Introduction à la logique

Stéphane Devismes **Pascal Lafourcade** Michel Lévy

Université Clermont Auvergne

Mathinfoly, Lyon, Août 2019

Organisation

Cours + TD

- Logique
- Projet de logique
- Cryptographie + Challenge
- Blockchain

Dimanche

- 9h00 - 10h30 Logic Lecture 1
- 11h00 - 12h30 TD of Logic 1
- 14h00 - 16h00 Cryptographic challenge

Lundi

- 9h00 - 10h30 Logic Lecture 2
- 11h00 - 12h30 TD of Logic 2
- 14h00 - 16h00 Begining of the projects

Mardi

- 9h00 - 10h30 Crypto Lecture
- 11h00 - 12h30 Blockchain

Par groupe de 5

1. Former une équipe
2. Choisir une sujet
3. Modéliser un jeu de votre choix
4. Coder pour utiliser un SAT-Solveur (glucose, minisat, zchaff, lingeling ...)
5. Afficher la solution
6. Présenter vos travaux

`https://en.wikipedia.org/wiki/Boolean_satisfiability_problem#SAT_problem_format`
`http://www.domagoj-babic.com/uploads/ResearchProjects/Spear/dimacs-cnf.pdf`

Exemple de sujets possibles

- Facile : N-Reine, Tout-noir-tout-blanc
- Moyen : Sudoku, coloration de graphe, Nori-Nori, Futoshiki, Dosun-fuwari, Squaro, Hanji
- Difficile : Tetravex, Takuzu, Hashiwokakero

Format DIMACS :

Standard international pour la représentation de formules en forme normale conjonctive, pour les SAT-solveurs.
Il commence par `p cnf 5 3`

- i `indique que la` i`-ème variable apparaît avec polarité positive dans la clause.`
- $-i$ `indique que la` i`-ème variable apparaît avec polarité négative dans la clause.`

`Il se termine par 0.`

Exemple :

$$(x_1 \vee \neg x_5 \vee x_4) \wedge (\neg x_1 \vee x_5 \vee x_3 \vee x_4) \wedge (\neg x_3 \vee \neg x_4)$$

```
c
c start with comments
c
c
p cnf 5 3
1 -5 4 0
-1 5 3 4 0
-3 -4 0
```

Exemple pour le problème des pigeons

Problème

Un colombophile possède n nids et p pigeons.

- chaque pigeon soit dans un nid,
- il y ait au plus un pigeon par nid.

Modélisation en logique du premier ordre

Le prédicat $P(i, j)$ représente le fait que le pigeon i est dans le nid j.

- Chaque pigeon est dans un nid : $\forall i, \exists j, P(i, j)$.
- Il y a au plus un pigeon par nid : $\forall i, \forall k, i \neq k \Longrightarrow \forall j, \overline{P(i, j)} \vee \overline{P(k, j)}$.

Modélisation en forme normale conjonctive

La variable booléenne $x_{i,j}$ représente le fait que le pigeon i est dans le nid j.
Pour un ensemble de pigeons $\{a, b, c\}$ et un ensemble de nids $\{1, 2, 3, 4\}$:

$$
\begin{aligned}
& (x_{a,1} \vee x_{a,2} \vee x_{a,3} \vee x_{a,4}) \\
\wedge\ & (x_{b,1} \vee x_{b,2} \vee x_{b,3} \vee x_{b,4}) \\
\wedge\ & (x_{c,1} \vee x_{c,2} \vee x_{c,3} \vee x_{c,4}) \\
\wedge\ & (\overline{x_{a,1}} \vee \overline{x_{b,1}}) \wedge (\overline{x_{a,1}} \vee \overline{x_{c,1}}) \wedge (\overline{x_{b,1}} \vee \overline{x_{c,1}}) \\
\wedge\ & (\overline{x_{a,2}} \vee \overline{x_{b,2}}) \wedge (\overline{x_{a,2}} \vee \overline{x_{c,2}}) \wedge (\overline{x_{b,2}} \vee \overline{x_{c,2}}) \\
\wedge\ & (\overline{x_{a,3}} \vee \overline{x_{b,3}}) \wedge (\overline{x_{a,3}} \vee \overline{x_{c,3}}) \wedge (\overline{x_{b,3}} \vee \overline{x_{c,3}}) \\
\wedge\ & (\overline{x_{a,4}} \vee \overline{x_{b,4}}) \wedge (\overline{x_{a,4}} \vee \overline{x_{c,4}}) \wedge (\overline{x_{b,4}} \vee \overline{x_{c,4}})
\end{aligned}
$$

Plan

Logique

Définitions

- La logique précise ce qu'est un raisonnement correct, indépendemment du domaine d'application.
- Un raisonnement est un moyen d'obtenir une conclusion à partir d'hypothèses données.
- Un raisonnement correct ne dit rien sur la vérité des hypothèses, il dit seulement que **de la vérité des hypothèses, on peut déduire la vérité de la conclusion**.

Exemple

- **Hypothèse I :** Tout ce qui est rare est cher
- **Hypothèse II :** Un cheval bon marché est rare
- **Conclusion :** Un cheval bon marché est cher !

Ajout d'une hypothèse

- **Hypothèse I :** Tout ce qui est rare est cher
- **Hypothèse II :** Un cheval bon marché est rare
- **Hypothèse III :** Tout ce qui est bon marché n'est pas cher
- **Conclusion :** Hypothèses contradictoires ! Car :
  - **Hypothèse I + hypothèse II :** Un cheval bon marché est cher
  - **Hypothèse III :** Un cheval bon marché n'est pas cher

Petit historique...

- **George Boole** (1815-1864)
  - *logique symbolique* : s'éloigne de la langue naturelle
- **Gottlob Frege** (1848-1925)
  - *calcul propositionnel* : formalisation des règles de raisonnement
  - *théorie de la démonstration* : démonstration = objet d'étude
- **Bertrand Russell** (1872-1970)
  - *logicisme* : programme de formalisation des mathématiques
  - *paradoxe* dans les premiers systèmes proposés
- **Kurt Gödel** (1906-1978)
  - *complétude* du calcul des prédicats du premier ordre
  - *théorème d'incomplétude* des systèmes incluant $\mathbb{N}$
- **Alonzo Church** (1903-1995)
  - *lambda-calcul* : représentation calculatoire des démonstrations

Applications

- **Hardware** (portes logique)
- **Vérification et correction des programmes** :
  - prouveurs COQ, PVS, Prover9, MACE, ...
  - applications industrielles (Meteor, Airbus...)
- **Intelligence artificielle** :
  - système expert (*MyCin*), ontologie
- **Programmation** : Prolog
  - intelligence artificielle
  - traitement de la langue
- **Preuves mathématiques**, **Sécurité**, **...**

Objectif du cours

- Modéliser et formaliser un problème décrit en langage naturel.

Logique propositionnelle

Définition

La logique propositionnelle est la logique *sans quantificateurs*.
Seules opérations logiques considérées :

- $\neg$ (négation)
- $\wedge$ (conjonction "et" aussi notée ou .)
- $\vee$ (disjonction "ou" aussi notée $+$)
- $\Rightarrow$ (implication)
- $\Leftrightarrow$ (équivalence)

Exemple : **Raisonnement formel**

Hypothèses :

- (H1) : Si Pierre est grand, alors Jean n'est pas le fils de Pierre
- (H2) : Si Pierre n'est pas grand, alors Jean est le fils de Pierre
- (H3) : Si Jean est le fils de Pierre alors Marie est la soeur de Jean

Conclusion (C) : Marie est la soeur de Jean ou Pierre est grand.

- p : "Pierre est grand"
- j : "Jean est le fils de Pierre"
- m : "Marie est la soeur de Jean"

- (H1) : $p \Rightarrow \neg j$
- (H2) : $\neg p \Rightarrow j$
- (H3) : $j \Rightarrow m$
- (C) : $m \vee p$

Il s'agira de montrer que $H1 \wedge H2 \wedge H3 \Rightarrow C$:

$$(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m) \Rightarrow m \vee p$$

est vraie quelque soit la valeur de vérité des propositions p, j, m.

Vocabulaire du langage

- Les constantes : $\top$ (vrai) et $\bot$ (faux)
- Les variables : par exemple x, y_1
- Les parenthèses
- Les connecteurs : $\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$

Taille d'une formule

Definition

La taille d'une formule A, notée $|A|$, est définie inductivement par :

- $|\top| = 0$ et $|\bot| = 0$.
- Si A est une variable alors $|A| = 0$.
- $|\neg A| = 1 + |A|$.
- $|(A \circ B)| = |A| + |B| + 1$.

Exemple

$|(a \vee (\neg b \wedge c))| =$

3.

Règles de priorité

Definition

Par ordre de priorités décroissantes : $\neg$, $\wedge$, $\vee$, $\Rightarrow$ et $\Leftrightarrow$.

Associativité à gauche

Pour deux connecteurs identiques $A \circ B \circ C \;=\; (A \circ B) \circ C$
sauf pour l'implication : $A \Rightarrow B \Rightarrow C \;=\; A \Rightarrow (B \Rightarrow C)$

Exemples de formules à priorité

Exemple

- $a \wedge b \wedge c$ est l'abréviation de

$((a \wedge b) \wedge c)$

- $a \wedge b \vee c$ est l'abréviation de

$((a \wedge b) \vee c)$

- $a \vee b \wedge c$ est l'abréviation de

$(a \vee (b \wedge c))$

Assignation d'une formule

Definition

Une assignation est une fonction qui,

à chaque variable d'une formule, associe une valeur dans $\{0, 1\}$.

$[A]_v$ dénote la valeur de la formule A dans l'assignation v.

Exemple : Soit v une assignation telle que $v(x) = 0$ et $v(y) = 1$.
Appliquer v à $x \vee y$ s'écrit $[x \vee y]_v$
$[x \vee y]_v = 0 \vee 1 = 1$
Conclusion : $x \vee y$ est vrai pour l'assignation v

Valeur d'une formule

Definition

Soient A, B des formules, x une variable et v une assignation.

- $[x]_v = v(x)$
- $[\top]_v = 1$, $[\bot]_v = 0$
- $[\neg A]_v = 1 - [A]_v$
- $[(A \vee B)]_v = max\{[A]_v, [B]_v\}$
- $[(A \wedge B)]_v = min\{[A]_v, [B]_v\}$
- $[(A \Rightarrow B)]_v =$ si $[A]_v = 0$ alors 1 sinon $[B]_v$
- $[(A \Leftrightarrow B)]_v =$ si $[A]_v = [B]_v$ alors 1 sinon 0

Table de vérité

Definition

Une table de vérité donne la valeur d'une formule pour **chaque** choix de valeurs des variables de A.

- une ligne de la table de vérité = une assignation
- une colonne = toutes les valeurs d'une formule.

Tables de base

On associe à chaque formule une *valeur* : 0 (faux) ou 1 (vrai).
La constante $\top$ vaut 1 et la constante $\bot$ vaut 0.

Table de vérité des connecteurs

x	y	$\neg x$	$x \vee y$	$x \wedge y$	$x \Rightarrow y$	$x \Leftrightarrow y$
0	0	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	1	0	0	0
1	1	0	1	1	1	1

Exemple :

Exemple

Donner la table de vérité des formules suivantes.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0	1	1	1	1	1
0	1	1	1	1	1	0
1	0	0	0	0	1	1
1	1	1	0	1	1	1

Formules équivalentes

Definition

Deux formules A et B sont équivalentes (noté $A \equiv B$ ou simplement $A = B$) si elles ont la même valeur pour toute assignation.

Exemple

$x \Rightarrow y \equiv \neg x \vee y$

Remarque :
Le connecteur logique $\Leftrightarrow$ ne signifie pas $A \equiv B$.

Valide, tautologie (1/2)

Definition

- Une formule est valide si elle a la valeur 1 pour toute assignation.
- Aussi appelée une tautologie.
- Noté $\models A$.

Exemple

- $(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$ est valide ;
- $x \Rightarrow y$ n'est pas valide car

elle est fausse pour $x = 1$ et $y = 0$.

Valide, tautologie (2/2)

Propriete

Les formules A et B sont équivalentes ($A \equiv B$)
si et seulement si
la formule $A \Leftrightarrow B$ est valide.

Cf table de vérité de $\Leftrightarrow$.

Modèle d'une formule

Definition

Une assignation v qui donne la valeur 1 à une formule est un modèle de cette formule.

On dit aussi que v satisfait A ou v rend A vraie.

Exemple

Un modèle de $x \Rightarrow y$ est :

$x = 1, y = 1$ (il y en a d'autres).

Par contre $x = 1, y = 0$ n'est pas un modèle de $x \Rightarrow y$.

Modèle d'un ensemble de formules

Definition

v est un modèle de l'ensemble $\{A_1, \ldots, A_n\}$
si et seulement si
elle est un modèle de chacune de ces formules.

Exemple

Un modèle de $\{a \Rightarrow b, b \Rightarrow c\}$ est :

$a = 0, b = 0$ (et c quelconque).

Propriété d'un modèle d'un ensemble de formules

Propriete

v est un modèle de $\{A_1, \ldots, A_n\}$
si et seulement si
v est un modèle de $A_1 \wedge \ldots \wedge A_n$.

Exemple

L'ensemble de formules $\{a \Rightarrow b, b \Rightarrow c\}$
et la formule $(a \Rightarrow b) \wedge (b \Rightarrow c)$
ont les mêmes modèles.

Contre-modèle

Definition

Une assignation v qui donne la valeur 0 à A est un contre-modèle de A.

On dit que v ne satisfait pas A ou que v rend la formule fausse.

