

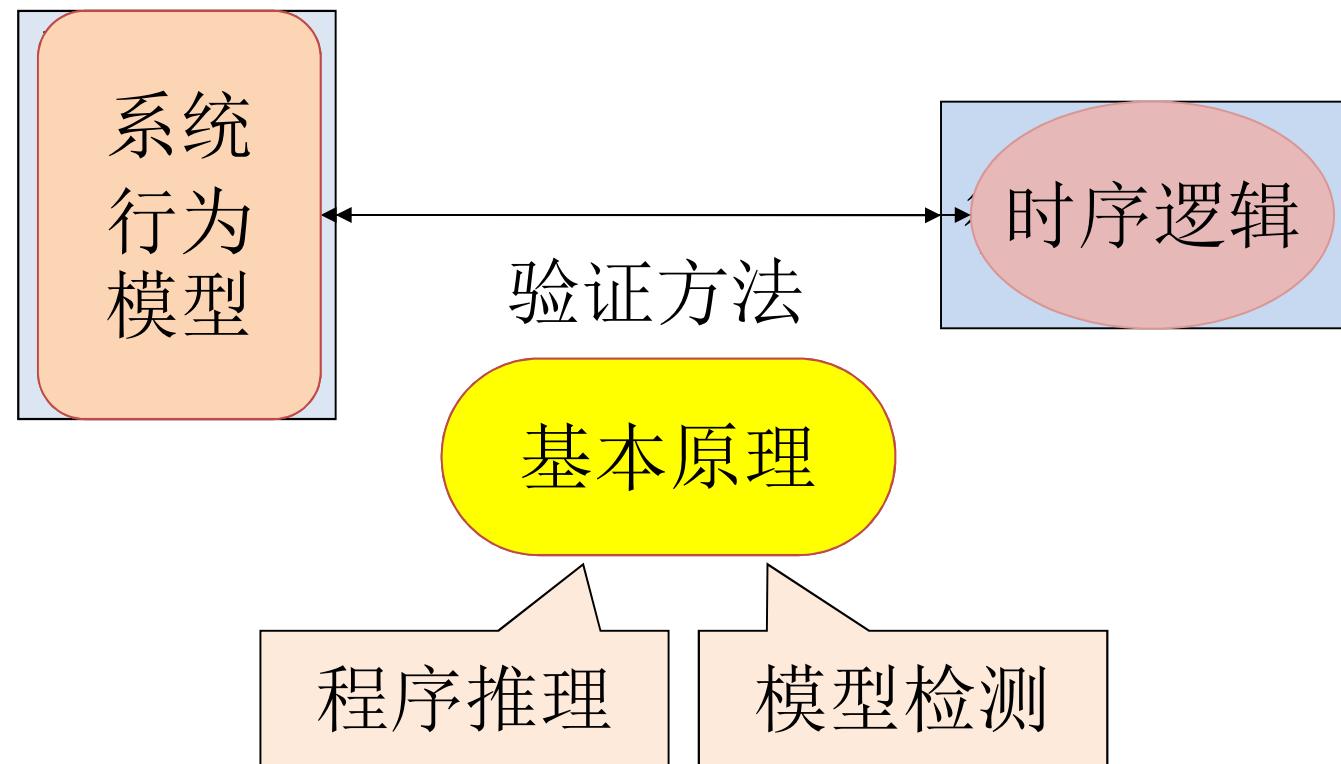
推理验证 --结构化程序模型

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



课程内容(3)

结构化程序模型

流程图模型

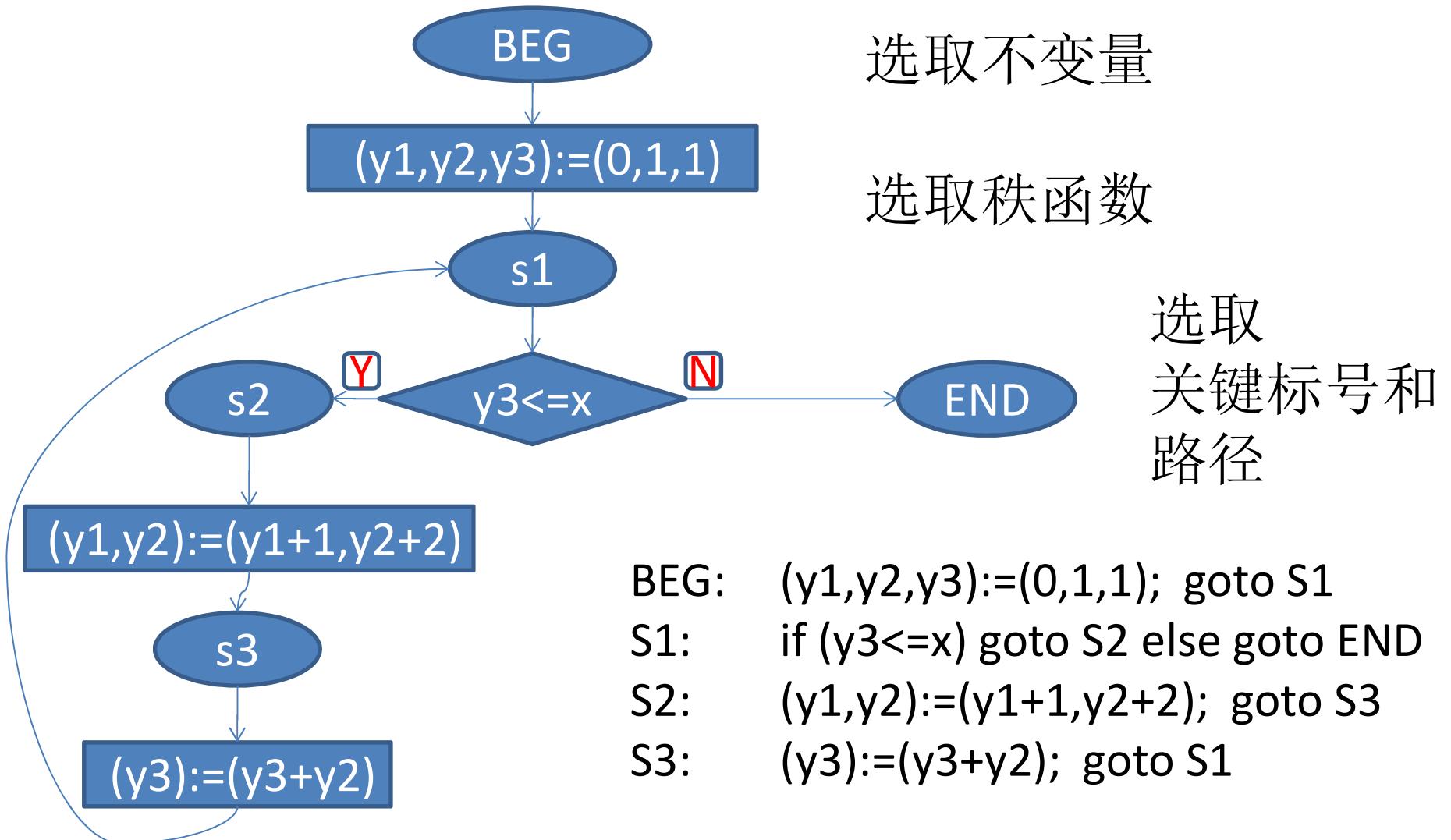
卫式迁移模型

基本原理

程序推理

模型检测

回顾：流程图模型



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
- $I: \text{Term} \rightarrow (\Sigma \rightarrow D)$
- $I: \text{WFF} \rightarrow (\Sigma \rightarrow \{0,1\})$

S: a program

Correctness (1)

Partial Correctness

DEF

$$I = , \{ \varphi \} S \{ \psi \}$$

iff

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

Correctness (2)

