Bases de l'Intelligence Artificielle

CM7 : Système à Base de Connaissances

Marie Lefevre

2025-2026

Université Claude Bernard Lyon 1

De quoi va-t-on parler ?

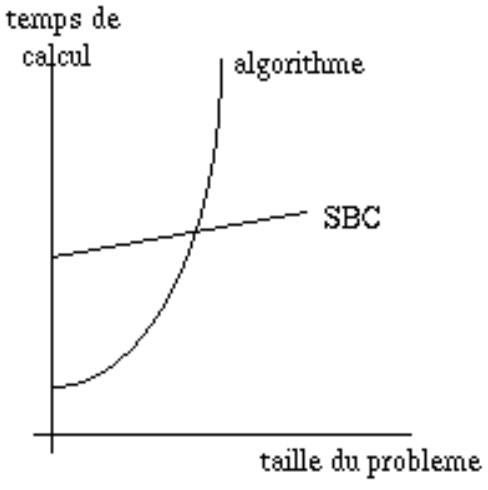
- SBC pour quoi faire ?
- L'approche SBC
- SBC à partir de règles
- Les mécanismes d'inférence
- Les types d'inférence
- Les méta-connaissances
- Pour aller plus loin

Domaines d'application

- Quand utiliser un SBC ?

➢ Quand les connaissances sont difficilement « codables »
  - Les connaissances sont éparses
  - Les connaissances sont d'origine expérimentale ou heuristique
  - Le savoir-faire des experts humains n'est pas suffisamment structuré pour que l'on dispose d'algorithmes
    - Exemple : Diagnostic médical, droit, éducation, …

➢ Quand les algorithmes dont on dispose sont de complexité exponentielle



Types de problèmes

- La prédiction des conséquences à partir de situations données
- Le diagnostic d'une défaillance à partir d'un ensemble d'observations
- La conception d'une configuration de composants à partir d'un ensemble de contraintes
- La planification d'une séquence d'actions pour accomplir un ensemble de buts à partir de certaines conditions de départ et en présence de certaines contraintes
- La réparation d'un dysfonctionnement
- L'interprétation ou la construction d'une description abstraite à partir de données
- Le contrôle du comportement d'un environnement complexe
- …

Problèmes adaptés ?

- Ensemble informel de critères pour déterminer si un problème est adapté à être résolu par un SBC :
  1. Le besoin d'une solution doit justifier le coût et l'effort de la construction d'un SBC
  2. L'expertise humaine n'est pas valable dans toutes les situations dont on a besoin
  3. Le problème peut être résolu en utilisant une technique de raisonnement symbolique
  4. Le domaine est bien structuré
  5. Le problème ne peut pas être résolu en utilisant des méthodes traditionnelles de calcul
  6. La coopération entre experts des différents domaines existe
  7. Le problème est de taille considérable

De quoi va-t-on parler ?

- SBC pour quoi faire ?
- L'approche SBC
- SBC à partir de règles
- Les mécanismes d'inférence
- Les types d'inférence
- Les méta-connaissances
- Pour aller plus loin

De quoi parle-t-on au juste ?

- 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
- 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 symbolique + Sémantique

➢ Pas de classement universel des différents types de connaissances (voir la tentative de Porphyre : http://fr.wikipedia.org/wiki/Arbre_de_Porphyre)

Où mettre la connaissance ?

- Représentations déclaratives
  - Logique du fonctionnement : « comment c'est fait »
  - ➢ Connaissances indépendantes de leur exploitation future
  - Architecture dans laquelle une « base de faits » est séparée d'une structure de procédures
    - Procédures assez générales pour pouvoir manipuler toutes sortes de faits
    - Base de faits particulière à un domaine de connaissance
  - Avantage : modularité
  - Inconvénient : architecture relativement lente car procédures interprétées

SBC

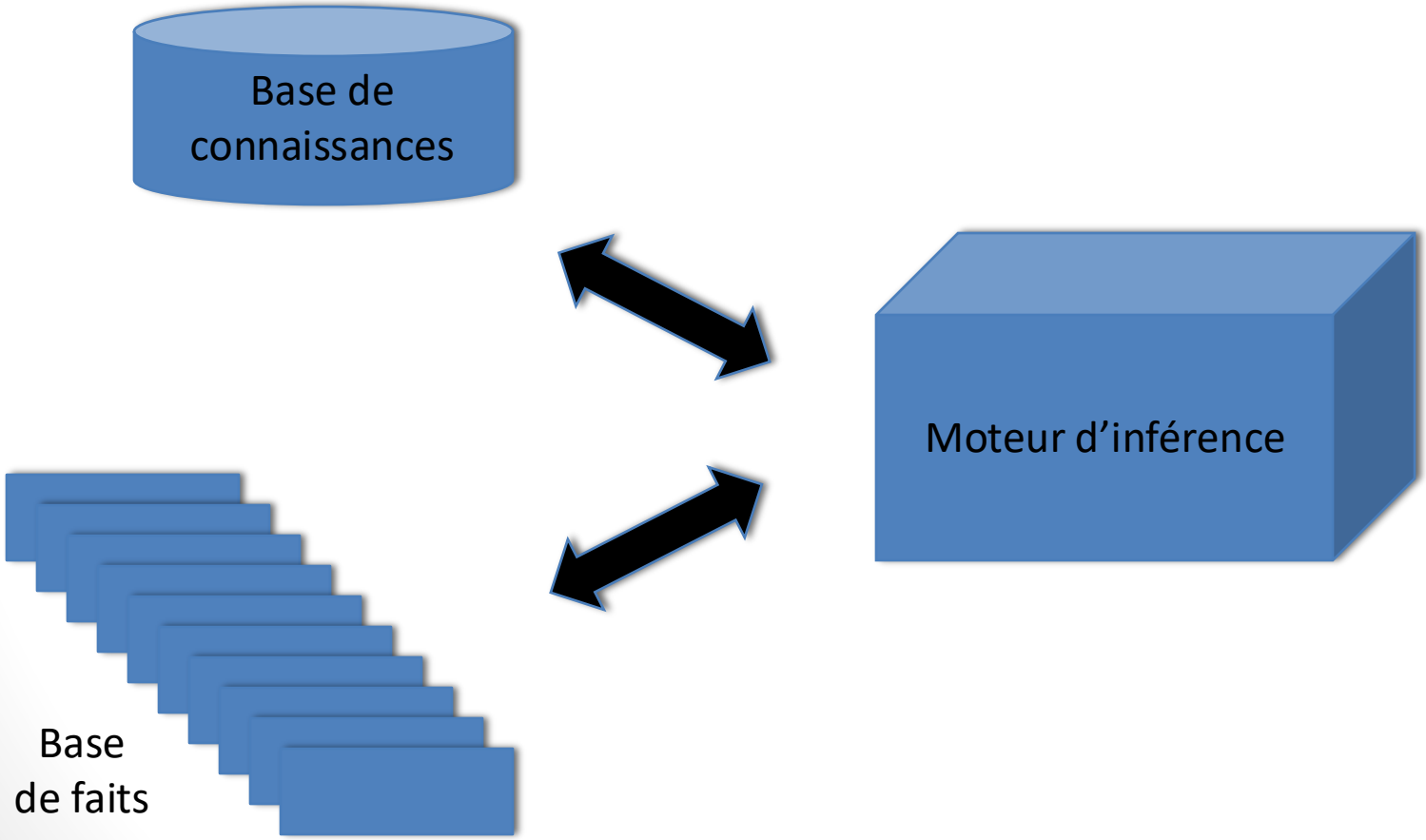
- Représentations procédurales
  - Logique de l'action : « comment s'en servir »
  - ➢ Connaissances contiennent leur mode d'emploi
  - Codification de toute la connaissance sous forme de procédures compilées
    - Données factuelles implicites
  - Avantage : rapidité d'exécution
  - Inconvénient : données difficilement accessibles car contenues dans les procédures compilées

