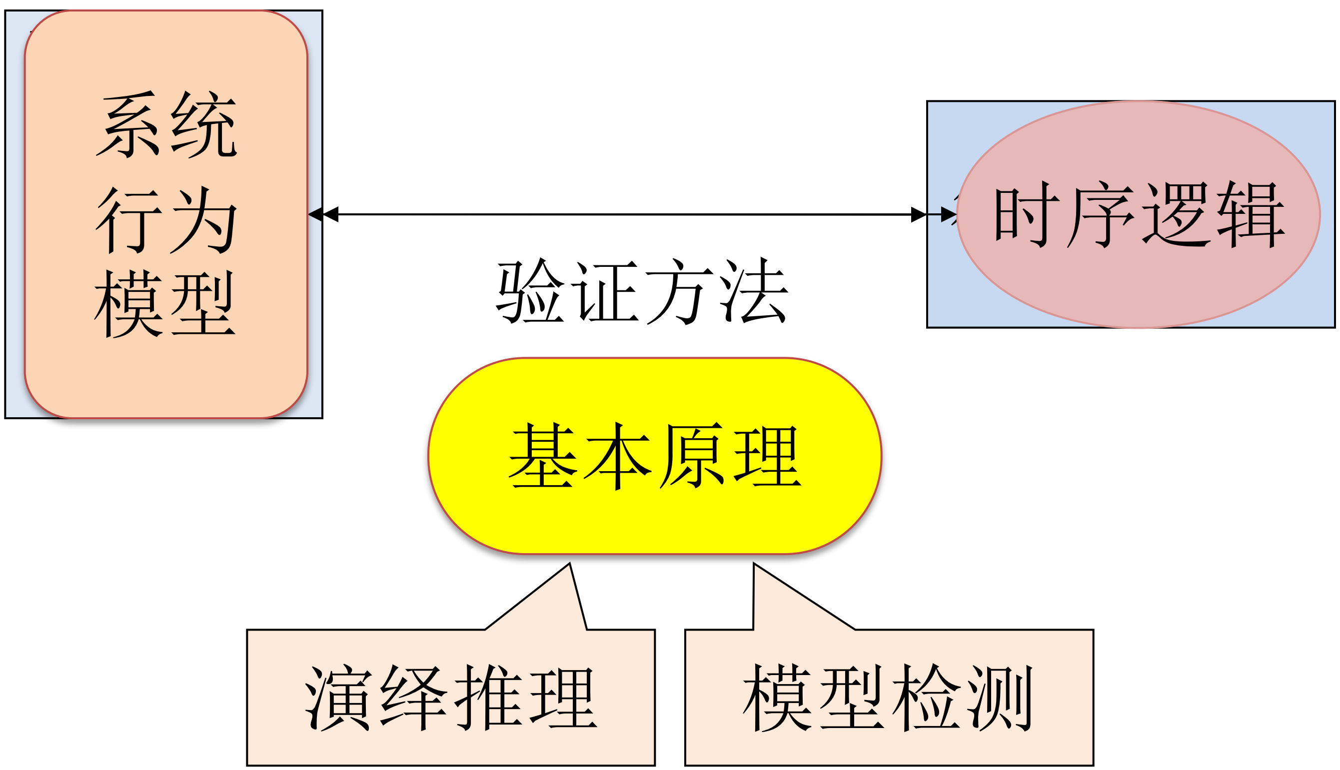
推理验证 --结构化程序模型

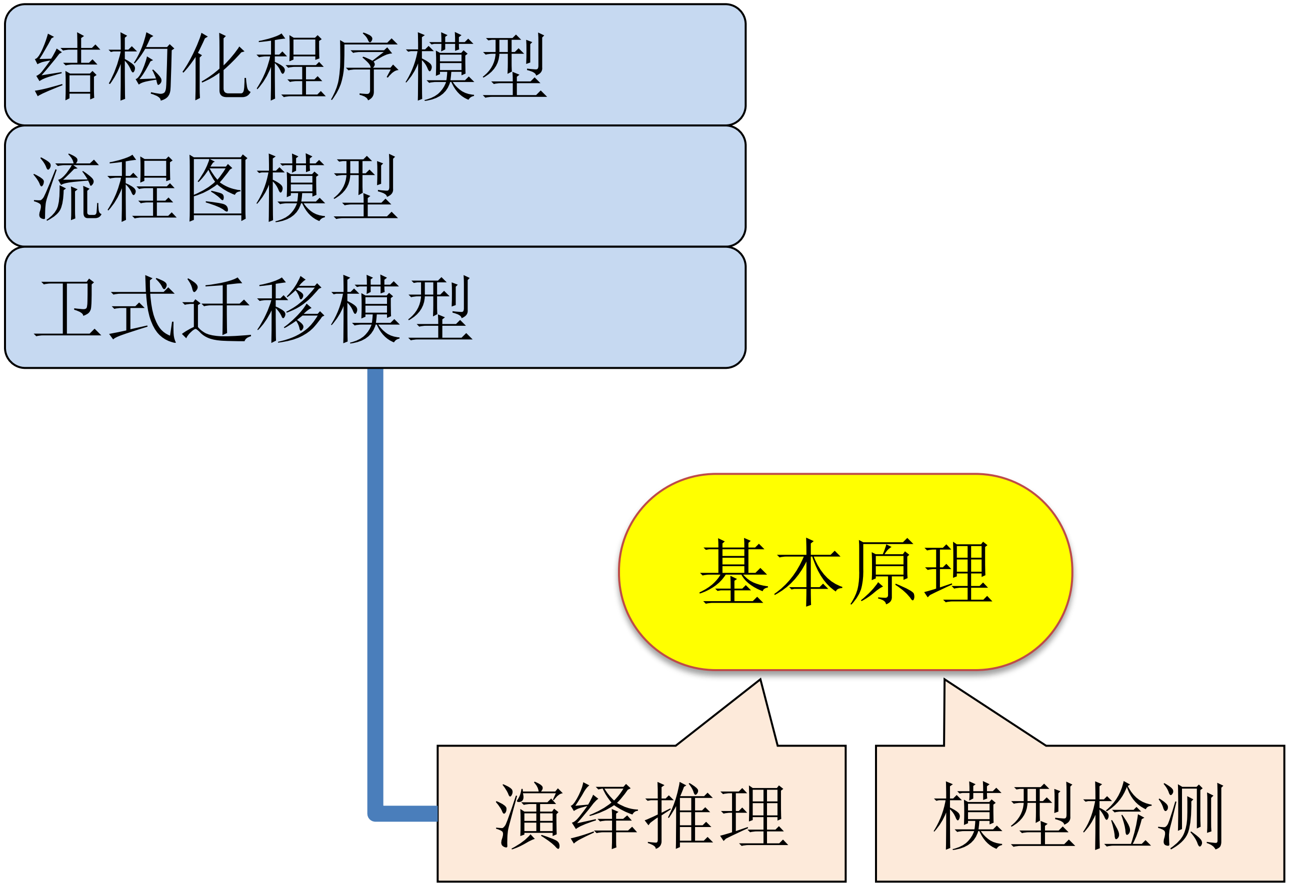
中国科学院软件研究所
计算机科学国家重点实验室
张文辉
http://lcs.ios.ac.cn/~zwh/

课程内容

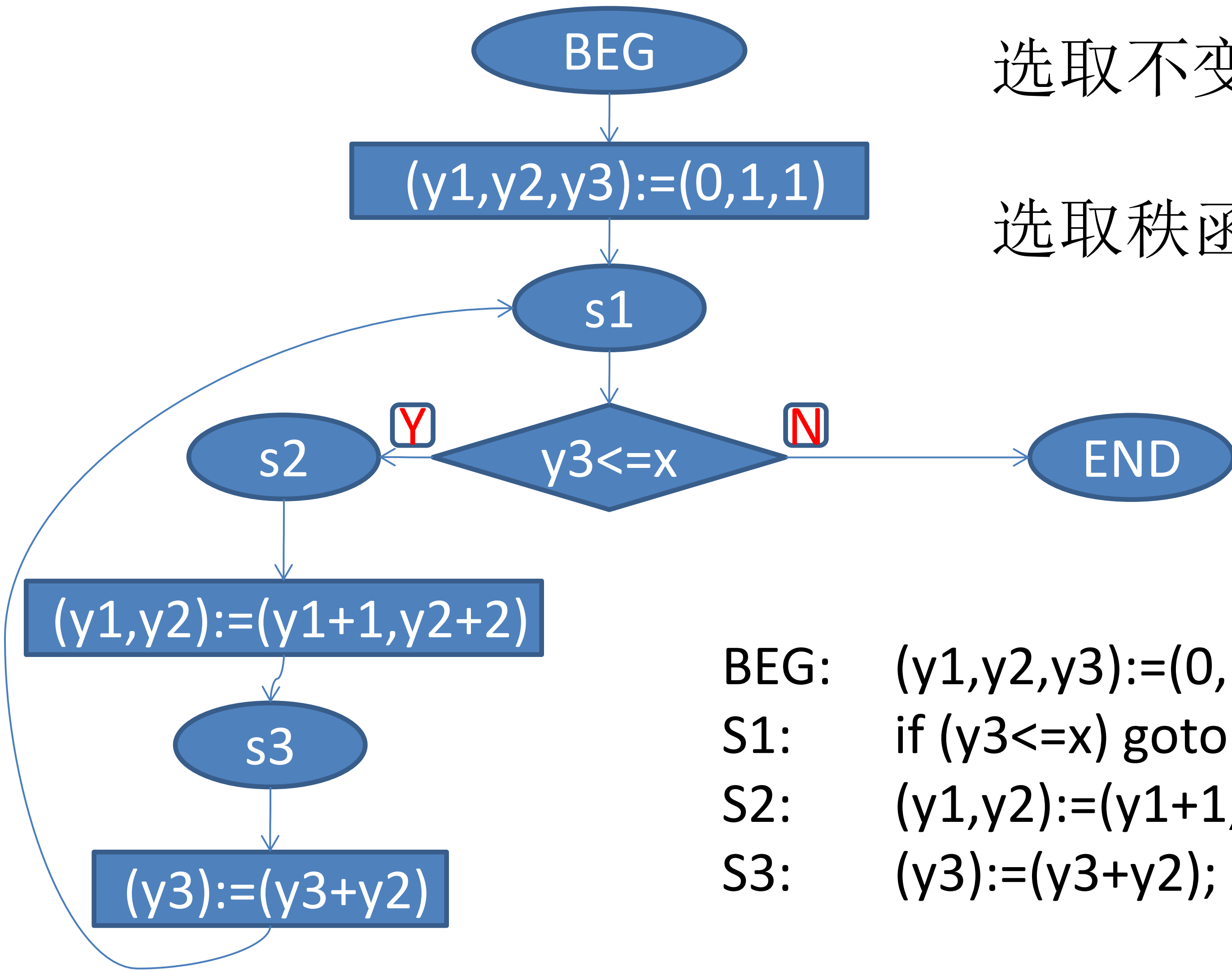


EE

课程内容(3)



