Manipulations syntaxiques

David Delahaye

Faculté des Sciences
David.Delahaye@lirmm.fr

Licence L3 2021-2022

Formes prénexes

Définition

- Une formule est dite en forme prénexe si et seulement si elle est de la forme $Q_1 x_1 . \ldots . Q_n x_n . \Phi$, où $Q_1, \ldots, Q_n$ 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
$\neg\exists x.\Phi$	$\forall x.\neg\Phi$
$(\exists x.\Phi) \wedge \Phi'$	$\exists x.\Phi \wedge \Phi'$
$(\exists x.\Phi) \vee \Phi'$	$\exists x.\Phi \vee \Phi'$
$(\exists x.\Phi) \Rightarrow \Phi'$	$\forall x.\Phi \Rightarrow \Phi'$
$\Phi' \wedge (\exists x.\Phi)$	$\exists x.\Phi' \wedge \Phi$
$\Phi' \vee (\exists x.\Phi)$	$\exists x.\Phi' \vee \Phi$
$\Phi' \Rightarrow (\exists x.\Phi)$	$\exists x.\Phi' \Rightarrow \Phi$

Formes prénexes

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 $(\Phi \Rightarrow \Phi') \wedge (\Phi' \Rightarrow \Phi)$.

Formes prénexes

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 :
$[\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I = [\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I_\rho = (\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \bigvee_{v \in D_I} (\neg_{\mathcal{B}} [\![\Phi]\!]^I_{\rho[v/x]} \vee_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = \bigvee_{v \in D_I} ([\![\Phi]\!]^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I$.

Formes prénexes

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 :
  $[\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I = [\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I_\rho = (\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \bigvee_{v \in D_I} (\neg_{\mathcal{B}} [\![\Phi]\!]^I_{\rho[v/x]} \vee_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = \bigvee_{v \in D_I} ([\![\Phi]\!]^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I$.

Formes prénexes

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 :
  $[\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I = [\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I_\rho = (\bigwedge_{v\in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \neg_{\mathcal{B}}(\bigwedge_{v\in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = (\bigvee_{v\in D_I} \neg_{\mathcal{B}}[\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \bigvee_{v\in D_I}(\neg_{\mathcal{B}}[\![\Phi]\!]^I_{\rho[v/x]} \vee_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = \bigvee_{v\in D_I}([\![\Phi]\!]^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I$.

Formes prénexes

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 :
  $[\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I = [\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I_\rho = (\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \Rightarrow_\mathcal{B} [\![\Phi']\!]^I_\rho = \neg_\mathcal{B}(\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_\mathcal{B} [\![\Phi']\!]^I_\rho = (\bigvee_{v \in D_I} \neg_\mathcal{B} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_\mathcal{B} [\![\Phi']\!]^I_\rho = \bigvee_{v \in D_I} (\neg_\mathcal{B} [\![\Phi]\!]^I_{\rho[v/x]} \vee_\mathcal{B} [\![\Phi']\!]^I_{\rho[v/x]}) = \bigvee_{v \in D_I} ([\![\Phi]\!]^I_{\rho[v/x]} \Rightarrow_\mathcal{B} [\![\Phi']\!]^I_{\rho[v/x]}) = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I$.

Formes prénexes

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 : $[\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I = [\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I_\rho = (\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \bigvee_{v \in D_I} (\neg_{\mathcal{B}} [\![\Phi]\!]^I_{\rho[v/x]} \vee_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = \bigvee_{v \in D_I} ([\![\Phi]\!]^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I$.

Formes prénexes

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 :
  $[\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I = [\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I_\rho = (\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \bigvee_{v \in D_I} (\neg_{\mathcal{B}} [\![\Phi]\!]^I_{\rho[v/x]} \vee_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = \bigvee_{v \in D_I} ([\![\Phi]\!]^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I$.

Formes prénexes

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 : $[\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I = [\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I_\rho = (\bigwedge_{v\in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \neg_{\mathcal{B}}(\bigwedge_{v\in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = (\bigvee_{v\in D_I} \neg_{\mathcal{B}}[\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \bigvee_{v\in D_I}(\neg_{\mathcal{B}}[\![\Phi]\!]^I_{\rho[v/x]} \vee_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = \bigvee_{v\in D_I}([\![\Phi]\!]^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I$.