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 »

Architecture d'un SBC



Architecture d'un SBC

- Séparation entre les connaissances et l'inférence qui permet
  - De séparer codage des connaissances et codage du moteur d'inférence
  - D'utiliser un codage différent
    - Par exemple : le langage naturel pour représenter les connaissances (sous forme Si .. ALORS..)
  - De modifier les connaissances sans avoir un effet sur le codage du moteur d'inférence
  - De pouvoir tester plusieurs types d'inférence sur la même base de connaissances

Donc 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
  - Il 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

De quoi va-t-on parler ?

- SBC pour quoi faire ?
- L'approche SBC
- SBC à partir de règles
- Les mécanismes d'inférence
- Les types d'inférence
- Les méta-connaissances
- Pour aller plus loin

SBC à partir de règles

- Chaque unité de connaissance est représentée par une règle de la forme : Si *Prémisses* Alors *Conclusions*
  - Prémisses = conditions de déclenchement de la règle
  - Conclusions = effets du tirage de la règle (ajouts ou retraits de faits, lancements d'actions)
- Une règle n'est pas désignée par un nom, mais par ses prémisses et ses conclusions : mode d'accès associatif
- Les connaissances sont déclaratives et (en principe) révisables
- L'ensemble des règles forme la base de connaissances, ou base de règles
- Les faits décrivent ce qui est vrai dans la situation d'exploitation de la base de règles (base de faits)

SBC / SBR / Système expert

- Les systèmes experts sont les SBC historiques
  - Les premiers SBC à partir de règles
  - « Expert » car spécialisé pour un domaine donné

- Le 1er : DENDRAL (1969) en chimie

- La suite : MYCIN (1972)
  - Dédié au diagnostic de maladies infectieuses du sang et à la prescription médicale
  - C'est le premier à bien séparer le moteur d'inférence de la base de connaissances et à pouvoir expliquer ses raisonnements
  - Le système initial comporte 200 règles mais atteint rapidement les 500 règles
  - ➢ Réussissait à diagnostiquer à un niveau proche des experts humains et considérablement meilleur que celui les jeunes médecins

➢ Les systèmes experts ont montré leur efficacité pour des tâches précises pour lesquelles aucune procédure algorithmique n'est connue, dans un domaine de connaissances réduit et technique

Exemple d'une base de règles : un système de conseil en voyages

- Règles de la logique des propositions (ordre zéro)

R1 : Si (distance.<.2km) Alors (aller.à.pied)

R2 : Si (¬(distance.<.2km) ^ distance.<.300km) Alors (prendre.le.train)

R3 : Si (¬(distance.<.300km)) Alors (prendre.l'avion)

R4 : Si (acheter.un.billet ^ avoir.le.téléphone) Alors (téléphoner.à.l'agence)

R5 : Si (acheter.un.billet ^ ¬ (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 (¬ (prendre.l'avion))

Exemple d'une base de faits : un système de conseil en voyages

- La base de faits contient, pour une situation donnée, les faits avérés ou à établir

F1 : ¬ (distance.<.300km)

F2 : (avoir.le.téléphone)

Ex.1 de moteur d'inférence : un système de conseil en voyages

- Le moteur d'inférence est un programme qui exploite la base de règles pour déduire de nouveaux faits
  - Il détient le contrôle du raisonnement

Boucle sur les règles : soit R une règle

Si les prémisses de R appartiennent à BF

Alors ajouter les conclusions de R à BF

FinSi

FinBoucle

R3 et F1 => F3 : prendre.l'avion

R6 et F3 => F4 : acheter.un.billet

- Problème : ne déclenche pas R4

Ex.2 de moteur d'inférence : un système de conseil en voyages

Changement ← vrai
Tant que changement
  Changement ← faux
  Boucle sur les règles : soit R une règle
    Si les prémisses de R appartiennent à BF
    Alors ajouter les conclusions de R à BF
          changement ← vrai
    FinSi
  FinBoucle
FinTQ

R3 et F1 => F3 : prendre.l'avion
R6 et F3 => F4 : acheter.un.billet
R3 et F1 => F3
R4 et F4 et F2 => F5 : téléphoner.à.l'agence
R6 et F3 => F4
R3 et F1 => F3, R4 et F4 et F2 => F5, R6 et F3 => F4 **...**

- Pb : ce programme boucle car les règles s'appliquent plusieurs fois

R1 : Si (distance.<.2km) Alors …

R2 : Si (¬(~~distance.<.2km~~) ^ distance.<.300km) Alors …

R3 : Si (¬(distance.<.300km)) Alors …

R4 : Si (acheter.un.billet ^ avoir.le.téléphone) Alors …

R5 : Si (acheter.un.billet ^ ¬ (avoir.le.téléphone)) Alors …

R6 : Si (prendre.l'avion) Alors …

R7 : Si (durée.>.2.jours ^ être.fonctionnaire) Alors …

F1 : ¬ (distance.<.300km)

