完全正当性(total correctness)のための証明規則

while規則(while rule)を以下のように変更する

（他の規則はそのまま）

$$\frac{\langle\!\langle\ \eta \land B \land 0 \leq E = E_0\ \rangle\!\rangle\ C\ \langle\!\langle\ \eta \land 0 \leq E < E_0\ \rangle\!\rangle}{\langle\!\langle\ \eta \land 0 \leq E\ \rangle\!\rangle\ \mathrm{while}\ B\ \{C\}\ \langle\!\langle\ \eta \land \neg B\ \rangle\!\rangle}$$

ただし,

E: プログラム変数を含む, 自然数を値とする式,

Cの実行で値が減少する

E_0: 論理変数, Cの実行直前のEの値を示す

大前提として, 変数の値は全て整数型

前件《 $\eta \wedge B \wedge E = E_0$ 》C《 $\eta \wedge 0 \leq E < E_0$ 》について

ここではE, E_0は自然数, $<$は自然数上の大小関係を用いたが,
一般的にはE, E_0はどのような領域Xの要素でもよく, “$<$”はX上での二項関係で
「整礎関係」(well-founded relation)であるとする.

集合X上の二項関係$R(x, y)$が整礎であるとは,
$R(x_i, x_{i+1})$であるような無限列$x_0, x_1, \ldots, x_i, \cdots \in X$(無限下降列)が存在しない

例:　集合N上の2項関係“$>$”は整礎である.
　　集合Z上の2項関係“$>$”は整礎でない.

問題

集合N×N上の2項関係$<_2$を

$(x_1, y_1) <_2 (x_2, y_2) \Leftrightarrow y_1 < y_2$または $y_1 = y_2$かつ $x_1 < x_2$

(ここで“$<$”は自然数上の通常の不等号)

と定義するとこれは整礎である.

このことを証明せよ.

(ヒント:無限下降列が存在しない(存在すると矛盾する)

ことを示す)

$(x_1, y_1) <_2 (x_2, y_2) \Leftrightarrow y_1 < y_2$または$y_1 = y_2$かつ$x_1 < x_2$

$(0, 0) <_2 (1, 0) <_2 (2, 0) <_2 (3, 0) <_2 \ldots$

$<_2 (0, 1) <_2 (1, 1) <_2 (2, 1) <_2 \ldots$

$<_2 (0, 2) <_2 \ldots$

演習

x, y, zを自然数とする.

このとき以下を証明せよ.

$$\vdash_{tot} \langle\!\langle\, y=1 \wedge z=0 \,\rangle\!\rangle \text{ while } (z!=x) \{z=z+1;y=y*z\} \langle\!\langle\, y=x! \,\rangle\!\rangle$$

Eとして何をとればいいか?

$\vdash_{par} \llparenthesis\, y=1 \wedge z=0 \,\rrparenthesis$ while $(z!=x)$ $\{z=z+1; y=y*z\}$ $\llparenthesis\, y=x! \,\rrparenthesis$

の証明

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{\llparenthesis y(z+1)=(z+1)! \rrparenthesis \; z=z+1 \; \llparenthesis yz=z! \rrparenthesis}{\llparenthesis y=z! \wedge z \neq x \rrparenthesis \; z=z+1 \; \llparenthesis yz=z! \rrparenthesis}
      \qquad
      \llparenthesis yz=z! \rrparenthesis y=y*z \llparenthesis y=z! \rrparenthesis
    }{\llparenthesis y=z! \wedge z \neq x \rrparenthesis \; z=z+1; y=y*z \llparenthesis y=z! \rrparenthesis}
  }{\llparenthesis y=z! \rrparenthesis \; \text{while } (z!=x)\{z=z+1;y=y*z\} \; \llparenthesis y=z! \wedge z=x \rrparenthesis}
}{\llparenthesis\, y=1 \wedge z=0 \,\rrparenthesis \; \text{while } (z!=x)\{z=z+1;y=y*z\} \; \llparenthesis\, y=x! \,\rrparenthesis}
$$

演習課題

以下を証明せよ. ただしx, yは自然数を領域とするプログラム変数,

x_0は自然数を領域とする論理変数とする.

$$\vdash_{tot} ⦅x=x_0 \wedge y=1⦆ \text{while } (x!=0)\ \{y=y*x;\ x=x-1\} ⦅y=x_0!⦆$$

⊢par ⦇$x=x_0 \land y=1$⦈while (x!=0) {y=y*x; x=x-1}⦇$y=x_0!$⦈ の証明

⦇$(x-1)!yx=x_0!$⦈ $y=y*x$⦇$(x-1)!y=x_0!$⦈

⦇$x!y=x_0! \land x\neq 0$⦈$y=y*x$⦇$(x-1)!y=x_0!$⦈　　　⦇$(x-1)!y=x_0!$⦈ $x=x-1$ ⦇$x!y=x_0!$⦈

⦇$x!y=x_0! \land x\neq 0$⦈ y=y*x; x=x-1 ⦇$x!y=x_0!$⦈

⦇$x!y=x_0!$⦈ while (x!=0){y=y*x; x=x-1}⦇$x!y=x_0! \land x=0$⦈

⦇$x=x_0 \land y=1$⦈ while (x!=0){y=y*x; x=x-1} ⦇$y=x_0!$⦈

Hoare論理の拡張

「古典的な」Hoare論理では扱えなかったプログラム構成要素

- 配列等のデータ構造
- goto文
- 変数宣言
- 手続き,関数 – 再帰的呼出し

ここでは配列の取扱いについて述べる.

