

Cours Module 15

Intelligence Artificielle

MIAGe

Marie-Pierre Gleizes
gleizes@irit.fr
IRIT - Université Paul Sabatier
Toulouse III
Année 2001- 2002

1



Cours Module 15

Intelligence Artificielle

MIAGe

Introduction à l'IA

Marie-Pierre Gleizes

2

Plan



- ✍ Introduction à l'I.A.
- ✍ La logique des propositions
- ✍ La logique des prédicats
- ✍ Le principe de résolution
- ✍ Les systèmes à base de connaissances
- ✍ PROLOG

3

Introduction



Intelligence ?

Ce que les gens font bien

Artificiel ?

Machine

MAIS machines fragiles si le contexte change

Test de Turing

1940 mathématicien anglais

4

Définitions

✍ CHARNIAK & MAC DERMOTT

L'IA est l'étude des facultés mentales à travers l'utilisation de modèles calculatoires

✍ MARVIN MINSKY

Domaine de la recherche visant à faire faire à des machines ce qui, d'après les humains, requiert de l'intelligence

✍ ELAINE RICH

L'IA est l'étude de l'art de faire faire par des machines des choses pour lesquelles les hommes sont meilleurs pour le moment

L'IA EST L'ÉTUDE DES CONCEPTS QUI PERMETTENT DE RENDRE
LES MACHINES PLUS INTELLIGENTES

5

Historique (1)

Périodes	Événements majeurs
	Logique formelle
Bases d'avant la 2 ^{ème} guerre mondiale	Psychologie cognitive
	1940-1944 Premiers ordinateurs : MarkI IBM
	1943 - 1946 ENIAC commercialisation
	Développement d'ordinateurs
Les années d'après guerre 1945-1954	H. Simon (économiste) "Administrative behavior"
	N. Wiener "Cybernetics"
	A.M. Turing "Computing Machinery and Intelligence"
Pré-intelligence artificielle	Conférence de MACY sur la cybernétique
	Langage de traitement des listes : IPL
Les années de formation 1955 - 1960	Séminaire de Dartmouth New-Hamshire US été 1956
	John Mac Carthy nomme la discipline IA
	Algorithme Alpha-Béta
Début des recherches en IA	LISP de Mac Carthy 1958 MIT
	Psychologie du traitement de l'information

6

Historique (2)



Périodes	Événements majeurs
Les années du développement et de la réorganisation	Résolveur général de problème GPS de Newell, Simon et shaw
1961 - 1970	Human Problem Solving Newell et Simon
	Heuristiques
Recherche de résolveurs de problèmes généraux	Robotique
	Programme d'échecs
	DENDRAL à Stanford en chimie moléculaire de Buchanan
	MYCIN de Shortliffe en 1976 :
Les années de spécialisation et du succès	diagnostic des maladies infectieuses du sang
1971 - 1980	Hearsay II de Erman et Lesser en 1975 en reconnaissance de la parole
	génie cognitif de nouveaux métiers
Découverte des systèmes à base de connaissances	EMYCIN Essential mycin : générateur de systèmes experts
	PROLOG de Colmerauer à marseille
	Sciences cognitives

7

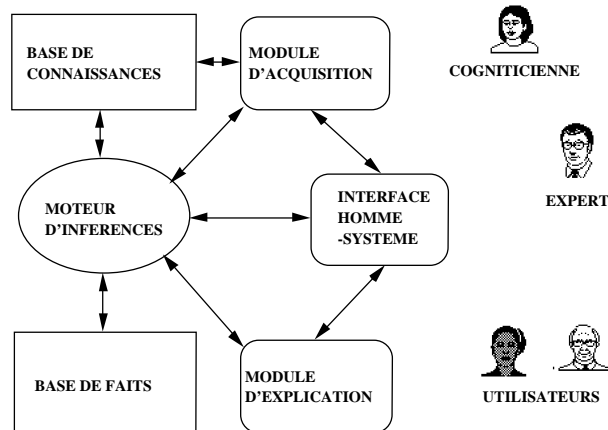
Historique (3)



Périodes	Événements majeurs
	PROSPECTOR de Duda 1978
Les applications 1981	Projet de la cinquième génération au JAPON
	premières machines PSI : machines LISP
Compétition internationale	Sociétés d'IA
IAD	IAD SMA
	Réseaux de neurones
	Algorithmes génétiques
	Vie artificielle

8

Système à base de connaissances



9

Objectifs

Systèmes capables de reproduire
Le comportement de l'être humain dans ses activités de raisonnement

🔪 Démarche cognitive

Etude des mécanismes de l'intelligence
L'IA cherche à comprendre les mécanismes de compréhension

🔪 Démarche technique

Emulation par l'ordinateur des comportements humains
Rendre les machines plus utiles

10

Caractéristiques



- ✍ Manipulation de symboles
- ✍ Utilisation de méthodes heuristiques
- ✍ Situations incomplètes, inexactes, conflictuelle
- ✍ Notion de connaissances
- ✍ Pluridisciplinarité de l'IA
 - <-- Logique, psychologie cognitive
 - <-- linguistique
 - <-- ergonomie
 - <-- Philosophie
 - <-- Neurosciences et biologie...

11

PROGRAMMATION CONVENTIONNELLE	PROGRAMMATION SYMBOLIQUE
ALGORITHMES	HEURISTIQUES
BASE DE DONNÉES À ADRESSAGE NUMÉRIQUES	BASE DE CONNAISSANCES À STRUCTURE SYMBOLIQUE AU SEIN D'UNE BASE DE FAITS GÉNÉRALE
ORIENTÉ VERS LE TRAITEMENT NUMÉRIQUE	ORIENTÉE VERS LE TRAITEMENT SYMBOLIQUE
PROCÉDURAL	DÉCLARATIVITÉ
EXPLICATION EN COURS IMPOSSIBLE	EXPLICATION EN COURS D'EXÉCUTION FACILE
MODIFICATION DIFFICILE	MODULARITÉ

12

Les langages de l'IA



- ✍ Manipulation listes
- ✍ Possibilité de définir divers types de données
- ✍ Modularité, interactivité, efficacité
- ✍ Filtrage
- ✍ Déduction automatique
- ✍ Structuration des connaissances
- ✍ Focalisation de l'attention
- ✍ Comportement piloté par le but
- ✍ Manipulation du déclaratif et du procédural

Aucun langage ne réunit toutes ces caractéristiques

13

Quelques langages



- ✍ IPL Information Processing Language - Newell 1960 - Traitement de listes
- ✍ LISP List processing language - John Mac Carthy AU MIT 1958
- ✍ MACHINES LISP
- LAMBDA MACHINE - MAIA - Texas Explorer - Symbolics...
- INTERLISP DIALECTE DE LISP (XEROX ET BBN)
- MACLISP DIALECTE LISP (MIT)
- SAIL - Swinehart 1971
- Dérivé d'algol avec des caractéristiques supplémentaires dont un support pour une mémoire associative
- ✍ PLANNER - Hewitt 1971
- Un des premiers langages qui facilite le raisonnement conduit par le but
- ✍ KRL - Bobrow 1977 Langage de type Frame
- ✍ PROLOG Colmerauer 1970 Marseille
- Langage de règles construit sur un démonstrateur de théorèmes en logique des prédicats

14

Différents champs de l'IA



- ✍ Processus de recherche
- ✍ Représentation des connaissances
- ✍ Domaine d'application

15

Domaines d'application



- ✍ Problèmes combinatoires : jeux
- ✍ Preuves de théorèmes
- ✍ Traduction automatique et traitement des langues naturelles
- ✍ Reconnaissance des formes : vision, image, audition
- ✍ Systèmes experts ou à base de connaissances
- ✍ Robotique
- ✍ Apprentissage
- ✍ Bases de données intelligentes

16

La représentation des connaissances

Marie-Pierre Gleizes

17

Préoccupation

- ✍ PHILOSOPHIE
- ✍ LOGIQUE
- ✍ LINGUISTIQUE
- ✍ PSYCHOLOGIE COGNITIVE
- ✍ INTELLIGENCE ARTIFICIELLE

CORRESPONDANCE

MONDE EXTERIEUR ----- SYSTEME SYMBOLIQUE

MANIPULABLE PAR ORDINATEUR

QU'EST-CE-QUE LA CONNAISSANCE ?

Un livre ne connaît pas l'information qu'il contient

CONNAITRE = MEMORISER + INFERER

REPRESENTER = FORMALISER + RAISONNER

18

REPRESENTATION

= ensemble de conventions syntaxiques et sémantiques
rendant possible la description d'objets

SA SYNTAXE

symboles qui peuvent être utilisés
façons dont ces symboles peuvent être assemblés

SA SEMANTIQUE

comment le sens se trouve intégré aux symboles et dans
les arrangements de symboles qui sont autorisés par la
syntaxe

19

Typologie de la connaissance - Daniel KAYSER

DE DEFINITION

--> toujours vraie

EVOLUTIVE / ATEMPORELLE

--> peut être modifiée

INCERTAINE / CERTAINE

--> pas avec certitude

--> évaluation difficile

TYPIQUE / UNIVERSELLE

--> plausible mais peut être contredit (habituellement)

AMBIGUË

--> plusieurs significations

20

Problèmes fondamentaux

✍ MODALITES POINTS DE VUE

Comment exprimer les nuances qui apparaissent dans l'expression d'une opinion? "je crois que", "il est probable que", "je pense que"... Différents statuts: intangibles, modifiables, certaines ou incertaines, valides ou périmées.