Formes prénexes

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 :
$[\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I = [\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I_\rho = (\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \Rightarrow_\mathcal{B} [\![\Phi']\!]^I_\rho = \neg_\mathcal{B}(\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_\mathcal{B} [\![\Phi']\!]^I_\rho = (\bigvee_{v \in D_I} \neg_\mathcal{B} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_\mathcal{B} [\![\Phi']\!]^I_\rho = \bigvee_{v \in D_I} (\neg_\mathcal{B} [\![\Phi]\!]^I_{\rho[v/x]} \vee_\mathcal{B} [\![\Phi']\!]^I_{\rho[v/x]}) = \bigvee_{v \in D_I} ([\![\Phi]\!]^I_{\rho[v/x]} \Rightarrow_\mathcal{B} [\![\Phi']\!]^I_{\rho[v/x]}) = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I$.

Formes prénexes

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 : $[\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I = [\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I_\rho = (\bigwedge_{v\in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \neg_{\mathcal{B}}(\bigwedge_{v\in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = (\bigvee_{v\in D_I} \neg_{\mathcal{B}}[\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \bigvee_{v\in D_I}(\neg_{\mathcal{B}}[\![\Phi]\!]^I_{\rho[v/x]} \vee_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = \bigvee_{v\in D_I}([\![\Phi]\!]^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I$.

Formes prénexes

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 :
$[\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I = [\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I_\rho = (\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \bigvee_{v \in D_I} (\neg_{\mathcal{B}} [\![\Phi]\!]^I_{\rho[v/x]} \vee_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = \bigvee_{v \in D_I} ([\![\Phi]\!]^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I$.

Formes prénexes

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 :
  $[\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I = [\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I_\rho = (\bigwedge_{v\in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \neg_{\mathcal{B}}(\bigwedge_{v\in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = (\bigvee_{v\in D_I} \neg_{\mathcal{B}}[\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \bigvee_{v\in D_I}(\neg_{\mathcal{B}}[\![\Phi]\!]^I_{\rho[v/x]} \vee_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = \bigvee_{v\in D_I}([\![\Phi]\!]^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I$.

Formes prénexes

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 :
$[\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I = [\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I_\rho = (\bigwedge_{v\in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \neg_{\mathcal{B}}(\bigwedge_{v\in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = (\bigvee_{v\in D_I} \neg_{\mathcal{B}}[\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \bigvee_{v\in D_I}(\neg_{\mathcal{B}}[\![\Phi]\!]^I_{\rho[v/x]} \vee_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = \bigvee_{v\in D_I}([\![\Phi]\!]^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I$.

Formes prénexes

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 :
  $[\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I = [\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I_\rho = (\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \neg_{\mathcal{B}}(\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = (\bigvee_{v \in D_I} \neg_{\mathcal{B}} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_{\mathcal{B}} [\![\Phi']\!]^I_\rho = \bigvee_{v \in D_I} (\neg_{\mathcal{B}} [\![\Phi]\!]^I_{\rho[v/x]} \vee_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = \bigvee_{v \in D_I} ([\![\Phi]\!]^I_{\rho[v/x]} \Rightarrow_{\mathcal{B}} [\![\Phi']\!]^I_{\rho[v/x]}) = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I$.

Formes prénexes

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 : $[\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I = [\![(\forall x.\Phi) \Rightarrow \Phi']\!]^I_\rho = (\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \Rightarrow_\mathcal{B} [\![\Phi']\!]^I_\rho = \neg_\mathcal{B}(\bigwedge_{v \in D_I} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_\mathcal{B} [\![\Phi']\!]^I_\rho = (\bigvee_{v \in D_I} \neg_\mathcal{B} [\![\Phi]\!]^I_{\rho[v/x]}) \vee_\mathcal{B} [\![\Phi']\!]^I_\rho = \bigvee_{v \in D_I} (\neg_\mathcal{B} [\![\Phi]\!]^I_{\rho[v/x]} \vee_\mathcal{B} [\![\Phi']\!]^I_{\rho[v/x]}) = \bigvee_{v \in D_I} ([\![\Phi]\!]^I_{\rho[v/x]} \Rightarrow_\mathcal{B} [\![\Phi']\!]^I_{\rho[v/x]}) = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I_\rho = [\![\exists x.\Phi \Rightarrow \Phi']\!]^I$.

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).

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).

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

- 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.

Skolémisation et herbrandisation

Formule existentielle/universelle

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

Skolémisation et herbrandisation

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).

Skolémisation et herbrandisation

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).

Skolémisation et herbrandisation

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).

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.

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

Skolémisation et herbrandisation

Fonctions de skolémisation et herbrandisation

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

Skolémisation et herbrandisation

Fonctions de skolémisation et herbrandisation

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

Skolémisation et herbrandisation

Fonctions de skolémisation et herbrandisation

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

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 : $\exists x_1. \dots .\exists x_n.h(\Phi)$, où $\{x_1, \dots, x_n\} = FV(h(\Phi))$.

