Cours Système Expert

Lotfi Nabli
National Engineering School of Monastir

Institut Supérieur des Sciences Appliquées et de Technologie de Kairouan

Cours de Mastère Professionnelle

Pilotage et Réseaux Industriels

Cours :

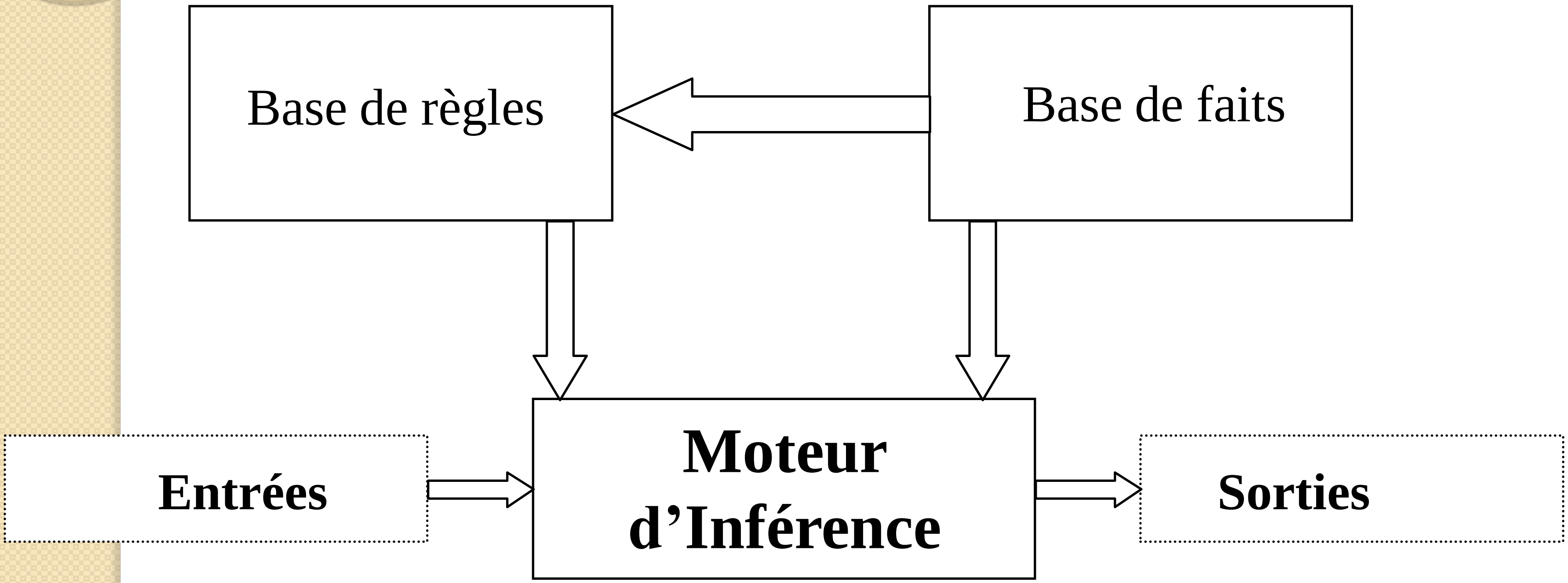
Système Expert et Applications

Systèmes Experts

Historique : Les systèmes experts ont eu leur heure de gloire dans les années 1980, où on a trop rapidement pensé qu'ils pourraient se développer massivement. En pratique, le développement de ce genre d'application est très lourd car, lorsque l'on dépasse la centaine de règles, il devient difficile de comprendre comment le système expert « raisonne » (manipule faits et règles en temps réel), et donc d'en assurer la mise au point finale puis la maintenance.

Définition : C'est un outil capable de reproduire les mécanismes cognitifs d'un expert, dans un domaine donné. Il s'agit de l'une des voies tentant d'aboutir à l'Intelligence Artificielle.

Systèmes Experts



1) Base de Connaissances

Définition : Une **base de connaissance** regroupe des connaissances spécifiques à un domaine spécialisé donné, sous une forme exploitable numérique.

Elle peut contenir des règles, dans ce cas on parle de base de règles, des faits ou d'autre représentations c'est-à-dire l'ensemble des propriétés générales de l'existant.

