

# Raisonnement automatique : L'approche "logique"

Matthieu Amiguet

2008 – 2009



---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

## Ingrédients nécessaires

3

- Un ordinateur est une machine symbolique qui n'est capable que de faire des manipulations d'ordre syntaxique
- Donc pour faire "raisonner" une machine, nous aurons besoin, en première approximation,
  - D'un formalisme pour représenter symboliquement des faits
  - D'un formalisme (le même ?) pour représenter symboliquement les rapports entre ces connaissances (implication, ...)
  - D'un "moteur" pour manipuler ces connaissances et en produire de nouvelles

---

---

---

---

---

---

---

---

## Une machine qui raisonne ?

2

- Nous avons vu jusqu'à maintenant des algorithmes pour faire "réfléchir" une machine. . .
- mais il s'agissait de "raisonnements" dans des cadres très délimités
  - Jeux de stratégie
  - Recherche de bonnes solutions à des problèmes complexes
- À la recherche d'une "intelligence artificielle", on pourrait tenter d'obtenir une machine qui est capable de raisonnement générique
  - "Socrate est un homme ; tous les hommes sont mortels ; donc Socrate est mortel"
  - "Je pense donc je suis" (!)

---

---

---

---

---

---

---

---

## Le langage

4

- Depuis la Grèce Antique (Aristote, IVe siècle av. JC), l'homme (occidental) s'est intéressé à une approche systématique du raisonnement : la logique
- À la fin du XIXe siècle, on va reprendre ces travaux pour en faire une approche purement "syntaxique" du raisonnement : la logique formelle
- Permet d'étudier la validité d'un raisonnement indépendamment de la vérité de ses composantes
  - S'il pleut, j'ouvre mon parapluie ; il pleut ; donc j'ouvre mon parapluie.
  - Tous les mammifères ont des os ; le chat a des os ; donc le chat est un mammifère.

---

---

---

---

---

---

---

---

# À quoi cela pourrait-il bien servir ?

- Science-fiction
  - un ordinateur (ou robot) pensant, intelligent et autonome
- Plus réaliste
  - Systèmes d'aide à la décision
  - Systèmes de capitalisation des connaissances
  - Outils d'exploration des mécanisme du raisonnement
  - ...

---

---

---

---

---

---

---

---

---

---



---

---

---

---

---

---

---

---

---

---

- 1 Introduction à la logique formelle
  - Logique des propositions
  - Logique des prédicats
- 2 Systèmes experts
- 3 Quoi d'autre ?
- 4 Conclusion

---

---

---

---

---

---

---

---

---

---

# Validité et vérité

- Un des concepts-clé de la logique formelle est la séparation de
  - la *validité* d'un raisonnement et de
  - la *vérité* de ses composants.
- En quoi est-ce utile pour l'IA ? on peut ramener la validité d'un raisonnement à des critères de type syntaxique. . .
- Par exemple, il est très difficile d'obtenir une machine qui "sait" ce que c'est que la rougeole, mais beaucoup plus facile de faire un logiciel qui présente comme conclusion valide d'un certain nombre de symptomes un diagnostic de rougeole

---

---

---

---

---

---

---

---

---

---

# Validité et vérité – 2

**Attention !**

- On peut tirer des conclusions vraies sur la base d'un raisonnement faux (non valide). . .
- . . . ou des conclusions fausses sur la base d'un raisonnement valide (si les prémisses sont fausses)

- Comment ramener la validité d'un raisonnement à une manipulation formelle réalisable automatiquement ?
- Premier essai : on se donne des variables qui représentent des affirmations (atomiques) pouvant être vraies ou fausses
  - $p \equiv$  "La neige est blanche"
  - $q \equiv$  "L'IA est une matière passionnante"
- Ces "éléments atomiques de vérité" s'appellent des *propositions*.

