

Série d'exercices #8

IFT-2035

November 13, 2023

8.1 Keep trying

Soit une macro `until` en ELisp, qui prend la forme suivante:

```
(until (CONDITION EXP))
```

Une telle expression devrait répéter `EXP`, jusqu'à ce que `COND` soit différent de `nil`. De plus elle renvoie comme valeur la dernière valeur de `COND`.

Une implantation naïve pourrait être

```
(defmacro until (ce)
  (let ((cond (car ce))
        (exp (cadr ce)))
    `(letrec ((loop (lambda ()
                      (if ,cond
                          ,cond
                          (progn ,exp (funcall loop))))))
      (funcall loop))))
```

Montrer le résultat de l'expansion de la macro pour les usages suivants:

- `(until (exp (setq exp (read-expression))))`
- `(until ((progn (message "new iteration") cond) (setq cond (read-answer))))`
- `(let ((loop nil)) (until (loop (setq loop (read-answer)))))`

Certaines de ces expansions sont incorrectes.

Indiquer précisément chacun des problèmes et montrer quel code aurait dû être renvoyé à la place.

Corriger la définition de la macro pour éliminer ces problèmes.

8.2 Définition par cas

Soit une macro `iftcase` en ELisp qui peut s'utiliser comme suit:

```
(iftcase exp
  ((1 3 5) exp1)
  ((4) exp2)
  (_ exp3))
```

qui signifierait: évalue d'abord `exp` puis évalue `exp1` si `exp` rend 1, 3, ou 5; ou évalue `exp2` si `exp` rend 4; ou évalue `exp3` sinon.

1. Montrer l'expansion désirée pour ce code
2. Identifier les risques possibles d'évaluation répétée excessive et de capture de nom, si la macro est définie trop naïvement.
3. Donner une définition de cette macro qui ne souffre pas de ces problèmes.

8.3 RPN en Lisp

Définir en ELisp la macro `postfix` qui prend une expression sous forme postfixée:

```
(let ((x 5))
  (postfix 1 x + 3 * 2 /))  $\implies$  9
```

Il suffira d'accepter les opérateurs `+`, `-`, `*`, `/`, `not`, `≥`, et `if`.

Montrer les étapes de l'évaluation de `(postfix 1 x + 3 * 2 /)` ci-dessus, jusqu'à l'obtention du résultat 9, en indiquant clairement quelles parties ont lieu lors de la compilation et quelles parties ont lieu à l'exécution.