Intelligence Artificielle

Logique et Systèmes Experts

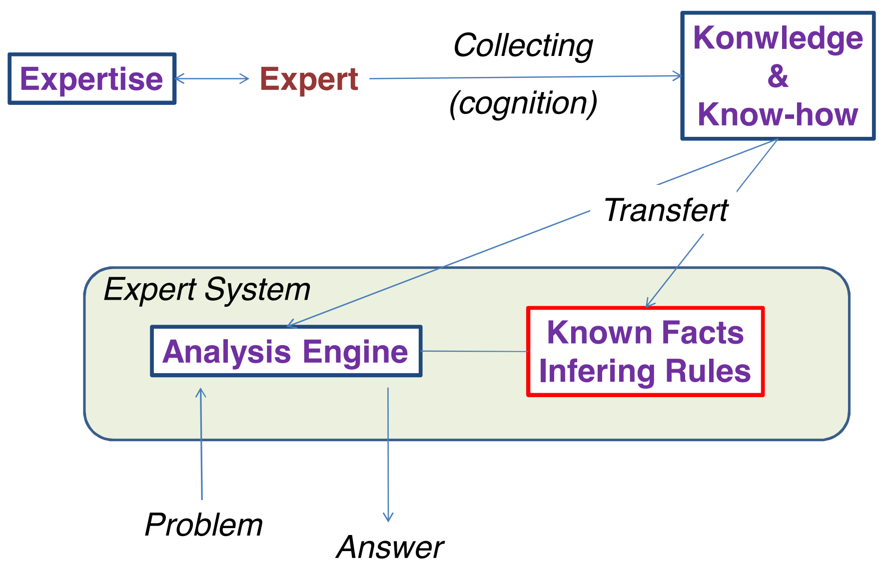
Introduction

System with Knowledge Base ?

- SBC (Système à Base de Connaissance)
  - Knowledge Base is related to a specific domain, a type of problem, …
  - Explicit representation of the knowledge

- Transfert of the knowledge
  - Implies a change of :
    - Media / support
    - Form
  - Linguistic modeling
    - Power of expression / statements
      - Is it possible to transfert the complete knowledge that the system shall take into account?
      - Is it possible to transfert all the knowledge that the expert owns?
    - Enables reasoning
      - Is the form well adapted to the processing that are required for solving a specific problem?
    - Efficient!

(C) Hervé Barbot, 2009 3

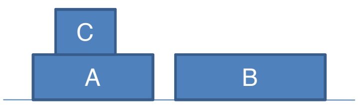


(C) Hervé Barbot, 2009 4

Logic

Introductory example

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



- Operations / Moving blocks:
  « remove » & « pile » actions
  - 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)
- How to move 'A' on top of 'B'?
  i.e. how to obtain « on_top(A,B) »?

- Facts & Rules

  `on_top(C,A) on_table(A) on_table(B) free(B) free(C)`

  R1: $free(x) \Leftrightarrow \neg (\exists y, on\_top(y,x))$

  R2: $on\_top(y,x) \wedge remove(y,x) \Rightarrow free(x) \wedge \neg on\_top(y,x)$

  R3: $free(x) \wedge free(y) \wedge pile(x,y) \Rightarrow on\_top(x,y)$
- « on_top(A,B) »?
  - R3 says: possible to obtain if we have « $free(A) \wedge free(B) \wedge pile(A,B)$ »
  - free(B)=true ; pile(A,B) is an action to execute
  - « free(A) »=?
    R1 says: yes if $\neg (\exists y, on\_top(y,A))$
  - We have on_top(C,A)! … How to obtain ¬on_top(C,A)?
    Need to apply R2:
    $on\_top(C,A) \wedge remove(C,A) \Rightarrow free(A) \wedge \neg on\_top(C,A)$
  - Thus: execute « remove(C,A) »:
    « $free(A) \wedge \neg on\_top(C,A)$ » becomes true
  - Then execute « pile(A,B) »:
    the goal « on_top(A,B) » becomes true!

(C) Hervé Barbot, 2009 7

Raisonnement à base de règles

- Principe
  - Trouver parmi les règles celles candidates
    règles « déclenchables »
  - Choisir une de ces règles
    stratégie / résolution de conflit
  - Exécuter la règle

  - Critère d'arrêt (fin du processus d'inférence)
    - Aucune règle déclenchable
    - Solution acceptable trouvée
    - Impossibilité de trouver une solution
    - …

(C) Hervé Barbot, 2009 8

- Base de règles
  - Connaissance + savoir-faire de l'expert
  - Prémisses + Conclusions
- Base de faits
  - Mémoire de travail du SBC
  - Variables en cours d'exécution
  - Base de faits initiaux
  - Faits déduits ou demandés à l'utilisateur

