

THÈSE

présentée par

**Catherine ORIAT**

pour obtenir le grade de DOCTEUR

de l'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

(Arrêté ministériel du 30 mars 1992)

Spécialité : Informatique

---

Étude des spécifications modulaires : constructions  
de colimites finies, diagrammes, isomorphismes

---

Thèse soutenue le 16 janvier 1996

Composition du jury :

Président : M. Jean-Pierre VERJUS

Rapporteurs : M. Michel BIDOIT

M. Christian LAIR

Examineurs : M. Michel SINTZOFF

M. Didier BERT

Laboratoire Logiciels Systèmes et Réseaux (LSR-IMAG)



## Remerciements

Je remercie tout d'abord mon directeur de thèse, M. Didier Bert, pour son soutien et pour la confiance qu'il m'a accordée dans la préparation de cette thèse.

Je remercie M. Jean-Pierre Verjus qui a accepté de présider ce jury.

Je remercie M. Michel Bidoit pour l'intérêt qu'il a porté à mon travail et pour avoir accepté d'être rapporteur de ma thèse.

Je remercie M. Christian Lair pour les améliorations qu'il m'a suggérées et pour les erreurs qu'il m'a signalées avec beaucoup d'indulgence et de gentillesse. Je suis particulièrement honorée qu'un catégoricien de sa compétence ait accepté de faire partie de mon jury en qualité de rapporteur.

Je remercie M. Michel Sintzoff pour m'avoir fait l'honneur de participer à ce jury.

Ce travail doit beaucoup à M. Jean-Claude Reynaud. Je le remercie d'une part pour le temps qu'il a consacré à suivre mon travail, et d'autre part pour toutes les suggestions qu'il m'a faites. Si, par hasard, ce travail contient quelques idées intéressantes, c'est probablement lui qui me les a soufflées (mais, en paraphrasant Mac Lane, je suis évidemment responsable des erreurs qui pourraient rester).

Je remercie enfin les membres de l'équipe SCOP, auxquels j'associe les occupants présents ou passés du bâtiment  $D$ , pour leur accueil et leur soutien tout au long de ce travail.



*“Le département [...] épousait l’opinion traditionnelle qui voulait que la recherche fût une occupation solitaire et érémitique, une mise à l’épreuve du caractère plus que de l’érudition, qu’un abus de contact humain pouvait vicier.”*

— *David Lodge, La chute du British Museum*



## Résumé

La composition de spécifications modulaires peut être modélisée, dans le formalisme des catégories, par des colimites de diagrammes. La somme amalgamée permet en particulier d’assembler deux spécifications en précisant les parties communes. Notre travail poursuit cette idée classique selon trois axes.

D’un point de vue syntaxique, nous définissons un langage pour représenter les spécifications modulaires construites à partir d’une catégorie de spécifications et de morphismes de spécifications de base. Ce langage est caractérisé formellement par une catégorie de termes finiment cocomplète.

D’un point de vue sémantique, nous proposons d’associer à tout terme un diagramme. Cette interprétation permet de faire abstraction de certains choix effectués lors de la construction de la spécification modulaire. Pour cela, nous définissons une catégorie de diagrammes “concrète”, c’est-à-dire dont les flèches peuvent être manipulées effectivement. En considérant le quotient par une certaine congruence, nous obtenons une complétion de la catégorie de base par colimites finies. Nous montrons que le calcul du diagramme associé à un terme définit une équivalence entre la catégorie des termes et la catégorie des diagrammes, ce qui prouve la correction de cette interprétation.

Enfin, nous proposons un algorithme pour décider si deux diagrammes sont isomorphes, dans le cas particulier où la catégorie de base est finie et sans cycle. Cela permet de détecter des isomorphismes “de construction” entre spécifications modulaires, c’est-à-dire des isomorphismes qui ne dépendent pas des spécifications de base, mais seulement de la manière dont celles-ci sont assemblées.

## Abstract

The composition of modular specifications can be modeled, in a category theoretic framework, by means of colimits of diagrams. Pushouts in particular allow us to gather two specifications sharing a common part. Our work extends this classic idea along three lines.

From a syntactic point of view, we define a language to represent modular specifications built from a category of base specifications and base specification morphisms. This language is formally characterized by a finitely cocomplete category of terms.

From a semantic point of view, we propose to associate with each term a diagram. This interpretation allows us to abstract some choices made while constructing a modular specification. We thus define a “concrete” category of diagrams, in which arrows can actually be handled. Considering the quotient by a certain congruence relation, we get a completion of the base category with finite colimits. We prove that this calculus defines an equivalence between the category of terms and the category of diagrams, which shows the soundness of this interpretation.

At last, we propose an algorithm to decide whether two diagrams are isomorphic, when the base category is finite and cycle free. This allows us to detect “construction isomorphisms” between modular specifications, i.e. isomorphisms which do not depend on the base specifications, but only on their combination.



# Table des matières

<b>Résumé</b>	<b>7</b>
<b>Introduction</b>	<b>13</b>
1  Spécifications algébriques . . . . .	13
2  Théorie des catégories . . . . .	14
3  Modularité . . . . .	15
4  Notre travail . . . . .	16
5  Comparaisons avec d'autres travaux . . . . .	18
6  Plan de ce mémoire . . . . .	20
<b>1 Motivation</b>	<b>23</b>
1.1 Préliminaires . . . . .	23
1.2 Spécifications . . . . .	29
1.3 Composition de spécifications en LPG . . . . .	33
1.4 Sommes amalgamées . . . . .	35
1.4.1 Somme amalgamée de deux ensembles . . . . .	36
1.4.2 Somme amalgamée de deux signatures . . . . .	39
1.4.3 Somme amalgamée de deux spécifications . . . . .	40
1.4.4 Syntaxe abstraite pour les constructions modulaires . . . . .	42
1.4.5 Somme disjointe de deux spécifications . . . . .	43
1.5 Spécifications modulaires d'anneaux . . . . .	44
1.6 Isomorphisme de construction . . . . .	50
1.7 Cadre général . . . . .	51
1.7.1 Institutions . . . . .	51
1.7.2 Terminologie et hypothèses . . . . .	53
<b>2 Graphes, catégories et diagrammes</b>	<b>55</b>
2.1 Graphes . . . . .	55
2.2 Catégories, foncteurs . . . . .	57
2.2.1 Catégories . . . . .	58
2.2.2 Foncteurs . . . . .	59
2.2.3 Transformations naturelles . . . . .	60
2.2.4 Adjonctions . . . . .	63
2.3 Diagrammes . . . . .	65
2.3.1 Catégorie des diagrammes $\text{DIAGR}(\mathcal{C})$ . . . . .	65
2.3.2 Colimites . . . . .	72

2.3.3	Exemples de colimites . . . . .	76
2.3.4	Adjonction $(\text{Colim}_{\mathcal{C}} \dashv I_{\mathcal{C}})$ . . . . .	79
2.3.5	Conservation de colimites . . . . .	81
2.3.6	Aplatissement . . . . .	87
2.3.7	Catégorie $\text{diagr}(\mathcal{C})$ . . . . .	93
2.3.8	La catégorie $\text{diagr}(\mathcal{C})$ est finiment cocomplète . . . . .	103
2.3.9	La catégorie $\text{diagr}(\mathcal{C})$ est une complétion de $\mathcal{C}$ par colimites finies . . . . .	109
2.4	Conclusion . . . . .	115
<b>3</b>	<b>Syntaxe : catégorie des termes</b> . . . . .	<b>117</b>
3.1	Précatégories, préfoncteurs et pré-colimites . . . . .	119
3.2	Problème de circularité termes – congruences . . . . .	124
3.3	Précatégories $\mathcal{C}_i$ . . . . .	125
3.4	Simplification des flèches . . . . .	130
3.5	Extensions conservatrices . . . . .	135
3.6	Précatégorie $\text{TERME}(\mathcal{C}_0)$ . . . . .	137
3.7	Catégorie $\text{Terme}(\mathcal{C}_0)$ . . . . .	140
3.8	Catégorie libre $\mathcal{L}(\mathcal{C}_0)$ . . . . .	142
3.9	Conclusion . . . . .	149
<b>4</b>	<b>Sémantique : des termes aux diagrammes</b> . . . . .	<b>151</b>
4.1	Sémantique . . . . .	151
4.1.1	Précatégorie $\text{DIAGR}(\mathcal{C}_0)$ . . . . .	151
4.1.2	Préfoncteur $\mathcal{D} : \text{TERME}(\mathcal{C}_0) \rightarrow \text{DIAGR}(\mathcal{C}_0)$ . . . . .	154
4.2	Exemple des anneaux . . . . .	155
4.2.1	Diagramme associé à la spécification $A_2$ . . . . .	155
4.2.2	Diagramme associé à la spécification $A'_2$ . . . . .	157
4.2.3	Diagramme associé à la spécification $A_3$ . . . . .	158
4.2.4	Diagramme associé à la spécification $A_4$ . . . . .	160
4.3	Équivalence entre les catégories $\text{Terme}(\mathcal{C}_0)$ et $\text{diagr}(\mathcal{C}_0)$ . . . . .	164
4.3.1	Équivalence entre catégories, équivalence entre précatégories . . . . .	164
4.3.2	Équivalence entre les précatégories $\text{DIAGR}(\mathcal{C}_0)$ et $\text{TERME}(\mathcal{C}_0)$ . . . . .	165
4.3.3	Correction du calcul de diagrammes . . . . .	170
4.4	Conclusion . . . . .	171
<b>5</b>	<b>Isomorphismes entre diagrammes</b> . . . . .	<b>173</b>
5.1	Transformations de diagrammes . . . . .	173
5.1.1	Substitution par objet isomorphe . . . . .	174
5.1.2	Suppression d'un arc . . . . .	176
5.1.3	Ajout d'un arc étiqueté . . . . .	180
5.1.4	Contraction d'un arc identité . . . . .	183
5.2	Catégorie $\mathcal{C}_0$ finie et sans cycle . . . . .	186
5.2.1	Hypothèses . . . . .	186
5.2.2	Fonction de complétion . . . . .	187
5.2.3	Zigzags élémentaires et liens . . . . .	190

---

5.2.4	Forme minimale . . . . .	194
5.2.5	Algorithme . . . . .	196
5.3	Cas général . . . . .	196
5.3.1	Zigzags . . . . .	197
5.3.2	Catégorie de base comportant des cycles . . . . .	198
5.3.3	Catégorie de base infinie . . . . .	198
<b>Conclusion</b>		<b>201</b>
1	Bilan . . . . .	201
2	Perspectives . . . . .	203
<b>A Preuves sur la syntaxe</b>		<b>207</b>
A.1	Preuve du théorème de forme normale (2) . . . . .	207
A.2	Preuve du théorème de forme normale (3) . . . . .	208
A.3	Preuve du théorème d'“unicité” de la forme normale . . . . .	209
A.4	Preuve du théorème 3.9 . . . . .	217
<b>Références bibliographiques</b>		<b>225</b>
<b>Index</b>		<b>231</b>



# Introduction

*Spécifier* un problème consiste à le décrire sans décider prématurément de la façon dont il sera résolu. En informatique, une spécification de programme est une description des fonctionnalités attendues du programme indépendante des choix de mise en œuvre.

Il existe plusieurs manières de spécifier un programme. On peut utiliser une langue naturelle, par exemple en écrivant un cahier des charges. Certaines activités, comme la validation de programme, la construction automatique d'un prototype, ou la production systématique de jeux de tests, nécessitent des spécifications *formelles*, qui possèdent une sémantique précise, en particulier non ambiguë.

Nous nous intéressons ici aux *spécifications algébriques*, qui font partie des spécifications *orientées propriétés*<sup>1</sup>. Les spécifications orientées propriétés sont basées sur un *système logique*, qui définit l'ensemble des propriétés que doit satisfaire tout modèle de la spécification.

## 1 Spécifications algébriques

Une spécification algébrique est composée d'une signature et d'un ensemble d'axiomes. La signature contient les symboles utilisés pour former des expressions, et les axiomes définissent certaines propriétés de ces expressions. La logique sous-jacente décrit l'ensemble des *théorèmes* que l'on peut déduire des axiomes. De plus, la *sémantique* des spécifications algébriques associe à toute spécification une classe de *modèles*, c'est-à-dire une classe d'algèbres multi-sortes qui satisfont les axiomes.

Historiquement, les spécifications algébriques sont issues de deux courants distincts : l'algèbre universelle en mathématiques [Coh65] et les types abstraits en génie logiciel. L'origine des types abstraits figure dans le concept de *classe* du langage de programmation Simula [DN66, BDMN73]. Un type abstrait, composé d'un ensemble de *domaines de valeurs* accompagné d'un ensemble d'*opérations* sur ces domaines de valeurs, permet de structurer les données. L'idée de base en spécification algébrique consiste à modéliser un type abstrait par une algèbre. Cette idée est motivée par l'analogie entre un type abstrait et une algèbre multi-sortes sur une signature. L'interprétation des sortes de la signature correspond en effet aux domaines de valeurs, et l'interprétation des opérateurs de la signature correspond aux opérations sur ces domaines de valeurs.

Les premiers travaux sur les spécifications algébriques datent du milieu des années 1970, avec les travaux de B. Liskov et S. Zilles [LZ74], de J. Guttag [Gut75,

---

<sup>1</sup>En anglais, *property-oriented*.

Gut77] et du groupe ADJ (composé de J. Goguen, J. Thatcher, E. Wagner et J. Wright) [GTWW75, GTWW77, GTW78]. Au départ, les spécifications algébriques étaient basées sur la logique équationnelle. Puis ont été introduites d'autres logiques, comme les clauses de Horn, la logique du premier ordre ou encore les spécifications avec contraintes [EM90]. D'autres extensions ont permis de prendre en compte la notion d'erreur [BBC86, BL93]. J. Goguen et R. Burstall ont proposé la *théorie des institutions*, qui offre un cadre général pour étudier les différents formalismes de spécification algébrique [GB84, GB90]. Il existe beaucoup de références sur les spécifications algébriques, parmi les bonnes synthèses, citons [EM85, MG85, Wir90].

Ces travaux ont donné naissance à de nombreux langages de spécification algébrique, comme Clear [BG77, BG80], ACT ONE et ACT TWO [EM85, EM90], ASL [SW83, Wir86], OBJ2 et OBJ3 [FGJM85, GKK<sup>+</sup>87], PLUSS [Gau84, Bid89], LPG [Ber83, BE86, B<sup>+</sup>90], GLIDER [Huf92]. On trouve un bon panorama des différents langages de spécification algébrique existants dans [Wir94].

## 2 Théorie des catégories

La théorie des catégories repose sur deux notions : l'*objet* et la *flèche*, contrairement à la théorie des ensembles qui repose sur le seul concept d'*ensemble*. En théorie des ensembles, un ensemble est caractérisé de façon *interne* par ses *éléments*, alors qu'en théorie des catégories, un *objet* est caractérisé de façon *externe* par les relations entretenues par l'intermédiaire des flèches avec les autres objets. Pour cette raison, un grand nombre de concepts de théorie des catégories sont définis à un *isomorphisme près*.

La théorie des catégories a été inventée au début des années 1940 par S. Mac Lane et S. Eilenberg. Bien que certains mathématiciens aient douté de l'intérêt des catégories, ce formalisme a joué un rôle unificateur en mathématiques, en particulier en algèbre et en topologie. Cette théorie définit les différents concepts de façon générale et abstraite, indépendamment de tout modèle. (C'est pour cette raison que ses détracteurs l'on surnommée "non-sens abstrait<sup>2</sup>".) La théorie des catégories a commencé à intéresser les informaticiens au début des années 1970. Par exemple, la découverte de l'équivalence entre le  $\lambda$ -calcul typé et les catégories cartésiennes fermées a permis de donner des solutions élégantes au problème des modèles du  $\lambda$ -calcul, lié à celui de la résolution des équations de domaines [SP77, Wan79, LS86]. Il existe de façon générale des liens très étroits entre la théorie des types et la théorie des catégories (cf. par exemple [See84, Poi92]).

L'ouvrage de référence sur les catégories est le livre de S. Mac Lane [McL71], plutôt destiné aux mathématiciens. Pour les informaticiens, le livre de M. Barr et C. Wells [BW90] est plus accessible, car les différents concepts sont présentés de façon assez élémentaire et les exemples sont tirés de l'informatique. Il existe beaucoup d'autres ouvrages qui traitent des catégories, par exemple [MB70, AM75, Gol79, AL91].

Le formalisme des spécifications algébriques s'appuie largement sur la théorie

---

<sup>2</sup>En anglais, *abstract nonsense*.

des catégories. Historiquement, le développement des spécifications algébriques (en particulier les travaux du groupe ADJ) a été influencé par les théories algébriques de Lawvere. Les théories algébriques de Lawvere permettent de modéliser les catégories d'algèbres en faisant abstraction de la syntaxe, c'est-à-dire indépendamment de la notion de signature [Law63, BW85, WBT85].

D'autre part, d'un point de vue pratique, on peut justifier l'utilisation des catégories par l'importance des *morphismes de spécifications*, qui jouent un rôle essentiel en spécification algébrique. Intuitivement, un morphisme de spécifications exprime une relation entre deux spécifications. Pour prendre un exemple mathématique, tout anneau peut être considéré comme un groupe, ce qui signifie qu'il existe un morphisme de spécifications entre la spécification des groupes et la spécification des anneaux. Il peut exister plusieurs morphismes de spécifications entre deux spécifications. Par exemple, on peut considérer tout anneau comme un monoïde *de deux façons différentes*, en considérant soit l'opérateur additif, soit l'opérateur multiplicatif de l'anneau comme l'opérateur du monoïde. Préciser quel opérateur de l'anneau sera l'opérateur du monoïde consiste exactement à définir un morphisme de spécifications entre la spécification des monoïdes et la spécification des anneaux.

### 3 Modularité

Le développement de spécifications de grande taille nécessite un découpage des spécifications en plusieurs spécifications plus simples, appelées *modules*. Chaque module correspond à un problème plus simple à résoudre. Cette méthode, qui consiste à "diviser pour mieux régner" est classique en génie logiciel [LCW85]. De façon générale, la modularité permet d'une part le développement indépendant des différents modules, et d'autre part une meilleure compréhension de chaque module, ce qui facilite la maintenance.

Cette vision de la modularité correspond à une approche descendante de résolution de problème, puisque l'accent est mis sur la *décomposition* du problème en plusieurs sous-problèmes plus simples. Dans notre travail, notre conception de la modularité est au contraire ascendante. Nous nous intéressons en effet à la *composition* des modules, c'est-à-dire à l'assemblage de modules élémentaires permettant de construire des modules plus complexes. Ces modules élémentaires sont regroupés dans une bibliothèque, la bibliothèque des *modules de base*. L'intérêt de la composition est de permettre la *réutilisation* des modules de base, préconisée en génie logiciel pour réduire les erreurs et les coûts de développement (cf. par exemple [Ber90]).

Un aspect de la modularité est la possibilité de définir des spécifications *génériques*, c'est-à-dire paramétrées [SST92]. En effet, une spécification paramétrée peut être instanciée de différentes manières par d'autres spécifications, ce qui évite de réécrire une nouvelle spécification pour chaque variation d'un même problème. Le but est alors de définir des spécifications génériques suffisamment générales pour permettre de spécifier le maximum de problèmes par une simple instantiation [Mar95].

Le développement modulaire des spécifications est assisté dans les langages de spécification algébrique par des *opérateurs de construction*<sup>3</sup> comme le *renommage*,

<sup>3</sup>En anglais, *specification-building operations*.

l'*enrichissement* ou l'*abstraction* [ST88, Wir90]. Ces primitives permettent de construire de nouvelles spécifications à partir de spécifications déjà définies. La composition de modules est un cas particulier d'opérateur de construction de spécifications, qui peut être modélisé par des *colimites de diagrammes*. Intuitivement, en théorie des catégories, un *diagramme* permet de décrire un assemblage de plusieurs objets en précisant des *partages* entre certains objets à l'aide de flèches. La *colimite* du diagramme permet de considérer le résultat de cette composition. Cette utilisation générale des colimites pour modéliser l'interconnexion de systèmes a été proposée par J. Goguen [Gog73, Gog92].

Un cas particulier de colimite est la *somme amalgamée*<sup>4</sup>, qui correspond à l'assemblage de deux objets  $B$  et  $C$ , en spécifiant une partie commune entre ces deux objets par un *objet partagé*  $A$  et deux flèches  $f : A \rightarrow B$  et  $g : A \rightarrow C$ . En spécification algébrique, la somme amalgamée peut modéliser d'une part la *composition* de deux spécifications qui ont une partie commune, et d'autre part l'*instanciation* d'une spécification générique.

## 4 Notre travail

Nous étudions la composition des spécifications algébriques modulaires. Nous supposons que nous disposons d'une bibliothèque de spécifications et de morphismes de spécifications *de base*, qui forment une catégorie *de base*, en général appelée  $\mathcal{C}_0$  dans ce mémoire. Les spécifications de base sont monolithiques, c'est-à-dire non décomposées. Notre travail repose sur une idée classique en spécification algébrique : la composition des spécifications peut être modélisée par des colimites de diagrammes finis. Nous proposons donc de considérer une *spécification modulaire* comme une spécification obtenue à partir de spécifications et morphismes de spécifications de base par une suite de constructions de colimites.

Notre travail est indépendant de la logique sous-jacente du langage de spécification algébrique. Nous pouvons donc nous placer dans le cadre général des *institutions*. Nous devons uniquement supposer que la catégorie des spécifications est *finiment cocomplète*, c'est-à-dire que tout diagramme fini a une colimite.

D'un point de vue syntaxique, c'est-à-dire en ce qui concerne le *langage* utilisé pour décrire ces constructions, nous ne considérons pas toutes les colimites finies, mais uniquement les sommes amalgamées. En effet, avec la spécification vide, objet initial de la catégorie des spécifications, les sommes amalgamées permettent de simuler la construction de toute colimite finie. Nous proposons la construction d'une catégorie de *termes*, appelée  $\text{Terme}(\mathcal{C}_0)$ , pour représenter la composition de spécifications modulaires en utilisant les sommes amalgamées.

En théorie des catégories, on travaille le plus souvent à un isomorphisme près. C'est une conséquence de la caractérisation externe des objets dans ce formalisme. Les colimites, par exemple, sont définies au départ à un isomorphisme près. La définition d'une syntaxe est au contraire un problème typiquement informatique, qui nécessite de faire des *choix de représentation*. C'est la raison pour laquelle nous devons utiliser des *sommes amalgamées choisies* au moment de la définition

<sup>4</sup>En anglais, *pushout*. Les catégorieiciens utilisent également le terme de *coproduit fibré*.

du langage d'expression pour les spécifications modulaires. Ces choix de sommes amalgamées correspondent à des choix de codage.

Un terme correspond donc à une spécification modulaire construite par une succession de sommes amalgamées. Dans une seconde étape, nous proposons d'interpréter un terme par un *diagramme* dont la colimite est la spécification modulaire dénotées par ce terme. Nous considérons les diagrammes comme une sémantique pour les spécifications modulaires, par opposition aux termes qui constituent la syntaxe.

L'interprétation des termes par des diagrammes nécessite de définir une *catégorie* de diagrammes puisqu'il faut non seulement associer à toute spécification modulaire un diagramme, mais aussi associer à tout morphisme de spécifications modulaires un morphisme de diagrammes. La définition d'un diagramme est relativement standard en théorie des catégories, bien qu'il existe quelques nuances. La définition de morphisme de diagrammes est moins connue, en particulier dans la littérature informatique où sont utilisées des définitions moins générales que celle dont nous nous servons ici.

Nous présentons deux catégories de diagrammes,  $\text{DIAGR}(\mathcal{C}_0)$  et  $\text{diagr}(\mathcal{C}_0)$ . Les objets de  $\text{DIAGR}(\mathcal{C}_0)$  sont des *diagrammes*, et les flèches de  $\text{DIAGR}(\mathcal{C}_0)$  des *morphismes de diagrammes*. La catégorie  $\text{diagr}(\mathcal{C}_0)$  est obtenue à partir de  $\text{DIAGR}(\mathcal{C}_0)$  par un quotient par une relation de congruence sur les flèches de  $\text{DIAGR}(\mathcal{C}_0)$ . C'est la catégorie  $\text{diagr}(\mathcal{C}_0)$  qui possède les propriétés théoriques intéressantes. Il s'agit en effet d'une catégorie *finiment cocomplète*, c'est-à-dire que tout diagramme fini sur  $\text{diagr}(\mathcal{C}_0)$  a une colimite finie dans  $\text{diagr}(\mathcal{C}_0)$ . D'autre part,  $\text{diagr}(\mathcal{C}_0)$  est une complétion par colimites finies de la catégorie  $\mathcal{C}_0$ . Sur le plan pratique, comme les flèches de  $\text{diagr}(\mathcal{C}_0)$  sont des classes d'équivalence de morphismes de diagrammes, on travaille en réalité avec des représentants, c'est-à-dire avec des flèches de  $\text{DIAGR}(\mathcal{C}_0)$ . La catégorie  $\text{DIAGR}(\mathcal{C}_0)$  permet donc la manipulation effective des flèches de  $\text{diagr}(\mathcal{C}_0)$ .

La représentation des termes par des diagrammes est définie par un foncteur

$$\mathcal{D} : \text{Terme}(\mathcal{C}_0) \rightarrow \text{diagr}(\mathcal{C}_0)$$

entre les catégories  $\text{Terme}(\mathcal{C}_0)$  et  $\text{diagr}(\mathcal{C}_0)$ . Nous montrons que ce foncteur définit une *équivalence de catégories* entre  $\text{Terme}(\mathcal{C}_0)$  et  $\text{diagr}(\mathcal{C}_0)$ , ce qui signifie que bien qu'elles ne soient pas isomorphes, ces catégories ont néanmoins essentiellement la même structure.

Après avoir représenté les spécifications modulaires par des diagrammes, nous nous intéressons au problème de l'*équivalence* entre deux spécifications modulaires. Deux spécifications modulaires sont considérées comme équivalentes lorsque celles-ci sont *isomorphes* en tant que *colimites de diagramme*. Détecter cet isomorphisme consiste en fait à détecter un isomorphisme entre deux diagrammes dans la catégorie  $\text{diagr}(\mathcal{C}_0)$ . Nous appelons cet isomorphisme un *isomorphisme de construction*, parce qu'il ne dépend pas de la définition effective des spécifications de base, mais seulement de la manière dont celles-ci sont combinées pour construire les spécifications modulaires. Dans l'hypothèse où la catégorie  $\mathcal{C}_0$  est finie et ne comporte pas de cycle, nous proposons un algorithme pour détecter si deux diagrammes sont isomorphes.

Des résultats partiels concernant l'interprétation des termes par des diagrammes et la détection d'isomorphismes de construction ont été présentés dans [Ori94, Ori95].

## 5 Comparaisons avec d'autres travaux

### Catégories de diagrammes

La définition de *diagramme* est, à quelques nuances près, standard. Chez S. Mac Lane [McL71], un diagramme sur une catégorie  $\mathcal{C}$  est un *foncteur* d'une catégorie  $J$  vers  $\mathcal{C}$ . La définition de M. Barr et C. Wells [BW90], est moins générale, puisqu'un diagramme est un *morphisme de graphes* d'un graphe vers le graphe sous-jacent de  $\mathcal{C}$ . Nous utilisons une définition "intermédiaire" : pour nous, un diagramme est un *foncteur* d'une catégorie *librement engendrée sur un graphe* vers  $\mathcal{C}$ . Partir d'un graphe nous permet de rester proche de l'informatique. Par contre, nous ne souhaitons pas nous restreindre à une *petite* catégorie  $\mathcal{C}$ .

La définition de *morphisme de diagrammes* est beaucoup moins standard. Par exemple dans Clear, un morphisme de diagrammes, appelé *morphisme basé*, correspond uniquement à une *inclusion* entre deux diagrammes.

La définition proposée par A. Tarlecki, R. Burstall et J. Goguen dans [TBG91], bien que plus générale que celle utilisée dans la sémantique du langage Clear, n'est pas encore suffisamment générale pour notre travail. En effet, cette définition ne permet pas de représenter tout morphisme de spécifications modulaires par un morphisme de diagrammes.

La définition de morphisme de diagrammes que nous utilisons n'est pas nouvelle en théorie des catégories. Une version duale de la catégorie  $\text{diag}(\mathcal{C}_0)$  est par exemple utilisée par M. Barr et C. Wells dans [BW94]. Néanmoins, la reformulation de la définition de catégorie de diagrammes que nous proposons dans le chapitre 2 présente un intérêt en informatique, d'une part parce que la catégorie  $\text{DIAGR}(\mathcal{C}_0)$  permet de manipuler effectivement les morphismes de diagrammes, et d'autre part parce que les catégories  $\text{DIAGR}(\mathcal{C}_0)$  et  $\text{diag}(\mathcal{C}_0)$  sont peu connues dans cette discipline.

### Calculs de colimites

R. Burstall et D. Rydeheard proposent des algorithmes pour *calculer* certains concepts de théorie des catégories [Bur80, BR86], en particulier les colimites de diagrammes. Le calcul général de colimite est paramétré par certaines colimites, en l'occurrence un objet initial, des sommes et des coégalisateurs, qui permettent d'évaluer la colimite d'un diagramme quelconque. Ainsi, ayant défini l'objet initial, la somme et le coégalisateur par exemple dans la catégorie des ensembles **Set**, on en déduit un moyen de calculer la colimite d'un diagramme quelconque dans **Set**. Cet algorithme est en fait basé sur une preuve du résultat suivant : si une catégorie a un objet initial, des sommes et des coégalisateurs, alors celle-ci est finiment cocomplète. Notre travail est basé sur un théorème similaire : si une catégorie a un objet initial et des sommes amalgamées alors celle-ci est finiment cocomplète. Notre travail est différent, dans la mesure où nous ne cherchons pas à calculer des colimites dans une catégorie fixée. En particulier, nous nous intéressons au problème général de l'isomorphisme entre deux diagrammes, qui correspond à un isomorphisme entre

leurs colimites respectives. Cet isomorphisme est indépendant de la catégorie sur laquelle ces diagrammes sont construits.

### Utilisation des colimites en spécification algébrique

L'utilisation des colimites pour représenter l'assemblage de spécifications est loin d'être nouvelle. R. Burstall et J. Goguen ont présenté cette idée dans la sémantique du langage de spécification Clear [BG80]. Dans Clear, la composition de plusieurs spécifications qui partagent un environnement commun, appelées *théories basées*, est en effet formalisée par la colimite d'un diagramme.

Les colimites ont ensuite été assez intensivement utilisées en spécification algébrique. La somme amalgamée en particulier permet de modéliser l'instanciation de spécifications génériques [TWW82, EM85]. H. Ehrig, R. Jimenez et F. Orejas [EJO93] ont également présenté une forme plus générale d'instanciation à l'aide de *sommes amalgamées multiples*<sup>5</sup>, nouvel exemple de colimite.

### Syntaxe pour les constructions modulaires

J. Bergstra, J. Heering et R. Klint [BHK90] d'une part, et G. Renardel de Lavalette [Ren91] d'autre part, ont proposé de structurer les opérateurs de construction de spécifications sous forme d'*algèbres de modules*. Une algèbre de modules est un langage permettant d'écrire des *expressions* de composition des spécifications algébriques. Cette approche est voisine de notre travail sur le langage  $\text{Terme}(\mathcal{C}_0)$ , puisque le problème de la composition des spécifications algébriques est abordé d'un point de vue *syntaxique*, et permet de comparer différentes spécifications modulaires. Par contre, nous ne nous intéressons pas aux mêmes opérateurs de constructions de spécifications, puisque nous ne considérons que les constructions de colimites.

Notre syntaxe pour représenter les spécifications algébriques modulaires est largement inspirée des travaux de J.-C. Reynaud, qui a proposé un système de types pour représenter les constructions de colimites [Rey90a, Rey90b, Rey93]. Ce système est fondé sur les théories algébriques généralisées de Cartmell [Car86] qui permettent de spécifier des types dépendants, c'est-à-dire des types paramétrés par des termes. Les types dépendants ont été également utilisés par T. Streicher et M. Wirsing [SW91] pour modéliser la composition de spécifications algébriques modulaires. D'autre part, H. Ehrig, R. Jimenez et F. Orejas ont également proposé une syntaxe pour les spécifications algébriques modulaires dans [EJO93]. Cette syntaxe est en partie basée sur des constructions de sommes amalgamées.

La principale difficulté en ce qui concerne la définition d'une syntaxe pour les constructions de colimites est le problème de circularité entre la définition d'une part des objets et des flèches de  $\text{Terme}(\mathcal{C}_0)$  et d'autre part de l'égalité entre deux flèches de  $\text{Terme}(\mathcal{C}_0)$ . En effet, il existe une flèche, que nous appelons *up* dans ce mémoire, entre une somme amalgamée et un objet à condition qu'une égalité entre deux flèches de  $\text{Terme}(\mathcal{C}_0)$  soit vérifiée. La définition des flèches *up* dépend donc de la définition de l'égalité des flèches dans  $\text{Terme}(\mathcal{C}_0)$ , qui elle-même présuppose que les flèches ont été définies.

---

<sup>5</sup>En anglais, *multiple pushout*.

J.-C. Reynaud contourne le problème en proposant une définition *concrète*, c'est-à-dire sémantique, d'une catégorie librement engendrée par objet initial choisi et sommes amalgamées choisies [Rey93]. H. Ehrig, R. Jimenez et F. Orejas [EJO93] ne définissent pas de syntaxe pour les flèches  $\text{up}$ . La syntaxe qu'ils proposent ne permet donc pas d'obtenir une catégorie *finiment cocomplète*.

Une solution pour spécifier une catégorie finiment cocomplète sans utiliser d'égalités pour définir les flèches a été présentée par F. Cury dans [Cur91]. F. Cury propose de dédoubler les flèches  $\text{up}$  en deux flèches dont la définition ne dépend pas d'une égalité. Ces deux flèches, qui sont des exemples particuliers de flèches  $\text{up}$ , permettent de reconstruire *a posteriori* toutes les flèches  $\text{up}$ .

La solution que nous proposons consiste à stratifier la construction de  $\text{Terme}(\mathcal{C}_0)$  en une suite de catégories  $\mathcal{C}_i$ . L'existence d'une flèche  $\text{up}$  dans la catégorie  $\mathcal{C}_i$  dépend uniquement d'égalités dans la catégorie  $\mathcal{C}_{i-1}$ . Nous évitons ainsi le problème de circularité entre la définition des flèches et la définition des égalités.

Notre catégorie  $\text{Terme}(\mathcal{C}_0)$  peut être comparée au *type* d'une esquisse, introduit par C. Ehresmann [Ehr68]. Dans les travaux de D. Duval et J.-C. Reynaud [DR94a, DR94b], le type d'une esquisse est une *catégorie librement engendrée* à partir d'un graphe par *sommes* et *produits*. Par conséquent, la construction du type d'une esquisse est basée sur les catégories à sommes et produits alors que la construction de  $\text{Terme}(\mathcal{C}_0)$  est basée sur les catégories à colimites finies. D'autre part, une esquisse contient des cônes et cocônes distingués, qui spécifient certaines limites ou colimites. Par contre, dans la construction de  $\text{Terme}(\mathcal{C}_0)$  que nous proposons, il n'est pas possible de spécifier des sommes amalgamées distinguées dans la catégorie de base  $\mathcal{C}_0$ .

## 6 Plan de ce mémoire

Le plan de ce mémoire est le suivant. Le chapitre 1 introduit les spécifications algébriques modulaires, présente le problème de la composition des spécifications et pose le cadre général de notre travail. Après quelques rappels sur les spécifications algébriques, nous proposons plusieurs exemples de spécifications modulaires d'*anneaux*, en utilisant quelques spécifications et morphismes de spécifications de base. Cet exemple académique présente d'une part la syntaxe pour les constructions modulaires, et d'autre part le problème de l'isomorphisme de construction entre deux spécifications modulaires.

Le chapitre 2 contient les bases théoriques de notre travail. Nous effectuons quelques rappels de théorie des graphes en section 2.1, puis de théorie des catégories en section 2.2. Enfin, en section 2.3, nous présentons les notions de diagrammes, de colimites finies ainsi que les catégories de diagrammes  $\text{DIAGR}(\mathcal{C}_0)$  et  $\text{diagr}(\mathcal{C}_0)$ . Nous montrons que la catégorie  $\text{diagr}(\mathcal{C}_0)$  est finiment cocomplète, et que  $\text{diagr}(\mathcal{C}_0)$  est une complétion par colimites finies de  $\mathcal{C}_0$ .

Le chapitre 3 est consacré à la syntaxe des spécifications modulaires. Dans ce chapitre, nous proposons une construction stratifiée de la catégorie  $\text{Terme}(\mathcal{C}_0)$  des termes, qui fournit un langage pour représenter les spécifications et morphismes de spécifications modulaires. Nous montrons d'une part que  $\text{Terme}(\mathcal{C}_0)$  est une catégorie finiment cocomplète, et d'autre part que  $\text{Terme}(\mathcal{C}_0)$  est une extension conser-

vatrice de  $\mathcal{C}_0$ . Enfin, nous construisons, à partir de  $\text{Terme}(\mathcal{C}_0)$ , une catégorie  $\mathcal{L}(\mathcal{C}_0)$  librement engendrée par objet initial choisi et sommes amalgamées choisies sur  $\mathcal{C}_0$ .

Dans le chapitre 4, nous interprétons les termes dénotant des spécifications modulaires par des diagrammes. Cette interprétation permet d'obtenir une représentation plus abstraite des spécifications modulaires que celle donnée par les termes, parce que certaines étapes spécifiques de la construction sont éliminées. Nous montrons que l'interprétation des termes par les diagrammes définit une équivalence entre les catégories  $\text{Terme}(\mathcal{C}_0)$  et  $\text{diagr}(\mathcal{C}_0)$ . Ce résultat nous permet de déduire que toute spécification modulaire est isomorphe à la colimite du diagramme qui la représente.

Le chapitre 5 est une application de cette représentation des spécifications modulaires par des diagrammes. Nous proposons de détecter un isomorphisme de construction entre deux spécifications modulaires en détectant un isomorphisme entre leurs diagrammes correspondants. Nous présentons un algorithme qui permet, dans l'hypothèse où la catégorie de base  $\mathcal{C}_0$  est finie et ne comporte pas de cycle, de détecter si deux diagrammes sont isomorphes.



# Chapitre 1

## Motivation

Le but de ce chapitre est de présenter le problème de la composition des spécifications modulaires. Nous étudions différentes spécifications modulaires des anneaux, construites à partir de quelques spécifications de base. Cet exemple est décrit en utilisant le langage de spécification LPG (Langage pour la Programmation Générique) développé par D. Bert et R. Echahed [BE86, B<sup>+</sup>90]. Nous introduisons informellement une *syntaxe* pour représenter les spécifications modulaires, et expliquons — également de façon informelle — qu’une spécification modulaire peut être considérée comme la *colimite d’un diagramme*. Nous motivons ensuite l’intérêt des *isomorphismes de construction*. Enfin, nous posons le cadre général de notre travail.

### 1.1 Préliminaires

Dans cette partie, nous rappelons quelques notions élémentaires de spécification algébrique, en particulier les définitions de signature, morphisme de signatures, algèbre, morphisme d’algèbres, équation et algèbre satisfaisant une équation. Cette présentation, loin d’être exhaustive, est inspirée de [GB90] et de [EM85].

**Définition 1.1** (*S*-ensemble et *S*-application) Soit un ensemble  $S$ .

Un *S*-ensemble  $A$  est une famille indexée par  $S$  d’ensembles disjoints  $(A_s)_{s \in S}$ .

Une *S*-application  $f$  entre deux *S*-ensembles  $A$  et  $B$ , notée  $f : A \rightarrow B$ , est une famille indexée par  $S$  d’applications  $(f_s : A_s \rightarrow B_s)_{s \in S}$ .

**Définition 1.2** (Ensemble des chaînes finies) Soit un ensemble  $A$ . L’ensemble des *chaînes finies* sur  $A$  est l’ensemble

$$A^* = \bigcup_{n=0}^{\infty} A^n.$$

Notations :

- La chaîne vide est notée  $\varepsilon$ .
- La chaîne composée des éléments  $s_1, s_2, \dots, s_n$  est notée  $s_1 s_2 \dots s_n$ .

L'ensemble des chaînes finies non vides sur  $A$  est

$$A^+ = A^* \times A = \bigcup_{n=1}^{\infty} A^n.$$

Les opérations  $_+^+$  et  $_+^*$  s'étendent aux applications entre deux ensembles. Soit une application  $f : A \rightarrow B$ . On définit les applications  $f^* : A^* \rightarrow B^*$  et  $f^+ : A^+ \rightarrow B^+$  de la façon suivante.

$$\begin{aligned} f^*(\varepsilon) &= \varepsilon; \\ f^*(a_1 \dots a_n) &= f(a_1) \dots f(a_n) \quad (\text{pour } n \geq 1); \\ f^+(a_1 \dots a_n) &= f(a_1) \dots f(a_n) \quad (\text{pour } n \geq 1). \end{aligned}$$

On vérifie facilement que pour tout ensemble  $A$ , on a

$$\begin{aligned} (id_A)^* &= id_{A^*} \\ (id_A)^+ &= id_{A^+} \end{aligned}$$

et pour toutes applications  $f : A \rightarrow B$  et  $g : B \rightarrow C$ , on a

$$\begin{aligned} (g \circ f)^* &= g^* \circ f^* \\ (g \circ f)^+ &= g^+ \circ f^+. \end{aligned}$$

Ces deux propriétés signifient que  $_+^*$  et  $_+^+$  sont des *foncteurs*  $_+^* : \mathbf{Set} \rightarrow \mathbf{Set}$  et  $_+^+ : \mathbf{Set} \rightarrow \mathbf{Set}$  dans la catégorie des ensembles  $\mathbf{Set}$ .

Une signature comprend les symboles utilisés pour former des expressions. Plus précisément, une signature est composée d'un ensemble de *symboles de sortes*, qui représentent des types de données, et d'un ensemble de *symboles d'opérateurs*, qui représentent des opérations sur ces données. À chaque symbole d'opérateur est associé un *profil*, qui représente le type des données d'entrée et de sortie.

**Définition 1.3** (Signature multi-sortes) Une *signature multi-sortes*, ou plus simplement *signature*, est un triplet  $(S, \Omega, \varpi)$  où

- $S$  est un ensemble fini, dont les éléments sont appelés *sortes*;
- $\Omega$  est un ensemble fini, dont les éléments sont appelés *opérateurs*;
- $\varpi$  est une application  $\varpi : \Omega \rightarrow S^+$ , qui à chaque opérateur  $\omega$  de  $\Omega$  associe une chaîne non vide sur  $S$ , appelée *profil* de  $\omega$ .

Notation : si  $\varpi(\omega) = s_1 \dots s_n s$  ( $n \geq 0$ ), on note  $\omega : s_1, \dots, s_n \rightarrow s$ . Si  $n = 0$ , on dit que  $\omega$  est une *constante*, et on note  $\omega : \rightarrow s$ .

**Remarque 1.1** Une signature est souvent définie comme un ensemble de sortes  $S$  accompagné d'une famille indicée par  $S^* \times S$  d'ensembles d'opérateurs. Nous avons préféré considérer ici un unique ensemble aplati  $\Omega$  d'opérateurs, le profil étant décrit par l'application  $\varpi : \Omega \rightarrow S^+$ . Nous avons choisi cette définition parce qu'elle permet de définir facilement la somme amalgamée de deux signatures à partir de la définition de la somme amalgamée de deux ensembles (cf. section 1.4).

**Exemple 1.1** (Exemples de signatures)

- La signature la plus simple est la signature vide, notée  $\Sigma_{\emptyset}$ , telle que l'ensemble des sortes et l'ensemble des opérateurs sont vides.
- Un exemple classique de signature est la signature  $\Sigma_{Nat} = (S_{Nat}, \Omega_{Nat}, \varpi_{Nat})$  des entiers naturels où  $S_{Nat}$  et  $\Omega_{Nat}$  et  $\varpi_{Nat}$  sont définis de la façon suivante :

$$\begin{aligned} S_{Nat} &= \{nat\}; \\ \Omega_{Nat} &= \{0, succ, +\}; \\ \varpi_{Nat}(0) &= nat; \\ \varpi_{Nat}(succ) &= nat\ nat; \\ \varpi_{Nat}(+) &= nat\ nat\ nat. \end{aligned}$$

Dans le langage de spécification LPG, les noms de sortes sont introduits par le mot clé `sorts` et les opérateurs par le mot clé `operators`. La signature  $\Sigma_{Nat}$  est donc définie en LPG par :

```

sorts      nat
operators  0 : -> nat
           succ : nat -> nat
           + : nat, nat -> nat

```

- Voici un autre exemple de signature, la signature  $\Sigma_M$  des monoïdes. Dans cette signature, on a une seule sorte  $s$ , une constante  $1$  et un opérateur binaire  $*$ . En LPG, on peut définir la signature  $\Sigma_M$  de la façon suivante :

```

sorts      s
operators  1 : -> s
           * : s, s -> s

```

Un *morphisme de signatures* entre deux signatures  $\Sigma$  et  $\Sigma'$  associe à toute sorte de  $\Sigma$  une sorte de  $\Sigma'$  et à tout opérateur de  $\Sigma$  un opérateur de  $\Sigma'$ , en conservant les profils des opérateurs.

**Définition 1.4** (Morphisme de signatures) Soit deux signatures  $\Sigma = (S, \Omega, \varpi)$  et  $\Sigma' = (S', \Omega', \varpi')$ . Un morphisme de signatures  $m$  de  $\Sigma$  vers  $\Sigma'$ , noté  $m : \Sigma \rightarrow \Sigma'$  est un couple  $(m^S, m^\Omega)$  où

- $m^S$  est une application  $m^S : S \rightarrow S'$ ;
- $m^\Omega$  est une application  $m^\Omega : \Omega \rightarrow \Omega'$ ;

tel que  $\varpi' \circ m^\Omega = m^{S+} \circ \varpi$ .

Détaillons un peu la condition  $\varpi' \circ m^\Omega = m^{S+} \circ \varpi$ . Considérons un opérateur  $\omega : s_1, \dots, s_n \rightarrow s$  de  $\Omega$ .

$$\begin{aligned}
\varpi'(m^\Omega(\omega)) &= m^{S^+}(\varpi(\omega)) \\
&= m^{S^+}(s_1 \dots s_n s) \\
&= m^S(s_1) \dots m^S(s_n) m^S(s)
\end{aligned}$$

L'opérateur  $m^\Omega(\omega)$  de  $\Omega'$  a donc pour profil  $m^\Omega(\omega) : m^S(s_1), \dots, m^S(s_n) \rightarrow m^S(s)$ .

On peut composer les morphismes de signatures. Soit trois signatures

$$\begin{aligned}
\Sigma &= (S, \Omega, \varpi) \\
\Sigma' &= (S', \Omega', \varpi') \\
\Sigma'' &= (S'', \Omega'', \varpi'')
\end{aligned}$$

et deux morphismes de signatures

$$\begin{aligned}
m &= (m^S, m^\Omega) : \Sigma \rightarrow \Sigma' \\
n &= (n^S, n^\Omega) : \Sigma' \rightarrow \Sigma''.
\end{aligned}$$

Pour composer  $m$  et  $n$ , il suffit de composer les applications sur les sortes et les applications sur les opérateurs. Autrement dit,

$$n \circ m = (n^S \circ m^S, n^\Omega \circ m^\Omega).$$

Cela définit bien un morphisme de signatures car

$$\begin{aligned}
&\varpi'' \circ n^\Omega \circ m^\Omega \\
&= n^{S^+} \circ \varpi' \circ m^\Omega \quad (n : \Sigma' \rightarrow \Sigma'' \text{ morphisme de signatures}) \\
&= n^{S^+} \circ m^{S^+} \circ \varpi \quad (m : \Sigma \rightarrow \Sigma' \text{ morphisme de signatures}) \\
&= (n^S \circ m^S)^+ \circ \varpi \quad (-^+ \text{ est un foncteur})
\end{aligned}$$

On a une catégorie des signatures, notée **Sig**, dont les objets sont les signatures, et les flèches sont les morphismes de signatures.

**Exemple 1.2** (Exemples de morphismes de signatures)

- Pour toute signature  $\Sigma = (S, \Omega, \varpi)$ , il existe un morphisme de signatures de la signature vide  $\Sigma_\emptyset$  vers  $\Sigma$ . Ce morphisme, noté  $j_\Sigma = (j_\Sigma^S, j_\Sigma^\Omega)$ , est constitué de l'application vide de l'ensemble vide des sortes vers  $S$ , et de l'application vide de l'ensemble vide des opérateurs vers  $\Omega$ . De plus, ce morphisme est unique. Dans la catégorie des signatures **Sig**,  $\Sigma_\emptyset$  est un *objet initial* puisqu'il existe une unique flèche de  $\Sigma_\emptyset$  vers tout autre objet de **Sig**.
- On a un morphisme de signatures de la signature des monoïdes vers la signature des entiers naturels  $m : \Sigma_M \rightarrow \Sigma_{Nat}$ , qui associe la sorte  $s$  de  $\Sigma_M$  à la sorte  $nat$  de  $\Sigma_{Nat}$ , la constante 1 à la constante 0, et l'opérateur binaire  $*$  à l'opérateur binaire  $+$ .

$$\begin{aligned}
m^S(s) &= nat; \\
m^\Omega(1) &= 0; \\
m^\Omega(*) &= +.
\end{aligned}$$

Soit une signature  $\Sigma$ . Définir une *algèbre sur  $\Sigma$* , ou  *$\Sigma$ -algèbre*, consiste à interpréter les sortes de  $\Sigma$  par des ensembles, et les opérateurs de  $\Sigma$  par des applications entre ces ensembles. Une  $\Sigma$ -algèbre est également appelée *modèle* ou *interprétation* de la signature  $\Sigma$ .

**Définition 1.5** ( $\Sigma$ -algèbre) Soit une signature  $\Sigma = (S, \Omega, \varpi)$ . Une  $\Sigma$ -algèbre  $A$  est définie par :

- un  $S$ -ensemble  $A$  ;
- pour tout opérateur  $\omega : s_1, s_2, \dots, s_n \rightarrow s \in \Omega$ , une *application*

$$\omega^A : A_{s_1} \times A_{s_2} \times \dots \times A_{s_n} \rightarrow A_s.$$

Si  $\omega$  est une constante  $\omega : \rightarrow s$ ,  $\omega^A$  est un élément de  $A_s$ .

**Définition 1.6** (Morphisme d'algèbres) Soit une signature  $\Sigma = (S, \Omega, \varpi)$  et deux algèbres  $A$  et  $B$ . Un *morphisme d'algèbres*  $h$  de  $A$  vers  $B$ , noté  $h : A \rightarrow B$  est une  $S$ -application  $(h_s : A_s \rightarrow B_s)_{s \in S}$  telle que pour tout opérateur  $\omega : s_1, s_2, \dots, s_n \rightarrow s$  de  $\Omega$ , pour tout  $n$ -uplet  $(a_1, a_2, \dots, a_n)$  de  $A_{s_1} \times A_{s_2} \times \dots \times A_{s_n}$ ,

$$h_s(\omega^A(a_1, a_2, \dots, a_n)) = \omega^B(h_{s_1}(a_1), h_{s_2}(a_2), \dots, h_{s_n}(a_n)).$$

Étant donné une signature  $\Sigma$ , on a une catégorie de  $\Sigma$ -algèbres, notée  $\text{Alg}(\Sigma)$ , dont les objets sont les  $\Sigma$ -algèbres et les flèches sont les morphismes de  $\Sigma$ -algèbres.

Une algèbre importante est l'algèbre des termes, qui permet de former des expressions à partir des symboles de la signature. Nous devons d'abord définir la notion de *variable* sur une signature.

**Définition 1.7** (Ensemble de variables sur une signature)

Soit une signature  $\Sigma = (S, \Omega, \varpi)$ . Un ensemble  $X$  de *variables sur la signature  $\Sigma$*  est un  $S$ -ensemble fini, tel que pour tout élément  $s \in S$ ,  $X_s$  est distinct de l'ensemble des opérateurs  $\Omega$ .

**Définition 1.8** (Algèbre des termes) Soit une signature  $\Sigma = (S, \Omega, \varpi)$  et un ensemble de variables  $X$  sur  $\Sigma$ . L'algèbre des termes  $T_\Sigma(X)$  est définie de la façon suivante.

- Pour toute sorte  $s \in S$ ,  $T_\Sigma(X)_s$  est le plus petit ensemble tel que :
  - $x \in X_s \Rightarrow x \in T_\Sigma(X)_s$  ;
  - $\omega : s_1, s_2, \dots, s_n \rightarrow s \in \Omega$ ,  $t_i \in T_\Sigma(X)_{s_i}$  ( $i = 1, \dots, n$ )  
 $\Rightarrow \omega(t_1, t_2, \dots, t_n) \in T_\Sigma(X)_s$ .

Les éléments de  $T_\Sigma(X)_s$  sont appelés *termes de sorte  $s$* .

- L'interprétation d'un opérateur  $\omega : s_1, s_2, \dots, s_n \rightarrow s \in \Omega$  est l'application
 
$$\omega^{T_\Sigma(X)} : \begin{array}{ccc} T_\Sigma(X)_{s_1} \times \dots \times T_\Sigma(X)_{s_n} & \rightarrow & T_\Sigma(X)_s \\ (t_1, \dots, t_n) & \mapsto & \omega(t_1, \dots, t_n). \end{array}$$

Lorsque l'ensemble des variables  $X$  est vide, on obtient l'algèbre des termes fermés  $T_\Sigma = T_\Sigma(\emptyset)$ .

Une *affectation* d'un ensemble de variables  $X$  dans une  $\Sigma$ -algèbre  $A$  associe à toute variable de  $X_s$  un élément de  $A_s$ . Autrement dit, une affectation  $aff$  de  $X$  dans  $A$  est une  $S$ -application  $aff : X \rightarrow A$ .

L'algèbre des termes a une propriété importante: c'est une algèbre *libre* dans la classe des  $\Sigma$ -algèbres. Cela signifie que toute affectation de  $X$  dans une algèbre  $A$  s'étend de façon unique en un morphisme d'algèbres de  $T_\Sigma(X)$  vers  $A$ .

**Théorème 1.1** (*L'algèbre  $T_\Sigma(X)$  est libre dans  $\text{Alg}(\Sigma)$* )

Notons  $i : X \rightarrow T_\Sigma(X)$  l'inclusion de  $X$  dans  $T_\Sigma(X)$ . Soit une  $\Sigma$ -algèbre  $A$ . Pour toute affectation  $aff : X \rightarrow A$ , il existe un unique morphisme d'algèbres  $\overline{aff} : T_\Sigma(X) \rightarrow A$  tel que  $\overline{aff} \circ i = aff$ .

Une équation est tout simplement composée d'un ensemble de variables et de deux termes.

**Définition 1.9** ( $\Sigma$ -équation) Soit une signature  $\Sigma = (S, \Omega, \varpi)$  et un ensemble de variables  $X$  sur  $\Sigma$ . Une  $\Sigma$ -équation sur  $\Sigma$  de sorte  $s$  est un triplet  $(X, t_1, t_2)$  où  $t_1$  et  $t_2$  sont des éléments de  $T_\Sigma(X)_s$ .

En LPG, une signature peut être accompagnée d'un ensemble d'équations. Toutes ces équations ont le même ensemble de variables, qui sont introduites une seule fois par le mot clé **variables**. Ensuite, les  $\Sigma$ -équations  $(X, t_1, t_2)$  sont introduites par le mot clé **equations** et notées  $t_1 == t_2$ .

**Définition 1.10** (Algèbre satisfaisant une équation,  $\Sigma$ - $E$ -algèbre)

Soit une signature  $\Sigma = (S, \Omega, \varpi)$ . On dit qu'une  $\Sigma$ -algèbre  $A$  *satisfait une équation*  $(X, t_1, t_2)$  si et seulement si pour toute affectation  $aff : X \rightarrow A$ ,  $\overline{aff}(t_1) = \overline{aff}(t_2)$ . Étant donné un ensemble d'équations  $E$  sur  $\Sigma$ , une  $\Sigma$ - $E$ -algèbre est une  $\Sigma$ -algèbre qui satisfait toutes les équations de  $E$ .

**Définition 1.11** (Fermeture sémantique d'un ensemble d'équations)

Soit une signature  $\Sigma$  et un ensemble d'équations  $E$ . La *fermeture sémantique*  $\overline{E}$  de  $E$  est l'ensemble des équations sur  $\Sigma$  qui sont satisfaites par toute  $\Sigma$ - $E$ -algèbre.

**Définition 1.12** (Extension aux termes d'un morphisme de signatures)

Soit deux signatures  $\Sigma = (S, \Omega, \varpi)$  et  $\Sigma' = (S', \Omega', \varpi')$ , ainsi qu'un morphisme de signatures  $m : \Sigma \rightarrow \Sigma'$ . Soit un ensemble de variables  $X$  sur  $\Sigma$ , dont les éléments sont distincts des constantes de  $\Omega'$ .

La *traduction* de  $X$  par le morphisme de signatures  $m : \Sigma \rightarrow \Sigma'$  est le  $S'$ -ensemble  $X^\#$  tel que pour toute sorte  $s'$  de  $S'$ ,

$$X_{s'}^\# = \bigcup_{s'=m^S(s)} X_s.$$

L'*extension* du morphisme de signatures  $m : \Sigma \rightarrow \Sigma'$  est une famille indicée par  $S$  d'applications  $m^\# = (m_s^\# : T_\Sigma(X)_s \rightarrow T_{\Sigma'}(X^\#)_{m^S(s)})_{s \in S}$ . Pour toute sorte  $s \in S$ , l'application  $m_s^\# : T_\Sigma(X)_s \rightarrow T_{\Sigma'}(X^\#)_{m^S(s)}$  est définie inductivement sur la structure des éléments de  $T_\Sigma(X)_s$  de la façon suivante.

- $\forall x \in X_s, m_s^\#(x) = x$  ( $x \in X_{m^S(s)}^\#$ );
- $\forall \omega : s_1, s_2, \dots, s_n \rightarrow s, \forall t_i \in T_\Sigma(X)_{s_i}$  ( $i = 1, \dots, n$ ),  

$$m_s^\#(\omega(t_1, t_2, \dots, t_n)) = m^\Omega(\omega)(m_{s_1}^\#(t_1), m_{s_2}^\#(t_2), \dots, m_{s_n}^\#(t_n)).$$

Étant donné un terme  $t \in T_\Sigma(X)_s$ ,  $m^\#(t) = m_s^\#(t)$  est appelé *traduction de  $t$  par  $m$* .

**Exemple 1.3** Considérons le morphisme de signatures  $m : \Sigma_M \rightarrow \Sigma_{Nat}$  et le terme  $1 * x$  de  $T_{\Sigma_M}(X)$ . La traduction du terme  $1 * x$  sur  $\Sigma_M$  par  $m$  est le terme sur  $\Sigma_{Nat}$

$$m^\#(1 * x) = m^\#(1) + m^\#(x) = 0 + x.$$

**Définition 1.13** (Traduction d'une équation par un morphisme de signatures)

Soit deux signatures  $\Sigma = (S, \Omega, \varpi)$  et  $\Sigma' = (S', \Omega', \varpi')$ , ainsi qu'un morphisme de signatures  $m : \Sigma \rightarrow \Sigma'$ . Soit une équation  $(X, t_1, t_2)$  sur  $\Sigma$ . La *traduction* de l'équation  $(X, t_1, t_2)$  par le morphisme de signatures  $m$  est l'équation  $(X^\#, m^\#(t_1), m^\#(t_2))$  sur la signature  $\Sigma'$ .

**Exemple 1.4** Considérons par exemple l'équation  $(\{x\}, 1 * x, x)$  sur la signature  $\Sigma_M$ . La traduction de cette équation par le morphisme de signatures  $m : \Sigma_M \rightarrow \Sigma_{Nat}$  est l'équation  $(\{x\}, 0 + x, x)$  sur  $\Sigma_{Nat}$ .

## 1.2 Spécifications

Dans ce paragraphe, nous présentons les spécifications *algébriques*, qui consistent en une *signature* et un ensemble de *formules*. Dans l'exemple des anneaux développé dans ce chapitre, les formules que nous considérons sont uniquement des *équations*. Pour cette raison, nous nous contentons de définir des spécifications équationnelles. Nous verrons en section 1.7 que cette hypothèse peut être affaiblie en se plaçant dans le cadre général des institutions.

**Définition 1.14** (Spécification équationnelle) Une *spécification équationnelle* est un couple  $(\Sigma, E)$ , où  $\Sigma$  est une signature et  $E$  est un ensemble de  $\Sigma$ -équations.

**Définition 1.15** (Morphisme de spécifications) Un *morphisme de spécifications* entre deux spécifications  $(\Sigma, E)$  et  $(\Sigma', E')$  est un morphisme de signatures  $m : \Sigma \rightarrow \Sigma'$  tel que pour toute équation  $e \in E$ ,  $m^\#(e) \in \overline{E'}$ . Rappelons que

- $m^\#(e)$  est la *traduction* de l'équation  $e$  par le morphisme de signatures  $m$ ;
- $\overline{E'}$  est la fermeture sémantique de  $E'$ , c'est-à-dire l'ensemble des équations sur  $\Sigma'$  qui sont satisfaites dans toute  $\Sigma'$ - $E'$ -algèbre.

Une autre façon de définir un morphisme de spécifications est d'imposer que la traduction par  $m^\#$  de toute équation de  $E$  appartienne à la théorie équationnelle de  $E'$ , c'est-à-dire peut être déduite des équations de  $E'$  par déduction en logique équationnelle. Il s'agit d'une définition plus *syntactique* de la notion de spécification. Nous avons choisi de suivre la définition donnée pour les institutions par J. Goguen et R. Burstall dans [GB90]. Dans le cadre des spécifications algébriques équationnelles, les deux définitions sont équivalentes puisque le système de déduction équationnel est complet. En effet, une équation  $e$  est satisfaite dans toute  $\Sigma$ - $E$ -algèbre si et seulement si on peut déduire  $e$  des équations de  $E$  par raisonnement équationnel.

On a une catégorie des spécifications équationnelles, notée **Spec**, dont les objets sont les spécifications (équationnelles) et les flèches sont les morphismes de spécifications.

### Exemple 1.5

- La spécification la plus simple est la spécification vide  $(\Sigma_\emptyset, \emptyset)$ , construite sur la signature vide  $\Sigma_\emptyset$ , et dont l'ensemble des équations est vide. Cette spécification est *initiale* dans la catégorie des spécifications, car pour toute spécification  $Sp$ , il existe un unique morphisme de spécifications de la spécification vide vers  $Sp$ .
- Voici un exemple de spécification des entiers écrit en LPG :

```

type      Nat
sorts     nat
operators 0 :  -> nat
          succ : nat -> nat
          + :  nat, nat -> nat
variables x, y : nat
equations 0 + y == y
          s(x) + y == s(x + y)

```

Le langage LPG permet de spécifier à la fois des *types*, comme le type **Nat**, qui ont pour interprétation une algèbre initiale (ou pour les types génériques une algèbre libre), et des *propriétés*, qui ont pour interprétation une classe d'algèbres satisfaisant les équations de la spécification. On peut consulter [Rey87] pour avoir une description plus détaillée de la sémantique de LPG. Dans l'exemple développé ici, nous utiliserons uniquement les propriétés, sans importation de type, ce qui signifie qu'un *modèle* ou une *interprétation* d'une spécification est tout simplement une algèbre qui satisfait les équations de la spécification. Le langage LPG permet également de déclarer des morphismes de spécifications. Un morphisme d'une spécification  $Sp$  vers une spécification  $Sp'$  est défini au moment de la déclaration du codomaine  $Sp'$  par une instruction commençant par le mot clé **satisfies** ou par le mot clé **inherits**. Le langage LPG permet également de considérer les compositions des morphismes de spécifications déclarés. Les règles de composition de ces morphismes et leur sémantique associée ont été décrits dans [BO95].

Nous allons maintenant présenter les spécifications équationnelles et les morphismes de spécifications qui nous permettront de définir plusieurs spécifications modulaires des anneaux. Rappelons qu'un ensemble  $A$ , muni de deux opérations binaires  $+$  et  $*$  est un *anneau* si et seulement si  $A$  muni de l'opération  $+$  est un groupe commutatif,  $A$  muni de l'opération  $*$  est un monoïde, et l'opération  $*$  est distributive par rapport à l'opération  $+$ .

Nous commençons par définir deux spécifications  $S$  et  $B$  qui comportent uniquement une signature et aucune équation. La spécification  $S$  contient une sorte unique  $s$ . Un modèle de  $S$  est donc un ensemble quelconque.

$$S = \begin{array}{|l} \text{property} \quad \text{UNE-SORTE} \\ \text{sorts} \quad \quad s \end{array}$$

La spécification  $B$  contient une sorte  $s$  et un opérateur binaire  $\text{op}$ . Un modèle de  $B$  consiste donc en un ensemble muni d'une opération binaire. Nous avons également spécifié un morphisme de spécifications  $s : S \rightarrow B$  par la proposition `satisfies UNE-SORTE[s]` dans  $B$ . Le morphisme de spécifications  $s$  associe la sorte  $s$  de  $S$  à la sorte  $s$  de  $B$ .

$$B = \begin{array}{|l} \text{property} \quad \text{OP-BIN} \\ \text{sorts} \quad \quad s \\ \text{operators} \quad \text{op} : s, s \rightarrow s \\ \text{satisfies} \quad \text{UNE-SORTE}[s] \end{array}$$

Nous définissons maintenant une spécification  $M$  pour les monoïdes. La signature de la spécification  $M$  comporte une sorte  $s$ , une opération binaire  $*$ , et une constante  $1$ . La spécification  $M$  comporte également trois équations qui indiquent que  $1$  est élément neutre pour  $*$ , et que  $*$  est associative. Un modèle de  $M$  est donc un ensemble muni d'une opération binaire associative et d'un élément neutre. Il s'agit donc bien d'un monoïde.

Nous avons également un morphisme de spécifications  $b : B \rightarrow M$ . Le morphisme  $b$  associe à la sorte  $s$  de  $B$  la sorte  $s$  de  $M$  et à l'opérateur  $\text{op}$  de  $B$  l'opérateur  $*$  de  $M$ . Ce morphisme de signatures est bien un morphisme de spécifications, puisque la spécification de l'opérateur binaire  $B$  ne comporte aucune équation.

$$M = \begin{array}{|l} \text{property} \quad \text{MONOIDE} \\ \text{sorts} \quad \quad s \\ \text{operators} \quad * : s, s \rightarrow s \\ \quad \quad \quad 1 : \rightarrow s \\ \text{variables} \quad x, y, z : s \\ \text{equations} \quad 1 * x == x \\ \quad \quad \quad x * 1 == x \\ \quad \quad \quad (x * y) * z == x * (y * z) \\ \text{satisfies} \quad \text{OP-BIN}[s,*] \end{array}$$

Pour obtenir un groupe commutatif, nous devons ajouter une opération inverse  $i$ , et la commutativité de l'opérateur binaire. Cela nous donne la spécification  $G$ .

$G =$	<b>property</b>	GRUPE-COMM
	<b>sorts</b>	$s$
	<b>operators</b>	$+ : s, s \rightarrow s$
		$0 : \rightarrow s$
		$i : s \rightarrow s$
	<b>variables</b>	$x, y : s$
	<b>equations</b>	$x + y == y + x$
		$i(x) + x == 0$
	<b>inherits</b>	MONOIDE[ $s, +, 0$ ]

Nous spécifions également un morphisme de signatures de la signature de  $M$  vers la signature de  $G$ ,  $m : \Sigma_M \rightarrow \Sigma_G$ , par la proposition **inherits** MONOIDE[ $s, +, 0$ ]. Ce morphisme de signatures associe la sorte  $s$  de  $M$  à la sorte  $s$  de  $G$ , l'opérateur  $*$  de  $M$  à l'opérateur  $+$  de  $G$ , et la constante  $1$  de  $M$  à la constante  $0$  de  $G$ . Le mot clé **inherits** signifie que l'on "recopie" les équations de la spécification  $M$  dans la spécification  $G$  via le morphisme de signatures  $m^\#$ . Autrement dit,  $G$  contient implicitement les équations

```
variables  x, y, z : s
equations  0 + x == x
           x + 0 == x
           (x + y) + z == x + (y + z)
```

Par conséquent, le morphisme de signatures  $m : \Sigma_M \rightarrow \Sigma_G$  est un morphisme de spécifications  $m : M \rightarrow G$ .

Enfin, la propriété  $D$  spécifie deux opérateurs  $+$  et  $*$  tels que  $*$  est distributif par rapport à  $+$ , ainsi que deux morphismes de spécifications  $m_+, m_* : B \rightarrow D$ .

$D =$	<b>property</b>	DISTRIBUTIVITE
	<b>sorts</b>	$s$
	<b>operators</b>	$+, * : s, s \rightarrow s$
	<b>variables</b>	$x, y, z : s$
	<b>equations</b>	$x * (y + z) == (x * y) + (x * z)$
	<b>satisfies</b>	OP-BIN[ $s, +$ ], OP-BIN[ $s, *$ ]

Les deux morphismes  $m_+, m_* : B \rightarrow D$  sont définis par :

```
m_+(s)  = s;
m_+(op) = +;
m_*(s)  = s;
m_*(op) = *.
```

Finalement, pour résumer, nous avons déclaré cinq spécifications  $S, B, M, G, D$ , et cinq morphismes de spécifications  $s, b, m, m_+, m_*$ . Nous pouvons évidemment

considérer également les morphismes de spécifications identités  $id_S$ ,  $id_B$ ,  $id_M$ ,  $id_G$  et  $id_D$  sur ces cinq spécifications, ainsi que toutes les compositions possibles pour les morphismes de spécifications. On vérifie facilement que l'on obtient ainsi quinze morphismes de spécifications différents. Autrement dit, nous considérons une *catégorie de spécifications*, appelée  $\mathcal{C}_0$ , qui comporte ces cinq spécifications et ces quinze morphismes de spécifications.

$$\begin{array}{ccccc}
 S & \xrightarrow{s} & B & \xrightarrow{b} & M & \xrightarrow{m} & G \\
 & & \downarrow m_+ & & \downarrow m_* & & \\
 & & D & & & & 
 \end{array}$$

Squelette de la catégorie  $\mathcal{C}_0$

Nous avons défini les spécifications équationnelles, qui forment une catégorie **Spec**, ainsi qu'une sous-catégorie finie  $\mathcal{C}_0$  de **Spec**, que nous appellerons catégorie des spécifications *de base*. La catégorie  $\mathcal{C}_0$  représente l'ensemble des spécifications disponibles en bibliothèque. Nous nous intéressons ensuite uniquement aux spécifications que l'on peut construire en assemblant des spécifications de base.

### 1.3 Composition de spécifications en LPG

Une spécification *modulaire* est soit une spécification de base, soit une spécification obtenue par un assemblage de plusieurs spécifications modulaires. Nous parlerons de *composition* de spécifications pour décrire l'assemblage de plusieurs spécifications permettant de construire une spécification plus complexe.

La composition de spécifications en LPG est réalisée par ce qu'on appelle une *combinaison* de spécifications. Pour réaliser cette opération, il faut définir une signature — la signature de la spécification résultat — et des morphismes de spécifications des spécifications combinées vers la spécification résultat.

Nous pouvons par exemple écrire une spécification des anneaux en combinant les propriétés MONOIDE, DISTRIBUTIVITE et GROUPE-COMM de la façon suivante.

$A_1 =$	property	ANNEAU
	sorts	s
	operators	+, * : s, s -> s
		0, 1 : -> s
		i : s -> s
combines	MONOIDE[s,*,1], DISTRIBUTIVITE[s,+,*], GROUPE-COMM[s,+,0,i]	

Sémantiquement, cette combinaison signifie que chaque fois qu'on a un modèle de MONOÏDE, un modèle de DISTRIBUTIVITE et un modèle de GROUPE-COMM, tels que *ces trois modèles ont la même interprétation pour la sorte  $s$ , que l'opérateur du monoïde est le second opérateur de la distributivité et l'opérateur du groupe commutatif est le premier opérateur de la distributivité*, alors on a un modèle de la spécification ANNEAU. Ces *partages* de sorte et d'opérateurs sont évidemment cruciaux. Il ne suffit pas d'avoir un monoïde, un groupe commutatif et deux opérateurs distributifs l'un par rapport à l'autre pour avoir un anneau. Par exemple, si nous avons écrit

```
combines ... DISTRIBUTIVITE[s,*,+]
```

à la place de

```
combines ... DISTRIBUTIVITE[s,+,*]
```

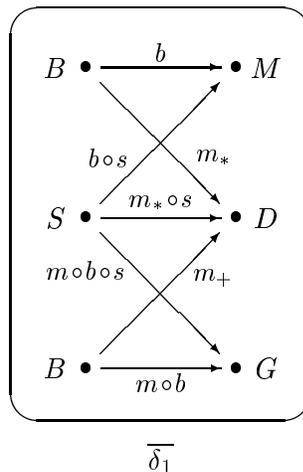
nous n'aurions pas spécifié un anneau. (La multiplication est distributive par rapport à l'addition, et non l'inverse.)

En LPG, les partages entre spécifications, en particulier lors de la construction de combinaisons de propriétés, sont exprimés de façon "implicite", par des morphismes de spécifications. En pratique, on voit que la sorte  $s$  est partagée par les trois spécifications par l'apparition de  $s$  dans la définition des trois morphismes.

```
combines MONOÏDE[s,*,1],
         DISTRIBUTIVITE[s,+,*],
         GROUPE-COMM[s,+,0,i]
```

Dans notre travail, ces partages effectués lors de la composition des spécifications sont exprimés par des *colimites* de *diagrammes*. La colimite est un concept de théorie des catégories qui sera présenté dans le chapitre 2. Cette construction permet, en particulier dans le cadre des spécifications algébriques, de modéliser l'assemblage de plusieurs objets en rendant explicites certains partages entre ces objets.

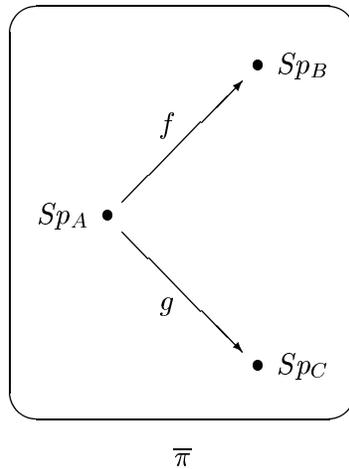
Dans l'exemple ci-dessus, la spécification modulaire  $A_1$  correspond à la colimite du diagramme  $\overline{\delta}_1$ .



Voici l'interprétation intuitive de ce diagramme :

- Les trois nœuds étiquetés par les spécifications  $M$ ,  $D$  et  $G$ , qui apparaissent comme des puits du graphe (aucun arc n'a pour source un nœud étiqueté par  $M$ ,  $D$  ou  $G$ ) sont les spécifications que l'on assemble.
- Les trois nœuds étiquetés par  $B$ ,  $S$  et  $B$  indiquent les partages.
  - La sorte  $\mathbf{s}$  est partagée par les trois spécifications  $M$ ,  $D$  et  $G$ . Ce partage est modélisé par les morphismes  $b \circ s : S \rightarrow M$ ,  $m_* \circ s : S \rightarrow D$  et  $m \circ b \circ s : S \rightarrow G$ .
  - L'opérateur binaire  $*$  est partagé par les spécifications  $M$  et  $D$ . Ce partage est modélisé par les morphismes  $b : B \rightarrow M$  et  $m_* : B \rightarrow D$ .
  - L'opérateur binaire  $+$  est partagé par les spécifications  $G$  et  $D$ . Ce partage est modélisé par les morphismes  $m \circ b : B \rightarrow G$  et  $m_+ : B \rightarrow D$ .

Par la suite, nous privilégions une construction particulière de colimite: la somme amalgamée. La somme amalgamée correspond à la colimite d'un diagramme  $\bar{\pi}$  qui ne comporte que trois spécifications  $Sp_A$ ,  $Sp_B$  et  $Sp_C$ , liées par deux morphismes de spécifications  $f : Sp_A \rightarrow Sp_B$  et  $g : Sp_A \rightarrow Sp_C$ .



La colimite de ce diagramme correspond à la composition des spécifications  $Sp_B$  et  $Sp_C$  où les partages entre ces deux spécifications sont indiqués par les deux morphismes  $f$  et  $g$ .

## 1.4 Sommes amalgamées

Dans cette section, nous définissons la somme amalgamée de deux ensembles, puis de deux signatures, et enfin de deux spécifications. Autrement dit, nous présentons la somme amalgamée dans trois catégories particulières: **Set** (catégorie des ensembles), **Sig** (catégorie des signatures) et **Spec** (catégorie des spécifications). La définition générale sera revue dans le chapitre 2.

### 1.4.1 Somme amalgamée de deux ensembles

Considérons trois ensembles  $A$ ,  $B$ , et  $C$ , ainsi que deux applications  $f : A \rightarrow B$  et  $g : A \rightarrow C$ . Intuitivement, faire la somme amalgamée de  $B$  et  $C$  par rapport à  $f$  et  $g$  consiste à faire l'union disjointe de  $B$  et  $C$  et à fusionner les éléments qui sont images par  $f$  et  $g$  d'un même élément  $a$  de  $A$ .

Plus formellement, on peut définir la somme disjointe de  $B$  et  $C$  par l'ensemble

$$B + C = (\{1\} \times B) \cup (\{2\} \times C).$$

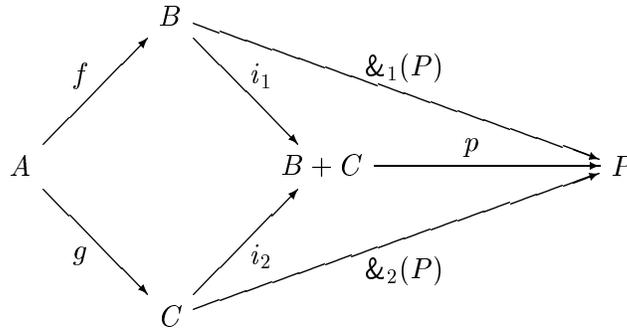
Il y a évidemment un choix de codage dans cette définition. L'idée est de faire la réunion des deux ensembles  $B$  et  $C$  après avoir renommé les éléments communs aux deux ensembles. Nous avons également deux applications  $i_1 : B \rightarrow B + C$  et  $i_2 : C \rightarrow B + C$  définies par

$$\begin{array}{ll} i_1 : B \rightarrow B + C & i_2 : C \rightarrow B + C \\ b \mapsto (1, b) & c \mapsto (2, c) \end{array}$$

Puis on définit une relation  $\mathcal{R}$  sur l'ensemble  $B + C$  :

$$\mathcal{R} = \{((1, f(a)), (2, g(a))) \ ; \ \forall a \in A\}.$$

On considère ensuite la relation d'équivalence  $\overline{\mathcal{R}}$  engendrée par  $\mathcal{R}$  et l'ensemble quotient  $P = (B + C)/\overline{\mathcal{R}}$ . On a une application de projection  $p : B + C \rightarrow P$ , qui à tout élément de  $B + C$  associe sa classe d'équivalence modulo  $\overline{\mathcal{R}}$ .



#### Notations

- L'ensemble  $P$  est noté  $\text{push}(A, B, C, f, g)$ .
- L'application  $p \circ i_1 : B \rightarrow P$  est notée  $\&_1(A, B, C, f, g)$ .
- L'application  $p \circ i_2 : C \rightarrow P$  est notée  $\&_2(A, B, C, f, g)$ .
- Pour alléger un peu les notations, si  $P \equiv \text{push}(A, B, C, f, g)$ , nous écrirons
  - $\&_1(P)$  à la place de  $\&_1(A, B, C, f, g)$  ;

–  $\&_2(P)$  à la place de  $\&_2(A, B, C, f, g)$ .

**Définition 1.16** (Somme amalgamée dans **Set**)

La somme amalgamée de  $B$  et  $C$  par rapport à  $f$  et  $g$  est le triplet

$$(\text{push}(A, B, C, f, g), \&_1(A, B, C, f, g), \&_2(A, B, C, f, g)).$$

Par abus de langage, l'ensemble  $\text{push}(A, B, C, f, g)$  sera également appelé somme amalgamée. Il ne faut pas oublier qu'en réalité les deux applications  $\&_1(A, B, C, f, g)$  et  $\&_2(A, B, C, f, g)$  font partie de la définition d'une somme amalgamée.

Nous formulons maintenant la propriété générale des sommes amalgamées en théorie des catégories. Nous reverrons cette propriété dans le chapitre 2.

**Propriété 1.1** (*Propriété caractéristique des sommes amalgamées*)

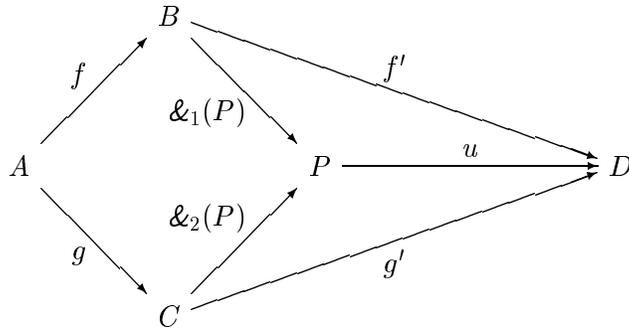
*Une somme amalgamée*

$$(\text{push}(A, B, C, f, g), \&_1(A, B, C, f, g), \&_2(A, B, C, f, g))$$

satisfait les propriétés suivantes.

1.  $\&_1(A, B, C, f, g) \circ f = \&_2(A, B, C, f, g) \circ g$ ;
2. pour tout objet  $D$  et flèches  $f' : B \rightarrow D$ ,  $g' : C \rightarrow D$  telles que  $f' \circ f = g' \circ g$ , il existe une flèche unique  $u : \text{push}(A, B, C, f, g) \rightarrow D$  telle que

$$\begin{cases} u \circ \&_1(A, B, C, f, g) = f' \\ u \circ \&_2(A, B, C, f, g) = g'. \end{cases}$$



Dans la catégorie des ensembles **Set**, les objets sont les ensembles, et les flèches sont les applications entre deux ensembles. La propriété 1.1 se reformule donc de la façon suivante :

$$1. \ \&_1(A, B, C, f, g) \circ f = \&_2(A, B, C, f, g) \circ g$$

(Il s'agit d'une égalité entre deux applications de  $A$  vers  $\text{push}(A, B, C, f, g)$ .)

2. Pour tout ensemble  $D$  et applications  $f' : B \rightarrow D$ ,  $g' : C \rightarrow D$  telles que  $f' \circ f = g' \circ g$ , il existe une application unique  $u : \text{push}(A, B, C, f, g) \rightarrow D$  telle que

$$\begin{cases} u \circ \&_1(A, B, C, f, g) = f' \\ u \circ \&_2(A, B, C, f, g) = g'. \end{cases}$$

**Exemple 1.6** Soit  $A = \{a_1, a_2\}$ ,  $B = \{x, y, z\}$  et  $C = \{x\}$ .  
Soit  $f = \{(a_1, x), (a_2, y)\}$  et  $g = \{(a_1, x), (a_2, x)\}$ .

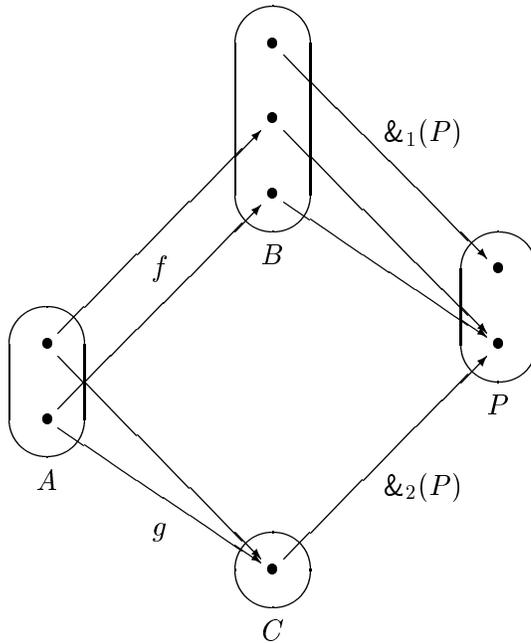
$$B + C = \{(1, x), (1, y), (1, z), (2, x)\}$$

$$P = \text{push}(A, B, C, f, g) = \{\{(1, x), (1, y), (2, x)\}, \{(1, z)\}\}$$

$$\&_1(P)(x) = \&_1(P)(y) = \{(1, x), (1, y), (2, x)\}$$

$$\&_1(P)(z) = \{(1, z)\}$$

$$\&_2(P)(x) = \{(1, x), (1, y), (2, x)\}$$



**Remarque 1.2** Nous avons en réalité défini ici un *codage* de la somme amalgamée pour les ensembles. Dans la théorie des catégories, la somme amalgamée est définie à un isomorphisme près. Dans l'exemple 1.6, en prenant pour  $P$  un ensemble quelconque à deux éléments, la propriété 1.1 est encore vérifiée.

### 1.4.2 Somme amalgamée de deux signatures

Nous définissons la somme amalgamée de deux signatures à partir de la définition de la somme amalgamée de deux ensembles. Soit trois signatures

$$\begin{aligned}\Sigma_A &= (S_A, \Omega_A, \varpi_A) \\ \Sigma_B &= (S_B, \Omega_B, \varpi_B) \\ \Sigma_C &= (S_C, \Omega_C, \varpi_C)\end{aligned}$$

et deux morphismes de signatures

$$\begin{aligned}f &= (f^S, f^\Omega) : \Sigma_A \rightarrow \Sigma_B \\ g &= (g^S, g^\Omega) : \Sigma_A \rightarrow \Sigma_C.\end{aligned}$$

On définit la signature  $\Sigma_P = (S_P, \Omega_P, \varpi_P)$ , somme amalgamée des signatures  $\Sigma_B$  et  $\Sigma_C$  par rapport aux morphismes  $f : \Sigma_A \rightarrow \Sigma_B$  et  $g : \Sigma_A \rightarrow \Sigma_C$  de la façon suivante. L'ensemble  $S_P$  est la somme amalgamée des ensembles  $S_B$  et  $S_C$  par rapport à  $f^S$  et  $g^S$ . Autrement dit,

$$S_P = \text{push}(S_A, S_B, S_C, f^S, g^S).$$

L'ensemble  $\Omega_P$  est la somme amalgamée des ensembles  $\Omega_B$  et  $\Omega_C$  par rapport à  $f^\Omega$  et  $g^\Omega$ . Autrement dit,

$$\Omega_P = \text{push}(\Omega_A, \Omega_B, \Omega_C, f^\Omega, g^\Omega).$$

Il reste à définir  $\varpi_P : \Omega_P \rightarrow S_P^+$  pour obtenir la signature  $\Sigma_P$ . On a l'égalité

$$\&_1(S_P)^+ \circ \varpi_B \circ f^\Omega = \&_2(S_P)^+ \circ \varpi_C \circ g^\Omega.$$

En effet,

$$\begin{aligned}\&_1(S_P)^+ \circ \varpi_B \circ f^\Omega &= \&_1(S_P)^+ \circ f^{S^+} \circ \varpi_A \quad (f : \Sigma_A \rightarrow \Sigma_B \text{ morphisme de signatures}) \\ &= (\&_1(S_P) \circ f^S)^+ \circ \varpi_A \quad (-^+ \text{ foncteur}) \\ &= (\&_2(S_P) \circ g^S)^+ \circ \varpi_A \quad (\text{propriété des sommes amalgamées}) \\ &= \&_2(S_P)^+ \circ g^{S^+} \circ \varpi_A \quad (-^+ \text{ foncteur}) \\ &= \&_2(S_P)^+ \circ \varpi_C \circ g^\Omega \quad (g : \Sigma_A \rightarrow \Sigma_C \text{ morphisme de signatures})\end{aligned}$$

donc, comme  $\Omega_P = \text{push}(\Omega_A, \Omega_B, \Omega_C, f^\Omega, g^\Omega)$  et d'après la propriété caractéristique des sommes amalgamées, il existe une unique application  $\varpi_P : \Omega_P \rightarrow S_P^+$  telle que

$$\varpi_P \circ \&_1(\Omega_P) = \&_1(S_P)^+ \circ \varpi_B \tag{1}$$

$$\varpi_P \circ \&_2(\Omega_P) = \&_2(S_P)^+ \circ \varpi_C \tag{2}$$

On pose

$$\begin{aligned}\text{push}(\Sigma_A, \Sigma_B, \Sigma_C, f, g) &= \Sigma_P; \\ \&_1(\Sigma_P) &= \&_1(\Sigma_A, \Sigma_B, \Sigma_C, f, g) = (\&_1(S_P), \&_1(\Omega_P)); \\ \&_2(\Sigma_P) &= \&_2(\Sigma_A, \Sigma_B, \Sigma_C, f, g) = (\&_2(S_P), \&_2(\Omega_P)).\end{aligned}$$

D'après les égalités (1) et (2) ci-dessus,  $\&_1(\Sigma_P)$  et  $\&_2(\Sigma_P)$  sont bien des morphismes de signatures.

**Théorème 1.2** *Le triplet*

$$(\text{push}(\Sigma_A, \Sigma_B, \Sigma_C, f, g), \&_1(\Sigma_A, \Sigma_B, \Sigma_C, f, g), \&_2(\Sigma_A, \Sigma_B, \Sigma_C, f, g))$$

est une somme amalgamée dans la catégorie des signatures **Sig**.

*Preuve.* On vérifie facilement que la propriété 1.1 est satisfaite.  $\square$

**Exemple 1.7** Considérons les signatures  $\Sigma_S$ ,  $\Sigma_M$  et  $\Sigma_G$  correspondant respectivement aux spécifications  $S$ ,  $M$  et  $G$  (propriétés `UNE_SORTE`, `MONOIDE` et `GROUPE_COMM`). Nous avons également deux morphismes de signatures  $b \circ s : \Sigma_S \rightarrow \Sigma_M$  et  $m \circ b \circ s : \Sigma_S \rightarrow \Sigma_G$ .

Une somme amalgamée de  $\Sigma_M$  et  $\Sigma_G$  par rapport à  $b \circ s$  et  $m \circ b \circ s$  est la signature  $\Sigma_P$  suivante :

```

sorts      s
operators  +, * : s, s -> s
           0, 1 : -> s
           i : s -> s

```

On doit fusionner la sorte  $s$  de  $\Sigma_M$  avec la sorte  $s$  de  $\Sigma_G$ . Remarquons que nous n'avons pas été obligés de renommer certaines sortes ou opérateurs. En effet, les deux sortes à fusionner ont le même nom dans les deux signatures  $\Sigma_M$  et  $\Sigma_G$ , et les opérateurs que l'on doit distinguer ont des noms distincts dans les deux signatures. Finalement comme la signature partagée  $\Sigma_S$  est l'*intersection* des signatures  $\Sigma_M$  et  $\Sigma_G$ , et comme les deux morphismes de signatures sont les inclusions correspondantes, la somme amalgamée  $\Sigma_P$  est la *réunion* de  $\Sigma_M$  et  $\Sigma_G$ .

### 1.4.3 Somme amalgamée de deux spécifications

On définit la somme amalgamée de deux spécifications de façon similaire: la signature de la spécification résultat est la somme amalgamée des signatures, et l'ensemble des équations de la spécification résultat est "l'union" (après traduction) des équations des deux spécifications.

