

Manipulations syntaxiques

David Delahaye

Faculté des Sciences
David.Delahaye@lirmm.fr

Licence L3 2019-2020

Formes prénexes

Définition

- Une formule est dite en forme prénexe si et seulement si elle est de la forme $Q_1x_1 \dots Q_nx_n.\Phi$, où Q_1, \dots, Q_n sont des quantificateurs et Φ est une formule sans quantificateur.

Théorème

- Pour toute formule Φ , il existe une formule en forme prénexe Φ' , calculable à partir de Φ , telle que Φ et Φ' sont (sémantiquement, resp. prouvablement) équivalentes.

Formes prénexes

Règles de transformation (pour \forall)

Forme initiale	Forme prénexe
$\neg \forall x. \Phi$	$\exists x. \neg \Phi$
$(\forall x. \Phi) \wedge \Phi'$	$\forall x. \Phi \wedge \Phi'$
$(\forall x. \Phi) \vee \Phi'$	$\forall x. \Phi \vee \Phi'$
$(\forall x. \Phi) \Rightarrow \Phi'$	$\exists x. \Phi \Rightarrow \Phi'$
$\Phi' \wedge (\forall x. \Phi)$	$\forall x. \Phi' \wedge \Phi$
$\Phi' \vee (\forall x. \Phi)$	$\forall x. \Phi' \vee \Phi$
$\Phi' \Rightarrow (\forall x. \Phi)$	$\forall x. \Phi' \Rightarrow \Phi$

Formes prénexes

Règles de transformation (pour \exists)

Forme initiale	Forme prénexe
$\neg \exists x. \Phi$	$\forall x. \neg \Phi$
$(\exists x. \Phi) \wedge \Phi'$	$\exists x. \Phi \wedge \Phi'$
$(\exists x. \Phi) \vee \Phi'$	$\exists x. \Phi \vee \Phi'$
$(\exists x. \Phi) \Rightarrow \Phi'$	$\forall x. \Phi \Rightarrow \Phi'$
$\Phi' \wedge (\exists x. \Phi)$	$\exists x. \Phi' \wedge \Phi$
$\Phi' \vee (\exists x. \Phi)$	$\exists x. \Phi' \vee \Phi$
$\Phi' \Rightarrow (\exists x. \Phi)$	$\exists x. \Phi' \Rightarrow \Phi$

Règles de transformation

- x ne doit pas être libre dans Φ' ;
- C'est le cas si la formule initiale est polie (propre) ;
- Cette transformation respecte les connecteurs de la formule initiale, sauf pour \Leftrightarrow , où il faut transformer $\Phi \Leftrightarrow \Phi'$ en $(\Phi \Rightarrow \Phi') \wedge (\Phi' \Rightarrow \Phi)$.

Justification des règles de transformation

- On peut les justifier syntaxiquement en utilisant d'autres règles de transformation (du tableau ou autre) :

$$(\forall x. \Phi) \Rightarrow \Phi' \rightsquigarrow \neg(\forall x. \Phi) \vee \Phi' \rightsquigarrow (\exists x. \neg \Phi) \vee \Phi' \rightsquigarrow \exists x. \neg \Phi \vee \Phi' \rightsquigarrow \exists x. \Phi \Rightarrow \Phi' ;$$

- On peut aussi les justifier sémantiquement :

$$\begin{aligned} \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I &= \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket_\rho^I = (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket_{\rho[v/x]}^I) \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket_\rho^I = \\ &= \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket_{\rho[v/x]}^I) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket_\rho^I = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} \llbracket \Phi \rrbracket_{\rho[v/x]}^I) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket_\rho^I = \\ &= \bigvee_{v \in D_I} (\neg_{\mathcal{B}} \llbracket \Phi \rrbracket_{\rho[v/x]}^I \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket_{\rho[v/x]}^I) = \bigvee_{v \in D_I} (\llbracket \Phi \rrbracket_{\rho[v/x]}^I \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket_{\rho[v/x]}^I) = \\ &= \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket_\rho^I = \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I. \end{aligned}$$

Justification des règles de transformation

- On peut les justifier syntaxiquement en utilisant d'autres règles de transformation (du tableau ou autre) :

$$(\forall x. \Phi) \Rightarrow \Phi' \rightsquigarrow \neg(\forall x. \Phi) \vee \Phi' \rightsquigarrow (\exists x. \neg \Phi) \vee \Phi' \rightsquigarrow \exists x. \neg \Phi \vee \Phi' \rightsquigarrow \exists x. \Phi \Rightarrow \Phi' ;$$

- On peut aussi les justifier sémantiquement :

$$\begin{aligned} \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I &= \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I_\rho = (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \neg_{\mathcal{B}} (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \bigvee_{v \in D_I} (\neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]} \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \bigvee_{v \in D_I} (\llbracket \Phi \rrbracket^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \\ &= \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I_\rho = \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I. \end{aligned}$$

Justification des règles de transformation

- On peut les justifier syntaxiquement en utilisant d'autres règles de transformation (du tableau ou autre) :

$$(\forall x. \Phi) \Rightarrow \Phi' \rightsquigarrow \neg(\forall x. \Phi) \vee \Phi' \rightsquigarrow (\exists x. \neg \Phi) \vee \Phi' \rightsquigarrow \exists x. \neg \Phi \vee \Phi' \rightsquigarrow \exists x. \Phi \Rightarrow \Phi' ;$$

- On peut aussi les justifier sémantiquement :

$$\begin{aligned} \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I &= \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I_\rho = (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \bigvee_{v \in D_I} (\neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]} \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \bigvee_{v \in D_I} (\llbracket \Phi \rrbracket^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \\ &= \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I_\rho = \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I. \end{aligned}$$

Justification des règles de transformation

- On peut les justifier syntaxiquement en utilisant d'autres règles de transformation (du tableau ou autre) :

$$(\forall x. \Phi) \Rightarrow \Phi' \rightsquigarrow \neg(\forall x. \Phi) \vee \Phi' \rightsquigarrow (\exists x. \neg \Phi) \vee \Phi' \rightsquigarrow \exists x. \neg \Phi \vee \Phi' \rightsquigarrow \exists x. \Phi \Rightarrow \Phi' ;$$

- On peut aussi les justifier sémantiquement :

$$\begin{aligned} \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I &= \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I_\rho = (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \bigvee_{v \in D_I} (\neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]} \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \bigvee_{v \in D_I} (\llbracket \Phi \rrbracket^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \\ &= \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I_\rho = \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I. \end{aligned}$$

Justification des règles de transformation

- On peut les justifier syntaxiquement en utilisant d'autres règles de transformation (du tableau ou autre) :

$$(\forall x. \Phi) \Rightarrow \Phi' \rightsquigarrow \neg(\forall x. \Phi) \vee \Phi' \rightsquigarrow (\exists x. \neg \Phi) \vee \Phi' \rightsquigarrow \exists x. \neg \Phi \vee \Phi' \rightsquigarrow \exists x. \Phi \Rightarrow \Phi' ;$$

- On peut aussi les justifier sémantiquement :

$$\begin{aligned} \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I &= \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I_\rho = (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \bigvee_{v \in D_I} (\neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]} \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \bigvee_{v \in D_I} (\llbracket \Phi \rrbracket^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \\ &= \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I_\rho = \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I. \end{aligned}$$

Justification des règles de transformation

- On peut les justifier syntaxiquement en utilisant d'autres règles de transformation (du tableau ou autre) :

$$(\forall x. \Phi) \Rightarrow \Phi' \rightsquigarrow \neg(\forall x. \Phi) \vee \Phi' \rightsquigarrow (\exists x. \neg \Phi) \vee \Phi' \rightsquigarrow \exists x. \neg \Phi \vee \Phi' \rightsquigarrow \exists x. \Phi \Rightarrow \Phi' ;$$

- On peut aussi les justifier sémantiquement :

$$\begin{aligned} \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I &= \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I_\rho = (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \bigvee_{v \in D_I} (\neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]} \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \bigvee_{v \in D_I} (\llbracket \Phi \rrbracket^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \\ &= \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I_\rho = \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I. \end{aligned}$$

Justification des règles de transformation

- On peut les justifier syntaxiquement en utilisant d'autres règles de transformation (du tableau ou autre) :

$$(\forall x. \Phi) \Rightarrow \Phi' \rightsquigarrow \neg(\forall x. \Phi) \vee \Phi' \rightsquigarrow (\exists x. \neg \Phi) \vee \Phi' \rightsquigarrow \exists x. \neg \Phi \vee \Phi' \rightsquigarrow \exists x. \Phi \Rightarrow \Phi' ;$$