✍ EVOLUTIVITE

Les connaissances changent au fur et à mesure que l'univers dans lequel elles sont utilisées se modifie.

✍ TYPICALITE ET PARTAGE DE PROPRIETE

Les Alsaciens aiment la bière n'a pas une valeur universelle
Elle exprime le fait qu'en général un Alsacien aime la bière, bien que certains individus puissent faire exception

✍ CONNAISSANCES INCOMPLETES IMPRECISES INCERTAINES

Comment gérer les incertitudes et traiter les données imprécises?

==> PAS DE FORMALISME IDEAL

21

Bon système de représentation des connaissances

✍ Adéquation représentationnelle

✍ Adéquation inférentielle

✍ Efficacité inférentielle

✍ Efficacité acquisitionnelle

✍ Simplicité

non informaticien puisse transmettre son savoir au système

✍ Connaissance explicite

recherche des erreurs et justification

22

Techniques pour représenter la connaissance



✚ REPRESENTATION DECLARATIVE

- ✚ les connaissances n'ont pas un caractère opératoire
- ✚ il n'est fait aucune hypothèse sur la façon de les utiliser
- ✚ indépendantes (clauses en PROLOG)
- ✚ liées (réseau sémantique)
- ✚ pertinence discutée a priori

✚ REPRESENTATION PROCEDURALE

Les connaissances ont un caractère opératoire et leur pertinence ne peut être jugée qu'après exécution du programme qui les manipule

23

✚ AVANTAGES D'UNE REPRESENTATION PROCEDURALE

- ✚ facilité pour représenter les connaissances sur la manière de faire les choses
- ✚ facilité pour représenter les connaissances qui ne rentrent pas dans de nombreux schémas déclaratifs simples (raisonnement par défaut...)
- ✚ facilité pour représenter des connaissances heuristiques sur la manière de faire les choses plus efficacement
- ✚ rectitude du raisonnement (heuristiques spécifiques pour induire un raisonnement naturel)
- ✚ facilité de codage
- ✚ facilité de compréhension du processus lui-même

✚ AVANTAGES D'UNE REPRESENTATION DECLARATIVE

- ✚ modularité
- ✚ chaque connaissance n'est stockée qu'une fois quel que soit le nombre de façons différentes dont on peut l'utiliser: gain de place

24

REPRESENTATION DES CONNAISSANCES UTILISANT LA LOGIQUE

- ✍ LOGIQUE DES PROPOSITIONS
- ✍ LOGIQUE DES PREDICATS
- ✍ LOGIQUES NON STANDARD
- ✍ LES REGLES DE PRODUCTION

REPRESENTATION STRUCTUREE DES CONNAISSANCES

- ✍ RESEAUX SEMANTIQUES
- ✍ OBJETS STRUCTURES
- ✍ FRAMES
- ✍ SCRIPTS
- ✍ SCHEMAS
- ✍ PROTOTYPES

25

Bibliographie

- ✍ BONNET A, L'intelligence artificielle : promesses et réalité, Editions InterEditions 1984
- ✍ KAYSER D, La représentation des connaissances : un cas typique de collaboration interdisciplinaire, Cognitiva 1985
- ✍ NILSSON M, principes de l'IA Cepadues Editions 1988
- ✍ WINSTON P, Intelligence Artificielle, InterEditions 1984

26




UNIVERSITE
PAUL
SABATIER
TOULOUSE III

Cours Module 15
Intelligence Artificielle
MIAGe



cnrs • inrp • ups
IRIT
Institut
Recherche
informatique
toulouse

La logique



cnrs • inrp • ups
IRIT
Institut
Recherche
informatique
toulouse




UNIVERSITE
PAUL
SABATIER
TOULOUSE III

Cours Module 15
Intelligence Artificielle
MIAGe




cnrs • inrp • ups
Institut
Recherche
informatique
de
toulouse




UNIVERSITE
PAUL
SABATIER
TOULOUSE III

La logique
des propositions



cnrs • inrp • ups
Institut
Recherche
informatique
de
toulouse




Jean est grand et Jean est brun
 --> JEAN_GRAND ET JEAN_BRUN ==> CONJONCTION

Jean est sage ou Jean est dissipé
 --> JEAN_SAGE OU JEAN_DISSIPE ==> DISJONCTION (ou inclusif)


Jean n'est pas petit <--> non JEAN PETIT
 --> NON JEAN_PETIT ==> NEGATION


si la voiture appartient à Jean alors elle est verte
 --> SI APPARTENIR_VOITURE_JEAN ALORS COULEUR_VOITURE_VERTE
 ==> IMPLICATION


31



Alphabet du langage

 **CONNECTEURS** $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

 **DELIMITEURS** $()$

 **ATOMES ou PROPOSITIONS ATOMIQUES**
 (majuscules de l'alphabet latin, et les concaténations de telles lettres) $\{A, B, Z, A_i, \dots\}$

32

Formules bien formées FBF ou WFF

Le **langage** = ensemble des FBFs défini comme suit:

- ✍ tout atome est une fbf
- ✍ si F et G sont des fbfs alors
- $(\neg G) (F \wedge G) (F \vee G) (F \rightarrow G) (F \leftrightarrow G)$ sont des fbfs
- ✍ toutes les fbfs sont obtenues à partir des 2 règles ci-dessus.

Exemple:

A, B, C, D sont des fbfs

$((A \vee (\neg B)) \wedge (C \vee D))$ est une fbf

$(A \vee B) (\wedge \vee C)$ n'est pas une fbf

33

✍ Ordre de priorité des connecteurs:

(Le plus prioritaire) $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

Exemple: $A \wedge \neg B \vee C \rightarrow D \wedge E$

c'est $((A \wedge (\neg B)) \vee C) \rightarrow (D \wedge E)$

✍ On omet par abus les parenthèses les plus externes

Exemple: $(A \vee B)$ devient $A \vee B$

✍ Quand il y a un seul connecteur, l'association se fait de gauche à droite.

Exemple: $A \rightarrow B \rightarrow C$ correspond à $((A \rightarrow B) \rightarrow C)$

34

Sémantique des propositions

INTERPRETATION du calcul propositionnel consiste à donner:

- 1- Domaine d'interprétation non vide D
- 2- Valuation des atomes dans D
- 3- Définition des connecteurs par des applications de D dans D pour \neg et de $D \times D$ dans D pour $\wedge, \vee, \rightarrow, \leftrightarrow$.

Interprétation classique de la logique des propositions

$D = \{ V, F \}$

Tout atome est vrai ou faux mais pas les deux à la fois

Définition des connecteurs par des tables de vérité:

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \Rightarrow B$	$A \leftrightarrow B$
V	V	F	V	V	V	V
V	F	F	F	V	F	F
F	V	V	F	V	V	F
F	F	V	F	F	V	V

35

Interprétation d'une fbf G

Pour une fbf G composée de différents atomes: $A_1 \dots A_n$, une interprétation de G est une assignation des valeurs de vérité à $A_1 \dots A_n$.

G comporte n atomes $\Rightarrow 2^n$ interprétations possibles

Exemple:

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

$\Rightarrow 8$ interprétations possibles

A	B	C	G
V	V	V	V
V	V	F	F
V	F	V	V
V	F	F	F
F	V	V	V
F	V	F	F
F	F	V	F
F	F	F	F

36

Une fbf G est vraie (respectivement fausse) dans une interprétation si la valeur de G est vraie (respectivement fausse)

Exemple:

$$G: (P \rightarrow (Q \vee (\neg R)))$$

Notation $I_1(P)$ peut s'écrire $v_{I_1}(P)$ et se lit valeur de P selon l'interprétation I_1

Soit I_1 telle que: $I_1(P) = I_1(R) = V$

$$I_1(Q) = F$$

G est fausse dans I_1

Soit I_2 telle que: $I_2(P) = I_2(R) = F$

$$I_2(Q) = V$$

G est vraie dans I_2

Formules équivalentes (1) A, B, C sont des fbfs

- | | | |
|----------------------------|----------|--|
| 1) $A \rightarrow B$ | \equiv | $\neg A \vee B$ |
| 2) $A \leftrightarrow B$ | \equiv | $(A \rightarrow B) \wedge (B \rightarrow A)$ |
| 3) Commutativité | | |
| a) $A \vee B$ | \equiv | $B \vee A$ |
| b) $A \wedge B$ | \equiv | $B \wedge A$ |
| 4) Associativité | | |
| a) $(A \vee B) \vee C$ | \equiv | $A \vee (B \vee C)$ |
| b) $(A \wedge B) \wedge C$ | \equiv | $A \wedge (B \wedge C)$ |
| 5) Distributivité | | |
| a) $A \vee (B \wedge C)$ | \equiv | $(A \vee B) \wedge (A \vee C)$ |
| b) $A \wedge (B \vee C)$ | \equiv | $(A \wedge B) \vee (A \wedge C)$ |
| 6) a) $A \vee V$ | \equiv | V |
| b) $A \wedge V$ | \equiv | A |

