

『プログラム言語論』 期末試験

2012年 7月 2日

解答用紙は2枚である。各解答用紙の上部に学籍番号と氏名を記入すること。解答用紙の裏面を使用してもよい。問題の順番通りに解答する必要はないが、必ず問題番号(「1-a」など)を記載してから解答を書くこと。

問1. (配点 30点) MiniC 言語で書かれた次のプログラムについて以下の問に答えよ。

```
int x; int y; int w;
int f (int z) {
    print x+y+z; /* C言語の printf("%d\n", x+y+z); と同じ */
    return z;
}
int g (int z) {
    int x;
    x = 100;
    if (z == 0) {
        int y; /* ここで局所変数を導入 */
        y = 200;
        return f(z);
    } else {
        y = 300;
        return f(z);
    }
}
int main () {
    x = 10; y = 20;
    w = g(f(0));
    return g(1);
}
```

上記のプログラムを以下のそれぞれの方式で実行すると、どのような値が印刷されるかを示しなさい。

- 1-a. 静的束縛かつ値呼び
- 1-b. 静的束縛かつ名前呼び
- 1-c. 動的束縛かつ値呼び
- 1-d. 動的束縛かつ名前呼び

1-e. 次に、上記のプログラムを静的束縛かつ値呼び方式で実行した時、`print x+y+z;` が実行される瞬間のスタックの内容を図示しなさい。(print が2回以上実行される場合は、その全ての瞬間のスタックを示しなさい。)

なお、1つのスタックフレームの中身として、局所変数とその値、Access Link, Control Link の3つのデータを書けばよい。また、「/* ここで局所変数を導入 */」というコメントがある部分も、ブロックを構成するものとする。(それを実行すると、スタックフレームが1つ積まれる。)

問 2. (配点 30 点) 次の 2 つの MiniML プログラムについて考える .

```
let rec f n =
  if n < 2 then
    1
  else
    f(n-2)+f(n-1)
in f 5 ;;

let rec g n =
  if n < 2 then
    (1,1)
  else
    let p = g(n-1) in
      ((snd p),(fst p)+(snd p))
in snd (g 5) ;;
```

ただし、授業で使った MiniML 言語は < や - という記号を持たなかったが、ここでは、C 言語や OCaml 言語と同様、大小比較および引き算の記号として使えるものとする。また、MiniML の (a,b) は、対 (ペア) のデータ構造を表す。また、fst と snd は、それぞれ、対の第 1 要素、第 2 要素を取り出す関数である。

これらのプログラムを MiniML で静的束縛かつ値呼び方式で実行する時、以下の問に答えなさい。

- 2-a. 上記の左のプログラムを実行し、スタックが最も深くなった瞬間 (最も多くのスタックフレームが詰められた瞬間) のスタックの中身を図示しなさい。
- 2-b. 上記の右のプログラムを実行すると、その途中で (g 3) という関数呼び出しをする。その関数呼び出しの瞬間と、プログラムが終了する瞬間の、スタックとヒープの中身を図示しなさい。なお、変数の値が決まっていなときは、「未定義 (undefined)」と書きなさい。
- 2-c. 「対」のデータはヒープに作成されるが、もし、これがスタックに作成されたとしたら、どのような問題が発生するか説明せよ。(上記のプログラムを例に取ればよい)。
- 2-d. 上記の 2 つの関数 f と g は同じ計算をするプログラムだが、実行効率が異なる。その理由を簡潔に説明せよ。
- 2-e. 上記の右側のプログラムを、更に次の図の左側のように書きかえた。(参考までに、OCaml での通常の見方をした同等のプログラムを右側に載せている。)

```
let rec h n = fun p ->
  if n < 2 then
    p
  else
    (h (n-1)) ((snd p),(fst p)+(snd p))
in snd ((h 5) (1,1)) ;;

let rec h n p =
  if n < 2 then
    p
  else
    h (n-1) (snd p,(fst p)+(snd p))
in snd (h 5 (1,1)) ;;
```

このような形のプログラムは末尾再帰と呼ばれる。この形式の利点について、簡潔に (2-3 行程度で) 説明せよ。

- 2-f. 上記のプログラムにおける関数 f と g の型を述べよ。ただし、データ a, b が A という型を持つとき、対 (a,b) は、 $A * A$ という型を持つ。(加点問題: 関数 h の型がわかるなら、それも述べると加点する。)

問 3. (配点 25 点) 次のプログラムは, Java 言語で 3 つのクラス Parent, Child, Test を定義したものである (実際には, 3 つは, それぞれ別のファイルに記述されている.)

```
// Parent.java
class Parent {
    public String method1 () {
        return "Parent";
    }
    Parent () {
    }
}

// Child.java
class Child extends Parent {
    public String method1 () {
        return "Child";
    }
    Child () {
    }
}

// Test.java
class Test {
    public static void main(String args[]) {
        Parent p = new Child();
        System.out.println(p.method1());
        method2(p);
    }
    public static void method2(Parent q) {
        System.out.println("No.1: " + q.method1());
    }
    public static void method2(Child q) {
        System.out.println("No.2: " + q.method1());
    }
}
```

Java では, extends というキーワードで継承を表す。よって, Child は Parent の subclasses である。System.out.println は文字列を標準出力へ出力する。p.method1() は, 変数 p に格納されているオブジェクトに対して, method1 というメソッドを呼び出す。

- 3-a. 上記プログラム (Test の main) を実行すると, 何が出力されるかを, 簡単な説明をつけて書きなさい。
- 3-b. この例をもとに, 動的ルックアップとは何か 2-3 行で説明しなさい。
- 3-c. この例をもとに, サブタイピングとは何か 2-3 行で説明しなさい。

問 4. (配点 15 点) 以下の事項から 3 つを選び, それぞれ 3-5 行程度で説明しなさい。

- 4-a. プログラムの静的情報と動的情報について。
- 4-b. 高階関数と関数クロージャについて。
- 4-c. 静的型付けと動的型付けについて。
- 4-d. メモリ管理におけるゴミ集めについて。
- 4-e. 抽象データ型とオブジェクト指向について。

以上.