TD de *Logique et Circuits* n° 9
(Correction)

Induction sur les entiers

Exercice 1 Soit R une relation binaire sur un ensemble E. Posons

$$R^0 = id, R^{i+1} = R \circ R^i.$$

Montrer que $\forall i, j \geq 0, R^{i+j} = R^i \circ R^j$.

Correction :

Par induction sur i:

- $i = 0$. Puisque $R^0 = id$, on a $R^0 \circ R^j = R^j$.
- $R^{i+1} \circ R^j = R \circ R^i \circ R^j = R \circ R^{i+j} = R^{i+j+1}$.

Exercice 2 Soit $\mathcal{M}$ l'ensemble de tous les mots sur un alphabet A donné. Montrer que pour tous mots u, v dans $\mathcal{M}$, on a:

$$u \cdot v = v \cdot u \Longleftrightarrow \exists w \in \mathcal{M}, \exists p, q \in \mathbb{N} \mid u = w^p \text{ et } v = w^q.$$

Correction :

Par induction sur $|u| + |v|$. Si $|u| + |v| = 0$ il faut montrer la validité de la phrase suivante:

$$\exists w \in \mathcal{M}, \exists p, q \in \mathbb{N} \mid \epsilon = w^p \text{ et } \epsilon = w^q.$$

Ceci est valable par exemple pour $w = \epsilon$.

Supposons que $\forall k < n$, la propriété soit vraie. Soient u, v t.q. $|u| + |v| = n$. Supposons que $|u| < |v|$ (par symétrie).

$\Rightarrow$)

L'hypothèse $u \cdot v = v \cdot u$ implique u préfixe de v. Si $u = \epsilon$ on prend $w = v$, $p = 0$ et $q = 1$. Si $u = v$, on prend $w = v$, $p = 1$ et $q = 1$. Sinon, soit v' tq $v = uv'$. On a $uv = uv'u = vu$, donc $v'u = v = uv'$. Comme $|v'| < |v|$, on a par récurrence sur $|u| + |v'| < |u| + |v|$:

$$\exists w \in \mathcal{M}, \exists p, q \in \mathbb{N}, u = w^p, v' = w^q.$$

Donc, on a $u = w^p$ et $v = w^{p+q}$.

$\Leftarrow$)

L'hypothèse $\exists w \in \mathcal{M}, \exists p, q \in \mathbb{N} \mid \epsilon = w^p$ et $\epsilon = w^q$ implique $u \cdot v = w^p \cdot w^q = w^{p+q} = w^q \cdot w^p = v \cdot u$.

Exercice 3 Une fonction récursive f peut être exprimée sous la forme d'une équation récursive de la forme:

$$f(x) = g(x, f(h(x)))$$

Où h et g sont des fonctions telles que $h(x) < x$. Déterminer la fonction f dans les cas suivants:

1. $f : \mathbb{N} \to \mathbb{N}$, $f(0) = 1$, $h(x) = x - 1$ et $g(x, y) = 2y$,

2. $f : \mathbb{N} \rightarrow \mathbb{N}$, $f(0) = 0$, $h(x) = x - 1$ et $g(x,y) = y + 2x - 1$,
3. $f : \mathbb{N} \rightarrow \mathbb{N}$, $f(0) = 0$, $h(x) = [x/2]$ et

$$g(x,y) = \begin{cases} 4y & : \text{ si } x \text{ pair} \\ 4y + 9(2x-1) & : \text{ sinon} \end{cases}$$

Correction :

1. $f(n) = 2^n$,
2. $f(n) = n^2$,
3. $f(n) = (3n)^2$.

Exercice 4 On appelle F_n le n-ième nombre de Fibonacci: $F_0 = F_1 = 1$ et $F_{n+2} = F_{n+1} + F_n$.

1. Écrire une fonction `let rec fibo n` qui calcule F_n.
2. Soit r_n le nombre d'appels de `fibo` pour le calcul de F_n. Montrer que $r_n = 2F_{n+1} - 1$.
3. Donner des arguments intuitifs au fait que le temps d'exécution de la fonction `fibo` est exponentiel.
4. Écrire une fonction `let rec iter n f x` qui calcule $f^n(x)$ $(f^0(x) = x, f^1(x) = f(x), \ldots, f^{i+1}(x) = f(f^i(x)))$.
5. On remarque que $(F_{n+2}, F_{n+1}) = s(F_{n+1}, F_n)$, où s est une fonction qui à une paire (x,y) associe la paire $(x+y,x)$. Montrer que $(F_n, F_{n-1}) = s^{n-1}((F_1, F_0))$ pour $n \geq 1$. En déduire une fonction `let fibp n` qui calcule F_n en un temps linéaire.
6. Soit la fonction $fib(n) = (F_n, F_{n-1})$. Exprimer $fib(n)$ sous la forme d'une équation récursive.

Correction :

1.
```
let rec fibo = function 0 -> 1
                      | 1 -> 1
                      | n -> fibo(n-1)+fibo(n-2);;
```

2. $r_0 = r_1 = 1$, et pour $n \geq 2$, $r_n = 1 + r_{n-1} + r_{n-2}$.
4. `let rec iter n f x = if n=0 then x else f(iter (n-1) f x);;`
5. `let fibp n = if n=0 then 0 else fst(iter (n-1) (function(x,y) -> (x+y,x)) (1,0));;`
6. $h(x) = x - 1$, si $y = (y_1, y_2)$, $g(x,y) = (y_1 + y_2, y_1)$.

Exercice 5 Soit `mod_m` la fonction définie comme suit:

```
let rec mod_m(n)= if m=0 then failwith''erreur''
                  else if n<m then n else mod_m(n-m);;
```

1. Que calcule cette fonction?

2. Exprimer `mod_m` sous la forme d'une équation récursive.

3. Montrer que cette fonction termine.

Correction :

1. `mod_m`(n) donne le reste de la division euclidienne de n par m.
2. `mod_m`(n)= $g_m(n,$ `mod_m` $(h_m(n)))$, $h_m(n) = n - m$, $g_m(x,y) = x$ si $x < m$, et $g_m(x,y) = y$ sinon.