Exemple

Un contre-modèle de $x \Rightarrow y$ est :

$x = 1, y = 0.$

Formule satisfaisable

Definition

Un (ensemble de) formule(s) est satisfaisable s'il admet un modèle.

Definition

Un (ensemble de) formule(s) est insatisfaisable s'il n'est pas satisfaisable.

Exemple

$x \wedge \neg x$ est insatisfaisable, mais $x \Rightarrow y$ est satisfaisable.

Attention

insatisfaisable = 0 modèle	satisfaisable = 1 modèle ou plus
invalide = 1 contre-modèle ou plus	valide = 0 contre-modèle

Propriété INCONTOURNABLE

Constamment utilisée dans les exercices. Vérifier la correction du raisonnemnt qui conclut B à partir de H_n

Propriete

Soit $H_n = A_1 \wedge \ldots \wedge A_n$.
Les 3 formulations suivantes sont équivalentes :

1. $A_1, \ldots, A_n \models B$
2. $H_n \Rightarrow B$ est valide.
3. $H_n \wedge \neg B$ est insatisfaisable.

Démonstration.

Elle se base sur les tables de vérité des connecteurs.
On procède en démontrant que $1 \Rightarrow 2$ puis $2 \Rightarrow 3$ et $3 \Rightarrow 1$. □

La disjonction, notée $\vee$ ou $+$

- associative $x \vee (y \vee z) \equiv (x \vee y) \vee z$
- commutative $x \vee y \equiv y \vee x$
- idempotente $x \vee x \equiv x$

Idem pour la conjonction, notée $\wedge$ ou .

Distributivité

- La conjonction est distributive sur la disjonction
  $x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$
- La disjonction est distributive sur la conjonction
  $x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$

Neutralité et Absortion

- 0 est l'élément neutre de la disjonction $0 \vee x \equiv x$
- 1 est l'élément neutre de la conjonction $1 \wedge x \equiv x$
- 1 est l'élément absorbant de la disjonction $1 \vee x \equiv 1$
- 0 est l'élément absorbant de la conjonction $0 \wedge x \equiv 0$

Négation

- Les lois de la négation :
  - $x \wedge \neg x \equiv 0$
  - $x \vee \neg x \equiv 1$ (Le tiers-exclus)
- $\neg\neg x \equiv x$
- $\neg 0 \equiv 1$
- $\neg 1 \equiv 0$

Les lois de De Morgan

- $\neg(x \wedge y) \equiv \neg x \vee \neg y$
- $\neg(x \vee y) \equiv \neg x \wedge \neg y$

Lois de simplification

Propriete

Pour tout x, y nous avons :

- $x \vee (x \wedge y) \equiv x$
- $x \wedge (x \vee y) \equiv x$
- $x \vee (\neg x \wedge y) \equiv x \vee y$

Substitution

Definition

Une substitution σ est une application de l'ensemble des variables dans l'ensemble des formules.

$A\sigma$ = remplacer dans la formule toute variable x par la formule $\sigma(x)$.

Exemple : $A = \neg(p \wedge q) \Leftrightarrow (\neg p \vee \neg q)$

- Soit σ la substitution suivante : $\sigma(p) = (a \vee b), \sigma(q) = (c \wedge d)$
- $A\sigma = \neg((a \vee b) \wedge (c \wedge d)) \Leftrightarrow (\neg(a \vee b) \vee \neg(c \wedge d))$

Définitions

Definition

- Un littéral est une variable ou la négation d'une variable.
- Un monôme est une conjonction de littéraux.
- Une clause est une disjonction de littéraux.
  (cas particuliers : 0 et 1)

Exemple

- $x, y, \neg z$ sont des littéraux.
- $x \wedge \neg y \wedge z$ est un monôme
- Le monôme $x \wedge \neg y \wedge z \wedge \neg x$ comporte x et $\neg x$: il vaut 0.
- $x \vee \neg y \vee z$ est une clause
- La clause $x \vee \neg y \vee z \vee \neg x$ comporte x et $\neg x$: elle vaut 1.

Forme normale

Definition

Formule en forme normale = seulement $\wedge, \vee$ et des négations sur les variables.

Exemple

La formule $\neg a \vee b$ est en forme normale.

$a \Rightarrow b$ est équivalente mais n'est pas en forme normale.

Toute formule admet une forme normale équivalente.

Mise en forme normale

1. **Élimination des équivalences**
   Remplacer $A \Leftrightarrow B$ par
   (a) $(\neg A \vee B) \wedge (\neg B \vee A)$
   OU
   (b) $(A \wedge B) \vee (\neg A \wedge \neg B)$
2. **Élimination des implications**
   Remplacer $A \Rightarrow B$ par $\neg A \vee B$
3. **Déplacement des négations vers les variables**
   Remplacer
   (a) $\neg\neg A$ par A
   (b) $\neg(A \vee B)$ par $\neg A \wedge \neg B$
   (c) $\neg(A \wedge B)$ par $\neg A \vee \neg B$

Remarque : simplifications

Simplifier le plus tôt possible :

1. Remplacer $\neg(A \Rightarrow B)$ par $A \wedge \neg B$
2. Remplacer une conjonction par 0 si elle comporte une formule et sa négation
3. Remplacer une disjonction par 1 si elle comporte une formule et sa négation
4. Appliquer :
   - l'idempotence de la conjonction et de la disjonction,
   - le caractère neutre ou absorbant de 0 et de 1,
   - remplacer $\neg 1$ par 0 et $\neg 0$ par 1.
5. Appliquer les simplifications :
   - $x \vee (x \wedge y) = x$,
   - $x \wedge (x \vee y) = x$,
   - $x \vee (\neg x \wedge y) = x \vee y$

Forme normale disjonctive

Definition

Une formule est en forme normale disjonctive (FND) si et seulement si elle est une disjonction (somme) de monômes.

Méthode : distribuer les conjonctions sur les disjonctions
$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
L'intérêt des FND est de mettre en évidence les modèles.

Exemple

$(x \wedge y) \vee (\neg x \wedge \neg y \wedge z)$ est une FND, qui a deux modèles principaux :

- $x = 1, y = 1$
- $x = 0, y = 0, z = 1$

Forme normale conjonctive

Definition

Une formule est en forme normale conjonctive (FNC) si et seulement si elle est une conjonction (produit) de clauses.

Appliquer la distributivité (!) de la disjonction sur la conjonction :

- $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$

L'intérêt des FNC est de mettre en évidence les contre-modèles.

Exemple

$(x \vee y) \wedge (\neg x \vee \neg y \vee z)$ est une FNC, qui a deux contre-modèles

- $x = 0, y = 0$
- $x = 1, y = 1, z = 0.$

Utilisée en modélisation (SAT-solvers)

Exemples

Mise en FND de :

$$(a \vee b) \wedge (c \vee d \vee e) \equiv$$

$$(a \wedge c) \vee (a \wedge d) \vee (a \wedge e) \vee (b \wedge c) \vee (b \wedge d) \vee (b \wedge e).$$

Mise en FNC de :

$$(a \wedge b) \vee (c \wedge d \wedge e) \equiv$$

$$(a \vee c) \wedge (a \vee d) \wedge (a \vee e) \wedge (b \vee c) \wedge (b \vee d) \wedge (b \vee e).$$

Théorèmes de Goëdel


En 1931

Théorème d'incomplètude

Dans n'importe quelle théorie récursivement axiomatisable, cohérente et capable de " formaliser l'arithmétique ", on peut construire un énoncé arithmétique qui ne peut être ni prouvé ni réfuté dans cette théorie.

Théorème d'incomplètude

Si T est une théorie cohérente qui satisfait des hypothèses analogues, la cohérence de T, qui peut s'exprimer dans la théorie T, n'est pas démontrable dans T.

Preuve simplifiée

Machine de Turing

Pour tout programme de taille finie, la machine de Truing répond VRAI ou FAUX à une affirmation qu'on lui donne, sans jamais se tromper.

Que signifie alors le théorème de Goëdel

Si un humain est capable de savoir si la phrase qu'il donne à la machine est vraie ou fausse, la machine est-elle aussi capable de découvrir la vérité ?

La phrase de Goëdel

"La machine ne répondra jamais VRAI à cette phrase"

La preuve ou que fait la machine ?

"La machine ne répondra jamais VRAI à cette phrase"

Deux réponses sont possibles

- VRAI
- FAUX

La preuve ou que fait la machine ?

"La machine ne répondra jamais VRAI à cette phrase"

VRAI

"La machine ne répondra jamais VRAI à cette phrase" est donc une affirmation vraie.
Si la machine ne se trompe pas, elle ne peut donc pas répondre VRAI.

FAUX

"La machine ne répondra jamais VRAI à cette phrase" est une affirmation fausse.
Si la machine ne se trompe pas, elle ne peut donc pas répondre FAUX.

La preuve ou que fait un homme ?

"La machine ne répondra jamais VRAI à cette phrase"

Nous venons de voir que la machine ne peut pas répondre VRAI.
Nous savons aussi que cette phrase est une vérité.
Pourtant la machine ne pourra pas la découvrir...

Question dans Don Quichotte :

À la frontière d'un pays, il faut dire la vérité sinon c'est la pendaison

Un garde frontière vous demande :
“ Pourquoi venez-vous ?”
Vous répondez :

“ Pour être pendu !”

Don quichotte de Miguel de Cervantes

"Par une ancienne loi de cette île, tout homme qui vient après la retraite sonnée pour passer ce pont est obligé de nous déclarer, sous la foi du serment, où il va. S'il dit la vérité nous le laissons passer sans obstacle ; s'il fait le moindre mensonge, il est pendu sur-le-champ à une potence dressée à l'autre bout de de ce pont. Cette loi est connue de tous les habitants de votre île. Tout à l'heure l'homme que voici s'est presenté pour passer : nous l'avons interrogé suivant l'usage ; il a levé la main et nous a répondu qu'il allait se faire pendre à cette potence."

Don quichotte

“Si nous le pendons en effet , il a dit vrai, et ne mérite pas la mort ; si nous le laissons passer, il a menti, et la loi veut qu’il soit pendu....
Mais écoutez : quelle que soit notre décision, nous manquerons toujours à la loi ;
s’il est pendu, nous sommes en faute, puisqu’il aura dit la vérité ;
s’il n’est pas pendu, nous sommes encore en faute, puisqu’il nous aura menti.
Nous n’avons donc que le choix de deux fautes : or, dans ce cas, nous devons choisir celle qui ne fait de mal qu’à nous. Qu’on laisse passer cet homme ; s’il aime tant à être pendu , nous le punissons assez en le contrariant pour aujourd’hui.”

Historique

Algorithme de Davis, Putnam, Logemann et Loveland

- Inventé par Martin Davis et Hilary Putnam en 1960, puis amélioré par Martin Davis, George Logemann et Donald Loveland en 1962
- Permet de savoir si un ensemble de clauses est satisfaisable.
- Base de SAT-solveurs complets comme **chaff, zchaff et satz**.

Principe I

Deux types de transformations de formules :

1. préservant le sens : transformant une formule en une formule équivalente
   - réduction
2. préservant seulement la satisfaisabilité : transformant une formule satisfaisable en une formule satisfaisable.
   - suppression des clauses qui ont des littéraux isolés
   - résolution unitaire

DPLL est efficace car il utilise ces 2 transformations.

Principe II

« Branchement/Retour-arrière »

- **Branchement** : Après simplification, affecter à **vrai** une variable choisit heuristiquement.
- Continuer récursivement l'algorithme.
- **Retour-arrière** : Si on arrive à une contradiction, on retourne au dernier choix, et on « branche » en affectant **faux** à la variable choisie.

Suppression des clauses qui ont des littéraux isolés.

Définition Littéral L **isolé**

Si aucune clause de Γ ne comporte de littéral complémentaire de L, noté L^c.

Lemme

Supprimer des clauses avec un littéral isolé préserve la satisfaisabilité.

Exemple

Soit Γ l'ensemble de clauses

(1) $p + q + r$

(2) $\bar{q} + \bar{r}$

(3) $q + s + p$

(4) $\bar{s} + t$

Simplifiez Γ en supprimant des clauses qui ont des littéraux isolés.

Les littéraux p et t sont isolés.
Donc on obtient

(2) $\bar{q} + \bar{r}$

(3) $q + s$

Les littéraux $\bar{r}$ et s sont isolés.
On obtient l'ensemble vide.
D'après le lemme, Γ a un modèle $p = 1, t = 1, r = 0, s = 1$.
Mais il existe des contre-modèle, *e.g.* $p = 0, q = 0, r = 0$! ! !

Résolution unitaire

Definition

Une clause unitaire est une clause qui ne comporte qu'un littéral.

Lemme

Soit L l'ensemble des littéraux des clauses unitaires de Γ. Soit Θ l'ensemble de clauses ainsi obtenu à partir de Γ :

- si L comporte deux littéraux complémentaires, alors $\Theta = \{\bot\}$.
- sinon Θ est obtenue ainsi
  - enlever les clauses qui comportent un élément de L
  - à l'intérieur des clauses restantes enlever les littéraux complémentaires des éléments de L

Γ a un modèle si et seulement si Θ en a un.

Exemple Résolution unitaire

Simplifiez les ensembles de clauses suivants par résolution unitaire :

- **Soit Γ l'ensemble de clauses : $p + q, \bar{p}, \bar{q}$**

  $\perp$ par la résolution unitaire, donc Γ n'a pas de modèle.

- **Soit Γ l'ensemble de clauses : $a + b + \bar{d}, \bar{a} + c + \bar{d}, \bar{b}, d, \bar{c}$.**

  1. $a, \bar{a}$.
  2. Clause vide.

  donc Γ n'a pas de modèle.