Formules équivalentes (2)

$$7) \text{ a) } A \vee \mathbf{F} \cong A \qquad \text{b) } A \wedge \mathbf{F} \cong \mathbf{F}$$

$$8) \text{ a) } A \vee \neg A \cong \mathbf{V} \qquad \text{b) } A \wedge \neg A \cong \mathbf{F}$$

9) Involution

$$(\neg(\neg A)) \cong A$$

10) Lois de de Morgan

$$\text{a) } \neg(A \vee B) \cong (\neg A) \wedge (\neg B)$$

$$\text{b) } \neg(A \wedge B) \cong (\neg A) \vee (\neg B)$$

$$11) \text{ a) } A \vee ((\neg A) \wedge B) \cong A \vee B$$

$$\text{b) } A \wedge ((\neg A) \vee B) \cong A \wedge B$$

$$12) \text{ a) } A \vee A \cong A \qquad \text{b) } A \wedge A \cong A$$

39

Exercice : développer

$$\text{a) } (\neg(A \wedge (B \vee C)))$$

$$\text{b) } ([\neg(\neg(A \wedge B) \vee (\neg D))] \wedge E \vee F)$$

$$\text{c) } (\neg((\neg A) \wedge B \wedge ((\neg C) \vee D) \vee (\neg E) \wedge F \wedge (\neg G)))$$

$$\text{d) } (\neg(A \vee (\neg B) \vee C) \wedge ([\neg(\neg D)] \vee (\neg E)) \vee (\neg F) \wedge G)$$

40

Une fbf est une **tautologie** (valide) \Leftrightarrow elle est vraie dans toute interprétation

Une fbf est **invalid** \Leftrightarrow elle n'est pas valide

Une fbf est **inconsistante** \Leftrightarrow elle est fausse dans toute interprétation

Une fbf est **consistante** \Leftrightarrow elle n'est pas inconsistante

- 1- Une fbf peut être à la fois invalide et consistante
- 2- G est valide $\Rightarrow (\neg G)$ est inconsistante
- 3- G est valide $\Rightarrow G$ est consistante
 G est inconsistante $\Rightarrow G$ est invalide
- 4- Il existe une procédure effective pour déterminer si une fbf est valide: la table de vérité

41

Exemple

Montrer que : $(P \rightarrow (Q \rightarrow P))$ est valide?

1^o) TABLE DE VERITE

P	Q	$Q \rightarrow P$	$P \rightarrow (Q \rightarrow P)$
V	V	V	V
V	F	V	V
F	V	F	V
F	F	V	V

La fbf est vraie selon toutes les interprétations donc elle est valide

42

2°) LOIS D'EQUIVALENCE

$$\begin{aligned}
 (P \rightarrow (Q \rightarrow P)) &\equiv (P \rightarrow (\neg Q \vee P)) \\
 &\equiv (\neg P \vee (\neg Q \vee P)) \\
 &\equiv (\neg P \vee (P \vee \neg Q)) \\
 &\equiv (\neg P \vee P) \vee \neg Q \\
 &\equiv V \vee \neg Q \\
 &\equiv V
 \end{aligned}$$

La fbf est toujours vraie donc elle est valide

43

Si G est vraie dans l'interprétation I on dit que

I satisfait G ou **I est un modèle de G**

Deux fbfs F et G sont **équivalentes** \Leftrightarrow les valeurs de vérité de F et de G sont les mêmes dans toute interprétation. Notation $F \equiv G$

CONSEQUENCE LOGIQUE

G est une conséquence logique de $F_1.. F_n$

\Leftrightarrow tout modèle de $F_1..F_n$ est un modèle de G

$\Leftrightarrow F_1 \wedge F_2 \wedge \dots \wedge F_n \rightarrow G$ est valide

$\Leftrightarrow F_1 \wedge F_2 \wedge \dots \wedge F_n \wedge (\neg G)$ est inconsistent

44



Littéral = atome ou négation d'un atome

FORMES NORMALES

Une fbf est mise sous **forme normale conjonctive** (fnc) \Leftrightarrow elle est de la forme :

$F_1 \wedge F_2 \wedge \dots \wedge F_n$ où chaque F_i est une disjonction de littéraux

Une fbf est mise sous **forme normale disjonctive** (fnd) \Leftrightarrow elle est de la forme :

$F_1 \vee F_2 \vee \dots \vee F_n$ où chaque F_i est une conjonction de littéraux

Exemple

$(A \vee B) \wedge ((\neg C) \vee D) \wedge E$ fnc

$(A \wedge B \wedge (\neg C)) \vee D$ fnd

45



La représentation des connaissances

FORMALISATION D'UN MONDE = UNIVERS DU DISCOURS
= REFERENTIEL

BESOIN = REPRESENTER DES CONNAISSANCES
EXPRIMEES EN LANGAGE NATUREL

AIDE: GRAMMAIRE

(1) *Jean n'est pas grand*

--> (2) *Il est faux que Jean est grand*

ou **NON** *Jean est grand*

PASSAGE (1) -- (2)

SENS CONSERVE -- VALEURS DE VERITE
PAS EN LINGUISTIQUE

46

CONJONCTION

$$P \wedge Q \equiv P \text{ et } Q$$

Q et P

à la fois P et Q

P, Q

P bien que Q

P quoique Q

P mais Q (sous-entendu
mais aussi)

Non seulement P mais Q

P et pourtant Q

P tandis que Q

DISJONCTION

$$P \vee Q \equiv$$

P ou Q

ou P ou Q

ou bien P ou bien Q

soit P soit Q

P à moins que Q

P sauf si Q

P ou Q ou les deux (OU
inclusif)

47

CONDITIONNEL

$$P \rightarrow Q \equiv$$

si P alors Q

P condition suffisante
de Q

Q condition nécessaire
de P

P alors Q

Q si P

Q lorsque P

P seulement si Q

Q pourvu que P

...

EQUIVALENCE

$$P \leftrightarrow Q \equiv$$

P si et seulement si Q

P si Q et Q si P

P condition nécessaire et
suffisante de Q

...

TRADUCTION LIEE AU CONTEXTE
PLUSIEURS TRADUCTIONS POSSIBLES

48

Exemples



- ✍ Jean et Pierre prirent le café et Gustave fit de même.
- ✍ Jean prit le café, et Pierre ou Gustave aussi
- ✍ Jean et Pierre ont dîné tous les deux, ou bien Jean et Gustave prirent le café
- ✍ Jean a dîné, ainsi que Gustave ou Pierre
- ✍ Pierre étudie bien à moins qu'il ne soit fatigué, auquel cas non.

49

UNIVERS DU DISCOURS



- J: Jean prend le café
- P: Pierre prend le café
- G: Gustave prend le café
- D: Jean a dîné
- E: Gustave a dîné
- F: Pierre a dîné
- ETUDIE: Pierre étudie bien
- FATIGUE: Pierre est fatigué

50



Jean et Pierre prirent le café et Gustave fit de même

$$J \wedge P \wedge G$$

Jean prit le café, et Pierre ou Gustave aussi

$$J \wedge (P \vee G)$$

Jean et Pierre ont dîné tous les deux, ou bien Jean et Gustave prirent le café

$$(D \wedge F) \vee (J \wedge G)$$

Jean a dîné, ainsi que Gustave ou Pierre

$$D \wedge (E \vee F)$$

Pierre étudie bien à moins qu'il ne soit fatigué, auquel cas non

$$\neg \text{ETUDIE} \leftrightarrow \text{FATIGUE}$$

51



Règles d'inférences, axiomes, théorèmes

REGLE D'INFERENCE

= représentation d'un procédé pour, à partir d'une ou de plusieurs fbfs dériver d'autres fbfs

Modus Ponens:

$$A, A \rightarrow B \vdash B$$

AXIOMES = fbfs choisies initialement

Tout système reposant sur des axiomes peuvent s'interpréter "Si ces expressions que j'appelle axiome sont correctes alors voici le système."