Termination

DEF

$\models, [\varphi] S \text{ [true]}$

iff

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

Correctness (3)

Total Correctness

DEF

$$I =_I [\varphi] S [\psi]$$

iff

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

Correctness

Total Correctness = Partial Correctness + Termination

Proposition:

$$\models, [\varphi] S [\psi]$$

iff

$$\models, \{\varphi\} S \{ \psi \} \text{ and } \models, [\varphi] S [\text{true}]$$

(II) Assertions

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

Assertions (Pre-Post-Conditions)

$\models, \{\varphi\} \vdash \{\psi\}$, iff

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

$\models, [\varphi] \vdash [\psi]$, iff

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

Weakest Liberal Pre-Condition (DEF)

DEF $\varphi = \text{wlp}(\mathcal{T}, \psi)$:

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

Theorem

$\varphi = \text{wlp}(\mathcal{T}, \psi)$, iff

$| = \{\varphi\} \vdash \{\psi\}$ and, if $| = \{\varphi'\} \vdash \{\psi\}$ then $(\varphi' \rightarrow \varphi)$

Theorem

$| = \{\varphi\} \vdash \{\psi\}$ iff $\varphi \rightarrow \text{wlp}(\mathcal{T}, \psi)$

WLP (Proof →)

$\varphi = \text{wlp}(\Gamma, \Psi)$:

$$I(\varphi)(\sigma) \leftrightarrow (((\Gamma; \varepsilon, \sigma) \rightarrow^* (\varepsilon, \sigma')) \rightarrow I(\Psi)(\sigma'))$$

$\varphi = \text{wlp}(\Gamma, \Psi) \rightarrow$

$I = \{\varphi\} \vdash \{\Psi\}$ and, if $I = \{\varphi'\} \vdash \{\Psi\}$ then $(\varphi' \rightarrow \varphi)$

- a. $I(\varphi)(\sigma) \rightarrow \forall \sigma'. (((\Gamma; \varepsilon, \sigma) \rightarrow^* (\varepsilon, \sigma')) \rightarrow I(\Psi)(\sigma'))$
- b. $\forall \sigma. (I(\varphi')(\sigma) \rightarrow \forall \sigma'. (((\Gamma; \varepsilon, \sigma) \rightarrow^* (\varepsilon, \sigma')) \rightarrow I(\Psi)(\sigma'))) \rightarrow \forall \sigma (I(\varphi')(\sigma) \rightarrow I(\varphi)(\sigma))$

WLP (Proof \leftarrow)

$\varphi = \text{wlp}(\mathsf{T}, \psi)$:

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

$\varphi = \text{wlp}(\mathsf{T}, \psi) \leftarrow$

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

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

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

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

Weakest Pre-Condition (DEF)

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

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

Theorem

$\varphi = \text{wp}(T, \psi)$, iff

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

Theorem

$\models [\varphi] T [\psi]$ iff $\varphi \rightarrow \text{wp}(T, \psi)$

WP (Proof \rightarrow)

$\varphi = \text{wp}(\mathsf{T}, \psi)$:

$$\mathsf{I}(\varphi)(\sigma) \leftrightarrow (((\mathsf{T}; \varepsilon, \sigma) \rightarrow^* (\varepsilon, \sigma')) \wedge \mathsf{I}(\psi)(\sigma'))$$

$\varphi = \text{wp}(\mathsf{T}, \psi) \rightarrow$

$\mathsf{I} = [\varphi] \mathsf{T} [\psi]$ and, if $\mathsf{I} = [\varphi'] \mathsf{T} [\psi]$ then $(\varphi' \rightarrow \varphi)$

- a. $\mathsf{I}(\varphi)(\sigma) \rightarrow \forall \sigma'. (((\mathsf{T}; \varepsilon, \sigma) \rightarrow^* (\varepsilon, \sigma')) \wedge \mathsf{I}(\psi)(\sigma'))$
- b. $\forall \sigma. (\mathsf{I}(\varphi')(\sigma) \rightarrow \forall \sigma'. (((\mathsf{T}; \varepsilon, \sigma) \rightarrow^* (\varepsilon, \sigma')) \wedge \mathsf{I}(\psi)(\sigma'))))$
 $\rightarrow \forall \sigma (\mathsf{I}(\varphi')(\sigma) \rightarrow \mathsf{I}(\varphi)(\sigma))$

WLP (Proof \leftarrow)

$\varphi = \text{wp}(\mathsf{T}, \psi)$:

$$\mathsf{I}(\varphi)(\sigma) \leftrightarrow (((\mathsf{T}; \varepsilon, \sigma) \rightarrow^* (\varepsilon, \sigma')) \wedge \mathsf{I}(\psi)(\sigma'))$$

$\varphi = \text{wp}(\mathsf{T}, \psi) \leftarrow$

$\mathsf{I} = [\varphi] \mathsf{T} [\psi]$ and, if $\mathsf{I} = [\varphi'] \mathsf{T} [\psi]$ then $(\varphi' \rightarrow \varphi)$

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

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

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

Strongest Post-Condition (DEF)

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

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

Theorem

$\psi = \text{sp}(T, \varphi)$ iff

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

Theorem

$| = \{\varphi\} T \{\psi\}$ iff $\text{sp}(T, \varphi) \rightarrow \psi$

Strongest Post-Condition (Proof \rightarrow)

$\Psi = \text{sp}(T, \varphi)$:

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

$\Psi = \text{sp}(T, \varphi) \rightarrow$

$| = \{\varphi\} T \{\Psi\}$ and, if $| = \{\varphi\} T \{\Psi'\}$ 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)

$\Psi = \text{sp}(T, \varphi)$:

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

$\Psi = \text{sp}(T, \varphi) \leftarrow$

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

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

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

Computation of WLP

$S ::= \varepsilon \mid T; \varepsilon$

$T ::= x := t \mid T; T \mid \text{if } (e) \text{ then } T \text{ else } T \text{ fi} \mid \text{while } (e) \text{ do } T \text{ od}$

Given T and ψ .

How to compute

$[T]\psi$

such that

$[T]\psi = \text{wlp}(T, \psi)$?

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

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

Computation of WLP (2)

- $[T_1; T_2] \psi = [T_1][T_2] \psi$

Computation of WLP (3)

[if (b) then T0 else T1 fi] ψ =

$$(b \rightarrow [T_0]\psi) \wedge (\neg b \rightarrow [T_1]\psi)$$

Computation of WLP (4)

- [while (b) do T0 od] $\psi = ?$

Analysis 1 (Fixpoint)

[while (b) do T0 od]ψ = ?

