

TD-TME 5

Exceptions

Soient les deux programmes suivants :

en Scheme	en O'Caml
<pre> ; \$Date: 2006/05/15 22:55:40 \$ (define (lastn i l) (if (pair? l) (let ((r (lastn i (cdr l)))) (cond ((not (integer? r)) r) ((= r i) l) (#t (+ r 1)))) 0)) </pre>	<pre> (* \$Date: 2006/05/15 22:55:40 \$ *) type 'a int_or_alpha_list = I of int L of 'a list;; let lastn i l = let rec lastn_aux l = match l with [] -> I 0 _::t -> let r = lastn_aux t in (match r with L al -> r I ir -> if ir = i then (L l) else (I (ir+1))) in match (lastn_aux l) with L al -> al _ -> [];; </pre>

qui retournent la liste des n derniers éléments de la liste l paramètre.

1. Quelle est la principale différence entre ces deux versions?
2. Ecrire une version C utilisant la structure de données `liste_entier` vue au td précédent.
3. On cherche à indiquer que le calcul n'a pas pu être effectué si la liste est trop courte. En utilisant un mécanisme d'exceptions, modifier les version O'Caml et C de ce programme. Pour la version C, utiliser l'ensemble de macros et de fonctions vu au cours 5.
4. On cherche maintenant à déclencher une exception dès que la liste cherchée est trouvée. Modifier en conséquence les programmes précédents.

Programmation fonctionnelle en C

On cherche à écrire directement en C les fonctions principales d'une bibliothèque sur les listes à la manière de la bibliothèque `List` d'O'Caml. Soit le programme suivant :

```

(* on suppose hd, tl et :: predefinies *)
(* definir : failwith et invalid_arg *)

```

```

let rec nth l n =
  if l = [] then failwith "nth"
  else let a = hd l and q = tl l in
        if n = 0 then a else
        if n > 0 then nth q (n-1) else
        invalid_arg "List.nth";;

```

```

let rec map f l =
  if l = [] then []
  else let a = hd l and q = tl l in

```

```

let r = f a in r :: map f q;;

let rec fold_right f l accu =
  if l = [] then accu
  else let a = hd l and q = tl l in
        f a (fold_right f q accu);;

```

1. Définir en C les fonctions `failwith` et `invalid_arg`.
2. Traduire les fonctions `nth`, `map` et `fold` en C.
3. A quel moment dans les fonctions que vous venez d'écrire, un GC pourrait se déclencher?

Implantation de continuations en C

On cherche à définir des continuations en C. Une continuation nécessite de conserver d'une part un contexte d'exécution et l'état de la pile, puis lors de son rappel de revenir au contexte d'exécution sauvegardé et de restaurer la pile d'exécution même si celle sauvegardée est plus grande que l'actuelle. On supposera que les registres sont bien traités par les `setjmp/longjmp`.

1. Définir un type C pour les continuations qui contienne au moins un contexte d'exécution (`jmp_buf`) et un champ pour l'état de la pile.
2. Ecrire une fonction qui sauve l'état de la pile dans un continuation, puis écrire une fonction `get_context` qui retourne une continuation en ayant posé un point de retour (`setjmp`). Vérifier le sens de la pile.
3. Ecrire une fonction qui restaure l'état de la pile tel qu'il est conservé dans la continuation. Attention de bien vérifier que le niveau de la pile actuel est suffisant pour la pile à restaurer. Effectuer un `longjmp` à la fin de cette fonction pour retourner au point d'exécution
4. Proposer une compialtion du `call/cc`.

Utilisation des programmes décrits

On trouvera sur les machines `ari-31-312-XY` les programmes suivants :

- `sigint.c` : traite le signal `SIGINT`.
- `sigsegv.c` : détermine le point de rupture.
- `compte_elt.c` : compte les éléments d'une liste et affiche le résultat intermédiaire en cas de signal `SIGINT`.
- `mult_liste_v1.c` : calcule le produit d'une liste d'entiers
- `mult_liste_v2.c` : même programme avec `setjmp/longjmp`.
- `exception.h` : l'entête pour se servir d'exceptions : `TRY`, `EXCEPT`, `ENDTRY`.
- `exception.c` : les déclarations globales et `_RaiseException`.
- `mult_liste_v3.c` : calcule le produit des éléments d'une liste avec exceptions.
- `lastn.c` : retourne les n derniers éléments d'une liste.

dans le catalogue `/users/Enseignants/chaillou/install/td5ex`.
N'hésitez pas à les compiler, les tester, les lire et les modifier!!!