Base de la démonstration automatique : Skolémisation

Stéphane Devismes Pascal Lafourcade Michel Lévy

Université Joseph Fourier, Grenoble I

25 novembre 2008

Plan

Plan

Introduction

Le théorème de Herbrand est applicable à la fermeture universelle d'un ensemble de formules sans quantificateur.

Introduction

Le théorème de Herbrand est applicable à la fermeture universelle d'un ensemble de formules sans quantificateur.

Aujourd'hui, nous allons étudier une transformation appelée skolémisation.

Introduction

Le théorème de Herbrand est applicable à la fermeture universelle d'un ensemble de formules sans quantificateur.

Aujourd'hui, nous allons étudier une transformation appelée skolémisation.

- La skolémisation **change un ensemble de formules fermées en formule universelle d'un ensemble de formules sans quantificateur**.

Introduction

Le théorème de Herbrand est applicable à la fermeture universelle d'un ensemble de formules sans quantificateur.

Aujourd'hui, nous allons étudier une transformation appelée skolémisation.

- La skolémisation **change un ensemble de formules fermées en formule universelle d'un ensemble de formules sans quantificateur**.
- La skolémisation **préserve l'existence d'un modèle**.

Plan

Exemples (1/2)

La formule $\exists x P(x)$ est skolémisée en $P(a)$.

On observe les relations suivantes entre ces deux formules :

Exemples (1/2)

La formule $\exists x P(x)$ est skolémisée en $P(a)$.

On observe les relations suivantes entre ces deux formules :

1. $P(a)$ a pour conséquence $\exists x P(x)$

Exemples (1/2)

La formule $\exists x P(x)$ est skolémisée en $P(a)$.

On observe les relations suivantes entre ces deux formules :

1. $P(a)$ a pour conséquence $\exists x P(x)$
2. $\exists x P(x)$ n'a pas pour conséquence $P(a)$ mais un modèle de $\exists x\, P(x)$ « donne » un modèle de $P(a)$.

Exemples (1/2)

La formule $\exists x P(x)$ est skolémisée en $P(a)$.

On observe les relations suivantes entre ces deux formules :

1. $P(a)$ a pour conséquence $\exists x P(x)$
2. $\exists x P(x)$ n'a pas pour conséquence $P(a)$ mais un modèle de $\exists x\, P(x)$ « donne » un modèle de $P(a)$.

   En effet soit I un modèle de $\exists x P(x)$. Donc il existe $d \in P_I$.

   Soit J l'interprétation telle que $P_J = P_I$ et $a_J = d$.

   J est modèle de $P(a)$.

Exemples (2/2)

La formule $\forall x \exists y Q(x, y)$ est skolémisée en $\forall x Q(x, f(x))$.

On observe les relations entre ces deux formules :

Exemples (2/2)

La formule $\forall x \exists y Q(x, y)$ est skolémisée en $\forall x Q(x, f(x))$.

On observe les relations entre ces deux formules :

1. $\forall x Q(x, f(x))$ a pour conséquence $\forall x \exists y Q(x, y)$

Exemples (2/2)

La formule $\forall x \exists y Q(x, y)$ est skolémisée en $\forall x Q(x, f(x))$.

On observe les relations entre ces deux formules :

1. $\forall x Q(x, f(x))$ a pour conséquence $\forall x \exists y Q(x, y)$
2. $\forall x \exists y Q(x, y)$ n'a pas pour conséquence $\forall x Q(x, f(x))$ mais un modèle de $\forall x \exists y Q(x, y)$ « donne » un modèle de $\forall x Q(x, f(x))$.

Exemples (2/2)

La formule $\forall x \exists y Q(x,y)$ est skolémisée en $\forall x Q(x, f(x))$.

On observe les relations entre ces deux formules :

1. $\forall x Q(x, f(x))$ a pour conséquence $\forall x \exists y Q(x,y)$
2. $\forall x \exists y Q(x,y)$ n'a pas pour conséquence $\forall x Q(x, f(x))$ mais un modèle de $\forall x \exists y Q(x,y)$ « donne » un modèle de $\forall x Q(x, f(x))$.

   Soit I un modèle de $\forall x \exists y Q(x,y)$ et soit D le domaine de I.

   Pour tout $d \in D$, l'ensemble $\{e \in D \mid (d,e) \in Q_I\}$ n'est pas vide, donc il existe $g : D \longrightarrow D$ une fonction telle que pour tout $d \in D$, $g(d) \in \{e \in D \mid (d,e) \in Q_I\}$.

   Soit J l'interprétation J telle que $Q_J = Q_I$ et $f_J = g$: J est modèle de $\forall x Q(x, f(x))$.

Propriétés

La skolémisation sert **à éliminer les quantificateurs existentiels** et **change une formule fermée** A **en une formule** B telle que :

- B a pour conséquence A
- tout modèle de A donne un modèle de B

Propriétés

La skolémisation sert **à éliminer les quantificateurs existentiels** et **change une formule fermée** A **en une formule** B telle que :

- B a pour conséquence A
- tout modèle de A donne un modèle de B

D'où, A **a un modèle si et seulement si** B **a un modèle** : la skolémisation préserve l'existence d'un modèle, on dit aussi qu'elle préserve la satisfaisabilité.

Plan

Définitions

Définitions

Definition (Formule propre)

Une formule fermée est dite propre si elle ne comporte pas de variable liée par deux quantificateurs distincts.

Définitions

Definition (Formule propre)

Une formule fermée est dite propre si elle ne comporte pas de variable liée par deux quantificateurs distincts.

Exemples :

- La formule $\forall x P(x) \vee \forall x Q(x)$ n'est pas propre.

Définitions

Definition (Formule propre)

Une formule fermée est dite propre si elle ne comporte pas de variable liée par deux quantificateurs distincts.

Exemples :

- La formule $\forall x P(x) \vee \forall x Q(x)$ n'est pas propre.
- La formule $\forall x P(x) \vee \forall y Q(y)$ est propre.

Définitions

Definition (Formule propre)

Une formule fermée est dite propre si elle ne comporte pas de variable liée par deux quantificateurs distincts.

Exemples :

- La formule $\forall x P(x) \vee \forall x Q(x)$ n'est pas propre.
- La formule $\forall x P(x) \vee \forall y Q(y)$ est propre.
- La formule $\forall x (P(x) \Rightarrow \exists x Q(x) \wedge \exists y R(x,y))$ n'est pas propre.

Définitions

Definition (Formule propre)

Une formule fermée est dite propre si elle ne comporte pas de variable liée par deux quantificateurs distincts.

Exemples :

- La formule $\forall x P(x) \vee \forall x Q(x)$ n'est pas propre.
- La formule $\forall x P(x) \vee \forall y Q(y)$ est propre.
- La formule $\forall x (P(x) \Rightarrow \exists x Q(x) \wedge \exists y R(x,y))$ n'est pas propre.
- La formule $\forall x (P(x) \Rightarrow \exists y R(x,y))$ est propre.

Définitions

Definition (Formule propre)

Une formule fermée est dite propre si elle ne comporte pas de variable liée par deux quantificateurs distincts.

Exemples :

- La formule $\forall x P(x) \vee \forall x Q(x)$ n'est pas propre.
- La formule $\forall x P(x) \vee \forall y Q(y)$ est propre.
- La formule $\forall x (P(x) \Rightarrow \exists x Q(x) \wedge \exists y R(x, y))$ n'est pas propre.
- La formule $\forall x (P(x) \Rightarrow \exists y R(x, y))$ est propre.

Definition (Formule normale)

Une formule normale est une formule **sans équivalence, ni implication, dont les négations portent uniquement sur les formules atomiques**.

Comment skolémiser une formule

Il y a plusieurs façons de skolémiser une formule fermée. Nous proposons ici un compromis entre l'efficacité et la simplicité de la transformation.

Comment skolémiser une formule

Il y a plusieurs façons de skolémiser une formule fermée. Nous proposons ici un compromis entre l'efficacité et la simplicité de la transformation.

1. Transformation de la formule fermée en une autre formule fermée normale équivalente.

Comment skolémiser une formule

Il y a plusieurs façons de skolémiser une formule fermée. Nous proposons ici un compromis entre l'efficacité et la simplicité de la transformation.

1. Transformation de la formule fermée en une autre formule fermée normale équivalente.
2. Transformation de la formule fermée normale en une formule fermée, normale et propre équivalente.

Comment skolémiser une formule

Il y a plusieurs façons de skolémiser une formule fermée. Nous proposons ici un compromis entre l'efficacité et la simplicité de la transformation.

1. Transformation de la formule fermée en une autre formule fermée normale équivalente.
2. Transformation de la formule fermée normale en une formule fermée, normale et propre équivalente.
3. Élimination des quantificateurs existentiels.
   Cette transformation préserve seulement l'existence de modèle.

Comment skolémiser une formule

Il y a plusieurs façons de skolémiser une formule fermée. Nous proposons ici un compromis entre l'efficacité et la simplicité de la transformation.

1. Transformation de la formule fermée en une autre formule fermée normale équivalente.
2. Transformation de la formule fermée normale en une formule fermée, normale et propre équivalente.
3. Élimination des quantificateurs existentiels.
   Cette transformation préserve seulement l'existence de modèle.
4. Transformation de chaque formule fermée, normale, propre et sans quantificateur existentiel en une formule normale sans quantificateur.

Comment skolémiser une formule

Il y a plusieurs façons de skolémiser une formule fermée. Nous proposons ici un compromis entre l'efficacité et la simplicité de la transformation.

