

プログラミング言語論 2021 年度 小テスト 解答例

問1 Little Quilt 言語の以下の各式 (1), (2), (3) が表すキルトを図示せよ。a, b, turn, sew の意味は以下の通りである。その他の Little Quilt 言語の構文要素 (let 式、val 宣言、fun 宣言) の意味は講義で説明したものとする。

- a, b は図 1, 図 2 のキルトを表す。

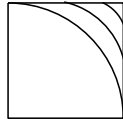


図 1: a が表すキルト

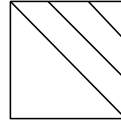
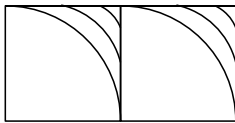


図 2: b が表すキルト

- turn (e) : キルト e を 90 度右回転させたキルトを表す。
- sew (e_1, e_2) : 高さが同じキルト e_1, e_2 を左右に並べ、縫い合わせたキルトを表す (左が e_1 、右が e_2)。

(1) sew (a, a)



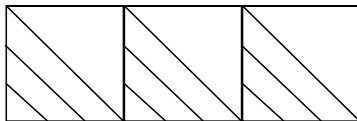
(2) let

```
    val x = turn (turn (b))
```

```
in
```

```
    sew (sew (x,x), x)
```

```
end
```



(3) let

```
    fun unturn (x) = turn (turn (turn (x)))
```

```
    fun pile (x,y) = unturn (sew (turn (y), turn (x)))
```

```
    val aa = sew (a, a)
```

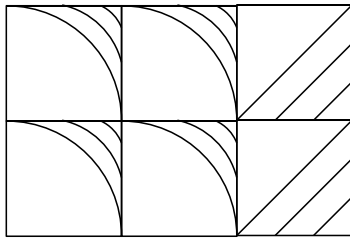
```
    val bb = sew (b, b)
```

```
    val aaaa = pile (aa, aa)
```

```
in
```

```
    sew (aaaa, turn (bb))
```

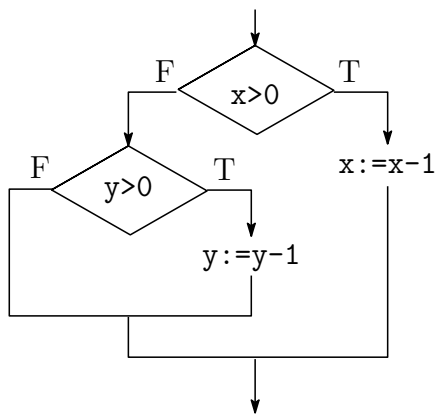
```
end
```



問2 命令型言語の制御フローについて以下の各問に答えよ。

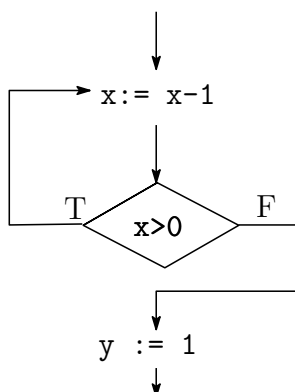
(1) 以下のプログラム断片の制御フローを図示せよ。

```
if x>0 then x := x - 1
else if y>0 then y := y - 1
```



(2) 以下のプログラム断片の制御フローを図示せよ。

```
L: x := x - 1;
if x>0 then goto L;
y := 1
```



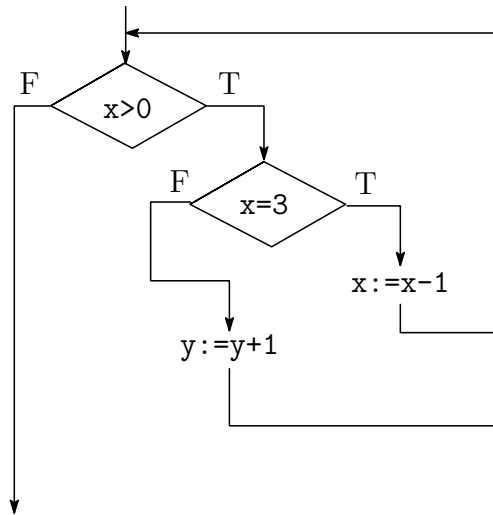
(3) 以下のプログラム断片の制御フローを図示せよ。

```
while x>0 do
begin
```