- On peut aussi les justifier sémantiquement :

$$\begin{aligned} \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I &= \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I_\rho = (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \bigvee_{v \in D_I} (\neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]} \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \bigvee_{v \in D_I} (\llbracket \Phi \rrbracket^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \\ &= \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I_\rho = \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I. \end{aligned}$$

Justification des règles de transformation

- On peut les justifier syntaxiquement en utilisant d'autres règles de transformation (du tableau ou autre) :

$$(\forall x. \Phi) \Rightarrow \Phi' \rightsquigarrow \neg(\forall x. \Phi) \vee \Phi' \rightsquigarrow (\exists x. \neg \Phi) \vee \Phi' \rightsquigarrow \exists x. \neg \Phi \vee \Phi' \rightsquigarrow \exists x. \Phi \Rightarrow \Phi' ;$$

- On peut aussi les justifier sémantiquement :

$$\begin{aligned} \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket' &= \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket'_\rho = (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket'_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket'_\rho = \\ &= \neg_{\mathcal{B}} (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket'_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket'_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} \llbracket \Phi \rrbracket'_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket'_\rho = \\ &= \bigvee_{v \in D_I} (\neg_{\mathcal{B}} \llbracket \Phi \rrbracket'_{\rho[v/x]} \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket'_{\rho[v/x]}) = \bigvee_{v \in D_I} (\llbracket \Phi \rrbracket'_{\rho[v/x]} \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket'_{\rho[v/x]}) = \\ &= \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket'_\rho = \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket'. \end{aligned}$$

Justification des règles de transformation

- On peut les justifier syntaxiquement en utilisant d'autres règles de transformation (du tableau ou autre) :

$$(\forall x. \Phi) \Rightarrow \Phi' \rightsquigarrow \neg(\forall x. \Phi) \vee \Phi' \rightsquigarrow (\exists x. \neg \Phi) \vee \Phi' \rightsquigarrow \exists x. \neg \Phi \vee \Phi' \rightsquigarrow \exists x. \Phi \Rightarrow \Phi' ;$$

- On peut aussi les justifier sémantiquement :

$$\begin{aligned} \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I &= \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I_\rho = (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \bigvee_{v \in D_I} (\neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]} \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \bigvee_{v \in D_I} (\llbracket \Phi \rrbracket^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \\ &= \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I_\rho = \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I. \end{aligned}$$

Justification des règles de transformation

- On peut les justifier syntaxiquement en utilisant d'autres règles de transformation (du tableau ou autre) :

$$(\forall x. \Phi) \Rightarrow \Phi' \rightsquigarrow \neg(\forall x. \Phi) \vee \Phi' \rightsquigarrow (\exists x. \neg \Phi) \vee \Phi' \rightsquigarrow \exists x. \neg \Phi \vee \Phi' \rightsquigarrow \exists x. \Phi \Rightarrow \Phi' ;$$

- On peut aussi les justifier sémantiquement :

$$\begin{aligned} \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I &= \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I_\rho = (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \bigvee_{v \in D_I} (\neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]} \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \bigvee_{v \in D_I} (\llbracket \Phi \rrbracket^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \\ &= \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I_\rho = \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I. \end{aligned}$$

Justification des règles de transformation

- On peut les justifier syntaxiquement en utilisant d'autres règles de transformation (du tableau ou autre) :

$$(\forall x. \Phi) \Rightarrow \Phi' \rightsquigarrow \neg(\forall x. \Phi) \vee \Phi' \rightsquigarrow (\exists x. \neg \Phi) \vee \Phi' \rightsquigarrow \exists x. \neg \Phi \vee \Phi' \rightsquigarrow \exists x. \Phi \Rightarrow \Phi' ;$$

- On peut aussi les justifier sémantiquement :

$$\begin{aligned} \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I &= \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I_\rho = (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \bigvee_{v \in D_I} (\neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]} \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \bigvee_{v \in D_I} (\llbracket \Phi \rrbracket^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \\ &= \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I_\rho = \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I. \end{aligned}$$

Justification des règles de transformation

- On peut les justifier syntaxiquement en utilisant d'autres règles de transformation (du tableau ou autre) :

$$(\forall x. \Phi) \Rightarrow \Phi' \rightsquigarrow \neg(\forall x. \Phi) \vee \Phi' \rightsquigarrow (\exists x. \neg \Phi) \vee \Phi' \rightsquigarrow \exists x. \neg \Phi \vee \Phi' \rightsquigarrow \exists x. \Phi \Rightarrow \Phi' ;$$

- On peut aussi les justifier sémantiquement :

$$\begin{aligned} \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I &= \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I_\rho = (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \bigvee_{v \in D_I} (\neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]} \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \bigvee_{v \in D_I} (\llbracket \Phi \rrbracket^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \\ &= \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I_\rho = \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I. \end{aligned}$$

Justification des règles de transformation

- On peut les justifier syntaxiquement en utilisant d'autres règles de transformation (du tableau ou autre) :

$$(\forall x. \Phi) \Rightarrow \Phi' \rightsquigarrow \neg(\forall x. \Phi) \vee \Phi' \rightsquigarrow (\exists x. \neg \Phi) \vee \Phi' \rightsquigarrow \exists x. \neg \Phi \vee \Phi' \rightsquigarrow \exists x. \Phi \Rightarrow \Phi' ;$$

- On peut aussi les justifier sémantiquement :

$$\begin{aligned} \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I &= \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I_\rho = (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \bigvee_{v \in D_I} (\neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]} \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \bigvee_{v \in D_I} (\llbracket \Phi \rrbracket^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \\ &= \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I_\rho = \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I. \end{aligned}$$

Justification des règles de transformation

- On peut les justifier syntaxiquement en utilisant d'autres règles de transformation (du tableau ou autre) :

$$(\forall x. \Phi) \Rightarrow \Phi' \rightsquigarrow \neg(\forall x. \Phi) \vee \Phi' \rightsquigarrow (\exists x. \neg \Phi) \vee \Phi' \rightsquigarrow \exists x. \neg \Phi \vee \Phi' \rightsquigarrow \exists x. \Phi \Rightarrow \Phi' ;$$

- On peut aussi les justifier sémantiquement :

$$\begin{aligned} \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I_\rho &= \llbracket (\forall x. \Phi) \Rightarrow \Phi' \rrbracket^I_\rho = (\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]}) \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_\rho = \\ &= \bigvee_{v \in D_I} (\neg_{\mathcal{B}} \llbracket \Phi \rrbracket^I_{\rho[v/x]} \vee_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \bigvee_{v \in D_I} (\llbracket \Phi \rrbracket^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} \llbracket \Phi' \rrbracket^I_{\rho[v/x]}) = \\ &= \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I_\rho = \llbracket \exists x. \Phi \Rightarrow \Phi' \rrbracket^I_\rho. \end{aligned}$$

Formes prénexes

Exemple

- $(\forall x. P(x) \Rightarrow Q(x)) \Rightarrow (\exists y. P(y)) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. (P(x) \Rightarrow Q(x)) \Rightarrow (\exists y. P(y)) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. \forall y. (P(x) \Rightarrow Q(x)) \Rightarrow P(y) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. \forall y. \exists z. (P(x) \Rightarrow Q(x)) \Rightarrow P(y) \Rightarrow Q(z);$
- Ordre de remontée des quantificateurs importe peu : l'ordre utilisé ici est de la gauche vers la droite (permet de respecter l'ordre initial des variables quantifiées).

Formes prénexes

Exemple

- $(\forall x. P(x) \Rightarrow Q(x)) \Rightarrow (\exists y. P(y)) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. (P(x) \Rightarrow Q(x)) \Rightarrow (\exists y. P(y)) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. \forall y. (P(x) \Rightarrow Q(x)) \Rightarrow P(y) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. \forall y. \exists z. (P(x) \Rightarrow Q(x)) \Rightarrow P(y) \Rightarrow Q(z);$
- Ordre de remontée des quantificateurs importe peu : l'ordre utilisé ici est de la gauche vers la droite (permet de respecter l'ordre initial des variables quantifiées).

Exemple

- $(\forall x. P(x) \Rightarrow Q(x)) \Rightarrow (\exists y. P(y)) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. (P(x) \Rightarrow Q(x)) \Rightarrow (\exists y. P(y)) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. \forall y. (P(x) \Rightarrow Q(x)) \Rightarrow P(y) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. \forall y. \exists z. (P(x) \Rightarrow Q(x)) \Rightarrow P(y) \Rightarrow Q(z) ;$
- Ordre de remontée des quantificateurs importe peu : l'ordre utilisé ici est de la gauche vers la droite (permet de respecter l'ordre initial des variables quantifiées).

