Université Paris 7 - DEUG-MIAS Année 2003-2004

TP de *Logique et Circuits* n° 3

Notions mathématiques préliminaires

Exercice 1 On cherche à représenter des ensembles (potentiellement infinis) d'entiers en OCaml. Pour cela, on représente un ensemble par sa fonction caractéristique de `int` dans `bool`.

1. Définir un type `ensemble` pour représenter des ensembles par exemple `type ensemble=Car of (int->bool);;`.
2. Définir une fonction `intersection` qui prend deux fonctions caractéristiques et renvoie la fonction caractéristique de l'ensemble intersection.
3. Définir de même une fonction `union`.
4. Définir une fonction `complémentaire`.
5. Définir la fonction caractéristique des nombres pairs.
6. Définir la fonction caractéristique des nombres divisibles par 5.
7. Tester l'intersection, l'union et le complémentaire de ces deux fonctions sur les entiers $0, 1, 2, 3, 4, 5, 6, 10$.

Exercice 2 On travaille ici de même avec des relations binaires. Une relation sur `'a*'b` est représentée comme une fonction de `'a*'b` vers `bool`.

1. Donner un type abstrait pour les relations (entre éléments de types quelconques).
2. Définir des fonctions d'union, d'intersection et de négation.
3. Représenter la relation entre entiers qui est vraie quand la somme des entiers est strictement plus petite que 3.
4. Représenter la relation entre entiers qui est vraie quand la somme des entiers est supérieure à 12.
5. Composer les fonctions d'union, d'intersection, et de négation et les appliquer aux relation définies.

Exercice 3 Dans cet exercice, on représente un ordre noté `ordre` comme une fonction de type `'a*'a->int`. On aura `ordre x y > 0` si `x` est supérieur à `y`, `ordre x y = 0` si `x` est égal à `y` et `ordre x y < 0` sinon.

1. Définir une fonction `let tri2 ordre x y` qui renvoie le couple trié composé de `x` et `y`.
2. Définir à l'aide de `tri2 ordre x y z` une fonction `tri3` triant suivant `ordre`.

3. Définir une fonction `ordre_standard` définissant l'ordre standard sur les entiers.
4. À l'aide de `tri3`, trier `54 2 36` pour tester `ordre_standard`.
5. Définir une fonction `ordre_deux` définissant l'ordre $0 < 1 < -1 < 2 < -2 < 3 < -3 < 4 < -4 \ldots$
6. Trier `54 2 (-2)`
7. Définir une fonction prenant deux ordres et définissant l'ordre lexicographique associé.
8. Trier `(8,0) (5,-5) (5,-3)` ainsi que `(8,0) (5,-5) (5,-12)`.