- Moteur d'inférence
  - Règle + faits = « ajouter » nouveaux faits
  - Différentes méthodes / catégories
    - Chaînage avant
    - Chaînage arrière
    - Chaînage mixte
  - Méta-règles

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 l phrase

Types de logique

- Logique des prédicats (« ordre 0 »)

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

  exemples:

  "Socrate est un homme" $\rightarrow$ HommeSocrate

  "Platon est un homme" $\rightarrow$ HommePlaton

  "Platon et Socrate sont des hommes"

  $\rightarrow$ HommeSocrate $\wedge$ HommePlaton

- Logique des propositions (« ordre 1 »)

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

  exemples:

  – Homme(Socrate) $\forall x$ Homme(x) $\Rightarrow$ Mortel(x)

Logique des Propositions

Syntaxe

- Propositions:
  « P » « Q » « R »
- Operators (symbols):
  $\neg \ \wedge \ \vee$ not or and
  $\Rightarrow$ IF … THEN …
  $\Leftrightarrow$ … IF AND ONLY IF …
- Examples:
  $\neg(P \wedge Q) \vee (P \Rightarrow (Q \Rightarrow R))$
  $P \wedge Q \wedge (Q \Rightarrow R)$

Truth table

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

Inference

- To infer = to establish the truth value of a sentence by applying syntactical modifications to the sentence
  - The sentence is the fact to prove
  - The modifications are made based on the « infering rules »
    - Example

      Knowledge

      A=true B=true (A∧B)⇒C

      Infering ‘C’

      yes because A and B are both true, and (A∧B)⇒C

Logique des Prédicats

Syntax

- Au lieu des symboles propositionnels P_1 , P_2 , Q …, utiliser des prédicats et des termes:
- exemples

1) Personne (Marie) Aime(Roméo, Juliette)

constantes "prédicats"
(ou simplement “prédicats”)
<u>sans signification prédéterminée</u>

constantes individuelles
(ou simplement “constantes”)
<u>sans signification prédéterminée</u>

2) “Jean donne Fifi à Marie”
Donne(Jean, Marie, Fifi)

3) Égal(Plus(2,2), 4)

Constantes	RoiJean, 2, UniGE, ...
Prédicats	Frère, >, ...
Fonctions	Sqrt(.), JambeGaucheDe(.), ...
Variables	x, y, a, b, ...
Connecteurs	$\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$
Égalité	=
Quantificateurs	$\forall$ $\exists$

(C) Hervé Barbot, 2009 19

Formula

Phrases atomiques = *prédicat(terme$_1$, ... , terme$_n$)*
ou *terme$_1$ = terme$_2$*

Terme = *fonction(terme$_1$, ... , terme$_n$)*
ou *constante* ou *variable*

Exemples

1) Frère(RoiJean, RichardCoeurDeLion)
2) > (Longueur(JambeGaucheDe(Richard)),15)

Phrases complexes = phrases atomiques réunies par des connecteurs

$\neg S$, $S_1 \wedge S_2$, $S_1 \vee S_2$, $S_1 \Rightarrow S_2$, $S_1 \Leftrightarrow S_2$

Exemples

1) Frère(RoiJean, Richard) $\Rightarrow$ Frère(Richard, RoiJean)
2) $> (1, 2) \vee \leq (1, 2)$
3) $> (1, 2) \wedge \neg > (1, 2)$

(C) Hervé Barbot, 2009 20

Quantification

- Used to define properties on sets of objects

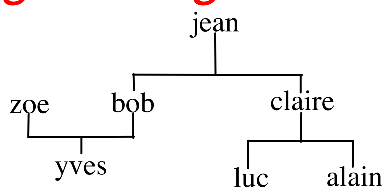
$\forall x$ …   whatever the value of x, … is true

$\exists x$ …   there exist a value of x such that … is true

Morgan law

…	*is the same as*	…
$\neg \exists x P$		$\forall x \neg P$
$\neg \forall x P$		$\exists x \neg P$
$\neg (P \vee Q)$		$\neg P \wedge \neg Q$
$\neg (P \wedge Q)$		$\neg P \vee \neg Q$

Example: genealogical tree



- Predicates:

  Father(x,y)

  father(yves,bob), father(claire,jean), …

  Mother(x,y)

  Allows a complete representation of the known facts

- Other relationships are derived from « father » and « mother » predicates by means of infering rules:

  father(x,y) ∨ mother(x,y) ⇒ parent(x,y)

  Or

  father(x,y) ⇒ parent(x,y)

  mother(x,y) ⇒ parent(x,y)

