Les contraintes sur les nombres réelles/rationnelles

Sur la page Web d'ECLIPSE CLP (`eclipseclp.org`) sous l'onglet `Contributions` vous trouvez l'archive ainsi que le mode d'emploi du module `clp(Q,R)` que nous utiliserons. Pour installer le module, il faut extraire l'archive `clpqr.tgz` dans le répertoire principal contenant votre installation de ECLIPSE CLP. On peut utiliser deux bibliothèques de contraintes, une pour les réelles (`use_module(library(clpr)).` ou `lib(clpr).`) et une pour les rationnelles (`(lib(clpq).`). Dans ce TD/TP nous allons travailler avec `clpr` ou `clpq` au choix (puisque les nombres réels sont en réalité des nombres floats avec une précision limitée et les rationnels par contre ont une précision "infinie", il faut faire attention à la différence concernant la convergence !). Sur les machines de l'UFR il faut utiliser la commande `eclipse-clp` pour lancer ECLIPSE CLP.

Les contraintes sont toujours incluses entre des accolades (par exemple `{A =< 4, B >= 5}`). Comme d'habitude "," signifie "et". Les contraintes peuvent contenir les fonctions suivantes: `+, -, *, /, abs,sin, cos, tan, pow, exp, min(e1,e2), max(e1,e2)` et les prédicats de base: `=, =:=, <, >, =<, >=, =\=`. Les contraintes écrites entre accolades sont traitées de façon spécifique par Prolog. Elles sont maintenues dans une forme résolues.

Exercice 1
On considère le programme (à prendre de la page web) suivant.

```
:- lib(clpr).
credit(Totale,Temps,_,_,Reste) :- {Temps = 0, Reste = Totale}.
credit(Totale,Temps,Taux,Montant,Reste) :- {Temps >= 1,
     NTotale = Totale + Totale*Taux - Montant, NTemps = Temps - 1},
     credit(NTotale,NTemps,Taux,Montant,Reste).
```

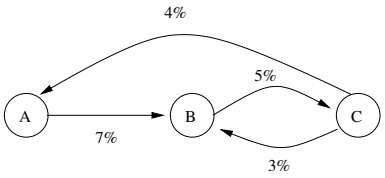
- Que fait ce programme ?
- Écrire une requête pour répondre à la question: Si on rembourse un crédit à taux 10% de 1000€ avec 150€ par an, combien d'argent reste-t-il à rembourser après 10 ans ?
  Remplacer `lib(clpr)` par `lib(clpq)` et observer la différence.
- Écrire une requête pour répondre à la question: Combien d'argent peut-on emprunter à 10% pendant 10 ans si on rembourse 150€ par an ?
- Écrire une requête pour répondre à la question: Quel est le coût global d'un crédit (différence entre les montant total des remboursement et la somme empruntée) d'un million d'Euros à un taux de 7% ?
- Écrire une requête pour répondre à la question: Quel est la différence du coût entre un crédit d'un million d'Euros à 3% et un crédit à 4% sur 10 ans ?
- Écrire une requête qui donne une suite de solutions à la question: Combien d'argent peut-on emprunter à 10% pendant combien d'année, si on rembourse 150€ par an ? Est-ce que cette suite converge (écrire une requête pour répondre à la question) ?

Exercice 2
Programmer le problème suivant: On considère un voyageur qui veut traverser avec un kayak (à une vitesse v_k) une rivière (d'une largeur l et l'eau se déplace avec une vitesse v_r) dans une forêt tropicale. On veut traverser le plus rapidement possible, puisque la rivière est pleine de crocodiles. De l'autre coté de la rivière se trouve une petite clairière (à distance d en aval du point de départ).

À quel distance en amont de la clairière doit commencer la traversée si la rivière a une largeur de 20m, le voyageur a une vitesse de 2m/s, et la rivière a une vitesse de 1m/s ?

Indication: Ce programme s'écrit en une ligne ...

Exercice 3
L'image suivante modélise le flot d'eau entre trois récipients.



Dans chaque unité de temps 7% de l'eau du récipient A coule dans le récipient B, 5% de l'eau coule du récipient B dans le récipient C, etc.

- Écrivez un programme qui calcule la **relation** entre
  - le contenu initial d'eau dans chaque récipient
  - et le contenu d'eau dans chaque récipient après N unités de temps.
- Écrire une requête pour : Combien d'eau est-ce qu'il y a dans chaque récipient après 100 unités de temps, si les récipients contiennent initialement le même montant ?
- Est-ce qu'il y a un point d'équilibre, c.-à-d. un moment où le contenu (initialement non vide) de chaque récipient ne change plus (en utilisant `clpr` ou `clpq`) ?
- (Bonus) Généralisez pour un nombre de récipients et des flôts quelconque. L'exemple ci-dessus pourrait être donné comme la liste `[[93,0,4],[7,95,3],[0,5,93]]`.