

## Examen du 15 mai 2007

1ère partie (10 points) - à traiter sur une copie séparée

### 1ère partie : Interprétation d'un langage fonctionnel

Le but de ce problème est d'implémenter un interprète du langage EXAM<sub>150507</sub>, sous-ensemble du langage SCHEME.

#### Question 1 : interprétation du langage EXAM<sub>150507</sub>

EXAM<sub>150507</sub> est un petit langage d'expressions. Une expression est soit une constante (booléen, entier, nil), soit un couple, soit un symbole, soit l'application d'une fonction à un argument, soit l'abstraction d'une fonction, soit la structure conditionnelle, soit la définition d'un symbole. Les principales structures de contrôle et de définition sont :

- l'application (`expr1 expr2`) : qui applique la fonction `expr1` à l'argument `expr2`;
- l'abstraction (`LAMBDA p1 (corps)`) : qui définit une fonction anonyme à 1 paramètre formel qui calculera `corps` lors de l'application en substituant le paramètre formel par l'argument d'appel.
- la conditionnelle (`IF c e1 e2`) : qui selon le résultat de l'évaluation de `c` évaluera `e1` ou `e2`;
- la définition (`DEFINE nom e`) : qui associe `nom` à la valeur de `e` dans l'environnement global

1. Ecrire un interprète pour le langage EXAM<sub>150507</sub> sous la forme d'une fonction prenant en paramètres un environnement `env` (liaison nom-valeur) et une expression `expr` du langage et retourne la valeur de l'expression `expr` calculée dans l'environnement `env`. Commencer par définir les types `expr`, `environnement` et `valeur` dans un des langages d'implantation proposés : C, Java ou OCaml.

#### Question 2 : interprète de la machine CAM

La mise en œuvre du langage Caml (l'ancêtre) repose sur la Machine Abstraite Catégorique (Categorical Abstract Machine ou CAM en grand breton) .

#### Instructions de la machine

instructions de la machine <i>Inst</i>	définition de <i>Val</i>
<i>Inst</i> ::=  <i>Inst<sub>F</sub></i> (instructions associées aux opérateurs de base), Push : met l'accumulateur sur le sommet de pile, Swap : échange le sommet de pile et l'accumulateur, Cons : consigne le sommet de pile à l'accumulateur, Fst : prend le premier élément de l'accumulateur, Snd : prend le deuxième élément de l'accumulateur, Id : NOP App : exécute le code de la fermeture du 1er élément de l'accumulateur Cur : fabrique une fermeture (code + environnement) Quote Val : n'évalue pas Val	<i>Val</i> ::=  C : constante de base Val : liste d'instructions (fermeture) (Val,Val) : couple de valeurs

### Schéma d'évaluation de la CAM

accumulateur	code	pile	accumulateur	code	pile
$(s, t)$	Fst;C	$S$	$s$	C	$S$
$(s, t)$	Snd;C	$S$	$t$	C	$S$
$s$	Id;C	$S$	$s$	C	$S$
$s$	Quote $c$ ;C	$S$	$c$	C	$S$
$s$	Cur (C);C1	$S$	$(s :: C)$	C1	$S$
$s$	Push;C	$S$	$s$	C	$s :: S$
$t$	Swap;C	$s :: S$	$s$	C	$t :: S$
$t$	Cons;C	$s :: S$	$(s, t)$	C	$S$
$(s :: C, t)$	App;C1	$S$	$(s, t)$	C;C1	$S$
$s$	Ins <sub>f</sub> ;C	$S$	$f(s)$	C	$S$

2. Ecrire un interprète de la CAM en suivant les schémas d'évaluation indiqués. Le langage d'implantation est toujours à choisir entre C, Java ou O'Caml.

### Question 3 : Schéma de compilation

On cherche maintenant à compiler EXAM<sub>150507</sub> vers la CAM. Voici la traduction du  $\lambda$ -calcul pur vers la CAM.

#### Traduction du $\lambda$ -calcul en CAM

Le schéma de compilation utilise un motif  $p$  pour mémoriser la place des variables.

$$\begin{aligned}
 \llbracket x \rrbracket_{(p,x)} &= Snd \\
 \llbracket x \rrbracket_{(x,p)} &= Fst \\
 \llbracket x \rrbracket_{(p_1,p_2)} &= (Snd; \llbracket x \rrbracket_{p_2})?(Fst; \llbracket x \rrbracket_{p_1}) \\
 \llbracket MN \rrbracket_p &= \langle \llbracket M \rrbracket_p, \llbracket N \rrbracket_p \rangle; App \\
 \llbracket (M, N) \rrbracket_p &= \langle \llbracket M \rrbracket_p, \llbracket N \rrbracket_p \rangle \\
 \llbracket \lambda v. M \rrbracket_p &= Cur(\llbracket M \rrbracket_{(p,v)})
 \end{aligned}$$

La notation  $e_1 ? e_2$  signifie la valeur de  $e_1$  si elle existe, sinon celle de  $e_2$ .

Le combinateur paire donne la suite d'instructions suivantes :

$$\langle M, N \rangle \rightarrow Push; M; Swap; N; Cons$$

3. Donner les schémas de compilation de langage EXAM<sub>150507</sub> vers la CAM en vous inspirant des schémas du  $\lambda$ -calcul. Indiquer la stratégie d'évaluation utilisée.

### Question 4 : compilateur EXAM<sub>150507</sub> vers CAM

4. En suivant les schémas de compilation que vous avez donné à la question précédente, écrire un compilateur du langage EXAM<sub>150507</sub> vers la CAM en utilisant un des trois langages d'implantation conseillés : C, Java ou O'Caml.