Maîtrise de Mathématiques **Université de Nantes**
Informatique **Année 1984–85**

Épreuve du 27/09/1985
(durée : 4 heures)

Rappels

$\mathbb{N}$ désigne l'ensemble des entiers naturels, $\mathbb{N}^n$ le produit cartésien de n facteurs égaux à $\mathbb{N}$ ($\mathbb{N}^0$ est réduit à un seul point). Un élément de $\mathbb{N}^n$ est noté $(x_1, x_2, \ldots, x_n)$, $\mathbb{N}^1$ étant identifié à $\mathbb{N}$, x_1 et (x_1) désignent le même objet. Si $(x_1, \ldots, x_n) \in \mathbb{N}^n$, et $(y_1, \ldots, y_p) \in \mathbb{N}^p$, l'élément $(x_1, \ldots, x_n, y_1, \ldots, y_p)$ de $\mathbb{N}^{n+p}$ sera noté :

$$(x_1, \ldots, x_n) * (y_1, \ldots, y_p)$$

On note $\mathbb{N} \xrightarrow{\mathcal{S}} \mathbb{N}$ la fonction successeur ($\mathcal{S}(x) = x + 1$), et $\mathbb{N}^n \xrightarrow{t_n} \mathbb{N}$, $\mathbb{N}^n \xrightarrow{q_n} \mathbb{N}^{n-1}$, les *projecteurs* définis pour $n \geq 1$ par :

$$\begin{aligned} t_n(x_1, \ldots, x_n) &= x_1 \\ q_n(x_1, \ldots, x_n) &= (x_2, \ldots, x_n) \end{aligned}$$

La classe des fonctions *primitivement récursives* (en abbrégé PR) est le plus petit ensemble contenant les fonctions $\mathcal{S}$, t_n, q_n ($n \geq 1$), la constante 0 (considérée comme une fonction de $\mathbb{N}^0$ vers $\mathbb{N}$), et stable par les opérations de composition, couplage et récursion primitive :

Composition : Si $\mathbb{N}^n \xrightarrow{f} \mathbb{N}^p$ et $\mathbb{N}^p \xrightarrow{g} \mathbb{N}^q$ sont PR, il en est de même de $\mathbb{N}^n \xrightarrow{g \circ f} \mathbb{N}^q$.

Couplage : Si $\mathbb{N}^n \xrightarrow{f} \mathbb{N}^p$ et $\mathbb{N}^n \xrightarrow{g} \mathbb{N}^q$ sont PR, il en est de même de $\mathbb{N}^n \xrightarrow{\langle f, g \rangle} \mathbb{N}^{p+q}$, définie par :

$$\langle f, g \rangle(x_1, \ldots, x_n) = f(x_1, \ldots, x_n) * g(x_1, \ldots, x_n).$$

Récursion primitive : Si $\mathbb{N}^{n-1} \xrightarrow{g} \mathbb{N}$ et $\mathbb{N}^{n+1} \xrightarrow{h} \mathbb{N}$ ($n \geq 1$) sont PR, il en est de même de $\mathbb{N}^n \xrightarrow{f} \mathbb{N}$ définie par :

$$\begin{aligned} f(0, x_2, \ldots, x_n) &= g(x_2, \ldots, x_n) \\ f(x_1 + 1, x_2, \ldots, x_n) &= h(f(x_1, \ldots, x_n), x_1, \ldots, x_n) \end{aligned}$$

Première partie

On considère les symboles suivants : O, S, C, P, R, J, T_n, Q_n, pour $n \geq 1$. On définit le langage typé $\mathcal{L}_{RJ}$ comme suit :

- Chaque expression de $\mathcal{L}_{RJ}$ a un *type*, qui est un couple d'entiers, et on note $\mathcal{L}_{RJ}(n, p)$ l'ensemble des expressions de $\mathcal{L}_{RJ}$ qui sont de type (n, p).
- Les expressions de $\mathcal{L}_{RJ}$ sont définies récursivement par :
  - $O \in \mathcal{L}_{RJ}(0, 1)$
  - $S \in \mathcal{L}_{RJ}(1, 1)$

