

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

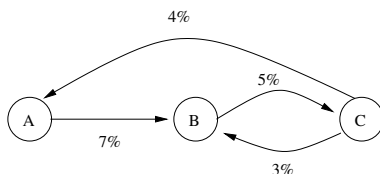
On considère le programme (à prendre de la page web) suivant.

```
:- 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 remboursements 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

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



Dans chaque unité de temps 7% de l'eau du récipient A coule (de façon synchrone) 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]]`.

Exercice 3

Écrire un programme qui, étant donnée une liste (de taille **arbitraire**) de la forme `[0,_,...,_,100]` donne une liste, où chaque élément intérieur de la liste est la moyenne de ses deux voisins. **Indication:** Pour avoir un résultat il faut poser par exemple la question : `L = [0,_,_,_,_,_,_,_,100], votre_predicat(L)`.

Le bon résultat pour l'exemple est `L=[0,12.5,25.0,37.5,50.0,62.5,75.0,87.5,100]`.

Modifier le programme pour faire en sorte que chaque élément de la liste soit la moyenne de ces deux prédécesseurs. On suppose que les deux premières valeurs soient connues. Par exemple, si on pose à PROLOG la question : `L = [0,100,_,_,_,_,_,_], votre_predicat2(L)`. la bonne réponse est `L = [0, 100, 50.0, 75.0, 62.5, 68.75, 65.625]`.

Exercice 4

On veut modéliser la température d'une feuille de métal. Pour cela, on découpe la feuille en une matrice de dimension $m * n$ de points. Si la feuille est dans un état stable, chaque point de la matrice a la même température que la moyenne de ses quatre voisins. Étant données les températures des points limites, les valeurs des autres points sont déterminées.

- Écrire un programme qui étant donné une matrice M avec les valeurs des points limites, calcule la valeur des autres points, par exemple:

```
M = [[100,100,100,100,100,100,100],
     [ 0,  -,  -,  -,  -,  -,  0],
     [ 0,  -,  -,  -,  -,  -,  0],
     [ 0,  -,  -,  -,  -,  -,  0],
     [ 0,  -,  -,  -,  -,  -,  0],
     [ 0,  0,  0,  0,  0,  0,  0]]
```

Rendez la réponse lisible.

Indication: Il suffit de parcourir la matrice et d'imposer les contraintes nécessaires. Utiliser un prédicat auxiliaire `troislignes(+L1,+L2,+L3)` qui impose les contraintes pour 3 lignes de la matrice.

- Considérer une feuille modélisée avec une matrice $9 * 9$. Supposons qu'on sait que la température du centre est 50 et que la température au milieu entre le centre et la partie en haut est 90 (pareil pour la partie gauche). Supposons aussi que les points limites de chaque coté ont la même valeur (H, G, D et B). La température des points dans les coins n'est pas importante. Trouvez les valeurs pour H, G, D et B. Expérimenter avec d'autres contraintes de température.