

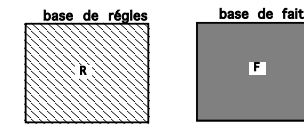
Contrôle dans les systèmes experts

Interprétation des connaissances dans les systèmes à base de connaissance

2004

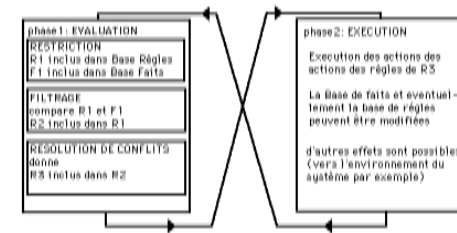
- Cycle de base de l'interpréteur
- Sans et avec variables
- Chaînage avant
- Chaînage arrière
- Modes de raisonnement

Cycle de base de l'interpréteur



cycle d'évaluation / exécution (H.FARRENY):

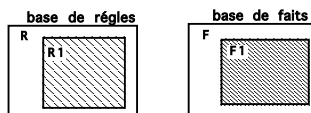
1. Phase d'évaluation
2. Phase d'exécution



Phase d'évaluation

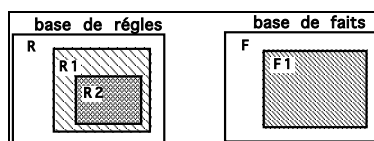
Etape de sélection ou restriction :

- Sélection d'un sous-ensemble de règles R1 et de faits F1, pouvant être rattaché à des problèmes, buts, hypothèses, contexte. (règles et/ou faits propres à l'analyse du sang etc)



Etape de filtrage (Pattern-matching) :

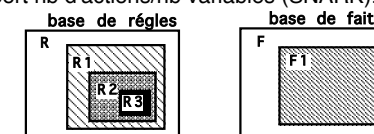
- Ensemble R2 de R1 de règles, dit ensemble de règles activables, dont les prémisses ont été jugées satisfaisantes/ F1.



Phase d'évaluation

Etape de choix ou résolution de conflits :

- L'ensemble $F1 \times R2$ est appelé "ensemble de conflit". L'interpréteur choisit un sous-ensemble R3 de règles à déclencher effectivement:
 - degré de fiabilité des règles,
 - signification dans un contexte de l'application,
 - selon des heuristiques tels que:
 - les règles étant arbitrairement ordonnées; (ex: sélectionner la première)
 - règles de "coût" minimal (ex: moins de conditions à vérifier..)
 - règles présentant un "gain" espéré plus élevé, (ex: plus de conséquences..)
 - règles qui ont le moins servi,
 - règles dont le rapport nb d'actions/nb variables (SNARK)....



Phase d'évaluation



R1 et F1 résultent de l'étape de sélection
R2 résulte de l'étape de filtrage sur F1
R3 résulte de l'étape de choix dans R2

Phase d'exécution

l'interpréteur déclenche les règles de R3 qui peuvent:

- **avoir des actions sur la base de faits:**
 - ajout, modification, destruction d'un fait
- **se désactiver ou désactiver des règles pour les cycles suivants:**
 - contrôle,...
- **avoir une action sur un interface:**
 - affichage à l'écran,
 - poser une question
 - chercher une connaissance ds Bdd
 - etc...

Phase d'exécution : particularités liées à l'utilisation de variables

2 étapes la phase d'exécution (SNARK):

- instanciation de la règle:
- déclenchement de la règle:

Étape d'instanciation de la règle

instancier les variables de la règles en fonction de la base de faits:

- 1• choix d'une prémisse dans la règle sélectionnée (R3) (critères de choix)
- 2• une instanciation de cette prémisse avec les faits de la base de faits
- 3• propagation de cette instanciation sur les autres prémisses de la règle
- 4• choix d'une autre prémisse parmi les prémisses restantes et itération (retour en 1)

- la propagation de l'instanciation d'une prémisse introduit de nouvelles contraintes dans l'instanciation des prémisses suivantes
- il peut arriver qu'une prémisse, compte tenu de ces contraintes ne puisse être instanciée
- il faut alors effectuer un retour-arrière sur les instanciations précédentes.