1. Transformation de la formule fermée en une autre formule fermée normale équivalente.
2. Transformation de la formule fermée normale en une formule fermée, normale et propre équivalente.
3. Élimination des quantificateurs existentiels.
   Cette transformation préserve seulement l'existence de modèle.
4. Transformation de chaque formule fermée, normale, propre et sans quantificateur existentiel en une formule normale sans quantificateur.

Definition (Forme de Skolem)

Soit A une formule fermée et B la formule normale sans quantificateur obtenue par la transformation ci-dessus : B est la forme de Skolem de A.

Transformation en formule normale

On enlève les équivalences et les implications et on déplace les négations vers les formules atomiques :

$$A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$$

$$A \Rightarrow B \equiv \neg A \vee B$$

$$\neg\neg A \equiv A$$

$$\neg(A \wedge B) \equiv \neg A \vee \neg B$$

$$\neg(A \vee B) \equiv \neg A \wedge \neg B$$

$$\neg \forall x A \equiv \exists x \neg A$$

$$\neg \exists x A \equiv \forall x \neg A$$

Astuce

Comme dans le cas propositionnel, en combinant déplacement de la négation et élimination de l'implication, c'est à dire en remplaçant $\neg(A \Rightarrow B)$ par $A \wedge \neg B$, on peut effectuer plus rapidement la transformation.

Exemple

La formule $\forall y(\forall x P(x,y) \Leftrightarrow Q(y))$ est transformée par élimination de l'équivalence et de l'implication en :

Exemple

La formule $\forall y(\forall x P(x,y) \Leftrightarrow Q(y))$ est transformée par élimination de l'équivalence et de l'implication en :

$$\forall y((\neg \forall x P(x,y) \vee Q(y)) \wedge (\neg Q(y) \vee \forall x P(x,y)))$$

Exemple

La formule $\forall y(\forall x P(x,y) \Leftrightarrow Q(y))$ est transformée par élimination de l'équivalence et de l'implication en :

$$\forall y((\neg\forall x P(x,y) \vee Q(y)) \wedge (\neg Q(y) \vee \forall x P(x,y)))$$

puis par déplacement de la négation en :

Exemple

La formule $\forall y(\forall x P(x,y) \Leftrightarrow Q(y))$ est transformée par élimination de l'équivalence et de l'implication en :

$$\forall y((\neg\forall x P(x,y) \vee Q(y)) \wedge (\neg Q(y) \vee \forall x P(x,y)))$$

puis par déplacement de la négation en :

$$\forall y((\exists x \neg P(x,y) \vee Q(y)) \wedge (\neg Q(y) \vee \forall x P(x,y)))$$

Transformation en formule propre

On change le nom des variables liées correctement, par exemple en choisissant de nouvelles variables à chaque changement de nom.

Transformation en formule propre

On change le nom des variables liées correctement, par exemple en choisissant de nouvelles variables à chaque changement de nom.

Exemples :

- La formule $\forall x P(x) \vee \forall x Q(x)$ est changée en $\forall x P(x) \vee \forall y Q(y)$

Transformation en formule propre

On change le nom des variables liées correctement, par exemple en choisissant de nouvelles variables à chaque changement de nom.

Exemples :

- La formule $\forall x P(x) \vee \forall x Q(x)$ est changée en $\forall x P(x) \vee \forall y Q(y)$
- La formule $\forall x (P(x) \Rightarrow \exists x Q(x) \wedge \exists y R(x, y))$ est changée en $\forall x (P(x) \Rightarrow \exists z Q(z) \wedge \exists y R(x, y))$

Élimination des quantificateurs existentiels

Théorème

[Élimination d'une occurence d'un quantificateur existentiel] Soit A une formule fermée normale et propre ayant une occurrence de la sous-formule $\exists y B$. Soient $x_1, \ldots x_n$ l'ensemble des variables libres de $\exists y B$, où $n \geq 0$. Soit f un symbole *ne figurant pas dans* A. Soit A' la formule obtenue en remplaçant cette occurrence de $\exists y B$ par $B < y := f(x_1, \ldots x_n) >$ (Si $n = 0$, f est une constante).

La formule A' est une formule fermée normale et propre vérifiant :

1. A' a pour conséquence A
2. Si A a un modèle alors A' a un modèle identique à celui de A sauf pour le sens de f.

Démonstration.

La preuve est donnée dans le poly. □

Remarque

D'après le théorème, il faut constater que la formule A' obtenue à partir de la formule A par élimination d'un quantificateur reste fermée, normale et propre.

Donc, en « appliquant » plusieurs fois le théorème, ce qui implique de choisir un *nouveau* symbole à chaque quantificateur éliminé, on peut transformer une formule A fermée, normale et propre en une formule B fermée, normale, propre et *sans quantificateur existentiel* telle que :

- La formule A est conséquence de la formule B
- Si A a un modèle, alors B a un modèle identique sauf pour le sens des nouveaux symboles

Exemple

En éliminant les quantificateurs existentiels de la formule $\exists x \forall y P(x,y) \wedge \exists z \forall u \neg P(z,u)$ on obtient $\forall y P(a,y) \wedge \forall u \neg P(b,u)$. Il est facile de voir que cette formule a un modèle.

Exemple

En éliminant les quantificateurs existentiels de la formule $\exists x \forall y P(x, y) \wedge \exists z \forall u \neg P(z, u)$ on obtient $\forall y P(a, y) \wedge \forall u \neg P(b, u)$. Il est facile de voir que cette formule a un modèle.

Remarque : Si on fait l'erreur d'éliminer les deux quantificateurs existentiels avec la même constante a, on obtient la formule $\forall y P(a, y) \wedge \forall u \neg P(a, u)$, qui est insatisfaisable, puisqu'elle a pour conséquence $P(a, a)$ et $\neg P(a, a)$.

Donc il faut impérativement utiliser un nouveau symbole lors de chaque élimination d'un quantificateur existentiel.

Transformation en formule universelle

Soit A une formule fermée, normale, propre et sans quantificateur existentiel.

On transforme A en une formule B telle que A est équivalente à $\forall(B)$ (la fermeture universelle de B).

B est obtenue en utilisant tous les remplacements possibles des sous formules de la forme

$(\forall x C) \wedge D$ par $\forall x(C \wedge D)$ où x non libre dans D

$(\forall x C) \vee D$ par $\forall x(C \wedge D)$ où x non libre dans D

$D \wedge (\forall x C)$ par $\forall x(D \wedge C)$ où x non libre dans D

$D \vee (\forall x C)$ par $\forall x(D \vee C)$ où x non libre dans D

Transformation en formule universelle

Soit A une formule fermée, normale, propre et sans quantificateur existentiel.

On transforme A en une formule B telle que A est équivalente à $\forall(B)$ (la fermeture universelle de B).

B est obtenue en utilisant tous les remplacements possibles des sous formules de la forme

$(\forall x C) \wedge D$ par $\forall x(C \wedge D)$ où x non libre dans D

$(\forall x C) \vee D$ par $\forall x(C \wedge D)$ où x non libre dans D

$D \wedge (\forall x C)$ par $\forall x(D \wedge C)$ où x non libre dans D

$D \vee (\forall x C)$ par $\forall x(D \vee C)$ où x non libre dans D

Théorème

La formule A est équivalente à $\forall(B)$.

Transformation en formule universelle

Soit A une formule fermée, normale, propre et sans quantificateur existentiel.

On transforme A en une formule B telle que A est équivalente à $\forall(B)$ (la fermeture universelle de B).

B est obtenue en utilisant tous les remplacements possibles des sous formules de la forme

$(\forall x C) \wedge D$ par $\forall x(C \wedge D)$ où x non libre dans D

$(\forall x C) \vee D$ par $\forall x(C \wedge D)$ où x non libre dans D

$D \wedge (\forall x C)$ par $\forall x(D \wedge C)$ où x non libre dans D

$D \vee (\forall x C)$ par $\forall x(D \vee C)$ où x non libre dans D

Théorème

La formule A est équivalente à $\forall(B)$.

Remarque : B est la forme de Skolem de A

Propriété de la skolémisation

Par construction, on obtient la propriété suivante :

Théorème

[Propriété de la skolémisation] Soit A une formule fermée et B la forme de Skolem de A.

- La formule $\forall(B)$ a pour conséquence la formule A
- si A a un modèle alors $\forall(B)$ a un modèle

Donc A a un modèle si et seulement si $\forall(B)$ a un modèle.

Démonstration.

Preuve donnée dans le poly. $\square$

Exemple

Soit $A = \forall x(P(x) \Rightarrow Q(x)) \Rightarrow (\forall x P(x) \Rightarrow \forall x Q(x))$. On skolémise $\neg A$.

Exemple

Soit $A = \forall x(P(x) \Rightarrow Q(x)) \Rightarrow (\forall x P(x) \Rightarrow \forall x Q(x))$. On skolémise $\neg A$.

1. $\neg A$ est transformée en la formule normale :
   $\forall x(\neg P(x) \vee Q(x)) \wedge \forall x P(x) \wedge \exists x \neg Q(x)$

Exemple

Soit $A = \forall x(P(x) \Rightarrow Q(x)) \Rightarrow (\forall x P(x) \Rightarrow \forall x Q(x))$. On skolémise $\neg A$.

1. $\neg A$ est transformée en la formule normale :
   $\forall x(\neg P(x) \vee Q(x)) \wedge \forall x P(x) \wedge \exists x \neg Q(x)$
2. La formule normale est transformée en la formule propre :
   $\forall x(\neg P(x) \vee Q(x)) \wedge \forall y P(y) \wedge \exists z \neg Q(z)$

Exemple

Soit $A = \forall x(P(x) \Rightarrow Q(x)) \Rightarrow (\forall x P(x) \Rightarrow \forall x Q(x))$. On skolémise $\neg A$.

