EFREI 2011/2012
L3

Aide à la Décision
Recherche dans un espace d'états – Systèmes experts et Logique

Support de cours

Intelligence Artificielle

Introduction à la Logique et aux Systèmes Experts à base de règles

Cours « Aide à la Décision »
EFREI – 2011/2012 – L3

- Introduction
  - SBC
  - Raisonnement à base de règles
- La logique
  - Exemple
  - Types de logique / Formalisme
  - Logique des propositions
  - Logique des prédicats
  - « Unification » : un exemple
- Arbre « ET / OU »
- Chaînage avant
- Chaînage arrière
- Cas particuliers
- *Etude de cas : gestion des inscriptions à un congrès*

(C) Hervé Barbot, 2009-2011 2

Introduction

 3

Raisonnement à base de règles

- Principe
  - Trouver parmi les règles celles candidates
    règles « déclenchables »
  - Choisir une de ces règles
    stratégie / résolution de conflit
  - Exécuter / appliquer la règle
  - **Critère d'arrêt (fin du processus d'inférence)**
    - **Aucune règle déclenchable**
    - **Solution acceptable trouvée**
    - **Impossibilité de trouver une solution**
    - **…**

 4

Logique

Exemple introductif

 5

- Comment procéder pour déplacer 'A' sur 'B' ?

  En ne prenant que des pièces sur lesquelles rien n'est posé ?

(C) Hervé Barbot, 2009-2011 6

Exemple introductif

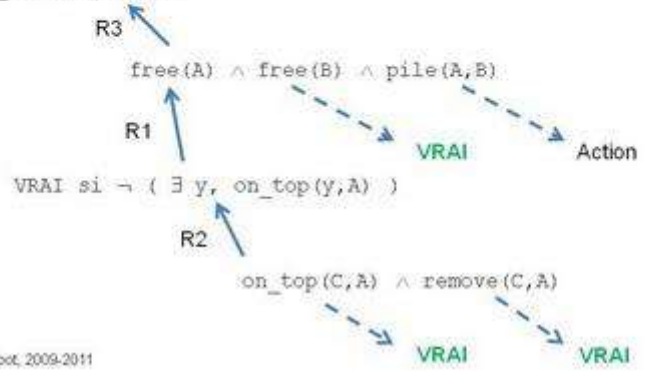
- Faits
  - on_top(C,A)
  - on_table(A) on_table(B)
  - free(B) free(C)

- Opérations / Déplacements de blocs (actions) :
  « remove » & « pile »
  - R1: free(x) ⇔ ¬ (∃ y, on_top(y,x))
  - R2: on_top(y,x) ∧ remove(y,x) ⇒ free(x) ∧ ¬ on_top(y,x)
  - R3: free(x) ∧ free(y) ∧ pile(x,y) ⇒ on_top(x,y)
- Comment procéder pour déplacer 'A' sur 'B' ?
  C.à d. comment obtenir « on_top(A,B) »?

(C) Hervé Barbot, 2009-2011 7

- Facts & Rules

  on_top(C,A) on_table(A) on_table(B) free(B) free(C)
  R1: free(x) ⇔ ¬ (∃ y, on_top(y,x))
  R2: on_top(y,x) ∧ remove(y,x) ⇒ free(x) ∧ ¬ on_top(y,x)
  R3: free(x) ∧ free(y) ∧ pile(x,y) ⇒ on_top(x,y)
- « on_top(A,B) » ?