====> on obtient différentes théories en changeant les axiomes

DEDUCTION A PARTIR DE h_1, h_2, \dots, h_n

= toute suite finie de formules f_1, f_2, \dots, f_p telles que pour tout $i \in \{1, 2, \dots, p\}$

a) f_i est un axiome

ou bien

b) f_i est l'une des formules h_1, h_2, \dots, h_n

ou bien

c) f_i est obtenue par application d'une règle $r_k \in R_s$ à partir des formules $f_{i_0}, f_{i_1}, \dots, f_{i_l}$ placées avant f_i (dans la démonstration)

52

Théorème

= toute formule t pour laquelle il existe une déduction à partir de χ (ensemble vide). Noté: $\vdash t$

Une déduction à partir de χ est appelée **déduction ou démonstration**; il s'agit d'une suite de formules $f_1 \dots f_p$ telles que pour tout $i \in \{1, 2, \dots, p\}$

a) f_i est un axiome ou

b) $f_{i_0}, f_{i_1}, \dots, f_{i_p} \vdash f_i$
 r_k

Cette formule se lit f_i se déduit à partir des fbfs $f_{i_0}, f_{i_1}, \dots, f_{i_p}$ grâce à la règle d'inférence r_k

DEMONSTRER UN THEOREME

= UTILISER UNE PROCEDURE DE DEMONSTRATION QUI ELLE-MEME
UTILISE DES REGLES D'INFERENCES

NON EQUIVALENT A MONTRER LA VALIDITE D'UNE EXPRESSION

= ACTIVITE SYNTAXIQUE INDEPENDANT DU SENS DES EXPRESSIONS

53

Système formel MIU

ALPHABET $E_S = \{M, I, U\}$

FORMULE $F_S = \{\text{toute suite finie d'éléments de } E_S\}$

AXIOME $A_S = \{MI\}$

REGLES D'INFERENCES

$R_S = \{r_1, r_2, r_3, r_4\}$

$r_1: f \vdash fU$

$r_2: Mf \vdash Mff$

$r_3: fIIIg \vdash fUg$

$r_4: fUUg \vdash fg$

54

Exemple

	f1: MI	axiome
	f2: MII	r2
	f3: MIII	r2
dérivation	f4: MIIIIU	r1
	f5: MUIU	r3
✓	f6: MUIUUIU	r2
	f7: MUII U	r4

MUIIU est un théorème

|---_s MUIIU

55

La logique des prédicats

Marie-Pierre Gleizes

56

La logique des prédicats



LOGIQUE DES PROPOSITIONS LIMITEE

✂ Manipulation de propriétés générales un peu complexes, de relations entre des objets

"Socrate est un homme" \Rightarrow SOCRATE_HOMME

"Platon est un homme" \Rightarrow PLATON_HOMME

\Rightarrow deux assertions distinctes et aucune similitude entre Socrate et Platon

\Rightarrow meilleure représentation
HOMME(Socrate), HOMME(Platon)

COMPOSANTS DE LA LOGIQUE DES PREDICATS:

PREDICATS, VARIABLES, FONCTIONS, CONSTANTES

PREDICATS exprime RELATION dans un CONTEXTE

Quelqu'un a chanté quelque chose

\Rightarrow A_CHANTE

+ 2 termes celui qui chante et ce qu'il chante

\Rightarrow A_CHANTE(MARIA_CALLAS, TRAVIATA)

FORMULE ATOMIQUE = PREDICAT + TERMES

57

besoin d'exprimer des relations ou propriétés, des fonctions



Le frère de Paul travaille avec le frère de Jacques

\rightarrow TRAVAILLER(frere(paul), frere(jacques))

frere = symbole de fonction

Dans N:

succ(X) \leftarrow fonction de N \rightarrow N

inf(X, Y) \leftarrow relation de N2 \rightarrow V, F

VERBE = prédicat en général

SUJET OU OBJET = terme

MAIS

La maison est verte

\rightarrow EST_VERTE(maison)

\rightarrow COULEUR(maison, verte)

VALEUR(couleur, maison, verte)

CONCEVOIR UNE REPRESENTATION

=

SELECTIONNER UN ALPHABET DES PREDICATS ET DES TERMES

+

DEFINIR LA SIGNIFICATION

besoin de quantifier

"Tous les hommes sont mortels"

58

Alphabet du langage

- CONNECTEURS** $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
- DELIMITEURS** $(), ,$
- CONSTANTES** (minuscules de l'alphabet latin et les concaténations de telles lettres)
 $C = \{a, b, c, \dots, z, aa, \dots\}$
- VARIABLES** (majuscules de l'alphabet latin, et les concaténations de telles lettres)
 $V = \{A, B, Z, AA, \dots\}$
- PREDICATS** (majuscules P)
 ARITE = nombre d'arguments d'un prédicat
 Arité d'un prédicat = un nombre positif
 Prédicat d'arité 0 = PROPOSITION
- FONCTIONS** (en minuscules: f, g successeur)
 Chaque symbole de fonction a une arité fixée
 ARITE d'une fonction = nombre strictement positif.
 Fonction d'arité 0 = CONSTANCE
- QUANTIFICATEURS**
 \exists "il existe" quantificateur existentiel
 \forall "quel que soit" quantificateur universel

59

Les termes

Par définition, tout **terme** est engendré par application des deux lois suivantes

- ✍ constantes et variables sont des termes
- ✍ si f est un symbole de fonction d'arité n ($n \geq 1$)
 et si $t_1..t_n$ sont des termes
 alors $f(t_1..t_n)$ est un terme

Exemple:

successeur(X) est un terme
 poids(b) est un terme
 successeur(poids(b)) est un terme
 P(X, bleu) n'est pas un terme
 poids(P(X)) n'est pas un terme

60

Les atomes



Par définition, tout **atome** est engendré par application des deux lois suivantes

- ✍ propositions sont des atomes
- ✍ si P est un prédicat d'arité n ($n \geq 1$)
et si $t_1..t_n$ sont des termes
alors $P(t_1..t_n)$ est un atome.

Exemple:

$P(X, \text{bleu})$ est un atome

VIDE est un atome

ENTRE(table, X , appui(fenetre)) est un atome

successeur(X) n'est pas un atome

appui(fenetre) n'est pas un atome

61

Formules bien formées : FBF ou WFF



Par définition, toute **fbf** est engendrée par application des trois lois suivantes

- ✍ atomes sont des fbfs
- ✍ si F et G sont des fbfs alors $(\neg G)$ $(F \wedge G)$ $(F \vee G)$ $(F \rightarrow G)$ $(F \leftrightarrow G)$ sont des fbfs
- ✍ si G est une fbf et X une variable alors $(\exists X)G$ et $(\forall X)G$ sont des fbfs.

Exemple:

$(\exists X) (\forall Y) (P(X, Y) \vee Q(X, Y) \rightarrow R(X))$

et $((\neg(P(a) \rightarrow P(b))) \rightarrow \neg P(b))$ sont des fbfs

$(\neg(f(a)))$

et $f(P(a))$ ne sont pas des fbfs

Ordre de priorité des connecteurs

(Le plus prioritaire) $\neg, \exists, \forall, \wedge, \vee, \rightarrow, \leftrightarrow$

62

Définitions

Champ ou portée d'un quantificateur = la fbf sur laquelle il s'applique

$$\forall X \exists Y (P(X, Y) \wedge \exists Z Q(X, Z))$$

le champ de

$$\forall X \text{ est } \exists Y (P(X, Y) \wedge \exists Z Q(X, Z))$$

$$\exists Y \text{ est } (P(X, Y) \wedge \exists Z Q(X, Z))$$

$$\exists Z \text{ est } Q(X, Z)$$

Occurrence de X est **liée** dans une fbf si elle est dans le champ d'un quantificateur " ou \$ qui l'utilise ou si elle le suit sinon cette occurrence est dite **libre**

$$A = \forall X (\exists Y P(X, Y) \wedge Q(X, Z)) \wedge R(X)$$

liée liée liée liée liée libre libre

$$B = \forall X ((\exists Y Q(X, Y)) \wedge P(X, Y, Z))$$

liée liée liée liée liée libre libre

Une **variable est libre** (resp. **liée**) si au moins une de ses occurrences est libre (resp. liée)

Variables libres de A = {Z, X}

Variables liées de A = {X, Y}

Variables libres de B = {Z, Y}

Variables liées de B = {X, Y}

FBF sans variable libre est dite **close ou fermée**

$$\forall X \exists Y (P(X, Y) \wedge \forall Z R(X, Y, Z))$$

liée liée liée liée liée

63

Sémantique

Une **interprétation** d'une fbf G est définie par les cinq étapes suivantes:

- 1- Choix d'un domaine d'interprétation non vide D
- 2- Assignation à chaque constante de G d'un élément de D
- 3- Assignation à chaque proposition de G d'un élément de $\{V, F\}$
- 4- Assignation à chaque prédicat d'arité n ($n \geq 1$) d'une application de D^n dans $\{V, F\}$
- 5- Assignation à chaque fonction d'arité ($n \geq 1$) d'une application de D^n dans D.