1. $\neg A$ est transformée en la formule normale :
   $\forall x(\neg P(x) \vee Q(x)) \wedge \forall x P(x) \wedge \exists x \neg Q(x)$
2. La formule normale est transformée en la formule propre :
   $\forall x(\neg P(x) \vee Q(x)) \wedge \forall y P(y) \wedge \exists z \neg Q(z)$
3. Le quantificateur existentiel est »remplacé« par une constante :
   $\forall x(\neg P(x) \vee Q(x)) \wedge \forall y P(y) \wedge \neg Q(a)$

Exemple

Soit $A = \forall x(P(x) \Rightarrow Q(x)) \Rightarrow (\forall xP(x) \Rightarrow \forall xQ(x))$. On skolémise $\neg A$.

1. $\neg A$ est transformée en la formule normale :
$\forall x(\neg P(x) \vee Q(x)) \wedge \forall xP(x) \wedge \exists x \neg Q(x)$
2. La formule normale est transformée en la formule propre :
$\forall x(\neg P(x) \vee Q(x)) \wedge \forall yP(y) \wedge \exists z \neg Q(z)$
3. Le quantificateur existentiel est »remplacé« par une constante :
$\forall x(\neg P(x) \vee Q(x)) \wedge \forall yP(y) \wedge \neg Q(a)$
4. Les quantificateurs universels sont enlevés :
$(\neg P(x) \vee Q(x)) \wedge P(y) \wedge \neg Q(a))$.

Exemple

Soit $A = \forall x(P(x) \Rightarrow Q(x)) \Rightarrow (\forall x P(x) \Rightarrow \forall x Q(x))$. On skolémise $\neg A$.

1. $\neg A$ est transformée en la formule normale :
   $\forall x(\neg P(x) \vee Q(x)) \wedge \forall x P(x) \wedge \exists x \neg Q(x)$
2. La formule normale est transformée en la formule propre :
   $\forall x(\neg P(x) \vee Q(x)) \wedge \forall y P(y) \wedge \exists z \neg Q(z)$
3. Le quantificateur existentiel est »remplacé« par une constante :
   $\forall x(\neg P(x) \vee Q(x)) \wedge \forall y P(y) \wedge \neg Q(a)$
4. Les quantificateurs universels sont enlevés :
   $(\neg P(x) \vee Q(x)) \wedge P(y) \wedge \neg Q(a))$.

Instancions la forme de Skolem de $\neg A$ en remplaçant x et y par a. On obtient la formule $(\neg P(a) \vee Q(a)) \wedge P(a) \wedge \neg Q(a)$ qui est insatisfaisable. Donc $\forall((\neg P(x) \vee Q(x)) \wedge P(y) \wedge \neg Q(a))$ est insatisfaisable. Puisque, la skolémisation préserve l'existence d'un modèle, $\neg A$ est insatisfaisable, donc A est valide.

Méthode

Théorème

Soit Γ un ensemble de formules fermées. La skolémisation de Γ consiste à appliquer la skolémisation à chaque formule de Γ, en choisissant un nouveau symbole pour chaque quantificateur existentiel éliminé à la troisième étape de la skolémisation.

On obtient ainsi un ensemble Δ de formules sans quantificateurs tel que :

- Tout modèle de $\forall(\Delta)$ est modèle de Γ
- Si Γ a un modèle alors $\forall(\Delta)$ en a un modèle qui ne différe de celui de Γ que par le sens des nouveaux symboles.

Plan

Idée

Nous définissons ultérieurement une généralisation au premier ordre de la méthode de résolution.

Aussi après la skolémisation des formules il faut les transformer en clauses.

Littéral, clause

Definition

Un littéral positif est une formule atomique.

Un littéral négatif est la négation d'une formule atomique.

Tout littéral est positif ou négatif.

Une clause est une somme de littéraux

Forme clausale d'une formule

Definition

Soit A une formule fermée. La forme clausale de A est un ensemble de clauses obtenue en deux étapes :

1. skolémiser A, autrement dire construire sa forme de Skolem B
2. remplacer B par un ensemble Γ équivalent de clauses obtenu par distributivité de la somme sur le produit

Propriété de la forme clausale d'une formule

Théorème

La fermeture de la forme clausale d'une formule fermée a un modèle si et seulement la formule a un modèle. Plus précisément

- la fermeture de la forme clausale d'une formule a pour conséquence la formule
- si la formule a un modèle alors la fermeture de sa forme clausale en a un

Démonstration.

La skolémisation préserve la satisfiabilité. Ensuite, comme Γ est obtenu par distributivité, B et Γ sont équivalents donc $\forall(B)$ et $\forall(\Gamma)$ sont aussi équivalents. $\square$

Forme clausale d'un ensemble de formules

Definition

Soit Γ un ensemble de formules fermées, la forme clausale de Γ est obtenu en faisant l'union des formes clausales de chacune des formules de Γ et en prenant soin au cours de la skolémisation d'éliminer chaque occurrence d'un quantificateur existentiel à l'aide d'un nouveau symbole.

Propriété de la forme clausale d'un ensemble de formules

Théorème

Soit Γ un ensemble de formules fermées et Δ la forme clausale de Γ. On a :

- $\forall(\Delta)$ a pour conséquence Γ
- Si Γ a un modèle alors $\forall(\Delta)$ a un modèle

Adaptation du théorème de Herbrand aux formes clausales

Théorème

Soit Γ un ensemble de formules fermées et Δ la forme clausale de Γ. Γ est insatisfaisable si et seulement s'il existe un sous-ensemble fini insatisfaisable d'instances des clauses de Δ sur la signature de Δ

Démonstration.

La skolémisation préserve la satisfaisablité, donc :
Γ est insatisfaisable si et seulement si $\forall(\Delta)$ est insatisfaisable.
D'après le corollaire du théorème de Herbrand, $\forall(\Delta)$ est insatisfaisable si et seulement si il existe un sous-ensemble fini insatisfaisable d'instances des clauses de Δ sur la signature de Δ. $\square$

Exemple (1/2)

Soit $A = \exists y \forall z (P(z,y) \Leftrightarrow \neg \exists x (P(z,x) \wedge P(x,z)))$. Calculons la forme clausale de A.

Exemple (1/2)

Soit $A = \exists y \forall z (P(z,y) \Leftrightarrow \neg \exists x (P(z,x) \wedge P(x,z)))$. Calculons la forme clausale de A.

1. Mettons A sous forme normale :
   $\exists y \forall z ((\neg P(z,y) \vee \forall x (\neg P(z,x) \vee \neg P(x,z))) \wedge \exists x (P(z,x) \wedge P(x,z)) \vee P(z,y))$

Exemple (1/2)

Soit $A = \exists y \forall z (P(z,y) \Leftrightarrow \neg \exists x (P(z,x) \wedge P(x,z)))$. Calculons la forme clausale de A.

1. Mettons A sous forme normale :
   $\exists y \forall z ((\neg P(z,y) \vee \forall x (\neg P(z,x) \vee \neg P(x,z))) \wedge \exists x (P(z,x) \wedge P(x,z)) \vee P(z,y))$
2. Rendons propre le résultat :
   $\exists y \forall z ((\neg P(z,y) \vee \forall x (\neg P(z,x) \vee \neg P(x,z))) \wedge \exists u (P(z,u) \wedge P(u,z)) \vee P(z,y))$

Exemple (1/2)

Soit $A = \exists y \forall z (P(z,y) \Leftrightarrow \neg \exists x (P(z,x) \wedge P(x,z)))$. Calculons la forme clausale de A.

1. Mettons A sous forme normale :
$\exists y \forall z ((\neg P(z,y) \vee \forall x (\neg P(z,x) \vee \neg P(x,z))) \wedge \exists x (P(z,x) \wedge P(x,z)) \vee P(z,y))$
2. Rendons propre le résultat :
$\exists y \forall z ((\neg P(z,y) \vee \forall x (\neg P(z,x) \vee \neg P(x,z))) \wedge \exists u (P(z,u) \wedge P(u,z)) \vee P(z,y))$
3. Éliminons les quantificateurs existentiels :
$\forall z ((\neg P(z,a) \vee \forall x (\neg P(z,x) \vee \neg P(x,z))) \wedge (P(z,f(z)) \wedge P(f(z),z)) \vee P(z,a))$

Exemple (1/2)

Soit $A = \exists y \forall z (P(z,y) \Leftrightarrow \neg \exists x (P(z,x) \wedge P(x,z)))$. Calculons la forme clausale de A.

1. Mettons A sous forme normale :
$\exists y \forall z ((\neg P(z,y) \vee \forall x (\neg P(z,x) \vee \neg P(x,z))) \wedge \exists x (P(z,x) \wedge P(x,z)) \vee P(z,y))$
2. Rendons propre le résultat :
$\exists y \forall z ((\neg P(z,y) \vee \forall x (\neg P(z,x) \vee \neg P(x,z))) \wedge \exists u (P(z,u) \wedge P(u,z)) \vee P(z,y))$
3. Éliminons les quantificateurs existentiels :
$\forall z ((\neg P(z,a) \vee \forall x (\neg P(z,x) \vee \neg P(x,z))) \wedge (P(z,f(z)) \wedge P(f(z),z)) \vee P(z,a))$
4. Supprimons les quantificateurs universels, on obtient la forme de Skolem de A :
$((\neg P(z,a) \vee (\neg P(z,x) \vee \neg P(x,z))) \wedge (P(z,f(z)) \wedge P(f(z),z)) \vee P(z,a))$

Exemple (1/2)

Soit $A = \exists y \forall z (P(z,y) \Leftrightarrow \neg \exists x (P(z,x) \wedge P(x,z)))$. Calculons la forme clausale de A.