Soit trois spécifications

$$\begin{aligned} Sp_A &= (\Sigma_A, E_A) \\ Sp_B &= (\Sigma_B, E_B) \\ Sp_C &= (\Sigma_C, E_C) \end{aligned}$$

et deux morphismes de spécifications

$$\begin{aligned} f &: Sp_A \rightarrow Sp_B \\ g &: Sp_A \rightarrow Sp_C. \end{aligned}$$

On définit la spécification  $Sp_P = (\Sigma_P, E_P)$  somme amalgamée des spécifications  $Sp_B$  et  $Sp_C$  par rapport aux morphismes de spécifications  $f$  et  $g$  de la façon suivante. La signature  $\Sigma_P$  est la somme amalgamée des signatures :

$$\Sigma_P = \text{push}(\Sigma_A, \Sigma_B, \Sigma_C, f, g).$$

On note

$$\begin{aligned}\&_1(\Sigma_P) &= \&_1(\Sigma_A, \Sigma_B, \Sigma_C, f, g); \\ \&_2(\Sigma_P) &= \&_2(\Sigma_A, \Sigma_B, \Sigma_C, f, g).\end{aligned}$$

Intuitivement,  $E_P$  contient les équations de  $E_B$  et de  $E_C$ . Comme les équations de  $E_B$  et de  $E_C$  sont respectivement des équations sur les signatures  $\Sigma_B$  et  $\Sigma_C$ , on les traduit respectivement par les morphismes de signatures  $\&_1(\Sigma_P) : \Sigma_B \rightarrow \Sigma_P$  et  $\&_2(\Sigma_P) : \Sigma_C \rightarrow \Sigma_P$ . On pose donc

$$\begin{aligned}E_P &= \{(X^\#, \&_1(\Sigma_P)^\#(t), \&_1(\Sigma_P)^\#(t')) ; \forall (X, t, t') \in E_B\} \\ &\cup \{(X^\#, \&_2(\Sigma_P)^\#(t), \&_2(\Sigma_P)^\#(t')) ; \forall (X, t, t') \in E_C\}.\end{aligned}$$

**Remarque 1.3** Les équations présentes dans  $E_A$  ne jouent pas directement de rôle. En effet, elles sont déjà “contenues” dans  $E_B$  et  $E_C$  puisque  $f$  et  $g$  sont des morphismes de spécifications.

**Lemme 1.1** *Les morphismes de signatures*

$$\begin{aligned}\&_1(\Sigma_P) &: \Sigma_B \rightarrow \Sigma_P \\ \&_2(\Sigma_P) &: \Sigma_C \rightarrow \Sigma_P\end{aligned}$$

sont des morphismes de spécifications.

*Preuve.* C'est évident, par définition de  $E_P$ . □

On pose

$$\begin{aligned}\&_1(Sp_P) &= \&_1(Sp_A, Sp_B, Sp_C, f, g) = \&_1(\Sigma_A, \Sigma_B, \Sigma_C, f, g) : Sp_B \rightarrow Sp_P; \\ \&_2(Sp_P) &= \&_2(Sp_A, Sp_B, Sp_C, f, g) = \&_2(\Sigma_A, \Sigma_B, \Sigma_C, f, g) : Sp_C \rightarrow Sp_P.\end{aligned}$$

**Théorème 1.3** *Le triplet*

$$(\text{push}(Sp_A, Sp_B, Sp_C, f, g), \&_1(Sp_A, Sp_B, Sp_C, f, g), \&_2(Sp_A, Sp_B, Sp_C, f, g))$$

est une somme amalgamée dans la catégorie des spécifications **Spec**.

*Preuve.* On vérifie facilement que la propriété 1.1 est satisfaite. □

**Exemple 1.8** Pour poursuivre l'exemple 1.7, on fait la somme amalgamée des spécifications  $M$  et  $G$  par rapport aux morphismes de spécifications  $b \circ s : S \rightarrow M$  et  $m \circ b \circ s : S \rightarrow G$ . Cela nous donne la spécification  $P$  suivante:

$$P = \text{push}(S, M, G, b \circ s, m \circ b \circ s).$$

Détaillons le contenu de la spécification  $P$ .

$P =$	property	PSEUDO_ANNEAU
	sorts	s
	operators	+, * : s, s -> s 0, 1 : -> s i : s -> s
	variables	x, y, z : s
	equations	1 * x == x x * 1 == x (x * y) * z == x * (y * z) 0 + x == x x + 0 == x (x + y) + z == x + (y + z) x + y == y + x i(x) + x == 0

Nous avons appelé cette spécification **PSEUDO\_ANNEAU** car il s'agit d'une spécification des anneaux sans distributivité entre les opérateurs  $*$  et  $+$ .

#### 1.4.4 Syntaxe abstraite pour les constructions modulaires

Nous récapitulons la syntaxe abstraite qui nous permet de représenter des constructions modulaires. Les deux constructions autorisées sont la spécification vide et la somme amalgamée de deux spécifications. En effet, ces deux seules constructions permettent de simuler la construction de toute colimite finie.

- La spécification vide  $(\Sigma_{\emptyset}, \emptyset)$  est notée  $\emptyset$ .

L'unique morphisme de spécifications de  $\emptyset$  vers une spécification quelconque  $Sp$  est noté  $j_{Sp} : \emptyset \rightarrow Sp$ .

- Étant donné trois spécifications  $Sp_A, Sp_B, Sp_C$  et deux morphismes de spécifications  $f : Sp_A \rightarrow Sp_B$  et  $g : Sp_A \rightarrow Sp_C$ , la somme amalgamée de  $Sp_B$  et  $Sp_C$  par rapport à  $f$  et  $g$  est donnée par une spécification

$$\text{push}(Sp_A, Sp_B, Sp_C, f, g)$$

et deux morphismes de spécifications

$$\begin{cases} \&_1(Sp_A, Sp_B, Sp_C, f, g) : Sp_B \rightarrow \text{push}(Sp_A, Sp_B, Sp_C, f, g) \\ \&_2(Sp_A, Sp_B, Sp_C, f, g) : Sp_C \rightarrow \text{push}(Sp_A, Sp_B, Sp_C, f, g). \end{cases}$$

De plus, étant donné une spécification  $Sp_D$  et deux flèches  $f' : Sp_B \rightarrow Sp_D$  et  $g' : Sp_C \rightarrow Sp_D$  telles que  $f' \circ f = g' \circ g$ , l'unique morphisme de spécifications

$$u : \text{push}(Sp_A, Sp_B, Sp_C, f, g) \rightarrow Sp_D$$

tel que

$$\begin{cases} u \circ \&_1(A, B, C, f, g) = f' \\ u \circ \&_2(A, B, C, f, g) = g' \end{cases}$$

est notée  $\text{up}(A, B, C, D, f, g, f', g') : \text{push}(Sp_A, Sp_B, Sp_C, f, g) \rightarrow Sp_D$ .

### 1.4.5 Somme disjointe de deux spécifications

Il est possible de coder la somme disjointe de deux spécifications en utilisant la spécification vide et une somme amalgamée. Soit deux spécifications  $Sp_A$  et  $Sp_B$ . La somme disjointe de  $Sp_A$  et  $Sp_B$  est la somme amalgamée de  $Sp_A$  et  $Sp_B$  par rapport à  $j_{Sp_A}$  et  $j_{Sp_B}$ .

$$Sp_A + Sp_B = \text{push}(\emptyset, Sp_A, Sp_B, j_{Sp_A}, j_{Sp_B})$$

De plus, les deux injections de  $Sp_A$  vers  $Sp_A + Sp_B$  et de  $Sp_B$  vers  $Sp_A + Sp_B$  sont les deux flèches

$$\begin{aligned} \&_1(\emptyset, Sp_A, Sp_B, j_{Sp_A}, j_{Sp_B}) : Sp_A &\rightarrow Sp_A + Sp_B; \\ \&_2(\emptyset, Sp_A, Sp_B, j_{Sp_A}, j_{Sp_B}) : Sp_B &\rightarrow Sp_A + Sp_B. \end{aligned}$$

**Exemple 1.9** Considérons par exemple la somme disjointe  $B_2$  de deux opérateurs binaires.

$$B_2 = \text{push}(\emptyset, B, B, j_B, j_B)$$

Cette somme amalgamée correspond à la propriété LPG suivante

$$B_2 = \begin{array}{|l|l|} \hline \text{property} & \text{DEUX\_OP\_BIN} \\ \hline \text{sorts} & \text{s1, s2} \\ \hline \text{operators} & \text{op1 : s1 -> s1} \\ & \text{op2 : s2 -> s2} \\ \hline \end{array}$$

Les deux morphismes de spécifications  $\&_1(B_2) : B \rightarrow B_2$  et  $\&_2(B_2) : B \rightarrow B_2$  sont définis par

$$\begin{aligned} \&_1(B_2)(\text{s}) &= \text{s1}; \\ \&_1(B_2)(\text{op}) &= \text{op1}; \\ \&_2(B_2)(\text{s}) &= \text{s2}; \\ \&_2(B_2)(\text{op}) &= \text{op2}. \end{aligned}$$

Nous avons deux morphismes de spécifications  $m_* : B \rightarrow D$  et  $m_+ : B \rightarrow D$ . Ces deux morphismes vérifient l'égalité  $m_* \circ j_B = m_+ \circ j_B$  car  $\emptyset$  est initial dans la catégorie **Spec**. Par conséquent, il existe un unique morphisme de spécifications

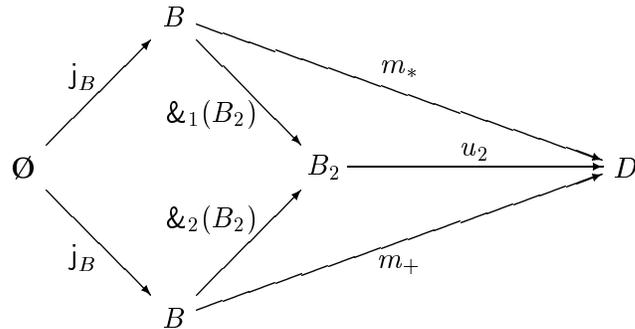
$$u_2 = \text{up}(\emptyset, B, B, D, j_B, j_B, m_*, m_+) : B_2 \rightarrow D$$

tel que

$$\begin{cases} u_2 \circ \&_1(B_2) = m_* \\ u_2 \circ \&_2(B_2) = m_+. \end{cases}$$

Le morphisme de spécifications  $u_2 : B_2 \rightarrow D$  est défini par

$$\begin{aligned}
u_2(s1) &= s; \\
u_2(s2) &= s; \\
u_2(op1) &= *; \\
u_2(op2) &= +.
\end{aligned}$$



## 1.5 Spécifications modulaires d'anneaux

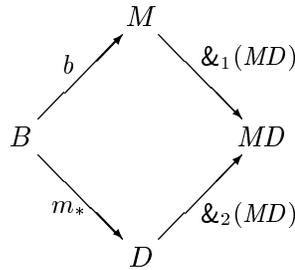
Nous avons vu en section 1.3 un exemple de spécification modulaire des anneaux, en combinant les trois spécifications  $M$ ,  $G$  et  $D$ . On peut coder cette combinaison en utilisant deux sommes amalgamées successives. Par exemple, on peut commencer par combiner  $M$  et  $D$ , puis combiner le résultat avec  $G$ .

$MD =$	property	MONOIDE-DISTRIBUTIF
	sorts	s
	operators	+, * : s, s -> s
		1 : -> s
	combines	MONOIDE[s,*,1], DISTRIBUTIVITE[s,+,*]

$A_2 =$	property	ANNEAU2
	sorts	s
	operators	+, * : s, s -> s
		0, 1 : -> s
		i : s -> s
combines	MONOIDE-DISTRIBUTIF[s,+,*,1], GROUPE-COMM[s,+,0,i]	

La combinaison des spécifications  $M$  et  $D$  peut être exprimée à l'aide d'une somme amalgamée, en partageant l'opérateur binaire  $*$  de  $M$  avec l'opérateur  $*$  de  $D$ . La spécification modulaire  $MD$  peut donc être exprimée de la façon suivante :

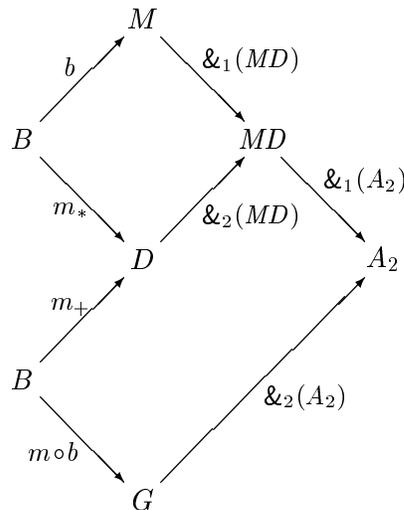
$$MD = \text{push}(B, M, D, b, m_*)$$



On peut remarquer que dans cette expression, il est nécessaire d'avoir exprimé la spécification  $B$  pour l'opérateur binaire partagé, alors qu'en LPG, cette spécification partagée reste implicite. En effet, on n'a pas fait référence à  $\text{OP-BIN}$  dans la spécification  $\text{MONOIDE-DISTRIBUTIF}$ .

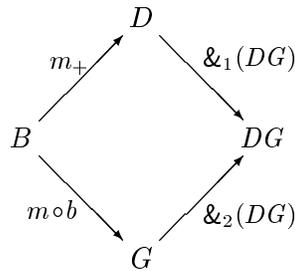
On peut ensuite exprimer  $A_2$  comme la somme amalgamée de  $MD$  et  $G$  en partageant l'opérateur binaire  $+$  de  $MD$  avec l'opérateur binaire  $+$  de  $G$ .

$$A_2 = \text{push}(B, MD, G, \&_2(MD) \circ m_+, m \circ b)$$

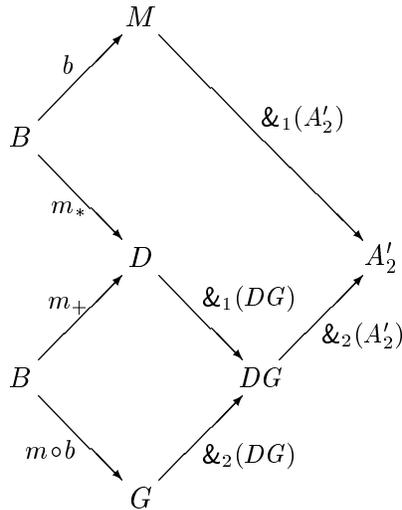


On peut effectuer ces deux combinaisons successives différemment, en combinant d'abord les spécifications  $G$  et  $D$  (en partageant l'opérateur  $+$ ), puis en ajoutant la spécification  $M$  (en partageant l'opérateur  $*$ ). Cela donne la spécification  $A'_2$ .

$$DG = \text{push}(B, D, G, m_+, m \circ b)$$



$$A'_2 = \text{push}(B, M, DG, b, \&_1(DG) \circ m_*)$$



On peut maintenant se poser la question suivante :

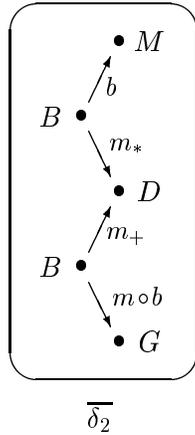
Avons-nous bien obtenu une spécification des anneaux ?

Dans ce cas particulier, on peut s'en convaincre assez facilement, en connaissant une définition indépendante des anneaux. Ici, on va s'intéresser à la question plus facile :

Est-ce que les spécifications  $A_2$  et  $A'_2$  sont équivalentes ?

Pour répondre à cette question, il est inutile d'avoir une définition indépendante des anneaux. On cherche simplement à savoir si les deux spécifications modulaires sont isomorphes. Ici, il est facile de voir que  $A_2$  et  $A'_2$  sont bien isomorphes. Intuitivement, les deux constructions sont équivalentes parce que l'opération de somme amalgamée est "associative" (dans un sens qui resterait à définir formellement). En

fait, les deux constructions  $A_2$  et  $A'_2$  sont un codage par des sommes amalgamées de la colimite du même diagramme  $\overline{\delta}_2$  suivant.



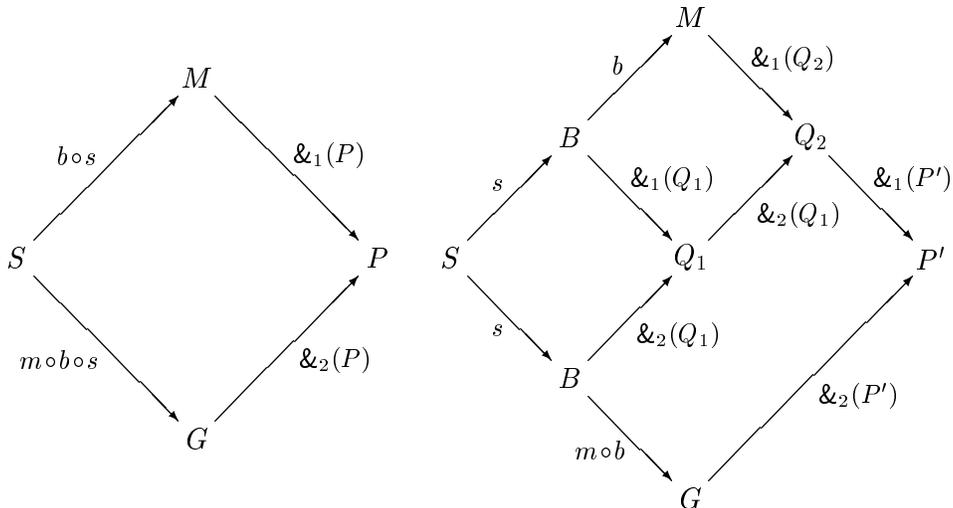
Mais des cas plus complexes peuvent se produire, comme le montre l'exemple suivant. Nous commençons par spécifier un pseudo-anneau, c'est-à-dire un anneau sans la propriété de distributivité. On peut faire cela par exemple avec le terme  $P$ , ou avec le terme  $P'$ .

$$P = \text{push}(S, M, G, b \circ s, m \circ b \circ s)$$

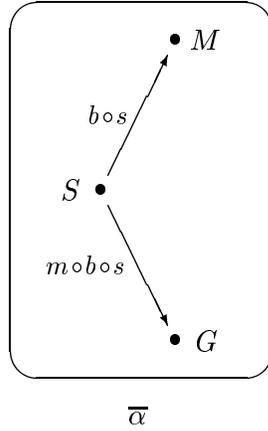
$$Q_1 = \text{push}(S, B, B, s, s)$$

$$Q_2 = \text{push}(B, M, Q_1, b, \&_1(Q_1))$$

$$P' = \text{push}(B, Q_2, G, \&_2(Q_2) \circ \&_2(Q_1), m \circ b)$$



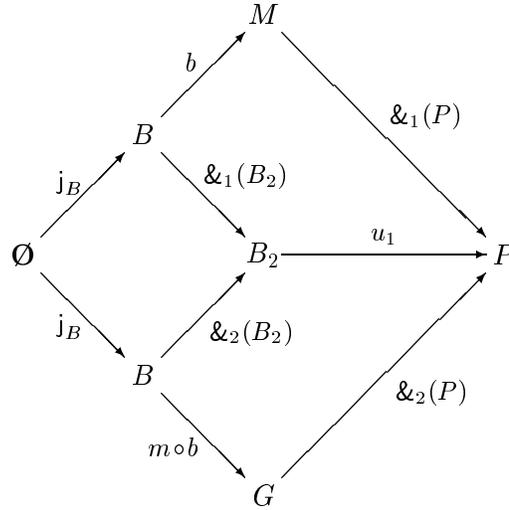
En fait,  $P$  et  $P'$  sont deux spécifications isomorphes des pseudo-anneaux, qui correspondent à la colimite du diagramme  $\overline{\alpha}$ .



Nous allons maintenant “ajouter la distributivité” à la spécification  $P$ . Pour cela, nous considérons le morphisme de spécifications

$$u_1 = \text{up}(\emptyset, B, B, P, j_B, j_B, \&_1(P) \circ b, \&_2(P) \circ m \circ b) : B_2 \rightarrow P$$

Cette flèche existe car comme  $\emptyset$  est initial,  $\&_1(P) \circ b \circ j_B = \&_2(P) \circ m \circ b \circ j_B$ .

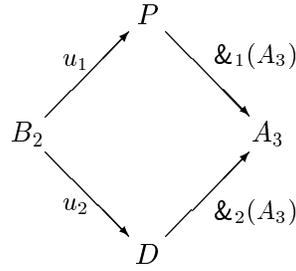


Nous avons vu en exemple 1.9 qu’il existe un morphisme de spécifications

$$u_2 = \text{up}(\emptyset, B, B, D, j_B, j_B, m_*, m_+) : B_2 \rightarrow D$$

On obtient alors une spécification  $A_3$  des anneaux en faisant la somme amalgamée de  $D$  et  $P$  par rapport à  $u_2$  et  $u_1$ . Cela revient à partager les deux opérateurs binaires de  $P$  et  $D$ .

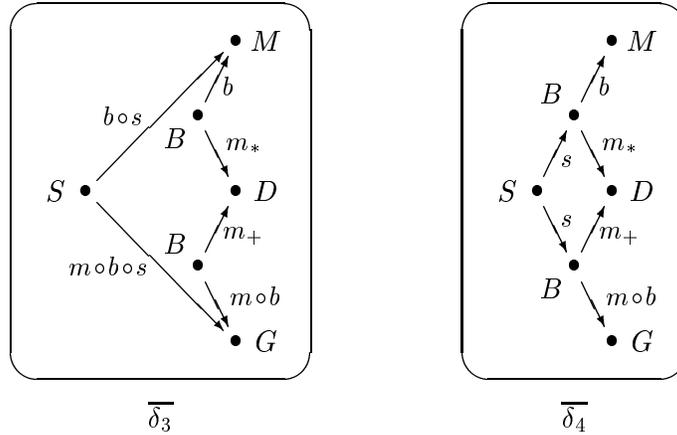
$$\begin{aligned} A_3 &= \text{push}(B_2, P, D, u_1, u_2) \\ &= \text{push}(\text{push}(\emptyset, B, B, j_B, j_B), P, D, \\ &\quad \text{up}(\emptyset, B, B, P, j_B, j_B, \&_1(P) \circ b, \&_2(P) \circ m \circ b), \\ &\quad \text{up}(\emptyset, B, B, D, j_B, j_B, m_*, m_+)) \end{aligned}$$



Enfin, voici une dernière spécification modulaire des anneaux, en utilisant la spécification  $P'$  des pseudo-anneaux à laquelle on ajoute la propriété de distributivité.

$$\begin{aligned}
 A_4 = \text{push} & (\text{push} (\emptyset, B, B, j_B, j_B), P', D, \\
 & \text{up} (\emptyset, B, B, P', j_B, j_B, \&_1(P') \circ \&_2(Q_2) \circ \&_1(Q_1), \\
 & \&_1(P') \circ \&_2(Q_2) \circ \&_2(Q_1)), \\
 & \text{up} (\emptyset, B, B, D, j_B, j_B, m_*, m_+))
 \end{aligned}$$

Les deux constructions modulaires d'anneaux  $A_3$  et  $A_4$  correspondent aux colimites des diagrammes  $\overline{\delta}_3$  et  $\overline{\delta}_4$  ci-dessous.



En utilisant la définition de colimite en théorie des catégorie, on peut vérifier que les *colimites* des diagrammes  $\overline{\delta}_1$ ,  $\overline{\delta}_2$ ,  $\overline{\delta}_3$  et  $\overline{\delta}_4$  sont isomorphes. Cet isomorphisme vient de l'égalité des morphismes de spécifications

$$m_+ \circ s = m_* \circ s.$$

Cette égalité signifie que le fait que les deux opérateurs  $+$  et  $*$  opèrent sur le même ensemble est contenu dans la spécification  $D$  des opérateurs distributifs.

Nous verrons dans le chapitre 2 que dans la catégorie de diagrammes  $\text{diagr}(\mathcal{C}_0)$ , les diagrammes  $\overline{\delta}_1$ ,  $\overline{\delta}_2$ ,  $\overline{\delta}_3$  et  $\overline{\delta}_4$  sont isomorphes.

## 1.6 Isomorphisme de construction

Nous venons de voir qu'il existe plusieurs manières équivalentes de spécifier des anneaux à partir d'une catégorie de spécifications de base donnée. Dans ce paragraphe, nous examinons quelques équivalences possibles entre deux spécifications. Formellement, deux spécifications sont équivalentes si elles sont isomorphes dans une certaine catégorie de spécifications. Nous présentons les isomorphismes de construction, qui nous semblent présenter un intérêt particulier dans l'étude des spécifications modulaires.

*Identité.* Deux spécifications sont équivalentes si et seulement si celles-ci sont identiques. Il s'agit de la plus faible équivalence possible, qui n'est pas très intéressante.

*Équivalence structurelle.* Deux spécifications sont équivalentes si et seulement si celles-ci ont été construites de la même façon, indépendamment de noms intermédiaires qui ont pu être utilisés pendant la construction. Avec cette approche, on a la possibilité de nommer des spécifications modulaires intermédiaires et de réutiliser ces noms par la suite. Cette équivalence n'est pas beaucoup plus intéressante que l'identité.

*Isomorphisme dans **Spec**.* Deux spécifications sont isomorphes si et seulement si il existe un isomorphisme entre ces deux spécifications dans la catégorie des spécifications **Spec**. La difficulté ici est tout d'abord d'exhiber l'isomorphisme entre les deux signatures, et surtout de vérifier qu'il s'agit bien d'un isomorphisme de spécifications. Avec la définition de morphisme de spécifications donnée ici, il faut être capable de savoir si une équation appartient à la fermeture sémantique d'un ensemble d'équations. Ce n'est pas trivial puisqu'il faut pouvoir dire si une équation est satisfaite par une classe d'algèbres spécifiée par un ensemble d'équations. On peut pour cela utiliser un système logique, mais ce système n'est pas forcément complet. De plus, même si l'on dispose d'un système de déduction complet, le problème est, en général, uniquement semi-décidable.

*Isomorphisme de construction.* Dans tout notre travail, les isomorphismes que nous considérons sont des isomorphismes *de construction*. Deux spécifications sont isomorphes si on peut le prouver en utilisant les propriétés générales des colimites. Il s'agit d'un isomorphisme dans la catégorie librement engendrée par certaines colimites finies sur la catégorie des spécifications de base  $\mathcal{C}_0$ . L'intérêt de cet isomorphisme réside d'une part dans le fait qu'il n'est pas *trop général*, parce qu'il reflète la *construction* de la spécification modulaire; et d'autre part dans le fait qu'il est suffisamment général pour faire abstraction des étapes particulières choisies pour la construction modulaire. Ces isomorphismes de construction ne dépendent pas de la définition effective des spécifications, mais seulement de la manière dont ces spécifications, liées par des morphismes de spécifications, sont combinées. En particulier, une modification "mineure" dans la catégorie des spécifications de base ne remettra pas en cause des isomorphismes de construction. (Par "mineure" on entend ici que la modification

ne change pas les égalités entre morphismes de spécifications de base existants). Par contre, si deux spécifications de base sont isomorphes et que cet isomorphisme ne fait pas partie de la catégorie des spécifications de base, alors celles-ci ne seront évidemment pas liées par un isomorphisme de construction.

## 1.7 Cadre général

Les spécifications que nous avons données au cours de ce chapitre sont des spécifications équationnelles. En effet, les seules formules utilisées sont des équations. Il existe évidemment des spécifications plus générales, par exemple des spécifications qui utilisent des clauses de Horn, ou la logique du premier ordre, ou encore les spécifications avec contraintes.

Pour formaliser la notion de système logique, J. Goguen et R. Burstall ont proposé les *institutions* [GB84, GB90]. Ce formalisme permet de parler de façon abstraite des signatures, des formules sur une signature, des modèles sur une signature et de la notion de satisfaction d'une formule par un modèle. L'intérêt des institutions est de fournir des résultats indépendants de la logique sous-jacente choisie.

Dans notre travail, nous étudions la composition des spécifications pour former des spécifications plus complexes à l'aide de colimites. Ce problème est identique qu'on se place par exemple dans le cadre des spécifications équationnelles, ou des spécifications avec clauses de Horn. En fait, on peut raisonner indépendamment de la logique sous-jacente en se plaçant dans le cadre général des institutions. La seule propriété que nous exigeons est l'existence de *colimites finies* dans la catégorie des spécifications.

### 1.7.1 Institutions

Rappelons d'abord la définition d'une *institution* [GB90].

**Définition 1.17** (Institution) Une institution consiste en

- une catégorie **Sig**, dont les objets sont appelés *signatures*, et les flèches *morphismes de signatures*;
- un foncteur  $Sen : \mathbf{Sig} \rightarrow \mathbf{Set}$  qui associe à chaque signature un ensemble de *propositions*, et à chaque morphisme de signatures une *application* entre deux ensembles de propositions;
- un foncteur  $Mod : \mathbf{Sig} \rightarrow \mathbf{Cat}^{\text{op}}$  qui associe à chaque signature une catégorie de *modèles*, et à chaque morphisme de signatures un foncteur contravariant;
- pour tout objet  $\Sigma$  de **Sig**, une relation  $\models_{\Sigma}$  entre les objets de  $Mod(\Sigma)$  et les éléments de  $Sen(\Sigma)$ , appelée relation de satisfaction, telle que pour tout morphisme de signatures  $m : \Sigma \rightarrow \Sigma'$ , tout modèle  $M'$  objet de  $Mod(\Sigma')$ , et toute équation  $e$  de  $Sen(\Sigma)$ ,

$$M' \models_{\Sigma'} Sen(m)(e) \Leftrightarrow Mod(m)(M') \models_{\Sigma} e. \quad (i)$$

Les spécifications équationnelles que nous avons utilisées dans ce chapitre forment une institution.

- **Sig** est la catégorie des signatures définie en section 1.1.
- *Sen* associe à toute signature  $\Sigma$  l'ensemble des équations sur  $\Sigma$ , et à tout morphisme de signatures  $m : \Sigma \rightarrow \Sigma'$  l'application  $m^\#$  qui traduit les équations.
- *Mod* associe à toute signature  $\Sigma$  la catégorie des  $\Sigma$ -algèbres  $\text{Alg}(\Sigma)$ , et à tout morphisme de signatures  $m : \Sigma \rightarrow \Sigma'$  le *foncteur d'oubli*

$$\mathcal{U}_m : \text{Alg}(\Sigma') \rightarrow \text{Alg}(\Sigma).$$

Ce foncteur d'oubli permet de considérer toute  $\Sigma'$ -algèbre comme une  $\Sigma$ -algèbre.

- Soit une signature  $\Sigma$ , une  $\Sigma$ -algèbre  $A$  et une équation  $e$  sur  $\Sigma$ ,

$$A \models_\Sigma e \Leftrightarrow \text{l'algèbre } A \text{ satisfait l'équation } e \text{ (cf. définition 1.10).}$$

La propriété (i) est vérifiée car pour toute algèbre  $A'$  de  $\text{Alg}(\Sigma')$ ,

$$A' \models m^\#(e) \Leftrightarrow \mathcal{U}_m(A') \models e$$

(cf. par exemple [GB90, EM90]).

**Définition 1.18** (Spécification) Une spécification est un couple  $(\Sigma, E)$  où  $\Sigma$  est un objet de **Sig**, et  $E$  un sous-ensemble de  $\text{Sen}(\Sigma)$ .

C'est ce que J. Goguen et R. Burstall appellent une  $\Sigma$ -présentation [GB90].

**Définition 1.19** (Fermeture sémantique d'un ensemble de propositions)

Soit une signature  $\Sigma$  et un ensemble de propositions  $E \subseteq \text{Sen}(\Sigma)$ . La *fermeture sémantique*  $\overline{E} \subseteq \text{Sen}(\Sigma)$  de  $E$  est définie par :

$$e \in \overline{E} \text{ si et seulement si pour tout modèle } M \text{ de } \text{Mod}(\Sigma),$$

$$\forall e_0 \in E, M \models_\Sigma e_0 \Rightarrow M \models_\Sigma e.$$

On dit que  $E$  est fermé si et seulement si  $E = \overline{E}$ .

**Définition 1.20** (Morphisme de spécifications) Soit deux spécifications  $(\Sigma, E)$  et  $(\Sigma', E')$ . Un *morphisme de spécifications* de  $(\Sigma, E)$  vers  $(\Sigma', E')$  est un morphisme de signatures  $m : \Sigma \rightarrow \Sigma'$ , tel que

$$\forall e \in E, \text{Sen}(m)(e) \in \overline{E'}.$$

Les spécifications, avec les morphismes de spécifications, forment une catégorie, notée **Spec**.

Notre but est de composer les spécifications par des constructions de colimites. Nous devons donc travailler dans une catégorie pour laquelle tout diagramme fini a une colimite, c'est-à-dire dans une catégorie finiment cocomplète. J. Goguen et R. Burstall ont montré qu'il suffit que la catégorie des signatures ait des colimites pour que la catégorie des spécifications associée ait des colimites.

**Théorème 1.4** ([GB90]) *La catégorie **Spec** est finiment cocomplète si et seulement si la catégorie associée de signatures **Sig** est finiment cocomplète.*

**Remarque 1.4** J. Goguen et R. Burstall ont montré ce résultat pour les *théories*, c'est-à-dire les spécifications  $(\Sigma, E)$  telles que  $E$  est fermé. On vérifie facilement qu'on peut adapter leur démonstration dans le cas des spécifications.

Cela permet de déduire que la catégorie des spécifications algébriques équationnelles, la catégorie des spécifications du premier ordre ... etc. sont finiment cocomplètes.

## 1.7.2 Terminologie et hypothèses

### Hypothèses

- Nous nous plaçons dans une institution, dans laquelle la catégorie des signatures **Sig** est finiment cocomplète. Cela implique que la catégorie des spécifications associée **Spec** est finiment cocomplète.
- On considère une sous-catégorie finie  $\mathcal{C}_0$  de **Spec**.

### Terminologie

Rappelons brièvement la terminologie employée.

*Spécification et morphisme de spécifications.* Une spécification et un morphisme de spécifications sont respectivement un objet et une flèche de la catégorie **Spec**.

*Spécification de base et morphisme de spécifications de base.* On fixe une sous-catégorie finie  $\mathcal{C}_0$  de **Spec**. Les objets de  $\mathcal{C}_0$  sont appelés *spécifications de base*, et les flèches de  $\mathcal{C}_0$  *morphismes de spécifications de base*. Intuitivement, cette catégorie représente l'ensemble des spécifications et morphismes de spécifications "déclarés" par l'utilisateur, qui forment une bibliothèque de spécifications réutilisables.

*Spécification modulaire et morphisme de spécifications modulaires.* Une spécification modulaire est soit une spécification de base soit une spécification construite à partir d'autres spécifications modulaires par certaines constructions de colimites. Les morphismes de spécifications modulaires sont des morphismes de spécifications également obtenus par des constructions de colimites.

### Angles d'études

Nous nous intéressons à la composition des spécifications de base sous différents angles :

*Syntaxe.* Peut-on définir une syntaxe, c'est-à-dire un langage pour décrire des constructions modulaires ? Dans le chapitre 3 nous proposons la construction d'une catégorie  $\text{Terme}(\mathcal{C}_0)$ . Cette catégorie permet de représenter la construction des spécifications modulaires à partir de spécifications de base en utilisant uniquement la spécification vide et des sommes amalgamées. Ce chapitre permet de justifier formellement la syntaxe abstraite utilisée dans l'exemple des anneaux.

*Interprétation.* Quel est le résultat de plusieurs constructions modulaires successives ? Nous montrons dans le chapitre 4 que des constructions successives de sommes amalgamées peuvent être vue comme le calcul d'une seule colimite. Nous montrons comment calculer le diagramme associé à un terme qui représente une spécification modulaire. Dans ce chapitre, une spécification modulaire est vue de façon plus abstraite comme la colimite d'un diagramme sur  $\mathcal{C}_0$ .

*Calcul.* Peut-on détecter des isomorphismes de construction ? Nous résolvons ce problème uniquement dans le cas simple où la catégorie  $\mathcal{C}_0$  est finie et ne comporte pas de cycle. Nous proposons un algorithme qui permet, par exemple, de déduire que les spécifications  $A_1$ ,  $A_2$ ,  $A_3$  et  $A_4$  des anneaux que nous avons présentées dans ce chapitre sont équivalentes. Cet algorithme est présenté dans le chapitre 5.

# Chapitre 2

## Graphes, catégories et diagrammes

Le but de ce chapitre est de présenter les bases formelles de notre travail, en particulier les concepts de *diagramme* et de *morphisme de diagrammes*, issus de la théorie des catégories. Pour cela nous avons besoin de quelques notions sur les graphes, présentées en section 2.1. Nous introduisons la définition des *morphismes généralisés de graphes*. En section 2.2, nous faisons quelques rappels de théorie des catégories. La plupart des résultats sont tirés de [McL71] ou [MB70]. Sur ce sujet, il est également intéressant de consulter [BW90] qui introduit les concepts de base de façon assez claire et élémentaire. En section 2.3, nous présentons les notions de diagramme et de morphisme de diagrammes.

La définition de morphisme de diagrammes que nous utilisons, bien qu'elle ne soit pas nouvelle, est plus générale que celles présentées classiquement dans la littérature informatique (cf. par exemple [TBG91]). Nous avons en effet besoin d'une définition suffisamment générale pour "réfléter" le maximum de flèches entre colimites de diagrammes par des morphismes entre les diagrammes correspondants. Nous nous basons ensuite sur cette définition pour reformuler les notions classiques de cône sur un diagramme et de colimite d'un diagramme.

Nous présentons deux catégories de diagrammes:  $\text{DIAGR}(\mathcal{C})$  et  $\text{diagr}(\mathcal{C})$ . La catégorie  $\text{DIAGR}(\mathcal{C})$ , qui est une généralisation de catégories de diagrammes utilisées en informatique, permet, de façon pratique, de manipuler les morphismes de diagrammes. La catégorie  $\text{diagr}(\mathcal{C})$ , obtenue comme un quotient de  $\text{DIAGR}(\mathcal{C})$  par une relation de congruence sur les flèches de  $\text{DIAGR}(\mathcal{C})$ , possède deux propriétés théoriques importantes. D'une part,  $\text{diagr}(\mathcal{C})$  est une catégorie finiment cocomplète (théorème 2.6). De plus,  $\text{diagr}(\mathcal{C})$  est une complétion de la catégorie  $\mathcal{C}$  par colimites finies (théorème 2.7).

### 2.1 Graphes

**Définition 2.1** (Graphe) Un graphe  $\alpha^\Phi$  est un ensemble de nœuds  $\text{Nœud}(\alpha^\Phi)$ , un ensemble d'arcs  $\text{Arc}(\alpha^\Phi)$ , et deux fonctions  $\text{Source}, \text{But} : \text{Arc}(\alpha^\Phi) \rightarrow \text{Nœud}(\alpha^\Phi)$ . Si  $a$  est un arc de source  $m$  et de but  $n$ , on note  $a : m \rightarrow n$ .

Un graphe est *fini* lorsque l'ensemble des nœuds et l'ensemble des arcs sont finis.

**Définition 2.2** (Morphisme de graphes) Soit deux graphes  $\alpha^\Phi$  et  $\beta^\Phi$ . Un morphisme de graphes  $\sigma^\Phi$  de  $\alpha^\Phi$  vers  $\beta^\Phi$ , noté  $\sigma^\Phi : \alpha^\Phi \rightarrow \beta^\Phi$ , est une application qui à tout nœud  $n$  de  $\alpha^\Phi$  associe un nœud  $\sigma^\Phi(n)$  de  $\beta^\Phi$  et à tout arc  $a : m \rightarrow n$  de  $\alpha^\Phi$  associe un arc  $\sigma^\Phi(a) : \sigma^\Phi(m) \rightarrow \sigma^\Phi(n)$  de  $\beta^\Phi$ .

*Composition des morphismes de graphes.* Soit trois graphes  $\alpha^\Phi$ ,  $\beta^\Phi$ ,  $\gamma^\Phi$ , et deux morphismes de graphes  $\sigma^\Phi : \alpha^\Phi \rightarrow \beta^\Phi$  et  $\tau^\Phi : \beta^\Phi \rightarrow \gamma^\Phi$ . La composition de  $\sigma^\Phi$  et  $\tau^\Phi$  est le morphisme  $\tau^\Phi \circ \sigma^\Phi : \alpha^\Phi \rightarrow \gamma^\Phi$  qui associe à tout nœud  $n$  de  $\alpha^\Phi$  le nœud  $\tau^\Phi(\sigma^\Phi(n))$  de  $\gamma^\Phi$ , et à tout arc  $a : m \rightarrow n$  de  $\alpha^\Phi$  associe l'arc  $\tau^\Phi(\sigma^\Phi(a)) : \tau^\Phi(\sigma^\Phi(m)) \rightarrow \tau^\Phi(\sigma^\Phi(n))$ . En résumé,  $\tau^\Phi \circ \sigma^\Phi$  est défini de la façon suivante.

$$\begin{aligned} \tau^\Phi \circ \sigma^\Phi & : & \alpha^\Phi & \rightarrow & \gamma^\Phi \\ & & n & \mapsto & \tau^\Phi(\sigma^\Phi(n)) \\ & & a : m \rightarrow n & \mapsto & \tau^\Phi(\sigma^\Phi(a)) : \tau^\Phi(\sigma^\Phi(m)) \rightarrow \tau^\Phi(\sigma^\Phi(n)) \end{aligned}$$

**Définition 2.3** (Zigzag sur un graphe) Soit un graphe  $\alpha^\Phi$ . Un *zigzag*  $Z$  sur  $\alpha^\Phi$  est un triplet  $(k, Z_N, Z_A)$  où

- $k$  est un entier naturel, appelé *longueur* du zigzag  $Z$ ;
- $Z_N$  est un  $(k + 1)$ -uplet  $(n_0, n_1, \dots, n_k)$  de nœuds de  $\alpha^\Phi$ ;
- $Z_A$  est un  $k$ -uplet  $(a_0, a_1, \dots, a_{k-1})$  d'arcs de  $\alpha^\Phi$ , tel que pour tout entier naturel  $i$  tel que  $0 \leq i \leq k - 1$ ,  $a_i$  est soit un arc de source  $n_i$  et de but  $n_{i+1}$ , soit un arc de source  $n_{i+1}$  et de but  $n_i$ .

On appelle  $n_0$  la *source* du zigzag  $Z$ , et  $n_k$  son *but*. On note  $Z : n_0 \rightsquigarrow n_k$ .

Pour tout nœud  $n$  de  $\alpha^\Phi$ , on a un *zigzag nul* de longueur nulle noté  $0_n : n \rightsquigarrow n$ .

Par la suite, pour plus de lisibilité, un zigzag  $Z : n_0 \rightsquigarrow n_k$  sera représenté de la façon suivante :

$$Z = n_0 \xrightarrow{a_0} n_1 \xleftarrow{a_1} n_2 \xleftarrow{a_2} n_3 \cdots n_{k-1} \xrightarrow{a_{k-1}} n_k.$$

On choisit une orientation arbitraire pour la représentation de chaque arc  $a_i$ .

Étant donné un graphe  $\alpha^\Phi$ , on a un graphe  $\text{Zigzag}(\alpha^\Phi)$ , dont les nœuds sont les nœuds de  $\alpha^\Phi$ , et les arcs les zigzags de  $\alpha^\Phi$ .

À tout zigzag  $Z : n_0 \rightsquigarrow n_k$

$$Z = n_0 \xrightarrow{a_0} n_1 \xleftarrow{a_1} n_2 \xleftarrow{a_2} n_3 \cdots n_{k-1} \xrightarrow{a_{k-1}} n_k,$$

on peut associer un *zigzag opposé*  $\bar{Z} : n_k \rightsquigarrow n_0$

$$\bar{Z} = n_k \xleftarrow{a_{k-1}} n_{k-1} \cdots n_3 \xrightarrow{a_2} n_2 \xrightarrow{a_1} n_1 \xleftarrow{a_0} n_0$$

obtenu en “parcourant le zigzag dans l'autre sens”. Plus formellement, on définit l'opposé d'un zigzag de la façon suivante.

**Définition 2.4** (Zigzag opposé) Soit un zigzag  $Z = (k, Z_N, Z_A)$  sur un graphe  $\alpha^\Phi$ , avec  $Z_N = (n_0, n_1, \dots, n_k)$  et  $Z_A = (a_0, a_1, \dots, a_{k-1})$ . L'opposé du zigzag

$$Z : n_0 \rightsquigarrow n_k$$

est le zigzag

$$\bar{Z} : n_k \rightsquigarrow n_0 = (k, Z'_N, Z'_A)$$

où

- $Z'_N = (n_k, n_{k-1}, \dots, n_0)$ ;
- $Z'_A = (a_{k-1}, \dots, a_0)$ .

**Définition 2.5** (Morphisme généralisé de graphes) Soit deux graphes  $\alpha^\Phi$  et  $\beta^\Phi$ . Un *morphisme généralisé de graphes*  $\sigma^\Phi : \alpha^\Phi \rightsquigarrow \beta^\Phi$  entre les graphes  $\alpha^\Phi$  et  $\beta^\Phi$  est un morphisme de graphes entre  $\alpha^\Phi$  et  $\text{Zigzag}(\beta^\Phi)$ .

Le morphisme généralisé de graphes  $\sigma^\Phi$  associe donc à tout nœud  $n$  de  $\alpha^\Phi$  un nœud  $\sigma^\Phi(n)$  de  $\beta^\Phi$ , et à tout arc  $a : m \rightarrow n$  de  $\alpha^\Phi$  un *zigzag*  $\sigma^\Phi(a) : \sigma^\Phi(m) \rightsquigarrow \sigma^\Phi(n)$  du graphe  $\beta^\Phi$ .

Tout morphisme de graphes est donc un morphisme généralisé, puisqu'un arc est un cas particulier de zigzag.

Un morphisme généralisé associe donc à tout arc un zigzag, en conservant les sources et les buts. On peut étendre la définition d'un morphisme généralisé de graphes  $\sigma^\Phi : \alpha^\Phi \rightsquigarrow \beta^\Phi$  en associant à tout zigzag un zigzag. Il suffit pour cela de "mettre les zigzags bout à bout". Étant donné un zigzag  $Z : n_0 \rightsquigarrow n_k$

$$Z = n_0 \xleftarrow{a_0} n_1 \xrightarrow{a_1} n_2 \cdots n_{k-1} \xrightarrow{a_{k-1}} n_k,$$

on définit le zigzag  $\sigma^\Phi(Z) : \sigma^\Phi(n_0) \rightsquigarrow \sigma^\Phi(n_k)$  par :

$$\sigma^\Phi(Z) = \sigma^\Phi(n_0) \xleftarrow{\sigma^\Phi(a_0)} \sigma^\Phi(n_1) \xrightarrow{\sigma^\Phi(a_1)} \sigma^\Phi(n_2) \cdots \sigma^\Phi(n_{k-1}) \xrightarrow{\sigma^\Phi(a_{k-1})} \sigma^\Phi(n_k).$$

Cela permet de composer les morphismes généralisés. Soit trois graphes  $\alpha^\Phi, \beta^\Phi, \gamma^\Phi$ , et deux morphismes généralisés  $\sigma^\Phi : \alpha^\Phi \rightsquigarrow \beta^\Phi, \tau^\Phi : \beta^\Phi \rightsquigarrow \gamma^\Phi$ . Le morphisme généralisé  $\tau^\Phi \circ \sigma^\Phi$  est défini de la façon suivante.

$$\begin{array}{lcl} \tau^\Phi \circ \sigma^\Phi : & \alpha^\Phi & \rightarrow \gamma^\Phi \\ & n & \mapsto \tau^\Phi(\sigma^\Phi(n)) \\ a : m \rightarrow n & \mapsto & \tau^\Phi(\sigma^\Phi(a)) : \tau^\Phi(\sigma^\Phi(m)) \rightsquigarrow \tau^\Phi(\sigma^\Phi(n)) \end{array}$$

## 2.2 Catégories, foncteurs ...

Dans cette partie, nous rappelons les définitions de catégorie, foncteur, transformation naturelle et adjonction. Ces définitions sont tirées de [McL71, MB70, BW90].

### 2.2.1 Catégories

**Définition 2.6** (Catégorie) Une catégorie  $\mathcal{C}$  est définie de la façon suivante.

- On a une classe d'objets, notés  $A, B, C \dots$
- Pour chaque couple d'objets  $(A, B)$ , on a un ensemble  $hom_{\mathcal{C}}(A, B)$  de flèches, noté plus simplement  $hom(A, B)$  s'il n'y pas d'ambiguïté sur la catégorie.  
 $f \in hom(A, B)$  est noté  $f : A \rightarrow B$ .  
 $A$  est appelé *domaine* de  $f$ , et  $B$  est appelé *codomaine* de  $f$ .

- Pour chaque triplet d'objets  $(A, B, C)$ , on a une opération de composition

$$\begin{aligned} \circ : hom(B, C) \times hom(A, B) &\rightarrow hom(A, C) \\ (g, f) &\mapsto g \circ f \end{aligned}$$

- Pour tout objet  $B$ , on a une flèche  $id_B : B \rightarrow B$ .
- Pour tout objet  $B$ , la flèche  $id_B$  est une *identité*:  
 $\forall f : A \rightarrow B, \forall g : B \rightarrow C, id_B \circ f = f$  et  $g \circ id_B = g$ .
- La composition est associative :  
 $\forall f : A \rightarrow B, \forall g : B \rightarrow C, \forall h : C \rightarrow D, h \circ (g \circ f) = (h \circ g) \circ f$ .

*Exemples de catégories*

- La catégorie des ensembles, notée **Set**, dont les objets sont les ensembles, et les flèches sont les applications entre deux ensembles.
- La catégorie des signatures, notée **Sig**, dont les objets sont les signatures, et les flèches sont les morphismes de signatures.
- La catégorie des spécifications, notée **Spec**, dont les objets sont les spécifications algébriques, et les flèches sont les morphismes de spécifications.
- La catégorie des graphes, notée **Graph**, dont les objets sont les graphes, et les flèches sont les morphismes de graphes.

**Définition 2.7** (Isomorphisme) Soit une catégorie  $\mathcal{C}$  et  $f : A \rightarrow B$  une flèche de  $\mathcal{C}$ . On dit que  $f : A \rightarrow B$  est un *isomorphisme* si et seulement si il existe une flèche  $g : B \rightarrow A$  de  $\mathcal{C}$  telle que  $g \circ f = id_A$  et  $f \circ g = id_B$ .  
 Si  $f$  est un isomorphisme, la flèche  $g$  est alors unique, et est notée  $f^{-1}$ . On dit alors que  $f$  et  $f^{-1}$  sont *inverses* et que les objets  $A$  et  $B$  sont *isomorphes*. On note  $A \cong B$ .

**Définition 2.8** (Petite catégorie) Lorsque la classe d'objets d'une catégorie  $\mathcal{C}$  est un *ensemble*, on dit que  $\mathcal{C}$  est une *petite catégorie*.

Par exemple, les catégories **Set**, **Sig**, **Spec** et **Graph** ne sont pas des petites catégories.

**Remarque 2.1**

- Toute petite catégorie est un graphe.
- Réciproquement, on peut associer à tout graphe  $\alpha^\Phi$  la catégorie *librement engendrée* sur ce graphe. Cette catégorie a pour objets les nœuds du graphe  $\alpha^\Phi$ , et pour flèches les *chaînes d'arcs composables* de  $\alpha^\Phi$ , où deux arcs  $a$  et  $a'$  sont composables si et seulement si  $\text{But}(a) = \text{Source}(a')$ . La composition est définie comme la concaténation des chaînes:  $(b_1 b_2 \dots b_l) \circ (a_1 a_2 \dots a_k) = (a_1 a_2 \dots a_k b_1 b_2 \dots b_l)$ . Les flèches identités sur un objet  $n$  sont les chaînes de longueur nulle de source et but  $n$ , notées  $\varepsilon_n$ .

**2.2.2 Foncteurs**

Un foncteur est l'équivalent d'un morphisme de graphes pour les catégories.

**Définition 2.9** (Foncteur) Soit  $\mathcal{C}$  et  $\mathcal{D}$  deux catégories. Un *foncteur*  $F$  de  $\mathcal{C}$  vers  $\mathcal{D}$ , noté  $F : \mathcal{C} \rightarrow \mathcal{D}$ , est une application qui à tout objet  $A$  de  $\mathcal{C}$  associe un objet  $F(A)$  de  $\mathcal{D}$ , et à toute flèche  $f : A \rightarrow B$  de  $\mathcal{C}$  associe une flèche  $F(f) : F(A) \rightarrow F(B)$  de  $\mathcal{D}$ , et telle que

- pour tout objet  $A$  de  $\mathcal{C}$ ,  $F(\text{id}_A) = \text{id}_{F(A)}$ ;
- pour toutes flèches  $f : A \rightarrow B$  et  $g : B \rightarrow C$  de  $\mathcal{C}$ ,  $F(g \circ f) = F(g) \circ F(f)$ .

Étant donné une catégorie  $\mathcal{C}$ , il y a foncteur identité  $\text{Id}_{\mathcal{C}} : \mathcal{C} \rightarrow \mathcal{C}$ , qui est l'identité sur les objets et sur les flèches.

Soit trois catégories  $\mathcal{C}$ ,  $\mathcal{D}$  et  $\mathcal{E}$ , et deux foncteurs  $F : \mathcal{C} \rightarrow \mathcal{D}$  et  $G : \mathcal{D} \rightarrow \mathcal{E}$ .

On définit la composition de  $F$  et  $G$  de façon évidente:

$$\begin{aligned} G \circ F : \quad \mathcal{C} &\rightarrow \mathcal{E} \\ A &\mapsto G(F(A)) \\ f : A \rightarrow B &\mapsto G(F(f)) : G(F(A)) \rightarrow G(F(B)) \end{aligned}$$

**Catégorie Cat**

La classe des petites catégories, avec les foncteurs, forme une catégorie **Cat**.

**Catégorie CAT**

Par la suite, nous aurons besoin d'une catégorie qui contient non seulement des petites catégories mais également des catégories comme **Set** ou **Spec** qui ne sont pas petites. Nous considérons pour cela la catégorie **CAT** des "grandes catégories", qui contient les petites catégories, ainsi que les autres catégories qui nous intéressent. Pour des problèmes de fondations, de même qu'on ne peut pas considérer l'ensemble de tous les ensembles, **CAT** n'est pas la catégorie de toutes les catégories. En particulier **CAT** n'est pas un objet de **CAT**. Pour plus de détails, on peut consulter [McL71] pages 21–24.

**Propriété 2.1** Soit un foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ . L'image par  $F$  d'un isomorphisme de  $\mathcal{C}$  est un isomorphisme de  $\mathcal{D}$ .

*Preuve.* Soit un foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ , et  $f : A \rightarrow B$  un isomorphisme de  $\mathcal{C}$ .  $F(f) : F(A) \rightarrow F(B)$  est un isomorphisme, et  $F(f)^{-1} = F(f^{-1}) : F(B) \rightarrow F(A)$ . En effet,

$$\begin{aligned} F(f) \circ F(f^{-1}) &= F(f \circ f^{-1}) \quad (F \text{ est un foncteur}) \\ &= F(id_B) \quad (\text{définition de } f^{-1}) \\ &= id_{F(B)} \quad (F \text{ est un foncteur}) \end{aligned}$$

De même,  $F(f^{-1}) \circ F(f) = id_{F(A)}$ , et donc  $F(f)$  est un isomorphisme.  $\square$

**Définition 2.10** (Endofoncteur) Un foncteur  $F : \mathcal{C} \rightarrow \mathcal{C}$  est appelé *endofoncteur*.

Si  $F$  est un endofoncteur, on pose

$$\begin{aligned} F^0 &= Id_{\mathcal{C}}; \\ F^i &= F \circ F^{i-1}, \quad \forall i \geq 1. \end{aligned}$$

**Définition 2.11** (Foncteur plein) Soit deux catégories  $\mathcal{C}$  et  $\mathcal{D}$ , et un foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ . On dit que  $F$  est *plein* si pour tout couple d'objets  $(A, B)$  de  $\mathcal{C}$ , et pour toute flèche  $g : F(A) \rightarrow F(B)$  de  $\mathcal{D}$ , il existe une flèche  $f : A \rightarrow B$  de  $\mathcal{C}$  telle que  $g = F(f)$ .

Intuitivement, entre deux objets de  $\mathcal{D}$  images par  $F$  d'objets de  $\mathcal{C}$ , il n'y a pas de flèches autres que les images par  $F$  de flèches de  $\mathcal{C}$ .  $F$  est, d'une certaine façon, "surjectif sur les flèches".

**Définition 2.12** (Foncteur fidèle) Soit deux catégories  $\mathcal{C}$  et  $\mathcal{D}$ , et un foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ . On dit que  $F$  est *fidèle* si pour tout couple d'objets  $(A, B)$  de  $\mathcal{C}$ , et pour tout couple de flèches  $f, f' : A \rightarrow B$ , on a  $F(f) = F(f') \Rightarrow f = f'$ .

Autrement dit,  $F$  est fidèle si  $F$  est injectif sur les flèches.

### 2.2.3 Transformations naturelles

**Définition 2.13** (Transformation naturelle) Soit deux catégories  $\mathcal{C}$  et  $\mathcal{D}$ . Soit deux foncteurs  $F, G : \mathcal{C} \rightarrow \mathcal{D}$ . Une *transformation naturelle*  $\sigma$  de  $F$  vers  $G$ , notée  $\sigma : F \rightarrow G$ , est une application qui à tout objet  $A$  de  $\mathcal{C}$  associe une flèche  $\sigma_A : F(A) \rightarrow G(A)$  de  $\mathcal{D}$ , telle que pour toute flèche  $f : A \rightarrow B$  de  $\mathcal{C}$ ,

$$G(f) \circ \sigma_A = \sigma_B \circ F(f).$$

$$\begin{array}{ccccc} A & & F(A) & \xrightarrow{\sigma_A} & G(A) \\ \downarrow f & & \downarrow F(f) & & \downarrow G(f) \\ B & & F(B) & \xrightarrow{\sigma_B} & G(B) \end{array}$$

**Catégorie des foncteurs de  $\mathcal{C}$  vers  $\mathcal{D}$** 

Soit trois foncteurs  $F, G, H : \mathcal{C} \rightarrow \mathcal{D}$ , et deux transformations naturelles  $\sigma : F \rightarrow G$  et  $\tau : G \rightarrow H$ . La composition de  $\sigma$  et  $\tau$  est la transformation naturelle  $\tau \cdot \sigma : F \rightarrow H$  telle que pour tout objet  $A$  de  $\mathcal{C}$ ,

$$(\tau \cdot \sigma)_A = \tau_A \circ \sigma_A$$

Cela définit bien une transformation naturelle. En effet,

$$\begin{aligned} H(f) \circ (\tau \cdot \sigma)_A &= H(f) \circ \tau_A \circ \sigma_A \quad (\text{définition de } \tau \cdot \sigma) \\ &= \tau_B \circ G(f) \circ \sigma_A \quad (\tau \text{ transformation naturelle}) \\ &= \tau_B \circ \sigma_B \circ F(f) \quad (\sigma \text{ transformation naturelle}) \\ &= (\tau \cdot \sigma)_B \circ F(f) \quad (\text{définition de } \tau \cdot \sigma) \end{aligned}$$

$$\begin{array}{ccccccc} & & A & & F(A) & \xrightarrow{\sigma_A} & G(A) & \xrightarrow{\tau_A} & H(A) \\ & & \downarrow f & & \downarrow F(f) & & \downarrow G(f) & & \downarrow H(f) \\ & & B & & F(B) & \xrightarrow{\sigma_B} & G(B) & \xrightarrow{\tau_B} & H(B) \end{array}$$

**Remarque 2.2** Cette composition des transformations naturelles est souvent appelée composition *verticale*, par opposition à la composition *horizontale* dont nous n'avons pas rappelé la définition ici.

Soit un foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ . On a une transformation naturelle identité, notée  $id_F : F \rightarrow F$ , qui associe à tout objet  $A$  de  $\mathcal{C}$  la flèche  $id_{F(A)}$ . On vérifie facilement que pour tous foncteurs  $F, G : \mathcal{C} \rightarrow \mathcal{D}$ , et toute transformation naturelle  $\sigma : F \rightarrow G$ , on a

$$\begin{aligned} \sigma \cdot id_F &= \sigma; \\ id_G \cdot \sigma &= \sigma. \end{aligned}$$

On vérifie facilement que la composition des transformations naturelles est associative, donc on a une catégorie, notée  $\mathcal{D}^{\mathcal{C}}$ , dont les objets sont les foncteurs de  $\mathcal{C}$  vers  $\mathcal{D}$ , et les flèches sont les transformations naturelles.

**Définition 2.14** (Isomorphisme naturel) Soit deux catégories  $\mathcal{C}$  et  $\mathcal{D}$ . Soit deux foncteurs  $F, G : \mathcal{C} \rightarrow \mathcal{D}$ . Un isomorphisme  $\sigma : F \rightarrow G$  de la catégorie  $\mathcal{D}^{\mathcal{C}}$  est appelé *isomorphisme naturel*. Si  $\sigma : F \rightarrow G$  est un isomorphisme naturel, on dit que les foncteurs  $F$  et  $G$  sont *naturellement isomorphes*, et on note  $F \cong G$ .

**Propriété 2.2** Soit deux catégories  $\mathcal{C}$  et  $\mathcal{D}$ . Soit deux foncteurs  $F, G : \mathcal{C} \rightarrow \mathcal{D}$ . Une transformation naturelle  $\sigma : F \rightarrow G$  est un isomorphisme naturel si et seulement si pour tout objet  $A$  de  $\mathcal{C}$ ,  $\sigma_A : F(A) \rightarrow G(A)$  est un isomorphisme.

De plus,  $\sigma^{-1} : G \rightarrow F$  est la transformation naturelle qui associe à tout objet  $A$  de  $\mathcal{C}$  la flèche  $\sigma_A^{-1} : G(A) \rightarrow F(A)$  de  $\mathcal{D}$ .

*Preuve.* Notons  $\tau_A = \sigma_A^{-1}$ . On vérifie d'abord que  $\tau$  est une transformation naturelle.

$$\begin{aligned} G(f) \circ \sigma_A &= \sigma_B \circ F(f) && (\sigma \text{ transformation naturelle}) \\ \Rightarrow \sigma_B^{-1} \circ G(f) &= F(f) \circ \sigma_A^{-1} && (\sigma_A \text{ et } \sigma_B \text{ isomorphismes}) \\ \Rightarrow \tau_B \circ G(f) &= F(f) \circ \tau_A && (\text{définition de } \tau_A \text{ et } \tau_B) \end{aligned}$$

donc  $\tau : G \rightarrow F$  est une transformation naturelle.

Montrons maintenant que  $\sigma$  et  $\tau$  sont inverses. Pour tout objet  $A$  de  $\mathcal{C}$ ,

$$(\sigma \cdot \tau)_A = \sigma_A \circ \tau_A = \sigma_A \circ \sigma_A^{-1} = id_{G(A)}$$

donc  $\sigma \cdot \tau = id_G$ . De même, on montre que  $\tau \cdot \sigma = id_F$ . Par conséquent,  $\sigma^{-1} = \tau$ .  $\square$

Considérons quatre catégories  $\mathcal{B}$ ,  $\mathcal{C}$ ,  $\mathcal{D}$  et  $\mathcal{E}$ ; et quatre foncteurs  $K : \mathcal{B} \rightarrow \mathcal{C}$ ,  $F, G : \mathcal{C} \rightarrow \mathcal{D}$ , et  $H : \mathcal{D} \rightarrow \mathcal{E}$ . Considérons une transformation naturelle  $\sigma : F \rightarrow G$ .

On a une transformation naturelle

$$H\sigma : H \circ F \rightarrow H \circ G$$

telle que pour tout objet  $A$  de  $\mathcal{C}$ ,

$$(H\sigma)_A = H(\sigma_A) : H(F(A)) \rightarrow H(G(A)) \quad (\text{flèche de } \mathcal{E})$$

Cela définit bien une transformation naturelle. Soit une flèche  $f : A \rightarrow B$  de  $\mathcal{C}$ .

$$\begin{aligned} G(f) \circ \sigma_A &= \sigma_B \circ F(f) && (\sigma \text{ transformation naturelle}) \\ \Rightarrow H(G(f) \circ \sigma_A) &= H(\sigma_B \circ F(f)) \\ \Rightarrow H(G(f)) \circ H(\sigma_A) &= H(\sigma_B) \circ H(F(f)) && (H \text{ foncteur}) \\ \Rightarrow (H \circ G)(f) \circ (H\sigma)_A &= (H\sigma)_B \circ (H \circ F)(f) && (\text{définition de } H\sigma) \end{aligned}$$

On a également une transformation naturelle

$$\sigma K : F \circ K \rightarrow G \circ K$$

telle que pour tout objet  $A$  de  $\mathcal{B}$ ,

$$(\sigma K)_A = \sigma_{K(A)} : F(K(A)) \rightarrow G(K(A)) \quad (\text{flèche de } \mathcal{D})$$

Cela définit bien une transformation naturelle. En effet, soit une flèche  $f : A \rightarrow B$  de  $\mathcal{B}$ .

$$\begin{aligned} G(K(f)) \circ \sigma_{K(A)} &= \sigma_{K(B)} \circ F(K(f)) && (\sigma \text{ transformation naturelle}) \\ \Rightarrow (G \circ K)(f) \circ (\sigma K)_A &= (\sigma K)_B \circ (F \circ K)(f) && (\text{définition de } \sigma K) \end{aligned}$$

**Propriété 2.3** Soit quatre catégories  $\mathcal{B}$ ,  $\mathcal{C}$ ,  $\mathcal{D}$  et  $\mathcal{E}$ ; et quatre foncteurs  $K : \mathcal{B} \rightarrow \mathcal{C}$ ,  $F, G : \mathcal{C} \rightarrow \mathcal{D}$ , et  $H : \mathcal{D} \rightarrow \mathcal{E}$ . Supposons  $F \cong G$ . Alors,

1.  $H \circ F \cong H \circ G$ ;
2.  $F \circ K \cong G \circ K$ .

*Preuve.* Soit  $\sigma : F \rightarrow G$  l'isomorphisme naturel entre  $F$  et  $G$ .

1. Pour tout objet  $A$  de  $\mathcal{C}$ ,  $\sigma_A$  est un isomorphisme, donc, d'après la propriété 2.1,  $H(\sigma_A) = (H\sigma)_A$  est un isomorphisme. Par conséquent, d'après la propriété 2.2,  $H\sigma : H \circ F \rightarrow H \circ G$  est un isomorphisme naturel, et donc  $H \circ F \cong H \circ G$ .
2. Pour tout objet  $A$  de  $\mathcal{B}$ ,  $\sigma_{K(A)} = (\sigma K)_A$  est un isomorphisme. Par conséquent,  $\sigma K : F \circ K \rightarrow G \circ K$  est un isomorphisme naturel, et donc  $F \circ K \cong G \circ K$ .

□

## 2.2.4 Adjonctions

Un des concepts les plus importants de la théorie des catégories est l'adjonction. Par exemple les constructions *libres* (comme les constructions d'*algèbres libres*, de *groupes libres* ou de *monoïdes libres*) sont caractérisées par des adjonctions.

**Définition 2.15** (Adjonction) Soit deux catégories  $\mathcal{C}$  et  $\mathcal{D}$ . Soit deux foncteurs  $F : \mathcal{C} \rightarrow \mathcal{D}$  et  $G : \mathcal{D} \rightarrow \mathcal{C}$ . Les foncteurs  $F$  et  $G$  forment une *adjonction* notée  $(F \dashv G)$  si et seulement si il existe une transformation naturelle  $\eta : Id_{\mathcal{C}} \rightarrow G \circ F$  telle que pour tout objet  $A$  de  $\mathcal{C}$ , pour tout objet  $B$  de  $\mathcal{D}$ , et pour toute flèche  $f : A \rightarrow G(B)$  de  $\mathcal{C}$ , il existe une unique flèche  $g : F(A) \rightarrow B$  de  $\mathcal{D}$  telle que  $G(g) \circ \eta_A = f$ .

$$\begin{array}{ccc}
 A & \xrightarrow{\eta_A} & G(F(A)) & & F(A) \\
 & \searrow f & \downarrow G(g) & & \downarrow g \\
 & & G(B) & & B
 \end{array}$$

La transformation naturelle  $\eta : Id_{\mathcal{C}} \rightarrow G \circ F$  est appelée *unité* de l'adjonction  $(F \dashv G)$ .

**Théorème 2.1** Soit deux catégories  $\mathcal{C}$  et  $\mathcal{D}$ , et deux foncteurs  $F : \mathcal{C} \rightarrow \mathcal{D}$  et  $G : \mathcal{D} \rightarrow \mathcal{C}$ . Alors,  $(F \dashv G)$  est une adjonction si et seulement si il existe une transformation naturelle  $\epsilon : F \circ G \rightarrow Id_{\mathcal{D}}$  telle que pour tout objet  $B$  de  $\mathcal{D}$ , pour tout objet  $A$  de  $\mathcal{C}$  et pour toute flèche  $g : F(A) \rightarrow B$  de  $\mathcal{D}$ , il existe une unique flèche  $f : A \rightarrow G(B)$  de  $\mathcal{C}$  telle que  $\epsilon_B \circ F(f) = g$ .

$$\begin{array}{ccc}
 B & \xleftarrow{\epsilon_B} & F(G(B)) & & G(B) \\
 & \swarrow g & \uparrow F(f) & & \uparrow f \\
 & & F(A) & & A
 \end{array}$$

La transformation naturelle  $\epsilon : F \circ G \rightarrow Id_{\mathcal{D}}$  est appelée coïunité de l'adjonction  $(F \dashv G)$ .

*Preuve.* On peut trouver une preuve de ce théorème dans [BW90], page 275.  $\square$

**Propriété 2.4** Soit une adjonction  $(F \dashv G)$ , ayant pour unité la transformation naturelle  $\eta : Id_{\mathcal{C}} \rightarrow G \circ F$  et pour coïunité la transformation naturelle  $\epsilon : F \circ G \rightarrow Id_{\mathcal{D}}$ . Alors, on a les égalités suivantes entre transformations naturelles :

1.  $G\epsilon \cdot \eta G = id_G$  ;
2.  $\epsilon F \cdot F\eta = id_F$ .

*Preuve.* On peut trouver une preuve de cette propriété dans [McL71], page 80.  $\square$

**Exemple 2.1** (Adjonction entre la catégorie **Graph** et la catégorie **Cat**)

À toute petite catégorie  $\mathcal{C}$ , on peut associer un graphe  $U(\mathcal{C})$ , dont les nœuds sont les objets de  $\mathcal{C}$ , et les arcs les flèches de  $\mathcal{C}$ . Réciproquement, à tout graphe  $\alpha^\Phi$ , on peut associer la catégorie  $F(\alpha^\Phi)$  librement engendrée sur le graphe  $\alpha^\Phi$  (cf. remarque 2.1).

$F$  et  $U$  sont en fait des foncteurs

$$\begin{array}{l}
 F : \mathbf{Graph} \rightarrow \mathbf{Cat} \\
 U : \mathbf{Cat} \rightarrow \mathbf{Graph}
 \end{array}$$

qui forment une adjonction  $(F \dashv U)$ . En effet, pour toute catégorie  $\mathcal{C}$ , pour tout graphe  $\alpha^\Phi$ , pour tout morphisme de graphes  $m : \alpha^\Phi \rightarrow U(\mathcal{C})$ , il existe une unique extension de  $m$  en un foncteur  $M : F(\alpha^\Phi) \rightarrow \mathcal{C}$ .

Par abus de notation, dans toute la suite la catégorie librement engendrée sur un graphe  $\alpha^\Phi$  sera également notée  $\alpha^\Phi$ .

## 2.3 Diagrammes

Dans cette section, nous définissons les diagrammes sur une catégorie de base  $\mathcal{C}$ . Intuitivement, un diagramme décrit un assemblage d'objets de  $\mathcal{C}$  à l'aide de flèches de  $\mathcal{C}$ . Nous rappelons la définition de la *colimite* d'un diagramme, qui peut être vue intuitivement comme le résultat de l'assemblage décrit par le diagramme.

Nous avons besoin de définir non seulement des diagrammes sur  $\mathcal{C}$ , mais également des morphismes de diagrammes, afin d'obtenir une *catégorie* de diagrammes. En effet, l'idée principale de notre travail est d'associer à toute spécification un diagramme, et à tout morphisme de spécifications un morphisme de diagrammes.

Dans cette section, nous commençons par définir la catégorie  $\text{DIAGR}(\mathcal{C})$ , qui a pour objets les *diagrammes* et pour flèches les *morphismes de diagrammes*. Notre définition de morphisme de diagrammes, basée sur les notions de *zigzag* sur un graphe et de *morphisme généralisé de graphe* est plus générale que celle par exemple donnée dans [TBG91]. Puis, nous reformulons la définition de cône et de cône colimite.

Nous définissons ensuite une congruence  $\approx$  sur les flèches de  $\text{DIAGR}(\mathcal{C})$  et la catégorie  $\text{diagr}(\mathcal{C})$  est le quotient de  $\text{DIAGR}(\mathcal{C}_0)$  par cette relation. La catégorie  $\text{diagr}(\mathcal{C}_0)$  a pour objets les objets de  $\text{DIAGR}(\mathcal{C})$ , et pour flèches les classes d'équivalence modulo  $\approx$  de flèches de  $\text{DIAGR}(\mathcal{C})$ . Nous montrons que la catégorie  $\text{diagr}(\mathcal{C})$  est finiment cocomplète et que le calcul d'une colimite consiste à *aplatir* un diagramme de diagrammes dans la catégorie  $\text{DIAGR}(\mathcal{C})$ . Nous montrons enfin que  $\text{diagr}(\mathcal{C})$  est une complétion de  $\mathcal{C}$  par colimites finies.

### 2.3.1 Catégorie des diagrammes $\text{DIAGR}(\mathcal{C})$

Un diagramme  $\bar{\alpha}$  sur une catégorie  $\mathcal{C}$  est un graphe  $\alpha^\Phi$ , dont les nœuds  $n$  sont étiquetés par des objets  $\alpha(n)$  de  $\mathcal{C}$ , et les arcs  $a : n \rightarrow n'$  sont étiquetés par des flèches  $\alpha(a) : \alpha(n) \rightarrow \alpha(n')$  de  $\mathcal{C}$ . De manière équivalente :

**Définition 2.16** (Diagramme) Un diagramme sur une catégorie  $\mathcal{C}$  est un couple

$$\bar{\alpha} = (\alpha^\Phi, \alpha : \alpha^\Phi \rightarrow \mathcal{C}),$$

avec :

- $\alpha^\Phi$  est une *catégorie librement engendrée* sur un graphe (ce graphe est également noté  $\alpha^\Phi$ );
- $\alpha : \alpha^\Phi \rightarrow \mathcal{C}$  est un foncteur.

Le graphe  $\alpha^\Phi$  est appelé *graphe sous-jacent* du diagramme  $\bar{\alpha}$ .

Un diagramme  $\bar{\alpha}$  est fini si et seulement si son graphe sous-jacent  $\alpha^\Phi$  est fini.

**Remarque 2.3**

1. S. Mac Lane [McL71] définit un diagramme comme un foncteur  $F : J \rightarrow \mathcal{C}$ , où  $J$  est une catégorie quelconque (“habituellement petite et souvent finie”). Notre définition est donc moins générale, dans la mesure où nous imposons que la catégorie  $\alpha^\Phi$  soit librement engendrée sur un graphe, donc petite. De plus la catégorie  $J$  peut contenir des égalités entre certaines flèches, ce qui n’est pas le cas pour une catégorie librement engendrée sur un graphe.
2. M. Barr et C. Wells [BW90] définissent un diagramme comme un morphisme de graphes entre le graphe  $\alpha^\Phi$  et le graphe sous-jacent de la catégorie  $\mathcal{C}$ . Leur définition est donc un peu moins générale que celle que nous utilisons, puisqu’ils imposent que la catégorie  $\mathcal{C}$  soit petite.
3. Lorsque le graphe  $\alpha^\Phi$  est fini, cela n’implique pas forcément que la catégorie  $\alpha^\Phi$  soit finie. En effet, si le graphe comporte une ou plusieurs boucles, c’est-à-dire des arcs qui ont mêmes source et but, alors la catégorie librement engendrée sur  $\alpha^\Phi$  comporte un nombre infini de flèches. La définition de diagramme fini donnée par S. Mac Lane, qui consiste à considérer un foncteur  $\alpha : \alpha^\Phi \rightarrow \mathcal{C}$  où la catégorie  $\alpha^\Phi$  est finie, exclut donc les diagrammes dont le graphe sous-jacent comporte des boucles.
4. Nous définissons un diagramme à partir d’un graphe parce que nous pensons que cette définition est plus proche de l’informatique: on a au départ, de façon effective, un graphe, et le fait de considérer la catégorie engendrée par ce graphe doit être vu comme un simple outil de formalisation.
5. Pour définir un foncteur  $\alpha : \alpha^\Phi \rightarrow \mathcal{C}$  où  $\alpha^\Phi$  est une catégorie librement engendrée sur un graphe, il suffit de définir  $\alpha$  sur les nœuds et les arcs du *graphe*  $\alpha^\Phi$ . Comme les flèches de la catégorie  $\alpha^\Phi$  sont les chaînes d’arcs composables sur le graphe  $\alpha^\Phi$ , il suffit de poser

$$\begin{aligned}\alpha(\varepsilon_n) &= id_{\alpha(n)} \\ \alpha(a_1 a_2 \dots a_n) &= \alpha(a_n) \circ \dots \circ \alpha(a_2) \circ \alpha(a_1)\end{aligned}$$

pour terminer la définition du foncteur  $\alpha : \alpha^\Phi \rightarrow \mathcal{C}$ .

6. Lorsque nous définissons un diagramme  $\bar{\alpha}$ , nous ne tenons pas compte des noms des nœuds et des arcs du graphe sous-jacent  $\alpha^\Phi$ . Autrement dit, le graphe  $\alpha^\Phi$  est considéré à *un isomorphisme près*. Plus précisément, soit deux diagrammes

$$\bar{\alpha} = (\alpha^\Phi, \alpha : \alpha^\Phi \rightarrow \mathcal{C}) \quad \text{et} \quad \bar{\beta} = (\beta^\Phi, \beta : \beta^\Phi \rightarrow \mathcal{C}).$$

Si on a un isomorphisme de graphes  $\sigma^\Phi : \alpha^\Phi \rightarrow \beta^\Phi$  (qui correspond à un foncteur entre les catégories  $\alpha^\Phi$  et  $\beta^\Phi$ ) tel que

$$\begin{aligned}\alpha &= \beta \circ \sigma^\Phi \\ \Leftrightarrow \quad &\begin{cases} \forall n \in \text{Nœud}(\alpha^\Phi), \alpha(n) = \beta(\sigma^\Phi(n)) \\ \forall a : n \rightarrow n' \in \text{Arc}(\alpha^\Phi), \alpha(a) = \beta(\sigma^\Phi(a)) \end{cases}\end{aligned}$$

alors, nous considérons que les diagrammes  $\bar{\alpha}$  et  $\bar{\beta}$  sont égaux.

Nous donnons maintenant quelques exemples de diagrammes sur une catégorie  $\mathcal{C}$ .

**Exemple 2.2** (Diagramme  $I_{\mathcal{C}}(A)$ ) Soit  $1^{\Phi}$  le graphe comportant un seul nœud, appelé  $*$ , et aucun arc. Soit  $A$  un objet de  $\mathcal{C}$ . On a un foncteur  $I_{\mathcal{C}}^A : 1^{\Phi} \rightarrow \mathcal{C}$  défini par  $I_{\mathcal{C}}^A(*) = A$ . On peut donc définir un diagramme

$$I_{\mathcal{C}}(A) = (1^{\Phi}, I_{\mathcal{C}}^A : 1^{\Phi} \rightarrow \mathcal{C}).$$

graphe  $1^{\Phi}$ diagramme  $I_{\mathcal{C}}(A)$ 

**Exemple 2.3** (Diagramme de somme) Soit  $2^{\Phi}$  le graphe comportant deux nœuds, et aucun arc. Soit  $A$  et  $B$  deux objets de  $\mathcal{C}$ . On a un foncteur  $\alpha : 2^{\Phi} \rightarrow \mathcal{C}$ , qui associe l'un des nœuds à  $A$  et l'autre des nœuds à  $B$ . On a donc un *diagramme de somme de  $A$  et  $B$*

$$\bar{\alpha} = (2^{\Phi}, \alpha : 2^{\Phi} \rightarrow \mathcal{C}).$$

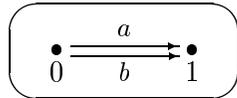
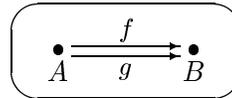
graphe  $2^{\Phi}$ diagramme  $\bar{\alpha}$ 

**Exemple 2.4** (Diagramme de coégalisateur) Soit  $\kappa^{\Phi}$  le graphe contenant deux nœuds 0 et 1, et deux arcs  $a$  et  $b$  de source 0 et de but 1. Si  $A$  et  $B$  sont deux objets de  $\mathcal{C}$ , et  $f, g : A \rightarrow B$  deux flèches de  $\mathcal{C}$ , on a *diagramme de coégalisateur*

$$\bar{\kappa} = (\kappa^{\Phi}, \kappa : \kappa^{\Phi} \rightarrow \mathcal{C})$$

défini par

$$\begin{aligned} \kappa(0) &= A \\ \kappa(1) &= B \\ \kappa(a) &= f \\ \kappa(b) &= g. \end{aligned}$$

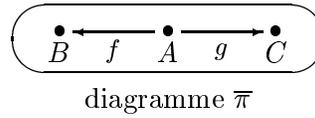
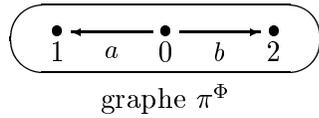
graphe  $\kappa^{\Phi}$ diagramme  $\bar{\kappa}$

**Exemple 2.5** (Diagramme de somme amalgamée) Considérons le graphe  $\pi^\Phi$ , appelé “graphe de somme amalgamée”, qui comporte trois nœuds 0, 1 et 2, et deux arcs  $a : 0 \rightarrow 1$  et  $b : 0 \rightarrow 2$ . Si  $A, B, C$  sont des objets de  $\mathcal{C}$  et  $f : A \rightarrow B, g : A \rightarrow C$  des flèches de  $\mathcal{C}$ , on a un *diagramme de somme amalgamée*

$$\bar{\pi} = (\pi^\Phi, \pi : \pi^\Phi \rightarrow \mathcal{C})$$

défini par

$$\begin{aligned}\pi(0) &= A \\ \pi(1) &= B \\ \pi(2) &= C \\ \pi(a) &= f \\ \pi(b) &= g.\end{aligned}$$



### Morphismes de diagrammes

Nous venons de définir les diagrammes finis. Pour obtenir une structure de catégorie, nous devons ajouter des flèches entre les diagrammes. Intuitivement, une flèche entre deux diagrammes  $\bar{\alpha}$  et  $\bar{\beta}$  sur  $\mathcal{C}$  est un ensemble de flèches de  $\mathcal{C}$ , une par nœud de  $\alpha^\Phi$ , qui ont certaines propriétés de commutation.

Différentes catégories de diagrammes ont été présentées dans la littérature. Par exemple, S. Mac Lane définit une catégorie de diagrammes, qu’il appelle “super-comma catégorie” ([McL71], exercice 5.(b), page 111). A. Tarlecki *et al.* ont défini une catégorie **Func**t( $\mathcal{C}$ ) [TBG91], qui peut être obtenue à partir de la “super-comma catégorie” de S. Mac Lane en dualisant. Dans cette catégorie, une flèche entre deux diagrammes

$$\bar{\alpha} = (\alpha^\Phi, \alpha : \alpha^\Phi \rightarrow \mathcal{C}) \text{ et } \bar{\beta} = (\beta^\Phi, \beta : \beta^\Phi \rightarrow \mathcal{C})$$

est un couple

$$\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta} = (\sigma^\Phi : \alpha^\Phi \rightarrow \beta^\Phi ; \sigma : \alpha \rightarrow \beta \circ \sigma^\Phi)$$

où

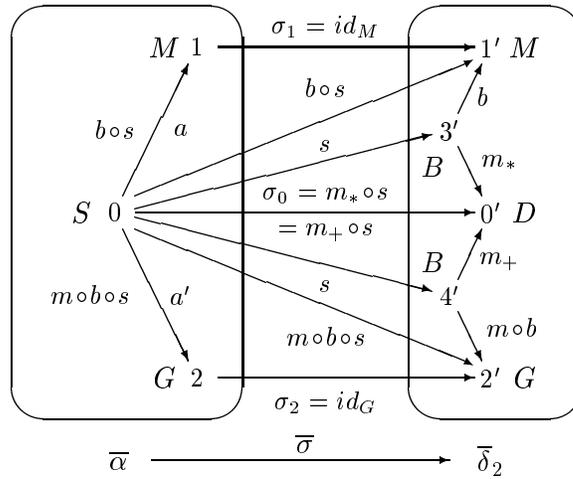
- $\sigma^\Phi$  est un foncteur entre les catégories  $\alpha^\Phi$  et  $\beta^\Phi$ , ou, de façon équivalente, un morphisme de graphes entre les graphes  $\alpha^\Phi$  et  $\beta^\Phi$  ;
- $\sigma : \alpha \rightarrow \beta \circ \sigma^\Phi$  est une transformation naturelle entre les foncteurs  $\alpha : \alpha^\Phi \rightarrow \mathcal{C}$  et  $\beta \circ \sigma^\Phi : \alpha^\Phi \rightarrow \mathcal{C}$ .

Ces catégories de diagrammes ne sont pas suffisamment générales pour notre propos, car elles ne comportent pas assez de flèches. Intuitivement, et en anticipant un peu sur la suite, nous souhaitons avoir une flèche entre deux diagrammes chaque fois qu'il existe une flèche entre les *colimites* des deux diagrammes. Avec la définition ci-dessus, certaines flèches entre colimites de diagrammes ne correspondent à aucune flèche entre les deux diagrammes.

**Exemple 2.6** Reprenons l'exemple des anneaux du chapitre 1. Nous avons un morphisme de diagrammes du diagramme  $\bar{\alpha}$  vers le diagramme  $\bar{\delta}_2$  qui correspond à un morphisme de spécifications de  $\text{Colim } \bar{\alpha}$  vers  $\text{Colim } \bar{\delta}_2$  :

$$\text{up}(S, M, G, A_2, b \circ s, m \circ b \circ s, \&_1(A_2) \circ \&_1(MD), \&_2(A_2)).$$

Mais il n'y a pas de morphisme de diagrammes de  $\bar{\alpha}$  vers  $\bar{\delta}_2$  avec la définition ci-dessus.



Pour cette raison, au lieu de considérer un morphisme de graphes  $\sigma^\Phi : \alpha^\Phi \rightarrow \beta^\Phi$  nous devons considérer un *morphisme généralisé* de graphes  $\sigma^\Phi : \alpha^\Phi \rightsquigarrow \beta^\Phi$ . Rappelons qu'un morphisme généralisé de graphes associe à tout nœud de  $\alpha^\Phi$  un nœud de  $\beta^\Phi$ , et à tout arc de  $\alpha^\Phi$  un *zigzag* de  $\beta^\Phi$ , c'est-à-dire une séquence linéaire d'arcs de  $\beta^\Phi$  (définition 2.5). De même, au lieu de considérer une transformation naturelle  $\sigma : \alpha \rightarrow \beta \circ \sigma^\Phi$ , nous devons considérer une *transformation naturelle généralisée*  $\sigma : \alpha \rightsquigarrow \beta \circ \sigma^\Phi$ .

**Exemple 2.6** (Suite) Avec cette définition, nous pouvons définir une flèche  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\delta}_2$ , qui consiste en un morphisme généralisé de graphes  $\sigma^\Phi : \alpha^\Phi \rightsquigarrow \delta_2^\Phi$  et une transformation naturelle généralisée  $\sigma : \alpha \rightsquigarrow \delta_2 \circ \sigma^\Phi$ .

- Le morphisme généralisé de graphes  $\sigma^\Phi$  peut être défini par exemple de la façon suivante :

$$\begin{array}{rcl}
\sigma^\Phi & : & \alpha^\Phi \rightsquigarrow \delta_2^\Phi \\
& & 0 \mapsto 0' \\
& & 1 \mapsto 1' \\
& & 2 \mapsto 2' \\
& & a \mapsto 0' \xleftarrow{m_*} 3' \xrightarrow{b} 1' \\
& & a' \mapsto 0' \xleftarrow{m_+} 4' \xrightarrow{m \circ b} 2'
\end{array}$$

- La transformation naturelle généralisée  $\sigma : \alpha \rightsquigarrow \delta_2 \circ \sigma^\Phi$  est définie par :

$$\begin{aligned}
\sigma_0 &= m_* \circ s = m_+ \circ s \\
\sigma_1 &= id_M \\
\sigma_2 &= id_G.
\end{aligned}$$

**Définition 2.17** (Relation de connexion entre deux flèches)

Soit un diagramme  $\bar{\delta} = (\delta^\Phi, \delta : \delta^\Phi \rightarrow \mathcal{C})$ , et deux nœuds  $n_0$  et  $n_k$  de  $\text{Nœud}(\delta^\Phi)$ .

Soit  $u : A \rightarrow \delta(n_0)$  et  $v : A \rightarrow \delta(n_k)$  deux flèches de  $\mathcal{C}$ .

On dit que les flèches  $u$  et  $v$  sont *connectées par le diagramme  $\bar{\delta}$*  (figure 2.1) si et seulement si il existe un zigzag

$$Z = n_0 \xrightarrow{a_0} n_1 \xleftarrow{a_1} n_2 \xleftarrow{a_2} n_2 \cdots n_{k-1} \xrightarrow{a_{k-1}} n_k$$

sur  $\delta^\Phi$ , et un ensemble de flèches de  $\mathcal{C}$

$$\{c_i : A \rightarrow \delta(n_i) ; i \in \{0, \dots, k\}\},$$

tels que :

- $u = c_0$  ;
- $v = c_k$  ;
- $\forall i \in \{0, \dots, k-1\}$  :  
 $\delta(a_i) \circ c_i = c_{i+1}$ , si  $a_i$  est orienté de  $n_i$  vers  $n_{i+1}$  ;  
 $\delta(a_i) \circ c_{i+1} = c_i$ , si  $a_i$  est orienté de  $n_{i+1}$  vers  $n_i$ .

On note  $u \sim_{\bar{\delta}} v$  ; ou  $u \sim_{\bar{\delta}} v [Z]$ , si l'on souhaite préciser le zigzag  $Z$ .

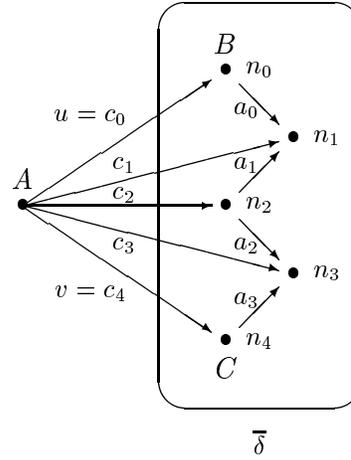
**Définition 2.18** (Catégorie des diagrammes finis  $\text{DIAGR}(\mathcal{C})$ )

Soit une catégorie  $\mathcal{C}$ .