回顾：流程图模型



选取不变量

选取秩函数

选取
关键标号和
路径

BEG: (y1,y2,y3):=(0,1,1); goto S1

S1: if (y3<=x) goto S2 else goto END

S2: (y1,y2):=(y1+1,y2+2); goto S3

S3: (y3):=(y3+y2); goto S1

例子

```
y1=0; y2=1; y3=1;
while (y3<=x) do
        y1=y1+1;
        y2=y2+2;
        y3=y3+y2;
od;
ε
```

选取不变量

选取秩函数

选取关键标号和路径的过程可以简化

模型具有更好的可组合性

结构化循环语句模型

有一些(更多的)程序结构的信息可用

关注一些特殊类型性质

可以发展具有针对性的方法

要点：循环语句 – 循环不变量

Contents

- Correctness
  - Partial Correctness
  - Termination
  - Total Correctness (Partial Correctness + Termination)

- Assertions
  - Preconditions/Postconditions
  - Weakest liberal preconditions
  - Weakest preconditions
  - Strongest postconditions

- Verification (Techniques and Examples)
  - Partial Correctness
  - Total Correctness

(I) Correctness (While-Programs)

- B=(F,P)
- V
- I=(D,I_0)
- Σ

- I: Term → (Σ→D)
- I: WFF → (Σ→ {0,1})

S: a program

Correctness (1)

Partial Correctness

DEF

$|=_I \{\varphi\}\ S\ \{\psi\}$

iff

$I(\varphi)(\sigma) \to ((S,\sigma) \to^* (\varepsilon,\sigma')) \to I(\psi)(\sigma'))$

Correctness (2)

Termination

DEF

$|=_I [\varphi]\ S\ [\text{true}]$

iff

$I(\varphi)(\sigma) \rightarrow ((S,\sigma) \rightarrow^* (\varepsilon,\sigma'))$

Correctness (3)

Total Correctness

DEF

$\models_I [\varphi]\ S\ [\psi]$

iff

$I(\varphi)(\sigma) \rightarrow ((S,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\psi)(\sigma'))$

Correctness

Total Correctness = Partial Correctness + Termination

Proposition:

$\models_{l} [\varphi]\ S\ [\ \psi\ ]$

iff

$\models_{l} \{\varphi\}\ S\ \{\ \psi\ \}$ and $\models_{l} [\varphi]\ S\ [\mathrm{true}]$

(II) Assertions

- Preconditions/postconditions
- Weakest liberal preconditions
- Weakest preconditions
- Strongest postconditions

Assertions (Pre-Post-Conditions, PC)

$|=_I \{\varphi\}\ T\ \{\psi\}$, iff

$$I(\varphi)(\sigma) \rightarrow \forall\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \rightarrow I(\psi)(\sigma'))$$

$$\frac{\varphi' \rightarrow \varphi \qquad |=_I \{\varphi\}\ T\ \{\psi\} \qquad \psi' \rightarrow \psi}{|=_I \{\varphi'\}\ T\ \{\psi'\}}$$

Assertions (Pre-Post-Conditions, TC)

$|=_I [\varphi]\ T\ [\psi]$, iff

$$I(\varphi)(\sigma) \rightarrow \exists\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\psi)(\sigma'))$$

$$\frac{\varphi' \rightarrow \varphi \qquad |=_I [\varphi]\ T\ [\psi] \qquad \psi' \rightarrow \psi}{|=_I [\varphi']\ T\ [\psi']}$$

Weakest Liberal Pre-Condition (DEF)

DEF $\varphi = wlp(T,\psi)$:

$I(\varphi)(\sigma) \leftrightarrow \forall\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \rightarrow I(\psi)(\sigma'))$

Theorem

$\varphi=wlp(T,\psi)$, iff

$\models \{\varphi\}\, T\{\psi\}$ and, if $\models \{\varphi'\}\ T\ \{\psi\}$ then $(\varphi' \rightarrow \varphi)$

Theorem

$\models \{\varphi\}\ T\ \{\psi\}$ iff $\varphi \rightarrow wlp(T,\psi)$

WLP (Proof $\rightarrow$)

$\varphi = wlp(T,\psi)$:

$I(\varphi)(\sigma) \leftrightarrow \forall\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \rightarrow I(\psi)(\sigma'))$

$\varphi=wlp(T,\psi) \rightarrow$

$|= \{\varphi\}\ T\{\psi\}$ and, if $|= \{\varphi'\}\ T\ \{\psi\}$ then $(\varphi' \rightarrow \varphi)$

a. $I(\varphi)(\sigma) \rightarrow \forall\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \rightarrow I(\psi)(\sigma'))$

b. $\forall\sigma.(I(\varphi')(\sigma)\rightarrow\forall\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \rightarrow I(\psi)(\sigma')))$

$\rightarrow \forall\sigma\ (I(\varphi')(\sigma) \rightarrow I(\varphi)(\sigma))$

WLP (Proof $\leftarrow$)

$\varphi = wlp(T,\psi)$:

$I(\varphi)(\sigma) \leftrightarrow \forall\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \rightarrow I(\psi)(\sigma'))$

$\varphi=wlp(T,\psi) \leftarrow$

$\models \{\varphi\}\ T\{\psi\}$ and, if $\models \{\varphi'\}\ T\ \{\psi\}$ then $(\varphi' \rightarrow \varphi)$

a. $I(\varphi)(\sigma) \rightarrow \forall\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \rightarrow I(\psi)(\sigma'))$

b. Let $I(\varphi')(\sigma) = \forall\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \rightarrow I(\psi)(\sigma'))$:

$\forall\sigma\ (\forall\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \rightarrow I(\psi)(\sigma'))) \rightarrow I(\varphi)(\sigma))$

Weakest Pre-Condition (DEF)

DEF $\varphi = wp(T,\psi)$:

$I(\varphi)(\sigma) \leftrightarrow \exists\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\psi)(\sigma'))$

Theorem

$\varphi=wp(T,\psi)$, iff

$|= [\varphi]\ T[\psi]$ and, if $|= [\varphi']\ T\ [\psi]$ then $(\varphi' \rightarrow \varphi)$

Theorem

$|= [\phi]\ T\ [\psi]$ iff $\phi \rightarrow wp(T,\psi)$

WP (Proof →)

$\varphi = wp(T,\psi)$:

$I(\varphi)(\sigma) \leftrightarrow \exists\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\psi)(\sigma'))$

$\varphi=wp(T,\psi) \rightarrow$

$\models [\varphi]\ T\ [\psi]$ and, if $\models [\varphi']\ T\ [\psi]$ then $(\varphi' \rightarrow \varphi)$

a. $I(\varphi)(\sigma) \rightarrow \exists\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\psi)(\sigma'))$

b. $\forall\sigma.(I(\varphi')(\sigma) \rightarrow \exists\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\psi)(\sigma')))$

$\rightarrow \forall\sigma\ (I(\varphi')(\sigma) \rightarrow I(\varphi)(\sigma))$

WLP (Proof $\leftarrow$)

$\varphi = wp(T,\psi)$:

$I(\varphi)(\sigma) \leftrightarrow \exists\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\psi)(\sigma'))$

$\varphi=wp(T,\psi) \leftarrow$

$\models [\varphi]\ T\ [\psi]$ and, if $\models [\varphi']\ T\ [\psi]$ then $(\varphi' \rightarrow \varphi)$

a. $I(\varphi)(\sigma) \rightarrow \exists\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\psi)(\sigma'))$

b. Let $I(\varphi')(\sigma) = \exists\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\psi)(\sigma')))$:

$\forall\sigma\ (\exists\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\psi)(\sigma')))\rightarrow I(\varphi)(\sigma))$

Strongest Post-Condition (DEF)

DEF $\psi = sp(T, \varphi)$:

$I(\psi)(\sigma') \leftrightarrow \exists\sigma.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\varphi)(\sigma))$

Theorem

$\psi = sp(T, \varphi)$ iff

$\models \{\varphi\}\ T\{\psi\}$ and, if $\models \{\varphi\}\ T\ \{\psi'\}$ then $(\psi \rightarrow \psi')$

Theorem

$\models \{\varphi\}\ T\ \{\psi\}$ iff $sp(T, \varphi) \rightarrow \psi$

Strongest Post-Condition (Proof $\rightarrow$)