on dit alors qu'on a une interprétation de G sur D

64

Exemple

Soient les fbfs

$$G1: (\forall X) P(X)$$

$$G2: (\forall X) (\exists Y) Q(X, Y)$$

$$G3: (\forall X) (R(X) \wedge T(f(X), a))$$

Soit une interprétation I1 de G1

I1: D1={1, 2} où P(1) = **F** et P(2) = **V**

Soit une interprétation I2 de G2

I2: D2={1, 3} où Q(1, 1) = **F**

$$Q(1, 3) = \mathbf{V}$$

$$Q(3, 1) = \mathbf{F}$$

$$Q(3, 3) = \mathbf{F}$$

Soit une interprétation I3 de G3

I3: D1={4, 5} a = 4 f(4) = 5 f(5) = 4

$$R(4) = \mathbf{V}$$

$$R(5) = \mathbf{F}$$

$$T(4, 4) = \mathbf{V}$$

$$T(5, 4) = \mathbf{V}$$

65

Valeur selon une interprétation

Soit une interprétation I de domaine D d'une fbf G :

- 1- Si G est une proposition alors la valeur qui lui est assignée par définition de I est appelée valeur de G selon I (ou dans I)
- 2- Si G est un littéral non propositionnel alors pour chaque choix de valeurs dans D pour les variables de G (s'il en existe) on obtiendra une valeur V ou F en suivant la définition de I. Cette valeur est dite: valeur de G selon I pour le choix des valeurs de variables.
Exemple: pour G3
 $T(f(X), a) = \mathbf{V}$ si $X = 4$ et $a = 4$
- 3- Si G est de la forme $(\forall X)G'$, la valeur de G sera **V** si la valeur de G' selon I pour toutes les valeurs de la variable (dans D) est **V** sinon la valeur de G sera **F**
Exemple: G1 est **F** selon l'interprétation I1
- 4- Si G est de la forme $(\exists X) G'$, la valeur de G sera **V** si la valeur de G' selon I pour au moins une valeur de X (dans D) est **V** sinon la valeur de G sera **F**
Exemple:
Valeur de $Q(X, Y)$ dans I2 est **V** quand $X = 1$ et $Y = 3$ donc
 $\exists Y Q(X, Y)$ est **V** selon I2 quand $X=1$

66

5- Si G est de la forme

$(\neg G')$ ou $(G' \wedge G'')$ ou $(G' \vee G'')$ ou $(G' \rightarrow G'')$ ou $(G' \leftrightarrow G'')$

Les connecteurs gardent la même sémantique qu'en calcul propositionnel. On définira la valeur de G selon I (quand les valeurs G' et G'' selon I seront définies) au moyen des tables de vérité.

REMARQUES

1- Il y a une infinité d'interprétations pour G

$G1 = \forall X \exists Y P(X, Y)$

Si $D = N$

I1: $P(X, Y)$ est équivalent à $X \geq Y$

$G1$ est V dans I1

I2: $P(X, Y)$ est équivalent à $X > Y$

$G1$ est F dans I2 si $X=0$ il n'existe pas $Y < 0$ dans N

2- On ne peut pas interpréter une fbf contenant des variables libres

$G4: Q(Y)$ Y libre dans G4

Soit I: D où $Q: D \in \{V, F\}$

Quel sens attribuer à la variable libre

soit constante

soit variable parcourant D $\Rightarrow \forall$

\Rightarrow On se limitera aux fbfs fermées

67

Si G est vraie selon une interprétation I on dit que **I satisfait G** ou que **I est un modèle de G**

Une fbf est **valide** \Leftrightarrow elle est vraie dans toute interprétation

Une fbf est **invalid** \Leftrightarrow elle n'est pas valide

Une fbf est **inconsistante** \Leftrightarrow elle est fausse dans toute interprétation

Une fbf est **consistante** \Leftrightarrow elle n'est pas inconsistante

Deux fbfs sont **équivalentes** \Leftrightarrow les valeurs de vérité de F et de G sont les mêmes dans toute interprétation

G est une **conséquence logique** de $F1..Fn$

\Leftrightarrow tout modèle de $F1..Fn$ est un modèle de G

$\Leftrightarrow F1 \wedge F2 \wedge \dots \wedge Fn \rightarrow G$ est valide

$\Leftrightarrow F1 \wedge F2 \wedge \dots \wedge Fn \wedge (\neg G)$ est inconsistent

68

Formules équivalentes

13. $(\forall X) G(X) = (\forall Y) G(Y)$
 $(\exists X) G(X) = (\exists Y) G(Y)$
14. $(\neg((\exists X) G(X))) = (\forall X) (\neg G(X))$
 $(\neg((\forall X) G(X))) = (\exists X) (\neg G(X))$
15. $(\forall X) (G(X) \wedge H(X)) = (\forall X) G(X) \wedge (\forall X) H(X)$
 $(\exists X) (G(X) \vee H(X)) = (\exists X) G(X) \vee (\exists X) H(X)$

ATTENTION:

$(\forall X) (G(X) \vee H(X))$ non équivalent à $(\forall X) G(X) \vee \forall X H(X)$
 $(\exists X) (G(X) \wedge H(X))$ non équivalent à $(\exists X) G(X) \wedge (\exists X) H(X)$

16. $(\forall X) F(X) \vee (\forall X) H(X) = (\forall X) F(X) \vee (\forall Y) H(Y)$
 $(\exists X) F(X) \vee (\exists X) H(X) = (\exists X) F(X) \vee (\exists Y) H(Y)$
 $(\forall X) F(X) \wedge (\forall X) H(X) = (\forall X) F(X) \wedge (\forall Y) H(Y)$
 $(\exists X) F(X) \wedge (\exists X) H(X) = (\exists X) F(X) \wedge (\exists Y) H(Y)$

69

Indécidabilité et semi-décidabilité de la logique des prédicats

Lorsqu'une formule ne contient pas de variable, on peut, comme en calcul propositionnel, en utilisant les tables de vérité, déterminer en un nombre fini d'opérations si cette formule est
 valide ou non
 inconsistante ou non.

La situation est plus complexe en présence de variables et donc de quantificateurs:
 il y a une infinité d'interprétations. On montre qu'il est impossible de proposer un algorithme général capable de décider en un nombre fini d'opérations de la validité ou de la non validité de n'importe quelle formule de la logique des prédicats du premier ordre.

On dit que la logique des prédicats est indécidable. (Théorème d'indécidabilité de Church)

Cependant, on peut proposer des algorithmes généraux pour décider de la validité de certaines familles de fbfs:

si la fbf est valide ils s'arrêteront
 si la fbf est non valide ils risquent de ne pas s'arrêter
 \Rightarrow la logique des prédicats est semi-décidable

Les principales techniques proposées sont:

le théorème d'Herbrand
 la méthode de Davis et Putman
 le principe de résolution

70

Traduction du langage naturel en logique des prédicats

UNIVERSELLE AFFIRMATIVE

Tous les F sont des G

$$\forall X (F(X) \rightarrow G(X))$$

Tout F est G

Tout ce qui est F est G

N'importe lequel F est G

Les F sont tous G

Si un être quelconque est F, il est G

Chaque F est G

Seuls les G sont F

UNIVERSELLE NEGATIVE

Aucun F n'est G

$$\forall X (F(X) \rightarrow \neg G(X))$$

Il n'y a aucun F et G

Rien n'est à la fois F et G

Les F et G n'existent pas

PARTICULIERE AFFIRMATIVE

Quelques F sont G

$$\exists X (F(X) \wedge G(X))$$

Quelque F est G

Il y a des F et G

Quelque chose est à la fois F et G

Il y a un F et G

Des F et G existent

PARTICULIERE NEGATIVE

Quelques F ne sont pas G

$$\exists X (F(X) \wedge \neg G(X))$$

Quelque F n'est pas G

Il y a des F et non G

Quelque chose est à la fois F et non G

Il y a un F et non G

Des F et non G existent

71

Exemple

- Marcus était un homme
 - Marcus était un pompéien
 - Tous les pompéiens étaient des romains
 - César était souverain
 - Tous les romains étaient fidèles à César, soit le haïssaient
 - Chacun est fidèle à quelqu'un
 - Les gens n'essayent d'assassiner que les souverains auxquels ils ne sont pas fidèles
 - Marcus a essayé d'assassiner César
- D = ensemble des êtres humains

