Cours 10

Calcul des prédicats : Formes normales

Établir la satisfaisabilité d'une formule du calcul des prédicats est un problème difficile. On peut alors essayer de travailler sur une version de forme plus simple mais la satisfaisabilité est équivalente à la formule initiale. Nous allons introduire `la forme clausale` *pour la logique des prédicats.*
Toute formule de la logique des prédicats du premier ordre admet une représentation sous forme de clause qui préserve sa satisfaisabilité. À la différence des formes normales, une clause n'est pas logiquement équivalente à la formule dont elle dérive. Nous décrivons, à présent, les différentes étapes qui mènent à une représentation sous forme clausale.

Rappelons les équivalences remarquables qui nous seront utiles dans la transformation.

- **Loi de conversion des quantificateurs :**

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

- **Loi de distribution des quantificateurs :**

  • $\forall x(A \wedge B) \equiv \forall x A \wedge \forall x B$ • $\exists x(A \vee B) \equiv \exists x A \vee \exists x B$

- **Loi de permutation des quantificateurs de même type :**

  • $\forall x \forall y A \equiv \forall y \forall x A$ • $\exists x \exists y A \equiv \exists y \exists x A$

- **Lois de renommage des variables(changement de variables liées).** On peut toujours renommer une variable liée et la variable du quantificateur au sein d'une formule. Cependant, le nouveau nom ne doit pas être un nom déjà utilisé pour une variable libre ou liée de la formule.
- **Lois de passage** : si x ne figure pas à titre d'occurrence libre dans A, on a les lois suivantes :

• $\forall x(A \wedge B) \equiv A \wedge \forall x B$	• $\exists x(A \wedge B) \equiv A \wedge \exists x B$
• $\forall x(A \vee B) \equiv A \vee \forall x B$	• $\exists x(A \vee B) \equiv A \vee \exists x B$
• $\forall x(A \Rightarrow B) \equiv A \Rightarrow \forall x B$	• $\exists x(A \Rightarrow B) \equiv A \Rightarrow \exists x B$
• $\boxed{\forall x(B \Rightarrow A) \equiv \exists x B \Rightarrow A}$	• $\boxed{\exists x(B \Rightarrow A) \equiv \forall x B \Rightarrow A}$

10.1 Formes prénexes

(Forme prénexe)

Définition 10.1. *. Une formule est sous forme prénexe lorsqu'elle a la forme :*

$$Q_1 x_1 Q_2 x_2 \ldots Q_n x_n G$$

où chaque Q_i est un quantificateur (existentiel ou universel), la partie $Q_1 x_1 Q_2 x_2 \cdots Q_n x_n$ étant appelée préfixe et G , étant qualifiée de matrice, ne contient aucun quantificateur.

Mettre une formule sous forme prénexe est : renvoyer tous les quantificateurs au début de la formule.

Théorème 10.1. *(Forme prénexe équivalente). Pour toute formule du calcul des prédicats il existe une formule équivalente en forme prénexe.*

L'algorithme de mise sous forme prénexe suit les étapes suivantes :
Etape 1 : Renommer les variables de façon à ce qu'aucune variable n'ait d'occurrence libre et liée et d'occurrences liées à des quantificateurs différents ;
Etape 2 : Appliquer les équivalences rappelés en haut.

Exemple 10.1. *Considérez la formule suivante :*

$$\neg\exists x \forall y R(x,y) \wedge \forall x(\exists y R(x,y) \Rightarrow R(x,x))$$

La première étape, renommage des variables, donne la formule suivante :

$$\neg\exists x \forall y R(x,y) \wedge \forall z(\exists w R(z,w) \Rightarrow R(z,z))$$