Exemple

- $(\forall x. P(x) \Rightarrow Q(x)) \Rightarrow (\exists y. P(y)) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. (P(x) \Rightarrow Q(x)) \Rightarrow (\exists y. P(y)) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. \forall y. (P(x) \Rightarrow Q(x)) \Rightarrow P(y) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. \forall y. \exists z. (P(x) \Rightarrow Q(x)) \Rightarrow P(y) \Rightarrow Q(z);$
- Ordre de remontée des quantificateurs importe peu : l'ordre utilisé ici est de la gauche vers la droite (permet de respecter l'ordre initial des variables quantifiées).

Exemple

- $(\forall x. P(x) \Rightarrow Q(x)) \Rightarrow (\exists y. P(y)) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. (P(x) \Rightarrow Q(x)) \Rightarrow (\exists y. P(y)) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. \forall y. (P(x) \Rightarrow Q(x)) \Rightarrow P(y) \Rightarrow \exists z. Q(z) \rightsquigarrow$
 $\exists x. \forall y. \exists z. (P(x) \Rightarrow Q(x)) \Rightarrow P(y) \Rightarrow Q(z);$
- Ordre de remontée des quantificateurs importe peu : l'ordre utilisé ici est de la gauche vers la droite (permet de respecter l'ordre initial des variables quantifiées).

Exemple

- Les règles de transformation reposent sur la logique classique ;
- Une formule qui admet une preuve en logique intuitionniste peut ne plus en admettre après transformation (et n'admettre qu'une preuve en logique classique) ;
- C'est le cas de l'exemple donné précédemment.

Formule existentielle/universelle

- Une formule existentielle est une formule de la forme $\exists x_1 \dots \exists x_n. \Phi$, où Φ est une formule sans quantificateur ;
- Une formule universelle est une formule de la forme $\forall x_1 \dots \forall x_n. \Phi$, où Φ est une formule sans quantificateur.

Théorème de Herbrand-Skolem

Pour toute formule Φ :

- Il existe une formule existentielle Φ' telle que Φ' est valide si et seulement si Φ est valide (Φ' est une forme de Herbrand de Φ) ;
- Il existe une formule universelle Φ' telle que Φ' est insatisfiable si et seulement si Φ est insatisfiable (Φ' est une forme de Skolem de Φ).

Intuition de la skolémisation

- Si on a une formule $\forall x. \exists y. P(x, y)$ vraie dans une certaine interprétation, on peut se dire que pour chaque x , il existe (au moins) un y tel que $P(x, y)$ soit vrai ;
- On peut donc supposer qu'il y a une fonction $f(x)$ qui nous fournit un « bon » y pour chaque x ;
- Ceci nous permet de réécrire la formule comme $\forall x. P(x, f(x))$, où f est un nouveau symbole de fonction (le langage est étendu).

Intuition de la skolémisation

- Si on a une formule $\forall x. \exists y. P(x, y)$ vraie dans une certaine interprétation, on peut se dire que pour chaque x , il existe (au moins) un y tel que $P(x, y)$ soit vrai ;
- On peut donc supposer qu'il y a une fonction $f(x)$ qui nous fournit un « bon » y pour chaque x ;
- Ceci nous permet de réécrire la formule comme $\forall x. P(x, f(x))$, où f est un nouveau symbole de fonction (le langage est étendu).

Intuition de la skolémisation

- Si on a une formule $\forall x. \exists y. P(x, y)$ vraie dans une certaine interprétation, on peut se dire que pour chaque x , il existe (au moins) un y tel que $P(x, y)$ soit vrai ;
- On peut donc supposer qu'il y a une fonction $f(x)$ qui nous fournit un « bon » y pour chaque x ;
- Ceci nous permet de réécrire la formule comme $\forall x. P(x, f(x))$, où f est un nouveau symbole de fonction (le langage est étendu).

Intuition de la skolémisation

- Il est plus simple de décrire la skolémisation pour une formule sous forme prénexe ;
- Chaque occurrence d'une variable y quantifiée existentiellement est remplacée par un terme $f(x_1, \dots, x_n)$ où x_1, \dots, x_n sont les variables quantifiées universellement dont les quantificateurs apparaissent avant celui de y .
- S'il n'y a aucun quantificateur universel avant celui de y , f sera une fonction d'arité 0, c'est-à-dire une constante ;
- Pour chaque variable à skolémiser, il faut choisir un nouveau nom de fonction qui n'apparaît pas déjà dans la formule : ce symbole est appelé symbole de Skolem.

Skolémisation et herbrandisation

Intuition de la skolémisation

- Il est plus simple de décrire la skolémisation pour une formule sous forme prénexe ;
- Chaque occurrence d'une variable y quantifiée existentiellement est remplacée par un terme $f(x_1, \dots, x_n)$ où x_1, \dots, x_n sont les variables quantifiées universellement dont les quantificateurs apparaissent avant celui de y .
- S'il n'y a aucun quantificateur universel avant celui de y , f sera une fonction d'arité 0, c'est-à-dire une constante ;
- Pour chaque variable à skolémiser, il faut choisir un nouveau nom de fonction qui n'apparaît pas déjà dans la formule : ce symbole est appelé symbole de Skolem.

Intuition de la skolémisation

- Il est plus simple de décrire la skolémisation pour une formule sous forme prénexe ;
- Chaque occurrence d'une variable y quantifiée existentiellement est remplacée par un terme $f(x_1, \dots, x_n)$ où x_1, \dots, x_n sont les variables quantifiées universellement dont les quantificateurs apparaissent avant celui de y .
- S'il n'y a aucun quantificateur universel avant celui de y , f sera une fonction d'arité 0, c'est-à-dire une constante ;
- Pour chaque variable à skolémiser, il faut choisir un nouveau nom de fonction qui n'apparaît pas déjà dans la formule : ce symbole est appelé symbole de Skolem.

Intuition de la skolémisation

- Il est plus simple de décrire la skolémisation pour une formule sous forme prénexe ;
- Chaque occurrence d'une variable y quantifiée existentiellement est remplacée par un terme $f(x_1, \dots, x_n)$ où x_1, \dots, x_n sont les variables quantifiées universellement dont les quantificateurs apparaissent avant celui de y .
- S'il n'y a aucun quantificateur universel avant celui de y , f sera une fonction d'arité 0, c'est-à-dire une constante ;
- Pour chaque variable à skolémiser, il faut choisir un nouveau nom de fonction qui n'apparaît pas déjà dans la formule : ce symbole est appelé symbole de Skolem.

Skolémisation et herbrandisation

Fonctions de skolémisation et herbrandisation

- Si Φ est atomique, $s(\Phi) = h(\Phi) = \Phi$;
- $s(\Phi \wedge \Phi') = s(\Phi) \wedge s(\Phi')$, $h(\Phi \wedge \Phi') = h(\Phi) \wedge h(\Phi')$;
- $s(\Phi \vee \Phi') = s(\Phi) \vee s(\Phi')$, $h(\Phi \vee \Phi') = h(\Phi) \vee h(\Phi')$;
- $s(\neg\Phi) = \neg h(\Phi)$, $h(\neg\Phi) = \neg s(\Phi)$;
- $s(\Phi \Rightarrow \Phi') = h(\Phi) \Rightarrow s(\Phi')$, $h(\Phi \Rightarrow \Phi') = s(\Phi) \Rightarrow h(\Phi')$;
- $s(\forall x.\Phi) = s(\Phi)$, $h(\forall x.\Phi) = h(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\forall x.\Phi)$;
- $s(\exists x.\Phi) = s(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\exists x.\Phi)$, $h(\exists x.\Phi) = h(\Phi)$.
- Ensuite, une fois le calcul terminé :
 - Skolémisation : $\forall x_1 \dots \forall x_n. s(\Phi)$, où $\{x_1, \dots, x_n\} = FV(s(\Phi))$;
 - Herbrandisation : $\exists x_1 \dots \forall x_n. h(\Phi)$, où $\{x_1, \dots, x_n\} = FV(h(\Phi))$.

Skolémisation et herbrandisation

Fonctions de skolémisation et herbrandisation

- Si Φ est atomique, $s(\Phi) = h(\Phi) = \Phi$;
- $s(\Phi \wedge \Phi') = s(\Phi) \wedge s(\Phi')$, $h(\Phi \wedge \Phi') = h(\Phi) \wedge h(\Phi')$;
- $s(\Phi \vee \Phi') = s(\Phi) \vee s(\Phi')$, $h(\Phi \vee \Phi') = h(\Phi) \vee h(\Phi')$;
- $s(\neg\Phi) = \neg h(\Phi)$, $h(\neg\Phi) = \neg s(\Phi)$;
- $s(\Phi \Rightarrow \Phi') = h(\Phi) \Rightarrow s(\Phi')$, $h(\Phi \Rightarrow \Phi') = s(\Phi) \Rightarrow h(\Phi')$;
- $s(\forall x.\Phi) = s(\Phi)$, $h(\forall x.\Phi) = h(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\forall x.\Phi)$;
- $s(\exists x.\Phi) = s(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\exists x.\Phi)$, $h(\exists x.\Phi) = h(\Phi)$.
- Ensuite, une fois le calcul terminé :
 - Skolémisation : $\forall x_1 \dots \forall x_n. s(\Phi)$, où $\{x_1, \dots, x_n\} = FV(s(\Phi))$;
 - Herbrandisation : $\exists x_1 \dots \forall x_n. h(\Phi)$, où $\{x_1, \dots, x_n\} = FV(h(\Phi))$.