- **Soit Γ' l'ensemble de clauses : $p, q, p + r, \bar{p} + r, q + \bar{r}, \bar{q} + s$.**

  Par résolution unitaire, on obtient : r, s.
  Cet ensemble de clauses a un modèle donc Γ' en a un.

Suppression de clauses valides

Lemme

Soit Θ l'ensemble de clauses obtenu en supprimant les clauses valides de Γ.

$$\Gamma \text{ a un modèle ssi } \Theta \text{ en a un.}$$

Preuve

- Supposons que Γ a un modèle v, comme Θ est un sous-ensemble des clauses de Γ, v est aussi modèle de Θ. Donc, Θ a un modèle.
- Supposons que Θ a un modèle v. Soit v' une assignation de Γ telle que $v'(x) = v(x)$ pour toute variable x présente à la fois dans Γ et Θ. Soit C une clause de Γ. Si C est aussi une clause de Θ, alors v' est un modèle de C car v et v' donnent la même valeur à C. Si C n'est pas une clause de Θ, alors C est valide, en conséquence toute assignation, v' en particulier, est modèle de C. D'où, Γ a un modèle : v'.

L'algorithme DPLL

bool fonction Algo_DPLL(Γ : ensemble de clauses)

0 Supprimer les clauses valides de Γ.
Si $\Gamma = \emptyset$, retourner (`vrai`).
Sinon retourner (`DPLL`(Γ))

bool fonction `DPLL`(Γ : ensemble de clauses non valides)

La fonction retourne vrai si et seulement si Γ est satisfaisable

1 **Si** $\perp \in \Gamma$, retourner(`faux`).
Si $\Gamma = \emptyset$, retourner (`vrai`).

2 Réduire Γ : il suffit d'enlever toute clause contenant une *autre* clause.

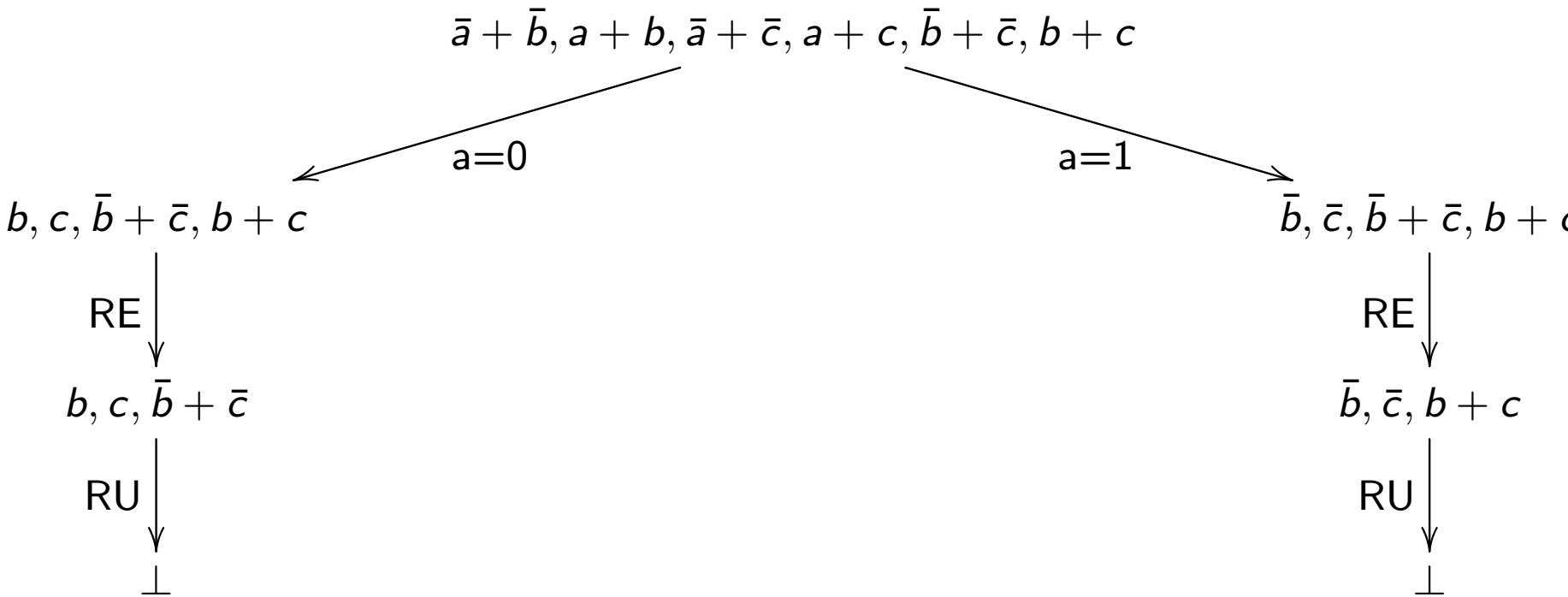
3 Enlever de Γ les clauses comportant des littéraux isolés.
Si l'ensemble Γ a été modifié, aller en 1.

4 Appliquer à Γ la résolution unitaire.
Si l'ensemble Γ a été modifié, aller en 1.

5 Choisir x une variable quelconque de Γ
retourner (`DPLL`($\Gamma[x := 0]$) ou alors `DPLL`($\Gamma[x := 1]$))

Exemple

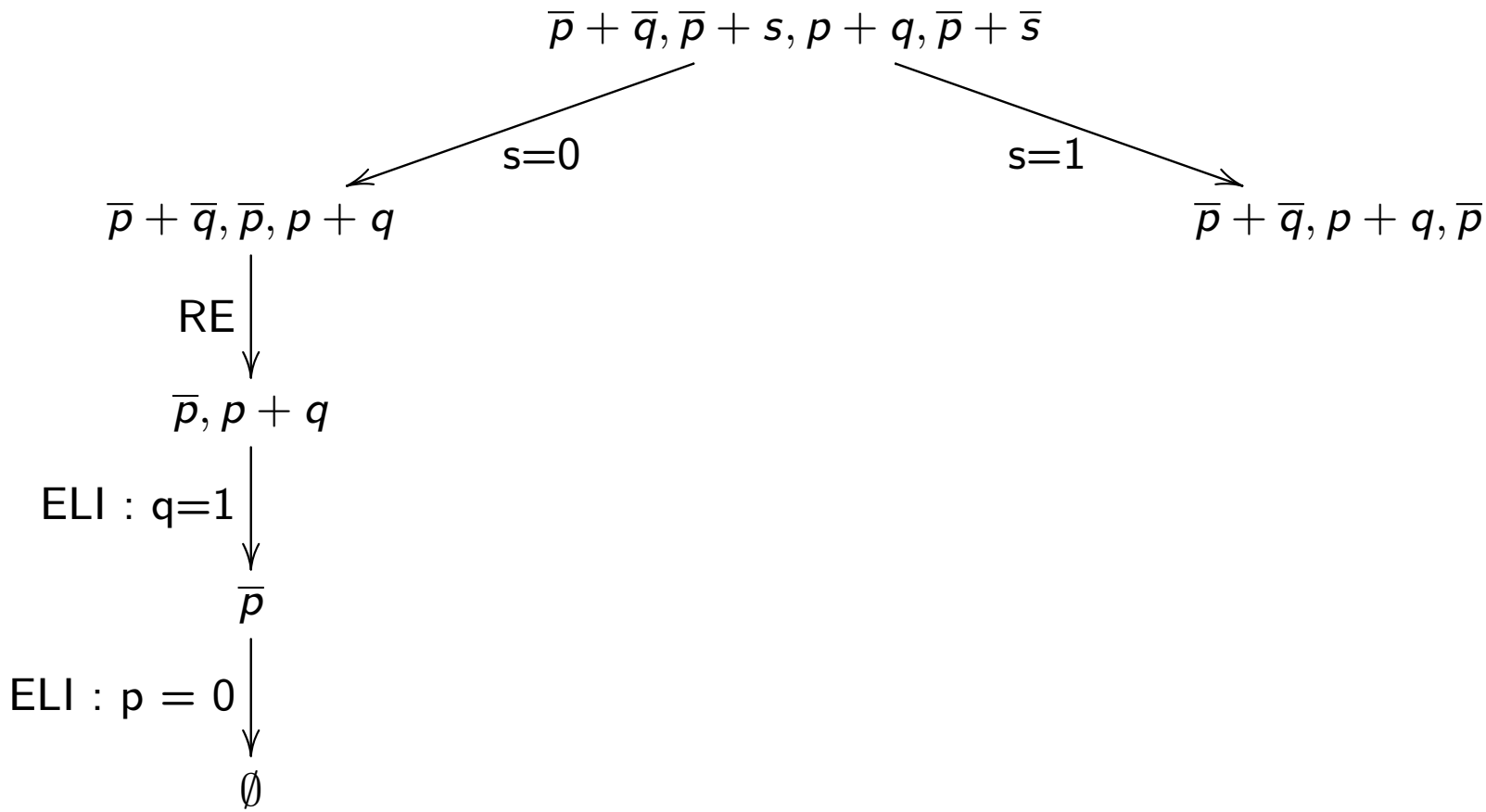
Soit Γ l'ensemble de clauses : $\bar{a}+\bar{b}, a+b, \bar{a}+\bar{c}, a+c, \bar{b}+\bar{c}, b+c$.



Puisque toutes les feuilles portent la clause vide, l'ensemble Γ est insatisfaisable. RE = Reduction

Exemple

Soit Γ l'ensemble de clauses : $\overline{p} + \overline{q}, \overline{p} + s, p + q, \overline{p} + \overline{s}$.



Puisqu'une feuille porte l'ensemble vide, l'ensemble Γ est satisfaisable. Il est inutile de poursuivre la construction de la branche droite.

Théorème

L'algorithme Algo_DPLL est correct et se termine.

Preuve de terminaison

- Le pas 0 n'est exécuté qu'une seule fois.
- Itération en 1 : le nombre de clauses diminue strictement, donc terminaison.
- Récursivité en 5 : le nombre de variables diminue strictement, donc terminaison.

Rappel de la propriété : Γ a un modèle ssi $\Gamma[x := 0]$ est satisfaisable ou $\Gamma[x := 1]$ est satisfaisable.

Preuve de correction

Invariant :

la valeur courante de Γ a un modèle ssi Γ a un modèle.

Vérifié au pas 0, 1 et 5 , donc réponses correctes. Supposons les appels récursifs corrects :

- si DPLL($\Gamma[x := 0]$) est vrai, alors par récurrence $\Gamma[x := 0]$ est satisfaisable donc Γ est satisfaisable, d'après la propriété. ce qui correspond à la valeur vrai de DPLL(Γ).
- si DPLL($\Gamma[x := 0]$) est faux, alors par récurrence $\Gamma[x := 0]$ est insatisfaisable. Dans ce cas, DPLL(Γ) vaut DPLL($\Gamma[x := 1]$) :
  - Supposons que DPLL($\Gamma[x := 1]$) est vrai, alors par récurrence $\Gamma[x := 1]$ est satisfaisable donc Γ est satisfaisable, ce qui correspond à la valeur vrai de DPLL(Γ).
  - Supposons que DPLL($\Gamma[x := 1]$) est faux, alors par récurrence $\Gamma[x := 1]$ est insatisfaisable. Donc Γ est insatisfaisable, ce qui correspond à la valeur faux de DPLL(Γ).

Remarques

- **Oubli de simplifications :** DPLL reste correcte si on oublie la réduction (2), enlèvement des littéraux isolés (3) et/ou réduction unitaire (4).
- **Choix de la variable :**
  - Un bon choix pour la variable x de l'étape (5), consiste à choisir la variable qui apparait le plus souvent.
  - Un meilleur choix consiste à choisir la variable qui va entrainer par la suite le plus de simplifications

Aperçu de la logique du premier ordre

Un domaine : les *objets* sur lesquels on raisonne

Trois catégories :

- les termes qui représentent des éléments du domaine
- les relations entre éléments du domaine
- les formules qui décrivent les interactions entre les relations

Deux nouveaux symboles (quantificateurs) dans les formules

$\forall$ (quantificateur universel) et $\exists$ (quantificateur existentiel)

Exemples :

- domaine = membres d'une famille
- le terme $pere(x)$ désigne un élément du domaine (le père de x)
- la relation *frere* détermine si deux éléments sont frères
- la formule $\forall x \exists y \; frere(y, x)$ signifie "tout individu a un frère".

Syllogisme

Tous les hommes sont mortels.
Socrate est un homme.
Donc Socrate est mortel.

$$\forall x(\mathit{homme}(x) \Rightarrow \mathit{mortel}(x))$$
$$\mathit{homme}(\mathit{Socrate})$$
$$\mathit{mortel}(\mathit{Socrate})$$

Quelques exemples

Traduire en logique du premier ordre :

- Il y a des gens qui s'aiment.

  $\exists x\ \exists y\ \ (a(x,y) \wedge a(y,x))$

- Si deux personnes s'aiment l'une l'autre, alors elles sont conjointes.

  $\forall x\ \forall y\ \ (a(x,y) \wedge a(y,x) \Rightarrow c(x) = y \wedge c(y) = x)$

- On ne peut pas aimer deux personnes à la fois.

  $\forall x\ \forall y\ \ (a(x,y) \Rightarrow \forall z\ \ (a(x,z) \Rightarrow y = z))$

  $\forall x\ \forall y\ \forall z\ \ (a(x,y) \wedge a(x,z) \Rightarrow y = z)$

Rappels

- D est un domaine non vide.
- I est une interprétation des symboles de la formule en tant que
  1. constantes ($\in D$)
  2. fonctions ($D^n \to D$), n est l'arité de la fonction
  3. variables propositionnelles ($\in \{0, 1\}$)
  4. relations ($\subseteq D^n$).
- e est un état des variables *libres* de la formule, qui associe à chacune un élément du domaine D.

