Base de la démontration automatique : Résolution au premier ordre

Stéphane Devismes Pascal Lafourcade Michel Lévy

Université Grenoble Alpes

Avril 2010

Plan

Plan

Idée

Avec la skolémisation, on obtient des formules sans quantificateur.

Aujourd'hui, nous proposons une généralisation au premier ordre de la résolution :

- Mise en forme clausale des formes de skolem.
- Définition de la généralisation de la résolution.
- Cohérence et complétude de la méthode.

Plan

Littéral, clause

Définition 5.2.19

Un littéral positif est une formule atomique. Ex : $P(x, y)$

Un littéral négatif est la négation d'une formule atomique. Ex : $\neg Q(a)$

Tout littéral est positif ou négatif.

Une clause est une somme de littéraux. Ex : $P(x, y) \vee \neg Q(a)$

Forme clausale d'une formule

Définition 5.2.20

Soit A une formule fermée. La forme clausale de A, $F(A)$ est un ensemble de clauses obtenu 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

Forme clausale d'une formule

Propriété 5.2.21

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

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

Preuve

Preuve.

Soient A une formule fermée, B sa forme de Skolem et Γ sa forme clausale. D'après les propriétés de la skolémisation :

- $\forall(B)$ a pour conséquence A.
- Si A a un modèle alors $\forall(B)$ a un modèle.

Puisque Γ est obtenu par distributivité, B et Γ sont équivalents donc $\forall(B)$ et $\forall(\Gamma)$ sont aussi équivalents. Par suite dans les deux propriétés ci-dessus, nous pouvons remplacer $\forall(B)$ par $\forall(\Gamma)$. □

Forme clausale d'un ensemble de formules

Définition 5.2.22

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

Forme clausale d'un ensemble de formules

Corollaire 5.2.23

Soient Γ un ensemble de formules fermées et Δ la forme clausale de Γ. Nous avons :

- $\forall(\Delta)$ a pour conséquence Γ, et
- si Γ a un modèle alors $\forall(\Delta)$ a un modèle.

Adaptation du théorème de Herbrand aux formes clausales

Théorème 5.2.24

Soient Γ un ensemble de formules fermées et Δ la forme clausale de Γ. L'ensemble 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.

Preuve.

D'après le corollaire 5.2.23, 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 5.1.18, $\forall(\Delta)$ est insatisfaisable si et seulement s'il existe un sous-ensemble fini insatisfaisable d'instances des clauses de Δ sur la signature de Δ. $\square$

Exemple 5.2.25 (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 5.2.25 (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 5.2.25 (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 5.2.25 (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 5.2.25 (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 5.2.25 (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 5.2.25 (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 5.2.25 (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 5.2.25 (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 5.2.25 (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 5.2.25 (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 5.2.25 (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

Unification : expression, solution

Définition 5.3.1

- Un terme ou un littéral est une **expression**.
- Une substitution σ (voir définition 5.1.3) est **solution** de l'équation $e_1 = e_2$ entre deux expressions, si les deux expressions $e_1\sigma$ et $e_2\sigma$ sont syntaxiquement identiques.
- Une substitution est solution d'un ensemble d'équations si elle est solution de chaque équation de l'ensemble.

Unification : support de substitution

Définition 5.3.3

Le support d'une substitution σ est l'ensemble des variables x telles que $x\sigma \neq x$.

Nous considérons que des substitutions à support fini (nombre fini de variables).

Définition 5.3.3

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 5.3.4

L'équation $P(x, f(y)) = P(g(z), z)$ a pour solution :

Le système d'équations $x = g(z), f(y) = z$ a pour solution :

Unification : exemple 5.3.4

L'équation $P(x, f(y)) = P(g(z), z)$ a pour solution :

$x := g(f(y)), z := f(y)$.

Le système d'équations $x = g(z), f(y) = z$ a pour solution :

$x := g(f(y)), z := f(y)$.