1. Mettons A sous forme normale :
   $\exists y \forall z ((\neg P(z,y) \vee \forall x (\neg P(z,x) \vee \neg P(x,z))) \wedge \exists x (P(z,x) \wedge P(x,z)) \vee P(z,y))$
2. Rendons propre le résultat :
   $\exists y \forall z ((\neg P(z,y) \vee \forall x (\neg P(z,x) \vee \neg P(x,z))) \wedge \exists u (P(z,u) \wedge P(u,z)) \vee P(z,y))$
3. Éliminons les quantificateurs existentiels :
   $\forall z ((\neg P(z,a) \vee \forall x (\neg P(z,x) \vee \neg P(x,z))) \wedge (P(z,f(z)) \wedge P(f(z),z)) \vee P(z,a))$
4. Supprimons les quantificateurs universels, on obtient la forme de Skolem de A :
   $((\neg P(z,a) \vee (\neg P(z,x) \vee \neg P(x,z))) \wedge (P(z,f(z)) \wedge P(f(z),z)) \vee P(z,a))$
5. Transformons en produit de sommes de littéraux, on obtient la forme clausale de A, qui est l'ensemble suivant de clauses :
   - $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
   - $C_2 = P(z,f(z)) \vee P(z,a)$
   - $C_3 = P(f(z),z) \vee P(z,a)$

Exemple (2/2)

- $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
- $C_2 = P(z,f(z)) \vee P(z,a)$
- $C_3 = P(f(z),z) \vee P(z,a)$

A n'a pas de modèle si et seulement si il y a un ensemble fini insatisfaisable d'instances de C_1, C_2, C_3 sur la signature de ces clauses.

On recherche ces instances :

Exemple (2/2)

- $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
- $C_2 = P(z,f(z)) \vee P(z,a)$
- $C_3 = P(f(z),z) \vee P(z,a)$

A n'a pas de modèle si et seulement si il y a un ensemble fini insatisfaisable d'instances de C_1, C_2, C_3 sur la signature de ces clauses.

On recherche ces instances :

- Soit C_1' obtenue avec $x = a, z = a$ dans C_1 : $C_1' = \neg P(a,a)$

Exemple (2/2)

- $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
- $C_2 = P(z,f(z)) \vee P(z,a)$
- $C_3 = P(f(z),z) \vee P(z,a)$

A n'a pas de modèle si et seulement si il y a un ensemble fini insatisfaisable d'instances de C_1, C_2, C_3 sur la signature de ces clauses.

On recherche ces instances :

- Soit C_1' obtenue avec $x = a, z = a$ dans C_1 : $C_1' = \neg P(a,a)$
- Soit C_1'' obtenue avec $x = a, z = f(a)$ dans C_1 :
  $C_1'' = \neg P(f(a),a) \vee \neg P(a,f(a))$

Exemple (2/2)

- $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
- $C_2 = P(z,f(z)) \vee P(z,a)$
- $C_3 = P(f(z),z) \vee P(z,a)$

A n'a pas de modèle si et seulement si il y a un ensemble fini insatisfaisable d'instances de C_1, C_2, C_3 sur la signature de ces clauses.

On recherche ces instances :

- Soit C_1' obtenue avec $x = a, z = a$ dans C_1 : $C_1' = \neg P(a,a)$
- Soit C_1'' obtenue avec $x = a, z = f(a)$ dans C_1 :
  $C_1'' = \neg P(f(a),a) \vee \neg P(a,f(a))$
- Soit C_2' obtenue avec $z = a$ dans C_2 : $C_2' = P(a,f(a)) \vee P(a,a)$

Exemple (2/2)

- $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
- $C_2 = P(z, f(z)) \vee P(z,a)$
- $C_3 = P(f(z), z) \vee P(z,a)$

A n'a pas de modèle si et seulement si il y a un ensemble fini insatisfaisable d'instances de C_1, C_2, C_3 sur la signature de ces clauses.

On recherche ces instances :

- Soit C_1' obtenue avec $x = a, z = a$ dans C_1 : $C_1' = \neg P(a,a)$
- Soit C_1'' obtenue avec $x = a, z = f(a)$ dans C_1 :
  $C_1'' = \neg P(f(a), a) \vee \neg P(a, f(a))$
- Soit C_2' obtenue avec $z = a$ dans C_2 : $C_2' = P(a, f(a)) \vee P(a,a)$
- Soit C_3' obtenue avec $z = a$ dans C_3 : $C_3' = P(f(a), a) \vee P(a,a)$

Exemple (2/2)

- $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
- $C_2 = P(z,f(z)) \vee P(z,a)$
- $C_3 = P(f(z),z) \vee P(z,a)$

A n'a pas de modèle si et seulement si il y a un ensemble fini insatisfaisable d'instances de C_1, C_2, C_3 sur la signature de ces clauses.

On recherche ces instances :

- Soit C_1' obtenue avec $x = a, z = a$ dans C_1 : $C_1' = \neg P(a,a)$
- Soit C_1'' obtenue avec $x = a, z = f(a)$ dans C_1 :
  $C_1'' = \neg P(f(a),a) \vee \neg P(a,f(a))$
- Soit C_2' obtenue avec $z = a$ dans C_2 : $C_2' = P(a,f(a)) \vee P(a,a)$
- Soit C_3' obtenue avec $z = a$ dans C_3 : $C_3' = P(f(a),a) \vee P(a,a)$

L'ensemble de ces instances est insatisfaisable, donc A est insatisfaisable !

Plan

Idée

Soit Γ un ensemble de clauses. Supposons que $\forall(\Gamma)$ n'a pas de modèle. Le système formel « factorisation, copie, résolution binaire » est un système formel permettant de déduire $\perp$ de Γ avec trois règles :

1. La factorisation qui de la prémisse $P(x, f(y)) \vee P(g(z), z) \vee Q(z, x)$ permet de déduire $P(g(f(y)), f(y)) \vee Q(f(y), g(f(y)))$. La clause déduite est obtenue en calculant la solution la plus générale de $P(x, f(y)) = P(g(z), z)$
2. La règle de copie qui permet de renommer les variables d'une clause.
3. La résolution binaire qui des 2 prémisses sans variable commune $P(x, a) \vee Q(x)$ et $\neg P(b, y) \vee R(f(y))$ permet de déduire le résolvant $Q(b) \vee R(f(a))$, en calculant la solution plus générale $x = b, y = a$ de $P(x, a) = P(b, y)$

La complétude de ce système formel est basée sur le théorème de Herbrand.

Pour trouver les instances contradictoires des clauses, les règles utilisent l'algorithme d'unification.

Unification : expression, solution

Definition

Appelons expression, un terme ou un littéral. Une substitution σ est solution de l'équation $e_1 = e_2$ entre deux expressions, si les deux expressions $e_1\sigma$ et $e_2\sigma$ sont *identiques*. Une substitution est solution d'un ensemble d'équations si elle est solution de chaque équation de l'ensemble.

Unification : support de substitution

Definition

Le support d'une substitution σ est l'ensemble des variables x telles que $x\sigma \neq x$. Dans la suite, on s'intéresse seulement aux substitutions à support fini, c'est à dire qui ne changent qu'un nombre fini de variables.

Une substitution σ à support fini est notée $< x_1 := t_1, \ldots, x_n := t_n >$ ou plus simplement $x_1 := t_1, \ldots, x_n := t_n$ quand il n'y a pas de risque d'ambiguïté.

Les variables $x_1, \ldots, x_n$ sont distinctes et la substitution vérifie :

- pour i de 1 à n, $x_i\sigma = t_i$
- pour toute variable y telle que $y \notin \{x_1, \ldots, x_n\}$, on a : $y\sigma = y$

Unification : exemple de solution

Exemple : La substitution $x := g(f(y)), z := f(y)$ est une solution de l'équation $P(x, f(y)) = P(g(z), z)$.

Elle est aussi solution du système d'équations : $x = g(z), f(y) = z$.

Unification : composition de substitution

Definition

Soient σ et τ 2 substitutions, on note $\sigma\tau$ la substitution telle que pour toute variable x, $x\sigma\tau = (x\sigma)\tau$.

La substitution $\sigma\tau$ est une instance de σ. Deux substitutions sont équivalentes si chacune d'elles est une instance de l'autre.

Unification : exemple

Considérons les substitutions

- $\sigma_1 =< x := g(z), y := z >$
- $\sigma_2 =< x := g(y), z := y >$
- $\sigma_3 =< x := g(a), y := a, z := a >$

On a les relations suivantes entre ces susbstitutions

- $\sigma_1 = \sigma_2 < y := z >$
- $\sigma_2 = \sigma_1 < z := y >$
- $\sigma_3 = \sigma_1 < z := a >$
- $\sigma_3 = \sigma_2 < y := a >$

Les substitutions σ_1 et σ_2 sont équivalentes.
La substitution σ_3 est une instance de σ_1 ainsi que de σ_2, mais ne leur est pas équivalente.

Unification : définition de la solution la plus générale

Definition

Une solution d'un système d'équations est appelée la plus générale si toute autre solution en est une instance. Notons que deux solutions « les plus générales » sont équivalentes.

Unification : définition de la solution la plus générale

Definition

Une solution d'un système d'équations est appelée la plus générale si toute autre solution en est une instance. Notons que deux solutions « les plus générales » sont équivalentes.