---

---

---

---

---

---

---

---

---

---



---

---

---

---

---

---

---

---

---

---

- Dans la logique "classique" (aussi appelée "bivalente"), chaque proposition peut être *vraie* ou *fausse*
- Une formule composée de  $n$  proposition peut donc être "plongée" dans  $2^n$  mondes en fonction de la valeur de vérité de ses propositions
- L'attribution d'une valeur de vérité à toutes les propositions d'une formule s'appelle une *interprétation* de cette formule
- On peut en déduire une valeur de vérité pour la formule complète sur la base de *tables de vérité*

---

---

---

---

---

---

---

---

---

---

- Les propositions peuvent être reliées entre elles pour former des expressions complexes à l'aide d'opérateurs logiques :

"et"  $\wedge$   
 "ou"  $\vee$   
 "non"  $\neg$   
 "implique"  $\rightarrow$   
 ...

- $p \rightarrow \neg q$  : "La neige est blanche donc l'IA n'est pas une matière passionnante".

---

---

---

---

---

---

---

---

---

---

$p$	$\neg p$
0	1
1	0

$p$	$q$	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

$p$	$q$	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

$p$	$q$	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

...

- La plupart des formules sont parfois vraies, parfois fausses, suivant dans quel monde on les plonge (formules *contingentes*)
- Certaines formules sont fausses dans tous les cas (formules *insatisfiables*)
- Certaines formules sont vraies dans toutes les interprétations possibles. Ces formules sont dites *valides*. Elles représentent les grans principes des raisonnements "corrects".

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

- La logique des propositions ne "descend" jamais au-dessous de la phrase complète
- Ainsi, un raisonnement comme "Tous les hommes sont mortels ; Socrate est un homme ; donc Socrate est mortel" se traduit au mieux comme  $p \wedge q \rightarrow r$
- Cette dernière formule n'est pas valide, alors que le raisonnement l'est !
- Pour rendre compte de ceci, on va être obligé de disséquer les phrases elles-mêmes.

---

---

---

---

---

---

---

---

- On dispose donc d'un outil complètement automatique pour calculer
  - La valeur de vérité d'une formule en fonction de la valeur de vérité de ses composants
  - La validité d'une formule
- À noter par contre que notre méthode de preuve est d'une complexité exponentielle en fonction du nombre de propositions. . .

---

---

---

---

---

---

---

---

- Un *prédicat* représente une propriété d'un objet ou une relation entre deux objets. Techniquement, c'est une fonction qui prend en paramètre un ou plusieurs "objets du monde" et retourne une valeur de vérité
- Exemples
  - majeur( *pierre* ) "Pierre est majeur"
  - frere( *pierre, paul* ) "Pierre est le frere de Paul"

- Sur la lancée, on va aussi introduire les *fonctions*, qui étant donné un ou plusieurs objets du monde retourne un objet du monde

$pere(pierre)$  "Le père de Pierre"

$majeur(pere(pierre))$  "Le père de Pierre est majeur"

---

---

---

---

---

---

---

---

- Une formule contenant des prédicats, des fonctions et des variables quantifiées joue le même rôle qu'une proposition dans la logique des propositions

$\forall x (homme(x) \rightarrow mortel(x))$

- On peut donc les combiner avec les connecteurs logiques vus précédemment

$(\forall x (homme(x) \rightarrow mortel(x)) \wedge homme(Socrate)) \rightarrow mortel(Socrate)$

---

---

---

---

---

---

---

---

- Plutôt que de travailler seulement avec des constantes ("Pierre", "Paul"), on pourrait aussi utiliser des variables. . .
- Oui mais. . . à première vue,  $majeur(x)$  ("x est majeur") peut avoir deux significations
  - Il existe au moins un  $x$  qui est majeur
  - Tous les  $x$  sont majeurs
- Pour distinguer ces deux significations, on va introduire les *quantificateurs*  $\exists$  ("il existe") et  $\forall$  ("pour tout")
  - $\exists x majeure(x)$
  - $\forall x majeure(x)$

---

---

---

---

---

---

---

---

- Pour l'interprétation (attribution d'une valeur de vérité aux formules), on ne peut plus travailler au niveau des propositions. Il nous faudra donc
  - Choisir un domaine d'interprétation  $D$  (les "objets du monde")
  - Attribuer à chaque fonction une fonction  $D^{(n)} \rightarrow D$
  - Attribuer à chaque prédicat une fonction  $D^{(n)} \rightarrow \{0, 1\}$
- On peut ensuite calculer récursivement la valeur d'une formule.