- Un objet  $\bar{\delta}$  de  $\text{DIAGR}(\mathcal{C})$  est un diagramme fini sur  $\mathcal{C}$ .
- Soit deux diagrammes  $\bar{\alpha} = (\alpha^\Phi, \alpha : \alpha^\Phi \rightarrow \mathcal{C})$  et  $\bar{\beta} = (\beta^\Phi, \beta : \beta^\Phi \rightarrow \mathcal{C})$ . Une flèche  $\bar{\tau} : \bar{\alpha} \rightarrow \bar{\beta}$  de  $\text{DIAGR}(\mathcal{C})$ , ou *morphisme de diagrammes*, est un couple

$$\bar{\tau} : \bar{\alpha} \rightarrow \bar{\beta} = (\tau^\Phi : \alpha^\Phi \rightsquigarrow \beta^\Phi, \tau : \alpha \rightsquigarrow \beta \circ \tau^\Phi),$$

avec :

Figure 2.1:  $u \sim_{\bar{\delta}} v$  [ $Z : n_0 \rightsquigarrow n_4$ ]

- $\tau^\Phi : \alpha^\Phi \rightsquigarrow \beta^\Phi$  est un morphisme généralisé de graphes.
- $\tau : \alpha \rightsquigarrow \beta \circ \tau^\Phi$  est une “transformation naturelle généralisée”, c’est-à-dire un ensemble de flèches

$$\tau_n : \alpha(n) \rightarrow \beta(\tau^\Phi(n)), \forall n \in \text{Nœud}(\alpha^\Phi)$$

telles que

$$\forall a : m \rightarrow n \in \text{Arc}(\alpha^\Phi), \tau_n \circ \alpha(a) \sim_{\bar{\beta}} \tau_m [\tau^\Phi(a)].$$

Remarquons que dans le cas particulier où  $\tau^\Phi$  est un morphisme de graphes, alors  $\tau^\Phi$  s’étend en un foncteur  $\tau^\Phi : \alpha^\Phi \rightarrow \beta^\Phi$ , et  $\tau$  est alors une transformation naturelle  $\tau : \alpha \rightsquigarrow \beta \circ \tau^\Phi$ .

- On définit la composition de deux flèches de  $\text{DIAGR}(\mathcal{C})$  de la façon suivante :

Soit trois diagrammes :

$$\begin{aligned} \bar{\alpha} &= (\alpha^\Phi, \alpha : \alpha^\Phi \rightarrow \mathcal{C}); \\ \bar{\beta} &= (\beta^\Phi, \beta : \beta^\Phi \rightarrow \mathcal{C}); \\ \bar{\gamma} &= (\gamma^\Phi, \gamma : \gamma^\Phi \rightarrow \mathcal{C}). \end{aligned}$$

Soit deux flèches :

$$\begin{aligned} \bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta} &= (\sigma^\Phi : \alpha^\Phi \rightsquigarrow \beta^\Phi, \sigma : \alpha \rightsquigarrow \beta \circ \sigma^\Phi); \\ \bar{\tau} : \bar{\beta} \rightarrow \bar{\gamma} &= (\tau^\Phi : \beta^\Phi \rightsquigarrow \gamma^\Phi, \tau : \beta \rightsquigarrow \gamma \circ \tau^\Phi). \end{aligned}$$

La composition  $\bar{\tau} \circ \bar{\sigma}$  de  $\bar{\sigma}$  et  $\bar{\tau}$  est le couple

$$\bar{\tau} \circ \bar{\sigma} : \bar{\alpha} \rightarrow \bar{\gamma} = (\tau^\Phi \circ \sigma^\Phi : \alpha^\Phi \rightsquigarrow \gamma^\Phi, \lambda : \alpha \rightsquigarrow \gamma \circ \tau^\Phi \circ \sigma^\Phi)$$

où

- $\tau^\Phi \circ \sigma^\Phi$  est la composition des deux morphismes de graphes généralisés ;
- $\lambda : \alpha \rightsquigarrow \gamma \circ \tau^\Phi \circ \sigma^\Phi$  est la *transformation naturelle généralisée* définie par

$$\forall n \in \text{Nœud}(\alpha^\Phi), \lambda_n = \tau_{\sigma^\Phi(n)} \circ \sigma_n.$$

$$\alpha(n) \xrightarrow{\sigma_n} \beta(\sigma^\Phi(n)) \xrightarrow{\tau_{\sigma^\Phi(n)}} \gamma(\tau^\Phi(\sigma^\Phi(n)))$$

On vérifie sans difficulté que  $\lambda$  est bien une transformation naturelle généralisée, c'est-à-dire que

$$\forall a : m \rightarrow n \in \text{Arc}(\alpha^\Phi), \tau_{\sigma^\Phi(n)} \circ \sigma_n \circ \alpha(a) \sim_{\overline{\gamma}} \tau_{\sigma^\Phi(m)} \circ \sigma_m [\tau^\Phi(\sigma^\Phi(a))].$$

- On vérifie que cette composition est associative.
- Pour tout diagramme  $\overline{\alpha}$ , on a une flèche identité

$$\overline{id_{\overline{\alpha}}} : \overline{\alpha} \rightarrow \overline{\alpha} = (id_{\alpha^\Phi} : \alpha^\Phi \rightsquigarrow \alpha^\Phi, id_{\overline{\alpha}} : \alpha \rightsquigarrow \alpha),$$

où  $id_{\overline{\alpha}} : \alpha \rightsquigarrow \alpha$  est la transformation naturelle identité  $id_\alpha : \alpha \rightarrow \alpha$ . Autrement dit,  $\forall n \in \text{Nœud}(\alpha^\Phi), (id_{\overline{\alpha}})_n = id_{\alpha(n)}$ .

### 2.3.2 Colimites

La façon standard de définir une colimite (comme par exemple dans [McL71]) est de commencer par fixer un graphe sous-jacent  $\alpha^\Phi$ , puis de définir un foncteur diagonal

$$\Delta : \mathcal{C} \rightarrow \mathcal{C}^{\alpha^\Phi},$$

où  $\mathcal{C}^{\alpha^\Phi}$  est la catégorie des foncteurs de  $\alpha^\Phi$  vers  $\mathcal{C}$ , qui correspond à une catégorie de diagrammes de graphe sous-jacent  $\alpha^\Phi$ . Cette présentation ne nous convient pas, parce que nous ne voulons pas fixer de graphe *a priori*. En définissant la catégorie  $\text{DIAGR}(\mathcal{C})$ , nous n'avons pas non plus fixé de graphe sous-jacent, puisqu'un diagramme est donné à la fois par un *graphe* et un *foncteur*.

Dans ce paragraphe, nous reformulons la définition de colimite dans notre cadre. Cette définition correspond à la définition standard, et évite de fixer trop tôt un graphe sous-jacent.

**Définition 2.19** (Foncteur d'inclusion  $I_{\mathcal{C}} : \mathcal{C} \rightarrow \text{DIAGR}(\mathcal{C})$ )

La catégorie  $\mathcal{C}$  peut être plongée dans  $\text{DIAGR}(\mathcal{C})$  en considérant tout objet de  $\mathcal{C}$  comme un diagramme sur le graphe  $1^\Phi$ . Autrement dit, on a un foncteur

$$\begin{array}{lcl} I_{\mathcal{C}} : & \mathcal{C} & \rightarrow \text{DIAGR}(\mathcal{C}) \\ & A & \mapsto \begin{array}{c} \bullet \\ \text{---} \\ \bullet \end{array} A \\ & f : A \rightarrow B & \mapsto \begin{array}{c} \bullet \quad \bullet \\ \text{---} \quad \text{---} \\ \bullet \quad \bullet \end{array} \end{array}$$

Le foncteur correspondant au diagramme  $I_C(A)$  est noté  $I_C^A : 1^\Phi \rightarrow \mathcal{C}$ .

La transformation naturelle généralisée correspondant au morphisme de diagrammes  $I_C(f)$  est notée  $I_C^f : I_C^A \rightsquigarrow I_C^B$ .

Nous avons donc :

$$\begin{aligned} I_C(A) &= (1^\Phi, I_C^A : 1^\Phi \rightarrow \mathcal{C}) \\ I_C(B) &= (1^\Phi, I_C^B : 1^\Phi \rightarrow \mathcal{C}) \\ I_C(f) &= (id_{1^\Phi}, I_C^f : I_C^A \rightsquigarrow I_C^B). \end{aligned}$$

Soit  $*$  l'unique nœud de  $1^\Phi$ . Nous avons :

$$\begin{aligned} I_C^A(*) &= A \\ I_C^B(*) &= B \\ (I_C^f)_* &= f : A \rightarrow B. \end{aligned}$$

**Définition 2.20** (Cône) Soit un diagramme  $\bar{\alpha} = (\alpha^\Phi, \alpha : \alpha^\Phi \rightarrow \mathcal{C})$ . Un *cône*<sup>1</sup> sur  $\bar{\alpha}$  est un couple

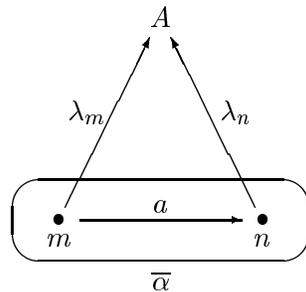
$$(A, \bar{\lambda} : \bar{\alpha} \rightarrow I_C(A))$$

où  $A$  est un objet de  $\mathcal{C}$ , et  $\bar{\lambda}$  est une flèche entre les diagrammes  $\bar{\alpha}$  et  $I_C(A)$ .

$\lambda^\Phi$  est l'unique morphisme généralisé de  $\alpha^\Phi$  vers  $1^\Phi$  : à tout nœud de  $\alpha^\Phi$ ,  $\lambda^\Phi$  associe l'unique nœud  $*$  de  $1^\Phi$ , et à tout arc de  $\alpha^\Phi$ ,  $\lambda^\Phi$  associe le zigzag de longueur nulle  $0_* : * \rightsquigarrow *$ . Remarquons que  $0_* : * \rightsquigarrow *$  est l'unique zigzag de  $1^\Phi$ .

**Remarque 2.4** La transformation naturelle généralisée  $\lambda : \alpha \rightsquigarrow I_C^A \circ \lambda^\Phi$  est en fait une transformation naturelle  $\lambda : \alpha \rightarrow I_C^A \circ \lambda^\Phi$ . En détaillant, on a

$$\forall m, n \in \text{Nœud}(\alpha^\Phi), \forall a : m \rightarrow n \in \text{Arc}(\alpha^\Phi), \lambda_n \circ \alpha(a) = \lambda_m.$$



**Définition 2.21** (Cône colimite)

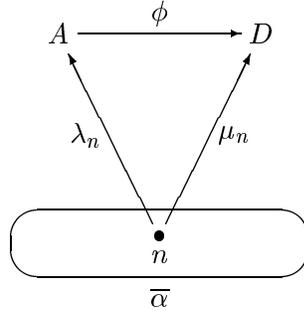
Le cône  $(A, \bar{\lambda} : \bar{\alpha} \rightarrow I_C(A))$  est un *cône colimite* du diagramme  $\bar{\alpha}$  si et seulement si pour tout cône  $(D, \bar{\mu} : \bar{\alpha} \rightarrow I_C(D))$ , il existe une unique flèche  $\phi : A \rightarrow D$  de  $\mathcal{C}$  telle que

$$I_C(\phi) \circ \bar{\lambda} = \bar{\mu}.$$

<sup>1</sup>Pour respecter la terminologie usuelle, nous devrions parler de *cône inductif* ou de *cocône*, car nous allons définir des *colimites* et non des *limites*. Dans la mesure où nous ne parlons ici jamais de limites, nous utiliserons le mot *cône*.

Détaillons un peu cette dernière égalité.

$$\begin{aligned}
I_{\mathcal{C}}(\phi) \circ \bar{\lambda} &= \bar{\mu} \\
\Leftrightarrow \forall n \in \text{Nœud}(\alpha^{\Phi}), I_{\mathcal{C}}(\phi)_{\lambda^{\Phi}(n)} \circ \lambda_n &= \mu_n \\
\Leftrightarrow \forall n \in \text{Nœud}(\alpha^{\Phi}), \phi \circ \lambda_n &= \mu_n
\end{aligned}$$



Si  $(A, \bar{\lambda} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(A))$  est un cône colimite, on appelle  $A$  la *colimite* du diagramme  $\bar{\alpha}$ , et on note  $A = \text{Colim}_{\mathcal{C}} \bar{\alpha}$ . Cette notation compacte ne doit pas faire oublier qu'en réalité, une colimite  $A$  est toujours définie en même temps qu'un cône, c'est-à-dire une flèche  $\bar{\lambda} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(A)$ .

Les deux lemmes suivants montrent qu'une colimite est définie à un isomorphisme près.

**Lemme 2.1** (*Deux colimites d'un même diagramme sont isomorphes*)

Soit un diagramme  $\bar{\alpha}$  sur  $\mathcal{C}$ . Si  $(A, \bar{\lambda} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(A))$  et  $(B, \bar{\mu} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(B))$  sont deux cônes colimites du diagramme  $\bar{\alpha}$ , alors  $A \cong B$ .

*Preuve.* Comme  $(B, \bar{\mu} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(B))$  est un cône sur  $\bar{\alpha}$ , il existe une unique flèche  $\phi : A \rightarrow B$  telle que

$$I_{\mathcal{C}}(\phi) \circ \bar{\lambda} = \bar{\mu} \tag{i}$$

Comme  $(A, \bar{\lambda} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(A))$  est un cône sur  $\bar{\alpha}$ , il existe une unique flèche  $\psi : B \rightarrow A$  telle que

$$I_{\mathcal{C}}(\psi) \circ \bar{\mu} = \bar{\lambda} \tag{ii}$$

Nous avons donc

$$\begin{aligned}
I_{\mathcal{C}}(\psi \circ \phi) \circ \bar{\lambda} &= I_{\mathcal{C}}(\psi) \circ I_{\mathcal{C}}(\phi) \circ \bar{\lambda} && (I_{\mathcal{C}} \text{ foncteur}) \\
&= I_{\mathcal{C}}(\psi) \circ \bar{\mu} && (\text{d'après (i)}) \\
&= \bar{\lambda} && (\text{d'après (ii)})
\end{aligned}$$

Comme  $(A, \bar{\lambda} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(A))$  est un cône colimite, il existe une unique flèche  $u : A \rightarrow A$  telle que  $I_{\mathcal{C}}(u) \circ \bar{\lambda} = \bar{\lambda}$ . Or  $\psi \circ \phi$  et  $id_A$  satisfont toutes les deux cette propriété, donc  $\psi \circ \phi = id_A$ . De même, on montre que  $\phi \circ \psi = id_B$ . Par conséquent,  $A$  et  $B$  sont isomorphes.  $\square$

**Lemme 2.2** Soit un diagramme  $\bar{\alpha}$  sur  $\mathcal{C}$ , ayant pour cône colimite

$$(A, \bar{\lambda} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(A)).$$

Soit  $B$  un objet de  $\mathcal{C}$  isomorphe à  $A$ . On note  $\phi : A \rightarrow B$  cet isomorphisme. Alors  $(B, I_{\mathcal{C}}(\phi) \circ \bar{\lambda} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(B))$  est un (autre) cône colimite du diagramme  $\bar{\alpha}$ .

*Preuve.* Il faut montrer que pour tout cône  $(D, \bar{\mu} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(D))$ , il existe un unique  $\psi : B \rightarrow D$  tel que  $I_{\mathcal{C}}(\psi) \circ I_{\mathcal{C}}(\phi) \circ \bar{\lambda} = \bar{\mu}$ .

*Existence.* Soit  $\xi : A \rightarrow D$  l'unique flèche telle que  $I_{\mathcal{C}}(\xi) \circ \bar{\lambda} = \bar{\mu}$ . On pose

$$\psi = \xi \circ \phi^{-1}.$$

$$\begin{aligned} I_{\mathcal{C}}(\psi) \circ I_{\mathcal{C}}(\phi) \circ \bar{\lambda} &= I_{\mathcal{C}}(\xi \circ \phi^{-1}) \circ I_{\mathcal{C}}(\phi) \circ \bar{\lambda} && \text{(définition de } \psi) \\ &= I_{\mathcal{C}}(\xi) \circ \bar{\lambda} && (I_{\mathcal{C}} \text{ foncteur}) \\ &= \bar{\mu} && \text{(définition de } \xi) \end{aligned}$$

*Unicité.* Soit  $\psi' : B \rightarrow D$  tel que  $I_{\mathcal{C}}(\psi') \circ I_{\mathcal{C}}(\phi) \circ \bar{\lambda} = \bar{\mu}$ . On montre que  $\psi = \psi'$ .

$$\begin{aligned} I_{\mathcal{C}}(\psi) \circ I_{\mathcal{C}}(\phi) \circ \bar{\lambda} &= I_{\mathcal{C}}(\psi') \circ I_{\mathcal{C}}(\phi) \circ \bar{\lambda} = \bar{\mu} && \text{(définitions de } \psi \text{ et } \psi') \\ \Rightarrow I_{\mathcal{C}}(\psi \circ \phi) \circ \bar{\lambda} &= I_{\mathcal{C}}(\psi' \circ \phi) \circ \bar{\lambda} = \bar{\mu} && (I_{\mathcal{C}} \text{ foncteur}) \end{aligned}$$

Comme  $(A, \bar{\lambda} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(A))$  est cône colimite du diagramme  $\bar{\alpha}$ , il existe une unique flèche  $u : A \rightarrow D$  de  $\mathcal{C}$  telle que

$$I_{\mathcal{C}}(u) \circ \bar{\lambda} = \bar{\mu}.$$

Comme  $\psi \circ \phi$  et  $\psi' \circ \phi$  satisfont cette propriété,

$$\psi \circ \phi = \psi' \circ \phi.$$

Or  $\phi$  est un isomorphisme, donc  $\psi = \psi'$ . □

**Remarque 2.5** (Choix de colimites)

Une colimite (avec le cône colimite correspondant) est donc définie à un isomorphisme près. On parle souvent abusivement de *la* colimite d'un diagramme, alors qu'on devrait parler d'*une* colimite d'un diagramme. Pour la suite, nous avons besoin de distinguer une colimite particulière pour un diagramme. Cela consiste à *choisir* un cône colimite privilégié pour ce diagramme. Dans ce cas, nous pouvons alors parler sans ambiguïté de *la* colimite de ce diagramme.

Lorsqu'on n'a pas fait de choix de colimites, la notation  $\text{Colim}_{\mathcal{C}} \bar{\alpha}$  représente donc *une colimite quelconque* du diagramme  $\bar{\alpha}$  (sachant qu'elles sont toutes isomorphes). Par contre, lorsqu'on a fait un choix de colimite, la notation  $\text{Colim}_{\mathcal{C}} \bar{\alpha}$  représente *la colimite choisie* du diagramme  $\bar{\alpha}$ .

Ces choix de colimites correspondent à des choix de représentation ou de codage, nécessaires en informatique, en particulier lorsqu'on définit une syntaxe.

### 2.3.3 Exemples de colimites

Dans ce paragraphe, nous considérons quelques exemples de colimites de diagrammes particuliers sur une catégorie  $\mathcal{C}$ . Nous rappelons d'abord la définition d'objet initial.

**Définition 2.22** (Objet initial) Un objet  $\emptyset$  de  $\mathcal{C}$  est initial si et seulement si pour tout objet  $A$  de  $\mathcal{C}$ , il existe une unique flèche  $j_A : \emptyset \rightarrow A$  dans  $\mathcal{C}$ .

**Exemple 2.7** (Objet initial)

L'ensemble vide est initial dans la catégorie **Set**.

La spécification vide est initiale dans la catégorie **Spec**.

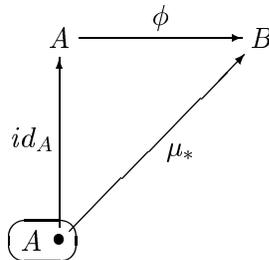
Le diagramme vide  $\circ$  (diagramme dont le graphe sous-jacent ne comporte aucun nœud et aucun arc) est initial dans  $\text{DIAGR}(\mathcal{C})$ .

Si le diagramme vide  $\circ$  a une colimite  $\emptyset$  dans  $\mathcal{C}$ , alors  $\emptyset$  est un objet initial de  $\mathcal{C}$ . Soit  $(\emptyset, \bar{\lambda} : \circ \rightarrow I_{\mathcal{C}}(\emptyset))$  le cône colimite de  $\circ$ . Soit  $A$  un objet de  $\mathcal{C}$ . Comme  $\circ$  est initial dans  $\text{DIAGR}(\mathcal{C})$ , on a une flèche  $\bar{\mu} : \circ \rightarrow I_{\mathcal{C}}(A)$ .  $\emptyset$  est colimite du diagramme  $\circ$ , donc il existe une unique flèche  $j_A : \emptyset \rightarrow A$  telle que  $I_{\mathcal{C}}(j_A) \circ \bar{\lambda} = \bar{\mu}$ . Soit  $j : \emptyset \rightarrow A$  une autre flèche de  $\mathcal{C}$ . On a  $I_{\mathcal{C}}(j) \circ \bar{\lambda} = \bar{\mu}$  car  $\circ$  est initial dans  $\text{DIAGR}(\mathcal{C})$ . Par conséquent,  $j = j_A$ .

**Exemple 2.8** (Colimite du diagramme  $I_{\mathcal{C}}(A)$ ) Soit un objet  $A$  de  $\mathcal{C}$ . Considérons le diagramme  $\bar{\alpha} = I_{\mathcal{C}}(A)$ . Le cône  $(A, \bar{id}_{\bar{\alpha}} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(A))$  est un cône colimite de  $\bar{\alpha}$ .

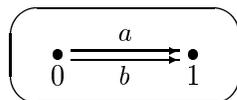
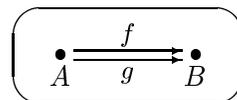
*Preuve.* Soit  $(B, \bar{\mu} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(B))$  un cône sur  $\bar{\alpha}$ . Il existe une unique flèche  $\phi : A \rightarrow B$  de  $\mathcal{C}$  telle que  $I_{\mathcal{C}}(\phi) \circ \bar{id}_{\bar{\alpha}} = \bar{\mu}$ . En effet,

$$\begin{aligned} I_{\mathcal{C}}(\phi) \circ \bar{id}_{\bar{\alpha}} &= \bar{\mu} \\ \Leftrightarrow \phi \circ id_A &= \mu_* \\ \Leftrightarrow \phi &= \mu_* \end{aligned}$$



□

**Exemple 2.9** (Coégalisateur) Soit deux objets  $A$  et  $B$ , ainsi que deux flèches  $f, g : A \rightarrow B$ . Le coégalisateur de  $f$  et  $g$  est la colimite du diagramme  $\bar{\kappa}$ , construit sur le graphe  $\kappa^{\Phi}$ .

graphe  $\kappa^{\Phi}$ diagramme  $\bar{\kappa}$

Soit un cône  $(D, \bar{\mu} : \bar{\kappa} \rightarrow I_C(D))$  sur  $\bar{\kappa}$ . Ce cône est entièrement déterminé par une flèche  $h : B \rightarrow D$  telle que

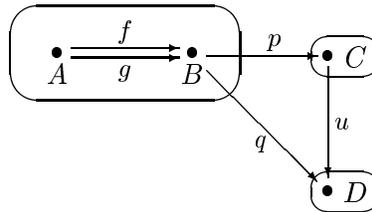
$$h \circ f = h \circ g.$$

En effet, il suffit de poser

$$\begin{aligned} \mu_0 &= h \circ f = h \circ g \\ \mu_1 &= h \end{aligned}$$

pour définir la flèche  $\bar{\mu} : \bar{\kappa} \rightarrow I_C(D)$ .

Soit  $(C, \bar{\lambda} : \bar{\kappa} \rightarrow I_C(C))$  le cône colimite de  $\bar{\kappa}$ . Alors, pour tout objet  $D$  et flèche  $q : B \rightarrow D$  telle que  $q \circ f = q \circ g$ , il existe une unique flèche  $u : C \rightarrow D$  telle que  $u \circ p = q$ .



Dans la catégorie des ensembles **Set**, le coégalisateur  $C$  est le quotient de l'ensemble  $B$  par la relation d'équivalence engendrée par  $\{(f(a), g(a)), \forall a \in A\}$ . L'application  $p : B \rightarrow C$  est la projection de  $B$  sur le quotient  $C$ . Une application  $q : B \rightarrow D$  telle que  $q \circ f = q \circ g$  est une application compatible avec la relation d'équivalence, qui se factorise donc à travers  $p$ . Il existe par conséquent une unique application  $u : C \rightarrow D$  telle que  $u \circ p = q$ .

La construction de somme amalgamée de deux ensembles proposée dans le chapitre 1 (section 1.4.1) est en réalité une façon générale de construire une somme amalgamée à partir d'une somme et d'un coégalisateur.

**Exemple 2.10** (Somme amalgamée) Soit trois objets  $A, B, C$  et deux flèches  $f : A \rightarrow B, g : A \rightarrow C$  dans  $\mathcal{C}$ . La somme amalgamée de  $B$  et  $C$  par rapport à  $f : A \rightarrow B$  et  $g : A \rightarrow C$  est la colimite du diagramme  $\bar{\pi}$ , construit sur le graphe  $\pi^\Phi$ .



Soit  $(P, \bar{\&} : \bar{\pi} \rightarrow I_C(P))$  le cône colimite de  $\bar{\pi}$ . Ce cône colimite consiste en trois flèches  $\&_0 : A \rightarrow P, \&_1 : B \rightarrow P$  et  $\&_2 : C \rightarrow P$  telles que

$$\&_0 = \&_1 \circ f = \&_2 \circ g.$$

Ce cône est donc entièrement déterminé par les deux flèches  $\&_1$  et  $\&_2$ .

Soit  $(D, \bar{\mu} : \bar{\pi} \rightarrow I_{\mathcal{C}}(D))$  un autre cône sur  $\bar{\pi}$ . Ce cône est constitué de trois flèches  $\mu_0 : A \rightarrow D$ ,  $f' = \mu_1 : B \rightarrow D$  et  $g' = \mu_2 : C \rightarrow D$  telles que

$$\mu_0 = f' \circ f = g' \circ g.$$

Un cône sur  $\bar{\pi}$  est donc entièrement déterminé par deux flèches  $f' : B \rightarrow D$  et  $g' : C \rightarrow D$  telles que  $f' \circ f = g' \circ g$ .

Comme  $P$  est la colimite de  $\bar{\pi}$ , il existe une unique flèche  $up : P \rightarrow D$  telle que

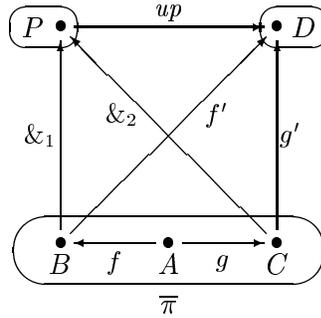
$$I_{\mathcal{C}}(up) \circ \bar{\&} = \bar{\mu}.$$

Cette égalité est équivalente aux deux égalités

$$up \circ \&_1 = f' \tag{i}$$

$$up \circ \&_2 = g' \tag{ii}$$

car l'égalité (i) — comme l'égalité (ii) — implique  $up \circ \&_0 = \mu_0$ .



somme amalgamée, colimite du diagramme  $\bar{\pi}$

**Définition 2.23** (Catégorie finiment cocomplète) Une catégorie  $\mathcal{C}$  est *finiment cocomplète* si et seulement si tout diagramme fini a une colimite.

**Théorème 2.2** Une catégorie  $\mathcal{C}$  est finiment cocomplète si et seulement si  $\mathcal{C}$  a un objet initial et des sommes amalgamées.

*Preuve.* Pour montrer ce théorème, on utilise le théorème suivant :

**Théorème 2.3** Une catégorie  $\mathcal{C}$  est finiment cocomplète si et seulement si  $\mathcal{C}$  a un objet initial, des sommes pour tout couple d'objets de  $\mathcal{C}$ , et des coégalisateurs pour tout couple de flèches  $f, g : A \rightarrow B$  de  $\mathcal{C}$ .

Le dual de ce théorème est démontré dans [McL71], page 109. On vérifie facilement qu'on peut adapter la démonstration de S. Mac Lane à notre définition de diagramme. Ensuite, pour montrer le théorème 2.2, il suffit de montrer qu'on peut simuler la construction d'une somme et d'un coégalisateur à l'aide d'un objet initial et de sommes amalgamées.  $\square$

### 2.3.4 Adjonction $(\text{Colim}_{\mathcal{C}} \dashv I_{\mathcal{C}})$

Dans ce paragraphe, nous considérons une catégorie  $\mathcal{C}$  finiment cocomplète. Tout diagramme fini

$$\bar{\alpha} = (\alpha^{\Phi}, \alpha : \alpha^{\Phi} \rightarrow \mathcal{C})$$

a donc un cône colimite que l'on va noter

$$(\text{Colim}_{\mathcal{C}} \bar{\alpha}, \bar{\eta}_{\bar{\alpha}} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\alpha})).$$

Nous montrons que l'on peut étendre la fonction  $\text{Colim}_{\mathcal{C}}$  sur les flèches de  $\text{DIAGR}(\mathcal{C})$  de façon à ce que  $\text{Colim}_{\mathcal{C}} : \text{DIAGR}(\mathcal{C}) \rightarrow \mathcal{C}$  soit un foncteur. La définition des flèches de  $\text{DIAGR}(\mathcal{C})$  a été justement choisie afin que cette extension soit possible.

Soit deux diagrammes

$$\bar{\alpha} = (\alpha^{\Phi}, \alpha : \alpha^{\Phi} \rightarrow \mathcal{C}) \quad \text{et} \quad \bar{\beta} = (\beta^{\Phi}, \beta : \beta^{\Phi} \rightarrow \mathcal{C})$$

ayant respectivement pour cônes colimites

$$(\text{Colim}_{\mathcal{C}} \bar{\alpha}, \bar{\eta}_{\bar{\alpha}} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\alpha})) \quad \text{et} \quad (\text{Colim}_{\mathcal{C}} \bar{\beta}, \bar{\eta}_{\bar{\beta}} : \bar{\beta} \rightarrow I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\beta})).$$

Soit  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta} = (\sigma^{\Phi} : \alpha^{\Phi} \rightsquigarrow \beta^{\Phi}, \sigma : \alpha \rightsquigarrow \beta \circ \sigma^{\Phi})$  une flèche de  $\text{DIAGR}(\mathcal{C})$ .  
Considérons le cône sur  $\bar{\alpha}$

$$(\text{Colim}_{\mathcal{C}} \bar{\beta}, \bar{\eta}_{\bar{\beta}} \circ \bar{\sigma} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\beta})).$$

Comme  $(\text{Colim}_{\mathcal{C}} \bar{\alpha}, \bar{\eta}_{\bar{\alpha}} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\alpha}))$  est un cône colimite de  $\bar{\alpha}$ , il existe une unique flèche

$$\text{Colim}_{\mathcal{C}} \bar{\sigma} : \text{Colim}_{\mathcal{C}} \bar{\alpha} \rightarrow \text{Colim}_{\mathcal{C}} \bar{\beta}$$

telle que

$$I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\sigma}) \circ \bar{\eta}_{\bar{\alpha}} = \bar{\eta}_{\bar{\beta}} \circ \bar{\sigma}.$$

**Théorème 2.4** *Soit  $\mathcal{C}$  une catégorie finiment cocomplète. Alors, on a les résultats suivants.*

1.  $\text{Colim}_{\mathcal{C}} : \text{DIAGR}(\mathcal{C}) \rightarrow \mathcal{C}$  est un foncteur.
2. La fonction  $\bar{\eta}$ , qui à tout diagramme  $\bar{\alpha}$  associe la flèche du cône colimite de  $\bar{\alpha}$  vers  $I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\alpha})$  est une transformation naturelle

$$\bar{\eta} : \text{Id}_{\text{DIAGR}(\mathcal{C})} \rightrightarrows I_{\mathcal{C}} \circ \text{Colim}_{\mathcal{C}}.$$

3.  $(\text{Colim}_{\mathcal{C}} \dashv I_{\mathcal{C}}) : \text{DIAGR}(\mathcal{C}) \rightarrow \mathcal{C}$  est une adjonction, dont l'unité est la transformation naturelle

$$\bar{\eta} : \text{Id}_{\text{DIAGR}(\mathcal{C})} \rightrightarrows I_{\mathcal{C}} \circ \text{Colim}_{\mathcal{C}}.$$

4. La coïunité de l'adjonction  $\epsilon : \text{Colim}_{\mathcal{C}} \circ I_{\mathcal{C}} \rightrightarrows \text{Id}_{\mathcal{C}}$  est un isomorphisme naturel.

*Preuve.* Ces résultats sont des conséquences immédiates de la définition de  $\text{Colim}_{\mathcal{C}}$ .

1. Soit un diagramme  $\bar{\alpha}$ .

$$\begin{aligned} & I_{\mathcal{C}}(id_{\text{Colim}_{\mathcal{C}} \bar{\alpha}}) \circ \bar{\eta}_{\bar{\alpha}} \\ &= \bar{id}_{I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\alpha})} \circ \bar{\eta}_{\bar{\alpha}} \quad (I_{\mathcal{C}} \text{ est un foncteur}) \\ &= \bar{\eta}_{\bar{\alpha}} \end{aligned}$$

Or,  $\text{Colim}_{\mathcal{C}} \bar{id}_{\bar{\alpha}}$  est l'unique flèche telle que  $I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{id}_{\bar{\alpha}}) \circ \bar{\eta}_{\bar{\alpha}} = \bar{\eta}_{\bar{\alpha}}$ , donc

$$\text{Colim}_{\mathcal{C}} \bar{id}_{\bar{\alpha}} = \bar{id}_{\text{Colim}_{\mathcal{C}} \bar{\alpha}}.$$

Soit trois diagrammes  $\bar{\alpha}$ ,  $\bar{\beta}$  et  $\bar{\gamma}$ , et deux flèches  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta}$  et  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\gamma}$ .

$$\begin{aligned} & I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\tau} \circ \text{Colim}_{\mathcal{C}} \bar{\sigma}) \circ \bar{\eta}_{\bar{\alpha}} \\ &= I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\tau}) \circ I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\sigma}) \circ \bar{\eta}_{\bar{\alpha}} \quad (I_{\mathcal{C}} \text{ est un foncteur}) \\ &= I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\tau}) \circ \bar{\eta}_{\bar{\beta}} \circ \bar{\sigma} \quad (\text{définition de } \text{Colim}_{\mathcal{C}} \bar{\sigma}) \\ &= \bar{\eta}_{\bar{\gamma}} \circ \bar{\tau} \circ \bar{\sigma} \quad (\text{définition de } \text{Colim}_{\mathcal{C}} \bar{\tau}) \end{aligned}$$

Or,  $\text{Colim}_{\mathcal{C}}(\bar{\tau} \circ \bar{\sigma})$  est l'unique flèche telle que

$$\text{Colim}_{\mathcal{C}}(\bar{\tau} \circ \bar{\sigma}) \circ \bar{\eta}_{\bar{\alpha}} = \bar{\eta}_{\bar{\gamma}} \circ \bar{\tau} \circ \bar{\sigma}$$

donc

$$\text{Colim}_{\mathcal{C}}(\bar{\tau} \circ \bar{\sigma}) = \text{Colim}_{\mathcal{C}} \bar{\tau} \circ \text{Colim}_{\mathcal{C}} \bar{\sigma}.$$

Par conséquent,  $\text{Colim}_{\mathcal{C}} : \text{DIAGR}(\mathcal{C}) \rightarrow \mathcal{C}$  est un foncteur.

2. Soit deux diagrammes  $\bar{\alpha}$ ,  $\bar{\beta}$  et une flèche  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta}$ . Par définition de  $\text{Colim}_{\mathcal{C}} \bar{\sigma}$ , on a bien

$$I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\sigma}) \circ \bar{\eta}_{\bar{\alpha}} = \bar{\eta}_{\bar{\beta}} \circ \bar{\sigma}.$$

3. Soit un diagramme  $\bar{\alpha}$ , un objet  $B$  de  $\mathcal{C}$ , et une flèche  $\bar{\sigma} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(B)$ . Nous avons un cône  $(B, \bar{\sigma} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(B))$  sur le diagramme  $\bar{\alpha}$ , donc il existe une unique flèche  $g : \text{Colim}_{\mathcal{C}} \bar{\alpha} \rightarrow B$  de  $\mathcal{C}$  telle que

$$I_{\mathcal{C}}(g) \circ \bar{\eta}_{\bar{\alpha}} = \bar{\sigma}.$$

4. D'après la propriété 2.2, il suffit de montrer que pour tout objet  $B$  de  $\mathcal{C}$ ,

$$\epsilon_B : \text{Colim}_{\mathcal{C}} I_{\mathcal{C}}(B) \rightarrow B$$

est un isomorphisme. Comme on a un adjonction  $(\text{Colim}_{\mathcal{C}} \dashv I_{\mathcal{C}})$ , et d'après la propriété 2.4,

$$\begin{aligned} & I_{\mathcal{C}} \epsilon \cdot \eta I_{\mathcal{C}} = id_{I_{\mathcal{C}}} \\ & \Rightarrow I_{\mathcal{C}}(\epsilon_B) \circ \bar{\eta}_{I_{\mathcal{C}}(B)} = \bar{id}_{I_{\mathcal{C}}(B)} \\ & \Rightarrow \epsilon_B \circ (\eta_{I_{\mathcal{C}}(B)})_* = id_B. \end{aligned}$$

D'autre part,

$$\begin{aligned}
I_{\mathcal{C}}(\epsilon_B) \circ \bar{\eta}_{I_{\mathcal{C}}(B)} &= \overline{id}_{I_{\mathcal{C}}(B)} \\
\Rightarrow \bar{\eta}_{I_{\mathcal{C}}(B)} \circ I_{\mathcal{C}}(\epsilon_B) \circ \bar{\eta}_{I_{\mathcal{C}}(B)} &= \bar{\eta}_{I_{\mathcal{C}}(B)} \\
\Rightarrow I_{\mathcal{C}}((\eta_{I_{\mathcal{C}}(B)})_* \circ \epsilon_B) \circ \bar{\eta}_{I_{\mathcal{C}}(B)} &= \bar{\eta}_{I_{\mathcal{C}}(B)} \quad (I_{\mathcal{C}} \text{ foncteur})
\end{aligned}$$

Comme  $(\text{Colim}_{\mathcal{C}} I_{\mathcal{C}}(B), \bar{\eta}_{I_{\mathcal{C}}(B)} : I_{\mathcal{C}}(B) \rightarrow I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} I_{\mathcal{C}}(B)))$  est un cône co-limite du diagramme  $I_{\mathcal{C}}(B)$ , il existe une unique flèche  $\phi : \text{Colim}_{\mathcal{C}} I_{\mathcal{C}}(B) \rightarrow \text{Colim}_{\mathcal{C}} I_{\mathcal{C}}(B)$  de  $\mathcal{C}$  telle que

$$I_{\mathcal{C}}(\phi) \circ \bar{\eta}_{I_{\mathcal{C}}(B)} = \bar{\eta}_{I_{\mathcal{C}}(B)}.$$

Or, les flèches  $(\eta_{I_{\mathcal{C}}(B)})_* \circ \epsilon_B$  et  $id_{\text{Colim}_{\mathcal{C}} I_{\mathcal{C}}(B)}$  satisfont cette propriété, donc

$$(\eta_{I_{\mathcal{C}}(B)})_* \circ \epsilon_B = id_{\text{Colim}_{\mathcal{C}} I_{\mathcal{C}}(B)}.$$

Donc  $\epsilon_B : \text{Colim}_{\mathcal{C}} I_{\mathcal{C}}(B) \rightarrow B$  est un isomorphisme, et  $\epsilon_B^{-1} = (\eta_{I_{\mathcal{C}}(B)})_*$ .

□

### 2.3.5 Conservation de colimites

Dans ce paragraphe, on considère deux catégories  $\mathcal{C}$  et  $\mathcal{D}$ , et un foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ . Nous définissons l'image d'un diagramme et d'un morphisme de diagrammes par ce foncteur : l'image d'un objet de  $\text{DIAGR}(\mathcal{C})$  est un objet de  $\text{DIAGR}(\mathcal{D})$ , et l'image d'une flèche de  $\text{DIAGR}(\mathcal{C})$  est une flèche de  $\text{DIAGR}(\mathcal{D})$ .

**Définition 2.24** (Image d'un diagramme par un foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ )

Soit un diagramme  $\bar{\alpha} = (\alpha^{\Phi}, \alpha : \alpha^{\Phi} \rightarrow \mathcal{C})$  sur  $\mathcal{C}$ . L'image par  $F$  de  $\bar{\alpha}$  est le diagramme sur  $\mathcal{D}$

$$F \circ \bar{\alpha} = (\alpha^{\Phi}, F \circ \alpha : \alpha^{\Phi} \rightarrow \mathcal{D}).$$

Intuitivement,  $F \circ \bar{\alpha}$  est un diagramme construit sur le graphe sous-jacent  $\alpha^{\Phi}$  du diagramme  $\bar{\alpha}$ . Les nœuds  $n$  de  $\alpha^{\Phi}$  sont étiquetés par les objets  $F(\alpha(n))$  de  $\mathcal{D}$ , et les arcs  $a : m \rightarrow n$  de  $\alpha^{\Phi}$  par les flèches  $F(\alpha(a)) : F(\alpha(m)) \rightarrow F(\alpha(n))$  de  $\mathcal{D}$ .

Par exemple, étant donné un objet  $A$  de  $\mathcal{C}$ , le diagramme  $F \circ I_{\mathcal{C}}(A)$  est le diagramme  $I_{\mathcal{D}}(F(A))$ .

$$\begin{array}{ccc}
\begin{array}{c} \bullet A \\ \text{diagramme } I_{\mathcal{C}}(A) \end{array} & & \begin{array}{c} \bullet F(A) \\ \text{diagramme } F \circ I_{\mathcal{C}}(A) = I_{\mathcal{D}}(F(A)) \end{array}
\end{array}$$

**Lemme 2.3** Soit un diagramme  $\bar{\alpha}$  sur  $\mathcal{C}$ . Soit  $n_0$  et  $n_k$  deux nœuds de  $\alpha^\Phi$ . Soit  $u : A \rightarrow \alpha(n_0)$  et  $v : A \rightarrow \alpha(n_k)$  deux flèches de  $\mathcal{C}$  telles que

$$u \sim_{\bar{\alpha}} v \quad [Z : n_0 \rightsquigarrow n_k].$$

Alors,

$$F(u) \sim_{F \circ \bar{\alpha}} F(v) \quad [Z : n_0 \rightsquigarrow n_k].$$

Remarquons que si  $Z : n_0 \rightsquigarrow n_k$  est un zigzag sur le diagramme  $\bar{\alpha}$ , alors c'est également un zigzag sur le diagramme  $F \circ \bar{\alpha}$ , puisque ces deux diagrammes sont construits sur le même graphe sous-jacent  $\alpha^\Phi$ .

*Preuve.* Application immédiate des définitions de  $F \circ \bar{\alpha}$ ,  $\sim_{\bar{\alpha}}$  et  $\sim_{F \circ \bar{\alpha}}$ .  $\square$

**Définition 2.25** (Image d'un morphisme de diagrammes par  $F$ )

Soit  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta} = (\sigma^\Phi : \alpha^\Phi \rightsquigarrow \beta^\Phi, \sigma : \alpha \rightsquigarrow \beta \circ \sigma^\Phi)$  une flèche de  $\text{DIAGR}(\mathcal{C})$ .

L'image par  $F$  de  $\bar{\sigma}$  est la flèche de  $\text{DIAGR}(\mathcal{D})$

$$F\bar{\sigma} = (\sigma^\Phi : \alpha^\Phi \rightsquigarrow \beta^\Phi, F\sigma : F \circ \alpha \rightsquigarrow F \circ \beta \circ \sigma^\Phi)$$

où  $F\sigma : F \circ \alpha \rightsquigarrow F \circ \beta \circ \sigma^\Phi$  est la transformation naturelle généralisée définie par

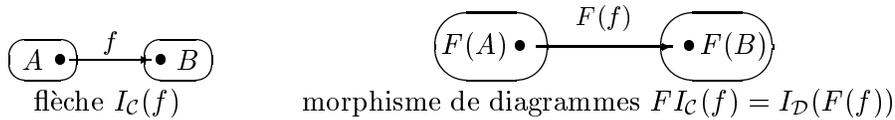
$$\forall n \in \text{Nœud}(\alpha^\Phi), (F\sigma)_n = F(\sigma_n).$$

On vérifie que cela définit bien une transformation naturelle généralisée. Considérons un arc  $a : m \rightarrow n$  de  $\alpha^\Phi$ .

$$\begin{aligned} \sigma_n \circ \alpha(a) &\sim_{\bar{\beta}} \sigma_m [\sigma^\Phi(a)] && (\sigma \text{ transformation nat. généralisée}) \\ \Rightarrow F(\sigma_n \circ \alpha(a)) &\sim_{F \circ \bar{\beta}} F(\sigma_m) [\sigma^\Phi(a)] && (\text{lemme 2.3}) \\ \Rightarrow F(\sigma_n) \circ F(\alpha(a)) &\sim_{F \circ \bar{\beta}} F(\sigma_m) [\sigma^\Phi(a)] && (F \text{ foncteur}) \\ \Rightarrow (F\sigma)_n \circ F(\alpha(a)) &\sim_{F \circ \bar{\beta}} (F\sigma)_m [\sigma^\Phi(a)] && (\text{définition de } F\sigma) \end{aligned}$$

Par conséquent,  $F\bar{\sigma}$  est bien une transformation naturelle généralisée.

Par exemple, soit une flèche  $f : A \rightarrow B$  de  $\mathcal{C}$ . La flèche  $F I_{\mathcal{C}}(f)$  est égale à  $I_{\mathcal{D}}(F(f))$ .



Pour tout diagramme  $\bar{\alpha}$  de  $\text{DIAGR}(\mathcal{C})$ , nous avons donc défini un diagramme  $F \circ \bar{\alpha}$  de  $\text{DIAGR}(\mathcal{D})$ ; et pour toute flèche  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta}$  de  $\text{DIAGR}(\mathcal{C})$ , nous avons défini une flèche  $F\bar{\sigma} : F \circ \bar{\alpha} \rightarrow F \circ \bar{\beta}$  de  $\text{DIAGR}(\mathcal{D})$ . Ces deux applications forment en fait un foncteur.

**Lemme 2.4** *L'application*

$$\begin{array}{ccc} \text{DIAGR}(F) : \text{DIAGR}(\mathcal{C}) & \rightarrow & \text{DIAGR}(\mathcal{D}) \\ \bar{\alpha} & \mapsto & F \circ \bar{\alpha} \\ \bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta} & \mapsto & F\bar{\sigma} : F \circ \bar{\alpha} \rightarrow F \circ \bar{\beta} \end{array}$$

*est un foncteur*  $\text{DIAGR}(F) : \text{DIAGR}(\mathcal{C}) \rightarrow \text{DIAGR}(\mathcal{D})$ .

Finalement, nous avons donc défini  $\text{DIAGR}$  à la fois sur les catégories (objets de  $\mathbf{CAT}$ ) et sur les foncteurs (flèches de  $\mathbf{CAT}$ ). De nouveau, nous avons défini un foncteur.

**Lemme 2.5** *L'application*

$$\begin{array}{ccc} \text{DIAGR} : \mathbf{CAT} & \rightarrow & \mathbf{CAT} \\ \mathcal{C} & \mapsto & \text{DIAGR}(\mathcal{C}) \\ F : \mathcal{C} \rightarrow \mathcal{D} & \mapsto & \text{DIAGR}(F) : \text{DIAGR}(\mathcal{C}) \rightarrow \text{DIAGR}(\mathcal{D}) \end{array}$$

*est un foncteur*  $\text{DIAGR} : \mathbf{CAT} \rightarrow \mathbf{CAT}$ .

**Propriété 2.5** *L'application*  $I$  *qui à toute catégorie*  $\mathcal{C}$  *associe le foncteur*  $I_{\mathcal{C}} : \mathcal{C} \rightarrow \text{DIAGR}(\mathcal{C})$  *est une transformation naturelle*

$$I : \text{Id}_{\mathbf{CAT}} \rightarrow \text{DIAGR}.$$

*Preuve.* Pour tout foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ , on a

$$\text{DIAGR}(F) \circ I_{\mathcal{C}} = I_{\mathcal{D}} \circ F.$$

□

**Lemme 2.6** *Soit deux foncteurs*  $F, F' : \mathcal{C} \rightarrow \mathcal{D}$ . *Alors,*

$$F \cong F' \Rightarrow \text{DIAGR}(F) \cong \text{DIAGR}(F').$$

*Preuve.* Soit  $\omega : F \rightarrow F'$  l'isomorphisme naturel entre ces deux foncteurs. On définit une transformation naturelle

$$\bar{\xi} : \text{DIAGR}(F) \rightarrow \text{DIAGR}(F').$$

À tout objet  $\bar{\alpha}$  de  $\text{DIAGR}(\mathcal{C})$ , on associe une flèche

$$\bar{\xi}_{\bar{\alpha}} : F \circ \bar{\alpha} \rightarrow F' \circ \bar{\alpha}$$

de  $\text{DIAGR}(\mathcal{D})$  définie la façon suivante.

- Les deux diagrammes  $F \circ \bar{\alpha}$  et  $F' \circ \bar{\alpha}$  sont construits sur le graphe  $\alpha^{\Phi}$  donc on peut définir  $\bar{\xi}_{\bar{\alpha}}^{\Phi} = id_{\alpha^{\Phi}}$ .

- Pour tout nœud  $n$  de  $\alpha^\Phi$ , on pose

$$(\xi_{\bar{\alpha}})_n = \omega_{\alpha(n)} : F(\alpha(n)) \rightarrow F'(\alpha(n)).$$

Comme  $\omega$  est une transformation naturelle, pour tout arc  $a : m \rightarrow n$  de  $\alpha^\Phi$ , on a

$$\begin{aligned} F'(\alpha(a)) \circ \omega_{\alpha(m)} &= \omega_{\alpha(n)} \circ F(\alpha(a)) \\ \Leftrightarrow F'(\alpha(a)) \circ (\xi_{\bar{\alpha}})_m &= (\xi_{\bar{\alpha}})_n \circ F(\alpha(a)) \end{aligned}$$

Par conséquent,  $\xi_{\bar{\alpha}} : F \circ \bar{\alpha} \rightarrow F' \circ \bar{\alpha}$  est une transformation naturelle, et donc  $\bar{\xi}_{\bar{\alpha}} : F \circ \bar{\alpha} \rightarrow F' \circ \bar{\alpha}$  est bien une flèche de  $\text{DIAGR}(\mathcal{D})$ .

De plus,  $\bar{\xi}_{\bar{\alpha}}$  est un isomorphisme, et pour tout nœud  $n$  de  $\alpha^\Phi$ ,  $(\xi_{\bar{\alpha}}^{-1})_n = \omega_{\alpha(n)}^{-1}$ .

On montre que

$$\bar{\xi} : \text{DIAGR}(F) \rightarrow \text{DIAGR}(F')$$

est une transformation naturelle. Pour toute flèche  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta}$  de  $\text{DIAGR}(\mathcal{C})$ , il faut vérifier que

$$\bar{\xi}_{\bar{\beta}} \circ \text{DIAGR}(F)(\bar{\sigma}) = \text{DIAGR}(F')(\bar{\sigma}) \circ \bar{\xi}_{\bar{\alpha}}.$$

Pour montrer cette égalité, on vérifie que les morphismes généralisés et les transformations naturelles généralisées qui correspondent aux deux membres coïncident. Notons

$$\begin{aligned} \bar{\tau} &= \bar{\xi}_{\bar{\beta}} \circ \text{DIAGR}(F)(\bar{\sigma}) \\ \bar{\tau}' &= \text{DIAGR}(F')(\bar{\sigma}) \circ \bar{\xi}_{\bar{\alpha}}. \end{aligned}$$

- $\tau^\Phi = \sigma^\Phi = \tau'^\Phi$ .
- Pour tout nœud  $n$  de  $\alpha^\Phi$ ,

$$\begin{aligned} \tau_n &= (\xi_{\bar{\beta}})_{\sigma^\Phi(n)} \circ F(\sigma_n) && \text{(définition de } \bar{\tau}\text{)} \\ &= \omega_{\beta(\sigma^\Phi(n))} \circ F(\sigma_n) && \text{(définition de } \bar{\xi}\text{)} \\ &= F'(\sigma_n) \circ \omega_{\alpha(n)} && \text{(\omega transformation naturelle)} \\ &= F'(\sigma_n) \circ (\xi_{\bar{\alpha}})_n && \text{(définition de } \bar{\xi}\text{)} \\ &= \tau'_n && \text{(définition de } \bar{\tau}'\text{)} \end{aligned}$$

Par conséquent, on a bien un isomorphisme naturel

$$\bar{\xi} : \text{DIAGR}(F) \rightarrow \text{DIAGR}(F').$$

□

**Définition 2.26** (Foncteur conservant une colimite)

Soit un diagramme  $\bar{\alpha} = (\alpha^\Phi, \alpha : \alpha^\Phi \rightarrow \mathcal{C})$ , ayant pour cône colimite

$$(A, \bar{\eta}_{\bar{\alpha}} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(A)).$$

On dit que le foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$  conserve la colimite de  $\bar{\alpha}$  si et seulement si

$$(F(A), \text{DIAGR}(F)(\bar{\eta}_{\bar{\alpha}}) : F \circ \bar{\alpha} \rightarrow I_{\mathcal{D}}(F(A)))$$

est un cône colimite du diagramme  $F \circ \bar{\alpha}$ .

**Définition 2.27** (Foncteur conservant les colimites finies)

Soit deux catégories finiment cocomplètes  $\mathcal{C}$  et  $\mathcal{D}$ . On dit qu'un foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$  conserve les colimites finies si et seulement si pour tout diagramme fini  $\bar{\alpha}$  sur  $\mathcal{C}$ ,  $F$  conserve la colimite de  $\bar{\alpha}$ .

**Propriété 2.6** Soit deux catégories finiment cocomplètes  $\mathcal{C}$  et  $\mathcal{D}$ , et un foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$  qui conserve les colimites finies. Alors, il existe un isomorphisme naturel

$$\text{Colim}_{\mathcal{D}} \circ \text{DIAGR}(F) \cong F \circ \text{Colim}_{\mathcal{C}}.$$

*Preuve.* Soit un diagramme  $\bar{\alpha}$  sur  $\mathcal{C}$ , ayant pour cône colimite

$$(\text{Colim}_{\mathcal{C}} \bar{\alpha}, \bar{\eta}_{\bar{\alpha}} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\alpha})).$$

On considère le diagramme  $F \circ \bar{\alpha}$  sur  $\mathcal{D}$ , qui a pour cône colimite

$$(\text{Colim}_{\mathcal{D}}(F \circ \bar{\alpha}), \bar{\eta}_{F \circ \bar{\alpha}} : F \circ \bar{\alpha} \rightarrow I_{\mathcal{D}}(\text{Colim}_{\mathcal{D}}(F \circ \bar{\alpha}))).$$

Comme  $F$  conserve la colimite de  $\bar{\alpha}$ , et d'après le lemme 2.1,

$$\text{Colim}_{\mathcal{D}}(F \circ \bar{\alpha}) \cong F(\text{Colim}_{\mathcal{C}} \bar{\alpha})$$

et l'isomorphisme

$$\phi_{\bar{\alpha}} : \text{Colim}_{\mathcal{D}}(F \circ \bar{\alpha}) \rightarrow F(\text{Colim}_{\mathcal{C}} \bar{\alpha})$$

est l'unique flèche telle que  $I_{\mathcal{D}}(\phi_{\bar{\alpha}}) \circ \bar{\eta}_{F \circ \bar{\alpha}} = \text{DIAGR}(F)(\bar{\eta}_{\bar{\alpha}})$ .

Il suffit de montrer qu'on a une transformation naturelle

$$\phi : \text{Colim}_{\mathcal{D}} \circ \text{DIAGR}(F) \rightarrow F \circ \text{Colim}_{\mathcal{C}}.$$

Soit un diagramme  $\bar{\beta}$ , ayant pour cône colimite

$$(\text{Colim}_{\mathcal{C}} \bar{\beta}, \bar{\eta}_{\bar{\beta}} : \bar{\beta} \rightarrow I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\beta})).$$

Soit

$$\phi_{\bar{\beta}} : \text{Colim}_{\mathcal{D}}(F \circ \bar{\beta}) \rightarrow F(\text{Colim}_{\mathcal{C}} \bar{\beta})$$

l'unique flèche telle que

$$I_{\mathcal{D}}(\phi_{\bar{\beta}}) \circ \bar{\eta}_{F \circ \bar{\beta}} = \text{DIAGR}(F)(\bar{\eta}_{\bar{\beta}}).$$

Soit  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta}$  une flèche de  $\text{DIAGR}(\mathcal{C})$ . Pour montrer que  $\phi$  est une transformation naturelle, il faut montrer que

$$F(\text{Colim}_{\mathcal{C}} \bar{\sigma}) \circ \phi_{\bar{\alpha}} = \phi_{\bar{\beta}} \circ \text{Colim}_{\mathcal{D}}(\text{DIAGR}(F)(\bar{\sigma})).$$

Pour cela, il suffit de montrer que

$$I_{\mathcal{D}}(F(\text{Colim}_{\mathcal{C}} \bar{\sigma}) \circ \phi_{\bar{\alpha}}) \circ \bar{\eta}_{F \circ \bar{\alpha}} = I_{\mathcal{D}}(\phi_{\bar{\beta}} \circ \text{Colim}_{\mathcal{D}}(\text{DIAGR}(F)(\bar{\sigma}))) \circ \bar{\eta}_{F \circ \bar{\alpha}}.$$

$$\begin{aligned} & I_{\mathcal{D}}(F(\text{Colim}_{\mathcal{C}} \bar{\sigma}) \circ \phi_{\bar{\alpha}}) \circ \bar{\eta}_{F \circ \bar{\alpha}} \\ &= I_{\mathcal{D}}(F(\text{Colim}_{\mathcal{C}} \bar{\sigma})) \circ I_{\mathcal{D}}(\phi_{\bar{\alpha}}) \circ \bar{\eta}_{F \circ \bar{\alpha}} && (I_{\mathcal{D}} \text{ est un foncteur}) \\ &= I_{\mathcal{D}}(F(\text{Colim}_{\mathcal{C}} \bar{\sigma})) \circ \text{DIAGR}(F)(\bar{\eta}_{\bar{\alpha}}) && (\text{définition de } \phi_{\bar{\alpha}}) \\ &= \text{DIAGR}(F)(I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\sigma})) \circ \text{DIAGR}(F)(\bar{\eta}_{\bar{\alpha}}) && (I : \text{Id}_{\mathbf{CAT}} \rightarrow \text{DIAGR}) \\ &= \text{DIAGR}(F)(I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\sigma}) \circ \bar{\eta}_{\bar{\alpha}}) && (\text{DIAGR}(F) \text{ est un foncteur}) \\ &= \text{DIAGR}(F)(\bar{\eta}_{\bar{\beta}} \circ \bar{\sigma}) && (\text{définition de } \text{Colim}_{\mathcal{C}} \bar{\sigma}) \\ &= \text{DIAGR}(F)(\bar{\eta}_{\bar{\beta}}) \circ \text{DIAGR}(F)(\bar{\sigma}) && (\text{DIAGR}(F) \text{ est un foncteur}) \\ &= I_{\mathcal{D}}(\phi_{\bar{\beta}}) \circ \bar{\eta}_{F \circ \bar{\beta}} \circ \text{DIAGR}(F)(\bar{\sigma}) && (\text{définition de } \phi_{\bar{\beta}}) \\ &= I_{\mathcal{D}}(\phi_{\bar{\beta}}) \circ I_{\mathcal{D}}(\text{Colim}_{\mathcal{D}}(\text{DIAGR}(F)(\bar{\sigma}))) \circ \bar{\eta}_{F \circ \bar{\alpha}} && (\text{déf. } \text{Colim}_{\mathcal{D}}(\text{DIAGR}(F)(\bar{\sigma}))) \\ &= I_{\mathcal{D}}(\phi_{\bar{\beta}} \circ \text{Colim}_{\mathcal{D}}(\text{DIAGR}(F)(\bar{\sigma}))) \circ \bar{\eta}_{F \circ \bar{\alpha}} && (I_{\mathcal{D}} \text{ est un foncteur}) \end{aligned}$$

Par conséquent, on a bien une transformation naturelle

$$\phi : \text{Colim}_{\mathcal{D}} \circ \text{DIAGR}(F) \rightarrow F \circ \text{Colim}_{\mathcal{C}}.$$

Comme pour tout diagramme  $\bar{\alpha}$ ,  $\phi_{\bar{\alpha}}$  est un isomorphisme, on a donc un isomorphisme naturel

$$\text{Colim}_{\mathcal{D}} \circ \text{DIAGR}(F) \cong F \circ \text{Colim}_{\mathcal{C}}.$$

□

Si on fait certains choix de colimites dans les catégories  $\mathcal{C}$  et  $\mathcal{D}$ , on peut renforcer la définition 2.26 en imposant que *la colimite choisie* du diagramme  $\bar{\alpha}$  sur  $\mathcal{C}$  soit envoyée sur *la colimite choisie* du diagramme  $F \circ \bar{\alpha}$  sur  $\mathcal{D}$ .

**Définition 2.28** (Foncteur conservant fortement une colimite)

Soit un diagramme  $\bar{\alpha} = (\alpha^{\Phi}, \alpha : \alpha^{\Phi} \rightarrow \mathcal{C})$ , ayant pour cône colimite choisi

$$(A, \bar{\eta}_{\bar{\alpha}} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(A)).$$

On dit que le foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$  *conservé fortement la colimite choisie* de  $\bar{\alpha}$  si et seulement si

$$(F(A), \text{DIAGR}(F)(\bar{\eta}_{\bar{\alpha}}) : F \circ \bar{\alpha} \rightarrow I_{\mathcal{D}}(F(A)))$$

est le cône colimite choisi du diagramme  $F \circ \bar{\alpha}$  sur  $\mathcal{D}$ .

Si  $F$  conserve fortement la colimite choisie de  $\bar{\alpha}$ , on a donc

$$F(\text{Colim}_{\mathcal{C}} \bar{\alpha}) = \text{Colim}_{\mathcal{D}}(F \circ \bar{\alpha}).$$

### 2.3.6 Aplatissement

**Diagrammes de diagrammes : catégorie  $\text{DIAGR}^2(\mathcal{C})$**

Étant donné une catégorie  $\mathcal{C}$ , on peut considérer la catégorie

$$\text{DIAGR}^2(\mathcal{C}) = \text{DIAGR}(\text{DIAGR}(\mathcal{C}))$$

qui a pour objets des “diagrammes de diagrammes”, c’est-à-dire des diagrammes sur  $\text{DIAGR}(\mathcal{C})$ . Un objet  $\overline{\Delta}$  de  $\text{DIAGR}^2(\mathcal{C})$  est donc un couple

$$\overline{\Delta} = (\Delta^\Phi, \Delta : \Delta^\Phi \rightarrow \text{DIAGR}(\mathcal{C}))$$

où  $\Delta^\Phi$  est un graphe, et  $\Delta : \Delta^\Phi \rightarrow \text{DIAGR}(\mathcal{C})$  est un foncteur. Détaillons l’action de ce foncteur sur les nœuds et les arcs de  $\Delta^\Phi$ .

- Pour tout nœud  $N$  du graphe  $\Delta^\Phi$ , on a un diagramme sur  $\mathcal{C}$

$$\overline{\Delta(N)} = (\Delta(N)^\Phi, \Delta(N) : \Delta(N)^\Phi \rightarrow \mathcal{C})$$

où

- $\Delta(N)^\Phi$  est un graphe;
- $\Delta(N) : \Delta(N)^\Phi \rightarrow \mathcal{C}$  est un foncteur, qui associe à tout nœud  $n$  du graphe  $\Delta(N)^\Phi$  un objet  $\Delta(N)(n)$  de  $\mathcal{C}$ , et à tout arc  $a : n \rightarrow n'$  du graphe  $\Delta(N)^\Phi$  une flèche  $\Delta(N)(a)$  de  $\mathcal{C}$ .
- Pour tout arc  $A : N \rightarrow N'$  du graphe  $\Delta^\Phi$ , on a un morphisme de diagrammes  $\overline{\Delta(A)} : \overline{\Delta(N)} \rightarrow \overline{\Delta(N')}$  sur  $\mathcal{C}$

$$\overline{\Delta(A)} = (\Delta(A)^\Phi : \Delta(N)^\Phi \rightsquigarrow \Delta(N')^\Phi, \Delta(A) : \Delta(N) \rightsquigarrow \Delta(N') \circ \Delta(A)^\Phi),$$

où

- $\Delta(A)^\Phi : \Delta(N)^\Phi \rightsquigarrow \Delta(N')^\Phi$  est un morphisme généralisé de graphes, qui associe à tout arc  $a : n \rightarrow n'$  de  $\Delta(N)^\Phi$  un zigzag  $\Delta(A)^\Phi(a) : \Delta(A)^\Phi(n) \rightsquigarrow \Delta(A)^\Phi(n')$  du graphe  $\Delta(N')^\Phi$ ;
- $\Delta(A) : \Delta(N) \rightsquigarrow \Delta(N') \circ \Delta(A)^\Phi$  est une transformation naturelle généralisée, qui associe à tout nœud  $n$  de  $\Delta(N)^\Phi$  une flèche  $\Delta(A)_n : \Delta(N)(n) \rightarrow \Delta(N')(\Delta(A)^\Phi(n))$  de  $\mathcal{C}$ .

**Aplatissement  $\text{Apl}_{\mathcal{C}} : \text{DIAGR}^2(\mathcal{C}) \rightarrow \text{DIAGR}(\mathcal{C})$**

Étant donné une catégorie  $\mathcal{C}$ , le foncteur  $I_{\mathcal{C}} : \mathcal{C} \rightarrow \text{DIAGR}(\mathcal{C})$  permet de créer un objet de  $\text{DIAGR}(\mathcal{C})$  à partir d’un objet de  $\mathcal{C}$ , et une flèche de  $\text{DIAGR}(\mathcal{C})$  à partir d’une flèche de  $\mathcal{C}$ . La fonction d’aplatissement

$$\text{Apl}_{\mathcal{C}} : \text{DIAGR}^2(\mathcal{C}) \rightarrow \text{DIAGR}(\mathcal{C})$$

permet au contraire de “détruire de la structure”, en transformant tout diagramme de diagrammes (c’est-à-dire tout objet de  $\text{DIAGR}^2(\mathcal{C})$ ) en un simple diagramme (c’est-à-dire en un objet de  $\text{DIAGR}(\mathcal{C})$ ).

Considérons par exemple le diagramme  $\overline{\overline{\Delta}}$  sur  $\text{DIAGR}(\mathcal{C})$ , figure 2.2.

$$\overline{\overline{\Delta}} = (\Delta^\Phi, \Delta : \Delta^\Phi \rightarrow \text{DIAGR}(\mathcal{C}))$$

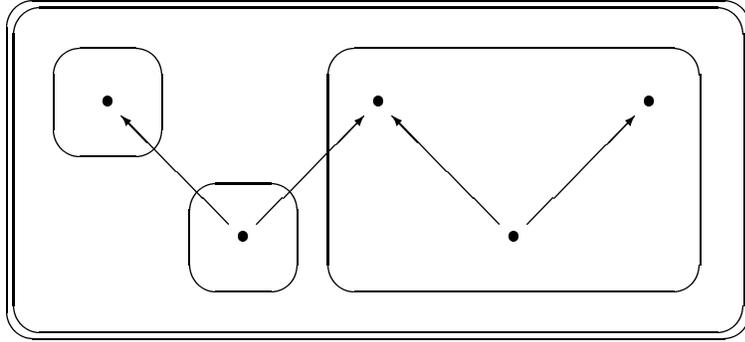


Figure 2.2 : diagramme  $\overline{\overline{\Delta}}$ , objet de  $\text{DIAGR}^2(\mathcal{C})$

Intuitivement, aplatiser le diagramme de diagrammes  $\overline{\overline{\Delta}}$  consiste à faire la réunion de tous les sous diagrammes de  $\overline{\overline{\Delta}}$ , et à transformer toutes les flèches de  $\text{DIAGR}(\mathcal{C})$  en (ensembles de) flèches de  $\mathcal{C}$  (cf. figure 2.3).

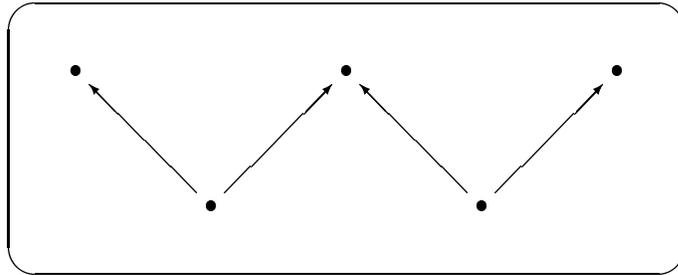


Figure 2.3 : Aplatissement du diagramme  $\overline{\overline{\Delta}}$   
 $\overline{\delta} = \text{Apl}_{\mathcal{C}}(\overline{\overline{\Delta}})$ , objet de  $\text{DIAGR}(\mathcal{C})$

**Définition 2.29** (Aplatissement  $\text{Apl}_{\mathcal{C}} : \text{DIAGR}^2(\mathcal{C}) \rightarrow \text{DIAGR}(\mathcal{C})$ )

Soit

$$\overline{\overline{\Delta}} = (\Delta^\Phi, \Delta : \Delta^\Phi \rightarrow \text{DIAGR}(\mathcal{C}))$$

un objet de  $\text{DIAGR}^2(\mathcal{C})$ . On définit le diagramme  $\overline{\delta} = \text{Apl}_{\mathcal{C}}(\overline{\overline{\Delta}})$  de  $\text{DIAGR}(\mathcal{C})$

$$\text{Apl}_{\mathcal{C}}(\overline{\overline{\Delta}}) = \overline{\delta} = (\delta^\Phi, \delta : \delta^\Phi \rightarrow \mathcal{C})$$

de la façon suivante.

- $\delta^\Phi$  est un graphe, déterminé par l'ensemble de ses nœuds et l'ensemble de ses arcs.

$$\text{Nœud}(\delta^\Phi) = \{ (N, n) ; N \in \text{Nœud}(\Delta^\Phi) ; n \in \text{Nœud}(\Delta(N)^\Phi) \}$$

$$\begin{aligned} \text{Arc}(\delta^\Phi) &= \{ (N, a) : (N, n) \rightarrow (N, n') ; \\ &\quad N \in \text{Nœud}(\Delta^\Phi) ; \\ &\quad n, n' \in \text{Nœud}(\Delta(N)^\Phi) ; \\ &\quad a : n \rightarrow n' \in \text{Arc}(\Delta(N)^\Phi) \} \\ &\cup \{ (A, n) : (N, n) \rightarrow (N', \Delta(A)^\Phi(n)) ; \\ &\quad N, N' \in \text{Nœud}(\Delta^\Phi) ; \\ &\quad A : N \rightarrow N' \in \text{Arc}(\Delta^\Phi) ; \\ &\quad n \in \text{Nœud}(\Delta(N)^\Phi) \} \end{aligned}$$

- $\delta : \delta^\Phi \rightarrow \mathcal{C}$  est un foncteur, défini sur les nœuds et arcs de  $\delta^\Phi$  de la façon suivante.

– Action sur les nœuds :

$$\delta(N, n) = \Delta(N)(n)$$

– Action sur les arcs :

$$\begin{aligned} \delta(N, a) &= \Delta(N)(a) \\ \delta(A, n) &= \Delta(A)_n \end{aligned}$$

**Définition 2.30** (Ensemble de flèches  $\{\overline{J_N} : \overline{\Delta(N)} \rightarrow \overline{\delta}, \forall N \in \text{Nœud}(\Delta^\Phi)\}$ )  
Pour tout nœud  $N$  de  $\Delta^\Phi$ , on définit une flèche  $\overline{J_N} : \overline{\Delta(N)} \rightarrow \overline{\delta}$  de  $\text{DIAGR}(\mathcal{C})$

$$\overline{J_N} : \overline{\Delta(N)} \rightarrow \overline{\delta} = (J_N^\Phi : \Delta(N)^\Phi \rightsquigarrow \delta^\Phi, J_N : \Delta(N) \rightsquigarrow \delta \circ J_N^\Phi)$$

Le foncteur généralisé  $J_N^\Phi : \Delta(N)^\Phi \rightsquigarrow \delta^\Phi$  est défini par

$$\begin{array}{ccc} J_N^\Phi : & \Delta(N)^\Phi & \rightsquigarrow & \delta^\Phi \\ & n & \mapsto & (N, n) \\ & a : n \rightarrow n' & \mapsto & (N, a) : (N, n) \rightarrow (N, n') \end{array}$$

Ce foncteur généralisé est en réalité un foncteur, car à tout arc de  $\Delta(N)^\Phi$  est associé un *arc* de  $\delta^\Phi$  (et non un zigzag quelconque).

La transformation naturelle généralisée  $J_N : \Delta(N) \rightsquigarrow \delta \circ J_N^\Phi$  associe à tout nœud  $n$  de  $\Delta(N)^\Phi$  la flèche de  $\mathcal{C}$

$$(J_N)_n = id_{\Delta(N)(n)} = id_{\delta(N, n)}.$$

Pour montrer que  $\overline{J_N} : \overline{\Delta(N)} \rightarrow \overline{\delta}$  est bien une flèche de  $\text{DIAGR}(\mathcal{C})$ , et comme  $J_N^\Phi$  est un foncteur, il suffit de vérifier que  $J_N$  est une transformation naturelle

$$J_N : \Delta(N) \rightarrow \delta \circ J_N^\Phi.$$

C'est le cas, car pour tout arc  $a : n \rightarrow n'$  de  $\Delta(N)^\Phi$ , par définition de  $\delta(N, a)$ , on a

$$(J_N)_n \circ \Delta(N)(a) = \delta(N, a) \circ (J_N)_{n'}.$$

**Remarque 2.6** Nous avons donc une famille de flèches

$$\overline{J_N} : \overline{\Delta(N)} \rightarrow \overline{\delta}, \quad \forall N \in \text{Nœud}(\Delta^\Phi).$$

Pour tout nœud  $n$  de  $\Delta(N)^\Phi$  on a donc une flèche

$$(J_N)_n : \Delta(N)(n) \rightarrow \delta(N, n).$$

Comme on a défini  $\delta(N, n)$  égal à  $\Delta(N)(n)$ , cela est cohérent avec la définition de départ

$$(J_N)_n = \text{id}_{\Delta(N)(n)} = \text{id}_{\delta(N, n)}.$$

En fait, il n'est pas toujours commode de confondre les objets  $\Delta(N)(n)$  et  $\delta(N, n)$ , parce que lors des raisonnements, le diagramme  $\overline{\Delta(N)}$  ou  $\overline{\delta}$  est en fait aussi important que l'objet  $\delta(N, n) = \Delta(N)(n)$  de  $\mathcal{C}$ .

Pour cette raison, pour tout arc  $a : n \rightarrow n'$  de  $\Delta(N)^\Phi$ , nous distinguerons les deux flèches

$$\begin{aligned} \Delta(N)(a) : \Delta(N)(n) &\rightarrow \Delta(N)(n'), \\ \delta(N, a) : \delta(N, n) &\rightarrow \delta(N, n'). \end{aligned}$$

De même, pour tout arc  $A : N \rightarrow N'$  de  $\Delta^\Phi$ , et pour tout nœud  $n$  de  $\Delta(N)^\Phi$ , nous distinguerons les deux flèches

$$\begin{aligned} \Delta(A)_n : \Delta(N)(n) &\rightarrow \Delta(N')(\Delta(A)^\Phi(n)), \\ \delta(A, n) : \delta(N, n) &\rightarrow \delta(N', \Delta(A)^\Phi(n)). \end{aligned}$$

Enfin, pour tout nœud  $N$  de  $\Delta^\Phi$ , et tout nœud  $n$  de  $\Delta(N)^\Phi$ , nous distinguerons également les trois flèches

$$\begin{aligned} (J_N)_n : \Delta(N)(n) &\rightarrow \delta(N, n), \\ \text{id}_{\Delta(N)(n)} : \Delta(N)(n) &\rightarrow \Delta(N)(n), \\ \text{id}_{\delta(N, n)} : \delta(N, n) &\rightarrow \delta(N, n). \end{aligned}$$

Pour respecter ces contraintes, nous n'utiliserons donc pas les deux égalités suivantes

$$\begin{aligned} \Delta(N)(a) &= \delta(N, a), \\ \Delta(A)_n &= \delta(A, n). \end{aligned}$$

À la place, nous utiliserons les deux propriétés équivalentes suivantes.

**Propriété 2.7**

1.  $\forall N \in \text{Nœud}(\Delta^\Phi), \forall a : n \rightarrow n' \in \text{Arc}(\Delta(N)^\Phi),$   
 $(J_N)_{n'} \circ \Delta(N)(a) = \delta(N, a) \circ (J_N)_n.$

$$\begin{array}{ccc}
 \delta(N, n) & \xrightarrow{\delta(N, a)} & \delta(N, n') \\
 \uparrow (J_N)_n & & \uparrow (J_N)_{n'} \\
 \Delta(N)(n) & \xrightarrow{\Delta(N)(a)} & \Delta(N)(n')
 \end{array}$$

2.  $\forall A : N \rightarrow N' \in \text{Arc}(\Delta^\Phi), \forall n \in \text{Nœud}(\Delta(N)^\Phi),$   
 $(J_{N'})_{\Delta(A)^\Phi(n)} \circ \Delta(A)_n = \delta(A, n) \circ (J_N)_n.$

$$\begin{array}{ccc}
 \delta(N, n) & \xrightarrow{\delta(A, n)} & \delta(N', \Delta(A)^\Phi(n)) \\
 \uparrow (J_N)_n & & \uparrow (J_{N'})_{\Delta(A)^\Phi(n)} \\
 \Delta(N)(n) & \xrightarrow{\Delta(A)_n} & \Delta(N')(\Delta(A)^\Phi(n))
 \end{array}$$

**Remarque 2.7** La famille de flèches de  $\text{DIAGR}(\mathcal{C})$

$$\overline{J_N} : \overline{\Delta(N)} \rightarrow \overline{\delta}, \quad \forall N \in \text{Nœud}(\Delta^\Phi)$$

ne permet pas de construire une flèche

$$\overline{J} : \overline{\Delta} \rightarrow I_{\text{DIAGR}(\mathcal{C})}(\overline{\delta})$$

dans la catégorie  $\text{DIAGR}^2(\mathcal{C})$ . En effet, il faudrait pour cela que pour tout arc  $A : N \rightarrow N'$  de  $\Delta^\Phi$ , on ait

$$\begin{aligned}
 \overline{J_{N'}} \circ \overline{\Delta(A)} &= \overline{J_N} \\
 \Leftrightarrow \forall n \in \text{Nœud}(\Delta(N)^\Phi), & (J_{N'})_{\Delta(A)^\Phi(n)} \circ \Delta(A)_n = (J_N)_n,
 \end{aligned}$$

ce qui n'est pas vérifié.

**Lemme 2.7** Pour tout diagramme  $\overline{\alpha}$  de  $\text{DIAGR}(\mathcal{C})$ , on a

$$\text{Apl}_{\mathcal{C}}(I_{\text{DIAGR}(\mathcal{C})}(\overline{\alpha})) = \overline{\alpha}.$$

*Preuve.* Posons  $\overline{\overline{\Delta}} = I_{\text{DIAGR}(\mathcal{C})}(\overline{\alpha})$  et  $\overline{\delta} = \text{Apl}_{\mathcal{C}}(\overline{\overline{\Delta}})$ . Le graphe  $\delta^{\Phi}$  est défini par

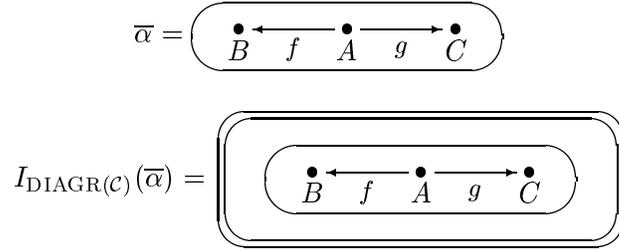
$$\begin{aligned} \text{Nœud}(\delta^{\Phi}) &= \{(*, n), \forall n \in \text{Nœud}(\alpha^{\Phi})\} \\ \text{Arc}(\delta^{\Phi}) &= \{(*, a) : (*, n) \rightarrow (*, n'), \forall a : n \rightarrow n' \in \text{Arc}(\alpha^{\Phi})\} \end{aligned}$$

donc les graphes  $\alpha^{\Phi}$  et  $\delta^{\Phi}$  sont isomorphes. D'autre part,

$$\begin{aligned} \forall n \in \text{Nœud}(\alpha^{\Phi}), \delta(*, n) &= \Delta(*) (n) = \alpha(n) \\ \forall a : n \rightarrow n' \in \text{Arc}(\alpha^{\Phi}), \delta(*, a) &= \Delta(*) (a) = \alpha(a). \end{aligned}$$

Par conséquent,  $\overline{\alpha} = \overline{\delta}$ . □

Donnons un exemple. On considère le diagramme  $\overline{\alpha}$  suivant.



Il est clair que l'aplatissement de  $I_{\text{DIAGR}(\mathcal{C})}(\overline{\alpha})$  donne le diagramme  $\overline{\alpha}$ .

**Lemme 2.8** *Pour tout diagramme  $\overline{\alpha}$  de  $\text{DIAGR}(\mathcal{C})$ , on a*

$$\text{Apl}_{\mathcal{C}}(\text{DIAGR}(I_{\mathcal{C}})(\overline{\alpha})) = \overline{\alpha}.$$

*Preuve.* Posons  $\overline{\overline{\Delta}} = \text{DIAGR}(I_{\mathcal{C}})(\overline{\alpha}) = I_{\mathcal{C}} \circ \overline{\alpha}$ , et  $\overline{\delta} = \text{Apl}_{\mathcal{C}}(\overline{\overline{\Delta}})$ . Le graphe  $\delta^{\Phi}$  est défini par

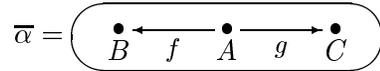
$$\begin{aligned} \text{Nœud}(\delta^{\Phi}) &= \{(n, *), \forall n \in \text{Nœud}(\alpha^{\Phi})\} \\ \text{Arc}(\delta^{\Phi}) &= \{(a, *) : (n, *) \rightarrow (n', *), \forall a : n \rightarrow n' \in \text{Arc}(\alpha^{\Phi})\} \end{aligned}$$

donc les graphes  $\alpha^{\Phi}$  et  $\delta^{\Phi}$  sont isomorphes. D'autre part,

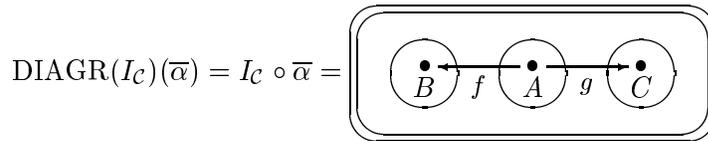
$$\begin{aligned} \delta(n, *) &= \Delta(n) (*) = I_{\mathcal{C}}(\alpha(n)) (*) = \alpha(n) \\ \delta(a, *) &= \Delta(a)_* = I_{\mathcal{C}}(\alpha(a))_* = \alpha(a). \end{aligned}$$

Par conséquent,  $\overline{\alpha} = \overline{\delta}$ . □

Illustrons ce résultat par un exemple. On considère le diagramme



On a donc



L'aplatissement de  $I_{\mathcal{C}} \circ \overline{\alpha}$  est bien égal au diagramme  $\overline{\alpha}$ .

### 2.3.7 Catégorie $\text{diagr}(\mathcal{C})$

Dans la catégorie  $\text{DIAGR}(\mathcal{C})$ , des morphismes de diagrammes différents peuvent avoir des colimites égales. Dans l'exemple 2.6, nous avons défini un morphisme de diagrammes  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\delta}_2$  tel que

$$\text{Colim } \bar{\sigma} = \text{up}(S, M, G, A_2, b \circ s, m \circ b \circ s, \&_1(A_2) \circ \&_1(MD), \&_2(A_2)).$$

La définition de  $\bar{\sigma}$  que nous avons donnée n'était pas la seule possible. Considérons par exemple le morphisme de diagrammes  $\bar{\tau} : \bar{\alpha} \rightarrow \bar{\delta}_2$  suivant.

- Morphisme généralisé de graphes  $\tau^\Phi$  :

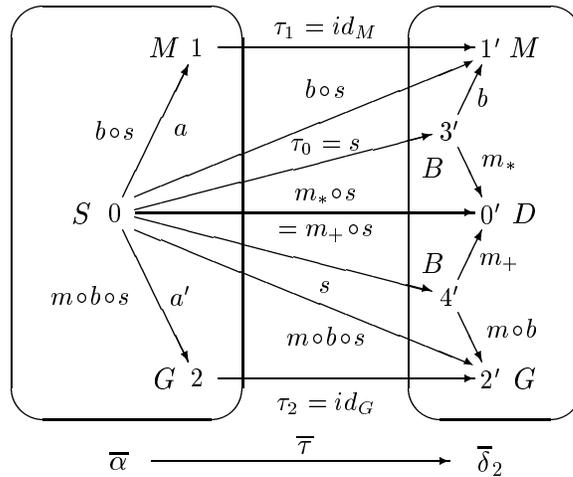
$$\begin{array}{lcl} \tau^\Phi & : & \alpha^\Phi \rightsquigarrow \delta_2^\Phi \\ & & 0 \mapsto 3' \\ & & 1 \mapsto 1' \\ & & 2 \mapsto 2' \\ & & a \mapsto b \\ & & a' \mapsto 3' \xrightarrow{m_*} 0' \xleftarrow{m_+} 4' \xrightarrow{m \circ b} 2' \end{array}$$

- Transformation naturelle  $\tau : \alpha \rightsquigarrow \delta_2 \circ \tau^\Phi$  :

$$\begin{array}{l} \tau_0 = s \\ \tau_1 = id_M \\ \tau_2 = id_G. \end{array}$$

Les colimites des morphismes de diagrammes  $\bar{\sigma}$  et  $\bar{\tau}$  sont égales.

$$\text{Colim } \bar{\tau} = \text{Colim } \bar{\sigma} = \text{up}(S, M, G, A_2, b \circ s, m \circ b \circ s, \&_1(A_2) \circ \&_1(MD), \&_2(A_2)).$$

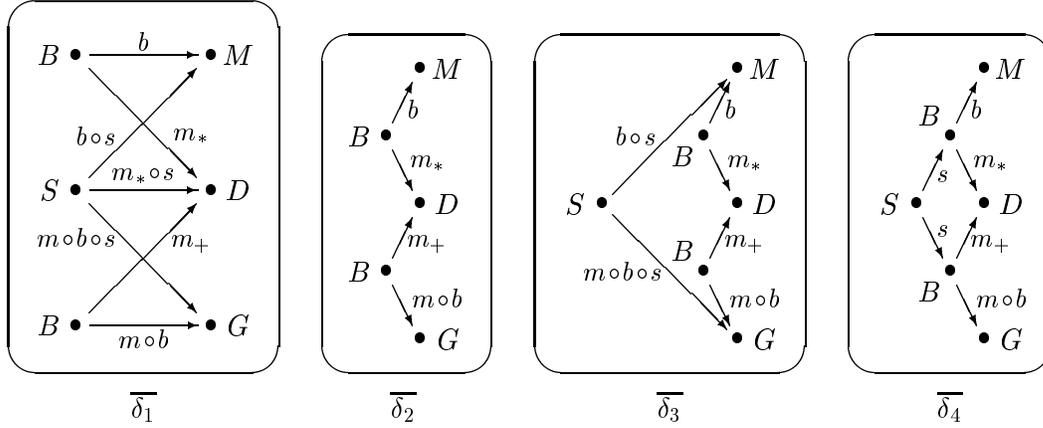


En fait, on peut choisir pour la flèche  $\tau_0$  une flèche quelconque du “faisceau” de flèches

$$\begin{array}{lcl} b \circ s & : & \alpha^\Phi(0) \rightarrow \delta_2^\Phi(1') \\ s & : & \alpha^\Phi(0) \rightarrow \delta_2^\Phi(3') \\ m_* \circ s = m_+ \circ s & : & \alpha^\Phi(0) \rightarrow \delta_2^\Phi(0') \\ s & : & \alpha^\Phi(0) \rightarrow \delta_2^\Phi(4') \\ m \circ b \circ s & : & \alpha^\Phi(0) \rightarrow \delta_2^\Phi(2'). \end{array}$$

Il faut évidemment définir  $\tau^\Phi$  en conséquence.

De même, deux diagrammes non isomorphes dans la catégorie  $\text{DIAGR}(\mathcal{C})$  peuvent avoir des colimites isomorphes. Par exemple, on peut vérifier que les diagrammes  $\bar{\delta}_1$ ,  $\bar{\delta}_2$ ,  $\bar{\delta}_3$  et  $\bar{\delta}_4$  définis dans le chapitre 1 (section 1.5) ont des colimites isomorphes.



Pour prouver que les colimites de ces quatre diagrammes sont isomorphes, il faut bien sûr utiliser les propriétés des colimites, ainsi que l'égalité entre flèches de  $\mathcal{C}_0$  :

$$m_+ \circ s = m_* \circ s.$$

Le but de ce paragraphe est de définir une catégorie dans laquelle les égalités entre flèches colimites et les isomorphismes entre objets colimites seront reflétées au niveau des diagrammes. Pour cela, nous définissons une relation de congruence  $\approx$  sur les flèches de  $\text{DIAGR}(\mathcal{C})$ , compatible avec les colimites, et  $\text{diagr}(\mathcal{C})$  est la catégorie quotient

$$\text{diagr}(\mathcal{C}) = \text{DIAGR}(\mathcal{C}) / \approx.$$

**Lemme 2.9** (La relation de connexion  $\sim_{\bar{\delta}}$  est une relation d'équivalence)

Soit un diagramme  $\bar{\delta}$  sur  $\mathcal{C}$ , et  $A$  un objet de  $\mathcal{C}$ . La relation  $\sim_{\bar{\delta}}$  est une relation d'équivalence sur l'ensemble de flèches de  $\mathcal{C}$

$$\{u : A \rightarrow \delta(n) ; n \in \text{Nœud}(\delta^\Phi)\}.$$

*Preuve.*

- La relation  $\sim_{\bar{\delta}}$  est réflexive. En effet, pour toute flèche  $u : A \rightarrow \delta(n)$  de  $\mathcal{C}$ ,

$$u \sim_{\bar{\delta}} u [0_n].$$

- La relation  $\sim_{\bar{\delta}}$  est symétrique. En effet, soit deux flèches  $u : A \rightarrow \delta(n_0)$  et  $v : A \rightarrow \delta(n_1)$  de  $\mathcal{C}$ . On a

$$u \sim_{\bar{\delta}} v [Z] \Rightarrow v \sim_{\bar{\delta}} u [\bar{Z}],$$

où  $\bar{Z} : n_1 \rightsquigarrow n_0$  est le zigzag opposé de  $Z : n_0 \rightsquigarrow n_1$  (cf. définition 2.4).

- La relation  $\sim_{\bar{\delta}}$  est transitive. En effet, soit trois flèches  $u : A \rightarrow \delta(n_0)$ ,  $v : A \rightarrow \delta(n_1)$  et  $w : A \rightarrow \delta(n_2)$  de  $\mathcal{C}$ . On a

$$u \sim_{\bar{\delta}} v [Z_1] \text{ et } v \sim_{\bar{\delta}} w [Z_2] \Rightarrow u \sim_{\bar{\delta}} w [Z],$$

où  $Z : n_0 \rightsquigarrow n_2$  est le zigzag obtenu en mettant bout à bout  $Z_1 : n_0 \rightsquigarrow n_1$  et  $Z_2 : n_1 \rightsquigarrow n_2$ .