HOMME(X): X est un homme

POMPEIEN(X): X est pompéien

SOUVERAIN(X): X est souverain

ROMAIN(X): X est romain

PERSONNE(X): X est une personne


FIDELE(X, Y): X est fidèle à Y

HAIR(X, Y): X hait Y

ESSAYER_ASSASSINER(X, Y): X essaye d'assassiner Y

marcus, césar : constantes

72



a) Marcus était un homme
HOMME(marcus)

b) Marcus était un pompéien
POMPEIEN(marcus)

c) Tous les pompéiens étaient des romains
 $\forall X (POMPEIEN(X) \rightarrow Romain(X))$

d) César était souverain
SOVERAIN(césar)

e) Tous les romains étaient fidèles à César, soit le haïssaient
 $\forall X (ROMAIN(X) \rightarrow FIDELE(X, césar) \vee HAIR(X, césar))$


ou
 $\forall X (ROMAIN(X) \rightarrow (FIDELE(X, césar) \vee HAIR(X, césar)) \wedge \neg(FIDELE(X, césar) \wedge HAIR(X, césar)))$

f) Chacun est fidèle à quelqu'un
 $\forall X \exists Y FIDELE(X, Y)$

g) Les gens n'essayent d'assassiner que les souverains auxquels ils ne sont pas fidèles
 $\forall X \forall Y (PERSONNE(X) \wedge SOVERAIN(Y) \wedge ESSAYER_ASSASSINER(X, Y) \rightarrow \neg FIDELE(X, Y))$


h) Marcus a essayé d'assassiner César
ESSAYER_ASSASSINER(marcus, césar)

73



UNIVERSITE
PAUL
SABATIER
TOULOUSE III

Cours Module 15
Intelligence Artificielle
MIAGe



Le principe de résolution

Marie-Pierre Gleizes

74

Préparation des formules pour la résolution



Objectif :Transformer une fbf sous la forme "ensemble de clauses" pour faciliter le travail

 MISE SOUS FORME PRENEXE

Une fbf en logique des prédicats est dite en forme normale prénexe (fnp) ssi elle est de la forme:

$$(Q_1 X_1) (Q_2 X_2) \dots (Q_n X_n) M(X_1, X_2, \dots, X_n)$$

PREFIXE

MATRICE

où chaque (Qi Xi) est soit $\forall Xi$ soit $\exists Xi$ et où M est une fbf ne contenant aucun quantificateur.

$$\forall X \exists Y (P(X, Y) \rightarrow P(Y, X))$$
$$\forall X \exists Y \forall Z (P(X) \wedge \neg Q(Y, Z) \wedge P(f(Y)))$$
 sont en fnp

$\forall X P(X) \wedge \exists Y Q(X, Y)$ n'est pas en fnp

Par application successive des théorèmes sur les paires (fbfs) équivalentes (lois d'équivalence) on peut trouver une fbf G' en fnp équivalente à une fbf G donnée.

75

Méthode de transformation d'une fbf en fnp



1) Eliminer les connecteurs \rightarrow et \leftrightarrow

$$(G \leftrightarrow F) \quad \equiv \quad (G \rightarrow F) \wedge (F \rightarrow G)$$
$$(G \rightarrow F) \quad \equiv \quad (\neg G \vee F)$$

2) Accoler les connecteurs \neg aux atomes concernés

$$\neg(\neg G) \quad \cong \quad G \quad \neg(F \wedge G) \quad \cong \quad \neg F \vee \neg G$$
$$\neg(F \vee G) \quad \equiv \quad \neg F \wedge \neg G \quad \neg((\forall X) P(X)) \quad \equiv \quad (\exists X) \neg P(X)$$
$$\neg((\exists X) P(X)) \quad \equiv \quad (\forall X) \neg P(X)$$

3) Rebaptiser les variables liées si nécessaire de sorte que chaque quantificateur gouverne une variable originale

$$(\forall X) P(X) \quad \equiv \quad (\forall Y) P(Y)$$
$$(\exists X) P(X) \quad \equiv \quad (\exists Y) P(Y)$$

4) Déplacer tous les quantificateurs à gauche de la formule (sans changer l'ordre relatif)

$$((Q1 \ X) \ F(X)) \vee ((Q2 \ Y) \ H(Y)) \quad \equiv \quad (Q1 \ X) \ (Q2 \ Y) \ (F(X) \vee H(Y))$$
$$((Q1\ X)\ F(X)) \wedge ((Q2\ Y)\ H(Y)) \quad \equiv \quad (Q1\ X)\ (Q2\ Y)\ (F(X) \wedge H(Y))$$

Au terme de ces 4 étapes on obtient une fnp de la fbf initiale qui lui est équivalente. On peut avoir diverses fnp pour une même fbf

76

Exemples : mise sous fnp

$$\begin{aligned}
 & (\forall X) P(X) \rightarrow (\exists X) Q(X) \\
 \equiv & \neg((\forall X) P(X)) \vee (\exists X) Q(X) \\
 \equiv & (\exists X) \neg P(X) \vee (\exists X) Q(X) \\
 \equiv & (\exists X) (\neg P(X) \vee Q(X))
 \end{aligned}$$

$$\begin{aligned}
 & (\exists X (P(X) \rightarrow Q(X))) \rightarrow (\forall X P(X) \rightarrow \exists X Q(X)) \\
 \equiv & (\neg(\exists X (P(X) \rightarrow Q(X)))) \vee ((\forall X) P(X) \rightarrow \exists X Q(X)) \\
 \equiv & (\neg(\exists X (\neg P(X) \vee Q(X)))) \vee ((\neg(\forall X P(X))) \vee \exists X Q(X)) \\
 \equiv & (\forall X (P(X) \wedge \neg Q(X))) \vee (\exists X (\neg P(X) \vee \exists X Q(X))) \\
 \equiv & (\forall X (P(X) \wedge \neg Q(X))) \vee (\exists X (\neg P(X) \vee Q(X))) \\
 \equiv & (\forall X (P(X) \wedge \neg Q(X))) \vee (\exists Y (\neg P(Y) \vee Q(Y))) \\
 \equiv & \forall X \exists Y ((P(X) \wedge \neg Q(X)) \vee (\neg P(Y) \vee Q(Y)))
 \end{aligned}$$

77

Mise sous forme standard de SKOLEM (1)

A partir d'une fnp G' d'une fbf G on peut produire une forme standard de Skolem par les transformations ci-après soit:

$$G' = (Q_1 X_1) (Q_2 X_2) \dots (Q_n X_n) M(X_1, X_2, \dots, X_n) \text{ fnp de } G$$

1) Eliminer les quantificateurs existentiels

Soit Q_r un existentiel dans le préfixe de G'

(On opère habituellement de la gauche vers la droite mais l'ordre importe peu)

a) Si aucun quantificateur universel n'apparaît avant Q_r c'est-à-dire dans $(Q_1 X_1) \dots (Q_{r-1} X_{r-1})$

✍ on choisit un symbole de constante c différent de toute constante apparaissant dans la matrice M

✍ on supprime $Q_r X_r$ du préfixe

✍ on remplace X_r par c dans la matrice M

$$(\exists X) (\forall Y) (P(X) \vee Q(Y))$$

Choix de la constante a/X ce qui donne

$$(\forall Y) (P(a) \vee Q(Y)) \text{ fss}$$

78

Mise sous forme standard de SKOLEM (2)



b) Si $Qs_1 Qs_2 \dots Qs_m$ sont m quantificateurs universels apparaissant avant Qr dans le préfixe

- ✍ on choisit un symbole de fonction f d'arité m différent de toute fonction apparaissant dans la matrice M
- ✍ on supprime $QrXr$ du préfixe
- ✍ on remplace tout Xr dans M par $f(Xs_1, Xs_2, \dots, Xs_m)$

2) On itère le processus jusqu'à ce qu'il n'y ait plus de quantificateur existentiel dans le préfixe

Fonctions et constantes introduites sont appelées : constantes et fonctions de Skolem.

Nous obtenons une fbf dite : forme standard de Skolem

79

Mise sous forme standard de SKOLEM (3)



$$\exists X \exists Y \forall Z \forall T \exists V P(X, Y, Z, T, V)$$

1^o) étape a/X

$$\exists Y \forall Z \forall T \exists V P(a, Y, Z, T, V)$$

2^o) étape b/Y

$$\forall Z \forall T \exists V P(a, b, Z, T, V)$$

3^o) étape $f(Z,T)/V$

$$\forall Z \forall T P(a, b, Z, T, f(Z, T))$$

Théorème

Soit G_s la forme standard de Skolem d'une fbf G

G inconsistant ssi G_s inconsistant

80

Passage sous forme de clauses

Clause = toutes fbf qui ont la forme d'une disjonction de littéraux
Cas particulier: les littéraux isolés

$A \vee B \vee C \vee \neg D$	clause
$R(Z, a, g(X)) \vee (\neg T(V))$	clause
$P(X)$	clause

OBTENTION D'UN ENSEMBLE DE CLAUSE

Soit G_s une forme standard de Skolem d'une fbf G

1) Eliminer tous les quantificateurs

Il ne reste que des quantificateurs universels. On allège la notation en les supprimant.

On suppose donc désormais que toutes les variables sont quantifiées universellement.

2) Passer sous forme normale conjonctive

3) Eliminer les connecteurs \wedge

La conjonction de clauses obtenues au 2) est considérée comme un ensemble de clauses S

S est dite insatisfiable ou insatisfaisable pour dire S inconsistante

4) Distinguer les variables des clauses distinctes si c'est nécessaire

81

Remarques

1- Il existe en général, plusieurs formes standards d'une même formule.