$\psi = sp(T, \varphi)$:

$I(\psi)(\sigma') \leftrightarrow \exists\sigma.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\varphi)(\sigma))$

$\psi = sp(T, \varphi) \rightarrow$

$\models \{\varphi\}\ T\{\psi\}$ and, if $\models \{\varphi\}\ T\ \{\psi'\}$ then $(\psi \rightarrow \psi')$

a. $I(\varphi)(\sigma'') \rightarrow \forall\sigma'.(((T;\varepsilon,\sigma'') \rightarrow^* (\varepsilon,\sigma')) \rightarrow I(\psi)(\sigma'))$

b. $\forall\sigma''.(I(\varphi)(\sigma'')\rightarrow\forall\sigma'.(((T;\varepsilon,\sigma'') \rightarrow^* (\varepsilon,\sigma'))\rightarrow I(\psi')(\sigma')))$

$\rightarrow\forall\sigma'.(\exists\sigma.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\varphi)(\sigma)) \rightarrow I(\psi')(\sigma'))$

Strongest Post-Condition (Proof $\leftarrow$)

$\psi = sp(T, \varphi)$:

$I(\psi)(\sigma') \leftrightarrow \exists\sigma.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\varphi)(\sigma))$

$\psi = sp(T, \varphi) \leftarrow$

$\models \{\varphi\}\ T\{\psi\}$ and, if $\models \{\varphi\}\ T\ \{\psi'\}$ then $(\psi \rightarrow \psi')$

a. $I(\varphi)(\sigma'') \rightarrow \forall\sigma'.(((T;\varepsilon,\sigma'') \rightarrow^* (\varepsilon,\sigma')) \rightarrow I(\psi)(\sigma'))$

b. $\forall\sigma''.(I(\varphi)(\sigma'')\rightarrow\forall\sigma'.(((T;\varepsilon,\sigma'') \rightarrow^* (\varepsilon,\sigma'))\rightarrow I(\psi')(\sigma')))$

$\rightarrow\ (I(\psi)(\sigma') \rightarrow I(\psi')(\sigma'))$

where $I(\psi')(\sigma') = \exists\sigma.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\varphi)(\sigma))$

Example

$\{x=0 \wedge y=0\}$

while (y<z) { x:=x+z; y:=y+1 };

$\{\varphi\}$

while (y>0) { x:=x-z; y:=y-1 }

$\{x=0 \wedge y=0\}$

sp:

$\varphi \equiv (y=z \wedge x=z*z)$

wp, wlp:

$\varphi \equiv (x=y*z)$

Computation of WLP

S ::= ε | T; ε

T ::= x:=t | T;T | if (e) then T else T fi | while (e) do T od

Given T and ψ .

How to compute

[T]ψ

such that

[T]ψ ≡ wlp(T,ψ) ?

Computation of WLP (1)

$\{\varphi\}\ T\ \{\psi\}$ iff $\varphi \rightarrow wlp(T,\psi)$

$\{\varphi\}\ x:=e\ \{\psi\}$ iff $\varphi \rightarrow\ ?$

$I(\varphi)(\sigma) \rightarrow ((x:=e;\varepsilon,\sigma) \rightarrow^{*} (\varepsilon,\sigma')) \rightarrow I(\psi)(\sigma'))$ iff

$I(\varphi)(\sigma) \rightarrow I(\psi(e/x))(\sigma)$

$\{\varphi\}\ x:=e\ \{\psi\}$ iff $\varphi \rightarrow \psi(e/x)$

$[x:=e]\psi \quad = \psi(e/x)$

Computation of WLP (2)

- [T1;T2] ψ = [T1][T2] ψ

Computation of WLP (3)

$$[\text{if (b) then T0 else T1 fi}]\ \psi \ =$$

$$(b \rightarrow [T0]\psi) \wedge (\neg b \rightarrow [T1]\psi)$$

Computation of WLP (4)

- [while (b) do T0 od] ψ = ?

Analysis 1 (Fixpoint)

$[\text{while } (b) \text{ do } T0 \text{ od}]\psi = ?$

$[\text{if } (b) \text{ then } T0;T \text{ else } x:=x \text{ fi}]\ \psi =$

$(b \rightarrow [T0;T]\psi) \wedge (\neg b \rightarrow \psi) =$

$(b \rightarrow [T0]\ [T]\psi) \wedge (\neg b \rightarrow \psi)$

$\phi = (b \rightarrow [T0]\phi) \wedge (\neg b \rightarrow \psi)$ not dir. computable

Analysis 2 (Invariant)

$\phi' \rightarrow (b \rightarrow [T0]\phi') \wedge (\neg b \rightarrow \psi)$

$\Rightarrow$

$\models \{\phi'\}$ while (b) do T0 od $\{\psi\}$

[while (b) do T0 od] ψ = ϕ :

(1) $\phi \rightarrow (b \rightarrow [T0]\phi) \wedge (\neg b \rightarrow \psi)$

(2) If $\phi' \rightarrow (b \rightarrow [T0]\phi') \wedge (\neg b \rightarrow \psi)$, then $\phi' \rightarrow \phi$

(Assume that such a ϕ is expressible)

Computation of WLP

LEMMA

$wlp(T, \psi) \equiv [T]\psi$

COROLLARY

$|=_I \{\varphi\}\ T\ \{\psi\}$ iff $\varphi \rightarrow [T]\psi$

Example 1

T: $y1=y1+1;\ y2=y2+2;\ y3=y3+y2;$

φ : $y1*y1<=x \wedge y2=2*y1+1 \wedge y3 =(y1+1)*(y1+1)$

wlp(T, φ) =

[T]φ =

[y1=y1+1; y2=y2+2] φ(y3/(y3+y2)) =

[y1=y1+1] φ(y3/(y3+y2))(y2/(y2+2)) =

φ(y3/(y3+y2))(y2/(y2+2))(y1/(y1+1) =

$(y1+1)*(y1+1)<=x \wedge y2=2*y1+1 \wedge y3 =(y1+1)*(y1+1)$

Example 2

T: y1=y1+1; y2=y2+2; y3=y3+y2;

φ : y1*y1<=x ∧ y2=2*y1+1 ∧y3 =(y1+1)*(y1+1)

$|=_I$ {(y3<=x) ∧ φ} T {φ}

iff

(y3<=x) ∧ φ → wlp(T,φ)

iff

(y3<=x) ∧ φ → [T] φ

iff

(y3<=x)∧φ → (y1+1)*(y1+1)<=x∧y2=2*y1+1 ∧y3 =(y1+1)*(y1+1)

iff

true

(III) Verification Techniques

- Partial Correctness
- Total Correctness

(III.a) Proof Rules (PC, Hoare Logic)

$\{\varphi\}\ T\ \{\psi\}$

$|=_I \{\varphi\}\ T\ \{\psi\}$

iff

$I(\varphi)(\sigma) \rightarrow (((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \rightarrow I(\psi)(\sigma'))$

Composition of Programs (1)

$\phi \rightarrow \varphi \qquad \models_I \{\varphi \wedge b\}\ T0\ \{\varphi\} \qquad \varphi \wedge \neg b \rightarrow \psi$

$\models_I \{\phi\}$ while (b) do T0 od $\{\psi\}$

$b \wedge \varphi \rightarrow [T0]\varphi$

$\neg b \wedge \varphi \rightarrow \psi \qquad \Rightarrow \qquad \varphi \rightarrow$ [while (b) do T0 od]ψ

$\phi \rightarrow \varphi \qquad \Rightarrow \qquad \phi \rightarrow$ [while (b) do T0 od]ψ

Composition of Programs (2)

$\models_{I} \{b \wedge \varphi\}\ T0\ \{\ \psi\ \}$ $\qquad\qquad$ $\models_{I} \{\neg b \wedge \varphi\}\ T1\ \{\ \psi\ \}$

$\models_{I} \{\varphi\}$ if (b) then T0 else T1 fi $\{\ \psi\ \}$

$b \wedge \varphi \rightarrow [T0]\ \psi$

$\neg b \wedge \varphi \rightarrow [T1]\ \psi$

$\Leftrightarrow \quad \varphi \rightarrow (b \rightarrow [T0]\psi) \wedge (\neg b \rightarrow [T1]\psi)$

$\Leftrightarrow \quad \varphi \rightarrow$ [if (b) then T0 else T1 fi] ψ

Composition of Programs (3)

$\models_{l} \{\varphi\}\ T0\ \{\varphi'\}$ $\quad\quad$ $\models_{l} \{\varphi'\}\ T1\ \{\psi\}$

$\models_{l} \{\varphi\}\ T0;T1\ \{\psi\}$

$\varphi \rightarrow [T0]\ \varphi'$

$\varphi' \rightarrow [T1]\ \psi$

$\Rightarrow \quad \varphi \rightarrow [T0][T1]\ \psi$

$\Rightarrow \quad \varphi \rightarrow [T0;T1]\ \psi$

Assignments (4)

$$\varphi \rightarrow \psi(t/x)$$

$$\models_{I} \{\varphi\}\ x{:=}t\ \{\psi\}$$

$\varphi \rightarrow [x{:=}t]\psi$

Consequence (5)

$$\frac{\varphi' \rightarrow \varphi \qquad \{\varphi\}\ T\ \{\psi\} \qquad \psi \rightarrow \psi'}{\{\varphi'\}\ T\ \{\psi'\}}$$

Integer Square Root (PC)

```
{ x>=0 }
y1=0;
y2=1;
y3=1;

while (y3<=x) do
        y1=y1+1;
        y2=y2+2;
        y3=y3+y2;
od;
{ y1*y1<=x ∧ x<(y1+1)*(y1+1) }
```