Exemple

Considérons la signature suivante.

- $Anne^{f0}$, $Bernard^{f0}$ et $Claude^{f0}$
- a^{r2} ($a(x, y)$ signifie « x aime y »)
- c^{f1} ($c(x)$ désigne le conjoint de x).

Soit I l'interprétation de domaine $D = \{0, 1, 2\}$ où :

- $Anne_I^{f0} = 0$, $Bernard_I^{f0} = 1$, et $Claude_I^{f0} = 2$.
- $a_I^{r2} = \{(0, 1), (1, 0), (2, 0)\}$
- $c_I^{f1}(0) = 1$, $c_I^{f1}(1) = 0$, $c_I^{f1}(2) = 2$.

  $c_I^{f1}(2)$ est défini artificiellement : Claude n'a pas de conjoint.

Sens des formules

1. Les connecteurs propositionnels ont le même sens qu'en logique propositionnelle.
2. Notons $e[x = d]$ l'état identique à l'état e, sauf pour x.

$$[\forall x B]_{(I,e)} = min_{d \in D}[B]_{(I,e[x=d])} = \prod_{d \in D}[B]_{(I,e[x=d])},$$

   vrai si $[B]_{(I,f)} = 1$ pour tout état f identique à e, sauf pour x.

3.

$$[\exists x B]_{(I,e)} = max_{d \in D}[B]_{(I,e[x=d])} = \sum_{d \in D}[B]_{(I,e[x=d])},$$

   vrai s'il y a un état f identique à e, sauf pour x, tel que $[B]_{(I,f)} = 1$.

Exemple

Utilisons l'interprétation I donnée dans l'exemple.
(Rappel $D = \{0, 1, 2\}$)

- $[\exists x\ a(x, x)]_I$

$$= \max\{[a(0,0)]_I, [a(1,1)]_I, [a(2,2)]_I\} = \mathit{faux}$$
$$= [a(0,0)]_I + [a(1,1)]_I + [a(2,2)]_I = \mathit{faux} + \mathit{faux} + \mathit{faux} = \mathit{faux}.$$

- $[\forall x \exists y\ a(x, y)]_I$

$$= \min\{\max\{[a(0,0)]_I,\ [a(0,1)]_I,\ [a(0,2)]_I\},$$
$$\max\{[a(1,0)]_I,\ [a(1,1)]_I,\ [a(1,2)]_I\},$$
$$\max\{[a(2,0)]_I,\ [a(2,1)]_I,\ [a(2,2)]_I\}\}$$

$$= \min\{\max\{\mathit{faux}, \mathit{vrai}, \mathit{faux}\},\ \max\{\mathit{vrai}, \mathit{faux}, \mathit{faux}\},$$
$$\max\{\mathit{vrai}, \mathit{faux}, \mathit{faux}\}\}$$

$$= \min\{\mathit{vrai}, \mathit{vrai}, \mathit{vrai}\} = \mathit{vrai}.$$

Exemple

- $[\exists y \forall x\ a(x,y)]_I$

$$
\begin{aligned}
&= [a(0,0)]_I.[a(1,0)]_I.[a(2,0)]_I + [a(0,1)]_I.[a(1,1)]_I.[a(2,1)]_I \\
&\quad + [a(0,2)]_I.\ [a(1,2)]_I.\ [a(2,2)]_I \\
&= \mathit{faux}.\mathit{vrai}.\mathit{vrai} + \mathit{vrai}.\mathit{faux}.\mathit{faux} + \mathit{faux}.\mathit{faux}.\mathit{faux} \\
&= \mathit{faux} + \mathit{faux} + \mathit{faux} = \mathit{faux}.
\end{aligned}
$$

Remarque

Les formules $\forall x \exists y\ a(x,y)$ et $\exists y \forall x\ a(x,y)$ n'ont pas la même valeur.
En intervertissant un $\exists$ et un $\forall$, on ne préserve pas le sens des formules.

Modèle, validité, conséquence, équivalence

Ces notions sont définies comme en logique propositionnelle mais...

Pour donner une valeur à une formule

- **En logique propositionnelle** : assignation $V \to \{0, 1\}$
- **En logique du premier ordre** : (I, e) où
  - I est une interprétation des symboles
  - e un état des variables.

... on utilise une interprétation au lieu d'une assignation.

La valeur d'une formule ne dépend que :

- de l'état de ses variables libres
- et de l'interprétation de ses symboles.

Idée

- Le sens de la formule $x + 2 = 4$ dépend de x
  x est libre dans cette formule
- $\forall x(x + 2 = 4)$ est fausse
  $\forall x(x + 0 = x)$ est vraie
  il n'y a pas à choisir de valeur pour x afin de déterminer leur valeur respective
  ces deux formules n'ont pas de variables libres

Occurences libres et liées

Definition

- Dans $\forall x\ A$ ou $\exists x\ A$, la portée de la liaison pour x est A.
- Une occurrence de x dans A est libre si elle n'est pas dans la portée d'une liaison pour x, sinon elle est dite liée

Si nous représentons une formule par un arbre :

- Une occurrence liée de x est

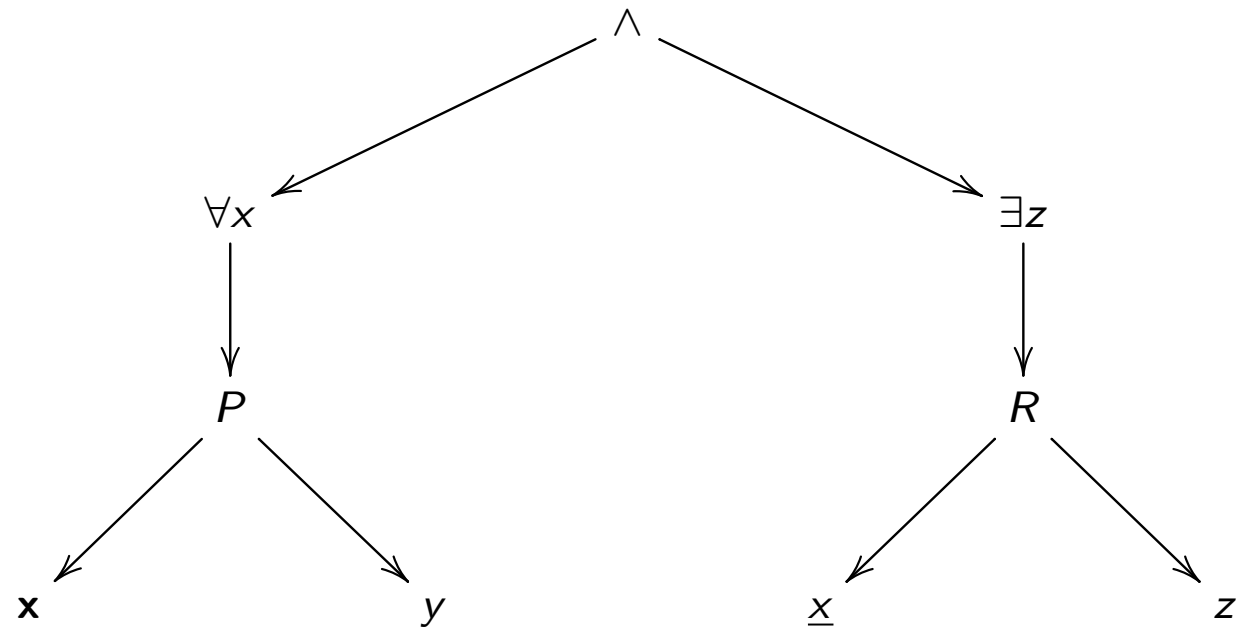
  en dessous d'un sommet $\exists x$ ou $\forall x$.

- Une occurrence de x est libre si

  elle n'est pas sous un tel sommet.

Exemple

$\forall x P(\mathbf{x}, y) \wedge \exists z R(\underline{x}, z)$



- L'occurrence en gras de x est liée.
- L'occurrence soulignée de x est libre.
- L'occurrence de z est liée.

Variables libres, liées

Definition

- La variable x est une variable libre d'une formule si et seulement s'il y a une occurrence libre de x dans la formule.
- Une variable x est une variable liée d'une formule si et seulement s'il y a une occurrence liée de x dans la formule.
- Une formule sans variable libre est aussi appelée une formule fermée.

remarque

Une variable peut-être à la fois libre et liée. Par exemple, dans la formule $\forall x P(x) \vee Q(x)$, x est à la fois libre et liée.

remarque

Par définition, une variable qui n'apparaît pas dans une formule (0 occurrence) est une variable **NON** libre de la formule.

Exemple

Les variables libres de la formule de l'exemple sont x et y.

Déclaration de symbole

Definition

Une déclaration de symbole est un triplet noté s^{gn} où :

- s est un symbole
- g une des lettres f (signifiant fonction) ou r (signifiant relation)
- n est un entier naturel.

Remarque

Si le contexte donne une déclaration implicite d'un symbole, nous omettons l'exposant.

Exemple : **égal** est toujours une relation à 2 arguments, on abrége la déclation $=^{r2}$ en $=$.

Déclaration de symbole : Exemple

Exemple

- $parent^{r2}$ est une **relation** (**r**) avec **2** arguments
- $*^{f2}$ est **fonction** (**f**) avec **2** arguments
- $homme^{r1}$ une **relation** unaire

Signature

Definition

Une signature est un ensemble de déclarations de symboles.

Soit $n > 0$ et Σ une signature le symbole s est :

1. une constante de la signature si et seulement si $s^{f0} \in \Sigma$
2. un symbole de fonction à n arguments de la signature, si et seulement si $s^{fn} \in \Sigma$
3. une variable propositionnelle de la signature si et seulement si $s^{r0} \in \Sigma$
4. un symbole de relation à n arguments de la signature, si et seulement si $s^{rn} \in \Sigma$

Exemples en mathématique (1/2)

$0^{f0}, 1^{f0}, +^{f2}, -^{f2}, *^{f2}, =^{r2}$ est une signature pour l'arithmétique.

Remarque :

- On écrit : 0, 1, $+$ et $-$ (à deux arguments), $*$ et $=$.
- Précisons que le plus et le moins ont deux arguments (nous n'utiliserons pas le symbole plus avec un seul argument).

Exemples en mathématique (2/2)

Exemple

théorie des ensembles Une signature possible est $\in, =$

les autres opérations peuvent être définies à partir de ces symboles.

Surcharge

Definition

Un symbole est surchargé dans une signature, lorsque cette signature comporte deux déclarations distinctes du même symbole.

Exemple : le signe moins est souvent surchargé.

- l'opposé d'un nombre
- la soustraction de deux nombres

Dans la suite de ce cours, nous nous interdirons d'utiliser des signatures comportant des symboles surchargés.

Terme sur une signature

Definition

Soit Σ une signature, un terme sur Σ est :

- soit une variable,
- soit une constante s où $s^{f0} \in \Sigma$,
- soit un terme de la forme $s(t_1, \ldots, t_n)$, où $n \geq 1$, $s^{fn} \in \Sigma$ et $t_1, \ldots, t_n$ sont des termes sur Σ.

L'ensemble des termes sur la signature Σ est noté T_Σ.

Formule atomique sur une signature

Definition

Soit Σ une signature, une formule atomique sur Σ est :

- soit une des constantes $\top, \bot$,
- soit une variable propositionnelle s où $s^{r0} \in \Sigma$,
- soit de la forme $s(t_1, \ldots, t_n)$ où $n \geq 1$, $s^{rn} \in \Sigma$ et où $t_1, \ldots, t_n$ sont des termes sur Σ.

Formule sur une signature

Definition

Une formule sur une signature Σ est une formule, dont les sous-formules atomiques sont des formules atomiques sur Σ (au sens de la définition.

Nous dénotons l'ensemble des formules sur la signature Σ par F_Σ.

Exemple

$\forall x\ (p(x) \Rightarrow \exists y\ q(x, y))$ est une formule sur la signature $\Sigma = \{p^{r1}, q^{r2}, h^{f1}, c^{f0}\}$.

Mais c'est aussi une formule sur la signature $\Sigma' = \{p^{r1}, q^{r2}\}$, puisque les symboles h et c ne figurent pas dans la formule.

Signature associée

Definition

La signature associée à une formule est la plus petite signature Σ telle que la formule est élément de F_Σ, c'est la plus petite signature permettant d'écrire la formule.

Exemple

La signature associée à la formule $\forall x\ (p(x) \Rightarrow \exists y\ q(x, y))$ est

p^{r1}, q^{r2}.

Signature associée

Definition

La signature associée à un ensemble de formules est l'union des signatures associées à chaque formule de l'ensemble.

Exemple

La signature associée à l'ensemble constitué des deux formules $\forall x(p(x) \Rightarrow \exists y\ q(x,y)), \forall u\ \forall v\ (u + s(v) = s(u) + v)$ est

$\Sigma = \{p^{r1}, q^{r2}, +^{f2}, s^{f1}, =^{r2}\}$.

Interprétation

Definition

Une interprétation I sur une signature Σ est définie par un domaine D non vide et une application qui à chaque déclaration de symbole $s^{gn} \in \Sigma$ associe sa valeur s_I^{gn} comme suit :

1. s_I^{f0} est un élément de D.
2. s_I^{fn} où $n \geq 1$ est une fonction de D^n dans D, autrement dit une fonction à n arguments.
3. s_I^{r0} vaut 0 ou 1.
4. s_I^{rn} où $n \geq 1$ est un sous-ensemble de D^n, autrement dit une relation à n arguments.

