A Veder (Bucket) algoritmus

Megoldások és kiterjesztések

- A p(X,Y) lekérdezés minden részcéljához minden megoldásnak tartalmaznia kell egy olyan nézetet (részcélt), amelynek kiterjesztése tartalmazza a p(x,y)-t.
- Egy nézethez (részcélhoz) tartozó „Veder” azon nézetek halmaza, amelyek „lefedik” a nézetet.
- Egy megoldás minden egyes vedrében legalább egy nézet kell, hogy legyen.

Ezenkívül …

- A nézetekben, a lekérdezésben, a megoldásokban és a kiterjesztésekben szereplő változók alapos vizsgálata szükséges a helyes megoldások megtalálásához szükséges kényszerek meghatározásához.

(Nem)Megkülönböztetett változók

- Egy változó, amely szerepel a lekérdezés fejében *megkülönböztetett* változó; egyébként nem megkülönböztetett.

```
p(X,Y) :- q(X,Z) & r(Z,Y)
```

Megkülönböztett   Nem megkülönböztetett

A kiterjesztések lokális változói

- Amikor kiterjesztünk egy részcélhoz tartozó nézetét egy megoldásnak, akkor a *nem megkülönböztetett* változók a nzetben lokálisak maradnak.
  - Egy lokális változó nem kell, hogy megjelenjen bárhol máshol a kiterjesztésben.
- A megoldás minden egyes változóját helyettesíteni kell egy nézetbeli megkülönböztetett változóval.

Példa

A nézet megkülönböztetett változóinak leképezése

```
v(X,Y)   :- p(X,Z) & q(Z,Y)
sol(U,V) :-      … & v(U,W) & …

exp(U,V) :- … & p(U,Z1) & q(Z1,W) …
```

Megkülönböztetettek a megoldásban; máshol is előfordulhatnak

Lokálisak a nézetben

Nem megkülönböztetettek

Látható változók

A lekérdezés változói

- Egy lekérdezés változója megosztott, ha több mint egyszer szerepel.
- Egy megkülönböztetett lekérdezés változó csak a vonatkozó megoldásbeli változóra képezhető le.
- Egy nem megkülönböztetett egyedi változó a lekérdezésben, bármely változóra leképezhető a kiterjesztésben.

Megosztott változók leképezése

- Két lehetőség létezik megosztott változókra:
  1. Leképezzük egy kiterjesztés egy lokális változójára.
  2. Kiterjesztjük egy látható változóra.
- Csak a második esetben lehet egy lekérdezés egyik nézetében szereplő megosztott változót leképezni több mint egy megoldásbeli részcélhoz.

Példa --- Leképezés

W duplikált
D meghatározható

```
v(X,Y)  :- p(X,Z) & q(Z,Y)
sol(U,V)  :-   …   v(U,W)   …       u(W,V)  …

exp(U,V)  :- p(U,Z1) & q(Z1,W) … r(W,T) …

que(A,B)  :-   …   q(C,D)   …   r(D,E)    …
```

A D megosztott változó leképzése a W látható változóra.
Másik nézetből egy másik példányt is
 leképezhetünk.

Példa

```
v(X,Y) :- p(X,Z) & q(Z,Y)
sol(U,V) :-       … & v(U,W) & …

exp(U,V) :- … & p(U,Z1) & q(Z1,W) …

que(U,V) :- … p(U,A) & q(A,D) …
```

Az A megosztott változó
minden előfordulása leképezhető a
lokális Z1 változóra

Vedrek

- Vedreket képzünk:
  1. Minden a lekérdezésben szereplő részcélhoz..
  2. Minden a lekérdezésben szereplő megosztott változóhoz.

Vedrek részcélokhoz

- Egy p(A,B) részcéhoz tartozó veder tartalma a következő:
  1. Egy *v nézet*.
  2. Egy *p* -részcél a *v törzsében*.
- A lekérdezés p(A,B) részcéljának és a a cél p(X,Y) nézetbeli elemnek feltételeknek kell megfelelnie.

Részcélok vedrei --- (2)

1. p(A,B) leképezhető kell, hogy legyen p(X,Y). Például ha A=B, akkor X=Y.
2. Ha mondjuk A megkülönböztetett a lekérdezésben, akkor X megkülönböztetett a nézetben.

.

Vedrek elosztott változókhoz

- Elosztott változók vedrei a következőt tartalmazzák:
  1. Egy *v nézet*
  2. A *v* –beli megoldás S elemeinek egy halmaza, amelyre a változók leképezése elvégezhető.
- A lekérdezésben megkülönböztetett változó megkülönböztetettre kell hogy legyen leképezve.

Example

```
v(X,Y) :- p(X,Z) & p(Z,Y)
w(U,V) :- p(U,S) & p(S,T) & p(T,V)
q(A,B) :- p(A,C) & p(C,D) & p(D,E)
          & p(E,F) & p(F,G) & p(G,B)
```

- v = "grandparent"; w = "great-grandparent"; query q = "sixth-generation ancestors."