[if (b) then T0;T else x:=x fi] ψ =

(b → [T0;T]ψ) ∧ (¬b → ψ) =

(b → [T0] [T]ψ) ∧ (¬b → ψ)

φ = (b → [T0]φ) ∧ (¬b → ψ) not dir. computable

Analysis 2 (Invariant)

$$\phi' \rightarrow (b \rightarrow [T_0]\phi') \wedge (\neg b \rightarrow \psi)$$

→

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

[while (b) do T0 od] $\psi = \phi :$

(1) $\phi \rightarrow (b \rightarrow [T_0]\phi) \wedge (\neg b \rightarrow \psi)$

(2) If $\phi' \rightarrow (b \rightarrow [T_0]\phi') \wedge (\neg b \rightarrow \psi)$, then $\phi' \rightarrow \phi$

(Assume that such a ϕ is expressible)

Computation of WLP

LEMMA

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

COROLLARY

$$\models_I \{\varphi\} T \{\psi\} \text{ iff } \varphi \rightarrow [T]\psi$$

Example 1

$T:$ $y_1 = y_1 + 1; \quad y_2 = y_2 + 2; \quad y_3 = y_3 + y_2;$

$\varphi :$ $y_1 * y_1 \leq x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1 + 1) * (y_1 + 1)$

wlp(T, φ) =

$[T]\varphi =$

$[y_1 = y_1 + 1; \quad y_2 = y_2 + 2] \varphi(y_3 / (y_3 + y_2)) =$

$[y_1 = y_1 + 1] \varphi(y_3 / (y_3 + y_2))(y_2 / (y_2 + 2)) =$

$\varphi(y_3 / (y_3 + y_2))(y_2 / (y_2 + 2))(y_1 / (y_1 + 1)) =$

$(y_1 + 1) * (y_1 + 1) \leq x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1 + 1) * (y_1 + 1)$

Example 2

T: $y_1 = y_1 + 1; y_2 = y_2 + 2; y_3 = y_3 + y_2;$

$\varphi :$ $y_1 * y_1 \leq x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1 + 1) * (y_1 + 1)$

$\models_I \{(y_3 \leq x) \wedge \varphi\} T \{\varphi\}$

iff

$(y_3 \leq x) \wedge \varphi \rightarrow \text{wlp}(T, \varphi)$

iff

$(y_3 \leq x) \wedge \varphi \rightarrow [T] \varphi$

iff

$(y_3 \leq x) \wedge \varphi \rightarrow (y_1 + 1) * (y_1 + 1) \leq x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1 + 1) * (y_1 + 1)$

iff

true

(III) Verification Techniques

- Partial Correctness
- Total Correctness

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

$\{\varphi\} \top \{\psi\}$

$\vdash_l \{\varphi\} \top \{\psi\}$

iff

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

Composition of Programs (1)

$$\phi \rightarrow \phi$$

$$|=, \{ \varphi \wedge b \} T_0 \{ \varphi \}$$

$$\varphi \wedge \neg b \rightarrow \psi$$

$$|=, \{ \phi \} \text{while } (b) \text{ do } T_0 \text{ od } \{ \psi \}$$

$$\begin{array}{l} b \wedge \phi \rightarrow [T_0]\phi \\ \neg b \wedge \phi \rightarrow \psi \end{array} \Rightarrow \phi \rightarrow [\text{while } (b) \text{ do } T_0 \text{ od }] \psi$$

$$\phi \rightarrow \phi \Rightarrow \phi \rightarrow [\text{while } (b) \text{ do } T_0 \text{ od }] \psi$$

Composition of Programs (2)

$$|=, \{b \wedge \varphi\} T_0 \{ \psi \}$$

$$|=, \{\neg b \wedge \varphi\} T_1 \{ \psi \}$$

$$|=, \{\varphi\} \text{if } (b) \text{ then } T_0 \text{ else } T_1 \text{ fi } \{ \psi \}$$

$$\begin{aligned} b \wedge \varphi &\rightarrow [T_0] \psi \\ \neg b \wedge \varphi &\rightarrow [T_1] \psi \end{aligned}$$

$$\Leftrightarrow \varphi \rightarrow (b \rightarrow [T_0]\psi) \wedge (\neg b \rightarrow [T_1]\psi)$$

$$\Leftrightarrow \varphi \rightarrow [\text{if } (b) \text{ then } T_0 \text{ else } T_1 \text{ fi}] \psi$$

Composition of Programs (3)

$$\models, \{\varphi\} T_0 \{\varphi'\}$$
$$\models, \{\varphi'\} T_1 \{ \psi \}$$

$$\models, \{\varphi\} T_0;T_1 \{ \psi \}$$
$$\varphi \rightarrow [T_0] \varphi'$$
$$\Rightarrow \varphi \rightarrow [T_0][T_1] \psi$$
$$\varphi' \rightarrow [T_1] \psi$$
$$\Rightarrow \varphi \rightarrow [T_0;T_1] \psi$$

Assignments (4)

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

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

Consequence (5)

 $\varphi' \rightarrow \varphi$ $\{\varphi\} \vdash \{\psi\}$ $\psi \rightarrow \psi'$

 $\{\varphi'\} \vdash \{\psi'\}$

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)

{ $x \geq 0$ }

$y_1 = 0;$

$y_2 = 1;$

$y_3 = 1;$

while ($y_3 \leq x$) do

$y_1 = y_1 + 1;$

$y_2 = y_2 + 2;$

$y_3 = y_3 + y_2;$

od;

{ $y_1 * y_1 \leq x \wedge x < (y_1 + 1) * (y_1 + 1)$ }

Integer Square Root

T:

T0; T1

T0:

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

T11:

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

T1:

while (y3<=x) do
 T11
od;

Integer Square Root

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

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

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

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

Integer Square Root

{ $x \geq 0$ }

T0;

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

{ φ }

while ($y3 \leq x$) do

 T11;

od;

{ $\neg (y3 \leq x) \wedge \varphi$ }

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

{ $x \geq 0$ } T0 { $0 \leq x \wedge y1 = 0 \wedge y2 = 1 \wedge y3 = 1$ }

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

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

Proof of $\{ x \geq 0 \} \text{ TO } \{ 0 \leq x \wedge y1 = 0 \wedge y2 = 1 \wedge y3 = 1 \}$

- | | | |
|--|--|--|
| (1) $\{ x \geq 0 \}$ | $y1 = 0$ | $\{ 0 \leq x \wedge y1 = 0 \}$ |
| (2) $\{ 0 \leq x \wedge y1 = 0 \}$ | $y2 = 1$ | $\{ 0 \leq x \wedge y1 = 0 \wedge y2 = 1 \}$ |
| (3) $\{ 0 \leq x \wedge y1 = 0 \wedge y2 = 1 \}$ | $y3 = 1$ | $\{ 0 \leq x \wedge y1 = 0 \wedge y2 = 1 \wedge y3 = 1 \}$ |
| | | |
| (4) $\{ x \geq 0 \}$ | $y1 = 0 ; y2 = 1$ | $\{ 0 \leq x \wedge y1 = 0 \wedge y2 = 1 \}$ |
| (5) $\{ x \geq 0 \}$ | $y1 = 0 ; y2 = 1 ; y3 = 1$ | $\{ 0 \leq x \wedge y1 = 0 \wedge y2 = 1 \wedge y3 = 1 \}$ |
| | | |
| $\{ x \geq 0 \} \text{ TO }$ | $\{ 0 \leq x \wedge y1 = 0 \wedge y2 = 1 \wedge y3 = 1 \}$ | |

Proof of $\{y_3 \leq x \wedge \varphi\} T11 \{ y_1 * y_1 \leq x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1 + 1) * (y_1 + 1)\}$

$$y_3 \leq x \wedge \varphi \rightarrow (y_1 + 1) * (y_1 + 1) \leq x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1 + 1) * (y_1 + 1)$$

$$\{ (y_1 + 1) * (y_1 + 1) \leq x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1 + 1) * (y_1 + 1) \}$$

$$y_1 = y_1 + 1;$$

$$\{ y_1 * y_1 \leq x \wedge y_2 + 2 = 2 * y_1 + 1 \wedge y_3 = (y_1) * (y_1) \}$$

$$y_2 = y_2 + 2;$$

$$\{ y_1 * y_1 \leq x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1) * (y_1) \}$$

$$y_3 = y_3 + y_2;$$

$$\{ y_1 * y_1 \leq x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1 + 1) * (y_1 + 1) \}$$

$\{y_3 \leq x \wedge \varphi\} T11 \{ y_1 * y_1 \leq x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1 + 1) * (y_1 + 1)\}$