F2 : (avoir.le.téléphone)

Ex.3 de moteur d'inférence : un système de conseil en voyages

Changement ← vrai
Tant que changement
  Changement ← faux
  Boucle sur les règles : soit R une règle
    Si R n'est pas marquée et si les prémisses de R appartiennent à BF
    Alors ajouter les conclusions de R à BF
          changement ← vrai
          marquer R
    FinSi
  FinBoucle
FinTQ

R3 et F1 => F3 : prendre.l'avion
R6 et F3 => F4 : acheter.un.billet
R4 et F4 et F2 => F5 : téléphoner.à.l'agence

➤ ☺

R1 : Si (distance.<.2km) Alors …

R2 : Si (¬(distance.<.2km) ^ distance.<.300km) Alors …

R3 : Si (¬(distance.<.300km)) Alors …

R4 : Si (acheter.un.billet ^ avoir.le.téléphone) Alors …

R5 : Si (acheter.un.billet ^ ¬ (avoir.le.téléphone)) Alors …

R6 : Si (prendre.l'avion) Alors …

R7 : Si (durée.>.2.jours ^ être.fonctionnaire) Alors …

F1 : ¬ (distance.<.300km)

F2 : (avoir.le.téléphone)

Et les contradictions ?

Avec une nouvelle base de faits :

R3 et F1 => F5 : prendre.l'avion
R6 et F5 => F6 : acheter.un.billet
R7 et F3 et F4 => F7 : ¬ prendre.l'avion

➢ ☹

Moteur d'inférence

Changement ← vrai
Tant que changement
    Changement ← faux
    Boucle sur les règles : soit R une règle
        Si R n'est pas marquée
            et si les prémisses de R appartiennent à BF
        Alors pour chaque conclusion C de R :
            Si ¬C appartient à BF
            Alors message d'erreur
            Sinon Ajouter C à BF
            FinSi
            Changement ← vrai
            Marquer R
        FinSi
    FinBoucle
FinTantQue

1. Détecter la modification de la base de faits
2. Savoir si une règle a déjà été utilisée
3. Gérer les contradictions

Possibilité d'explications

- Un SBR doit être capable de :
  - Dialoguer avec l'expert et l'utilisateur
  - Fournir des explications sur son raisonnement
  - Apprendre de nouvelles connaissances grâce à ce dialogue

- Pour dialoguer
  - 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 de Mycin : http://lazax.com/software/Mycin/mycin.html

Générateur de SBR (noyau)

- Moteur d'inférence
- Langage d'expression externe des connaissances
- Ensemble de structures et conventions de représentation internes de ces connaissances

➢ On instancie ce noyau à un domaine
en garnissant la base de règles

De quoi va-t-on parler ?

- SBC pour quoi faire ?
- L'approche SBC
- SBC à partir de règles
- Les mécanismes d'inférence
- Les types d'inférence
- Les méta-connaissances
- Pour aller plus loin

Schéma général de fonctionnement d'un moteur d'inférence

Phase d'évaluation :

- Constituer l'ensemble des règles déclenchables (Conflict Set)
  - Sélection des faits de BF et des règles de BR qui méritent d'être comparés, i.e. qui sont pertinents / problème
  - Filtrage par comparaison des prémisses de chaque règle R de BR avec les faits de BF
- Résoudre les conflits
  - Choisir les règles à déclencher selon une stratégie

Phase d'exécution :

- Déclencher les règles
  - Mise à jour de BF, avec détection d'incohérence
  - Effectuer une action : passer le contrôle à une autre entité

Mécanismes de raisonnement

- Un moteur d'inférence peut raisonner de différentes façons
  - Chaînage avant
    - Le raisonnement est guidé par les données
  - Chaînage arrière
    - Le raisonnement est guidé par le but
  - Chaînage mixte

- Deux régimes
  - Irrévocable : déclenchement d'une règle non remis en cause
  - Par tentatives : des suites infructueuses de « déclenchements » de règles peuvent être annulées et remplacées par d'autres

- Monotonie
  - Monotone : les faits produits ne sont pas remis en cause
  - Non-monotone : une action peut consister à ajouter ou retirer un fait

Prenons un exemple…

- Ce que nous savons :
  - SI X aime la musique classique et les maths ALORS il pourrait aimer Bach
  - SI X aime la musique classique et est de tempérament romantique ALORS il pourrait aimer Schubert
  - SI X est de tempérament romantique ALORS il pourrait aimer Cabrel
  - SI X écrit des poèmes ALORS il est de tempérament romantique

- Nous pouvons modéliser ces quatre règles avec la logique de prédicats :
- Base de règles :
  - R1 : Si aime(X, musiqueC) ^ aime(X, maths) Alors peutAimer(X, bach)
  - R2 : Si aime(X, musiqueC) ^ romantique(X) Alors peutAimer(X, schubert)
  - R3 : Si romantique(X) Alors peutAimer(X, cabrel)
  - R4 : Si ecritPoeme(X) Alors romantique(X)

- Base de faits :
  - F1 : aime(toto, musiqueC)
  - F2 : ecritPoeme(toto)

Chainage avant

- Régime irrévocable & monotone
- Sans but : le système déclenche des règles jusqu'à épuisement ou arrêt
  - Plusieurs stratégies :
    - Intégration immédiate des conclusions des règles
    - Production et intégration des faits en « largeur d'abord », après épuisement du Conflict set
    - Privilégier les règles qui contiennent en prémisse un fait qui vient d'être établi (profondeur d'abord)
- Avec but :
  - Calcul de la distance à ce but pour choisir la règle à appliquer
    - En profondeur d'abord
    - En largeur d'abord

Sur notre exemple… sans but

- Base de règles :
  - R1 : Si aime(X, musiqueC) ^ aime(X, maths) Alors peutAimer(X, bach)
  - R2 : Si aime(X, musiqueC) ^ romantique(X) Alors peutAimer(X, schubert)
  - R3 : Si romantique(X) Alors peutAimer(X, cabrel)
  - R4 : Si ecritPoeme(X) Alors romantique(X)