Exemple : $\sigma_1 =< x := g(z), y := z >$, $\sigma_2 =< x := g(y), z := y >$, $\sigma_3 =< x := g(a), y := a, z := a >$ sont 3 solutions de l'équation $f(x, g(z)) = f(g(y), x)$.

σ_1 et σ_2 en sont les solutions les plus générales.

Unificateur

Definition

Soit σ une substitution et E un ensemble d'expressions.
$E\sigma = \{t\sigma \mid t \in E\}$.
La substitution σ est un unificateur de E si et seulement si l'ensemble $E\sigma$ n'a qu'un élément.

Soit $\{e_i | 1 \leq i \leq n\}$ un ensemble fini d'expressions. La substitution σ est un unificateur de cet ensemble si et seulement si elle est solution du système d'équations $\{e_i = e_{i+1} | 1 \leq i < n\}$.

Unification : l'algorithme (plan)

L'algorithme calculant la solution la plus générale d'un système d'équations est appelé algorithme d'unification car la recherche d'un unificateur le plus général d'un ensemble d'expression se réduit à la recherche de la solution la plus générale d'un système d'équations.

L'algorithme sépare les équations en équations à résoudre, notées par une égalité et équations résolues, notées par le signe $:=$.

Initialement, il n'y a pas d'équations résolues.

L'algorithme applique les règles énoncées ci-après. Il s'arrête quand il n'y a plus d'équations à résoudre ou quand il a déclaré que le système à résoudre n'a pas de solution.

Quand il s'arrête sans avoir déclaré l'absence de solution, la liste des équations résolues est la solution la plus générale du système initial d'équations.

Unification : l'algorithme (les règles)

- **Supprimer.** Si les 2 membres d'une équation sont identiques, il supprime l'équation
- **Décomposer.** Si les 2 membres d'une équation sont distincts, l'équation ayant la forme
  - $\neg A = \neg B$, il la remplace par $A = B$.
  - $f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)$, il la remplace par les équations $s_1 = t_1, \ldots, s_n = t_n$.
    Pour $n = 0$ cette décomposition supprime l'équation.
- **Echec de la décomposition** Si une équation à résoudre est de la forme $f(s_1, \ldots, s_n) = g(t_1, \ldots, t_p)$ avec $f \neq g$ alors l'algorithme déclare qu'il n'y a pas de solution.

  En particulier il y a évidemment un échec, si l'on cherche à résoudre une équation entre un littéral positif et un littéral négatif.

Unification : l'algorithme (les règles)

- **Orienter.** Si une équation est de la forme $t = x$ où t est un terme qui n'est pas une variable et x une variable, alors on remplace l'équation par $x = t$
- **Élimination d'une variable.** Si une équation à résoudre est de la forme $x = t$ où x est une variable et t un terme *ne contenant pas* x
  1. il l'enlève des équations à résoudre
  2. il remplace x par t dans toutes les équations (non résolues *et résolues*)
  3. il ajoute $x := t$ à la partie résolue
- **Echec de l'élimination.** Si une équation à résoudre est de la forme $x = t$ où x est une variable et t un terme distinct de x et *contenant* x alors l'algorithme déclare qu'il n'y a pas de solution.

Unification : l'algorithme (exemples)

1. Résoudre $f(x, g(z)) = f(g(y), x)$.
   Par décomposition, on obtient : $x = g(y), g(z) = x$
   Par élimination de x, on obtient : $x := g(y), g(z) = g(y)$
   Par décomposition, on obtient : $x := g(y), z = y$
   Par élimination de z, on obtient la solution : $x := g(y), z := y$

Unification : l'algorithme (exemples)

1. Résoudre $f(x, g(z)) = f(g(y), x)$.
   Par décomposition, on obtient : $x = g(y), g(z) = x$
   Par élimination de x, on obtient : $x := g(y), g(z) = g(y)$
   Par décomposition, on obtient : $x := g(y), z = y$
   Par élimination de z, on obtient la solution : $x := g(y), z := y$
2. Résoudre $f(x, x, a) = f(g(y), g(a), y)$.
   Par décomposition, on obtient : $x = g(y), x = g(a), a = y$
   Par élimination de x, grâce à la première équation, on obtient :
   $x := g(y), g(y) = g(a), a = y$
   Par décomposition, on obtient : $x := g(y), y = a, a = y$
   Par élimination de y, on obtient : $x := g(a), y := a, a = a$
   Par suppression de l'identité, on obtient : $x := g(a), y := a$

Unification : l'algorithme (exemples)

1. Résoudre $f(x,x,x) = f(g(y),g(a),y)$.
Par décomposition, on obtient : $x = g(y), x = g(a), x = y$
Par élimination de x, on obtient :
$x := g(y), g(y) = g(a), g(y) = y$
Par orientation des équations, on obtient :
$x := g(y), g(y) = g(a), y = g(y)$
L'équation $y = g(y)$ engendre un échec. Donc l'équation $f(x,x,x) = f(g(y),g(a),y)$ n'a pas de solution.

Résolution : Règles

Nous allons maintenant définir 5 règles de résolutions.

Résolution : Règles

Nous allons maintenant définir 5 règles de résolutions.

Pour simplifier la description de ces règles, nous identifierons, dans cette description, une clause, qui est une somme de littéraux, avec l'ensemble de ses littéraux.

Résolution : Règles (factorisation)

Définition La clause C' est un facteur de la clause C si $C' = C$ ou s'il existe un sous-ensemble E de C tel que E a au moins deux éléments, E est unifiable et $C' = C\sigma$ où σ est l'unificateur le plus général de E.

Résolution : Règles (factorisation)

Définition La clause C' est un facteur de la clause C si $C' = C$ ou s'il existe un sous-ensemble E de C tel que E a au moins deux éléments, E est unifiable et $C' = C\sigma$ où σ est l'unificateur le plus général de E.

Exemple La clause $\underline{P(x)} \vee Q(g(x,y)) \vee \underline{P(f(a))}$ a deux facteurs, elle-même et le facteur $P(f(a)) \vee Q(g(f(a),y))$ obtenu en appliquant à la clause, l'unificateur le plus général $x := f(a)$ des deux littéraux soulignés.

Résolution : Règles (factorisation)

Définition La clause C' est un facteur de la clause C si $C' = C$ ou s'il existe un sous-ensemble E de C tel que E a au moins deux éléments, E est unifiable et $C' = C\sigma$ où σ est l'unificateur le plus général de E.

Exemple La clause $\underline{P(x)} \vee Q(g(x,y)) \vee \underline{P(f(a))}$ a deux facteurs, elle-même et le facteur $P(f(a)) \vee Q(g(f(a),y))$ obtenu en appliquant à la clause, l'unificateur le plus général $x := f(a)$ des deux littéraux soulignés.

Cohérence de la règle Soit C' un facteur de C. Puisque C' est une instance de C, la fermeture universelle de C' est une conséquence de celle de C

Résolution : Règles (copie d'une clause)

Définition Soit C une clause et σ une substitution dont la *restriction* aux variables de C est une *bijection* entre ces variables et celles de la clause $C\sigma$. Les clauses C et $C\sigma$ sont des copies l'une de l'autre. La substitution σ est aussi appelée un renommage de C.
Soit σ un renommage de C et soit σ_C^{-1} la substitution ainsi définie : si x est une variable de $C\sigma$, $x\sigma_C^{-1} = y$, où $y\sigma = x$ sinon $x\sigma_C^{-1} = x$. On montre que $C\sigma\sigma_C^{-1} = C$.

Résolution : Règles (copie d'une clause)

Définition Soit C une clause et σ une substitution dont la *restriction* aux variables de C est une *bijection* entre ces variables et celles de la clause $C\sigma$. Les clauses C et $C\sigma$ sont des copies l'une de l'autre. La substitution σ est aussi appelée un renommage de C.
Soit σ un renommage de C et soit σ_C^{-1} la substitution ainsi définie : si x est une variable de $C\sigma$, $x\sigma_C^{-1} = y$, où $y\sigma = x$ sinon $x\sigma_C^{-1} = x$. On montre que $C\sigma\sigma_C^{-1} = C$.

Exemple Soit $\sigma = < x := u, y := v >$. σ est un renommage de $P(x, y)$. Le renommage inverse, *relativement* à $P(x, y)$ est $< u := x, v := y >$

Résolution : Règles (copie d'une clause)

Définition Soit C une clause et σ une substitution dont la *restriction* aux variables de C est une *bijection* entre ces variables et celles de la clause $C\sigma$. Les clauses C et $C\sigma$ sont des copies l'une de l'autre. La substitution σ est aussi appelée un renommage de C.
Soit σ un renommage de C et soit σ_C^{-1} la substitution ainsi définie : si x est une variable de $C\sigma$, $x\sigma_C^{-1} = y$, où $y\sigma = x$ sinon $x\sigma_C^{-1} = x$. On montre que $C\sigma\sigma_C^{-1} = C$.

Exemple Soit $\sigma = < x := u, y := v >$. σ est un renommage de $P(x, y)$. Le renommage inverse, *relativement* à $P(x, y)$ est $< u := x, v := y >$

Propriété Soient 2 clauses copies l'une de l'autre. Puisque chacune des clauses est instance de l'autre, leurs fermetures universelles sont équivalentes.

Résolution : Règles (résolvant binaire)

Définition Soient C et D deux clauses *n'ayant pas de variable commune.*

La clause E est un résolvant binaire de C et D s'il y a un littéral $L \in C$ et un littéral $M \in D$ tels que L et M^c (le littéral opposé à M) sont unifiables et si $E = ((C - \{L\}) \cup (D - \{M\}))\sigma$ où σ est la solution principale de l'équation $L = M^c$.