Nous appliquons ensuite la deuxième étape

$$\begin{aligned}
\neg\exists x\forall y R(x,y) \wedge & \forall z(\exists w R(z,w) \Rightarrow R(z,z)) \\
& \equiv \forall x\exists y\neg R(x,y) \wedge \forall z(\exists w R(z,w) \Rightarrow R(z,z)) \\
& \equiv \forall x(\exists y\neg R(x,y) \wedge \forall z(\exists w R(z,w) \Rightarrow R(z,z))) \\
& \equiv \forall x(\exists y(\neg R(x,y) \wedge \forall z(\exists w R(z,w) \Rightarrow R(z,z)))) \\
& \equiv \forall x(\exists y\forall z((\neg R(x,y) \wedge (\exists w R(z,w) \Rightarrow R(z,z))))) \\
& \equiv \forall x\exists y\forall z((\neg R(x,y) \wedge (\forall w(R(z,w) \Rightarrow R(z,z))))) \\
& \equiv \forall x\exists y\forall z\forall w \quad (\neg R(x,y) \wedge (R(z,w) \Rightarrow R(z,z)))
\end{aligned}$$

La nécessite de renommer les variables peut se comprendre si on essaie d'appliquer les règles de passage sans un renommage préalable. Ainsi :

$$\begin{aligned}
\neg\exists x\forall y R(x,y) \wedge & \forall x(\exists y R(x,y) \Rightarrow R(x,x)) \\
& \equiv \forall x\exists y\neg R(x,y) \wedge \forall x(\exists y R(x,y) \Rightarrow R(x,x)) \\
& \equiv \forall x\Big(\exists y\neg R(x,y) \wedge (\exists y R(x,y) \Rightarrow R(x,x))\Big) \\
& \equiv \forall x\Big(\exists y\neg R(x,y) \wedge \forall y(R(x,y) \Rightarrow R(x,x))\Big)
\end{aligned}$$

Maintenant, on doit obligatoirement faire un renommage, sinon on ne peut pas appliquer les lois de passages. Après renommage on obtient la formule équivalente suivante :

$$\equiv \forall x\Big(\exists y\neg R(x,y) \wedge \forall w(R(x,w) \Rightarrow R(x,x))\Big)$$

Là, on a le choix entre la remplacer par :

$$\equiv \forall x\forall w\exists y\Big(\neg R(x,y) \wedge (R(x,w) \Rightarrow R(x,x))\Big)$$

ou

$$\equiv \forall x\exists y\forall w\Big(\neg R(x,y) \wedge (R(x,w) \Rightarrow R(x,x))\Big)$$

On remarque la stratégie d'application des règles de transformations en formes prénexes peut influer sur le nombre de variables de la formule finale.

Nous allons voir dans la section suivante, que le dernier choix est plus avantageux. Mais si on veut être prudent, on commence par rendre la formule propre en faisant les renommage nécessaires, après on applique les lois de passages.

Attention. En général on ne peut pas intervertir des quantificateurs différents, mais dans des cas particuliers :
par exemple $\exists x\forall y(P(x) \wedge Q(y)) \equiv \forall y\exists x(P(x) \wedge Q(y))$.

10.2 Forme de Skolem

Une formule est sous forme de Skolem lorsqu'elle est sous forme prénexe et qu'elle ne contient que des quantifications universelles. Pour effectuer une skolémisation, on part donc d'une formule sous forme prénexe et on ń supprime ż les quantificateurs existentiels, en appliquant de façon itérée la règle suivante.

Règle de Skolémisation

$$\frac{\exists x A}{A < x : f(z_1, \ldots, z_m) >}$$

où

- $z_1, \ldots, z_m$ sont les variables libres de la formule $\exists x A$.
- f un *nouveau* symbole de fonction (dit de Skolem) d'arité m.

La skolémisation sert donc 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.

Par suite A a un modèle si et seulement si B a un modèle : la skolémisation préserve `l'existence d'un modèle`, nous disons aussi qu'elle `préserve la satisfaisabilité.`

Regardons ce qu'il en est sur deux exemples simples.