- Base de faits :
  - F1 : aime(toto, musiqueC)
  - F2 : ecritPoeme(toto)

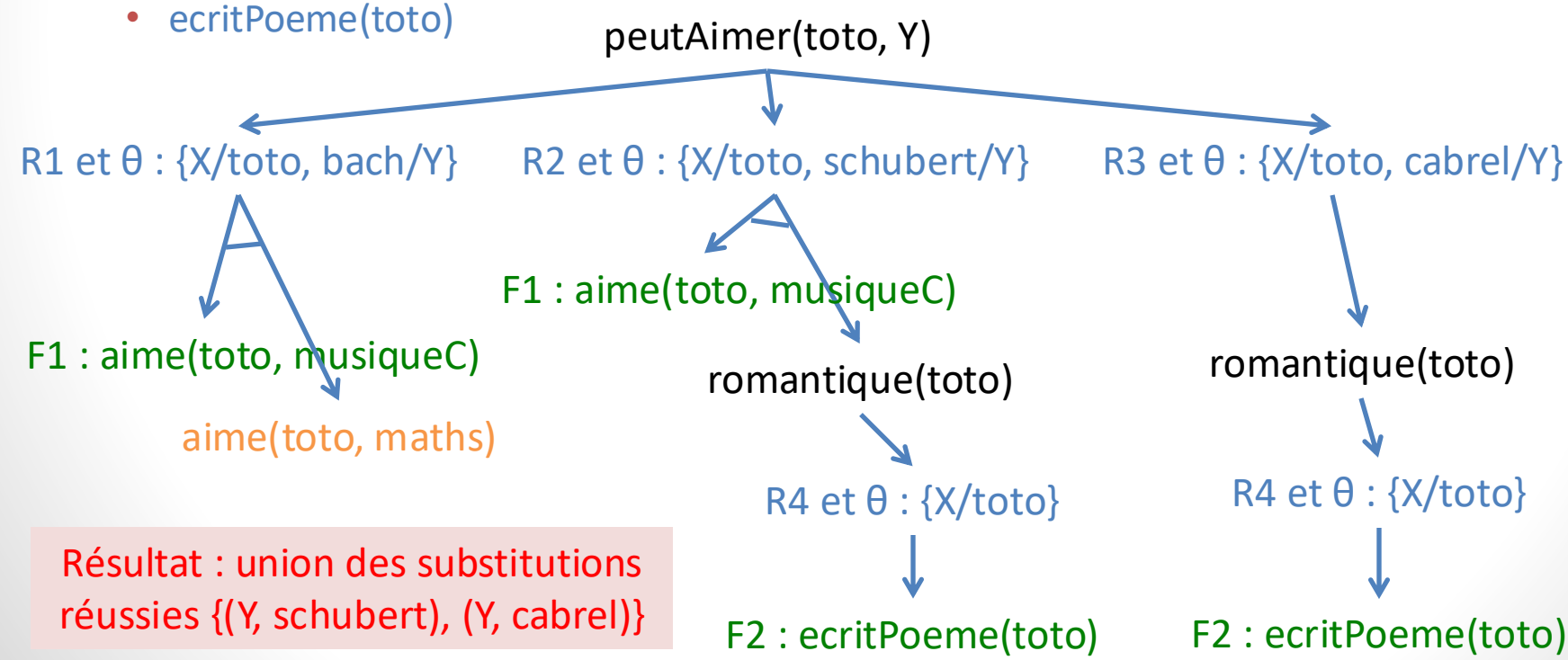
- Raisonnement dans l'ordre des R_i avec intégration immédiate des faits
  - R4 et F2 => F3 : romantique(toto)
  - R2 et F1/F3 => F4 : peutAimer(toto, schubert)
  - R3 et F3 => F5 : peutAimer(toto, cabrel)
  - Arret

Chainage arrière

- Un but est assigné système
  - Unification de la partie conclusion des règles avec ce but
  - Nouveaux buts = les prémisses de la règle sélectionnée
  - Donc développement d'un arbre ET/OU de buts,
    les feuilles pouvant être validées ou invalidées par la base de faits
- Régime par tentatives & monotone
  - Tentatives ⇔ substitution de buts
- Deux stratégies :
  - Production de buts en profondeur d'abord
  - Profondeur d'abord sauf si une règle est immédiatement concluante
- Cas de feuilles non vérifiables :
  - Poser la question à l'utilisateur
  - Faire une hypothèse jusqu'à ce qu'un élément nouveau vienne la valider ou l'invalider

Sur notre exemple…

- Base de règles :
  - R1 : Si aime(X, musiqueC) ^ aime(X, maths) Alors peutAimer(X, bach)
  - R2 : Si aime(X, musiqueC) ^ romantique(X) Alors peutAimer(X, schubert)
  - R3 : Si romantique(X) Alors peutAimer(X, cabrel)
  - R4 : Si ecritPoeme(X) Alors romantique(X)
- Base de faits :
  - aime(toto, musiqueC)
  - ecritPoeme(toto)



Résultat : union des substitutions réussies {(Y, schubert), (Y, cabrel)}

Chainage mixte

- Régime par tentatives & monotone
  - Panacher chaînage avant et arrière :
    - Tant que des règles sont déclenchables : chainage avant
    - Puis on choisit une règle « presque déclenchable » et on essaie d'en évaluer les prémisses inconnues en chaînage arrière
    - En cas de succès, on repart en chaînage avant

- Régime par tentatives & non-monotone
  - La partie déclencheur des règles = un but + des prémisses :
    - « Pour prouver B sachant que F est établi,
      il suffit d'exécuter l'action A et de prouver B2 »
  - Une action peut consister à ajouter ou retirer un fait
  - Déclenchement en profondeur par empilement / dépilement des buts
  - Le retour arrière en cas d'échec nécessite de restaurer le contexte

Exemple de moteur avec des variables : PROLOG (standard)

- Chaînage arrière
- Régime par Tentatives
- Monotonie

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

De quoi va-t-on parler ?