On a intérêt à introduire des fonctions de Skolem aussi simples que possible

==> essayer de repousser le plus à gauche le quantificateur existentiel.

2- Si F peut s'écrire $F_1 \wedge F_2 \wedge \dots \wedge F_n$ un ensemble de clauses pour F peut être obtenu comme union des ensembles de clauses S_i de chaque F_i .

$$F = F_1 \wedge F_2 \wedge \dots \wedge F_n$$

$$fs = fs_1 \wedge \dots \wedge fs_n \text{ " SKOLEM}$$

$$S = S_1 \wedge \dots \wedge S_n \text{ " Clauses}$$

c'est-à-dire on fait clauses puis Skolem

$$\forall X P(X) \wedge \exists Y Q(Y)$$

Skolem	→	$\forall X \exists Y (P(X) \wedge Q(Y))$
--------	---	--

Clauses	→	$\forall X (P(X) \wedge Q(f(X)))$
---------	---	-----------------------------------

Ens. clauses	→	$S = \{P(X), Q(f(X))\}$
--------------	---	-------------------------

Clauses	→	$\forall X P(X)$	$\exists Y Q(Y)$
---------	---	------------------	------------------

Skolem	→	$\forall X P(X)$	$Q(a)$
--------	---	------------------	--------

Ens. clauses	→	$S = \{P(X), Q(a)\}$
--------------	---	----------------------

82

Résumé

- ✍ Fbf G équivalente à G' fnp de G
 - ✍ Fbf G non équivalente (si G est consistante) à G'' (fss) ou forme clausale on a uniquement si G est inconsistante alors G'' forme clausale inconsistante.
- Donc pour étudier la validité d'une fbf G on étudie l'inconsistance de $\neg G$ et donc de la fss de $\neg G$ on dit que l'on procède par REFUTATION

G VALIDE $\iff \neg G$ inconsistant
 \iff fnp ($\neg G$) inconsistante
 \iff fss ($\neg G$) inconsistante
 \iff S de ($\neg G$) insatisfiable

Dans les applications de la logique des prédicats, en général on veut montrer qu'une fbf H est conséquence logique de fbf $G_1 \dots G_n$ c'est-à-dire /

$G_1 \wedge \dots \wedge G_n \rightarrow H$ valide
 $\iff \neg(G_1 \wedge \dots \wedge G_n \rightarrow H)$ inconsistant
 $\iff G_1 \wedge \dots \wedge G_n \wedge \neg H$ inconsistant
 $\iff G_1 G_2 \dots G_n \neg H$ inconsistant

83

Le principe de résolution en logique des propositions

REGLE D'INFERENCE APPLIQUEE A DES CLAUSES

Clause résolvante

Si C1 et C2 sont 2 clauses et si $L_1 = \neg L_2$ et L1 est dans C1 et L2 est dans C2

C = la disjonction des clauses restantes après suppression des littéraux L1 et L2.
 = CLAUSE RESOLVANTE
 = résolvant de C1 et de C2 et L1 et L2 sont les littéraux résolus

$$\left. \begin{array}{l} C1: E1 \vee E2 \\ C2: \neg E2 \vee E3 \end{array} \right\} \longrightarrow C = E1 \vee E3$$

$$\left. \begin{array}{l} C1: P \\ C2: \neg P \end{array} \right\} \longrightarrow C = F$$

La clause Faux est notée χ c'est la **CLAUSE VIDE**
 Le résolvant C de deux clauses C1 et C2 est une conséquence logique de C1 et de C2

84

Déduction (ou résolution) d'une clause C à partir d'un ensemble de clauses S
= séquence finie $R_1, R_2, \dots, R_n = C$ de clauses telle que chaque R_i est:

- soit une clause de S
 - soit un résolvant de clauses le précédant
- $$S = \{R \vee Q, \neg R, \neg Q \vee P, \neg P \vee R\}$$
- $$R_1 = R \vee Q$$

S'il existe une déduction de χ à partir de S alors S est insatisfiable

Réfutation = déduction de χ à partir de S

ALGORITHME DE RESOLUTION

F valide $\iff \neg F$ inconsistant
 $\iff S \neg F$ insatisfiable
 \iff s'il existe une déduction de χ

- 1) Ecrire la négation de F
- 2) Mettre F sous forme d'un ensemble de clauses
- 3) Jusqu'à ce que χ soit rencontrée ou qu'il n'existe plus de paires réductibles chercher des clauses résolvantes et ajouter ce résultat à la liste des clauses
- 4) Si on trouve χ alors F est valide sinon F est invalide

85

EXEMPLE

Soit $S = \{\neg P \vee \neg Q \vee R, \neg R, P, \neg T \vee Q, T\}$
C1 C2 C3 C4 C5

$S \neg F$ est l'ensemble des clauses d'une fbf $\neg F$

$C_1 = \neg P \vee \neg Q \vee R$ } \Rightarrow C_6
 $C_2 = \neg R$

$C_6 = \neg P \vee \neg Q$ } \Rightarrow C_7
 $C_3 = P$

$C_7 = \neg Q$ } \Rightarrow C_8
 $C_4 = \neg T \vee Q$

$C_8 = \neg T$ } \Rightarrow χ \Rightarrow F valide
 $C_5 = T$

86

L'unification

GENERALISATION DU PRINCIPE DE RESOLUTION A LA LOGIQUE DES PREDICATS

=> ETENDRE DEFINITIONS DE "littéraux complémentaires" et de "résolvant"

Soient $C1 = P(X) \vee Q(a, X)$ et $C2 = \neg P(g(Y)) \vee R(Y, b)$

Peut-on dire des littéraux P et $\neg P$ qu'ils sont complémentaires? Peut-on définir un résolvant?

Si $[g(a)/X]$ et $[a/Y]$ on obtient:

$$C1' = P(g(a)) \vee Q(a, g(a))$$

$$C2' = \neg P(g(a)) \vee R(a, b)$$

$C1'$ et $C2'$ ont un résolvant $C = Q(a, g(a)) \vee R(a, b)$

C est conséquence logique de $C1'$ et de $C2'$ et donc de $C1$ et de $C2$. On peut trouver d'autres instances de $C1$ et de $C2$ $[g(g(b))/X]$ et $[g(b)/Y]$...

On peut continuer sur d'autres instances $C1$ et $C2$ dont les littéraux en P sont complémentaires.

Tous les résolvants obtenus sont des instances de $C = Q(a, X) \vee R(Y, b)$

Objectif: essayer de produire directement C qui paraît plus général. il peut être obtenu comme résolvant de $D1 = P(g(Y)) \vee Q(a, g(Y))$ instance de $C1$ $[g(Y)/X]$

et $D2 = \neg P(g(Y)) \vee R(Y, b)$ instance de $C2$

La substitution $[g(Y)/X]$ a permis de rendre les deux expressions en P symboliquement complémentaires

87

L'unification

Unification= procédé qui consiste à trouver des affectations de variables de façon à rendre des expressions identiques symboliquement

Substitution θ = ensemble fini de couples (t_i, V_i) où chaque V_i est une variable, chaque t_i un terme différent de V_i et où aucun couple d'éléments de l'ensemble n'a la même variable V_i

$$\theta_1 = \{ f(Z)/X, t/Y \}$$

$$\theta_2 = \{ a/X, f(g(a))/Y, g(t)/Z \}$$

Soit θ une substitution, E une expression. L'expression obtenue à partir de E en remplaçant simultanément toute occurrence V_i par t_i pour tout i est notée $E.\theta$ et est appelée **instance** de E

$$E = P(X, Y) \vee Q(g(X), Z)$$

$$\theta = \{ f(a)/X, g(t)/Y, t/Z \}$$

$$E.\theta = P(f(a), g(t)) \vee Q(g(f(a)), t)$$

88

Composition de θ et de σ

Soient $\theta = \{ t1/X1, t2/X2, \dots, tn/Xn \}$

$\sigma = \{ u1/Y1, u2/Y2, \dots, um/Ym \}$

la composition de θ et de σ (notée $\sigma \circ \theta$) = substitution obtenue à partir de $\{ t1.\sigma/X1, t2.\sigma/X2, \dots, tn.\sigma/Xn, u1/Y1, u2/Y2, \dots, um/Ym \}$ en éliminant les couples:

a) ui/Yi si $Yi \in \{ X1, X2, \dots, Xn \}$

b) $tj.\sigma/Xj$ si $Xj = tj.\sigma$

$\theta = \{ f(T)/X, Z/Y \}$

$\sigma = \{ a/X, b/T, Y/Z \}$

$\sigma \circ \theta = \{ f(T).\sigma/X, Z.\sigma/Y, a/X, b/T, Y/Z \}$

$= \{ f(b)/X, Y/Y, a/X, b/T, Y/Z \}$

$= \{ f(b)/X, b/T, Y/Z \}$

$E . (\sigma \circ \theta) = (E . \theta) . \sigma$

Substitutions interdites : $?$ /constante, $f(X)/X$, $?$ /fonction

La composition est associative et possède un élément neutre (substitution vide notée ϵ)

