

TP 1 - Programmation logique - Prolog

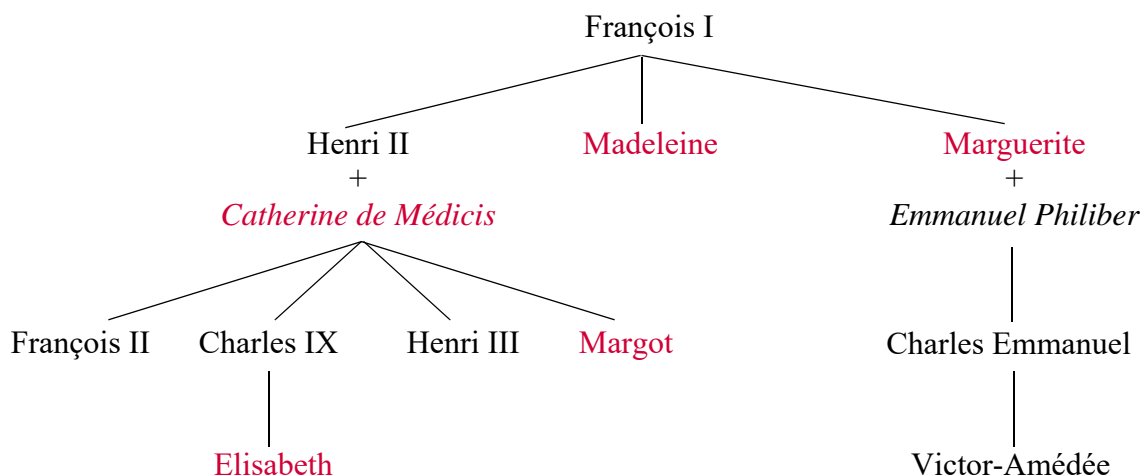
Arbre généalogique et manipulation de listes

Exercice 1

Les prédicats définis seront **testés au fur et à mesure** de leur écriture par des questions/réponses. L'ensemble des tests effectués sera reporté dans le compte-rendu de TP.

De plus, toutes les questions apparaissant explicitement dans l'énoncé seront posées au programme Prolog et les réponses seront commentées. On justifiera notamment l'ordre d'apparition des solutions trouvées par le programme et on expliquera les éventuelles occurrences multiples d'une même solution.

On considère la dynastie de François Ier :



- 1.1 Définir les prédicats homme, femme et parent. Le prédicat homme (resp. femme) permet de retrouver tous les hommes (resp. toutes les femmes) de la figure ci-dessus. Le prédicat parent(X,Y) est vrai si X est parent de Y.
- 1.2 A partir des prédicats ci-dessus, définir les prédicats pere(X,Y) vrai si X est père de Y et mere(X,Y) vrai si X est mère de Y. Qui est le père d'Henri II ?
- 1.3 Définir le prédicat fille(X,Y) vrai si X est fille de Y.
- 1.4 Définir le prédicat procreer(X,Y) vrai si l'homme X et la femme Y ont procréé ensemble. Quels sont les couples qui ont des enfants ?
- 1.5 Définir le prédicat gdpere(X,Y) vrai si X est un grand parent de Y.
- 1.6 Définir les prédicats gdpere(X,Y) et gdmere(X,Y).

- 1.7 Définir le prédicat **ancetre**(X,Y) vrai si X est un ancêtre de Y. Quels sont les ancêtres de la reine Margot ? Quelle est la descendance de Catherine de Médicis ?
- 1.8 Définir le prédicat **frere**(X,Y) vrai si X est un frère de Y (X et Y ont au moins un parent en commun). Définir de manière similaire le prédicat **soeur**(X,Y). De qui Henri II est-il le frère ? De qui Charles IX est-il le frère ? Quelles sont les soeurs de Charles IX ?
- 1.9 Définir le prédicat **oncle**(X,Y) vrai si X est un oncle de Y. Définir également le prédicat **tante**(X,Y). Quels sont les couples oncles/nièces et tantes/neveux ?
-

Exercice 2

- 2.1 Ecrivez le predicat **doubler**(L, LD) qui réussit si LD est la liste obtenue en doublant chacun des éléments de la liste L.
Exemple: ?- doubler([a,b,c], LD).
LD = [a,a,b,b,c,c].
- 2.2 Ecrivez le prédicat **const_liste**(N, X, L) qui réussit si la liste L est composée de N fois l'élément X.
Exemple: ?- const_liste(3, a, L).
L = [a,a,a].
- 2.3 Ecrivez le predicat **developper**(N, L, LD) qui réussit si LD est la liste obtenue en répétant N fois chaque element de la liste L.
Exemple: ?- developper(3, [a,b,a], LD).
LD = [a,a,a,b,b,b,a,a,a].
-

Exercice 3

- 3.1 Définir le prédicat **traduit**(L, M) qui traduit une liste de nombres vers les mots correspondants. Par exemple traduit([1,2,5], M) devrait donner M = [un, deux, cinq].
- 3.2 Définir les prédicats **plus_symb**(M1, M2, R) et **fois_symb**(M1, M2, R) qui, étant donnés les mots M1 et M2, réalisent respectivement leur addition et multiplication numérique. Le résultat est l'entier R.
- 3.3 Quelles sont les réponses à la question **plus_symb**(M1, M2, 4) ?
- 3.4 Modifier les prédicats précédents de façon à ne produire que les couples solutions (M1, M2) où le chiffre associé à M1 est inférieur ou égal à celui associé à M2. Ainsi, la réponse attendue à la question **plus_symb_bis**(M1, M2, 4) est :
- M1 = zero, M2 = quatre;
M1 = un, M2 = trois;
M1 = deux, M2 = deux;
- Quelle est la réponse à la question ?- **plus_symb_bis**(trois, X, 4).