(C) Hervé Barbot, 2009-2011 8

Logique

 9

La logique

- Formalisme / « langage » de représentation des connaissances
  - Logiques = langages formels pour représenter
    - l'information
    - le mécanisme de raisonnement
      - i.e. la méthode pour en tirer des conclusions
- Mécanisme de manipulation et de raisonnement sur l'information sémantique
  - contenue dans une phrase,
  - par manipulation des symboles constituant la phrase

 10

Types de logique

- Logique des propositions (« ordre 0 »)

  une suite de symboles séparés par des conjonctions (et), des disjonctions (ou) ou des négations (non)

  exemples:

  "Socrate est un homme" → HommeSocrate

  "Platon est un homme" → HommePlaton

  "Platon et Socrate sont des hommes"

  → HommeSocrate ∧ HommePlaton

- Logique des prédicats (« ordre 1 »)

  une suite de symboles, de variables et de relations avec des quantificateurs universels et existentiels

  exemples:

  – Homme(Socrate) ∀x Homme(x) ⇒ Mortel(x)

 11

Logique des Propositions

 12

- Permet d'exprimer :
  - Des faits sur le monde
    « Jean aime Marie »
  - Des négations
    « Marie n'aime pas Jean »
  - Des conjonctions et des disjonctions
  - Des phrases avec des « conséquences » logiques
    « si Jean n'aime pas Marie, elle ne l'aime pas non plus »

- Proposition = phrase
  dont la valeur de vérité est « vrai » ou « faux »

 13

Syntaxe

- Symboles de propositions :
  « P » « Q » « R »
- Phrases spéciales :
  « vrai » « faux »
- Operators (symbols) :
  ¬ ∧ ∨ not or and
  ⇒ IF ... THEN ...
  ⇔ ... IF AND ONLY IF ...
- Examples de phrases :
  $\neg (P \wedge Q) \vee (P \Rightarrow (Q \Rightarrow R))$
  $P \wedge Q \wedge (Q \Rightarrow R)$

 14

Table de vérité

P	¬P	Q	P ∧ Q	P ∨ Q	P ⇒ Q	P ⇔ Q
V	F	V	V	V	V	V
V	F	F	F	V	F	F
F	V	V	F	V	V	F
F	V	F	F	F	V	V

Pour déterminer la « valeur de vérité » d'une phrase

 15

Inférence

- Inférer = établir la valeur de vérité d'une phrase en appliquant des modifications syntaxiques à la phrase
  - La phrase est le fait à déduire / à prouver / à déterminer
    - Le problème posé !
  - Les modifications sont faites sur la base de « règles d'inférences »
    - Exemple

      Savoir :

      « A = true » « B = true » « (A ∧ B) ⇒ C »

      Inférer 'C' :

      « A » et « B » sont 2 phrases 'vraies',
      donc '(A∧B) ' est 'vrai'
      et puisque « (A∧B)⇒C » est également vrai
      par conséquence C ne peut qu'être vrai

      … cf la « table de vérité »

 16

Logique des Prédicats

 17

Syntaxe

- Prédicats
  Des « constantes », ou « fonctions »
  - Ex :
    Homme() Mortel()
- Termes
  Constantes manipulées au moyen des prédicats
  - Ex :
    Platon Socrate
- Phrase
  Combinaison de prédicats et de termes reliés par des opérateurs
  - Ex :
    Homme(Socrate) Homme(Platon)
    Homme(?x) ⇒ Mortel(?x)

(C) Hervé Barbot, 2009-2011 18

Table de vérité

- Valeur de vérité :
  par rapport à un modèle et une interprétation
  - Modèle :
    objets + relations entre les objets
  - Interprétation :
    Constantes → objets
    Prédicats → relations
  - Phrase atomique = vrai *ssi*
    les objets représentés par les termes entretiennent les uns par rapports aux autres la relation représentée par le prédicat

(C) Hervé Barbot, 2009-2011 19

Quantification

- Permet de définir une propriété sur un ensemble d'objets

∀x ...
quelle que soit la valeur(*) de 'x', « ... » est 'vrai'

∃x ...
il existe une valeur de 'x' telle que « ... » soit 'vrai'

(*) il ne s'agit bien évidemment pas ici de la « valeur de vérité »...