Unification : composition de substitution

Définition 5.3.5

- 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 5.3.6

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 :

Unification : exemple 5.3.6

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

Définition 5.3.7 (mgu)

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

Définition 5.3.7 (mgu)

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 5.3.8

Considérons l'équation $f(x, g(z)) = f(g(y), x)$.

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

sont 3 solutions.

σ_1 et σ_2 en sont les solutions les plus générales.

Unificateur

Définition 5.3.2

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 $\{e_i = e_{i+1} | 1 \leq i < n\}$.

Unificateur le plus général

Définition 5.3.9

Soit E un ensemble d'expressions. Nous rappelons qu'une expression est un terme ou un littéral. Un unificateur de E est appelé le plus général (ou encore principal), si tout autre unificateur en est une instance.

Unificateur le plus général et solution la plus générale

Remarque 5.3.10

Soit $E = \{e_i \mid 1 \leq i \leq n\}$ un ensemble d'expressions.
Dans la définition d'un unificateur, nous avons indiqué que σ est un unificateur de E si et seulement si σ est solution du système $S = \{e_i = e_{i+1} \mid 1 \leq i < n\}$.
Donc l'unificateur le plus général de E est la solution la plus générale de S.

Unification : l'algorithme (plan)

L'algorithme sépare les équations en :

- équations à résoudre, notées par une égalité
- équations résolues, notées par le signe $:=$

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

L'algorithme s'arrête quand :

- il n'y a plus d'équations à résoudre : la liste des équations résolues est la solution la plus générale du système initial d'équations.
- ou quand il a déclaré que le système à résoudre n'a pas de solution.

Unification : l'algorithme (les règles)

- **Supprimer l'équation.** Si les 2 membres d'une équation sont identiques.
- **Décomposer.** Si les 2 membres d'une équation sont distincts :
  - $\neg A = \neg B$, devient $A = B$.
  - $f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)$, devient $s_1 = t_1, \ldots, s_n = t_n$.
    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 $f \neq g$ ou $n \neq p$ 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. l'enlever des équations à résoudre
  2. remplacer x par t dans toutes les équations (non résolues et résolues)
  3. ajouter $x := t$ 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 (exemple 5.3.11)

Unification : l'algorithme (exemple 5.3.11)

1. Résoudre $f(x, g(z)) = f(g(y), x)$.

Unification : l'algorithme (exemple 5.3.11)

1. Résoudre $f(x, g(z)) = f(g(y), x)$.

Unification : l'algorithme (exemple 5.3.11)

1. Résoudre $f(x, g(z)) = f(g(y), x)$.

Par décomposition, on obtient : $x = g(y), g(z) = x$

Unification : l'algorithme (exemple 5.3.11)

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 : $x := g(y), g(z) = g(y)$

Unification : l'algorithme (exemple 5.3.11)

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 : $x := g(y), z = y$

Unification : l'algorithme (exemple 5.3.11)

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 (exemple 5.3.11)

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 $f(x, x, a) = f(g(y), g(a), y)$.

Unification : l'algorithme (exemple 5.3.11)

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 : $x = g(y), x = g(a), a = y$

Unification : l'algorithme (exemple 5.3.11)

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 :
   $x := g(y), g(y) = g(a), a = y$

Unification : l'algorithme (exemple 5.3.11)

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 : $x := g(y), y = a, a = y$

Unification : l'algorithme (exemple 5.3.11)

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 : $x := g(a), y := a, a = a$

Unification : l'algorithme (exemple 5.3.11)

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 (exemple 5.3.11)

1. Résoudre $f(x, x, x) = f(g(y), g(a), y)$.

Unification : l'algorithme (exemple 5.3.11)

1. Résoudre $f(x,x,x) = f(g(y),g(a),y)$.

Par décomposition, on obtient : $x = g(y), x = g(a), x = y$

Unification : l'algorithme (exemple 5.3.11)

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 : $x := g(y), g(y) = g(a), g(y) = y$

Unification : l'algorithme (exemple 5.3.11)

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 :
$x := g(y), g(y) = g(a), y = g(y)$

Unification : l'algorithme (exemple 5.3.11)

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.

