

Introduction aux machines virtuelles

Examen du 25 juin 2009

Duré : 3h. Docum nts autorisés

I) Caml (~6 points)**Exercice 1** Traduire n byt cod Caml l s phras s suivant s :

- $8*6/4$
- `let double x = x+x in double 6`
- `let = 10 in let decuple x = * x in decuple 5`
- `let = (1,2) in fst + snd`

Indic : `fst` `t` `snd` sont traduit dir ct m nt n primitiv s sur l s blocs.

Cit r un autr phras Caml qui pourrait donn r l mêm byt cod .

Exercice 2 Étant donné la portion d byt cod Caml suivant :

```

      closurerec 1, 0
      const [0: 10 [0: 20 0 ]]
      push
        cc 1
        pply 1
      push
      const 0
      ddint
      return 2
L1:   cc 0
      br nchifnot L2
      cc 0
      getfield 1
      push
      offsetclosure 0
      pply 1
      push
      const 1
      ddint
      return 1
L2:   const 0
      return 1

```

- On xécute c byt cod jusqu'au pr mi r passag par l'instruction `offsetclosure`. Décrire alors l'état d la machin virtu ll Caml après c tt instruction : cont nu d l'accumulat ur, d la pil t du tas. Justifi r brièv m nt.
- R trouv r un phras Caml dont la compilation produit c byt cod . Associ r à chaqu élém nt d c tt phras la parti byt cod qui corr spond. Quel st l résultat d l'xécution ?

II) Ja a (~6 points)

Exercice 3

a) Le langage Logo

Voici les instructions du notre langage mini-Logo. Nous ne les utilisons pas directement, elles ne sont décrites ici que pour aider à la compréhension des instructions du bytecode de la section suivante.

- **Avance n** : déplace la tortue de n dans sa direction courante, sans changer son orientation.
- **Reculer n** : déplace la tortue de n dans la direction opposée à son orientation actuelle. L'orientation reste préservée lors du déplacement.
- **Girer n** : l'orientation est augmentée de n degrés.
- **Droite n** : l'orientation est diminuée de n degrés.
- **LeveCrayon** : suspend le tracé lors des déplacements à venir de la tortue.
- **BaisseCrayon** : reprend le tracé lors des déplacements à venir de la tortue.
- **Repete n [...]** : itère n fois les instructions présentes entre les crochets.
- **Pour p ... Fin** : associée à la chaîne de caractères p la procédure constituée de la suite d'instructions allant jusqu'au **Fin**
- p : si la chaîne de caractères p correspond à une procédure **Pour p ... Fin**, exécute les instructions correspondantes à cette procédure

Par exemple, un programme Logo dessinant un carré peut alors s'écrire :

```
Avance 10 Girer 90 Avance 10 Girer 90 Avance 10 Girer 90 Avance 10 Girer 90
```

ou aussi :

```
Repete 4 [ Avance 10 Girer 90 ]
```

ou encore :

```
Pour CARRE
```

```
  Repete 4 [ Avance 10 Girer 90 ]
```

```
Fin
```

```
CARRE
```

b) Un bytecode Logo

Au lieu de travailler avec les instructions "haut niveau" précédentes, on va utiliser une version "bytecode", dans laquelle chaque instruction correspond à un nombre entier ou double, le premier nombre étant toujours le code de l'instruction :

- **STOP** (cod = 0) : arrête le programme.
- **AVANCE n** (cod = 1) : avance la tortue de n , cette quantité pouvant être négative.
- **TOURNE n** (cod = 2) : ajoute n degrés à l'orientation de la tortue (n peut être négatif).
- **LEVECRAYON** (cod = 3)
- **BAISSECRAYON** (cod = 4)
- **REPETE n** (cod = 5) : instruction marquant le début d'un bloc à répéter n fois
- **FINREPETE o** (cod = 6) : fin d'un zone à répéter. L'entier o est un décalage (offset) permettant de revenir juste après l'instruction **REPETE** correspondante (cf. exemple ci-dessous).
- **FINPOUR** (cod = 7) : instruction marquant la fin d'une procédure et retour à l'endroit de l'appel.

- **LANCE** o (cod =8) : lanc m nt d la procédur situé o nti rs plus loin dans l byt cod .

Ainsi l s trois x mpl s précéd nts corr spond nt aux trois portions d byt cod suivants :

1 10 2 90 1 10 2 90 1 10 2 90 1 10 0

5 4 1 10 2 90 6 -6 0

8 1 0 5 4 1 10 2 90 6 -6 7

Exercice 5

1. On considèr d'abord un Machin Virtuel Logo (MVL) n traitant qu l s instructions d byt cod 0 à 4. Qu ls sont l s r gistr s néc ssair s au fonctionn m nt d c tt MVL? L'usag d'un pil st-il néc ssair ?
2. Qu faut-il n plus pour obt nir un MVL traitant aussi l s instructions 5 t 6? Votr solution p rm t- ll d'imbriqu r d s répétitions? Comm nt trait r c cas?
3. Qu faut-il n plus pour obt nir un MVL traitant aussi l s instructions 7 t 8? Votr solution p rm t- ll d'imbriqu r d s app ls d procédur s dans d'autr s procédur s? Comm nt trait r c cas?
4. L'argum nt o d l'instruction 6 (**FINREPETE**) st-il indispsabl ? Si c n' st pas l cas, qu faut-il n plus comm structur pour s' n pass r?

Exercice 6 En utilisant l langag d votr choix (parmi Ocaml, Java t C), écri r un Machin Virtuel Logo, c' st-à-dir un fonction r c vant un tabl au d nombr s corr spondant à du byt cod Logo t réalisant l d ssin corr spondant.

On pourra suppos r l' xist nc d'un fonction **dessine_ligne** qui att nd quatr nti rs x_1 y_1 x_2 y_2 t trac l s gm nt ntr (x_1, y_1) t (x_2, y_2) .

Indication : Dans un pr mi r t mps, vous pouv z vous conc ntr r sur la réalisation d'un machin virtuel traitant s ul m nt l s instructions 0 à 4, puis 0 à 6. Évid mm nt, à vaincr sans péril, on triomph sans gloir , t on a moins d points...