Exemple

Soit l'interprétation I de domaine $D = \{1, 2, 3\}$ où la relation binaire *ami* est vraie pour les couples $(1, 2)$, $(1, 3)$ et $(2, 3)$, c'est-à-dire,
$ami_I^{r2} = \{(1, 2), (1, 3), (2, 3)\}$.

$ami(2, 3)$ est vraie dans l'interprétation I. En revanche, $ami(2, 1)$ est fausse dans l'interprétation I.

Remarque

Dans toute interprétation I, la valeur du symbole $=$ est l'ensemble $\{(d, d) \mid d \in D\}$, autrement dit dans toute interprétation le sens de l'égalité est l'identité sur le domaine de l'interprétation.

Exemple

Considérons une signature suivante.

- $Anne^{f0}$, $Bernard^{f0}$ et $Claude^{f0}$: les prénoms Anne, Bernard, et Claude dénotent des constantes,
- a^{r2} : la lettre a dénote une relation à deux arguments (nous lisons $a(x, y)$ comme x aime y) et
- c^{f1} : la lettre c dénote une fonction à un argument (nous lisons $c(x)$ comme le copain ou la copine de x).

Une interprétation possible sur cette signature est l'interprétation I de domaine $D = \{0, 1, 2\}$ où :

- $Anne_I^{f0} = 0$, $Bernard_I^{f0} = 1$, et $Claude_I^{f0} = 2$.
- $a_I^{r2} = \{(0, 1), (1, 0), (2, 0)\}$.
- $c_I^{f1}(0) = 1, c_I^{f1}(1) = 0, c_I^{f1}(2) = 2$. Notons que la fonction c_I^{f1} a comme domaine D, ce qui oblige à définir artificiellement $c_I^{f1}(2)$: Claude, dénoté par 2, n'a ni copain, ni copine.

Interprétation d'un ensemble de formules

Definition

L'interprétation d'un ensemble de formules est une interprétation qui définit seulement le sens de la signature associée à l'ensemble des formules.

Etat, assignation

Definition

Un état e d'une interprétation est une application de l'ensemble des variables dans le domaine de l'interprétation.

Definition

Une assignation est un couple (I, e) composé d'une interprétation I et d'un état e.

Exemple

Soient le domaine $D = \{1, 2, 3\}$ et l'interprétation I où la relation binaire *ami* est vraie uniquement pour les couples $(1, 2)$, $(1, 3)$ et $(2, 3)$, c'est-à-dire,
$ami_I^{r2} = \{(1, 2), (1, 3), (2, 3)\}$.

Soit e l'état qui associe 2 à x et 1 à y.

L'assignation (I, e) rend la relation $ami(x, y)$ fausse.

Remarque

La valeur d'une formule ne dépend que de ses variables libres et de ses symboles, aussi pour évaluer une formule sans variable libre, l'état des variables est inutile. Nous avons alors deux possibilités :

- Pour une formule sans variables libres, il suffit de donner une interprétation I des symboles de la formule. Dans ce cas, les assignations (I, e) et (I, e') donnerons la même valeur à la formule pour tous états e et e'. Ainsi pour tout état e, nous identifierons (I, e) et I. Selon le contexte, I sera considéré comme soit une interprétation soit une assignation dont l'état est quelconque.
- Pour une formule avec des variables libres, nous avons donc besoin d'une assignation.

Termes

Définition

Évaluation Nous donnons la définition inductive de l'évaluation d'un terme t :

1. si t est une variable, alors $[\![t]\!]_{(I,e)} = e(t)$,
2. si t est une constante alors $[\![t]\!]_{(I,e)} = t_I^{f0}$,
3. si $t = s(t_1, \ldots, t_n)$ où s est un symbole et $t_1, \ldots, t_n$ sont des termes, alors $[\![t]\!]_{(I,e)} = s_I^{fn}([\![t_1]\!]_{(I,e)}, \ldots, [\![t_n]\!]_{(I,e)})$.

Exemple

Soit I l'interprétation de domaine $\mathbb{N}$ qui donne aux déclarations de symboles $1^{f0}, *^{f2}, +^{f2}$ leur sens usuel sur les entiers.
Soit e l'état tel que $x = 2, y = 3$.
Calculons $[\![x * (y + 1)]\!]_{(I,e)}$.

$$[\![x * (y + 1)]\!]_{(I,e)} = [\![x]\!]_{(I,e)} * [\![(y + 1)]\!]_{(I,e)} =$$
$$[\![x]\!]_{(I,e)} * ([\![y]\!]_{(I,e)} + [\![1]\!]_{(I,e)}) = e(x) * (e(y) + 1) = 2 * (3 + 1) = 8.$$

Formules

Définition

Sens des formules atomiques Le sens des formules atomiques est donné par les règles inductives suivantes :

1. $[\top]_{(I,e)} = 1, [\bot]_{(I,e)} = 0$. Dans les exemples, nous autorisons à remplacer $\top$ par sa valeur 1 et $\bot$ par sa valeur 0.
2. Soit s une variable propositionnelle, $[s]_{(I,e)} = s_I^{r0}$.
3. Soit $A = s(t_1, \ldots, t_n)$ où s est un symbole et $t_1, \ldots, t_n$ sont des termes. Si $([\![t_1]\!]_{(I,e)}, \ldots, [\![t_n]\!]_{(I,e)}) \in s_I^{rn}$ alors $[A]_{(I,e)} = 1$ sinon $[A]_{(I,e)} = 0$.

Exemple

Considérons une signature suivante.

- $Anne^{f0}$, $Bernard^{f0}$ et $Claude^{f0}$: les prénoms Anne, Bernard, et Claude dénotent des constantes,
- a^{r2} : la lettre a dénote une relation à deux arguments (nous lisons $a(x, y)$ comme x aime y) et
- c^{f1} : la lettre c dénote une fonction à un argument (nous lisons $c(x)$ comme le copain ou la copine de x).

Soit I l'interprétation de domaine $D = \{0, 1, 2\}$ sur cette signature où :

- $Anne_I^{f0} = 0$, $Bernard_I^{f0} = 1$, et $Claude_I^{f0} = 2$.
- $a_I^{r2} = \{(0, 1), (1, 0), (2, 0)\}$.
- $c_I^{f1}(0) = 1, c_I^{f1}(1) = 0, c_I^{f1}(2) = 2$. Notons que la fonction c_I^{f1} a comme domaine D, ce qui oblige à définir artificiellement $c_I^{f1}(2)$: Claude, dénoté par 2, n'a ni copain, ni copine.

Exemple

Nous obtenons :

- $[a(Anne, Bernard)]_I =$

  1 car $(\llbracket Anne \rrbracket_I, \llbracket Bernard \rrbracket_I) = (0, 1) \in a_I^{r2}$.

- $[a(Anne, Claude)]_I =$

  0 car $(\llbracket Anne \rrbracket_I, \llbracket Claude \rrbracket_I) = (0, 2) \notin a_I^{r2}$.

Exemple

Soit e l'état $x = 0, y = 2$. Nous avons :

- $[a(x, c(x))]_{(I,e)} =$

  1 car $([\![x]\!]_{(I,e)}, [\![c(x)]\!]_{(I,e)}) = (0, c_I^{f1}([\![x]\!]_{(I,e)})) = (0, c_I^{f1}(0)) = (0, 1) \in a_I^{r2}$.

- $[a(y, c(y))]_{(I,e)} =$

  0 car $([\![y]\!]_{(I,e)}, [\![c(y)]\!]_{(I,e)}) = (2, c_I^{f1}([\![y]\!]_{(I,e)})) = (2, c_I^{f1}(2)) = (2, 2) \notin a_I^{r2}$.

Attention à distinguer (suivant le contexte), les éléments du domaine $0, 1$ et les valeurs de vérité $0, 1$.

Exemple

Nous avons :

- $[(Anne = Bernard)]_I =$

  0, car $([\![Anne]\!]_I, [\![Bernard]\!]_I) = (0,1) \not\in =_I^{r2}$.

- $[(c(Anne) = Anne)]_I =$

  0, car $([\![c(Anne)]\!]_I, [\![Anne]\!]_I) = (c_I^{f1}([\![Anne]\!]_I), 0) = (c_I^{f1}(0), 0) = (1,0) \not\in =_I^{r2}$.

- $[(c(c(Anne)) = Anne)]_I =$

  1, car $([\![c(c(Anne))]\!]_I, [\![Anne]\!]_I) = (c_I^{f1}([\![c(Anne)]\!]_I), 0) = (c_I^{f1}(c_I^{f1}(0)), 0) = (c_I^{f1}(1), 0) = (0,0) \in =_I^{r2}$.

Sens des formules

1. Les connecteurs propositionnels ont le même sens qu'en logique propositionnelle.
2. Soient x une variable et B une formule. $[\forall x B]_{(I,e)} = 1$ si et seulement si $[B]_{(I,f)} = 1$ pour tout état f identique à e, sauf pour x. Soit $d \in D$. Notons $e[x = d]$ l'état identique à l'état e, sauf pour la variable x, auquel l'état $e[x = d]$ associe la valeur d. La définition ci-dessus peut être mise sous la forme suivante :

$$[\forall x B]_{(I,e)} = min_{d \in D} [B]_{(I,e[x=d])} = \prod_{d \in D} [B]_{(I,e[x=d])},$$

   où le produit est le produit booléen.
3. $[\exists x B]_{(I,e)} = 1$ si et seulement s'il y a un état f identique à e, sauf pour x, tel que $[B]_{(I,f)} = 1$. La définition ci-dessus peut être mise sous la forme suivante :

$$[\exists x B]_{(I,e)} = max_{d \in D} [B]_{(I,e[x=d])} = \sum_{d \in D} [B]_{(I,e[x=d])},$$

   où la somme est la somme booléenne.

Exemple

Soit l'interprétation I de domaine $D = \{1, 2, 3\}$ où la relation binaire *ami* est vraie pour les couples $(1, 2)$, $(1, 3)$ et $(2, 3)$, c'est-à-dire,
$ami_I^{r2} = \{(1, 2), (1, 3), (2, 3)\}$.

La formule $ami(1, 2) \wedge ami(2, 3) \Rightarrow ami(1, 3)$ est vraie dans l'interprétation I, *i.e.*, $[ami(1, 2) \wedge ami(2, 3) \Rightarrow ami(1, 3)]_I = 1$.

Exemple

Utilisons l'interprétation I donnée dans l'exemple.
(Rappel $D = \{0, 1, 2\}$)

- $[\exists x\ a(x, x)]_I =$

$\max\{[a(0,0)]_I, [a(1,1)]_I, [a(2,2)]_I\} = 0$ car $(0,0), (1,1), (2,2) \notin a_I^{r2}$.
D'après la définition, nous avons : $[\exists x\ a(x,x)]_I =$
$[a(0,0)]_I + [a(1,1)]_I + [a(2,2)]_I = 0 + 0 + 0 = 0$.

- $[\forall x \exists y\ a(x, y)]_I =$

$\min\{\max\{[a(0,0)]_I, [a(0,1)]_I, [a(0,2)]_I\}, \max\{[a(1,0)]_I, [a(1,1)]_I,$
$[a(1,2)]_I\}, \max\{[a(2,0)]_I, [a(2,1)]_I, [a(2,2)]_I\}\} = \min\{\max\{0,1,0\},$
$\max\{1,0,0\}, \max\{1,0,0\}\} = \min\{1,1,1\} = 1$.

D'après la définition, nous avons : $[\forall x \exists y\ a(x,y)]_I =$
$([a(0,0)]_I + [a(0,1)]_I + [a(0,2)]_I).\ ([a(1,0)]_I + [a(1,1)]_I + [a(1,2)]_I).$
$([a(2,0)]_I + [a(2,1)]_I + [a(2,2)]_I) =$
$(0+1+0).(1+0+0).(1+0+0) = 1.1.1 = 1$.

Exemple

- $[\exists y \forall x\ a(x,y)]_I =$

$\max\{\min\{[a(0,0)]_I, [a(1,0)]_I, [a(2,0)]_I\}, \min\{[a(0,1)]_I, [a(1,1)]_I, [a(2,1)]_I\}, \min\{[a(0,2)]_I, [a(1,2)]_I, [a(2,2)]_I\}\} = \max\{\min\{0,1,1\}, \min\{1,0,0\}, \min\{0,0,0\}\} = \max\{0,0,0\} = 0.$

D'après la définition, nous avons : $[\exists y \forall x\ a(x,y)]_I = [a(0,0)]_I.[a(1,0)]_I.[a(2,0)]_I + [a(0,1)]_I.[a(1,1)]_I.[a(2,1)]_I + [a(0,2)]_I.[a(1,2)]_I.\ [a(2,2)]_I = 0.1.1 + 1.0.0 + 0.0.0 = 0 + 0 + 0 = 0.$

Remarque

Les formules $\forall x \exists y\ a(x,y)$ et $\exists y \forall x\ a(x,y)$ n'ont pas la même valeur. Donc en intervertissant un quantificateur existentiel et un quantificateur universel, nous ne préservons pas le sens des formules.

Modèle, validité, conséquence, équivalence

Ces notions sont définies comme en logique propositionnelle.

Une assignation

- **En logique propositionnelle** : $V \to \{0, 1\}$
- **En logique du premier ordre** : (I, e) où
  - I est une interprétation des symboles
  - e un état des variables.

La valeur d'une formule ne dépend que
de ses variables libres et de ses symboles.

L'état des variables est inutile pour évaluer
une formule sans variables libres.

On utilise une interprétation au lieu d'une assignation.

Instanciation

Definition

Soit x une variable, t un terme et A une formule.