- SBC pour quoi faire ?
- L'approche SBC
- SBC à partir de règles
- Les mécanismes d'inférence
- Les types d'inférence
- Les méta-connaissances
- Pour aller plus loin

Les types d'inférence

- Raisonnement déductif ou déduction
  - But : dériver les conséquences d'un ensemble d'informations données
  - Raisonnement utilisé en logique propositionnelle
  - Principal avantage :
    - Si nous dérivons une conséquence d'un ensemble d'informations qui sont vraies,
      alors nous sommes sûrs que la conséquence est vraie aussi

  - Exemple : permet de déduire
    - « Marie est un docteur » à partir des informations
      « Marie est un docteur ou un professeur »
      et « Marie n'est pas un professeur »
    - « Titi a des ailes » à partir de
      « Titi est un oiseau » et « Tout oiseau a des ailes »

Les types d'inférence

- Raisonnement abductif ou abduction
  - But : expliquer des observations
  - Raisonnement utilisé couramment dans la vie de tous les jours
    - Notamment dans le diagnostic médical : on cherche une cause possible (maladie) qui pourrait expliquer les observations (des symptômes du patient) en présence de connaissances générales (les connaissances sur la médecine)

  - Exemple : permet de déduire « La patiente a la grippe »

    étant donné que « La patiente a de la fièvre, de la toux, et mal à la tête »
    et que « La grippe peut provoquer des fièvres, maux de gorge, maux de tête, fatigue, toux, et des douleurs musculaires »
    … mais on aurait aussi pu déduire « La patiente a une bronchite combinée avec un cancer du cerveau » !!!

Les types d'inférence

- Raisonnement inductif ou induction
  - But : produire des connaissances générales à partir d'observations
  - Raisonnement équivalent à un apprentissage
  - Principal limite :
    - Contrairement au raisonnement déductif, les conclusions du raisonnement inductif peuvent s'avérer fausses
  - Principal avantage :
    - Permet d'agir dans un environnement incertain ou lorsque les informations dont on dispose sur celui-ci sont incomplètes

  - Exemple : permet de déduire « Tous les cygnes sont blancs »

    étant donné que tous les cygnes que l'agent a vu dans sa vie étaient blancs
    …. il existe des cygnes noirs !

Les types d'inférence

- Raisonnement par analogie
  - But : mettre en correspondance une situation antérieure et une situation nouvelle qui lui ressemble, afin de déduire la nature ou des aspects de cette situation nouvelle
  - Type particulier de raisonnement inductif

  - Exemple : « Piaget est à la psychologie du développement ce que Freud est à … »

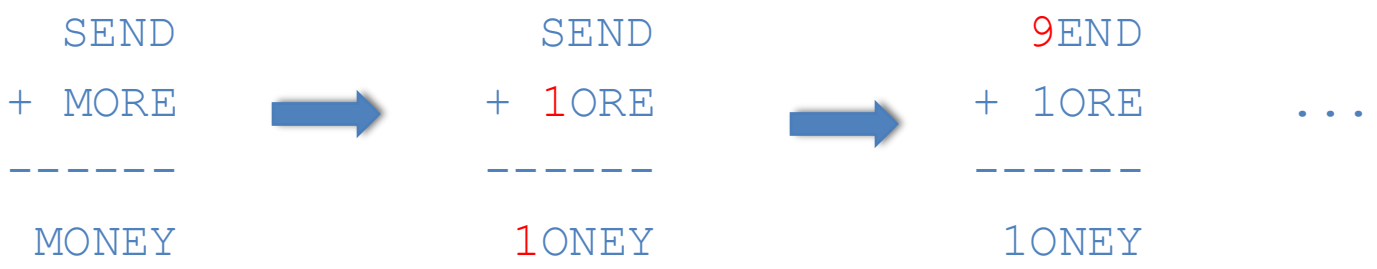
    étant donné que Piaget est un pionnier de la psychologie du développement
    on cherche de quoi Freud est le pionnier
    sachant qu'une ressemblance doit exister (en général, même catégorie)

    on peut prendre « psychanalyse »

    … mais on aurait pu trouver d'autres relations

Les types d'inférence

- Sens commun
  - Introduction des heuristiques
    - Voir le problème de SEND + MORE = MONEY
      http://www.greylabyrinth.com/solution/puzzle001



Synthèse des types d'inférence

- Déduction
  - Si ((A → B est vrai) et (A est vrai))
    Alors B est vrai
- Induction (généralisation)
  - Si (P est vraie pour a,b,c de {a,b,c,…,x}
    Alors (P est vraie pour tout élément de l'ensemble)
- Abduction (hypothèse)
  - Si ((B est vrai) et (A → B 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 ↔ B' est similaire à B)

De quoi va-t-on parler ?

- SBC pour quoi faire ?
- L'approche SBC
- SBC à partir de règles
- Les mécanismes d'inférence
- Les types d'inférence
- Les méta-connaissances
- Pour aller plus loin

Méta-connaissance

- Une méta-connaissance est
  une connaissance sur les connaissances

- Utilisation surtout à partir des années 80 dans les systèmes experts
  - Début dès 1966 : Pitrat
    pour la démonstration de théorèmes

➢ Mais pour quoi faire ?

Exemple humain (J-M. Fouet)

- Lorsque je concocte un sujet d'examen, mon but est d'obtenir une répartition Gaussienne des notes. Pour cela, je prends en compte :
  - ce que je pense qu'une majorité d'étudiants a compris
  - ce que je pense que seuls quelques étudiants maitrisent
- Lorsqu'ils reçoivent le sujet, les étudiants adoptent l'un de deux comportements :
  - les étudiants "amateurs" commencent à répondre à la 1ère question ;
  - les étudiants "professionnels" lisent l'énoncé en entier.
    Cela leur permet, connaissant le temps qui leur est imparti, et éventuellement le coefficient de chaque question, d'attribuer un quota de temps à chaque question, e s] : telle question est facile, r Telle question est dans mon i fléchir. Telle question m lui accorde une heure.

Où sont les méta-connaissances ?