Skolémisation et herbrandisation

Fonctions de skolémisation et herbrandisation

- Si Φ est atomique, $s(\Phi) = h(\Phi) = \Phi$;
- $s(\Phi \wedge \Phi') = s(\Phi) \wedge s(\Phi')$, $h(\Phi \wedge \Phi') = h(\Phi) \wedge h(\Phi')$;
- $s(\Phi \vee \Phi') = s(\Phi) \vee s(\Phi')$, $h(\Phi \vee \Phi') = h(\Phi) \vee h(\Phi')$;
- $s(\neg\Phi) = \neg h(\Phi)$, $h(\neg\Phi) = \neg s(\Phi)$;
- $s(\Phi \Rightarrow \Phi') = h(\Phi) \Rightarrow s(\Phi')$, $h(\Phi \Rightarrow \Phi') = s(\Phi) \Rightarrow h(\Phi')$;
- $s(\forall x.\Phi) = s(\Phi)$, $h(\forall x.\Phi) = h(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\forall x.\Phi)$;
- $s(\exists x.\Phi) = s(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\exists x.\Phi)$, $h(\exists x.\Phi) = h(\Phi)$.
- Ensuite, une fois le calcul terminé :
 - Skolémisation : $\forall x_1 \dots \forall x_n. s(\Phi)$, où $\{x_1, \dots, x_n\} = FV(s(\Phi))$;
 - Herbrandisation : $\exists x_1 \dots \forall x_n. h(\Phi)$, où $\{x_1, \dots, x_n\} = FV(h(\Phi))$.

Skolémisation et herbrandisation

Fonctions de skolémisation et herbrandisation

- Si Φ est atomique, $s(\Phi) = h(\Phi) = \Phi$;
- $s(\Phi \wedge \Phi') = s(\Phi) \wedge s(\Phi')$, $h(\Phi \wedge \Phi') = h(\Phi) \wedge h(\Phi')$;
- $s(\Phi \vee \Phi') = s(\Phi) \vee s(\Phi')$, $h(\Phi \vee \Phi') = h(\Phi) \vee h(\Phi')$;
- $s(\neg\Phi) = \neg h(\Phi)$, $h(\neg\Phi) = \neg s(\Phi)$;
- $s(\Phi \Rightarrow \Phi') = h(\Phi) \Rightarrow s(\Phi')$, $h(\Phi \Rightarrow \Phi') = s(\Phi) \Rightarrow h(\Phi')$;
- $s(\forall x.\Phi) = s(\Phi)$, $h(\forall x.\Phi) = h(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\forall x.\Phi)$;
- $s(\exists x.\Phi) = s(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\exists x.\Phi)$, $h(\exists x.\Phi) = h(\Phi)$.
- Ensuite, une fois le calcul terminé :
 - Skolémisation : $\forall x_1 \dots \forall x_n. s(\Phi)$, où $\{x_1, \dots, x_n\} = FV(s(\Phi))$;
 - Herbrandisation : $\exists x_1 \dots \forall x_n. h(\Phi)$, où $\{x_1, \dots, x_n\} = FV(h(\Phi))$.

Skolémisation et herbrandisation

Fonctions de skolémisation et herbrandisation

- Si Φ est atomique, $s(\Phi) = h(\Phi) = \Phi$;
- $s(\Phi \wedge \Phi') = s(\Phi) \wedge s(\Phi')$, $h(\Phi \wedge \Phi') = h(\Phi) \wedge h(\Phi')$;
- $s(\Phi \vee \Phi') = s(\Phi) \vee s(\Phi')$, $h(\Phi \vee \Phi') = h(\Phi) \vee h(\Phi')$;
- $s(\neg\Phi) = \neg h(\Phi)$, $h(\neg\Phi) = \neg s(\Phi)$;
- $s(\Phi \Rightarrow \Phi') = h(\Phi) \Rightarrow s(\Phi')$, $h(\Phi \Rightarrow \Phi') = s(\Phi) \Rightarrow h(\Phi')$;
- $s(\forall x.\Phi) = s(\Phi)$, $h(\forall x.\Phi) = h(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\forall x.\Phi)$;
- $s(\exists x.\Phi) = s(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\exists x.\Phi)$, $h(\exists x.\Phi) = h(\Phi)$.
- Ensuite, une fois le calcul terminé :
 - Skolémisation : $\forall x_1 \dots \forall x_n. s(\Phi)$, où $\{x_1, \dots, x_n\} = FV(s(\Phi))$;
 - Herbrandisation : $\exists x_1 \dots \forall x_n. h(\Phi)$, où $\{x_1, \dots, x_n\} = FV(h(\Phi))$.

Skolémisation et herbrandisation

Fonctions de skolémisation et herbrandisation

- Si Φ est atomique, $s(\Phi) = h(\Phi) = \Phi$;
- $s(\Phi \wedge \Phi') = s(\Phi) \wedge s(\Phi')$, $h(\Phi \wedge \Phi') = h(\Phi) \wedge h(\Phi')$;
- $s(\Phi \vee \Phi') = s(\Phi) \vee s(\Phi')$, $h(\Phi \vee \Phi') = h(\Phi) \vee h(\Phi')$;
- $s(\neg\Phi) = \neg h(\Phi)$, $h(\neg\Phi) = \neg s(\Phi)$;
- $s(\Phi \Rightarrow \Phi') = h(\Phi) \Rightarrow s(\Phi')$, $h(\Phi \Rightarrow \Phi') = s(\Phi) \Rightarrow h(\Phi')$;
- $s(\forall x.\Phi) = s(\Phi)$, $h(\forall x.\Phi) = h(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\forall x.\Phi)$;
- $s(\exists x.\Phi) = s(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\exists x.\Phi)$, $h(\exists x.\Phi) = h(\Phi)$.
- Ensuite, une fois le calcul terminé :
 - Skolémisation : $\forall x_1 \dots \forall x_n. s(\Phi)$, où $\{x_1, \dots, x_n\} = FV(s(\Phi))$;
 - Herbrandisation : $\exists x_1 \dots \forall x_n. h(\Phi)$, où $\{x_1, \dots, x_n\} = FV(h(\Phi))$.

Fonctions de skolémisation et herbrandisation

- Si Φ est atomique, $s(\Phi) = h(\Phi) = \Phi$;
- $s(\Phi \wedge \Phi') = s(\Phi) \wedge s(\Phi')$, $h(\Phi \wedge \Phi') = h(\Phi) \wedge h(\Phi')$;
- $s(\Phi \vee \Phi') = s(\Phi) \vee s(\Phi')$, $h(\Phi \vee \Phi') = h(\Phi) \vee h(\Phi')$;
- $s(\neg\Phi) = \neg h(\Phi)$, $h(\neg\Phi) = \neg s(\Phi)$;
- $s(\Phi \Rightarrow \Phi') = h(\Phi) \Rightarrow s(\Phi')$, $h(\Phi \Rightarrow \Phi') = s(\Phi) \Rightarrow h(\Phi')$;
- $s(\forall x.\Phi) = s(\Phi)$, $h(\forall x.\Phi) = h(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\forall x.\Phi)$;
- $s(\exists x.\Phi) = s(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\exists x.\Phi)$, $h(\exists x.\Phi) = h(\Phi)$.
- Ensuite, une fois le calcul terminé :
 - Skolémisation : $\forall x_1 \dots \forall x_n. s(\Phi)$, où $\{x_1, \dots, x_n\} = FV(s(\Phi))$;
 - Herbrandisation : $\exists x_1 \dots \forall x_n. h(\Phi)$, où $\{x_1, \dots, x_n\} = FV(h(\Phi))$.