Phase d'exécution : particularités liées à l'utilisation de variables

Étape de déclenchement de la règle

Lorsque toutes les prémisses sont instanciées:

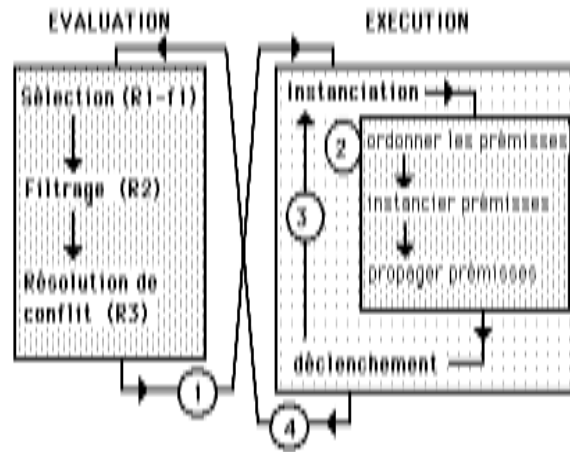
- on a alors une instance de la règle,
- cette instance peut être déclenchée et conduire par exemple à la modification de la base de fait.

Une fois cette instance de règle exécutée:

- on retourne dans l'étape précédente afin de construire une nouvelle instance de la règle

Cycle d'interprétation dans SNARK

(règles avec variables)



Modes d'inférence de l'interpréteur

Chaînage avant :

permet de répondre à la question:

"QUE PEUT-ON DEDUIRE DES FAITS SUIVANTS?"

à partir des faits initiaux, on déclenche itérativement toutes les règles qui peuvent l'être, jusqu'à :

- satisfaction d'un but donné
- ou saturation complète de la base de faits

Chaînage arrière :

permet de répondre à la question:

"A PARTIR DES FAITS SUIVANTS, PEUT-ON ATTEINDRE CE BUT?"

- à partir d'un but donné par un utilisateur, ou choisi par le système, on remonte récursivement vers les faits initiaux,
- on décrit ainsi un arbre de recherche ET/OU (un fait peut être démontré de plusieurs façons s'il est en partie droite de plusieurs règles -> nœud OU;
- tous les faits en partie gauche d'une règle doivent être démontrés pour affirmer la partie droite ET);
- le système peut poser des questions à l'utilisateur pour tenter de compléter la base de faits initiale.

Inférence en chaînage avant (ou dirigé par les données)

Les faits = données dont la valeur de vérité est

- soit déjà établie: **raisonnement déductif**,
- soit supposée: **raisonnement hypothético-déductif**.

Principe général:

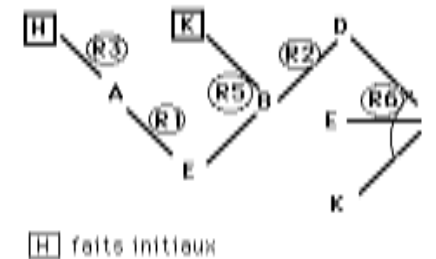
Règles déclenchées entre états de la base de faits

R3-> R1-> R5-> R2-> R6->

base de règles	base de faits					
BR	BFi (état initial)	BF1	BF2	BF3	BF4	BF5
R1: A->E	H	H	H	H	H	
R2: B->D	K	K	K	K	K	
R3: H->A		A	A	A	A	
R4: E et G->C			E	E	E	E
R5: E et K->B				B	B	B
R6: D et E et K->C					D	D
R7: G et K et F->A						C

Inférence en chaînage avant (ou dirigé par les données)

Soit le graphe:



Le chaînage avant correspond à l'application du **"modus-ponens"** défini en logique formelle:

P (et) P => Q -> Q

Inférence en chaînage avant en logique d'ordre 1

- Chaînage plus **difficile** à réaliser qu'en logique d'ordre 0.
- Il s'appuie sur l'existence
 - d'une règle dont la condition $F(x)$ et de conclusion $C(x)$,
où x est une variable,
 - d'un fait établi $F(a)$, pour introduire $C(a)$ dans la base de faits.

On a une combinaison de :

• *"modus-ponens"*

et de

• *"spécialisation universelle"*

Avantages et inconvénients du chaînage avant

Interpréteur fonctionnant en chaînage avant:

Avantages :

- Exécution de toutes les règles sélectionnées dans la phase d'évaluation,
=> saturation de la base de faits.
- Simple à assurer,

Inconvénients :

- Toutes les règles sont déclenchées, même si les conclusions de certaines n'offrent aucun intérêt,
- Il est nécessaire de charger dans l'espace de travail toutes les connaissances que l'on possède, sans savoir si tous les éléments seront utiles et s'il n'en manque pas.

