

L3 - Programmation logique - TP1

1 Généalogie

On considère les liens de parenté suivants :

- `john` est le fils de `annie` et `ipfix`,
- `nolife` est le fils de `baldelix` et `felicie`,
- `annie` est la fille de `massmedia` et `osterlix`,
- `ipfix` et `baldelix` sont les enfants de `meacupla` et `ursul`,
- `felicie` est la fille de `osterlix` et `ida` (`osterlix` ayant trompé `massmedia` à la suite d'une terrible bataille...),
- `ursul` et `ida` sont les enfants de `cosinus` et `isabeau`.

1.1

Exprimez tous ces liens de parentés avec des faits prolog spécifiés à l'aide de trois prédicats :

- `parentDe/2`, où `parentDe(X,Y)` signifie que `X` est le père ou la mère de `Y`,
- `femme/1`, où `femme(X)` signifie que `X` est une femme,
- `homme/1`, où `homme(X)` signifie que `X` est un homme.

1.2

Vérifiez le fonctionnement du prédicat défini par les clauses suivantes :

```
 filsDe(P,E) :- parentDe(P,E),homme(E).
```

en lançant les buts suivants :

- `filsDe(X,Y)`.
- `filsDe(ursul,Y)`.
- `filsDe(X,ursul)`.
- `filsDe(nolife,john)`.

1.3

Spécifiez le prédicat `memeFratrerie/2` tel que `memeFratrerie(A,B)` soit vrai si et seulement si `A` et `B` ont le même père ou la même mère.

Expliquez pourquoi le but `memeFratrerie(A,baldelix)` donne deux fois la réponse `ipfix` et deux fois la réponse `baldelix`.

1.4

Spécifiez le prédicat `grandMere/2` tel que `grandMere(G,H)` soit vrai si et seulement si `G` est une grand-mère de `H`. Utilisez ce prédicat pour rechercher les grand mères de `john` et les petits-enfants de `massmedia`.

1.5

Vérifiez le fonctionnement du prédicat défini par les clauses suivantes :

```
 ancetre(A,D) :- parentDe(A,D).
 ancetre(A,D) :- parentDe(A,X),ancetre(X,D).
```

en recherchant les ancêtres de `john`.

Tentez de prévoir le résultat de la recherche des ancêtres de `john` si le prédicat `ancetre` est défini de la manière suivante :

```
 ancetre(A,D) :- parentDe(A,D).
 ancetre(A,D) :- ancetre(X,D),parentDe(A,X).
```

puis vérifiez votre réponse.

1.6

Spécifiez le prédicat `cousinGermain/2` tel que `cousinGermain(X,Y)` soit vrai si et seulement si l'un des parents de `X` est frère ou soeur d'un des parents de `Y`. Vérifiez cette spécification avec des buts pertinents.

1.7

Spécifiez le prédicat `LienParentee/2` tel que `LienParentee(X,Y)` soit vrai si et seulement si `X` et `Y` ont un ancêtre commun. Vérifiez cette spécification avec des buts pertinents.

1.8

En généalogie, on appelle `implex` une personne qui est ancêtre à la fois de la mère et du père d'un même individu. Spécifiez le prédicat `implex/1` tel que `implex(X)` soit vrai si et seulement si `X` est un `implex`. Vérifiez cette spécification en recherchant s'il existe un `implex` dans la base généalogique d'exemple.

2 Calcul numérique

2.1

Essayez les buts suivants pour vérifier les modes d'utilisation des opérateurs arithmétiques et de comparaison.

```
1 >= 2.  
1 >= 0.  
X >= 2.  
X is 1 + 2.  
X is 1 + Y.  
1 + 2 is 3.
```

2.2

Spécifiez un prédicat `fact/2` tel que `fact(N,R)` soit vrai si et seulement si $R = N!$. Utilisez ce prédicat pour calculer $7!$. Pourquoi ne peut-on pas l'utiliser pour déterminer l'entier dont la factorielle est 5040?

3 Listes

Les spécifications demandées doivent être réalisées si possible sans l'utilisation de la coupure (!). Chaque prédicat spécifié devra être testé avec *tous les modes d'utilisation possibles*.

3.1

Spécifiez le prédicat `dernier/2` tel que `dernier(L,D)` soit vrai si et seulement si `D` est le dernier élément de la liste `L`.

3.2

Spécifiez le prédicat `longueur/2` tel que `longueur(L,N)` soit vrai si et seulement si `N` est un entier égal à la longueur de la liste `L`.

3.3

Spécifiez le prédicat `concat/3` tel que `concat(A,B,R)` soit vrai si et seulement si `R` est la concaténation des listes `A` et `B`.

3.4

Spécifiez le prédicat `appartient/2` tel que `appartient(X,L)` soit vrai si et seulement si `X` est un élément de la liste `L`.

3.5

Spécifiez le prédicat `delete/3` tel que `delete(X,L,R)` soit vrai si et seulement si `R` est la liste obtenue en retirant de `L` toutes les occurrences de `X`.

4 Recherche de chemins dans un graphe

Cet exercice est optionnel. A faire uniquement si vous avez parfaitement maîtrisé les exercices précédents. On représente un graphe à l'aide des faits suivants :

```
arc(a,b).  
arc(b,c).  
arc(a,d).  
arc(d,b).  
arc(c,d).
```

On dit qu'il existe un chemin de longueur 0 entre un sommet `X` et lui même. Pour tout $N > 0$, on dit qu'il existe un chemin de longueur `N` entre un sommet `X` et un sommet `Y` si et seulement si il existe un arc reliant `X` à `T` et un chemin de longueur `N-1` entre `T` et `Y`.

En vous basant sur cette définition, spécifiez un prédicat `chemin/3` tel que `chemin(X,Y,N)` soit vrai si et seulement si il existe un chemin de longueur `N` entre `X` et `Y`.

Utilisez ce prédicat pour rechercher tous les chemins de longueur 1, 2, 3, 4 et 5 partant du sommet `a`.