---

---

---

---

---

---

---

---

- Comme pour la logique des propositions, on peut définir les formules
  - Valides (vraies dans toutes les interprétations)
  - Insatisfiables (fausses dans toutes les interprétations)
  - Contingentes (parfois vraies, parfois fausses)
- ... Mais le "dans toutes les interprétations" représente un terrain *beaucoup* plus vaste qu'avant (et même infini...)

---

---

---

---

---

---

---

---

- La logique des prédicats permet de ramener (en partie) le raisonnement logique dans le domaine du calculable
- L'utilisation de ces techniques pour implémenter différentes formes de raisonnement automatique forme le domaine des *systèmes experts*
- Les systèmes experts diffèrent selon
  - Le type de logique utilisé
  - Utilisation du langage complet ou d'une sous-partie (efficacité...)
  - "Sens" d'utilisation de la logique (des prémisses vers les conclusions ou l'inverse)
  - ...

---

---

---

---

---

---

---

---

- Avec la logique des prédicats, il est toujours facile de calculer la valeur de vérité d'une formule dans une interprétation donnée...
- ... mais on ne peut plus démontrer la validité d'une formule en parcourant toutes les interprétations possibles
- On a donc recours à une forme de démonstration par l'absurde nommée *principe de résolution*
  - Dans le cas général, la complexité d'une preuve en utilisant ce principe est toujours exponentielle
  - Mais si on restreint le langage, on peut se ramener à une complexité *linéaire*

---

---

---

---

---

---

---

---

1 Introduction à la logique formelle

- 2 Systèmes experts
- Généralités
  - La base de connaissances
  - Moteurs d'inférence
  - La fin de l'IA ?

3 Quoi d'autre ?

4 Conclusion

Système expert (SE)

Un système expert est un logiciel qui reproduit le comportement d'un expert humain accomplissant une tâche intellectuelle dans un domaine précis.

"Il est plus facile de simuler un géologue qu'un enfant de 5 ans."

- Un SE est généralement conçu pour résoudre des problèmes de *classification* ou de *décision*
  - diagnostic médical, prescription thérapeutique, analyse de résultats, ...

---

---

---

---

---

---

---

---

---

---

- Un SE n'est concevable que pour les domaines dans lesquels il existe des *experts humains*
- Un expert est quelqu'un qui connaît un domaine et qui est (plus ou moins) capable de *transmettre* ce qu'il sait
  - Ce n'est par exemple pas le cas d'un enfant par rapport à sa langue maternelle.

---

---

---

---

---

---

---

---

---

---

- Au-delà de la définition généraliste, le terme "système experts" s'est spécialisé pour définir un système basé sur la logique formelle et ayant la structure suivante :
  - Une *base de connaissances*, souvent composée de deux parties :
    - une *base de règles*, qui modélise la connaissance des *mécanismes* du domaine considéré
    - une *base de faits*, qui contient les *informations* concernant le cas que l'on est en train de traiter
  - Un *moteur d'inférence* capable de *raisonner* à partir des informations contenues dans la base de connaissances, de faire des déductions, etc.
- Un tel système est également appelé "système à base de règles"