□

**Lemme 2.10** ( $\sim_{\bar{\delta}}$  est une demi-congruence à droite)

Soit  $u : A \rightarrow \delta(n_0)$ ,  $v : A \rightarrow \delta(n_k)$  et  $w : A' \rightarrow A$ . On a

$$u \sim_{\bar{\delta}} v [Z] \Rightarrow u \circ w \sim_{\bar{\delta}} v \circ w [Z]$$

(cf. figure 2.4).

*Preuve.* Application immédiate de la définition de  $\sim_{\bar{\delta}}$ .

□

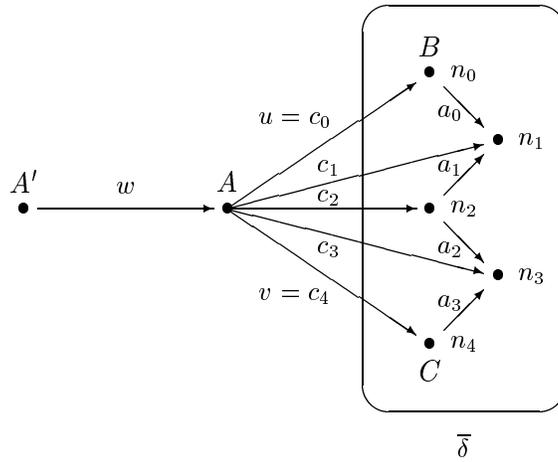


Figure 2.4 :  $u \circ w \sim_{\bar{\delta}} v \circ w [Z]$

**Lemme 2.11** Soit  $\bar{\alpha}$  et  $\bar{\beta}$  deux diagrammes sur  $\mathcal{C}$ . Soit  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta}$  une flèche de  $\text{DIAGR}(\mathcal{C})$ . Soit  $u : A \rightarrow \alpha(m)$  et  $v : A \rightarrow \alpha(n)$  deux flèches de  $\mathcal{C}$ . Alors,

$$u \sim_{\bar{\alpha}} v [Z] \Rightarrow \sigma_m \circ u \sim_{\bar{\beta}} \sigma_n \circ v [\sigma^\Phi(Z)].$$

*Preuve.* Application de la définition de  $\sim_{\bar{\beta}}$  et de  $\bar{\sigma}$ .  $\square$

**Définition 2.31** (Congruence  $\approx$ )

Soit deux diagrammes  $\bar{\alpha}$  et  $\bar{\beta}$ . Soit  $\bar{\sigma}, \bar{\tau} : \bar{\alpha} \rightarrow \bar{\beta}$  deux flèches de  $\text{DIAGR}(\mathcal{C})$ . On définit la relation  $\approx$  sur les flèches de  $\text{DIAGR}(\mathcal{C})$  par

$$\bar{\sigma} \approx \bar{\tau} \Leftrightarrow \forall n \in \text{Nœud}(\alpha^\Phi), \sigma_n \sim_{\bar{\beta}} \tau_n.$$

Remarquons que c'est une bonne définition car  $\forall n \in \text{Nœud}(\alpha^\Phi)$ ,

$$\begin{aligned} \sigma_n &: \alpha(n) \rightarrow \beta(\sigma^\Phi(n)), \\ \tau_n &: \alpha(n) \rightarrow \beta(\tau^\Phi(n)). \end{aligned}$$

**Proposition 2.1** *La relation  $\approx$  est une congruence.*

*Preuve.*

- La relation  $\approx$  est réflexive car  $\sim_{\bar{\beta}}$  est réflexive. Soit  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta}$  une flèche de  $\text{DIAGR}(\mathcal{C})$ .

$$\forall n \in \text{Nœud}(\alpha^\Phi), \sigma_n \sim_{\bar{\beta}} \sigma_n \Rightarrow \bar{\sigma} \approx \bar{\sigma}$$

- La relation  $\approx$  est symétrique car  $\sim_{\bar{\beta}}$  est symétrique. Soit  $\bar{\sigma}, \bar{\tau} : \bar{\alpha} \rightarrow \bar{\beta}$  deux flèches de  $\text{DIAGR}(\mathcal{C})$ .

$$\begin{aligned} \bar{\sigma} \approx \bar{\tau} &\Rightarrow \forall n \in \text{Nœud}(\alpha^\Phi), \sigma_n \sim_{\bar{\beta}} \tau_n && \text{(définition de } \approx) \\ &\Rightarrow \forall n \in \text{Nœud}(\alpha^\Phi), \tau_n \sim_{\bar{\beta}} \sigma_n && (\sim_{\bar{\beta}} \text{ symétrique)} \\ &\Rightarrow \bar{\tau} \approx \bar{\sigma} && \text{(définition de } \approx) \end{aligned}$$

- De même, la relation  $\approx$  est transitive car  $\sim_{\bar{\beta}}$  est transitive.
- Soit  $\bar{\sigma}, \bar{\tau} : \bar{\alpha} \rightarrow \bar{\beta}$  et  $\bar{\chi} : \bar{\delta} \rightarrow \bar{\alpha}$  trois flèches de  $\text{DIAGR}(\mathcal{C})$ . Montrons que

$$\bar{\sigma} \approx \bar{\tau} \Rightarrow \bar{\sigma} \circ \bar{\chi} \approx \bar{\tau} \circ \bar{\chi}.$$

Supposons  $\bar{\sigma} \approx \bar{\tau}$ . Il suffit de montrer que

$$\forall n \in \text{Nœud}(\delta^\Phi), \sigma_{\chi^\Phi(n)} \circ \chi_n \sim_{\bar{\beta}} \tau_{\chi^\Phi(n)} \circ \chi_n.$$

$$\begin{aligned} \bar{\sigma} \approx \bar{\tau} &\Rightarrow \forall n \in \text{Nœud}(\delta^\Phi), \sigma_{\chi^\Phi(n)} \sim_{\bar{\beta}} \tau_{\chi^\Phi(n)} && \text{(définition de } \approx) \\ &\Rightarrow \forall n \in \text{Nœud}(\delta^\Phi), \sigma_{\chi^\Phi(n)} \circ \chi_n \sim_{\bar{\beta}} \tau_{\chi^\Phi(n)} \circ \chi_n && \text{(lemme 2.10)} \end{aligned}$$

- Soit  $\bar{\sigma}, \bar{\tau} : \bar{\alpha} \rightarrow \bar{\beta}$  et  $\bar{\xi} : \bar{\beta} \rightarrow \bar{\gamma}$  trois flèches de  $\text{DIAGR}(\mathcal{C})$ . Montrons que

$$\bar{\sigma} \approx \bar{\tau} \Rightarrow \bar{\xi} \circ \bar{\sigma} \approx \bar{\xi} \circ \bar{\tau}.$$

Supposons que  $\bar{\sigma} \approx \bar{\tau}$ . Il suffit de montrer que

$$\forall n \in \text{Nœud}(\alpha^\Phi), \xi_{\sigma^\Phi(n)} \circ \sigma_n \sim_{\bar{\gamma}} \xi_{\tau^\Phi(n)} \circ \tau_n.$$

$$\begin{aligned} \bar{\sigma} \approx \bar{\tau} &\Rightarrow \forall n \in \text{Nœud}(\alpha^\Phi), \sigma_n \sim_{\bar{\beta}} \tau_n && \text{(définition de } \approx \text{)} \\ &\Rightarrow \forall n \in \text{Nœud}(\alpha^\Phi), \xi_{\sigma^\Phi(n)} \circ \sigma_n \sim_{\bar{\gamma}} \xi_{\tau^\Phi(n)} \circ \tau_n && \text{(lemme 2.11)} \end{aligned}$$

□

**Lemme 2.12** *Soit deux catégories  $\mathcal{C}$  et  $\mathcal{D}$  et un foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ . Soit deux flèches  $\bar{\sigma}, \bar{\tau} : \bar{\alpha} \rightarrow \bar{\beta}$  de  $\text{DIAGR}(\mathcal{C})$ . Alors,*

$$\bar{\sigma} \approx \bar{\tau} \Rightarrow \text{DIAGR}(F)(\bar{\sigma}) \approx \text{DIAGR}(F)(\bar{\tau}).$$

*Preuve.*

$$\begin{aligned} \bar{\sigma} \approx \bar{\tau} & \\ \Rightarrow \forall n \in \text{Nœud}(\alpha^\Phi), \sigma_n \sim_{\bar{\beta}} \tau_n && \text{(définition de } \approx \text{)} \\ \Rightarrow \forall n \in \text{Nœud}(\alpha^\Phi), F(\sigma_n) \sim_{F \circ \bar{\beta}} F(\tau_n) && \text{(lemme 2.3)} \\ \Rightarrow \text{DIAGR}(F)(\bar{\sigma}) \approx \text{DIAGR}(F)(\bar{\tau}) && \text{(définitions de } \text{DIAGR}(F) \text{ et } \approx \text{)} \end{aligned}$$

□

**Définition 2.32** (Catégorie  $\text{diagr}(\mathcal{C})$ )

Comme  $\approx$  est une congruence (proposition 2.1), nous pouvons définir la catégorie  $\text{diagr}(\mathcal{C})$  comme le quotient de  $\text{DIAGR}(\mathcal{C})$  par  $\approx$ .

$$\text{diagr}(\mathcal{C}) = \text{DIAGR}(\mathcal{C}) / \approx.$$

Les objets de  $\text{diagr}(\mathcal{C})$  sont les objets de  $\text{DIAGR}(\mathcal{C})$ . Étant donné deux objets  $\bar{\alpha}$  et  $\bar{\beta}$  de  $\text{diagr}(\mathcal{C})$ , une flèche entre  $\bar{\alpha}$  et  $\bar{\beta}$  est une classe d'équivalence modulo  $\approx$  de flèches de  $\text{DIAGR}(\mathcal{C})$  entre  $\bar{\alpha}$  et  $\bar{\beta}$ .

On a un foncteur de projection

$$\begin{aligned} [-]_{\mathcal{C}} : \text{DIAGR}(\mathcal{C}) &\rightarrow \text{diagr}(\mathcal{C}) \\ \bar{\alpha} &\mapsto \bar{\alpha} \\ \bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta} &\mapsto [\bar{\sigma}] : \bar{\alpha} \rightarrow \bar{\beta} \end{aligned}$$

qui est l'identité sur les objets, et associe à toute flèche  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta}$  de  $\text{DIAGR}(\mathcal{C})$  sa classe d'équivalence  $[\bar{\sigma}] : \bar{\alpha} \rightarrow \bar{\beta}$  modulo  $\approx$  dans  $\text{diagr}(\mathcal{C})$ .

Comme le foncteur de projection  $[-]_{\mathcal{C}} : \text{DIAGR}(\mathcal{C}) \rightarrow \text{diagr}(\mathcal{C})$  est l'identité sur les objets, on notera de la même façon un objet  $\bar{\alpha}$  de  $\text{DIAGR}(\mathcal{C})$  et son image par  $[-]_{\mathcal{C}}$  dans  $\text{diagr}(\mathcal{C})$ .

**Propriété 2.8** Soit  $F : \text{DIAGR}(\mathcal{C}) \rightarrow \mathcal{D}$  un foncteur compatible avec  $\approx$ , c'est-à-dire tel que pour toutes flèches  $\bar{\sigma}, \bar{\tau} : \bar{\alpha} \rightarrow \bar{\beta}$  de  $\text{DIAGR}(\mathcal{C})$ ,

$$\bar{\sigma} \approx \bar{\tau} \Rightarrow F(\bar{\sigma}) = F(\bar{\tau}).$$

Alors, le foncteur  $F$  se factorise à travers  $[-]_{\mathcal{C}}$ . Autrement dit, il existe un unique foncteur  $G : \text{diagr}(\mathcal{C}) \rightarrow \mathcal{D}$  tel que  $F = G \circ [-]_{\mathcal{C}}$ .

**Définition 2.33** (Foncteur  $\text{diagr}(F)$ ) Soit deux catégories  $\mathcal{C}$  et  $\mathcal{D}$ , et un foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ . D'après le lemme 2.12, le foncteur  $[-]_{\mathcal{D}} \circ \text{DIAGR}(F) : \text{DIAGR}(\mathcal{C}) \rightarrow \text{diagr}(\mathcal{D})$  est compatible avec  $\approx$ . Donc, d'après la propriété 2.8, il existe un unique foncteur

$$\text{diagr}(F) : \text{diagr}(\mathcal{C}) \rightarrow \text{diagr}(\mathcal{D})$$

tel que  $\text{diagr}(F) \circ [-]_{\mathcal{C}} = [-]_{\mathcal{D}} \circ \text{DIAGR}(F)$ .

**Lemme 2.13** L'application

$$\begin{array}{ccc} \text{diagr} : & \mathbf{CAT} & \rightarrow & \mathbf{CAT} \\ & \mathcal{C} & \mapsto & \text{diagr}(\mathcal{C}) \\ & F : \mathcal{C} \rightarrow \mathcal{D} & \mapsto & \text{diagr}(F) : \text{diagr}(\mathcal{C}) \rightarrow \text{diagr}(\mathcal{D}) \end{array}$$

est un foncteur  $\text{diagr} : \mathbf{CAT} \rightarrow \mathbf{CAT}$ .

*Preuve.* Application immédiate de la définition de  $\text{diagr}$  sur les flèches.  $\square$

**Propriété 2.9**

L'application  $[-]$  qui à toute catégorie  $\mathcal{C}$  associe le foncteur de projection

$$[-]_{\mathcal{C}} : \text{DIAGR}(\mathcal{C}) \rightarrow \text{diagr}(\mathcal{C})$$

est une transformation naturelle

$$[-] : \text{DIAGR} \rightarrow \text{diagr}.$$

*Preuve.* Résulte immédiatement de la définition de  $\text{diagr}(F)$ .  $\square$

**Lemme 2.14** Soit deux foncteurs  $G, G' : \text{diagr}(\mathcal{C}) \rightarrow \mathcal{D}$ . Alors,

$$G \circ [-]_{\mathcal{C}} \cong G' \circ [-]_{\mathcal{C}} \Rightarrow G \cong G'.$$

*Preuve.* Soit  $\omega : G \circ [-]_{\mathcal{C}} \rightarrow G' \circ [-]_{\mathcal{C}}$  l'isomorphisme naturel entre  $G \circ [-]_{\mathcal{C}}$  et  $G' \circ [-]_{\mathcal{C}}$ . Comme  $[-]_{\mathcal{C}} : \text{DIAGR}(\mathcal{C}) \rightarrow \text{diagr}(\mathcal{C})$  est l'identité sur les objets on peut définir une transformation naturelle  $\xi : G \rightarrow G'$  par

$$\xi_{\bar{\alpha}} = \omega_{\bar{\alpha}}.$$

Cela définit bien une transformation naturelle car pour toute flèche  $\bar{\tau} : \bar{\alpha} \rightarrow \bar{\beta}$  de  $\text{diagr}(\mathcal{C})$ , il existe une flèche  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta}$  de  $\text{diagr}(\mathcal{C})$  telle que  $\bar{\tau} = [\bar{\sigma}]$ . Comme  $\omega$  est une transformation naturelle,

$$\begin{aligned} \omega_{\bar{\beta}} \circ G([\bar{\sigma}]) &= G'([\bar{\sigma}]) \circ \omega_{\bar{\alpha}} \\ \Rightarrow \xi_{\bar{\beta}} \circ G(\bar{\tau}) &= G'(\bar{\tau}) \circ \xi_{\bar{\alpha}} \end{aligned}$$

De plus,  $\xi_{\bar{\alpha}}$  est un isomorphisme pour tout diagramme  $\bar{\alpha}$  de  $\text{diagr}(\mathcal{C})$ , donc on a bien un isomorphisme naturel

$$G \cong G'.$$

□

Jusqu'à la fin de ce paragraphe, nous considérons une catégorie  $\mathcal{C}$  finiment cocomplète. Nous allons montrer que les colimites sont compatibles avec la relation de congruence  $\approx$ , c'est-à-dire que le foncteur  $\text{Colim}_{\mathcal{C}}$  envoie des flèches équivalentes sur des colimites égales. Intuitivement, cette propriété prouve que la relation de congruence n'est pas trop "forte", c'est-à-dire n'identifie pas trop de flèches, et permet de définir un foncteur

$$\text{colim}_{\mathcal{C}} : \text{diagr}(\mathcal{C}) \rightarrow \mathcal{C}.$$

**Lemme 2.15** *Soit un diagramme  $\bar{\alpha}$  de  $\text{diagr}(\mathcal{C})$ ,  $B$  un objet de  $\mathcal{C}$ , et deux flèches  $\bar{\sigma}, \bar{\tau} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(B)$ . Alors,*

$$\bar{\sigma} \approx \bar{\tau} \Rightarrow \bar{\sigma} = \bar{\tau}.$$

*Preuve.* D'abord,  $\sigma^{\Phi} = \tau^{\Phi}$  car  $I_{\mathcal{C}}(B)$  est construit sur le graphe  $1^{\Phi}$ , qui ne comporte qu'un seul nœud  $*$ , et qu'un seul zigzag  $0_* : * \rightsquigarrow *$ . D'autre part, comme  $\bar{\sigma} \approx \bar{\tau}$ , pour tout nœud  $n$  de  $\alpha^{\Phi}$ , on a  $\sigma_n \sim_{I_{\mathcal{C}}(B)} \tau_n$ . Comme  $0_* : * \rightsquigarrow *$  est le seul zigzag de  $1^{\Phi}$ , on a nécessairement  $\sigma_n = \tau_n$ . Par conséquent,  $\bar{\sigma} = \bar{\tau}$ . □

**Proposition 2.2** *(Le foncteur  $\text{Colim}_{\mathcal{C}}$  est compatible avec  $\approx$ )*

*Soit  $\mathcal{C}$  une catégorie finiment cocomplète. Le foncteur  $\text{Colim}_{\mathcal{C}}$  est compatible avec  $\approx$ . Autrement dit, soit  $\bar{\alpha}$  et  $\bar{\beta}$  deux diagrammes et  $\bar{\sigma}, \bar{\tau} : \bar{\alpha} \rightarrow \bar{\beta}$  deux flèches de  $\text{DIAGR}(\mathcal{C})$ . Alors,*

$$\bar{\sigma} \approx \bar{\tau} \Rightarrow \text{Colim}_{\mathcal{C}} \bar{\sigma} = \text{Colim}_{\mathcal{C}} \bar{\tau}.$$

*Preuve.* Soit  $(A, \bar{\lambda} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(A))$  et  $(B, \bar{\lambda}' : \bar{\beta} \rightarrow I_{\mathcal{C}}(B))$  les cônes colimites des diagrammes  $\bar{\alpha}$  et  $\bar{\beta}$ .

$$\begin{aligned}
\bar{\sigma} &\approx \bar{\tau} \\
\Rightarrow \bar{\lambda}' \circ \bar{\tau} &\approx \bar{\lambda}' \circ \bar{\sigma} && (\approx \text{ est une congruence}) \\
\Rightarrow \bar{\lambda}' \circ \bar{\tau} &= \bar{\lambda}' \circ \bar{\sigma} && (\text{lemme 2.15}) \\
\Rightarrow I_{\mathcal{C}}(\text{Colim}_{\mathcal{C}} \bar{\tau}) \circ \bar{\lambda} &= \bar{\lambda}' \circ \bar{\sigma} && (\text{définition de } \text{Colim}_{\mathcal{C}} \bar{\tau}) \\
\Rightarrow \text{Colim}_{\mathcal{C}} \bar{\sigma} &= \text{Colim}_{\mathcal{C}} \bar{\tau} && (\text{définition de } \text{Colim}_{\mathcal{C}} \bar{\sigma})
\end{aligned}$$

□

**Corollaire 2.1** *D'après la propriété 2.8, le foncteur  $\text{Colim}_{\mathcal{C}} : \text{DIAGR}(\mathcal{C}) \rightarrow \mathcal{C}$  se factorise à travers  $[-]_{\mathcal{C}}$ . Autrement dit, il existe un unique foncteur*

$$\text{colim}_{\mathcal{C}} : \text{diagr}(\mathcal{C}) \rightarrow \mathcal{C}$$

tel que  $\text{colim}_{\mathcal{C}} \circ [-]_{\mathcal{C}} = \text{Colim}_{\mathcal{C}}$ .

**Remarque 2.8** Dans la catégorie  $\text{diagr}(\mathcal{C})$ , il existe plus d'isomorphismes entre diagrammes, et plus d'égalités entre flèches de diagrammes que dans la catégorie  $\text{DIAGR}(\mathcal{C})$ . Néanmoins, dans le cas général,

1.  $\text{colim}_{\mathcal{C}} \bar{\sigma} = \text{colim}_{\mathcal{C}} \bar{\tau} \not\cong \bar{\sigma} = \bar{\tau}$ ;
2.  $\text{colim}_{\mathcal{C}} \bar{\alpha} \cong \text{colim}_{\mathcal{C}} \bar{\beta} \not\cong \bar{\alpha} \cong \bar{\beta}$ .

La raison intuitive est qu'on peut avoir des égalités entre flèches de  $\mathcal{C}$  qui ne proviennent pas des propriétés générales des colimites, mais de la catégorie  $\mathcal{C}$  elle-même.

*Contreexemple.* On considère la catégorie  $(\mathbb{N}, /)$  définie de la façon suivante.

- Les objets de  $(\mathbb{N}, /)$  sont les entiers naturels.
- On a une flèche unique de  $n$  vers  $m$  si et seulement si  $n$  divise  $m$ .

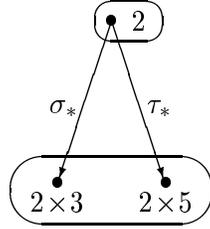
$(\mathbb{N}, /)$  est une catégorie finiment cocomplète. (Par exemple, la somme catégorielle de  $m$  et  $n$  est le *ppcm* de  $m$  et  $n$ .)

Les diagrammes  $\bar{\alpha}$  et  $\bar{\beta}$  dessinés ci-dessous, ne sont pas isomorphes dans  $\text{diagr}(\mathbb{N}, /)$ , et ont pourtant des colimites isomorphes.



Les deux diagrammes ont en effet pour colimite l'entier  $2 \times 3 \times 5$ .

De même, les flèches  $\bar{\sigma}$  et  $\bar{\tau}$ , dessinées ci-dessous, ne sont pas égales dans  $\text{diagr}(\mathbb{N}, /)$ , et ont pourtant des colimites égales : la flèche de l'entier 2 vers l'entier  $2 \times 3 \times 5$ .



Ici, la définition de la catégorie indique que deux flèches entre deux objets sont forcément égales. C'est la raison pour laquelle  $\text{colim}_{\mathcal{C}} \bar{\sigma} = \text{colim}_{\mathcal{C}} \bar{\tau}$ . Dans une autre catégorie, il n'y a *a priori* aucune raison pour que cette égalité soit vérifiée.

**Théorème 2.5** *Soit  $\mathcal{C}$  une catégorie finiment cocomplète.*

1. *Les deux foncteurs*

$$\begin{aligned} \text{colim}_{\mathcal{C}} : \text{diagr}(\mathcal{C}) &\rightarrow \mathcal{C} \\ [-]_{\mathcal{C}} \circ I_{\mathcal{C}} : \mathcal{C} &\rightarrow \text{diagr}(\mathcal{C}) \end{aligned}$$

*forment une adjonction*

$$(\text{colim}_{\mathcal{C}} \dashv [-]_{\mathcal{C}} \circ I_{\mathcal{C}}).$$

2. *La coïunité de cette adjonction est la coïunité de l'adjonction*

$$(\text{Colim}_{\mathcal{C}} \dashv I_{\mathcal{C}}).$$

*Preuve.*

1. On définit une transformation naturelle  $\bar{\eta}' : Id_{\text{diagr}(\mathcal{C})} \rightrightarrows [-]_{\mathcal{C}} \circ I_{\mathcal{C}} \circ \text{colim}_{\mathcal{C}}$  de la façon suivante. Pour tout objet  $\bar{\alpha}$  de  $\text{diagr}(\mathcal{C})$ , on pose

$$\bar{\eta}'_{\bar{\alpha}} = [\bar{\eta}_{\bar{\alpha}}]$$

où  $\bar{\eta} : Id_{\text{DIAGR}(\mathcal{C})} \rightrightarrows I_{\mathcal{C}} \circ \text{Colim}_{\mathcal{C}}$  est l'unité de l'adjonction  $(\text{Colim}_{\mathcal{C}} \dashv I_{\mathcal{C}})$ . C'est une bonne définition car le foncteur  $[-]_{\mathcal{C}}$  est l'identité sur les objets. On vérifie facilement que  $\bar{\eta}'$  est bien une transformation naturelle.

Montrons maintenant que  $(\text{colim}_{\mathcal{C}} \dashv [-]_{\mathcal{C}} \circ I_{\mathcal{C}})$  est une adjonction dont l'unité est  $\bar{\eta}'$ . Soit un diagramme  $\bar{\alpha}$  de  $\text{diagr}(\mathcal{C})$ ,  $B$  un objet de  $\mathcal{C}$ , et  $[\bar{\sigma}] : \bar{\alpha} \rightarrow I_{\mathcal{C}}(B)$  une flèche de  $\text{diagr}(\mathcal{C})$ . Il faut montrer qu'il existe une unique flèche de  $\mathcal{C}$

$$\phi : \text{colim}_{\mathcal{C}} \bar{\alpha} \rightarrow B$$

telle que

$$[I_{\mathcal{C}}(\phi)] \circ [\bar{\eta}'_{\bar{\alpha}}] = [\bar{\sigma}].$$

*Existence.* Comme  $(\text{Colim}_{\mathcal{C}} \dashv I_{\mathcal{C}})$  est une adjonction dont l'unité est  $\bar{\eta} : \text{Id}_{\text{DIAGR}(\mathcal{C})} \dashv I_{\mathcal{C}} \circ \text{Colim}_{\mathcal{C}}$ , il existe une unique flèche  $\phi : \text{Colim}_{\mathcal{C}} \bar{\alpha} \rightarrow B$  telle que

$$I_{\mathcal{C}}(\phi) \circ \bar{\eta}_{\bar{\alpha}} = \bar{\sigma}.$$

En considérant  $\bar{\alpha}$  comme un objet de  $\text{diagr}(\mathcal{C})$ , la flèche  $\phi$  a pour domaine  $\text{colim}_{\mathcal{C}} \bar{\alpha}$ .

$$\begin{aligned} I_{\mathcal{C}}(\phi) \circ \bar{\eta}_{\bar{\alpha}} &= \bar{\sigma} \\ \Rightarrow [I_{\mathcal{C}}(\phi) \circ \bar{\eta}_{\bar{\alpha}}] &= [\bar{\sigma}] \\ \Rightarrow [I_{\mathcal{C}}(\phi)] \circ [\bar{\eta}_{\bar{\alpha}}] &= [\bar{\sigma}] \quad ([-]_{\mathcal{C}} \text{ foncteur}) \\ \Rightarrow [I_{\mathcal{C}}(\phi)] \circ \bar{\eta}'_{\bar{\alpha}} &= [\bar{\sigma}] \quad (\text{définition de } \bar{\eta}') \end{aligned}$$

*Unicité.* Soit  $\phi, \psi : \text{colim}_{\mathcal{C}} \alpha \rightarrow B$  deux flèches de  $\mathcal{C}$  telles que

$$\begin{aligned} [I_{\mathcal{C}}(\phi)] \circ \bar{\eta}'_{\bar{\alpha}} &= [\bar{\sigma}], \\ [I_{\mathcal{C}}(\psi)] \circ \bar{\eta}'_{\bar{\alpha}} &= [\bar{\sigma}]. \end{aligned}$$

On montre que  $\phi = \psi$ .

$$\begin{aligned} [I_{\mathcal{C}}(\phi)] \circ \bar{\eta}'_{\bar{\alpha}} &= [I_{\mathcal{C}}(\psi)] \circ \bar{\eta}'_{\bar{\alpha}} \\ \Rightarrow [I_{\mathcal{C}}(\phi)] \circ [\bar{\eta}_{\bar{\alpha}}] &= [I_{\mathcal{C}}(\psi)] \circ [\bar{\eta}_{\bar{\alpha}}] \quad (\text{définition de } \bar{\eta}') \\ \Rightarrow [I_{\mathcal{C}}(\phi) \circ \bar{\eta}_{\bar{\alpha}}] &= [I_{\mathcal{C}}(\psi) \circ \bar{\eta}_{\bar{\alpha}}] \quad ([-]_{\mathcal{C}} \text{ foncteur}) \\ \Rightarrow I_{\mathcal{C}}(\phi) \circ \bar{\eta}_{\bar{\alpha}} &\approx I_{\mathcal{C}}(\psi) \circ \bar{\eta}_{\bar{\alpha}} \quad (\text{définition de } \approx) \\ \Rightarrow I_{\mathcal{C}}(\phi) \circ \bar{\eta}_{\bar{\alpha}} &= I_{\mathcal{C}}(\psi) \circ \bar{\eta}_{\bar{\alpha}} \quad (\text{lemme 2.15}) \\ \Rightarrow \phi &= \psi \quad (\text{adjonction } (\text{Colim}_{\mathcal{C}} \dashv I_{\mathcal{C}})) \end{aligned}$$

2. Considérons  $\epsilon : \text{Colim}_{\mathcal{C}} \circ I_{\mathcal{C}} \dashv \text{Id}_{\mathcal{C}}$  la coïunité de l'adjonction  $(\text{Colim}_{\mathcal{C}} \dashv I_{\mathcal{C}})$ . Comme  $\text{Colim}_{\mathcal{C}} = \text{colim}_{\mathcal{C}} \circ [-]_{\mathcal{C}}$ ,  $\epsilon$  est bien une transformation naturelle

$$\epsilon : \text{colim}_{\mathcal{C}} \circ [-]_{\mathcal{C}} \circ I_{\mathcal{C}} \dashv \text{Id}_{\mathcal{C}}.$$

Pour montrer que  $\epsilon$  est la coïunité de l'adjonction  $(\text{colim}_{\mathcal{C}} \dashv [-]_{\mathcal{C}} \circ I_{\mathcal{C}}) : \text{diagr}(\mathcal{C}) \rightarrow \mathcal{C}$ , on utilise le théorème 2.1.

Soit une flèche  $g : \text{colim}_{\mathcal{C}} \bar{\alpha} \rightarrow B$  de  $\mathcal{C}$ . Il faut montrer qu'il existe une unique flèche  $[\bar{\sigma}] : \bar{\alpha} \rightarrow I_{\mathcal{C}}(B)$  de  $\text{diagr}(\mathcal{C})$  telle que

$$\epsilon_B \circ \text{colim}_{\mathcal{C}}[\bar{\sigma}] = g.$$

*Existence.* Comme  $(\text{Colim}_{\mathcal{C}} \dashv I_{\mathcal{C}})$  est une adjonction, il existe une unique flèche  $\bar{\sigma} : \bar{\alpha} \rightarrow I_{\mathcal{C}}(B)$  de  $\text{DIAGR}(\mathcal{C})$  telle que

$$\begin{aligned} \epsilon_B \circ \text{Colim}_{\mathcal{C}} \bar{\sigma} &= g \\ \Leftrightarrow \epsilon_B \circ \text{colim}_{\mathcal{C}}[\bar{\sigma}] &= g \quad (\text{définition de } \text{colim}_{\mathcal{C}}) \end{aligned}$$

*Unicité.* Soit  $[\bar{\sigma}], [\bar{\tau}] : \bar{\alpha} \rightarrow I_{\mathcal{C}}(B)$  deux flèches de  $\text{diagr}(\mathcal{C})$  telles que

$$\begin{aligned}\epsilon_B \circ \text{colim}_{\mathcal{C}}[\bar{\sigma}] &= g, \\ \epsilon_B \circ \text{colim}_{\mathcal{C}}[\bar{\tau}] &= g.\end{aligned}$$

On montre que  $[\bar{\sigma}] = [\bar{\tau}]$ .

$$\begin{aligned}\epsilon_B \circ \text{colim}_{\mathcal{C}}[\bar{\sigma}] &= \epsilon_B \circ \text{colim}_{\mathcal{C}}[\bar{\tau}] \\ \Rightarrow \epsilon_B \circ \text{Colim}_{\mathcal{C}} \bar{\sigma} &= \epsilon_B \circ \text{Colim}_{\mathcal{C}} \bar{\tau} \quad (\text{définition de } \text{colim}_{\mathcal{C}}) \\ \Rightarrow \bar{\sigma} &= \bar{\tau} \quad (\text{adjonction } (\text{Colim}_{\mathcal{C}} \dashv I_{\mathcal{C}})) \\ \Rightarrow [\bar{\sigma}] &= [\bar{\tau}].\end{aligned}$$

□

### 2.3.8 La catégorie $\text{diagr}(\mathcal{C})$ est finiment cocomplète

Dans ce paragraphe, nous considérons une catégorie  $\mathcal{C}$  quelconque, et nous allons montrer que la catégorie  $\text{diagr}(\mathcal{C})$  est finiment cocomplète. Pour cela nous devons montrer que tout diagramme  $\overline{\overline{\Gamma}}$  de  $\text{DIAGR}(\text{diagr}(\mathcal{C}))$  a une colimite dans  $\text{diagr}(\mathcal{C})$ . Intuitivement, pour construire une colimite de  $\overline{\overline{\Gamma}}$ , on commence par construire un diagramme  $\overline{\overline{\Delta}}$  de  $\text{DIAGR}^2(\mathcal{C})$  construit sur le même graphe sous-jacent que  $\overline{\overline{\Gamma}}$ , tel que pour tout arc  $A : N \rightarrow N'$ ,  $\overline{\overline{\Delta}}(A)$  est un représentant de la classe d'équivalence  $\overline{\overline{\Gamma}}(A)$  de flèches de  $\text{DIAGR}(\mathcal{C})$ . Ensuite, l'aplatissement de  $\overline{\overline{\Delta}}$  donne une colimite du diagramme  $\overline{\overline{\Gamma}}$ .

#### Lemme 2.16

Soit un objet  $\overline{\overline{\Gamma}}$  de  $\text{DIAGR}(\text{diagr}(\mathcal{C}))$ . Il existe un objet  $\overline{\overline{\Delta}}$  de  $\text{DIAGR}^2(\mathcal{C})$  tel que

$$\overline{\overline{\Gamma}} = \text{DIAGR}([\_ ]_{\mathcal{C}})(\overline{\overline{\Delta}}) = [\_ ]_{\mathcal{C}} \circ \overline{\overline{\Delta}}.$$

*Preuve.* On définit un diagramme

$$\overline{\overline{\Delta}} = (\Delta^{\Phi}, \Delta : \Delta^{\Phi} \rightarrow \text{DIAGR}(\mathcal{C}))$$

de la façon suivante :

- Le graphe sous-jacent de  $\overline{\overline{\Delta}}$  est le graphe sous-jacent de  $\overline{\overline{\Gamma}}$ . Autrement dit,

$$\Delta^{\Phi} = \Gamma^{\Phi}.$$

- Le foncteur  $\Delta : \Delta^{\Phi} \rightarrow \text{DIAGR}(\mathcal{C})$  est défini par son action sur les nœuds et sur les arcs du graphe  $\Delta^{\Phi}$ .

- Tout nœud  $N$  de  $\Delta^{\Phi}$  est étiqueté par  $\overline{\overline{\Gamma}}(N)$ .

$$\overline{\overline{\Delta}}(N) = \overline{\overline{\Gamma}}(N);$$

- Pour tout arc  $A : N \rightarrow N'$  de  $\Delta^\Phi$ ,  $\overline{\Gamma(A)} : \overline{\Gamma(N)} \rightarrow \overline{\Gamma(N')}$  est une flèche de  $\text{diagr}(\mathcal{C})$ , c'est-à-dire une classe d'équivalence de flèches de  $\text{DIAGR}(\mathcal{C})$ . On *choisit* un représentant  $\overline{\Delta(A)}$  de cette classe d'équivalence. Autrement dit,  $\overline{\Delta(A)}$  est tel que

$$\overline{\Gamma(A)} = [\overline{\Delta(A)}].$$

Cela définit bien un foncteur car

$$\overline{\Delta(A)} : \overline{\Delta(N)} \rightarrow \overline{\Delta(N')}.$$

On vérifie immédiatement que

$$\overline{\overline{\Gamma}} = [-]_{\mathcal{C}} \circ \overline{\overline{\Delta}}.$$

□

Soit  $\overline{\overline{\Gamma}}$  un objet de  $\text{DIAGR}(\text{diagr}(\mathcal{C}))$ . D'après le lemme 2.16, il existe un objet  $\overline{\overline{\Delta}}$  de  $\text{DIAGR}^2(\mathcal{C})$  tel que  $\overline{\overline{\Gamma}} = [-]_{\mathcal{C}} \circ \overline{\overline{\Delta}}$ . Comme dans le paragraphe 2.3.6, on note

$$\overline{\delta} = \text{Apl}_{\mathcal{C}}(\overline{\overline{\Delta}})$$

l'aplatissement de  $\overline{\overline{\Delta}}$ , et pour tout nœud  $N$  de  $\Delta^\Phi$ , on a une famille de flèches de  $\text{DIAGR}(\mathcal{C})$

$$\overline{J_N} : \overline{\Delta(N)} \rightarrow \overline{\delta}.$$

On avait vu en remarque 2.7 que la famille de flèches

$$\overline{J_N} : \overline{\Delta(N)} \rightarrow \overline{\delta}$$

de  $\text{DIAGR}(\mathcal{C})$  ne permet pas de définir une flèche

$$\overline{\overline{J}} : \overline{\overline{\Delta}} \rightarrow I_{\text{DIAGR}(\mathcal{C})}(\overline{\delta})$$

dans la catégorie  $\text{DIAGR}^2(\mathcal{C})$ . Par contre on a une flèche de  $\overline{\overline{\Gamma}}$  vers  $I_{\text{diagr}(\mathcal{C})}(\overline{\delta})$  dans la catégorie  $\text{DIAGR}(\text{diagr}(\mathcal{C}))$ , grâce au lemme suivant.

**Lemme 2.17** *Pour tout arc  $A : N \rightarrow N'$  de  $\Delta^\Phi$ ,*

$$\overline{J_{N'}} \circ \overline{\Delta(A)} \approx \overline{J_N}.$$

*Preuve.* Il suffit de montrer que

$$\forall n \in \text{Nœud}(\Delta(N)^\Phi), (J_{N'})_{\Delta(A)^\Phi(n)} \circ \Delta(A)_n \sim_{\overline{\delta}} (J_N)_n.$$

Cette relation est vérifiée, car par définition de  $\delta$ , on a

$$(J_{N'})_{\Delta(A)^\Phi(n)} \circ \Delta(A)_n = \delta(A, n) \circ (J_N)_n.$$

□

On définit une famille de flèches de  $\text{diagr}(\mathcal{C})$ , en posant, pour tout nœud  $N$  de  $\Delta^\Phi$ ,

$$\overline{\eta_N} = [\overline{J_N}] : \overline{\Gamma(N)} \rightarrow \overline{\delta}.$$

**Lemme 2.18**  $(\overline{\delta}, \overline{\eta} : \overline{\overline{\Gamma}} \rightarrow I_{\text{diagr}(\mathcal{C})}(\overline{\delta}))$  est un cône sur  $\overline{\overline{\Gamma}}$ .

*Preuve.* Il suffit de montrer que  $\overline{\eta} : \overline{\Gamma} \rightarrow I_{\text{diagr}(\mathcal{C})}(\overline{\delta})$  est une flèche de la catégorie  $\text{DIAGR}(\text{diagr}(\mathcal{C}))$ , c'est-à-dire que pour tout arc  $A : N \rightarrow N'$  de  $\Delta(N)^\Phi$ , on a

$$\overline{\eta}_{N'} \circ \overline{\Gamma}(A) = \overline{\eta}_N.$$

Pour tout arc  $A : N \rightarrow N'$  de  $\Gamma(N)^\Phi$ , on a

$$\begin{aligned} \overline{J}_{N'} \circ \overline{\Delta}(A) &\approx \overline{J}_N && \text{(lemme 2.17)} \\ \Rightarrow [\overline{J}_{N'} \circ \overline{\Delta}(A)] &= [\overline{J}_N] && \text{(définition de } [-]_{\mathcal{C}} \text{)} \\ \Rightarrow [\overline{J}_{N'}] \circ [\overline{\Delta}(A)] &= [\overline{J}_N] && \text{([} [-]_{\mathcal{C}} \text{ est un foncteur)} \\ \Rightarrow [\overline{J}_{N'}] \circ \overline{\Gamma}(A) &= [\overline{J}_N] && \text{(définition de } \overline{\Delta}(A) \text{)} \\ \Rightarrow \overline{\eta}_{N'} \circ \overline{\Gamma}(A) &= \overline{\eta}_N && \text{(définition de } \overline{\eta}_N \text{ et } \overline{\eta}_{N'} \text{)} \end{aligned}$$

Par conséquent,  $(\overline{\delta}, \overline{\eta} : \overline{\Gamma} \rightarrow I_{\text{diagr}(\mathcal{C})}(\overline{\delta}))$  est un cône sur  $\overline{\Gamma}$ .  $\square$

**Lemme 2.19** *Soit un ensemble de flèches de  $\text{DIAGR}(\mathcal{C})$*

$$\{\overline{K}_N : \overline{\Delta}(N) \rightarrow \overline{\alpha}, \forall N \in \text{Nœud}(\Delta^\Phi)\}$$

*tel que*

$$\forall A : N \rightarrow N' \in \text{Arc}(\Delta^\Phi), \overline{K}_{N'} \circ \overline{\Delta}(A) \approx \overline{K}_N.$$

*Alors,*

1. *il existe une flèche  $\overline{\sigma} : \overline{\delta} \rightarrow \overline{\alpha}$  de  $\text{DIAGR}(\mathcal{C})$  telle que*

$$\forall N \in \text{Nœud}(\Delta^\Phi), \overline{\sigma} \circ \overline{J}_N \approx \overline{K}_N;$$

2. *pour toute flèche  $\overline{\tau} : \overline{\delta} \rightarrow \overline{\alpha}$  de  $\text{DIAGR}(\mathcal{C})$  telle que*

$$\forall N \in \text{Nœud}(\Delta^\Phi), \overline{\tau} \circ \overline{J}_N \approx \overline{K}_N,$$

*on a*

$$\overline{\sigma} \approx \overline{\tau}.$$

*Preuve.* Soit un ensemble de flèches de  $\text{DIAGR}(\mathcal{C})$

$$\{\overline{K}_N : \overline{\Delta}(N) \rightarrow \overline{\alpha}, \forall N \in \text{Nœud}(\Delta^\Phi)\}$$

tel que  $\forall A : N \rightarrow N' \in \text{Arc}(\Delta^\Phi), \overline{K}_{N'} \circ \overline{\Delta}(A) \approx \overline{K}_N$ .

1. Considérons la flèche de  $\text{DIAGR}(\mathcal{C})$

$$\overline{\sigma} : \overline{\delta} \rightarrow \overline{\alpha} = (\sigma^\Phi : \delta^\Phi \rightsquigarrow \alpha^\Phi, \sigma : \delta \rightsquigarrow \alpha \circ \sigma^\Phi)$$

définie de la façon suivante.

- $\sigma^\Phi : \delta^\Phi \rightsquigarrow \alpha^\Phi$  est un morphisme généralisé de graphes.

- $\forall (N, n) \in \text{Nœud}(\delta^\Phi)$ ,  $\sigma^\Phi(N, n) = K_N^\Phi(n)$ .
- $\forall (N, a) : (N, n) \rightarrow (N, n') \in \text{Arc}(\delta^\Phi)$ ,  
 $\sigma^\Phi(N, a) = K_N^\Phi(a) : K_N^\Phi(n) \rightarrow K_N^\Phi(n')$ .
- $\forall (A, n) : (N, n) \rightarrow (N', \Delta(A)^\Phi(n)) \in \text{Arc}(\delta^\Phi)$ ,  $\sigma^\Phi(A, n) = Z$ ,  
 où  $Z : K_N^\Phi(n) \rightsquigarrow K_{N'}^\Phi(\Delta(A)(n))$  est le zigzag tel que

$$(K_{N'}^\Phi)_{\Delta(A)^\Phi(n)} \circ \Delta(A)_n \sim_{\bar{\alpha}} (K_N)_n [Z].$$

- $\sigma : \delta \rightsquigarrow \alpha \circ \sigma^\Phi$  est la transformation naturelle généralisée définie par

$$\forall (N, n) \in \text{Nœud}(\delta^\Phi), \sigma_{(N, n)} = (K_N)_n \circ (J_N)_n^{-1}.$$

Pour prouver que cela définit bien une flèche de  $\text{DIAGR}(\mathcal{C})$ , il faut montrer que  $\sigma : \delta \rightsquigarrow \alpha \circ \sigma^\Phi$  est bien une transformation naturelle généralisée c'est-à-dire que

- (a)  $\forall (N, a) : (N, n) \rightarrow (N, n') \in \text{Arc}(\delta^\Phi)$ ,  
 $\sigma_{(N, n')} \circ \delta(N, a) \sim_{\bar{\alpha}} \sigma_{(N, n)} [\sigma^\Phi(N, a)]$
- (b)  $\forall (A, n) : (N, n) \rightarrow (N', \Delta(A)^\Phi(n)) \in \text{Arc}(\delta^\Phi)$ ,  
 $\sigma_{(N', \Delta(A)^\Phi(n))} \circ \delta(A, n) \sim_{\bar{\alpha}} \sigma_{(N, n)} [\sigma^\Phi(A, n)]$

- (a) Comme  $K_N$  est une flèche de  $\text{DIAGR}(\mathcal{C})$ ,

$$\begin{aligned} & (K_N)_{n'} \circ \Delta(N)(a) \sim_{\bar{\alpha}} (K_N)_n [\sigma^\Phi(N, a)] \\ \Rightarrow & (K_N)_{n'} \circ \Delta(N)(a) \circ (J_N)_{n'}^{-1} \sim_{\bar{\alpha}} (K_N)_n \circ (J_N)_n^{-1} \quad (\text{lemme 2.10}) \\ \Rightarrow & (K_N)_{n'} \circ (J_N)_{n'} \circ \delta(N, a) \sim_{\bar{\alpha}} (K_N)_n \circ (J_N)_n^{-1} \quad (\text{définition de } \delta(N, a)) \\ \Rightarrow & \sigma_{(N, n')} \circ \delta(N, a) \sim_{\bar{\alpha}} \sigma_{(N, n)} [\sigma^\Phi(N, a)] \quad (\text{définition de } \sigma_{(N, n)}) \end{aligned}$$

- (b) Par définition de  $\sigma(A, n)^\Phi$ ,

$$\begin{aligned} & (K_{N'}^\Phi)_{\Delta(A)^\Phi(n)} \circ \Delta(A)_n \sim_{\bar{\alpha}} (K_N)_n [\sigma^\Phi(A, n)] \\ \Rightarrow & (K_{N'}^\Phi)_{\Delta(A)^\Phi(n)} \circ \Delta(A)_n \circ (J_N)_{n'}^{-1} \sim_{\bar{\alpha}} (K_N)_n \circ (J_N)_n^{-1} [\sigma^\Phi(A, n)] \\ \Rightarrow & (K_{N'}^\Phi)_{\Delta(A)^\Phi(n)} \circ (J_{N'}^\Phi)_{\Delta(A)^\Phi(n)}^{-1} \circ \delta(A, n) \sim_{\bar{\alpha}} (K_N)_n \circ (J_N)_n^{-1} [\sigma^\Phi(A, n)] \\ \Rightarrow & \sigma_{(N', \Delta(A)^\Phi(n))} \circ \delta(A, n) \sim_{\bar{\alpha}} \sigma_{(N, n)} [\sigma(A, n)^\Phi] \end{aligned}$$

Il reste à montrer que  $\forall N \in \text{Nœud}(\Delta^\Phi)$ ,  $\bar{\sigma} \circ \overline{J_N} \approx \overline{K_N}$ . Par définition de  $\bar{\sigma}$ ,

$$\begin{aligned} & \forall (N, n) \in \text{Nœud}(\delta^\Phi), \sigma_{(N, n)} = (K_N)_n \circ (J_N)_n^{-1} \\ \Rightarrow & \forall (N, n) \in \text{Nœud}(\delta^\Phi), \sigma_{(N, n)} \circ (J_N)_n = (K_N)_n \\ \Rightarrow & \forall N \in \text{Nœud}(\Delta^\Phi), \forall n \in \text{Nœud}(\Delta(N)^\Phi), \sigma_{(N, n)} \circ (J_N)_n \sim_{\bar{\alpha}} (K_N)_n \\ \Rightarrow & \forall N \in \text{Nœud}(\Delta^\Phi), \bar{\sigma} \circ \overline{J_N} \approx \overline{K_N} \end{aligned}$$

2. Soit  $\bar{\tau} : \bar{\delta} \rightarrow \bar{\alpha}$  une flèche de  $\text{DIAGR}(\mathcal{C})$  telle que

$$\forall N \in \text{Nœud}(\Delta^\Phi), \bar{\tau} \circ \overline{J_N} \approx \overline{K_N}.$$

On montre que  $\bar{\sigma} \approx \bar{\tau}$ .

$$\begin{aligned}
& \forall N \in \text{Nœud}(\Delta^\Phi), \bar{\tau} \circ \overline{J_N} \approx \overline{K_N} \\
& \Rightarrow \forall N, \forall n \in \text{Nœud}(\Delta(N)^\Phi), \tau_{(N,n)} \circ (J_N)_n \sim_{\bar{\alpha}} (K_N)_n \\
& \Rightarrow \forall (N, n) \in \text{Nœud}(\delta^\Phi), \tau_{(N,n)} \circ (J_N)_n \sim_{\bar{\alpha}} (K_N)_n \\
& \Rightarrow \forall (N, n) \in \text{Nœud}(\delta^\Phi), \tau_{(N,n)} \sim_{\bar{\alpha}} (K_N)_n \circ (J_N)_n^{-1} \\
& \Rightarrow \forall (N, n) \in \text{Nœud}(\delta^\Phi), \tau_{(N,n)} \sim_{\bar{\alpha}} \sigma_{(N,n)} \\
& \Rightarrow \bar{\tau} \approx \bar{\sigma}
\end{aligned}$$

□

**Lemme 2.20** *Le cône  $(\bar{\delta}, \bar{\eta} : \bar{\Gamma} \rightarrow I_{\text{diagr}(\mathcal{C})}(\bar{\delta}))$  est un cône colimite de  $\bar{\Gamma}$ .*

*Preuve.* Soit  $(\bar{\alpha}, \bar{\lambda} : \bar{\Gamma} \rightarrow I_{\text{diagr}(\mathcal{C})}(\bar{\alpha}))$  un autre cône sur  $\bar{\Gamma}$ . Nous devons montrer qu'il existe une unique flèche

$$[\bar{\sigma}] : \bar{\delta} \rightarrow \bar{\alpha}$$

de  $\text{diagr}(\mathcal{C})$  telle que

$$I_{\text{diagr}(\mathcal{C})}([\bar{\sigma}]) \circ \bar{\eta} = \bar{\lambda}.$$

*Existence.* Pour tout nœud  $N$  de  $\Gamma^\Phi$ ,  $\overline{\lambda_N} : \overline{\Gamma(N)} \rightarrow \bar{\alpha}$  est une flèche de  $\text{diagr}(\mathcal{C})$ . On choisit un représentant  $\overline{K_N} : \overline{\Gamma(N)} \rightarrow \bar{\alpha}$  dans  $\text{DIAGR}(\mathcal{C})$  de  $\overline{\lambda_N}$ . Autrement dit, on a

$$\forall N \in \text{Nœud}(\Gamma^\Phi), [\overline{K_N}] = \overline{\lambda_N}.$$

Comme  $\bar{\lambda} : \bar{\Gamma} \rightarrow I_{\text{diagr}(\mathcal{C})}(\bar{\alpha})$  est une flèche de  $\text{DIAGR}(\text{diagr}(\mathcal{C}))$ , pour tout arc  $A : N \rightarrow N'$  du graphe  $\Gamma^\Phi = \Delta^\Phi$ , on a

$$\begin{aligned}
& \overline{\lambda_{N'}} \circ \overline{\Gamma(A)} = \overline{\lambda_N} \\
& \Rightarrow [\overline{K_{N'}}] \circ [\overline{\Delta(A)}] = [\overline{K_N}] \quad (\text{définition de } \overline{K_N}, \overline{K_{N'}} \text{ et } \overline{\Delta}) \\
& \Rightarrow [\overline{K_{N'}} \circ \overline{\Delta(A)}] = [\overline{K_N}] \quad ([-]_{\mathcal{C}} \text{ foncteur}) \\
& \Rightarrow \overline{K_{N'}} \circ \overline{\Delta(A)} \approx \overline{K_N} \quad (\text{définition de } \text{diagr}(\mathcal{C}))
\end{aligned}$$

donc, d'après le lemme 2.19 (1), il existe une flèche  $\bar{\sigma} : \bar{\delta} \rightarrow \bar{\alpha}$  de  $\text{DIAGR}(\mathcal{C})$  telle que

$$\forall N \in \text{Nœud}(\Delta^\Phi), \bar{\sigma} \circ \overline{J_N} \approx \overline{K_N}.$$

$$\begin{aligned}
& \forall N \in \text{Nœud}(\Gamma^\Phi), \bar{\sigma} \circ \overline{J_N} \approx \overline{K_N} \\
& \Rightarrow \forall N \in \text{Nœud}(\Gamma^\Phi), [\bar{\sigma} \circ \overline{J_N}] = [\overline{K_N}] \\
& \Rightarrow \forall N \in \text{Nœud}(\Gamma^\Phi), [\bar{\sigma}] \circ [\overline{J_N}] = [\overline{K_N}] \\
& \Rightarrow \forall N \in \text{Nœud}(\Gamma^\Phi), [\bar{\sigma}] \circ \overline{\eta_N} = \overline{\lambda_N} \\
& \Rightarrow I_{\text{diagr}(\mathcal{C})}([\bar{\sigma}]) \circ \bar{\eta} = \bar{\lambda}
\end{aligned}$$

*Unicité.* Soit  $[\bar{\tau}] : \bar{\delta} \rightarrow \bar{\alpha}$  une flèche de  $\text{diagr}(\mathcal{C})$  telle que

$$I_{\text{diagr}(\mathcal{C})}([\bar{\tau}]) \circ \bar{\eta} = \bar{\lambda}.$$

$$\begin{aligned} I_{\text{diagr}(\mathcal{C})}([\bar{\sigma}]) \circ \bar{\eta} &= \bar{\lambda} \\ \Rightarrow \forall N \in \text{Nœud}(\Gamma^\Phi), [\bar{\tau}] \circ \bar{\eta}_N &= \bar{\lambda}_N \\ \Rightarrow \forall N \in \text{Nœud}(\Gamma^\Phi), [\bar{\tau}] \circ [\bar{J}_N] &= [\bar{K}_N] \\ \Rightarrow \forall N \in \text{Nœud}(\Gamma^\Phi), [\bar{\tau} \circ \bar{J}_N] &= [\bar{K}_N] \\ \Rightarrow \forall N \in \text{Nœud}(\Gamma^\Phi), \bar{\tau} \circ \bar{J}_N &\approx \bar{K}_N \end{aligned}$$

donc d'après le lemme 2.19 (2),  $\sigma \approx \tau$ , et par conséquent  $[\sigma] = [\tau]$ . □

**Théorème 2.6** *La catégorie  $\text{diagr}(\mathcal{C})$  est finiment cocomplète.*

*Preuve.* Conséquence immédiate du lemme 2.20. □

**Remarque 2.9** Dans cette preuve, nous avons montré l'existence d'une colimite pour le diagramme  $\bar{\Gamma}$ . Pour chaque arc de  $\Gamma^\Phi$ , nous avons effectué un choix de représentant  $\bar{\Delta}(A)$  pour la flèche  $\bar{\Gamma}(A)$ . Un choix différent de représentant conduit à un autre diagramme colimite (isomorphe) pour  $\bar{\Gamma}$ . Nous avons donc défini la colimite de  $\bar{\Gamma}$  à un isomorphisme près.

Une colimite est définie à un isomorphisme près, mais lorsque les classes d'équivalence  $\bar{\Gamma}(A)$  ne contiennent qu'un élément, alors il n'existe qu'un seul diagramme  $\bar{\Delta}$  tel que  $\bar{\Gamma} = [-]_{\mathcal{C}} \circ \bar{\Delta}$ . Soit un diagramme  $\bar{\alpha}$  de  $\text{DIAGR}(\mathcal{C})$ . Considérons le diagramme de  $\text{DIAGR}(\text{diagr}(\mathcal{C}))$

$$\bar{\Gamma} = \text{DIAGR}([-]_{\mathcal{C}} \circ I_{\mathcal{C}})(\bar{\alpha}) = [-]_{\mathcal{C}} \circ I_{\mathcal{C}} \circ \bar{\alpha}.$$

Pour tout arc  $a : n \rightarrow n'$  du graphe  $\Gamma^\Phi = \alpha^\Phi$ ,

$$\Gamma(a) = [I_{\mathcal{C}}(\alpha(a))],$$

et le seul élément de la classe  $\Gamma(a)$  est  $I_{\mathcal{C}}(\alpha(a))$ . Donc

$$\bar{\Delta} = I_{\mathcal{C}} \circ \bar{\alpha} = \text{DIAGR}(I_{\mathcal{C}})(\bar{\alpha}).$$

D'après le lemme 2.8,

$$\text{Apl}_{\mathcal{C}}(\bar{\Delta}) = \text{Apl}_{\mathcal{C}}(\text{DIAGR}(I_{\mathcal{C}})(\bar{\alpha})) = \bar{\alpha}.$$

Finalement, la colimite de  $\bar{\Gamma}$  est donc  $\bar{\alpha}$ .

**Lemme 2.21** *Soit une catégorie  $\mathcal{C}$ .*

$$\text{Colim}_{\text{diagr}(\mathcal{C})} \circ \text{DIAGR}([-]_{\mathcal{C}} \circ I_{\mathcal{C}}) = [-]_{\mathcal{C}}.$$

*Preuve.*

- Pour tout diagramme  $\bar{\alpha}$  de  $\text{DIAGR}(\mathcal{C})$ ,

$$\text{Colim}_{\text{diagr}(\mathcal{C})}(\text{DIAGR}([-]_{\mathcal{C}} \circ I_{\mathcal{C}})(\bar{\alpha})) = \bar{\alpha}.$$

- Soit une flèche  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta}$  de  $\text{DIAGR}(\mathcal{C})$ . Soit

$$(\bar{\alpha}, \bar{\eta} : [-]_{\mathcal{C}} \circ I_{\mathcal{C}} \circ \bar{\alpha} \rightarrow I_{\text{diagr}(\mathcal{C})}(\bar{\alpha}))$$

et

$$(\bar{\beta}, \bar{\eta}' : [-]_{\mathcal{C}} \circ I_{\mathcal{C}} \circ \bar{\beta} \rightarrow I_{\text{diagr}(\mathcal{C})}(\bar{\beta}))$$

les cône colimites respectifs des diagrammes  $[-]_{\mathcal{C}} \circ I_{\mathcal{C}} \circ \bar{\alpha}$  et  $[-]_{\mathcal{C}} \circ I_{\mathcal{C}} \circ \bar{\beta}$ .

Posons

$$\bar{\bar{\Sigma}} = [-]_{\mathcal{C}} \circ I_{\mathcal{C}} \circ \bar{\sigma}.$$

Il faut montrer que

$$\text{Colim}_{\text{diagr}(\mathcal{C})} \bar{\bar{\Sigma}} = [\bar{\sigma}].$$

Comme  $\text{Colim}_{\text{diagr}(\mathcal{C})} \bar{\bar{\Sigma}}$  est l'unique flèche telle que

$$I_{\text{diagr}(\mathcal{C})}(\text{Colim}_{\text{diagr}(\mathcal{C})} \bar{\bar{\Sigma}}) \circ \bar{\eta} = \bar{\eta}' \circ \bar{\bar{\Sigma}},$$

il suffit de prouver

$$\begin{aligned} I_{\text{diagr}(\mathcal{C})}([\bar{\sigma}]) \circ \bar{\eta} &= \bar{\eta}' \circ \bar{\bar{\Sigma}} \\ \Leftrightarrow \forall n \in \text{Nœud}(\alpha^{\Phi}), [\bar{\sigma}] \circ \bar{\eta}_n &= \bar{\eta}'_{\sigma^{\Phi}(n)} \circ \bar{\bar{\Sigma}}_n. \end{aligned}$$

□

**Remarque 2.10** Ce lemme montre que le foncteur  $[-]_{\mathcal{C}} \circ I_{\mathcal{C}} : \mathcal{C} \rightarrow \text{diagr}(\mathcal{C})$  est dense, c'est-à-dire que tout diagramme  $\bar{\alpha}$  est colimite d'un diagramme construit sur les images des objets et flèches de  $\mathcal{C}$  dans  $\text{diagr}(\mathcal{C})$ . En effet, tout diagramme  $\bar{\alpha}$  de  $\text{diagr}(\mathcal{C})$  est la colimite du diagramme  $\bar{\bar{\Gamma}} = \text{DIAGR}([-]_{\mathcal{C}} \circ I_{\mathcal{C}})(\bar{\alpha})$ .

### 2.3.9 La catégorie $\text{diagr}(\mathcal{C})$ est une complétion de $\mathcal{C}$ par colimites finies

La catégorie  $\text{diagr}(\mathcal{C})$  a une propriété importante: c'est une complétion de  $\mathcal{C}$  par colimites finies. Intuitivement, cela signifie que  $\text{diagr}(\mathcal{C})$  est une catégorie "minimum" qui contient  $\mathcal{C}$ , et toutes les colimites finies que l'on peut construire à partir d'objets et de flèches de  $\mathcal{C}$ .

Soit une catégorie  $\mathcal{C}$  quelconque, et  $\mathcal{D}$  une catégorie finiment cocomplète. Soit un foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ . On considère le foncteur

$$G = \text{Colim}_{\mathcal{D}} \circ \text{DIAGR}(F) : \text{DIAGR}(\mathcal{C}) \rightarrow \mathcal{D}.$$

**Lemme 2.22** On a un isomorphisme naturel  $G \circ I_{\mathcal{C}} \cong F$ .

*Preuve.* D'après le théorème 2.4,  $\epsilon : \text{Colim}_{\mathcal{D}} \circ I_{\mathcal{D}} \rightarrow Id_{\mathcal{D}}$ , coïté de l'adjonction  $(\text{Colim}_{\mathcal{D}} \dashv I_{\mathcal{D}})$  est un isomorphisme naturel. Par conséquent,

$$\begin{aligned} \text{Colim}_{\mathcal{D}} \circ I_{\mathcal{D}} &\cong Id_{\mathcal{D}} \\ \Rightarrow \text{Colim}_{\mathcal{D}} \circ I_{\mathcal{D}} \circ F &\cong F && \text{(composition par } F) \\ \Rightarrow \text{Colim}_{\mathcal{D}} \circ \text{DIAGR}(F) \circ I_{\mathcal{C}} &\cong F && (I : Id_{\mathbf{CAT}} \rightarrow \text{DIAGR}) \\ \Rightarrow G \circ I_{\mathcal{C}} &\cong F && \text{(définition de } G) \end{aligned}$$

□

Soit  $\overline{\overline{\Delta}}$  un objet de  $\text{DIAGR}^2(\mathcal{C})$ ,  $\overline{\delta} = \text{Apl}_{\mathcal{C}}(\overline{\overline{\Delta}})$  l'aplatissement de  $\overline{\overline{\Delta}}$ , et la famille de flèches de  $\text{DIAGR}(\mathcal{C})$

$$\overline{J}_N : \overline{\Delta(N)} \rightarrow \overline{\delta}, \quad \forall N \in \text{Nœud}(\Delta^{\Phi}).$$

Pour tout nœud  $N$  de  $\Delta^{\Phi}$ , on considère la flèche

$$\lambda_N = G(\overline{J}_N) : G(\overline{\Delta(N)}) \rightarrow G(\overline{\delta})$$

dans  $\mathcal{D}$ .

Nous allons considérer les diagrammes suivants.

- Le diagramme

$$F \circ \overline{\delta} = (\delta^{\Phi}, F \circ \delta : \delta^{\Phi} \rightarrow \mathcal{D})$$

est un objet de  $\text{DIAGR}(\mathcal{D})$ , qui a pour cône colimite

$$(G(\overline{\delta}), \overline{\eta}_{F \circ \overline{\delta}} : F \circ \overline{\delta} \rightarrow I_{\mathcal{D}}(G(\overline{\delta}))).$$

- Pour tout nœud  $N$  de  $\Delta^{\Phi}$ , on a un diagramme

$$F \circ \overline{\Delta(N)} = (\Delta(N)^{\Phi}, F \circ \Delta(N) : \Delta(N)^{\Phi} \rightarrow \mathcal{D})$$

dans  $\text{DIAGR}(\mathcal{D})$ , qui a pour cône colimite

$$(G(\overline{\Delta(N)}), \overline{\eta}_{F \circ \overline{\Delta(N)}} : F \circ \overline{\Delta(N)} \rightarrow I_{\mathcal{D}}(G(\overline{\Delta(N)}))).$$

Par définition,  $\lambda_N = G(\overline{J}_N) = \text{Colim}_{\mathcal{D}}(\text{DIAGR}(F)(\overline{J}_N))$ , donc  $\lambda_N$  est l'unique flèche telle que

$$I_{\mathcal{D}}(\lambda_N) \circ \overline{\eta}_{F \circ \overline{\Delta(N)}} = \overline{\eta}_{F \circ \overline{\delta}} \circ \text{DIAGR}(F)(\overline{J}_N). \quad (1)$$

**Lemme 2.23**  $\overline{\lambda} : G \circ \overline{\overline{\Delta}} \rightarrow I_{\mathcal{D}}(G(\overline{\delta}))$  est une flèche de  $\text{DIAGR}(\mathcal{D})$ .

*Preuve.* Il suffit de montrer que pour tout arc  $A : N \rightarrow N'$  de  $\Delta^\Phi$ , on a

$$\lambda_{N'} \circ G(\overline{\Delta(A)}) = \lambda_N.$$

$$\begin{aligned} \overline{J_{N'}} \circ \overline{\Delta(A)} &\approx \overline{J_N} && \text{(lemme 2.17)} \\ \Rightarrow \text{DIAGR}(F)(\overline{J_{N'}} \circ \overline{\Delta(A)}) &\approx \text{DIAGR}(F)(\overline{J_N}) && \text{(lemme 2.12)} \\ \Rightarrow \text{Colim}_{\mathcal{D}}(\text{DIAGR}(F)(\overline{J_{N'}} \circ \overline{\Delta(A)})) & && \\ &= \text{Colim}_{\mathcal{D}}(\text{DIAGR}(F)(\overline{J_N})) && \text{(proposition 2.2)} \\ \Rightarrow G(\overline{J_{N'}} \circ \overline{\Delta(A)}) &= G(\overline{J_N}) && \text{(définition de } G\text{)} \\ \Rightarrow G(\overline{J_{N'}}) \circ G(\overline{\Delta(A)}) &= G(\overline{J_N}) && \text{(} G \text{ foncteur)} \\ \Rightarrow \lambda_{N'} \circ G(\overline{\Delta(A)}) &= \lambda_N && \text{(définition de } \lambda_N \text{ et } \lambda_{N'}) \end{aligned}$$

□

**Lemme 2.24** ( $G(\overline{\delta})$ ,  $\overline{\lambda} : G \circ \overline{\Delta} \rightarrow I_{\mathcal{D}}(G(\overline{\delta}))$ ) est un cône colimite du diagramme  $G \circ \overline{\Delta}$  de  $\text{DIAGR}(\mathcal{D})$ .

*Preuve.* Soit un cône  $(A, \overline{\mu} : G \circ \overline{\Delta} \rightarrow I_{\mathcal{D}}(A))$  sur le diagramme  $G \circ \overline{\Delta}$ .  $\overline{\mu}$  est donc une flèche de  $\text{DIAGR}(\mathcal{D})$ . Nous allons montrer qu'il existe une unique flèche

$$\phi : G(\overline{\delta}) \rightarrow A$$

de  $\mathcal{D}$  telle que

$$I_{\mathcal{D}}(\phi) \circ \overline{\lambda} = \overline{\mu}.$$

*Existence.* Pour tout nœud  $N$  de  $\Delta^\Phi$  et tout nœud  $n$  de  $\Delta(N)^\Phi$ , on pose

$$\psi_{N,n} = \mu_N \circ (\eta_{F \circ \Delta(N)})_n \circ F((J_N)_n^{-1}) : F(\delta(N, n)) \rightarrow A.$$

Cette définition est correcte car

$$\begin{aligned} F((J_N)_n^{-1}) &: F(\delta(N, n)) \rightarrow F(\Delta(N)(n)) \\ (\eta_{F \circ \Delta(N)})_n &: F(\Delta(N)(n)) \rightarrow G(\overline{\Delta(N)}) \\ \mu_N &: G(\overline{\Delta(N)}) \rightarrow A. \end{aligned}$$

On montre que

$$\overline{\psi} : F \circ \overline{\delta} \rightarrow I_{\mathcal{D}}(A)$$

est une flèche de  $\text{DIAGR}(\mathcal{D})$ .

- Pour tout arc  $(N, a) : (N, m) \rightarrow (N, n)$  de  $\delta^\Phi$ ,

$$\begin{aligned} \psi_{N,n} \circ F(\delta(N, a)) & \\ &= \mu_N \circ (\eta_{F \circ \Delta(N)})_n \circ F((J_N)_n^{-1}) \circ F(\delta(N, a)) && \text{(définition de } \psi_{N,n}\text{)} \\ &= \mu_N \circ (\eta_{F \circ \Delta(N)})_n \circ F((J_N)_n^{-1} \circ \delta(N, a)) && \text{(} F \text{ foncteur)} \\ &= \mu_N \circ (\eta_{F \circ \Delta(N)})_n \circ F(\Delta(N)(a) \circ (J_N)_m^{-1}) && \text{(définition de } \delta(N, a)\text{)} \\ &= \mu_N \circ (\eta_{F \circ \Delta(N)})_n \circ F(\Delta(N)(a)) \circ F((J_N)_m^{-1}) && \text{(} F \text{ foncteur)} \\ &= \mu_N \circ (\eta_{F \circ \Delta(N)})_m \circ F((J_N)_m^{-1}) && \text{(\overline{\eta}_{F \circ \Delta(N)} \text{ cône)} \\ &= \psi_{N,m} && \text{(définition de } \psi_{N,m}\text{)} \end{aligned}$$

- Pour tout arc  $(A, m) : (M, m) \rightarrow (N, n)$  de  $\delta^\Phi$ , où  $A : M \rightarrow N \in \text{Arc}(\Delta^\Phi)$ ,  $m \in \text{Nœud}(\Delta(M)^\Phi)$ , et  $n = \Delta(A)^\Phi(m) \in \text{Nœud}(\Delta(N)^\Phi)$ ,

$$\begin{aligned}
& \psi_{N,n} \circ F(\delta(A, m)) \\
&= \mu_N \circ (\eta_{F \circ \Delta(N)})_n \circ F((J_N)_n^{-1}) \circ F(\delta(A, m)) \quad (\text{définition de } \psi_{N,n}) \\
&= \mu_N \circ (\eta_{F \circ \Delta(N)})_n \circ F((J_N)_n^{-1} \circ \delta(A, m)) \quad (F \text{ foncteur}) \\
&= \mu_N \circ (\eta_{F \circ \Delta(N)})_n \circ F(\Delta(A)_m) \circ F((J_M)_m^{-1}) \quad (\text{définition de } \delta(A, m)) \\
&= \mu_N \circ G(\overline{\Delta(A)}) \circ (\eta_{F \circ \Delta(M)})_m \circ F((J_M)_m^{-1}) \quad (\text{définition de } G) \\
&= \mu_M \circ (\eta_{F \circ \Delta(M)})_m \circ F((J_M)_m^{-1}) \quad (\bar{\mu} \text{ cône}) \\
&= \psi_{M,m} \quad (\text{définition de } \psi_{M,m})
\end{aligned}$$

On a donc bien une flèche

$$\bar{\psi} : F \circ \bar{\delta} \rightarrow I_{\mathcal{D}}(A)$$

dans  $\mathcal{D}$ . De plus on vérifie immédiatement que

$$\bar{\psi} \circ \text{DIAGR}(F)(\overline{J_N}) = I_{\mathcal{D}}(\mu_N) \circ \bar{\eta}_{F \circ \Delta(N)}. \quad (2)$$

Comme on a un cône

$$(A, \bar{\psi} : F \circ \bar{\delta} \rightarrow I_{\mathcal{D}}(A)),$$

il existe une unique flèche  $\phi : G(\bar{\delta}) \rightarrow A$  de  $\mathcal{D}$  telle que

$$I_{\mathcal{D}}(\phi) \circ \bar{\eta}_{F \circ \bar{\delta}} = \bar{\psi}. \quad (3)$$

Il reste à montrer que

$$\begin{aligned}
& I_{\mathcal{D}}(\phi) \circ \bar{\lambda} = \bar{\mu} \\
& \Leftrightarrow \forall N \in \text{Nœud}(\Delta^\Phi), \phi \circ \lambda_N = \mu_N : G(\overline{\Delta(N)}) \rightarrow A
\end{aligned}$$

Comme

$$(G(\overline{\Delta(N)}), \bar{\eta}_{F \circ \Delta(N)} : F \circ \overline{\Delta(N)} \rightarrow I_{\mathcal{D}}(G(\overline{\Delta(N)})))$$

est cône colimite de  $\overline{\Delta(N)}$ , il suffit de montrer que

$$I_{\mathcal{D}}(\phi \circ \lambda_N) \circ \bar{\eta}_{F \circ \Delta(N)} = I_{\mathcal{D}}(\mu_N) \circ \bar{\eta}_{F \circ \Delta(N)}.$$

$$\begin{aligned}
& I_{\mathcal{D}}(\phi \circ \lambda_N) \circ \bar{\eta}_{F \circ \Delta(N)} \\
&= I_{\mathcal{D}}(\phi) \circ I_{\mathcal{D}}(\lambda_N) \circ \bar{\eta}_{F \circ \Delta(N)} \quad (I_{\mathcal{D}} \text{ foncteur}) \\
&= I_{\mathcal{D}}(\phi) \circ \bar{\eta}_{F \circ \bar{\delta}} \circ \text{DIAGR}(F)(\overline{J_N}) \quad (1) \\
&= \bar{\psi} \circ \text{DIAGR}(F)(\overline{J_N}) \quad (3) \\
&= I_{\mathcal{D}}(\mu_N) \circ \bar{\eta}_{F \circ \Delta(N)} \quad (2)
\end{aligned}$$

*Unicité.* Soit  $\phi' : G(\bar{\delta}) \rightarrow A$  une flèche de  $\mathcal{D}$  telle que

$$I_{\mathcal{D}}(\phi') \circ \bar{\lambda} = \bar{\mu}.$$

On montre que  $\phi = \phi'$ . Pour cela, il suffit de montrer que

$$\begin{aligned} I_{\mathcal{D}}(\phi') \circ \bar{\eta}_{F \circ \bar{\delta}} &= \bar{\psi} \\ \Leftrightarrow \forall (N, n) \in \text{Nœud}(\delta^\Phi), \phi'_{(N, n)} \circ (\eta_{F \circ \bar{\delta}})_{(N, n)} &= \psi_{N, n} \end{aligned}$$

$$\begin{aligned} \phi'_{(N, n)} \circ (\eta_{F \circ \bar{\delta}})_{(N, n)} & \\ &= \phi'_{(N, n)} \circ \lambda_N \circ (\eta_{F \circ \Delta(N)})_n \circ F((J_N)_n^{-1}) \quad (1) \\ &= \mu_N \circ (\eta_{F \circ \Delta(N)})_n \circ F((J_N)_n^{-1}) \quad (\text{définition de } \phi') \\ &= \psi_{N, n} \quad (\text{définition de } \psi_{N, n}) \end{aligned}$$

□

**Proposition 2.3** *Soit une catégorie  $\mathcal{C}$  et une catégorie finiment cocomplète  $\mathcal{D}$ . Soit un foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ . Posons*

$$\begin{aligned} G &= \text{Colim}_{\mathcal{D}} \circ \text{DIAGR}(F) : \text{DIAGR}(\mathcal{C}) \rightarrow \mathcal{D} \\ H &= \text{colim}_{\mathcal{D}} \circ \text{diagr}(F) : \text{diagr}(\mathcal{C}) \rightarrow \mathcal{D}. \end{aligned}$$

Alors,

1.  $H \circ [-]_{\mathcal{C}} = G$  ;
2.  $H \circ [-]_{\mathcal{C}} \circ I_{\mathcal{C}} \cong F$  ;
3.  $H$  conserve les colimites finies.

*Preuve.* On prouve successivement les points 1, 2 et 3.

1. On a

$$\begin{aligned} H \circ [-]_{\mathcal{C}} &= \text{colim}_{\mathcal{D}} \circ \text{diagr}(F) \circ [-]_{\mathcal{C}} \quad (\text{définition de } H) \\ &= \text{colim}_{\mathcal{D}} \circ [-]_{\mathcal{D}} \circ \text{DIAGR}(F) \quad (\text{définition de } \text{diagr}(F)) \\ &= \text{Colim}_{\mathcal{D}} \circ \text{DIAGR}(F) \quad (\text{définition de } \text{colim}_{\mathcal{D}}) \\ &= G \quad (\text{définition de } G) \end{aligned}$$

2. D'après le lemme 2.22,  $G \circ I_{\mathcal{C}} \cong F$ , donc, d'après le point 1,  $H \circ [-]_{\mathcal{C}} \circ I_{\mathcal{C}} \cong F$ .

3. Soit un diagramme  $\bar{\Gamma}$  de  $\text{DIAGR}(\text{diagr}(\mathcal{C}))$ . On considère un diagramme  $\bar{\Delta}$  de  $\text{DIAGR}^2(\mathcal{C})$  tel que

$$\bar{\Gamma} = [-]_{\mathcal{C}} \circ \bar{\Delta}.$$

On pose  $\bar{\delta} = \text{Apl}_{\mathcal{C}}(\bar{\Delta})$ , et le cône colimite du diagramme  $\bar{\Gamma}$

$$(\bar{\delta}, \bar{\eta} : \bar{\Gamma} \rightarrow I_{\text{diagr}(\mathcal{C})}(\bar{\delta}))$$

avec  $\overline{\eta_N} = [\overline{J_N}]$ .

On a

$$H \circ \overline{\Gamma} = H \circ [-]_{\mathcal{C}} \circ \overline{\Delta} = G \circ \overline{\Delta}$$

et

$$\lambda_N = G(\overline{J_N}) = H([\overline{J_N}]) = H(\overline{\eta_N}).$$

D'après le lemme 2.24,

$$(G(\overline{\delta}), \overline{\lambda} : G \circ \overline{\Delta} \rightarrow I_{\mathcal{D}}(G(\overline{\delta})))$$

est un cône colimite de  $G \circ \overline{\Delta}$ , donc

$$(H(\overline{\delta}), H\overline{\eta} : H \circ \overline{\Gamma} \rightarrow I_{\mathcal{D}}(H(\overline{\delta})))$$

est un cône colimite du diagramme  $H \circ \overline{\Gamma}$ . Par conséquent,  $H$  conserve les colimites finies.

□

**Théorème 2.7 (Complétion)** *Soit  $\mathcal{C}$  une catégorie quelconque. La catégorie  $\text{diagr}(\mathcal{C})$  est une complétion de  $\mathcal{C}$  par colimites finies. Autrement dit, pour toute catégorie finiment cocomplète  $\mathcal{D}$ , et tout foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ , il existe un foncteur  $H : \text{diagr}(\mathcal{C}) \rightarrow \mathcal{D}$ , unique à un isomorphisme naturel près, qui conserve les colimites finies, et tel que :*

$$H \circ [-]_{\mathcal{C}} \circ I_{\mathcal{C}} \cong F.$$

*Preuve.* Soit une catégorie  $\mathcal{C}$  quelconque, et une catégorie  $\mathcal{D}$  finiment cocomplète. Soit un foncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ . On montre l'existence, puis l'unicité à un isomorphisme près, d'un foncteur

$$H : \text{diagr}(\mathcal{C}) \rightarrow \mathcal{D}$$

qui conserve les colimites finies et tel que

$$H \circ [-]_{\mathcal{C}} \circ I_{\mathcal{C}} \cong F.$$

*Existence.* On pose  $H = \text{colim}_{\mathcal{D}} \circ \text{diagr}(F)$ . D'après la proposition 2.3,  $H \circ [-]_{\mathcal{C}} \circ I_{\mathcal{C}} \cong F$  et  $H$  conserve les colimites finies.

*Unicité de  $H$  à un isomorphisme naturel près.*

Soit deux foncteurs  $H, H' : \text{diagr}(\mathcal{C}) \rightarrow \mathcal{D}$  qui conservent les colimites finies, et tels que

$$H \circ [-]_{\mathcal{C}} \circ I_{\mathcal{C}} \cong H' \circ [-]_{\mathcal{C}} \circ I_{\mathcal{C}}.$$

On montre qu'il existe un isomorphisme naturel

$$H \cong H'.$$

$$\begin{aligned}
H \circ [-]_{\mathcal{C}} \circ I_{\mathcal{C}} &\cong H' \circ [-]_{\mathcal{C}} \circ I_{\mathcal{C}} \\
\Rightarrow \text{DIAGR}(H \circ [-]_{\mathcal{C}} \circ I_{\mathcal{C}}) &\cong \text{DIAGR}(H' \circ [-]_{\mathcal{C}} \circ I_{\mathcal{C}}) && \text{(lemme 2.6)} \\
\Rightarrow \text{DIAGR}(H) \circ \text{DIAGR}([-]_{\mathcal{C}} \circ I_{\mathcal{C}}) &\cong \\
&\text{DIAGR}(H') \circ \text{DIAGR}([-]_{\mathcal{C}} \circ I_{\mathcal{C}}) && \text{(DIAGR foncteur)} \\
\Rightarrow \text{Colim}_{\mathcal{D}} \circ \text{DIAGR}(H) \circ \text{DIAGR}([-]_{\mathcal{C}} \circ I_{\mathcal{C}}) &\cong \\
&\text{Colim}_{\mathcal{D}} \circ \text{DIAGR}(H') \circ \text{DIAGR}([-]_{\mathcal{C}} \circ I_{\mathcal{C}}) && \text{(propriété 2.3)} \\
\Rightarrow H \circ \text{Colim}_{\text{diagr}(\mathcal{C})} \circ \text{DIAGR}([-]_{\mathcal{C}} \circ I_{\mathcal{C}}) &\cong \\
&H' \circ \text{Colim}_{\text{diagr}(\mathcal{C})} \circ \text{DIAGR}([-]_{\mathcal{C}} \circ I_{\mathcal{C}}) && \text{(propriété 2.6)} \\
\Rightarrow H \circ [-]_{\mathcal{C}} \cong H' \circ [-]_{\mathcal{C}} &&& \text{(lemme 2.21)} \\
\Rightarrow H \cong H' &&& \text{(lemme 2.14)}
\end{aligned}$$

□

**Remarque 2.11** Nous avons montré que la catégorie  $\text{diagr}(\mathcal{C})$  est finiment cocomplète. Par contre, nous n'avons pas montré que  $\text{diagr}(\mathcal{C})$  a des colimites choisies. En effet, nous n'avons pas montré l'existence d'une colimite *canonique* pour un objet de  $\text{DIAGR}(\text{diagr}(\mathcal{C}))$ , puisque pour montrer l'existence d'une colimite, nous devons choisir un représentant pour une classe d'équivalence de flèches de  $\text{DIAGR}(\mathcal{C})$ .

## 2.4 Conclusion

Dans ce chapitre, nous avons défini une catégorie de diagrammes  $\text{DIAGR}(\mathcal{C})$ , dont les objets sont les diagrammes finis, et les flèches sont les morphismes de diagrammes (définitions 2.16 et 2.18). Nous avons rappelé la définition de *colimite* d'un diagramme et de colimite d'un morphisme de diagrammes (paragraphe 2.3.2). Nous avons défini une opération d'*aplatissement*  $\text{Apl}_{\mathcal{C}} : \text{DIAGR}^2(\mathcal{C}) \rightarrow \text{DIAGR}(\mathcal{C})$  qui permet de considérer tout diagramme de diagrammes comme un simple diagramme (définition 2.29).

Nous avons défini une relation de congruence sur les morphismes de diagrammes, ce qui nous a permis de considérer la catégorie quotient  $\text{diagr}(\mathcal{C})$  (définition 2.32). Nous avons montré que la catégorie  $\text{diagr}(\mathcal{C})$  est finiment cocomplète, c'est-à-dire que tout diagramme fini sur  $\text{diagr}(\mathcal{C})$  a une colimite dans  $\text{diagr}(\mathcal{C})$  (théorème 2.6). Cette colimite d'un objet  $\overline{\Gamma}$  de  $\text{DIAGR}(\text{diagr}(\mathcal{C}))$  peut être construite en *aplatissant* un diagramme  $\overline{\Delta}$  de  $\text{DIAGR}^2(\mathcal{C})$  obtenu à partir de  $\overline{\Gamma}$  en remplaçant chaque classe d'équivalence qui étiquète un arc par un représentant de cette classe.

Enfin, nous avons montré que la catégorie  $\text{diagr}(\mathcal{C})$  est une complétion de  $\mathcal{C}$  par colimites finies (théorème 2.7).



# Chapitre 3

## Syntaxe : catégorie des termes

*“Nous sommes certain que la syntaxe est indispensable mais nous voulons soutenir qu’elle ne saurait être syntaxe que dans la mesure où elle envisage non pas des relations entre objets quelconques, mais des relations entre des “signes” ou des “expressions”. Or ce n’est que comme concept pragmatique que nous pourrions aborder la notion de “signe” ou d’“expression”, car un événement n’est un signe que s’il est émis ou reçu comme tel.”*

— Léo Apostel, *Logique et connaissance scientifique*

Dans ce chapitre, nous considérons une catégorie  $\mathcal{C}_0$  de base dont les objets sont considérés comme “atomiques”, c’est-à-dire ne peuvent pas être divisés. Nous supposons que cette catégorie  $\mathcal{C}_0$  est *petite*. Nous souhaitons pouvoir construire des objets composites en réunissant ces objets élémentaires, en fusionnant éventuellement certaines parties communes.

L’assemblage de plusieurs objets est modélisé à l’aide de constructions de colimites. Par exemple, dans la catégorie des ensembles, la somme permet de modéliser l’union disjointe de deux ensembles. La somme amalgamée permet de réunir deux ensembles en fusionnant certains éléments. Dans le cadre des spécifications algébriques, la somme amalgamée permet également de combiner deux spécifications en précisant les parties fusionnées.

Le but de ce chapitre est de présenter une *syntaxe*, c’est-à-dire un langage de description des constructions de colimites. Ces constructions de colimites font partie d’une *catégorie*, parce qu’à partir des objets et flèches de  $\mathcal{C}_0$ , on peut non seulement construire des objets colimites, mais également des flèches entre objets colimites. En effet, d’un point de vue formel, la fonction qui à tout diagramme associe sa colimite peut s’appliquer sur les flèches et être ainsi étendue en un foncteur. Dans ce chapitre, nous construisons donc une catégorie de termes, appelée  $\text{Terme}(\mathcal{C}_0)$ , dont les objets représentent des constructions de colimites.

Notre objectif n’est pas d’avoir une syntaxe pour *toutes* les constructions de coli-

mites finies, mais de sélectionner quelques constructions qui permettent d'obtenir un objet isomorphe à n'importe quelle colimite finie. Il s'agit donc de pouvoir décrire une catégorie finiment cocomplète. Ici, nous choisissons d'avoir une représentation pour un objet initial, et pour des sommes amalgamées. Ce choix est purement arbitraire, on aurait pu prendre d'autres colimites, comme des sommes ou des co-égalisateurs.

Nous avons donc besoin de constructions syntaxiques pour l'objet initial et les sommes amalgamées. Nous utilisons le terme  $\emptyset$  pour représenter l'objet initial, et le terme  $\text{push}(A, B, C, f, g)$  pour représenter la somme amalgamée construite sur le diagramme qui comporte trois nœuds étiquetés par  $A$ ,  $B$ , et  $C$ , et deux flèches étiquetées par  $f : A \rightarrow B$  et  $g : A \rightarrow C$ . Comme nos constructions sont purement syntaxiques, la seule égalité dont nous disposons, au départ, est *l'identité syntaxique*. Nous définissons donc des *relations* entre les flèches pour décrire les propriétés de l'objet initial et des sommes amalgamées. Techniquement, nous définissons donc une *précatégorie*, c'est-à-dire une catégorie dans laquelle certaines égalités sont remplacées par des équivalences sur les flèches. La catégorie correspondante est le quotient de cette précatégorie par la relation d'équivalence.

Comme l'existence de certains termes est conditionnée par une relation entre deux autres termes (cf. règle (10)), on ne peut pas définir successivement les termes, puis la relation sur les termes. Pour résoudre ce problème, nous proposons une construction stratifiée de la syntaxe. Nous définissons une suite de précatégories de termes  $\mathcal{C}_i$ , telles que, dans la précatégorie  $\mathcal{C}_i$ , l'introduction de termes n'est conditionnée que par des relations entre termes de  $\mathcal{C}_{i-1}$ . Finalement, la précatégorie des termes  $\text{TERME}(\mathcal{C}_0)$  est la "limite" de la suite de précatégories  $\mathcal{C}_i$ .

## Syntaxe, sémantique, catégories et choix

En informatique, les problèmes ont souvent deux aspects : un aspect *syntactique*, qui correspond au langage qui décrit les objets manipulés (par exemple un programme) ; et un aspect *sémantique*, qui correspond au sens des objets manipulés (par exemple le résultat de l'exécution d'un programme). Du point de vue syntaxique, on fait des choix de représentation, qui correspondent à la définition du langage utilisé. Du point de vue sémantique, on travaille indépendamment du choix de représentation, "à un codage près".

L'intérêt des catégories réside dans le fait que l'on travaille le plus souvent "à un isomorphisme près", ce qui correspond à travailler "à un codage près" en informatique. Par exemple, en théorie des catégories, on définit les colimites à un isomorphisme près : un diagramme peut avoir plusieurs colimites, qui sont alors isomorphes. Lorsque l'on parle de "la" colimite d'un diagramme, il s'agit en fait souvent d'"une" colimite quelconque d'un diagramme, sachant que tant que l'on travaille à un isomorphisme près, on peut en choisir une quelconque. L'ensemble  $(\{1\} \times A) \cup (\{2\} \times B)$  est un exemple de choix de colimite pour  $A + B$  dans la catégorie des ensembles.

Ce degré de liberté est trop important lorsque l'on parle de syntaxe. On a en effet besoin de faire des choix de représentation ou de codage. Si on veut décrire une syntaxe pour des constructions de colimites, on ne peut plus parler des colimites

à un isomorphisme près. En effet le fait d'avoir une représentation syntaxique par exemple pour une somme impose que l'on ait fait un choix particulier de somme. Cela explique pourquoi dans toute la suite, nous parlons d'“objet initial choisi”, ou de “somme amalgamée choisie”. Techniquement, on doit faire des choix afin d'avoir une construction *libre*. L'introduction de colimites *choisies* dans une catégorie, due à C. Ehresmann [Ehr65, Ehr68], permet de donner un statut algébrique à cette catégorie.