配列

配列とは

変数に通し番号がついているものの様にも見える

$a[0], a[1], a[2], \ldots$ ←比較→ $a_0, a_1, a_2, \ldots$

$⟪\phi[t/a[2]]⟫\ a[2]=t⟪\phi⟫$ ←比較→ $⟪\phi[t/a_2]⟫\ a_2:=t⟪\phi⟫$

$⟪4=4⟫a[2]=4\ ⟪a[2]=4⟫$ ←比較→ $⟪x=1 \wedge a[2]=4⟫\ a[x+1]=4\ ⟪x=1 \wedge a[2]=4⟫$

$\because (a[2]=4)[4/a[x+1]] \equiv a[2]=4$ 代入できない!

配列

配列aに対してa(t; i)という表記を導入

配列aのi番目の要素a[i]をtに変更して得られる配列全体

$$a(t;\ i)[j] = \begin{cases} t & \text{if } i=j \\ a[j] & \text{if } i \neq j \end{cases}$$

配列の代入公理

$$⟪\ \phi[a(t;\ i)/a]\ ⟫\ a[i]=t\ ⟪\ \phi\ ⟫$$

$$(\text{cf. } ⟪\ \phi[t/a]\ ⟫ a=t ⟪\ \phi\ ⟫)$$

演習

$$\vdash_{par} (\!| a[i]=y \land a[i+1]=z |\!) x=a[i];\ a[i]=a[i+1];\ a[i+1]=x (\!| a[i+1]=y \land a[i]=z |\!)$$

を検証せよ.

演習

$$\vdash_{par} (\!(i=0)\!)\ \text{while}\ i \neq n \{ b[i]=a[i];\ i=i+1 \} (\!(\forall j (0 \leq j < n \supset a[j]=b[j]))\!)$$

を検証せよ.

演習

配列要素の入替（バブルソートの一部）

《$a[i+1] \leq a[i] \wedge \forall k<i.\, a[k] \leq a[i]$》 t1=a[i]; a[i]=a[i+1]; a[i+1]=t1 《$\forall k<i+1.\, a[k] \leq a[i+1]$》

演習

次のバブルソートプログラムを検証せよ.

$\langle\!\langle \forall i \le n.a[i]=b[i] \rangle\!\rangle$ P $\langle\!\langle A_0(a, b, n, 0) \rangle\!\rangle$

```
P::
    j=n;
    while j>0 {
            i=0;
                while i<j {
                        if a[i+1]<a[i]{ t1=a[i]; a[i]=a[i+1]; a[i+1]=t1};
                        i=i+1}
            j=j-1
            }
}
```

$A_0(a, b, n, j)$は以下の条件が成立つことを示す.

b[0], ... , b[n]はa[0], ... , a[n]の置換になっている. (bは論理変数:実行開始時の配列の値)

a[j+1], ... , a[n]の範囲はソートされている.

$0 \le h \le j$, $j+1 \le k \le n$なるh, kに対して, $a[h] \le a[k]$.

ヒント: 外側のwhile文のループ不変表明は $A_0(a, b, n, j)$,

内側のwhile文のループ不変表明は $i \le j \wedge A_0(a, b, n, j) \wedge \forall k<i.a[k] \le a[i]$

局所変数宣言

変数xのスコープがプログラムPであるような局所変数宣言を以下の形で与えるとする.

new x; P

例:

《a=1∧b=2》 new a; { a=7; b=a+b } 《a=1∧b=9》 (postconditionでa=7ではない!)

《a=1∧b=2》 new n; { n=7; b=n+b } 《a=1∧b=9》 でも同じ

局所変数宣言の規則

$$\frac{《A》\ P[n/x]\ 《B》}{《A》\ \text{new } x;\ P\ 《B》}$$

ただしnはA, B, Pに現れない新しい変数

例

$$\frac{\langle\!\langle a=1 \land b=2 \rangle\!\rangle\ \{\ n=7;\ b=n+b\ \}\ \langle\!\langle a=1 \land b=9 \rangle\!\rangle}{\langle\!\langle a=1 \land b=2 \rangle\!\rangle\ \mathrm{new}\ a;\ \{\ a=7;\ b=a+b\ \}\ \langle\!\langle a=1 \land b=9 \rangle\!\rangle}$$

レポート問題

$\vdash_{par} ⦅\forall i(0 \le i \le x \supset y[i]=1) \land z=0⦆ \text{ while } z \ne x \ \{z=z+1; y[z]=y[z-1]*z\} \ ⦅\forall i(0 \le i \le x \supset y[i]=i!)⦆$

を検証せよ.

締切　8月6日(水)15:15

提出先　manaba