Lógica informática (2011–12)

Tema 17: Implementación en Prolog de la resolución

José A. Alonso Jiménez
Andrés Cordón Franco
María J. Hidalgo Doblado

Grupo de Lógica Computacional
Departamento de Ciencias de la Computación e I.A.
Universidad de Sevilla

Tema 17: Implementación en Prolog de la resolución

1. Regla de resolución proposicional

2. Demostraciones por resolución

Tema 17: Implementación en Prolog de la resolución

Regla de resolución proposicional

- `complementario(+L1,-L2)` se verifica si `L2` es el complementario del literal `L1`.

```
complementario(-A, A) :- !.
complementario(A, -A).
```

- `resolvente(+C1,+C2,-C3)` se verifica si `C3` es una resolvente de las cláusulas `C1` y `C2`. Por ejemplo,

```
?- resolvente([q, -p],[p, -q],C).   =>  C = [p, -p] ; C = [q, -q] ; No
?- resolvente([q, -p], [q, r], C).  =>  No
?- resolvente([p],[-p],C).          =>  C = [] ; No
```

```
resolvente(C1,C2,C) :-
   member(L1,C1),
   complementario(L1,L2),
   member(L2,C2),
   delete(C1, L1, C1P),
   delete(C2, L2, C2P),
   append(C1P, C2P, C3),
   sort(C3,C).
```

Tema 17: Implementación en Prolog de la resolución

1. Regla de resolución proposicional

2. Demostraciones por resolución

Demostraciones por resolución

- `refutación(+S,-R)` se verifica si `R` es una refutación por resolución del conjunto de cláusulas `S`. Por ejemplo,

```
?- refutación([[-p,-q],[p,q],[-p,q],[-q,p]],R).
R = [[p,-q],[q,-p],[p,q],[-p,-q],
     [q,-q],[p,-p],[-p],[q],[p],[]]
?- refutación([[p,q],[-p,q],[-q,p]],R).
No
```

- Ejemplo con traza de resolventes:

```
?- refutación([[-p,-q],[p,q],[-p,q],[-q,p]],R).
[q, -q] resolvente de [-p, -q] y [p, q]
[p, -p] resolvente de [-p, -q] y [p, q]
[-p]    resolvente de [-p, -q] y [q, -p]
[q]     resolvente de [-p] y [p, q]
[p]     resolvente de [q] y [p, -q]
[]      resolvente de [p] y [-p]

R = [[p,-q],[q,-p],[p,q],[-p,-q],
     [q,-q],[p,-p],[-p],[q],[p],[]]
```

Búsqueda elemental de refutación

- Def. de `refutación`:

```
refutación(S,R) :-
   maplist(sort,S,S1),
   refutación_aux(S1,R).

refutación_aux(S,R) :-
   member([],S), !,
   reverse(S,R).
refutación_aux(S,R) :-
   member(C1,S),
   member(C2,S),
   resolvente(C1,C2,C),
   \+ member(C, S),
   % format('~N~w resolvente de ~w y ~w~n', [C,C1,C2]),
   refutación_aux([C|S],R).
```

Bibliografía

- Alonso, J.A. y Borrego, J. *Deducción automática (Vol. 1: Construcción lógica de sistemas lógicos)* (Ed. Kronos, 2002)
  - Cap. 4.3: Resolución.
- Ben–Ari, M. *Mathematical Logic for Computer Science (2nd ed.)* (Springer, 2001)
- Chang, C.–L. y Lee, R.C.–T. *Symbolic Logic and Mechanical Theorem Proving* (Academic Press, 1973)
- Fitting, M. *First-Order Logic and Automated Theorem Proving (2nd ed.)* (Springer, 1995)
- Nerode,A. y Shore, R.A. *Logic for Applications* (Springer, 1997)
- Schöning, U. *Logic for Computer Scientists* (Birkäuser, 1989)