2017年度 『プログラム言語論』 期末試験 (誤植修正版)

解答用紙は 2 枚であり、表と裏の両方を使用できる。すべての解答用紙の上部に、学籍番号と氏名を明記すること．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 35 点)

授業の演習では、micro-ML 言語 (一階の関数型言語、ML のサブセット) に対する環境渡しインタプリタを用いた。これに関して以下の設問に答えなさい。

1-a. 下の図は, Fibonacci 関数 f (つまり、$f(1) = f(2) = 1$ および $x > 2$ となる x に対して $f(x) = f(x-2)+f(x-1)$ となる関数) を定義して，$f(4)$ を計算するプログラム (左が OCaml, 右が micro-ML) である。ただし、整数の大小比較<=をプリミティブ演算として持つよう拡張されているものとする。

このプログラムを、演習で用いた環境渡し方式のインタプリタ (eval) で実行したとき、$f(4)$ の実行過程で、環境がどのように変化するか書きなさい．つまり，環境が変化するごとに，その環境を書きなさい。ただし、最初の時点では環境は空であるとし、静的束縛かつ値呼びであるとする。環境の表記は自分流でよいが、たとえば、$[x \to 10,\ x \to 20]$ と書けばよい。

```
let rec f x =
  if x <= 2 then 1
  else f (x-2) + f (x-1)
in   f 4
```

```
Letfun("f", "x",
 If(Prim("<=", Var("x"), CstI(2)),
   CstI(1),
   Prim("+",Call(Var("f"),Prim("-",Var("x"),CstI(2))),
            Call(Var("f"),Prim("-",Var("x"),CstI(1))))),
 Call(Var("f"),CstI(4)))
```

1-b. 前問のように静的束縛かつ値呼びで $f(4)$ を実行すると，足し算 (加算 $+$) が合計して何回実行されるか、答えなさい。

1-c. 下のように数学的表記で定義される関数 (を micro-ML で書いたもの) を g および h とする。

$$g(x) = x * x + x * 2 + 1$$
$$h(x) = 13$$

これらの定義のもとに $g(g(4))$ および $h(g(4))$ を実行するとき、(1) 静的束縛かつ値呼びのとき、(2) 静的束縛かつ名前呼びのとき、(3) 静的束縛かつ必要呼びのとき、の 3 通りの場合において、足し算が実行される回数を答えなさい。($g(g(4))$ と $h(g(4))$ の 2 つに対して、3 通りがあるので、合計で 6 個の数を答えることになる。)

1-d. 今度は、micro-ML を、高階関数を持つ言語に拡張した言語を考える．これを micro-ML+と呼ぶことにして，その言語で記述された以下のプログラムの実行について考える。 (左は OCaml での記述，右は micro-ML+での記述)

```
let x = 3 in
let f y = x + y in
let x = 5 in
let g z = x + (f z) in
let x = 7 in
    f (g 20)
```

```
Let("x", CstI(3),
Letfun("f","y",Prim("+",Var("x"),Var("y")),
Let("x", CstI(5),
Letfun("g","z",Prim("+",Var("x"),Call(Var("f"),Var("z"))),
Let("x", CstI(7),
 Call(Var("f"),Call(Var("g"),CstI(20))))))))
```

このプログラムを静的束縛かつ値呼びで実行する．最後の行である f(g 20) を実行する直前の環境，および，上記プログラムの実行結果 (返す値) を書きなさい．関数クロージャは Closure(...) といった形で適宜書けばよい．

1-e. 前問のプログラムを、動的束縛かつ値呼びで実行したとき、計算結果が何になるか答えなさい。(環境を書く必要はない．)

問 2. (配点 26 点)

Java は静的型付けを行なうオブジェクト指向言語である．下記の Java プログラムについて考えなさい。ただし、String は文字列型を表し、(後のプログラムで) System.out.println(s) は 文字列 s を印刷するメソッドである．ClassB は ClassA を継承している。toString, foo 等はメソッドの名前である。public というキーワードは無視してよい．

```
class ClassA {
  public String toString () { return "ClassA";    }
  public String foo ()      { return "A-foo";     }
  public String goo ()      { return "A-goo";     }
  ClassA ()                 {} // construcing an object of ClassA
}
class ClassB extends ClassA {  // inheritance
  public String toString () { return "ClassB";    }  // overwrite
  public String hoo ()      { return "B-hoo";     }
  public String foo (String s){ return "B-foo";   }  // overload
  ClassB ()                 {} // construcing an object of ClassB
}
```