Fonctions de skolémisation et herbrandisation

- Si Φ est atomique, $s(\Phi) = h(\Phi) = \Phi$;
- $s(\Phi \wedge \Phi') = s(\Phi) \wedge s(\Phi')$, $h(\Phi \wedge \Phi') = h(\Phi) \wedge h(\Phi')$;
- $s(\Phi \vee \Phi') = s(\Phi) \vee s(\Phi')$, $h(\Phi \vee \Phi') = h(\Phi) \vee h(\Phi')$;
- $s(\neg\Phi) = \neg h(\Phi)$, $h(\neg\Phi) = \neg s(\Phi)$;
- $s(\Phi \Rightarrow \Phi') = h(\Phi) \Rightarrow s(\Phi')$, $h(\Phi \Rightarrow \Phi') = s(\Phi) \Rightarrow h(\Phi')$;
- $s(\forall x.\Phi) = s(\Phi)$, $h(\forall x.\Phi) = h(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\forall x.\Phi)$;
- $s(\exists x.\Phi) = s(\Phi)[f(x_1, \dots, x_n)/x]$, où $\{x_1, \dots, x_n\} = FV(\exists x.\Phi)$, $h(\exists x.\Phi) = h(\Phi)$.
- Ensuite, une fois le calcul terminé :
 - ▶ Skolémisation : $\forall x_1 \dots \forall x_n. s(\Phi)$, où $\{x_1, \dots, x_n\} = FV(s(\Phi))$;
 - ▶ Herbrandisation : $\exists x_1 \dots \forall x_n. h(\Phi)$, où $\{x_1, \dots, x_n\} = FV(h(\Phi))$.

Skolémisation et herbrandisation

Fonctions de skolémisation et herbrandisation

- La formule de départ doit être polie (propre) ;
- La formule de départ n'est pas forcément en forme prénexe ;
- Après le calcul, on obtient une forme prénexe (tous les quantificateurs universels ou existentiels sont en tête), mais la formule obtenue *n'est pas* logiquement équivalente à la formule de départ (voir le théorème de Herbrand-Skolem).

Skolémisation et herbrandisation

Fonctions de skolémisation et herbrandisation

- La formule de départ doit être polie (propre) ;
- La formule de départ n'est pas forcément en forme prénexe ;
- Après le calcul, on obtient une forme prénexe (tous les quantificateurs universels ou existentiels sont en tête), mais la formule obtenue *n'est pas* logiquement équivalente à la formule de départ (voir le théorème de Herbrand-Skolem).

Fonctions de skolémisation et herbrandisation

- La formule de départ doit être polie (propre) ;
- La formule de départ n'est pas forcément en forme prénexe ;
- Après le calcul, on obtient une forme prénexe (tous les quantificateurs universels ou existentiels sont en tête), mais la formule obtenue *n'est pas* logiquement équivalente à la formule de départ (voir le théorème de Herbrand-Skolem).

Exemple

- Skolémisation de $\forall x. \exists y. \forall z. P(x, y, z)$;
- $s(\forall x. \exists y. \forall z. P(x, y, z)) =$
 $s(\exists y. \forall z. P(x, y, z)) =$
 $s(\forall z. P(x, y, z))[f(x)/y] =$
 $s(P(x, y, z))[f(x)/y] =$
 $P(x, y, z)[f(x)/y] =$
 $P(x, f(x), z)$;
- On obtient donc : $\forall x. \forall z. P(x, f(x), z)$.

Exemple

- Skolémisation de $\forall x. \exists y. \forall z. P(x, y, z)$;
- $s(\forall x. \exists y. \forall z. P(x, y, z)) =$
 $s(\exists y. \forall z. P(x, y, z)) =$
 $s(\forall z. P(x, y, z))[f(x)/y] =$
 $s(P(x, y, z))[f(x)/y] =$
 $P(x, y, z)[f(x)/y] =$
 $P(x, f(x), z)$;
- On obtient donc : $\forall x. \forall z. P(x, f(x), z)$.

Exemple

- Skolémisation de $\forall x. \exists y. \forall z. P(x, y, z)$;
- $s(\forall x. \exists y. \forall z. P(x, y, z)) =$
 $s(\exists y. \forall z. P(x, y, z)) =$
 $s(\forall z. P(x, y, z))[f(x)/y] =$
 $s(P(x, y, z))[f(x)/y] =$
 $P(x, y, z)[f(x)/y] =$
 $P(x, f(x), z)$;
- On obtient donc : $\forall x. \forall z. P(x, f(x), z)$.

Exemple

- Skolémisation de $\forall x. \exists y. \forall z. P(x, y, z)$;
- $s(\forall x. \exists y. \forall z. P(x, y, z)) =$
 $s(\exists y. \forall z. P(x, y, z)) =$
 $s(\forall z. P(x, y, z))[f(x)/y] =$
 $s(P(x, y, z))[f(x)/y] =$
 $P(x, y, z)[f(x)/y] =$
 $P(x, f(x), z)$;
- On obtient donc : $\forall x. \forall z. P(x, f(x), z)$.

Exemple

- Skolémisation de $\forall x. \exists y. \forall z. P(x, y, z)$;
- $s(\forall x. \exists y. \forall z. P(x, y, z)) =$
 $s(\exists y. \forall z. P(x, y, z)) =$
 $s(\forall z. P(x, y, z))[f(x)/y] =$
 $s(P(x, y, z))[f(x)/y] =$
 $P(x, y, z)[f(x)/y] =$
 $P(x, f(x), z)$;
- On obtient donc : $\forall x. \forall z. P(x, f(x), z)$.

Exemple

- Skolémisation de $\forall x. \exists y. \forall z. P(x, y, z)$;
- $s(\forall x. \exists y. \forall z. P(x, y, z)) =$
 $s(\exists y. \forall z. P(x, y, z)) =$
 $s(\forall z. P(x, y, z))[f(x)/y] =$
 $s(P(x, y, z))[f(x)/y] =$
 $P(x, y, z)[f(x)/y] =$
 $P(x, f(x), z)$;
- On obtient donc : $\forall x. \forall z. P(x, f(x), z)$.

Exemple

- Skolémisation de $\forall x.\exists y.\forall z.P(x, y, z)$;
- $s(\forall x.\exists y.\forall z.P(x, y, z)) =$
 $s(\exists y.\forall z.P(x, y, z)) =$
 $s(\forall z.P(x, y, z))[f(x)/y] =$
 $s(P(x, y, z))[f(x)/y] =$
 $P(x, y, z)[f(x)/y] =$
 $P(x, f(x), z)$;
- On obtient donc : $\forall x.\forall z.P(x, f(x), z)$.

Exemple

- Skolémisation de $\forall x. \exists y. \forall z. P(x, y, z)$;
- $s(\forall x. \exists y. \forall z. P(x, y, z)) =$
 $s(\exists y. \forall z. P(x, y, z)) =$
 $s(\forall z. P(x, y, z))[f(x)/y] =$
 $s(P(x, y, z))[f(x)/y] =$
 $P(x, y, z)[f(x)/y] =$
 $P(x, f(x), z)$;
- On obtient donc : $\forall x. \forall z. P(x, f(x), z)$.

Principe

- On skolemise : on obtient une formule universelle $\forall \vec{x}. \Phi$;
- On élimine les quantificateurs, puis on met Φ en forme cnf ;
- Les clauses seront les membres de la cnf (quantification implicite) ;
- Vocabulaire :
 - ▶ un atome est de la forme $P(t_1, \dots, t_n)$ où $P \in \mathcal{P}$ et $t_1, \dots, t_n \in \mathcal{T}$;
 - ▶ Un littéral est un atome ou la négation d'un atome ;
 - ▶ Une clause est une disjonction de littéraux.

Clausification

Mise en forme cnf

$$\neg\neg F \rightsquigarrow F \quad \neg\top \rightsquigarrow \perp \quad \neg\perp \rightsquigarrow \top$$

$$\neg(F_1 \wedge F_2) \rightsquigarrow \neg F_1 \vee \neg F_2 \quad \neg(F_1 \vee F_2) \rightsquigarrow \neg F_1 \wedge \neg F_2$$

$$F_1 \Rightarrow F_2 \rightsquigarrow \neg F_1 \vee F_2$$

$$F_1 \wedge \top \rightsquigarrow F_1 \quad \top \wedge F_1 \rightsquigarrow F_1 \quad F_1 \wedge \perp \rightsquigarrow \perp \quad \perp \wedge F_1 \rightsquigarrow \perp$$

$$F_1 \vee \top \rightsquigarrow \top \quad \top \vee F_1 \rightsquigarrow \top \quad F_1 \vee \perp \rightsquigarrow F_1 \quad \perp \vee F_1 \rightsquigarrow F_1$$

$$(F_1 \wedge F_2) \vee F_3 \rightsquigarrow (F_1 \vee F_3) \wedge (F_2 \vee F_3)$$

$$F_3 \vee (F_1 \wedge F_2) \rightsquigarrow (F_3 \vee F_1) \wedge (F_3 \vee F_2)$$

Clausification

Exemple

- Formule initiale : $\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))$;
- $s(\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(h((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(s(\forall x. P(x) \vee Q(x)) \Rightarrow h(P(a) \vee Q(a))) =$
 $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- Formule skolémisée : $\forall x. \neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x)) \vee P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x))) \wedge \neg P(a) \wedge \neg Q(a) =$
 $(P(x) \vee Q(x)) \wedge \neg P(a) \wedge \neg Q(a)$;
- Ensemble des clauses : $S = \{P(x) \vee Q(x), \neg P(a), \neg Q(a)\}$.