(C) Hervé Barbot, 2009-2011 20

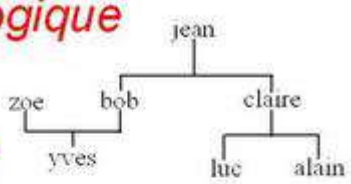
Lois de Morgan

...	est équivalent à	...
$\neg \exists x P$		$\forall x \neg P$
$\neg \forall x P$		$\exists x \neg P$
$\neg (P \vee Q)$		$\neg P \wedge \neg Q$
$\neg (P \wedge Q)$		$\neg P \vee \neg Q$

Permet de ré-écrire et manipuler les phrases

(C) Hervé Barbot, 2009-2011 21

Exemple: arbre généalogique



Représentation de l'environnement

père(yves,bob)
mère(yves,zoe)
père(bob,jean)
mère(luc,claire)
mère(alain,claire)
père(claire,jean)

- Savoir

Prédicats:

père(x,y)

père(yves,bob), père(claire,jean), ...

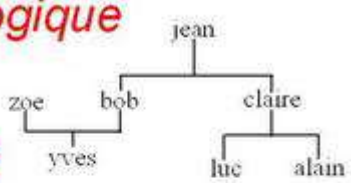
mère(x,y)

Permet la représentation complète des 'faits' connus

Ce qui n'est pas représenté n'est pas nécessairement 'faux', mais on ne le considèrera pas comme étant 'vrai'

(C) Hervé Barbot, 2009-2011 22

Exemple: arbre généalogique



Règles de manipulation

- Savoir-faire

Autres relations dérivées de « père » et de « mère » au moyen de règles d'inférence :

$père(x,y) \vee mère(x,y) \Rightarrow parent(x,y)$

Ou

$père(x,y) \Rightarrow parent(x,y)$

$mère(x,y) \Rightarrow parent(x,y)$

(C) Hervé Barbot, 2009-2011 23

Questions possibles :

- P(x) est-il 'vrai' ?
  Ex. grand_père(yves,jean) ?
- Quels sont les 'x' tels que P(x)=vrai ?
  Ex. parent(yves,?)

 24

Hypothèse de discussion pour la suite :
règles de la forme

IF A and B and ... THEN X

$(A \wedge B \wedge \ldots) \Rightarrow X$

 25

Graphe ou Arbre ET/OU

 26

Base de règles (« savoir-faire »)

$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$

Base de faits (« savoir »)

A = vrai
B = vrai

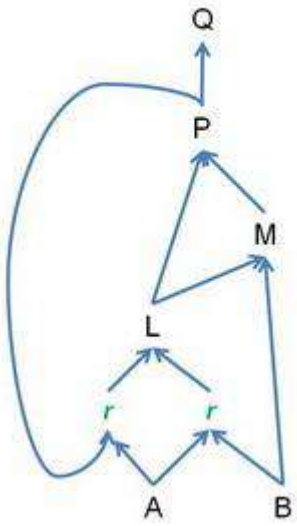
 27

Base de règles (« savoir-faire »)

$P \Rightarrow Q$
$L \wedge M \Rightarrow P$
$B \wedge L \Rightarrow M$
$A \wedge P \Rightarrow L$
$A \wedge B \Rightarrow L$

Base de faits (« savoir »)

A = vrai
B = vrai

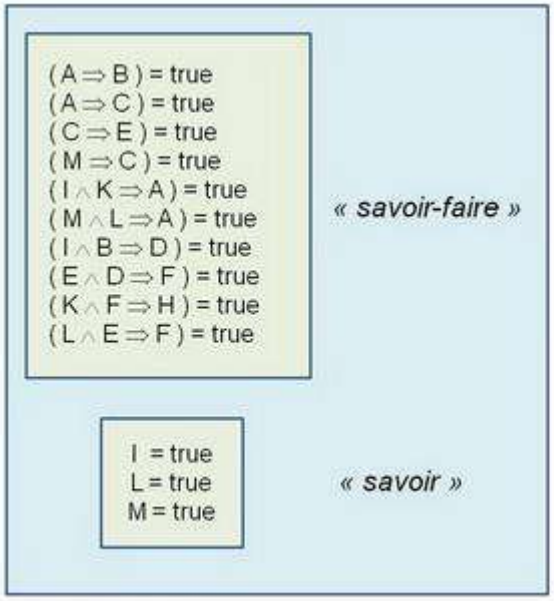


 28

Chaînage avant

Forward chaining

 29



(C) Hervé Barbot, 2009-2011 30

Infering rules:
R1: A→B
R2: A→C
R3: C→E
R4: M→C
R5: I&K→A
R6: M&L→A
R7: I&B→D
R8: E&D→F
R9: K&F→H
R10: L&E→F