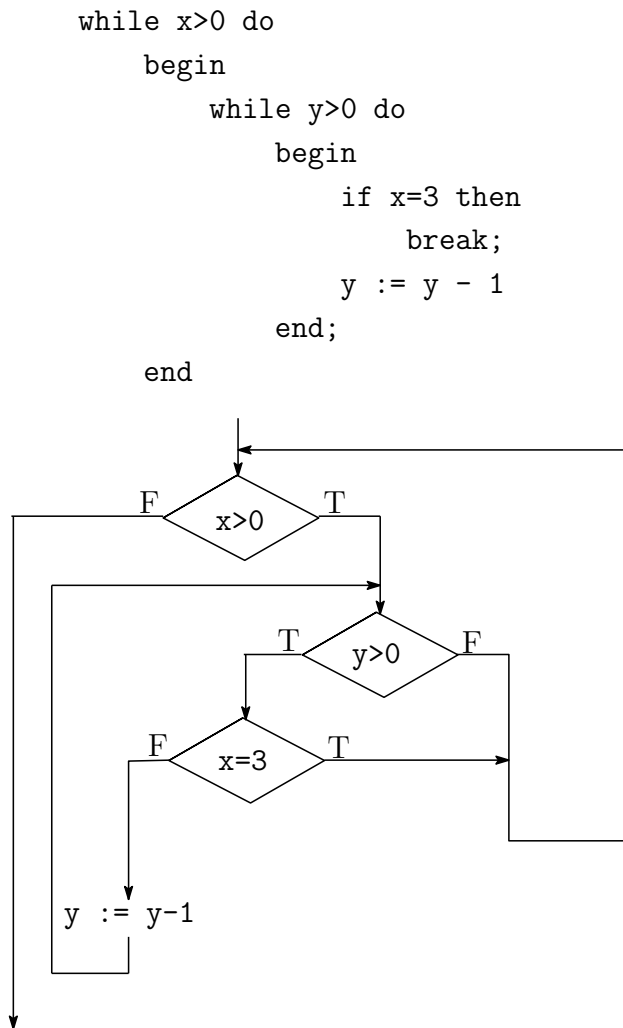
```

    if x=3 then
      begin
        x := x - 1;
        continue
      end;
    y := y + 1;
  end

```



(4) 以下のプログラム断片の制御フローを図示せよ。



(5) (4) のプログラム断片における if 文 (if x=3 then break;) の入口、出口はそれぞれいくつが答えよ。

入口 1 つ、出口 2 つ

問 3 以下の Hoare triple(1), (2) を講義中に提示した規則を使って導け。

(1) $\{a = 40\} a := a - 20 \{a = 20\}$

$$\frac{a = 40 \Rightarrow a - 20 = 40 \quad \frac{}{\{a - 20 = 40\} a := a - 20 \{a = 40\}} \text{(assign)} \quad a = 40 \Rightarrow a = 40}{\{a = 40\} a := a - 20 \{a = 20\}} \text{(conseq)}$$

授業時にも言った通り、上記証明の中で、 $a = 40 \Rightarrow a = 40$ の部分は $\Rightarrow$ の左右が同じなので省略し、以下のように書いてもよい。

$$\frac{a = 40 \Rightarrow a - 20 = 40 \quad \frac{}{\{a - 20 = 40\} a := a - 20 \{a = 40\}} \text{(assign)} \quad \text{(conseq)}}{\{a = 40\} a := a - 20 \{a = 20\}}$$

(2) $\{a = 3\} a := a + 1; a := a + 3 \{a = 7\}$

$$\frac{\frac{a = 3 \Rightarrow a + 1 = 4 \quad \overline{\{a + 1 = 4\} a := a + 1 \{a = 4\}} \text{ (assign)}}{\{a = 3\} a := a + 1 \{a = 4\}} \text{ (conseq)} \quad \frac{\frac{a = 4 \Rightarrow a + 3 = 7 \quad \overline{\{a + 3 = 7\} a := a + 3 \{a = 7\}} \text{ (assign)}}{\{a = 4\} a := a + 3 \{a = 7\}} \text{ (conseq)}}{\{a = 3\} a := a + 1; a := a + 3 \{a = 7\}} \text{ (composition)}$$

この証明では、2カ所の consequence rule の適用のところで、 $a = 4 \Rightarrow a = 4$ と $a = 7 \Rightarrow a = 7$ を省略した。