Si les algorithmes de manipulation de faits et de règles sont nombreux et connus, la détermination de l'ensemble des faits et règles qui vont composer la base de connaissances est un problème délicat.

Ainsi, comment décrire le comportement d'un expert face à un problème particulier, et sa manière de le résoudre, là est la question. Car ce que l'on souhaite obtenir n'est ni plus ni moins que l'expérience, la connaissance pratique de l'expert, et non la théorie que l'on peut trouver dans les livres ni exclusivement les règles logiques d'inférence.

Equivalents des méthodes d'analyse de l'informatique traditionnelle, des méthodes d'acquisition des connaissances sont développées.

2. Moteur d'Inférence

Définition : Le moteur d'inférence, du verbe « inférer » qui signifie « déduire », est capable d'utiliser faits et règles pour produire de nouveaux faits, jusqu'à parvenir à la réponse à la question experte posée.

Il existe de nombreux types de moteur, capables de traiter différentes formes de règles logiques pour déduire de nouveaux faits à partir de la base de connaissance

On distingue souvent trois catégories, basées sur la manière dont les problèmes sont résolus :

- Les moteurs - dits à « *chaînage avant* » - qui partent des faits et règles de la base de connaissance, et tentent de s'approcher des faits recherchés par le problème.
- Les moteurs - dits à « *chaînage arrière* » - qui partent des faits recherchés par le problème, et tentent par l'intermédiaire des règles, de « remonter » à des faits connus,
- Les moteurs - dits à « *chaînage mixte* » - qui utilisent une combinaison de ces deux approches *chaînage avant* et *chaînage arrière*.

- Certains moteurs d’inférence peuvent être partiellement pilotés ou contrôlés par des méta-règles qui modifient leur fonctionnement et leurs modalités de raisonnement.
- Dans un système expert à base de règles, le **chaînage avant** est une méthode de déduction qui applique des règles en partant des symptômes pour en déduire de nouvelles conclusions. Ces conclusions enrichissent la mémoire de travail et peuvent devenir les prémisses d’autres règles. Par opposition, le chainage arrière (en) part des conclusions pour essayer de « remonter » aux axiomes.

Exemples de moteurs d'inférence

- ***CLIPS*** **: Chaînage avant, Contrôle irrévocable, Moteur d'ordre 1, Logique non monotone, Monde fermé**
- ***Moteur de MYCIN :'*** **système de diagnostic médical,**
- ***GOSSEYN :*** **moteur d'ordre 1 en chaînage avant, développé en Fortran par Jean-Marc Fouet,**
- ***PROLOG II :*** **moteur d'ordre 1 en chaînage arrière, développé par Alain Colmerauer,**
- ***Kadviser :*** **moteur d'ordre 1 à propagation de contraintes, créé en 1988 par la société KADE-TECH et développé aujourd'hui par la société NIMTOTH,**
- ***MACISTE*** **moteur d'ordre 2, développé par Jacques Pitrat,**
- ***SMECI :*** **développé en LISP par la société ILOG,**
- ***SNARK :*** **développé par Jean-Louis Laurière.**

3. Raisonnement

Le **raisonnement** est un processus cognitif qui permet d'obtenir de nouveaux résultats ou de vérifier un fait en faisant appel à différentes "lois" ou expériences, quel que soit leur domaine d'application : système d’informations, système industrielle, mathématique, système juridique, système qualité, système pédagogique, etc.

Objectifs des raisonnements

On conduit des raisonnements pour des objectifs différents, qui peuvent se combiner :

- Prise de décision ;
- Test d'une argumentation ;
- Conduite d'une démonstration, d'un théorème, de la confirmation d'une hypothèse ;

On dit que l'individu effectue des inférences et que le mécanisme d'élaboration de ces inférences s'appelle raisonnement.

Les différents raisonnements

Un raisonnement s'appuyant davantage sur des règles formulées en langage mathématique sera dit plutôt rationaliste. Cette forme de raisonnement est prépondérante dans les systèmes avec modèle physique.

Un raisonnement s'appuyant davantage sur des expériences vécues sera dit plutôt empirique. Cette forme de raisonnement, qui n'exclut pas la rigueur, y compris dans la méthode expérimentale, est prépondérante dans les systèmes sans modèle physique.