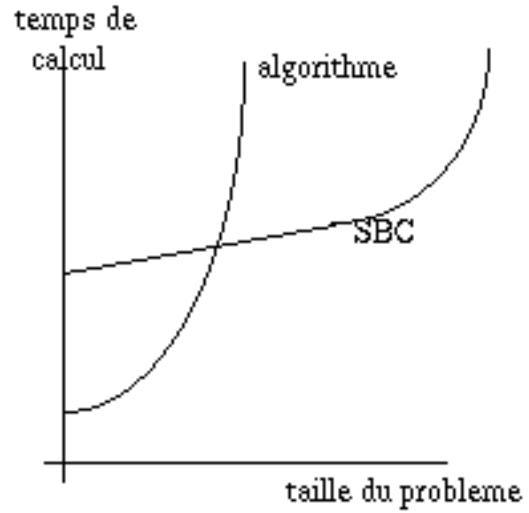
- Ainsi le professionnel, même s'il est moins bon que son voisin amateur, obtiendra un meilleur résultat.
  En particulier, l'amateur risquera de ne pas avoir le temps de traiter la dernière question, à laquelle il aurait pourtant su répondre

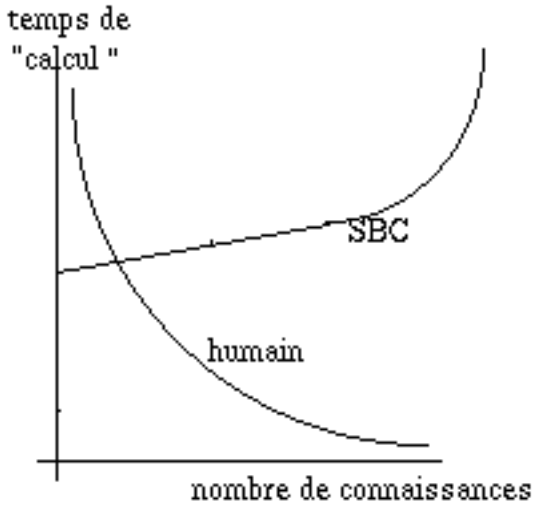
Exemple du MI (J-M. Fouet)

- Un moteur d'inférence contient un certain nombre de boucles imbriquées

- Rien d'étonnant
  - Lorsqu'on augmente le nombre de connaissances, les temps de calcul commencent à grimper exponentiellement
- Ce qui est étonnant par contre
  - L'expert est d'autant plus efficace qu'il sait de choses

- Seule explication possible
  - L'expert n'utilise pas un algorithme pour utiliser ses connaissances, il utilise des méta-connaissances





Opérer à plusieurs niveaux

- Pour être « intelligent »
  un système informatique doit pouvoir
  - Réfléchir sur l'énoncé des problèmes qu'il traite
  - Disposer d'un mécanisme analogue à la conscience humaine pour savoir ce qu'il a fait,
    et pourquoi il l'a fait
  - Comprendre les raisons de ses succès et de ses échecs
  - Observer ses progrès vers son but
  - Modifier ses plans en fonction de ses observations

Métaconnaissances (J. Pitrat)

- Connaissances sur les connaissances
  - Propriétés des connaissances
  - Connaissances sur les connaissances d'un individu
  - Connaissances pour manipuler des connaissances

- Les métaconnaissances sont des connaissances

➢ L'homme est conscient des niveaux méta
  - ➢ « Je ne sais rien, mais je sais que je ne sais rien » (Socrate)
  - ➢ Pierre sait que Jacques ne connait pas son téléphone

Les connaissances qui sont des propriétés des connaissances

- Historique d'une connaissance
- Véracité d'une connaissance
- Précision d'une connaissance
- Classification des connaissances
- Intérêt d'une connaissance

Les connaissances sur les connaissances des individus

- Quel individu modélise-t-on ?
  - Soi-même
  - Un autre système d'IA
  - Des êtres humains
- Pourquoi utiliser un modèle ?
  - Pour comprendre pourquoi un individu a fait quelque chose
  - Pour prédire ce que quelqu'un va faire
- Que contient un modèle ?
  - Savoir ce qu'un individu sait
  - Savoir ce qu'un individu veut faire
  - Savoir ce qu'un individu peut faire
  - Savoir comment un individu fait quelque chose
  - Connaître les habitudes d'un individu

Connaissances pour manipuler des connaissances (1)

- Objectif : accroître la quantité et la qualité des connaissances
  - Connaissances pour acquérir des connaissances
    - Aider l'utilisateur
    - Diagnostiquer les connaissances fournies
    - Compléter les connaissances
  - Connaissances pour stocker des connaissances
    - Où
    - Comment
  - Connaissances pour découvrir de nouvelles connaissances
    - Créer une connaissance
      par généralisation, analogie, …
    - Modifier des connaissances
      par spécialisation, généralisation, élimination

Connaissances pour manipuler des connaissances (2)

- Objectif : résoudre un problème
  - Connaissances pour rechercher les connaissances en mémoire
  - Connaissances pour utiliser les connaissances
    - Définir le mode d'emploi des connaissances
    - Choisir les connaissances les plus pertinentes
    - Compiler les connaissances
- Objectif : communiquer avec un utilisateur
  - Connaissances pour exprimer des connaissances
    - Choisir ce qu'on va dire
    - Quand
    - Comment

Mais où les mettre…
Dans la base de connaissances ?

- Idée = pour rendre le MI plus efficace, on ordonne :
- Les règles
  - Physiquement
    - En imposant au moteur de lire le fichier de haut en bas, et de ne pas revenir en arrière (c'est ce que fait Prolog)
    - En découpant le fichier en « paquets » : le moteur tourne dans un paquet, puis passe au paquet suivant
  - Numériquement
    - En affectant une priorité à chaque règle
      - Si toutes les priorités sont différentes, on retrouve l'ordre ci-dessus
      - Sinon on retrouve les paquets
- Les prémisses
  - Par exemple en imposant au moteur de les évaluer de gauche à droite (ce que fait Prolog)
- Les variables

Mais où les mettre…
Dans la base de connaissances ?

- Mais …. On viole le principe du « vrac » qui est à la base des SBC
  - On recrée un langage de programmation où l'instruction coûte très cher !

- On impose un ordre qui ne tient pas compte du problème particulier
  - Par exemple, si je veux appliquer une règle aux étudiantes, j'écrirai :
    etudiant(x) ^ femme(x) ^ … -> …
  - Dans une petite ville : 5% d'étudiants et 50% de femme ☺
  - Dans une université : 90% d'étudiants et 45% de femmes ☹

- On impose un ordre qui ne tient pas compte de l'évolution du problème
  - Par exemple, si je veux gérer ma production, j'écrirai :
    pièce.terminée(x) ^ sur.chariot(x) ^ … -> …
  - Le matin : peu de pièces terminées et beaucoup de pièces en circulation
  - Mais le soir, c'est catastrophique!

Mais où les mettre…
Dans le moteur d'inférence ?