Integer Square Root

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

od;
{¬ (y3<=x) ∧ φ}
{ y1*y1<=x ∧ x<((y1+1)*(y1+1)) }
```

Summary

Problem:

{ $x \geq 0$ }

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

{ $y1 * y1 \leq x \wedge x < (y1+1)^2$ }

Let T_0 be

$y1=0; y2=1; y3=1;$

Let T_{11} be

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

Let T_1 be

$\text{while } (y3 \leq x) \text{ do } T_{11} \text{ od;}$

Let T be

$T_0; T_1$

Need to prove:

$\{x \geq 0\} \vdash \{ y1 * y1 \leq x \wedge x < (y1+1)^2 \}$

Let φ be $y_1 * y_1 \leq x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1 + 1) * (y_1 + 1)$

We have: $\{ \varphi(1/y_3)(1/y_2)(0/y_1) \} \text{ T0 } \{ \varphi \}$

Since we have $x \geq 0 \rightarrow \varphi(1/y_3)(1/y_2)(0/y_1)$,
we have $\{ x \geq 0 \} \text{ T0 } \{ \varphi \}$

We have: $\{ \varphi((y_3+y_2)/y_3)((y_2+2)/y_2)((y_1+1)/y_1) \} \text{ T11 } \{ \varphi \}$

Since we have

$(y_3 \leq x) \wedge \varphi \rightarrow \varphi((y_3+y_2)/y_3)((y_2+2)/y_2)((y_1+1)/y_1)$

we have $\{ (y_3 \leq x) \wedge \varphi \} \text{ T11 } \{ \varphi \}$

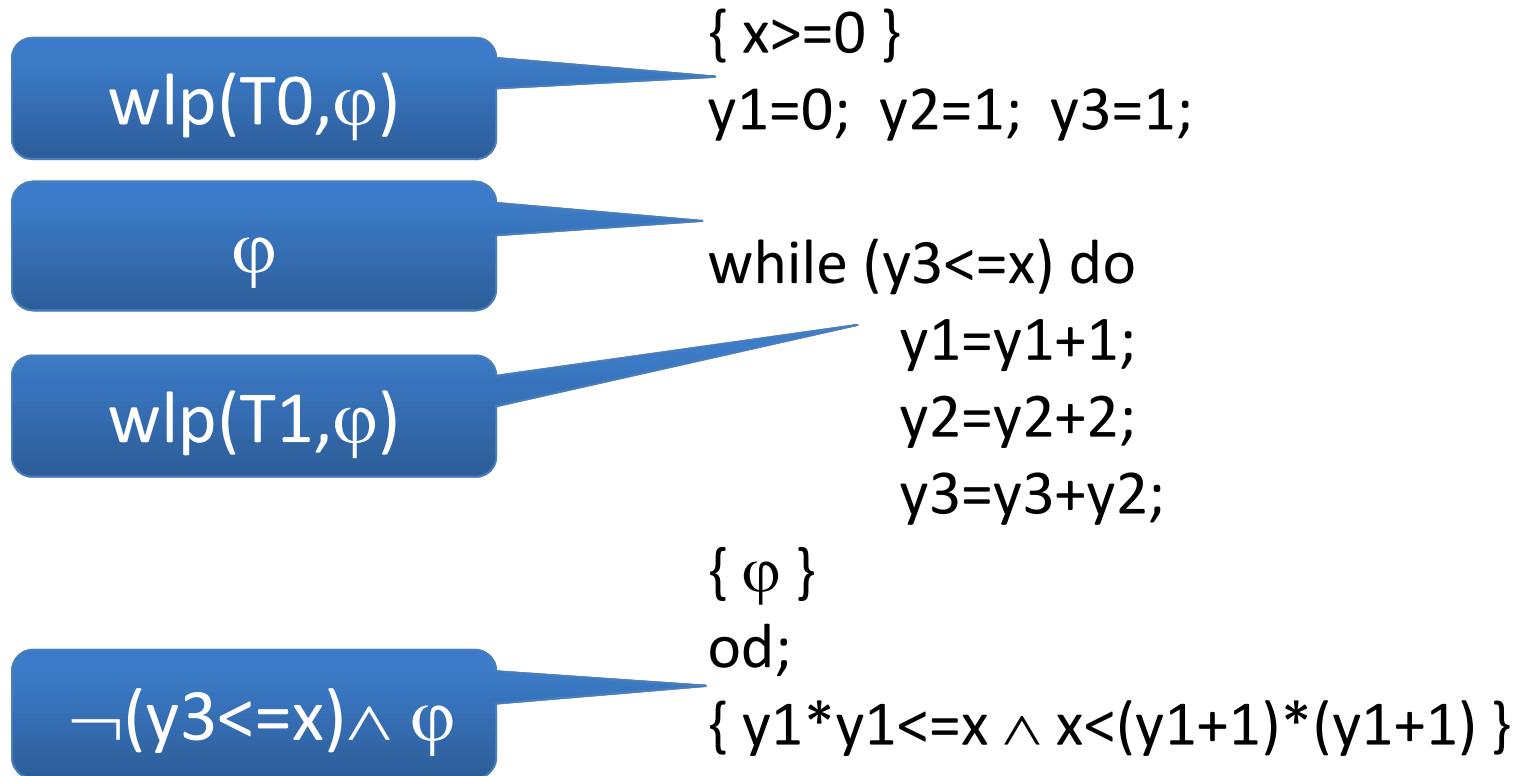
Therefore we have $\{ \varphi \} \text{ T1 } \{ \neg(y_3 \leq x) \wedge \varphi \}$

