Université de Lorraine

Faculté de Science et Technologie

Master 1 Informatique

Projet de LMC

Algorithme d'unification

Martelli Montanari

Auteurs :
Aurélien Thirion
Nicolas Blin

10 décembre 2016

Introduction

Nous avons voulu, dans ce projet, essayer d'être le plus clair et simple possible en disposant dans différents fichiers le code source de notre programme. Ainsi dans le code principal, il y a le minimum de ligne pour comprendre réellement de processus de l'unification de Martelli-Montanari. Notre code principal se situe dans **main.pl**, des prédicats sont appelés lors du processus dans les fichiers suivants : **operateur.pl** , **predicatsRelais.pl** et **reglesTest.pl**.

operateur.pl
Contient l'opérateur **?=** et le code pour le echo contenu dans le mail.

predicatsRelais.pl
Fonction de décomposition d'une équation E pour obtenir la partie gauche de l'opérateur **?=** et la partie droite.

reglesTest.pl
Toutes les conditions nécessaire pour appliquer la réduction (prédicat reduit(R,E,P,Q)).

Question 1

Mise en place

```
:-
    [operateurs],
    [predicatsRelais],
    [reglesTest].
```

Listing 1 – Dans le fichier *main.pl*

Un exemple du prédicat regle(E,R) dans le code :

```
regle(E,rename)
```

Listing 2 – Dans le fichier *main.pl*

Qui va appeler le prédicat dans le fichier **reglesTest.pl** :

```
regle(E, rename):-
    splitEquation(E,X,T),
    var(T),
    var(X).
```

Listing 3 – Dans le fichier *reglesTest.pl*

Qui va appeler le prédicat dans le fichier **predicatsRelais.pl** :

```
splitEquation(E,X,T):-
    arg(1,E,L),
    arg(2,E,R),
    X = L,
    T = R.
```

Listing 4 – Dans le fichier *predicatsRelais.pl*

Code source

Dans le main

```
occur_check(V,T):-
    compound(T),
    var(V),
    contains_var(V,T).

unif(P,S) :-
    clr_echo,
    unifie(P,S).

trace_unif(P,S) :-
    set_echo,
    (unifie(P,S),
     echo("Yes"),
     !;
     echo("No")).

unifie([], _) :- !.
unifie([]) :- !.


unifie(P):-
    unifie(P, regle),
    !.

unifie(P, regle):- unifie(P, rename).
unifie(P, regle):- unifie(P, simplify).
unifie(P, regle):- unifie(P, expand).
unifie(P, regle):- unifie(P, check).
unifie(P, regle):- unifie(P, orient).
unifie(P, regle):- unifie(P, decompose).
unifie(P, regle):- unifie(P, clash).


unifie(P, rename) :-
    P = [E |_],
    regle(E, rename),
    reduit(rename, E, P, Q),
    unifie(Q, regle),!.


unifie(P, simplify):-
    P = [E |_],
    regle(E, simplify),
    reduit(simplify, E, P, Q),
    unifie(Q, regle),!.

unifie(P, expand):-
    P = [E |_],
    regle(E, expand),
    reduit(expand, E, P, Q),
    unifie(Q, regle),!.

```

```
53  unifie(P, check):-
54      P = [E |_],
55      regle(E, check),
56      reduit(check, E, P, Q),
57      unifie(Q, regle),!.
58
59  unifie(P, orient):-
60      P = [E |_],
61      regle(E, orient),
62      reduit(orient, E, P, Q),
63      unifie(Q, regle),!.
64
65  unifie(P, decompose):-
66      P = [E |_],
67      regle(E, decompose),
68      reduit(decompose, E, P, Q),
69      unifie(Q, regle),!.
70
71  unifie(P, clash):-
72      P = [E |_],
73      regle(E, clash),
74      reduit(clash, E, P, Q),
75      unifie(Q, regle),!.
76
77
78  reduit(decompose, E, P, Q):-
79      splitEquation(E,X,T),
80      functor(X,_,ArityX),
81      functor(T,_,_),
82      P = [_|Tail],
83      repet(X,T,ArityX,Tail,Q),
84      echo("decompose: "),echo(Q),nl.
85
86  repet(_,_,0,T,Q):- Q = T, !.
87  repet(X,T,N,Tail,Q) :-
88      N > 0,
89      arg(N,X,ValX),
90      arg(N,T,ValT),
91      Var = [ValX?=ValT|Tail],
92      N1 is N - 1,
93      repet(X,T,N1,Var,Q).
94
95  reduit(rename, E, P, Q):-
96      splitEquation(E,X,T),
97      X = T,
98      P = [_|Q].
99
100 reduit(simplify, E, P, Q):-
101     splitEquation(E,X,T),
102     X = T,
103     P = [_|Q].
104
105 reduit(expand, E, P, Q):-
106     splitEquation(E,X,T),
107     X = T,
108     P = [_|Q].
109
110 reduit(check, _, _, _):-
111     fail,
112     !.
113
114 reduit(orient, E, P, Q):-
115     splitEquation(E,X,T),
116     P = [_|Tail],
117     Q = [T ?= X | Tail].
118
```

```
119
120 reduit(clash, _, _, _):-
121     fail,
122     !.
```

Listing 5 – *main.pl*