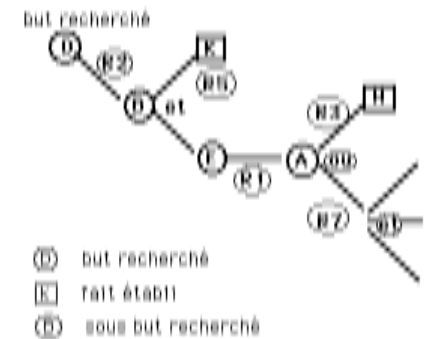
Inférence en chaînage arrière

Problème : but recherché : D

Base de règles	Base de faits
BR	BFi (état initial)
R1: $A \rightarrow E$	H
R2: $B \rightarrow D$	K
R3: $H \rightarrow A$	
R4: $E \text{ et } G \rightarrow C$	
R5: $E \text{ et } K \rightarrow B$	
R6: $D \text{ et } E \text{ et } K \rightarrow C$	
R7: $G \text{ et } K \text{ et } F \rightarrow A$	

Inférence en chaînage arrière

Soit le graphe d'enchaînement:



Le chaînage arrière correspond à l'application du *"modus-tollens"* défini en logique formelle

Chaînage arrière en logique d'ordre 1

- Face à un problème $P(x)$, le chaînage arrière utilise la spécialisation universelle pour invoquer une règle (instanciation de la règle), dont la condition est de la forme $Q(x)$.
 - La **phase d'évaluation** des règles de l'interpréteur se réfère maintenant à un problème en suspens pouvant être sélectionné dans un agenda de problèmes.
 - Dans la **phase d'exécution**, de nouveaux problèmes définis par les prémisses de la règle peuvent être introduits dans la base de faits et s'intégrer à l'agenda.
 - Les problèmes sont considérés comme résolus lorsque les problèmes introduits par le déclenchement de la règle sont soit d'office résolus (faits établis) soit résolus par unification.
- L'exécution d'une telle règle, n'est éventuellement réalisée qu'après le déclenchement, et non en même temps comme c'est le cas pour le chaînage avant.

Avantages du chaînage arrière

- L'interpréteur peut poser lui-même des questions après avoir exploité toutes les possibilités d'y répondre par lui-même.
- Nombre de filtrages réduit de membres de gauche de règles,
=> l'explosion combinatoire réduite.

Inconvénients du chaînage arrière

- Problème des **raisonnements circulaires**:
cas où un but est un sous-but de sa propre arborescence de recherche.

Modes de raisonnements dans les systèmes experts

Le raisonnement de surface:

- ne fait pas appel à un modèle théorique plus profond.
- symptômes considéré sans relation de cause à effet: règles rattachant des symptômes à la probabilité de différentes bactéries

Le raisonnement en profondeur:

- fait appel à un modèle théorique permettant de considérer des relations de cause à effet. Exemple, pour un diagnostic médical, il reliera des symptômes à un modèle de simulation de la maladie; la thérapie suggérée est alors dérivée du modèle

En classification:

- utilisation de règles de surface, car plus faciles à définir et en partie parce que la connaissance disponible peut être incomplète.

En conception:

- systèmes basés sur des connaissances complètes, des raisonnements en profondeur car le but est de construire des modèles.

En aide à la décision:

- les 2 types de raisonnement peuvent être combinés.

Autre typologie des raisonnements:

Raisonnement formel :

- = manipulation syntaxique de données pour déduire de nouveaux faits suivant des règles d'inférences prescrites. La logique mathématique est la représentation formelle de la connaissance et le calcul de prédicat la technique de déduction.

Raisonnement procédural :

- utilise la simulation de modèles du domaine pour poser des questions et résoudre les problèmes.

Raisonnement par analogie :

- mode très naturel de pensée pour les humains, difficile à mettre en œuvre sur des ordinateurs.

Généralisation et abstraction :

- ce sont aussi des processus de raisonnement très naturels chez l'être humain, difficile à implanter aussi.

Méta-raisonnement :

- raisonnement sur de la méta-connaissance
- de récentes recherches en psychologie et en intelligence artificielle indiquent que le méta-raisonnement joue un rôle central dans tous les processus cognitifs humains et en particulier d'apprentissages.