Résolution : Règles (résolvant binaire)

Définition Soient C et D deux clauses *n'ayant pas de variable commune.*
La clause E est un résolvant binaire de C et D s'il y a un littéral $L \in C$ et un littéral $M \in D$ tels que L et M^c (le littéral opposé à M) sont unifiables et si $E = ((C - \{L\}) \cup (D - \{M\}))\sigma$ où σ est la solution principale de l'équation $L = M^c$.

Exemple Soit $C = P(x,y) \vee P(y,k(z))$ et $D = \neg P(a,f(a,y_1))$.
$< x := a, y := f(a,y_1) >$ est la solution la plus générale de $P(x,y) = P(a,f(a,y_1))$, donc $P(f(a,y_1),k(z))$ est un résolvant binaire des clauses C et D.

Résolution : Règles (résolvant binaire)

Définition Soient C et D deux clauses *n'ayant pas de variable commune.*
La clause E est un résolvant binaire de C et D s'il y a un littéral $L \in C$ et un littéral $M \in D$ tels que L et M^c (le littéral opposé à M) sont unifiables et si $E = ((C - \{L\}) \cup (D - \{M\}))\sigma$ où σ est la solution principale de l'équation $L = M^c$.

Exemple Soit $C = P(x,y) \vee P(y,k(z))$ et $D = \neg P(a,f(a,y_1))$.
$< x := a, y := f(a,y_1) >$ est la solution la plus générale de $P(x,y) = P(a,f(a,y_1))$, donc $P(f(a,y_1),k(z))$ est un résolvant binaire des clauses C et D.

Cohérence de la règle Soit E un résolvant binaire des clauses C et D : $\forall(C), \forall(D) \models \forall(E)$.

Résolution : Règles (preuve)

Définition Soit Γ un ensemble de clauses et C une clause. Une preuve de C à partir de Γ est une suite de clauses se terminant par C, toute clause de la preuve étant un élément de Γ, un facteur d'une clause la précédant dans la preuve, une copie d'une clause la précédant dans la preuve ou un résolvant binaire de 2 clauses la précédant dans la preuve.

C est déduite de Γ au premier ordre par les 3 règles factorisation, copie et résolution binaire (ce qui est noté $\Gamma \vdash_{1fcb} C$) s'il y a une preuve de C à partir de Γ. Quand il n'y a pas d'ambiguïté sur le système formel utilisé, on remplace $\vdash_{1fcb}$ par $\vdash$.

Résolution : Règles (preuve)

Définition Soit Γ un ensemble de clauses et C une clause. Une preuve de C à partir de Γ est une suite de clauses se terminant par C, toute clause de la preuve étant un élément de Γ, un facteur d'une clause la précédant dans la preuve, une copie d'une clause la précédant dans la preuve ou un résolvant binaire de 2 clauses la précédant dans la preuve.

C est déduite de Γ au premier ordre par les 3 règles factorisation, copie et résolution binaire (ce qui est noté $\Gamma \vdash_{1fcb} C$) s'il y a une preuve de C à partir de Γ. Quand il n'y a pas d'ambiguïté sur le système formel utilisé, on remplace $\vdash_{1fcb}$ par $\vdash$.

Cohérence Soit Γ un ensemble de clauses et C une clause. Notons $\forall(\Gamma)$ l'ensemble des fermetures universelles des formules de Γ.

Si $\Gamma \vdash_{1fcb} C$ alors $\forall(\Gamma) \models \forall(C)$

Cette propriété est une conséquence immédiate de la cohérence de la factorisation, de la copie et de la résolution binaire.

Résolution : Règles (exemples)

Soient les deux clauses

1. $C_1 = P(x, y) \vee P(y, x)$
2. $C_2 = \neg P(u, z) \vee \neg P(z, u)$

Montrons par résolution que $\forall(C_1, C_2)$ n'a pas de modèle.

Résolution : Règles (exemples)

Soient les deux clauses

1. $C_1 = P(x,y) \vee P(y,x)$
2. $C_2 = \neg P(u,z) \vee \neg P(z,u)$

Montrons par résolution que $\forall(C_1, C_2)$ n'a pas de modèle.

1. $P(x,y) \vee P(y,x)$ Hyp C_1

Résolution : Règles (exemples)

Soient les deux clauses

1. $C_1 = P(x, y) \vee P(y, x)$
2. $C_2 = \neg P(u, z) \vee \neg P(z, u)$

Montrons par résolution que $\forall(C_1, C_2)$ n'a pas de modèle.

1. $P(x, y) \vee P(y, x)$ Hyp C_1
2. $P(y, y)$ Facteur de 1 par $< x := y >$

Résolution : Règles (exemples)

Soient les deux clauses

1. $C_1 = P(x,y) \vee P(y,x)$
2. $C_2 = \neg P(u,z) \vee \neg P(z,u)$

Montrons par résolution que $\forall(C_1, C_2)$ n'a pas de modèle.

1. $P(x,y) \vee P(y,x)$ Hyp C_1
2. $P(y,y)$ Facteur de 1 par $< x := y >$
3. $\neg P(u,z) \vee \neg P(z,u)$ Hyp C_2

Résolution : Règles (exemples)

Soient les deux clauses

1. $C_1 = P(x,y) \vee P(y,x)$
2. $C_2 = \neg P(u,z) \vee \neg P(z,u)$

Montrons par résolution que $\forall(C_1, C_2)$ n'a pas de modèle.

1. $P(x,y) \vee P(y,x)$ Hyp C_1
2. $P(y,y)$ Facteur de 1 par $< x := y >$
3. $\neg P(u,z) \vee \neg P(z,u)$ Hyp C_2
4. $\neg P(z,z)$ Facteur de 3 par $< u := z >$

Résolution : Règles (exemples)

Soient les deux clauses

1. $C_1 = P(x, y) \vee P(y, x)$
2. $C_2 = \neg P(u, z) \vee \neg P(z, u)$

Montrons par résolution que $\forall(C_1, C_2)$ n'a pas de modèle.

1. $P(x, y) \vee P(y, x)$ Hyp C_1
2. $P(y, y)$ Facteur de 1 par $< x := y >$
3. $\neg P(u, z) \vee \neg P(z, u)$ Hyp C_2
4. $\neg P(z, z)$ Facteur de 3 par $< u := z >$
5. $\perp$ res bin 2, 4 par $< y := z >$

Résolution : Règles (exemples)

Soient les deux clauses

1. $C_1 = P(x, y) \vee P(y, x)$
2. $C_2 = \neg P(u, z) \vee \neg P(z, u)$

Montrons par résolution que $\forall(C_1, C_2)$ n'a pas de modèle.

1. $P(x, y) \vee P(y, x)$ Hyp C_1
2. $P(y, y)$ Facteur de 1 par $< x := y >$
3. $\neg P(u, z) \vee \neg P(z, u)$ Hyp C_2
4. $\neg P(z, z)$ Facteur de 3 par $< u := z >$
5. $\perp$ res bin 2, 4 par $< y := z >$

Cet exemple montre, a contrario, que la résolution binaire seule est incomplète, sans la factorisation, on ne peut pas déduire la clause vide.

Résolution : Règles (exemples)

Soient les trois clauses

1. $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
2. $C_2 = P(z,f(z)) \vee P(z,a)$
3. $C_3 = P(f(z),z) \vee P(z,a)$

On donne une preuve par résolution de ce que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

Dans cette preuve RB 1(3), 3(1) signifie par résolution binaire sur le 3^{o} littéral de la clause 1 et le 1^{o} littéral de la clause 3.

Résolution : Règles (exemples)

Soient les trois clauses

1. $C_1 = \neg P(z, a) \vee \neg P(z, x) \vee \neg P(x, z)$
2. $C_2 = P(z, f(z)) \vee P(z, a)$
3. $C_3 = P(f(z), z) \vee P(z, a)$

On donne une preuve par résolution de ce que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

Dans cette preuve RB 1(3), 3(1) signifie par résolution binaire sur le 3° littéral de la clause 1 et le 1° littéral de la clause 3.

1. $\neg P(z, a) \vee \neg P(z, x) \vee \neg P(x, z)$ Hyp C_1

Résolution : Règles (exemples)

Soient les trois clauses

1. $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
2. $C_2 = P(z,f(z)) \vee P(z,a)$
3. $C_3 = P(f(z),z) \vee P(z,a)$

On donne une preuve par résolution de ce que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

Dans cette preuve RB 1(3), 3(1) signifie par résolution binaire sur le 3^o littéral de la clause 1 et le 1^o littéral de la clause 3.

1. $\neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$ Hyp C_1
2. $P(z,f(z)) \vee P(z,a)$ Hyp C_2

Résolution : Règles (exemples)

Soient les trois clauses

1. $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
2. $C_2 = P(z,f(z)) \vee P(z,a)$
3. $C_3 = P(f(z),z) \vee P(z,a)$

On donne une preuve par résolution de ce que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

Dans cette preuve RB 1(3), 3(1) signifie par résolution binaire sur le 3° littéral de la clause 1 et le 1° littéral de la clause 3.

1. $\neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$ Hyp C_1
2. $P(z,f(z)) \vee P(z,a)$ Hyp C_2
3. $P(v_0,f(v_0)) \vee P(v_0,a)$ Copie 2 par $< z := v_0 >$

Résolution : Règles (exemples)

Soient les trois clauses

1. $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
2. $C_2 = P(z,f(z)) \vee P(z,a)$
3. $C_3 = P(f(z),z) \vee P(z,a)$

On donne une preuve par résolution de ce que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

Dans cette preuve RB 1(3), 3(1) signifie par résolution binaire sur le 3^o littéral de la clause 1 et le 1^o littéral de la clause 3.