Exemple 10.2. *La formule $\exists x P(x)$ est skolémisée en $P(a)$. Nous observons les relations 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 $P(a)$.

Exemple 10.3. *La formule $\forall x \exists y Q(x, y)$ est skolémisée en $\forall x Q(x, f(x))$. Nous observons 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* $\forall x Q(x, f(x))$.
   En effet soient I un modèle de $\forall x Q(x, f(x))$ et D le domaine de I. Pour tout $d \in D$, l'ensemble $\{e \in D | (d, e) \in Q_I\}$ n'est pas vide, donc il existe $g : D \longrightarrow D$ une fonction telle que pour tout $d \in D, g(d) \in \{e \in D | (d, e) \in Q_I\}$. Soit J l'interprétation J telle que $Q_J = Q_I$ et $f_J = g$. : J est modèle de $\forall x Q(x, f(x))$.

Exemple 10.4.

1. *Soit $F = \forall x \exists y P(x, y) \Rightarrow \forall x \exists y P(y, x)$.*
   La mise sous forme prénexe donne la formule équivalente $F' = \exists x \forall y \forall z \exists w (P(x, y) \Rightarrow P(w, z))$.
   La skolémisation donne :
   $F_{s1} = \forall y \forall z \exists w (P(a, y) \Rightarrow P(w, z))$.
   $F_{s2} = \forall y \forall z (P(a, y) \Rightarrow P(f(y, z), z))$.

2. *Considérons la formule $\exists x_1 \forall x_2 \forall x_3 \exists x_4 \forall x_5 \exists x_6 P(x_1, x_2, x_3, x_4, x_5, x_6)$. On obtient sa formule de Skolem de la façon suivante :*

$$\forall x_2 \forall x_3 \exists x_4 \forall x_5 \exists x_6 P(a, x_2, x_3, x_4, x_5, x_6).$$
$$\forall x_2 \forall x_3 \forall x_5 \exists x_6 P(a, x_2, x_3, f(x_2, x_3), x_5, x_6).$$
$$\forall x_2 \forall x_3 \forall x_5 P(a, x_2, x_3, f(x_2, x_3), x_5, g(x_2, x_3, x_5)).$$

La version skolémisée d'une formule ne lui est pas en général équivalente (le langage étant étendu), néanmoins :

- tout modèle de la formule skolémisée est modèle de la formule initiale.
- tout modèle de la formule initiale peut s'étendre en un modèle de la formule skolémisée, obtenu en conservant les interprétations des symboles de la signature initiale, et en interprétant correctement les nouveaux symboles de fonction introduits pas la skolémisation.
- une formule close et sa forme de Skolem sont dites équisatisfaisables : si l'une possède un modèle, l'autre également et réciproquement.

Equisatisfaisabilité

Proposition 10.1. *Si F_s est obtenue par skolémisation à partir de F alors F_s est satisfaisable si et seulement si D est satisfaisable.*

10.3 Forme clausale

Après la skolémisation des formules et avant de pouvoir appliquer une généralisation au premier ordre de la méthode de résolution il faut transformer les formes de Skolem en clauses comme nous l'avons fait en logique propositionnelle. Nous étendons donc les notions essentielles à cette transformation avant de donner les règles permettant cette transformation.

Définition 10.2. *(Forme clausale). Une formule close est sous forme clausale si elle est*

1. *en forme prénexe,*

2. *elle est universelle (tous ses quantificateurs sont universels), et*
3. *sa matrice est sous forme normale conjonctive.*

Définition 10.3.

- *Un littéral est une formule atomique ou négation d'une formule atomique.*
- *Soit F une formule avec $\ell(F) = \{x_1, ..., x_n\}$. Sa fermeture universelle est la formule*

$$\forall x_1 \ldots \forall x_n F.$$

- *Une clause (de premier ordre) est la fermeture universelle d'une disjonction de littéraux.*

Lorsqu'une formule est sous forme clausale, on peut ensuite la décomposer en appliquant la règle $\forall x(A \wedge B) \equiv (\forall x A) \wedge (\forall x B)$. On obtient ainsi un ensemble de clauses (de premier ordre) equisatisfaisable à la formule donnée.