1. $A < x := t >$ est la formule obtenue en remplaçant dans la formule A toute occurrence libre de x par le terme t.
2. Le terme t est libre pour x dans A si les variables de t ne sont pas liées dans les occurrences libres de x.

Instantiation : Exemple

Exemple

- Le terme z est libre pour x dans la formule $\exists y p(x, y)$.
- Par contre le terme y, comme tout terme comportant la variable y, n'est pas libre pour x dans cette formule.
- Par définition, le terme x est libre pour lui-même dans toute formule.
- Soit A la formule $(\forall x P(x) \vee Q(\mathbf{x}))$, la formule $A < x := b >$ vaut

$(\forall x P(x) \vee Q(b))$ car seule l'occurrence en gras de x est libre.

Propriétés

Theoreme

Soient A une formule et t un terme libre pour la variable x dans A.
Soient I une interprétation et e un état de l'interprétation. Nous avons $[A < x := t >]_{(I,e)} = [A]_{(I,e[x=d])}$, où $d = [\![t]\!]_{(I,e)}$.

Corollaire

Soient A une formule et t un terme libre pour x dans A.
Les formules $\forall x A \Rightarrow A < x := t >$ et $A < x := t > \Rightarrow \exists x A$ sont valides.

La condition sur t est nécessaire :

La condition t terme libre est nécessaire dans **le théorème**

Exemple

Soient I l'interprétation de domaine $\{0,1\}$ avec $p_I = \{(0,1)\}$ et e, un état où $y = 0$. Soient A la formule $\exists y\ p(x,y)$ et t le terme y.
Ce terme n'est pas libre pour x dans A

- $A < x := t >=$

$$\exists y\ p(y,y)$$

et $[A < x := t >]_{(I,e)} =$

$[\exists y\ p(y,y)]_{(I,e)} = \max\{[p(0,0)]_{(I,e)}, [p(1,1)]_{(I,e)}\} = max\{0,0\} = 0.$

La condition sur t est nécessaire :

Exemple

- Soit $d = [\![t]\!]_{(I,e)} = [\![y]\!]_{(I,e)} = 0$. Dans l'assignation $(I, e[x = d])$, nous avons $x = 0$. Donc $[A]_{(I,e[x=d])} =$

$[\exists y\ p(x, y)]_{(I,e[x=d])} = \max\{[p(0, 0)]_{(I,e)}, [p(0, 1)]_{(I,e)}\} =$
$max\{0, 1\} = 1.$

Ainsi, $[A < x := t >]_{(I,e)} \neq [A]_{(I,e[x=d])}$, pour $d = [\![t]\!]_{(I,e)}$.

Modèle fini

Définition

Un modèle fini d'une formule fermée est une interprétation de la formule de domaine fini, qui rend vraie la formule.

Remarque

- Le nom des éléments du domaine est sans importance.
- Ainsi pour un modèle avec n éléments, nous utiliserons le domaine des entiers naturels inférieurs à n.

Construire un modèle fini

Idée naïve : Pour savoir si une formule fermée a un modèle de domaine $\{0, \ldots, n-1\}$, il suffit de

- **énumérer** toutes les interprétations possibles de la signature associée à la formule
- **évaluer** la formule pour ces interprétations.

Exemple

Soit $\Sigma = \{a^{f0}, f^{f1}, P^{r2}\}$, plus éventuellement l'égalité de sens fixé.

Sur un domaine à 5 éléments, Σ a $5 \times 5^5 \times 2^{25}$ interprétations !

Cette méthode est inutilisable en pratique.

Logiciel pour construire un modèle fini

MACE

- **traduction** des formules du premier ordre en formules propositionnelles
- **algorithmes performants pour trouver la satisfaisabilité** d'une formule propositionnelle (par exemple des variantes de l'algorithme de DPLL)

`http://www.cs.unm.edu/~mccune/prover9/mace4.pdf`

Méthode pour la recherche d'un modèle fini

Cas simple : Recherche de modèles à n éléments **par réduction au cas propositionnel** pour une formule n'ayant aucun symbole de fonction et aucune constante, sauf des représentations d'entiers inférieurs à n.

Construction du modèle à n éléments

1. suppression des quantificateurs par expansion sur un domaine à n éléments,
2. remplacement des égalités par leur valeur
3. recherche d'une assignation propositionnelle qui soit modèle.

Expansion d'une formule

Definition

Soient A une formule et n un entier. La n-expansion de A est la formule qui consiste à remplacer :

- toute sous-formule de A de la forme $\forall x B$ par la conjonction $(\prod_{i<n} B < x := \underline{i} >)$
- toute sous-formule de A de la forme $\exists x B$ par la disjonction $(\sum_{i<n} B < x := \underline{i} >)$

où $\underline{i}$ est la représentation décimale de l'entier i.

Exemple

La 2-expansion de la formule $\exists x P(x) \Rightarrow \forall x P(x)$ est la formule

$$P(0) + P(1) \Rightarrow P(0).P(1)$$

Propriété de la n-expansion

Theoreme

Soient n un entier et A une formule ne comportant que des représentations d'entiers de valeur inférieure à n.
Soit B la n-expansion de A.
Toute interprétation de domaine $\{0, \ldots, n-1\}$ attribue la même valeur à A et à B.

La condition sur A est nécessaire car si A comporte une représentation d'un entier au moins égal à n, la valeur de cette représentation ne sera pas dans le domaine de l'interprétation.
La preuve du théorème est une récurrence sur la taille des formules.

Idée de la récurrence : élimination d'un quantificateur universel

Rappel : théorème

Soient A une formule et t un terme libre pour la variable x dans A. Soient I une interprétation et e un état de l'interprétation. Nous avons $[A < x := t >]_{(I,e)} = [A]_{(I,e[x=d])}$, où $d = [\![t]\!]_{(I,e)}$.

Soit $\forall x B$ une sous-formule de A. Soient (I, e) une interprétation et un état de domaine $\{0, \ldots, n-1\}$ donnant à la représentation d'un entier, la valeur de l'entier représenté. Par définition :

$$[\forall x B]_{(I,e)} = \prod_{i<n} [B]_{(I,e[x=i])}$$

D'après le théorème et le fait que la valeur de la représentation de l'entier i est i, nous avons :

$$[B]_{(I,e[x=i])} = [B < x := \underline{i} >]_{(I,e)}$$

Par suite : $[\forall x B]_{(I,e)} = \prod_{i<n} [B < x := \underline{i} >]_{(I,e)} = [\prod_{i<n} B < x := \underline{i} >]_{(I,e)}$.

De l'assignation à l'interprétation

Soient n un entier et A une formule fermée, sans quantificateur, sans égalité, sans symbole de fonction, sans constante sauf des représentations d'entiers inférieurs à n. Soit P l'ensemble des formules atomiques de A (sauf $\top$ et $\bot$ dont le sens est fixé).

Theoreme

Soit v une assignation propositionnelle de P dans $\{0, 1\}$ alors il existe une interprétation I de A telle que $[A]_I = [A]_v$.

Cf. poly

Exemple

Soit v l'assignation définie par $p(0) = 1, p(1) = 0$.

v donne la valeur 0 à la formule $(p(0) + p(1)) \Rightarrow (p(0).p(1))$.

Donc l'interprétation I définie par $p_I = \{0\}$ donne aussi la valeur 0 à cette même formule.

Cet exemple montre que v et I sont deux façons analogues de présenter une interprétation, la deuxième étant souvent plus concise.

De l'interprétation à l'assignation

Soient n un entier et A une formule fermée, sans quantificateur, sans égalité, sans symbole de fonction, sans constante sauf des représentations d'entiers inférieurs à n. Soit P l'ensemble des formules atomiques de A (sauf $\top$ et $\bot$ dont le sens est fixé).

Theoreme

Soit I une interprétation de A alors il existe une assignation v de P telle que

$$[A]_I = [A]_v.$$

Cf. poly

Recherche d'un modèle fini d'une formule fermée **sans** symbole de fonction

Procédure sous les mêmes hypothèses.

1. Remplacer A par sa n-expansion B
2. Dans B,
   - remplacer les égalités par leur valeur, cad $\underline{i} = \underline{j}$ est remplacée par 0 si $i \neq j$ et par 1 si $i = j$.
   - Simplifier par les identités $x \vee 0 = x, x \vee 1 = 1, x \wedge 0 = 0, x \wedge 1 = x$.

   Soit C la formule ainsi obtenue.
3. Chercher une assignation propositionnelle v des formules atomiques de C, qui soit modèle de C :
   - si une telle assignation n'existe pas, A n'a pas de modèle
   - sinon l'interprétation I déduite de v est modèle de A.

Correction de la méthode

1. Supposons qu'il n'a pas d'assignation propositionnelle modèle de C, mais que A ait un modèle I.
   - D'après le théorème, I est modèle de B, donc de C.
   - D'après le théorème, il y a une assignation propositionnelle modèle de C.

   De cette contradiction, nous déduisons que A n'a pas de modèle à n éléments.
2. Supposons l'existence d'une assignation propositionnelle v des formules atomiques de C qui soit modèle de C.
   Donc, l'interprétation I construite comme il est indiqué dans le théorème est modèle de C.
   Donc elle est modèle de B.
   Donc d'après le théorème, elle est modèle de A.

Exemple

$A = \exists x P(x) \wedge \exists x \neg P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y))$

A n'a pas modèle à un élément, car nous avons P et sa négation.

2-expansion de A

$(P(0) + P(1)).(\overline{P(0)} + \overline{P(1)}).(P(0).P(0) \Rightarrow 0 = 0).$
$(P(0).P(1) \Rightarrow 0 = 1).(P(1).P(0) \Rightarrow 1 = 0).(P(1).P(1) \Rightarrow 1 = 1).$

En remplaçant les égalités par leur valeurs

$(P(0) + P(1)).(\overline{P(0)} + \overline{P(1)}).(P(0).P(0) \Rightarrow 1).$
$(P(0).P(1) \Rightarrow 0).(P(1).P(0) \Rightarrow 0).(P(1).P(1) \Rightarrow 1).$
Ce qui ce simplifie en :

$$(P(0) + P(1)).(\overline{P(0)} + \overline{P(1)})$$

L'assignation $P(0) = 1, P(1) = 0$ est un des modèles propositionnels de la formule ci-dessus, donc l'interprétation I de domaine $\{0, 1\}$ où $P_I = \{0\}$ est modèle de A.

Recherche d'un modèle fini d'une formule fermée **avec** symbole de fonction

Soit A une formule fermée pouvant comporter des représentations d'entiers de valeur inférieure à n.

Procédure

- Remplacer A par son expansion.
- Supprimer les égalités.
- Enumérer les choix des valeurs des symboles, en propageant le plus possible chacun des choix effectués.

Similaire à *l'algorithme de* `DPLL`.

Exemple : $A = \exists y P(y) \Rightarrow P(a)$

Chercher un contre-modèle à 2 éléments

2-expansion de A

$$P(0) + P(1) \Rightarrow P(a)$$

Trouver les valeurs de $P(0)$, $P(1)$, a. On choisit (arbitrairement) $a = 0$.

$$P(0) + P(1) \Rightarrow P(0)$$

$P(0) = 0, P(1) = 1$ est un contre-modèle propositionnel, cad une interprétation telle que $P = \{1\}$.

Un contre-modèle est l'interprétation de domaine $\{0, 1\}$ telle que $P = \{1\}$ et $a = 0$.

Exemple : $P(a), \forall x(P(x) \Rightarrow P(f(x))), \neg P(f(b))$

1. 2-expansion :
   $F = \{P(a), (P(0) \Rightarrow P(f(0))).(P(1) \Rightarrow P(f(1))), \neg P(f(b))\}$.
2. Trouver les valeurs de $P(0)$, $P(1)$, a, b, $f(0)$ et $f(1)$ modèle de F. $P(a) = 1$, $(P(0) \Rightarrow P(f(0))) = 1$, $(P(1) \Rightarrow P(f(1))) = 1$, $P(f(b)) = 0$
3. Choisissons $a = 0$
   - De $P(a) = 1$ et $a = 0$, on déduit : $P(0) = 1$
   - De $P(0) = 1$ et de $(P(0) \Rightarrow P(f(0))) = 1$, on déduit : $P(f(0)) = 1$
   - De $P(f(b)) = 0$ et de $P(f(0)) = 1$, on déduit $f(0) \neq f(b)$ donc que $b \neq 0$, donc : $b = 1$
   - De $P(f(b)) = 0$, $P(0) = 1$ et $b = 1$, on déduit $f(b) = f(1) \neq 0$ donc : $f(1) = 1$ et $P(1) = 0$
   - De $P(f(0)) = 1$ et $P(1) = 0$, on déduit : $f(0) = 0$

Modèle de domaine $\{0, 1\}$:

$a = 0, b = 1, P = \{0\}, f(0) = 0, f(1) = 1$

Substitution

L'application d'une substitution à une formule propositionnelle valide donne une formule valide, s'étend à la logique du premier ordre.

Exemple :

Soit $\sigma(p) = \forall x\ q(x)$.
$p \vee \neg p$ est valide, il en est de même de la formule

$$\sigma(p \vee \neg p) = \forall x\ q(x) \vee \neg \forall x\ q(x)$$

Remplacement

Le principe de remplacement s'étend avec le même énoncé de la logique propositionnelle à la logique du premier ordre car il est déduit des propriétés élémentaires suivantes :

Pour toutes formules A et B et toute variable x :

- $(A \Leftrightarrow B) \models (\forall x A \Leftrightarrow \forall x B)$
- $(A \Leftrightarrow B) \models (\exists x A \Leftrightarrow \exists x B)$

Relation entre $\forall$ et $\exists$

Lemme

Soient A une formule et x une variable.