1. $\neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$ Hyp C_1
2. $P(z,f(z)) \vee P(z,a)$ Hyp C_2
3. $P(v_0,f(v_0)) \vee P(v_0,a)$ Copie 2 par $< z := v_0 >$
4. $\neg P(f(v_0),a) \vee \neg P(f(v_0),v_0) \vee P(v_0,a)$ RB 1(3), 3(1) par $< z := f(v_0); x := v_0 >$

Résolution : Règles (exemples)

Soient les trois clauses

1. $C_1 = \neg P(z, a) \vee \neg P(z, x) \vee \neg P(x, z)$
2. $C_2 = P(z, f(z)) \vee P(z, a)$
3. $C_3 = P(f(z), z) \vee P(z, a)$

On donne une preuve par résolution de ce que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

Dans cette preuve RB 1(3), 3(1) signifie par résolution binaire sur le 3^o littéral de la clause 1 et le 1^o littéral de la clause 3.

1. $\neg P(z, a) \vee \neg P(z, x) \vee \neg P(x, z)$ Hyp C_1
2. $P(z, f(z)) \vee P(z, a)$ Hyp C_2
3. $P(v_0, f(v_0)) \vee P(v_0, a)$ Copie 2 par $< z := v_0 >$
4. $\neg P(f(v_0), a) \vee \neg P(f(v_0), v_0) \vee P(v_0, a)$ RB 1(3), 3(1) par $< z := f(v_0); x := v_0 >$
5. $\neg P(f(a), a) \vee P(a, a)$ Fact 4 par $< v_0 := a >$

Résolution : Règles (exemples)

Soient les trois clauses

1. $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
2. $C_2 = P(z,f(z)) \vee P(z,a)$
3. $C_3 = P(f(z),z) \vee P(z,a)$

On donne une preuve par résolution de ce que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

Dans cette preuve RB 1(3), 3(1) signifie par résolution binaire sur le 3° littéral de la clause 1 et le 1° littéral de la clause 3.

1. $\neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$ Hyp C_1
2. $P(z,f(z)) \vee P(z,a)$ Hyp C_2
3. $P(v_0,f(v_0)) \vee P(v_0,a)$ Copie 2 par $< z := v_0 >$
4. $\neg P(f(v_0),a) \vee \neg P(f(v_0),v_0) \vee P(v_0,a)$ RB 1(3), 3(1) par $< z := f(v_0); x := v_0 >$
5. $\neg P(f(a),a) \vee P(a,a)$ Fact 4 par $< v_0 := a >$
6. $\neg P(a,a)$ Fact 1 par $< x := a; z := a >$

Résolution : Règles (exemples)

Soient les trois clauses

1. $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
2. $C_2 = P(z, f(z)) \vee P(z,a)$
3. $C_3 = P(f(z), z) \vee P(z,a)$

On donne une preuve par résolution de ce que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

Dans cette preuve RB 1(3), 3(1) signifie par résolution binaire sur le 3^o littéral de la clause 1 et le 1^o littéral de la clause 3.

1. $\neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$ Hyp C_1
2. $P(z, f(z)) \vee P(z,a)$ Hyp C_2
3. $P(v_0, f(v_0)) \vee P(v_0, a)$ Copie 2 par $< z := v_0 >$
4. $\neg P(f(v_0), a) \vee \neg P(f(v_0), v_0) \vee P(v_0, a)$ RB 1(3), 3(1) par $< z := f(v_0); x := v_0 >$
5. $\neg P(f(a), a) \vee P(a,a)$ Fact 4 par $< v_0 := a >$
6. $\neg P(a,a)$ Fact 1 par $< x := a; z := a >$
7. $P(f(z), z) \vee P(z,a)$ Hyp C_3

Résolution : Règles (exemples)

Soient les trois clauses

1. $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
2. $C_2 = P(z,f(z)) \vee P(z,a)$
3. $C_3 = P(f(z),z) \vee P(z,a)$

On donne une preuve par résolution de ce que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

Dans cette preuve RB 1(3), 3(1) signifie par résolution binaire sur le 3° littéral de la clause 1 et le 1° littéral de la clause 3.

1. $\neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$ Hyp C_1
2. $P(z,f(z)) \vee P(z,a)$ Hyp C_2
3. $P(v_0,f(v_0)) \vee P(v_0,a)$ Copie 2 par $< z := v_0 >$
4. $\neg P(f(v_0),a) \vee \neg P(f(v_0),v_0) \vee P(v_0,a)$ RB 1(3), 3(1) par $< z := f(v_0); x := v_0 >$
5. $\neg P(f(a),a) \vee P(a,a)$ Fact 4 par $< v_0 := a >$
6. $\neg P(a,a)$ Fact 1 par $< x := a; z := a >$
7. $P(f(z),z) \vee P(z,a)$ Hyp C_3
8. $P(f(v_0),v_0) \vee P(v_0,a)$ Copie 7 par $< z := v_0 >$

Résolution : Règles (exemples)

Soient les trois clauses

1. $C_1 = \neg P(z, a) \vee \neg P(z, x) \vee \neg P(x, z)$
2. $C_2 = P(z, f(z)) \vee P(z, a)$
3. $C_3 = P(f(z), z) \vee P(z, a)$

On donne une preuve par résolution de ce que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

Dans cette preuve RB 1(3), 3(1) signifie par résolution binaire sur le 3° littéral de la clause 1 et le 1° littéral de la clause 3.

1. $\neg P(z, a) \vee \neg P(z, x) \vee \neg P(x, z)$ Hyp C_1
2. $P(z, f(z)) \vee P(z, a)$ Hyp C_2
3. $P(v_0, f(v_0)) \vee P(v_0, a)$ Copie 2 par $< z := v_0 >$
4. $\neg P(f(v_0), a) \vee \neg P(f(v_0), v_0) \vee P(v_0, a)$ RB 1(3), 3(1) par $< z := f(v_0); x := v_0 >$
5. $\neg P(f(a), a) \vee P(a, a)$ Fact 4 par $< v_0 := a >$
6. $\neg P(a, a)$ Fact 1 par $< x := a; z := a >$
7. $P(f(z), z) \vee P(z, a)$ Hyp C_3
8. $P(f(v_0), v_0) \vee P(v_0, a)$ Copie 7 par $< z := v_0 >$
9. $P(f(a), a)$ RB 6, 8 par $< v_0 := a >$

Résolution : Règles (exemples)

Soient les trois clauses

1. $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
2. $C_2 = P(z,f(z)) \vee P(z,a)$
3. $C_3 = P(f(z),z) \vee P(z,a)$

On donne une preuve par résolution de ce que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

Dans cette preuve RB 1(3), 3(1) signifie par résolution binaire sur le 3° littéral de la clause 1 et le 1° littéral de la clause 3.

1. $\neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$ Hyp C_1
2. $P(z,f(z)) \vee P(z,a)$ Hyp C_2
3. $P(v_0,f(v_0)) \vee P(v_0,a)$ Copie 2 par $< z := v_0 >$
4. $\neg P(f(v_0),a) \vee \neg P(f(v_0),v_0) \vee P(v_0,a)$ RB 1(3), 3(1) par $< z := f(v_0); x := v_0 >$
5. $\neg P(f(a),a) \vee P(a,a)$ Fact 4 par $< v_0 := a >$
6. $\neg P(a,a)$ Fact 1 par $< x := a; z := a >$
7. $P(f(z),z) \vee P(z,a)$ Hyp C_3
8. $P(f(v_0),v_0) \vee P(v_0,a)$ Copie 7 par $< z := v_0 >$
9. $P(f(a),a)$ RB 6, 8 par $< v_0 := a >$
10. $P(a,a)$ RB 5, 9

Résolution : Règles (exemples)

Soient les trois clauses

1. $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
2. $C_2 = P(z,f(z)) \vee P(z,a)$
3. $C_3 = P(f(z),z) \vee P(z,a)$

On donne une preuve par résolution de ce que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

Dans cette preuve RB 1(3), 3(1) signifie par résolution binaire sur le 3^o littéral de la clause 1 et le 1^o littéral de la clause 3.

1. $\neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$ Hyp C_1
2. $P(z,f(z)) \vee P(z,a)$ Hyp C_2
3. $P(v_0,f(v_0)) \vee P(v_0,a)$ Copie 2 par $< z := v_0 >$
4. $\neg P(f(v_0),a) \vee \neg P(f(v_0),v_0) \vee P(v_0,a)$ RB 1(3), 3(1) par $< z := f(v_0); x := v_0 >$
5. $\neg P(f(a),a) \vee P(a,a)$ Fact 4 par $< v_0 := a >$
6. $\neg P(a,a)$ Fact 1 par $< x := a; z := a >$
7. $P(f(z),z) \vee P(z,a)$ Hyp C_3
8. $P(f(v_0),v_0) \vee P(v_0,a)$ Copie 7 par $< z := v_0 >$
9. $P(f(a),a)$ RB 6, 8 par $< v_0 := a >$
10. $P(a,a)$ RB 5, 9
11. $\perp$ Res Bin 6, 10

Plan

Résolution 1° ordre

On définit une *nouvelle* règle, la résolution au 1° ordre, qui est une combinaison des trois règles de factorisation, copie et résolution binaire.

Résolution 1° ordre

On définit une *nouvelle* règle, la résolution au 1° ordre, qui est une combinaison des trois règles de factorisation, copie et résolution binaire.

