

『プログラム言語論』 演習の解答例

亀山

課題 1-1. ファイル ex11.c の関数に、適当な引数 n を与えて MiniC 言語処理系で実行したときのスタックの様子を書きなさい。また、 $n = 5$ のときに処理系が必要とするスタックの深さの最大値 (実行中に最も多くのスタックフレームが積まれる瞬間のスタックフレームの個数) を答えなさい。ただし、整数の計算が「桁あふれ」して異常終了することはないものと仮定する。

解答例 スタックの様子は、MiniC 処理系で show 関数を呼べばよいだけなので省略する。

スタックサイズ (スタックフレームの個数) の最大値は 5 である。

課題 1-2.

MiniC 処理系の 0,1,2,3 の 4 つのモードは、以下のいずれに該当するか調べよ。

- 動的束縛か静的束縛か。
- 複数の引数を持つ関数において、引数を左から右に順番に計算 (評価) するか、右から左に計算するか。

解答例 例題ファイル ex5.c を走らせると、モード 0,2 のとき 30、モード 1,3 のとき 25 が返る。よって、モード 0,2 が動的束縛、1,3 が静的束縛である。

例題ファイル ex6.c を走らせると、モード 0,3 のとき、3,2,1 の順に印刷し、モード 1,2 のとき、1,2,3 の順に印刷する。よって、モード 1,2 が「左から右」、モード 0,3 が「右から左」である。

課題 2-2. (発展課題)

Java, JavaScript, Ruby, Scheme, OCaml, Haskell などの言語のうち 1 つを取りあげ、

- 動的束縛か静的束縛か。
- 値呼びか、名前呼びか、必要呼びか、
- 関数の引数を左から評価するか右からか

について、(文献で調べるのではなく) プログラム例をつくって調べなさい。

解答例 発展課題であるので具体的なプログラム例は省略するが、

JavaScript, Scheme, OCaml, Ruby などでの関数呼び出しは、静的束縛、値呼びである。これらの言語での関数の引数の評価の順番は、かなり微妙だが、OCaml では「カーリー化」という仕組みで複数の引数を持つ関数を実現しており、標準的処理系では「右から左」に評価する。(OCaml の言語仕様上は、C 言語と同様「不定」である。) Scheme 等は「左から右」である。

Haskell は、静的束縛かつ必要呼びである。(必要呼びであるので、「引数をどの順番に評価するか」というのは意味を持たない。)

なお、Java などのオブジェクト指向言語におけるメソッドの呼び出しは、関数呼び出しと似た機能ではあるが、「その関数名が、どの関数定義をあらわすか」という部分が動的に定まるので、(授業で述べたのとは別のところで) 動的束縛の要素をもっている。これについては、オブジェクト指向についての説明をする週 (第 8 週か第 9 週) で解説する予定である。