Therefore we have $\{ x \geq 0 \} \text{ T } \{ \neg(y_3 \leq x) \wedge \varphi \}$

Since $\neg(y_3 \leq x) \wedge \varphi \rightarrow y_1 * y_1 \leq x \wedge x < (y_1 + 1) * (y_1 + 1)$

we have $\{ x \geq 0 \} \text{ T } \{ y_1 * y_1 \leq x \wedge x < (y_1 + 1) * (y_1 + 1) \}$

Reasoning with wlp()



$$(y_3 \leq x) \wedge \varphi \rightarrow \text{wlp}(T_1, \varphi)$$

$$\neg(y_3 \leq x) \wedge \varphi \rightarrow y_1 * y_1 \leq x \wedge x < (y_1 + 1) * (y_1 + 1)$$

$$(x \geq 0) \rightarrow \text{wlp}(T_0, \varphi)$$

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

{ $x=c$ }

Pre-Condition

$y1:=0; y2:=1; y3:=1;$

while ($y3 \leq x$) {

$y1:=(y1+1); y2 := (y2+2);$

$y3:=(y2+y3);$

{ $x=c \wedge y1 * y1 \leq x \wedge y3 = (y1+1) * (y1+1) \wedge$

$y2=2 * y1 + 1$ }

}

{ $y1 * y1 \leq c \wedge c \leq (y1+1) * (y1+1)$ }

Invariant

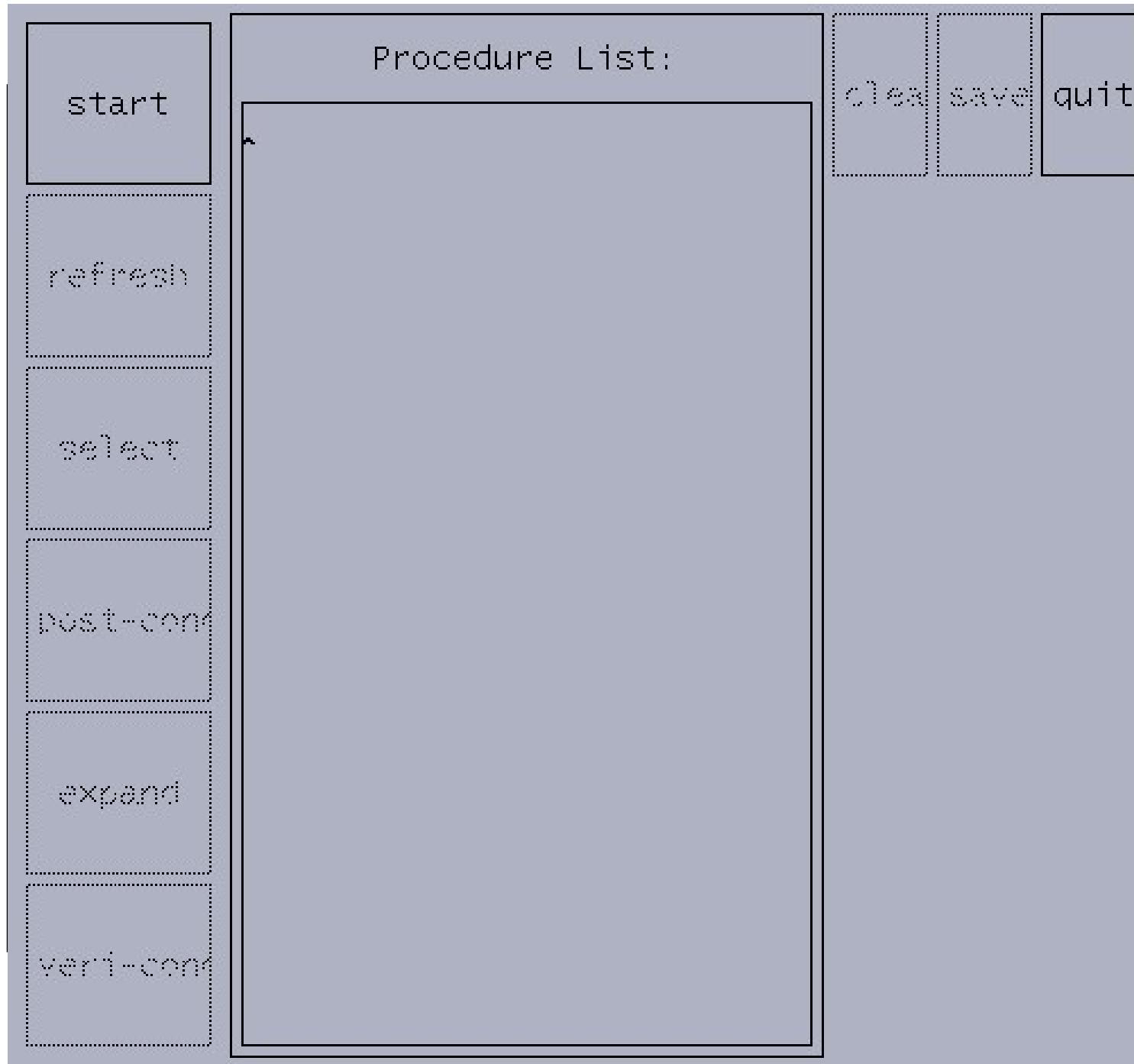
Post-Condition

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



Please Provide Procedure Text or a File Name:

ve.ex3

Ok

Cancel

add-proc
refresh
select
post-cond
expand
veri-cond

Procedure List:

wl

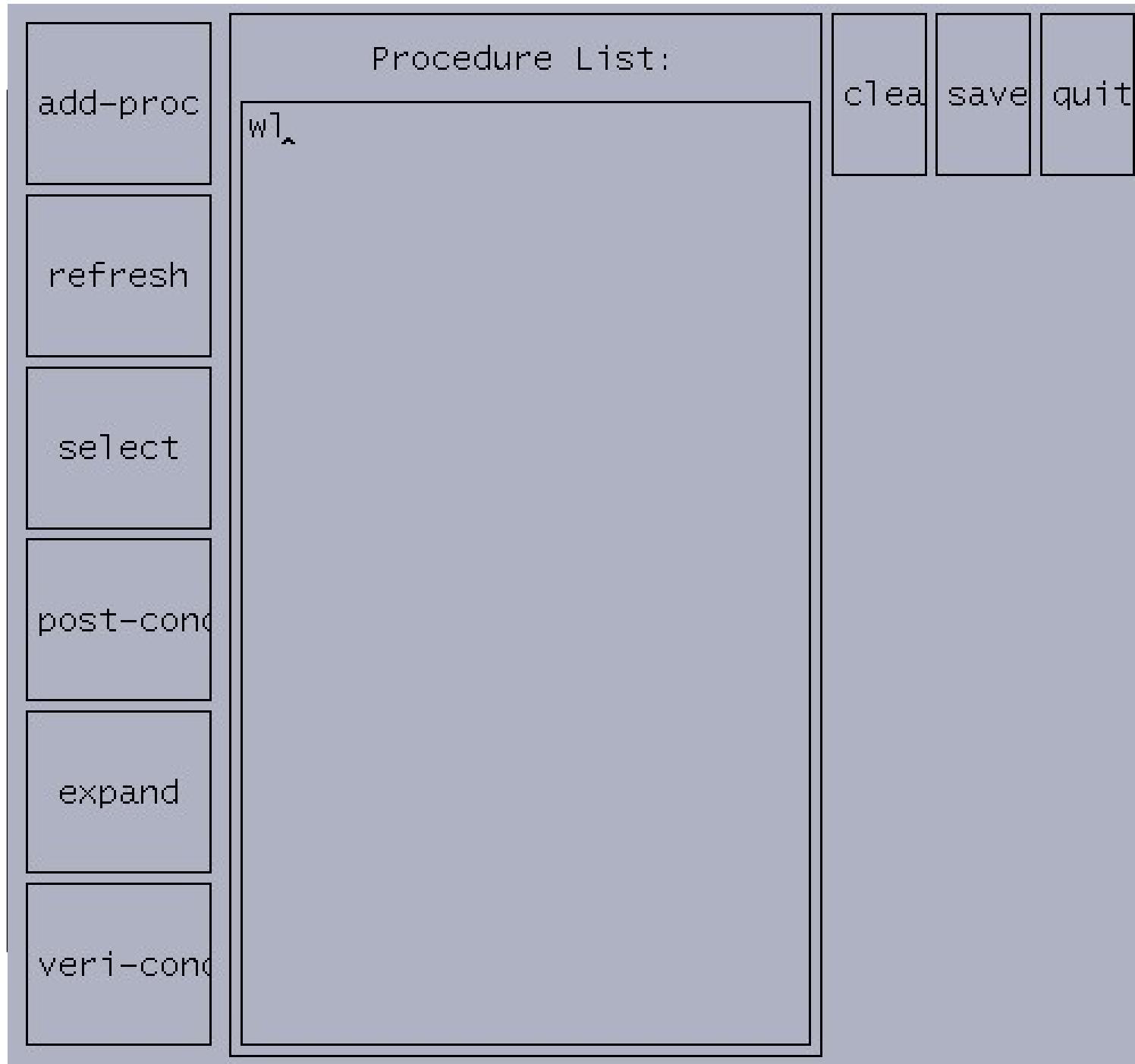
clear save quit

Procedure w1:

