

Introduction aux machines virtuelles – TD n° 4

Bytecode Caml

Exercice 1 Pour chacune des phrases Caml suivantes, déterminez une traduction en bytecode, comparez vos propositions avec la sortie de `ocaml -dinstr`, puis décrivez l'évolution de la pile et du tas lors de l'exécution de ce bytecode. Vérifier cette évolution de la pile avec l'interpréteur fourni.

- `1-2+5;;`
- `match [1;2;3] with [] -> 50 | [a] -> a | t::q -> t;;`
- `type t = C1 | C2 of int | C3 of int*int;;`
`match C3 (2,3) with C1 -> 0 | C2 n -> n | C3 (a,b) -> a+b;;`
- `(fun x -> x+1) 42 + 0;;`
- `(fun x -> x+1) 42;;`
- `let rec fact n = if n=0 then 1 else (fact (n-1))*n`
`in fact 5;;`
- `let rec pow a n = if n=0 then 1 else a*(pow a (n-1))`
`in pow 3 5;;`
- `let rec pow a n = if n=0 then 1 else a*(pow a (n-1))`
`in let f = pow 3`
`in f 5;;`
- `let rec fact' a n = if n=0 then a else fact' (a*n) (n-1)`
`in fact' 1 5;;`
- `let f =`
`let a = 10 in`
`let g x = x+1 in`
`(fun x -> g x + a)`
`in f 42 * 1;;`
- `let app f = f in`
`app (fun x->x+1) 10`

Exercice 2 Ecrire en bytecode une fonction à trois arguments prenant une liste `l`, un nombre `n` et une donnée `a` et remplaçant le `n`-ième élément de la liste `l` par `a`. Ceci devra se faire *en place* (pas de nouvelle allocation de liste). Tester votre fonction. Que se passe-t'il si la donnée `a` n'a pas le même type que les autres éléments de la liste? Que se passe-t'il si par erreur on modifie une zone de pointeur dans la liste au lieu d'une zone de donnée?

Exercice 3 Ecrire une portion de bytecode qui commence par définir deux fonctions du même type (disons `f` et `g`) puis agit en place sur la représentation de ces fonctions pour faire en sorte que `g` soit la fonction exécutée lorsque `f` est lancée. Application : imaginer comment réaliser l'équivalent du `#trace`.