Known facts:
I
L
M

Problème :

Selon :

Les <u>faits</u> que nous savons être vrais
Formant l'état initial de la recherche,
Représentant la description effective et actuelle du « monde »

Les <u>règles d'inférence</u>
Qui représentent le savoir-faire permettant de résoudre le problème

Quels sont tous les faits qui sont 'vrais' ?

(C) Hervé Barbot, 2009-2011 31

- Iterate on:
  - Select a rule that is applicable
    - That is for which the « conditions » part is true
  - Apply the rule
    - That is: add the conclusion in the « known facts » set
- Stop when there is no more applicable rule

- Applying a rule adds known facts
  - Thus, a rule not applicable at one time may be applicable later…

(C) Hervé Barbot, 2009-2011 32

Infering rules:
R1: A→B
R2: A→C
R3: C→E
R4: M→C
R5: I&K→A
R6: M&L→A
R7: I&B→D
R8: E&D→F
R9: K&F→H
R10: L&E→F

Initial facts = {I,L,M}

Rule applied	Knowledge	
	ILM	
R4	ILMC	
R6	ILMCA	
R1	ILMCAB	
R2	ILMCAB	*no new fact added!*
R3	ILMCABE	
R7	ILMCABED	
R8	ILMCABEDF	
R10	ILMCABEDF	*no new fact added!*

Order may differ depending on the strategy to select a rule at each step!

But the result is the same...

(C) Hervé Barbot, 2009-2011 33

```
facts_set forward_chaining(
               rules_set rules
               facts_set initial_facts)

rules_set to_fire ← rules
facts_set facts ← initial_facts

WHILE there_is_applicable_rule(to_fire,facts) DO

   rule r ← select_rule(to_fire,facts)
   to_fire ← to_fire - r
   facts ← facts + conclusion(r)

ENDWHILE

RETURN facts
```

(C) Hervé Barbot, 2009-2011 34

- Variants
  - Each time a rule is selected, add its conclusions in the knowledge base and fire (new) rules accordingly
    - See previous exemple and pseudo-code
  - Fire rules only based on the initial facts
    - No change to the knowledge base

(C) Hervé Barbot, 2009-2011 35

Chaînage arrière

Backward chaining

 36

Infering rules:
R1: A→B
R2: A→C
R3: C→E
R4: M→C
R5: I&K→A
R6: M&L→A
R7: I&B→D
R8: E&D→F
R9: K&F→H
R10: L&E→F

Known facts:
I
L
M

Problem statement :

According to:

Facts that we know are true
representing the initial state of the search,
Representing the actual description of the world

Infering rules (they are true as well!...)
representing the know-how of the problem

Is a given fact (e.g. F) true or not ?

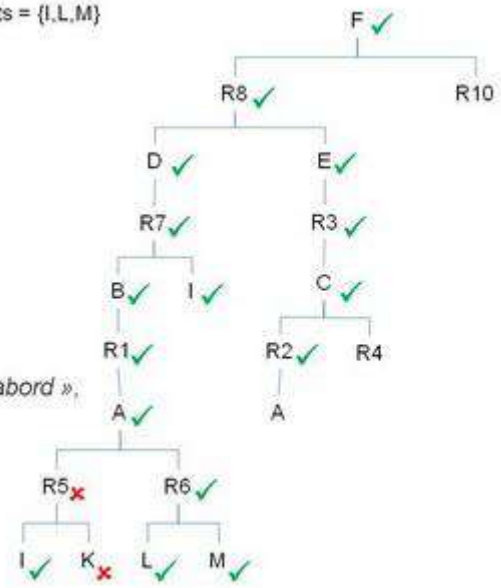
 37

- Iterate on all rules concluding to the fact:
  - Check if conditions are true
  - If yes, decide that the fact is true
- Checking if a condition is true is recursion!