1. $\neg\forall xA \equiv \exists x\neg A$
2. $\forall xA \equiv \neg\exists x\neg A$
3. $\neg\exists xA \equiv \forall x\neg A$
4. $\exists xA \equiv \neg\forall x\neg A$

Prouvons les deux premières identités, les deux autres sont données en exercice.

Preuve de $\neg\forall xA \equiv \exists x\neg A$

Soit I une interprétation de domaine D et e un état

Evaluons $[\neg\forall xA]_{(I,e)}$

$= 1 - [\forall xA]_{(I,e)}$

$= 1 - \prod_{d\in D}[A]_{(I,e[x=d])}$ d'après le sens de $\forall$

$= \sum_{d\in D}\left(1 - [A]_{(I,e[x=d])}\right)$ par les lois de de Morgan généralisées

$= \sum_{d\in D}[\neg A]_{(I,e[x=d])}$ par définition du sens de $\neg$

$= [\exists x\neg A]_{(I,e)}$ par définition du sens de $\exists$

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

Evaluons $\forall x A$

$\equiv \neg\neg\forall x A$ identité de la double négation

$\equiv \neg\exists x \neg A$ par l'identité

Déplacement des quantificateurs

Soient x, y deux variables et A, B deux formules.

1. $\forall x \forall y A \equiv \forall y \forall x A$
2. $\exists x \exists y A \equiv \exists y \exists x A$
3. $\forall x (A \wedge B) \equiv (\forall x A \wedge \forall x B)$
4. $\exists x (A \vee B) \equiv (\exists x A \vee \exists x B)$
5. Soit Q un des quantificateurs $\forall, \exists$, soit $\circ$ une des opérations $\wedge, \vee, \Rightarrow$. Supposons que x n'est pas une variable libre de A.
   - 5.1 $QxA \equiv A$,
   - 5.2 $Qx(A \circ B) \equiv (A \circ QxB)$

Exemple

Nous éliminons de ces deux formules les quantificateurs inutiles :

- $\forall x \exists x P(x) \equiv$

$\exists x P(x)$

- $\forall x (\exists x P(x) \vee Q(x)) \equiv$

$\exists x P(x) \vee \forall x Q(x)$

Changement de variables liées (1/4)

Theoreme

Soit Q un des quantificateurs $\forall, \exists$. Supposons que y soit une variable qui ne figure pas dans QxA alors : $QxA \equiv QyA < x := y >$.

exemple

- $\forall x\ p(x,z) \equiv \forall y\ p(y,z)$.
- $\forall x\ p(x,z) \not\equiv \forall z\ p(z,z)$.

Changement de variables liées (2/4)

Definition

Deux formules sont égales à un changement près de variables liées si nous pouvons obtenir l'une à partir de l'autre par des remplacements de sous-formules de la forme QxA par

$$QyA < x := y >$$

où Q est un quantificateur et y est une variable qui ne figure pas dans QxA.

Les deux formules sont α-équivalentes ou copie l'une de l'autre ou encore, notée $A =_\alpha B$

Changement de variables liées (3/4)

Theoreme

Si deux formules sont égales à un changement près de variables liées alors elles sont équivalentes.

Changement de variables liées (4/4)

Exemple

Montrons que les formules $\forall x \exists y P(x, y)$ et $\forall y \exists x P(y, x)$ sont égales par changement des variables liées et donc équivalentes.

$$\forall x \exists y P(x, y)$$

$$\equiv \forall u \exists y P(u, y)$$

$$\equiv \forall u \exists x P(u, x)$$

$$\equiv \forall y \exists x P(y, x)$$

Savoir si deux formules sont α-équivalentes

Technique

- Tracer des traits entre chaque quantificateur et les variables qu'il lie.
- Effacer les noms des variables liées.

Si après cette transformation, les deux formules deviennent identiques, c'est qu'elles sont égales à un changement près des variables liées.

Exemple

Soit $\forall x \exists y P(y, x)$ et $\forall y \exists x P(x, y)$ deux formules.

$$\underline{\forall | \exists | P(\underline{|}, |)}$$

Exercice

Calculer la transformation pour

- $A = \forall x \forall y \ R(x, y, y)$
- $B = \forall x \forall y \ R(x, x, y)$

A et B sont-elles α-équivalentes ? non

Propriété $=_\alpha$

Theoreme

1. Soit A une formule atomique, $A =_\alpha A'$ si et seulement si $A' = A$
2. $\neg B =_\alpha A'$ si et seulement si $A' = \neg B'$ et $B =_\alpha B'$
3. $(B \circ C) =_\alpha A'$ si et seulement si $A' = (B' \circ C')$ et $B =_\alpha B'$ et $C =_\alpha C'$. où $\circ$ est l'un des connecteurs $\wedge, \vee, \Rightarrow, \Leftrightarrow$.
4. Si $\forall x B =_\alpha A'$ alors $A' = \forall x' B'$ et pour toute variable z absente des formules B et B', nous avons : $B < x := z >=_\alpha B' < x' := z >$.
5. Si $\exists x B =_\alpha A'$ alors $A' = \exists x' B'$ et pour toute variable z absente des formules B et B', nous avons : $B < x := z >=_\alpha B' < x' := z >$.
6. S'il existe une variable z absente des formules B et B' telle que $B < x := z >=_\alpha B' < x' := z >$ alors $\forall x B =_\alpha \forall x' B'$ et $\exists x B =_\alpha \exists x' B'$.

Algorithme pour le test de l'alpha-équivalence

Les données du test sont deux formules A et A'.
Le résultat est oui si $A =_\alpha A'$, non si $A \neq_\alpha A'$.

Exemple

On traite uniquement le cas où $A = \forall x B$.

1. Si A' n'est pas de la forme $\forall x' B'$, alors, d'après le point (4) du théorème, la réponse est non.
2. Si $A' = \forall x' B'$ alors nous choisissons une variable z quelconque absente de B et B'.
   - 2.1 Si $B < x := z >=_\alpha B' < x' := z >$ alors, d'après point (6) du théorème, la réponse est oui.
   - 2.2 Si $B < x := z > \neq_\alpha B' < x' := z >$ alors, d'après le point (4) du théorème la réponse est non.

Introduction

Rappel : En logique du premier ordre, il n'y a pas d'algorithme pour décider si une formule est valide ou non valide.

Programme semi-décidable :

1. S'il termine alors il décide correctement si la formule est valide ou non.
   Lorsque la formule est valide, la décision est généralement accompagnée d'une preuve.
2. Si la formule est valide, alors il termine. Cependant, l'exécution peut être longue !

Notons que si la formule n'est pas valide, la terminaison de ce programme n'est pas garantie.

Nous étudions maintenant un tel programme.

Fermeture universelle

Definition

Soit C une formule ayant pour variables libres $x_1, \ldots, x_n$.

La fermeture universelle de C, notée $\forall(C)$, est la formule $\forall x_1 \ldots \forall x_n C$.

Cette notion est définie à l'ordre près des variables libres de C.

Soit Γ un ensemble de formules, $\forall(\Gamma) = \{\forall(A) \mid A \in \Gamma\}$.

Exemple

$\forall(P(x) \wedge R(x,y)) =$

$\forall x \forall y (P(x) \wedge R(x,y))$ ou $\forall y \forall x (P(x) \wedge R(x,y))$

Généralisation de la substitution

Definition

Une substitution est une application des variables dans les termes.

Soient A une formule et σ une substitution.

$A\sigma$ est la formule obtenue en remplaçant toute occurrence libre d'une variable par son image dans l'application.

La formule $A\sigma$ est une instance de A.

Hypothèses

Nous considérons que

- des formules qui ne contiennent ni le symbole égal, ni les constantes propositionnelles $\top$ et $\bot$, car leur sens est fixé dans toute interprétation
- toute signature comporte au moins une constante.

Quitte à ajouter **la constante *a***.

Domaine et base de Herbrand

Definition

1. Le domaine de Herbrand pour Σ est l'ensemble des termes fermés (*i.e.*, sans variable) de cette signature, noté $\boldsymbol{D_\Sigma}$.

   Remarque : cet ensemble n'est jamais vide, car $a \in D_\Sigma$.

2. La base de Herbrand pour Σ est l'ensemble des formules atomiques fermées de cette signature, notée $\boldsymbol{B_\Sigma}$.

Definition

(Rappel)

- Un terme sur Σ est : soit une variable, soit une constante s où $s^{f0} \in \Sigma$, soit un terme de la forme $s(t_1, \ldots, t_n)$ où $n \geq 1$, $s^{fn} \in \Sigma$ et où $t_1, \ldots, t_n$ sont des termes sur Σ.
- Une formule atomique sur Σ est : soit une des constantes $\top, \bot$, soit une variable propositionnelle s où $s^{r0} \in \Sigma$, soit de la forme $s(t_1, \ldots, t_n)$ où $n \geq 1$, $s^{rn} \in \Sigma$ et où $t_1, \ldots, t_n$ sont des termes sur Σ.

Exemple

1. Soit $\Sigma = \{a^{f0}, b^{f0}, P^{r1}, Q^{r1}\}$, $D_\Sigma = \{a, b\}$ et $B_\Sigma =$

$\{P(a), P(b), Q(a), Q(b)\}$.

2. Soit $\Sigma = \{a^{f0}, f^{f1}, P^{r1}\}$, $D_\Sigma = \{f^n(a) \mid n \in \mathbb{N}\}$ et $B_\Sigma =$

$\{P(f^n(a)) \mid n \in \mathbb{N}\}$

Interprétation de Herbrand

Definition

Soient Σ une signature et $E \subseteq B_\Sigma$. L'interprétation de Herbrand $H_{\Sigma,E}$ a pour domaine D_Σ et donne aux symboles le sens suivant :

1. Si le symbole s est une constante de la signature, il vaut lui-même dans cette interprétation.
2. Si s est un symbole de fonction à $n \geq 1$ arguments de la signature et si $t_1, \ldots, t_n \in D_\Sigma$ alors $s^{fn}_{H_{\Sigma,E}}(t_1, \ldots, t_n) = s(t_1, \ldots, t_n)$.
3. Si le symbole s est une variable propositionnelle, il vaut 1, autrement dit il est vrai, si et seulement si $s \in E$.
4. Si s est un symbole de relation de la signature à $n \geq 1$ arguments et si $t_1, \ldots, t_n \in D_\Sigma$ alors $s^{rn}_{H_{\Sigma,E}} = \{(t_1, \ldots, t_n) \mid t_1, \ldots, t_n \in D_\Sigma \wedge s(t_1, \ldots, t_n) \in E\}$.

Propriété de l'interprétation de Herbrand

Propriete

Soient Σ une signature et $E \subseteq B_\Sigma$. Dans l'interprétation de Herbrand $H_{\Sigma,E}$:

1. La valeur d'un terme sans variable est lui-même
2. L'interprétation est modèle d'une formule atomique fermée si et seulement si elle est élément de E.

La preuve est une conséquence directe de la définition d'une interprétation de Herbrand.

Notons ici, avec un exemple, pourquoi on a supposé que les formules ne contiennent pas les symboles de relation $\top, \bot, =$, dont le sens est fixé dans toutes les interprétations.

Supposons au contraire que $\top$ soit élément de la base mais non élément de E. D'après le point 2, l'interprétation $H_{\Sigma,E}$ donnerait $\top$ la valeur 0, alors que $\top$ vaut 1 dans toute interprétation.

Exemple

Soit $\Sigma = \{a^{f0}, b^{f0}, P^{r1}, Q^{r1}\}$

L'ensemble $E = \{P(b), Q(a)\}$ définit l'interprétation de Herbrand H de domaine $D_\Sigma = \{a, b\}$ où :

- les constantes a et b ont pour valeur elles-mêmes et
- $P_H = \{b\}$ et $Q_H = \{a\}$.

Formule universelle et modèle de Herbrand

Theoreme

Soit Γ un ensemble de formules sans quantificateur sur la signature Σ.

$\forall(\Gamma)$ a un modèle si et seulement si $\forall(\Gamma)$ a un modèle qui est une interprétation de Herbrand de Σ.

Preuve, Cf. Poly

Exemple

Soit $\Sigma = \{a^{f0}, b^{f0}, P^{r1}, Q^{r1}\}$

L'ensemble $E = \{P(b), Q(a)\}$ définit l'interprétation de Herbrand H de domaine $D_\Sigma = \{a, b\}$ où :

- les constantes a et b ont pour valeur elles-mêmes et
- $P_H = \{b\}$ et $Q_H = \{a\}$.

Soit I l'interprétation de domaine $\{0, 1\}$ où :

- $a_I = 0$, $b_I = 1$,
- $P_I = \{1\}$ et $Q_I = \{0\}$.

I est modèle de $\forall(\Gamma)$, où Γ est un ensemble de formules sans quantificateur sur la signature Σ, ssi H est un modèle de Herbrand de $\forall(\Gamma)$

Théorème de Herbrand

Theoreme

Soit Γ un ensemble de formules sans quantificateur de signature Σ.

$\forall(\Gamma)$ a un modèle *si et seulement si*
tout ensemble fini d'instances fermées sur la signature Σ des formules de Γ a un modèle propositionnel application de la base de Herbrand B_Σ dans l'ensemble $\{0, 1\}$.

Rappel : la signature comporte au moins une constante et le signe égal n'est pas utilisé.

Exemple

Soit $\Gamma = \{P(x) \vee Q(x), \neg P(a), \neg Q(b)\}$.