Hypothesis : rules such as

IF A and B and … THEN X

AND/OR Tree or Graph

- Basic example

rules

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

$B \wedge L \Rightarrow M$

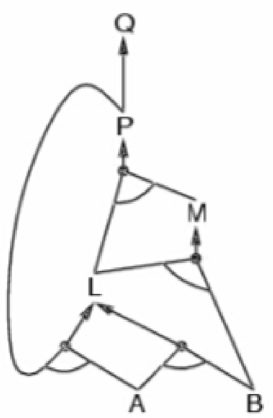
$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B

graph



Exercice

Construction d'un graphe ET/OU

Example

R1 : A -> B	R6 : M+L -> A
R2 : A -> C	R7 : I+B -> D
R3 : C -> E	R8 : E+D -> F
R4 : M -> C	R9 : K+F -> H
R5 : I+K -> A	R10 : L+E -> F

Construction de l'arbre « et / ou »

?

Chaînage avant

Forward chaining

$(A \Rightarrow B) = true$
$(A \Rightarrow C) = true$
$(C \Rightarrow E) = true$
$(M \Rightarrow C) = true$
$(I \wedge K \Rightarrow A) = true$
$(M \wedge L \Rightarrow A) = true$
$(I \wedge B \Rightarrow D) = true$
$(E \wedge D \Rightarrow F) = true$
$(K \wedge F \Rightarrow H) = true$
$(L \wedge E \Rightarrow F) = true$

« know-how »

Knowledge base

I = true
L = true
M = true

« know-what »

(C) Hervé Barbot, 2009 31

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

What other facts can we determine as true?

(C) Hervé Barbot, 2009 32

Lorsqu'un nouveau fait *p* est ajouté à la base de connaissances:

pour chaque règle telle que *p* s'unifie avec un prémisse
et si les autres prémisses sont connus
alors ajouter la conclusion à la base de connaissances et
continuer le chaînage

- le chaînage avant est piloté par les données (*"data-driven"*)

càd, on infère des propriétés et des catégories à partir des séquences perceptives

(C) Hervé Barbot, 2009 33

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 34

Chaînage arrière

Backward chaining

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

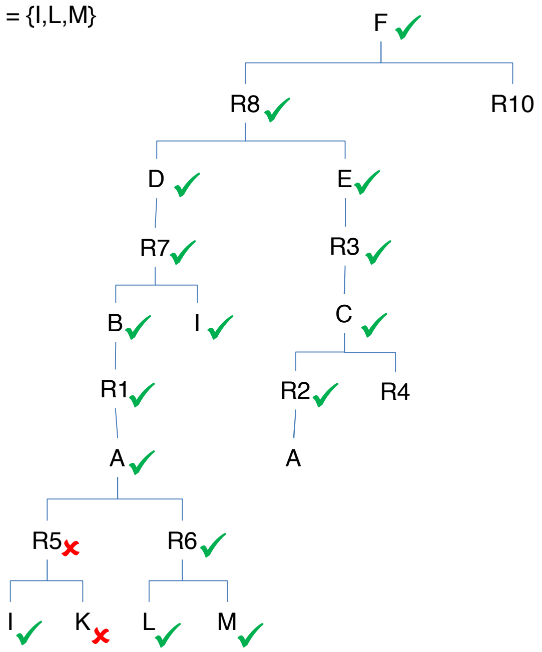
Is a given fact (e.g. F) true or not ?

- 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!

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 ?



If all rules are of the form :

IF a set of facts that shall be true
THEN a set of facts that will be considered as true

- i.e. only the « and » operator is used

$$\text{truth\_value}(F) = \mathrm{OR}_{R,\ F \in C(R)}\ \text{fire\_ability}(R)$$

$$\text{fire\_ability}(R) = \mathrm{AND}_{F,\ F \in P(R)}\ \text{truth\_value}(F)$$

C(R)=set of facts as conclusion of a rule R
P(R)=set of facts in the condition part of rule R

After the « basic » rules and chaining algorithms…

Let's see more issues!

« OR » in conditions

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

« NOT »

- Domain of value :
  { true , undetermined }
  or
  { true , false , undetermined }

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 true
```


Multiple conclusions

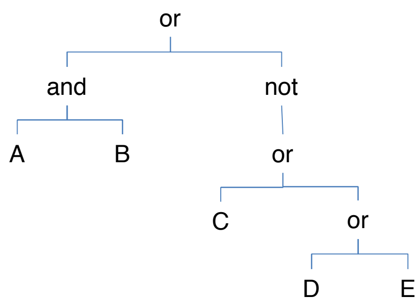
- IF A THEN X AND Y AND Z
- IF A THEN X OR Y
  - At least one of ‘X’ and ‘Y’ shall be set to ‘true’
  - How to do if we don’t know that X or Y is false?
- IF A THEN X XOR Y
  - Exactly one shall be true
  - How to do if both are ‘undetermined’?

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



- But what about « conclusion » parts?

  What to decide?

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