```
{ x=c }
%PROC w1(%INP/x:INT;%IOP/y1:INT)==
%LOC [y2,y3:INT]
%STM [
    LB=START=>$0y1=0^$0y2=1^$0y3=1^$OLB=12;
    * [LB=12^1e(y3,x)=>($OLB=13|$OLB=END)
        LB=13=>$0y1=+(y1,1)^$0y2=+(y2,2)^$OLB=14;
        LB=14=>$0y3=+(y2,y3)^$OLB=12;
        {x=c^1e(*(y1,y1),x)^y3=*(+y1,1),+y1,1))^y2=+
    ]
]
{ 1e(*(y1,y1),c)^1t(c,*(+y1,1),+y1,1))) }
```

Show

Cancel



Verification Conditions for w1:

```
(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

$$\begin{aligned} & y_3 \leq x \wedge x=c \wedge y_1 * y_1 \leq x \wedge y_3 = (y_1 + 1) * (y_1 + 1) \wedge y_2 = 2 * y_1 + 1 \\ \rightarrow & x=c \wedge (y_1 + 1) * (y_1 + 1) \leq x \wedge \\ & (y_2 + 2) + y_3 = ((y_1 + 1) + 1) * ((y_1 + 1) + 1) \wedge y_2 + 2 = 2 * (y_1 + 1) + 1 \end{aligned}$$
$$\begin{aligned} & \neg y_3 \leq x \wedge x=c \wedge y_1 * y_1 \leq x \wedge y_3 = (y_1 + 1) * (y_1 + 1) \wedge y_2 = 2 * y_1 + 1 \\ \rightarrow & y_1 * y_1 \leq c \wedge c < (y_1 + 1) * (y_1 + 1) \end{aligned}$$
$$\begin{aligned} & x=c \\ \rightarrow & x=c \wedge 0 * 0 \leq x \wedge 1 = (0 + 1) * (0 + 1) \wedge 1 = 2 * 0 + 1 \end{aligned}$$

Verification Conditions for w1:

```
le(*(+y1,1),+(y1,1)),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)),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

$$(y_1+1) * (y_1+1) \leq c, y_1 * y_1 \leq c \rightarrow 2 + (1 + y_1 * 2) = 1 + 2 * (y_1+1)$$

$$y_1 * y_1 \leq c \rightarrow c < (y_1+1) * (y_1+1), (y_1+1) * (y_1+1) \leq c$$

$$T \rightarrow 1 = 1 + 2 * 0$$

$$(y_1+1) * (y_1+1) \leq c, y_1 * y_1 \leq c$$

$$\rightarrow (2 + (1 + y_1 * 2)) + (y_1+1) * (y_1+1) = (1 + (y_1+1)) * (1 + (y_1+1))$$

$$T \rightarrow 0 * 0 \leq c$$

$$T \rightarrow 1 = (1 + 0) * (1 + 0)$$

Please Provide an Axiom:

$\text{le}(*(0,0), c)$

Simplify

Cancel

Verification Conditions for w1:

$\text{le}(*(+(\text{y1}, 1), +(\text{y1}, 1)), \text{c}), \text{le}(*(\text{y1}, \text{y1}), \text{c})$
=>
 $+(\text{2}, +(1, *(\text{y1}, 2))) = +(1, *(\text{2}, +(\text{y1}, 1)))$

$\text{le}(*(\text{y1}, \text{y1}), \text{c})$
=>
 $\text{lt}(\text{c}, *(+(\text{y1}, 1), +(\text{y1}, 1))), \text{le}(*(+(\text{y1}, 1), +(\text{y1}, 1)), \text{c})$

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

$\text{le}(*(+(\text{y1}, 1), +(\text{y1}, 1)), \text{c}), \text{le}(*(\text{y1}, \text{y1}), \text{c})$
=>
 $+((\text{2}, +(1, *(\text{y1}, 2))), *(+(\text{y1}, 1), +(\text{y1}, 1))) = *(+(1, +(\text{y1}, 1))$

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

Simplify

Cancel

Simplified Verification Conditions

$$(y_1+1) * (y_1+1) \leq c, y_1 * y_1 \leq c \rightarrow 2 + (1 + y_1 * 2) = 1 + 2 * (y_1+1)$$

$$y_1 * y_1 \leq c \rightarrow c < (y_1+1) * (y_1+1), (y_1+1) * (y_1+1) \leq c$$

$$T \rightarrow 1 = 1 + 2 * 0$$

$$(y_1+1) * (y_1+1) \leq c, y_1 * y_1 \leq c$$

$$\rightarrow (2 + (1 + y_1 * 2)) + (y_1+1) * (y_1+1) = (1 + (y_1+1)) * (1 + (y_1+1))$$

$$T \rightarrow 1 = (1 + 0) * (1 + 0)$$

(III.b) Proof Rules (TC)

Extended Hoare Logic:

$[\varphi] \top [\psi]$

$\vdash_L [\varphi] \top [\psi],$

iff

$I(\varphi)(\sigma) \rightarrow \exists \sigma'. (((\top; \varepsilon, \sigma) \rightarrow^* (\varepsilon, \sigma')) \wedge I(\psi)(\sigma'))$

Composition of Programs (1)

w, W, t:

$$\models_1 \phi \wedge b \rightarrow w(t/x)$$

$$\models_1 [\phi \wedge b \wedge t=v] \text{ T0 } [\phi \wedge t < v]$$

$$\models_1 [\phi] \text{ while } (b) \text{ do T0 od } [\neg b \wedge \phi]$$

$$\varphi \rightarrow \phi$$

$$\models_1 \dots \dots$$

$$\models_1 \dots \dots$$

$$\phi \wedge \neg b \rightarrow \psi$$

$$\models_1 [\varphi] \text{ while } (b) \text{ do T0 od } [\psi]$$

Composition of Programs (2)

$$\models, [b \wedge \varphi] T_0 [\psi]$$
$$\models, [\neg b \wedge \varphi] T_1 [\psi]$$

$$\models, [\varphi] \text{ if } (b) \text{ then } T_0 \text{ else } T_1 \text{ fi } [\psi]$$

Composition of Programs (3)

$$\vdash , [\varphi] T_0 [\varphi']$$
$$\vdash , [\varphi'] T_1 [\psi]$$

$$\vdash , [\varphi] T_0;T_1 [\psi]$$

Assignments (4)

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

$| = , [\varphi] x:=t [\psi]$

Consequence (5)

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

 $[\varphi'] \top [\psi']$

例子 

Integer Square Root (TC)

[$x \geq 0$]

$y1 = 0;$

$y2 = 1;$

$y3 = 1;$

while ($y3 \leq x$) do

$y1 = y1 + 1;$

$y2 = y2 + 2;$

$y3 = y3 + y2;$

od;

[$y1 * y1 \leq x \wedge x < (y1 + 1) * (y1 + 1)$]

Integer Square Root

T:

T0; T1

T0:

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

T11:

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

T1:

while (y3<=x) do
 T11
od;

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 ∧ y1=0 ∧ y2=1 ∧ y3 =1]

[y3<=x ∧ φ ∧ ?=v] T11 [y1*y1<=x ∧ y2=2*y1+1 ∧ y3 =(y1+1)*(y1+1) ∧ ?<v]

[φ] while (y3<=x) do T11; od [¬ (y3<=x) ∧ φ]

Integer Square Root

W: NAT

w: $x \geq 0$

$t = x + 1 - y^3$

$y^3 \leq x \wedge \varphi \rightarrow (x + 1 - y^3) > 0$ and

$[y^3 \leq x \wedge \varphi \wedge t = v] \text{ T11 } [y_1 * y_1 \leq x \wedge y_2 = 2 * y_1 + 1 \wedge y^3 = (y_1 + 1)^*(y_1 + 1) \wedge t < v]$

$[\varphi] \text{ while } (y^3 \leq x) \text{ do T11; od } [\neg (y^3 \leq x) \wedge \varphi]$

Summary

Problem:

[$x \geq 0$]

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

[$y1 * y1 \leq x \wedge x < (y1+1)^2$]

Let T_0 be

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

Let T_{11} be

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

Let T_1 be

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

Let T be