Clausification

Exemple

- Formule initiale : $\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))$;
- $s(\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(h((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(s(\forall x. P(x) \vee Q(x)) \Rightarrow h(P(a) \vee Q(a))) =$
 $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- Formule skolémisée : $\forall x. \neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x)) \vee P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x))) \wedge \neg P(a) \wedge \neg Q(a) =$
 $(P(x) \vee Q(x)) \wedge \neg P(a) \wedge \neg Q(a)$;
- Ensemble des clauses : $S = \{P(x) \vee Q(x), \neg P(a), \neg Q(a)\}$.

Clausification

Exemple

- Formule initiale : $\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))$;
- $s(\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(h((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(s(\forall x. P(x) \vee Q(x)) \Rightarrow h(P(a) \vee Q(a))) =$
 $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- Formule skolémisée : $\forall x. \neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x)) \vee P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x))) \wedge \neg P(a) \wedge \neg Q(a) =$
 $(P(x) \vee Q(x)) \wedge \neg P(a) \wedge \neg Q(a)$;
- Ensemble des clauses : $S = \{P(x) \vee Q(x), \neg P(a), \neg Q(a)\}$.

Clausification

Exemple

- Formule initiale : $\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))$;
- $s(\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(h((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(s(\forall x. P(x) \vee Q(x)) \Rightarrow h(P(a) \vee Q(a))) =$
 $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- Formule skolémisée : $\forall x. \neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x)) \vee P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x))) \wedge \neg P(a) \wedge \neg Q(a) =$
 $(P(x) \vee Q(x)) \wedge \neg P(a) \wedge \neg Q(a)$;
- Ensemble des clauses : $S = \{P(x) \vee Q(x), \neg P(a), \neg Q(a)\}$.

Clausification

Exemple

- Formule initiale : $\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))$;
- $s(\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(h((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(s(\forall x. P(x) \vee Q(x)) \Rightarrow h(P(a) \vee Q(a))) =$
 $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- Formule skolémisée : $\forall x. \neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x)) \vee P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x))) \wedge \neg P(a) \wedge \neg Q(a) =$
 $(P(x) \vee Q(x)) \wedge \neg P(a) \wedge \neg Q(a)$;
- Ensemble des clauses : $S = \{P(x) \vee Q(x), \neg P(a), \neg Q(a)\}$.

Exemple

- Formule initiale : $\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))$;
- $s(\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(h((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(s(\forall x. P(x) \vee Q(x)) \Rightarrow h(P(a) \vee Q(a))) =$
 $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- Formule skolémisée : $\forall x. \neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x)) \vee P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x))) \wedge \neg P(a) \wedge \neg Q(a) =$
 $(P(x) \vee Q(x)) \wedge \neg P(a) \wedge \neg Q(a)$;
- Ensemble des clauses : $S = \{P(x) \vee Q(x), \neg P(a), \neg Q(a)\}$.

Exemple

- Formule initiale : $\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))$;
- $s(\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(h((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(s(\forall x. P(x) \vee Q(x)) \Rightarrow h(P(a) \vee Q(a))) =$
 $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- Formule skolémisée : $\forall x. \neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x)) \vee P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x))) \wedge \neg P(a) \wedge \neg Q(a) =$
 $(P(x) \vee Q(x)) \wedge \neg P(a) \wedge \neg Q(a)$;
- Ensemble des clauses : $S = \{P(x) \vee Q(x), \neg P(a), \neg Q(a)\}$.

Exemple

- Formule initiale : $\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))$;
- $s(\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(h((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(s(\forall x. P(x) \vee Q(x)) \Rightarrow h(P(a) \vee Q(a))) =$
 $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- Formule skolémisée : $\forall x. \neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x)) \vee P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x))) \wedge \neg P(a) \wedge \neg Q(a) =$
 $(P(x) \vee Q(x)) \wedge \neg P(a) \wedge \neg Q(a)$;
- Ensemble des clauses : $S = \{P(x) \vee Q(x), \neg P(a), \neg Q(a)\}$.

Exemple

- Formule initiale : $\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))$;
- $s(\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(h((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(s(\forall x. P(x) \vee Q(x)) \Rightarrow h(P(a) \vee Q(a))) =$
 $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- Formule skolémisée : $\forall x. \neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x)) \vee P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x))) \wedge \neg P(a) \wedge \neg Q(a) =$
 $(P(x) \vee Q(x)) \wedge \neg P(a) \wedge \neg Q(a)$;
- Ensemble des clauses : $S = \{P(x) \vee Q(x), \neg P(a), \neg Q(a)\}$.

Exemple

- Formule initiale : $\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))$;
- $s(\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(h((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(s(\forall x. P(x) \vee Q(x)) \Rightarrow h(P(a) \vee Q(a))) =$
 $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- Formule skolémisée : $\forall x. \neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x)) \vee P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x))) \wedge \neg P(a) \wedge \neg Q(a) =$
 $(P(x) \vee Q(x)) \wedge \neg P(a) \wedge \neg Q(a)$;
- Ensemble des clauses : $S = \{P(x) \vee Q(x), \neg P(a), \neg Q(a)\}$.

Exemple

- Formule initiale : $\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))$;
- $s(\neg((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(h((\forall x. P(x) \vee Q(x)) \Rightarrow P(a) \vee Q(a))) =$
 $\neg(s(\forall x. P(x) \vee Q(x)) \Rightarrow h(P(a) \vee Q(a))) =$
 $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- Formule skolémisée : $\forall x. \neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a))$;
- $\neg(P(x) \vee Q(x) \Rightarrow P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x)) \vee P(a) \vee Q(a)) =$
 $\neg(\neg(P(x) \vee Q(x))) \wedge \neg P(a) \wedge \neg Q(a) =$
 $(P(x) \vee Q(x)) \wedge \neg P(a) \wedge \neg Q(a)$;
- Ensemble des clauses : $S = \{P(x) \vee Q(x), \neg P(a), \neg Q(a)\}$.

Définition

- Une substitution σ est une application de \mathbb{V} vers \mathcal{T} telle que l'ensemble $dom(\sigma) = \{x \in \mathbb{V} \mid \sigma(x) \neq x\}$, appelé le domaine de σ , est fini ;
- L'image de σ est par définition $ran(\sigma) = \{\sigma(x) \mid x \in dom(\sigma)\}$ et on pose $yield(\sigma) = \bigcup_{t \in ran(\sigma)} FV(t)$;
- Nous écrivons aussi σ sous la forme $[\sigma(x_1)/x_1, \dots, \sigma(x_n)/x_n]$, où x_1, \dots, x_n contiennent toutes les variables de $dom(\sigma)$ et sont distinctes deux à deux ;
- En particulier, $[]$ est la substitution vide (ou identité).

Substitution sur les termes

Définition

- La notion de substitution s'étend aux termes et se note $t\sigma$ (de manière postfixe), où t est un terme et σ une substitution ;
- Elle se définit par récurrence structurelle sur les termes :
 - ▶ Si $x \in \mathbb{V}$ alors $x\sigma = \sigma(x)$;
 - ▶ Si $f \in \mathcal{F}$ et $t_1, \dots, t_n \in \mathcal{T}$ alors $f(t_1, \dots, t_n)\sigma = f(t_1\sigma, \dots, t_n\sigma)$.

Exemples

- $f(x, g(y, z))[a/x, h(b)/y, c/z] = f(a, g(h(b), c))$;
- $f(x, g(y, z))[a/x, h(b)/y] = f(a, g(h(b), z))$.

Substitution sur les termes

Définition

- La notion de substitution s'étend aux termes et se note $t\sigma$ (de manière postfixe), où t est un terme et σ une substitution ;
- Elle se définit par récurrence structurelle sur les termes :
 - ▶ Si $x \in \mathbb{V}$ alors $x\sigma = \sigma(x)$;
 - ▶ Si $f \in \mathcal{F}$ et $t_1, \dots, t_n \in \mathcal{T}$ alors $f(t_1, \dots, t_n)\sigma = f(t_1\sigma, \dots, t_n\sigma)$.

Exemples

- $f(x, g(y, z))[a/x, h(b)/y, c/z] = f(a, g(h(b), c))$;
- $f(x, g(y, z))[a/x, h(b)/y] = f(a, g(h(b), z))$.