例子

Integer Square Root (PC)

```
{ x>=0 }
y1=0;
y2=1;
y3=1;

while (y3<=x) do
        y1=y1+1;
        y2=y2+2;
        y3=y3+y2;
od;
{ y1*y1<=x ∧ x<(y1+1)*(y1+1) }
```

Integer Square Root

T:

```
T0; T1
```

T0:

```
y1=0;
y2=1;
y3=1;
```

T1:

```
while (y3<=x) do
      T11
od;
```

T11:

```
y1=y1+1;
y2=y2+2;
y3=y3+y2;
```

Integer Square Root

```
{ x>=0 }
T0;
while (y3<=x) do
        T11;
od;
{ y1*y1<=x ∧ x<(y1+1)*(y1+1) }
```

$\{ x>=0 \}$ T0 $\{ 0<=x \wedge y1=0 \wedge y2=1 \wedge y3 =1\}$

$\{y3<=x \wedge \varphi \}$ T11 $\{ y1*y1<=x \wedge y2=2*y1+1 \wedge y3 =(y1+1)*(y1+1) \}$

$\{\varphi \}$ while (y3<=x) do T11; od $\{\neg (y3<=x) \wedge \varphi\}$

Integer Square Root

$\{ x>=0 \}$
T0;
$\{ 0<=x \wedge y1=0 \wedge y2=1 \wedge y3 =1\}$
$\{\varphi \}$
while (y3<=x) do
    T11;
od;
$\{\neg (y3<=x) \wedge\varphi\}$
$\{ y1*y1<=x \wedge x<(y1+1)*(y1+1) \}$

$\{ x>=0 \}$ T0 $\{ 0<=x \wedge y1=0 \wedge y2=1 \wedge y3 =1\}$

$\{y3<=x\wedge\varphi \}$ T11 $\{ y1*y1<=x \wedge y2=2*y1+1 \wedge y3 =(y1+1)*(y1+1) \}$

$\{\varphi \}$ while (y3<=x) do T11; od $\{\neg (y3<=x) \wedge\varphi\}$

Proof of { $x>=0$ } T0 { $0<=x \wedge y1=0 \wedge y2=1 \wedge y3=1$ }

(1) { $x>=0$ } $y1=0$ { $0<=x \wedge y1=0$ }

(2) { $0<=x \wedge y1=0$ } $y2=1$ { $0<=x \wedge y1=0 \wedge y2=1$ }

(3) { $0<=x \wedge y1=0 \wedge y2=1$ } $y3=1$ { $0<=x \wedge y1=0 \wedge y2=1 \wedge y3=1$ }

(4) { $x>=0$ } $y1=0 ; y2=1$ { $0<=x \wedge y1=0 \wedge y2=1$ }

(5) { $x>=0$ } $y1=0 ; y2=1; y3=1$ { $0<=x \wedge y1=0 \wedge y2=1 \wedge y3=1$ }

{ $x>=0$ } T0 { $0<=x \wedge y1=0 \wedge y2=1 \wedge y3=1$ }

Proof of $\{y3<=x \wedge \varphi\}$ T11 $\{ y1*y1<=x \wedge y2=2*y1+1 \wedge y3=(y1+1)*(y1+1) \}$

$y3<=x \wedge \varphi \rightarrow (y1+1)*(y1+1)<=x \wedge y2=2*y1+1 \wedge y3=(y1+1)*(y1+1)$

$\{ (y1+1)*(y1+1)<=x \wedge y2=2*y1+1 \wedge y3=(y1+1)*(y1+1) \}$
y1=y1+1;
$\{ y1*y1<=x \wedge y2+2=2*y1+1 \wedge y3=(y1)*(y1) \}$
y2=y2+2;
$\{ y1*y1<=x \wedge y2=2*y1+1 \wedge y3=(y1)*(y1) \}$
y3=y3+y2;
$\{ y1*y1<=x \wedge y2=2*y1+1 \wedge y3=(y1+1)*(y1+1) \}$

$\{y3<=x \wedge \varphi\}$ T11 $\{ y1*y1<=x \wedge y2=2*y1+1 \wedge y3=(y1+1)*(y1+1) \}$

Integer Square Root

$\{ x>=0 \}$

y1=0; $\{ 0<=x \wedge y1=0 \}$

y2=1; $\{ 0<=x \wedge y1=0 \wedge y2=1 \}$

y3=1; $\{ 0<=x \wedge y1=0 \wedge y2=1 \wedge y3 =1\}$

$\{\varphi\}$

while (y3<=x) do $\{ (y3<=x) \wedge\varphi\}$

$\{(y1+1)*(y1+1)<=x \wedge y2=2*y1+1 \wedge y3 =(y1+1)*(y1+1) \}$

y1=y1+1; $\{ y1*y1<=x \wedge y2+2=2*y1+1 \wedge y3 =(y1)*(y1) \}$

y2=y2+2; $\{ y1*y1<=x \wedge y2=2*y1+1 \wedge y3 =(y1)*(y1) \}$

y3=y3+y2; $\{ y1*y1<=x \wedge y2=2*y1+1 \wedge y3 =(y1+1)*(y1+1) \}$

od;

$\{\neg (y3<=x) \wedge\varphi\}$

$\{ y1*y1<=x \wedge x<(y1+1)*(y1+1) \}$

Summary

Problem:

{ x>=0 }

```
y1=0;  y2=1;  y3=1;
while (y3<=x) do
        y1=y1+1;  y2=y2+2;  y3=y3+y2;
od;
```

{ y1*y1<=x ∧ x<(y1+1)*(y1+1) }

Let T0 be	y1=0; y2=1; y3=1;
Let T11 be	y1=y1+1; y2=y2+2; y3=y3+y2;
Let T1 be	while (y3<=x) do T11 od;
Let T be	T0; T1

Need to prove: {x>=0} T { y1*y1<=x ∧ x<(y1+1)*(y1+1) }

Let φ be y1*y1<=x $\wedge$ y2=2*y1+1 $\wedge$ y3 =(y1+1)*(y1+1)

We have: { φ(1/y3)(1/y2)(0/y1) } T0 { φ }
Since we have x>=0 $\rightarrow$ φ(1/y3)(1/y2)(0/y1) ,
we have { x>=0 } T0 { φ }

We have: { φ((y3+y2)/y3)((y2+2)/y2)((y1+1)/y1) } T11 { φ }
Since we have
(y3<=x) $\wedge\varphi$ $\rightarrow$ φ((y3+y2)/y3)((y2+2)/y2)((y1+1)/y1)
we have { (y3<=x) $\wedge\varphi$ } T11 { φ }

Therefore we have { φ } T1 { $\neg$(y3<=x) $\wedge\varphi$ }
Therefore we have { x>=0 } T { $\neg$(y3<=x) $\wedge\varphi$ }

Since $\neg$(y3<=x) $\wedge\varphi$ $\rightarrow$ y1*y1<=x $\wedge$ x<(y1+1)*(y1+1)
we have { x>=0 } T { y1*y1<=x $\wedge$ x<(y1+1)*(y1+1) }

Reasoning with wlp()

wlp(T0,φ)

φ

wlp(T1,φ)

¬(y3<=x)∧ φ

```
{ x>=0 }
y1=0;  y2=1;  y3=1;

while (y3<=x) do
        y1=y1+1;
        y2=y2+2;
        y3=y3+y2;
{ φ }
od;
{ y1*y1<=x ∧ x<(y1+1)*(y1+1) }
```

(y3<=x)∧ φ → wlp(T1, φ)

¬(y3<=x)∧ φ → y1*y1<=x ∧ x<(y1+1)*(y1+1)

(x>=0) → wlp(T0,φ)

Integer Square Root (PC)

```
{ x>=0 }
y1=0;
y2=1;
y3=1;

while (y3<=x) do
        y1=y1+1;
        y2=y2+2;
        y3=y3+y2;  { y1*y1<=x ∧ y2=2*y1+1 ∧ y3 =(y1+1)*(y1+1) }
od;
{ y1*y1<=x ∧ x<(y1+1)*(y1+1) }
```

程序推理辅助工具XYZ/VERI-II

功能：

给定简单程序和性质以及不变式，

生成验证条件、进行验证条件的化简

下载：

lcs.ios.ac.cn/~zwh/veri2/veri2.tar.gz

Example

```
{x=c}
   y1:=0; y2:=1; y3:=1;
   while (y3<=x) {
        y1:=(y1+1); y2 := (y2+2);
        y3:=(y2+y3);
        { x=c∧ y1*y1<=x∧y3=(y1+1)*(y1+1)∧
          y2=2*y1+1}
   }
{ y1*y1<=c ∧ c<=(y1+1)*(y1+1) }
```