89

EXEMPLE

$E = P(X, Y) \vee Q(Y, Z) \vee R(Y, X)$

$\theta = \{ f(T)/X, Z/Y \}$

$\sigma = \{ a/X, b/T, Y/Z \}$

$E . \theta = P(f(T), Z) \vee Q(Z, Z) \vee R(Z, f(T))$

$(E . \theta) . \sigma = P(f(b), Y) \vee Q(Y, Y) \vee R(Y, f(b))$

$\sigma \circ \theta = \{ f(b)/X, b/T, Y/Z \}$

$E . (\sigma \circ \theta) = P(f(b), Y) \vee Q(Y, Y) \vee R(Y, f(b))$

90

Exemples

a)

$$\sigma = \{ a/X, g(Z)/Y, g(Z)/W \}$$

$$\theta = \{ f(Y)/X, b/T, c/Z \}$$

b)

$$\sigma = \{ c/V, f(Y)/Z \}$$

$$\theta = \{ a/X, V/Z \}$$

c)

$$\sigma = \{ a/Z, b/X, f(Z)/V \}$$

$$\theta = \{ f(X)/Z, b/X, g(Z)/V \}$$

91

Unifieur : Substitution θ est un unifieur de l'ensemble W des expressions $\{E_1, \dots, E_k\}$

$$\Leftrightarrow E_1 \cdot \theta = E_2 \cdot \theta = \dots = E_k \cdot \theta$$

W est dit unifiable.

σ est l'unifieur le plus général UPG \Leftrightarrow pour tout unifieur θ de W , il existe δ tel que: $\theta = \delta \circ \sigma$

$$W = \{ P(X, Y), P(f(T), Z) \}$$

$$\sigma_1 = \{ f(T)/X, Z/Y \} \text{ et } \sigma_2 = \{ f(T)/X, Y/Z \} \text{ upgs}$$

$$\theta = \{ f(a)/X, g(g(a))/Y, g(g(a))/Z, a/T \} \text{ unifieur}$$

$$\delta_1 \text{ tel que } \theta = \delta_1 \circ \sigma_1 \text{ est } \{ a/T, g(g(a))/Z \}$$

$$\delta_2 \text{ tel que } \theta = \delta_2 \circ \sigma_2 \text{ est } \{ a/T, g(g(a))/Y \}$$

ALGORITHME D'UNIFICATION :

consiste en la recherche d'un upg d'un ensemble d'expressions.

L'ensemble de discordance d'un ensemble non vide d'expressions est obtenu:

en localisant la première position à partir de la gauche pour laquelle toutes les expressions n'ont pas le même symbole et en extrayant dans chaque expression, la sous-expression qui commence en cette position

$$W_1 = \{ P(X, g(X), f(X, Y)), P(X, g(X), f(g(t, Y), Z)) \}$$

$$\text{Ensemble de discordance : } D_1 = \{ X, g(t, Y) \}$$

92

Algorithme

Soit W l'ensemble fini à unifier

✂ ETAPE 1: $k = 0$; $W_k = W$; $\sigma_k = \epsilon$ (ϵ est la substitution vide)

✂ ETAPE 2:

SI W_k est un singleton

ALORS σ_k upg de W

SINON trouver D_k ensemble de discordance de W_k ;

✂ ETAPE 3: SI il existe des éléments V_k et t_k de D_k tels que:

- V_k soit une variable

- t_k soit un terme ne contenant pas V_k

ALORS aller à l'étape 4

SINON W non unifiable

✂ ETAPE 4: $\sigma_{k+1} = \{ t_k/V_k \} \circ \sigma_k$;

$W_{k+1} = W_k \cdot \{ t_k/V_k \}$

✂ ETAPE 5: $k := k+1$;

aller à l'étape 2

SI W est un ensemble fini non vide unifiable alors l'algorithme s'arrête toujours à l'étape 2 et σ_k est upg.

93

Exemple

$W = \{ P(a, X, f(g(Y))), P(Z, f(Z), f(U)) \}$

1) $k = 0$; $W_0 = W$; $\sigma_0 = \epsilon$

2) $D_0 = \{ a, Z \}$

Z variable, a terme ne contenant pas Z

3) $\sigma_1 = \{ a/Z \} \circ \sigma_0 = \{ a/Z \}$

$W_1 = W_0 \cdot \{ a/Z \} = \{ P(a, X, f(g(Y))), P(a, f(a), f(U)) \}$

4) $D_1 = \{ X, f(a) \}$

$V_1 = X$, $t_1 = f(a)$ terme ne contenant pas X

5) $\sigma_2 = \{ f(a)/X \} \circ \{ a/Z \} = \{ f(a)/X, a/Z \}$

$W_2 = W_1 \cdot \{ f(a)/X \} = \{ P(a, f(a), f(g(Y))), P(a, f(a), f(U)) \}$

6) $D_2 = \{ g(Y), U \}$

$V_2 = U$, $t_2 = g(Y)$ terme ne contenant pas U

7) $\sigma_3 = \{ g(Y)/U \} \circ \sigma_2 = \{ g(Y)/U, f(a)/X, a/Z \}$

$W_3 = W_2 \cdot \{ g(Y)/U \} = \{ P(a, f(a), f(g(Y))) \}$

8) W_3 est un singleton $\Rightarrow \sigma_3$ est l'upg de W

W est unifiable

94

Le principe de résolution en logique des prédicats



FACTEUR D'UNE CLAUSE

Si deux ou plusieurs littéraux d'une même clause C (de même signe) ont un upg σ alors $C.\sigma$ est appelé facteur de C

$$\begin{aligned} C &= \neg P(Y) \vee \neg P(f(X)) \vee P(a) \vee Q(X, Y) \\ \{ \neg P(Y), \neg P(f(X)) \} &\text{ ont un upg } \sigma = \{ f(X)/Y \} \\ C.\sigma &= \neg P(f(X)) \vee P(a) \vee Q(X, f(X)) \end{aligned}$$

RESOLVANT BINAIRE

Soient C1 et C2, deux clauses qui n'ont aucune variable en commun, On dit que C1 et C2 sont des clauses parentes.

Soient L1 et L2, deux littéraux de C1 et de C2 resp.

Si L1 et $\neg L2$ ont un upg s alors la clause $(C1.\sigma - L1.\sigma) \cup (C2.\sigma - L2.\sigma)$ est appelée résolvant binaire de C1 et de C2

Les littéraux L1 et L2 = littéraux à résoudre

95

Résolvant



Un résolvant des clauses C1 et C2 est un des résolvants binaires suivants:

- ✚résolvant binaire de C1 et de C2
- ✚résolvant binaire de C1 et d'un facteur de C2
- ✚résolvant binaire d'un facteur de C1 et de C2
- ✚résolvant binaire d'un facteur de C1 et d'un facteur de C2

$$\begin{aligned} C1 &= P(a, X) \vee R(X) \\ C2 &= \neg P(T, f(T)) \vee Q(T, T) \\ \{ P(a, X), P(T, f(T)) \} &\text{ a un upg } \sigma = \{ a/T, f(a)/X \} \\ C1.\sigma &= P(a, f(a)) \vee R(f(a)) \\ C2.\sigma &= \neg P(a, f(a)) \vee Q(a, a) \\ C &= R(f(a)) \vee Q(a, a) \text{ résolvant binaire de C1 et de C2} \end{aligned}$$

1- Pour éviter des contraintes inutiles sur les variables ne pas oublier de travailler avec des clauses parentes

2 - Ne pas oublier d'utiliser les facteurs des clauses pour obtenir tous les résolvants

96

Théorèmes

Le résolvant C de deux clauses C1 et C2 est la conséquence logique de ces clauses.

Si C'1 et C'2 sont des instances de C1 et de C2 resp. et si C' est un résolvant de C'1 et de C'2
alors il existe un résolvant C de C1 et de C2 dont C' est une instance.

S insatisfiable \Leftrightarrow il existe une déduction par résolution de la clause vide
(La définition de la déduction est la même qu'en logique des propositions)

F valide \Leftrightarrow $\neg F$ inconsistant
 \Leftrightarrow $S \neg F$ insatisfiable
 \Leftrightarrow on peut déduire χ par résolution

97

Bibliographie

- ✍ BLANCHE Robert , Introduction à la logique temporelle, Collection U Armand Colin 1968
- ✍ DELAHAYE Jean-Paul, Outils logique pour l'intelligence artificielle, Editions eyrolles 1988
- ✍ FARGUES J, la logique mathématique et l'intelligence artificielle, AFCET Interfaces n°44 Juin pp 11-19
- ✍ MARCOUX A;, MAUREL C, Eléments de logique, Polycopié de cours de licence informatique à l'UPS 1988

98