---

---

---

---

---

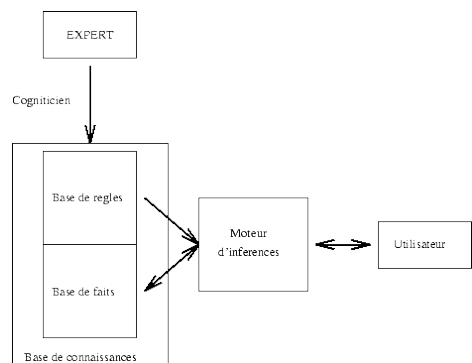
---

---

---

---

---



- Rôle du cogniticien
  - “*soutirer*” leurs connaissances aux experts du domaine et
  - *traduire* ces connaissances dans un formalisme se prêtant à un traitement automatique
- Ces deux tâches sont aussi délicates l'une que l'autre :
  - Un expert est la plupart du temps inconscient de la majeure partie de son savoir...
  - ... et s'il arrive à en exprimer une partie, c'est souvent sous une forme qui ne se laisse pas facilement formaliser.
- De plus l'extraction de connaissances peut avoir un effet perturbant sur l'expert
  - On a observé que si l'on demande – par exemple – à un maître aux échecs de réfléchir à sa manière de jouer, on observera une baisse de ses performances dans les jours ou les semaines qui suivent cet effort d'introspection !

---

---

---

---

---

---

---

---

---

---

1) Please enter information about the patient.

Name : Sally  
Age : 42 years  
Sex : Female  
Race : Caucasian

[...]

13) Has SALLY recently has symptoms of persistent headache or other abnormal neurologic symptoms (dizziness, lethargy, etc) ?

Yes

---

---

---

---

---

---

---

---

---

---

- SE conçu entre 1972 et 1974 pour aider au diagnostic des maladies infectieuses du sang
- Initialement, contenait environ 200 règles
  - Complétée quelques années après pour tenir compte des méningites (300 règles)
- Même dans la version de 1974, les résultats sont concluants :
  - 15 cas représentatifs d'infections bactériennes ont été soumis à MYCIN d'une part et à 5 experts d'autre part
  - Les 5 experts ont approuvé les recommandations thérapeutiques de MYCIN dans 11 cas sur 15 (55 accords pour 75 comparaisons)...
  - ... Dans les autres cas, les 5 experts étaient en désaccord entre eux !

---

---

---

---

---

---

---

---

---

---

14) Has SALLY recently had objective evidence of abnormal neurologic signs (nuchal rigidity, coma, seizures, etc.) documented by physician observation or examination?

Yes

The CSF cultures will be considered to be associated with MENINGITIS.

[...]

---



38) Is SALLY a burn patient?

Why?

[4.0] This will aid in determining the organisms (other than those seen on cultures or smears) which might be causing the infection.

It has already been established that

[4.1] the infection which requires therapy is meningitis, and

[4.2] organisms were not seen on the stain of the pending csf culture (CULTURE-1), and

[4.3] the type of the infection may be bacterial

---

---

---

---

---

---

---

---

Therefore, if

[4.4] SALLY has been seriously burned then

there is suggestive evidence that *Pseudomonas-aeruginosa* is one of the organisms (other than those seen on cultures or smears) which might be causing the infection [RULES578]

---

---

---

---

---

---

---

---

- DENDRAL, 1969, chimie, recherche la formule développée d'un corps organique à partir de la formule brute et du spectrogramme de masse du corps considéré
- CRYNALIS, 1979, chimie, recherche la structure de protéines à partir de résultats d'analyse cristallographique
- MOLGEN, 1977, biologie, engendre un plan de manipulations génétiques en vue de construire une entité biologique donnée
- PROSPECTOR, 1978, géologie, aide le géologue à évaluer l'intérêt d'un site en vue d'une prospection minière (1600 règles)
- ...

---

---

---

---

---

---

---

---

- Un système expert concerne un domaine précis d'application
- Un *système expert général* (ou *générateur de systèmes experts*) est un outil de développement de systèmes experts particuliers
- Un système expert général contient un moteur d'inférences et un système de représentation des connaissances.