Substitution sur les termes

Instances, composition, généralité

- Les instances d'un terme t sont tous les termes de la forme $t\sigma$, où σ est une substitution ;
- La composition $\sigma\sigma'$ de deux substitutions est définie par $t(\sigma\sigma') = (t\sigma)\sigma'$, où t est un terme ;
- Une substitution σ' est dite moins générale que σ , ce que nous notons $\sigma' \leq \sigma$, si et seulement s'il existe une substitution σ'' telle que $\sigma\sigma'' = \sigma'$.

Substitution sur les formules

Définition

- La notion de substitution s'étend aux formules et se note $\Phi\sigma$ (de manière postfixe), où Φ est un terme et σ une substitution ;
- Elle se définit par récurrence structurelle sur les formules :
 - ▶ Si $P \in \mathcal{S}_p$ d'arité n et $t_1, \dots, t_n \in \mathcal{T}$ alors
 $P(t_1, \dots, t_n)\sigma = P(t_1\sigma, \dots, t_n\sigma)$;
 - ▶ $\perp\sigma = \perp$, $\top\sigma = \top$;
 - ▶ Si $\Phi \in \mathcal{F}$ alors $(\neg\Phi)\sigma = \neg(\Phi\sigma)$;
 - ▶ Si $\Phi, \Phi' \in \mathcal{F}$ alors $(\Phi \oplus \Phi')\sigma = \Phi\sigma \oplus \Phi'\sigma$, où $\oplus \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$;
 - ▶ Si $x \in \mathcal{V}$ et $\Phi \in \mathcal{F}$ alors $(Qx.\Phi)\sigma = Qx'.\Phi[x'/x]\sigma$, où $Q \in \{\forall, \exists\}$ et $x' \notin \text{dom}(\sigma) \cup \text{yield}(\sigma) \cup (FV(\Phi) \setminus \{x\})$.

Exemples

- $P(f(x), g(y, z))[a/x, h(b)/y, c/z] = P(f(a), g(h(b), c))$;
- $(\forall x.P(f(x), g(y, z)))[a/x, h(b)/y, c/z] = \forall x'.P(f(x'), g(h(b), c))$.

Substitution sur les formules

Définition

- La notion de substitution s'étend aux formules et se note $\Phi\sigma$ (de manière postfixe), où Φ est un terme et σ une substitution ;
- Elle se définit par récurrence structurelle sur les formules :
 - ▶ Si $P \in \mathcal{S}_p$ d'arité n et $t_1, \dots, t_n \in \mathcal{T}$ alors
 $P(t_1, \dots, t_n)\sigma = P(t_1\sigma, \dots, t_n\sigma)$;
 - ▶ $\perp\sigma = \perp$, $\top\sigma = \top$;
 - ▶ Si $\Phi \in \mathcal{F}$ alors $(\neg\Phi)\sigma = \neg(\Phi\sigma)$;
 - ▶ Si $\Phi, \Phi' \in \mathcal{F}$ alors $(\Phi \oplus \Phi')\sigma = \Phi\sigma \oplus \Phi'\sigma$, où $\oplus \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$;
 - ▶ Si $x \in \mathcal{V}$ et $\Phi \in \mathcal{F}$ alors $(Qx.\Phi)\sigma = Qx'.\Phi[x'/x]\sigma$, où $Q \in \{\forall, \exists\}$ et $x' \notin \text{dom}(\sigma) \cup \text{yield}(\sigma) \cup (FV(\Phi) \setminus \{x\})$.

Exemples

- $P(f(x), g(y, z))[a/x, h(b)/y, c/z] = P(f(a), g(h(b), c))$;
- $(\forall x.P(f(x), g(y, z)))[a/x, h(b)/y, c/z] = \forall x'.P(f(x'), g(h(b), c))$.

Unification

Définition

- Une substitution σ est un unificateur de deux termes s et t si et seulement si $s\sigma = t\sigma$;
- Une substitution ρ est un renommage si et seulement si elle envoie des variables vers des variables, et elle est bijective.

Exemples

- $f(x, g(a))$ et $f(b, y)$ sont unifiables, l'unificateur est $[b/x, g(a)/y]$;
- $\rho = [x'/x, y'/y]$ est un renommage : $f(x, g(y))\rho = f(x', g(y'))$.

Unification

Définition

- Une substitution σ est un unificateur de deux termes s et t si et seulement si $s\sigma = t\sigma$;
- Une substitution ρ est un renommage si et seulement si elle envoie des variables vers des variables, et elle est bijective.

Exemples

- $f(x, g(a))$ et $f(b, y)$ sont unifiables, l'unificateur est $[b/x, g(a)/y]$;
- $\rho = [x'/x, y'/y]$ est un renommage : $f(x, g(y))\rho = f(x', g(y'))$.

Unification

Théorème

- Étant donnés deux termes s et t , soit il n'existe aucun unificateur de s et t , et nous disons alors que s et t ne sont pas unifiables, ou il existe un unificateur le plus général ou *mgu* ("most general unifier" en anglais), c'est-à-dire une substitution σ telle que :
 - ▶ σ est un unificateur de s et t , autrement dit, $s\sigma = t\sigma$;
 - ▶ Tout unificateur σ' de s et t est une instance de σ , c'est-à-dire qu'il existe une substitution σ'' telle que $\sigma' = \sigma\sigma''$.

Question

- Comment calculer le *mgu* de deux termes ?

Unification

Théorème

- Étant donnés deux termes s et t , soit il n'existe aucun unificateur de s et t , et nous disons alors que s et t ne sont pas unifiables, ou il existe un unificateur le plus général ou *mgu* ("most general unifier" en anglais), c'est-à-dire une substitution σ telle que :
 - ▶ σ est un unificateur de s et t , autrement dit, $s\sigma = t\sigma$;
 - ▶ Tout unificateur σ' de s et t est une instance de σ , c'est-à-dire qu'il existe une substitution σ'' telle que $\sigma' = \sigma\sigma''$.

Question

- Comment calculer le *mgu* de deux termes ?

Algorithme d'unification de Robinson

- $G\{s_1 = t_1, \dots, s_n = t_n\} \hookrightarrow \{x_1 = u_1, \dots, x_m = u_m\}$
où x_i sont des variables distinctes et $x_i \notin u_j$;
- Règles :
 - ▶ $G \cup \{t = t\} \hookrightarrow G$ (delete);
 - ▶ $G \cup \{f(s_1, \dots, s_n) = f(t_1, \dots, t_n)\} \hookrightarrow G \cup \{s_1 = t_1, \dots, s_n = t_n\}$ (decompose);
 - ▶ $G \cup \{f(s_1, \dots, s_n) = g(t_1, \dots, t_m)\} \hookrightarrow \perp$, si $f \neq g$ ou $n \neq m$ (conflict);
 - ▶ $G \cup \{f(s_1, \dots, s_n) = x\} \hookrightarrow G \cup \{x = f(s_1, \dots, s_n)\}$ (swap);
 - ▶ $G \cup \{x = t\} \hookrightarrow G[t/x] \cup \{x = t\}$, si $x \notin t$ et $x \in G$ (eliminate);
 - ▶ $G \cup \{x = f(s_1, \dots, s_n)\} \hookrightarrow \perp$, si $x \in f(s_1, \dots, s_n)$ (check).

Unification

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(x, g(a)) = f(b, y)\}$;
- $\{f(x, g(a)) = f(b, y)\} \xrightarrow{\text{decompose}} \{x = b, g(a) = y\} \xrightarrow{\text{swap}} \{x = b, y = g(a)\}$;
- $\text{mgu}(f(x, g(a)), f(b, y)) = [b/x, g(a)/y]$.

Unification

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(x, g(a)) = f(b, y)\}$;
- $\{f(x, g(a)) = f(b, y)\} \hookrightarrow_{\text{decompose}} \{x = b, g(a) = y\} \hookrightarrow_{\text{swap}} \{x = b, y = g(a)\}$;
- $\text{mgu}(f(x, g(a)), f(b, y)) = [b/x, g(a)/y]$.

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(x, g(a)) = f(b, y)\}$;
- $\{f(x, g(a)) = f(b, y)\} \hookrightarrow_{\text{decompose}} \{x = b, g(a) = y\} \hookrightarrow_{\text{swap}} \{x = b, y = g(a)\}$;
- $\text{mgu}(f(x, g(a)), f(b, y)) = [b/x, g(a)/y]$.

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(x, g(a)) = f(b, y)\}$;
- $\{f(x, g(a)) = f(b, y)\} \hookrightarrow_{\text{decompose}} \{x = b, g(a) = y\} \hookrightarrow_{\text{swap}} \{x = b, y = g(a)\}$;
- $\text{mgu}(f(x, g(a)), f(b, y)) = [b/x, g(a)/y]$.

