

# Systemes à base de connaissances

Une introduction  
A Mille  
2007-2011  
[Avec emprunts à N. Guin et K. Boukerche]

---

---

---

---

---

---

---

---

## Connaissance ? [1]

- \* Cognition : faculté de « connaître » -> activités mentales (perception, raisonnement, mémoire, représentation, apprentissage, langage, conscience, émotions, ...)
- \* Inférence : production d'une connaissance
  - \* Avec représentation de la connaissance (déclarative)
  - \* Sans représentation de la connaissance (incorporée)
- \* Raisonnement : enchaînement d'inférences avec un objectif

SBC A Mille 2007-2011

2

---

---

---

---

---

---

---

---

## Connaissance ? [2]

- \* Démontrer de la connaissance => démontrer une capacité à mobiliser des informations pour agir ou produire d'autres capacités à agir (connaissances)
- \* Connaissance = information + mode d'emploi dans un contexte donné
- \* En IA: Connaissance = Information + Sémantique
- \* Pas de classement universel des différents types de connaissances (voir la tentative de Porphyre)

SBC A Mille 2007-2011

3

---

---

---

---

---

---

---

---

## Objectifs, attentes d'un SBC

- \* Inscrire les connaissances en tant que connaissance (pas seulement en tant qu'information) dans un système :
  - \* Pour « conserver » des savoirs, des savoir-faire et leur sémantique associée
- \* Disposer d'un « moteur » permettant d'enchaîner des inférences sur ces inscriptions de connaissances :
  - \* Pour « exploiter » les savoirs et savoir-faire ainsi « conservés »

SBC A Mile 2007-2011

4

---

---

---

---

---

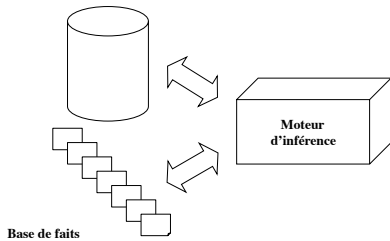
---

---

---

## Architecture d'un SBC

Base de (inscriptions de) connaissances



SBC A Mile 2007-2011

5

---

---

---

---

---

---

---

---

## Un SBC ...

- \* Inscrit des connaissances issues de l'expertise ou/et de la pratique (on dit que les connaissances sont « représentées » dans un système informatique).
  - \* Est donc spécialisé sur une expertise ou une pratique donnée
- \* Fonde le « raisonnement » sur des mécanismes d'inférence logique ou analogique
- \* Intègre une représentation symbolique
- \* Autorise parfois une certaine prise en compte de l'incertitude
- \* Les heuristiques sont des connaissances spécifiques au domaine qui guident la recherche de solutions
- \* Est orienté décision, résolution de problème et doit fournir des explications

SBC A Mile 2007-2011

6

---

---

---

---

---

---

---

---

## SBC à partir de règles

- \* Il s'agit des SBC historiques que l'on appelle « systèmes experts » s'ils sont spécialisés
- \* Les connaissances expertes sont représentées par des règles de la forme
  - \* Si (prémisses) Alors (conclusions)
  - \* Prémisses = conditions de déclenchement de la règle
  - \* Conclusions = effets du tirage de la règle
- \* Les connaissances sont déclaratives (révisables en principe)
- \* L'ensemble des règles forme « la base de connaissances ».
- \* Les faits décrivent ce qui est vrai dans la situation d'exploitation de la base de règles (base de faits).

SBC A Mile 2007-2011

7

---

---

---

---

---

---

---

---

## Exemple d'un langage (ordre 0) de règles

- \* R1 : Si (distance.<.2km) Alors (aller.à.pied)
- \* R2 : Si ((non distance.<.2km) ^ distance.<.300km) Alors (prendre.le.train)
- \* R3 : Si (non distance.<.300km) Alors (prendre.l'avion)
- \* R4 : Si (acheter.un.billet ^ avoir.le.téléphone) Alors (téléphoner.à.l'agence)
- \* R5 : Si (acheter.un.billet ^ (non avoir.le.téléphone)) Alors (aller.à.l'agence)
- \* R6 : Si (prendre.l'avion) Alors (acheter.un.billet)
- \* R7 : Si (durée.>.2.jours ^ être.fonctionnaire) Alors (non prendre.l'avion)

SBC A Mile 2007-2011

8

---

---

---

---

---

---

---

---

## Base de faits

- \* F1 : (non distance.<.300km)
- \* F2 : (avoir.le.téléphone)