Egy példa

- Családi kapcsolat

Globális nézet: szülő kapcsolat
p(X,Y)

Example --- p(A,C)

```
v(X,Y) :- p(X,Z) & p(Z,Y)
w(U,V) :- p(U,S) & p(S,T) & p(T,V)
q(A,B) :- p(A,C) & p(C,D) & p(D,E)
          & p(E,F) & p(F,G) & p(G,B)
```

- The bucket for p(A,C) is empty.
  - *A* is distinguished; *C* is shared.
  - No view subgoal has distinguished variables in both positions.

Example --- p(C,D)

```
v(X,Y) :- p(X,Z) & p(Z,Y)
w(U,V) :- p(U,S) & p(S,T) & p(T,V)
q(A,B) :- p(A,C) & p(C,D) & p(D,E)
          & p(E,F) & p(F,G) & p(G,B)
```

- The bucket for p(C,D) is empty.
  - Both *C* and *D* are shared.
  - No view subgoal has distinguished variables in both positions.
- Likewise, all subgoals of *q* have empty buckets.

Example --- Shared Variable *C*

```
v(X,Y) :- p(X,Z) & p(Z,Y)
w(U,V) :- p(U,S) & p(S,T) & p(T,V)
q(A,B) :- p(A,C) & p(C,D) & p(D,E)
          & p(E,F) & p(F,G) & p(G,B)
```

- The bucket for *C* :
  1. {p(X,Z), p(Z,Y)} from *v*.
     - Important: *X* is distinguished (since *A* maps to *X*).
  2. {p(U,S), p(S,T)} from *w*.
     - Important: *U* is distinguished (since *A* maps to *U*).

Shared Variable *C* --- Continued

```
v(X,Y) :- p(X,Z) & p(Z,Y)
w(U,V) :- p(U,S) & p(S,T) & p(T,V)
q(A,B) :- p(A,C) & p(C,D) & p(D,E)
          & p(E,F) & p(F,G) & p(G,B)
```

- The bucket for *C* does not contain {p(S,T), p(T,V)} from *w*.
  - Because distinguished variable *A* of the query would have to map to *S*, which is local in the view definition.

Example --- Shared Variable *D*

```
v(X,Y) :- p(X,Z) & p(Z,Y)
w(U,V) :- p(U,S) & p(S,T) & p(T,V)
q(A,B) :- p(A,C) & p(C,D) & p(D,E)
              & p(E,F) & p(F,G) & p(G,B)
```

- The bucket for *D* :
  1. {p(X,Z), p(Z,Y)} from *v*.
  2. {p(U,S), p(S,T)} and {p(S,T), p(T,V)} from *w*.
     - Either is OK, since neither *C* nor *E* is distinguished.
- *E*, *F* like *D* ; *G* like *A*.

Example --- Continued

- Each of the six query subgoals must be covered by at least one member of a bucket.
- Since the subgoals themselves have empty buckets, we must group them according to their shared variables and cover them, in groups, from the buckets for the variables.

Example --- Continued

- One possibility: use the members from *v* in the buckets for *C*, *E*, and *G*.
- Since shared variables D and F map to distinguished variables of the view definition, we can use *v* three times in the solution, and equate the corresponding variables.

First Solution

```
v(X,Y):- p(X,Z) & p(Z,Y)
w(U,V):- p(U,S) & p(S,T) & p(T,V)
q(A,B):- p(A,C) & p(C,D) & p(D,E)
         & p(E,F) & p(F,G) & p(G,B)
s(A,B):- v(A,J) & v(J,K) & v(K,B)
e(A,B):- p(A,Z1) & p(Z1,J) & p(J,Z2)
         & p(Z2,K) & p(K,Z3) & p(Z3,B)
```

Example --- Continued

- Another possibility is to use one copy of w to cover the first three query subgoals and another copy of w to cover the last three.
- The first copy covers shared variables *C* and *D*; the second covers *F* and *G*.
- Shared variable *E* maps to distinguished variables of w.

Second Solution

```
v(X,Y) :- p(X,Z) & p(Z,Y)
w(U,V) :- p(U,S) & p(S,T) & p(T,V)
q(A,B) :- p(A,C) & p(C,D) & p(D,E)&
              p(E,F) & p(F,G) & p(G,B)
s(A,B) :- w(A,J) & w(J,B)
e(A,B) :- p(A,S1) & p(S1,T1) & p(T1,J) &
              p(J,S2) & p(S2,T2) & p(T2,B)
```

Why There Are No More Solutions

- For instance, we cannot use one *v* subgoal v(A,J) in the solution to cover shared variable *C* and another v(K,L) to cover *D*.
- v(A,J) expands to p(A,Z1) & p(Z1,J), forcing *D* to map to *J*.
- But v(K,L) expands to p(K,Z2) & p(Z2,L), forcing *D* to map to Z2.

Látható változók

- Variables of the expansion that have substituted for distinguished variables of a view.
- These are the only variables that may appear in subgoals belonging to the expansion of two different solution subgoals.