Définition La clause E est un résolvant au 1° ordre des clauses C et D si E est un résolvant binaire de C' et D' où C' est un facteur de C et D' est une copie sans variable commune avec C' d'un facteur de D.
La règle qui de C et D permet de déduire E est appelée la résolution de 1° ordre.

Résolution 1° ordre

On définit une *nouvelle* règle, la résolution au 1° ordre, qui est une combinaison des trois règles de factorisation, copie et résolution binaire.

Définition La clause E est un résolvant au 1° ordre des clauses C et D si E est un résolvant binaire de C' et D' où C' est un facteur de C et D' est une copie sans variable commune avec C' d'un facteur de D.
La règle qui de C et D permet de déduire E est appelée la résolution de 1° ordre.

Exemple Soient $C = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$ et $D = P(z,f(z)) \vee P(z,a)$.
$C' = \neg P(a,a)$ est un facteur de C. La clause $P(a,f(a))$ est un résolvant binaire de C' et de D (qui est facteur de lui-même) donc c'est un résolvant de C et D.

Trois notions de preuve par résolution

Soit Γ un ensemble de clauses et C une clause.

Trois notions de preuve par résolution

Soit Γ un ensemble de clauses et C une clause.

1. On note $\Gamma \vdash_p C$ le fait qu'il y a une preuve de C à partir de Γ obtenue par résolution propositionnelle, autrement dit sans substitution.

Trois notions de preuve par résolution

Soit Γ un ensemble de clauses et C une clause.

1. On note $\Gamma \vdash_p C$ le fait qu'il y a une preuve de C à partir de Γ obtenue par résolution propositionnelle, autrement dit sans substitution.
2. On rappelle que $\Gamma \vdash_{1fcb} C$ signifie qu'il y a une preuve de C à partir de Γ par factorisation, copie et résolution binaire.

Trois notions de preuve par résolution

Soit Γ un ensemble de clauses et C une clause.

1. On note $\Gamma \vdash_p C$ le fait qu'il y a une preuve de C à partir de Γ obtenue par résolution propositionnelle, autrement dit sans substitution.
2. On rappelle que $\Gamma \vdash_{1fcb} C$ signifie qu'il y a une preuve de C à partir de Γ par factorisation, copie et résolution binaire.
3. On note $\Gamma \vdash_{1r} C$ le fait qu'il y a une preuve de C à partir de Γ obtenue par résolution de 1^{o} ordre.

Trois notions de preuve par résolution

Soit Γ un ensemble de clauses et C une clause.

1. On note $\Gamma \vdash_p C$ le fait qu'il y a une preuve de C à partir de Γ obtenue par résolution propositionnelle, autrement dit sans substitution.
2. On rappelle que $\Gamma \vdash_{1fcb} C$ signifie qu'il y a une preuve de C à partir de Γ par factorisation, copie et résolution binaire.
3. On note $\Gamma \vdash_{1r} C$ le fait qu'il y a une preuve de C à partir de Γ obtenue par résolution de 1° ordre.

Puisque la résolution du premier ordre est une combinaison des règles factorisation, copie et résolution binaire, on a :
$\Gamma \vdash_{1r} C$ implique $\Gamma \vdash_{1fcb} C$

Lemme du relèvement (1/3)

Théorème

Soient C et D deux clauses. Soit C' une instance de C et D' une instance de D. Soit E' un résolvant propositionnel de C' et D', il existe E un résolvant premier ordre de C et D qui a pour instance E'.

Lemme du relèvement (1/3)

Théorème

Soient C et D deux clauses. Soit C' une instance de C et D' une instance de D. Soit E' un résolvant propositionnel de C' et D', il existe E un résolvant premier ordre de C et D qui a pour instance E'.

Exemple : Soient $C = P(x) \vee P(y) \vee R(u) \vee R(v)$ et $D = \neg Q(x) \vee P(x) \vee \neg R(x) \vee P(y)$.
Les clauses $C' = P(a) \vee R(a)$ et $D' = \neg Q(a) \vee P(a) \vee \neg R(a)$ sont des instances respectivement de C et D.
La clause $E' = P(a) \vee \neg Q(a)$ est un résolvant propositionnel de C' et D'.
La clause $E = P(x) \vee \neg Q(x)$ est un résolvant au 1^{o} ordre de C et D qui a pour instance E'.

Lemme du relèvement (2/3)

Théorème

Soit Γ un ensemble de clauses et Δ un ensemble d'instances des clauses de Γ. Soit $C_1, \ldots, C_n$ une preuve par résolution propositionnelle à partir de Δ. Il existe une preuve $D_1, \ldots, D_n$ par résolution 1° ordre à partir Γ telle que pour i de 1 à n, la clause C_i est une instance de D_i.

Démonstration.

Preuve par récurrence. □

Lemme du relèvement (3/3)

Corollaire du relèvement

Soit Γ un ensemble de clauses et Δ un ensemble d'instances des clauses de Γ. Supposons que $\Delta \vdash_p C$. Il existe D telle que $\Gamma \vdash_{1r} D$ et C est une instance de D.

Lemme du relèvement : exemple

Soit l'ensemble de clauses
$P(f(x)) \vee P(u), \neg P(x) \vee Q(z), \neg Q(x) \vee \neg Q(y)$.
La fermeture universelle de cet ensemble de clauses est insatisfaisable et nous le montrons de trois manières

1. Par instanciation sur le domaine de Herbrand $a, f(a), f(f(a)), \ldots$:
   $P(f(x)) \vee P(u)$ est instanciée par $x := a, u := f(a)$ en $P(f(a))$
   $\neg P(x) \vee Q(z)$ est instanciée par $x := f(a), z := a$ en $\neg P(f(a)) \vee Q(a)$
   $\neg Q(x) \vee \neg Q(y)$ est instanciée par $x := a, y :=$ en $\neg Q(a)$
   L'ensemble de ces 3 instanciations est insatisfaisable, comme le montre la preuve par résolution propositionnelle ci-dessous :

$$\dfrac{\dfrac{P(f(a)) \qquad \neg P(f(a)) \vee Q(a)}{Q(a)} \qquad \neg Q(a)}{\perp}$$

Lemme du relèvement : exemple

Soit l'ensemble de clauses
$P(f(x)) \vee P(u), \neg P(x) \vee Q(z), \neg Q(x) \vee \neg Q(y)$.
La fermeture universelle de cet ensemble de clauses est insatisfaisable et nous le montrons de trois manières

Lemme du relèvement : exemple

Soit l'ensemble de clauses
$P(f(x)) \vee P(u), \neg P(x) \vee Q(z), \neg Q(x) \vee \neg Q(y)$.
La fermeture universelle de cet ensemble de clauses est insatisfaisable et nous le montrons de trois manières

2. Cette preuve par résolution propositionnelle est relevée en une preuve par la règle de résolution au premier ordre :

$$\dfrac{\dfrac{P(f(x))\vee P(u) \qquad \neg P(x)\vee Q(z)}{Q(z)} \qquad \neg Q(x)\vee\neg Q(y)}{\perp}$$

Lemme du relèvement : exemple

Soit l'ensemble de clauses
$P(f(x)) \vee P(u), \neg P(x) \vee Q(z), \neg Q(x) \vee \neg Q(y)$.
La fermeture universelle de cet ensemble de clauses est insatisfaisable et nous le montrons de trois manières

2. Cette preuve par résolution propositionnelle est relevée en une preuve par la règle de résolution au premier ordre :

$$\dfrac{\dfrac{P(f(x)) \vee P(u) \qquad \neg P(x) \vee Q(z)}{Q(z)} \qquad \neg Q(x) \vee \neg Q(y)}{\perp}$$

3. Chaque règle de résolution au premier ordre est décomposée en factorisation, copie et résolution binaire :

$$\dfrac{\dfrac{\dfrac{P(fx)) \vee P(u)}{P((f(x))}\,fact \quad \dfrac{\neg P(x) \vee Q(z)}{\neg P(y) \vee Q(z)}\,copie}{Q(z)}\,rd \qquad \dfrac{\neg Q(x) \vee \neg Q(y)}{\neg Q(x)}\,fact}{\perp}\,rb$$

Complétude réfutationnelle de la résolution au 1° ordre

Théorème

Soit Γ un ensemble de clauses. Les propositions suivantes sont équivalentes :

1. $\Gamma \vdash_{1r} \bot$
2. $\Gamma \vdash_{1fcb} \bot$
3. $\forall(\Gamma) \models \bot$

Complétude réfutationnelle de la résolution au 1° ordre

Théorème

Soit Γ un ensemble de clauses. Les propositions suivantes sont équivalentes :

1. $\Gamma \vdash_{1r} \perp$
2. $\Gamma \vdash_{1fcb} \perp$
3. $\forall(\Gamma) \models \perp$

Démonstration.

On sait que (1) implique (2), car la résolution au 1° ordre est une combinaison de factorisation, copie et résolution binaire. On sait que (2) implique (3), car la factorisation, la copie et la résolution binaire sont cohérentes. Il nous reste à montrer que (3) implique (1).

Supposons que $\forall(\Gamma) \models \perp$, autrement dit $\forall(\Gamma)$ est insatisfaisable. D'après le théorème de Herbrand, il y a Δ un ensemble fini d'instances sans variable de clauses de Γ qui n'a pas de modèle propositionnel. Par complétude de la résolution propositionnelle, on a : $\Delta \vdash_p \perp$. D'après le corollaire au relèvement, il existe D telle que $\Gamma \vdash_{1r} D$ et $\perp$ instance de D. Mais dans ce cas, on a $D = \perp$. $\square$