```
T0; T1
```

Need to prove: [$x \geq 0$] \vdash [$y1 * y1 \leq x \wedge x < (y1+1)^2$]

Let φ be $y_1 * y_1 \leq x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1 + 1) * (y_1 + 1)$

We have: $[\varphi(1/y_3)(1/y_2)(0/y_1)] \text{ T0 } [\varphi]$

Since we have $x \geq 0 \rightarrow \varphi(1/y_3)(1/y_2)(0/y_1)$,
we have $[x \geq 0] \text{ T0 } [\varphi]$

Let W be NAT, and w be $x \geq 0$

Let t be $x+1-y_3$. We have $y_3 \leq x \wedge \varphi \rightarrow (x+1-y_3) \geq 0$

Since $y_3 \leq x \wedge \varphi \wedge t = v \rightarrow$
 $(\varphi \wedge t < v) ((y_3+y_2)/y_3)((y_2+2)/y_2)((y_1+1)/y_1)$

we have $[y_3 \leq x \wedge \varphi \wedge t = v] \text{ T11 } [\varphi \wedge t < v]$

Therefore $[\varphi] \text{ T1 } [\neg(y_3 \leq x) \wedge \varphi]$

Therefore $[x \geq 0] \text{ T } [\neg(y_3 \leq x) \wedge \varphi]$

Therefore $[x \geq 0] \text{ T } [y_1 * y_1 \leq x \wedge x < (y_1 + 1) * (y_1 + 1)]$

(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.

$SP := \begin{array}{l} x := t \mid \\ P_1; P_2 \mid \\ \text{if } (b) \{ P_1 \} \text{ else } \{ P_2 \} \mid \\ \text{while } (b) \{ P_1 \} \end{array}$

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

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

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

$\models_I \{\varphi\}P\{\psi\}$ if

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

$\models \{\varphi\}P\{\psi\}$ if

for all I , $\models_I \{\varphi\}P\{\psi\}$

Logical Consequence

Let W be a set of formulas.

$W \models \{\varphi\}P\{\psi\}$ if

for every I ,

if $\models_I W$, then we have $\models_I \{\varphi\}P\{\psi\}$.

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.

$I(\{\text{true}\} \text{ while } (x \neq 10) \{ x := x + 1 \} \{x = 10\})\sigma = \text{true}$
for all σ .

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

Example (3)

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

Let I be given as usual. Then $\Vdash_I \{\varphi\}P\{\psi\}$

Let $W = \{y + 1 > y\}$.

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

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

$W \Vdash \{\varphi\}P\{\psi\}$

On Hoare Logic

Hoare logic is a first order logic:

$$\vdash_I \{\varphi\} P \{\psi\} \Leftrightarrow \text{th}(I) \models \{\varphi\} P \{\psi\}$$

Hoare Logic (1)

- Axiom:

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

Hoare Logic (2)

- Sequential Composition

$$\{\psi_0\} P_1 \{\psi_1\}$$
$$\{\psi_1\} P_2 \{\psi_2\}$$

$$\{\psi_0\} P_1;P_2 \{\psi_2\}$$

Hoare Logic (3)

- Conditional Composition

$$\{b \wedge \varphi\} P_1 \{\psi\}$$
$$\{\neg b \wedge \varphi\} P_2 \{\psi\}$$

$$\{\varphi\} \text{ if } (b) \{P_1\} \text{ else } \{P_2\} \{\psi\}$$

Hoare Logic (4)

- Loop Composition

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

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

Hoare Logic (5)

- Consequence

$$\varphi' \rightarrow \varphi$$

$$\{\varphi\} P \{\psi\}$$

$$\psi \rightarrow \psi'$$

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

Soundness

- The proof system is sound:

$$\vdash_I \{\varphi\} P \{\psi\} \rightarrow \models_I \{\varphi\} P \{\psi\}$$

by structural induction.

Lemma

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

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

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

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

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

\rightarrow

$| =_I \{\varphi\} P1; P2 \{\psi\}$

Lemma

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

→

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

Lemma

$| =_l \{\varphi \wedge b\} P1 \{\varphi\}$

\rightarrow

$| =_l \{\varphi\} \text{while } (b) \{P1\} \{\varphi \wedge \neg b\}$

Lemma

$\Vdash_1 \varphi' \rightarrow \varphi, \Vdash_1 \{\varphi\} \Vdash \{\psi\}$ and $\Vdash_1 \psi \rightarrow \psi'$

\rightarrow

$\Vdash_1 \{\varphi'\} \Vdash \{\psi'\}$

Relative Completeness

- The proof system is relatively complete.

$$\Vdash_{\mathcal{I}} \{\varphi\} \text{ P } \{\psi\} \rightarrow \Vdash_{\mathcal{I}} \{\neg\varphi\} \text{ P } \{\psi\}$$

- 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

$$| =_1 \{\varphi\} \mathsf{P} \{\psi\} \rightarrow | \dashv_1 \{\varphi\} \mathsf{P} \{\psi\}$$

by structural induction.

Lemma

$$\Vdash_{\mathcal{I}} \{\varphi\} x := t \{ \psi \} \rightarrow \Vdash_{\mathcal{I}} \varphi \rightarrow \psi[x/t]$$

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

Lemma

$|=_{\mathcal{I}} \{\phi\} P1; P2 \{\psi\}$, and $\mathcal{I}(\psi')$ is WLP of $P2$ and $\mathcal{I}(\psi)$



$|=_{\mathcal{I}} \{\phi\} P1 \{\psi'\}$ and $|=_{\mathcal{I}} \{\psi'\} P2 \{\psi\}$

Lemma

$|=_{\mathcal{I}} \{\varphi\} \text{ if } (b) \{ P_1 \} \text{ else } \{ P_2 \} \{\psi\}$

\rightarrow

$|=_{\mathcal{I}} \{b \wedge \varphi\} P_1 \{\psi\} \text{ and } |=_{\mathcal{I}} \{\neg b \wedge \varphi\} P_2 \{\psi\}$

Lemma

$\models_1 \{\varphi\} \text{while } (b) \{P_1\} \{\psi\}$

→

$\models_1 \{\varphi\} \text{if } (b) \{ P_1; \text{while } (b) \{P_1\} \} \text{else } \{x:=x\} \{\psi\}$

$\models_1 \{\varphi\} \text{while } (b) \{P_1\} \{\psi\}$, and

$I(\psi')$ is the WLP of “while $(b) \{P_1\}$ ” and $I(\psi)$

→

$\models_1 \varphi \rightarrow \psi'$, $\models_1 \{\psi' \wedge b\} P_1 \{\psi'\}$, and $\models_1 \psi' \wedge \neg b \rightarrow \psi$

Examples

Example 1

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

$y := 1;$

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

$\text{while } (x > 0) \{$

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

$y := y^* x;$

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

$x := x - 1$

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

}

$\{y = n!\}$

Example 2

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

while ($\neg(x = y)$) {

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

else { $y := y - x$ } $\{ \text{gcd}(x, y) = \text{gcd}(a, b) \}$

}

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

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

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

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

$\models_I [\varphi]P[\psi]$ if

