# Examen Prolog — ESSI2

**( Documents autorisés:** *tous***)**

Michel Rueher

Mai 1999

1. (2 points) Donnez le résultat de la requète:

```
> [A,_,C|D]=[[a,b,c],d,e,f,g]
```

2. (3 points) Soit le prédicat défini par les clauses :

```
appartient(Sous_liste,Sur_liste):-
        conc(L1,Sous_liste,L2),
        conc(L2,L3,Sur_liste).
```

Quel est le comportement de ce prédicat pour un appel `appartient(S,L)` où `S` est une liste complètement instanciée et `L` une liste dont la taille est connue.

3. (3 points) Ecrire un prédicat `frere(F,X)` qui est vrai si F est le frere de X.
Vous devez définir les prédicats que vous utilisez.

4. (3 points) Programme "tracassin" (cf. annexe). Quelle est la version la plus efficace? Expliquer pourquoi.

5. (3 points) Ecrire un prédicat `palindrome(L)` qui vérifie que la liste L est un palindrome (version "naive" et version utilisant les DCG). On supposera que L est complètement instanciée (cf. exemples ci-dessous).

```
> palindrome([e,l,l,e]).        ou   > palindrome([e,l,l,e],[])
yes                                  yes
> palindrome([e,l,l,e,s]).      ou   > palindrome([e,l,l,e,s],[])
no                                   no
> palindrome([l,a,v,a,l]).      ou   >palindrome([l,a,v,a,l],[]).
yes                                  yes
```

6. Soit le langage formé des mots de la forme $a^n b^n$.
   (a) (2 points) Utilisez les `DCG` pour analyser les mots de ce langage.
   (b) (2 points) Donner le programme `prolog` généré à partir de la `DCG` par l'interprète `Sicstus`;
   (c) (2 points) Donner la `DCG` qui permet d'analyser les mots de ce langage et de calculer $n$.



Exemple:

```
> anbn(N,[a,a,a,b,b,b],[]).
> N=3
> anbn(N,[a,a,a,b,b],[]).
> no
```

Annexe

```
                    /* Tracassin V1 */
tr(P) :-
        etat_i(E1),  etat_f(E2),
        transfo(E1,E2,[E1],P).

transfo(Ef,Ef,_,[]).
transfo(Ec,Ef,LE,[Op|A]):-
        ope(Ec,Es,Op),
        hors(Es,LE),
        transfo(Es,Ef,[Ec|LE],A).

ope(E1,E2,r1) :- subst([x,b],[b,x],E1,E2)),!.
ope(E1,E2,r2) :- subst([b,o],[o,b],E1,E2)),!.
ope(E1,E2,r3) :- subst([x,o,b],[b,o,x],E1,E2)),!.
ope(E1,E2,r4) :- subst([b,x,o],[o,x,b],E1,E2)),!.

subst(C1,C2,L1,L2) :-
        conc(L0,R,L1),   conc(X,C1,L0),
        conc(X,C2,Li),   conc(Li,R,L2).

                    /* Tracassin V2 */
tr(P) :-
        etat_i(E1),  etat_f(E2),
        transfo(E1,E2,[E1],P).

transfo(Ef,Ef,_,[]).
transfo(Ec,Ef,LE,[Op|A]):-
        ope(Ec,Es,Op),
        hors(Es,LE),
        transfo(Es,Ef,[Ec|LE],A).
ope(E1,E2,r1) :- subst([x,b|L],[b,x|L],E1,E2).
ope(E1,E2,r2) :- subst([b,o|L],[o,b|L],E1,E2).
ope(E1,E2,r3) :- subst([x,o,b|L],[b,o,x|L],E1,E2).
ope(E1,E2,r4) :- subst([b,x,o|L],[o,x,b|L],E1,E2).

subst(L1,L2,L1,L2).
subst(C1,C2,[_|L1],[_|L2]) :-
          subst(C1,C2,L1,L2).
```