### 3.1 Précatégories, préfoncteurs et pré-colimites

Nous commençons par définir la notion de précatégorie, similaire à celle de catégorie. Dans une précatégorie, il n'y a, *a priori*, pas d'égalité entre flèches, mais seulement des *équivalences*. Chaque ensemble de flèches est donc muni d'une relation de d'équivalence.

**Définition 3.1** (Précatégorie) Une *précatégorie*  $\mathcal{C}$  est définie par

- une classe d'*objets*  $\text{Obj}(\mathcal{C})$ , (si  $A$  est un objet de  $\text{Obj}(\mathcal{C})$ , on notera  $A \in \text{Obj}(\mathcal{C})$ , même si  $\text{Obj}(\mathcal{C})$  n'est pas un ensemble);
- une famille  $\text{Arr}(\mathcal{C})$  indicée par  $\text{Obj}(\mathcal{C}) \times \text{Obj}(\mathcal{C})$  de *flèches* — autrement dit,  $\forall A, B \in \text{Obj}(\mathcal{C})$ , on a un ensemble de flèches  $\text{Arr}(\mathcal{C})(A, B)$ ;
- une famille  $\mathcal{R}(\mathcal{C})$  indicée par  $\text{Obj}(\mathcal{C}) \times \text{Obj}(\mathcal{C})$  de relations sur  $\text{Arr}(\mathcal{C})$  — autrement dit,  $\forall A, B \in \text{Obj}(\mathcal{C})$ , on a une relation  $\mathcal{R}(\mathcal{C})(A, B)$  sur l'ensemble  $\text{Arr}(\mathcal{C})(A, B)$ ;
- $\forall A, B, C \in \text{Obj}(\mathcal{C})$ , on a une opération de composition

$$\circ : \text{Arr}(\mathcal{C})(B, C) \times \text{Arr}(\mathcal{C})(A, B) \rightarrow \text{Arr}(\mathcal{C})(A, C);$$

- $\forall A \in \text{Obj}(\mathcal{C})$ , on a une flèche identité  $id_A \in \text{Arr}(\mathcal{C})(A, A)$ ;
- $\forall A, B \in \text{Obj}(\mathcal{C})$ ,  $\forall f \in \text{Arr}(\mathcal{C})(A, B)$ ,

$$\begin{aligned} (f \circ id_A, f) &\in \mathcal{R}(\mathcal{C})(A, B) \\ (id_B \circ f, f) &\in \mathcal{R}(\mathcal{C})(A, B); \end{aligned}$$

- $\forall A, B, C, D \in \text{Obj}(\mathcal{C})$ ,  
 $\forall f \in \text{Arr}(\mathcal{C})(A, B)$ ,  $\forall g \in \text{Arr}(\mathcal{C})(B, C)$ ,  $\forall h \in \text{Arr}(\mathcal{C})(C, D)$ ,

$$((h \circ g) \circ f, h \circ (g \circ f)) \in \mathcal{R}(\mathcal{C})(A, D);$$

- $\mathcal{R}(\mathcal{C})$  est une congruence, c'est-à-dire

$$- \forall A, B \in \text{Obj}(\mathcal{C}), \mathcal{R}(\mathcal{C})(A, B) \text{ est une relation d'équivalence};$$

$$\begin{aligned}
& - \forall A, B, C \in \text{Obj}(\mathcal{C}), \forall f, f' \in \text{Arr}(\mathcal{C})(A, B), \forall g, g' \in \text{Arr}(\mathcal{C})(B, C), \\
& \left. \begin{array}{l} (f, f') \in \mathcal{R}(\mathcal{C})(A, B) \\ (g, g') \in \mathcal{R}(\mathcal{C})(B, C) \end{array} \right\} \Rightarrow (g \circ f, g' \circ f') \in \mathcal{R}(\mathcal{C})(A, C).
\end{aligned}$$

**Lemme 3.1** *On peut associer à toute précatégorie  $\mathcal{C}$  une catégorie, qu'on notera également, par abus de langage,  $\mathcal{C}$ . Cette catégorie est définie de la façon suivante.*

- Les objets de la catégorie  $\mathcal{C}$  sont les objets de la précatégorie  $\mathcal{C}$ .
- Comme  $\mathcal{R}(\mathcal{C})(A, B)$  est une relation d'équivalence sur  $\text{Arr}(\mathcal{C})(A, B)$ , on peut définir l'ensemble des flèches de  $A$  vers  $B$  de la catégorie  $\mathcal{C}$  comme l'ensemble quotient

$$\text{hom}_{\mathcal{C}}(A, B) = \text{Arr}(\mathcal{C})(A, B) / \mathcal{R}(\mathcal{C})(A, B).$$

On notera  $[f] : A \rightarrow B$  la classe d'équivalence de  $f \in \text{Arr}(\mathcal{C})(A, B)$ .

- Étant donné un objet  $A$  de  $\mathcal{C}$ , l'identité sur  $A$  est  $[id_A] : A \rightarrow A$ .
- La composition de deux flèches  $[f] : A \rightarrow B$  et  $[g] : B \rightarrow C$  est la classe  $[g \circ f] : A \rightarrow C$ . Cette définition est indépendante des représentants  $f$  et  $g$  choisis car  $\mathcal{R}(\mathcal{C})$  est une congruence.

*Preuve.* On vérifie immédiatement que

- les flèches  $[id_A] : A \rightarrow A$  sont bien des identités ;
- la composition est associative.

□

Réciproquement, on peut considérer toute catégorie comme une précatégorie, en prenant pour relation d'équivalence la relation d'égalité sur les flèches.

**Définition 3.2** (Identité syntaxique) Soit une précatégorie  $\mathcal{C}$ . L'égalité dans  $\mathcal{C}$  (à ne pas confondre avec l'équivalence  $\mathcal{R}(\mathcal{C})$ ) est appelée *identité syntaxique*.

En effet, dans une précatégorie de termes, deux termes sont égaux lorsqu'ils sont *syntactiquement identiques*, c'est-à-dire composés des mêmes symboles.

### Notations

- L'identité syntaxique sera notée  $\equiv$ .
- Pour plus de lisibilité, et comme deux flèches équivalentes dans la précatégorie  $\mathcal{C}$  sont *égales* dans la catégorie  $\mathcal{C}$  associée,  $(f, f') \in \mathcal{R}(\mathcal{C})(A, B)$  sera noté

$$f = f' \in \text{Arr}(\mathcal{C})(A, B).$$

Nous rappelons le plus souvent possible l'ensemble auquel les flèches appartiennent, parce que nous manipulons plusieurs précatégories, et certaines flèches appartiennent à plusieurs ensembles de flèches dans des précatégories différentes.

**Définition 3.3** (Petite précatégorie) Une précatégorie  $\mathcal{C}$  est *petite* si et seulement si la catégorie associée est petite, c'est-à-dire si et seulement si  $\text{Obj}(\mathcal{C})$  est un ensemble.

L'analogue d'un foncteur entre deux catégories est un *préfoncteur* entre deux précatégories.

**Définition 3.4** (Préfoncteur) Soit deux précatégories  $\mathcal{C}$  et  $\mathcal{D}$ . Un *préfoncteur*  $F$  de  $\mathcal{C}$  vers  $\mathcal{D}$ , noté  $F : \mathcal{C} \rightarrow \mathcal{D}$  est une application qui à tout objet  $A$  de  $\mathcal{C}$  associe un objet  $F(A)$  de  $\mathcal{D}$ , et à toute flèche  $f \in \text{Arr}(\mathcal{C})(A, B)$  associe une flèche  $F(f) \in \text{Arr}(\mathcal{D})(F(A), F(B))$ , et telle que

$$\bullet \forall f, f' \in \text{Arr}(\mathcal{C})(A, B),$$

$$f = f' \in \text{Arr}(\mathcal{C})(A, B) \Rightarrow F(f) = F(f') \in \text{Arr}(\mathcal{D})(F(A), F(B));$$

$$\bullet \forall A \in \text{Obj}(\mathcal{C}), F(\text{id}_A) = \text{id}_{F(A)} \in \text{Arr}(\mathcal{D})(F(A), F(A));$$

$$\bullet \forall f \in \text{Arr}(\mathcal{C})(A, B) \text{ et } \forall g \in \text{Arr}(\mathcal{C})(B, C),$$

$$F(g \circ f) = F(g) \circ F(f) \in \text{Arr}(\mathcal{D})(F(A), F(C)).$$

À tout préfoncteur entre deux précatégories on peut associer un *foncteur* entre les catégories correspondantes.

**Lemme 3.2** Soit deux précatégories  $\mathcal{C}$  et  $\mathcal{D}$  et un préfoncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ . Alors on a un foncteur, noté également  $F : \mathcal{C} \rightarrow \mathcal{D}$  entre les catégories  $\mathcal{C}$  et  $\mathcal{D}$ , défini par

$$\begin{array}{ccc} F : & \mathcal{C} & \rightarrow & \mathcal{D} \\ & A & \mapsto & A \\ & [f] & \mapsto & [F(f)] \end{array}$$

La définition de  $F([f])$  est bien indépendante du représentant  $f$  choisi car

$$f = f' \in \text{Arr}(\mathcal{C})(A, B) \Rightarrow F(f) = F(f') \in \text{Arr}(\mathcal{D})(F(A), F(B)).$$

Deux préfoncteurs  $F, G : \mathcal{C} \rightarrow \mathcal{D}$  sont *syntactiquement identiques* lorsque

$$\bullet \forall A \in \text{Obj}(\mathcal{C}), F(A) \equiv G(A);$$

$$\bullet \forall f \in \text{Arr}(\mathcal{C})(A, B), F(f) \equiv G(f).$$

On note  $F \equiv G$ .

On peut reformuler de nombreuses notions de théorie des catégories en les adaptant aux précatégories. Il suffit de remplacer les égalités entre flèches dans chaque catégorie par des équivalences dans la précatégorie correspondante.

**Définition 3.5** (Isomorphisme) Soit une précatégorie  $\mathcal{C}$ . La flèche  $f \in \text{Arr}(\mathcal{C})(A, B)$  est un *isomorphisme* (dans la précatégorie  $\mathcal{C}$ ) si et seulement si il existe une flèche  $g \in \text{Arr}(\mathcal{C})(B, A)$  telle que  $g \circ f = \text{id}_A \in \text{Arr}(\mathcal{C})(A, A)$  et  $f \circ g = \text{id}_B \in \text{Arr}(\mathcal{C})(B, B)$ .

On vérifie facilement qu'une flèche  $f$  dans une précatégorie  $\mathcal{C}$  est un isomorphisme si et seulement si  $[f]$  est un isomorphisme dans la catégorie associée.

**Définition 3.6** (Préfoncteur plein) Soit deux précatégories  $\mathcal{C}$  et  $\mathcal{D}$ , et un préfoncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ . Le préfoncteur  $F$  est *plein* si et seulement si pour toute flèche  $g \in \text{Arr}(\mathcal{D})(F(A), F(B))$ , il existe une flèche  $f \in \text{Arr}(\mathcal{C})(A, B)$  telle que

$$g = F(f) \in \text{Arr}(\mathcal{D})(F(A), F(B)).$$

**Définition 3.7** (Préfoncteur fidèle) Soit deux précatégories  $\mathcal{C}$  et  $\mathcal{D}$ , et un préfoncteur  $F : \mathcal{C} \rightarrow \mathcal{D}$ . Le préfoncteur  $F$  est *fidèle* si et seulement si pour toutes flèches  $f, f' \in \text{Arr}(\mathcal{C})(A, B)$ ,

$$F(f) = F(f') \in \text{Arr}(\mathcal{D})(F(A), F(B)) \Rightarrow f = f' \in \text{Arr}(\mathcal{C})(A, B).$$

On vérifie facilement qu'un préfoncteur  $F$  est *plein* (respectivement *fidèle*) si et seulement si le foncteur associé est *plein* (respectivement *fidèle*).

On peut enfin reformuler la notion de colimite d'un diagramme pour une précatégorie. Nous définissons ici uniquement les notions d'*objet pré-initial* et de *pré-sommes amalgamées*.

**Définition 3.8** (Objet pré-initial) Soit une précatégorie  $\mathcal{C}$ . Un objet  $\emptyset$  de  $\mathcal{C}$  est *pré-initial* si et seulement si

1.  $\forall A \in \text{Obj}(\mathcal{C})$ , il existe une flèche  $j_A \in \text{Arr}(\mathcal{C})(\emptyset, A)$ ;
2.  $\forall f \in \text{Arr}(\mathcal{C})(\emptyset, A)$ ,  $f = j_A \in \text{Arr}(\mathcal{C})(\emptyset, A)$ .

**Lemme 3.3** *Un objet  $\emptyset$  est pré-initial dans une précatégorie  $\mathcal{C}$  si et seulement si  $\emptyset$  est initial dans la catégorie associée  $\mathcal{C}$ .*

**Définition 3.9** (Pré-somme amalgamée) Soit une précatégorie  $\mathcal{C}$ . Soit trois objets  $A, B$  et  $C$  de  $\mathcal{C}$ , ainsi que deux flèches  $f \in \text{Arr}(\mathcal{C})(A, B)$  et  $g \in \text{Arr}(\mathcal{C})(A, C)$ . Le triplet  $(P, \&_1, \&_2)$ , où  $P \in \text{Obj}(\mathcal{C})$ ,  $\&_1 \in \text{Arr}(\mathcal{C})(B, P)$  et  $\&_2 \in \text{Arr}(\mathcal{C})(C, P)$ , est une *pré-somme amalgamée* de  $B$  et  $C$  par rapport aux flèches  $f$  et  $g$  si et seulement si

1.  $\&_1 \circ f = \&_2 \circ g \in \text{Arr}(\mathcal{C})(A, P)$ ;
2. si  $D \in \text{Obj}(\mathcal{C})$ ,  $f' \in \text{Arr}(\mathcal{C})(B, D)$ ,  $g' \in \text{Arr}(\mathcal{C})(C, D)$  et  $f' \circ f = g' \circ g \in \text{Arr}(\mathcal{C})(A, D)$ , alors,

(a) il existe une flèche  $u \in \text{Arr}(\mathcal{C})(P, D)$  telle que

$$\begin{aligned} u \circ \&_1 &= f' \in \text{Arr}(\mathcal{C})(B, D) \\ u \circ \&_2 &= g' \in \text{Arr}(\mathcal{C})(C, D); \end{aligned}$$

(b) pour toute flèche  $v \in \text{Arr}(\mathcal{C})(P, D)$  telle que

$$\begin{aligned} v \circ \&_1 &= f' \in \text{Arr}(\mathcal{C})(B, D) \\ v \circ \&_2 &= g' \in \text{Arr}(\mathcal{C})(C, D), \end{aligned}$$

on a  $u = v \in \text{Arr}(\mathcal{C})(P, D)$ .

**Lemme 3.4** *Le triplet  $(P, \&_1, \&_2)$  est une pré-somme amalgamée de  $B$  et  $C$  par rapport aux flèches  $f$  et  $g$  dans la pré-catégorie  $\mathcal{C}$  si et seulement si le triplet  $(P, [\&_1], [\&_2])$  est une somme amalgamée de  $B$  et  $C$  par rapport aux flèches  $[f]$  et  $[g]$  dans la catégorie associée  $\mathcal{C}$ .*

On peut définir, de façon similaire, la *pré-colimite* d'un diagramme sur une pré-catégorie. Une pré-catégorie est *finiment pré-cocomplète* lorsque tout diagramme fini a une pré-colimite.

De façon générale, une pré-colimite dans une pré-catégorie correspond à une colimite dans la catégorie associée. Par conséquent, une pré-catégorie est finiment pré-cocomplète si et seulement si la catégorie associée est finiment cocomplète. De plus, une pré-catégorie est finiment pré-cocomplète si et seulement si elle a un objet pré-initial et des pré-sommes amalgamées.

### Pré-catégorie $\text{DIAGR}(\mathcal{C})$

Étant donné une catégorie  $\mathcal{C}$ , la catégorie  $\text{DIAGR}(\mathcal{C})$ , avec la relation de congruence  $\approx$ , est une pré-catégorie. Plus précisément,

- l'identité syntaxique dans la pré-catégorie  $\text{DIAGR}(\mathcal{C})$  est l'égalité dans la catégorie  $\text{DIAGR}(\mathcal{C})$  :

– si  $\bar{\alpha}$  et  $\bar{\beta}$  sont deux objets de  $\text{DIAGR}(\mathcal{C})$ ,

$$\bar{\alpha} \equiv \bar{\beta} \text{ (dans la pré-catégorie)} \Leftrightarrow \bar{\alpha} = \bar{\beta} \text{ (dans la catégorie)} ;$$

– si  $\bar{\sigma}, \bar{\tau} : \bar{\alpha} \rightarrow \bar{\beta}$  sont deux flèches de  $\text{DIAGR}(\mathcal{C}_0)$ ,

$$\bar{\sigma} \equiv \bar{\tau} \text{ (dans la pré-catégorie)} \Leftrightarrow \bar{\sigma} = \bar{\tau} \text{ (dans la catégorie)} ;$$

- la relation d'équivalence dans la pré-catégorie  $\text{DIAGR}(\mathcal{C})$  est la congruence  $\approx$  sur les flèches de la catégorie  $\text{DIAGR}(\mathcal{C})$  : si  $\bar{\sigma}, \bar{\tau} : \bar{\alpha} \rightarrow \bar{\beta}$  sont deux flèches de  $\text{DIAGR}(\mathcal{C}_0)$ ,

$$\bar{\sigma} = \bar{\tau} \in \text{Arr}(\text{DIAGR}(\mathcal{C}))(\bar{\alpha}, \bar{\beta}) \Leftrightarrow \bar{\sigma} \approx \bar{\tau}.$$

La catégorie associée à la pré-catégorie  $\text{DIAGR}(\mathcal{C})$  est  $\text{diagr}(\mathcal{C})$ . De plus, le foncteur  $I_{\mathcal{C}} : \mathcal{C} \rightarrow \text{DIAGR}(\mathcal{C})$  peut être considéré comme un préfoncteur dont le foncteur associé est  $[-]_{\mathcal{C}} \circ I_{\mathcal{C}} : \mathcal{C} \rightarrow \text{diagr}(\mathcal{C})$ .

D'autre part, la pré-catégorie  $\text{DIAGR}(\mathcal{C})$  est finiment pré-cocomplète, d'après les lemmes 2.17 et 2.19. Remarquons que ce sont justement ces deux lemmes qui nous ont permis de montrer que la catégorie  $\text{diagr}(\mathcal{C})$  est finiment cocomplète. L'aplatissement

$$\text{Apl}_{\mathcal{C}} : \text{DIAGR}^2(\mathcal{C}) \rightarrow \text{DIAGR}(\mathcal{C})$$

donne un choix de pré-colimites dans  $\text{DIAGR}(\mathcal{C})$ . En particulier, la pré-catégorie  $\text{DIAGR}(\mathcal{C})$  a un objet pré-initial choisi (le diagramme vide) et des pré-sommes amalgamées choisies (obtenues par aplatissement).

### 3.2 Problème de circularité termes – congruences

Soit  $\mathcal{C}_0$  une petite catégorie. Nous cherchons à définir une catégorie qui contient  $\mathcal{C}_0$ , qui a un objet initial et des sommes amalgamées. Nous allons en fait définir une pré-catégorie avec un objet pré-initial et des pré-sommes amalgamées.

#### Objet pré-initial

Pour l'objet pré-initial, il nous suffit d'introduire un nouvel objet  $\emptyset$ , et pour tout objet  $A$ , une flèche  $j_A$  de  $\emptyset$  vers  $A$ . De plus, pour chaque couple de flèches  $(f, g)$  de  $\emptyset$  vers  $A$ , on introduit la relation  $f = g$ .

#### Pré-sommes amalgamées

Étant donné trois objets  $A, B$  et  $C$ , ainsi que deux flèches  $f : A \rightarrow B$  et  $g : A \rightarrow C$ , nous introduisons un nouvel objet

$$\text{push}(A, B, C, f, g)$$

deux flèches

$$\begin{aligned} \&_1(A, B, C, f, g) : B \rightarrow \text{push}(A, B, C, f, g) \\ \&_2(A, B, C, f, g) : C \rightarrow \text{push}(A, B, C, f, g) \end{aligned}$$

et la relation

$$\&_1(A, B, C, f, g) \circ f = \&_2(A, B, C, f, g) \circ g.$$

Il y a donc une circularité entre la définition des objets et la définition des flèches, puisque l'introduction d'une nouvelle flèche permet d'introduire ensuite des nouveaux objets. Cette circularité ne pose, *a priori*, pas de problème car il s'agit d'une circularité au niveau des *termes*.

De plus, étant donné un objet  $D$  et deux flèches  $f' : B \rightarrow D$  et  $g' : C \rightarrow D$  telles que

$$f' \circ f = g' \circ g$$

on introduit une flèche

$$\text{up}(A, B, C, D, f, g, f', g') : \text{push}(A, B, C, f, g) \rightarrow D$$

et deux relations

$$\begin{aligned} \text{up}(A, B, C, D, f, g, f', g') \circ \&_1(A, B, C, f, g) &= f' \\ \text{up}(A, B, C, D, f, g, f', g') \circ \&_2(A, B, C, f, g) &= g'. \end{aligned}$$

Il faut également spécifier qu'il y a une unique flèche qui satisfait ces deux relations. L'introduction de la flèche

$$\text{up}(A, B, C, D, f, g, f', g') : \text{push}(A, B, C, f, g) \rightarrow D$$

pose un problème. Jusqu'à présent, nous avons défini d'abord des termes, puis des relations entre les termes. Ici, nous devons introduire un nouveau terme à *condition*

qu'une relation soit vérifiée. Il y a donc une circularité entre la définition des termes et la définition des relations.

Les théories algébriques généralisées, qui sont une généralisation des algèbres multi-sortes, ont été proposées par Cartmell pour spécifier des types *dépendants*, c'est-à-dire paramétrés par des termes [Car86]. Par exemple le “type”  $\text{Arr}(A, B)$  dépend de deux termes  $A$  et  $B$ . Cependant, les types dépendants de Cartmell ne permettent pas de spécifier des termes dont l'existence dépend d'une équivalence entre deux autres termes. T. Streicher et M. Wirsing, qui préconisent l'utilisation de types dépendants [SW91], ne précisent pas comment résoudre ce problème.

Une solution est de remplacer la flèche  $\text{up}$  qui pose problème par des flèches dont l'existence ne dépend pas d'une équivalence entre deux termes. Cette approche a été proposée par F. Cury [Cur91]. Les flèches  $\text{up}$  sont remplacées par deux flèches  $\text{p}$  et  $\text{d}$ , qui sont des flèches  $\text{up}$  particulières dont l'existence n'est pas conditionnée par une équivalence. Toute flèche  $\text{up}$  peut être reconstruite *a posteriori* à l'aide des flèches  $\text{p}$  et  $\text{d}$ .

Dans notre travail, nous voulons d'une part rester proche de la définition classique de la somme amalgamée, et d'autre part ne pas multiplier les flèches et les règles à considérer. Pour ces raisons, nous conservons les flèches  $\text{up}$  et proposons une construction stratifiée de la syntaxe, qui permet d'éliminer le problème de circularité. Nous définissons une suite de précatégories  $\mathcal{C}_i$  telles que, dans la précatégorie  $\mathcal{C}_{i+1}$ , l'introduction d'une flèche  $\text{up}$  dépend uniquement d'une relation dans  $\mathcal{C}_i$ . Étant donné des objets  $A, B, C, D$ , et des flèches  $f : A \rightarrow B, g : A \rightarrow C, f' : B \rightarrow D$  et  $g' : C \rightarrow D$  de  $\mathcal{C}_i$ , tels qu'on ait la relation  $f \circ f' = g \circ g'$  dans  $\mathcal{C}_i$ , on introduit une flèche

$$\text{up}(A, B, C, D, f, g, f', g') : \text{push}(A, B, C, f, g) \rightarrow D$$

dans  $\mathcal{C}_{i+1}$ . Dans chaque  $\mathcal{C}_i$ , nous définissons donc d'abord les objets — l'ensemble  $\text{Obj}(\mathcal{C}_i)$  —, ensuite les flèches — la famille d'ensembles  $\text{Arr}(\mathcal{C}_i)$  —, et enfin les équivalences entre flèches — la famille de relations  $\mathcal{R}(\mathcal{C}_i)$ .

### 3.3 Précatégories $\mathcal{C}_i$

Soit  $\mathcal{C}_0$  une petite catégorie. On considère la précatégorie associée  $\mathcal{C}_0$ . On a donc un ensemble d'objets  $\text{Obj}(\mathcal{C}_0)$ , une famille de flèches  $\text{Arr}(\mathcal{C}_0)$  et une famille de relations  $\mathcal{R}(\mathcal{C}_0)$ . Si  $A$  et  $B$  sont des objets de  $\mathcal{C}_0$ , la relation  $\mathcal{R}(\mathcal{C}_0)(A, B)$  sur l'ensemble  $\text{Arr}(\mathcal{C}_0)(A, B)$  est l'égalité dans la catégorie  $\mathcal{C}_0$ .

Dans ce paragraphe, nous définissons, pour tout entier naturel  $i$ , une petite précatégorie  $\mathcal{C}_{i+1}$ , en définissant successivement les ensembles et familles d'ensembles  $\text{Obj}(\mathcal{C}_{i+1})$ ,  $\text{Arr}(\mathcal{C}_{i+1})$  et  $\mathcal{R}(\mathcal{C}_{i+1})$ , à partir de  $\text{Obj}(\mathcal{C}_i)$ ,  $\text{Arr}(\mathcal{C}_i)$  et  $\mathcal{R}(\mathcal{C}_i)$ . Chaque ensemble est caractérisé par des règles de la forme

$$\frac{x \in X \ ; \ y \in Y}{z(x, y) \in Z}$$

Cette règle définit partiellement l'ensemble  $Z$ , et signifie :

“Si le terme  $x$  appartient à  $X$ , et si le terme  $y$  appartient à  $Y$ , alors le terme  $z(x, y)$  appartient à  $Z$ .”

Ensuite, on considère le plus petit ensemble  $Z$  qui satisfait toutes les règles qui le définissent.

### Règles qui définissent l'ensemble $\text{Obj}(\mathcal{C}_{i+1})$

Les objets de  $\mathcal{C}_{i+1}$  sont les objets de  $\mathcal{C}_0$  (règle (1)), et les objets que l'on peut construire à partir d'objets et de flèches de  $\mathcal{C}_i$ , c'est-à-dire objet pré-initial (règle (2)) et pré-somme amalgamée (règle (3)).

$$\frac{A \in \text{Obj}(\mathcal{C}_0)}{A \in \text{Obj}(\mathcal{C}_{i+1})} \quad (1)$$

$$\frac{}{\emptyset \in \text{Obj}(\mathcal{C}_{i+1})} \quad (2)$$

$$\frac{A, B, C \in \text{Obj}(\mathcal{C}_i) ; f \in \text{Arr}(\mathcal{C}_i)(A, B) ; g \in \text{Arr}(\mathcal{C}_i)(A, C)}{\text{push}(A, B, C, f, g) \in \text{Obj}(\mathcal{C}_{i+1})} \quad (3)$$

### Règles qui définissent la famille d'ensembles $\text{Arr}(\mathcal{C}_{i+1})$

Les flèches de  $\mathcal{C}_{i+1}$  sont les flèches de  $\mathcal{C}_0$  (règle (4)), les compositions de flèches de  $\mathcal{C}_{i+1}$  (règle (5)), les identités entre objets de  $\mathcal{C}_{i+1}$  (règle (6)), les flèches de cône colimite (règles (8), (9)), et les uniques flèches d'un objet colimite vers le sommet d'un cône (règles (7), (10)).

$$\frac{A, B \in \text{Obj}(\mathcal{C}_0) ; f \in \text{Arr}(\mathcal{C}_0)(A, B)}{f \in \text{Arr}(\mathcal{C}_{i+1})(A, B)} \quad (4)$$

$$\frac{A, B, C \in \text{Obj}(\mathcal{C}_{i+1}) ; f \in \text{Arr}(\mathcal{C}_{i+1})(A, B) ; g \in \text{Arr}(\mathcal{C}_{i+1})(B, C)}{g \circ f \in \text{Arr}(\mathcal{C}_{i+1})(A, C)} \quad (5)$$

$$\frac{A \in \text{Obj}(\mathcal{C}_{i+1})}{\text{id}_A \in \text{Arr}(\mathcal{C}_{i+1})(A, A)} \quad (6)$$

$$\frac{A \in \text{Obj}(\mathcal{C}_{i+1})}{\text{j}_A \in \text{Arr}(\mathcal{C}_{i+1})(\emptyset, A)} \quad (7)$$

$$\frac{A, B, C \in \text{Obj}(\mathcal{C}_i) ; f \in \text{Arr}(\mathcal{C}_i)(A, B) ; g \in \text{Arr}(\mathcal{C}_i)(A, C)}{\&_1(A, B, C, f, g) \in \text{Arr}(\mathcal{C}_{i+1})(B, \text{push}(A, B, C, f, g))} \quad (8)$$

$$\frac{A, B, C \in \text{Obj}(\mathcal{C}_i) ; f \in \text{Arr}(\mathcal{C}_i)(A, B) ; g \in \text{Arr}(\mathcal{C}_i)(A, C)}{\&_2(A, B, C, f, g) \in \text{Arr}(\mathcal{C}_{i+1})(C, \text{push}(A, B, C, f, g))} \quad (9)$$

$$\frac{A, B, C, D \in \text{Obj}(\mathcal{C}_i) ; f \in \text{Arr}(\mathcal{C}_i)(A, B) ; g \in \text{Arr}(\mathcal{C}_i)(A, C) ; f' \in \text{Arr}(\mathcal{C}_i)(B, D) ; g' \in \text{Arr}(\mathcal{C}_i)(C, D) ; f' \circ f = g' \circ g \in \text{Arr}(\mathcal{C}_i)(A, D)}{\text{up}(A, B, C, D, f, g, f', g') \in \text{Arr}(\mathcal{C}_{i+1})(\text{push}(A, B, C, f, g), D)} \quad (10)$$

Remarquons que l'existence de la flèche  $\text{up}$  de  $\mathcal{C}_{i+1}$  dépend d'une relation entre deux flèches de  $\mathcal{C}_i$ . L'introduction d'un terme dans une pré-catégorie ne dépend jamais d'une relation dans cette pré-catégorie.

**Règles qui définissent la famille de relations  $\mathcal{R}(\mathcal{C}_{i+1})$** 

La relation  $\mathcal{R}(\mathcal{C}_{i+1})$  contient la relation  $\mathcal{R}(\mathcal{C}_0)$  (règle (11)).  $\mathcal{R}(\mathcal{C}_{i+1})$  est une relation d'équivalence (règles (12)–(14)), et une congruence (règle (15)). Pour que  $\mathcal{C}_{i+1}$  soit une précatégorie, il faut que la composition soit associative (règle (16)), et que les flèches  $\text{id}_A$  soient des flèches identités (règles (17), (18)). Les règles (19)–(23) servent à assurer que  $\emptyset$  est pré-initial, et que  $\text{push}(A, B, C, f, g)$  est une pré-somme amalgamée.

$$\frac{A, B \in \text{Obj}(\mathcal{C}_0) ; f, g \in \text{Arr}(\mathcal{C}_0)(A, B) \quad f = g \in \text{Arr}(\mathcal{C}_0)(A, B)}{f = g \in \text{Arr}(\mathcal{C}_{i+1})(A, B)} \quad (11)$$

$$\frac{A, B \in \text{Obj}(\mathcal{C}_{i+1}) ; f \in \text{Arr}(\mathcal{C}_{i+1})(A, B)}{f = f \in \text{Arr}(\mathcal{C}_{i+1})(A, B)} \quad (12)$$

$$\frac{A, B \in \text{Obj}(\mathcal{C}_{i+1}) ; f, g \in \text{Arr}(\mathcal{C}_{i+1})(A, B) \quad f = g \in \text{Arr}(\mathcal{C}_{i+1})(A, B)}{g = f \in \text{Arr}(\mathcal{C}_{i+1})(A, B)} \quad (13)$$

$$\frac{A, B \in \text{Obj}(\mathcal{C}_{i+1}) ; f, g, h \in \text{Arr}(\mathcal{C}_{i+1})(A, B) \quad f = g \in \text{Arr}(\mathcal{C}_{i+1})(A, B) \quad g = h \in \text{Arr}(\mathcal{C}_{i+1})(A, B)}{f = h \in \text{Arr}(\mathcal{C}_{i+1})(A, B)} \quad (14)$$

$$\frac{A, B, C \in \text{Obj}(\mathcal{C}_{i+1}) ; f, f' \in \text{Arr}(\mathcal{C}_{i+1})(A, B) ; g, g' \in \text{Arr}(\mathcal{C}_{i+1})(B, C) \quad f = f' \in \text{Arr}(\mathcal{C}_{i+1})(A, B) ; g = g' \in \text{Arr}(\mathcal{C}_{i+1})(B, C)}{g \circ f = g' \circ f' \in \text{Arr}(\mathcal{C}_{i+1})(A, C)} \quad (15)$$

$$\frac{A, B, C, D \in \text{Obj}(\mathcal{C}_{i+1}) \quad f \in \text{Arr}(\mathcal{C}_{i+1})(A, B) ; g \in \text{Arr}(\mathcal{C}_{i+1})(B, C) ; h \in \text{Arr}(\mathcal{C}_{i+1})(C, D)}{(h \circ g) \circ f = h \circ (g \circ f) \in \text{Arr}(\mathcal{C}_{i+1})(A, D)} \quad (16)$$

$$\frac{A, B \in \text{Obj}(\mathcal{C}_{i+1}) ; f \in \text{Arr}(\mathcal{C}_{i+1})(A, B)}{f \circ \text{id}_A = f \in \text{Arr}(\mathcal{C}_{i+1})(A, B)} \quad (17)$$

$$\frac{A, B \in \text{Obj}(\mathcal{C}_{i+1}) ; f \in \text{Arr}(\mathcal{C}_{i+1})(A, B)}{\text{id}_B \circ f = f \in \text{Arr}(\mathcal{C}_{i+1})(A, B)} \quad (18)$$

$$\frac{A \in \text{Obj}(\mathcal{C}_{i+1}) ; f, g \in \text{Arr}(\mathcal{C}_{i+1})(\emptyset, A)}{f = g \in \text{Arr}(\mathcal{C}_{i+1})(\emptyset, A)} \quad (19)$$

$$\frac{A, B, C \in \text{Obj}(\mathcal{C}_i) ; f \in \text{Arr}(\mathcal{C}_i)(A, B) ; g \in \text{Arr}(\mathcal{C}_i)(A, C)}{\&_1(A, B, C, f, g) \circ f = \&_2(A, B, C, f, g) \circ g \in \text{Arr}(\mathcal{C}_{i+1})(A, \text{push}(A, B, C, f, g))} \quad (20)$$

$$\frac{A, B, C, D \in \text{Obj}(\mathcal{C}_i) ; f \in \text{Arr}(\mathcal{C}_i)(A, B) ; g \in \text{Arr}(\mathcal{C}_i)(A, C) \quad f' \in \text{Arr}(\mathcal{C}_i)(B, D) ; g' \in \text{Arr}(\mathcal{C}_i)(C, D) ; f' \circ f = g' \circ g \in \text{Arr}(\mathcal{C}_i)(A, D)}{\text{up}(A, B, C, D, f, g, f', g') \circ \&_1(A, B, C, f, g) = f' \in \text{Arr}(\mathcal{C}_{i+1})(B, D)} \quad (21)$$

$$\frac{A, B, C, D \in \text{Obj}(\mathcal{C}_i) ; f \in \text{Arr}(\mathcal{C}_i)(A, B) ; g \in \text{Arr}(\mathcal{C}_i)(A, C) \\ f' \in \text{Arr}(\mathcal{C}_i)(B, D) ; g' \in \text{Arr}(\mathcal{C}_i)(C, D) ; f' \circ f = g' \circ g \in \text{Arr}(\mathcal{C}_i)(A, D)}{\text{up}(A, B, C, D, f, g, f', g') \circ \&_2(A, B, C, f, g) = g' \in \text{Arr}(\mathcal{C}_{i+1})(C, D)} \quad (22)$$

$$\frac{A, B, C, D \in \text{Obj}(\mathcal{C}_i) ; f \in \text{Arr}(\mathcal{C}_i)(A, B) ; g \in \text{Arr}(\mathcal{C}_i)(A, C) \\ u, v \in \text{Arr}(\mathcal{C}_{i+1})(\text{push}(A, B, C, f, g), D) \\ u \circ \&_1(A, B, C, f, g) = v \circ \&_1(A, B, C, f, g) \in \text{Arr}(\mathcal{C}_{i+1})(B, D) \\ u \circ \&_2(A, B, C, f, g) = v \circ \&_2(A, B, C, f, g) \in \text{Arr}(\mathcal{C}_{i+1})(C, D)}{u = v \in \text{Arr}(\mathcal{C}_{i+1})(\text{push}(A, B, C, f, g), D)} \quad (23)$$

Pour obtenir une suite de précatégories  $\mathcal{C}_i$ , il faut préciser les flèches identités et les compositions. Les flèches identités sont évidemment les termes  $\text{id}_A \in \text{Arr}(\mathcal{C}_i)(A, A)$ , et les opérations de composition sont les fonctions

$$\begin{aligned} \circ : \text{Arr}(\mathcal{C}_i)(B, C) \times \text{Arr}(\mathcal{C}_i)(A, B) &\rightarrow \text{Arr}(\mathcal{C}_i)(A, C) \\ (g, f) &\mapsto g \circ f. \end{aligned}$$

**Lemme 3.5** *Pour tout  $i \geq 0$ , on a :*

1.  $\text{Obj}(\mathcal{C}_i) \subseteq \text{Obj}(\mathcal{C}_{i+1})$  ;
2.  $\forall A, B \in \text{Obj}(\mathcal{C}_i), \text{Arr}(\mathcal{C}_i)(A, B) \subseteq \text{Arr}(\mathcal{C}_{i+1})(A, B)$  ;
3.  $\forall A, B \in \text{Obj}(\mathcal{C}_i), \mathcal{R}(\mathcal{C}_i)(A, B) \subseteq \mathcal{R}(\mathcal{C}_{i+1})(A, B)$ .

*Preuve.* Par induction sur  $i$ . Pour  $i = 0$ , le point 1 provient de la règle (1), le point 2 provient de la règle (4), et le point 3 provient de la règle (11). Pour  $i + 1$ , on prouve les trois points en parallèle, par induction sur la structure des éléments de  $\text{Obj}(\mathcal{C}_{i+1})$ ,  $\text{Arr}(\mathcal{C}_{i+1})$  et  $\mathcal{R}(\mathcal{C}_{i+1})$ . Cela nécessite de considérer toutes les règles (1)–(23).  $\square$

**Lemme 3.6** *Pour tout  $i \geq 0$ , on a :*

1.  $\mathcal{R}(\mathcal{C}_{i+1})$  est une congruence ;
2.  $\mathcal{C}_{i+1}$  est une précatégorie ;
3. l'objet  $\emptyset$  est pré-initial dans  $\mathcal{C}_{i+1}$ .

*Preuve.* Toutes les propositions dérivent de façon évidente des règles qui définissent  $\text{Obj}(\mathcal{C}_{i+1})$ ,  $\text{Arr}(\mathcal{C}_{i+1})$  et  $\mathcal{R}(\mathcal{C}_{i+1})$ . Plus précisément :

1. règles (12)–(15) ;
2. règles (5), (6) et (16)–(18) ;
3. règles (7) et (19).

$\square$

**Exemple 3.1** Supposons que  $\mathcal{C}_0$  contient trois objets  $A$ ,  $B$  et  $C$ , et deux flèches  $f : A \rightarrow B$  et  $g : A \rightarrow C$ . On peut construire une pré-somme amalgamée de deux façons différentes :

$$\frac{A, B, C \in \text{Obj}(\mathcal{C}_0) ; f \in \text{Arr}(\mathcal{C}_0)(A, B) ; g \in \text{Arr}(\mathcal{C}_0)(A, C)}{P_1 \equiv \text{push}(A, B, C, f, g) \in \text{Obj}(\mathcal{C}_1)} \quad (3)$$

$$\frac{A, C, B, \in \text{Obj}(\mathcal{C}_0) ; g \in \text{Arr}(\mathcal{C}_0)(A, C) ; f \in \text{Arr}(\mathcal{C}_0)(A, B)}{P_2 \equiv \text{push}(A, C, B, g, f) \in \text{Obj}(\mathcal{C}_1)} \quad (3)$$

On a également :

$$\frac{A, C, B, \in \text{Obj}(\mathcal{C}_0) ; g \in \text{Arr}(\mathcal{C}_0)(A, C) ; f \in \text{Arr}(\mathcal{C}_0)(A, B)}{\&_1(A, C, B, g, f) \circ g = \&_2(A, C, B, g, f) \circ f \in \text{Arr}(\mathcal{C}_1)(A, P_2)} \quad (20)$$

$$\frac{A, B, C \in \text{Obj}(\mathcal{C}_0) ; f \in \text{Arr}(\mathcal{C}_0)(A, B) ; g \in \text{Arr}(\mathcal{C}_0)(A, C)}{\&_1(A, B, C, f, g) \circ f = \&_2(A, B, C, f, g) \circ g \in \text{Arr}(\mathcal{C}_1)(A, P_1)} \quad (20)$$

Par la règle (10), on peut donc construire les termes

$$\begin{aligned} u &\equiv \text{up}(A, B, C, P_2, f, g, \&_1(A, C, B, g, f), \&_2(A, C, B, g, f)) \in \text{Arr}(\mathcal{C}_2)(P_1, P_2), \\ v &\equiv \text{up}(A, C, B, P_1, f, g, \&_1(A, B, C, f, g), \&_2(A, B, C, f, g)) \in \text{Arr}(\mathcal{C}_2)(P_2, P_1). \end{aligned}$$

D'après la règle (12), on a

$$\begin{aligned} \&_1(A, B, C, f, g) &= \&_1(A, B, C, f, g) \in \text{Arr}(\mathcal{C}_1)(B, P_1), \\ \&_2(A, B, C, f, g) &= \&_2(A, B, C, f, g) \in \text{Arr}(\mathcal{C}_1)(C, P_1). \end{aligned}$$

En appliquant la règle (23), on en déduit que

$$u \circ v = \text{id}_{P_2} \in \text{Arr}(\mathcal{C}_2)(P_2, P_2).$$

De même, on montre que

$$v \circ u = \text{id}_{P_1} \in \text{Arr}(\mathcal{C}_2)(P_1, P_1).$$

Nous en déduisons donc que les deux constructions  $P_1$  et  $P_2$  sont isomorphes dans la précatégorie  $\mathcal{C}_2$ , (mais  $P_1$  et  $P_2$  ne sont pas isomorphes dans la précatégorie  $\mathcal{C}_1$ ).

**Remarque 3.1** Les triplets

$$(\text{push}(A, B, C, f, g), \&_1(A, B, C, f, g), \&_2(A, B, C, f, g))$$

ne sont pas, en général, des pré-sommes amalgamées. Le terme  $\text{push}(A, B, C, f, g)$ , dans l'exemple 3.1, n'est pas une pré-somme amalgamée dans  $\mathcal{C}_1$  car la flèche  $u$  ne fait pas partie de  $\mathcal{C}_1$ . Par contre, lorsque nous considérerons la précatégorie des termes, limite de la suite de précatégories  $\mathcal{C}_i$  (section 3.6), les triplets

$$(\text{push}(A, B, C, f, g), \&_1(A, B, C, f, g), \&_2(A, B, C, f, g))$$

seront bien des pré-sommes amalgamées.

**Lemme 3.7** *L'ensemble des flèches d'un objet de  $\mathcal{C}_0$  vers l'objet pré-initial est vide. Autrement dit,*

$$\forall i \geq 1, \forall A \in \text{Obj}(\mathcal{C}_0), \text{Arr}(\mathcal{C}_i)(A, \emptyset) = \emptyset.$$

*Preuve.* Par induction sur la définition de  $\text{Arr}(\mathcal{C}_i)$ . □

### 3.4 Simplification des flèches

Dans cette section, nous définissons pour tout entier  $i \geq 0$ , un *système de simplification* des flèches de la précatégorie  $\mathcal{C}_{i+1}$ . Il s'agit ici d'un outil technique qui nous permet de démontrer en section 3.5 que, pour tout  $i$ , la précatégorie  $\mathcal{C}_i$  est une extension conservatrice de  $\mathcal{C}_0$ .

**Définition 3.10** (Flèche élémentaire) On dit qu'une flèche  $f_j \in \text{Arr}(\mathcal{C}_i)(A, B)$  est *élémentaire* si et seulement si l'une des conditions suivantes est satisfaite.

1.  $f_j \in \text{Arr}(\mathcal{C}_0)(A, B)$ ;
2.  $f_j \equiv \&_k(A, B, C, f, g)$ ;
3.  $f_j \equiv \text{up}(A, B, C, D, f, g, f', g')$ ;
4.  $f_j \equiv \text{id}_X$  et  $f_j \notin \text{Arr}(\mathcal{C}_0)(X, X)$ ;
5.  $f_j \equiv \text{j}_X$ .

Dans tout le reste de cette section, nous considérons les flèches de  $\mathcal{C}_i$  comme des compositions

$$f \equiv f_n \circ f_{n-1} \circ \cdots \circ f_1$$

avec, pour tout  $j \in \{1, \dots, n\}$ ,  $f_j$  est une flèche élémentaire. L'entier  $n$  est appelé *longueur* du terme  $f$ .

Cette présentation simplifiée est justifiée par la règle (16), c'est-à-dire l'associativité de la composition.

Le système de simplification est composé de *règles de simplification*, qui définissent une relation  $\longrightarrow$  sur les flèches de la précatégorie  $\mathcal{C}_{i+1}$ . On considère ensuite la fermeture réflexive et transitive, notée  $\longrightarrow^*$ , de la relation  $\longrightarrow$ .

Nous considérons des règles de simplification de l'une des deux formes suivantes.

$$\frac{P}{f \longrightarrow f'} \quad (\text{Axiome})$$

$$\frac{P}{u \longrightarrow u' \Rightarrow f \longrightarrow f'} \quad (\text{Règle de contexte})$$

où  $f, f', u$  et  $u'$  sont des flèches de la précatégorie  $\mathcal{C}_{i+1}$ , et  $P$  est l'ensemble des conditions qui assurent que les termes  $f, f', u$  et  $u'$  sont bien formés.

Intuitivement, si  $f \longrightarrow^* f'$  (c'est-à-dire si la flèche  $f$  peut se simplifier en la flèche  $f'$ ), alors  $f$  et  $f'$  sont équivalentes dans la précatégorie  $\mathcal{C}_{i+1}$ . La réciproque n'est pas vraie. Autrement dit, le système de simplification est cohérent, mais pas complet, par rapport à l'équivalence dans  $\mathcal{C}_{i+1}$ .

On considère le système de simplification des flèches de  $\mathcal{C}_{i+1}$  composé des règles suivantes.

**Axiomes**

$$\frac{A, B \in \text{Obj}(\mathcal{C}_{i+1}) ; f \in \text{Arr}(\mathcal{C}_{i+1})(A, B)}{f \circ \text{id}_A \longrightarrow f} \quad (\rightarrow 17)$$

$$\frac{A, B \in \text{Obj}(\mathcal{C}_{i+1}) ; f \in \text{Arr}(\mathcal{C}_{i+1})(A, B)}{\text{id}_B \circ f \longrightarrow f} \quad (\rightarrow 18)$$

$$\frac{A, B, C, D \in \text{Obj}(\mathcal{C}_i) ; f \in \text{Arr}(\mathcal{C}_i)(A, B) ; g \in \text{Arr}(\mathcal{C}_i)(A, C) \\ f' \in \text{Arr}(\mathcal{C}_i)(B, D) ; g' \in \text{Arr}(\mathcal{C}_i)(C, D) ; f' \circ f = g' \circ g \in \text{Arr}(\mathcal{C}_i)(A, D)}{\text{up}(A, B, C, D, f, g, f', g') \circ \&_1(A, B, C, f, g) \longrightarrow f'} \quad (\rightarrow 20)$$

$$\frac{A, B, C, D \in \text{Obj}(\mathcal{C}_i) ; f \in \text{Arr}(\mathcal{C}_i)(A, B) ; g \in \text{Arr}(\mathcal{C}_i)(A, C) \\ f' \in \text{Arr}(\mathcal{C}_i)(B, D) ; g' \in \text{Arr}(\mathcal{C}_i)(C, D) ; f' \circ f = g' \circ g \in \text{Arr}(\mathcal{C}_i)(A, D)}{\text{up}(A, B, C, D, f, g, f', g') \circ \&_2(A, B, C, f, g) \longrightarrow g'} \quad (\rightarrow 21)$$

**Règles de contexte**

$$\frac{A, B, C \in \text{Obj}(\mathcal{C}_{i+1}) ; f, f' \in \text{Arr}(\mathcal{C}_{i+1})(A, B) ; g \in \text{Arr}(\mathcal{C}_{i+1})(B, C)}{f \longrightarrow f' \Rightarrow g \circ f \longrightarrow g \circ f'} \quad (\rightarrow 15 \text{ i})$$

$$\frac{A, B, C \in \text{Obj}(\mathcal{C}_{i+1}) ; f \in \text{Arr}(\mathcal{C}_{i+1})(A, B) ; g, g' \in \text{Arr}(\mathcal{C}_{i+1})(B, C)}{g \longrightarrow g' \Rightarrow g \circ f \longrightarrow g' \circ f} \quad (\rightarrow 15 \text{ ii})$$

$$\frac{A, B, C, D \in \text{Obj}(\mathcal{C}_i) ; f \in \text{Arr}(\mathcal{C}_i)(A, B) ; g \in \text{Arr}(\mathcal{C}_i)(A, C) \\ f' \in \text{Arr}(\mathcal{C}_i)(B, D) ; g' \in \text{Arr}(\mathcal{C}_i)(C, D) ; f'' \in \text{Arr}(\mathcal{C}_i)(B, D) \\ f' \circ f = g' \circ g \in \text{Arr}(\mathcal{C}_i)(A, D)}{f' \longrightarrow f'' \Rightarrow \text{up}(A, B, C, D, f, g, f', g') \longrightarrow \text{up}(A, B, C, D, f, g, f'', g')} \quad (\rightarrow 23 \text{ i})$$

$$\frac{A, B, C, D \in \text{Obj}(\mathcal{C}_i) ; f \in \text{Arr}(\mathcal{C}_i)(A, B) ; g \in \text{Arr}(\mathcal{C}_i)(A, C) \\ f' \in \text{Arr}(\mathcal{C}_i)(B, D) ; g' \in \text{Arr}(\mathcal{C}_i)(C, D) ; g'' \in \text{Arr}(\mathcal{C}_i)(C, D) \\ f' \circ f = g' \circ g \in \text{Arr}(\mathcal{C}_i)(A, D)}{g' \longrightarrow g'' \Rightarrow \text{up}(A, B, C, D, f, g, f', g') \longrightarrow \text{up}(A, B, C, D, f, g, f', g'')} \quad (\rightarrow 23 \text{ ii})$$

**Remarque 3.2** Ce système de simplification ressemble à un système de réécriture (cf. par exemple [DJ90]). Il y a cependant deux différences avec les systèmes de réécriture classiques.

1. Nous avons besoin de prémisses pour assurer que les termes manipulés sont bien formés.

2. Nous n'autorisons pas toutes les règles de passage au contexte. Nous avons une règle de passage au contexte uniquement pour l'opérateur de composition (règles ( $\rightarrow$  15 i) et ( $\rightarrow$  15 ii)), et pour les deux derniers paramètres de l'opérateur  $\text{up}$  (règles ( $\rightarrow$  23 i) et ( $\rightarrow$  23 ii)).

**Lemme 3.8** *Soit  $f \in \text{Arr}(\mathcal{C}_i)(A, B)$ . alors,*

1.  $f \longrightarrow f' \Rightarrow f' \in \text{Arr}(\mathcal{C}_i)(A, B)$  ;
2.  $f \longrightarrow f' \Rightarrow f = f' \in \text{Arr}(\mathcal{C}_i)(A, B)$ .

*Preuve.* Aucune difficulté. □

**Lemme 3.9** (*Terminaison du système de simplification*) *Le système de simplification termine, c'est-à-dire qu'il n'existe pas de séquence infinie*

$$f_1 \longrightarrow f_2 \longrightarrow \cdots f_n \longrightarrow \cdots.$$

*Preuve.* Pour chaque axiome

$$\frac{P}{f \longrightarrow f'}$$

$f'$  est un sous terme de  $f$ . □

**Lemme 3.10** (*Confluence du système de simplification*)

*Le système de simplification est confluente, c'est-à-dire que si  $f \longrightarrow f_1$  et  $f \longrightarrow f_2$ , alors il existe  $f'$  tel que  $f_1 \longrightarrow^* f'$  et  $f_2 \longrightarrow^* f'$ .*

*Preuve.* Aucune difficulté. □

**Définition 3.11** (*Forme normale*) Soit une flèche  $f \in \text{Arr}(\mathcal{C}_{i+1})(A, B)$ . On dit que la flèche  $f$  est sous *forme normale*, ou *irréductible*, si il n'existe aucune simplification  $f \longrightarrow f'$ .

**Proposition 3.1** *Toute flèche  $f \in \text{Arr}(\mathcal{C}_i)(A, B)$  a une forme normale.*

*Preuve.* C'est une conséquence immédiate de la terminaison et confluence du système de simplification. □

La forme normale de  $f$  sera notée  $\mathcal{N}(f)$ .

**Théorème 3.1** (*Cohérence de la simplification*) *Soit  $f \in \text{Arr}(\mathcal{C}_i)(A, B)$ . On a*

$$f = \mathcal{N}(f) \in \text{Arr}(\mathcal{C}_i)(A, B).$$

*Preuve.* Par induction sur la longueur de la simplification  $f \rightarrow^* \mathcal{N}(f)$ .

Pour une simplification de longueur 0, le résultat est évident.

Supposons  $f \rightarrow f' \rightarrow^* \mathcal{N}(f')$ . D'après le lemme 3.8,  $f = f' \in \text{Arr}(\mathcal{C}_i)(A, B)$ .

Ensuite, par hypothèse d'induction, on a  $f' = \mathcal{N}(f') \in \text{Arr}(\mathcal{C}_i)(A, B)$ , d'où  $f = \mathcal{N}(f') \in \text{Arr}(\mathcal{C}_i)(A, B)$ .  $\square$

Le théorème suivant affirme que toute flèche pour laquelle le domaine et le codomaine sont dans  $\mathcal{C}_0$  a pour forme normale une flèche de  $\mathcal{C}_0$ .

**Théorème 3.2** (*Forme normale (1)*)

Soit  $A, B \in \text{Obj}(\mathcal{C}_0)$  et  $m \in \text{Arr}(\mathcal{C}_i)(A, B)$ . On a

$$\mathcal{N}(m) \in \text{Arr}(\mathcal{C}_0)(A, B).$$

*Preuve.* Soit  $A, B \in \text{Obj}(\mathcal{C}_0)$  et  $m \in \text{Arr}(\mathcal{C}_i)(A, B)$ . Soit

$$\mathcal{N}(m) \equiv m_n \circ m_{n-1} \cdots m_2 \circ m_1,$$

avec pour tout  $j \in \{1, \dots, n\}$ ,  $m_j$  est une flèche élémentaire.

Pour tout  $j \in \{1, \dots, n\}$ , on a

- Si  $m_j$  n'est pas une flèche de  $\mathcal{C}_0$ , alors  $m_j \neq \text{id}_X$ , sinon nécessairement  $n \geq 2$  et donc  $\mathcal{N}(m)$  n'est pas irréductible.
- $m_j \neq j_X$ , car d'après le lemme 3.7,  $\text{Arr}(\mathcal{C}_i)(A, \emptyset) = \emptyset$ .

Par conséquent, pour tout  $j \in \{1, \dots, n\}$ , l'une des conditions suivantes est vérifiée

1.  $m_j \in \text{Arr}(\mathcal{C}_0)(X, Y)$ ;
2.  $m_j \equiv \&k(A_0, A_1, A_2, f_1, f_2)$ , avec  $k \in \{1, 2\}$ ;
3.  $m_j \equiv \text{up}(A_0, A_1, A_2, A_3, f_1, f_2, f'_1, f'_2)$ .

Par l'absurde, supposons qu'il existe  $j$  tel que  $m_j \notin \text{Arr}(\mathcal{C}_0)(X, Y)$ . Soit  $j_0$  le plus petit  $j$  qui satisfait cette propriété. On a donc  $m_{j_0} \in \text{Arr}(\mathcal{C}_i)(X, Y)$ , avec  $X \in \text{Obj}(\mathcal{C}_0)$ .

Par conséquent

$$m_{j_0} \equiv \&k(A_0, A_1, A_2, f_1, f_2).$$

Si  $m_{j_0+1} \equiv \text{up}(A_0, A_1, A_2, A_3, f_1, f_2, f'_1, f'_2)$ , alors  $\mathcal{N}(m)$  n'est pas irréductible, donc

$$m_{j_0+1} \equiv \&k'(A'_0, A'_1, A'_2, f'_1, f'_2).$$

Finalement, on obtient

$$m_n \equiv \&k''(A_0'', A_1'', A_2'', f_1'', f_2'').$$

ce qui est en contradiction avec  $B \in \text{Obj}(\mathcal{C}_0)$ . Par conséquent, pour tout  $j \in \{1, \dots, n\}$ ,  $m_j$  est une flèche de  $\mathcal{C}_0$ , et donc  $\mathcal{N}(m) \in \text{Arr}(\mathcal{C}_0)(A, B)$ .  $\square$

**Théorème 3.3** (*Forme normale (2)*)

Soit  $A \in \text{Obj}(\mathcal{C}_0)$ ,  $B \equiv \text{push}(A_0, A_1, A_2, f_1, f_2) \in \text{Obj}(\mathcal{C}_i)$ ,  $m \in \text{Arr}(\mathcal{C}_i)(A, B)$ . Posons

$$\mathcal{N}(m) \equiv m_n \circ m_{n-1} \circ \cdots \circ m_1,$$

avec  $n \geq 1$  et  $\forall j \in \{1, \dots, n\}$ ,  $m_j$  est une flèche élémentaire. Alors,

1.  $m_n \equiv \&_k(A_0, A_1, A_2, f_1, f_2)$ , avec  $k \in \{1, 2\}$ ;
2. si  $n \geq 2$ , alors  $m_{n-1} \circ \cdots \circ m_1 \in \text{Arr}(\mathcal{C}_{i-1})(A, A_k)$ .

*Preuve.* Similaire à la preuve du théorème 3.2. La preuve complète est développée en appendice A.1.  $\square$

**Théorème 3.4** (*Forme normale (3)*)

Soit  $A \equiv \text{push}(A_0, A_1, A_2, f_1, f_2) \in \text{Obj}(\mathcal{C}_i)$ ,  $B \in \text{Obj}(\mathcal{C}_0)$  et  $m \in \text{Arr}(\mathcal{C}_i)(A, B)$ . Posons

$$\mathcal{N}(m) \equiv m_n \circ m_{n-1} \circ \cdots \circ m_1,$$

avec  $n \geq 1$  et  $\forall j \in \{1, \dots, n\}$ ,  $m_j$  est une flèche élémentaire. Alors

$$m_1 \equiv \text{up}(A_0, A_1, A_2, A_3, f_1, f_2, g_1, g_2).$$

*Preuve.* Similaire à la preuve du théorème 3.2. La preuve complète est développée en appendice A.2.  $\square$

La forme normale d'une flèche n'est pas unique relativement à l'équivalence dans  $\mathcal{C}_i$ . Dans le cas général,

$$m = m' \in \text{Arr}(\mathcal{C}_i)(X, Y) \not\equiv \mathcal{N}(m) \equiv \mathcal{N}(m').$$

Par exemple, on a

$$\&_1(A, B, C, f, g) \circ f = \&_2(A, B, C, f, g) \circ g \in \text{Arr}(\mathcal{C}_i)(A, \text{push}(A, B, C, f, g)).$$

Ces deux flèches sont irréductibles et équivalentes dans  $\mathcal{C}_i$ , mais ne sont pas identiques. Par contre, on a

$$\forall X, Y \in \text{Obj}(\mathcal{C}_0), m = m' \in \text{Arr}(\mathcal{C}_i)(X, Y) \Rightarrow \mathcal{N}(m) = \mathcal{N}(m') \in \text{Arr}(\mathcal{C}_0)(X, Y).$$

Pour montrer ce résultat, nous démontrons le théorème suivant.

**Théorème 3.5** (*“Unicité” de la forme normale*)

Soit  $m, m' \in \text{Arr}(\mathcal{C}_i)(X, Y)$ , tels que  $m = m' \in \text{Arr}(\mathcal{C}_i)(X, Y)$ .

1. Si  $X, Y \in \text{Obj}(\mathcal{C}_0)$ , alors

$$\mathcal{N}(m) = \mathcal{N}(m') \in \text{Arr}(\mathcal{C}_0)(X, Y).$$

2. Si  $X \in \text{Obj}(\mathcal{C}_0)$ ,  $Y \equiv \text{push}(A_0, A_1, A_2, f_1, f_2)$ , alors  
 $\forall Z \in \text{Obj}(\mathcal{C}_0)$ ,  $\forall h \in \text{Arr}(\mathcal{C}_i)(Y, Z)$ , on a

$$\mathcal{N}(h \circ m) = \mathcal{N}(h \circ m') \in \text{Arr}(\mathcal{C}_0)(X, Z).$$

3. Si  $X \equiv \text{push}(A_0, A_1, A_2, f_1, f_2)$ ,  $Y \in \text{Obj}(\mathcal{C}_0)$ , alors  
 $\forall W \in \text{Obj}(\mathcal{C}_0)$ ,  $\forall p \in \text{Arr}(\mathcal{C}_i)(W, X)$ , on a

$$\mathcal{N}(m \circ p) = \mathcal{N}(m' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Y).$$

4. Si  $X \equiv \text{push}(A_0, A_1, A_2, f_1, f_2)$ ,  $Y \equiv \text{push}(A'_0, A'_1, A'_2, f'_1, f'_2)$ , alors  
 $\forall W, Z \in \text{Obj}(\mathcal{C}_0)$ ,  $\forall h \in \text{Arr}(\mathcal{C}_i)(Y, Z)$ ,  $\forall p \in \text{Arr}(\mathcal{C}_i)(W, X)$ , on a

$$\mathcal{N}(h \circ m \circ p) = \mathcal{N}(h \circ m' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Z).$$

*Preuve.* On prouve les points 1 à 4 par induction sur  $i$ . Pour  $i = 0$ , le point 1 est évident, et il n'y a rien à prouver pour les points 2 à 4. Pour  $i + 1$ , on doit faire une induction sur la longueur de  $\mathcal{N}(h)$  pour prouver les points 2 et 4. On prouve alors les points 1, 2, 3, 4 en parallèle, par induction sur la longueur de la preuve que

$$m = m' \in \text{Arr}(\mathcal{C}_{i+1})(X, Y).$$

Cette preuve est détaillée en Appendice A.3. □

### 3.5 Extensions conservatrices

La suite de précatégories  $\mathcal{C}_i$  satisfait deux propriétés importantes : on n'introduit pas de nouvelles flèches entre des objets de  $\mathcal{C}_0$ , et on n'introduit pas de nouvelles équivalences entre des flèches de  $\mathcal{C}_0$ . On dit que  $\mathcal{C}_i$  est une *extension conservatrice* de  $\mathcal{C}_0$ .

Pour tout  $i \leq j$ , on définit un préfoncteur  $J_{i,j} : \mathcal{C}_i \rightarrow \mathcal{C}_j$  par :

$$\begin{array}{ccc} J_{i,j} : \mathcal{C}_i & \rightarrow & \mathcal{C}_j \\ & A & \mapsto A \\ & f & \mapsto f \end{array}$$

$J_{i,j}$  est bien un préfoncteur car d'après le lemme 3.5,

- $A \in \text{Obj}(\mathcal{C}_i) \Rightarrow A \in \text{Obj}(\mathcal{C}_j)$  ;

- $f \in \text{Arr}(\mathcal{C}_i)(A, B) \Rightarrow f \in \text{Arr}(\mathcal{C}_j)(A, B)$  ;
- $f = f' \in \text{Arr}(\mathcal{C}_i)(A, B) \Rightarrow f = f' \in \text{Arr}(\mathcal{C}_j)(A, B)$ .

**Théorème 3.6** *Pour tout entier  $j$ , la précatégorie  $\mathcal{C}_j$  est une extension conservatrice de  $\mathcal{C}_0$ . Autrement dit, le préfoncteur  $J_{0,j}$  est plein et fidèle.*

Ce théorème est équivalent à :

$\forall j \in \mathbb{N}, \forall A, B \in \text{Obj}(\mathcal{C}_0)$ ,

1.  $\forall h \in \text{Arr}(\mathcal{C}_j)(A, B), \exists h' \in \text{Arr}(\mathcal{C}_0)(A, B) ; h = h' \in \text{Arr}(\mathcal{C}_j)(A, B)$  ;
2.  $\forall h', h'' \in \text{Arr}(\mathcal{C}_0)(A, B)$ ,

$$h' = h'' \in \text{Arr}(\mathcal{C}_j)(A, B) \Rightarrow h' = h'' \in \text{Arr}(\mathcal{C}_0)(A, B).$$

*Preuve.* C'est une conséquence immédiate des théorèmes 3.1, 3.2, et 3.5.

1. Il suffit de prendre  $h' \equiv \mathcal{N}(h)$ . D'après le théorème 3.2,  $\mathcal{N}(h) \in \text{Arr}(\mathcal{C}_0)(A, B)$ .
2.  $h' = h'' \in \text{Arr}(\mathcal{C}_j)(A, B)$ , donc d'après le théorème 3.5,

$$\mathcal{N}(h') = \mathcal{N}(h'') \in \text{Arr}(\mathcal{C}_0)(A, B).$$

De plus, d'après le théorème 3.1, on a

$$\begin{aligned} \mathcal{N}(h') &= h' \in \text{Arr}(\mathcal{C}_0)(A, B) \\ \mathcal{N}(h'') &= h'' \in \text{Arr}(\mathcal{C}_0)(A, B) \end{aligned}$$

d'où le résultat. □

**Remarque 3.3** En général, pour  $1 \leq i < j$ ,  $\mathcal{C}_j$  n'est pas une extension conservatrice de  $\mathcal{C}_i$ . En effet :

1. Le préfoncteur  $J_{i,j}$  n'est pas plein, car la règle (10) peut introduire dans la précatégorie  $\mathcal{C}_{i+1}$  des nouvelles flèches entre des objets de  $\mathcal{C}_i$ . Dans l'exemple 3.1,  $u$  est une flèche entre deux objets de  $\mathcal{C}_1$  qui fait partie de  $\mathcal{C}_2$ , mais pas de  $\mathcal{C}_1$ .
2. Le préfoncteur  $J_{i,j}$  n'est pas fidèle, car deux flèches de  $\mathcal{C}_i$  peuvent être équivalentes à une troisième flèche de  $\mathcal{C}_{i+1}$  (et donc, par transitivité, équivalentes dans  $\mathcal{C}_{i+1}$ ), sans être équivalentes dans  $\mathcal{C}_i$ . On peut donc introduire dans  $\mathcal{C}_{i+1}$  des nouvelles équivalences entre flèches de  $\mathcal{C}_i$ .

### 3.6 Précatégorie $\text{TERME}(\mathcal{C}_0)$

Nous pouvons maintenant définir la précatégorie des termes, qui va nous servir de syntaxe pour les constructions de colimites. Les objets de cette précatégorie sont les éléments de  $\text{Obj}(\mathcal{C}_i)$ , pour  $i$  quelconque, et les flèches entre deux objets  $A$  et  $B$  sont les éléments de  $\text{Arr}(\mathcal{C}_j)(A, B)$ , pour tous les  $j$  tels que  $A, B \in \text{Obj}(\mathcal{C}_j)$ . De plus, deux flèches sont équivalentes si et seulement si elles sont équivalentes dans un des ensembles  $\text{Arr}(\mathcal{C}_j)(A, B)$ . On définit donc la précatégorie  $\text{TERME}(\mathcal{C}_0)$  comme la “limite” de la suite de précatégories  $\mathcal{C}_i$ .

**Définition 3.12** (Précatégorie  $\text{TERME}(\mathcal{C}_0)$ )

La précatégorie des termes  $\text{TERME}(\mathcal{C}_0)$  est définie de la façon suivante.

- L'ensemble des objets est

$$\text{Obj}(\text{TERME}(\mathcal{C}_0)) = \bigcup_{i=0}^{\infty} \text{Obj}(\mathcal{C}_i).$$

- Soit  $A, B \in \text{Obj}(\text{TERME}(\mathcal{C}_0))$ . Il existe  $k \in \mathbb{N}$  tel que  $A, B \in \text{Obj}(\mathcal{C}_k)$ . L'ensemble des flèches de  $A$  vers  $B$  est

$$\text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B) = \bigcup_{i=k}^{\infty} \text{Arr}(\mathcal{C}_i)(A, B).$$

- Soit  $A, B \in \text{Obj}(\text{TERME}(\mathcal{C}_0))$ . Il existe  $k \in \mathbb{N}$  tel que  $A, B \in \text{Obj}(\mathcal{C}_k)$ . La relation d'équivalence sur  $\text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$  est

$$\mathcal{R}(\text{TERME}(\mathcal{C}_0))(A, B) = \bigcup_{i=k}^{\infty} \mathcal{R}(\mathcal{C}_i)(A, B).$$

D'après le lemme 3.5, ces ensembles sont bien définis. En particulier, la définition des ensembles  $\text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$  et  $\mathcal{R}(\text{TERME}(\mathcal{C}_0))(A, B)$  est indépendante de l'entier  $k$  choisi.

**Proposition 3.2**  $\text{TERME}(\mathcal{C}_0)$  est une précatégorie.

Soit  $J : \mathcal{C}_0 \rightarrow \text{TERME}(\mathcal{C}_0)$  le préfoncteur défini par

$$\begin{array}{ccc} J : \mathcal{C}_0 & \rightarrow & \text{TERME}(\mathcal{C}_0) \\ A & \mapsto & A \\ f & \mapsto & f. \end{array}$$

**Théorème 3.7** La précatégorie  $\text{TERME}(\mathcal{C}_0)$  est une extension conservatrice de  $\mathcal{C}_0$ . Autrement dit, le préfoncteur  $J : \mathcal{C}_0 \rightarrow \text{TERME}(\mathcal{C}_0)$  est plein et fidèle.

Par conséquent, dans  $\text{TERME}(\mathcal{C}_0)$ , on n'ajoute pas de flèche entre des objets de  $\mathcal{C}_0$ , et on n'ajoute pas d'équivalence entre flèches de  $\mathcal{C}_0$ . Cela montre la *cohérence hiérarchique* de la construction.

*Preuve.* D'après le théorème 3.6,  $\forall k \in \mathbb{N}$ , la précatégorie  $\mathcal{C}_k$  est une extension conservatrice de  $\mathcal{C}_0$ .

1. Soit  $A, B \in \text{Obj}(\mathcal{C}_0)$ , et  $h \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$ . Il existe  $k \in \mathbb{N}$  tel que  $h \in \text{Arr}(\mathcal{C}_k)(A, B)$ . Comme  $\mathcal{C}_k$  est une extension conservatrice de  $\mathcal{C}_0$ , il existe  $h' \in \text{Arr}(\mathcal{C}_0)(A, B)$ , tel que  $h = h' \in \text{Arr}(\mathcal{C}_k)(A, B)$ , et donc

$$h = h' \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B).$$

2. Soit  $A, B \in \text{Obj}(\mathcal{C}_0)$ , et  $h', h'' \in \text{Arr}(\mathcal{C}_0)(A, B)$ , tels que

$$h' = h'' \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B).$$

Il existe  $k \in \mathbb{N}$  tel que  $h' = h'' \in \text{Arr}(\mathcal{C}_k)(A, B)$ . Comme  $\mathcal{C}_k$  est une extension conservatrice de  $\mathcal{C}_0$ ,

$$h' = h'' \in \text{Arr}(\mathcal{C}_0)(A, B).$$

□

**Théorème 3.8** *La précatégorie  $\text{TERME}(\mathcal{C}_0)$  est finiment pré-cocomplète. Plus précisément,*

1. l'objet  $\emptyset$  est un choix d'objet pré-initial dans  $\text{TERME}(\mathcal{C}_0)$  ;
2. si  $A, B, C \in \text{Obj}(\text{TERME}(\mathcal{C}_0))$ ,  
 $f \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$  et  $g \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, C)$ , alors le triplet

$$(\text{push}(A, B, C, f, g), \&_1(A, B, C, f, g), \&_2(A, B, C, f, g))$$

est un choix de pré-somme amalgamée de  $B$  et  $C$  par rapport à  $f$  et  $g$  dans  $\text{TERME}(\mathcal{C}_0)$ .

*Preuve.*

1. L'objet  $\emptyset$  est pré-initial. Soit  $A \in \text{Obj}(\text{TERME}(\mathcal{C}_0))$ . Il existe  $k \geq 1$  tel que  $A \in \text{Obj}(\mathcal{C}_k)$ . Donc on a une flèche  $j_A \in \text{Arr}(\mathcal{C}_k)(\emptyset, A) \subseteq \text{Arr}(\text{TERME}(\mathcal{C}_0))(\emptyset, A)$ . Soit deux flèches  $u, v \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(\emptyset, A)$ . Il existe  $k' \geq 1$  tel que  $u, v \in \text{Arr}(\mathcal{C}_{k'})$ , donc d'après la règle (19),  $u = v \in \text{Arr}(\mathcal{C}_{k'})$  et donc  $u = v \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(\emptyset, A)$ .

2. La preuve est similaire.

□

Considérons une précatégorie  $\mathcal{E}$  ayant un objet pré-initial choisi  $\emptyset^\mathcal{E}$  et des pré-sommes amalgamées choisies. Si  $A, B$  et  $C$  sont trois objets de  $\mathcal{E}$ , et  $f \in \text{Arr}(\mathcal{E})(A, B)$ ,  $g \in \text{Arr}(\mathcal{E})(A, C)$  deux flèches de  $\mathcal{E}$ , on note

$$(push^\mathcal{E}(A, B, C, f, g), \&_1^\mathcal{E}(A, B, C, f, g), \&_2^\mathcal{E}(A, B, C, f, g))$$

la pré-somme amalgamée choisie de  $B$  et  $C$  par rapport à  $f$  et  $g$  dans  $\mathcal{E}$ . On suppose de plus que, pour tout objet  $A$  de  $\mathcal{E}$ , on a un choix de flèche

$$j_A^\mathcal{E} \in \text{Arr}(\mathcal{E})(\emptyset^\mathcal{E}, A).$$

De même, si  $A, B, C, D \in \text{Obj}(\mathcal{E})$ ,  $f \in \text{Arr}(\mathcal{E})(A, B)$ ,  $g \in \text{Arr}(\mathcal{E})(A, C)$ ,  $f' \in \text{Arr}(\mathcal{E})(B, D)$ ,  $g' \in \text{Arr}(\mathcal{E})(C, D)$ , tel que  $f' \circ f = g' \circ g \in \text{Arr}(\mathcal{E})(A, D)$ , on a un choix de flèche

$$up^\mathcal{E}(A, B, C, D, f, g, f', g') \in \text{Arr}(\mathcal{E})(push^\mathcal{E}(A, B, C, f, g), D)$$

telle que

$$\begin{aligned} up^\mathcal{E}(A, B, C, D, f, g, f', g') \circ \&_1^\mathcal{E}(A, B, C, f, g) &= f' \in \text{Arr}(\mathcal{E})(B, D) \\ up^\mathcal{E}(A, B, C, D, f, g, f', g') \circ \&_2^\mathcal{E}(A, B, C, f, g) &= g' \in \text{Arr}(\mathcal{E})(C, D). \end{aligned}$$

**Théorème 3.9** (*La précatégorie TERME( $\mathcal{C}_0$ ) est librement engendrée par objet pré-initial choisi et pré-sommes amalgamées choisies*)

*Soit un préfoncteur  $F : \mathcal{C}_0 \rightarrow \mathcal{E}$ . Alors il existe un unique préfoncteur*

$$G : \text{TERME}(\mathcal{C}_0) \rightarrow \mathcal{E}$$

*tel que*

- $G \circ J \equiv F$  ;
- $G(g \circ f) \equiv G(g) \circ G(f)$  ;
- $G(id_A) \equiv id_{G(A)}$  ;
- $G(\emptyset) \equiv \emptyset^\mathcal{E}$  ;
- $G(j_A) \equiv j_{G(A)}^\mathcal{E}$  ;
- $G(push(A, B, C, f, g)) \equiv push^\mathcal{E}(G(A), G(B), G(C), G(f), G(g))$  ;
- $G(\&_1(A, B, C, f, g)) \equiv \&_1^\mathcal{E}(G(A), G(B), G(C), G(f), G(g))$  ;
- $G(\&_2(A, B, C, f, g)) \equiv \&_2^\mathcal{E}(G(A), G(B), G(C), G(f), G(g))$  ;
- $G(up(A, B, C, D, f, g, f', g')) \equiv up^\mathcal{E}(G(A), G(B), G(C), G(D), G(f), G(g), G(f'), G(g'))$ .

La précatégorie  $\text{TERME}(\mathcal{C}_0)$  est donc *librement engendrée par objet pré-initial choisi et pré-sommes amalgamées choisies*. Intuitivement cela signifie que la précatégorie  $\text{TERME}(\mathcal{C}_0)$  satisfait les propriétés suivantes.

1. Toutes les constructions (de pré-colimites sélectionnées, c'est-à-dire utilisant l'objet pré-initial et les pré-sommes amalgamées) sont distinctes.
2.  $\text{TERME}(\mathcal{C}_0)$  ne contient pas d'objets ou flèches autres que ceux de  $\mathcal{C}_0$  et ceux obtenus par les constructions de pré-colimites sélectionnées.

Ces deux propriétés correspondent respectivement à l'existence et l'unicité du préfoncteur  $G$ .

*Preuve.* On construit un préfoncteur  $G : \text{TERME}(\mathcal{C}_0) \rightarrow \mathcal{E}$  par induction sur la structure des objets et flèches de  $\text{TERME}(\mathcal{C}_0)$ . On montre ensuite que ce préfoncteur est unique. La preuve est détaillée en Appendice A.4.  $\square$

Ce théorème permet d'*interpréter*, c'est-à-dire de donner une sémantique aux termes. Une précatégorie d'interprétation est une précatégorie avec un objet pré-initial choisi et des pré-sommes amalgamées choisies. À partir d'une interprétation de  $\mathcal{C}_0$ , c'est-à-dire un préfoncteur

$$\mathcal{S}_0 : \mathcal{C}_0 \rightarrow \mathcal{E}$$

on obtient une interprétation des termes, c'est-à-dire un préfoncteur

$$\mathcal{S} : \text{TERME}(\mathcal{C}_0) \rightarrow \mathcal{E}.$$

Cette précatégorie  $\mathcal{E}$  peut par exemple être la catégorie **Spec** (considérée comme une précatégorie, et accompagnée d'un choix d'objet pré-initial et de choix de pré-sommes amalgamées), ou une catégorie de classes d'algèbres permettant d'interpréter les spécifications de base. Dans le chapitre 4, la précatégorie d'interprétation considérée est  $\text{DIAGR}(\mathcal{C}_0)$ . L'interprétation des objets et flèches de  $\mathcal{C}_0$  est donnée par le préfoncteur  $I_{\mathcal{C}_0} : \mathcal{C}_0 \rightarrow \text{DIAGR}(\mathcal{C}_0)$ . D'après le théorème 3.9, cette interprétation peut être prolongée sur les termes en un préfoncteur

$$\mathcal{D} : \text{TERME}(\mathcal{C}_0) \rightarrow \text{DIAGR}(\mathcal{C}_0).$$

C'est ce préfoncteur qui va nous permettre, dans le chapitre 4, d'associer à tout terme un diagramme.

### 3.7 Catégorie $\text{Terme}(\mathcal{C}_0)$

La catégorie  $\text{Terme}(\mathcal{C}_0)$  est la catégorie associée à la précatégorie  $\text{TERME}(\mathcal{C}_0)$ .

**Définition 3.13** (Catégorie des termes) La catégorie  $\text{Terme}(\mathcal{C}_0)$  est définie de la façon suivante.

- Les objets de  $\text{Terme}(\mathcal{C}_0)$  sont les objets de  $\text{TERME}(\mathcal{C}_0)$ .

- Pour toute flèche  $f \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$ ,  $[f] : A \rightarrow B$ , classe d'équivalence de  $f$  modulo  $\mathcal{R}(\text{TERME}(\mathcal{C}_0))(A, B)$ , est une flèche de  $\text{Terme}(\mathcal{C}_0)$ .

On a un *foncteur*  $J : \mathcal{C}_0 \rightarrow \text{Terme}(\mathcal{C}_0)$  associé au préfoncteur  $J : \mathcal{C}_0 \rightarrow \text{TERME}(\mathcal{C}_0)$ .

**Théorème 3.10** *La catégorie  $\text{Terme}(\mathcal{C}_0)$  est une extension conservatrice de  $\mathcal{C}_0$ . Autrement dit, le foncteur  $J : \mathcal{C}_0 \rightarrow \text{Terme}(\mathcal{C}_0)$  est plein et fidèle.*

*Preuve.* D'après le théorème 3.7, le préfoncteur  $J : \mathcal{C}_0 \rightarrow \text{TERME}(\mathcal{C}_0)$  est plein et fidèle, donc le foncteur associé  $J : \mathcal{C}_0 \rightarrow \text{Terme}(\mathcal{C}_0)$  est également plein et fidèle.  $\square$

**Théorème 3.11**  *$\text{Terme}(\mathcal{C}_0)$  est une catégorie finiment cocomplète. Plus précisément,*

1. *l'objet  $\emptyset$  est initial dans  $\text{Terme}(\mathcal{C}_0)$  ;*
2. *si  $A, B$  et  $C$  sont des objets de  $\text{Terme}(\mathcal{C}_0)$ ,  $[f] : A \rightarrow B$  et  $[g] : A \rightarrow C$  sont des flèches de  $\text{Terme}(\mathcal{C}_0)$ , alors le triplet*

$$(\text{push}(A, B, C, f, g), [\&_1(A, B, C, f, g)], [\&_2(A, B, C, f, g)])$$

*est une somme amalgamée de  $B$  et  $C$  par rapport à  $[f]$  et  $[g]$  dans  $\text{Terme}(\mathcal{C}_0)$ .*

*Preuve.* Application des lemmes 3.3 et 3.4.  $\square$

La catégorie  $\text{Terme}(\mathcal{C}_0)$ , bien que finiment cocomplète, n'a pas de sommes amalgamées *choisies*. En effet, si  $f'$  est un autre représentant de  $[f]$ , et  $g'$  un autre représentant de  $[g]$ , c'est-à-dire si  $[f] = [f']$  et  $[g] = [g']$  dans  $\text{Terme}(\mathcal{C}_0)$ , alors les triplets

$$\begin{aligned} &(\text{push}(A, B, C, f, g), [\&_1(A, B, C, f, g)], [\&_2(A, B, C, f, g)]) \\ &(\text{push}(A, B, C, f', g'), [\&_1(A, B, C, f', g')], [\&_2(A, B, C, f', g')]) \end{aligned}$$

sont deux *choix différents* de somme amalgamée de  $B$  et  $C$  par rapport aux flèches  $[f]$  et  $[g]$ .

Par conséquent, la catégorie  $\text{Terme}(\mathcal{C}_0)$  n'est pas *librement engendrée par objet initial choisi et sommes amalgamées choisies*. Il y a en effet "trop de choix de sommes amalgamées" dans  $\text{Terme}(\mathcal{C}_0)$ . Intuitivement, les constructeurs  $\text{push}$ ,  $\&_1$  et  $\&_2$  dépendent des représentants choisis pour  $[f]$  et  $[g]$ , et donc les *choix* de pré-colimites de  $\text{TERME}(\mathcal{C}_0)$  ne sont pas conservés en passant au quotient. Par contre, on peut retrouver une catégorie *librement engendrée par objet initial choisi et sommes amalgamées choisies* en "fusionnant les choix multiples de sommes amalgamées". Cette construction est décrite dans le paragraphe suivant.

### 3.8 Catégorie libre $\mathcal{L}(\mathcal{C}_0)$

Dans ce paragraphe, nous construisons, à partir de la précatégorie  $\text{TERME}(\mathcal{C}_0)$ , une catégorie  $\mathcal{L}(\mathcal{C}_0)$  librement engendrée par objet initial choisi et sommes amalgamées choisies. L'idée est de définir une relation d'équivalence à la fois sur les objets et les flèches de  $\text{TERME}(\mathcal{C}_0)$ , afin de fusionner les choix multiples de sommes amalgamées. La relation d'équivalence sur les flèches est plus forte que celle de  $\text{TERME}(\mathcal{C}_0)$ , c'est-à-dire vérifie

$$f = g \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B) \Rightarrow f \sim g.$$

Cette construction est inspirée de la "réécriture d'objets" proposée par F. Cury [Cur91].

Nous commençons par définir la relation d'équivalence  $\sim$  sur  $\text{Obj}(\text{TERME}(\mathcal{C}_0))$ . Deux objets équivalents sont isomorphes, mais deux objets isomorphes ne sont pas nécessairement équivalents. Nous définissons d'abord une sous-précatégorie  $\text{Iso}(\mathcal{C}_0)$  de  $\text{TERME}(\mathcal{C}_0)$ . La précatégorie  $\text{Iso}(\mathcal{C}_0)$  a pour objets les objets de  $\text{TERME}(\mathcal{C}_0)$ , et pour flèches *certaines isomorphismes* de  $\text{TERME}(\mathcal{C}_0)$ . Ces isomorphismes vont correspondre aux équivalences entre objets de  $\text{TERME}(\mathcal{C}_0)$ .

**Définition 3.14** (Précatégorie  $\text{Iso}(\mathcal{C}_0)$ )

- L'ensemble des objets de  $\text{Iso}(\mathcal{C}_0)$  est l'ensemble des objets de  $\text{TERME}(\mathcal{C}_0)$ .
- L'ensemble des flèches de  $\text{Iso}(\mathcal{C}_0)$  est défini inductivement de la façon suivante.

- Pour tout objet  $A$  de  $\text{TERME}(\mathcal{C}_0)$ ,

$$\text{id}_A \in \text{Arr}(\text{Iso}(\mathcal{C}_0))(A, A).$$

- Soit  $f \in \text{Arr}(\text{Iso}(\mathcal{C}_0))(A, B)$  et  $g \in \text{Arr}(\text{Iso}(\mathcal{C}_0))(B, C)$ . Alors,

$$g \circ f \in \text{Arr}(\text{Iso}(\mathcal{C}_0))(A, C).$$

- Soit six objets  $A_1, B_1, C_1, A_2, B_2, C_2$  de  $\text{TERME}(\mathcal{C}_0)$ , et trois flèches de  $\text{Iso}(\mathcal{C}_0)$

$$\phi_A \in \text{Arr}(\text{Iso}(\mathcal{C}_0))(A_1, A_2),$$

$$\phi_B \in \text{Arr}(\text{Iso}(\mathcal{C}_0))(B_1, B_2),$$

$$\phi_C \in \text{Arr}(\text{Iso}(\mathcal{C}_0))(C_1, C_2).$$

Soit quatre flèches de  $\text{TERME}(\mathcal{C}_0)$

$$f_1 \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A_1, B_1),$$

$$g_1 \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A_1, C_1),$$

$$f_2 \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A_2, B_2),$$

$$g_2 \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A_2, C_2),$$

telles que

$$\phi_B \circ f_1 = f_2 \circ \phi_A \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A_1, B_2),$$

$$\phi_C \circ g_1 = g_2 \circ \phi_A \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A_1, C_2).$$

Il existe alors une flèche

$$\begin{aligned} & \text{up}(A_1, B_1, C_1, \text{push}(A_2, B_2, C_2, f_2, g_2), f_1, g_1, \\ & \quad \&_1(A_2, B_2, C_2, f_2, g_2) \circ \phi_B, \&_2(A_2, B_2, C_2, f_2, g_2) \circ \phi_C) \end{aligned}$$

de  $\text{push}(A_1, B_1, C_1, f_1, g_1)$  vers  $\text{push}(A_2, B_2, C_2, f_2, g_2)$ . Cette flèche, que l'on note

$$\text{iso}(f_1, g_1, f_2, g_2, \phi_A, \phi_B, \phi_C)$$

est, par définition, une flèche de la précatégorie  $\text{Iso}(\mathcal{C}_0)$ .

- Soit deux flèches  $f, g \in \text{Arr}(\text{Iso}(\mathcal{C}_0))(A, B)$ . Par définition, on pose

$$f = g \in \text{Arr}(\text{Iso}(\mathcal{C}_0))(A, B) \Leftrightarrow f = g \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B).$$

$\text{Iso}(\mathcal{C}_0)$  est évidemment une précatégorie. De plus, toutes les flèches de  $\text{Iso}(\mathcal{C}_0)$  sont des isomorphismes dans  $\text{TERME}(\mathcal{C}_0)$ , et également dans  $\text{Iso}(\mathcal{C}_0)$ . Pour montrer ce résultat, on définit une fonction  $\text{inv}$  qui associe à toute flèche de  $\text{Iso}(\mathcal{C}_0)$  une inverse dans  $\text{Iso}(\mathcal{C}_0)$ .

**Définition 3.15** (Fonction  $\text{inv} : \text{Arr}(\text{Iso}(\mathcal{C}_0))(A, B) \rightarrow \text{Arr}(\text{Iso}(\mathcal{C}_0))(B, A)$ )

La fonction  $\text{inv}$ , qui associe à toute flèche de  $\text{Iso}(\mathcal{C}_0)$  de  $A$  vers  $B$  une flèche de  $\text{Iso}(\mathcal{C}_0)$  de  $B$  vers  $A$ , est définie inductivement sur la structure des flèches de  $\text{Iso}(\mathcal{C}_0)$  de la façon suivante.

- $\text{inv}(\text{id}_A) = \text{id}_A$ ;
- $\text{inv}(g \circ f) = \text{inv}(f) \circ \text{inv}(g)$ ;
- $\text{inv}(\text{iso}(f_1, g_1, f_2, g_2, \phi_A, \phi_B, \phi_C))$   
 $= \text{iso}(f_2, g_2, f_1, g_1, \text{inv}(\phi_A), \text{inv}(\phi_B), \text{inv}(\phi_C)).$

On vérifie sans difficulté que, pour toute flèche  $f \in \text{Arr}(\text{Iso}(\mathcal{C}_0))(A, B)$ , on a bien

- $\text{inv}(f) \in \text{Arr}(\text{Iso}(\mathcal{C}_0))(B, A)$ ;
- $\text{inv}(f) \circ f = \text{id}_A \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, A)$ ;
- $f \circ \text{inv}(f) = \text{id}_B \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(B, B)$ .

Nous pouvons maintenant définir la relation d'équivalence sur les objets de  $\text{TERME}(\mathcal{C}_0)$ .

**Définition 3.16** (Relation d'équivalence  $\sim$  sur les objets de  $\text{TERME}(\mathcal{C}_0)$ )

Deux objets  $A$  et  $B$  de  $\text{TERME}(\mathcal{C}_0)$  sont équivalents si et seulement si il existe un isomorphisme de  $A$  vers  $B$  dans  $\text{Iso}(\mathcal{C}_0)$ .

*Notation.* Si  $A$  et  $B$  sont équivalents, on note  $A \sim B$ , ou, pour préciser l'isomorphisme  $\phi$  de  $A$  vers  $B$ ,  $\phi \vdash A \sim B$ . Intuitivement, on peut considérer la flèche  $\phi : A \rightarrow B$  comme la preuve que  $A \sim B$ .