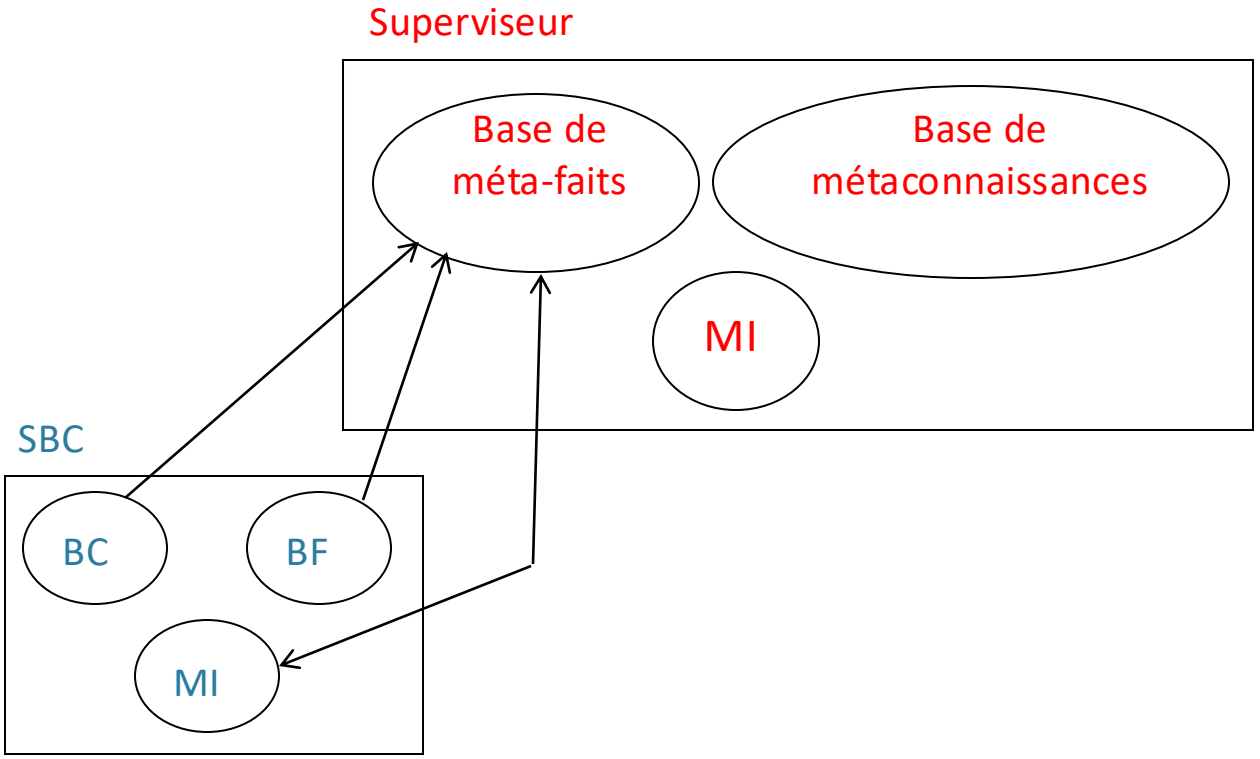
Si (phase = diagnostic)

Alors (ignorer les règles qui parlent de médicaments)

- Mais … même inconvénients que pour la base de connaissances

- Avec en plus :
  - Un moteur dorénavant spécialisé,
    pour le diagnostic médical, avec une notion de *phase*
  - Chaque fois qu'une erreur sera détectée,
    il faudra rappeler l'informaticien pour debugger le code