- Historiques :
  - VP-Expert
  - SNARK
  - GURU
  - SMECI
  - ...
- Actuels
  - PROLOG (langage de programmation indépendant, à base de règles)
  - CLIPS (langage de programmation indépendant ou intégration dans un autre langage)
  - JESS (moteur de règles intégré à Java)
  - ...

---

---

---

---

---

---

---

---

---

---

- On peut classer les SE en fonction des types de faits qu'ils gèrent :
  - Un SE d'ordre 0 n'est capable de traiter que des faits booléens
  - Un SE d'ordre 0+ peut gérer des faits de type symbolique ( $\cong$  énuméré) ou numérique, mais *sans utiliser de variables*
  - Un SE d'ordre 1 admet tous les types de faits et est capable de gérer les variables.

---

---

---

---

---

---

---

---

---

---

- La base de faits est la *mémoire de travail* du SE
- Elle évolue donc au cours du temps
  - Au départ, elle contient les éléments connus du cas à étudier
  - Au fil du temps, elle sera complétée par les faits produits par le moteur d'inférence et/ou les informations demandées à l'utilisateur

---

---

---

---

---

---

---

---

---

---

- Un SE d'ordre 0 pourra par exemple contenir une condition du style

```
actif ou !retraité
```
- Pour l'ordre 0+, on pourrait avoir

```
actif et (profession médecin ou
rémunération 100'000)
```
- Et pour l'ordre 1

```
maladie(X) et X!=grippe et
symptome(X)="forte fièvre"
```

- Dans certains cas, la base de connaissance ne suffit pas, il faut des connaissances *sur* les connaissances, ou des *métaconnaissances*
- Par exemple, il faut savoir si une valeur a été attribuée à un fait. Dans le cas contraire, on peut la demander à l'utilisateur...
- ... mais si l'utilisateur ne sait pas, il faut s'en souvenir pour ne pas lui reposer sans arrêt la même question !

---

---

---

---

---

---

---

---

- La base de règles rassemble la connaissance et le savoir-faire de l'expert
- Elle n'évolue donc pas au cours d'une session de travail
- Une règle est souvent de la forme  
Si <conditions> alors <conclusion>  
où les conclusions sont de la forme <Fact> = <valeur>
- Exemple : **Si** intérêt\_cours < 0 **et** fatigué **alors** dort=vrai

---

---

---

---

---

---

---

---

- La valeur d'un fait peut par exemple être
  - *connue* si une valeur a été attribuée au fait
  - *inconnue* sinon, et si aucune question n'a été posée à son sujet
  - *indéterminée* si aucune valeur n'est attribuée au fait, que l'utilisateur a été interrogé et qu'il a répondu "je ne sais pas"
- Ainsi, valeur (profession) est un métafait symbolique et valeur (profession) =connue est une métacondition

---

---

---

---

---

---

---

---

- Avec des règles comme ci-dessus, il est facile de "simuler" en plusieurs règles des règles de la forme
  - **Si A ou B alors C**
  - **Si A alors B et C**
- Par contre, il n'est pas possible de représenter
  - **Si A alors B ou C**

---

---

---

---

---

---

---

---

- De même qu'il est parfois nécessaire d'avoir de la connaissance sur la connaissance, il peut être utile d'avoir des règles sur l'utilisation des règles : les *métarègles*

- Exemple tiré de MYCIN :

**Si** le patient est un hôte à risque **et** s'il existe des règles mentionnant des pseudo-monias dans une prémisse **et** s'il existe des règles mentionnant des klebsiellas dans une prémisse **alors** il est probable qu'il faille utiliser les premières avant les secondes

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

- L'idée est de partir de la base de règles et de générer des nouveaux faits jusqu'à ce qu'on ne puisse plus rien faire
- Dans certains cas, on cherche à produire un fait particulier ; on peut donc s'arrêter dès qu'on l'a trouvé
- D'autres fois, on cherche l'ensemble des faits déductibles de la base
- Pour choisir quelle règle appliquer à quel fait, on peut utiliser les métarègles ou une heuristique
- L'algorithme "naïf" est simple, mais devient vite très inefficace !
  - Il existe une optimisation assez efficace, bien que gourmande en mémoire, connue sous le nom d'*algorithme de Rete*
- Exemples : CLIPS, JESS

