

## Trois méthodes d'interrogation avec des règles Datalog

Nous reprenons ici un exercice fait en chaînage avant et ajoutons les méthodes de chaînage arrière et de réécriture de requête.

On considère la base de connaissances suivante :

- Règles (*intuitivement*, “sg” signifie “same generation”)

R1:  $\text{flat}(x1,y1) \rightarrow \text{sg}(x1,y1)$

R2 :  $\text{up}(x2,y2) \wedge \text{sg}(y2,z2) \wedge \text{up}(t2,z2) \rightarrow \text{sg}(x2,t2)$

- Faits (où a,b,c,d,e,f,g sont des constantes)

$\text{flat}(a,b) \text{ flat}(b,c) \text{ flat}(a,c) \quad \text{up}(d,a) \text{ up}(d,b) \text{ up}(e,c) \text{ up}(f,d) \text{ up}(g,e)$

ainsi que la requête booléenne :

$q() = \{ \text{sg}(x,y) , \text{up}(y,z), \text{flat}(z,c) \}$  où x, y et z sont des variables.

### 1. Chainage avant (effectué en TD)

1) **Saturez** la base de faits avec les règles, en procédant **en largeur** (cf. algorithme FC du cours). A chaque étape, on ne considère que les **nouveaux** homomorphismes. On dit qu’une application de règle est **utile** si elle produit un fait qui n’appartient pas à la base de faits courante.

Etape	Règle applicable	Homomorphisme	Fait produit	Application utile ?
n° étape	n° règle	...	...	oui/non
...	...	...	...	...

2)

- Comment reconnaît-on qu’un homomorphisme est *nouveau* ?
- On dit qu’un prédicat est *intentionnel* s’il apparaît au moins une fois en tête de règle : ici, *sg* est un prédicat intentionnel, et c’est le seul (ceux qui n’apparaissent pas en tête de règle sont dits *extensionnels*). L’ensemble de règles ci-dessus a une particularité : le corps de chaque règle contient au plus un atome avec un prédicat intentionnel. Un tel ensemble de règles est appelé *linéaire*. Comment exploiter le fait qu’un ensemble de règles soit linéaire pour ne calculer que les homomorphismes nouveaux à chaque étape de largeur ?

3)

La base de connaissances répond-elle positivement à q ? Justifiez votre réponse en vous basant sur le mécanisme de *chaînage avant*.

## 2. Chainage arrière

---

1) Montrer que la base de connaissances répond positivement à  $q()$  en utilisant le chaînage arrière.

Au lieu de prendre à chaque étape le premier atome de la requête courante, vous adopterez les priorités suivantes :

1. Les atomes ayant au moins une constante
  2. Les atomes avec un prédicat extensionnel
  3. Les atomes avec un prédicat qui apparaît aussi dans les faits.
- 2) Calculer l'ensemble des réponses à  $q(x,y,z)$  sur la base de connaissances, c'est-à-dire en considérant  $x,y$  et  $z$  comme des variables réponses. Ceci nécessite donc de mémoriser les substitutions des variables réponses au cours du calcul.

## 3. Réécriture de requête

---

On considère maintenant le mécanisme d'interrogation par réécriture de requête, qui décompose le processus en deux phases :

- a. Réécriture de la requête  $q$  (une CQ) en une union de requête conjonctive  $Q$ , en utilisant les règles, sans considérer les faits.
- b. Interrogation de la base de faits avec  $Q$  (si la base de faits est stockée dans une base de données relationnelle,  $Q$  est transformée en une requête SQL).

Cette méthode est-elle utilisable avec la requête  $q()$  de l'énoncé (ou sa variante  $q(x,y,z)$ ) ?