Descartes affirmait : « Il n’y a pas d’autres voies qui s’offrent aux hommes, pour arriver à une connaissance certaine de la vérité, que l’intuition évidente et la déduction nécessaire ». Il admettait l'importance de l'intuition, qui fut confirmée par Spinoza et Bergson. Pourtant, toute intuition n'est pas forcément évidente tant qu'elle n'est pas partagée.

La logique générale s'appuie sur la tradition des syllogismes (voir logique).

Dans une logique mathématique (logique des propositions, logique des prédicats, logique modale, etc.), on s'accorde à considérer trois moyens de construction de raisonnements :

- la déduction : *raisonnement par déduction*
- l'induction : *raisonnement par induction*
- l'abduction : *raisonnement par abduction*

Elles se présentent schématiquement ainsi, en s'appuyant sur les notations classiques de la logique (→ pour l'implication) :

Déduction	Abduction	Induction
$\dfrac{a \quad a\to b}{b}$	$\dfrac{b \quad a\to b}{a}$	$\dfrac{a \quad b}{a\to b}$

La règle de la déduction se lit ainsi :

si *a* est vrai

et si *a est vrai alors b l'est aussi* est vrai

→ alors *b* est vrai.

La règle de l'abduction se lit ainsi:

si *b* est vrai

et si *a est vrai alors b l'est aussi* est vrai

→ alors *a* est vrai.

Le processus de construction d'un raisonnement simple consiste à appliquer au moins l'une de ces trois règles sur une théorie initiale ; c'est donc un moyen d'y ajouter de nouvelles propositions.

Un raisonnement est dit *déductif* s'il ne s'appuie que sur la règle de déduction ; il est dit *hypothétique* s'il s'appuie sur au moins l'une des règles d'abduction ou d'induction.

Seule la déduction conserve la cohérence d'une théorie : si la théorie initiale est cohérente, alors toute théorie qui en est une conséquence déductive reste cohérente.

Une n-conséquence déductive D d'une théorie initiale I est une théorie obtenue après application d'un nombre quelconque mais fini de déductions sur I.

La clôture déductive D d'une théorie initiale I est sa n-conséquence déductive, n étant infini.

Une théorie est dite maximalement cohérente si sa clôture déductive ne contient pas la proposition **faux**.

Dans la plupart des systèmes du calcul des propositions, on trouve les règles suivantes

modus ponens	modus tollens
$$\frac{a \qquad a \rightarrow b}{b}$$	$$\frac{\neg b \qquad a \rightarrow b}{\neg a}$$

Le *modus tollens* est considéré en général comme une règle dérivée. La déduction naturelle y ajoute des règles d'introduction et d'élimination. Le calcul des séquents ne considère que des règles d'introduction et en plus la règle de coupure.

4. Calcul des propositions

Introduction générale

La notion de *proposition* a fait l'objet de nombreux débats au cours de l'histoire de la logique ; l'idée consensuelle est qu'une proposition est une construction syntaxique pour laquelle il fait sens de parler de vérité.

Le **calcul des propositions** ou **calcul propositionnel** est une théorie logique qui définit les lois formelles du raisonnement. C'est la version moderne de la logique stoïcienne. C'est aussi la première étape dans la construction des outils de la logique mathématique.

En logique mathématique, le calcul des propositions est la première étape dans la définition de la logique et du raisonnement. Il définit les règles de déduction qui relient les propositions entre elles, sans en examiner le contenu ; il est ainsi une première étape dans la construction du calcul des prédicats, qui lui s'intéresse au contenu des propositions et qui est une formalisation achevée du raisonnement mathématique.

Si l'on se place dans la logique classique, le calcul de propositions est une structure algébrique que l'on appelle algèbre de Boole.

Définition d'une proposition

Quoique le calcul des propositions ne se préoccupe pas du contenu des propositions, mais seulement de leurs relations, il peut être intéressant de discuter ce que pourrait être ce contenu.

Une proposition donne une information sur un état de chose. Ainsi « 2 + 2 = 4 » ou « le livre est ouvert » sont deux propositions. En logique classique, une proposition peut prendre uniquement les valeurs *vrai* ou *faux*.