SBC A Mile 2007-2011

9

---

---

---

---

---

---

---

---

## Moteur d'inférence (1)

```
ça marche
tant que ça marche
  ça ne marche pas
  boucle sur les  $R_i$ 
    boucle sur les  $F_j$ 
      si  $R_i$  est de la forme « Si  $F_j$  Alors  $F_k$  »
        ajouter  $F_k$  à la BdF
        ça marche
      finsi
    finboucle
  finboucle
fintant
```

*Ce moteur ne sait pas gérer les prémisses de type  
conjonctions de faits... Comme  
(acheter.un.billet ^ avoir.le.téléphone)*

SBC A Mile 2007-2011

10

---

---

---

---

---

---

---

---

---

---

## Moteur d'inférence (2)

```
ça marche
tant que ça marche
  ça ne marche pas
  boucle sur les  $R_i$ 
    boucle sur les  $F_j$ 
      si  $R_i$  est de la forme « Si  $F_j$  Alors  $F_k$  »
        ajouter  $F_k$  à la BdF
        ça marche
      sinon
        boucle sur les  $F_1$ 
          si  $R_i$  est de la forme « Si  $F_j$  ^  $F_1$  Alors ... »
            ajouter  $F_n = (F_j \wedge F_1)$  à la BdF
            ça marche
          finsi
        finboucle
      finsi
    finboucle
  finboucle
fintant
```

*Ce programme boucle car on ne  
marque pas les règles déjà « tirées »*

SBC A Mile 2007-2011

11

---

---

---

---

---

---

---

---

---

---

## Moteur d'inférence (3)

```
ça marche
tant que ça marche
  ça ne marche pas
  boucle sur les  $R_i$ 
    boucle sur les  $F_j$  non marqués
      si  $R_i$  est de la forme « Si  $F_j$  Alors  $F_k$  »
        ajouter  $F_k$  à la BdF
        marquer  $F_j$ 
        ça marche
      sinon
        boucle sur les  $F_1$ 
          si  $R_i$  est de la forme « Si  $F_j$  ^  $F_1$  Alors ... »
            ajouter  $F_n = (F_j \wedge F_1)$  à la BdF
            marquer  $F_j$ 
            ça marche
          finsi
        finboucle
      finsi
    finboucle
  finboucle
fintant
```

*Ce programme est correct, donne la réponse en  
un temps fini et quel que soit l'ordre des  $R_i$  et  $F_j$*

SBC A Mile 2007-2011

12

---

---

---

---

---

---

---

---

---

---

## Gérer les contradictions ?

Considérons une nouvelle base de faits :

- F1 :  $\neg$ distance < 300km
- F2 : avoir\_le\_téléphone
- F3 : durée > 2jours
- F4 : être\_fonctionnaire

- R3 et F1  $\rightarrow$  F5 : prendre\_l'avion
- R7 et F3 et F4  $\rightarrow$  F6 :  $\neg$ prendre\_l'avion

**Solution : tester la présence d'un fait contradictoire dans la BdF avant d'y ajouter un nouveau fait -> signaler le problème à l'utilisateur.**

SBC A Mile 2007-2011

13

---

---

---

---

---

---

---

---

## Fournir des « explications »

- \* Le SBC peut « expliquer » chaque fait produit par la trace de son exécution.
- \* Les règles et les faits étant exprimés à un haut niveau d'abstraction (symbolique), ces explications sont réputées « lisibles » par les opérateurs humains. -> exemple Mycin
- \* Certains systèmes donnent la possibilité d'accéder aux « documents » justifiant l'inscription de telle ou telle connaissance.

SBC A Mile 2007-2011

14

---

---

---

---

---

---

---

---

## Schéma général de fonctionnement SBC / Règles

- \* *Constituer l'ensemble des règles déclenchantes* = Sélection des faits et des règles pertinentes / pb + Filtrage par comparaison des prémisses de chaque règle avec les faits de la BdF  $\Leftrightarrow$  Conflict Set
- \* *Choisir les règles à déclencher* selon une stratégie
- \* *Déclencher les règles* = mise à jour de la BdF avec détection des contradictions, effectuer une action == passer le contrôle à une autre entité.
- \* Recommencer....