2-a. 上記プログラムとあわせて以下のプログラムを実行する．以下のプログラムで testN-M というコメントがあるすべての行について、(1) コンパイルエラーを起こす，(2) 実行時エラーを起こす，(3) エラーは起こさない，のいずれであるか、また、System.out.println がある行がエラーを起こさない場合、何が印刷されるかを答えなさい。なお、たとえば test2-1 がコンパイルエラーのときは、その行と一緒にコンパイルする必要がある test2-2 などの行もコンパイルエラーと判断しなさい。

```
class Test1 {
  public static void main(String args[]) {
  ClassA  a;  ClassB  b;      // declare variables
  ClassA  a2;  ClassB  b2;
  a = new ClassA();   // create an object of ClassA and store it in a
  b = new ClassB();
  System.out.println(b.foo());       // test1-1
  System.out.println(b.foo("abc"));  // test1-2
  System.out.println(b.goo());       // test1-3
  a2 = b;                            // test2-1
  System.out.println(a2.toString()); // test2-2
  System.out.println(a2.foo());      // test2-3
  System.out.println(a2.hoo());      // test2-4
  b2 = a;                            // test3-1
  System.out.println(b2.toString()); // test3-2
  System.out.println(b2.foo());      // test3-3
  System.out.println(b2.hoo());      // test3-4
} }
```

2-b. 上記プログラム例 (の適当な部分) を用いて，動的ルックアップとサブタイピングとは何かを、簡潔に説明しなさい．

問 3. (配点 15 点)

以下のプログラムに、型がつくか (型を整合的に与えることができるか) どうかを答えなさい。(YES/NO だけでなく、その理由と、型がつく場合はその型を答えなさい。) ただし、真理値型を `bool`, 整数型を `int` と記しなさい。

3-a. 以下の式の型付け (ヒント: この式は関数 f そのものを返そうとしているので、型がつくとすれば $A \to B$ の形である。)

(micro-ML+の構文)

```
Letfun("f","x",
       Prim("*",CstI(2),Call(Var("f"),Prim("+",Var("x"),CstI(3)))),
       Var("f"))
```

(OCaml の構文) `let rec f x = 2 * f (x + 3) in f`

3-b. 以下の式の型付け

(micro-ML+の構文) `Letfun("f","x",Prim("+",Var("x"),CstI(1)),Call(Var("f"),Var("f")))`

(OCaml の構文) `let f x = x + 1 in f f`

3-c. 以下の式の型付け (型がつくとすれば多相型を使う)

(micro-ML+の構文)

```
Letfun("f","x",Var("x"),
 If(Call(Var("f"),CstB(true)),
    Call(Var("f"),CstI(10)),
    Call(Var("f"),CstI(20))))
```

(OCaml の構文) `let f x = x in if (f true) then (f 10) else (f 20)`

問 4. (配点 24 点)

以下の問のうち 3 つ以上に、それぞれ、3 行以上答えなさい。(4 つとも答えた場合は、点数が良いものから 3 つを選ぶ。)

4-a コンパイラとインタプリタの定義を述べた上で、コンパイラで作成したコードの実行がインタプリタでの実行より高速であることが多い理由を説明しなさい。

4-b. 末尾呼び出しの関数の例を 1 つあげた上で (例を書く言語は OCaml でも micro-ML でも C 言語でも Lisp でもよい)、末尾呼び出しとは何か、また、末尾呼び出しはどのように有用なのかを簡潔に述べなさい。

4-c. 抽象データ型の概念と情報隠蔽について説明しなさい。また、これらが、どのように有用であるかを簡潔に述べなさい。

4-d. 多相型の一種であるパラメータ多相とサブタイプ多相について説明しなさい。また、これらが、どのように有用であるかを簡潔に述べなさい。

以上.