(3) $\{a = 4\} \text{ if } a = 4 \text{ then } a := a + 3 \text{ else } a := a - 3 \{a = 7\}$

スペースの関係で、3 つに分けて記述する。

$$\frac{\frac{\text{この部分は下に記述}}{\{a = 4 \wedge a = 4\} a := a + 3 \{a = 7\}} \text{ (conseq)} \quad \frac{\frac{\text{この部分は下に記述}}{\{a = 4 \wedge \neg a = 4\} a := a - 3 \{a = 7\}} \text{ (conseq)}}{\{a = 4\} \text{ if } a = 4 \text{ then } a := a + 3 \text{ else } a := a - 3 \{a = 7\}} \text{ (conditional)}$$

$$\frac{a = 4 \wedge a = 4 \Rightarrow a + 3 = 7 \quad \overline{\{a + 3 = 7\} a := a + 3 \{a = 7\}} \text{ (assign)}}{\{a = 4 \wedge a = 4\} a := a + 3 \{a = 7\}} \text{ (conseq)}$$

$$\frac{a = 4 \wedge \neg a = 4 \Rightarrow a - 3 = 7 \quad \overline{\{a - 3 = 7\} a := a - 3 \{a = 7\}} \text{ (assign)}}{\{a = 4 \wedge \neg a = 4\} a := a - 3 \{a = 7\}} \text{ (conseq)}$$

(4) $\{a = 2\} \text{ while } a < 5 \text{ do } a := a + 1 \{a = 5\}$

スペースの関係で、2カ所に分けて記述する。

$$\frac{\frac{\text{この部分は下に記述。}}{a = 2 \Rightarrow a \leq 5 \quad \overline{\{a \leq 5\} \text{ while } a < 5 \text{ do } a := a + 1 \{a \leq 5 \wedge \neg a < 5\}}} \quad a \leq 5 \wedge \neg a < 5 \Rightarrow a = 5}{\{a = 2\} \text{ while } a < 5 \text{ do } a := a + 1 \{a = 5\}} \text{ (conseq)}$$

$$\frac{\frac{a \leq 5 \wedge a < 5 \Rightarrow a + 1 \leq 5 \quad \overline{\{a + 1 \leq 5\} a := a + 1 \{a \leq 5\}} \text{ (assign)}}{\{a \leq 5 \wedge a < 5\} a := a + 1 \{a \leq 5\}} \text{ (conseq)} \quad a \leq 5 \Rightarrow a \leq 5}{\{a \leq 5\} \text{ while } a < 5 \text{ do } a := a + 1 \{a \leq 5 \wedge \neg a < 5\}} \text{ (while)}$$

この下側の証明図でも、 $a \leq 5 \Rightarrow a \leq 5$ の部分は $\Rightarrow$ の左右が同じなので、省略して以下のように書いてよい。

$$\frac{\frac{a \leq 5 \wedge a < 5 \Rightarrow a + 1 \leq 5 \quad \overline{\{a + 1 \leq 5\} a := a + 1 \{a \leq 5\}} \text{ (assign)}}{\{a \leq 5 \wedge a < 5\} a := a + 1 \{a \leq 5\}} \text{ (conseq)}}{\{a \leq 5\} \text{ while } a < 5 \text{ do } a := a + 1 \{a \leq 5 \wedge \neg a < 5\}} \text{ (while)}$$

上記証明において、assignment axiom を assign, consequence rule を conseq, while rule を while, composition rule を composition, conditional rule を conditional と略記した。

問4 以下のように Pascal 言語で関数 sum が宣言されている状況下において、関数呼び出し式 sum(5) の値を書け。

```

function sum(n:integer):integer;
begin
    if n=0 then sum:=0 else sum:=n+sum(n-1);
end;

```

15

問5 以下の Pascal プログラムを実行したときの出力結果を示せ。手続きの仮引数に var がついている場合、call by reference であることを表す。writeln は引数の値を出力後改行する。

```

program test;
var x : integer;
var y : integer;
procedure swap
    (var x: integer;
     var y : integer);
var z : integer;
begin
    z := x; x := y; y := z
end;
                                begin
                                x := 3;
                                y := 4;
                                swap (x,y);
                                writeln (x);
                                writeln (y)
                                end.

```

4

3