Représentation de connaissances

Interrogation de bases de connaissances

avec règles Datalog

- Chaînage avant (Forward Chaining , « matérialisation »)
- **Chaînage arrière classique** (Backward Chaining)
- Variante de BC par réécriture de requête

Marie-Laure Mugnier

EXAMPLE (FORWARD CHAINING)

R: play(x,y), movie(y) → movieActor(x)

« find all movie actors »

q(x) = movieActor(x)

F*

movie(m1)
movie(m2)
movie(m3)
movieActor(a)
movieActor(b)
play(a,m1)
play(a,m2)
play(c,m3)

movieActor(c)

Saturation

x = a
x = b
x = c

Query Answering

1. **Forward chaining**: compute F* (can be done offline)
2. Evaluate queries on F* (based on homomorphisms to F*)

Example (Backward Chaining - Classical)

R: play(x,y), movie(y) → movieActor(x)

q(x) = movieActor(x)

movie(m1)
movie(m2)
movie(m3)
movieActor(a)
movieActor(b)
play(a,m1)
play(a,m2)
play(c,m3)

Rewrite queries with the rules and the facts until obtaining empty sets of atoms
Compose substitutions to « trace » answers

- q unified with facts

x = a
x = b

- q unified with head(R):

Renaming of R's variables (to avoid confusion)
R: play(x',y'), movie(y') → movieActor(x')

x → x' unifier of q and head(R)

⇨ q1: play(x',y'), movie(y')

- part of q1 unified with fact play(c,m3)

x' → c, y' → m3

⇨ q2: movie(m3)

- q2 unified with fact movie(m3)

⇨ ø

x = c

x = a
x = b
x = c

UNIFIERS

R1: p(x1,y1), p(y1,z1) → gp(x1,z1)
R2: mo(x2,y2) → p(x2,y2)
R3: fa(x3,y3) →p(x3,y3)

A = gp(x,a)

A **unifier** *u* of atoms *A* and *B* is a substitution (of **variables**) such that *u(A) = u(B)*

A **most general unifier** (mgu) of *A* and *B* is a unifier *u* of *A* and *B* such that every other unifier of *A* and *B* can be written as *s* o *u* where *s* is a substitution

A = gp(x,a)

B = gp(x1,z1)

u1:
x → x1
z1 → a

u2:
x1 → x
z1 → a

mgu

Il peut y avoir plusieurs mgu de A et B,
mais ils sont « équivalents » :
on passe de l'un à l'autre par une substitution s qui ne fait que renommer bijectivement les variables

u2 = { x1 → x } o u1
u2 = {x → x1 } o u2

u3:
x → a
z1 → a
x1 → a

u3 = { x1 → a } o u1

Direct Rewriting

R1: p(x1,y1), p(y1,z1) → gp(x1,z1)
R2: mo(x2,y2) → p(x2,y2)
R3: fa(x3,y3) →p(x3,y3)

Basic step: computation of a **direct rewriting** of a query *q*

1. look for a mgu *u* of an atom A in *q* and an atom in the head of a rule *R*
2. the direct rewriting of *q* with *R* according to *u* is

$$rew(q,R,u) = u(body(R)) \cup u(q \setminus A)$$

q = gp(x,a) | head(R1) = gp(x1,z1) | u: x → x1, z1 → a

rew(q,R1,u) = p(x1,y1),p(y1,a)

q' = gp(x,a), p(x,b) | head(R1) = gp(x1,z1) | u: x → x1, z1 → a

rew(q',R1,u) = p(x1,y1), p(y1,a), p(x1,b)

Backward Chaining pour CQ booléenne

Algorithme BC(K,Q)

// Donnée : K = (F,R) et Q un but (liste d'atomes)

*// Remarque : on voit un fait **p(a1,…,ak)** comme une règle → **p(a1,…,ak)***

// Résultat : vrai ssi K répond positivement à Q

Début

Si Q = ø, retourner vrai

Pour toute règle R = B → H de **F** et **R**

telle que premier(Q) s'**unifie** avec H par un unificateur u

// ceci peut nécessiter de renommer les variables de R

// pour que les ensembles de variables de Q et R soient disjoints

Q' ← u(B) U u(reste(Q))

Si BC(K,Q') = vrai, retourner vrai

FinPour

Retourner faux

Fin

BACKWARD CHAINING POUR CQ BOOLÉENNE

Algorithme BC(K,Q)

// Donnée : K = (F,R) et Q un but (liste d'atomes)

*// Remarque : on voit un fait **p(a1,…,ak)** comme une règle → **p(a1,…,ak)***

// Résultat : vrai ssi K répond positivement à Q

Début

Si Q = ø, retourner vrai

Pour toute règle R = B → H de **F** et **R**

telle que premier(Q) s'**unifie** avec H par un unificateur u

// ceci peut nécessiter de renommer les variables de R
// pour que les ensembles de variables de Q et R soient disjoints

Q' ← u(B) U u(reste(Q))
Si BC(K,Q') = vrai, retourner vrai

FinPour

Retourner faux

Fin

BACKWARD CHAINING POUR CQ

// Q(x1,...,xk) est une CQ vue comme un ensemble d'atomes

// S = {(x1,x1,),...,(xk,xk)} initialement

Algorithme BC(K,Q,S)

// Donnée : K = (F,R) et Q une CQ, S substitution de {x1,...,xk}

// Résultat : Ensemble des {(x1,a1), ...(xk,ak) | (a1,...ak) est une réponse à Q dans K}

Début

Rep ← ø // sera le résultat

Pour toute règle R = B → H de **F** et **R** telle que premier(Q) s'**unifie** avec H par u

// si besoin, renommer les variables de R pour que variables(Q) ∩ variables(R) = ø

Q' ← u(B) U u(reste(Q))

S ← u o S *// mise à jour de (xi,y) par (xi,z) si (y,z) ∊ u*

Si Q' = ø, ajouter S à Rep

Sinon Rep ← Rep U BC(K,Q',S)

FinPour

Retourner Rep

Fin

Exemple BC

R1: p(x1,y1), p(y1,z1) → gp(x1,z1)
R2: mo(x2,y2) → p(x2,y2)
R3: fa(x3,y3) →p(x3,y3)

Q(x) = gp(x,a)

« Trouver les grand-parents de a »