Théorème 10.2. *(Satisfaisabilité dun ensemble de clauses). Soit S un ensemble de clauses résultant de la mise sous forme clausale d'une formule. Alors F est insatisfaisable si et seulement si S est insatisfaisable.*

Ce théorème forme la base de nombreux démonstrateurs automatiques utilisant une représentation des formules sous forme de clauses. Il établit que la recherche de linsatisfaisabilité dune formule F est équivalente à la recherche dinsatisfaisabilité de sa représentation sous forme clausale S. Cependant F et S ne sont pas logiquement équivalentes : seule la satisfaisabilité est préservée.

Exemple 10.5. *Soit*

$$F_0 := \exists x \forall y \Big(\big(P(x,y) \Rightarrow \forall x \exists y P(y,x)\big) \wedge Q(x) \Big).$$

La mise sous forme prénexe donne la formule

$$F_1 := \exists x \forall y \forall z \exists w ((P(x,y)) \Rightarrow P(w,z)) \wedge Q(x)).$$

Par skolémisation, nous obtenons

$$F_2 := \forall y \forall z ((P(c,y)) \Rightarrow P(f(y,z),z)) \wedge Q(c)).$$

Nous pouvons ensuite mettre la matrice sous forme normale conjonctive :

$$F_3 := \forall y \forall z ((\neg P(c,y) \vee P(f(y,z),z)) \wedge Q(c)),$$

et obtenir ainsi lensemble de clauses :

$$S := \{\forall y \forall z (\neg P(c,y) \vee P(f(y,z),z)), \forall y \forall z Q(c)\}.$$

Cet ensemble est equisatisfaisable avec la formule F_0.

Souvent on écrit S comme

$$S' := \{(\neg P(c,y) \vee P(f(y,z),z)), Q(c)\}.$$

F_0 est satisfaisable ssi la fermeture universelle de S' est satisfaisable.

Exemple 10.6. *Soit $F = \exists y \forall z (P(z,y) \Leftrightarrow \neg \exists x (P(z,x) \wedge P(x,z)))$. Calculons la forme clausale de F avec une méthode différente.*

1. *Mettre F sous forme normale (éliminer les équivalences, les implications, déplacer les négations)*

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

2. *Rendre la formule propre. (Faire des renommages)*

$$\exists y \forall z \Big((\neg P(z,y) \vee \forall x (\neg P(z,x) \vee \neg P(x,z))) \wedge (\exists u (P(z,u) \wedge P(u,z)) \vee P(z,y)) \Big)$$

3. *Supprimer les quantificateurs existentiels*

$$\forall z \Big((\neg P(z,a) \vee \forall x (\neg P(z,x) \vee \neg P(x,z))) \wedge ((P(z,f(z)) \wedge P(f(z),z)) \vee P(z,a)) \Big)$$

4. *Supprimer les quantificateurs universels, nous obtenons la forme Skolem de F.*

$$\Big((\neg P(z,a) \vee (\neg P(z,x) \vee \neg P(x,z))) \wedge ((P(z,f(z)) \wedge P(f(z),z)) \vee P(z,a)) \Big)$$

5. *Transformons en produits de sommes de littéraux, on obtient la forme clausale de F, qui est l'ensemble des clauses :*
   - $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
   - $C_2 = P(z,f(z)) \vee P(z,a)$
   - $C_3 = P(f(z),z) \vee P(z,a)$

Severine Fratani. *Cours Logique et Calculabilité. Université Aix Marseille.*
Michel Levy et al. *Logique et démonstration automatique - Introduction à la logique propositionnelle et à la logique du premier ordre.*