

プログラム言語論

亀山幸義

筑波大学 情報科学類

No. 5

- ▶ micro ML (4章): 一階、関数型言語
- ▶ 名前なし (5章): 高階、関数型言語
- ▶ micro C (7章): 命令型言語
- ▶ オブジェクト指向言語

micro ML

Sestoft テキストでの micro ML:

- ▶ ML や F# の小さなサブセット
- ▶ 一階の言語 (not 高階言語)
- ▶ 関数型言語 (not 命令型言語)

micro ML の構文

```

type expr =
  | CstI of int   | CstB of bool   | Var of string
  | Let of string * expr * expr
  | Prim of string * expr * expr
  | If of expr * expr * expr
  | Letfun of string * string * expr * expr
  | Call of expr * expr

```

ただし、Call(e1,e2) において e1 は、Var s の形でなければいけない。
(「一階言語である」ことの制限)

```

CstB true
If(Prim("=", Var "x", CstI 20), CstI 30, CstI 40)
  ... if x=20 then 30 else 40
Letfun("f", "x", Prim("+", Var "x", CstI 10),
  Call(Var "f", CstI 20))
  ... let f x = x + 10 in f 20

```

micro ML の特徴

- ▶ 「プログラム=(値を返す)式」である(「プログラム=命令」でない)
- ▶ (再帰)関数の定義、関数呼出しがある
- ▶ 関数そのものは値とならない(変数に格納できない)
- ▶ 副作用はない(変数の値の書換え、入出力)

例:

```
z + 8
let f x = x + 7 in f 2
let f x = if x = 0 then 1 else 2 + f(x-1) in f 7
```

注意点: micro ML での let は再帰関数を定義する (OCaml での let ではなく let rec に相当する)。

関数クロージャの必要性

例:

```
let x = 10 in
let f y = x + y in
let x = 20 in
  f 30
```

質問: f 中の x は 10 なのか 20 なのか?

- ▶ 静的束縛 (lexical, static binding)
- ▶ 動的束縛 (dynamic binding)

静的束縛を実現するには、f が $y \rightarrow x + y$ であるという情報だけでは不足!

関数クロージャの必要性

関数クロージャ(関数閉包, function closure): 関数定義と環境をセットにしたもの。

```
let x = 10 in
let f y = x + y in
let x = 20 in
  f 30
```

上記の f に対応する関数クロージャ

(f y = x + y, [(x,10)])

評価器での表現

```
type value =
  | Int of int
  | Closure of string * string * expr * value env
```

上記の f に対応する関数クロージャ

```
Closure("f", "y", Prim("+", ...), [(x,10)])
```

関数クロージャの必要性

関数クロージャへの対応

```
let rec eval e env =
  match e with
  | CstI i    -> i
  ...
  | Letfun(f, x, fBody, letBody) ->
    let bodyEnv = (f, Closure(f, x, fBody, env)) :: env
    in eval letBody bodyEnv
  | Call(Var f, eArg) ->
    let fClosure = lookup env f in
    match fClosure with
    | Closure (f,x,fBody,fDeclEnv) ->
      let xVal = Int(eval eArg env) in
      let fBodyEnv = (x,xVal)::(f,fClosure)::fDeclEnv in
        eval fBody fBodyEnv
    ...
```

疑問

いろいろな疑問：

- ▶ これで本当に静的束縛になるのか？
- ▶ 動的束縛にするのにはどうしたらよいか？
- ▶ 関数クロージャは、動的に (実行時に) 作られるが、そうすると、メモリを大量に消費するし、遅くなるのではないかと不安ではないか？

課題

今週は、全員必須の宿題は出ませんが、来週の演習にそなえて、スライドをよく読みかえておいてください。

なお、テキストのプログラムは~kam/plm2015/の下に置いてあります。

来週は2コマとも演習なので注意してください。