 38

Infering rules:
R1: A→B
R2: A→C
R3: C→E
R4: M→C
R5: I&K→A
R6: M&L→A
R7: I&B→D
R8: E&D→F
R9: K&F→H
R10: L&E→F

Initial facts = {I,L,M}
F ?



Parcours « en profondeur d'abord », ici de gauche à droite.

 39

```
bool truth_value: fact      F       IN
                  rules_set rules   IN/OUT
                  facts_set facts   IN/OUT

IF F ∈ facts THEN RETURN true

WHILE there_is_a_concluding_rule(rules,F) DO

   rule r ← select_a_concluding_rule(rules,F)
   rules ← rules - r
   IF fire_ability(r,rules,facts)
          THEN facts ← facts + F
               RETURN true

ENDWHILE

RETURN false
```

 40

```
bool fire_ability: rule      R           IN
                   rules_set rules       IN/OUT
                   facts_set facts       IN/OUT

facts_set facts_to_check ← condition(R)

WHILE facts_to_check ≠ ∅ DO

   fact F ← select_fact(facts_to_check)
   IF truth_value(F,rules,facts) = false
          THEN RETURN false
   facts_to_check ← facts_to_check - F

ENDWHILE

RETURN true
```

 41

After the « basic » rules and chaining algorithms…

Let's see more issues!

Cas particuliers !

 42

« OR » in conditions

- IF A OR B THEN C
- IF A THEN C
  IF B THEN C

 43

« NOT »

- IF not A THEN B
  IF C THEN not B

Domain of value :

~~{ true , undetermined }~~

{ true , false , undetermined }

 44

General boolean expressions

- IF <bool_expression> THEN A

```
bool fire_ability: rule R                IN
                   rules_set rules       IN/OUT
                   facts_set facts       IN/OUT

      Evaluate according to the semantic tree of the
      boolean expression

RETURN eval_result
```

 45

Multiple conclusions

- IF A THEN X AND Y AND Z
- IF A THEN X OR Y
  - At least one of 'X' and 'Y' shall be set to 'true'
  - How to do if we don't know that X or Y is false?
- IF A THEN X XOR Y
  - Exactly one shall be true
  - How to do if both are 'undetermined'?

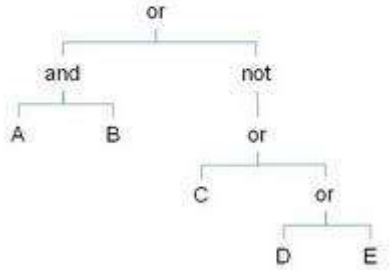
 46

More generally…

IF <bool_exp> THEN <bool_exp>

- Boolean expression leads to evaluation tree

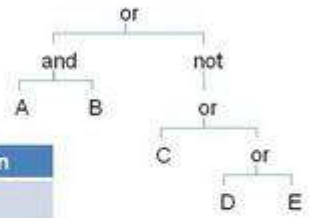
this is « simple » to evaluate as a « truth value » of a condition



 47

- But what about « conclusion » parts?

What to decide?



A	B	C	D	E	Boolean expression
✓	?	×	?	?	TRUE

- Is there any possible combination so that the truth table be correct?
- If several solutions, which one to select?
- What about indetermined facts?

(C) Hervé Barbot, 2009-2011 48

Interactive backward chaining

- Idea:
  ask the user to help the system

- Principle 1:
  the system is an expert, and shall investigate as much as possible before requiring any help

- Principle 2:
  ask as few help as possible

(C) Hervé Barbot, 2009-2011 49