Université de Reims
Champagne-Ardenne
U.F.R. de Sciences
Exactes et Naturelles

Licence 3 INFO / PASSERELLE
INFO0502
2020/2021
J.-C. Boisson

Interrogation de TP (1h)

Quand vous aurez terminé ou que le temps imparti sera écoulé, vous rendrez tous les fichiers produits suivant les consignes que vous avez reçu.

Exercice 1 (économe ou énergivore)
Ouvrez un fichier que vous nommerez `Conso.prolog` et mettez les faits suivants à l'intérieur :

```
econome('A').
econome('B').
econome('C').

energivore('E').
energivore('F').
energivore('G').
```

1) Ajouter la règle `economeEnergivore` qui est valide si le premier argument valide un fait `econome` et le second un fait `energivore`.

2) Ajouter la règle `duoEnergivore` qui est valide si les deux arguments sont différents et valident chacun un fait `energivore`.

Exercice 2 (Comptons)
Ouvrez un fichier que vous nommerez `compte.prolog` et incluez les règles suivantes :

1) Une règle `estPremier(N)` qui est valide si N est un nombre premier. On ne tiendra pas compte du cas où N vaut 0. Pour rappel, un nombre est considéré comme premier s'il est uniquement divisible par lui-même et par 1.

2) une règle `baseTrinaire(N)` qui est valide si N est un nombre et l'affiche en base 3. On rappelle que la division entière se fait avec l'opérateur `div` et le modulo par l'opérateur `mod`. Enfin, le cas où N vaut 0 n'amènera à aucun affichage.

Exercice 3 (Vive les listes)
Ouvrez un fichier que vous nommerez `liste.prolog` et incluez les règles suivantes :

1) La règle `identiquesDecal1D(L1,L2)` qui indique si la liste $L2$ est identique à la liste $L1$ à un décalage à droite près. Cela signifie juste que $L2$ est la liste $L1$ avec une valeur en plus au début (donc si on ne tient pas compte de la première valeur de $L2$, les listes doivent être identiques).

2) La règle `estNombre(I,L)` qui est valide si le I-ème élément de L est un nombre.

3) Insérer le prédicat suivant dans votre fichier et testez-le :

```
truc([], [], []).
truc([X|L], [X|LA], LN) :- not(number(X)), truc(L, LA, LN).
truc([X|L], LA, [X|LN]) :- number(X), truc(L, LA, LN).
```

Une fois testé, donnez un nom plus cohérent à ce prédicat et indiquez ce qu'il fait par un commentaire (lignes commençant par le symbole %).