**Lemme 3.11** *La relation  $\sim$  est une relation d'équivalence sur  $\text{Obj}(\text{TERME}(\mathcal{C}_0))$ .*

*Preuve.* La relation  $\sim$  est

- réflexive car  $\text{id}_A \vdash A \sim A$  ;
- symétrique car  $\phi \vdash A \sim B \Rightarrow \text{inv}(\phi) \vdash B \sim A$  ;
- transitive car  $\phi_1 \vdash A \sim B, \phi_2 \vdash B \sim C \Rightarrow \phi_2 \circ \phi_1 \vdash A \sim C$ .

□

Soit un objet  $A$  de  $\text{TERME}(\mathcal{C}_0)$ . On note  $[[A]]$  la classe d'équivalence de  $A$  modulo  $\sim$ . Autrement dit,

$$[[A]] = \{B \in \text{Obj}(\text{TERME}(\mathcal{C}_0)); A \sim B\}.$$

À chaque preuve de  $A \sim B$  est associé un isomorphisme  $\phi \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$ . La proposition suivante indique que l'isomorphisme  $\phi$  ne dépend pas de cette preuve.

**Proposition 3.3** *Si  $\phi_1 \vdash A \sim B$  et  $\phi_2 \vdash A \sim B$ , alors*

$$\phi_1 = \phi_2 \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B).$$

*Esquisse de la preuve.* On définit un système de réécriture sur les flèches de  $\text{Iso}(\mathcal{C}_0)$  de la façon suivante. (Les prémisses qui assurent que tous les termes sont bien formés sont, pour faciliter la lecture, omis.)

### Axiomes

- $\phi_3 \circ (\phi_2 \circ \phi_1) \longrightarrow (\phi_3 \circ \phi_2) \circ \phi_1$
- $\phi \circ \text{id}_A \longrightarrow \phi$
- $\text{id}_A \circ \phi \longrightarrow \phi$
- $\text{iso}(f_2, g_2, f_3, g_3, \psi_A, \psi_B, \psi_C) \circ \text{iso}(f_1, g_1, f_2, g_2, \phi_A, \phi_B, \phi_C) \longrightarrow \text{iso}(f_1, g_1, f_3, g_3, \psi_A \circ \phi_A, \psi_B \circ \phi_B, \psi_C \circ \phi_C)$

### Règles de contexte

- $\phi \longrightarrow \phi', \psi \longrightarrow \psi' \Rightarrow \psi \circ \phi \longrightarrow \psi' \circ \phi'$
- $\phi_A \longrightarrow \phi'_A, \phi_B \longrightarrow \phi'_B, \phi_C \longrightarrow \phi'_C \Rightarrow \text{iso}(f_1, g_1, f_2, g_2, \phi_A, \phi_B, \phi_C) \longrightarrow \text{iso}(f_1, g_1, f_2, g_2, \phi'_A, \phi'_B, \phi'_C)$

On montre alors successivement les points suivants.

1. Si  $\phi \vdash A \sim B$  et  $\phi \longrightarrow^* \phi'$ , alors  $\phi' \in \text{Arr}(\text{Iso}(\mathcal{C}_0))(A, B)$ .
2. Le système de réécriture est cohérent. Si  $\phi \vdash A \sim B$  et  $\phi \longrightarrow^* \phi'$ , alors  $\phi = \phi' \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$ .

3. Le système de réécriture est confluent.
4. Le système de réécriture termine.
5. D'après 3 et 4, toute flèche  $\phi$  de  $\text{Arr}(\text{Iso}(\mathcal{C}_0))(A, B)$  a donc une forme normale notée  $\mathcal{V}(\phi)$ . D'après 2, on a  $\phi = \mathcal{V}(\phi) \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$ .
6. On montre, par induction sur la structure de  $A$  et  $B$ , que

$$\phi_1 \vdash A \sim B, \phi_2 \vdash A \sim B \Rightarrow \mathcal{V}(\phi_1) \equiv \mathcal{V}(\phi_2).$$

7. On déduit immédiatement de 5 et 6 que

$$\phi_1 \vdash A \sim B, \phi_2 \vdash A \sim B \Rightarrow \phi_1 = \phi_2 \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B).$$

□

**Remarque 3.4** Si  $\phi \vdash A \sim B$ , l'isomorphisme  $\phi$  correspond à une preuve que  $A \sim B$  et  $\mathcal{V}(\phi)$  correspond à la normalisation de cette preuve.

*Notation.* Si  $\phi \vdash A \sim B$ , la flèche  $\phi : A \rightarrow B$  de  $\text{Iso}(\mathcal{C}_0)$  est notée  $\langle A, B \rangle : A \rightarrow B$ . Cette notation est justifiée par la proposition 3.3.

À partir de cette relation d'équivalence sur les objets de  $\text{TERME}(\mathcal{C}_0)$ , nous définissons une relation d'équivalence sur les flèches de  $\text{TERME}(\mathcal{C}_0)$ .

**Définition 3.17** (Relation d'équivalence  $\sim$  sur les flèches de  $\text{TERME}(\mathcal{C}_0)$ )  
Soit deux flèches  $f \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$  et  $f' \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A', B')$ . Les flèches  $f$  et  $f'$  sont équivalentes si et seulement si

$$A \sim A', B \sim B' \text{ et } f' \circ \langle A, A' \rangle = \langle B, B' \rangle \circ f \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B').$$

On note  $f \sim f'$ .

**Lemme 3.12**

*La relation  $\sim$  sur les flèches de  $\text{TERME}(\mathcal{C}_0)$  est une relation d'équivalence.*

*Preuve.* La relation  $\sim$  est

- réflexive car  $\langle A, A \rangle = \text{id}_A \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, A)$  ;
- symétrique car  $\text{inv}\langle A, B \rangle = \langle B, A \rangle \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(B, A)$  ;
- transitive car  $\langle B, C \rangle \circ \langle A, B \rangle = \langle A, C \rangle \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, C)$ .

□

Soit  $f \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$ . On note  $[[f]]$  la classe d'équivalence de  $f$  modulo  $\sim$ . Autrement dit,

$$[[f]] = \{f' \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A', B') ; f \sim f'\}.$$

**Lemme 3.13**  $\forall f, f' \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$ ,

$$f = f' \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B) \Rightarrow f \sim f'.$$

*Preuve.* On a  $f' \circ \langle A, A \rangle = \langle B, B \rangle \circ f \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$  car

$$\begin{aligned}\langle A, A \rangle &= \text{id}_A \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, A) \\ \langle B, B \rangle &= \text{id}_B \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(B, B).\end{aligned}$$

□

Nous définissons maintenant la catégorie  $\mathcal{L}(\mathcal{C}_0)$ . Intuitivement,  $\mathcal{L}(\mathcal{C}_0)$  a pour objets les classes d'équivalence modulo  $\sim$  d'objets de  $\text{TERME}(\mathcal{C}_0)$  et pour flèches les classes d'équivalence modulo  $\sim$  de flèches de  $\text{TERME}(\mathcal{C}_0)$ .

**Définition 3.18** (Catégorie  $\mathcal{L}(\mathcal{C}_0)$ )

- *Objets.* Soit un objet  $A$  de  $\text{TERME}(\mathcal{C}_0)$ . Alors,  $[[A]]$  est un objet de  $\mathcal{L}(\mathcal{C}_0)$ .
- *Flèches.* Soit  $f \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$ . Alors,  $[[f]] : [[A]] \rightarrow [[B]]$  est une flèche de  $\mathcal{L}(\mathcal{C}_0)$ .

Cette définition est correcte car si

$$f \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B), f' \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A', B') \text{ et } f \sim f',$$

alors, on a  $A \sim A'$  et  $B \sim B'$ .

- *Composition.* Soit  $f \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$  et  $g \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(C, D)$ , avec  $B \sim C$ . On pose  $[[g]] \circ [[f]] = [[g \circ \langle B, C \rangle \circ f]]$ .

Cette définition est indépendante des représentants  $f$  et  $g$  choisis. En effet, soit  $f' \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A', B')$  et  $g' \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(C', D')$  tels que  $f \sim f'$  et  $g \sim g'$ . On a les égalités suivantes dans  $\text{Arr}(\text{TERME}(\mathcal{C}_0))(A, D')$ :

$$\begin{aligned}\langle D, D' \rangle \circ g \circ \langle B, C \rangle \circ f & \\ &= g' \circ \langle C, C' \rangle \circ \langle B, C \rangle \circ f \quad (\text{car } g \sim g') \\ &= g' \circ \langle B, C' \rangle \circ f \quad (\text{car } \langle C, C' \rangle \circ \langle B, C \rangle = \langle B, C' \rangle) \\ &= g' \circ \langle B', C' \rangle \circ \langle B, B' \rangle \circ f \quad (\text{car } \langle B', C' \rangle \circ \langle B, B' \rangle = \langle B, C' \rangle) \\ &= g' \circ \langle B', C' \rangle \circ f' \circ \langle A, A' \rangle \quad (\text{car } f \sim f')\end{aligned}$$

Par conséquent,  $g \circ \langle B, C \rangle \circ f \sim g' \circ \langle B', C' \rangle \circ f'$ , et donc

$$[[g \circ \langle B, C \rangle \circ f]] = [[g' \circ \langle B', C' \rangle \circ f']].$$

La composition est évidemment associative.

- *Identités.* Soit  $[[A]]$  un objet de  $\mathcal{L}(\mathcal{C}_0)$ . Alors  $[[\text{id}_A]] : [[A]] \rightarrow [[A]]$  est une flèche identité.

**Remarque 3.5** Si  $A \sim B$ , alors  $\text{id}_A \sim \langle A, B \rangle$ . Par conséquent, les flèches  $\langle A, B \rangle$  de  $\text{Iso}(\mathcal{C}_0)$  correspondent à des identités  $[[\langle A, B \rangle]] = [[\text{id}_A]] = [[\text{id}_B]]$  dans  $\mathcal{L}(\mathcal{C}_0)$ .

On a un préfoncteur  $[[ - ]]$  :  $\text{TERME}(\mathcal{C}_0) \rightarrow \mathcal{L}(\mathcal{C}_0)$  défini de la façon suivante :

$$\begin{array}{lcl} [[ - ]] & : & \text{TERME}(\mathcal{C}_0) \rightarrow \mathcal{L}(\mathcal{C}_0) \\ & & A \mapsto [[A]] \\ & & f \mapsto [[f]]. \end{array}$$

C'est bien un préfoncteur d'après le lemme 3.13. Ce préfoncteur est associé à un foncteur  $[[ - ]]$  :  $\text{Terme}(\mathcal{C}_0) \rightarrow \mathcal{L}(\mathcal{C}_0)$ , défini par :

$$\begin{array}{lcl} [[ - ]] & : & \text{Terme}(\mathcal{C}_0) \rightarrow \mathcal{L}(\mathcal{C}_0) \\ & & A \mapsto [[A]] \\ & & [f] \mapsto [[f]]. \end{array}$$

**Proposition 3.4** *La catégorie  $\mathcal{L}(\mathcal{C}_0)$  est une extension conservatrice de  $\mathcal{C}_0$ . Autrement dit, le foncteur  $[[ - ]]$   $\circ J : \mathcal{C}_0 \rightarrow \mathcal{L}(\mathcal{C}_0)$  est plein et fidèle.*

*Preuve.* On utilise le fait que le foncteur  $J : \mathcal{C}_0 \rightarrow \text{Terme}(\mathcal{C}_0)$  est plein et fidèle (théorème 3.10). On remarque également que la classe d'équivalence d'un objet de  $\mathcal{C}_0$  ne contient que cet objet. Autrement dit, pour tout objet  $A$  de  $\mathcal{C}_0$ ,  $[[J(A)]] = \{J(A)\}$ . En effet, dans la définition 3.14, on n'introduit pas d'équivalence entre objets de  $\mathcal{C}_0$  autres que les identités.

*Le foncteur  $[[ - ]]$   $\circ J$  est plein.*

Soit deux objets  $A$  et  $B$  de  $\mathcal{C}_0$ , et une flèche  $h : [[J(A)]] \rightarrow [[J(B)]]$  de  $\mathcal{L}(\mathcal{C}_0)$ . Par définition de  $[[ - ]]$ , il existe une flèche  $g$  de  $\text{TERME}(\mathcal{C}_0)$  telle que  $[[g]] = h$ , qui a pour domaine  $J(A)$  et pour codomaine  $J(B)$ . Comme le foncteur  $J$  est plein, il existe donc une flèche  $f : A \rightarrow B$  de  $\mathcal{C}_0$  telle que  $J(f) = g$ , ce qui implique que  $[[J(f)]] = [[g]] = h$ .

*Le foncteur  $[[ - ]]$   $\circ J$  est fidèle.*

Soit deux flèches  $f, g : A \rightarrow B$  de  $\mathcal{C}_0$  telles que  $[[J(f)]] = [[J(g)]]$ .

$$\begin{aligned} [[J(f)]] &= [[J(g)]] \\ \Rightarrow J(f) &\sim J(g) \\ \Rightarrow \langle J(B), J(B) \rangle \circ J(f) &= J(g) \circ \langle J(A), J(A) \rangle \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B) \\ \Rightarrow J(f) &= J(g) \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B) \\ \Rightarrow f &= g \qquad \qquad \qquad (\text{car } J \text{ est fidèle}) \end{aligned}$$

□

**Théorème 3.12** *La catégorie  $\mathcal{L}(\mathcal{C}_0)$  est finiment cocomplète. Plus précisément,*

1. *l'objet  $[[\emptyset]]$  est un objet initial choisi dans  $\mathcal{L}(\mathcal{C}_0)$  ;*
2. *si  $[[A]]$ ,  $[[B]]$  et  $[[C]]$  sont des objets de  $\mathcal{L}(\mathcal{C}_0)$ ,  $[[f]] : [[A]] \rightarrow [[B]]$  et  $[[g]] : [[A]] \rightarrow [[C]]$  sont des flèches de  $\mathcal{L}(\mathcal{C}_0)$ , alors le triplet*

$$([[push(A, B, C, f, g)]], [[\&_1(A, B, C, f, g)]], [[\&_2(A, B, C, f, g)]])$$

*est un choix de somme amalgamée de  $[[B]]$  et  $[[C]]$  par rapport à  $[[f]]$  et  $[[g]]$  dans  $\mathcal{L}(\mathcal{C}_0)$ .*

On vérifie facilement que le choix de somme amalgamée ne dépend pas des représentants  $A$ ,  $B$ ,  $C$ ,  $f$  et  $g$  respectivement choisis pour  $[[A]]$ ,  $[[B]]$ ,  $[[C]]$ ,  $[[f]]$  et  $[[g]]$ . Le triplet

$$([[push(A, B, C, f, g)]], [[\&_1(A, B, C, f, g)]], [[\&_2(A, B, C, f, g)]])$$

est donc bien une somme amalgamée choisie.

On montre enfin que la catégorie  $\mathcal{L}(\mathcal{C}_0)$  est libre.

**Théorème 3.13** ( $\mathcal{L}(\mathcal{C}_0)$  est une extension libre) *La catégorie  $\mathcal{L}(\mathcal{C}_0)$  est librement engendrée par objet initial choisi et sommes amalgamées choisies. Autrement dit : Soit  $\mathcal{E}$  une catégorie avec un objet initial choisi et des sommes amalgamées choisies. Soit  $F : \mathcal{C}_0 \rightarrow \mathcal{E}$  un foncteur. Alors il existe un unique foncteur*

$$H : \mathcal{L}(\mathcal{C}_0) \rightarrow \mathcal{E}$$

qui conserve fortement l'objet initial choisi, les sommes amalgamées choisies, et tel que

$$H \circ [[-]] \circ J = F.$$

*Esquisse de la preuve.* On considère le foncteur  $G : \text{Terme}(\mathcal{C}_0) \rightarrow \mathcal{E}$  associé au préfoncteur  $G : \text{TERME}(\mathcal{C}_0) \rightarrow \mathcal{E}$  du théorème 3.9.

*Existence.* On montre que

$$\phi \vdash A \sim A' \Rightarrow G(A) = G(A') \text{ et } G(\phi) = id_{G(A)},$$

par induction sur la longueur de la preuve que  $\phi \vdash A \sim A'$ . On en déduit que

$$f \sim f' \Rightarrow G(f) = G(f').$$

Par conséquent, il existe un unique foncteur  $H : \mathcal{L}(\mathcal{C}_0) \rightarrow \mathcal{E}$  tel que  $H \circ [[-]] = G$ .

On a donc  $H \circ [[-]] \circ J = G \circ J = F$ .

On vérifie que le foncteur  $H$  conserve fortement l'objet initial choisi et les sommes amalgamées choisies.

*Unicité.* Soit  $H' : \mathcal{L}(\mathcal{C}_0) \rightarrow \mathcal{E}$  un foncteur qui conserve fortement l'objet initial choisi, les sommes amalgamées choisies, et tel que

$$H' \circ [[-]] \circ J = F.$$

D'après le théorème 3.9, on a donc  $H \circ [[-]] = H' \circ [[-]]$ . Comme  $[[-]]$  est surjectif sur les flèches et les objets, on en déduit que  $H = H'$ .

□

### 3.9 Conclusion

Nous avons proposé la construction d'une précatégorie  $\text{TERME}(\mathcal{C}_0)$  à objet pré-initial choisi et pré-sommes amalgamées choisies pour décrire des constructions de colimites sur une petite catégorie de base  $\mathcal{C}_0$ . Cette précatégorie de termes constitue une *syntaxe* pour les constructions modulaires.

Nous avons montré que  $\text{TERME}(\mathcal{C}_0)$  est une extension conservatrice de  $\mathcal{C}_0$  (théorème 3.10), c'est-à-dire que le préfoncteur d'inclusion  $J$  de  $\mathcal{C}_0$  dans  $\text{TERME}(\mathcal{C}_0)$  est plein et fidèle. Cela signifie d'une part que  $\text{TERME}(\mathcal{C}_0)$  ne contient pas de flèches entre des objets de  $\mathcal{C}_0$  autres que celles contenues dans  $\mathcal{C}_0$ , et d'autre part que  $\text{TERME}(\mathcal{C}_0)$  ne contient pas d'égalités entre flèches de  $\mathcal{C}_0$  autres que celles qui proviennent de  $\mathcal{C}_0$ .

Nous avons montré que la précatégorie  $\text{TERME}(\mathcal{C}_0)$  est librement engendrée par objet pré-initial choisi et pré-sommes amalgamées choisies (théorème 3.9).

En passant au quotient, on obtient une catégorie finiment cocomplète  $\text{Terme}(\mathcal{C}_0)$ , qui est également une extension conservatrice de  $\mathcal{C}_0$  (théorèmes 3.10 et 3.11).

Enfin, nous avons montré comment construire, à partir de  $\text{TERME}(\mathcal{C}_0)$ , une catégorie  $\mathcal{L}(\mathcal{C}_0)$  librement engendrée par objet initial choisi et sommes amalgamées choisies sur  $\mathcal{C}_0$  (théorème 3.13).



# Chapitre 4

## Sémantique : des termes aux diagrammes

Le but de ce chapitre est d'associer à toute spécification modulaire un *diagramme*, et à tout morphisme de spécifications modulaires un *morphisme de diagrammes*. On appelle  $\mathcal{C}_0$  la catégorie des spécifications et morphismes de spécifications de base, et on décrit cette association par un préfoncteur

$$\mathcal{D} : \text{TERME}(\mathcal{C}_0) \rightarrow \text{DIAGR}(\mathcal{C}_0).$$

Intuitivement, étant donné un objet  $A$  de  $\text{TERME}(\mathcal{C}_0)$  qui représente une spécification modulaire,  $\mathcal{D}(A)$  est un diagramme dont la colimite est isomorphe à cette spécification. Le foncteur  $\mathcal{D}$  décrit une *interprétation* des termes par des diagrammes, il s'agit donc d'une *sémantique* pour les spécifications algébriques modulaires, par opposition aux termes qui constituent la *syntaxe*.

Après avoir défini le préfoncteur  $\mathcal{D} : \text{TERME}(\mathcal{C}_0) \rightarrow \text{DIAGR}(\mathcal{C}_0)$ , nous détaillons les calculs de diagrammes associés aux spécifications modulaires  $A_2$ ,  $A_2'$ ,  $A_3$  et  $A_4$  des anneaux présentées dans le chapitre 1.

Dans la dernière partie, nous montrons que les précatégories  $\text{TERME}(\mathcal{C}_0)$  et  $\text{DIAGR}(\mathcal{C}_0)$  sont équivalentes, ce qui implique par passage au quotient que les catégories associées  $\text{Terme}(\mathcal{C}_0)$  et  $\text{diagr}(\mathcal{C}_0)$  sont équivalentes. Intuitivement, cela signifie que ces deux catégories ont “la même structure”, “aux isomorphismes entre objets près”. Nous en déduisons que deux objets de  $\text{Terme}(\mathcal{C}_0)$  sont isomorphes si et seulement si leurs diagrammes associés sont isomorphes dans la catégorie  $\text{diagr}(\mathcal{C}_0)$ . Enfin, nous montrons que le calcul de diagrammes est correct, c'est-à-dire qu'une spécification est isomorphe à la colimite du diagramme qui la représente.

### 4.1 Sémantique

#### 4.1.1 Précatégorie $\text{DIAGR}(\mathcal{C}_0)$

Dans tout ce chapitre,  $\mathcal{C}_0$  est une petite catégorie. Rappelons que la catégorie  $\text{DIAGR}(\mathcal{C}_0)$  peut être considérée comme une *précatégorie* de la façon suivante (cf. chapitre 3, page 123) :

- l'identité syntaxique dans la pré-catégorie  $\text{DIAGR}(\mathcal{C}_0)$  est l'égalité dans la catégorie  $\text{DIAGR}(\mathcal{C}_0)$  ;
- l'équivalence dans la pré-catégorie  $\text{DIAGR}(\mathcal{C}_0)$  est définie par la congruence  $\approx$  sur les flèches de la catégorie  $\text{DIAGR}(\mathcal{C}_0)$ .

Par conséquent, la catégorie associée à la pré-catégorie  $\text{DIAGR}(\mathcal{C}_0)$  est  $\text{diagr}(\mathcal{C}_0)$ .

La pré-catégorie  $\text{DIAGR}(\mathcal{C}_0)$  est finiment pré-cocomplète. Plus précisément, la pré-catégorie  $\text{DIAGR}(\mathcal{C}_0)$  a un objet pré-initial choisi et des pré-sommes amalgamées choisies.

### Objet pré-initial

Rappelons que le diagramme vide, noté  $\bigcirc$ , est le diagramme dont le graphe sous-jacent ne contient aucun nœud et aucun arc.

**Lemme 4.1** *Le diagramme vide est pré-initial dans la pré-catégorie  $\text{DIAGR}(\mathcal{C}_0)$ . Autrement dit,*

1. pour tout diagramme  $\bar{\alpha}$  de  $\text{DIAGR}(\mathcal{C}_0)$ , il existe une flèche

$$\bar{j}_{\bar{\alpha}} : \bigcirc \rightarrow \bar{\alpha}$$

dans  $\text{DIAGR}(\mathcal{C}_0)$  ;

2. pour toute flèche  $\bar{\sigma} : \bigcirc \rightarrow \bar{\alpha}$ , on a  $\bar{j}_{\bar{\alpha}} \approx \bar{\sigma}$ .

Remarquons qu'en réalité le diagramme vide est *initial* dans  $\text{DIAGR}(\mathcal{C}_0)$ . En effet, pour toute flèche  $\bar{\sigma} : \bigcirc \rightarrow \bar{\alpha}$ , on a  $\bar{j}_{\bar{\alpha}} = \bar{\sigma}$ .

### Pré-sommes amalgamées

Soit trois diagrammes  $\bar{\alpha}$ ,  $\bar{\beta}$  et  $\bar{\gamma}$ , et deux morphismes de diagrammes  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta}$  et  $\bar{\tau} : \bar{\alpha} \rightarrow \bar{\gamma}$  de  $\text{DIAGR}(\mathcal{C}_0)$ . On considère le diagramme de somme amalgamée

$$\bar{\bar{\Pi}} = (\Pi^\Phi, \Pi : \Pi^\Phi \rightarrow \text{DIAGR}(\mathcal{C}_0))$$

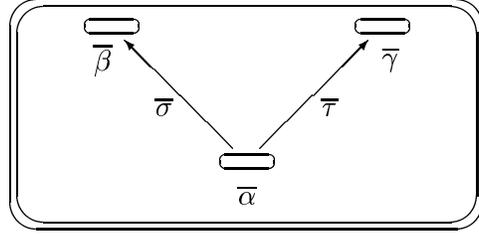
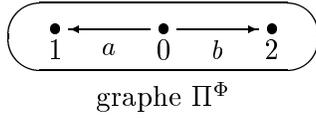
où

$$\begin{aligned} \Pi(0) &= \bar{\alpha}; \\ \Pi(1) &= \bar{\beta}; \\ \Pi(2) &= \bar{\gamma}; \\ \Pi(a) &= \bar{\sigma}; \\ \Pi(b) &= \bar{\tau}. \end{aligned}$$

Ce diagramme sera noté

$$\bar{\bar{\Pi}} = \text{PushDiagr}(\bar{\alpha}, \bar{\beta}, \bar{\gamma}, \bar{\sigma}, \bar{\tau}).$$

Le diagramme  $\bar{\bar{\Pi}}$  est un objet de  $\text{DIAGR}^2(\mathcal{C}_0)$ , construit sur le graphe  $\Pi^\Phi$ .



En aplatissant le diagramme  $\overline{\overline{\Pi}}$ , on obtient un diagramme  $\overline{\delta}$  de  $\text{DIAGR}(\mathcal{C}_0)$  (cf. définition 2.29, page 88) :

$$\overline{\delta} = \text{Apl}_{\mathcal{C}_0}(\overline{\overline{\Pi}}).$$

On a également deux flèches  $\overline{J}_1 : \overline{\beta} \rightarrow \overline{\delta}$  et  $\overline{J}_2 : \overline{\gamma} \rightarrow \overline{\delta}$  dans  $\text{DIAGR}(\mathcal{C}_0)$  (cf. définition 2.30, page 89) qui seront notées

$$\begin{aligned} \overline{J}_1 &= \overline{J}_1(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\sigma}, \overline{\tau}) : \overline{\beta} \rightarrow \overline{\delta} \\ \overline{J}_2 &= \overline{J}_2(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\sigma}, \overline{\tau}) : \overline{\gamma} \rightarrow \overline{\delta}. \end{aligned}$$

**Lemme 4.2** *Le triplet*

$$(\text{Apl}_{\mathcal{C}_0}(\text{PushDiagr}(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\sigma}, \overline{\tau})), \overline{J}_1(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\sigma}, \overline{\tau}), \overline{J}_2(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\sigma}, \overline{\tau}))$$

*est une pré-somme amalgamée de  $\overline{\beta}$  et  $\overline{\gamma}$  par rapport aux flèches  $\overline{\sigma}$  et  $\overline{\tau}$ . Cela signifie qu'on a les propriétés suivantes.*

1.  $\overline{J}_1(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\sigma}, \overline{\tau}) \circ \overline{\sigma} \approx \overline{J}_2(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\sigma}, \overline{\tau}) \circ \overline{\tau}$ .
2. Soit un diagramme  $\overline{\delta'}$  et deux flèches

$$\begin{aligned} \overline{K}_1 &: \overline{\beta} \rightarrow \overline{\delta'} \\ \overline{K}_2 &: \overline{\gamma} \rightarrow \overline{\delta'} \end{aligned}$$

de  $\text{DIAGR}(\mathcal{C}_0)$  telles que

$$\overline{K}_1 \circ \overline{\sigma} \approx \overline{K}_2 \circ \overline{\tau}.$$

Alors,

- (a) en posant  $\overline{\delta} = \text{Apl}_{\mathcal{C}_0}(\text{PushDiagr}(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\sigma}, \overline{\tau}))$ , il existe une flèche

$$\overline{U}(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\delta'}, \overline{\sigma}, \overline{\tau}, \overline{K}_1, \overline{K}_2) : \overline{\delta} \rightarrow \overline{\delta'}$$

de  $\text{DIAGR}(\mathcal{C}_0)$  telle que

$$\begin{cases} \overline{U}(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\delta'}, \overline{\sigma}, \overline{\tau}, \overline{K}_1, \overline{K}_2) \circ \overline{J}_1(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\sigma}, \overline{\tau}) \approx \overline{K}_1 \\ \overline{U}(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\delta'}, \overline{\sigma}, \overline{\tau}, \overline{K}_1, \overline{K}_2) \circ \overline{J}_2(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\sigma}, \overline{\tau}) \approx \overline{K}_2; \end{cases}$$

- (b) pour toute flèche  $\overline{U'} : \overline{\delta} \rightarrow \overline{\delta'}$  de  $\text{DIAGR}(\mathcal{C}_0)$  telle que

$$\begin{cases} \overline{U'} \circ \overline{J}_1(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\sigma}, \overline{\tau}) \approx \overline{K}_1 \\ \overline{U'} \circ \overline{J}_2(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\sigma}, \overline{\tau}) \approx \overline{K}_2 \end{cases}$$

on a

$$\overline{U}(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\delta'}, \overline{\sigma}, \overline{\tau}, \overline{K}_1, \overline{K}_2) \approx \overline{U'}.$$

*Preuve.*

1. Il s'agit d'une reformulation du lemme 2.17, page 104.
2. Il s'agit d'une reformulation du lemme 2.19, page 105. Remarquons que la preuve de ce lemme est constructive, c'est-à-dire construit explicitement un choix de flèche

$$\overline{U}(\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\delta'}, \overline{\sigma}, \overline{\tau}, \overline{K_1}, \overline{K_2}) : \overline{\delta} \rightarrow \overline{\delta'}.$$

□

#### 4.1.2 Préfoncteur $\mathcal{D} : \text{TERME}(\mathcal{C}_0) \rightarrow \text{DIAGR}(\mathcal{C}_0)$

Rappelons qu'on a un préfoncteur  $I_{\mathcal{C}_0} : \mathcal{C}_0 \rightarrow \text{DIAGR}(\mathcal{C}_0)$ . D'après les lemmes 4.1 et 4.2, on peut appliquer le théorème 3.9 page 139. Il existe donc un unique préfoncteur

$$\mathcal{D} : \text{TERME}(\mathcal{C}_0) \rightarrow \text{DIAGR}(\mathcal{C}_0)$$

tel que

1.  $\forall A \in \text{Obj}(\mathcal{C}_0), \mathcal{D}(A) \equiv I_{\mathcal{C}_0}(A)$  ;
2.  $\mathcal{D}(\emptyset) \equiv \bigcirc$  ;
3.  $\mathcal{D}(\text{push}(A, B, C, f, g)) \equiv \text{Apl}_{\mathcal{C}_0}(\text{PushDiagr}(\mathcal{D}(A), \mathcal{D}(B), \mathcal{D}(C), \mathcal{D}(f), \mathcal{D}(g)))$  ;
4.  $\forall f \in \text{Arr}(\mathcal{C}_0)(A, B), \mathcal{D}(f) \equiv I_{\mathcal{C}_0}(f)$  ;
5.  $\mathcal{D}(g \circ f) \equiv \mathcal{D}(g) \circ \mathcal{D}(f)$  ;
6.  $\mathcal{D}(\text{id}_A) \equiv \overline{id}_{\mathcal{D}(A)}$  ;
7.  $\mathcal{D}(j_A) \equiv \overline{j}_{\mathcal{D}(A)}$  ;
8.  $\mathcal{D}(\&_1(A, B, C, f, g)) \equiv \overline{J}_1(\mathcal{D}(A), \mathcal{D}(B), \mathcal{D}(C), \mathcal{D}(f), \mathcal{D}(g))$  ;
9.  $\mathcal{D}(\&_2(A, B, C, f, g)) \equiv \overline{J}_2(\mathcal{D}(A), \mathcal{D}(B), \mathcal{D}(C), \mathcal{D}(f), \mathcal{D}(g))$  ;
10.  $\mathcal{D}(\text{up}(A, B, C, D, f, g, f', g')) \equiv \overline{U}(\mathcal{D}(A), \mathcal{D}(B), \mathcal{D}(C), \mathcal{D}(D), \mathcal{D}(f), \mathcal{D}(g), \mathcal{D}(f'), \mathcal{D}(g'))$ .

Ces règles donnent une procédure de calcul du diagramme ou morphisme de diagrammes associé à un terme. Remarquons que nous avons remplacé la condition  $\mathcal{D} \circ J \equiv I_{\mathcal{C}_0}$  du théorème 3.9 par les points 1 et 4, qui sont évidemment équivalents.

Enfin, au préfoncteur  $\mathcal{D} : \text{TERME}(\mathcal{C}_0) \rightarrow \text{DIAGR}(\mathcal{C}_0)$  est évidemment associé à un foncteur

$$\mathcal{D} : \text{Terme}(\mathcal{C}_0) \rightarrow \text{diagr}(\mathcal{C}_0).$$

## 4.2 Exemple des anneaux

Dans cette section, nous calculons les diagrammes associés aux spécifications modulaires des anneaux  $A_2$ ,  $A'_2$ ,  $A_3$  et  $A_4$  présentées dans le chapitre 1. Pour chaque spécification modulaire  $A$ , nous calculons donc  $\mathcal{D}(A)$ , en utilisant les règles de calcul du préfoncteur

$$\mathcal{D} : \text{TERME}(\mathcal{C}_0) \rightarrow \text{DIAGR}(\mathcal{C}_0)$$

décrites dans le paragraphe 4.1.2.

### 4.2.1 Diagramme associé à la spécification $A_2$

Dans le chapitre 1, nous avons défini une spécification modulaire des anneaux  $A_2$  de la façon suivante (cf. page 45). Nous commençons par construire une spécification  $MD$  en combinant la spécification  $M$  des monoïdes avec la spécification  $D$  qui contient deux opérateurs distributifs, et en partageant la spécification de l'opérateur binaire multiplicatif. Puis on obtient une spécification des anneaux en combinant  $MD$  avec la spécification  $G$  des groupes, et en partageant l'opérateur additif.

$$\begin{aligned} MD &\equiv \text{push}(B, M, D, b, m_*) \\ A_2 &\equiv \text{push}(B, MD, G, \&_2(MD) \circ m_+, m \circ b) \end{aligned}$$

Nous calculons d'abord le diagramme associé à  $MD$ .

$$\begin{aligned} \mathcal{D}(MD) &\equiv \mathcal{D}(\text{push}(B, M, D, b, m_*)) \\ &\equiv \text{Apl}_{\mathcal{C}_0}(\text{PushDiagr}(\mathcal{D}(B), \mathcal{D}(M), \mathcal{D}(D), \mathcal{D}(b), \mathcal{D}(m_*))) \end{aligned}$$

$$\begin{aligned} \mathcal{D}(B) &\equiv I_{\mathcal{C}_0}(B) \equiv \boxed{\bullet B} \\ \mathcal{D}(M) &\equiv I_{\mathcal{C}_0}(M) \equiv \boxed{\bullet M} \\ \mathcal{D}(D) &\equiv I_{\mathcal{C}_0}(D) \equiv \boxed{\bullet D} \\ \mathcal{D}(b) &\equiv I_{\mathcal{C}_0}(b) \equiv \boxed{B \bullet} \xrightarrow{b} \boxed{\bullet M} \\ \mathcal{D}(m_*) &\equiv I_{\mathcal{C}_0}(m_*) \equiv \boxed{B \bullet} \xrightarrow{m_*} \boxed{\bullet D} \end{aligned}$$

$$\mathcal{D}(MD) \equiv \text{Apl}_{\mathcal{C}_0} \left( \begin{array}{c} \boxed{\begin{array}{c} \bullet M \\ \nearrow b \\ B \bullet \\ \searrow m_* \\ \bullet D \end{array}} \end{array} \right) \equiv \begin{array}{c} \bullet M \\ \nearrow b \\ B \bullet \\ \searrow m_* \\ \bullet D \end{array}$$

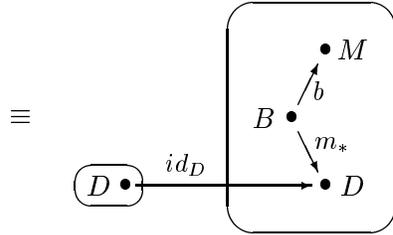
Nous calculons maintenant le diagramme associé à  $A_2$ .

$$\begin{aligned} \mathcal{D}(A_2) &\equiv \mathcal{D}(\text{push}(B, MD, G, \&_2(MD) \circ m_+, m \circ b)) \\ &\equiv \text{Apl}_{\mathcal{C}_0}(\text{PushDiagr}(\mathcal{D}(B), \mathcal{D}(MD), \mathcal{D}(G), \mathcal{D}(\&_2(MD) \circ m_+), \mathcal{D}(m \circ b))) \end{aligned}$$

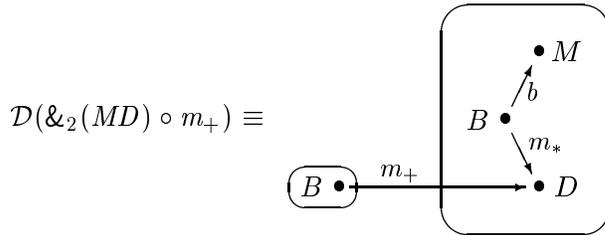
$$\mathcal{D}(G) \equiv I_{C_0}(G) \equiv \langle \bullet G \rangle$$

$$\mathcal{D}(\&_2(MD) \circ m_+) \equiv \mathcal{D}(\&_2(MD)) \circ \mathcal{D}(m_+)$$

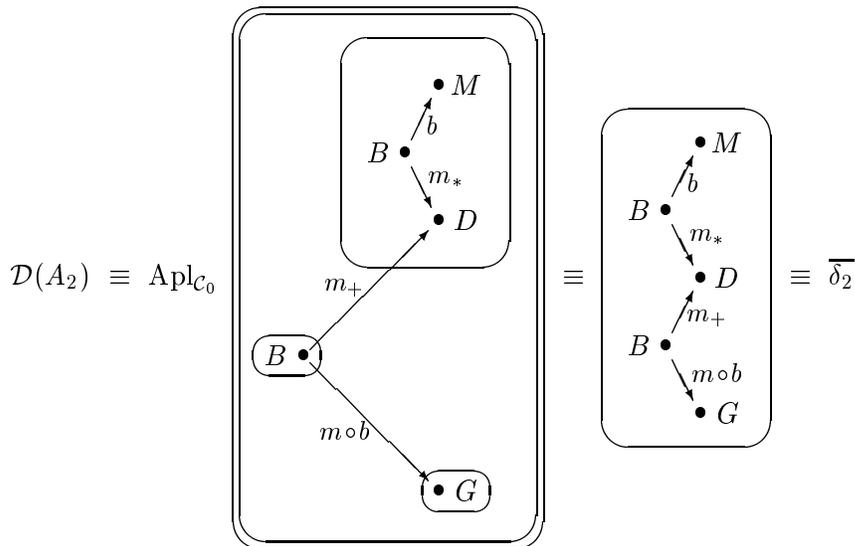
$$\mathcal{D}(\&_2(MD)) \equiv \overline{J}_2(\mathcal{D}(B), \mathcal{D}(M), \mathcal{D}(D), \mathcal{D}(b), \mathcal{D}(m_*))$$



$$\mathcal{D}(m_+) \equiv \langle B \bullet \xrightarrow{m_+} \bullet D \rangle$$



$$\mathcal{D}(m \circ b) \equiv \langle B \bullet \xrightarrow{m \circ b} \bullet G \rangle$$



Finalement, le diagramme associé à la spécification  $A_2$  est le diagramme  $\overline{\delta}_2$  que nous avons donné page 47.

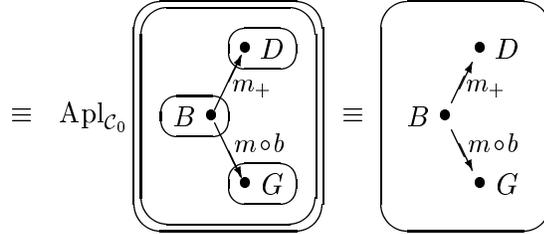
### 4.2.2 Diagramme associé à la spécification $A'_2$

Nous calculons maintenant le diagramme associé à la spécification  $A'_2$  des anneaux. La spécification  $A'_2$  a été construite en commençant par combiner les spécifications  $D$  et  $G$ , puis en ajoutant  $M$  (cf. page 46).

$$DG \equiv \text{push}(B, D, G, m_+, m \circ b)$$

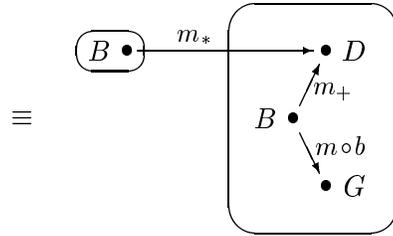
$$A'_2 \equiv \text{push}(B, M, DG, b, \&_1(DG) \circ m_*)$$

$$\mathcal{D}(DG) \equiv \text{Apl}_{\mathcal{C}_0}(\text{PushDiagr}(\mathcal{D}(B), \mathcal{D}(D), \mathcal{D}(G), \mathcal{D}(m_+), \mathcal{D}(m) \circ \mathcal{D}(b)))$$

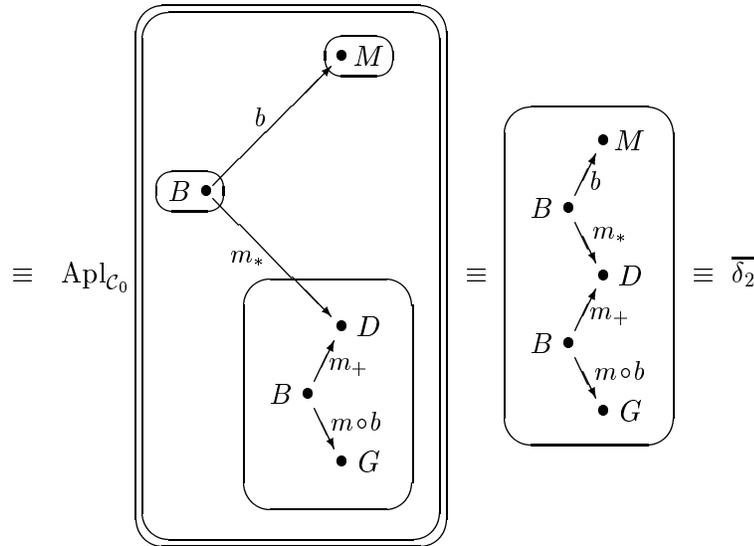


$$\mathcal{D}(\&_1(DG) \circ m_*) \equiv \mathcal{D}(\&_1(DG)) \circ \mathcal{D}(m_*)$$

$$\equiv \overline{J}_1(\mathcal{D}(B), \mathcal{D}(D), \mathcal{D}(G), \mathcal{D}(m_+), \mathcal{D}(m) \circ \mathcal{D}(b)) \circ \mathcal{D}(m_*)$$



$$\mathcal{D}(A'_2) \equiv \text{Apl}_{\mathcal{C}_0}(\text{PushDiagr}(\mathcal{D}(B), \mathcal{D}(M), \mathcal{D}(DG), \mathcal{D}(\&_1(DG)) \circ \mathcal{D}(m_*)))$$



Le diagramme associé à la spécification  $A'_2$  est donc, comme pour la spécification  $A_2$ , le diagramme  $\overline{\delta}_2$ .

### 4.2.3 Diagramme associé à la spécification $A_3$

Pour la spécification  $A_3$  des anneaux, nous avons utilisé deux spécifications intermédiaires : la spécification  $B_2$  qui regroupe deux opérateurs binaires, et la spécification  $P$  des pseudo-anneaux (cf. page 47). Rappelons qu'un pseudo-anneau est un anneau sans la propriété de distributivité entre les deux opérateurs.

$$\begin{aligned}
 B_2 &\equiv \text{push}(\emptyset, B, B, j_B, j_B) \\
 u_2 &\equiv \text{up}(\emptyset, B, B, D, j_B, j_B, m_*, m_+) : B_2 \rightarrow D \\
 P &\equiv \text{push}(S, M, G, b \circ s, m \circ b \circ s) \\
 u_1 &\equiv \text{up}(\emptyset, B, B, P, j_B, j_B, \&_1(P) \circ b, \&_2(P) \circ m \circ b) : B_2 \rightarrow P \\
 A_3 &\equiv \text{push}(B_2, P, D, u_1, u_2)
 \end{aligned}$$

Commençons par calculer le diagramme associé à la spécification  $B_2$ .

$$\mathcal{D}(B_2) \equiv \text{Apl}_{C_0}(\text{PushDiagr}(\mathcal{D}(\emptyset), \mathcal{D}(B), \mathcal{D}(B), \mathcal{D}(j_B), \mathcal{D}(j_B)))$$

$$\mathcal{D}(\emptyset) \equiv \circ$$

$$\mathcal{D}(B) \equiv \boxed{\bullet B}$$

$$\mathcal{D}(j_B) \equiv \circ \rightarrow \boxed{\bullet B}$$

$$\mathcal{D}(B_2) \equiv \text{Apl}_{C_0} \left( \begin{array}{c} \boxed{\boxed{\bullet B}} \\ \circ \\ \boxed{\boxed{\bullet B}} \end{array} \right) \equiv \boxed{\begin{array}{c} \bullet B \\ \\ \bullet B \end{array}}$$

Calculons le morphisme de diagrammes associé au morphisme de spécifications  $u_2 : B_2 \rightarrow D$ .

$$\mathcal{D}(u_2) \equiv \mathcal{D}(\text{up}(\emptyset, B, B, D, j_B, j_B, m_*, m_+))$$

$$\equiv \overline{U}(\mathcal{D}(\emptyset), \mathcal{D}(B), \mathcal{D}(B), \mathcal{D}(D), \mathcal{D}(j_B), \mathcal{D}(j_B), \mathcal{D}(m_*), \mathcal{D}(m_+))$$

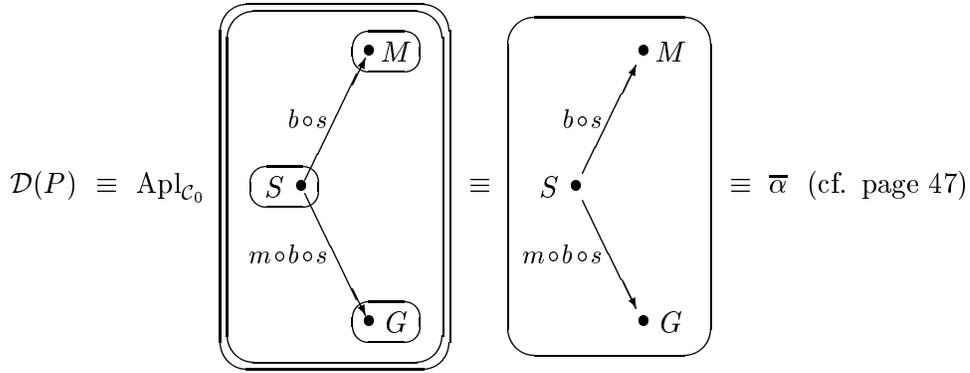
$$\mathcal{D}(D) \equiv \boxed{\bullet D}$$

$$\mathcal{D}(m_*) \equiv \boxed{B \bullet} \xrightarrow{m_*} \boxed{\bullet D}$$

$$\mathcal{D}(m_+) \equiv \boxed{B \bullet} \xrightarrow{m_+} \boxed{\bullet D}$$

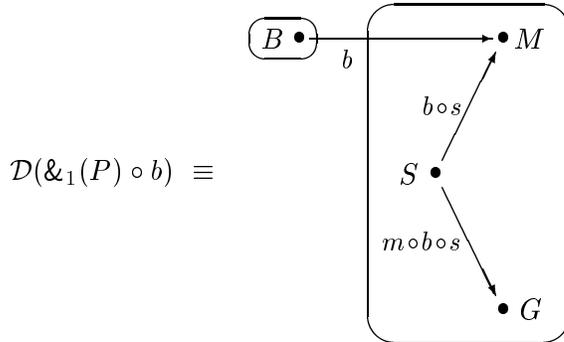
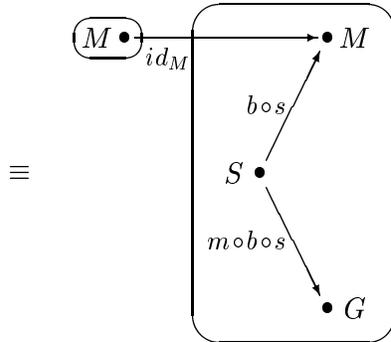
$$\mathcal{D}(u_2) \equiv \begin{array}{c} \boxed{\begin{array}{c} B \bullet \\ \\ B \bullet \end{array}} \begin{array}{l} \xrightarrow{m_*} \\ \xrightarrow{m_+} \end{array} \boxed{\bullet D} \end{array}$$

Calculons le diagramme associé à la spécification  $P$  des pseudo-anneaux.

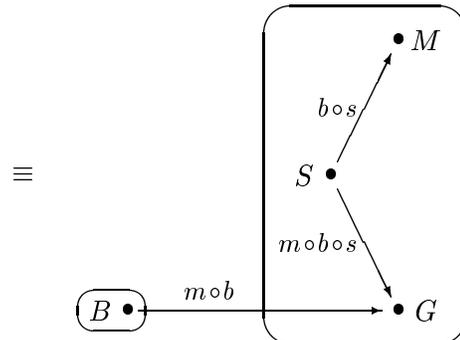


Pour calculer le morphisme de diagrammes associé au morphisme de spécifications  $u_1 : B_2 \rightarrow P$ , nous devons d'abord calculer  $\mathcal{D}(\&_1(P) \circ b)$  et  $\mathcal{D}(\&_2(P) \circ m \circ b)$ .

$$\mathcal{D}(\&_1(P)) \equiv \overline{J}_1(\mathcal{D}(S), \mathcal{D}(M), \mathcal{D}(G), \mathcal{D}(b \circ s), \mathcal{D}(m \circ b \circ s))$$

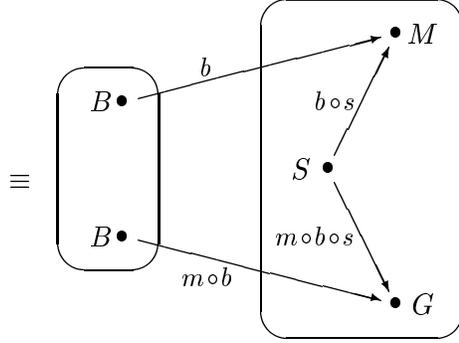


$$\mathcal{D}(\&_2(P) \circ m \circ b) \equiv \overline{J}_2(\mathcal{D}(S), \mathcal{D}(M), \mathcal{D}(G), \mathcal{D}(b \circ s), \mathcal{D}(m \circ b \circ s)) \circ \mathcal{D}(m) \circ \mathcal{D}(b)$$



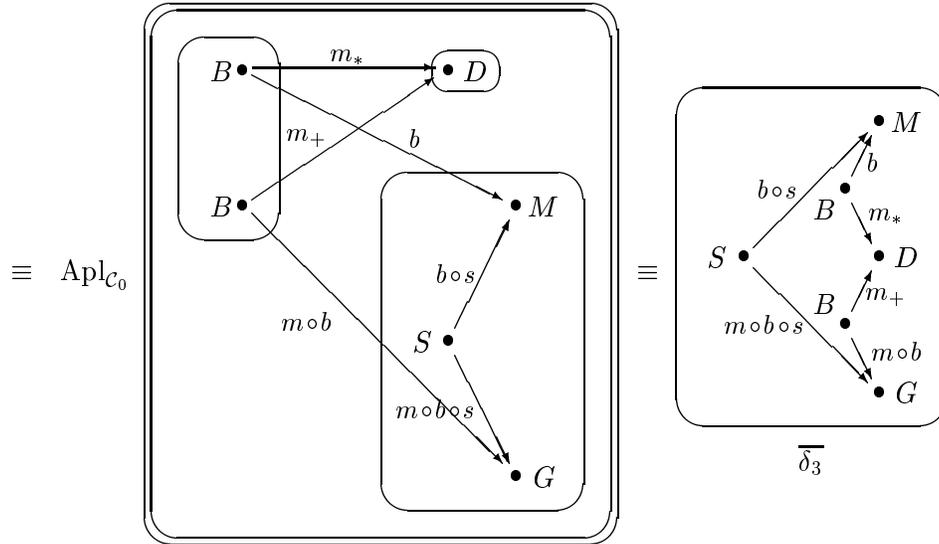
Nous calculons maintenant le morphisme de diagrammes associé à  $u_1 : B_2 \rightarrow P$ .

$$\begin{aligned} \mathcal{D}(u_1) &\equiv \mathcal{D}(\text{up}(\emptyset, B, B, P, j_B, j_B, \&_1(P) \circ b, \&_2(P) \circ m \circ b)) \\ &\equiv \overline{U}(\mathcal{D}(\emptyset), \mathcal{D}(B), \mathcal{D}(B), \mathcal{D}(P), \\ &\quad \mathcal{D}(j_B), \mathcal{D}(j_B), \mathcal{D}(\&_1(P) \circ b), \mathcal{D}(\&_2(P) \circ m \circ b)) \end{aligned}$$



Nous calculons enfin le diagramme associé à la spécification  $A_3$ .

$$\begin{aligned} \mathcal{D}(A_3) &\equiv \mathcal{D}(\text{push}(B_2, P, D, u_1, u_2)) \\ &\equiv \text{Apl}_{\mathcal{C}_0}(\text{PushDiagr}(\mathcal{D}(B_2), \mathcal{D}(P), \mathcal{D}(D), \mathcal{D}(u_1), \mathcal{D}(u_2))) \end{aligned}$$



Le diagramme associé à la spécification  $A_3$  est donc le diagramme  $\overline{\delta}_3$  donné page 49.

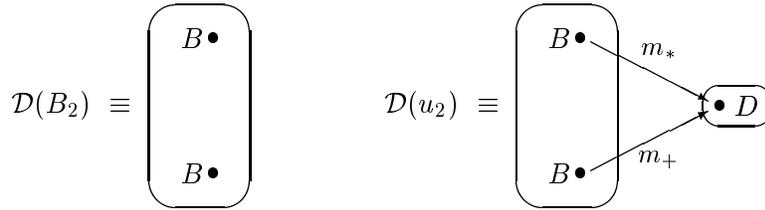
#### 4.2.4 Diagramme associé à la spécification $A_4$

Rappelons la définition de la spécification  $A_4$  des anneaux. Cette spécification utilise la spécification  $B_2$  que nous avons déjà vue dans le paragraphe précédent, ainsi qu'une autre spécification  $P'$  des pseudo-anneaux.

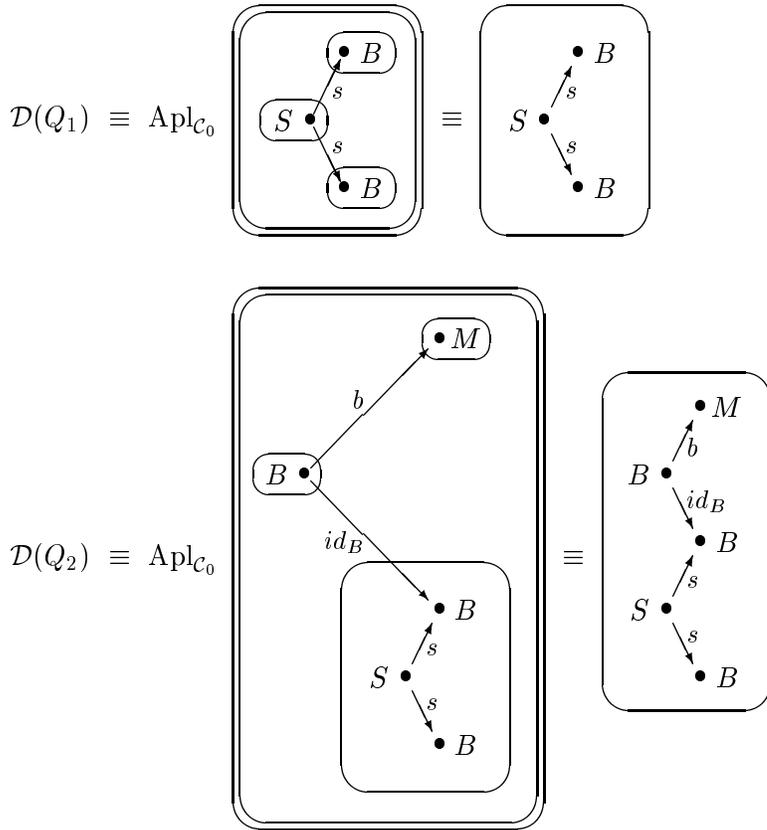
$$\begin{aligned} B_2 &\equiv \text{push}(\emptyset, B, B, j_B, j_B) \\ u_2 &\equiv \text{up}(\emptyset, B, B, D, j_B, j_B, m_*, m_+) : B_2 \rightarrow D \end{aligned}$$

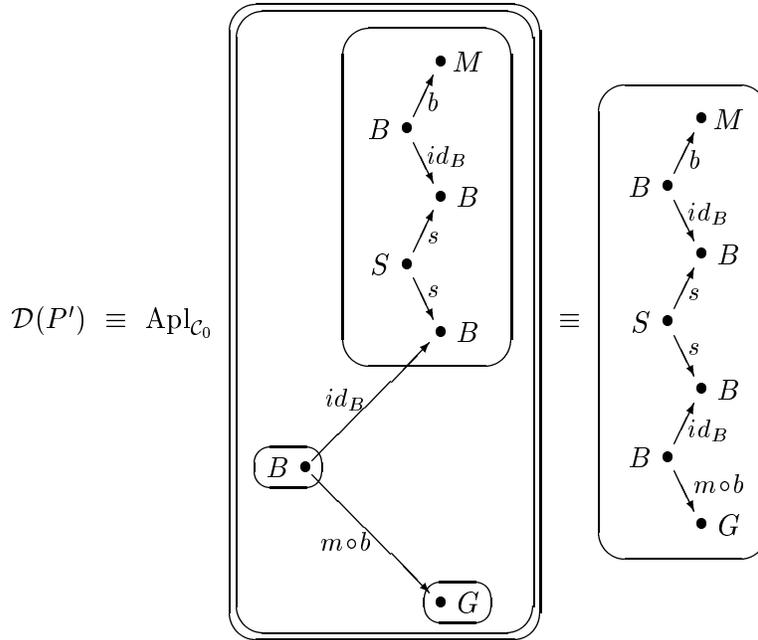
$$\begin{aligned}
 Q_1 &\equiv \text{push}(S, B, B, s, s) \\
 Q_2 &\equiv \text{push}(B, M, Q_1, b, \&_1(Q_1)) \\
 P' &\equiv \text{push}(B, Q_2, G, \&_2(Q_2) \circ \&_2(Q_1), m \circ b) \\
 u_3 &\equiv \text{up}(\emptyset, B, B, P', j_B, j_B, \&_1(P') \circ \&_2(Q_2) \circ \&_1(Q_1), \\
 &\quad \&_1(P') \circ \&_2(Q_2) \circ \&_2(Q_1) : B_2 \rightarrow P') \\
 A_4 &\equiv \text{push}(B_2, P', D, u_3, u_2)
 \end{aligned}$$

Nous avons vu dans le paragraphe précédent le diagramme associé à la spécification  $B_2$ , ainsi que le morphisme de diagrammes associé au morphisme de spécifications  $u_2 : B_2 \rightarrow D$ .

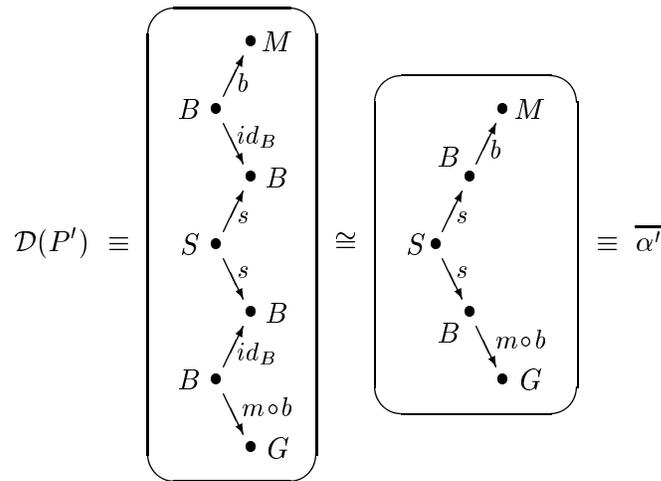


Calculons maintenant les diagrammes associés aux spécifications  $Q_1$ ,  $Q_2$  et  $P'$ .

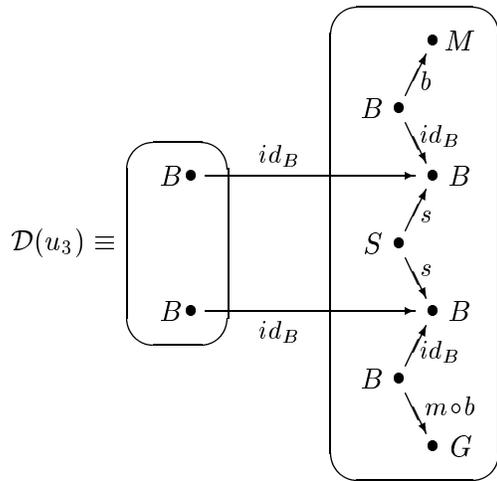




Si un diagramme  $\bar{\delta}$  contient un arc  $a : n \rightarrow n'$  étiqueté par la flèche identité (c'est-à-dire tel que  $\delta(n) = \delta(n')$  et  $\delta(a) = id_{\alpha(n)}$ ), alors on obtient un diagramme isomorphe en fusionnant les nœuds  $n$  et  $n'$  et en supprimant l'arc  $a : n \rightarrow n'$ . Ce résultat sera démontré dans le chapitre 5. Dans notre exemple, le diagramme  $\mathcal{D}(P')$  est donc isomorphe au diagramme  $\bar{\alpha}'$  ci-dessous.

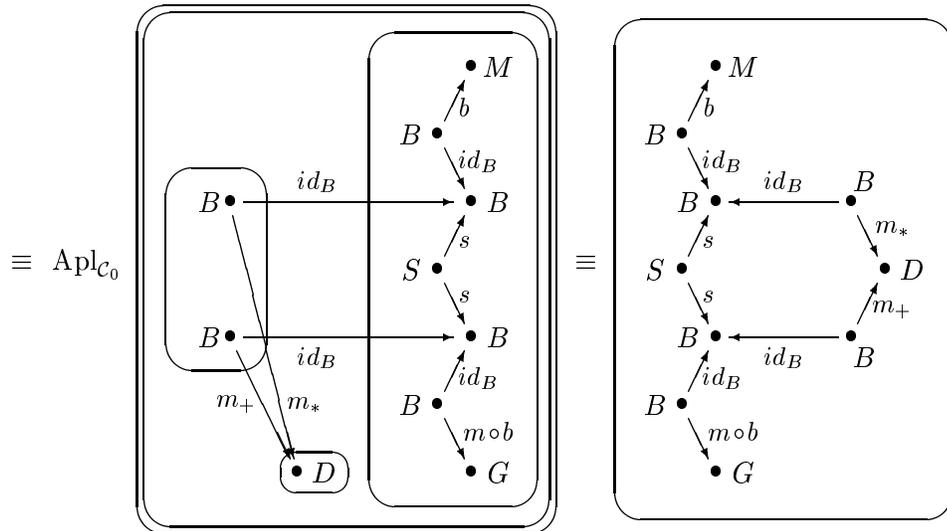


Revenons au calcul de  $\mathcal{D}(A_4)$ . Il reste à calculer le morphisme de diagrammes associé au morphisme de spécifications  $u_3 : B_2 \rightarrow P'$ .

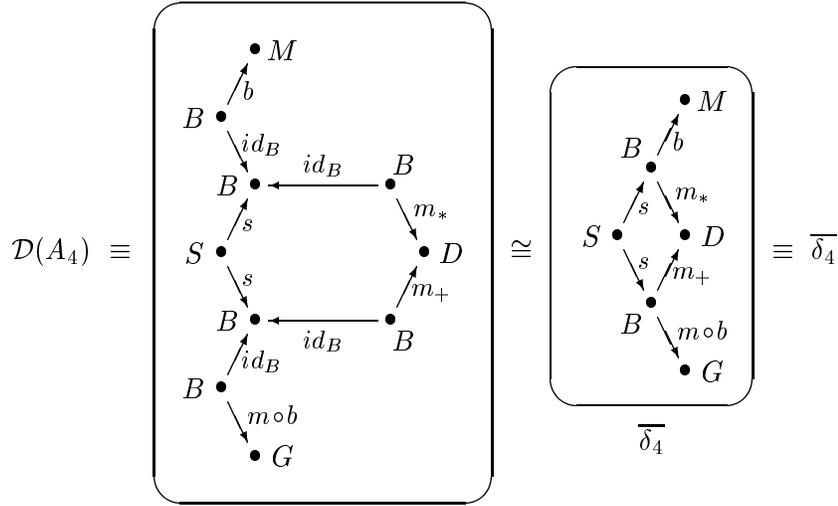


On en déduit enfin le diagramme associé à la spécification  $A_4$ .

$\mathcal{D}(A_4) \equiv \mathcal{D}(\text{push}(B_2, P^l, D, u_3, u_2))$



En supprimant les quatre arcs étiquetés par des flèches identités, on obtient un diagramme  $\bar{\delta}_4$ , isomorphe à  $\mathcal{D}(A_4)$ .



### 4.3 Équivalence entre les catégories $\text{Terme}(\mathcal{C}_0)$ et $\text{diagr}(\mathcal{C}_0)$

Dans cette partie, nous montrons que les catégories  $\text{Terme}(\mathcal{C}_0)$  et  $\text{diagr}(\mathcal{C}_0)$  sont équivalentes. Pour cela nous montrons en fait que les *précatégories*  $\text{TERME}(\mathcal{C}_0)$  et  $\text{DIAGR}(\mathcal{C}_0)$  sont équivalentes.

#### 4.3.1 Équivalence entre catégories, équivalence entre précatégories

Nous commençons par définir la notion de *transformation naturelle entre préfoncteurs*.

**Définition 4.1** (Transformation naturelle entre deux préfoncteurs) Soit deux précatégories  $\mathcal{C}$  et  $\mathcal{C}'$ . Soit deux préfoncteurs  $F, G : \mathcal{C} \rightarrow \mathcal{C}'$ . Une *transformation naturelle*  $\sigma$  de  $F$  vers  $G$ , notée  $\sigma : F \rightarrow G$ , est une application qui à tout objet  $A$  de  $\mathcal{C}$  associe une flèche  $\sigma_A \in \text{Arr}(\mathcal{C}')(F(A), G(A))$ , telle que pour toute flèche  $f \in \text{Arr}(\mathcal{C})(A, B)$ ,

$$G(f) \circ \sigma_A = \sigma_B \circ F(f) \in \text{Arr}(\mathcal{C}')(F(A), G(B)).$$

Par conséquent,  $\sigma$  est une transformation naturelle entre les deux préfoncteurs  $F$  et  $G$  si et seulement si  $\sigma$  est une transformation naturelle entre les foncteurs  $F$  et  $G$ .

**Définition 4.2** (Isomorphisme naturel entre deux préfoncteurs) Une transformation naturelle  $\sigma : F \rightarrow G$  entre deux préfoncteurs  $F, G : \mathcal{C} \rightarrow \mathcal{C}'$  est un *isomorphisme naturel* si et seulement si pour tout objet  $A$  de  $\mathcal{C}$ ,  $\sigma_A \in \text{Arr}(\mathcal{C}')(F(A), G(A))$  est un isomorphisme.

Si  $\sigma : F \rightarrow G$  est un isomorphisme naturel, on dit que les préfoncteurs  $F$  et  $G$  sont *naturellement isomorphes*, et on note  $F \cong G$ . On vérifie facilement que deux préfoncteurs sont naturellement isomorphes si et seulement si les foncteurs associés sont naturellement isomorphes.

Rappelons la définition d'*équivalence* entre deux catégories.

**Définition 4.3** (Équivalence) Deux catégories  $\mathcal{C}$  et  $\mathcal{C}'$  sont *équivalentes* si et seulement si il existe deux foncteurs  $F : \mathcal{C} \rightarrow \mathcal{C}'$  et  $G : \mathcal{C}' \rightarrow \mathcal{C}$  tels qu'on ait deux isomorphismes naturels  $G \circ F \cong \text{Id}_{\mathcal{C}}$  et  $F \circ G \cong \text{Id}_{\mathcal{C}'}$ .

On en déduit une définition d'*équivalence* entre deux *précatégories*.

**Définition 4.4** Deux précatégories  $\mathcal{C}$  et  $\mathcal{C}'$  sont *équivalentes* si et seulement si il existe deux préfoncteurs  $F : \mathcal{C} \rightarrow \mathcal{C}'$  et  $G : \mathcal{C}' \rightarrow \mathcal{C}$  tels qu'on ait deux isomorphismes naturels  $G \circ F \cong \text{Id}_{\mathcal{C}}$  et  $F \circ G \cong \text{Id}_{\mathcal{C}'}$ .

Deux précatégories sont équivalentes si et seulement si les catégories associées sont équivalentes.

### 4.3.2 Équivalence entre les précatégories $\text{DIAGR}(\mathcal{C}_0)$ et $\text{TERME}(\mathcal{C}_0)$

Dans ce paragraphe, nous montrons qu'il existe une équivalence entre les précatégories  $\text{DIAGR}(\mathcal{C}_0)$  et  $\text{TERME}(\mathcal{C}_0)$ . Intuitivement, cela signifie que ces deux précatégories ont "la même structure", "aux isomorphismes entre objets près".

Nous commençons par reformuler le théorème 2.7 page 114 en utilisant des précatégories.

**Théorème 4.1** La précatégorie  $\text{DIAGR}(\mathcal{C}_0)$  est une complétion de  $\mathcal{C}_0$  par pré-colimites finies. Autrement dit, pour toute précatégorie  $\mathcal{E}$  finiment pré-cocomplète, et tout préfoncteur  $F : \mathcal{C}_0 \rightarrow \mathcal{E}$ , il existe un préfoncteur

$$H : \text{DIAGR}(\mathcal{C}_0) \rightarrow \mathcal{E},$$

unique à un isomorphisme naturel près, qui conserve les pré-colimites finies, tel que

$$H \circ I_{\mathcal{C}_0} \cong F.$$

Comme la précatégorie  $\text{TERME}(\mathcal{C}_0)$  est finiment pré-cocomplète, et comme on a un préfoncteur  $J : \mathcal{C}_0 \rightarrow \text{TERME}(\mathcal{C}_0)$ , il existe un préfoncteur

$$\mathcal{T} : \text{DIAGR}(\mathcal{C}_0) \rightarrow \text{TERME}(\mathcal{C}_0),$$

qui conserve les pré-colimites finies, tel que

$$\mathcal{T} \circ I_{\mathcal{C}_0} \cong J.$$

D'autre part, nous avons vu au paragraphe 4.1.2 que le préfoncteur

$$\mathcal{D} : \text{TERME}(\mathcal{C}_0) \rightarrow \text{DIAGR}(\mathcal{C}_0)$$

conserve l'objet pré-initial et les pré-sommes amalgamées. Le préfoncteur  $\mathcal{D}$  conserve donc les pré-colimites finies. De plus, on a

$$\mathcal{D} \circ J \equiv I_{\mathcal{C}_0}.$$

**Lemme 4.3** *On a les isomorphismes naturels suivants :*

1.  $\mathcal{D} \circ \mathcal{T} \circ I_{\mathcal{C}_0} \cong I_{\mathcal{C}_0}$  ;
2.  $\mathcal{T} \circ \mathcal{D} \circ J \cong J$ .

*Preuve.* C'est une conséquence immédiate de  $\mathcal{T} \circ I_{\mathcal{C}_0} \cong J$  et  $\mathcal{D} \circ J \equiv I_{\mathcal{C}_0}$ . □

**Proposition 4.1** *On a un isomorphisme naturel*

$$\mathcal{D} \circ \mathcal{T} \cong Id_{\text{DIAGR}(\mathcal{C}_0)}.$$

*Preuve.* D'après le lemme 4.3–1, on a  $\mathcal{D} \circ \mathcal{T} \circ I_{\mathcal{C}_0} \cong I_{\mathcal{C}_0}$ . De plus, les préfoncteurs  $\mathcal{D} \circ \mathcal{T} : \text{DIAGR}(\mathcal{C}_0) \rightarrow \text{DIAGR}(\mathcal{C}_0)$  et  $Id_{\text{DIAGR}(\mathcal{C}_0)} : \text{DIAGR}(\mathcal{C}_0) \rightarrow \text{DIAGR}(\mathcal{C}_0)$  conservent les pré-colimites finies. Par conséquent, d'après le théorème 4.1,

$$\mathcal{D} \circ \mathcal{T} \cong Id_{\text{DIAGR}(\mathcal{C}_0)}.$$

□

**Lemme 4.4** *Soit  $A, B, C \in \text{Obj}(\text{TERME}(\mathcal{C}_0))$ ,  $f \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$  et  $g \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, C)$ . Alors, le triplet*

$$\left( \begin{array}{l} (\mathcal{T} \circ \mathcal{D})(\text{push}(A, B, C, f, g)), \\ (\mathcal{T} \circ \mathcal{D})(\&_1(A, B, C, f, g)), \\ (\mathcal{T} \circ \mathcal{D})(\&_2(A, B, C, f, g)) \end{array} \right)$$

*est une pré-somme amalgamée de  $(\mathcal{T} \circ \mathcal{D})(B)$  et  $(\mathcal{T} \circ \mathcal{D})(C)$  par rapport aux flèches  $(\mathcal{T} \circ \mathcal{D})(f)$  et  $(\mathcal{T} \circ \mathcal{D})(g)$ .*

*Preuve.* Comme les préfoncteurs  $\mathcal{D}$  et  $\mathcal{T}$  conservent les pré-sommes amalgamées, le préfoncteur  $\mathcal{T} \circ \mathcal{D}$  conserve également les pré-sommes amalgamées. □

**Proposition 4.2** *On a un isomorphisme naturel*

$$\Psi : \mathcal{T} \circ \mathcal{D} \xrightarrow{\cong} Id_{\text{TERME}(\mathcal{C}_0)}.$$

*Preuve.* Comme  $\mathcal{T} \circ \mathcal{D}$  et  $\text{Id}_{\text{TERME}(\mathcal{C}_0)}$  sont des préfoncteurs de la précatégorie  $\text{TERME}(\mathcal{C}_0)$  vers  $\text{TERME}(\mathcal{C}_0)$ , il faut donner, pour tout objet  $U$  de  $\text{TERME}(\mathcal{C}_0)$ , un isomorphisme

$$\Psi_U : (\mathcal{T} \circ \mathcal{D})(U) \rightarrow U$$

dans  $\text{TERME}(\mathcal{C}_0)$ . On définit cet isomorphisme par induction sur la structure des objets de  $\text{TERME}(\mathcal{C}_0)$ , et on vérifie qu'il s'agit bien d'une transformation naturelle par induction sur la structure des flèches de  $\text{TERME}(\mathcal{C}_0)$ .

Soit un objet  $U$  de  $\text{TERME}(\mathcal{C}_0)$ . On définit  $\Psi_U$  par induction sur la structure des objets de  $\text{TERME}(\mathcal{C}_0)$ .

*Règle (1)*  $U \equiv J(A)$ , où  $A$  est un objet de  $\mathcal{C}_0$ .

D'après le lemme 4.3–2,  $\mathcal{T} \circ \mathcal{D} \circ J \cong J$ . Par conséquent, il existe un isomorphisme

$$\Psi_U : (\mathcal{T} \circ \mathcal{D})(U) \rightarrow U.$$

*Règle (2)*  $U \equiv \emptyset$ .

$(\mathcal{T} \circ \mathcal{D})(\emptyset) \equiv \mathcal{T}(\emptyset) \cong \emptyset$  car  $\mathcal{T}$  conserve l'objet pré-initial. Soit

$$\Psi_{\emptyset} : (\mathcal{T} \circ \mathcal{D})(\emptyset) \rightarrow \emptyset$$

cet isomorphisme.

*Règle (3)*  $U \equiv \text{push}(A, B, C, f, g)$ .

Par hypothèse d'induction, on a trois isomorphismes

$$\begin{aligned} \Psi_A &: (\mathcal{T} \circ \mathcal{D})(A) \rightarrow A \\ \Psi_B &: (\mathcal{T} \circ \mathcal{D})(B) \rightarrow B \\ \Psi_C &: (\mathcal{T} \circ \mathcal{D})(C) \rightarrow C \end{aligned}$$

tels que

$$\begin{aligned} f \circ \Psi_A &= \Psi_B \circ (\mathcal{T} \circ \mathcal{D})(f) \\ g \circ \Psi_A &= \Psi_C \circ (\mathcal{T} \circ \mathcal{D})(g). \end{aligned}$$

D'après le lemme 4.4, le triplet

$$\left( \begin{array}{l} (\mathcal{T} \circ \mathcal{D})(\text{push}(A, B, C, f, g)), \\ (\mathcal{T} \circ \mathcal{D})(\&_1(A, B, C, f, g)), \\ (\mathcal{T} \circ \mathcal{D})(\&_2(A, B, C, f, g)) \end{array} \right)$$

est une pré-somme amalgamée de  $(\mathcal{T} \circ \mathcal{D})(B)$  et  $(\mathcal{T} \circ \mathcal{D})(C)$  par rapport aux flèches  $(\mathcal{T} \circ \mathcal{D})(f)$  et  $(\mathcal{T} \circ \mathcal{D})(g)$ .

D'autre part, le triplet

$$(\text{push}(A, B, C, f, g), \&_1(A, B, C, f, g), \&_1(A, B, C, f, g))$$

est évidemment une pré-somme amalgamée de  $B$  et  $C$  par rapport aux flèches  $f$  et  $g$  dans  $\text{TERME}(\mathcal{C}_0)$ .

Par conséquent, il existe un unique isomorphisme

$$\Psi_U = \Psi_{\text{push}(A,B,C,f,g)} : (\mathcal{T} \circ \mathcal{D})(\text{push}(A, B, C, f, g)) \rightarrow \text{push}(A, B, C, f, g)$$

tel que

$$\begin{aligned} \Psi_U \circ (\mathcal{T} \circ \mathcal{D})(\&_1(A, B, C, f, g)) &= \&_1(A, B, C, f, g) \circ \Psi_B \\ \Psi_U \circ (\mathcal{T} \circ \mathcal{D})(\&_2(A, B, C, f, g)) &= \&_2(A, B, C, f, g) \circ \Psi_C. \end{aligned}$$

On vérifie maintenant qu'on définit bien une *transformation naturelle*

$$\Psi : \mathcal{T} \circ \mathcal{D} \rightarrow \text{Id}_{\text{TERME}(\mathcal{C}_0)},$$

par induction sur la structure des flèches de  $\text{TERME}(\mathcal{C}_0)$ .

*Règle (4)* Soit  $f : A \rightarrow B$  une flèche de  $\mathcal{C}_0$ . Comme  $\Psi_A$  et  $\Psi_B$  sont les isomorphismes correspondant à l'isomorphisme naturel  $\mathcal{T} \circ \mathcal{D} \circ J \cong J$ , on a donc

$$\Psi_B \circ (\mathcal{T} \circ \mathcal{D})(f) = f \circ \Psi_A.$$

*Règle (5)* Soit deux flèches  $f : A \rightarrow B$  et  $g : B \rightarrow C$  de  $\text{TERME}(\mathcal{C}_0)$ .

$$\begin{aligned} \Psi_C \circ (\mathcal{T} \circ \mathcal{D})(g \circ f) &= \Psi_C \circ (\mathcal{T} \circ \mathcal{D})(g) \circ (\mathcal{T} \circ \mathcal{D})(f) && (\mathcal{T} \circ \mathcal{D} \text{ préfoncteur}) \\ &= g \circ \Psi_B \circ (\mathcal{T} \circ \mathcal{D})(f) && (\text{hypothèse d'induction sur } g) \\ &= g \circ f \circ \Psi_A && (\text{hypothèse d'induction sur } f) \end{aligned}$$

*Règle (6)*  $\Psi_A \circ (\mathcal{T} \circ \mathcal{D})(\text{id}_A) = \text{id}_A \circ \Psi_A$

*Règle (7)*  $\Psi_A \circ (\mathcal{T} \circ \mathcal{D})(j_A) = j_A \circ \Psi_\emptyset$  car ces deux flèches ont pour domaine un objet pré-initial de  $\text{TERME}(\mathcal{C}_0)$ .

*Règle (8)* Par définition de  $\Psi_{\text{push}(A,B,C,f,g)}$ , on a

$$\Psi_{\text{push}(A,B,C,f,g)} \circ (\mathcal{T} \circ \mathcal{D})(\&_1(A, B, C, f, g)) = \&_1(A, B, C, f, g) \circ \Psi_B.$$

*Règle (9)* Par définition de  $\Psi_{\text{push}(A,B,C,f,g)}$ , on a

$$\Psi_{\text{push}(A,B,C,f,g)} \circ (\mathcal{T} \circ \mathcal{D})(\&_2(A, B, C, f, g)) = \&_2(A, B, C, f, g) \circ \Psi_C.$$

*Règle (10)* Posons

$$\begin{aligned} \&_1 &= \&_1(A, B, C, f, g) \\ \&_2 &= \&_2(A, B, C, f, g) \\ \text{up} &= \text{up}(A, B, C, D, f, g, f', g') \\ \Psi_{\text{push}} &= \Psi_{\text{push}(A,B,C,f,g)} \end{aligned}$$

Il faut montrer que

$$\Psi_D \circ (\mathcal{T} \circ \mathcal{D})(\text{up}) = \text{up} \circ \Psi_{\text{push}}.$$

Pour cela, il suffit de montrer

$$\begin{aligned}\Psi_D \circ (\mathcal{T} \circ \mathcal{D})(\text{up}) \circ (\mathcal{T} \circ \mathcal{D})(\&_1) &= \text{up} \circ \Psi_{\text{push}} \circ (\mathcal{T} \circ \mathcal{D})(\&_1) \\ \Psi_D \circ (\mathcal{T} \circ \mathcal{D})(\text{up}) \circ (\mathcal{T} \circ \mathcal{D})(\&_2) &= \text{up} \circ \Psi_{\text{push}} \circ (\mathcal{T} \circ \mathcal{D})(\&_2).\end{aligned}$$

$$\begin{aligned}\Psi_D \circ (\mathcal{T} \circ \mathcal{D})(\text{up}) \circ (\mathcal{T} \circ \mathcal{D})(\&_1) & \\ &= \Psi_D \circ (\mathcal{T} \circ \mathcal{D})(\text{up} \circ \&_1) && (\mathcal{T} \circ \mathcal{D} \text{ préfoncteur}) \\ &= \Psi_D \circ (\mathcal{T} \circ \mathcal{D})(f') && (\text{définition de up}) \\ &= f' \circ \Psi_B && (\text{hypothèse d'induction sur } f') \\ &= \text{up} \circ \&_1 \circ \Psi_B && (\text{définition de up}) \\ &= \text{up} \circ \Psi_{\text{push}} \circ (\mathcal{T} \circ \mathcal{D})(\&_1) && (\text{règle 8})\end{aligned}$$

On montre de la même façon que

$$\Psi_D \circ (\mathcal{T} \circ \mathcal{D})(\text{up}) \circ (\mathcal{T} \circ \mathcal{D})(\&_2) = \text{up} \circ \Psi_{\text{push}} \circ (\mathcal{T} \circ \mathcal{D})(\&_2).$$

□

**Théorème 4.2** *Les précatégories  $\text{TERME}(\mathcal{C}_0)$  et  $\text{DIAGR}(\mathcal{C}_0)$  sont équivalentes.*

*Preuve.* Conséquence immédiate des propositions 4.1 et 4.2. □

**Corollaire 4.1** *Les catégories  $\text{Terme}(\mathcal{C}_0)$  et  $\text{diagr}(\mathcal{C}_0)$  sont équivalentes.*

La proposition 4.2 nous permet de montrer que deux objets de  $\text{Terme}(\mathcal{C}_0)$  sont isomorphes si et seulement si leurs diagrammes associés sont isomorphes, et deux flèches de  $\text{Terme}(\mathcal{C}_0)$  sont égales si et seulement si leurs morphismes de diagrammes associés sont égaux dans  $\text{diagr}(\mathcal{C}_0)$ .

**Théorème 4.3** *Considérons le foncteur  $\mathcal{D} : \text{Terme}(\mathcal{C}_0) \rightarrow \text{diagr}(\mathcal{C}_0)$ .*

- Soit  $A$  et  $B$  deux objets de  $\text{Terme}(\mathcal{C}_0)$ . Alors,

$$A \cong B \Leftrightarrow \mathcal{D}(A) \cong \mathcal{D}(B).$$

- Soit  $f, g : A \rightarrow B$  deux flèches de  $\text{Terme}(\mathcal{C}_0)$ . Alors,

$$f = g \Leftrightarrow \mathcal{D}(f) = \mathcal{D}(g).$$

*Preuve.* Les deux implications de gauche à droite sont évidentes car  $\mathcal{D}$  est un foncteur. Dans l'autre sens, d'après la proposition 4.2, il existe un isomorphisme naturel  $\Psi : \mathcal{T} \circ \mathcal{D} \xrightarrow{\sim} \text{Id}_{\text{Terme}(\mathcal{C}_0)}$  entre les foncteurs  $\mathcal{T} \circ \mathcal{D} : \text{Terme}(\mathcal{C}_0) \rightarrow \text{Terme}(\mathcal{C}_0)$  et  $\text{Id}_{\text{Terme}(\mathcal{C}_0)} : \text{Terme}(\mathcal{C}_0) \rightarrow \text{Terme}(\mathcal{C}_0)$ .

- Soit deux objets  $A$  et  $B$  de  $\text{Terme}(\mathcal{C}_0)$ .

$$\begin{aligned}\mathcal{D}(A) \cong \mathcal{D}(B) & \\ \Rightarrow (\mathcal{T} \circ \mathcal{D})(A) \cong (\mathcal{T} \circ \mathcal{D})(B) & \\ \Rightarrow A \cong B & \quad (\Psi : \mathcal{T} \circ \mathcal{D} \xrightarrow{\sim} \text{Id}_{\text{Terme}(\mathcal{C}_0)})\end{aligned}$$

- Soit deux flèches  $f, g : A \rightarrow B$  de  $\text{Terme}(\mathcal{C}_0)$  telles que  $\mathcal{D}(f) = \mathcal{D}(g)$  dans  $\text{diagr}(\mathcal{C}_0)$ .

$$\begin{aligned}
\mathcal{D}(f) &= \mathcal{D}(g) \\
\Rightarrow (\mathcal{T} \circ \mathcal{D})(f) &= (\mathcal{T} \circ \mathcal{D})(g) \\
\Rightarrow \Psi_B \circ (\mathcal{T} \circ \mathcal{D})(f) &= \Psi_B \circ (\mathcal{T} \circ \mathcal{D})(g) \\
\Rightarrow f \circ \Psi_A &= g \circ \Psi_A && (\Psi : \mathcal{T} \circ \mathcal{D} \dashrightarrow \text{Id}_{\text{Terme}(\mathcal{C}_0)}) \\
\Rightarrow f &= g && (\Psi_A \text{ isomorphisme})
\end{aligned}$$

□

### 4.3.3 Correction du calcul de diagrammes

La proposition 4.2 nous permet également de déduire que le calcul de diagrammes décrit par le foncteur  $\mathcal{D} : \text{Terme}(\mathcal{C}_0) \rightarrow \text{diagr}(\mathcal{C}_0)$  est *correct* c'est-à-dire que la colimite d'un diagramme associé à une spécification modulaire est isomorphe à cette spécification.

Considérons une catégorie d'interprétation  $\mathcal{E}$  à objet initial choisi et sommes amalgamées choisies, par exemple la catégorie des spécifications **Spec**. Soit une interprétation de  $\mathcal{C}_0$ , c'est-à-dire un foncteur

$$\mathcal{S}_0 : \mathcal{C}_0 \rightarrow \mathcal{E}.$$

D'après le théorème 2.7, ce foncteur s'étend aux diagrammes en un foncteur

$$\mathcal{S}_D : \text{diagr}(\mathcal{C}_0) \rightarrow \mathcal{E}.$$

On a en fait  $\mathcal{S}_D = \text{colim}_{\mathcal{E}} \circ \text{diagr}(\mathcal{S}_0)$ .

D'après le théorème 3.9, le foncteur  $\mathcal{S}_0$  s'étend également aux termes en un foncteur

$$\mathcal{S}_T : \text{Terme}(\mathcal{C}_0) \rightarrow \mathcal{E}.$$

De plus, comme  $\mathcal{S}_D$  et  $\mathcal{S}_T$  conservent les colimites, d'après le théorème 2.7, on a

$$\mathcal{S}_D \cong \mathcal{S}_T \circ \mathcal{T}.$$

Par conséquent,  $\mathcal{S}_D \circ \mathcal{D} \cong \mathcal{S}_T \circ \mathcal{T} \circ \mathcal{D}$ , et donc, d'après la proposition 4.2,

$$\mathcal{S}_D \circ \mathcal{D} \cong \mathcal{S}_T.$$

Ce résultat exprime que la colimite d'un diagramme associé à une spécification modulaire est isomorphe à cette spécification.

## 4.4 Conclusion

Dans ce chapitre, nous avons montré comment associer à tout terme qui représente une spécification modulaire un diagramme dont la colimite est isomorphe à cette spécification. Cette association est décrite par le préfoncteur

$$\mathcal{D} : \text{TERME}(\mathcal{C}_0) \rightarrow \text{DIAGR}(\mathcal{C}_0).$$

Le calcul d'un diagramme associé à un terme consiste simplement à appliquer les règles de calcul de  $\mathcal{D}$  définies dans le paragraphe 4.1.2. Une des règles principales est la règle 3 qui *aplatit* un diagramme de diagrammes, c'est-à-dire transforme un objet de  $\text{DIAGR}^2(\mathcal{C}_0)$  en un objet de  $\text{DIAGR}(\mathcal{C}_0)$ .

Nous avons montré que les précatégories  $\text{TERME}(\mathcal{C}_0)$  et  $\text{DIAGR}(\mathcal{C}_0)$  sont équivalentes, ce qui signifie qu'elles ont essentiellement la même structure "aux isomorphismes entre objets près". Il faut néanmoins remarquer que  $\text{TERME}(\mathcal{C}_0)$  et  $\text{DIAGR}(\mathcal{C}_0)$  *ne sont pas isomorphes*. En fait, pour qu'elles soient isomorphes, il faudrait que ces deux précatégories aient les mêmes choix de pré-colimites, ce qui n'est pas le cas. Par exemple,  $\text{push}(A, B, C, f, g)$  et  $\text{push}(A, C, B, g, f)$  sont deux pré-sommes amalgamées différentes dans  $\text{TERME}(\mathcal{C}_0)$  qui sont envoyées sur le même diagramme dans  $\text{DIAGR}(\mathcal{C}_0)$ .



# Chapitre 5

## Isomorphismes entre diagrammes

Nous considérons une catégorie de base  $\mathcal{C}_0$ , et nous nous intéressons dans ce chapitre à la détection d'isomorphismes entre deux diagrammes dans la catégorie  $\text{diagr}(\mathcal{C}_0)$ . Nous avons vu dans le chapitre 4 que deux objets  $A$  et  $B$  de la catégorie  $\text{Terme}(\mathcal{C}_0)$  sont isomorphes si et seulement si leurs diagrammes associés  $\mathcal{D}(A)$  et  $\mathcal{D}(B)$  sont isomorphes (théorème 4.3 page 169). Une solution pour savoir si deux termes  $A$  et  $B$  sont isomorphes consiste donc à détecter un isomorphisme entre leurs diagrammes associés.