Pre-Condition

Invariant

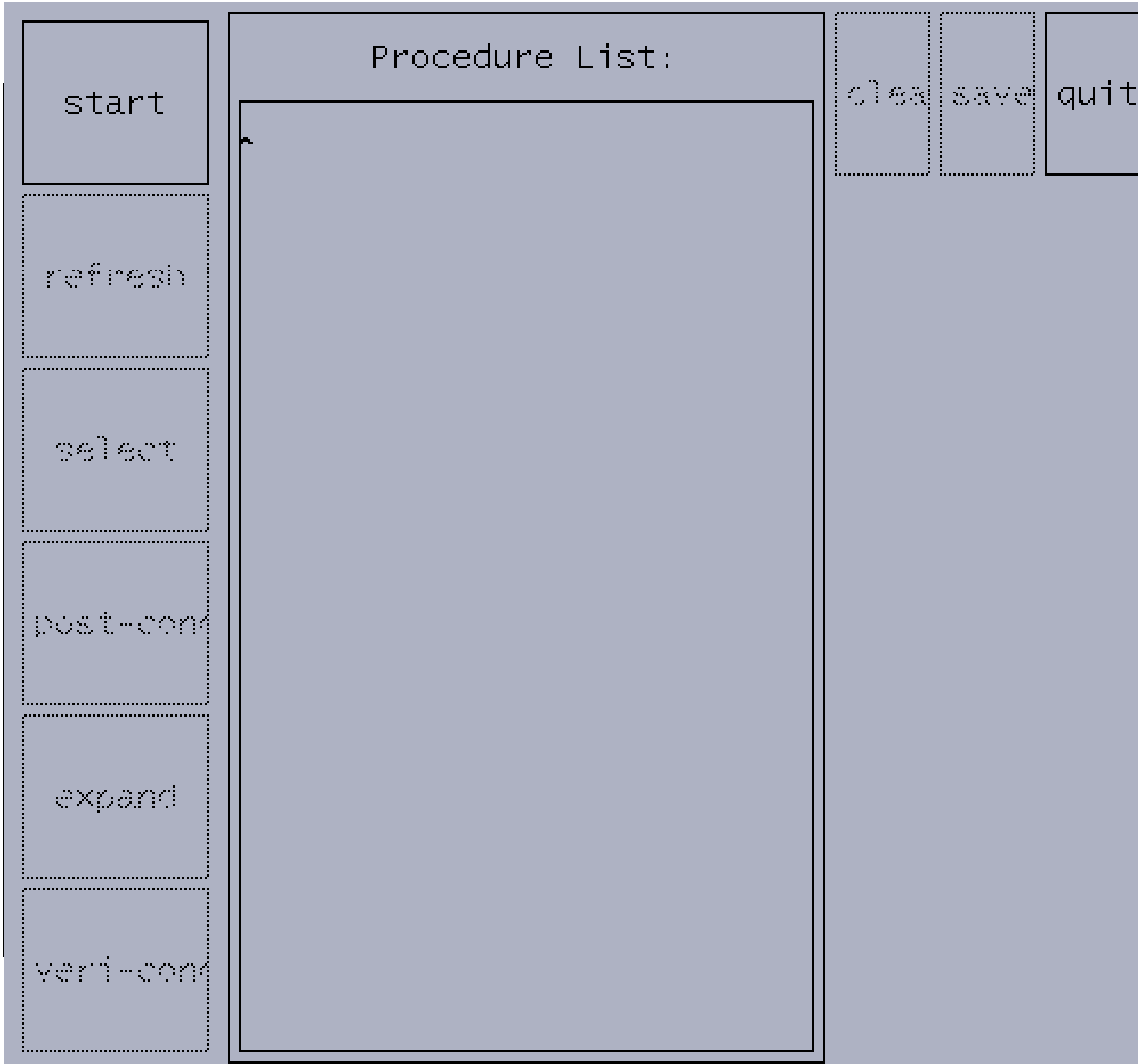
Post-Condition

Program in XYZ/SE

```
{x=c}
%PROC wl(%INP/x:INT;%IOP/y1:INT)==
%LOC [y2,y3:INT]
%STM [
      LB=START => $Oy1=0 /\ $Oy2=1 /\ $Oy3=1 /\ $OLB=l2;
      *[ LB=l2  /\ (le(y3,x)) => ($OLB=l3 | $OLB=END);
          LB=l3  => $Oy1=+(y1,1) /\ $Oy2 = +(y2,2) /\ $OLB=l4;
          LB=l4  => $Oy3=+(y2,y3) /\ $OLB=l2;
          { x=c /\ le(*(y1,y1),x) /\ y3=*(+(y1,1),+(y1,1)) /\ y2=+(*(2,y1),1) }
        ] ]
{ le(*(y1,y1),c) /\ lt(c,*(+(y1,1),+(y1,1))) }
```

XYZ/VERI-II:

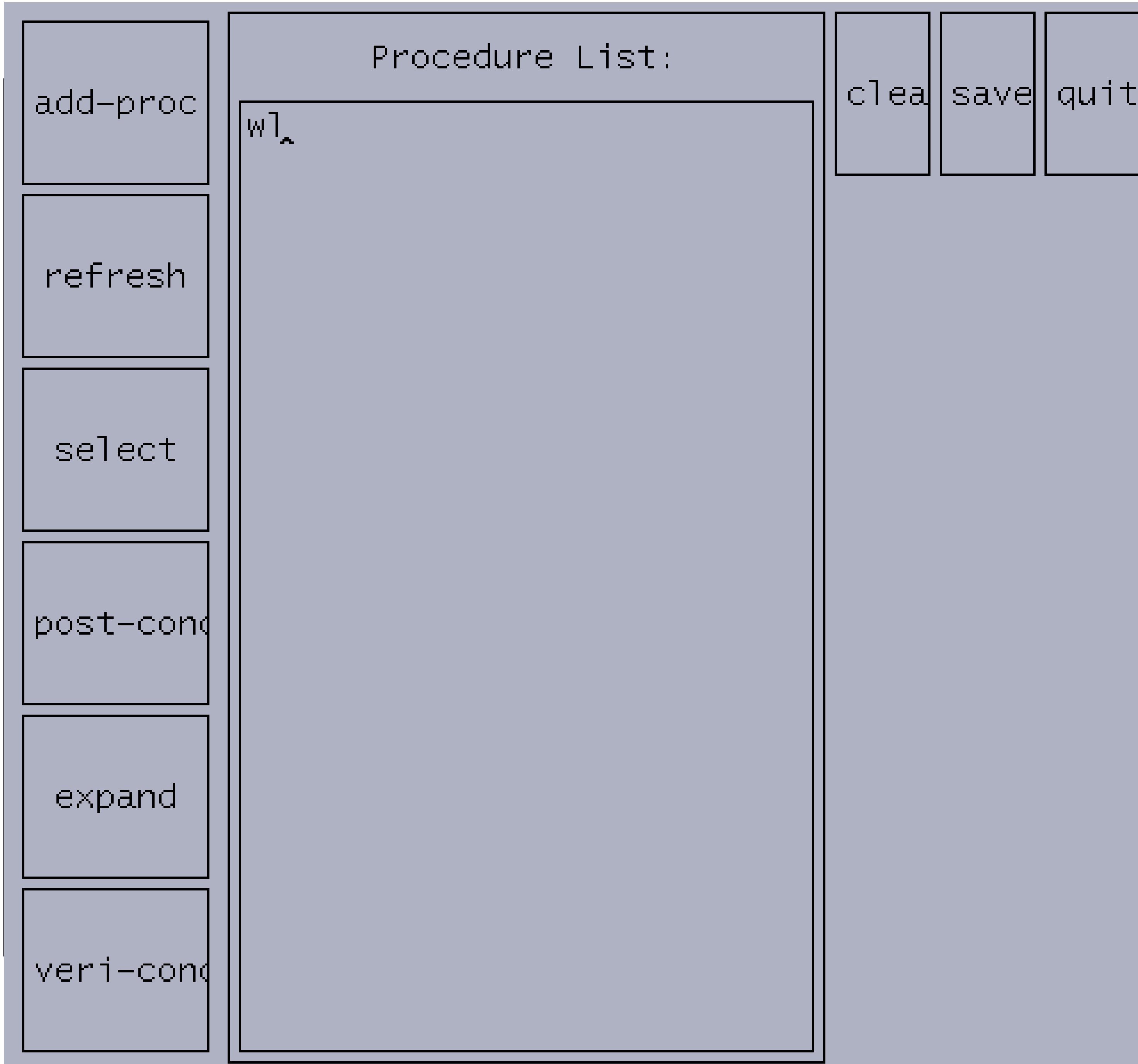
- User-Interface and Functionalities


start
refresh
select
post-con
expand
veri-con
Procedure List:
clea
save
quit

Please Provide Procedure Text or a File Name:

ve.ex3

Ok | Cancel


add-proc
refresh
select
post-conc
expand
veri-conc
Procedure List:
wl
clea
save
quit

Procedure wl:

```
{ x=c }
%PROC wl(%INP/x:INT;%IOP/y1:INT)==
%LOC [y2,y3:INT]
%STM [
        LB=START=>$Oy1=0^$Oy2=1^$Oy3=1^$OLB=l2;
        *[LB=l2^le(y3,x)=>($OLB=l3|$OLB=END)
            LB=l3=>$Oy1=+(y1,1)^$Oy2=+(y2,2)^$OLB=l4;
            LB=l4=>$Oy3=+(y2,y3)^$OLB=l2;
            {(x=c^(le(*(y1,y1),x)^(y3=*(+(y1,1),+(y1,1))^y2=+
           ]
]
{ (le(*(y1,y1),c)^lt(c,*(+(y1,1),+(y1,1)))) }
```

Show | Cancel


add-proc
refresh
select
post-conc
expand
veri-conc
Procedure List:
wl
clea
save
quit

Verification Conditions for wl:

```
(le(y3,x)^(x=c^(le(*(y1,y1),x)^
    (y3=*(+(y1,1),+(y1,1))^
y2=+(*(2,y1),1)))))
 =>
  (x=c^(le(*(+(y1,1),+(y1,1)),x)^
   (+(+(y2,2),y3)=*(+(+(y1,1),1),+(+(y1,1),1))^
+(y2,2)=+(*(2,+(y1,1)),1))))

(~le(y3,x)^(x=c^(le(*(y1,y1),x)^
    (y3=*(+(y1,1),+(y1,1))^
y2=+(*(2,y1),1)))))
 =>
  (le(*(y1,y1),c)^
lt(c,*(+(y1,1),+(y1,1))))

x=c
 =>
  (x=c^(le(*(0,0),x)^(1=*(+(0,1),+(0,1))^
1=+(*(2,0),1))))
```