Skolémisation et herbrandisation

Fonctions de skolémisation et herbrandisation

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

Skolémisation et herbrandisation

Fonctions de skolémisation et herbrandisation

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

Skolémisation et herbrandisation

Fonctions de skolémisation et herbrandisation

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

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).

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

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 : $\forall x.\forall z.P(x,f(x),z)$.

Skolémisation et herbrandisation

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 : $\forall x.\forall z.P(x, f(x), z)$.

Skolémisation et herbrandisation

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 : $\forall x.\forall z.P(x, f(x), z)$.

Skolémisation et herbrandisation

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 : $\forall x.\forall z.P(x, f(x), z)$.

Skolémisation et herbrandisation

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 : $\forall x.\forall z.P(x,f(x),z)$.

Skolémisation et herbrandisation

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 : $\forall x.\forall z.P(x,f(x),z)$.

Skolémisation et herbrandisation

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 : $\forall x.\forall z.P(x, f(x), z)$.

Skolémisation et herbrandisation

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 : $\forall x.\forall z.P(x,f(x),z)$.

Clausification

Principe

- On skolémise : 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 $P(t_1, \ldots, t_n)$ où $P \in \mathcal{P}$ et $t_1, \ldots, t_n \in \mathcal{T}$;
  - Un littéral est un atome ou la négation d'un atome ;
  - Une clause est une disjonction de littéraux.

Clausification

Mise en forme cnf

$$\neg\neg F \rightsquigarrow F \qquad \neg\top \rightsquigarrow \bot \qquad \neg\bot \rightsquigarrow \top$$
$$\neg(F_1 \land F_2) \rightsquigarrow \neg F_1 \lor \neg F_2 \qquad \neg(F_1 \lor F_2) \rightsquigarrow \neg F_1 \land \neg F_2$$
$$F_1 \Rightarrow F_2 \rightsquigarrow \neg F_1 \lor F_2$$
$$F_1 \land \top \rightsquigarrow F_1 \qquad \top \land F_1 \rightsquigarrow F_1 \qquad F_1 \land \bot \rightsquigarrow \bot \qquad \bot \land F_1 \rightsquigarrow \bot$$
$$F_1 \lor \top \rightsquigarrow \top \qquad \top \lor F_1 \rightsquigarrow \top \qquad F_1 \lor \bot \rightsquigarrow F_1 \qquad \bot \lor F_1 \rightsquigarrow F_1$$
$$(F_1 \land F_2) \lor F_3 \rightsquigarrow (F_1 \lor F_3) \land (F_2 \lor F_3)$$
$$F_3 \lor (F_1 \land F_2) \rightsquigarrow (F_3 \lor F_1) \land (F_3 \lor F_2)$$

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 : $S = \{P(x) \vee Q(x), \neg P(a), \neg Q(a)\}$.

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 : $S = \{P(x) \vee Q(x), \neg P(a), \neg Q(a)\}$.

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 : $S = \{P(x) \vee Q(x), \neg P(a), \neg Q(a)\}$.

Substitution

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 $[\sigma(x_1)/x_1, \ldots, \sigma(x_n)/x_n]$, où $x_1, \ldots, x_n$ 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 $t_1, \ldots, t_n \in \mathcal{T}$ alors $f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)$.

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 $t_1, \ldots, t_n \in \mathcal{T}$ alors $f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)$.

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 $P \in \mathcal{S}_\mathcal{P}$ d'arité n et $t_1, \ldots, t_n \in \mathcal{T}$ alors $P(t_1, \ldots, t_n)\sigma = P(t_1\sigma, \ldots, t_n\sigma)$;
  - $\bot\sigma = \bot$, $\top\sigma = \top$;
  - 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 $x' \notin dom(\sigma) \cup yield(\sigma) \cup (FV(\Phi) \backslash \{x\})$.

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 $P \in \mathcal{S}_\mathcal{P}$ d'arité n et $t_1, \ldots, t_n \in \mathcal{T}$ alors $P(t_1, \ldots, t_n)\sigma = P(t_1\sigma, \ldots, t_n\sigma)$;
  - $\bot\sigma = \bot$, $\top\sigma = \top$;
  - 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 $x' \notin dom(\sigma) \cup yield(\sigma) \cup (FV(\Phi)\backslash\{x\})$.

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 ?

Unification

Algorithme d'unification de Robinson

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

Unification

Algorithme d'unification de Robinson

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

Unification

Algorithme d'unification de Robinson

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

Unification

Algorithme d'unification de Robinson

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

Unification

Algorithme d'unification de Robinson

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

Unification

Algorithme d'unification de Robinson

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

Unification

Algorithme d'unification de Robinson

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

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

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

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

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.

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.

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.

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.

Unification

Extension aux atomes des formules

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

On est prêt

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).