Unification : l'algorithme (exemple 5.3.11)

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.

Remarque : les preuves de correction et terminaison l'algorithme d'unification sont dans le poly.

Plan

Idée

Soit Γ un ensemble de clauses. Supposons que $\forall(\Gamma)$ n'a pas de modèle. Que faire ?

Idée

Soit Γ un ensemble de clauses. Supposons que $\forall(\Gamma)$ n'a pas de modèle. Que faire ?

Le système formel $\ll$ factorisation, copie, résolution binaire $\gg$ est un système formel permettant de déduire $\bot$ de Γ.

Idée

Soit Γ un ensemble de clauses. Supposons que $\forall(\Gamma)$ n'a pas de modèle. Que faire ?

Le système formel $\ll$ factorisation, copie, résolution binaire $\gg$ est un système formel permettant de déduire $\bot$ de Γ.

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.

Trois règles

1. La factorisation qui de la prémisse $P(x, f(y)) \vee P(g(z), z) \vee Q(z, x)$ déduit $P(g(f(y)), f(y)) \vee Q(f(y), g(f(y)))$. La clause déduite est obtenue en calculant la solution la plus générale $x := g(f(y)), z := f(y)$ 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 (RB) qui des deux prémisses sans variable commune $P(x, a) \vee Q(x)$ et $\neg P(b, y) \vee R(f(y))$ déduit le résolvant $Q(b) \vee R(f(a))$, en calculant la solution plus générale $x := b, y := a$ de $P(x, a) = P(b, y)$.

Résolution : 3 Règles de résolutions

1. factorisation,
2. copie,
3. résolvant

Résolution : 3 Règles de résolutions

1. factorisation,
2. copie,
3. résolvant

Une clause, qui est une somme de littéraux, est identifiée avec l'ensemble de ses littéraux.

Factorisation

Définition 5.4.2

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.

Factorisation

Définition 5.4.2

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 5.4.3

La clause $\underline{P(x)} \vee Q(g(x,y)) \vee \underline{P(f(a))}$ a deux facteurs :

Factorisation

Définition 5.4.2

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 5.4.3

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.

Factorisation

Propriété 5.4.1

Soient A une formule sans quantificateur et B une instance de A.
$\forall(A) \models \forall(B)$

Preuve.

Cf. Poly □

Factorisation

Propriété 5.4.1

Soient A une formule sans quantificateur et B une instance de A.
$\forall(A) \models \forall(B)$

Preuve.

Cf. Poly □

Propriété 5.4.4

Soit C' un facteur de la clause C.
$\forall(C) \models \forall(C')$

Preuve.

Puisque C' est une instance de C, c'est une conséquence de la propriété 5.4.1. □

Copie

Définition 5.4.5

Soient C une clause et σ une substitution, qui ne change que les variables de C et dont la restriction aux variables de C est une bijection entre ces variables et celles de la clause $C\sigma$.

La clause $C\sigma$ est une copie de la clause C.

La substitution σ est aussi appelée un renommage de C.

Copie

Définition 5.4.6

Soient C une clause et σ un renommage de C. Soit f la restriction de σ aux variables de C et f^{-1} l'application réciproque de f. Soit σ_C^{-1} la substitution ainsi définie pour toute variable x :

- Si x est une variable de $C\sigma$ alors $x\sigma_C^{-1} = xf^{-1}$
- Sinon $x\sigma_C^{-1} = x$.

Cette substitution est appelée l'inverse du renommage σ de C.

Copie

Exemple 5.4.7

Soit $\sigma = < x := u, y := v >$.

σ est un renommage de $P(x, y)$.

Le littéral $P(u, v)$, où $P(u, v) = P(x, y)\sigma$, est une copie de $P(x, y)$.

Soit $\tau = < u := x, v := y >$. τ est l'inverse du renommage σ de $P(x, y)$.