- $T_n \in \mathcal{L}_{RJ}(n,1)$ (pour $n \geq 1$)
- $Q_n \in \mathcal{L}_{RJ}(n,n-1)$ (pour $n \geq 1$)
- Si $A \in \mathcal{L}_{RJ}(p,q)$ et $B \in \mathcal{L}_{RJ}(n,p)$ alors $CAB \in \mathcal{L}_{RJ}(n,q)$
- Si $A \in \mathcal{L}_{RJ}(n,p)$ et $B \in \mathcal{L}_{RJ}(n,q)$, alors $PAB \in \mathcal{L}_{RJ}(n,p+q)$
- Si $A \in \mathcal{L}_{RJ}(n-1,1)$ et $B \in \mathcal{L}_{RJ}(n+1,1)$ alors $RAB \in \mathcal{L}_{RJ}(n,1)$
- Si $A \in \mathcal{L}_{RJ}(n,n)$ alors $JA \in \mathcal{L}_{RJ}(n+1,n)$

On note $\mathcal{L}_R$ le sous–langage (sous–ensemble) de $\mathcal{L}_{RJ}$ formé des expressions qui ne contiennent pas J, et $\mathcal{L}_J$ le sous–langage de $\mathcal{L}_{RJ}$ formé des expressions qui ne contiennent pas R.

Question 1. Montrer que $ROCST_2$ et $RROCST_2CST_3$, notées respectivement I et Σ, sont des expressions de $\mathcal{L}_R$, et calculer leurs types.

On définit maintenant la *sémantique* ou *interprétation* d'une expression E de $\mathcal{L}_{RJ}(n,p)$ comme une application de $\mathbb{N}^n$ vers $\mathbb{N}^p$, notée $[\![E]\!]$, comme suit :

- $[\![O]\!] = 0$
- $[\![S]\!] = \mathcal{S}$
- $[\![T_n]\!] = t_n$
- $[\![Q_n]\!] = q_n$
- $[\![CAB]\!] = [\![A]\!] \circ [\![B]\!]$
- $[\![PAB]\!] = \langle [\![A]\!], [\![B]\!] \rangle$
- $[\![RAB]\!](0, x_2, \ldots, x_n) = [\![A]\!](x_2, \ldots, x_n)$
- $[\![RAB]\!](x_1+1, x_2, \ldots, x_n) = [\![B]\!]([\![RAB]\!](x_1, \ldots, x_n), x_1, \ldots, x_n)$
- $[\![JA]\!](x_0, x_1, \ldots, x_n) = [\![A]\!]^{x_0}(x_1, \ldots, x_n)$

(où $[\![A]\!]^{x_0}$ représente $[\![A]\!] \circ \ldots \circ [\![A]\!]$ (x_0 facteurs)).

Question 2. Montrer que pour tous x et y de $\mathbb{N}$, on a :

$$\begin{array}{rcl} [\![I]\!](x) & = & x \\ [\![\Sigma]\!](x,y) & = & x+y \end{array}$$

Question 3. Montrer que pour tout E de $\mathcal{L}_R$, $[\![E]\!]$ est PR.

Question 4. Montrer que pour toute fonction PR f, il existe une expression E de $\mathcal{L}_R$, telle que $[\![E]\!] = f$.

Question 5. Montrer que pour tous A et B de $\mathcal{L}_{RJ}$, tels qur RAB soit dans $\mathcal{L}_{RJ}(n,1)$, on a :

$$\begin{array}{rcl} [\![RAB]\!] & = & [\![CT_{n+1}PRABPT_nQ_n]\!] \\ [\![PRABPT_nQ_n]\!] & = & [\![CJ\varphi PT_nPCAQ_nPCOCQ_1T_nQ_n]\!] \end{array}$$

où $\varphi = PBPCSCT_nQ_{n+1}CQ_nQ_{n+1}$

Question 6. Définir récursivement une application $\mathcal{D}$ de $\mathcal{L}_R$ vers $\mathcal{L}_J$, telle que pour tout E de $\mathcal{L}_R$, on ait : $[\![\mathcal{D}(E)]\!] = [\![E]\!]$.

Deuxième partie

On considère maintenant une machine dont voici la description.

- La machine exécute des programmes, qui sont des suites d'instructions (dont certaines sont des parenthèses ouvrantes ou fermantes). L'ensemble de ces instructions est le suivant :

  `Cut Push Swap Append Incr Zero Car Cdr ( )`

  Les instructions d'un programme sont disposées en ligne de la gauche vers la droite.
- La machine manipule en outre un *registre* qui peut contenir une suite (finie) d'entiers, et une *pile* qui peut contenir une suite $A_1, \ldots, A_n$, où les A_i sont eux–mêmes des suites d'entiers. A_1 est appelé le *sommet* de la pile.
- Tous les cas qui ne sont pas prévus dans le tableau provoquent l'émission d'un message d'erreur et l'arrêt de la machine.

La signification des instructions est donnée par le tableau suivant :

Fonctionnement de la machine à registre et pile.

