TD4 : Systèmes à base de règles (ordre 0)

Exercice 1. Chaînage avant (application directe du cours)

Soit la base de faits BF = {A,D,G}.
Soit la base de règles BR = {R1, …, R5}, avec :

R1 : $A \wedge B \rightarrow C$
R2 : $A \wedge C \rightarrow E$
R3 : $D \wedge F \rightarrow E$
R4 : $E \wedge F \rightarrow H$
R5 : $G \rightarrow F$

1) Quels faits peuvent être ajoutés à BF par chaînage avant avec BR ?
2) Y-a-t-il des règles qui ne sont pas déclenchées sur cet exemple ?

Exercice 2. Complexité de l'algorithme de chaînage avant (complément au cours)

On considère l'algorithme de chaînage avant à base de compteurs vu en cours (principe : à chaque règle est associé un compteur, dont la valeur correspond au nombre d'atomes de son hypothèse qui n'ont pas encore été reconnus comme faits ; la règle devient applicable lorsque son compteur passe à zéro) :

```
Algorithme FC(K) // saturation de la base K
// Données : K = (BF, BR)
// Résultat : BF saturée par application des règles de BR
Début
  ATraiter ← BF
  Pour toute règle R de BR
    Compteur(R)← Nombre de symboles de l'hypothèse de R
  Tant que ATraiter n'est pas vide
    Retirer un atome A de ATraiter
    Pour toute règle de BR ayant A dans son hypothèse
      Décrémenter Compteur(R)
      Si Compteur(R) = 0 // R est applicable
            Soit C la conclusion de R
            Si C ∉ BF // l'application de R est utile
                  Ajouter C à ATraiter
                  Ajouter C à BF
    FinPour
  FinTantQue
Fin
```

1) Montrer sur un exemple que la condition « $C \notin BF$ » (test de l'utilité de l'application de R) n'a pas seulement pour but d'éviter d'ajouter des faits en double dans BF mais que sans elle l'algorithme serait incorrect : il produirait des faits qui ne devraient pas être produits.
2) Imaginer des structures de données qui permettent :
   a. de ne considérer que les règles dont l'hypothèse contient A, lorsque A est traité
   b. de tester en temps constant si C est dans BF.

   Etant données ces structures, montrer que l'algorithme a une complexité linéaire en la taille de K (c'est-à-dire effectue un nombre d'opérations borné par *une constante* x *taille(K)*).

Exercice 3. Adéquation et complétude du chaînage avant (complément au cours)

On rappelle quelques définitions concernant la sémantique de la logique des propositions :

- une *interprétation* d'un ensemble de symboles (atomes) associe à chaque symbole la valeur vrai ou faux ;
- une interprétation est dite *modèle* d'une formule si elle rend vraie cette formule ;
- Etant données deux formules F1 et F2, on dit que F2 est *conséquence* de F1 si tout modèle de F1 est un modèle de F2 (« à chaque fois que F1 est vraie, F2 l'est aussi »).

On peut voir la base de connaissances comme une formule K obtenue en faisant la conjonction des faits et des règles. Montrer que le chaînage avant est *adéquat* et *complet* (« sound » and « complete » in English) par rapport à la conséquence logique, c'est-à-dire qu'il ne produit que des faits qui sont conséquences de K, et qu'il les produit tous :

- Adéquation : « pour tout atome A, si A est produit par l'algorithme alors A est conséquence de K »
- Complétude : « pour tout atome A, si A est conséquence de K alors A est produit par l'algorithme ».