Notons que $P(u, v)\tau = P(x, y)$: le littéral $P(x, y)$ est une copie de $P(u, v)$ par le renommage τ.

Copie

Propriété 5.4.8

Soient C une clause et σ un renommage de C.

1. σ_C^{-1} est un renommage de $C\sigma$.
2. Pour toute expression ou clause E, dont les variables sont celles de C, $E\sigma\sigma_C^{-1} = E$.

Donc $C\sigma\sigma_C^{-1} = C$ et par suite C est une copie de $C\sigma$.

Preuve.

Soit f la restriction de σ aux variables de C. Par définition du renommage, f est une bijection entre les variables de C et celles de $C\sigma$.

1. Par définition de σ_C^{-1}, cette substitution ne change que les variables de $C\sigma$ et sa restriction aux variables de $C\sigma$ est la bijection f^{-1}. Donc, σ_C^{-1} est un renommage de $C\sigma$.
2. Soit x une variable de C. Par définition de f, $x\sigma\sigma_C^{-1} = xff^{-1} = x$. Donc, par une récurrence sur les termes, littéraux et clauses, pour toute expression ou clause E, dont les variables sont celles de C, nous avons $E\sigma\sigma_C^{-1} = E$.

□

Copie

Propriété 5.4.9

Soient deux clauses copies l'une de l'autre, leurs fermetures universelles sont équivalentes.

Preuve.

Soit C' une copie de C. Par définition, C' est une instance de C et par la propriété précédente, C est une copie de C', donc une instance de C'.

Donc par la propriété 5.4.1, la fermeture universelle de C est conséquence de celle de C' et inversement. Par suite, ces deux fermetures universelles sont équivalentes. □

Résolvant binaire

Définition 5.4.10

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 sont unifiables et si $E = ((C - \{L\}) \cup (D - \{M\}))\sigma$ où σ est la solution la plus générale de l'équation $L = M^c$.

Résolvant binaire

Définition 5.4.10

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 sont unifiables et si $E = ((C - \{L\}) \cup (D - \{M\}))\sigma$ où σ est la solution la plus générale de l'équation $L = M^c$.

Exemple 5.4.11

Soit $C = P(x, y) \vee P(y, k(z))$ et $D = \neg P(a, f(a, y_1))$.

Résolvant binaire

Définition 5.4.10

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 sont unifiables et si $E = ((C - \{L\}) \cup (D - \{M\}))\sigma$ où σ est la solution la plus générale de l'équation $L = M^c$.

Exemple 5.4.11

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.

Résolvant binaire

Propriété 5.4.12

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

Preuve.

Cf. Poly □

Résolution :

Définition 5.4.13

Soient 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 est

- 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 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, nous remplaçons $\vdash_{1fcb}$ par $\vdash$.

Résolution : Cohérence

Propriété 5.4.14

Soient Γ un ensemble de clauses et C une clause.

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. Cette preuve est une induction demandée dans l'exercice 97.

Résolution : Exemple 5.4.15

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 : Exemple 5.4.15

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 : Exemple 5.4.15

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 : Exemple 5.4.15

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 : Exemple 5.4.15

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 : Exemple 5.4.15

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. $\perp$ RB 2, 4 par $< y := z >$

Résolution : Exemple 5.4.15

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. $\perp$ RB 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 : Exemple 5.4.16

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 que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

Résolution : Exemple 5.4.16

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 que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

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

Résolution : Exemple 5.4.16

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 que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

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 : Exemple 5.4.16

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 que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

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 : Exemple 5.4.16

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 que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

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 : Exemple 5.4.16

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 que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

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 : Exemple 5.4.16

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 que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

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 : Exemple 5.4.16

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 que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

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 : Exemple 5.4.16

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 que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

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. $P(f(a),a)$ RB 6(1), 7(2) par $< z := a >$

Résolution : Exemple 5.4.16

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 que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

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. $P(f(a),a)$ RB 6(1), 7(2) par $< z := a >$

Résolution : Exemple 5.4.16

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 que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

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. $P(f(a),a)$ RB 6(1), 7(2) par $< z := a >$
9. $P(a,a)$ RB 5(1), 8(1)

Résolution : Exemple 5.4.16

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 que $\forall(C_1, C_2, C_3)$ n'a pas de modèle.