SBC A Mile 2007-2011

15

---

---

---

---

---

---

---

---

Mode d'exploitation par chaînage avant, sans but, irrévocable et monotone

- \* Pas de but ⇔ déclenchement des règles jusqu'à épuisement des faits possibles à produire (ou arrêt).
- \* Par intégration immédiate des conclusions des règles
- \* En largeur d'abord
- \* Irrévocable ⇔ déclenchement d'une règle non remis en cause
- \* Monotone ⇔ les faits produits ne sont pas remis en cause.
- \* (avec un but ⇔ une « distance » au but est calculée pour choisir la règle à appliquer)

---

---

---

---

---

---

---

---

Chaînage arrière, par tentatives + monotone

- \* Un but est assigné au système
- \* La partie « conclusions » des règles est unifiée avec ce but. En cas de succès, les prémisses de la règle unifiée sont les nouveaux buts assignés.
  - \* Il s'agit donc d'un arbre ET/OU de buts dont les feuilles sont vérifiées ou non par les faits
- \* Par tentatives ⇔ les buts sont substitués lorsque le chaînage arrière est infructueux.
- \* Stratégies :
  - \* Production de buts en profondeur
  - \* Profondeur d'abord sauf si une règle conclue immédiatement
- \* Cas de feuilles non vérifiables
  - \* Poser la question à l'utilisateur
  - \* Formuler une hypothèse jusqu'à vérification ou contradiction.

---

---

---

---

---

---

---

---

Chaînage mixte  
Tentatives + monotone

- \* Tant que des règles sont déclenchables ⇔ chaînage avant
- \* Puis, on choisit une règle « presque » déclenchable et on essaie de vérifier les prémisses inconnues par chaînage arrière
- \* En cas de succès, on repart en chaînage avant.

---

---

---

---

---

---

---

---

## Chaînage mixte Tentatives, non monotonie

- \* La partie déclencheur de règles = un but B et des prémisses  
Ex: Pour prouver B quand F est vérifié, il suffit d'exécuter l'action A et de prouver B'  
Action = ajouter ou retirer un fait ⇔ non monotonie
- \* Déclenchement en profondeur avec empilement et dépilement des buts
- \* En cas d'échec, retour arrière avec restauration du contexte initial

SBC A Mille 2007-2011

19

---

---

---

---

---

---

---

---

## Exemple de moteur avec des variables : PROLOG (standard)

- \* R1 : papy(X,Y) :- pere(X,Z), pere(Z,Y).
- \* F1 : pere(pierre,jean).
- \* F2 : pere(jean,rené).
- \* But : papy(U,V).

- \* Chaînage arrière
- \* Tentatives
- \* Monotonie

SBC A Mille 2007-2011

20

---

---

---

---

---

---

---

---

## Meta-connaissances (Jacques Pitrat)

- \* Connaissances sur les connaissances
  - \* Mécanisme de « contrôle » du raisonnement en cours et de sa validité
  - \* Possibilité d'organiser progressivement les connaissances par un mécanisme de supervision
  - \* Mécanisme « empilable » ⇔ Meta-connaissances sur des meta-connaissances
  - \* Exploitation pour « contextualiser » l'usage d'un SBC – à l'individu, à la situation, au lieu, à l'instant, à la nécessité de précision ou non, etc...

SBC A Mille 2007-2011

21

---

---

---

---

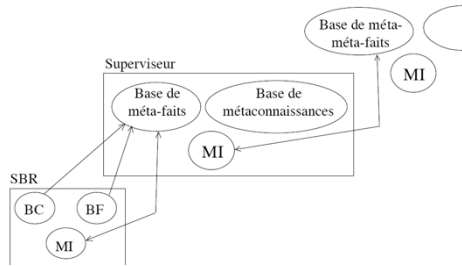
---

---

---

---

## Supervision / Meta-connaissances



SBC A Mile 2007-2011

22

---

---

---

---

---

---

---

---

## Grandes familles d'inférence pour le raisonnement

- \* **Déduction**
  - \* Si  $(A \rightarrow B \text{ est vrai})$  et  $(A \text{ est vrai})$  Alors  $B$  est vrai.
- \* **Induction (généralisation)**
  - \* Si  $(P \text{ est vraie pour } a, b, c \text{ de } \{a, b, c, \dots, x\})$  Alors  $(P \text{ est vraie pour tout élément de l'ensemble})$
- \* **Abduction (hypothèse)**
  - \* Si  $(B \text{ est vrai})$  et que  $(A \rightarrow B \text{ est vrai})$  Alors  $A$  est vrai.
- \* **Analogie (hypothèse)**
  - \* Les  $A'$  sont à  $B'$  ce que les  $A$  sont à  $B$  ( $A'$  est similaire à  $A \Leftrightarrow B'$  est similaire à  $B$ )
- \* **« Sens commun »**
  - \* Introduction des heuristiques (voir le problème de [Send + More](#))

SBC A Mile 2007-2011

23

---

---

---

---

---

---

---

---

## En savoir plus

- \* <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-871-knowledge-based-applications-systems-spring-2005/lecture-notes/>

SBC A Mile 2007-2011

24

---

---

---

---

---

---

---

---