

言語処理工学A 期末テスト 2009/01/30 井上克郎

教科書、ノート、その他持ち込み なし！ ただし、[1]は解答用紙表紙に、[2]は内側の2ページに、[3]、裏表紙に書け。

[1] 次のプログラムの実行環境について答えよ(表紙のページに答えを書け)

```

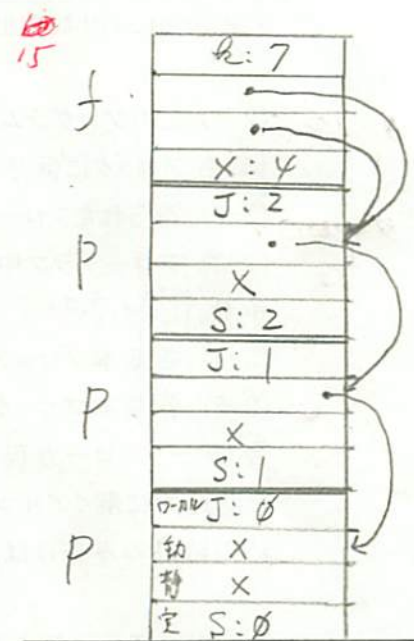
program main(input,output);

  procedure p(s:integer);
    j: integer;

    function f(x:integer);
      k: integer;
      begin
        k:= x + 1+j; { ← Here}
        return(k)
      end;

    begin { of p }
      j := s;
      if s = 2 then j:= f(s*2)
        else p(s+1);
      writeln(j);
    end; { of p}

  begin { of main}
    p(0)
  end.
  
```



(1-1) このプログラムの出力を書け。 ⇒

10 { 7, 2, 1, 2, 0 }

(1-2)このプログラムの実行時、最初に Here を通過した直後の、スタックの中の各呼出のフレーム(駆動レコード)を次の要領で書け。

ただしフレーム内の配列はいろいろあり

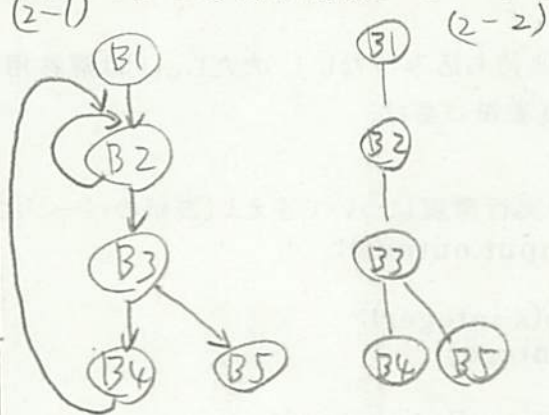
スタック上での各フレームの重なり(ただし、各フレームの持ち主の手続き/関数の名前を書くこと、そして、フレーム内の以下の情報を書くこと

- 実引数 ([変数名:値] の組で書く)
- 動的リンク(ポインタの指すところを有向辺で書け)
- 静的リンク(ポインタの指すところを有向辺で書け)
ただし大域変数の環境へのポインタは×で書くこと
- ローカル変数 ([変数名:値] の組で書く)

他の情報は不要。

[2] 次の3番地コードに関して答えよ。(解答用紙2-3ページに答えを書くこと)

B1	① s = 0 ② a = 1 ③ b = 2
B2	④ L1: t = a * b ⑤ s = s + t ⑥ if s <= 10 goto L1
B3	⑦ if a < 10 goto L2
B4	⑧ a = a + 1 ⑨ goto L1
B5	⑩ L2: print S



(注: print は引数を入力するマクロで、引数を参照するのみで変更しない)

- 8 ~~10~~ (2-1) このプログラムの基本ブロックに分け、フローグラフを書け。プログラムの上から順に、各ブロックに番号 B1~をつけ、各ブロックに入るコードを行番号で明示せよ。
- 8 ~~10~~ (2-2) 得られたフローグラフの支配木 (dominator tree) を書け。
- 10 (2-3) フローグラフ中の各バックエッジを挙げ、それぞれが構成するループの各頂点を示せ。 $\left\{ (B2 \rightarrow B2 : B2), (B4 \rightarrow B2 : B2, B3, B4) \right\}$
- 10 (2-4) 各基本ブロックの Gen 集合、Kill 集合を求めよ。
- 15 (2-5) 各基本ブロックの入り口 (IN)、出口 (OUT) で出現しうる変数定義の行番号の集合 (データフロー方程式の解) を示せ。プリントでは、データフロー方程式を立てて、それを漸近的に解くアルゴリズムを示したが、ここでは、解く途中経過を示す必要はない。最終の結果のみ書けばよい。

[3] 次の問いに答えよ(解答用紙の裏表紙に答えを書くこと)

10 (3-1) 設問(2-5)の結果のデータフロー解析の結果を用いることによって可能になる最適化の一つを説明せよ。具体的に何行目の文がこう変わる、というふうにプログラムの変化を示すとともに、なぜ、それが可能なのか、説明せよ。

10 (3-2) 動的変数領域(ヒープ)とは何か、また、どのように管理しなければならないか説明せよ。

(2-4)

	GEN	KILL
B1	{①, ②, ③}	{⑤, ⑧}
B2	{④, ⑤}	{①}
B3	\emptyset	\emptyset
B4	{⑧}	{②}
B5	\emptyset	\emptyset

(2-5)

	IN	OUT
B1	\emptyset	{①, ②, ③}
B2	{①, ②, ③, ④, ⑤, ⑧}	{②, ③, ④, ⑤, ⑧}
B3	{②, ③, ④, ⑤, ⑧}	{②, ③, ④, ⑤, ⑧}
B4	{②, ③, ④, ⑤, ⑧}	{③, ④, ⑤, ⑧}
B5	{②, ③, ④, ⑤, ⑧}	{②, ③, ④, ⑤, ⑧}

(3-1)

④ $L1: t = a * 2 \Rightarrow L1: t = a * 2 \Rightarrow L1: t = a + a$
 又は $L1: t = \text{シフト}(a)$
 \therefore B2のINで a に関する定数は③のみ。従って定数の伝播の最適化で a を 2 に書きかえられ、
 (更に $a * 2$ は軽い計算のたし算, 又はシフト命令に置きかえられる。③も以降で a が参照されないため前段で置き)

(3-2)

ヒープは, Cの malloc や Pascalの New等の実行時に要求される変数を置く場所。実行前は日付まで分かっている。実行中にヒープの領域が不足した場合, ガベージコレクションを行ったり, 不要なものを削除して領域を再利用したりすることで効率の良い利用手法が必要になる。