Exercice 4. Règles avec négation (*dite « classique »*)

L'algorithme DPLL et la méthode de résolution (en largeur) sont des mécanismes *adéquats* et *complets* par rapport à la conséquence logique. Autrement dit, ils permettent de déterminer pour deux formules *quelconques* A et B de la logique des propositions si A est conséquence de B (en résolvant le problème d'insatisfiabilité de la forme clausale associée à $B \land \neg A$). Nous avons vu que le chaînage avant était adéquat et complet sur des clauses définies. **Le reste-t-il si on considère des règles plus complexes ?**

Considérons des règles avec **négation** : une règle a maintenant pour hypothèse une conjonction de *littéraux* et pour conclusion un *littéral* (rappelons qu'un littéral est un atome ou la négation d'un atome). On étend la notion d'application de règle : une règle est applicable si chacun des littéraux de son hypothèse est présent dans la base de faits, et l'appliquer consiste à ajouter sa conclusion à la base de faits. Réfléchir à partir des exemples suivants :

a) Règle : $A \rightarrow B$ Fait : $\neg B$
[Question : a-t-on $\neg A$?]

a') Règles : R1 : $A \rightarrow B$ **;** R2 : $\neg A \rightarrow O$ Fait : $\neg B$
[Question : a-t-on O ?]

b) Règles : R1 : $A \rightarrow B$ **;** R2 : $A \rightarrow C$ **;** R3 : $B \land C \rightarrow D$ Fait : $\neg D$
[Question : a-t-on $\neg A$?]

c) Règles : R1 : $A \rightarrow B$ **;** R2 : $\neg A \rightarrow B$ Pas de faits
[Question : a-t-on B ?]

Exercice 5 : Graphe ET-OU

A toute base de règles, on peut aussi associer un graphe "ET-OU" construit de la façon suivante :

- on a un sommet par atome (ou « symbole propositionnel »)
- on a aussi un sommet par règle (ces sommets sont appelés "sommets ET")
- pour chaque règle $R = A1 \land \ldots \land An \rightarrow B$, soit R le sommet ET associé ; on a les arcs (Ai,R), pour i de 1 à n, et l'arc (R,B).

On peut voir le chaînage avant comme un parcours de ce graphe. On marque les symboles qui sont aussi des faits, et on propage les marques à travers le graphe de la façon suivante : pour passer un noeud ET (ce qui correspond à déclencher la règle associer), il faut avoir marqué tous ses prédécesseurs ; on peut alors marquer son successeur.

Construire le graphe ET-OU de l'exemple "réunion d'amis" (rappelé ci-dessous) et effectuer le chaînage avant par parcours de ce graphe.

Réunion d'amis [1]

R1 : si Benoît et Djamel et Emma alors Félix
R2 : si Gaëlle et Djamel alors Amandine
R3 : si Cloé et Félix alors Amandine
R4 : si Benoît alors Xéna
R5 : si Xéna et Amandine alors Habiba
R6 : si Cloé alors Djamel
R7 : si Xéna et Cloé alors Amandine
R8 : si Xéna et Benoît alors Djamel

Exercice 8 : Chaînage arrière

En utilisant un algorithme de chaînage arrière sur la base de règles « réunion d'amis » :

- déterminer quelle suite d'applications de règles permettent de prouver que Habiba doit être invité (c'est-à-dire Q = Habiba) ;
- comprendre pourquoi le graphe de l'exercice précédent est appelé "graphe ET-OU".

Exercice 9 : Chaînage arrière (avec questions à l'environnement)

Soit la base de règles suivante :

R1 : si A alors B
R2 : si B alors D
R3 : si H alors A
R4 : Si G et E alors C
R5 : si E et K alors B
R6 : si D et E et K alors C
R7 : si G et K et F alors A
R8: si C alors G

Les faits H et K sont établis.

Question 1 : peut-on prouver C ? Essayer avec le chaînage arrière.

Question 2 : on suppose que les informations qui ne figurent en conclusion d'aucune règle sont "observables", c'est-à-dire que le système peut demander ces informations ("a-t-on X ?") à son environnement (capteurs, utilisateur, …) si elles ne figurent pas dans la base de faits et qu'il en a besoin pour atteindre un certain but. Dans l'exemple précédent : E, F, H et K sont observables.
Illustrer ce mécanisme avec C comme but.

---

[1] Adapté de E. Adam