---

---

---

---

---

---

---

---

- Quand on a des faits et des règles, il faut un "moteur" qui permette d'appliquer les règles et d'en déduire des nouveaux faits : on l'appelle le *moteur d'inférence*
- Habituellement, le moteur d'inférence est (presque) complètement indépendant du domaine d'application
- Il en existe essentiellement trois catégories
  - moteurs en chaînage avant
  - moteurs en chaînage arrière
  - moteurs en chaînage mixte.

---

---

---

---

---

---

---

---

- Le mécanisme de chaînage arrière consiste à
  - partir du fait que l'on souhaite établir
  - rechercher toutes les règles qui concluent sur ce fait
  - établir la liste des faits qu'il suffit de prouver pour qu'elles puissent se déclencher
  - appliquer récursivement le même mécanisme aux faits contenus dans ces listes
- Le moteur en chaînage arrière le plus connu est le moteur Prolog.

---

---

---

---

---

---

---

---

- Il existe toute une variété de systèmes avec diverses combinaisons des chaînages avant et arrière.
- On parle alors de moteurs en chaînage mixte
- Une idée simple peut être de générer (en chaînage avant) tous les faits possibles, puis déterminer (en chaînage arrière) une question pertinente à poser à l'utilisateur, avant de recommencer
- Un exemple plus complexe est le système de propagation de contraintes TROPIC

---

---

---

---

---

---

---

---

---

---

- Une fois les règles entrées dans TROPIC, le système fonctionne sur une alternance de chaînage avant et de chaînage arrière :
  - La décomposition en sous-problèmes se fait en chaînage arrière
  - Une fois les sous-problèmes résolus, on en tire toutes les conséquences en chaînage avant
  - Si la solution est contradictoire ou non-satisfaisante, on fait du *retour arrière* pour trouver une autre décomposition.
- Attention : chaînage arrière  $\neq$  retour arrière!!!

---

---

---

---

---

---

---

---

---

---

- Développé par Jean-Claude Latombe en 1977
- Domaine visé : conception assistée par ordinateur
- Le système comporte deux types de règles
  - Les *règles de décomposition*, qui expliquent comment le problème se décompose en sous problème, en tenant compte des contraintes qui en résultent
  - Les *règles d'analyse*, qui permettent de tirer toutes les conséquences d'une décomposition donnée
- Exemple : pour concevoir un avion, il faut concevoir des ailes, un fuselage, etc (règles de décomposition). Le poids total de l'appareil est le total des poids des composants (règle d'analyse).

---

---

---

---

---

---

---

---

---

---

- Nous avons vu une manière d'obtenir un système qui effectue des raisonnements. . .
- . . . cela résout-il tous les problèmes de l'IA d'un coup ?
- Dans les faits, non !
  - Problèmes de complexité des bases de connaissances
  - Problèmes d'efficacité
  - Problèmes de puissance d'expression de la logique
  - Et même si tout cela est résolu. . . il reste Gödel !

---

---

---

---

---

---

---

---

---

---

- 1 Introduction à la logique formelle
- 2 Systèmes experts
- 3 **Quoi d'autre ?**
- 4 Conclusion

---

---

---

---

---

---

---

---

---

---



---

---

---

---

---

---

---

---

---

---

Raisonnement automatique  
Quoi d'autre ?