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

$\models [\varphi]P[\psi]$ if

for all I , $\models_I [\varphi]P[\psi]$

Logical Consequence

Let W be a set of formulas.

$W \models [\varphi]P[\psi]$ if
for every I such that, $\models_I W$,
we have $\models_I [\varphi]P[\psi]$.

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

$$[\psi_0] P_1 [\psi_1]$$
$$[\psi_1] P_2 [\psi_2]$$

$$[\psi_0] P_1;P_2 [\psi_2]$$

-

Extended Hoare Logic (3)

- Conditional Composition

$$[b \wedge \varphi] P_1 [\psi]$$
$$[\neg b \wedge \varphi] P_2 [\psi]$$

$$[\varphi] \text{ if } (b) \{P_1\} \text{ else } \{P_2\} [\psi]$$

Extended Hoare Logic (4)

- Loop Composition

$$b \wedge \psi \rightarrow w[x/t] \quad [b \wedge \psi \wedge t=v] \vdash [\psi \wedge t < v]$$

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

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

Extended Hoare Logic (5)

- Consequence

$$\varphi' \rightarrow \varphi$$

$$[\varphi] P [\psi]$$

$$\psi \rightarrow \psi'$$

$$[\varphi'] P [\psi']$$

Examples

Example 1

$[x \geq 0 \wedge x = n]$

$y := 1;$

$\text{while } (x > 0) \{$ { $t = (x); w = (x \geq 0)$ }

$y := y * x;$

$x := x - 1$

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

}

$[y = n!]$

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 { $y := y - x$ }

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

}

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

Verification Condition Generation

$\text{vcg}(\phi, T, \psi)$:

$\Phi = \{\}; p = \text{vc}(\phi, T, \psi); \Phi = \Phi \cup \{p\}; \text{return } \Phi$

$\text{vc}(\phi, T; x := e, \psi) \equiv \text{vc}(\phi, T, \psi(e/x))$

$\text{vc}(\phi, T; \text{if } (b) \text{ then } T_0 \text{ else } T_1 \text{ fi}, \psi) \equiv$
 $\text{vc}(\phi, T, \text{vc}(b, T_0, \psi) \wedge \text{vc}(\neg b, T_1, \psi))$

$\text{vc}(\phi, T; \text{while } (b) \text{ do } T_0 \{\varphi\} \text{ od}, \psi) \equiv \text{vc}(\phi, T, \varphi)$

UPDATE: $\Phi = \Phi \cup \{\text{vc}(b \wedge \varphi, T_0, \varphi), \neg b \wedge \varphi \rightarrow \psi\}$

$\text{vc}(\phi, \varepsilon, \psi) \equiv \phi \rightarrow \psi$

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

Let T be $T0; T1$

Integer Square Root (PC)

$$\begin{aligned} \forall c(x >= 0, T, y_1 * y_1 <= x \wedge x < (y_1 + 1) * (y_1 + 1)) &\equiv \\ \forall c(x >= 0, T_0, y_1 * y_1 <= x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1 + 1) * (y_1 + 1)) &\equiv \\ \forall c(x >= 0, \varepsilon, 0 <= x \wedge 1 = 2 * 0 + 1 \wedge 1 = (0 + 1) * (0 + 1)) &\equiv \\ x >= 0 \rightarrow 0 <= x \wedge 1 = 2 * 0 + 1 \wedge 1 = (0 + 1) * (0 + 1) \end{aligned}$$

Integer Square Root (PC)

$$\begin{aligned} \text{vc}(y_3 \leq x \wedge \varphi, T11, \varphi) &\equiv \\ \text{vc}(y_3 \leq x \wedge \varphi, \varepsilon, \varphi((y_3+y_2)/y_3)((y_2+2)/y_2)((y_1+1)/y_1)) &\equiv \\ y_3 \leq x \wedge \varphi \rightarrow \varphi((y_3+y_2)/y_3)((y_2+2)/y_2)((y_1+1)/y_1) \end{aligned}$$
$$\neg y_3 \leq x \wedge \varphi \rightarrow y_1 * y_1 \leq x \wedge x < (y_1+1) * (y_1+1)$$

$$\begin{aligned} \text{vc}(x >= 0, T, y_1 * y_1 \leq x \wedge x < (y_1+1) * (y_1+1)) &\equiv \\ \text{vc}(x >= 0, T0, y_1 * y_1 \leq x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1+1) * (y_1+1)) &\equiv \\ \text{vc}(x >= 0, \varepsilon, 0 \leq x \wedge 1 = 2 * 0 + 1 \wedge 1 = (0+1) * (0+1)) &\equiv \\ x >= 0 \rightarrow 0 \leq x \wedge 1 = 2 * 0 + 1 \wedge 1 = (0+1) * (0+1) \end{aligned}$$

Verification Condition Generation (TC)

$\text{vcg}(\phi, T, \psi)$:

$\Phi = \{\}; p = \text{vc}(\phi, T, \psi); \Phi = \Phi \cup \{p\}; \text{return } \Phi$

$\text{vc}(\phi, T; x := e, \psi) \equiv \text{vc}(\phi, T, \psi(e/x))$

$\text{vc}(\phi, T; \text{if } (b) \text{ then } T_0 \text{ else } T_1 \text{ fi}, \psi) \equiv$
 $\text{vc}(\phi, T, \text{vc}(b, T_0, \psi) \wedge \text{vc}(\neg b, T_1, \psi))$

$\text{vc}(\phi, T; \text{while } (b) \text{ do } \{t, w\} T_0 \{\varphi\} \text{ od}, \psi) \equiv \text{vc}(\phi, T, \varphi)$

UPDATE:

$\Phi = \Phi \cup \{\text{vc}(b \wedge \varphi \wedge t = v, T_0, \varphi \wedge t < v), \neg b \wedge \varphi \rightarrow \psi, b \wedge \varphi \rightarrow w(x/t)\}$

$\text{vc}(\phi, \varepsilon, \psi) \equiv \phi \rightarrow \psi$

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

Let T be T0; T1

Integer Square Root (TC)

$$\begin{aligned} \forall c(x >= 0, T, y_1 * y_1 <= x \wedge x < (y_1 + 1) * (y_1 + 1)) &\equiv \\ \forall c(x >= 0, T_0, y_1 * y_1 <= x \wedge y_2 = 2 * y_1 + 1 \wedge y_3 = (y_1 + 1) * (y_1 + 1)) &\equiv \\ \forall c(x >= 0, \varepsilon, 0 <= x \wedge 1 = 2 * 0 + 1 \wedge 1 = (0 + 1) * (0 + 1)) &\equiv \\ x >= 0 \rightarrow 0 <= x \wedge 1 = 2 * 0 + 1 \wedge 1 = (0 + 1) * (0 + 1) \end{aligned}$$

Integer Square Root (TC)

$$\text{vc}(\text{y3} \leq \text{x} \wedge \varphi \wedge \text{x} + 1 - \text{y3} = \text{v}, \text{T11}, \varphi \wedge \text{x} + 1 - \text{y3} < \text{v}) \equiv$$

...

$$\neg \text{y3} \leq \text{x} \wedge \varphi \rightarrow \text{y1} * \text{y1} \leq \text{x} \wedge \text{x} < (\text{y1} + 1) * (\text{y1} + 1)$$
$$\text{y3} \leq \text{x} \wedge \varphi \rightarrow \text{x} + 1 - \text{y3} \geq 0$$
$$\text{vc}(\text{x} \geq 0, \text{T}, \text{y1} * \text{y1} \leq \text{x} \wedge \text{x} < (\text{y1} + 1) * (\text{y1} + 1)) \equiv$$
$$\text{vc}(\text{x} \geq 0, \text{T0}, \text{y1} * \text{y1} \leq \text{x} \wedge \text{y2} = 2 * \text{y1} + 1 \wedge \text{y3} = (\text{y1} + 1) * (\text{y1} + 1)) \equiv$$
$$\text{vc}(\text{x} \geq 0, \varepsilon, 0 \leq \text{x} \wedge 1 = 2 * 0 + 1 \wedge 1 = (0 + 1) * (0 + 1)) \equiv$$
$$\text{x} \geq 0 \rightarrow 0 \leq \text{x} \wedge 1 = 2 * 0 + 1 \wedge 1 = (0 + 1) * (0 + 1)$$

(V) Summary

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

练习1

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

给定 \mathbf{l} 为 \mathbf{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 $\mathbf{l} \models_l \{ y=k*a+l*b \wedge (x=i*a+j*b) \} T \{ x=i*a+j*b \}$ 。

练习2

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

给定以下程序 T :

```
i:=1; j:=0; k:=0; l:=1;
while  $\neg(x = y)$  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:

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

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