Une phrase optative (qui exprime un souhait comme par exemple « Que Dieu nous protège !», une phrase impérative (« viens !», « tais-toi !») ou une interrogation n'est pas une proposition. « Que Dieu nous protège !»ne pouvant être ni vraie ni fausse: elle exprime uniquement un souhait du locuteur. En revanche, une phrase comme « Dans ce calcul, *toutes les variables informatiques sont strictement positives* » est une proposition dont le contenu a été modifié par le quantificateur *toutes* et qui est supposée s'avérer dans la durée. Ce type de proposition est étudié dans la logique modale, plus précisément dans la logique temporelle dans ce cas, à cause l'affirmation de sa pérennité

Définition d'un système déductif

Un *calcul* ou un *système déductif* est un ensemble de règles permettant en un nombre fini d'étapes et selon des règles explicites de déterminer si une proposition est vraie. Un tel procédé s'appelle une *démonstration*.

On associe aussi aux propositions une structure mathématique qui permet de garantir que ces raisonnements ou démonstrations ont du sens, on dit qu'on lui a donné une sémantique. En calcul des propositions classique, cette sémantique est constituée des deux éléments *vrai* et *faux* (souvent notés *1* et *0*).

Structure

Dans les théories de la logique mathématique, on considère donc deux points de vue dits syntaxique et sémantique, c'est le cas en calcul des propositions.

Aspect syntaxique : il s’agit de définir le langage du calcul des propositions par les règles d’écriture des propositions.

Aspect sémantique : il s’agit ici de donner un sens aux symboles représentant les connecteurs logiques en fonction de la valeur de vérité des propositions de base (ainsi signifie non). Ce sens est donné, par exemple, par des tables de vérité ou par des modèles de Kripke.

Puis de décrire les règles d'inférence permettant la déduction de propositions à partir d'autres. Ces règles de déduction permettent d'engendrer des propositions spécifiques que l'on appelle des *théorèmes*.

Le fait que la déduction corresponde parfaitement à la sémantique s'appelle la *complétude*.

Le système exposé ci-dessous se situe dans le cadre de la logique booléenne, qui est la branche de la logique mathématique la plus utilisée en mathématiques.

Les constituants du langage

A la base de la syntaxe du calcul des propositions sont les **variables propositionnelles** ou **propositions atomiques**, notées *p*, *q*, etc., formant un ensemble infini dénombrable.

Les deuxièmes constituants de base du langage du calcul des propositions sont les **opérateurs** ou **connecteurs**. Ce sont des symboles qui permettent de construire des propositions plus élaborées. Les plus courants de ces connecteurs logiques sont : *et* $\wedge$, *ou* $\vee$, *non* $\neg$, *implique* $\rightarrow$, *équivaut* $\leftrightarrow$. On considère souvent aussi la constante $\perp$ qui représente le *faux*.

À côté de ces symboles on utilise des parenthèses pour lever les ambiguïtés dans les formules.

Un ensemble de connecteurs propositionnels est dit *complet*si tout connecteur peut se définir au moyen des connecteurs de l'ensemble. Ainsi l'ensemble $\{\neg, \rightarrow\}$ est complet : la disjonction se définit par: $A \vee B = (\neg A) \rightarrow B$.

L'ensemble constitué du seul connecteur NON-ET (noté « | » par Henry Sheffer) est également complet, $\neg P$ étant équivalent à $P|P$ et$P \vee Q$ à $(P|P) \mid (Q|Q)$. Cette particularité est utilisée pour la construction de circuits logiques, une seule porte suffisant alors pour concevoir tous les circuits existants.

Calcul des propositions

Le calcul des propositions repose de plus sur des règles de formation indiquant comment construire des propositions complexes syntaxiques bien construites ou « bien formées ».

Une **proposition** ou **formule bien formée** (notée par la suite *A*, *B*, *C* ou *P*, *Q*, *R*) est définie par induction sur la structure des expressions[4] comme suit :

- une variable propositionnelle *p* est une proposition,
- 0 (ou ⊥) et 1 sont des propositions.
- si *P* et *Q* sont des propositions, alors $(P \wedge Q)$, $(P \vee Q)$, $(P \rightarrow Q)$, $(P \leftrightarrow Q)$ et $\neg P$ sont des propositions.