## Autres logiques 55

- La logique "habituelle" (logique des prédicats du premier ordre) ne permet pas (ou difficilement) . . .
  - ... de gérer les aspects temporels : "S'il pleut aujourd'hui, il y aura des champignons demain"  $\rightsquigarrow$  logique temporelle
  - ... de gérer des connaissances incomplètes ou des connaissances sur les connaissances  $\rightsquigarrow$  logiques modales
  - ... de gérer la révision des connaissances : un fait établi l'est définitivement  $\rightsquigarrow$  logique non-monotone
  - ... de gérer l'incertitude sur les croyances  $\rightsquigarrow$  logique floue
  - ... de travailler sur les caractéristiques (prédicats) : "Pour toute caractéristique, si un objet a cette caractéristique alors. . ."  $\rightsquigarrow$  logiques d'ordre supérieur
  - ...

---

---

---

---

---

---

---

---

---

---

Raisonnement automatique  
Quoi d'autre ?

## Alternatives et extensions 54

- Nous l'avons vu, les SE (dans le sens "système à base de règles) ne résolvent pas tout
- "Some Guidelines For Deciding Whether To Use A Rules Engine" :  
<http://www.jessrules.com/jess/guidelines.shtml>
- Si un SE classique ne semble pas adéquat, quelles sont les alternatives ?
  - Algorithmes ad-hoc
  - Systèmes à base de règles, mais utilisant *une logique étendue*
  - Autres modèles de raisonnement (notamment modèles connexionnistes)

---

---

---

---

---

---

---

---

---

---

Raisonnement automatique  
Quoi d'autre ?

## Modèles connexionnistes 56

- Si le problème est difficile à représenter sous la forme d'un ensemble de règles, il est parfois plus simple de le représenter sous la forme d'un graphe
  - Réseaux de neurones – se prêtent bien à la classification et à l'apprentissage
  - Réseaux bayésiens – outil de raisonnement probabiliste
  - ...

---

---

---

---

---

---

---

---

---

---

- 1 Introduction à la logique formelle
- 2 Systèmes experts
- 3 Quoi d'autre ?
- 4 Conclusion

---

---

---

---

---

---

---

---

---

---



---

---

---

---

---

---

---

---

---

---

Raisonnement automatique  
Conclusion

## Les points sensibles des SE 59

- Importance de la cognitive
- Il n'est pas toujours facile – ou même possible — de formaliser complètement le savoir-faire
- Gros investissement nécessaire (temps, coûts et hommes)
- Algorithmique des meta-règles/connaissances

---

---

---

---

---

---

---

---

---

---

Raisonnement automatique  
Conclusion

## Les points forts des SE 58

- Formalisation du savoir-faire de l'entreprise
  - Meilleure transmissibilité !
- Baisse du niveau d'expertise exigé pour certaines tâches (pas toujours !)
  - Parfois, les systèmes experts ne peuvent être utilisés que par... des experts !
- Gestion simultanée de nombreuses ressources et contraintes
- Utilisation du SE comme outil d'analyse et de simulation

---

---

---

---

---

---

---

---

---

---

Raisonnement automatique  
Conclusion

## Conclusion 60

- Les SE sont une des applications de l'IA qui ont quitté les laboratoires de recherche pour être utilisées dans le monde professionnel
- De nombreux systèmes experts ont été implémentés avec succès pour résoudre des problèmes concrets
- Mais : les grandes difficultés que l'on rencontre lorsqu'on cherche à extraire des experts leur connaissance, puis lorsqu'on tente de formaliser ces connaissances sont peut être signes d'une faiblesse intrinsèque des SE
  - Hypothèse fondamentale des SE : la structuration de nos aptitudes en règles symboliques parfaitement identifiables
  - Majoritaire chez les psychologues et chercheurs en intelligence artificielle dans les années 70, cette hypothèse est de plus en plus contestée...

- Les modèles psychologiques de l'IA ont évolué vers plus de souplesse
  - Modèles connexionnistes (Réseaux de neurones, ...)
  - Divers types de raisonnement incertain (Réseaux bayésiens, ...)
  - Modèles adaptatifs (Q-learning, algorithmes génétiques, ...)
  - Modèles décentralisés (Systèmes multi-agents, ...)
- L'avenir des SE se situe probablement dans des systèmes hybrides avec ces différentes techniques (ou d'autres...)

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---