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. $P(f(a),a)$ RB 6(1), 7(2) par $< z := a >$
9. $P(a,a)$ RB 5(1), 8(1)
10. $\perp$ RB 6(1), 9(1)

Plan

Résolution 1^o ordre

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

Définition 5.4.17

La clause E est un résolvant au 1^o 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^o ordre.

Exemple 5.4.18

Soient $C = \neg P(z, a) \vee \neg P(z, x) \vee \neg P(x, z)$ et $D = P(z, f(z)) \vee P(z, a)$.

Exemple 5.4.18

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 au premier ordre de C et D.

Trois notions de preuve par résolution

Soient Γ un ensemble de clauses et C une clause.

Notations

Trois notions de preuve par résolution

Soient Γ un ensemble de clauses et C une clause.

Notations

1. $\Gamma \vdash_p C$: preuve de C à partir de Γ par résolution propositionnelle (sans substitution).

Trois notions de preuve par résolution

Soient Γ un ensemble de clauses et C une clause.

Notations

1. $\Gamma \vdash_p C$: preuve de C à partir de Γ par résolution propositionnelle (sans substitution).
2. $\Gamma \vdash_{1fcb} C$: preuve de C à partir de Γ par factorisation, copie et résolution binaire.

Trois notions de preuve par résolution

Soient Γ un ensemble de clauses et C une clause.

Notations

1. $\Gamma \vdash_p C$: preuve de C à partir de Γ par résolution propositionnelle (sans substitution).
2. $\Gamma \vdash_{1fcb} C$: preuve de C à partir de Γ par factorisation, copie et résolution binaire.
3. $\Gamma \vdash_{1r} C$: preuve de C à partir de Γ obtenue par résolution de 1^o ordre.

Trois notions de preuve par résolution

Soient Γ un ensemble de clauses et C une clause.

Notations

1. $\Gamma \vdash_p C$: preuve de C à partir de Γ par résolution propositionnelle (sans substitution).
2. $\Gamma \vdash_{1fcb} C$: preuve de C à partir de Γ par factorisation, copie et résolution binaire.
3. $\Gamma \vdash_{1r} C$: preuve de C à partir de Γ obtenue par résolution de 1^o ordre.

Par définition nous avons : $\Gamma \vdash_{1r} C$ implique $\Gamma \vdash_{1fcb} C$

Théorème du relèvement (1/3)

Théorème 5.4.19

Soient C et D deux clauses. Soient 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'.

Preuve.

Cf. Poly. □

Théorème du relèvement (1/3)

Théorème 5.4.19

Soient C et D deux clauses. Soient 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'.

Preuve.

Cf. Poly. □

Exemple 5.4.20

Soient $C = P(x) \vee P(y) \vee R(y)$ et $D = \neg Q(x) \vee P(x) \vee \neg R(x) \vee P(y)$.

Théorème du relèvement (1/3)

Théorème 5.4.19

Soient C et D deux clauses. Soient 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'.

Preuve.

Cf. Poly. □

Exemple 5.4.20

Soient $C = P(x) \vee P(y) \vee R(y)$ 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.

Théorème du relèvement (1/3)

Théorème 5.4.19

Soient C et D deux clauses. Soient 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'.

Preuve.

Cf. Poly. □

Exemple 5.4.20

Soient $C = P(x) \vee P(y) \vee R(y)$ 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'.

Théorème du relèvement (1/3)

Théorème 5.4.19

Soient C et D deux clauses. Soient 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'.

Preuve.

Cf. Poly. □

Exemple 5.4.20

Soient $C = P(x) \vee P(y) \vee R(y)$ 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° ordre de C et D qui a pour instance E'.