$\Sigma = \{a^{f0}, b^{f0}, P^{r1}, Q^{r1}\}$, $B_\Sigma = \{P(a), P(b), Q(a), Q(b)\}$

$\forall(\Gamma) = \{\forall x(P(x) \vee Q(x)), \neg P(a), \neg Q(b)\}$

L'ensemble des instances fermées de Γ est

$$\{P(a) \vee Q(a), P(b) \vee Q(b), \neg P(a), \neg Q(b)\}$$

$v(P(a)) = 0, v(P(b)) = 1, v(Q(a)) = 1, v(Q(b)) = 0$ est un modèle propositionnel application de la base de Herbrand B_Σ dans l'ensemble $\{0, 1\}$.

Donc, $\forall(\Gamma)$ a un modèle. En effet, l'interprétation I de domaine $\{0, 1\}$ définie comme suit est modèle de $\forall(\Gamma)$:

$$a_I = 0, b_I = 1, P_I = \{1\}, Q_I = \{0\}$$

Idées de la preuve (1/2)

$\Rightarrow$ Supposons que $\forall(\Gamma)$ a un modèle I.
Les instances des formules de Γ sont conséquences de $\forall(\Gamma)$ donc ont pour modèle I.
Ce modèle I peut être vu comme un modèle propositionnel v de domaine B_Σ, la base de Herbrand de la signature Σ, où pour tout $A \in B_\Sigma$, $v(A) = [A]_I$.

Donc v est modèle propositionnel de tout ensemble d'instances fermés sur Σ des formules de Γ.

Idées de la preuve (2/2)

$\Leftarrow$ Supposons que tout ensemble fini d'instances fermées sur la signature Σ des formules de Γ a un modèle propositionnel de domaine B_Σ.
D'après le théorème de compacité (théorème), l'ensemble de toutes les instances fermées sur la signature Σ a alors un modèle propositionnel v de domaine B_Σ.
Ce modèle propositionnel peut être vu comme le modèle de Herbrand de $\forall(\Gamma)$ associé à l'ensemble des éléments de la base de Herbrand dont v est modèle. D'après le théorème, $\forall(\Gamma)$ a un modèle.

Variante du théorème de Herbrand

Corollaire

Soit Γ un ensemble de formules sans quantificateur de signature Σ.

$\forall(\Gamma)$ est insatisfaisable *si et seulement s'il existe*
un ensemble fini insatisfaisable d'instances fermées sur la signature Σ des formules de Γ

Preuve : Le corollaire est obtenu en remplaçant chaque côté de l'équivalence du théorème de Herbrand par sa négation.

Procédure de semi-décision : insatisfaisabilité de $\forall(\Gamma)$

Soit Γ un ensemble fini de formules sans quantificateur.
Énumérer l'ensemble des instances fermées des formules de Γ sur la signature Σ et arrêter dès que :

- (1) un ensemble est insatisfaisable, donc $\forall(\Gamma)$ est insatisfaisable.
- (2) terminaison sans contradiction (dans ce cas, le domaine de Herbrand ne comprend que des constantes) donc $\forall(\Gamma)$ est satisfaisable, on a un modèle.
- (3) on est « fatigué » ! donc on ne peut pas conclure : le corollaire nous dit que si $\forall(\Gamma)$ est insatisfaisable, et si l'on avait été plus courageux, on aurait obtenu une contradiction !

Exemple (1/5)

Soit $\Gamma = \{P(x), Q(x), \neg P(a) \vee \neg Q(b)\}$ et
$\Sigma = \{a^{f0}, b^{f0}, P^{r1}, Q^{r1}\}$.

$D_\Sigma = \{a, b\}$. $B_\Sigma = \{P(a), P(b), Q(a), Q(b)\}$.

L'ensemble $\{P(a), Q(b), \neg P(a) \vee \neg Q(b)\}$ d'instances sur le domaine de Herbrand est insatisfaisable, donc $\forall(\Gamma)$ est insatisfaisable.

Exemple (2/5)

Soit $\Gamma = \{P(x) \vee Q(x), \neg P(a), \neg Q(b)\}$ et
$\Sigma = \{a^{f0}, b^{f0}, P^{r1}, Q^{r1}\}$.

$D_\Sigma = \{a, b\}$. $B_\Sigma = \{P(a), P(b), Q(a), Q(b)\}$.

L'ensemble de toutes les instances sur le domaine de Herbrand $\{P(a) \vee Q(a), P(b) \vee Q(b), \neg P(a), \neg Q(b)\}$ a un modèle propositionnel caractérisé par $E = \{P(b), Q(a)\}$.

Donc l'interprétation de Herbrand associée à E est modèle de $\forall(\Gamma)$.

À partir de E, nous pouvons fabriquer une interprétation modèle de $\forall(\Gamma)$, c'est-à-dire, un modèle de $\{\forall x(P(x) \vee Q(x)), \neg P(a), \neg Q(b)\}$: soit I l'interprétation de domaine $\{0, 1\}$ où $a_I = 0$, $b_I = 1$, $P_I = \{1\}$ et $Q_I = \{0\}$; I est modèle de $\forall(\Gamma)$.

Exemple (3/5)

Soit $\Gamma = \{P(x), \neg P(f(x))\}$ et $\Sigma = \{a^{f0}, f^{f1}, P^{r1}\}$.

$D_\Sigma = \{f^n(a) | n \in \mathbb{N}\}$. $B_\Sigma = \{P(f^n(a)) \mid n \in \mathbb{N}\}$.

L'ensemble $\{P(f(a)), \neg P(f(a))\}$ d'instances sur le domaine de Herbrand est insatisfaisable, donc $\forall(\Gamma)$ est insatisfaisable.

Exemple (4/5)

Soit $\Gamma = \{P(x) \vee \neg P(f(x)), \neg P(a), P(f(f(a)))\}$ et $\Sigma = \{a^{f0}, f^{f1}, P^{r1}\}$.

$D_\Sigma = \{f^n(a) | n \in \mathbb{N}\}$. $B_\Sigma = \{P(f^n(a)) \mid n \in \mathbb{N}\}$.

L'ensemble
$\{P(a) \vee \neg P(f(a)), P(f(a)) \vee \neg P(f(f(a))), \neg P(a), P(f(f(a)))\}$
d'instances sur le domaine de Herbrand est insatisfaisable, donc $\forall(\Gamma)$ est insatisfaisable.

Remarque : observez qu'il a fallu prendre 2 instances de la première formule de Γ pour obtenir une contradiction.

Exemple (5/5)

Soit $\Gamma = \{R(x, s(x)), R(x, y) \wedge R(y, z) \Rightarrow R(x, z), \neg R(x, x)\}$ et $\Sigma = \{a^{f0}, s^{f1}, R^{r2}\}$.

$D_\Sigma = \{s^n(a) \mid n \in \mathbb{N}\}$. $B_\Sigma = \{R(s^n(a), s^m(a)) \mid n, m \in \mathbb{N}\}$. D_Σ et B_Σ sont infinis.

$\forall(\Gamma)$ a un modèle infini : l'interprétation I de domaine $\mathbb{N}$ avec $\forall n \in \mathbb{N}$, $s_I(n) = n + 1$ et $R_I = \{(n, p) \mid n < p\}$, en bref $R(x, y) = x < y$.

$\forall(\Gamma)$ n'a aucun modèle fini, autrement dit il est inutile de chercher des modèles finis avec les méthodes du chapitre précédent.

Puisque $\forall(\Gamma)$ a un modèle, on est dans une situation où la procédure évoquée précédemment ne pourra jamais donner de réponses, aussi longtemps que l'on poursuive l'énumération des instances des formules de Γ.

Introduction

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

Pour des formules avec quantificateur existentiel utiliser la skolémisation.

Cette transformation est dûe à Thoralf Albert Skolem (1887 - 1963) mathématicien et logicien norvégien.

La skolémisation

- change un ensemble de formules fermées en la fermeture universelle d'un ensemble de formules sans quantificateur.
- préserve l'existence d'un modèle.

Exemple

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

On observe les relations suivantes entre ces deux formules :

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

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

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

   J est modèle de $P(a)$.

Exemple

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

On observe les relations entre ces deux formules :

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

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

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

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

Propriétés

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

- B a pour conséquence A, $(B \models A)$
- tout modèle de A « donne » un modèle de B

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

Définitions

Definition

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

Exemple

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

Définitions : forme normale généralisée

En logique du premier ordre une formule est en forme **normale** si elle est sans équivalence, ni implication, et dont les négations portent uniquement sur les formules atomiques.

Comment skolémiser une formule fermée A ?

Definition

(skolémisation) Soient A une formule fermée et E la formule normale sans quantificateur obtenue par la transformation ci-après : E est la forme de Skolem de A.

1. B = normalisation de A
2. C = rendre B propre
3. D= Élimination des quantificateurs existentiels de C.
   Cette transformation préserve seulement l'existence de modèle.
4. E = Transformation de la formule fermée, normale, propre et sans quantificateur existentiel D en une formule normale sans quantificateur.

Normalisation

1. Enlever les équivalences
2. Enlever les implications
3. Déplacer les négations vers les formules atomiques

Règles

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

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

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

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

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

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

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

Astuce : remplacer $\neg(A \Rightarrow B)$ par $A \wedge \neg B$

Exemple

La forme normale de $\forall y(\forall x P(x,y) \Leftrightarrow Q(y))$ est :

Tout d'abord, suppression de l'équivalence :

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

puis par déplacement de la négation en :

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

Transformation en formule propre

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

Exemple

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

$$\forall x P(x) \vee \forall y Q(y)$$

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

$$\forall x (P(x) \Rightarrow \exists z Q(z) \wedge \exists y R(x, y))$$

Élimination des quantificateurs existentiels

Theoreme

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

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

Preuve du Théorème

Montrons que A' a pour conséquence A.

Puisque la formule A est fermée et propre, les variables libres de $\exists yB$, qui sont liées à l'extérieur de $\exists yB$, ne sont pas liées par des quantificateurs dans B (sinon la propriété propre ne serait pas respectée), donc le terme $f(x_1, \ldots x_n)$ est libre pour y dans B.

D'après le corollaire : $B < y := f(x_1, \ldots x_n) >$ a pour conséquence $\exists yB$. D'où, nous en déduisons que A' a pour conséquence A.

Preuve du théorème

Montrons que tout modèle de A donne un modèle de A'.

Supposons que A a un modèle I où I est une interprétation de domaine D. Soit $c \in D$. Pour tous $d_1, \ldots, d_n, \in D$, soit $E_{d_1,\ldots,d_n}$ l'ensemble des éléments $d \in D$ tels que la formule B vaut 1 dans l'interprétation I et l'état $x_1 = d_1, \ldots, x_n = d_n, y = d$ de ses variables libres. Soit $g : D^n \to D$ une fonction telle que si $E_{d_1,\ldots,d_n} \neq \emptyset$ alors $g(d_1, \ldots, d_n) \in E_{d_1,\ldots,d_n}$ sinon $g(d_1, \ldots, d_n) = c$. Soit J l'interprétation identique à I sauf que $f_J = g$. Nous avons :

1. $[\exists y B]_{(I,e)} = [B < y := f(x_1, \ldots, x_n) >]_{(J,e)}$, ceci d'après l'interprétation de f et le théorème, pour tout état e des variables,
2. $[\exists y B]_{(I,e)} = [\exists y B]_{(J,e)}$, puisque le symbole f est nouveau, la valeur de $\exists y B$ ne dépend pas du sens de f.
3. $\exists y B \Leftrightarrow B < y := f(x_1, \ldots, x_n) > \models A \Leftrightarrow A'$, d'après la propriété du remplacement, qui est aussi vraie en logique du premier ordre.

D'après ces trois points, nous obtenons $[A]_{(J,e)} = [A']_{(J,e)}$ et puisque f n'est pas dans A et que les formules A et A' n'ont pas de variables libres, nous avons $[A]_I = [A']_J$. Puisque I est modèle de A, J est modèle de A'.

Remarque

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

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

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

Exemple

En éliminant les quantificateurs existentiels de la formule $\exists x \forall y P(x, y) \wedge \exists z \forall u \neg P(z, u)$ on obtient $\forall y P(a, y) \wedge \forall u \neg P(b, u)$.

Il est facile de voir que cette formule a un modèle.

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

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

Transformation en formule universelle

Theoreme

Soit A une formule fermée, normale, propre et sans quantificateur existentiel. Soit B la formule obtenue en enlevant de A tous les quantificateurs universels (B est la forme de Skolem de A).

La formule A est équivalente à la fermeture universelle de B.

Démonstration.

Avec les conditions posées sur A, la transformation de A en $\forall(B)$ revient à effectuer tous les remplacements possibles des sous formules de la forme

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

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

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

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

Puisque chacun de ces remplacements change une formule en une autre équivalente, les formules A et $\forall(B)$ sont équivalentes.

□

Propriété de la skolémisation

Propriété

Soit A une formule fermée et B la forme de Skolem de A.

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

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

Démonstration.

Soit C la formule fermée en forme normale et propre, obtenue au terme des deux premières étapes de la skolémisation de A. Soit D le résultat de l'élimination des quantificateurs existentiels appliquée à C. D'après la remarque nous avons :

- La formule D a pour conséquence la formule C
- si C a un modèle alors D a un modèle.

Puisque les deux premières étapes changent des formules en des formules équivalentes, A et C sont équivalentes. D'après le théorème, D est équivalent à $\forall(B)$. Donc nous pouvons remplacer ci-dessus D par $\forall(B)$ et C par A, CQFD. □

Exemple

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

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

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

Skolémiser un ensemble de formules

Corollaire

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

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

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

Devise d'Oxford

The more I study, the more I know
The more I know, the more I forget
The more I forget, the less I know