

プログラム言語論 第8週 補足資料  
MiniML について (改訂版)  
亀山幸義, 2009/06/08

## 1 miniML の構文 (修正版)

変数  $x$  と定数  $c$  の構文は、miniC と同じ (定数は、整数定数と真偽値定数)。

$$\begin{aligned} \text{式 } e ::= & x \mid c \mid e + e' \mid e * e' \mid e = e' \mid e > e' \\ & \mid \text{if } e \text{ then } e \text{ else } e \mid \text{fun } x \rightarrow e \mid e e' \\ & \mid \text{let } x = e \text{ in } e' \mid \text{let rec } x y = e \text{ in } e' \\ & \mid \text{print } e \mid \text{show } e \\ & \mid (e, e') \mid \text{fst } e \mid \text{snd } e \end{aligned}$$

注意すべき点:

- (再帰的) 関数の定義は、`let rec f x = e in e'`
- 関数は 1 引数のものしかない。
- なお、`let f = fun x -> e in e'` とすると  $f$  は再帰的でない関数になる。
- `print e` は、 $e$  の計算結果 (値) を印刷する。
- `show e` は、 $e$  の値によって、「スタックの印刷」か「ヒープの印刷」を行なう。(  $e$  が 0 のとき前者、1 のとき後者。 )
- $(e_1, e_2)$  の形の式は、ペア (対)、あるいは、2 個組 (タプル)。
- ペアから、左要素あるいは右要素を取り出すのは `fst e` あるいは `snd e`。

`print` と `show` の例:

```
print (1, (2, fun x -> x+1));; (プリント文はどんな値でも適用可能)
```

```
show 0 ;; (スタックの表示)
```

```
show 1 ;; (ヒープの表示)
```