Simplify | Cancel

Verification Conditions

$y3 <= x \wedge x=c \wedge y1*y1<=x \wedge y3=(y1+1)*(y1+1) \wedge y2=2*y1+1$

$\rightarrow x=c \wedge (y1+1)*(y1+1)<=x \wedge$

$(y2+2)+y3=((y1+1)+1)*((y1+1)+1) \wedge y2+2=2*(y1+1)+1$

$\neg y3<=x \wedge x=c \wedge y1*y1<=x \wedge y3=(y1+1)*(y1+1) \wedge y2=2*y1+1$

$\rightarrow y1*y1<=c \wedge c<(y1+1)*(y1+1)$

$x=c$

$\rightarrow x=c \wedge 0*0<=x \wedge 1=(0+1)*(0+1) \wedge 1=2*0+1$

Verification Conditions for wl:

```
le(*(+(y1,1),+(y1,1)),c),le(*(y1,y1),c)
 =>
  +(2,+(1,*(y1,2)))=+(1,*(2,+(y1,1)))

le(*(y1,y1),c)
 =>
  lt(c,*(+(y1,1),+(y1,1))),le(*(+(y1,1),+(y1,1)),c)

$T
 =>
  1=+(1,*(2,0))

le(*(+(y1,1),+(y1,1)),c),le(*(y1,y1),c)
 =>
  +(+(2,+(1,*(y1,2))),*(+(y1,1),+(y1,1)))=*(+(1,+(y1,1)

$T
 =>
  le(*(0,0),c)

$T
 =>
  1=*(+(1,0),+(1,0))
```

Simplify | Cancel

Simplified Verification Conditions

(y1+1)*(y1+1)<=c, y1*y1<=c → 2+(1+y1*2) = 1+2*(y1+1)
y1*y1<= c → c < (y1+1)*(y1+1), (y1+1)*(y1+1)<= c
T → 1=1+2*0
(y1+1)*(y1+1) <= c, y1*y1<= c → (2+(1+y1*2))+(y1+1)*(y1+1)=(1+(y1+1))*(1+(y1+1))
T → 0*0<=c
T →1=(1+0)*(1+0)

Please Provide an Axiom:

```
le(*(0,0),c)
```

Simplify

Cancel

Verification Conditions for wl:

```
le(*(+(y1,1),+(y1,1)),c),le(*(y1,y1),c)
 =>
  +(2,+(1,*(y1,2)))=+(1,*(2,+(y1,1)))

le(*(y1,y1),c)
 =>
  lt(c,*(+(y1,1),+(y1,1))),le(*(+(y1,1),+(y1,1)),c)

$T
 =>
  1=+(1,*(2,0))

le(*(+(y1,1),+(y1,1)),c),le(*(y1,y1),c)
 =>
  +(+(2,+(1,*(y1,2))),*(+(y1,1),+(y1,1)))=*(+(1,+(y1,1)

$T
 =>
  1=*(+(1,0),+(1,0))
```

Simplify | Cancel

Simplified Verification Conditions

(y1+1)*(y1+1)<=c, y1*y1<=c → 2+(1+y1*2) = 1+2*(y1+1)
y1*y1<= c → c < (y1+1)*(y1+1), (y1+1)*(y1+1)<= c
T → 1=1+2*0
(y1+1)*(y1+1) <= c, y1*y1<= c → (2+(1+y1*2))+(y1+1)*(y1+1)=(1+(y1+1))*(1+(y1+1))
T →1=(1+0)*(1+0)

(III.b) Proof Rules (TC)

Extended Hoare Logic:

$[\varphi]\ T\ [\psi]$

$\models_I [\varphi]\ T\ [\psi]$,

iff

$I(\varphi)(\sigma) \rightarrow \exists\sigma'.(((T;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \wedge I(\psi)(\sigma'))$

Composition of Programs (1)

w, W, t:

$|=_I \phi \wedge b \rightarrow w(t/x)$ $|=_I [\phi \wedge b \wedge t=v]\ T0\ [\phi \wedge t<v]$

$|=_I [\phi]$ while (b) do T0 od $[\neg b \wedge \phi]$

$\varphi \rightarrow \phi$ $|=_I$ …… $|=_I$ …… $\phi \wedge \neg b \rightarrow \psi$

$|=_I [\varphi]$ while (b) do T0 od $[\psi]$

Composition of Programs (2)

$|=_{l} [b \wedge \varphi]\ T0\ [\ \psi\ ]$ $\qquad$ $|=_{l} [\neg b \wedge \varphi] T1\ [\ \psi\ ]$

$|=_{l} [\varphi]$ if (b) then T0 else T1 fi $[\ \psi\ ]$

Composition of Programs (3)

$\models_I [\varphi]\ T0\ [\varphi']$ $\models_I [\varphi']\ T1\ [\psi]$

$\models_I [\varphi]\ T0;T1\ [\psi]$

Assignments (4)

$$\varphi \rightarrow \psi(t/x)$$

$$\models_I [\varphi]\ x{:=}t\ [\psi]$$

Consequence (5)

$$\frac{\varphi' \rightarrow \varphi \qquad [\varphi]\ T\ [\psi] \qquad \psi \rightarrow \psi'}{[\varphi']\ T\ [\psi']}$$

例子

Integer Square Root (TC)

```
[ x>=0 ]
y1=0;
y2=1;
y3=1;

while (y3<=x) do
            y1=y1+1;
            y2=y2+2;
            y3=y3+y2;
od;
[ y1*y1<=x ∧ x<(y1+1)*(y1+1) ]
```

Integer Square Root

T:

```
    T0; T1
```

T0:

```
    y1=0;
    y2=1;
    y3=1;
```

T1:

```
    while (y3<=x) do
        T11
    od;
```

T11:

```
    y1=y1+1;
    y2=y2+2;
    y3=y3+y2;
```

Integer Square Root

```
[x>=0 ]
T0;
while (y3<=x) do
        T11;
od;
[ y1*y1<=x ∧ x<(y1+1)*(y1+1) ]
```

$[\, x>=0 \,]\ T0\ [\, 0<=x \wedge y1=0 \wedge y2=1 \wedge y3 =1]$

$[y3<=x \wedge \varphi \wedge ?=v \,]\ T11\ [\, y1*y1<=x \wedge y2=2*y1+1 \wedge y3 =(y1+1)*(y1+1) \wedge ?<v]$

$[\varphi]$ while (y3<=x) do T11; od $[\neg (y3<=x) \wedge \varphi]$

Integer Square Root

W: NAT

w: x>=0

t = x+1-y3

$y3 <= x \wedge \varphi \rightarrow (x+1-y3>=0)$ and

$[y3<=x \wedge \varphi \wedge t=v\ ]\ T11\ [\ y1*y1<=x \wedge y2=2*y1+1 \wedge y3 =(y1+1)*(y1+1) \wedge t<v]$

$[\varphi]$ while (y3<=x) do T11; od $[\neg\ (y3<=x) \wedge \varphi]$

Summary

Problem:

```
[ x>=0 ]
        y1=0;  y2=1;  y3=1;
        while (y3<=x) do
                y1=y1+1;  y2=y2+2;  y3=y3+y2;
        od;
[ y1*y1<=x ∧ x<(y1+1)*(y1+1) ]
```

Let T0 be	y1=0; y2=1; y3=1;
Let T11 be	y1=y1+1; y2=y2+2; y3=y3+y2;
Let T1 be	while (y3<=x) do T11 od;
Let T be	T0; T1

Need to prove: [x>=0]   T [y1*y1<=x ∧ x<(y1+1)*(y1+1)]

Let φ be y1*y1<=x $\wedge$ y2=2*y1+1 $\wedge$ y3 =(y1+1)*(y1+1)

We have: [φ(1/y3)(1/y2)(0/y1)] T0 [φ]
Since we have x>=0 $\rightarrow$ φ(1/y3)(1/y2)(0/y1) ,
we have [x>=0] T0 [φ]

Let W be NAT, and w be x>=0
Let t be x+1-y3. We have y3<=x$\wedge\varphi$ $\rightarrow$ (x+1-y3>=0)

Since y3<=x$\wedge\varphi\wedge$t=v $\rightarrow$
($\varphi\wedge$t<v) ((y3+y2)/y3)((y2+2)/y2)((y1+1)/y1)
we have [y3<=x$\wedge\varphi\wedge$t=v] T11 [$\varphi\wedge$t<v]
Therefore [φ] T1 [$\neg$(y3<=x) $\wedge\varphi$]
Therefore [x>=0] T [$\neg$(y3<=x) $\wedge\varphi$]
Therefore [x>=0] T [y1*y1<=x $\wedge$ x<(y1+1)*(y1+1)]

(IV) Hoare Logic (Floyd-Hoare Logic)

公式:　　$\{ \varphi \} T \{ \psi \}$

语义

可满足性

推论

推理系统

推理系统的完备性

Hoare Logic

Let SP be the set of programs specified as follows.

SP := x:=t |
P1;P2 |
if (b) { P1 } else { P2 } |
while (b) { P1 }

The set of Hoare logic formulas:

- If P $\in$ SP and $\varphi,\psi \in$QFF, then $\{\varphi\}P\{\psi\}$ is a Hoare logic formula (Hoare Triple).