Instruction exécutée	État avant exécution		État après exécution		Prochaine instruction à exécuter
	Registre	Pile	Registre	Pile	
Cut	$(x_0, x_1, \ldots, x_n)$	$(A_1, \ldots)$	$(x_1, \ldots, x_n)$	$((x_0), A_1, \ldots)$	suivante
Push	A_0	$(A_1, \ldots)$	A_0	$(A_0, A_1, \ldots)$	suivante
Swap	A_0	$(A_1, A_2, \ldots)$	A_1	$(A_0, A_2, \ldots)$	suivante
Append	A_0	$(A_1, A_2, \ldots)$	$A_1 * A_2$	$(A_2, \ldots)$	suivante
Incr	(x)	$(A_1, \ldots)$	$(x+1)$	$(A_1, \ldots)$	suivante
Zero	A_0	$(A_1, \ldots)$	(0)	$(A_1, \ldots)$	suivante
Car	$(x_1, x_2, \ldots, x_n)$	$(A_1, \ldots)$	(x_1)	$(A_1, \ldots)$	suivante
Cdr	$(x_1, x_2, \ldots, x_n)$	$(A_1, \ldots)$	$(x_2, \ldots, x_n)$	$(A_1, \ldots)$	suivante
(	A_0	$((x+1), A_1, \ldots)$	A_0	$((x), A_1, \ldots)$	suivante
(	A_0	$((0), A_1, \ldots)$	A_0	$(A_1, \ldots)$	Note 1.
)	A_0	$(A_1, \ldots)$	A_0	$(A_1, \ldots)$	Note 2.

Note 1. Instruction qui vient immédiatement après la parenthèse fermante correspondante.

Note 2. Parenthèse ouvrante correspondante.

Note 3. L'exécution d'un programme commence à l'instruction la plus à gauche et se termine quand la machine cherche à exécuter une instruction qui se trouverait immédiatement à droite de l'instruction la plus à droite du programme.

On note $\mathcal{P}$ l'ensemble de tous les programmes de cette machine, c'est à dire l'ensemble de toutes les suites finies d'instructions. On note $\overline{\mathbb{N}}$ l'ensemble de toutes les suites finies d'entiers. Noter qu'on a des inclusions canoniques : $\mathbb{N}^n \longrightarrow \overline{\mathbb{N}}$.

Pour tout $P \in \mathcal{P}$, la *sémantique* de P est l'application (*a priori* partielle, c'est–à–dire non nécessairement définie partout) :

$$\overline{\mathbb{N}} \xrightarrow{[\![P]\!]} \overline{\mathbb{N}}$$

définie comme suit. Soit $(x_1, \ldots, x_n) \in \overline{\mathbb{N}}$. On fait démarrer la machine avec une pile vide (i.e. contenant la suite vide), et $(x_1, \ldots, x_n)$ dans le registre. Si la machine ne s'arrête pas où émet un message d'erreur,

$[\![P]\!](x_1, \dots, x_n)$ n'est pas défini. Si la machine s'arrête sans émettre de message d'erreur, $[\![P]\!](x_1, \dots, x_n)$ est le contenu du registre après l'arrêt.

On définit maintenant une application $\mathcal{L}_J \xrightarrow{[\,]} \mathcal{P}$ appelée *compilateur*, de la façon suivante :

- $[O] =$ Zero
- $[S] =$ Incr
- $[T_n] =$ Car
- $[Q_n] =$ Cdr
- $[CAB] = [B][A]$
- $[PAB] =$ Push $[A]$ Swap $[B]$ Append
- $[JA] =$ Cut ($[A]$)

Question 7. Montrer simultanément, et par récurrence sur la structure de E, que pour tout E de $\mathcal{L}_J(n, p)$, et tout $(x_1, \dots, x_n)$ de $\mathbb{N}^n$,

(a) $[\![[E]]\!](x_1, \dots, x_n)$ est défini,

(b) $[\![[E]]\!](x_1, \dots, x_n) = [\![E]\!](x_1, \dots, x_n)$

(c) L'état de la pile avant l'exécution de $[E]$ n'a pas d'influence sur le résultat, et se retrouve inchangée après l'exécution de $[E]$.

Question 8. Montrer que pour toute fonction PR, $\mathbb{N}^n \xrightarrow{f} \mathbb{N}^p$, il existe un programme $P \in \mathcal{P}$, tel que :

$$[\![P]\!](x_1, \dots, x_n) = f(x_1, \dots, x_n)$$

pour tous $x_1, \dots, x_n$. Autrement–dit, toute fonction PR est calculable par cette machine à registre et pile.