Exemples : Si P, Q et R sont des propositions,

$(P \to Q) \to (\neg Q \to \neg P)$ est une proposition.$(P \to \bot) \to \bot$ est une proposition. $P \land \neg P$ est une proposition.$(P \land Q) \lor R$ est une proposition.

$P \land Q$ n'est pas une proposition.

Règle de résolution

La **règle de résolution** ou **principe de résolution** de Robinson est une règle d'inférence logique que l'on peut voir comme une généralisation du modus ponens. Cette règle est principalement utilisée dans les systèmes de preuve automatiques, elle est à la base du langage de programmation logique Prolog.

Systèmes formels

Pour construire une langue (par exemple le français), on a besoin de 4 choses :

•un alphabet (a, b, ..., z, blanc, virgule, parenthèse ouvrante, ...)

•un procédé de formation des mots, qui est la concaténation

•un dictionnaire, qui permet de savoir que "chien" est français, alors que "cien" ne l'est pas

•des règles de grammaire, qui permettent de savoir que "chattes" est français, alors qu'il n'est pas dans le dictionnaire.

Définition

Pour construire un système formel, nous aurons besoin de 4 choses analogues :

- un alphabet, ensemble de symboles pas nécessairement réduit à des caractères,
- un procédé de formation des expressions, pas nécessairement la concaténation,
- un ensemble d'*axiomes*, c'est-à-dire d'expressions obéissant aux deux premiers points ci-dessus, et dont on décide arbitrairement qu'ils appartiennent au système
- des règles de dérivation qui, à partir des axiomes, permettent de produire des *théorèmes* (c'est-à-dire des expressions appartenant au système), et peuvent ensuite s'appliquer aux théorèmes pour en produire d'autres

Exemple

Considérons le système "peu" :

alphabet = l'ensemble des trois symboles "p" , "e" , et "u"

p.f.e. = concaténation

axiome = upueuu

règles :

R1 : si une expression de la forme AeB est un théorème (où "A" désigne n'importe quelle suite de "u", de "p", et de "e", et B de même), alors l'expression uAeBu est aussi un théorème.

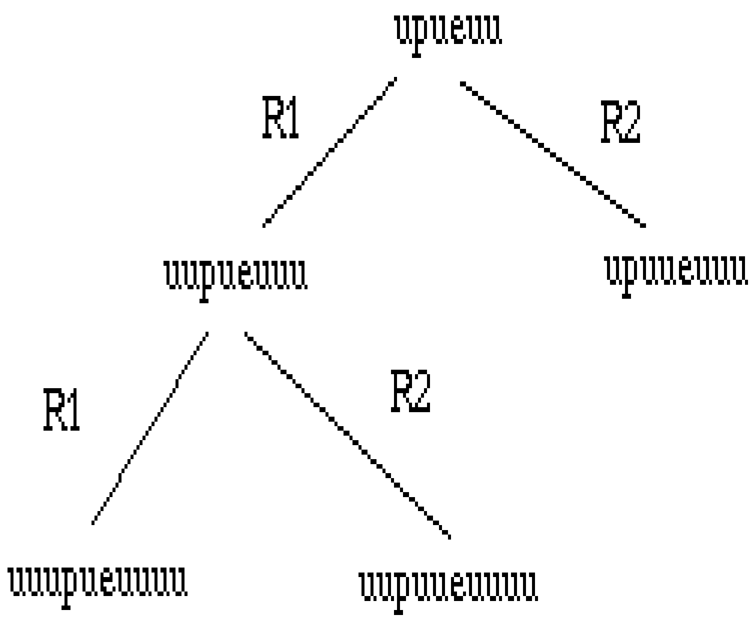
R2 : si une expression de la forme AeB est un théorème, alors l'expression AueuB est aussi un théorème

Questions :

- Q1 = uupuueuuuu est-il un théorème?
- Q2 = upuueuuuu ?
- Q3 = upupueuuu ?

Réponses :

Q1, oui. Preuve? On développe l'arbre :



Q2, non : il y a un nombre impair de "u", ce qui n'est pas possible

Q3, non : il y a deux "p "