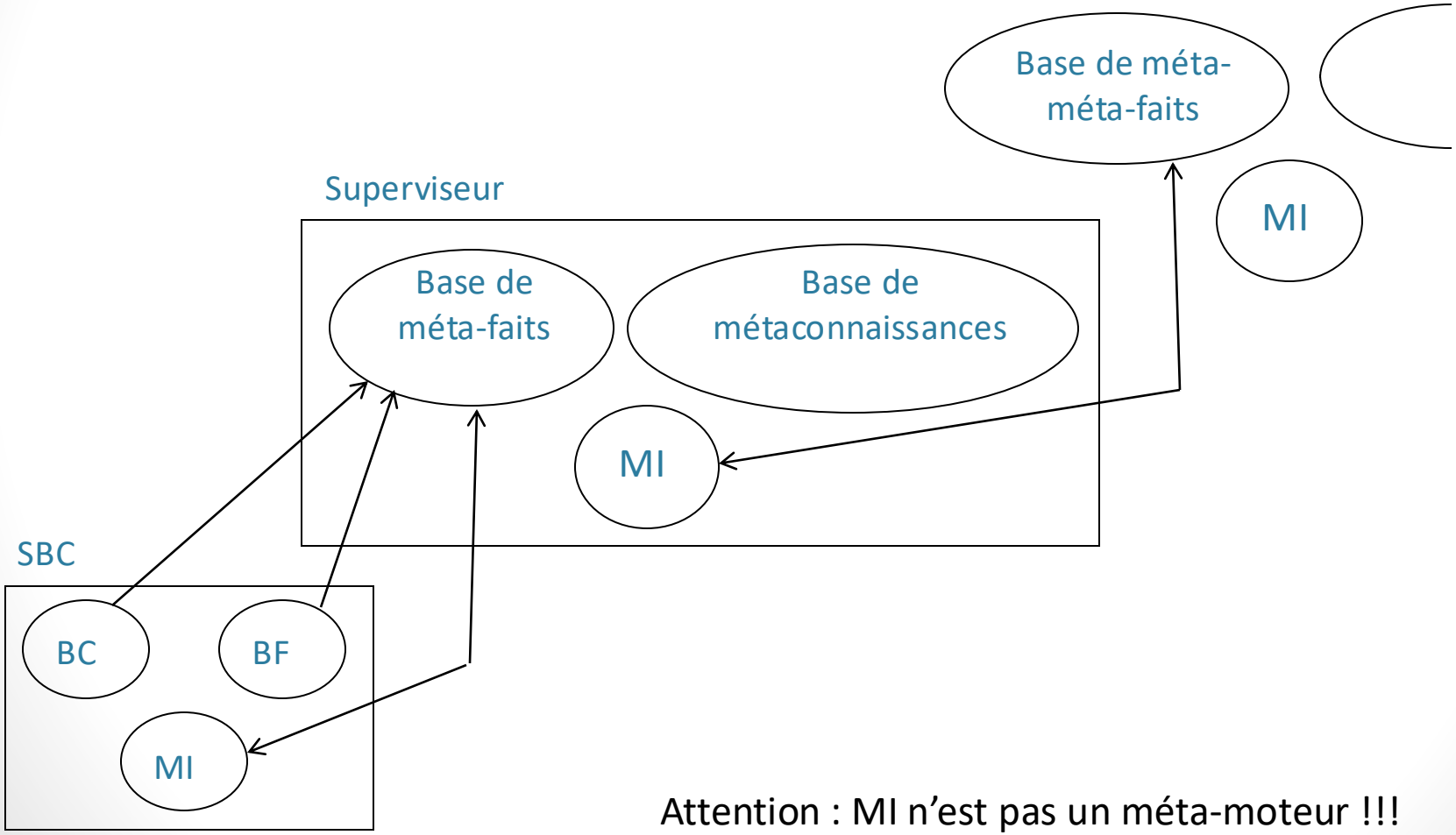
Méta-connaissance ailleurs



Tout ce qui concerne les stratégies du MI est mis dans un superviseur

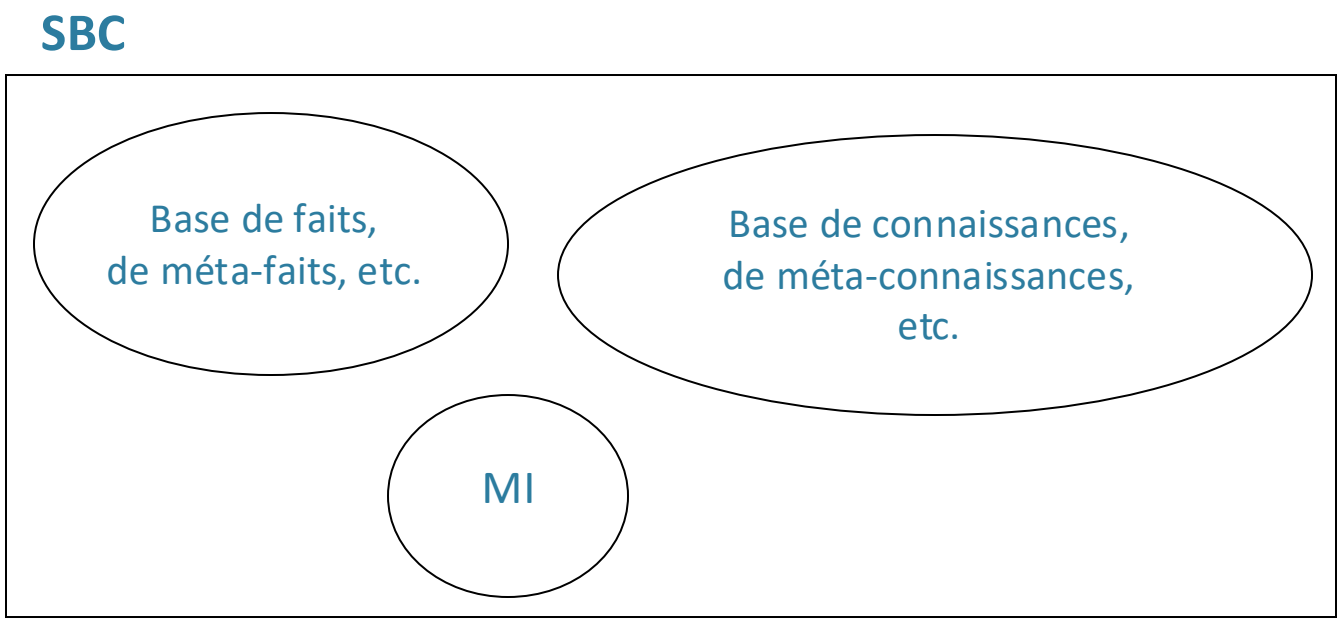
- Quand le MI a un problème, il le pose au superviseur, qui regarde tout, et répond en utilisant ses méta-connaissances

Ascension des niveaux méta



Attention : MI n'est pas un méta-moteur !!!

En réalité…


SBC
Base de faits,
de méta-faits, etc.
Base de connaissances,
de méta-connaissances,
etc.
MI

Utiliser la réflexivité pour arrêter l'ascension des niveaux méta

M1 : Si la règle R1 est identique à la règle R2
Alors éliminer la règle R2

M2 : Si la règle R1 est identique à la règle R2
Alors éliminer la règle R2

M3 : Si une règle R a deux prémisses identiques
Si une règle R a deux prémisses identiques
Alors la règle R est anormale

Ces règles sont des méta-règles, puisqu'elle parle de règles

Elles sont réflexives puisqu'elle s'appliquent à elles-même

- Il y a réflexivité chaque fois qu'un système s'applique à lui-même

En résumé…

- Méta-connaissance = connaissance 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 » : méta-connaissances sur des méta-connaissances

- Exploitation pour « contextualiser » l'usage d'un SBC
  - A l'individu, à la situation, au lieu, à l'instant, à la nécessité de précision ou non, etc…

De quoi va-t-on parler ?

- SBC pour quoi faire ?
- L'approche SBC
- L'architecture d'un SBC
- Le moteur d'inférence
- Les méta-connaissances
- Pour aller plus loin

Pour aller plus loin

- Sur les SBC
  - http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-871-knowledge-based-applications-systems-spring-2005/lecture-notes/
- Sur les méta-connaissances de Jacques Pitrat
  - Métaconnaissances, futur de l'intelligence artificielle, Hermès, 1990
  - Penser autrement l'informatique, Hermès, 1993
  - De la machine à l'intelligence, Hermès, 1995
  - Artificial Beings - The conscience of a conscious machine ISTE, Wiley, 2009
  - A Step toward an Artificial AI Scientist : http://jacques.pitrat.pagesperso-orange.fr/A%20Step%20toward%20an%20Artificial%20AI%20Scientist.pdf
- Sources pour construire ce cours
  - Les cours de Nathalie Guin, Alain Mille et Jean-Marc Fouet : http://liris.cnrs.fr/alain.mille/enseignements/Master_PRO/BIA/sommaire.htm