Interpretation

$I(\{\varphi\}P\{\psi\})\sigma$ = true, if

$I(\varphi)\sigma$ implies $((P;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma'))$ implies $I(\psi)\sigma'$)

$|=_I \{\varphi\}P\{\psi\}$ if

for all $\sigma \in \Sigma$, $I(\{\varphi\}P\{\psi\})\sigma$ = true

$|= \{\varphi\}P\{\psi\}$ if

for all I, $|=_I \{\varphi\}P\{\psi\}$

Logical Consequence

Let W be a set of formulas.

W $|= \{\varphi\}P\{\psi\}$ if

for every I,

if $|=_I$ W, then we have $|=_I \{\varphi\}P\{\psi\}$.

Example (1)

$\{x>5\}$ x:=2*x $\{x>20\}$

Let I be given as usual.

$I(\{x>5\}\ x:=2*x\ \{x>20\})\sigma = \text{true iff } \sigma(x)\leq 5 \vee \sigma(x)>10$

Example (2)

{true} while (x$\neq$10) { x:=x+1 } {x=10}

Let I be given as usual.

I({true} while (x$\neq$10) { x:=x+1 } {x=10})σ = true

for all σ.

$|=_I \{\varphi\}P\{\psi\}$

Example (3)

{true} x:=y+1 {x>y}

Let I be given as usual. Then $|=_I \{\varphi\}P\{\psi\}$

Let W = {y+1>y}.

$I(\{true\}\ x:=y+1\ \{x>y\})(\sigma) = true$

$\Leftrightarrow I(y+1>y)(\sigma) = true \Leftrightarrow$ I is a model of W

$W\ |= \{\varphi\}P\{\psi\}$

On Hoare Logic

Hoare logic is a first order logic:

$$|=_I \{\varphi\}\ P\ \{\psi\} \Leftrightarrow th(I) \ |= \{\varphi\}\ P\ \{\psi\}$$

Hoare Logic (1)

- Axiom:

$$\{\psi[x/t]\}\ x{:=}t\ \{\psi\}$$

Hoare Logic (2)

- Sequential Composition

$$\frac{\{\psi 0\}\ P1\ \{\psi 1\} \qquad \{\psi 1\}\ P2\ \{\psi 2\}}{\{\psi 0\}\ P1;P2\ \{\psi 2\}}$$

Hoare Logic (3)

- Conditional Composition

$$\frac{\{b \wedge \varphi\}\ P1\ \{\psi\} \qquad \{\neg b \wedge \varphi\}\ P2\ \{\psi\}}{\{\varphi\}\ \text{if (b) \{P1\} else \{P2\}}\ \{\psi\}}$$

Hoare Logic (4)

- Loop Composition

$\{b \wedge \psi\}\ P\ \{\psi\}$

$\{\psi\}$ while (b) $\{P\}$ $\{\psi \wedge \neg b\}$

Hoare Logic (5)

- Consequence

$$\frac{\varphi' \rightarrow \varphi \qquad \{\varphi\}\ P\ \{\psi\} \qquad \psi \rightarrow \psi'}{\{\varphi'\}\ P\ \{\psi'\}}$$

Soundness

- The proof system is sound:

$$|-_{l} \{\varphi\}\ P\ \{\psi\} \Rightarrow\ |=_{l} \{\varphi\}\ P\ \{\psi\}$$

by structural induction.

Lemma

$$I(\psi[x/t])\sigma = I(\psi)\sigma[x/I(t)\sigma]$$

$$\models_I \{\psi[x/t]\}\ x:=t\ \{\psi\}$$

Lemma

$I(\varphi)\sigma \rightarrow (((P1;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')) \rightarrow I(\psi')\sigma')$ and

$I(\psi')\sigma' \rightarrow (((P2;\varepsilon,\sigma') \rightarrow^* (\varepsilon,\sigma'')) \rightarrow I(\psi)\sigma'')$

$\rightarrow$

$I(\varphi)\sigma \rightarrow (((P1;P2;\varepsilon,\sigma) \rightarrow^* (P2;\varepsilon,\sigma'')) \rightarrow I(\psi)\sigma'')$

$\models_I \{\varphi\}\ P1\ \{\psi'\}$ and $\models_I \{\psi'\}\ P2\ \{\psi\}$

$\rightarrow$

$\models_I \{\varphi\}\ P1;P2\ \{\psi\}$

Lemma

$\models_l \{b \wedge \varphi\}$ P1 $\{\psi\}$ and $\models_l \{\neg b \wedge \varphi\}$ P2 $\{\psi\}$

$\Rightarrow$

$\models_l \{\varphi\}$ if (b) { P1 } else { P2 } $\{\psi\}$

Lemma

$\models_l \{\varphi \wedge b\}\ P1\ \{\varphi\}$

$\Rightarrow$

$\models_l \{\varphi\}\ \text{while (b) } \{P1\}\ \{\varphi \wedge \neg b\}$

Lemma

$\models_l \varphi' \rightarrow \varphi$, $\models_l \{\varphi\}\ P\ \{\psi\}$ and $\models_l \psi \rightarrow \psi'$

$\Rightarrow$

$\models_l \{\varphi'\}\ P\ \{\psi'\}$

Relative Completeness

- The proof system is relatively complete.

$$\models_l \{\varphi\}\ P\ \{\psi\} \Rightarrow \vdash_l \{\varphi\}\ P\ \{\psi\}$$

- Relative to the expressive power of the underlying first order logic and the completeness of the underlying proof system

Example

$\{x=0 \wedge y=0\}$

```
while (y<z) { x:=x+z; y:=y+1 };
while (y>0) { x:=x-z; y:=y-1 }
```

$\{x=0 \wedge y=0\}$

Proof

$\models_l \{\varphi\}\ P\ \{\psi\} \Rightarrow \vdash_l \{\varphi\}\ P\ \{\psi\}$

by structural induction.

Lemma

$$|=_I \{\varphi\}\ x{:=}t\ \{\psi\} \Rightarrow |=_I \varphi \rightarrow \psi[x/t]$$

$$wlp(x{:=}t,\ \psi) \equiv \psi[x/t]$$

Lemma

$|=_I \{\varphi\}$ P1;P2 $\{\psi\}$, and $I(\psi')$ is WLP of P2 and $I(\psi)$

➔

$|=_I \{\varphi\}$ P1 $\{\psi'\}$ and $|=_I \{\psi'\}$ P2 $\{\psi\}$

Lemma

$|=_l \{\varphi\}$ if (b) { P1 } else { P2 } $\{\psi\}$

➔

$|=_l \{b \wedge \varphi\}$ P1 $\{\psi\}$ and $|=_l \{\neg b \wedge \varphi\}$ P2 $\{\psi\}$

Lemma

$\models_I \{\varphi\}$ while (b) {P1} $\{\psi\}$

$\Rightarrow$

$\models_I \{\varphi\}$ if (b) { P1; while (b) {P1} } else {x:=x} $\{\psi\}$

$\models_I \{\varphi\}$ while (b) {P1} $\{\psi\}$, and

$I(\psi')$ is the WLP of "while (b) {P1}" and $I(\psi)$

$\Rightarrow$

$\models_I \varphi \rightarrow \psi'$, $\models_I \{\psi' \wedge b\}$ P1 $\{\psi'\}$, and $\models_I \psi' \wedge \neg b \rightarrow \psi$

Examples

Example 1

$\{x \geq 0 \wedge x=n\}$

```
y:=1;
while (x>0) {
        y:=y*x;
        x:=x-1
}
```

$\{y=n!\}$

$\{ x \geq 0 \wedge y*x!=n! \}$

$\{x-1 \geq 0 \wedge y*x*(x-1)!=n! \}$

$\{ x-1 \geq 0 \wedge y*(x-1)!=n! \}$

$\{ x \geq 0 \wedge y*x!=n! \}$

Example 2

$\{x \geq 0 \wedge y \geq 0 \wedge x = a \wedge y = b\}$

```
while (¬(x=y)) {
        if (x>y) { x:=x−y }
        else { y:=y−x }                { gcd(x,y)=gcd(a,b) }
}
```

$\{x = gcd(a,b)\}$

Extended Hoare Logic

The set of extended Hoare logic formulas:

- If $P \in SP$ and $\varphi, \psi \in QFF(V)$, then $[\varphi]P[\psi]$ is an extended Hoare logic formula.

Interpretation

$I([\varphi]P[\psi])\sigma$ = true, if

$I(\varphi)\sigma$ implies $((P;\varepsilon,\sigma) \rightarrow^* (\varepsilon,\sigma')$ and $I(\psi)\sigma')$

$|=_I [\varphi]P[\psi]$ if

for all $\sigma \in \Sigma$, $I([\varphi]P[\psi])\sigma$ = true

$|= [\varphi]P[\psi]$ if

for all I, $|=_I [\varphi]P[\psi]$

Logical Consequence

Let W be a set of formulas.

W $|= [\varphi]P[\psi]$ if

for every I such that, $|=_I$ W,

we have $|=_I [\varphi]P[\psi]$.

On Extended Hoare Logic

Extended Hoare logic is not a first order logic.

Example 1

```
y:=1;
while (x>0) {
        y:=y*x;
        x:=x-1
}
```

Extended Hoare Logic (1)

- Axiom:

$$[\psi[x/t]]x:=t[\psi]$$

Extended Hoare Logic (2)

- Sequential Composition

$$\frac{[\psi 0]\ P1\ [\psi 1] \qquad [\psi 1]\ P2\ [\psi 2]}{[\psi 0]\ P1;P2\ [\psi 2]}$$

-

Extended Hoare Logic (3)

- Conditional Composition

$$\frac{[b \wedge \varphi]\ P1\ [\psi] \qquad [\neg b \wedge \varphi]\ P2\ [\psi]}{[\varphi]\ \text{if } (b)\ \{P1\}\ \text{else}\ \{P2\}\ [\psi]}$$

Extended Hoare Logic (4)

- Loop Composition

$$\frac{b \wedge \psi \rightarrow w[x/t] \qquad [b \wedge \psi \wedge t=v]\ P\ [\psi \wedge t<v]}{[\psi]\ \text{while (b) \{P\}}\ [\psi \wedge \neg b]}$$

where $<\in P$, and w characterizes a well-founded set.