En section 5.1, nous décrivons des transformations de diagrammes, qui sont des fonctions sur l'ensemble des diagrammes sur  $\mathcal{C}_0$ . Les transformations qui nous intéressent particulièrement sont les transformations *stables par isomorphisme*, c'est-à-dire telles que le diagramme résultat est isomorphe au diagramme de départ. Ces transformations nous serviront à calculer la *complétion* d'un diagramme.

En section 5.2, nous présentons un algorithme pour détecter si deux diagrammes sont isomorphes, dans le cas particulier où la catégorie  $\mathcal{C}_0$  est finie et sans cycle. Cet algorithme consiste à compléter puis à calculer la *forme minimale* du diagramme. Ensuite, deux diagrammes sont isomorphes si et seulement si ils ont la même forme minimale.

Enfin, en section 5.3, nous montrons les difficultés rencontrées lorsque la catégorie de base  $\mathcal{C}_0$  comporte des cycles ou est infinie.

### 5.1 Transformations de diagrammes

Nous présentons quelques transformations de diagrammes : la *substitution par objet isomorphe*, la *suppression d'un arc*, l'*ajout d'un arc étiqueté* et la *contraction d'un arc identité*.

**Définition 5.1** (Transformation de diagrammes) Une transformation de diagrammes est une fonction sur l'ensemble des objets de  $\text{diagr}(\mathcal{C}_0)$

$$\mathcal{F} : \text{diagr}(\mathcal{C}_0) \rightarrow \text{diagr}(\mathcal{C}_0).$$

Les transformations intéressantes sont les transformations *stables par isomorphisme* c'est-à-dire qui produisent un diagramme isomorphe au diagramme de départ.

**Définition 5.2** (Transformation stable par isomorphisme) Une transformation de diagrammes  $\mathcal{F} : \text{diagr}(\mathcal{C}_0) \rightarrow \text{diagr}(\mathcal{C}_0)$  est *stable par isomorphisme* si et seulement si pour tout diagramme  $\bar{\delta}$ ,

$$\mathcal{F}(\bar{\delta}) \cong \bar{\delta}.$$

Deux diagrammes  $\bar{\alpha}$  et  $\bar{\beta}$  sur  $\mathcal{C}_0$  sont isomorphes dans  $\text{diagr}(\mathcal{C}_0)$  s'il existe deux flèches  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta}$  et  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\alpha}$  de  $\text{DIAGR}(\mathcal{C}_0)$  telles que

$$\begin{cases} [\bar{\sigma}] \circ [\bar{\tau}] = [\overline{id_{\bar{\beta}}}] \\ [\bar{\tau}] \circ [\bar{\sigma}] = [\overline{id_{\bar{\alpha}}}] \end{cases} \Leftrightarrow \begin{cases} \bar{\sigma} \circ \bar{\tau} \approx \overline{id_{\bar{\beta}}} \\ \bar{\tau} \circ \bar{\sigma} \approx \overline{id_{\bar{\alpha}}}. \end{cases}$$

Rappelons qu'une flèche  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta}$  de  $\text{DIAGR}(\mathcal{C}_0)$  est un couple (cf. définition 2.18 page 70)

$$\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta} = (\sigma^\Phi : \alpha^\Phi \rightsquigarrow \beta^\Phi, \sigma : \alpha \rightsquigarrow \beta \circ \sigma^\Phi)$$

où  $\sigma^\Phi : \alpha^\Phi \rightsquigarrow \beta^\Phi$  est un morphisme généralisé de graphes et  $\sigma : \alpha \rightsquigarrow \beta \circ \sigma^\Phi$  est une transformation naturelle généralisée.

- Le morphisme généralisé de graphes

$$\sigma^\Phi : \alpha^\Phi \rightsquigarrow \beta^\Phi$$

associe à tout nœud de  $\alpha^\Phi$  un nœud de  $\beta^\Phi$ , et à tout arc  $a : m \rightarrow n$  de  $\alpha^\Phi$  un *zigzag*  $\sigma^\Phi(a) : \sigma^\Phi(m) \rightsquigarrow \sigma^\Phi(n)$  sur le graphe  $\beta^\Phi$ .

- La transformation naturelle généralisée  $\sigma : \alpha \rightsquigarrow \beta \circ \sigma^\Phi$  associe à tout nœud  $n$  de  $\alpha^\Phi$  une flèche  $\sigma_n : \alpha(n) \rightarrow \beta(\sigma^\Phi(n))$ . De plus, la condition suivante doit être satisfaite: pour tout arc  $a : m \rightarrow n$  de  $\alpha^\Phi$ ,

$$\sigma_n \circ \alpha(a) \sim_{\bar{\beta}} \sigma_m [\sigma^\Phi(a)].$$

### 5.1.1 Substitution par objet isomorphe

Soit un diagramme  $\bar{\delta}$  sur  $\mathcal{C}_0$ , un nœud  $n_0$  de  $\delta^\Phi$ , et un objet  $B$  de  $\mathcal{C}_0$  isomorphe à  $\delta(n_0)$ . Intuitivement, la *substitution par objet isomorphe* consiste à remplacer l'étiquette  $\delta(n_0)$  du nœud  $n_0$  par l'objet  $B$ . On obtient alors un diagramme isomorphe au diagramme de départ  $\bar{\delta}$ .

**Définition 5.3** (Subst\_Obj\_Iso) Soit un diagramme  $\bar{\delta}$ , un nœud  $n_0$  de  $\delta^\Phi$ , et un objet  $B$  de  $\mathcal{C}_0$  isomorphe à  $\delta(n_0)$ . Soit  $\phi : \delta(n_0) \rightarrow B$  l'isomorphisme entre  $\delta(n_0)$  et  $B$ . Le diagramme

$$\beta = \text{Subst\_Obj\_Iso}(\bar{\delta}, n_0, B, \phi)$$

est défini de la façon suivante.

- Graphe  $\beta^\Phi : \beta^\Phi = \delta^\Phi$ .

- Foncteur  $\beta : \beta^\Phi \rightarrow \mathcal{C}_0$ .
  - Action sur les nœuds.
    - $\forall n \in \text{Nœud}(\delta^\Phi), n \neq n_0 : \beta(n) = \delta(n)$ ;
    - $\beta(n_0) = B$ .
  - Action sur les arcs. Soit un arc  $a : n \rightarrow n' \in \text{Arc}(\delta^\Phi)$ .
    - si  $n \neq n_0$  et  $n' \neq n_0 : \beta(a) = \delta(a)$ ;
    - si  $n = n_0$  et  $n' \neq n_0 : \beta(a) = \delta(a) \circ \phi^{-1}$ ;
    - si  $n \neq n_0$  et  $n' = n_0 : \beta(a) = \phi \circ \delta(a)$ ;
    - si  $n = n_0$  et  $n' = n_0 : \beta(a) = \phi \circ \delta(a) \circ \phi^{-1}$ .

**Proposition 5.1** *La transformation Subst\_Obj\_Iso est stable par isomorphisme.*

*Preuve.* Soit un diagramme  $\bar{\delta}$ , un nœud  $n_0$  de  $\delta^\Phi$ , et un isomorphisme  $\phi : \delta(n_0) \rightarrow B$  de  $\mathcal{C}_0$ . Soit  $\bar{\beta} = \text{Subst\_Obj\_Iso}(\bar{\delta}, n_0, B, \phi)$ . On construit deux flèches de  $\text{DIAGR}(\mathcal{C}_0)$ ,  $\bar{\sigma} : \bar{\delta} \rightarrow \bar{\beta}$  et  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\delta}$  telles que  $\bar{\sigma} \circ \bar{\tau} \approx \overline{id_{\bar{\beta}}}$  et  $\bar{\tau} \circ \bar{\sigma} \approx \overline{id_{\bar{\delta}}}$ .

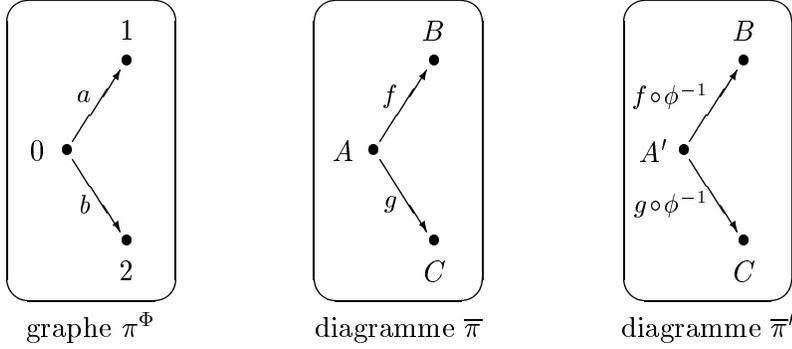
- On définit  $\bar{\sigma} : \bar{\delta} \rightarrow \bar{\beta}$  de la façon suivante.
  - Le morphisme généralisé de graphes  $\sigma^\Phi : \delta^\Phi \rightarrow \beta^\Phi$  est l'identité.
    - $\sigma^\Phi = id_{\delta^\Phi}$ .
  - La transformation naturelle généralisée  $\sigma : \delta \rightsquigarrow \beta \circ \sigma^\Phi$  est définie par :
    - $\forall n \in \text{Nœud}(\delta^\Phi)$  tel que  $n \neq n_0, \sigma_n = id_{\delta(n)}$ ;
    - $\sigma_{n_0} = \phi$ .
 Cela définit bien une transformation naturelle généralisée car pour tout arc  $a : n \rightarrow n' \in \text{Arc}(\delta^\Phi)$ , on a  $\sigma_{n'} \circ \delta(a) = \beta(a) \circ \sigma_n$ .
- On définit  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\delta}$  de la façon suivante.
  - Le morphisme généralisé de graphes  $\tau^\Phi : \beta^\Phi \rightarrow \delta^\Phi$  est l'identité.
    - $\tau^\Phi = id_{\beta^\Phi}$ .
  - La transformation naturelle généralisée  $\tau : \beta \rightsquigarrow \delta \circ \tau^\Phi$  est définie par :
    - $\forall n \in \text{Nœud}(\beta^\Phi)$  tel que  $n \neq n_0, \tau_n = id_{\beta(n)}$ ;
    - $\tau_{n_0} = \phi^{-1}$ .
 Cela définit bien une transformation naturelle généralisée car pour tout arc  $a : n \rightarrow n' \in \text{Arc}(\beta^\Phi)$ , on a  $\tau_{n'} \circ \beta(a) = \delta(a) \circ \tau_n$ .
- On vérifie que  $\bar{\sigma} \circ \bar{\tau} = \overline{id_{\bar{\beta}}}$ .
- On vérifie également que  $\bar{\tau} \circ \bar{\sigma} = \overline{id_{\bar{\delta}}}$ .

Par conséquent  $\bar{\delta} \cong \bar{\beta}$  dans la catégorie  $\text{diagr}(\mathcal{C}_0)$ . En fait, les diagrammes  $\bar{\delta}$  et  $\bar{\beta}$  sont ici également isomorphes dans la catégorie  $\text{DIAGR}(\mathcal{C}_0)$ .

□

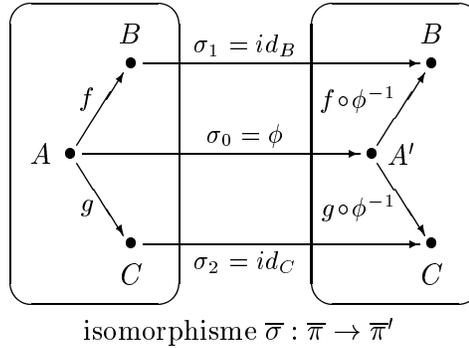
**Exemple 5.1** Considérons le diagramme  $\bar{\pi}$ , construit sur le graphe  $\pi^\Phi$ . Soit  $\phi : A \rightarrow A'$  un isomorphisme de  $\mathcal{C}_0$ . Alors, nous pouvons considérer le diagramme

$$\bar{\pi}' = \text{Subst\_Obj\_Iso}(\bar{\pi}, 0, A', \phi).$$



L'isomorphisme  $\bar{\sigma} : \bar{\pi} \rightarrow \bar{\pi}'$  est composé du morphisme de graphes  $\sigma^\Phi = id_{\pi^\Phi}$  et de la transformation naturelle généralisée  $\sigma : \pi \rightsquigarrow \pi' \circ \sigma^\Phi$  définie par

$$\sigma_0 = \phi, \quad \sigma_1 = id_B, \quad \sigma_2 = id_C.$$



### 5.1.2 Suppression d'un arc

Soit un diagramme  $\bar{\delta}$  sur  $\mathcal{C}_0$ . Intuitivement, *supprimer l'arc*  $a_0 : n_0 \rightarrow n'_0$  du diagramme  $\bar{\delta}$  consiste à retirer l'arc  $a_0$  du graphe  $\delta^\Phi$  et à conserver les mêmes étiquettes pour les arcs et les nœuds restants.

**Définition 5.4** (Suppr\_Arc) Soit un diagramme  $\bar{\delta}$  et un arc  $a_0 : n_0 \rightarrow n'_0 \in \text{Arc}(\delta^\Phi)$ . Le diagramme

$$\bar{\beta} = \text{Suppr\_Arc}(\bar{\delta}, a_0)$$

est défini de la façon suivante.

- Le graphe  $\beta^\Phi$  est le graphe  $\delta^\Phi$  auquel on retire l'arc  $a_0$ .
  - $\text{Nœud}(\beta^\Phi) = \text{Nœud}(\delta^\Phi)$ ;
  - $\text{Arc}(\beta^\Phi) = \text{Arc}(\delta^\Phi) - \{a_0\}$ .

- Le foncteur  $\beta : \beta^\Phi \rightarrow \mathcal{C}_0$  est la restriction de  $\delta : \delta^\Phi \rightarrow \mathcal{C}_0$  à  $\beta^\Phi$ .
  - $\forall n \in \text{Nœud}(\beta^\Phi), \beta(n) = \delta(n)$  ;
  - $\forall a : n \rightarrow n' \in \text{Arc}(\beta^\Phi), \beta(a) = \delta(a)$ .

Dans le cas général, la transformation  $\text{Suppr\_Arc}$  n'est évidemment pas stable par isomorphisme.

**Exemple 5.2** Considérons le diagramme  $\bar{\kappa}$ , construit sur le graphe  $\kappa^\Phi$ .



Posons  $\bar{\kappa}' = \text{Suppr\_Arc}(\bar{\kappa}, b)$ .



Dans le cas général, les diagrammes  $\bar{\kappa}$  et  $\bar{\kappa}'$  ne sont pas isomorphes dans la catégorie  $\text{diagr}(\mathcal{C}_0)$ .

Par contre, la suppression de certains arcs conduit à des diagrammes isomorphes. En particulier, on peut d'une part supprimer les *boucles identités* et d'autre part l'un des arcs d'un *doublet*.

Une *boucle* sur un graphe  $\delta^\Phi$  est un arc  $a : n \rightarrow n$  de  $\delta^\Phi$  qui a le même nœud pour source et but.

**Définition 5.5** (Boucle) Soit un graphe  $\delta^\Phi$ . Un arc  $a$  de  $\delta^\Phi$  est une *boucle* si et seulement si  $\text{Source}(a) = \text{But}(a)$ .

Une *boucle identité* sur un diagramme  $\bar{\delta}$  est une boucle de  $\delta^\Phi$  étiquetée par une flèche identité.

**Définition 5.6** (Boucle identité) Soit un diagramme  $\bar{\delta}$  sur  $\mathcal{C}_0$ . Une *boucle identité* sur  $\bar{\delta}$  est une boucle  $a : n \rightarrow n$  de  $\delta^\Phi$  telle que  $\delta(a) = \text{id}_{\delta(n)}$ .

**Proposition 5.2** Soit un diagramme  $\bar{\delta}$  et une boucle identité  $a_0 : n_0 \rightarrow n_0$  sur  $\bar{\delta}$ . Alors, dans la catégorie  $\text{diagr}(\mathcal{C}_0)$ ,

$$\text{Suppr\_Arc}(\bar{\delta}, a_0) \cong \bar{\delta}.$$

*Preuve.* Considérons un diagramme  $\bar{\delta}$  et une boucle identité  $a_0 : n_0 \rightarrow n_0$  de  $\bar{\delta}$ . Soit  $\bar{\beta} = \text{Suppr\_Arc}(\bar{\delta}, a_0)$ . On construit deux flèches de  $\text{DIAGR}(\mathcal{C}_0)$ ,  $\bar{\sigma} : \bar{\delta} \rightarrow \bar{\beta}$  et  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\delta}$  telles que  $\bar{\sigma} \circ \bar{\tau} \approx \overline{id_{\bar{\beta}}}$  et  $\bar{\tau} \circ \bar{\sigma} \approx \overline{id_{\bar{\delta}}}$ .

- On définit  $\bar{\sigma} : \bar{\delta} \rightarrow \bar{\beta}$  de la façon suivante.

- Morphisme généralisé de graphes  $\sigma^\Phi : \delta^\Phi \rightarrow \beta^\Phi$ .
  - $\forall n \in \text{Nœud}(\delta^\Phi)$ ,  $\sigma^\Phi(n) = n$ ;
  - $\forall a \in \text{Arc}(\delta^\Phi)$ ,  $a \neq a_0$ ,  $\sigma^\Phi(a) = a$ ;
  - $\sigma^\Phi(a_0) = 0_{n_0}$  (zigzag nul de source et but  $n_0$ ).
- Transformation naturelle généralisée  $\sigma : \delta \rightsquigarrow \beta \circ \sigma^\Phi$ .
  - $\forall n \in \text{Nœud}(\delta^\Phi)$ ,  $\sigma_n = id_{\delta(n)}$ .

Cela définit bien une transformation naturelle généralisée car  $\delta(a_0) = id_{\delta(n_0)}$ .

- On définit  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\delta}$  de la façon suivante.

- Morphisme généralisé de graphes  $\tau^\Phi : \beta^\Phi \rightarrow \delta^\Phi$ .
  - $\forall n \in \text{Nœud}(\beta^\Phi)$ ,  $\tau^\Phi(n) = n$ ;
  - $\forall a \in \text{Arc}(\beta^\Phi)$ ,  $\tau^\Phi(a) = a$ .
- Transformation naturelle généralisée  $\tau : \beta \rightsquigarrow \delta \circ \tau^\Phi$ .
  - $\forall n \in \text{Nœud}(\beta^\Phi)$ ,  $\tau_n = id_{\beta(n)}$ .

- On a  $\bar{\sigma} \circ \bar{\tau} = \overline{id_{\bar{\beta}}}$ .

- On vérifie également que  $\bar{\tau} \circ \bar{\sigma} \approx \overline{id_{\bar{\delta}}}$ .

Par conséquent, les diagrammes  $\bar{\delta}$  et  $\bar{\beta}$  sont isomorphes dans la catégorie  $\text{diagr}(\mathcal{C}_0)$ .  $\square$

**Exemple 5.3** Soit  $A$  un objet de  $\mathcal{C}_0$ . Considérons le diagramme  $I_{\mathcal{C}_0}(A)$ , construit sur le graphe  $1^\Phi$ . Rappelons que le graphe  $1^\Phi$  contient un seul nœud appelé  $*$ .

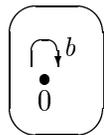


graphe  $1^\Phi$



diagramme  $I_{\mathcal{C}_0}(A)$

Considérons le graphe  $\rho^\Phi$  qui contient un nœud appelé  $0$  et un arc  $b : 0 \rightarrow 0$ . Étant donné un objet  $A$  et une flèche  $f : A \rightarrow A$  de  $\mathcal{C}_0$ , on note  $\bar{\rho}(A, f)$  le diagramme construit sur le graphe  $\rho^\Phi$ , qui associe au nœud  $0$  l'objet  $A$  et à l'arc  $b$  la flèche  $f$ . Nous considérons ici le diagramme  $\bar{\rho}(A, id_A)$ .



graphe  $\rho^\Phi$

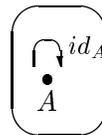


diagramme  $\bar{\rho}(A, id_A)$

Les diagrammes  $I_{\mathcal{C}_0}(A)$  et  $\bar{\rho}(A, id_A)$  sont isomorphes, car

$$I_{\mathcal{C}_0}(A) = \text{Suppr\_Arc}(\bar{\rho}(A, id_A), b)$$

et  $b$  est une boucle identité sur  $\bar{\rho}(A, id_A)$ .

Un doublet sur un diagramme  $\bar{\delta}$  est un couple d'arcs qui ont tous les deux la même source et le même but, et étiquetés par la même flèche de  $\mathcal{C}_0$ .

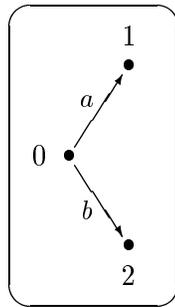
**Définition 5.7** (Doublet) Soit un diagramme  $\bar{\delta}$  sur  $\mathcal{C}_0$ . Un *doublet* sur  $\bar{\delta}$  est un couple d'arcs  $(a_0, a_1)$  tel que

$$\text{Source}(a_0) = \text{Source}(a_1), \text{ But}(a_0) = \text{But}(a_1) \text{ et } \delta(a_0) = \delta(a_1).$$

**Exemple 5.4**

- Soit deux objets  $A$  et  $B$  et une flèche  $f : A \rightarrow B$  de  $\mathcal{C}_0$ . Considérons le diagramme  $\bar{\pi}_1$ , construit sur le graphe  $\pi^\Phi$ , défini par

$$\begin{aligned} \pi(0) &= A; \\ \pi(1) &= \pi(2) = B; \\ \pi(a) &= \pi(b) = f. \end{aligned}$$



graphe  $\pi^\Phi$

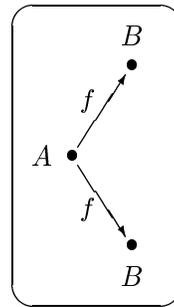
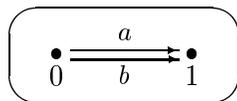


diagramme  $\bar{\pi}_1$

Le couple  $(a, b)$  n'est pas un doublet sur  $\bar{\pi}_1$ , bien que les arcs  $a$  et  $b$  soient étiquetés par la même flèche  $f$  de  $\mathcal{C}_0$ , car ces arcs n'ont pas le même but.

- Considérons le diagramme  $\bar{\kappa}_1$ , construit sur le graphe  $\kappa^\Phi$ .



graphe  $\kappa^\Phi$

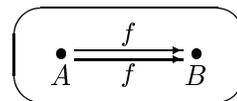


diagramme  $\bar{\kappa}_1$

Le couple  $(a, b)$  est un doublet de  $\bar{\kappa}_1$ .

**Proposition 5.3** Soit un diagramme  $\bar{\delta}$  sur  $\mathcal{C}_0$  et un doublet  $(a_0, a_1)$  de  $\bar{\delta}$ . Alors, dans la catégorie  $\text{diagr}(\mathcal{C}_0)$ ,

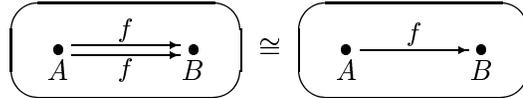
$$\text{Suppr\_Arc}(\bar{\delta}, a_0) \cong \bar{\delta}.$$

*Preuve.* Soit un diagramme  $\bar{\delta}$  et un doublet  $(a_0, a_1)$  de  $\bar{\delta}$ . Soit  $\bar{\beta} = \text{Suppr\_Arc}(\bar{\delta}, a_0)$ . On construit deux flèches de  $\text{DIAGR}(\mathcal{C}_0)$ ,  $\bar{\sigma} : \bar{\delta} \rightarrow \bar{\beta}$  et  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\delta}$  telles que  $\bar{\sigma} \circ \bar{\tau} \approx \text{id}_{\bar{\beta}}$  et  $\bar{\tau} \circ \bar{\sigma} \approx \text{id}_{\bar{\delta}}$ .

- On définit  $\bar{\sigma} : \bar{\delta} \rightarrow \bar{\beta}$  de la façon suivante.
  - Morphisme généralisé de graphes  $\sigma^\Phi : \delta^\Phi \rightarrow \beta^\Phi$ .
    - $\forall n \in \text{Nœud}(\delta^\Phi), \sigma^\Phi(n) = n$ ;
    - $\forall a \in \text{Arc}(\delta^\Phi), a \neq a_0, \sigma^\Phi(a) = a$ ;
    - $\sigma^\Phi(a_0) = a_1$ .
  - Transformation naturelle généralisée  $\sigma : \delta \rightsquigarrow \beta \circ \sigma^\Phi$ .
    - $\forall n \in \text{Nœud}(\delta^\Phi), \sigma_n = \text{id}_{\delta(n)}$ .
 Cela définit bien une transformation naturelle généralisée. En effet, on a  $\delta(a_0) = \delta(a_1)$ .
- On définit  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\delta}$  de la façon suivante.
  - Morphisme généralisé de graphes  $\tau^\Phi : \beta^\Phi \rightarrow \delta^\Phi$ .
    - $\forall n \in \text{Nœud}(\beta^\Phi), \tau^\Phi(n) = n$ ;
    - $\forall a \in \text{Arc}(\beta^\Phi), \tau^\Phi(a) = a$ .
  - Transformation naturelle généralisée  $\tau : \beta \rightsquigarrow \delta \circ \tau^\Phi$ .
    - $\forall n \in \text{Nœud}(\beta^\Phi), \tau_n = \text{id}_{\beta(n)}$ .
- On a  $\bar{\sigma} \circ \bar{\tau} = \text{id}_{\bar{\beta}}$ .
- On vérifie également que  $\bar{\tau} \circ \bar{\sigma} \approx \text{id}_{\bar{\delta}}$ .

Par conséquent, les diagrammes  $\bar{\delta}$  et  $\bar{\beta}$  sont isomorphes dans la catégorie  $\text{diagr}(\mathcal{C}_0)$ .  $\square$

**Exemple 5.5** Les diagrammes suivants sont isomorphes dans  $\text{diagr}(\mathcal{C}_0)$ .



### 5.1.3 Ajout d'un arc étiqueté

Étant donné un diagramme  $\bar{\delta}$ , l'opération d'*ajout d'arc étiqueté* consiste à ajouter un arc au diagramme sous-jacent  $\delta^\Phi$  et à étiqueter cet arc par une flèche de  $\mathcal{C}_0$ .

**Définition 5.8** (*Ajout\_Arc*) Soit un diagramme  $\bar{\delta}$ , deux nœuds  $n_0$  et  $n_1$  de  $\delta^\Phi$  et une flèche  $f : \delta(n_0) \rightarrow \delta(n_1)$  de  $\mathcal{C}_0$ . L'ajout d'un arc de  $n_0$  vers  $n_1$  étiqueté par  $f$  dans le diagramme  $\bar{\delta}$  est le diagramme

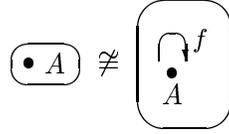
$$\bar{\beta} = \text{Ajout\_Arc}(\bar{\delta}, n_0, n_1, f)$$

défini de la façon suivante.

- Le graphe  $\beta^\Phi$  est le graphe  $\delta^\Phi$  auquel on ajoute un nouvel arc  $a_x : n_0 \rightarrow n_1$ .
  - $\text{Nœud}(\beta^\Phi) = \text{Nœud}(\delta^\Phi)$ ;
  - $\text{Arc}(\beta^\Phi) = \text{Arc}(\delta^\Phi) \cup \{a_x\}$ , où  $a_x \notin \text{Arc}(\delta^\Phi)$ , avec  $\text{Source}(a_x) = n_0$  et  $\text{But}(a_x) = n_1$ .
- Le foncteur  $\beta : \beta^\Phi \rightarrow \mathcal{C}_0$  étend le foncteur  $\delta$  en associant à l'arc  $a_x$  la flèche  $f$ .
  - Action sur les nœuds.
    - $\forall n \in \text{Nœud}(\beta^\Phi), \beta(n) = \delta(n)$ .
  - Action sur les arcs.
    - $\forall a \in \text{Arc}(\beta^\Phi), a \neq a_x, \beta(a) = \delta(a)$ ;
    - $\beta(a_x) = f$ .

La transformation `Ajout_Arc` n'est évidemment pas stable par isomorphisme dans le cas général.

**Exemple 5.6** Soit un objet  $A$  et une flèche  $f : A \rightarrow A$  dans  $\mathcal{C}_0$ . Dans le cas général, les diagrammes  $I_{\mathcal{C}_0}(A)$  et  $\bar{\rho}(A, f)$  ne sont pas isomorphes dans  $\text{diagr}(\mathcal{C}_0)$ .



**Proposition 5.4** (*Ajout de compositions*) Soit un diagramme  $\bar{\delta}$  sur  $\mathcal{C}_0$ . Soit deux arcs  $a_0 : n_0 \rightarrow n_1$  et  $a_1 : n_1 \rightarrow n_2$  de  $\delta^\Phi$ . Alors, dans la catégorie  $\text{diagr}(\mathcal{C}_0)$ ,

$$\text{Ajout\_Arc}(\bar{\delta}, n_0, n_2, \delta(a_1) \circ \delta(a_0)) \cong \bar{\delta}.$$

*Preuve.* Soit un diagramme  $\bar{\delta}$  et deux arcs  $a_0 : n_0 \rightarrow n_1$  et  $a_1 : n_1 \rightarrow n_2$  de  $\delta^\Phi$ . Posons  $\bar{\beta} = \text{Ajout\_Arc}(\bar{\delta}, n_0, n_2, \delta(a_1) \circ \delta(a_0))$ . On construit deux flèches de  $\text{DIAGR}(\mathcal{C}_0)$ ,  $\bar{\sigma} : \bar{\delta} \rightarrow \bar{\beta}$  et  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\delta}$  telles que  $\bar{\sigma} \circ \bar{\tau} \approx \text{id}_{\bar{\beta}}$  et  $\bar{\tau} \circ \bar{\sigma} \approx \text{id}_{\bar{\delta}}$ .

- On définit  $\bar{\sigma} : \bar{\delta} \rightarrow \bar{\beta}$  de la façon suivante.
  - Morphisme généralisé de graphes  $\sigma^\Phi : \delta^\Phi \rightarrow \beta^\Phi$ .
    - $\forall n \in \text{Nœud}(\delta^\Phi), \sigma^\Phi(n) = n$ ;
    - $\forall a \in \text{Arc}(\delta^\Phi), \sigma^\Phi(a) = a$ .
  - Transformation naturelle généralisée  $\sigma : \delta \rightsquigarrow \beta \circ \sigma^\Phi$ .
    - $\forall n \in \text{Nœud}(\delta^\Phi), \sigma_n = \text{id}_{\delta(n)}$ .
- On définit  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\delta}$  de la façon suivante.
  - Morphisme généralisé de graphes  $\tau^\Phi : \beta^\Phi \rightarrow \delta^\Phi$ .
    - $\forall n \in \text{Nœud}(\beta^\Phi), \tau^\Phi(n) = n$ ;

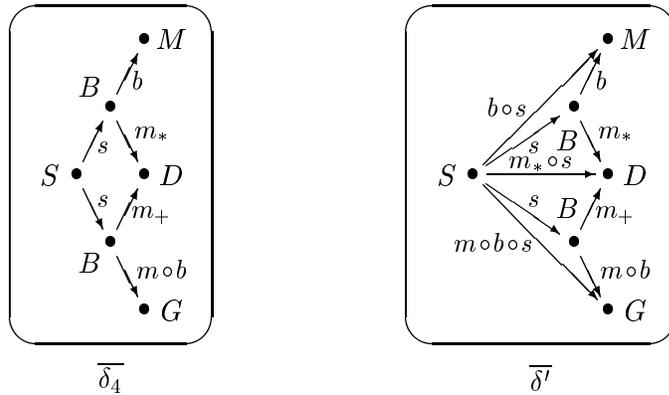
- $\forall a \in \text{Arc}(\beta^\Phi), a \neq a_x, \tau^\Phi(a) = a;$
- $\tau^\Phi(a_x) = n_0 \xrightarrow{a_0} n_1 \xrightarrow{a_1} n_2$  (zigzag de longueur de 2).
- Transformation naturelle généralisée  $\tau : \beta \rightsquigarrow \delta \circ \tau^\Phi.$ 
  - $\forall n \in \text{Nœud}(\beta^\Phi), \tau_n = \text{id}_{\beta(n)}.$

Cela définit bien une transformation naturelle généralisée. En effet, on a  $\beta(a_x) = \delta(a_1) \circ \delta(a_0).$

- On a  $\overline{\tau} \circ \overline{\sigma} = \overline{id_{\overline{\delta}}}$ .
- On vérifie également que  $\overline{\sigma} \circ \overline{\tau} \approx \overline{id_{\overline{\beta}}}$ .

□

**Exemple 5.7** Considérons le diagramme  $\overline{\delta}_4$  sur  $\mathcal{C}_0$ . Nous avons vu une première fois ce diagramme dans le chapitre 1, page 49, et une seconde fois dans le chapitre 4, page 164. Nous pouvons ajouter trois arcs étiquetés dans  $\overline{\delta}_4$ : un arc étiqueté par  $b \circ s$  entre (les nœuds étiquetés par)  $S$  et  $M$ , un arc étiqueté par  $m_* \circ s$  entre  $S$  et  $D$  et un arc étiqueté par  $m \circ b \circ s$  entre  $S$  et  $G$ . Nous obtenons ainsi un diagramme  $\overline{\delta}'$ . Nous n'avons pas ajouté d'arc étiqueté par  $m_+ \circ s$  entre  $S$  et  $D$  car il existe déjà un arc étiqueté par  $m_* \circ s$  entre ces deux nœuds et, dans la catégorie  $\mathcal{C}_0$ , on a l'égalité  $m_+ \circ s = m_* \circ s$ .



**Proposition 5.5** (Ajout de factorisations à droite) Soit un diagramme  $\overline{\delta}$  sur  $\mathcal{C}_0$ . Soit deux arcs  $a_1 : n_1 \rightarrow n_0$  et  $a_2 : n_2 \rightarrow n_0$  de  $\delta^\Phi$ . Soit une flèche  $h : \delta(n_1) \rightarrow \delta(n_2)$  de  $\mathcal{C}_0$  telle que  $\delta(a_2) \circ h = \delta(a_1)$ . Alors, dans la catégorie  $\text{diagr}(\mathcal{C}_0)$ ,

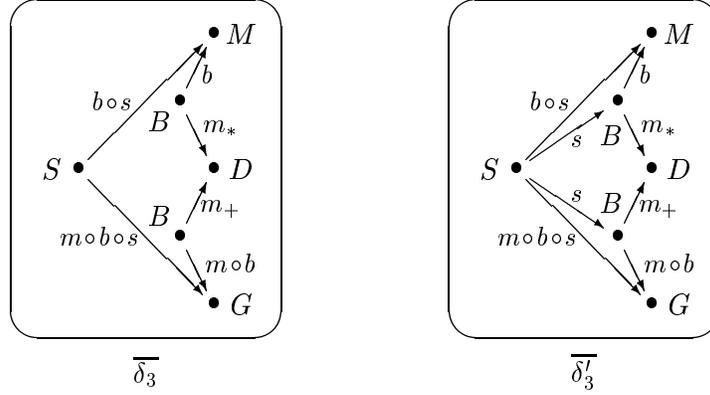
$$\text{Ajout\_Arc}(\overline{\delta}, n_1, n_2, h) \cong \overline{\delta}.$$

*Preuve.* La preuve est similaire à la preuve de la proposition 5.4. On pose

$$\tau^\Phi(a_x) = n_1 \xrightarrow{a_1} n_0 \xleftarrow{a_2} n_2.$$

Cela définit bien une transformation naturelle généralisée  $\tau : \beta \rightsquigarrow \delta \circ \tau^\Phi$ . En effet,  $\delta(a_2) \circ h = \delta(a_1)$ . □

**Exemple 5.8** Considérons le diagramme  $\overline{\delta}_3$ , vu une première fois dans le chapitre 1, page 49, et revu dans le chapitre 4, page 160. Nous obtenons le diagramme  $\overline{\delta}'_3$  à partir du diagramme  $\overline{\delta}_3$  en ajoutant deux factorisations à droite.



#### 5.1.4 Contraction d'un arc identité

Soit un diagramme  $\overline{\delta}$ . Un *arc identité* du diagramme  $\overline{\delta}$  est un arc de  $\delta^\Phi$  étiqueté par une flèche identité.

**Définition 5.9** (Arc identité) Soit un diagramme  $\overline{\delta}$  sur  $\mathcal{C}_0$ . Un arc  $a_0 : n_0 \rightarrow n_1$  du graphe  $\delta^\Phi$  est un *arc identité* si et seulement si  $\delta(n_0) = \delta(n_1)$  et  $\delta(a_0) = id_{\delta(n_0)}$ .

Intuitivement, *contracter un arc identité*  $a_0 : n_0 \rightarrow n_1$  dans un diagramme  $\overline{\delta}$  consiste à fusionner les nœuds  $n_0$  et  $n_1$  et à supprimer l'arc  $a_0$ .

**Définition 5.10** (Contract\_Ld) Soit un diagramme  $\overline{\delta}$  et un arc identité  $a_0 : n_0 \rightarrow n_1$  de  $\overline{\delta}$ , tel que les nœuds  $n_0$  et  $n_1$  sont distincts. Le diagramme

$$\overline{\beta} = \text{Contract\_ld}(\overline{\delta}, a_0)$$

est défini de la façon suivante.

- Graphe  $\beta^\Phi$ .
  - $\text{Nœud}(\beta^\Phi) = \text{Nœud}(\delta^\Phi) - \{n_1\}$
  - L'ensemble  $\text{Arc}(\beta^\Phi)$  est défini à partir de l'ensemble  $\text{Arc}(\delta^\Phi)$  de la façon suivante. Soit un arc  $a : n \rightarrow n'$  de  $\delta^\Phi$ .
    - $n \neq n_1$  et  $n' \neq n_1 \Rightarrow a : n \rightarrow n' \in \text{Arc}(\beta^\Phi)$ ;
    - $n = n_1$  et  $n' \neq n_1 \Rightarrow a : n_0 \rightarrow n' \in \text{Arc}(\beta^\Phi)$ ;
    - $n \neq n_1$  et  $n' = n_1$  et  $a \neq a_0 \Rightarrow a : n \rightarrow n_0 \in \text{Arc}(\beta^\Phi)$ ;
    - $n = n_1$  et  $n' = n_1 \Rightarrow a : n_0 \rightarrow n_0 \in \text{Arc}(\beta^\Phi)$ .
- Foncteur  $\beta : \beta^\Phi \rightarrow \mathcal{C}_0$ .

- $\forall n \in \text{Nœud}(\beta^\Phi), \beta(n) = \delta(n)$  ;
- $\forall a \in \text{Arc}(\beta^\Phi), \beta(a) = \delta(a)$ .

Nous avons supposé que les nœuds  $n_0$  et  $n_1$  sont distincts, car si  $n_0 = n_1$ , cela n'a pas de sens de fusionner ces deux nœuds. De plus, si  $n_0 = n_1$ , alors  $a_0$  est une *boucle identité*, qui peut être supprimée par l'opération `Suppr_Arc` (cf. proposition 5.2).

**Proposition 5.6** *La transformation `Contract_Id` est stable par isomorphisme.*

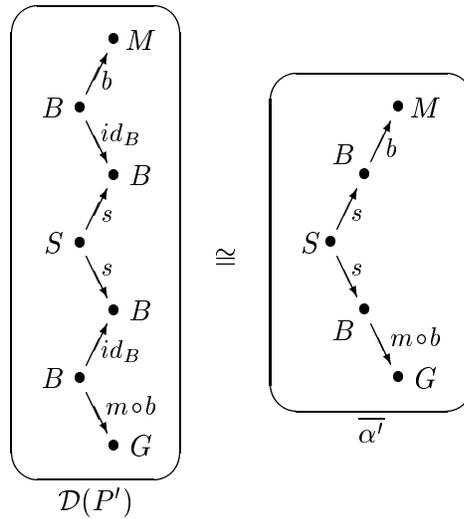
*Preuve.* Soit un diagramme  $\bar{\delta}$  et un arc identité  $a_0 : n_0 \rightarrow n_1$  de  $\bar{\delta}$ , tel que  $n_0 \neq n_1$ . Posons  $\bar{\beta} = \text{Contract\_Id}(\bar{\delta}, a_0)$ . On construit deux flèches de  $\text{DIAGR}(\mathcal{C}_0)$ ,  $\bar{\sigma} : \bar{\delta} \rightarrow \bar{\beta}$  et  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\delta}$  telles que  $\bar{\sigma} \circ \bar{\tau} \approx \overline{id_{\bar{\beta}}}$  et  $\bar{\tau} \circ \bar{\sigma} \approx \overline{id_{\bar{\delta}}}$ .

- On définit  $\bar{\sigma} : \bar{\delta} \rightarrow \bar{\beta}$  de la façon suivante.
  - Morphisme généralisé de graphes  $\sigma^\Phi : \delta^\Phi \rightarrow \beta^\Phi$ .
    - $\forall n \in \text{Nœud}(\delta^\Phi), n \neq n_1, \sigma^\Phi(n) = n$  ;
    - $\sigma^\Phi(n_1) = n_0$  ;
    - $\forall a \in \text{Arc}(\delta^\Phi), a \neq a_0, \sigma^\Phi(a) = a$  ;
    - $\sigma^\Phi(a_0) = 0_{n_0}$  (zigzag nul ayant pour source et but le nœud  $n_0$  de  $\beta^\Phi$ ).
  - Transformation naturelle généralisée  $\sigma : \delta \rightsquigarrow \beta \circ \sigma^\Phi$ .
    - $\forall n \in \text{Nœud}(\delta^\Phi), \sigma_n = id_{\delta(n)}$ .
 Cela définit bien une transformation naturelle généralisée car  $\delta(a_0) = id_{\delta(n_0)}$ .
- On définit  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\delta}$  de la façon suivante.
  - Morphisme généralisé de graphes  $\tau^\Phi : \beta^\Phi \rightarrow \delta^\Phi$ .
    - $\forall n \in \text{Nœud}(\beta^\Phi), \tau^\Phi(n) = n$  ;
    - $\forall a \in \text{Arc}(\beta^\Phi), \tau^\Phi(a) = a$ .
  - Transformation naturelle généralisée  $\tau : \beta \rightsquigarrow \delta \circ \tau^\Phi$ .
    - $\forall n \in \text{Nœud}(\beta^\Phi), \tau_n = id_{\beta(n)}$ .
- On a  $\bar{\sigma} \circ \bar{\tau} = \overline{id_{\bar{\beta}}}$ .
- On vérifie également que  $\bar{\tau} \circ \bar{\sigma} \approx \overline{id_{\bar{\delta}}}$ .

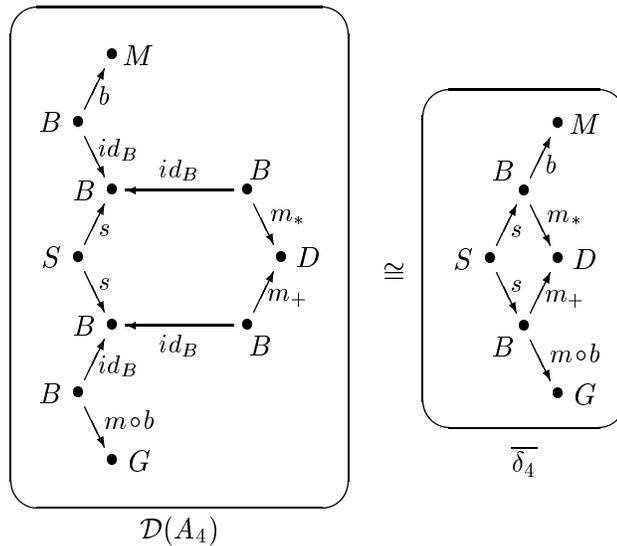
□

**Exemple 5.9**

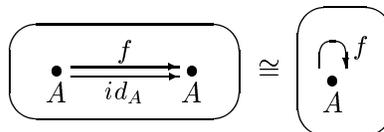
1. Les diagrammes  $\mathcal{D}(P')$  et  $\overline{\alpha'}$  (cf. chapitre 4, page 162) sont isomorphes dans la catégorie  $\text{diagr}(\mathcal{C}_0)$ .



2. Les diagrammes  $\mathcal{D}(A_4)$  et  $\overline{\delta_4}$  (cf. chapitre 4, page 163) sont isomorphes dans la catégorie  $\text{diagr}(\mathcal{C}_0)$ .



3. Les diagrammes suivants sont isomorphes dans la catégorie  $\text{diagr}(\mathcal{C}_0)$ .



Cet exemple montre qu'à partir d'un diagramme dont le graphe sous-jacent n'est pas cyclique, la contraction d'un arc identité peut produire un diagramme dont le graphe sous-jacent est cyclique.

## 5.2 Catégorie $\mathcal{C}_0$ finie et sans cycle

Dans cette partie, nous nous intéressons à la détection d'isomorphismes lorsque la catégorie  $\mathcal{C}_0$  est finie et sans cycle. Nous commençons par définir un *cycle* dans une catégorie.

**Définition 5.11** (Cycle) Étant donné une catégorie  $\mathcal{C}$ , un *cycle* dans  $\mathcal{C}$  est une flèche  $f : A \rightarrow A$  ayant même domaine et codomaine  $A$  et telle que  $f \neq id_A$ .

Nous dirons qu'une catégorie  $\mathcal{C}$  ne comporte pas de cycle si et seulement si pour toute flèche  $f : A \rightarrow A$  de  $\mathcal{C}$ ,  $f = id_A$ .

### 5.2.1 Hypothèses

Nous supposons que la catégorie de base  $\mathcal{C}_0$  est finie et ne comporte pas de cycle. Cette hypothèse est compatible avec le langage de spécification LPG, puisque d'une part on ne peut déclarer qu'un nombre fini de spécifications et morphismes de spécifications, et d'autre part, la déclaration d'un morphisme de spécifications ayant la même spécification pour domaine et codomaine est syntaxiquement interdite.

D'autre part, nous supposons qu'il n'existe pas d'isomorphisme entre objets dans la catégorie de base  $\mathcal{C}_0$ . Cette hypothèse est justifiée par la transformation de substitution par objet isomorphe (proposition 5.1). En effet, supposons qu'on a, au départ, une catégorie  $\mathcal{X}_0$  qui ne possède pas cette propriété. Pour chaque classe d'isomorphisme entre objets de  $\mathcal{X}_0$ , on choisit un représentant. On considère ensuite la sous-catégorie pleine  $\mathcal{C}_0$  de  $\mathcal{X}_0$  ayant pour objets ces représentants. La catégorie  $\mathcal{C}_0$  satisfait alors la propriété suivante :

$$\text{pour tout couple d'objets } (A, B) \text{ de } \mathcal{C}_0, \quad A \cong B \Rightarrow A = B.$$

Tout diagramme sur  $\mathcal{X}_0$  peut être transformé en un diagramme sur  $\mathcal{C}_0$  en utilisant la proposition 5.1.

Nous faisons également l'hypothèse qu'on sait détecter une égalité entre deux morphismes de spécifications de base, c'est-à-dire que l'égalité est décidable dans  $\mathcal{C}_0$ . Enfin nous supposons que toutes les factorisations à droite sont connues.

### Hypothèses

1.  $\mathcal{C}_0$  est finie.
2.  $\mathcal{C}_0$  ne comporte pas de cycle.
3. Pour tout couple d'objets  $(A, B)$  de  $\mathcal{C}_0$ ,  $A \cong B \Rightarrow A = B$ .
4. L'égalité sur  $\mathcal{C}_0$  est décidable. Soit deux flèches  $f, g : A \rightarrow B$  de  $\mathcal{C}_0$ . On sait si  $f = g$  ou  $f \neq g$ .
5. On sait effectuer des factorisations à droite dans  $\mathcal{C}_0$ . Étant donné deux flèches  $f : A \rightarrow B$  et  $g : C \rightarrow B$ , on connaît l'ensemble des flèches  $h : A \rightarrow C$  telles que  $f = g \circ h$ .

**Remarque 5.1** Les hypothèses 1 et 2 sont distinctes. En effet, il existe des catégories infinies qui ne comportent pas de cycle, et il existe des catégories finies qui comportent des cycles.

La combinaison des hypothèses 2 et 3 est très forte, comme le montre le lemme suivant.

**Lemme 5.1** *Soit deux objets  $A$  et  $B$  ainsi que deux flèches  $f : A \rightarrow B$  et  $g : B \rightarrow A$  dans  $\mathcal{C}_0$ . Alors,  $A = B$  et  $f = g = id_A$ .*

*Preuve.* D'après l'hypothèse 2,  $g \circ f = id_A$  et  $f \circ g = id_B$ , d'où  $A \cong B$ . D'après l'hypothèse 3, on a donc  $A = B$ . On en déduit que  $f$  et  $g$  sont des flèches de  $A$  vers  $A$ . Par conséquent, en utilisant de nouveau l'hypothèse 2, on obtient  $f = g = id_A$ .  $\square$

## 5.2.2 Fonction de complétion

Dans ce paragraphe, nous détaillons la fonction de *complétion* d'un diagramme. Compléter un diagramme  $\bar{\alpha}$  sur  $\mathcal{C}_0$  consiste à contracter les arcs identités, supprimer les boucles identités, supprimer les doublets, ajouter les compositions d'arcs et les factorisations à droite.

Fonction Complétion ( $\bar{\alpha}$  : diagramme) : diagramme

$\bar{\delta}, \bar{\delta}_0$  : diagramme ;

Début

$\bar{\delta} := \bar{\alpha}$  ;

Répéter

$\bar{\delta}_0 := \bar{\delta}$  ;

Pour tout arc identité  $a$  de  $\bar{\delta}$  : (1)

$\bar{\delta} := \text{Contract\_Id}(\bar{\delta}, a)$  ;

Pour toute boucle identité  $a$  de  $\bar{\delta}$  : (2)

$\bar{\delta} := \text{Suppr\_Arc}(\bar{\delta}, a)$  ;

Pour tout doublet  $(a_0, a_1)$  de  $\bar{\delta}$  : (3)

$\bar{\delta} := \text{Suppr\_Arc}(\bar{\delta}, a_0)$  ;

Pour tout couple d'arcs  $(a_0 : n_0 \rightarrow n_1, a_1 : n_1 \rightarrow n_2)$  de  $\bar{\delta}$  tel que (4)  
il n'existe pas d'arc  $a : n_0 \rightarrow n_2$  étiqueté par  $\delta(a_1) \circ \delta(a_0)$  dans  $\bar{\delta}$ ,

$\bar{\delta} := \text{Ajout\_Arc}(\bar{\delta}, n_0, n_2, \delta(a_1) \circ \delta(a_0))$  ;

Pour tout couple d'arcs  $(a_1 : n_1 \rightarrow n_0, a_2 : n_2 \rightarrow n_0)$  de  $\bar{\delta}$  tel que (5)

- il existe une flèche  $h : \delta(n_1) \rightarrow \delta(n_2)$  de  $\mathcal{C}_0$  vérifiant  $\delta(a_2) \circ h = \delta(a_1)$ ,
- il n'existe pas d'arc  $a : n_1 \rightarrow n_2$  étiqueté par  $h$  dans  $\bar{\delta}$ ,

$$\bar{\delta} := \text{Ajout\_Arc}(\bar{\delta}, n_1, n_2, h);$$

Jusqu'à  $\bar{\delta} = \bar{\delta}_0$ ;

Résultat :  $\bar{\delta}$ ;

Fin Complétion

**Proposition 5.7** *La fonction de complétion s'arrête.*

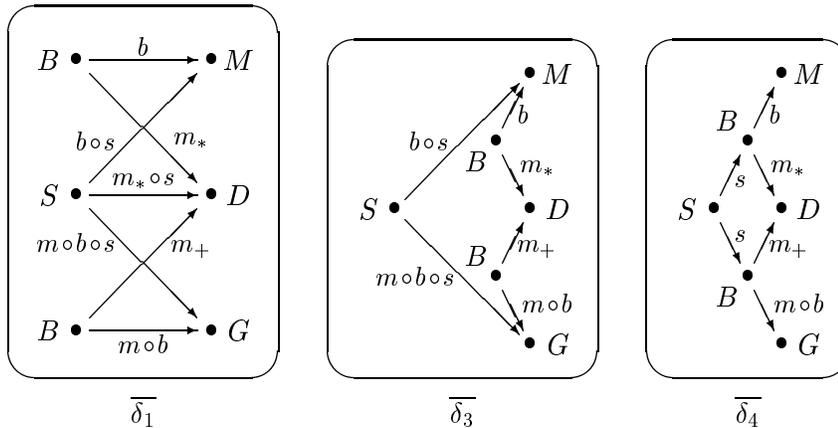
*Preuve.* On appelle *itération* le fait d'exécuter successivement les étapes 1 à 5. D'après la condition d'arrêt, on effectue une itération tant que le diagramme a été modifié par l'itération précédente. Chaque étape de l'itération termine: les étapes 1, 2 et 3 parce qu'elles suppriment un arc dans le graphe, et les étapes 4 et 5 parce que la catégorie  $\mathcal{C}_0$  est finie et qu'on n'ajoute pas une nouvelle composition ou factorisation à droite si l'arc correspondant appartient déjà au diagramme.

On ne peut effectuer qu'un nombre fini d'itérations. En effet, seule la contraction d'un arc identité (étape 1) peut permettre ensuite d'ajouter des nouvelles compositions d'arcs ou des nouvelles factorisations à droite. Comme la contraction supprime un nœud dans le graphe, et comme aucune transformation n'ajoute de nœud dans le graphe, on ne peut effectuer qu'un nombre fini d'itérations.  $\square$

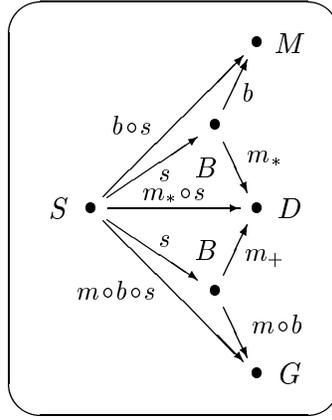
**Proposition 5.8** *La complétion est une transformation stable par isomorphisme. Pour tout diagramme  $\bar{\delta}$  sur  $\mathcal{C}_0$ ,  $\bar{\delta} \cong \text{Complétion}(\bar{\delta})$  dans  $\text{diagr}(\mathcal{C}_0)$ .*

*Preuve.* Toutes les transformations appliquées sur le diagramme  $\bar{\delta}$  par la fonction de complétion sont stables par isomorphisme, d'après les propositions 5.2, 5.3, 5.4, 5.5, et 5.6.  $\square$

**Exemple 5.10** Considérons les diagrammes  $\bar{\delta}_1$ ,  $\bar{\delta}_3$  et  $\bar{\delta}_4$ .



La complétion de ces trois diagrammes produit le même diagramme  $\overline{\delta'}$  :  
 Complétion( $\overline{\delta_1}$ ) = Complétion( $\overline{\delta_3}$ ) = Complétion( $\overline{\delta_4}$ ) =  $\overline{\delta'}$ .

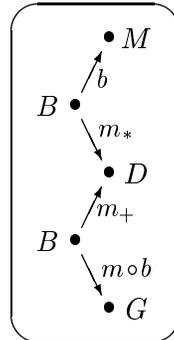


$\overline{\delta'}$

Un diagramme est *complet* si celui-ci est égal à sa complétion.

**Définition 5.12** (Diagramme complet) Un diagramme  $\overline{\delta}$  sur  $\mathcal{C}_0$  est *complet* si et seulement si  $\overline{\delta} = \text{Complétion}(\overline{\delta})$ .

**Exemple 5.11** Le diagramme  $\overline{\delta_2}$  est complet :  $\overline{\delta_2} = \text{Complétion}(\overline{\delta_2})$ .



$\overline{\delta_2}$

**Lemme 5.2** Soit un diagramme complet  $\overline{\delta}$  sur  $\mathcal{C}_0$ . Alors il n'existe aucune boucle dans le graphe  $\delta^\Phi$ .

*Preuve.* Supposons par l'absurde qu'on a une boucle, c'est-à-dire un arc  $a : n \rightarrow n$  dans  $\delta^\Phi$ . On a soit  $\delta(a) = id_{\delta(n)}$ , soit  $\delta(a) \neq id_{\delta(n)}$ . La supposition  $\delta(a) = id_{\delta(n)}$  est en contradiction avec l'hypothèse que  $\overline{\delta}$  est complet. La supposition  $\delta(a) \neq id_{\delta(n)}$  est en contradiction avec l'hypothèse que  $\mathcal{C}_0$  ne comporte pas de cycle. Par conséquent, il n'existe pas de boucle dans le graphe  $\delta^\Phi$ .  $\square$

**Lemme 5.3** Soit un diagramme complet  $\overline{\delta}$  sur  $\mathcal{C}_0$ . Soit deux nœuds  $n_1, n_2$  et un arc  $a : n_1 \rightarrow n_2$  dans le graphe  $\delta^\Phi$ . Alors, il n'existe aucun arc de  $n_2$  vers  $n_1$  dans  $\delta^\Phi$ .

*Preuve.* S'il existe un arc  $a' : n_2 \rightarrow n_1$  dans  $\delta^\Phi$ , alors comme  $\bar{\delta}$  est complet, il existe un arc  $a_1 : n_1 \rightarrow n_1$  étiqueté par la flèche  $\delta(a') \circ \delta(a)$ , ce qui est impossible d'après le lemme 5.2.  $\square$

### 5.2.3 Zigzags élémentaires et liens

La complétion d'un diagramme permet d'obtenir une forme un peu plus "canonique" pour les diagrammes. Cependant, deux diagrammes complets peuvent être isomorphes dans  $\text{diagr}(\mathcal{C}_0)$  sans être identiques. Par exemple, les diagrammes  $\bar{\delta}_2$  et  $\bar{\delta}'$  sont complets et isomorphes, mais ne sont pas identiques. Pour détecter cet isomorphisme, nous devons introduire la notion de *zigzag élémentaire*.

**Définition 5.13** (Zigzag élémentaire) Un *zigzag élémentaire* sur un graphe  $\delta^\Phi$  est un zigzag de longueur 2 de la forme

$$m_0 \xleftarrow{a_0} m_1 \xrightarrow{a_1} m_2$$

tel que les arcs  $a_0$  et  $a_1$  sont distincts.

Nous avons également besoin d'une relation d'*inclusion* sur les zigzags. Intuitivement, le zigzag  $Z$  est inclus dans le zigzag  $Z'$  si  $Z$  est un "sous-zigzag" de  $Z'$ .

**Définition 5.14** (Inclusion de zigzags) Soit un graphe  $\delta^\Phi$ . Considérons deux zigzags  $Z$  et  $Z'$  sur  $\delta^\Phi$ . Posons  $Z = (k, Z_N, Z_A)$  et  $Z' = (k', Z'_N, Z'_A)$ , avec

$$\begin{aligned} Z_N &= (n_0, n_1, \dots, n_k) \quad \text{et} \quad Z_A = (a_0, a_1, \dots, a_{k-1}); \\ Z'_N &= (n'_0, n'_1, \dots, n'_{k'}) \quad \text{et} \quad Z'_A = (a'_0, a'_1, \dots, a'_{k'-1}). \end{aligned}$$

On dit que le zigzag  $Z$  est *inclus* dans le zigzag  $Z'$ , et on note  $Z \subseteq Z'$  si et seulement si il existe un entier  $j$ ,  $0 \leq j \leq k' - k$ , tel que

- $\forall i, 0 \leq i \leq k, n_i = n'_{i+j}$ ;
- $\forall i, 0 \leq i \leq k - 1, a_i = a'_{i+j}$ .

Remarquons que si  $Z \subseteq Z'$ , alors nécessairement  $k \leq k'$ .

Soit un diagramme complet  $\bar{\delta}$ . Les zigzags élémentaires sur le *diagramme*  $\bar{\delta}$  sont les zigzags élémentaires sur le *graphe*  $\delta^\Phi$ . Nous définissons maintenant une relation d'ordre sur les zigzags élémentaires d'un diagramme complet.

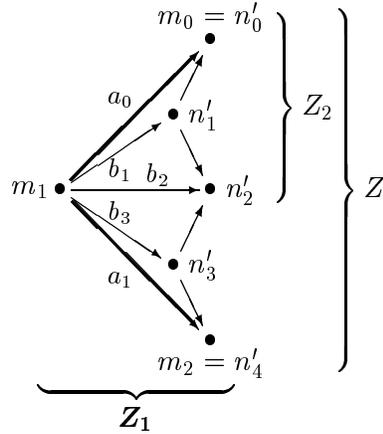
**Définition 5.15** Soit un diagramme complet  $\bar{\delta}$  sur  $\mathcal{C}_0$ . On définit une *relation d'ordre*  $\leq$  sur les zigzags élémentaires du diagramme complet  $\bar{\delta}$  de la façon suivante. Soit deux zigzags élémentaires  $Z_1$  et  $Z_2$  sur  $\bar{\delta}$ . Posons  $Z_1 = m_0 \xleftarrow{a_0} m_1 \xrightarrow{a_1} m_2$ .

- On a  $Z_1 < Z_2$  si et seulement si il existe un zigzag  $Z' = (k', Z'_N, Z'_A)$ , avec  $Z'_N = (n'_0, n'_1, \dots, n'_{k'})$  tel que
  - $m_0 = n'_0$  et  $m_2 = n'_{k'}$ ;

- $Z_2 \subseteq Z'$ ;
- $\delta(a_0) \sim_{\bar{\delta}} \delta(a_1) [Z']$ ;
- $\forall i, 0 \leq i \leq k', m_1 \neq n'_i$ .

- On a  $Z_1 \leq Z_2$  si et seulement si  $Z_1 = Z_2$  ou  $Z_1 < Z_2$ .

Dans le dessin ci-dessous,  $Z_1 < Z_2$ .



**Remarque 5.2** Supposons  $Z_1 < Z_2$ .

1. C'est le zigzag  $Z_1$  qui paraît "le plus grand". Nous avons choisi le sens  $Z_1 \leq Z_2$  car intuitivement, toute l'information sur les partages contenue dans le zigzag  $Z_1$  est contenue dans les zigzags élémentaires de  $Z'$ .
2. Soit  $\{c_i : \delta(m_1) \rightarrow \delta(n'_i) ; i \in \{0, \dots, k'\}\}$  l'ensemble des flèches qui correspondent à la connexion  $\delta(a_0) \sim_{\bar{\delta}} \delta(a_1) [Z']$  (cf. définition 2.17, page 70). Alors, comme le diagramme  $\bar{\delta}$  est complet, pour tout entier  $i, 0 \leq i \leq k'$ , il existe un arc  $b_i : m_1 \rightarrow n'_i$  dans  $\delta^\Phi$  tel que  $\delta(b_i) = c_i$ .

**Proposition 5.9** Soit un diagramme complet  $\bar{\delta}$  sur  $\mathcal{C}_0$ . La relation  $\leq$  sur les zigzags élémentaires sur  $\bar{\delta}$  est une relation d'ordre.

*Preuve.* On utilise la définition de  $\leq$  ainsi que le lemme 5.3.

*Réflexivité.* La relation  $\leq$  est réflexive par définition.

*Antisymétrie.* Soit deux zigzags élémentaires sur  $\bar{\delta}$

$$\begin{aligned} Z_1 &= m_0 \xleftarrow{a_0} m_1 \xrightarrow{a_1} m_2, \\ Z_2 &= n_0 \xleftarrow{b_0} n_1 \xrightarrow{b_1} n_2. \end{aligned}$$

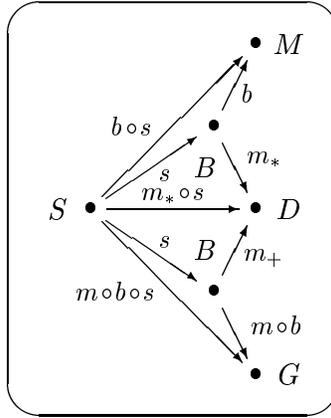
Supposons  $Z_1 \leq Z_2$  et  $Z_2 \leq Z_1$ . Par l'absurde, si  $Z_1 \neq Z_2$ , alors  $Z_1 < Z_2$  et  $Z_2 < Z_1$ . Par conséquent, on a  $m_1 \neq n_1$ . De plus, il existe un arc  $a : m_1 \rightarrow n_1$  et un arc  $a' : n_1 \rightarrow m_1$  dans  $\delta^\Phi$ . Cela est impossible d'après le lemme 5.3.

*Transitivité.* Soit trois zigzags élémentaires  $Z_1, Z_2$  et  $Z_3$  sur un diagramme complet  $\bar{\delta}$ . Supposons  $Z_1 \leq Z_2$  et  $Z_2 \leq Z_3$ . Si  $Z_1 = Z_2$  ou  $Z_2 = Z_3$  alors on a évidemment  $Z_1 \leq Z_3$ . Si  $Z_1 < Z_2$  et  $Z_2 < Z_3$ , alors on vérifie que  $Z_1 < Z_3$ .

□

Par exemple, dans le diagramme  $\bar{\delta}'$ ,

$$\begin{aligned} M \xleftarrow{b \circ s} S \xrightarrow{m \circ b \circ s} G &\leq M \xleftarrow{b} B \xrightarrow{m_*} D, \\ M \xleftarrow{b \circ s} S \xrightarrow{m \circ b \circ s} G &\leq D \xleftarrow{m_+} B \xrightarrow{m \circ b} G, \\ M \xleftarrow{b \circ s} S \xrightarrow{m_* \circ s} D &\leq M \xleftarrow{b} B \xrightarrow{m_*} D, \\ D \xleftarrow{m_* \circ s} S \xrightarrow{m \circ b \circ s} G &\leq D \xleftarrow{m_+} B \xrightarrow{m \circ b} G. \end{aligned}$$



$\bar{\delta}'$

Comme la relation  $\leq$  est une relation d'ordre sur l'ensemble fini des zigzags élémentaires de  $\bar{\delta}$ , cet ensemble contient des éléments *maximaux*. Intuitivement, toute l'information sur les partages est contenue dans les zigzags élémentaires maximaux.

**Définition 5.16** (Zigzag élémentaire maximal sur un diagramme complet  $\bar{\delta}$ )

Soit un diagramme complet  $\bar{\delta}$ . Un *zigzag élémentaire maximal* sur  $\bar{\delta}$  est un élément maximal pour la relation d'ordre  $\leq$  sur l'ensemble des zigzags élémentaires de  $\bar{\delta}$ .

Par exemple, les zigzags élémentaires

$$\begin{aligned} M \xleftarrow{b} B \xrightarrow{m_*} D \\ D \xleftarrow{m_+} B \xrightarrow{m \circ b} G \end{aligned}$$

sont maximaux dans le diagramme  $\bar{\delta}'$ . Par contre, les zigzags élémentaires

$$\begin{aligned} M \xleftarrow{b \circ s} S \xrightarrow{m \circ b \circ s} G \\ M \xleftarrow{b \circ s} S \xrightarrow{m_* \circ s} D \\ D \xleftarrow{m_* \circ s} S \xrightarrow{m \circ b \circ s} G \end{aligned}$$

ne sont pas maximaux dans  $\bar{\delta}'$ .

Nous avons encore besoin de quelques définitions avant de donner un critère d'isomorphisme entre deux diagrammes. Nous commençons par définir un *nœud puits* dans un graphe. Un nœud puits est un nœud d'où ne part aucune flèche.

**Définition 5.17** (Nœud puits) Soit un graphe  $\delta^\Phi$ . Un *nœud puits* de  $\delta^\Phi$  est un nœud  $n \in \text{Nœud}(\delta^\Phi)$  qui vérifie la condition suivante: il n'existe aucun arc  $a \in \text{Arc}(\delta^\Phi)$  tel que  $\text{Source}(a) = n$ .

Un *lien* sur un diagramme complet est un zigzag élémentaire maximal entre deux nœuds puits.

**Définition 5.18** (Lien sur un diagramme complet) Soit un diagramme complet  $\bar{\delta}$  sur  $\mathcal{C}_0$ . Un *lien*  $Z$  sur  $\bar{\delta}$  est un zigzag élémentaire maximal sur  $\bar{\delta}$

$$Z = m_0 \xleftarrow{a_0} m_1 \xrightarrow{a_1} m_2$$

tel que  $m_0$  et  $m_2$  sont des nœuds puits de  $\delta^\Phi$ .

L'ensemble des liens d'un diagramme  $\bar{\delta}$  est noté  $\text{Liens}(\bar{\delta})$ .

**Proposition 5.10** *L'ensemble des liens d'un diagramme complet  $\bar{\delta}$  est calculable.*

*Preuve.* On calcule l'ensemble des zigzags élémentaires de  $\bar{\delta}$ . Cet ensemble est fini. Pour chaque zigzag élémentaire

$$Z_1 = m_0 \xleftarrow{a_0} m_1 \xrightarrow{a_1} m_2,$$

on calcule l'ensemble des zigzags  $Z'$  tels que

- $\delta(a_0) \sim_{\bar{\delta}} \delta(a_1)$  [ $Z'$ ]
- $Z'$  est formé d'une suite de zigzags élémentaires, c'est-à-dire est de la forme

$$Z' = n'_0 \xleftarrow{a'_0} n'_1 \xrightarrow{a'_1} n'_2 \dots n'_{k'-2} \xleftarrow{a'_{k'-2}} n'_{k'-1} \xrightarrow{a'_{k'-1}} n'_{k'},$$

où  $\forall i \in \{0, \dots, \frac{k'}{2} - 1\}$ ,  $n'_{2i} \xleftarrow{a'_{2i}} n'_{2i+1} \xrightarrow{a'_{2i+1}} n'_{2i+2}$  est un zigzag élémentaire.

- $\forall i \in \{0, \dots, k'\}$ ,  $m_1 \neq n'_i$ .

Le zigzag élémentaire  $Z_1$  est alors strictement plus petit que chaque zigzag élémentaire de  $Z'$ . Puis on ne conserve que les zigzags élémentaires maximaux entre nœuds puits.  $\square$

### 5.2.4 Forme minimale

Calculer la forme minimale d'un diagramme complet consiste à ne conserver que les nœuds puits, ainsi que les nœuds et les arcs qui font partie d'un lien du diagramme.

**Définition 5.19** (Forme minimale) Soit un diagramme complet  $\bar{\delta}$ . La forme minimale du diagramme  $\bar{\delta}$  est le diagramme

$$\bar{\beta} = \text{Minimal}(\bar{\delta})$$

défini de la façon suivante.

- Graphe  $\beta^\Phi$  :
  - $\text{Nœud}(\beta^\Phi) = \{n \in \text{Nœud}(\delta^\Phi), \text{ tel que } n \text{ est un puits de } \delta^\Phi \text{ ou } n \text{ appartient à un lien de } \delta^\Phi\}$
  - $\text{Arc}(\beta^\Phi) = \{a \in \text{Arc}(\delta^\Phi), \text{ tel que } a \text{ appartient à un lien de } \delta^\Phi\}$
- Foncteur  $\beta : \beta^\Phi \rightarrow \mathcal{C}_0$ .
  - $\forall n \in \text{Nœud}(\beta^\Phi), \beta(n) = \delta(n)$  ;
  - $\forall a \in \text{Arc}(\beta^\Phi), \beta(a) = \delta(a)$ .

**Proposition 5.11** Pour tout diagramme complet  $\bar{\delta}$ ,

$$\bar{\delta} \cong \text{Minimal}(\bar{\delta}).$$

*Esquisse de la preuve.* Soit  $\bar{\beta} = \text{Minimal}(\bar{\delta})$ . On définit deux flèches de  $\text{DIAGR}(\mathcal{C}_0)$ ,  $\bar{\sigma} : \bar{\delta} \rightarrow \bar{\beta}$  et  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\delta}$  telles que  $\bar{\sigma} \circ \bar{\tau} \approx \text{id}_{\bar{\beta}}$  et  $\bar{\tau} \circ \bar{\sigma} \approx \text{id}_{\bar{\delta}}$ .

- On définit  $\bar{\sigma} : \bar{\delta} \rightarrow \bar{\beta}$  de la façon suivante.
  - Morphisme généralisé de graphes  $\sigma^\Phi : \delta^\Phi \rightsquigarrow \beta^\Phi$ .
    - Pour tout nœud  $n$  de  $\delta^\Phi$ , tel que  $n$  est un nœud puits ou fait partie d'un lien de  $\bar{\delta}$ , on pose  $\sigma^\Phi(n) = n$ .  
Pour tout autre nœud  $n$  de  $\delta^\Phi$ , il existe un nœud puits  $n'$  et un arc  $a_n : n \rightarrow n'$  dans  $\delta^\Phi$ . On pose  $\sigma^\Phi(n) = n'$ .
    - Pour tout arc  $a : n \rightarrow n'$  de  $\delta^\Phi$  qui appartient à un lien de  $\bar{\delta}$ , on pose  $\sigma^\Phi(a) = a$ .  
Pour tout autre arc  $a : n \rightarrow n'$  de  $\delta^\Phi$ ,  $\sigma^\Phi(a)$  est défini un peu plus loin.
  - Transformation naturelle généralisée  $\sigma : \delta \rightsquigarrow \beta \circ \sigma^\Phi$ .
    - Si  $n$  est un nœud de  $\beta^\Phi$ , on pose  $\sigma_n = \text{id}_{\delta(n)}$ .
    - Sinon, on pose  $\sigma_n = \delta(a_n)$  où  $a_n : n \rightarrow n'$  est un arc de  $\bar{\delta}$  tel que  $n' = \sigma^\Phi(n)$ .

Pour tout arc  $a : n \rightarrow n'$  de  $\delta^\Phi$ , si  $a$  appartient à un lien de  $\bar{\delta}$ , on a alors évidemment

$$\sigma_{n'} \circ \delta(a) = \beta(a) \circ \sigma_n.$$

Si  $a$  n'appartient pas à un lien de  $\bar{\delta}$ , il existe un zigzag  $Z$  tel que

$$\sigma_{n'} \circ \delta(a) \sim_{\bar{\beta}} \sigma_n [Z].$$

On pose  $\sigma^\Phi(a) = Z$ .

- La flèche  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\delta}$  est définie de façon évidente.
- On a  $\bar{\sigma} \circ \bar{\tau} = \overline{id_{\bar{\beta}}}$ .
- On vérifie également que  $\bar{\tau} \circ \bar{\sigma} \approx \overline{id_{\bar{\delta}}}$ .

□

**Théorème 5.1** *Deux diagrammes complets sont isomorphes si et seulement si leurs formes minimales sont égales.*

*Esquisse de la preuve.* Si les formes minimales de deux diagrammes sont égales, alors les diagrammes sont évidemment isomorphes. Réciproquement, soit deux diagrammes  $\bar{\alpha}$  et  $\bar{\beta}$  sous forme minimale. Soit deux flèches  $\bar{\sigma} : \bar{\alpha} \rightarrow \bar{\beta}$  et  $\bar{\tau} : \bar{\beta} \rightarrow \bar{\alpha}$  de  $\text{DIAGR}(\mathcal{C}_0)$  telles que  $\bar{\sigma} \circ \bar{\tau} \approx \overline{id_{\bar{\beta}}}$  et  $\bar{\tau} \circ \bar{\sigma} \approx \overline{id_{\bar{\alpha}}}$ .

1. On montre que pour tout nœud puits  $n$  de  $\alpha^\Phi$ ,  $\sigma^\Phi(n)$  est un nœud puits de  $\beta^\Phi$ ,  $\alpha(n) = \beta(\sigma^\Phi(n))$  et  $\sigma_n = id_{\alpha(n)}$ .
2. Soit un zigzag élémentaire

$$m_0 \xleftarrow{a_0} m_1 \xrightarrow{a_1} m_2$$

de  $\bar{\alpha}$ . On pose  $n_0 = \sigma^\Phi(m_0)$  et  $n_2 = \sigma^\Phi(m_2)$ . Les nœuds  $n_0$  et  $n_2$  sont donc des nœuds puits de  $\beta^\Phi$ , et on a

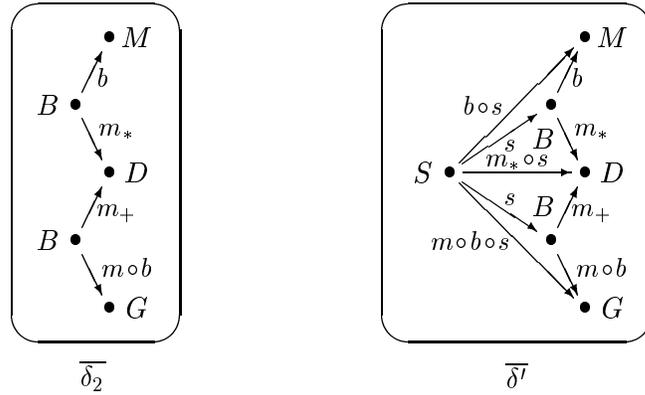
$$\begin{aligned} \alpha(m_0) &= \beta(n_0), \quad \alpha(m_2) = \beta(n_2), \\ \sigma_{m_0} &= id_{\alpha(m_0)}, \quad \sigma_{m_2} = id_{\alpha(m_2)}. \end{aligned}$$

On montre qu'il existe un nœud  $n_1$  dans  $\beta^\Phi$  tel que  $\alpha(m_1) = \beta(n_1)$ . Enfin, on montre qu'il existe deux arcs  $b_0 : n_1 \rightarrow n_0$  et  $b_1 : n_1 \rightarrow n_2$  dans  $\beta^\Phi$  tel que  $\alpha(a_0) = \beta(b_0)$  et  $\alpha(a_1) = \beta(b_1)$ .

3. Finalement, les diagrammes  $\bar{\alpha}$  et  $\bar{\beta}$  ont donc les mêmes nœuds puits, reliés par les mêmes liens. Les diagrammes  $\bar{\alpha}$  et  $\bar{\beta}$  sont donc identiques.

□

**Exemple 5.12** Revenons aux diagrammes  $\overline{\delta_2}$  et  $\overline{\delta'}$ .



Le diagramme  $\overline{\delta_2}$  est sous forme minimale; le diagramme  $\overline{\delta'}$  a pour forme minimale le diagramme  $\overline{\delta_2}$ .

$$\begin{aligned} \text{Minimal}(\overline{\delta_2}) &= \overline{\delta_2}; \\ \text{Minimal}(\overline{\delta'}) &= \overline{\delta_2}. \end{aligned}$$

Finalement, les diagrammes  $\overline{\delta_2}$  et  $\overline{\delta'}$  sont donc isomorphes.

### 5.2.5 Algorithme

Finalement, un algorithme pour détecter si deux diagrammes sont isomorphes dans la catégorie  $\text{diagr}(\mathcal{C}_0)$  consiste à calculer la complétion puis la forme minimale de chaque diagramme, et enfin à comparer les formes minimales. Pour calculer les formes minimales, il est nécessaire de calculer l'ensemble des liens associés à chaque diagramme.

Fonction Isorphe ( $\overline{\alpha_0}, \overline{\beta_0}$  : diagramme) : booléen

$\overline{\alpha}, \overline{\beta}$  : diagramme ;

Début

$\overline{\alpha} := \text{Minimal}(\text{Complétion}(\overline{\alpha_0}))$  ;  
 $\overline{\beta} := \text{Minimal}(\text{Complétion}(\overline{\beta_0}))$  ;  
 Résultat :  $\overline{\alpha} = \overline{\beta}$  ;

Fin Isorphe

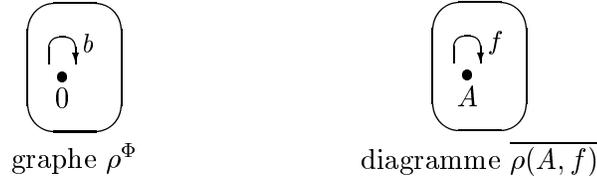
## 5.3 Cas général

Dans cette partie, nous examinons les problèmes soulevés lorsque la catégorie  $\mathcal{C}_0$  est infinie ou contient des cycles.

### 5.3.1 Zigzags

Dans ce paragraphe, nous revenons un instant sur les zigzags. Considérons un diagramme fini  $\bar{\delta}$ . Le graphe  $\delta^\Phi$  est fini. Par contre, le nombre de zigzags sur  $\delta^\Phi$  est en général infini. En effet, rien n'interdit de passer plusieurs fois par le même arc.

- Si nous considérons un graphe avec une ou plusieurs boucles, il existe évidemment un nombre infini de zigzags sur ce graphe.



Nous pouvons en effet considérer les zigzags suivants.

$$\begin{aligned} 0 &\xrightarrow{b} 0 \\ 0 &\xrightarrow{b} 0 \xrightarrow{b} 0 \\ 0 &\xrightarrow{b} 0 \xrightarrow{b} 0 \dots 0 \xrightarrow{b} 0 \end{aligned}$$

- Un diagramme, même s'il est construit sur un graphe sans cycle, contient en général une infinité de zigzags. Considérons par exemple le diagramme  $\bar{\pi}$  construit sur le graphe  $\pi^\Phi$ .



Nous pouvons considérer les zigzags suivants.

$$\begin{aligned} 1 &\xleftarrow{a} 0 \xrightarrow{b} 2 \\ 1 &\xleftarrow{a} 0 \xrightarrow{b} 2 \xleftarrow{b} 0 \xrightarrow{a} 1 \\ 1 &\xleftarrow{a} 0 \xrightarrow{b} 2 \xleftarrow{b} 0 \xrightarrow{a} 1 \xleftarrow{a} 0 \xrightarrow{b} 2 \\ &\dots \end{aligned}$$

De façon générale, il n'existe pas forcément de borne supérieure pour la longueur des zigzags à considérer lorsqu'on veut tester si deux flèches  $u$  et  $v$  sont connectées, c'est-à-dire s'il existe un zigzag  $Z$  tel que  $u \sim_{\bar{\delta}} v [Z]$ .

### 5.3.2 Catégorie de base comportant des cycles

Nous supposons dans ce paragraphe que la catégorie de base  $\mathcal{C}_0$  contient des cycles. Remarquons d'abord que la contraction d'un arc identité dans un diagramme peut former des *boucles* dans le graphe sous-jacent du diagramme, comme dans l'exemple 5.9–3.

$$\left( \begin{array}{ccc} & f & \\ \bullet & \xrightarrow{\quad} & \bullet \\ A & \xrightarrow{id_A} & A \end{array} \right) \cong \left( \begin{array}{c} \curvearrowright f \\ \bullet \\ A \end{array} \right)$$

Si  $\mathcal{C}_0$  comporte des cycles, l'algorithme présenté en section 5.2 n'est pas correct. En effet, le principal de l'information contenue dans le diagramme n'est plus concentrée dans les nœuds puits. Certains diagrammes, comme  $\rho(A, f)$  n'ont pas de nœud puits.

$$\rho(A, f) = \left( \begin{array}{c} \curvearrowright f \\ \bullet \\ A \end{array} \right)$$

Si l'ensemble  $\{f^n, n \in \mathbb{N}\}$  est infini, c'est-à-dire si toutes les flèches  $f^n$  sont différentes, alors on peut ajouter une infinité d'arcs au diagramme. On peut en effet ajouter un arc étiqueté par  $f^2 = f \circ f$ , un arc étiqueté par  $f^3 = f \circ f \circ f$ , etc. Autrement dit, pour tout entier  $n$ , les deux diagrammes ci-dessous sont isomorphes.

$$\left( \begin{array}{c} \curvearrowright f \\ \bullet \\ A \end{array} \right) \cong \left( \begin{array}{c} f^n \\ \curvearrowright \\ \bullet \\ A \end{array} \right)$$

Par conséquent, si l'ensemble  $\{f^n, n \in \mathbb{N}\}$  est infini, alors la fonction de complétion appliquée sur le diagramme  $\rho(A, f)$  boucle.

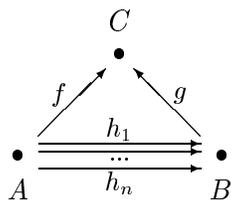
### 5.3.3 Catégorie de base infinie

Lorsque la catégorie de base  $\mathcal{C}_0$  est infinie, en particulier lorsque  $\mathcal{C}_0$  comporte un nombre infini de flèches entre deux objets, la procédure qui consiste à ajouter les factorisations à droite peut ne pas s'arrêter, ou, ce qui revient au même, conduire à un diagramme non fini.

**Exemple 5.13** Soit deux flèches  $f : A \rightarrow C$  et  $g : B \rightarrow C$  de  $\mathcal{C}_0$ . Supposons qu'on a une famille indexée par  $\mathbb{N}$  (donc infinie) de flèches

$$\{h_i : A \rightarrow B, i \in \mathbb{N}\}$$

telle que  $\forall i \in \mathbb{N}, g \circ h_i = f$ .



Si on considère un diagramme qui comporte deux arcs

$$n_0 \xrightarrow{a} n_1 \xleftarrow{b} n_2$$

étiquetés respectivement par  $f$  et  $g$ , on peut alors réaliser une infinité de factorisations à droite. Par conséquent, la fonction de complétion boucle.



# Conclusion

## 1 Bilan

Nous avons proposé un cadre théorique pour étudier les spécifications algébriques modulaires. Nous avons repris une idée classique en spécification algébrique: la composition de spécifications peut être modélisée par des colimites de diagrammes finis. La construction principale est la somme amalgamée qui permet de représenter la composition de deux spécifications en précisant quelles parties sont partagées. Notre travail a consisté à poursuivre cette idée bien connue selon trois directions.

- D'un point de vue syntaxique, nous avons défini un langage de termes pour les constructions de colimites finies sur une catégorie de base  $\mathcal{C}_0$ . Ce langage est défini formellement par la catégorie  $\text{Terme}(\mathcal{C}_0)$ .
- D'un point de vue sémantique, nous avons proposé d'interpréter les termes par des diagrammes finis sur  $\mathcal{C}_0$ . Cette représentation est plus abstraite que la représentation par des termes car elle permet d'éliminer une partie des choix effectués lors de la construction de la spécification modulaire.
- Nous avons défini la notion d'*isomorphisme de construction* pour les spécifications modulaires. Un isomorphisme de construction entre deux spécifications modulaires correspond à un isomorphisme entre les diagrammes correspondants. Nous avons enfin proposé un algorithme pour détecter si deux diagrammes sont isomorphes, dans le cas particulier où la catégorie de base  $\mathcal{C}_0$  est finie et ne comporte pas de cycle.

Notre travail est articulé en cinq chapitres.

1. Dans le chapitre 1, après quelques rappels sur les spécifications algébriques équationnelles, nous avons présenté plusieurs spécifications modulaires d'anneaux, qui nous ont permis d'introduire la définition d'isomorphisme de construction. Nous avons également précisé le cadre général de notre travail: la théorie des institutions, avec l'hypothèse que la catégorie des spécifications est finiment cocomplète.
2. Dans le chapitre 2, nous avons défini les notions de diagramme et morphisme de diagrammes en utilisant une formulation proche de l'informatique. Notre définition de diagramme est en effet basée sur celle de graphe, et non sur une catégorie quelconque. Nous avons reformulé les définitions de cône et de

colimite dans ce cadre. Nous avons défini deux catégories de diagrammes :  $\text{DIAGR}(\mathcal{C}_0)$  et  $\text{diagr}(\mathcal{C}_0)$ . L'intérêt de la catégorie  $\text{diagr}(\mathcal{C}_0)$  se situe sur le plan théorique puisque

- la catégorie  $\text{diagr}(\mathcal{C}_0)$  est finiment cocomplète ;
- la catégorie  $\text{diagr}(\mathcal{C}_0)$  est une complétion de  $\mathcal{C}_0$  par colimites finies.

La catégorie  $\text{DIAGR}(\mathcal{C}_0)$ , généralisation de certaines catégories utilisées en informatique — comme celle présentée dans [TBG91] —, est plus concrète que  $\text{diagr}(\mathcal{C}_0)$  dans la mesure où elle permet de manipuler les morphismes de diagrammes de façon effective.

3. Le chapitre 3 est consacré à la définition de la précatégorie  $\text{TERME}(\mathcal{C}_0)$ , qui offre un langage pour les constructions modulaires. Les objets de  $\text{TERME}(\mathcal{C}_0)$  sont des termes qui dénotent des spécifications modulaires, et les flèches de  $\text{TERME}(\mathcal{C}_0)$  sont des termes qui dénotent des morphismes de spécifications modulaires. Nous proposons une construction stratifiée de cette précatégorie, afin de résoudre le problème de circularité entre la définition des flèches et de l'équivalence dans  $\text{TERME}(\mathcal{C}_0)$ . On obtient, en passant au quotient, une catégorie  $\text{Terme}(\mathcal{C}_0)$ . Nous avons montré que

- la catégorie  $\text{Terme}(\mathcal{C}_0)$  est finiment cocomplète ;
- la catégorie  $\text{Terme}(\mathcal{C}_0)$  est une extension conservatrice de  $\mathcal{C}_0$ .

Enfin, nous avons montré qu'on peut construire une catégorie  $\mathcal{L}(\mathcal{C}_0)$  librement engendrée sur  $\mathcal{C}_0$  par objet initial choisi et sommes amalgamées choisies. Cette catégorie est obtenue en quotientant  $\text{TERME}(\mathcal{C}_0)$  par une congruence sur les objets et flèches de cette précatégorie.

4. Nous proposons dans le chapitre 4 une interprétation des termes représentant des spécifications modulaires par des diagrammes. Cette représentation, qui permet de faire abstraction de certaines étapes particulières choisies pour la construction de la spécification modulaire, est décrite par un préfoncteur

$$\mathcal{D} : \text{TERME}(\mathcal{C}_0) \rightarrow \text{DIAGR}(\mathcal{C}_0)$$

qui correspond à un foncteur entre les catégories correspondantes

$$\mathcal{D} : \text{Terme}(\mathcal{C}_0) \rightarrow \text{diagr}(\mathcal{C}_0).$$

Nous avons montré que

- le foncteur  $\mathcal{D}$  définit une équivalence entre les catégories  $\text{Terme}(\mathcal{C}_0)$  et  $\text{diagr}(\mathcal{C}_0)$  ;
- deux objets de  $\text{Terme}(\mathcal{C}_0)$  sont isomorphes si et seulement si leurs diagrammes associés sont isomorphes ;
- l'interprétation  $\mathcal{D}$  est correcte, c'est-à-dire qu'une spécification modulaire est isomorphe à la colimite du diagramme qui la représente.

5. La représentation des constructions modulaires par des diagrammes, bien que plus abstraite que celle des termes ne permet pas de détecter immédiatement des isomorphismes de constructions. En effet, deux spécifications modulaires isomorphes sont représentées par des diagrammes qui sont isomorphes, mais pas forcément identiques. Dans le chapitre 5, nous proposons une normalisation des diagrammes, dans le cas particulier où la catégorie des spécifications de base est finie et ne comporte pas de cycle. Ayant montré que deux diagrammes sont isomorphes si et seulement si ils ont la même forme normale, nous disposons donc d'une procédure pour décider si deux diagrammes sont isomorphes.

Le formalisme proposé est *compositionnel*, dans la mesure où des compositions successives de spécifications peuvent être vues comme un assemblage unique. En effet, une suite de constructions de sommes amalgamées est équivalente à la colimite d'un diagramme plus complexe. Cette propriété provient de la structure de *monade* des précatégories  $\text{TERME}(\mathcal{C}_0)$  et  $\text{DIAGR}(\mathcal{C}_0)$ .

Nous espérons avoir montré que la modularité, en particulier le problème des partages entre plusieurs spécifications, est correctement modélisé par les colimites finies. En effet, l'utilisation des colimites permet de faire abstraction d'une part de la définition effective des spécifications et d'autre part des étapes particulières choisies pour la construction d'une spécification modulaire. On peut ainsi se concentrer uniquement sur l'assemblage des différentes spécifications de base.