Détail de l'exemple 5.4.20

Détail de l'exemple 5.4.20

- $C' = P(x) \vee R(x)$ est un facteur de $C = P(x) \vee P(y) \vee R(y)$ par $< y := x >$

Détail de l'exemple 5.4.20

- $C' = P(x) \vee R(x)$ est un facteur de $C = P(x) \vee P(y) \vee R(y)$ par $< y := x >$
- $\neg Q(x) \vee P(x) \vee \neg R(x)$ est un facteur de $D = \neg Q(x) \vee P(x) \vee \neg R(x) \vee P(y)$ par $< y := x >$

Détail de l'exemple 5.4.20

- $C' = P(x) \vee R(x)$ est un facteur de $C = P(x) \vee P(y) \vee R(y)$ par $< y := x >$
- $\neg Q(x) \vee P(x) \vee \neg R(x)$ est un facteur de $D = \neg Q(x) \vee P(x) \vee \neg R(x) \vee P(y)$ par $< y := x >$
- $D' = \neg Q(x_0) \vee P(x_0) \vee \neg R(x_0)$ est une copie de $\neg Q(x) \vee P(x) \vee \neg R(x)$ par $< x := x_0 >$

Détail de l'exemple 5.4.20

- $C' = P(x) \vee R(x)$ est un facteur de $C = P(x) \vee P(y) \vee R(y)$ par $< y := x >$
- $\neg Q(x) \vee P(x) \vee \neg R(x)$ est un facteur de $D = \neg Q(x) \vee P(x) \vee \neg R(x) \vee P(y)$ par $< y := x >$
- $D' = \neg Q(x_0) \vee P(x_0) \vee \neg R(x_0)$ est une copie de $\neg Q(x) \vee P(x) \vee \neg R(x)$ par $< x := x_0 >$
- $P(x) \vee \neg Q(x)$ est un résolvant binaire de C' et D' par $< x_0 := x >$ donc un résolvant au 1° ordre de C et D qui a pour instance E'

Théorème du relèvement (2/3)

Théorème 5.4.21

Soient Γ un ensemble de clauses et Δ un ensemble d'instances des clauses de Γ, et $C_1, \ldots, C_n$ une preuve par résolution propositionnelle à partir de Δ.

Il existe une preuve $D_1, \ldots, D_n$ par résolution 1^o ordre à partir Γ telle que pour i de 1 à n, la clause C_i est une instance de D_i.

Théorème du relèvement (2/3)

Théorème 5.4.21

Soient Γ un ensemble de clauses et Δ un ensemble d'instances des clauses de Γ, et $C_1, \ldots, C_n$ une preuve par résolution propositionnelle à partir de Δ.

Il existe une preuve $D_1, \ldots, D_n$ par résolution 1^o ordre à partir Γ telle que pour i de 1 à n, la clause C_i est une instance de D_i.

Preuve.

Par récurrence sur n. Soit $C_1, \ldots, C_n, C_{n+1}$ une preuve par résolution propositionnelle à partir de Δ. Par récurrence, il existe une preuve $D_1, \ldots, D_n$ par résolution 1^o ordre à partir Γ telle que pour i de 1 à n, la clause C_i est une instance de D_i.

1. Supposons $C_{n+1} \in \Delta$. Il existe $E \in \Gamma$ dont C_{n+1} est une instance donc nous prenons $D_{n+1} = E$.
2. Supposons que C_{n+1} est un résolvant propositionnel de C_j et C_k où $j, k \leq n$. D'après le transparent précédent, il existe E résolvant au 1^o ordre de D_j et D_k : nous prenons $D_{n+1} = E$.

Théorème du relèvement (3/3)

Corollaire 5.4.22

Soient 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.

