Licence 3
 Fondements

Université d'Angers

Feuille de travaux dirigés
Fonctions et prédicats récursifs primitifs

Exercice 1 (*Au pied de la lettre)
Montrer que les fonctions suivantes sont récursives primives.

1. la fonction $Z \begin{cases} \mathbb{N} \to \mathbb{N} \\ x \mapsto 0 \end{cases}$
2. les fonctions constantes égales à 3 d'arité 0, 1 et 2
3. la fonction d'addition : $(x, y) \to x + y$
4. la fonction de multiplication : $(x, y) \to x * y$
5. la fonction prédécésseur Pred : $x \mapsto \begin{cases} x - 1 \text{ si x} > 0 \\ 0 \text{ sinon} \end{cases}$
6. la fonction différence tronquée Diff : $(x, y) \mapsto x - y = \begin{cases} x - y \text{ si x} > \text{y} \\ 0 \text{ sinon} \end{cases}$

Exercice 2 (Division et reste)
Montrer que les fonctions suivantes sont récursives primives.

1. la fonction *div* où $div(n, m)$ est le quotient de la division entière de n par m
2. la fonction *mod* où $mod(n, m)$ est le reste de la division entière de n par m

Exercice 3 (Puissance, racine, et log)
Montrer que les fonctions suivantes sont récursives primives.

1. la fonction *pow* où $pow(n, m) = m^n$
2. la fonction *root* où $root(n, m)$ est la racine m-ème de n, c'est-à-dire le plus grand entier k tel que $k^m \leq n$
3. la fonction *log* où $log(n, m)$ est le logarithme en base m de n, c'est-à-dire le plus grand entier k tel que $m^k \leq n$ si $n > 0$, 0 sinon.

Exercice 4 (Nombres premiers)
Trouver une fonction récursives primitives *premier*, telle que $premier(n) = 0$ si n n'est pas premier, 1 sinon.

Exercice 5
Montrer que le prédicat $x > y$ est récursif primitif.

Exercice 6
Montrer que les constructeurs suivants sont récursifs primitifs (c'est à dire que s'ils sont utilisés sur des fonctions récursives primitives du bon type alors les fonctions qu'ils construisent sont bien récursives primitives).

1. pour $k \in \mathbb{N}$, si P est un prédicat d'arité k, f et g sont des fonctions d'arité k, alors $Si(P, f, g)$ est la fonction h d'arité k telle que lorsque P est vrai $h = f$ et lorsque P est faux $h = g$.
2. pour $k \in \mathbb{N}$, si f est une fonction d'arité $k + 1$ alors $Somme(f)$ est la fonction h d'arité $k + 1$ telle que :

$$h(y, x_1, \ldots, x_k) = \Sigma_{i=0}^{y} f(i, x_1, \ldots, x_k)$$

3. pour $k \in \mathbb{N}$, si f est une fonction d'arité $k+1$ alors *Produit*(f) est la fonction h d'arité $k+1$ telle que :

$$h(y, x_1, \ldots, x_k) = \Pi_{i=0}^{y} f(i, x_1, \ldots, x_k)$$

Exercice 7

Montrer que les constructeurs suivants sont récursifs primitifs :

1. Minimisation bornée (autre version que celle du cours). Pour $l \in \mathbb{N}$, si P est un prédicat d'arité $l+1$, alors $\mu \leq (f)$ est la fonction h d'arité $l+1$ telle que :

$$h(n, x_1, \ldots, x_l) = \begin{cases} \text{le plus petit } j \leq n \text{ tel que } P(j, x_1, \ldots, x_l) \text{ s'il en existe un} \\ n+1 \text{ sinon} \end{cases}$$

   Ecrire un algorithme implantant cette fonction.

Exercice 8 **(*Récursif, mais primitif ? ? ? : Tours de Hanoï)**

Le problème des tours de Hanoï consiste à déplacer des disques de diamètres différents d'une tour de départ à une tour d'arrivée en s'aidant d'une tour intermédiaire, et ceci en un minimum de coups, tout en respectant les règles suivantes :

— on ne peut déplacer plus d'un disque à la fois,

— on ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide.

On suppose que cette dernière règle est également respectée dans la configuration de départ.

1. donner un algorithme permettant de déplacer une tour de n disques d'un emplacement de départ D à un emplacement d'arrivée A en s'aidant d'un emplacement intermédiaire I.
2. maintenant, les disques sont répartis de façon aléatoire sur les 3 emplacements. Ecrire un algorithme pour regrouper les n disques sur l'emplacement d'arrivée A ;