Examen **“Programmation en Logique”** n^o 2
ESSI2 — 30 Mai 1995

Michel Rueher

Documents autorisés: *supports (et notes) de cours de TD*

1 Utilisation de Prolog

Une banque décide d’offrir à sa clientèle la possibilité d’effectuer via le minitel la plupart des opérations proposées dans l’un quelconque de ses guichets.

Vous êtes chargé de la pré-étude, de la conception et de la réalisation des traitements informatiques. Expliquez dans quelles phases de ce travail, l’utilisation du langage Prolog serait pertinente.

P.S. Vous êtes libre de définir les hypothèses de votre choix pour préciser le contexte ou la nature des opérations “automatisées”, mais vous devez expliciter les hypothèses utilisées !

2 DCGs

- Ecrivez un parseur DCG pour la grammaire suivante:
  ```
  <atome> ::== 'a' | 'b' | 'c'
  <sexpr> ::== <atome> | '(' <sexpr>+ ')'
  ```
- Donnez la traduction en Prolog du parseur DCG.
- Etendez votre parseur DCG pour construire un arbre syntaxique. Il n’est pas necessaire de donner la traduction en Prolog pour le parseur étendu.
- Quel est l’arbre donné par le parseur étendu pour l’entrée `((a b) c)`, i.e., la liste de tokens `['(','(',a,b,')',c,')']` ?

3 Explication de programme

Les morceaux de code ci-dessous sont extraits d’une des solutions proposées pour le TD4-5.

1. Expliquez la différence entre les prédicats *valeur_semantique*/2 n^o (1) et n^o (2). On distinguera la sémantique opérationnelle de la sémantique dénotationnelle.

   (1)
   ```
   valeur_semantique(et(P,Q),V) :-
               valeur_semantique(P,V1),
               valeur_semantique(Q,V2),
               et(V1,V2,V).
   ```
   (2)
   ```
   valeur_semantique(et(P,Q),vrai) :-
               valeur_semantique(P,vrai),
               valeur_semantique(Q,vrai),!.
   valeur_semantique(et(P,Q),faux) :- !.
   ```

2. Donner une définition précise et **concise** du travail effectué par le prédicat *subst*/3 ci dessous. Justifiez le traitement effectué dans les clauses A et B.

```
subst(existe(X,P),[X,I],existe(X,P)).                    (A)
subst(existe(Y,P),[X,I],existe(Y,P1)) :-
               X ≡ Y,
               subst(P,[X,I],P1).
subst(tout(X,P),[X,I],tout(X,P)).                        (B)
subst(tout(Y,P),[X,I],tout(Y,P1)) :-
               X ≡ Y,
               subst(P,[X,I],P1).
subst(et(U,V),S,et(U1,V1)):-
               subst(U,S,U1),
               subst(V,S,V1).
subst(ou(U,V),S,ou(U1,V1)):-
               subst(U,S,U1),
               subst(V,S,V1).
subst(non(U),S,non(U1)):-
               subst(U,S,U1).
subst(L,S,L1):-
               L =.. [P|U],
               symbole_relationnel(P,A),
               subst_terme(A,U,S,U1),
               L1 =.. [P|U1].
subst_terme(1,[T],S,[T1]):-
               terme(T),
               subst1(T,S,T1).
subst_terme(N,[T|Ts],S,[T1|Ts1]):-
               N > 1,
               terme(T),
               subst1(T,S,T1),
               N1 is N - 1,
               subst_terme(N1,Ts,S,Ts1).
```