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(x, g(a)) = f(b, y)\}$;
- $\{f(x, g(a)) = f(b, y)\} \hookrightarrow_{\text{decompose}} \{x = b, g(a) = y\} \hookrightarrow_{\text{swap}} \{x = b, y = g(a)\}$;
- $\text{mgu}(f(x, g(a)), f(b, y)) = [b/x, g(a)/y]$.

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(x, g(a)) = f(b, y)\}$;
- $\{f(x, g(a)) = f(b, y)\} \hookrightarrow_{\text{decompose}} \{x = b, g(a) = y\} \hookrightarrow_{\text{swap}} \{x = b, y = g(a)\}$;
- $\text{mgu}(f(x, g(a)), f(b, y)) = [b/x, g(a)/y]$.

Unification

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(g(x), x) = f(y, a)\}$;
- $\{f(g(x), x) = f(y, a)\} \hookrightarrow_{\text{decompose}}$
 $\{g(x) = y, x = a\} \hookrightarrow_{\text{swap}}$
 $\{y = g(x), x = a\} \hookrightarrow_{\text{eliminate}}$
 $\{y = g(a), x = a\}$;
- $mgu(f(g(x), x), f(y, a)) = [a/x, g(a)/y]$.

Unification

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(g(x), x) = f(y, a)\}$;
- $\{f(g(x), x) = f(y, a)\} \hookrightarrow_{\text{decompose}}$
 $\{g(x) = y, x = a\} \hookrightarrow_{\text{swap}}$
 $\{y = g(x), x = a\} \hookrightarrow_{\text{eliminate}}$
 $\{y = g(a), x = a\}$;
- $mgu(f(g(x), x), f(y, a)) = [a/x, g(a)/y]$.

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(g(x), x) = f(y, a)\}$;
- $\{f(g(x), x) = f(y, a)\} \hookrightarrow_{\text{decompose}}$
 $\{g(x) = y, x = a\} \hookrightarrow_{\text{swap}}$
 $\{y = g(x), x = a\} \hookrightarrow_{\text{eliminate}}$
 $\{y = g(a), x = a\}$;
- $mgu(f(g(x), x), f(y, a)) = [a/x, g(a)/y]$.

Unification

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(g(x), x) = f(y, a)\}$;
- $\{f(g(x), x) = f(y, a)\} \hookrightarrow_{\text{decompose}} \{g(x) = y, x = a\} \hookrightarrow_{\text{swap}} \{y = g(x), x = a\} \hookrightarrow_{\text{eliminate}} \{y = g(a), x = a\}$;
- $mgu(f(g(x), x), f(y, a)) = [a/x, g(a)/y]$.

Unification

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(g(x), x) = f(y, a)\}$;
- $\{f(g(x), x) = f(y, a)\} \hookrightarrow_{\text{decompose}}$
 $\{g(x) = y, x = a\} \hookrightarrow_{\text{swap}}$
 $\{y = g(x), x = a\} \hookrightarrow_{\text{eliminate}}$
 $\{y = g(a), x = a\}$;
- $mgu(f(g(x), x), f(y, a)) = [a/x, g(a)/y]$.

Unification

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(g(x), x) = f(y, a)\}$;
- $\{f(g(x), x) = f(y, a)\} \hookrightarrow_{\text{decompose}}$
 $\{g(x) = y, x = a\} \hookrightarrow_{\text{swap}}$
 $\{y = g(x), x = a\} \hookrightarrow_{\text{eliminate}}$
 $\{y = g(a), x = a\}$;
- $mgu(f(g(x), x), f(y, a)) = [a/x, g(a)/y]$.

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(g(x), x) = f(y, a)\}$;
- $\{f(g(x), x) = f(y, a)\} \hookrightarrow_{\text{decompose}}$
 $\{g(x) = y, x = a\} \hookrightarrow_{\text{swap}}$
 $\{y = g(x), x = a\} \hookrightarrow_{\text{eliminate}}$
 $\{y = g(a), x = a\}$;
- $\text{mgu}(f(g(x), x), f(y, a)) = [a/x, g(a)/y]$.

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(g(x), x) = f(y, a)\}$;
- $\{f(g(x), x) = f(y, a)\} \hookrightarrow_{\text{decompose}}$
 $\{g(x) = y, x = a\} \hookrightarrow_{\text{swap}}$
 $\{y = g(x), x = a\} \hookrightarrow_{\text{eliminate}}$
 $\{y = g(a), x = a\}$;
- $mgu(f(g(x), x), f(y, a)) = [a/x, g(a)/y]$.

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(x, b) = f(a, x)\}$;
- $\{f(x, b) = f(a, x)\} \hookrightarrow_{\text{decompose}}$
 $\{x = a, b = x\} \hookrightarrow_{\text{eliminate}}$
 $\{x = a, b = a\} \hookrightarrow_{\text{conflict}}$
 \perp ;
- $f(x, b)$ et $f(a, x)$ ne sont pas unifiables.

Unification

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(x, b) = f(a, x)\}$;
- $\{f(x, b) = f(a, x)\} \hookrightarrow_{\text{decompose}}$
 $\{x = a, b = x\} \hookrightarrow_{\text{eliminate}}$
 $\{x = a, b = a\} \hookrightarrow_{\text{conflict}}$
 \perp ;
- $f(x, b)$ et $f(a, x)$ ne sont pas unifiables.

Unification

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(x, b) = f(a, x)\}$;
- $\{f(x, b) = f(a, x)\} \hookrightarrow_{\text{decompose}}$
 $\{x = a, b = x\} \hookrightarrow_{\text{eliminate}}$
 $\{x = a, b = a\} \hookrightarrow_{\text{conflict}}$
 \perp ;
- $f(x, b)$ et $f(a, x)$ ne sont pas unifiables.

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(x, b) = f(a, x)\}$;
- $\{f(x, b) = f(a, x)\} \hookrightarrow_{\text{decompose}}$
 $\{x = a, b = x\} \hookrightarrow_{\text{eliminate}}$
 $\{x = a, b = a\} \hookrightarrow_{\text{conflict}}$
 \perp ;
- $f(x, b)$ et $f(a, x)$ ne sont pas unifiables.

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(x, b) = f(a, x)\}$;
- $\{f(x, b) = f(a, x)\} \hookrightarrow_{\text{decompose}}$
 $\{x = a, b = x\} \hookrightarrow_{\text{eliminate}}$
 $\{x = a, b = a\} \hookrightarrow_{\text{conflict}}$
 \perp ;
- $f(x, b)$ et $f(a, x)$ ne sont pas unifiables.

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(x, b) = f(a, x)\}$;
- $\{f(x, b) = f(a, x)\} \hookrightarrow_{\text{decompose}}$
 $\{x = a, b = x\} \hookrightarrow_{\text{eliminate}}$
 $\{x = a, b = a\} \hookrightarrow_{\text{conflict}}$
 \perp ;
- $f(x, b)$ et $f(a, x)$ ne sont pas unifiables.

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(x, b) = f(a, x)\}$;
- $\{f(x, b) = f(a, x)\} \hookrightarrow_{\text{decompose}}$
 $\{x = a, b = x\} \hookrightarrow_{\text{eliminate}}$
 $\{x = a, b = a\} \hookrightarrow_{\text{conflict}}$
 \perp ;
- $f(x, b)$ et $f(a, x)$ ne sont pas unifiables.

Algorithme d'unification de Robinson

- Ensemble initial d'équations : $\{f(x, b) = f(a, x)\}$;
- $\{f(x, b) = f(a, x)\} \hookrightarrow_{\text{decompose}}$
 $\{x = a, b = x\} \hookrightarrow_{\text{eliminate}}$
 $\{x = a, b = a\} \hookrightarrow_{\text{conflict}}$
 \perp ;
- $f(x, b)$ et $f(a, x)$ ne sont pas unifiables.

Extension aux atomes des formules

- Deux atomes $P(t_1, \dots, t_n)$ et $Q(t'_1, \dots, t'_m)$, avec $P, Q \in \mathcal{P}$ et $t_1, \dots, t_n, t'_1, \dots, t'_m \in \mathcal{T}$ sont unifiables si et seulement si :
 - ▶ $P = Q$ (même symbole de prédicat) ;
 - ▶ $n = m$ (même arité) ;
 - ▶ $\{t_1 = t'_1, \dots, t_n = t'_n\} \not\vdash \perp$ (on a une solution).

Méthodes de déduction automatique

- Résolution, tableaux clausaux, solveurs SMT, etc. ;
- Méthodes par réfutation :
 - ▶ On nie la formule initiale (que l'on doit montrer insatisfiable) ;
 - ▶ On la clausifie (skolémisation + mise en forme cnf).
- L'unification est utilisée dans les règles de recherche de preuve de ces méthodes (pas forcément de la même manière).