Exemple 5.4.23

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 $a, f(a), f(f(a)), \ldots$:

Exemple 5.4.23

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 $x := a, y := a$ en $\neg Q(a)$
   L'ensemble de ces 3 instanciations est insatisfaisable, comme le montre la preuve par résolution propositionnelle ci-dessous :

Exemple 5.4.23

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 $x := a, y := a$ en $\neg Q(a)$
   L'ensemble de ces 3 instanciations est insatisfaisable, comme le montre la preuve par résolution propositionnelle ci-dessous :

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

Exemple 5.4.23

$$P(f(x)) \vee P(u), \neg P(x) \vee Q(z), \neg Q(x) \vee \neg Q(y).$$

Exemple 5.4.23

$P(f(x)) \vee P(u), \neg P(x) \vee Q(z), \neg Q(x) \vee \neg Q(y)$.

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

Exemple 5.4.23

$P(f(x)) \vee P(u), \neg P(x) \vee Q(z), \neg Q(x) \vee \neg Q(y)$.

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

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

Exemple 5.4.23

$P(f(x)) \vee P(u), \neg P(x) \vee Q(z), \neg Q(x) \vee \neg Q(y)$.

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

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

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

Exemple 5.4.23

$P(f(x)) \vee P(u), \neg P(x) \vee Q(z), \neg Q(x) \vee \neg Q(y).$

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

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

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

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

Complétude réfutationnelle de la résolution au 1^o ordre

Théorème 5.4.24

Soit Γ un ensemble de clauses. Les propositions : (1) $\Gamma \vdash_{1r} \bot$, (2) $\Gamma \vdash_{1fcb} \bot$, et (3) $\forall(\Gamma) \models \bot$ sont équivalentes.

Complétude réfutationnelle de la résolution au 1^o ordre

Théorème 5.4.24

Soit Γ un ensemble de clauses. Les propositions : (1) $\Gamma \vdash_{1r} \bot$, (2) $\Gamma \vdash_{1fcb} \bot$, et (3) $\forall(\Gamma) \models \bot$ sont équivalentes.

Démonstration.

- (1) implique (2) car la résolution au 1^o ordre est une combinaison de factorisation, copie et résolution binaire.

Complétude réfutationnelle de la résolution au 1^o ordre

Théorème 5.4.24

Soit Γ un ensemble de clauses. Les propositions : (1) $\Gamma \vdash_{1r} \bot$, (2) $\Gamma \vdash_{1fcb} \bot$, et (3) $\forall(\Gamma) \models \bot$ sont équivalentes.

Démonstration.

- (1) implique (2) car la résolution au 1^o ordre est une combinaison de factorisation, copie et résolution binaire.
- (2) implique (3) car la factorisation, la copie et la résolution binaire sont cohérentes.

Complétude réfutationnelle de la résolution au 1^o ordre

Théorème 5.4.24

Soit Γ un ensemble de clauses. Les propositions : (1) $\Gamma \vdash_{1r} \bot$, (2) $\Gamma \vdash_{1fcb} \bot$, et (3) $\forall(\Gamma) \models \bot$ sont équivalentes.

Démonstration.

- (1) implique (2) car la résolution au 1^o ordre est une combinaison de factorisation, copie et résolution binaire.
- (2) implique (3) car la factorisation, la copie et la résolution binaire sont cohérentes.
- Prouvons (3) implique (1). Supposons que $\forall(\Gamma) \models \bot$, 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, nous avons : $\Delta \vdash_p \bot$. D'après le corollaire au relèvement 5.4.22, il existe D telle que $\Gamma \vdash_{1r} D$ et $\bot$ est instance de D. Mais dans ce cas, nous avons $D = \bot$.

Plan

Aujourd'hui

- Unification
- Résolution au premier ordre
- Complétude de la résolution au premier ordre

Plan du Semestre

AUJOURD'HUI

- Logique propositionnelle
- Résolution propositionnelle
- Déduction naturelle propositionnelle

PARTIEL

- Logique du premier ordre
- Base de la démonstration automatique ("résolution au premier ordre") *
- Déduction naturelle au premier ordre

EXAMEN

Prochaine fois

Déduction naturelle au premier ordre

- Règles
- Exemples
- Tactiques

Conclusion

Merci de votre attention.

Questions ?