## 2 Perspectives

Citons quelques voies possibles pour poursuivre ce travail.

### Contraintes sémantiques

De nombreux langages de spécification permettent de préciser des *contraintes sémantiques*. Un exemple de contrainte sémantique est la *persistance* d'un foncteur de synthèse entre deux classes d'algèbres. Cette contrainte assure la préservation par exemple d'un module importé, ou encore du paramètre effectif lors d'une instantiation (cf. par exemple [EM85]). Une spécification  $B$  qui importe une spécification  $A$  est une flèche  $f : A \rightarrow B$ . J.-C. Reynaud [Rey90b, Rey93] a proposé de *distinguer* les flèches auxquelles est associée une contrainte sémantique. Dans notre cadre, l'utilisation de morphismes distingués ne paraît pas suffisante. En effet, comme les deux diagrammes suivants sont isomorphes, l'information d'importation entre  $A$  et  $B$  est perdue lors du passage à la colimite.

$$\left( \begin{array}{ccc} \bullet & \xrightarrow{f} & \bullet \\ A & & B \end{array} \right) \cong \left( \bullet \right) B$$

Par conséquent, le concept de colimite, qui convient à la modélisation des partages, ne peut pas être appliqué directement pour prendre en considération des contraintes sur des morphismes de spécifications.

## Enrichissement et abstraction

Deux opérateurs de constructions de spécifications sont fondamentaux en spécification algébrique : l'*enrichissement*, qui permet à partir d'une spécification d'obtenir une nouvelle spécification en ajoutant des sortes et des opérateurs ; et l'*abstraction*, qui permet d'obtenir une nouvelle spécification en supprimant des sortes et des opérateurs.

Il semble possible d'étendre le formalisme proposé afin de prendre en compte les enrichissements. Pour nous, un enrichissement d'une spécification modulaire  $S$  est une spécification  $S'$  accompagnée d'un morphisme de spécifications  $p : S \rightarrow S'$  qui modélise l'inclusion de  $S$  dans  $S'$ . La spécification  $S$  fait partie de  $\text{Terme}(\mathcal{C}_0)$ , mais pas  $S'$ , ni par conséquent  $f$ . Une solution pour permettre d'ajouter de telles flèches est de considérer la catégorie  $T'$  qui contient  $\text{Terme}(\mathcal{C}_0)$ , l'objet  $S$  et la flèche  $f$ . Ensuite, on considère la sous-catégorie pleine  $\mathcal{C}'_0$  de  $T'$  qui a pour objets d'une part l'objet  $S$  et d'autre part les objets de  $\mathcal{C}_0$ . Finalement, la catégorie  $\text{Terme}(\mathcal{C}'_0)$  contient  $\text{Terme}(\mathcal{C}_0)$ , ainsi que toutes les constructions qui ont été ajoutées par l'introduction de l'enrichissement  $S'$ . La flèche  $p : S \rightarrow S'$  appartient bien à la catégorie  $\text{Terme}(\mathcal{C}'_0)$  car celle-ci a pu être reconstruite à l'aide des flèches  $\text{up}$ .

Le problème de l'abstraction semble plus difficile à résoudre. Dans notre cadre, une abstraction d'une spécification  $S$  de  $\text{Terme}(\mathcal{C}_0)$  est une spécification  $S'$  accompagnée d'un morphisme de spécifications  $q : S' \rightarrow S$ . Dans ce cas, la construction décrite dans le cas de l'enrichissement ne fonctionne pas, car la flèche  $q$  n'appartient pas à la nouvelle catégorie des termes  $\text{Terme}(\mathcal{C}'_0)$ . Une voie de recherche pourrait être ici la prise en considération de colimites distinguées dans la catégorie de base  $\mathcal{C}_0$ .

## Colimites distinguées dans $\mathcal{C}_0$

En théorie des esquisses [Ehr68, DR94a, DR94b], une esquisse contient des cônes et des cocônes *distingués*. Le type d'une esquisse correspond à une catégorie librement engendrée par sommes et produits, en conservant les limites et colimites spécifiées.

Dans l'état actuel de notre travail, s'il existe par exemple une somme amalgamée dans la catégorie de base  $\mathcal{C}_0$ , cette somme est "perdue" lors de la construction de la catégorie  $\text{Terme}(\mathcal{C}_0)$ . Il serait intéressant de généraliser la construction de  $\text{Terme}(\mathcal{C}_0)$  de façon à pouvoir conserver certaines colimites.

La prise en considération de certaines colimites dans la catégorie de base peut résoudre le problème de l'abstraction. En effet, on peut imaginer ajouter à la catégorie de base la spécification  $S$  ainsi que les colimites qui ont permis de construire  $S$ , puis d'ajouter  $S'$  et le morphisme de spécifications  $q : S' \rightarrow S$ . On obtient ainsi une catégorie  $\mathcal{C}'_0$ , qui contient comme colimites distinguées d'une part les colimites distinguées de  $\mathcal{C}_0$ , et d'autre part les colimites qui ont permis de construire  $S$ . Finalement, la catégorie  $\text{Terme}(\mathcal{C}'_0)$  contient  $\text{Terme}(\mathcal{C}_0)$ ,  $S'$ ,  $q : S' \rightarrow S$ , ainsi que toutes les constructions qui utilisent  $S'$ .

## Généralisation de l'algorithme

En ce qui concerne les diagrammes, nous avons proposé un critère pour décider si deux diagrammes sont isomorphes dans  $\text{diagr}(\mathcal{C}_0)$ . Il resterait à évaluer la complexité de cet algorithme. D'autre part, nous avons supposé que la catégorie des spécifications de base  $\mathcal{C}_0$  est finie et ne comporte pas de cycle. Cette hypothèse est compatible avec le langage de spécification LPG. Il serait néanmoins intéressant soit de généraliser l'algorithme proposé, soit de montrer l'indécidabilité de ce problème.

## Application aux langages de spécification et de programmation

Notre travail a des retombées pratiques sur le traitement de la modularité dans les langages de spécification et de programmation.

Tout d'abord, la somme amalgamée est couramment utilisée dans les langages de spécification algébrique; par contre les flèches "up", à part dans le langage LPG, sont ignorées. D'un point de vue théorique, cela implique que les catégories considérées ne sont pas finiment cocomplètes. D'un point de vue pratique, cela signifie qu'on ne peut pas modéliser toutes les compositions de spécifications, en particulier certains assemblages décrits dans le chapitre 1 sur l'exemple des anneaux.

D'autre part, le langage de termes proposé, qui permet de coder finement les assemblages de modules, a l'avantage d'introduire une gestion explicite des partages entre spécifications. Remarquons également que l'instanciation des modules génériques peut être traitée de façon uniforme dans le même cadre.

Ces éléments doivent permettre de définir, pour les langages de la prochaine génération, un concept de modularité plus général et bien fondé d'un point de vue théorique.

## Réutilisation

Enfin, le formalisme que nous avons proposé peut également être appliqué à la réutilisation de programmes. On suppose que l'on dispose d'une bibliothèque de spécifications de base, accompagnées de différentes implantations. Comme il peut être nécessaire d'implanter une même spécification de façons différentes dans diverses parties du programme, par exemple pour des raisons d'efficacité, cette bibliothèque doit contenir différentes implantations pour chaque spécification de base. D'autre part, cette bibliothèque peut également contenir des implantations de spécifications modulaires construites sur les spécifications de base.

L'objectif est d'implanter une nouvelle spécification modulaire, en réutilisant au mieux les implantations disponibles. Si cette spécification est isomorphe à une construction déjà implantée, on peut bien évidemment réutiliser cette implantation. De façon plus générale, il s'agit de détecter les implantations réutilisables et les spécifications qui restent à implanter, et, parallèlement, les contraintes posées sur ces implantations par les partages entre les spécifications de base utilisées. L'idée est d'extraire des informations du diagramme associé à la spécification modulaire à implanter, ce diagramme pouvant être considéré comme le degré de liberté dont on dispose pour coder les spécifications de base.



# Appendice A

## Preuves sur la syntaxe

### A.1 Preuve du théorème de forme normale (2)

Nous montrons le théorème 3.3, page 134.

**Théorème 3.3** (*Forme normale (2)*)

Soit  $A \in \text{Obj}(\mathcal{C}_0)$ ,  $B \equiv \text{push}(A_0, A_1, A_2, f_1, f_2) \in \text{Obj}(\mathcal{C}_i)$ ,  $m \in \text{Arr}(\mathcal{C}_i)(A, B)$ . Posons

$$\mathcal{N}(m) \equiv m_n \circ m_{n-1} \circ \cdots \circ m_1,$$

avec  $n \geq 1$  et  $\forall j \in \{1, \dots, n\}$ ,  $m_j$  est une flèche élémentaire. Alors,

1.  $m_n \equiv \&_k(A_0, A_1, A_2, f_1, f_2)$ , avec  $k \in \{1, 2\}$  ;
2. si  $n \geq 2$ , alors  $m_{n-1} \circ \cdots \circ m_1 \in \text{Arr}(\mathcal{C}_{i-1})(A, A_k)$ .

*Preuve.* Nous montrons successivement les points 1 et 2.

1. On a

- $m_n \notin \text{Arr}(\mathcal{C}_0)(B', B)$ , car  $B \notin \text{Obj}(\mathcal{C}_0)$ .
- $m_n \neq \text{id}_X$ , sinon nécessairement  $n \geq 2$  et donc  $\mathcal{N}(m)$  n'est pas irréductible.
- $m_n \neq j_X$ , car  $\text{Arr}(\mathcal{C}_j)(A, \emptyset) = \emptyset$ .

On fait une induction sur  $n$ . Si  $n = 1$ , alors comme  $A \in \text{Obj}(\mathcal{C}_0)$ , on a

$$m_1 \equiv \&_k(A_0, A_1, A_2, f_1, f_2).$$

Pas d'induction: par l'absurde, si  $m_n \neq \&_k(A_0, A_1, A_2, f_1, f_2)$ , alors

$$m_n \equiv \text{up}(A'_0, A'_1, A'_2, B, f'_1, f'_2, g'_1, g'_2) \in \text{Arr}(\mathcal{C}_i)(\text{push}(A'_0, A'_1, A'_2, f'_1, f'_2), B).$$

Par hypothèse d'induction,

$$m_{n-1} \equiv \&_{k'}(A'_0, A'_1, A'_2, f'_1, f'_2),$$

ce qui est en contradiction avec  $\mathcal{N}(m)$  irréductible. On a donc

$$m_n \equiv \&_k(A_0, A_1, A_2, f_1, f_2).$$

2. Par induction sur  $n$ . Si  $n = 2$ , alors

$$\mathcal{N}(m) \equiv \&_k(A_0, A_1, A_2, f_1, f_2) \circ m_1.$$

D'après le lemme 3.7,  $A_k \neq \emptyset$ .

Si  $A_k \in \text{Obj}(\mathcal{C}_0)$ , alors  $m_1 \in \text{Arr}(\mathcal{C}_0)(A, A_k)$ .

Sinon,  $A_k \equiv \text{push}(A'_0, A'_1, A'_2, f'_1, f'_2) \in \text{Obj}(\mathcal{C}_{i-1})$ , et donc

$$m_1 \equiv \&_{k'}(A'_0, A'_1, A'_2, f'_1, f'_2) \in \text{Arr}(\mathcal{C}_{i-1})(A, A_k).$$

Pas d'induction :

$$\mathcal{N}(m) \equiv \&_k(A_0, A_1, A_2, f_1, f_2) \circ m_{n-1} \circ \cdots \circ m_1 \in \text{Arr}(\mathcal{C}_i)(A, B).$$

D'après le lemme 3.7,  $A_k \neq \emptyset$ .

Si  $A_k \in \text{Obj}(\mathcal{C}_0)$ , alors d'après le théorème 3.2,

$$m_{n-1} \circ \cdots \circ m_1 \in \text{Arr}(\mathcal{C}_0)(A, A_k)$$

Si  $A_k \equiv \text{push}(A'_0, A'_1, A'_2, f'_1, f'_2) \in \text{Obj}(\mathcal{C}_{i-1})$ , alors

$$m_{n-1} \equiv \&_{k'}(A'_0, A'_1, A'_2, f'_1, f'_2) \in \text{Arr}(\mathcal{C}_{i-1})(A'_{k'}, A_k)$$

et, par hypothèse d'induction,

$$m_{n-2} \circ \cdots \circ m_1 \in \text{Obj}(\mathcal{C}_{i-2})(A, A'_{k'}),$$

d'où

$$m_{n-1} \circ \cdots \circ m_1 \in \text{Arr}(\mathcal{C}_{i-1})(A, A_k).$$

□

## A.2 Preuve du théorème de forme normale (3)

Nous montrons le théorème 3.4, page 134.

**Théorème 3.4** (*Forme normale (3)*)

Soit  $A \equiv \text{push}(A_0, A_1, A_2, f_1, f_2) \in \text{Obj}(\mathcal{C}_i)$ ,  $B \in \text{Obj}(\mathcal{C}_0)$  et  $m \in \text{Arr}(\mathcal{C}_i)(A, B)$ . Posons

$$\mathcal{N}(m) \equiv m_n \circ m_{n-1} \circ \cdots \circ m_1,$$

avec  $n \geq 1$  et  $\forall j \in \{1, \dots, n\}$ ,  $m_j$  est une flèche élémentaire. Alors

$$m_1 \equiv \text{up}(A_0, A_1, A_2, A_3, f_1, f_2, g_1, g_2).$$

*Preuve.* D'abord, on a

- $m_1 \notin \text{Arr}(\mathcal{C}_0)(A, A')$  car  $A \notin \text{Obj}(\mathcal{C}_0)$ .
- $m_1 \neq \text{id}_X$ , sinon nécessairement,  $n \geq 2$  et donc  $\mathcal{N}(m)$  n'est pas irréductible.
- $m_1 \neq j_X$ , car  $A \neq \emptyset$ .

Par conséquent, on a soit

$$m_1 \equiv \text{up}(A_0, A_1, A_2, B, f_1, f_2, g_1, g_2),$$

soit

$$m_1 \equiv \&_{k'}(A'_0, A'_1, A'_2, f'_1, f'_2).$$

On montre par induction sur  $n$  que c'est la première condition qui est satisfaite. Si  $n = 1$ , alors comme  $B \in \text{Obj}(\mathcal{C}_0)$ , on a

$$m_1 \equiv \text{up}(A_0, A_1, A_2, B, f_1, f_2, g_1, g_2).$$

Pas d'induction: par l'absurde, si

$$m_1 \neq \text{up}(A_0, A_1, A_2, A_3, f_1, f_2, g_1, g_2),$$

alors

$$m_1 \equiv \&_{k'}(A'_0, A'_1, A'_2, f'_1, f'_2).$$

Par hypothèse d'induction,

$$m_2 \equiv \text{up}(A'_0, A'_1, A'_2, A'_3, f'_1, f'_2, g'_1, g'_2),$$

ce qui est en contradiction avec  $\mathcal{N}(m)$  irréductible. Par conséquent,

$$m_1 \equiv \text{up}(A_0, A_1, A_2, A_3, f_1, f_2, g_1, g_2).$$

□

### A.3 Preuve du théorème d'“unicité” de la forme normale

Nous montrons le théorème 3.5, page 135.

**Théorème 3.5** (“Unicité” de la forme normale)

Soit  $m, m' \in \text{Arr}(\mathcal{C}_i)(X, Y)$ , tels que  $m = m' \in \text{Arr}(\mathcal{C}_i)(X, Y)$ .

1. Si  $X, Y \in \text{Obj}(\mathcal{C}_0)$ , alors

$$\mathcal{N}(m) = \mathcal{N}(m') \in \text{Arr}(\mathcal{C}_0)(X, Y).$$

2. Si  $X \in \text{Obj}(\mathcal{C}_0)$ ,  $Y \equiv \text{push}(A_0, A_1, A_2, f_1, f_2)$ , alors  
 $\forall Z \in \text{Obj}(\mathcal{C}_0)$ ,  $\forall h \in \text{Arr}(\mathcal{C}_i)(Y, Z)$ , on a

$$\mathcal{N}(h \circ m) = \mathcal{N}(h \circ m') \in \text{Arr}(\mathcal{C}_0)(X, Z).$$

3. Si  $X \equiv \text{push}(A_0, A_1, A_2, f_1, f_2)$ ,  $Y \in \text{Obj}(\mathcal{C}_0)$ , alors  
 $\forall W \in \text{Obj}(\mathcal{C}_0)$ ,  $\forall p \in \text{Arr}(\mathcal{C}_i)(W, X)$ , on a

$$\mathcal{N}(m \circ p) = \mathcal{N}(m' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Y).$$

4. Si  $X \equiv \text{push}(A_0, A_1, A_2, f_1, f_2)$ ,  $Y \equiv \text{push}(A'_0, A'_1, A'_2, f'_1, f'_2)$ , alors  
 $\forall W, Z \in \text{Obj}(\mathcal{C}_0)$ ,  $\forall h \in \text{Arr}(\mathcal{C}_i)(Y, Z)$ ,  $\forall p \in \text{Arr}(\mathcal{C}_i)(W, X)$ , on a

$$\mathcal{N}(h \circ m \circ p) = \mathcal{N}(h \circ m' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Z).$$

*Preuve.* Nous montrons les points 1 à 4 par induction sur  $i$ . Pour  $i = 0$ , le point 1 est évident, et il n'y a rien à prouver pour les points 2 à 4. Pour  $i + 1$ , on prouve les points 2 et 4 par induction sur la longueur de  $\mathcal{N}(h)$ . Nous prouvons alors les points 1 à 4 en parallèle, par induction sur la longueur de la preuve que

$$m = m' \in \text{Arr}(\mathcal{C}_{i+1})(X, Y).$$

Il faut considérer les règles (11) à (23).

En réalité, nous avons besoin de l'hypothèse d'induction sur la longueur de  $\mathcal{N}(h)$  uniquement pour la règle (20). Dans tous les autres cas, il est inutile de prouver le cas de base, puis le pas d'induction. Les points 2 et 4 sont donc montrés directement, pour toutes les règles sauf la règle (20), et par induction sur la longueur de  $\mathcal{N}(h)$  pour la règle (20).

Règle (11)  $m = m' \in \text{Arr}(\mathcal{C}_0)(X, Y)$ .

1. D'après le théorème 3.1, on a

$$\begin{aligned} m &= \mathcal{N}(m) \in \text{Arr}(\mathcal{C}_0)(X, Y) \\ m' &= \mathcal{N}(m') \in \text{Arr}(\mathcal{C}_0)(X, Y) \end{aligned}$$

$$\text{et donc } \mathcal{N}(m) = \mathcal{N}(m') \in \text{Arr}(\mathcal{C}_0)(X, Y).$$

Il n'y a rien à démontrer pour les points 2 à 4.

Règle (12) Les 4 points sont évidents.

Règle (13) Les 4 points sont évidents, par symétrie de l'égalité dans  $\mathcal{C}_0$ .

Règle (14) Les 4 points sont évidents, par transitivité de l'égalité dans  $\mathcal{C}_0$ .

Règle (15)  $m \equiv g \circ f$ ,  $m' \equiv g' \circ f'$ , avec  $f = f' \in \text{Arr}(\mathcal{C}_{i+1})(A, B)$  et  $g = g' \in \text{Arr}(\mathcal{C}_{i+1})(B, C)$ .

1.  $A, C \in \text{Obj}(\mathcal{C}_0)$ .

- $B \in \text{Obj}(\mathcal{C}_0)$ . Par hypothèse d'induction, on a
 
$$\mathcal{N}(f) = \mathcal{N}(f') \in \text{Arr}(\mathcal{C}_0)(A, B) \quad (i)$$

$$\mathcal{N}(g) = \mathcal{N}(g') \in \text{Arr}(\mathcal{C}_0)(B, C) \quad (ii)$$

$$\begin{aligned} \mathcal{N}(g \circ f) &\equiv \mathcal{N}(\mathcal{N}(g) \circ \mathcal{N}(f)) \in \text{Arr}(\mathcal{C}_0)(A, C) \\ &= \mathcal{N}(g) \circ \mathcal{N}(f) \in \text{Arr}(\mathcal{C}_0)(A, C) \quad (\text{théorème 3.1}) \\ &= \mathcal{N}(g') \circ \mathcal{N}(f') \in \text{Arr}(\mathcal{C}_0)(A, C) \quad (\text{d'après (i) et (ii)}) \\ &= \mathcal{N}(\mathcal{N}(g') \circ \mathcal{N}(f')) \in \text{Arr}(\mathcal{C}_0)(A, C) \quad (\text{théorème 3.1}) \\ &\equiv \mathcal{N}(g' \circ f') \in \text{Arr}(\mathcal{C}_0)(A, C) \end{aligned}$$

- $B \equiv \text{push}(A_0, A_1, A_2, f_1, f_2)$ . Par hypothèse d'induction, on a :
 
$$\forall Z \in \text{Obj}(\mathcal{C}_0), \forall h \in \text{Arr}(\mathcal{C}_{i+1})(C, Z),$$

$$\mathcal{N}(h \circ f) = \mathcal{N}(h \circ f') \in \text{Arr}(\mathcal{C}_0)(A, Z) \quad (i)$$

$$\forall W \in \text{Obj}(\mathcal{C}_0), \forall p \in \text{Arr}(\mathcal{C}_{i+1})(W, A),$$

$$\mathcal{N}(g \circ p) = \mathcal{N}(g' \circ p) \in \text{Arr}(\mathcal{C}_{i+1})(W, C) \quad (ii)$$

Par conséquent,

$$\begin{aligned} \mathcal{N}(g \circ f) &= \mathcal{N}(g \circ f') \in \text{Arr}(\mathcal{C}_0)(A, C) \quad ((i), \text{ avec } h \equiv g) \\ &= \mathcal{N}(g' \circ f') \in \text{Arr}(\mathcal{C}_0)(A, C) \quad ((ii), \text{ avec } p \equiv f') \end{aligned}$$

- $B \equiv \emptyset$  : impossible d'après le lemme 3.7.

2.  $A \in \text{Obj}(\mathcal{C}_0)$ ,  $C \equiv \text{push}(A'_0, A'_1, A'_2, f'_1, f'_2)$ . Il faut montrer que
 
$$\forall Z \in \text{Obj}(\mathcal{C}_0), \forall h \in \text{Arr}(\mathcal{C}_{i+1})(C, Z),$$

$$\mathcal{N}(h \circ g \circ f) = \mathcal{N}(h \circ g' \circ f') \in \text{Arr}(\mathcal{C}_0)(A, Z).$$

- $B \in \text{Obj}(\mathcal{C}_0)$ . Par hypothèse d'induction, on a
 
$$\mathcal{N}(f) = \mathcal{N}(f') \in \text{Arr}(\mathcal{C}_0)(A, B) \quad (i)$$

$$\forall Z \in \text{Obj}(\mathcal{C}_0), \forall h \in \text{Arr}(\mathcal{C}_{i+1})(C, Z),$$

$$\mathcal{N}(h \circ g) = \mathcal{N}(h \circ g') \in \text{Arr}(\mathcal{C}_0)(B, Z) \quad (ii)$$

Par conséquent,

$$\begin{aligned} \mathcal{N}(h \circ g \circ f) &\equiv \mathcal{N}(\mathcal{N}(h \circ g) \circ \mathcal{N}(f)) \\ &= \mathcal{N}(h \circ g) \circ \mathcal{N}(f) \in \text{Arr}(\mathcal{C}_0)(A, Z) \quad (\text{théorème 3.1}) \\ &= \mathcal{N}(h \circ g') \circ \mathcal{N}(f') \in \text{Arr}(\mathcal{C}_0)(A, Z) \quad (\text{d'après (i) et (ii)}) \\ &= \mathcal{N}(\mathcal{N}(h \circ g') \circ \mathcal{N}(f')) \quad (\text{théorème 3.1}) \\ &\equiv \mathcal{N}(h \circ g' \circ f') \end{aligned}$$

- $B \equiv \text{push}(A_0, A_1, A_2, f_1, f_2)$ . Par hypothèse d'induction, on a
 
$$\forall Z \in \text{Obj}(\mathcal{C}_0), \forall h' \in \text{Arr}(\mathcal{C}_{i+1})(B, Z),$$

$$\mathcal{N}(h' \circ f) = \mathcal{N}(h' \circ f') \in \text{Arr}(\mathcal{C}_0)(A, Z) \quad (i)$$

$$\forall W, Z \in \text{Obj}(\mathcal{C}_0), \forall p' \in \text{Arr}(\mathcal{C}_{i+1})(W, B), \forall h \in \text{Arr}(\mathcal{C}_{i+1})(C, Z),$$

$$\mathcal{N}(h \circ g \circ p') = \mathcal{N}(h \circ g' \circ p') \in \text{Arr}(\mathcal{C}_0)(W, Z) \quad (ii)$$

Par conséquent,

$$\begin{aligned} \mathcal{N}(h \circ g \circ f) &= \mathcal{N}(h \circ g \circ f') \in \text{Arr}(\mathcal{C}_0)(A, Z) \quad ((i), \text{ avec } h' \equiv h \circ g) \\ &= \mathcal{N}(h \circ g' \circ f') \in \text{Arr}(\mathcal{C}_0)(A, Z) \quad ((ii), \text{ avec } p' \equiv f') \end{aligned}$$

- $B \equiv \emptyset$  : impossible d'après le lemme 3.7.

3.  $A \equiv \text{push}(A_0, A_1, A_2, f_1, f_2)$ ,  $C \in \text{Obj}(\mathcal{C}_0)$ . On doit montrer que  
 $\forall W \in \text{Obj}(\mathcal{C}_0)$ ,  $\forall p \in \text{Arr}(\mathcal{C}_{i+1})(W, A)$ ,

$$\mathcal{N}(g \circ f \circ p) = \mathcal{N}(g' \circ f' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, C).$$

- $B \in \text{Obj}(\mathcal{C}_0)$ . Par hypothèse d'induction, on a  
 $\forall W \in \text{Obj}(\mathcal{C}_0)$ ,  $\forall p \in \text{Arr}(\mathcal{C}_{i+1})(W, A)$ ,  
 $\mathcal{N}(f \circ p) = \mathcal{N}(f' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, B)$  (i)

et

$$\mathcal{N}(g) = \mathcal{N}(g') \in \text{Arr}(\mathcal{C}_0)(B, C) \quad (ii)$$

Par conséquent,

$$\begin{aligned} \mathcal{N}(g \circ f \circ p) &\equiv \mathcal{N}(\mathcal{N}(g) \circ \mathcal{N}(f \circ p)) \\ &= \mathcal{N}(g) \circ \mathcal{N}(f \circ p) \in \text{Arr}(\mathcal{C}_0)(W, C) && \text{(théorème 3.1)} \\ &= \mathcal{N}(g') \circ \mathcal{N}(f' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, C) && \text{(cf. (i) et (ii))} \\ &= \mathcal{N}(\mathcal{N}(g') \circ \mathcal{N}(f' \circ p)) \in \text{Arr}(\mathcal{C}_0)(W, C) && \text{(théorème 3.1)} \\ &\equiv \mathcal{N}(g' \circ f' \circ p) \end{aligned}$$

- $B \equiv \text{push}(A'_0, A'_1, A'_2, f'_1, f'_2)$ . Par hypothèse d'induction, on a  
 $\forall W, Z \in \text{Obj}(\mathcal{C}_0)$ ,  $\forall p \in \text{Arr}(\mathcal{C}_{i+1})(W, A)$ ,  $\forall h \in \text{Arr}(\mathcal{C}_{i+1})(B, Z)$ ,  
 $\mathcal{N}(h \circ f \circ p) = \mathcal{N}(h \circ f' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Z)$  (i)  
 et  $\forall W \in \text{Obj}(\mathcal{C}_0)$ ,  $\forall p' \in \text{Arr}(\mathcal{C}_{i+1})(W, B)$ ,  
 $\mathcal{N}(g \circ p') = \mathcal{N}(g' \circ p') \in \text{Arr}(\mathcal{C}_0)(W, C)$  (ii)

Par conséquent,

$$\begin{aligned} \mathcal{N}(g \circ f \circ p) &= \mathcal{N}(g \circ f' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, C) && ((i), \text{ avec } h \equiv g) \\ &= \mathcal{N}(g' \circ f' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, C) && ((ii), \text{ avec } p' \equiv f' \circ p) \end{aligned}$$

- $B \equiv \emptyset$ . Il n'existe pas de flèche  $p \in \text{Arr}(\mathcal{C}_{i+1})(W, A)$ , sinon, on aurait une flèche  $f \circ p \in \text{Arr}(\mathcal{C}_{i+1})(W, \emptyset)$ , avec  $W \in \text{Obj}(\mathcal{C}_0)$ . Par conséquent, on a le résultat.

4.  $A \equiv \text{push}(A_0, A_1, A_2, f_1, f_2)$  et  $C \equiv \text{push}(A_0'', A_1'', A_2'', f_1'', f_2'')$ . On doit montrer que

$$\forall W, Z \in \text{Obj}(\mathcal{C}_0), \forall p \in \text{Arr}(\mathcal{C}_{i+1})(W, A), \forall h \in \text{Arr}(\mathcal{C}_{i+1})(C, Z),$$

$$\mathcal{N}(h \circ g \circ f \circ p) = \mathcal{N}(h \circ g' \circ f' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Z).$$

- $B \in \text{Obj}(\mathcal{C}_0)$ . Par hypothèse d'induction, on a  
 $\forall W \in \text{Obj}(\mathcal{C}_0)$ ,  $\forall p \in \text{Arr}(\mathcal{C}_{i+1})(W, A)$ ,  
 $\mathcal{N}(f \circ p) = \mathcal{N}(f' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, B)$  (i)  
 $\forall Z \in \text{Obj}(\mathcal{C}_0)$ ,  $\forall h \in \text{Arr}(\mathcal{C}_{i+1})(C, Z)$ ,  
 $\mathcal{N}(h \circ g) = \mathcal{N}(h \circ g') \in \text{Arr}(\mathcal{C}_0)(B, Z)$  (ii)

Par conséquent,

$$\begin{aligned}
\mathcal{N}(h \circ g \circ f \circ p) & \\
&\equiv \mathcal{N}(\mathcal{N}(h \circ g) \circ \mathcal{N}(f \circ p)) \\
&= \mathcal{N}(h \circ g) \circ \mathcal{N}(f \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Z) \quad (\text{théorème 3.1}) \\
&= \mathcal{N}(h \circ g') \circ \mathcal{N}(f' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Z) \quad (\text{d'après (i) et (ii)}) \\
&= \mathcal{N}(\mathcal{N}(h \circ g') \circ \mathcal{N}(f' \circ p)) \quad (\text{théorème 3.1}) \\
&\equiv \mathcal{N}(h \circ g' \circ f' \circ p)
\end{aligned}$$

- $B \equiv \text{push}(A'_0, A'_1, A'_2, f'_1, f'_2)$ . Par hypothèse d'induction, on a  
 $\forall W, Z \in \text{Obj}(\mathcal{C}_0), \forall p \in \text{Arr}(\mathcal{C}_{i+1})(W, A), \forall h' \in \text{Arr}(\mathcal{C}_{i+1})(B, Z),$   
 $\mathcal{N}(h' \circ f \circ p) = \mathcal{N}(h' \circ f' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Z) \quad (i)$   
 $\forall W, Z \in \text{Obj}(\mathcal{C}_0), \forall p' \in \text{Arr}(\mathcal{C}_{i+1})(W, B), \forall h \in \text{Arr}(\mathcal{C}_{i+1})(C, Z),$   
 $\mathcal{N}(h \circ g \circ p') = \mathcal{N}(h \circ g' \circ p') \in \text{Arr}(\mathcal{C}_0)(W, Z) \quad (ii)$

Par conséquent,

$$\begin{aligned}
\mathcal{N}(h \circ g \circ f \circ p) & \\
&= \mathcal{N}(h \circ g \circ f' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Z) \quad ((i), \text{ avec } h' \equiv h \circ g) \\
&= \mathcal{N}(h \circ g' \circ f' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Z) \quad ((ii), \text{ avec } p' \equiv f' \circ p)
\end{aligned}$$

- $B \equiv \emptyset$ . Il n'existe pas de flèche  $p \in \text{Arr}(\mathcal{C}_{i+1})(W, A)$ , sinon, on aurait une flèche  $f \circ p \in \text{Arr}(\mathcal{C}_{i+1})(W, \emptyset)$ , avec  $W \in \text{Obj}(\mathcal{C}_0)$ . Par conséquent, on a le résultat.

Règle (16)  $m \equiv (h \circ g) \circ f$  et  $m' \equiv h \circ (g \circ f)$ .

1.  $\mathcal{N}((h \circ g) \circ f) \equiv \mathcal{N}(h \circ (g \circ f))$ .

Les points 2 à 4 sont aussi évidents.

Règle (17)  $m \equiv f \circ \text{id}_A$  et  $m' \equiv f$ , avec  $f \in \text{Arr}(\mathcal{C}_i)(A, B)$ .

1. On a

$$\mathcal{N}(m) \equiv \mathcal{N}(m').$$

2.  $\forall Z \in \text{Obj}(\mathcal{C}_0), \forall h \in \text{Arr}(\mathcal{C}_{i+1})(B, Z),$

$$\mathcal{N}(h \circ m) \equiv \mathcal{N}(h \circ m').$$

3.  $\forall W \in \text{Obj}(\mathcal{C}_0), \forall p \in \text{Arr}(\mathcal{C}_{i+1})(W, A),$

$$\mathcal{N}(m \circ p) \equiv \mathcal{N}(m' \circ p).$$

4.  $\forall W, Z \in \text{Obj}(\mathcal{C}_0), \forall p \in \text{Arr}(\mathcal{C}_{i+1})(W, A), \forall h \in \text{Arr}(\mathcal{C}_{i+1})(B, Z),$

$$\mathcal{N}(h \circ m \circ p) \equiv \mathcal{N}(h \circ m' \circ p).$$

Règle (18) Cas similaire au précédent.

Règle (19) Il n'y a rien à démontrer.

Règle (20)  $m \equiv \&_1(A_0, A_1, A_2, f_1, f_2) \circ f_1$  et  $m' \equiv \&_2(A_0, A_1, A_2, f_1, f_2) \circ f_2$ .

1. Il n'y a rien à démontrer.
2.  $X \equiv A_0 \in \text{Obj}(\mathcal{C}_0)$  et  $Y \equiv \text{push}(A_0, A_1, A_2, f_1, f_2)$ . Il faut montrer que  $\forall Z \in \text{Obj}(\mathcal{C}_0), \forall h \in \text{Arr}(\mathcal{C}_{i+1})(Y, Z)$ ,

$$\mathcal{N}(h \circ m) = \mathcal{N}(h \circ m').$$

Induction sur la longueur de  $\mathcal{N}(h)$ . Cas de base:  $\mathcal{N}(h)$  est de longueur 1. D'après le théorème 3.4,

$$\mathcal{N}(h) \equiv \text{up}(A_0, A_1, A_2, Z, f_1, f_2, g_1, g_2)$$

$$\begin{aligned} \mathcal{N}(h \circ m) &\equiv \mathcal{N}(\text{up}(A_0, \dots, g_1, g_2) \circ \&_1(A_0, \dots, f_2) \circ f_1) \\ &\equiv \mathcal{N}(g_1 \circ f_1) \\ \mathcal{N}(h \circ m') &\equiv \mathcal{N}(\text{up}(A_0, \dots, g_1, g_2) \circ \&_2(A_0, \dots, f_2) \circ f_2) \\ &\equiv \mathcal{N}(g_2 \circ f_2) \end{aligned}$$

Or, on a  $g_1 \circ f_1 = g_2 \circ f_2 \in \text{Arr}(\mathcal{C}_i)(X, Z)$ , donc par hypothèse d'induction sur  $i$ ,

$$\mathcal{N}(g_1 \circ f_1) = \mathcal{N}(g_2 \circ f_2) \in \text{Arr}(\mathcal{C}_0)(X, Z).$$

Par conséquent,

$$\mathcal{N}(h \circ m) = \mathcal{N}(h \circ m') \in \text{Arr}(\mathcal{C}_0)(X, Z).$$

Pas d'induction: on suppose le résultat démontré pour  $\mathcal{N}(h)$  de longueur  $n$ , et on le montre pour  $\mathcal{N}(h)$  de longueur  $n + 1$ . D'après le théorème 3.4,

$$\mathcal{N}(h) \equiv h' \circ \text{up}(A_0, A_1, A_2, A_3, f_1, f_2, g_1, g_2).$$

$$\begin{aligned} \mathcal{N}(h \circ m) &\equiv \mathcal{N}(h' \circ \text{up}(A_0, \dots, g_1, g_2) \circ \&_1(A_0, \dots, f_2) \circ f_1) \\ &\equiv \mathcal{N}(h' \circ g_1 \circ f_1) \\ \mathcal{N}(h \circ m') &\equiv \mathcal{N}(h' \circ \text{up}(A_0, \dots, g_1, g_2) \circ \&_2(A_0, \dots, f_2) \circ f_2) \\ &\equiv \mathcal{N}(h' \circ g_2 \circ f_2) \end{aligned}$$

Or, on a  $g_1 \circ f_1 = g_2 \circ f_2 \in \text{Arr}(\mathcal{C}_i)(X, Y)$ , et  $h'$  est une forme normale de longueur  $n$ , donc par hypothèse d'induction sur  $n$ ,

$$\mathcal{N}(h' \circ g_1 \circ f_1) = \mathcal{N}(h' \circ g_2 \circ f_2) \in \text{Arr}(\mathcal{C}_0)(X, Z).$$

Par conséquent,

$$\mathcal{N}(h \circ m) = \mathcal{N}(h \circ m') \in \text{Arr}(\mathcal{C}_0)(X, Z).$$

3. Il n'y a rien à démontrer.

4.  $X \equiv A_0 \equiv \text{push}(A'_0, A'_1, A'_2, f'_1, f'_2)$  et  $Y \equiv \text{push}(A_0, A_1, A_2, f_1, f_2)$ . Il faut montrer que

$$\forall W, Z \in \text{Obj}(\mathcal{C}_0), \forall p \in \text{Arr}(\mathcal{C}_{i+1})(W, X), \forall h \in \text{Arr}(\mathcal{C}_{i+1})(Y, Z),$$

$$\mathcal{N}(h \circ m \circ p) = \mathcal{N}(h \circ m' \circ p).$$

Remarquons d'abord que  $X \in \text{Obj}(\mathcal{C}_i)$ , donc d'après le théorème 3.3–2,  $\mathcal{N}(p) \in \text{Arr}(\mathcal{C}_i)(W, X)$ .

Cas de base sur la longueur de  $\mathcal{N}(h)$  :

$$\mathcal{N}(h) \equiv \text{up}(A_0, A_1, A_2, Z, f_1, f_2, g_1, g_2).$$

$$\begin{aligned} \mathcal{N}(h \circ m \circ p) &\equiv \mathcal{N}(\text{up}(A_0, \dots, g_1, g_2) \circ \&_1(A_0, \dots, f_2) \circ f_1 \circ p) \\ &\equiv \mathcal{N}(g_1 \circ f_1 \circ \mathcal{N}(p)) \\ \mathcal{N}(h \circ m' \circ p) &\equiv \mathcal{N}(\text{up}(A_0, \dots, g_1, g_2) \circ \&_2(A_0, \dots, f_2) \circ f_2 \circ p) \\ &\equiv \mathcal{N}(g_2 \circ f_2 \circ \mathcal{N}(p)) \end{aligned}$$

$\mathcal{N}(p) \in \text{Arr}(\mathcal{C}_i)$ ,  $g_1 \circ f_1 = g_2 \circ f_2 \in \text{Arr}(\mathcal{C}_i)(X, Z)$ , donc par hypothèse d'induction sur  $i$ , on a

$$\mathcal{N}(g_1 \circ f_1 \circ \mathcal{N}(p)) = \mathcal{N}(g_2 \circ f_2 \circ \mathcal{N}(p)) \in \text{Arr}(\mathcal{C}_0)(W, Z)$$

d'où

$$\mathcal{N}(h \circ m \circ p) = \mathcal{N}(h \circ m' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Z).$$

Pas d'induction: on suppose le résultat démontré pour  $\mathcal{N}(h)$  de longueur  $n$ , et on le montre pour  $\mathcal{N}(h)$  de longueur  $n + 1$ .

$$\mathcal{N}(h) \equiv h' \circ \text{up}(A_0, A_1, A_2, A_3, f_1, f_2, g_1, g_2)$$

$$\begin{aligned} \mathcal{N}(h \circ m \circ p) &\equiv \mathcal{N}(h' \circ \text{up}(A_0, \dots, g_1, g_2) \circ \&_1(A_0, \dots, f_2) \circ f_1 \circ p) \\ &\equiv \mathcal{N}(h' \circ g_1 \circ f_1 \circ p) \\ \mathcal{N}(h \circ m' \circ p) &\equiv \mathcal{N}(h' \circ \text{up}(A_0, \dots, g_1, g_2) \circ \&_2(A_0, \dots, f_2) \circ f_2 \circ p) \\ &\equiv \mathcal{N}(h' \circ g_2 \circ f_2 \circ p) \end{aligned}$$

On a  $g_1 \circ f_1 = g_2 \circ f_2 \in \text{Arr}(\mathcal{C}_i)(X, Z)$ ,  $h'$  est une forme normale de longueur  $n$ , donc par hypothèse d'induction sur  $n$ , on a

$$\mathcal{N}(h' \circ g_1 \circ f_1 \circ p) = \mathcal{N}(h' \circ g_2 \circ f_2 \circ p),$$

d'où

$$\mathcal{N}(h \circ m \circ p) = \mathcal{N}(h \circ m' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Z).$$

Règle (21)  $m \equiv \text{up}(A, B, C, D, f, g, f', g') \circ \&_1(A, B, C, f, g)$ ,  $m' \equiv f'$ .

1. On a

$$\mathcal{N}(m) \equiv \mathcal{N}(m').$$

2.  $\forall Z \in \text{Obj}(\mathcal{C}_0), \forall h \in \text{Arr}(\mathcal{C}_{i+1})(B, Z),$

$$\mathcal{N}(h \circ m) \equiv \mathcal{N}(h \circ m').$$

3.  $\forall W \in \text{Obj}(\mathcal{C}_0), \forall p \in \text{Arr}(\mathcal{C}_{i+1})(W, A),$

$$\mathcal{N}(m \circ p) \equiv \mathcal{N}(m' \circ p).$$

4.  $\forall W, Z \in \text{Obj}(\mathcal{C}_0), \forall p \in \text{Arr}(\mathcal{C}_{i+1})(W, A), \forall h \in \text{Arr}(\mathcal{C}_{i+1})(B, Z),$

$$\mathcal{N}(h \circ m \circ p) \equiv \mathcal{N}(h \circ m' \circ p).$$

*Règle (22)* Cas similaire au cas précédent.

*Règle (23)*  $m = m'$  parce que

$$\begin{aligned} m \circ \&_1(A_0, A_1, A_2, f_1, f_2) &= m' \circ \&_1(A_0, A_1, A_2, f_1, f_2) \in \text{Arr}(\mathcal{C}_{i+1})(A_1, Y), \\ m \circ \&_2(A_0, A_1, A_2, f_1, f_2) &= m' \circ \&_2(A_0, A_1, A_2, f_1, f_2) \in \text{Arr}(\mathcal{C}_{i+1})(A_2, Y). \end{aligned}$$

1. Il n'y a rien à démontrer.

2. Il n'y a rien à démontrer.

3.  $X \equiv \text{push}(A_0, A_1, A_2, f_1, f_2)$  et  $Y \in \text{Obj}(\mathcal{C}_0)$ . Il faut montrer que  
 $\forall W \in \text{Obj}(\mathcal{C}_0), \forall p \in \text{Arr}(\mathcal{C}_{i+1})(W, X),$

$$\mathcal{N}(m \circ p) = \mathcal{N}(m' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Y).$$

Soit  $\mathcal{N}(p) \equiv p_n \circ p_{n-1} \circ \cdots \circ p_1$ , avec  $n \geq 1$ . D'après le théorème 3.3,

$$p_n \equiv \&_k(A_0, A_1, A_2, f_1, f_2).$$

Si  $n = 1$ , on a par hypothèse d'induction

$$\begin{aligned} \mathcal{N}(m \circ \&_k(A_0, A_1, A_2, f_1, f_2)) \\ = \mathcal{N}(m' \circ \&_k(A_0, A_1, A_2, f_1, f_2)) \in \text{Arr}(\mathcal{C}_0)(W, Y) \end{aligned}$$

d'où

$$\mathcal{N}(m \circ p) = \mathcal{N}(m' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Y).$$

Si  $n \geq 2$ , on a par hypothèse d'induction

$$\begin{aligned} \mathcal{N}(m \circ \&_k(A_0, A_1, A_2, f_1, f_2) \circ p_{n-1} \circ \cdots \circ p_1) \\ = \mathcal{N}(m' \circ \&_k(A_0, A_1, A_2, f_1, f_2) \circ p_{n-1} \circ \cdots \circ p_1) \in \text{Arr}(\mathcal{C}_0)(W, Y) \end{aligned}$$

d'où

$$\mathcal{N}(m \circ p) = \mathcal{N}(m' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Y).$$

4.  $X \equiv \text{push}(A_0, A_1, A_2, f_1, f_2)$  et  $Y \equiv \text{push}(A'_0, A'_1, A'_2, f'_1, f'_2)$ . Il faut montrer que

$$\forall W, Z \in \text{Obj}(\mathcal{C}_0), \forall p \in \text{Arr}(\mathcal{C}_{i+1})(W, X), \forall h \in \text{Arr}(\mathcal{C}_{i+1})(Y, Z),$$

$$\mathcal{N}(h \circ m \circ p) = \mathcal{N}(h \circ m' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Z).$$

Soit  $\mathcal{N}(p) \equiv p_n \circ p_{n-1} \circ \cdots \circ p_1$ , avec  $n \geq 1$ . D'après le théorème 3.3,

$$p_n \equiv \&_k(A_0, A_1, A_2, f_1, f_2).$$

Si  $n = 1$ , on a par hypothèse d'induction

$$\begin{aligned} & \mathcal{N}(h \circ m \circ \&_k(A_0, A_1, A_2, f_1, f_2)) \\ &= \mathcal{N}(h \circ m' \circ \&_k(A_0, A_1, A_2, f_1, f_2)) \in \text{Arr}(\mathcal{C}_0)(W, Z) \end{aligned}$$

d'où

$$\mathcal{N}(h \circ m \circ p) = \mathcal{N}(h \circ m' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Z).$$

Si  $n \geq 2$ , on a par hypothèse d'induction

$$\begin{aligned} & \mathcal{N}(h \circ m \circ \&_k(A_0, A_1, A_2, f_1, f_2) \circ p_{n-1} \circ \cdots \circ p_1) \\ &= \mathcal{N}(h \circ m' \circ \&_k(A_0, A_1, A_2, f_1, f_2) \circ p_{n-1} \circ \cdots \circ p_1) \\ &\in \text{Arr}(\mathcal{C}_0)(W, Z) \end{aligned}$$

d'où

$$\mathcal{N}(h \circ m \circ p) = \mathcal{N}(h \circ m' \circ p) \in \text{Arr}(\mathcal{C}_0)(W, Z).$$

□

## A.4 Preuve du théorème 3.9

Nous montrons le théorème 3.9 page 139.

**Théorème 3.9** (La pré-catégorie  $\text{TERME}(\mathcal{C}_0)$  est librement engendrée par objet pré-initial choisi et pré-sommes amalgamées choisies)

Considérons une pré-catégorie  $\mathcal{E}$  ayant un objet pré-initial choisi  $\mathcal{O}^\mathcal{E}$  et des pré-sommes amalgamées choisies. Si  $A, B, C$  sont trois objets de  $\mathcal{E}$ , et  $f \in \text{Arr}(\mathcal{E})(A, B)$ ,  $g \in \text{Arr}(\mathcal{E})(A, C)$  deux flèches de  $\mathcal{E}$ , on note

$$(\text{push}^\mathcal{E}(A, B, C, f, g), \&_1^\mathcal{E}(A, B, C, f, g), \&_2^\mathcal{E}(A, B, C, f, g))$$

la pré-somme amalgamée choisie de  $B$  et  $C$  par rapport à  $f$  et  $g$  dans  $\mathcal{E}$ .

On suppose de plus que, pour tout objet  $A$  de  $\mathcal{E}$ , on a un choix de flèche

$$j_A^\mathcal{E} \in \text{Arr}(\mathcal{E})(\mathcal{O}^\mathcal{E}, A).$$

De même, si  $A, B, C, D \in \text{Obj}(\mathcal{E})$ ,  $f \in \text{Arr}(\mathcal{E})(A, B)$ ,  $g \in \text{Arr}(\mathcal{E})(A, C)$ ,  $f' \in \text{Arr}(\mathcal{E})(B, D)$ ,  $g' \in \text{Arr}(\mathcal{E})(C, D)$ , tel que  $f' \circ f = g' \circ g \in \text{Arr}(\mathcal{E})(A, D)$ , on a un choix de flèche

$$up^\mathcal{E}(A, B, C, D, f, g, f', g') \in \text{Arr}(\mathcal{E})(\text{push}^\mathcal{E}(A, B, C, f, g), D)$$

telle que

$$\begin{aligned} up^{\mathcal{E}}(A, B, C, D, f, g, f', g') \circ \&_1^{\mathcal{E}}(A, B, C, f, g) &= f' \in \text{Arr}(\mathcal{E})(B, D) \\ up^{\mathcal{E}}(A, B, C, D, f, g, f', g') \circ \&_2^{\mathcal{E}}(A, B, C, f, g) &= g' \in \text{Arr}(\mathcal{E})(C, D). \end{aligned}$$

Soit un préfoncteur  $F : \mathcal{C}_0 \rightarrow \mathcal{E}$ . Alors il existe un unique préfoncteur

$$G : \text{TERME}(\mathcal{C}_0) \rightarrow \mathcal{E}$$

tel que

- $G \circ J \equiv F$  ;
- $G(g \circ f) \equiv G(g) \circ G(f)$  ;
- $G(id_A) \equiv id_{G(A)}$  ;
- $G(\emptyset) \equiv \emptyset^{\mathcal{E}}$  ;
- $G(j_A) \equiv j_{G(A)}^{\mathcal{E}}$  ;
- $G(push(A, B, C, f, g)) \equiv push^{\mathcal{E}}(G(A), G(B), G(C), G(f), G(g))$  ;
- $G(\&_1(A, B, C, f, g)) \equiv \&_1^{\mathcal{E}}(G(A), G(B), G(C), G(f), G(g))$  ;
- $G(\&_2(A, B, C, f, g)) \equiv \&_2^{\mathcal{E}}(G(A), G(B), G(C), G(f), G(g))$  ;
- $G(up(A, B, C, D, f, g, f', g')) \equiv up^{\mathcal{E}}(G(A), G(B), G(C), G(D), G(f), G(g), G(f'), G(g'))$ .

Nous commençons par montrer l'existence du préfoncteur

$$G : \text{TERME}(\mathcal{C}_0) \rightarrow \mathcal{E}.$$

Pour cela, nous définissons pour tout entier  $i$  un préfoncteur  $G_i : \mathcal{C}_i \rightarrow \mathcal{E}$ , par induction sur  $i$ . Nous posons  $G_0 \equiv F$ , et définissons le préfoncteur  $G_{i+1} : \mathcal{C}_{i+1} \rightarrow \mathcal{E}$  à partir du préfoncteur  $G_i : \mathcal{C}_i \rightarrow \mathcal{E}$ .

### Hypothèses d'induction

Pour  $i \in \mathbb{N}$ , on fait les hypothèses d'induction suivantes.

1. Si  $i \geq 1$ ,
  - (a)  $\forall A \in \text{Obj}(\mathcal{C}_{i-1}), G_i(A) \equiv G_{i-1}(A)$  ;
  - (b)  $\forall A, B \in \text{Obj}(\mathcal{C}_{i-1}), \forall f \in \text{Arr}(\mathcal{C}_{i-1})(A, B), G_i(f) \equiv G_{i-1}(f)$  ;
2.  $h = h' \in \text{Arr}(\mathcal{C}_i)(A, B) \Rightarrow G_i(h) = G_i(h') \in \text{Arr}(\mathcal{E})(G_i(A), G_i(B))$  ;
3.  $G_i : \mathcal{C}_i \rightarrow \mathcal{E}$  est un préfoncteur.

Pour  $i = 0$ , on pose  $G_0 \equiv F : \mathcal{C}_0 \rightarrow \mathcal{E}$ . Les hypothèses d'induction sont bien vérifiées.

Nous définissons maintenant le préfoncteur  $G_{i+1} : \mathcal{C}_{i+1} \rightarrow \mathcal{E}$ , en supposant que les hypothèses d'induction sont satisfaites au rang  $i$ .

**Définition du préfoncteur**  $G_{i+1} : \mathcal{C}_{i+1} \rightarrow \mathcal{E}$ 

Le préfoncteur  $G_{i+1}$  est défini à partir de  $G_i$ , par induction sur la structure des objets et des flèches de  $\mathcal{C}_{i+1}$ .

- Action sur les objets :

$$\text{Règle (1) } G_{i+1}(A) \equiv G_0(A), \text{ pour } A \in \text{Obj}(\mathcal{C}_0);$$

$$\text{Règle (2) } G_{i+1}(\emptyset) \equiv \emptyset^{\mathcal{E}};$$

$$\begin{aligned} \text{Règle (3) } G_{i+1}(\text{push}(A, B, C, f, g)) \\ \equiv \text{push}^{\mathcal{E}}(G_i(A), G_i(B), G_i(C), G_i(f), G_i(g)). \end{aligned}$$

- Action sur les flèches :

$$\text{Règle (4) } G_{i+1}(f) \equiv G_0(f), \text{ pour } f \in \text{Arr}(\mathcal{C}_0)(A, B);$$

$$\text{Règle (5) } G_{i+1}(g \circ f) \equiv G_{i+1}(g) \circ G_{i+1}(f);$$

$$\text{Règle (6) } G_{i+1}(\text{id}_A) \equiv \text{id}_{G_{i+1}(A)}, \text{ identité de } \mathcal{E};$$

$$\text{Règle (7) } G_{i+1}(j_A) \equiv j_{G_{i+1}(A)}^{\mathcal{E}};$$

$$\text{Règle (8) } G_{i+1}(\&_1(A, B, C, f, g)) \equiv \&_1^{\mathcal{E}}(G_i(A), G_i(B), G_i(C), G_i(f), G_i(g));$$

$$\text{Règle (9) } G_{i+1}(\&_2(A, B, C, f, g)) \equiv \&_2^{\mathcal{E}}(G_i(A), G_i(B), G_i(C), G_i(f), G_i(g));$$

$$\begin{aligned} \text{Règle (10) } G_{i+1}(\text{up}(A, B, C, D, f, g, f', g')) \\ \equiv \text{up}^{\mathcal{E}}(G_i(A), G_i(B), G_i(C), G_i(D), G_i(f), G_i(g), G_i(f'), G_i(g')). \end{aligned}$$

Cette flèche existe car on a  $f' \circ f = g' \circ g \in \text{Arr}(\mathcal{C}_i)(A, D)$ . Par hypothèse d'induction 3,  $G_i$  est un préfoncteur, donc

$$G_i(f') \circ G_i(f) = G_i(g') \circ G_i(g) \in \text{Arr}(\mathcal{E})(G_i(A), G_i(D)).$$

**Pas d'induction**

Nous montrons maintenant que les hypothèses d'induction sont bien vérifiées au rang  $i + 1$ , c'est-à-dire que

1. (a)  $\forall A \in \text{Obj}(\mathcal{C}_i), G_{i+1}(A) \equiv G_i(A);$   
 (b)  $\forall A, B \in \text{Obj}(\mathcal{C}_i), \forall f \in \text{Arr}(\mathcal{C}_i)(A, B), G_{i+1}(f) \equiv G_i(f);$
2.  $h = h' \in \text{Arr}(\mathcal{C}_{i+1})(A, B) \Rightarrow G_{i+1}(h) = G_{i+1}(h') \in \text{Arr}(\mathcal{E})(G_{i+1}(A), G_{i+1}(B));$
3.  $G_{i+1} : \mathcal{C}_{i+1} \rightarrow \mathcal{E}$  est un préfoncteur.

*Preuve.*

*Point 1.* On montre le point 1 (a) par induction sur la structure des objets de  $\mathcal{C}_{i+1}$ , le point 1 (b) par induction sur la structure des flèches de  $\mathcal{C}_{i+1}$ .

*Point 2.* On montre le point 2 par induction sur la longueur de la preuve que  $h = h' \in \text{Arr}(\mathcal{C}_{i+1})(A, B)$ .

*Règle (11)* Si  $h = h' \in \text{Arr}(\mathcal{C}_0)(A, B)$ , alors  $G_0(h) = G_0(h')$ , et donc, par définition de  $G_{i+1}$ ,  $G_{i+1}(h) = G_{i+1}(h')$ .

*Règle (12)* Si  $h \equiv h'$ , alors on a évidemment  $G_{i+1}(h) \equiv G_{i+1}(h')$ , d'où le résultat.

*Règle (13)* Si  $h' = h \in \text{Arr}(\mathcal{C}_{i+1})(A, B)$ , alors par hypothèse d'induction,  $G_{i+1}(h') = G_{i+1}(h)$ , d'où  $G_{i+1}(h) = G_{i+1}(h')$ .

*Règle (14)* Si  $h = g \in \text{Arr}(\mathcal{C}_{i+1})(A, B)$  et  $g = h' \in \text{Arr}(\mathcal{C}_{i+1})(A, B)$ , alors, par hypothèse d'induction, on a  $G_{i+1}(h) = G_{i+1}(g)$  et  $G_{i+1}(g) = G_{i+1}(h')$ , d'où  $G_{i+1}(h) = G_{i+1}(h')$ .

*Règle (15)* On suppose que  $h \equiv g \circ f$ ,  $h' \equiv g' \circ f'$ ,  $f = f' \in \text{Arr}(\mathcal{C}_{i+1})(A, X)$ , et  $g = g' \in \text{Arr}(\mathcal{C}_{i+1})(X, B)$ . Par hypothèse d'induction, on a  $G_{i+1}(f) = G_{i+1}(f')$  et  $G_{i+1}(g) = G_{i+1}(g')$ . Donc  $G_{i+1}(g) \circ G_{i+1}(f) = G_{i+1}(g') \circ G_{i+1}(f')$ , et donc, par définition de  $G_{i+1}$ ,  $G_{i+1}(g \circ f) = G_{i+1}(g' \circ f')$ , d'où  $G_{i+1}(h) = G_{i+1}(h')$ .

*Règle (16)* Si  $h \equiv (k \circ g) \circ f$  et  $h' \equiv k \circ (g \circ f)$ , par définition de  $G_{i+1}$ ,

$$\begin{aligned} G_{i+1}(h) &\equiv G_{i+1}((k \circ g) \circ f) \\ &\equiv G_{i+1}(k \circ g) \circ G_{i+1}(f) \\ &\equiv (G_{i+1}(k) \circ G_{i+1}(g)) \circ G_{i+1}(f) \\ &\equiv G_{i+1}(k) \circ (G_{i+1}(g) \circ G_{i+1}(f)) \\ &\equiv G_{i+1}(k \circ (g \circ f)) \\ &\equiv G_{i+1}(h') \end{aligned}$$

*Règle (17)* Si  $h \equiv f \circ \text{id}_A$ , et  $h' \equiv f$ , on a

$$\begin{aligned} G_{i+1}(h) &\equiv G_{i+1}(f \circ \text{id}_A) \\ &\equiv G_{i+1}(f) \circ G_{i+1}(\text{id}_A) \quad (\text{définition de } G_{i+1}) \\ &\equiv G_{i+1}(f) \circ \text{id}_{G_{i+1}(A)} \quad (\text{définition de } G_{i+1}) \\ &= G_{i+1}(f) \quad (\text{identité dans } \mathcal{E}) \\ &\equiv G_{i+1}(h') \end{aligned}$$

*Règle (18)* Preuve similaire à celle de la règle (17).

*Règle (19)* Si  $h, h' \in \text{Arr}(\mathcal{C}_{i+1})(\emptyset, A)$ , comme  $\emptyset^\mathcal{E}$  est pré-initial,  $G_{i+1}(h) = G_{i+1}(h')$ .

*Règle (20)*  $h \equiv \&_1(A, B, C, f, g) \circ f$  et  $h' \equiv \&_2(A, B, C, f, g) \circ g$ .

$$\begin{aligned} G_{i+1}(h) &\equiv G_{i+1}(\&_1(A, B, C, f, g) \circ f) \\ &\equiv G_{i+1}(\&_1(A, B, C, f, g)) \circ G_{i+1}(f) \\ &\equiv G_{i+1}(\&_1(A, B, C, f, g)) \circ G_i(f) && (\text{d'après 1 (b)}) \\ &\equiv \&_1^\mathcal{E}(G_i(A), G_i(B), G_i(C), G_i(f), G_i(g)) \circ G_i(f) \\ \\ G_{i+1}(h') &\equiv G_{i+1}(\&_2(A, B, C, f, g) \circ g) \\ &\equiv G_{i+1}(\&_2(A, B, C, f, g)) \circ G_{i+1}(g) \\ &\equiv G_{i+1}(\&_2(A, B, C, f, g)) \circ G_i(g) && (\text{d'après 1 (b)}) \\ &\equiv \&_2^\mathcal{E}(G_i(A), G_i(B), G_i(C), G_i(f), G_i(g)) \circ G_i(g) \end{aligned}$$

Comme le triplet

$$\left( \begin{array}{l} \text{push}^{\mathcal{E}}(G_i(A), G_i(B), G_i(C), G_i(f), G_i(g)), \\ \&_1^{\mathcal{E}}(G_i(A), G_i(B), G_i(C), G_i(f), G_i(g)), \\ \&_2^{\mathcal{E}}(G_i(A), G_i(B), G_i(C), G_i(f), G_i(g)) \end{array} \right)$$

est une pré-somme amalgamée dans  $\mathcal{E}$ , on a

$$\begin{aligned} & \&_1^{\mathcal{E}}(G_i(A), G_i(B), G_i(C), G_i(f), G_i(g)) \circ G_i(f) \\ &= \&_2^{\mathcal{E}}(G_i(A), G_i(B), G_i(C), G_i(f), G_i(g)) \circ G_i(g), \end{aligned}$$

et donc  $G_{i+1}(h) = G_{i+1}(h')$ .

Règle (21)  $h \equiv \text{up}(A, B, C, D, f, g, f', g') \circ \&_1(A, B, C, f, g)$  et  $h' \equiv f'$ .

$$\begin{aligned} G_{i+1}(h) &\equiv G_{i+1}(\text{up}(A, B, C, D, f, g, f', g') \circ \&_1(A, B, C, f, g)) \\ &\equiv G_{i+1}(\text{up}(A, B, C, D, f, g, f', g')) \circ G_{i+1}(\&_1(A, B, C, f, g)) \\ &= G_i(f') \\ &\equiv G_{i+1}(f') \end{aligned}$$

d'après la définition de  $G_{i+1}(\text{up}(A, B, C, D, f, g, f', g'))$ .

Règle (22) Preuve similaire à celle de la règle (21).

Règle (23)  $h \equiv u$  et  $h \equiv v$ , avec

$$\begin{aligned} u \circ \&_1(A, B, C, f, g) &= v \circ \&_1(A, B, C, f, g) \in \text{Arr}(\mathcal{C}_{i+1})(B, D) \\ u \circ \&_2(A, B, C, f, g) &= v \circ \&_2(A, B, C, f, g) \in \text{Arr}(\mathcal{C}_{i+1})(C, D). \end{aligned}$$

Par hypothèse d'induction, on a

$$\begin{aligned} G_{i+1}(u \circ \&_1(A, B, C, f, g)) &= G_{i+1}(v \circ \&_1(A, B, C, f, g)) \\ G_{i+1}(u \circ \&_2(A, B, C, f, g)) &= G_{i+1}(v \circ \&_2(A, B, C, f, g)) \end{aligned}$$

et donc

$$\begin{aligned} G_{i+1}(u) \circ G_{i+1}(\&_1(A, B, C, f, g)) &= G_{i+1}(v) \circ G_{i+1}(\&_2(A, B, C, f, g)) \\ G_{i+1}(u) \circ G_{i+1}(\&_1(A, B, C, f, g)) &= G_{i+1}(v) \circ G_{i+1}(\&_2(A, B, C, f, g)). \end{aligned}$$

Par conséquent, par définition de  $G_{i+1}(\text{up}(A, B, C, D, f, g, f', g'))$ , on a

$$G_{i+1}(u) = G_{i+1}(v).$$

Cela termine la preuve du point 2.

Point 3. On montre sans difficulté que  $G_{i+1} : \mathcal{C}_{i+1} \rightarrow \mathcal{E}$  est bien un préfoncteur.  $\square$

Nous avons donc montré le résultat suivant.

**Lemme A.1** *Pour tout entier  $i$ ,*

1. (a)  $\forall A \in \text{Obj}(\mathcal{C}_i), G_{i+1}(A) \equiv G_i(A)$  ;  
 (b)  $\forall A, B \in \text{Obj}(\mathcal{C}_i), \forall f \in \text{Arr}(\mathcal{C}_i)(A, B), G_{i+1}(f) \equiv G_i(f)$  ;
2.  $G_i : \mathcal{C}_i \rightarrow \mathcal{E}$  est un préfoncteur.

**Définition du préfoncteur  $G : \text{TERME}(\mathcal{C}_0) \rightarrow \mathcal{E}$** 

On définit maintenant le préfoncteur  $G : \text{TERME}(\mathcal{C}_0) \rightarrow \mathcal{E}$  par son action sur les objets et les flèches de  $\text{TERME}(\mathcal{C}_0)$ .

- Action sur les objets.

Pour tout  $A \in \text{Obj}(\text{TERME}(\mathcal{C}_0))$ , il existe  $k \in \mathbb{N}$  tel que  $A \in \text{Obj}(\mathcal{C}_k)$ . On pose

$$G(A) \equiv G_k(A).$$

D'après le lemme A.1 (a), cette définition est indépendante de l'entier  $k$  choisi.

- Action sur les flèches.

Pour tout  $f \in \text{Arr}(\text{TERME}(\mathcal{C}_0))$ , il existe  $k \in \mathbb{N}$  tel que  $f \in \text{Arr}(\mathcal{C}_k)(A, B)$ . On pose

$$G(f) \equiv G_k(f).$$

D'après le lemme A.1 (b), cette définition est indépendante de l'entier  $k$  choisi.

- Compatibilité avec les équivalences.

Si  $h = h' \in \text{Arr}(\text{TERME}(\mathcal{C}_0))(A, B)$ , alors il existe  $k \in \mathbb{N}$  tel que  $h = h' \in \text{Arr}(\mathcal{C}_k)(A, B)$ . Comme  $G_k$  est un préfoncteur,  $G_k(h) = G_k(h')$ . Par conséquent,  $G(h) = G(h')$ .

On vérifie que  $G$  est bien un préfoncteur tel que

- $G \circ J \equiv F$  ;
- $G(g \circ f) \equiv G(g) \circ G(f)$  ;
- $G(\text{id}_A) \equiv \text{id}_{G(A)}$  ;
- $G(\emptyset) \equiv \emptyset^{\mathcal{E}}$  ;
- $G(j_A) \equiv j_{G(A)}^{\mathcal{E}}$  ;
- $G(\text{push}(A, B, C, f, g)) \equiv \text{push}^{\mathcal{E}}(G(A), G(B), G(C), G(f), G(g))$  ;
- $G(\&_1(A, B, C, f, g)) \equiv \&_1^{\mathcal{E}}(G(A), G(B), G(C), G(f), G(g))$  ;
- $G(\&_2(A, B, C, f, g)) \equiv \&_2^{\mathcal{E}}(G(A), G(B), G(C), G(f), G(g))$  ;
- $G(\text{up}(A, B, C, D, f, g, f', g')) \equiv \text{up}^{\mathcal{E}}(G(A), G(B), G(C), G(D), G(f), G(g), G(f'), G(g'))$ .

**Unicité du préfoncteur  $G$** 

Soit un préfoncteur  $G' : \text{TERME}(\mathcal{C}_0) \rightarrow \mathcal{E}$  tel que

- $G' \circ J \equiv F$  ;
- $G'(g \circ f) \equiv G'(g) \circ G'(f)$  ;

- $G'(\text{id}_A) \equiv \text{id}_{G'(A)}$  ;
- $G'(\emptyset) \equiv \emptyset^{\mathcal{E}}$  ;
- $G'(j_A) \equiv j_{G'(A)}^{\mathcal{E}}$  ;
- $G'(\text{push}(A, B, C, f, g)) \equiv \text{push}^{\mathcal{E}}(G'(A), G'(B), G'(C), G'(f), G'(g))$  ;
- $G'(\&_1(A, B, C, f, g)) \equiv \&_1^{\mathcal{E}}(G'(A), G'(B), G'(C), G'(f), G'(g))$  ;
- $G'(\&_2(A, B, C, f, g)) \equiv \&_2^{\mathcal{E}}(G'(A), G'(B), G'(C), G'(f), G'(g))$  ;
- $G'(\text{up}(A, B, C, D, f, g, f', g'))$   
 $\equiv \text{up}^{\mathcal{E}}(G'(A), G'(B), G'(C), G'(D), G'(f), G'(g), G'(f'), G'(g'))$ .

Par induction sur la structure des objets et flèches de  $\text{TERME}(\mathcal{C}_0)$ , on a bien  $G \equiv G'$ .



# Références bibliographiques

- [AL91] A. Asperti and G. Longo. *Categories, Types and Structures, An Introduction to Category Theory for the Working Computer Scientist*. Foundations of Computing Science, MIT Press, 1991.
- [AM75] M. A. Arbib and E. G. Manes. *Arrows, Structures and Functors: The Categorical Imperative*. Academic Press, 1975.
- [B<sup>+</sup>90] D. Bert et al. Reference manual of the specification language LPG. Technical Report 59, LIFIA, Mars 1990. Anonymous ftp at `imag.fr`, in `/pub/SC0P/LPG/NewSun4/man_lpg.dvi`.
- [BBC86] G. Bernot, M. Bidoit, and C. Choppy. Abstract data types with exception handling: An initial approach based on a distinction between exception and errors. *Theoretical Computer Science*, 46(1):13–45, 1986.
- [BDMN73] G. Birtwistle, O.-J. Dahl, B. Myrhaug, and K. Nygaard. *Simula Begin*. Auerbach Pub., New York, 1973.
- [BE86] D. Bert and R. Echahed. Design and implementation of a generic, logic and functional programming language. In *Proceedings of ESOP'86*, number 213 in LNCS, pages 119–132. Springer-Verlag, 1986.
- [Ber83] D. Bert. Refinements of generic specifications with algebraic tools. In *Proceedings of IFIP'83, Paris*, pages 815–820, 1983.
- [Ber90] D. Bert. Spécification de logiciels réutilisables. Technical Report RR-828-I-IMAG-116, LIFIA, Octobre 1990.
- [BG77] R. M. Burstall and J. A. Goguen. Putting theories together to make specifications. In *Proceedings of the 5<sup>th</sup> International Joint Conference on Artificial Intelligence*, pages 1045–1058, 1977.
- [BG80] R. M. Burstall and J. A. Goguen. The semantics of CLEAR, a specification language. In *Proceedings of Advanced Course on Abstract Software Specification*, number 86 in LNCS, pages 292–332. Springer-Verlag, 1980.
- [BHK90] J. A. Bergstra, J. Heering, and R. Klint. Module algebra. *J. ACM*, 37(2):335–372, Apr. 1990.

- [Bid89] M. Bidoit. PLUSS, un langage pour le développement de spécifications algébriques modulaires, Mai 1989. Thèse d'État, Université de Paris-Sud.
- [BL93] G. Bernot and P. Le Gall. Exception handling and term labelling. In *Proceedings of TAPSOFT'93*, number 668 in LNCS, pages 421–436. Springer-Verlag, 1993.
- [BO95] D. Bert and C. Oriat. A model inference system for generic specification with application to code sharing. In *Proceedings of TAPSOFT'95*, number 915 in LNCS, pages 741–755. Springer-Verlag, 1995.
- [BR86] R. M. Burstall and D. Rydeheard. Computing with categories. Technical Report ECS-LFCS-86-9, University of Edinburgh, September 1986.
- [Bur80] R. M. Burstall. Electronic category theory. In *Proceedings of the 9<sup>th</sup> Symposium on Mathematical Foundations of Computer Science*, 1980.
- [BW85] S. L. Bloom and E. G. Wagner. Many-sorted theories and their algebras with some applications to data types. In M. Nivat and J. C. Reynolds, editors, *Algebraic Methods in Semantics*, chapter 4, pages 133–168. Cambridge University Press, 1985.
- [BW90] M. Barr and C. Wells. *Category Theory for Computing Science*. Prentice-Hall International, 1990.
- [BW94] M. Barr and C. Wells. The categorical theory generated by a limit sketch, Nov. 1994.
- [Car86] J. Cartmell. Generalized algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32:209–243, 1986.
- [Coh65] P. M. Cohn. *Universal Algebra*. Harper and Row, 1965. Revised version 1980.
- [Cur91] F. Cury. Catégories lax-localement-cartésiennes et catégories localement cartésiennes : un exemple de suffisante complétude connexe de sémantiques initiales. In *diagrammes*, volume 25, pages 1–155, Université Paris 7, Juillet 1991.
- [DJ90] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In *Handbook of Theoretical Computer Science*, chapter 15. Elsevier Science Publishers B.V., 1990.
- [DN66] O.-J. Dahl and K. Nygaard. Simula – an Algol-based simulation language. *Communications of the ACM*, 9:671–678, Sept. 1966.
- [DR94a] D. Duval and J.-C. Reynaud. Sketches and computation (part 1): Basic definitions and static evaluation. *Mathematical Structures in Computer Science*, 4:185–238, 1994.

- [DR94b] D. Duval and J.-C. Reynaud. Sketches and computation (part 2): Dynamic evaluation and applications. *Mathematical Structures in Computer Science*, 4:239–271, 1994.
- [Ehr65] C. Ehresmann. *Catégories et structures*. Dunod, Paris, 1965.
- [Ehr68] C. Ehresmann. Esquisses et types de structures algébriques. *Bulletin de l'Institut Polytechnique*, Iasi 14, 1968.
- [EJO93] H. Ehrig, R. M. Jimenez, and F. Orejas. Compositionality results for different types of parameterization and parameter passing in specification languages. In *Proceedings of the 4<sup>th</sup> International Joint Conference CAAP/FASE*, number 668 in LNCS, pages 31–45. Springer-Verlag, 1993.
- [EM85] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1. Equations and Initial Semantics*, volume 6 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1985.
- [EM90] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 2. Module Specifications and Constraints*, volume 21 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1990.
- [FGJM85] K. Futatsugi, J. A. Goguen, J.-P. Jouannaud, and J. Meseguer. Principles of OBJ2. In *Proceedings of Principles of Programming Languages*, pages 52–66, 1985.
- [Gau84] M.-C. Gaudel. A first introduction to PLUSS. Technical report, Université d'Orsay, France, 1984.
- [GB84] J. A. Goguen and R. M. Burstall. Introducing institutions. In *Proceedings of the Workshop on Logic of Programming*, number 164 in LNCS, pages 221–256. Springer-Verlag, 1984.
- [GB90] J. A. Goguen and R. M. Burstall. Institutions: Abstract model theory for specification and programming. Research Report ECS-LFCS-90-106, University of Edinburgh, January 1990.
- [GKK<sup>+</sup>87] J. A. Goguen, C. Kirchner, H. Kirchner, A. Mégard, J. Meseguer, and T. Winkler. An introduction to OBJ3. In *Proceedings of the 1<sup>st</sup> International Workshop on Conditional Term Rewriting Systems*, number 308 in LNCS, pages 258–263. Springer-Verlag, 1987.
- [Gog73] J. A. Goguen. Categorical foundations for general systems theory. In *Advances in Cybernetics and System Research*, pages 121–130. Transcripita Books, 1973.
- [Gog92] J. A. Goguen. Sheaf semantics for concurrent interacting objects. *Mathematical Structures in Computer Science*, 2:159–191, 1992.

- [Gol79] R. Goldblatt. *Topoi: The Categorical Analysis of Logic*, volume 98 of *Studies in Logic and the Foundations of Mathematics*. North Holland, 1979.
- [GTW78] J. A. Goguen, J. W. Thatcher, and E. G. Wagner. An initial algebra approach to the specification, correctness, and implementation of abstract data types. In R. T. Yeh, editor, *Current Trends in Programming Methodology*, volume 4: Data Structuring, pages 80–149. Prentice-Hall, 1978.
- [GTWW75] J. A. Goguen, J. W. Thatcher, E. G. Wagner, and J. B. Wright. Abstract data types as initial algebras and the correctness of data representation. In *Computer Graphics, Pattern Recognition and Data Structure*, pages 89–93, 1975.
- [GTWW77] J. A. Goguen, J. W. Thatcher, E. G. Wagner, and J. B. Wright. Initial algebra semantics and continuous algebra. *J. ACM*, 24:68–95, 1977.
- [Gut75] J. V. Guttag. *The specification and application to programming of abstract data types*. PhD thesis, University of Toronto, 1975.
- [Gut77] J. V. Guttag. Abstract data types and the development of data structures. *Communication of the ACM*, 6:396–404, 1977.
- [Huf92] J.-M. Hufflen. Proposal for GLIDER version 1.0: Principles and main features. ICARUS Technical Report, INRIA-LORRAINE & CRIN, 1992.
- [Law63] W. Lawvere. Functorial semantics of algebraic theories. *Proc. Nat. Acad. Sci.*, 50:869–873, 1963.
- [LCW85] D. Lorge Parnas, P. C. Clements, and D. M. Weiss. The modular structure of complex systems. *IEEE Transactions on Software Engineering*, SE-11(3):259–266, March 1985.
- [LS86] J. Lambek and P. J. Scott. *Introduction to higher order categorical logic*. Cambridge studies in advanced mathematics, 1986.
- [LZ74] B. Liskov and S. Zilles. Programming with abstract data types. *ACM SIGPLAN Notices*, 9(4):50–59, 1974.
- [Mar95] A. Martins. *La généralisation: un outil pour la réutilisation*. PhD thesis, INPG, Grenoble, Mars 1995.
- [MB70] S. Mac Lane and G. Birkhoff. *Algèbre. 1. Structures fondamentales*. Gauthier-Villars, 1970.
- [McL71] S. Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, 1971.

- [MG85] J. Meseguer and J. A. Goguen. Initiality, induction, and computability. In M. Nivat and J. C. Reynolds, editors, *Algebraic Methods in Semantics*, chapter 14, pages 459–541. Cambridge University Press, 1985.
- [Ori94] C. Oriat. Representing modular specifications as diagrams. In *Compte Rendu des Journées du GDR Programmation, Lille*, pages 53–68, Septembre 1994.
- [Ori95] C. Oriat. Detecting isomorphisms of modular specifications with diagrams. In *Proceedings of AMAST'95*, number 936 in LNCS, pages 184–198. Springer-Verlag, 1995.
- [Poi92] A. Poigné. Basic category theory. In *Handbook of Logic in Computer Science, Volume 1. Background: Mathematical Structures*, pages 413–640. Oxford Science Publication, 1992.
- [Ren91] G. Renardel de Lavalette. Logical semantics of modularisation. In *Proceedings of CSL'91*, number 626 in LNCS, pages 306–315. Springer-Verlag, 1991.
- [Rey87] J.-C. Reynaud. Sémantique de LPG. Research Report 651 I IMAG, LIFIA, Mars 1987.
- [Rey90a] J.-C. Reynaud. Putting algebraic components together: A dependent type approach. Research Report 810 I IMAG, LIFIA, Avril 1990. Extended version.
- [Rey90b] J.-C. Reynaud. Putting algebraic components together: A dependent type approach. In *Proceedings of DISCO'90*, number 429 in LNCS. Springer-Verlag, 1990.
- [Rey93] J.-C. Reynaud. Isomorphism of typed algebraic specifications. Internal Report, LGI-IMAG, Avril 1993.
- [See84] R. A. G. Seely. Locally cartesian closed categories and type theory. *Math. Proc. Camb. Phil. Soc.*, 95(33):33–48, 1984.
- [SP77] M. B. Smyth and G. D. Plotkin. The category-theoretic solution of recursive domain equations. In *Proceedings of the 18<sup>th</sup> IEEE Symposium on Foundations of Computer Science*, pages 13–17, 1977.
- [SST92] D. Sannella, S. Sokolowski, and A. Tarlecki. Toward formal development of programs from algebraic specifications: Parameterisation revisited. *Acta Informatica*, 29:689–736, 1992.
- [ST88] D. Sannella and A. Tarlecki. Specifications in an arbitrary institution. *Information and Computation*, 76:165–210, 1988.
- [SW83] D. Sannella and M. Wirsing. A kernal language for algebraic specification and implementation. In *Proceedings of the 11<sup>th</sup> Colloquium on Foundations of Computation Theory*, number 158 in LNCS, pages 413–427. Springer-Verlag, 1983.

- [SW91] T. Streicher and M. Wirsing. Dependent types considered necessary for specification languages. In *Proceedings of the 7<sup>th</sup> Workshop on Specification of Abstract Data Types*, number 534 in LNCS, pages 323–339, 1991.
- [TBG91] A. Tarlecki, R. M. Burstall, and J. A. Goguen. Some fundamental algebraic tools for the semantics of computation: Part 3. Indexed categories. *Theoretical Computer Science*, 91:239–264, 1991.
- [TWW82] J. W. Thatcher, E. G. Wagner, and J. B. Wright. Data type specification: Parameterization and the power of specification techniques. *ACM Trans. Prog. Lang. Syst.*, 4:711–773, 1982.
- [Wan79] M. Wand. Fixed-point constructions in order-enriched categories. *Theoretical Computer Science*, 8:13–30, 1979.
- [WBT85] E. G. Wagner, S. L. Bloom, and J. W. Thatcher. Why algebraic theories? In M. Nivat and J. C. Reynolds, editors, *Algebraic Methods in Semantics*, chapter 17, pages 607–634. Cambridge University Press, 1985.
- [Wir86] M. Wirsing. Structured algebraic specifications: A kernel language. *Theoretical Computer Science*, 42:123–249, 1986.
- [Wir90] M. Wirsing. Algebraic specification. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, chapter 13, pages 677–788. Elsevier Science Publishers B.V., 1990.
- [Wir94] M. Wirsing. Algebraic specification languages: An overview. In *Recent Trends in Data Type Specification, 10<sup>th</sup> Workshop on Specification of Abstract Data Types*, number 906 in LNCS, 1994.

# Index

- $\subseteq$ , 190
- $0_n$ , 56
- $\approx$ , 96
- $\cdot$ , 61
- $\cong$ , 58, 61
- $\mathcal{D}^c$ , 61
- $=$ , 120
- $\varepsilon$ , 23
- $\sim_{\delta}$ , 70
- $\equiv$ , 120, 121
- $\&_1 \&_2$ , 42, 124, 126
- $\longrightarrow$ , 130
- $<$ , 190
- $\leq$ , 190
- $-^*$ , 23
- $-^+$ , 23
- $\emptyset$ , 42, 124, 126
- Alg, 27
- Arr, 119, 126
- $j_A$ , 42, 126
- $\mathcal{N}$ , 132
- Obj, 119, 126
- push, 42, 124, 126
- $\mathcal{R}$ , 119, 127
- $T_{\Sigma}, T_{\Sigma}(X)$ , 27
- up, 43, 124, 126
  
- adjonction, 63
- affectation, 28
- ajout
  - d'un arc, 180
  - d'une composition, 181
  - d'une factorisation à droite, 182
- algèbre, 27
  - des termes, 27
  - libre, 28
  - satisfaisant une équation, 28
- anneau, 31
  
- aplatissement, 88
- application
  - $S$ - —, 23
- arc, 55
  - identité, 183
  
- boucle, 177
  - identité, 177
- but, 55
  
- CAT**, 59
- Cat**, 59
- catégorie, 14, 58
  - associée à une précatégorie, 120
  - de base, 117
  - de diagrammes, 70
  - des catégories, 59
  - des ensembles, 37, 58
  - des foncteurs, 61
  - des graphes, 58
  - des signatures, 26, 52, 58
  - des spécifications, 30, 52, 58
  - diagr( $\mathcal{C}$ ), 97
  - DIAGR( $\mathcal{C}$ ), 70
  - finiment cocomplète, 78
  - finiment pré-cocomplète, 123
  - $\mathcal{L}(\mathcal{C}_0)$ , 146
  - librement engendrée sur un graphe, 59
  - Terme( $\mathcal{C}_0$ ), 140
  - petite —, 58
  - super-comma —, 68
  
- chaîne, 23
- choix
  - de codage, 36, 38
  - de colimites, 75, 118
- codomaine, 58
- coégalisateur, 67, 76
- colimite, 34, 72–86

- cône —, 73
- flèche —, 79
- objet —, 74
- combinaison, 33
- complétion
  - d'un diagramme, 187
  - d'une catégorie, 109
  - théorème de —, 114
- composition
  - dans  $\text{DIAGR}(\mathcal{C})$ , 71
  - de flèches, 58
  - de foncteurs, 59
  - de morphismes de graphes, 56
  - de spécifications, 33
  - de transformations naturelles, 61
- cône, 73
  - colimite, 73
- connexion, 70
- contraction
  - d'un arc identité, 183
- coüinité, 64
- cycle, 186
- diagramme, 34, 65
  - complet, 189
  - fini, 65
  - vide, 76
- domaine, 58
- doublet, 179
- élémentaire
  - flèche —, 130
  - zigzag —, 190
- endofoncteur, 60
- ensemble
  - $S$ - —, 23
- équation, 28
- équivalence, 165
- extension, 28
  - conservatrice, 135–137, 141, 147
  - libre, 140, 148
- factorisation
  - à droite, 182
- fermeture sémantique, 28, 52
- flèche, 58
  - élémentaire, 130
- foncteur, 59
  - associé à un préfoncteur, 121
  - $\text{colim}_{\mathcal{C}}$ , 100
  - $\text{Colim}_{\mathcal{C}}$ , 79
  - conservant fortement une colimite, 86
  - conservant les colimites, 85
  - conservant une colimite, 84
  - diagr, 98
  - $\text{diagr}(F)$ , 98
  - $\text{DIAGR}$ , 83
  - $\text{DIAGR}(F)$ , 83
  - fidèle, 60
  - $I_{\mathcal{C}}$ , 72
  - plein, 60
- forme
  - minimale, 194
  - normale, 132
- Graph**, 58
- graphe, 55
  - fini, 56
  - sous-jacent, 65
- identité
  - syntaxique, 120
- inclusion
  - de zigzag, 190
- institution, 51, 51–53
- interprétation, 30, 151
- inverse, 58
- irréductible, 132
- isomorphisme, 58, 59, 121
  - de construction, 50
  - entre diagrammes, 173–199
  - naturel, 61, 164
- lien, 193
- LPG, 23–45
- modèle, 27, 30
- modularité, 15
- morphisme
  - d'algèbres, 27
  - de diagrammes, 70
  - de graphes, 56

- de signatures, 25
- de spécifications, 29, 52, 53
  - de base, 53
  - modulaires, 53
- généralisé de graphes, 57
- nœud, 55
  - puits, 193
- objet, 58
  - initial, 76
  - pré-initial, 122, 124, 152
- opérateur, 24
- ordre
  - relation d'— sur les zigzags élémentaires, 190
- précatégorie, 119
  - associée à une catégorie, 120
  - $\text{DIAGR}(\mathcal{C})$ , 123, 151
  - $\text{Iso}(\mathcal{C}_0)$ , 142
  - $\text{TERME}(\mathcal{C}_0)$ , 137
  - petite —, 121
- pré-colimite, 123
- préfoncteur, 121
  - $\mathcal{D}$ , 154
  - fidèle, 122
  - plein, 122
- pré-somme
  - amalgamée, 122, 124, 152
- profil, 24
- Set**, 37, 58
- Sig**, 26, 52, 58
- signature, 24
  - vide, 25
- somme, 67
  - amalgamée, 35–44, 68, 77
  - disjointe, 43
- sorte, 24
- source, 55
- spécification, 13, 52, 53
  - algébrique, 13, 29
  - de base, 53
  - équationnelle, 29
  - modulaire, 33, 53
  - vide, 30, 42
- Spec**, 30, 52, 58
- substitution
  - par objet isomorphe, 174
- suppression
  - d'un arc, 176
  - d'un arc d'un doublet, 179
  - d'une boucle identité, 177
- traduction, 29
- transformation
  - **Ajout\_Arc**, 180
  - Complétion, 187
  - **Contract\_Id**, 183
  - de diagrammes, 173
  - Minimal, 194
  - naturelle, 164
  - stable par isomorphisme, 174
  - **Subst\_Obj\_Iso**, 174
  - **Suppr\_Arc**, 176
- transformation naturelle, 60
  - $[-]$ , 98
  - généralisée, 69, 71
  - $I$ , 83
- unité, 63
- variable, 27
- zigzag, 56
  - élémentaire, 190
    - maximal, 192
  - nul, 56
  - opposé, 56





## Résumé

La composition de spécifications modulaires peut être modélisée, dans le formalisme des catégories, par des colimites de diagrammes. La somme amalgamée permet en particulier d’assembler deux spécifications en précisant les parties communes. Notre travail poursuit cette idée classique selon trois axes.

D’un point de vue syntaxique, nous définissons un langage pour représenter les spécifications modulaires construites à partir d’une catégorie de spécifications et de morphismes de spécifications de base. Ce langage est caractérisé formellement par une catégorie de termes finiment cocomplète.

D’un point de vue sémantique, nous proposons d’associer à tout terme un diagramme. Cette interprétation permet de faire abstraction de certains choix effectués lors de la construction de la spécification modulaire. Pour cela, nous définissons une catégorie de diagrammes “concrète”, c’est-à-dire dont les flèches peuvent être manipulées effectivement. En considérant le quotient par une certaine congruence, nous obtenons une complétion de la catégorie de base par colimites finies. Nous montrons que le calcul du diagramme associé à un terme définit une équivalence entre la catégorie des termes et la catégorie des diagrammes, ce qui prouve la correction de cette interprétation.

Enfin, nous proposons un algorithme pour décider si deux diagrammes sont isomorphes, dans le cas particulier où la catégorie de base est finie et sans cycle. Cela permet de détecter des isomorphismes “de construction” entre spécifications modulaires, c’est-à-dire des isomorphismes qui ne dépendent pas des spécifications de base, mais seulement de la manière dont celles-ci sont assemblées.

## Abstract

The composition of modular specifications can be modeled, in a category theoretic framework, by means of colimits of diagrams. Pushouts in particular allow us to gather two specifications sharing a common part. Our work extends this classic idea along three lines.

From a syntactic point of view, we define a language to represent modular specifications built from a category of base specifications and base specification morphisms. This language is formally characterized by a finitely cocomplete category of terms.

From a semantic point of view, we propose to associate with each term a diagram. This interpretation allows us to abstract some choices made while constructing a modular specification. We thus define a “concrete” category of diagrams, in which arrows can actually be handled. Considering the quotient by a certain congruence relation, we get a completion of the base category with finite colimits. We prove that this calculus defines an equivalence between the category of terms and the category of diagrams, which shows the soundness of this interpretation.

At last, we propose an algorithm to decide whether two diagrams are isomorphic, when the base category is finite and cycle free. This allows us to detect “construction isomorphisms” between modular specifications, i.e. isomorphisms which do not depend on the base specifications, but only on their combination.