Extended Hoare Logic (5)

- Consequence

$$\frac{\varphi' \rightarrow \varphi \qquad [\varphi]\ P\ [\psi] \qquad \psi \rightarrow \psi'}{[\varphi']\ P\ [\psi']}$$

Examples

Example 1

```
[x≥0∧x=n]
y:=1;
while (x>0) {                                   { t=(x); w=(x≥0) }
        y:=y*x;
        x:=x-1
        { x≥0∧y*x!=n! }
}
[y=n!]
```

Example 2

$[x>0 \wedge y>0 \wedge x=a \wedge y=b]$

while $(\neg(x=y))$ { { $t=(x+y)$; $w=(x \geq 0)$ }

if $(x>y)$ { $x:=x-y$ }

else { $y:=y-x$ }

{ $x>0 \wedge y>0 \wedge gcd(x,y)=gcd(a,b)$ }

}

$[x=gcd(a,b)]$

Verification Condition Generation

vcg(ϕ,T,ψ):

Φ={}; p=vc(ϕ,T,ψ); $\Phi = \Phi \cup \{p\}$; return Φ

----

vc(ϕ,T;x:=e,ψ) $\equiv$ vc(ϕ,T,ψ(e/x))

vc(ϕ,T;if (b) then T0 else T1 fi,ψ) $\equiv$
vc(ϕ,T,vc(b,T0,ψ)$\wedge$vc($\neg$b,T1,ψ))

vc(ϕ,T;while (b) do T0 {φ} od,ψ) $\equiv$ vc(ϕ,T,φ)

UPDATE: $\Phi = \Phi \cup \{vc(b \wedge \varphi, T0, \varphi), \neg b \wedge \varphi \rightarrow \psi\}$

vc(ϕ,ε,ψ) $\equiv \phi \rightarrow \psi$

Integer Square Root (PC)

```
{ x>=0 }
y1=0; y2=1;  y3=1;
while (y3<=x) do
        y1=y1+1;
        y2=y2+2;
        y3=y3+y2;  { y1*y1<=x ∧ y2=2*y1+1 ∧ y3 =(y1+1)*(y1+1) }
od;
{ y1*y1<=x ∧ x<(y1+1)*(y1+1) }
```

Let T0 be	y1=0; y2=1; y3=1;
Let T11 be	y1=y1+1; y2=y2+2; y3=y3+y2;
Let T1 be	while (y3<=x) do T11 {φ} od;
Let T be	T0; T1

Integer Square Root (PC)

$vc(x>=0,T,y1*y1<=x \wedge x<(y1+1)*(y1+1)) \equiv$

$vc(x>=0,T0,y1*y1<=x \wedge y2=2*y1+1 \wedge y3 =(y1+1)*(y1+1)) \equiv$

$vc(x>=0,\varepsilon,0<=x \wedge 1=2*0+1 \wedge 1 =(0+1)*(0+1)) \equiv$

$x>=0 \rightarrow 0<=x \wedge 1=2*0+1 \wedge 1 =(0+1)*(0+1))$

Integer Square Root (PC)

$vc(y3<=x \wedge \varphi, T11, \varphi) \equiv$

$vc(y3<=x \wedge \varphi, \varepsilon, \varphi((y3+y2)/y3)((y2+2)/y2)((y1+1)/y1)\ ) \equiv$

$y3<=x \wedge \varphi \rightarrow \varphi((y3+y2)/y3)((y2+2)/y2)((y1+1)/y1)$

$\neg y3<=x \wedge \varphi \rightarrow y1*y1<=x \wedge x<(y1+1)*(y1+1)$

$vc(x>=0,T,y1*y1<=x \wedge x<(y1+1)*(y1+1)) \equiv$

$vc(x>=0,T0,y1*y1<=x \wedge y2=2*y1+1 \wedge y3=(y1+1)*(y1+1)) \equiv$

$vc(x>=0,\varepsilon,0<=x \wedge 1=2*0+1 \wedge 1=(0+1)*(0+1)) \equiv$

$x>=0 \rightarrow 0<=x \wedge 1=2*0+1 \wedge 1=(0+1)*(0+1))$

Verification Condition Generation (TC)

vcg(ϕ,T,ψ):

Φ={}; p=vc(ϕ,T,ψ); $\Phi = \Phi \cup$ {p}; return Φ

----

vc(ϕ,T;x:=e,ψ) $\equiv$ vc(ϕ,T,ψ(e/x))

vc(ϕ,T;if (b) then T0 else T1 fi,ψ) $\equiv$
vc(ϕ,T,vc(b,T0,ψ)$\wedge$vc($\neg$b,T1,ψ))

vc(ϕ,T;while (b) do {t,w} T0 {φ} od,ψ) $\equiv$ vc(ϕ,T,φ)

UPDATE:

$\Phi = \Phi \cup$ {vc(b$\wedge\varphi\wedge$t=v,T0,$\varphi\wedge$t<v),$\neg$b$\wedge\varphi \rightarrow \psi$,b$\wedge\varphi \rightarrow$w(x/t)}

vc(ϕ,ε,ψ) $\equiv \phi \rightarrow \psi$

Integer Square Root (TC)

```
{ x>=0 }
y1=0; y2=1;  y3=1;
while (y3<=x) do    {x+1-y3,x>=0}
        y1=y1+1;
        y2=y2+2;
        y3=y3+y2;  { y1*y1<=x ∧ y2=2*y1+1 ∧ y3 =(y1+1)*(y1+1) }
od;
{ y1*y1<=x ∧ x<(y1+1)*(y1+1) }
```

Let T0 be	y1=0; y2=1; y3=1;
Let T11 be	y1=y1+1; y2=y2+2; y3=y3+y2;
Let T1 be	while (y3<=x) do {x+1-y3,x>=0} T11 {φ} od;
Let T be	T0; T1

Integer Square Root (TC)

$vc(x>=0, T, y1*y1<=x \wedge x<(y1+1)*(y1+1)) \equiv$

$vc(x>=0, T0, y1*y1<=x \wedge y2=2*y1+1 \wedge y3 =(y1+1)*(y1+1)) \equiv$

$vc(x>=0, \varepsilon, 0<=x \wedge 1=2*0+1 \wedge 1 =(0+1)*(0+1)) \equiv$

$x>=0 \rightarrow 0<=x \wedge 1=2*0+1 \wedge 1 =(0+1)*(0+1))$

Integer Square Root (TC)

vc(y3<=x ∧ φ ∧ x+1-y3=v, T11, φ ∧ x+1-y3<v) ≡ … ≡
y3<=x ∧ φ ∧ x+1-y3=v
→φ((y3+y2)/y3)((y2+2)/y2)((y1+1)/y1)∧x+1-(y3+y2+2)<v

y3<=x ∧ φ → x+1-y3>=0
¬y3<=x ∧ φ → y1*y1<=x ∧ x<(y1+1)*(y1+1)

vc(x>=0,T,y1*y1<=x ∧ x<(y1+1)*(y1+1)) ≡
vc(x>=0,T0,y1*y1<=x ∧ y2=2*y1+1 ∧ y3 =(y1+1)*(y1+1)) ≡
vc(x>=0,ε,0<=x ∧ 1=2*0+1 ∧1 =(0+1)*(0+1)) ≡
x>=0 → 0<=x ∧ 1=2*0+1 ∧1 =(0+1)*(0+1))

(V) Summary

- Correctness/Properties
- Assertions (Basic Theories)
- Verification Techniques
- Hoare Logic

基于演绎推理的验证

卫式迁移模型 | 顺序流程图模型 | 结构化程序模型

- 模型和程序的语义
- 断言、前后断言
- 最弱宽松前断言、最弱前断言、最强后断言
- 正确性、部分正确性、终止性、完全正确性
- 计算方法、推理规则、逻辑

练习1

设 $B = (\{i, j, k, l, x, y, a, b\}, \{0, 1, 2, 3, ..., +, -, *\}, \{<, =, >\})$

给定I为B在整数上的正常解释。

记以下程序为T 。

if (x>y) then x:=x-y; i:=i-k; j:=j-l; else y:=y-x; k:=k-i; l:=l-j;

计算最弱宽松前断言wlp(T,(x=i*a+j*b))，

并证明$\models_I$ { y=k*a+l*b $\wedge$(x=i*a+j*b) } T { x=i*a+j*b}。

练习2

设 $B = (\{i, j, k, l, x, y, a, b\}, \{0, 1, 2, 3, ..., +, -, *\}, \{<, =, >\})$

给定以下程序 T：

```
i:=1; j:=0; k:=0; l:=1;
while ¬(x = y) do
    if x > y then  x:=x-y; i:=i-k; j:=j-l;
    else  y:=y-x; k:=k-i; l:=l-j;
od
```

给定 I 为 B 在整数上的正常解释。证明以下命题成立：

$$\vdash_I \{x = a \wedge y = b \wedge a \geq 0 \wedge b \geq 0\}\ T\ \{x = gcd(a, b) \wedge x = i * a + j * b\}$$

$$\vdash_I [x = a \wedge y = b \wedge a > 0 \wedge b > 0\ ]\ T\ [x = gcd(a, b) \wedge x = i * a + j * b]$$