

『プログラム言語論』 期末試験

2014年6月26日(木)

解答用紙は2枚であり、表と裏の両方を使用できる。2枚とも、学籍番号と氏名を記入すること。問題用紙は持ち帰ってよいので下書き等のために使ってよい。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 z;
int f (int x) {
    print x;          /* C言語の printf("%d\n", x); と同じ*/
    return x;
}
int g (int x, int y) {
    z = 40;
    return x * x * x;
}
int main () {
    int z;
    z = 30;
    print g(f(10),f(20));
    print z;
}
```

上記のプログラムを以下のそれぞれの方式で実行したとき、印刷される値を、印刷される順番に示しなさい。

- 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. 静的束縛かつ必要呼び
- 1-d. 動的束縛かつ値呼び

MiniC 言語における $f(e)$ という形の関数呼出し (関数 f を実引数 e に適用) に対して、以下の文がそれぞれ正しいかどうか判定し、その根拠を簡潔に (1-2行で) 書きなさい。

- 1-e. 関数 f の本体で引数が使われていないとき、名前呼び戦略と必要呼び戦略においては、 e の計算を行わずに済むため、値呼び戦略より有利である。
- 1-f. 関数 f の本体で引数が2回以上使われるとき、名前呼び戦略と必要呼び戦略においては、 e の計算を2回以上行うため、値呼び戦略より不利である。

問 2. (配点 30 点、加点問題あり) 次の抽象構文で定義される言語 L に対する抽象機械について考える。

$$e ::= \text{Var}(x) \mid \text{Int}(n) \mid \text{Plus}(e, e) \mid \text{Let}(x, e, e) \mid \text{Lam}(x, e) \mid \text{App}(e, e)$$

ただし、 x は変数を表し、 n は整数定数 (-1 や 3 など) を表し、また、授業で定義した言語から $\text{Times}(e, e)$ を除いた。

この抽象機械は CEK 機械の一種であり、状態は、 e (初期状態)、 $\text{eval}\langle e \mid E \mid K \rangle$ の形 (eval 状態)、 $\text{apply}\langle K \mid E \rangle$ の形 (apply 状態)、 v (最終状態) のいずれかである。この状態の間の遷移は以下の規則で与えられる。

$$\begin{aligned} e &\rightarrow \text{eval}\langle e \mid [] \mid \text{init} \rangle \\ \text{eval}\langle \text{Var}(x) \mid E \mid K \rangle &\rightarrow \text{apply}\langle K \mid v \rangle \quad (v = \text{lookup}(x, E)) \\ \text{eval}\langle \text{Int}(n) \mid E \mid K \rangle &\rightarrow \text{apply}\langle K \mid n \rangle \\ \text{eval}\langle \text{Plus}(e, e') \mid E \mid K \rangle &\rightarrow \text{eval}\langle e \mid E \mid \text{plus1}(e', E)::K \rangle \\ \text{eval}\langle \text{Let}(x, e, e') \mid E \mid K \rangle &\rightarrow \text{eval}\langle e \mid E \mid \text{letin}(x, e', E)::K \rangle \\ \text{eval}\langle \text{Lam}(x, e) \mid E \mid K \rangle &\rightarrow \text{apply}\langle K \mid \text{Closure}(x, e, E) \rangle \\ \text{eval}\langle \text{App}(e, e') \mid E \mid K \rangle &\rightarrow \text{eval}\langle e \mid E \mid \text{apply1}(e', E)::K \rangle \\ \text{apply}\langle \text{plus1}(e, E)::K \mid v \rangle &\rightarrow \text{eval}\langle e \mid E \mid \text{plus2}(v)::K \rangle \\ \text{apply}\langle \text{plus2}(n_1)::K \mid n_2 \rangle &\rightarrow \text{apply}\langle K \mid n \rangle \quad (n_1 + n_2 = n) \\ \text{apply}\langle \text{letin}(x, e, E)::K \mid v \rangle &\rightarrow \text{eval}\langle e \mid E[x = v] \mid K \rangle \\ \text{apply}\langle \text{apply1}(e, E)::K \mid v \rangle &\rightarrow \text{eval}\langle e \mid E \mid \text{apply2}(v)::K \rangle \\ \text{apply}\langle \text{apply2}(\text{Closure}(x, e, E))::K \mid v \rangle &\rightarrow \text{eval}\langle e \mid E[x = v] \mid K \rangle \\ \text{apply}\langle \text{init} \mid v \rangle &\rightarrow v \quad (\text{最終状態}) \end{aligned}$$

ここで、 $\text{init}, \text{lookup}(x, E), E[x = v]$ は環境 E に対する操作、 $+$ は加算、 $X::K$ はスタック K に対する操作、値 v は、整数 n か、 $\text{Closure}(x, e, E)$ の形をした式であり、すべて授業で与えた通りである。

このとき、以下のプログラムを初期状態として、最終状態に至るまでの状態遷移列を書きなさい。

2-a. $\text{Plus}(\text{Int}(10), \text{Int}(20))$

2-b. $\text{Let}(x, \text{Int}(10), \text{Plus}(\text{Var}(x), \text{Int}(20)))$

2-c. $\text{App}(\text{Lam}(x, \text{App}(\text{Var}(x), \text{Int}(10))), \text{Lam}(y, \text{Plus}(\text{Var}(y), \text{Int}(20))))$

また、次の問題に答えなさい。

2-d. $\text{Closure}(x, e, E)$ は関数クロージャと呼ばれる。関数クロージャを必要とする (それがないと、うまく実行できない) プログラムの例を 1 つあげなさい。なお、このプログラムに対する計算 (状態遷移列) を書く必要はない。

2-e. 上の抽象機械は、値呼び、名前呼び、必要呼びのどの方式に対応したものであるか、簡潔な根拠とともに述べよ。

2-f. (加点問題) 上の言語に、「引数 e に 1 を足す」ことを意味する $\text{Inc}(e)$ という式を加えたい。たとえば、 $\text{Inc}(\text{Int}(17))$ の実行が 18 という最終状態で終わるようにしたい。このために、どのような状態遷移を追加すればよいか、示しなさい。

問 3. (配点 40 点、加点問題あり)

以下の 4 つの事項のすべてについて、それぞれ 5 行程度で説明しなさい。

- 3-a. MiniC 言語 (あるいは C, Java, ML などの言語) において、以下の情報が静的に決定できるかどうか答えた上で、プログラムにおける「静的情報」とは何かを説明しなさい。(i) 関数の実行において、その引数の値が実際に 1 回以上使われるかどうか。(ii) 整数に対する足し算の実行において、その引数が実際に整数であるかどうか。
- 3-b. ヒープとは何か、また、プログラム言語の処理系がヒープを必要とする具体的な理由は何か、そして、ヒープの適切な管理のためにプログラム言語の処理系が必要とする機能は何か、について、説明しなさい。
- 3-c. 抽象データ型と情報の隠蔽について、説明しなさい。
- 3-d. オブジェクト指向言語における override (メソッドの上書き) とは、どのような機能であるか、またどのような利点があるか、利用にあたって注意すべき点は何か、など、を説明しなさい。(overload について言及する必要はない。)

また、以下の事項のうち 1 つ以上を選択して、5 行程度で説明しなさい。(2 つ以上を選択して解答した場合は、特に良い解答の時に加点する。)

- 4-e. 高階関数、その利点、その処理 (ML など、具体的な言語について論じてもよい)
- 4-f. 静的型付けと動的型付け (Java と Ruby など、具体的な言語について論じてもよい)
- 4-g. 多相型 (polymorphic type)
- 4-h. オブジェクト指向言語における動的ルックアップ (Java など、具体的な言語について論じてもよい)
- 4-i. オブジェクト指向言語における継承とサブタイピング (Java など、具体的な言語について論じてもよい)
- 4-j. 汎用言語とドメイン特化言語 (DSL)
- 4-k. 末尾再帰とは何か、その利点、その処理

以上.