Lógica
Aula 21

Renata Wassermann

renata@ime.usp.br

2020

Fases da verificação

1. Partindo de uma descrição informal D, gerar φ_D.
2. Escrever programa P.
3. Provar que $P \vdash \varphi_D$.

Triplas de Hoare

Uma especificação é dada por uma tripla

$$(|\varphi|)P(|\psi|)$$

"Ao rodar P num estado que satisfaz φ, chegamos a um estado que satisfaz ψ."

φ é a pré-condição (pode ser vazia $(|\top|)$)
ψ é a pós-condição

Triplas de Hoare

"Se a entrada x é um número positivo, calcule um número y cujo quadrado seja menor que a entrada x."

Triplas de Hoare

"Se a entrada x é um número positivo, calcule um número y cujo quadrado seja menor que a entrada x."

$$(|x > 0|)P(|y.y < x|)$$

Triplas de Hoare

"Se a entrada x é um número positivo, calcule um número y cujo quadrado seja menor que a entrada x."

$$(|x > 0|)P(|y.y < x|)$$

```
P1:     y = 0
```

Triplas de Hoare

"Se a entrada x é um número positivo, calcule um número y cujo quadrado seja menor que a entrada x."

$$(|x > 0|)P(|y.y < x|)$$

```
P1:     y = 0
```

```
P2:     y = 0;
        while (y*y<x) {
           y = y+1;
        y = y-1;
```

Correção Parcial

$$\models_{par} (|\varphi|)P(|\psi|)$$

$$\Updownarrow$$

Para qualquer estado satisfazendo φ, se P termina, o estado final satisfaz ψ.

Correção Parcial

$$\models_{par} (|\varphi|)P(|\psi|)$$

$$\Updownarrow$$

Para qualquer estado satisfazendo φ, se P termina, o estado final satisfaz ψ.

Qualquer programa que não para é parcialmente correto.

Correção Total

$$\models_{tot} (|\varphi|)P(|\psi|)$$
$$\Updownarrow$$
$$\models_{par} (|\varphi|)P(|\psi|) \text{ e } P \text{ termina.}$$

Linguagem para Descrição de Programas

Expressões inteiras:

$$E ::= n|x|(-E)|(E+E)|(E*E)$$

Expressões booleanas:

$$B ::= \texttt{true}|\texttt{false}|(!B)|(B\&B)|(B||B)|(E<E)$$

$$E_1 == E_2 \equiv !(E_1 < E_2)\&!(E_2 < E_1)$$

Comandos:

$$C ::= x = E|C;C|\texttt{if}\ \ B\{C\}\texttt{else}\ \ \{C\}|\texttt{while}\ \ B\{C\}$$

Variáveis lógicas

Não aparecem no programa!

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

$(|x = 3|)\texttt{Soma}\ (|z = 6|)$

$(|x = x_0 \land x \geq 0|)\texttt{Soma}\ (|z = (x_0.(x_0 + 1))/2|)$

Estado dá valor para variáveis do programa, mas não para variáveis lógicas.

Cálculo para Prova de Correção

Composição

$$\frac{(|\varphi|)\,C_1\,(|\chi|) \qquad (|\chi|)\,C_2\,(|\psi|)}{(|\varphi|)\,C_1;\,C_2\,(|\psi|)}$$

Cálculo para Prova de Correção

Composição

$$\frac{(|\varphi|)C_1(|\chi|) \qquad (|\chi|)C_2(|\psi|)}{(|\varphi|)C_1;C_2(|\psi|)}$$

Atribuição

$$\frac{}{(|\psi[E/x]|)x = E(|\psi|)}$$

Para escrever uma prova

$$\vdash_{par} (|\varphi|)P(|\psi|)$$

Seja P:

C_1;

C_2;

.

.

.

C_n;

Para escrever uma prova

$$\vdash_{par} (|\varphi|)P(|\psi|)$$

Seja P:

$C_1;$

$C_2;$

.

.

.

$C_n;$

$(|\varphi_n|)$

Para escrever uma prova

$$\vdash_{par} (|\varphi|)P(|\psi|)$$

Seja P:

$C_1;$

$C_2;$

.

.

.

$(|\varphi_{n-1}|)$

$C_n;$

$(|\varphi_n|)$

Para escrever uma prova

$$\vdash_{par} (|\varphi|)P(|\psi|)$$

Seja P:

$C_1;$

$C_2;$
  $(|\varphi_2|)$
.
.
.
  $(|\varphi_{n-1}|)$
$C_n;$
  $(|\varphi_n|)$

Para escrever uma prova

$$\vdash_{par} (|\varphi|) P (|\psi|)$$

Seja P:

$$\begin{array}{l} C_1; \\ \quad (|\varphi_1|) \\ C_2; \\ \quad (|\varphi_2|) \\ . \\ . \\ . \\ \quad (|\varphi_{n-1}|) \\ C_n; \\ \quad (|\varphi_n|) \end{array}$$

Para escrever uma prova

$$\vdash_{par} (|\varphi|)P(|\psi|)$$

Seja P:

$$\begin{array}{l} \quad (|\varphi_0|) \\ C_1; \\ \quad (|\varphi_1|) \\ C_2; \\ \quad (|\varphi_2|) \\ . \\ . \\ . \\ \quad (|\varphi_{n-1}|) \\ C_n; \\ \quad (|\varphi_n|) \end{array}$$

Para escrever uma prova

Cada φ_i deve valer no ponto em que aparece.

Para escrever uma prova

Cada φ_i deve valer no ponto em que aparece.

Cada transição

$$\begin{array}{l} \quad (|\varphi_{i-1}|) \\ C_i; \\ \quad (|\varphi_i|) \end{array}$$

usa alguma regra do cálculo e parte de φ_i para calcular a *pré-condição mais fraca* φ_{i-1}.

Cálculo para Prova de Correção

Implicação

$$\frac{\vdash \varphi' \to \varphi \qquad (|\varphi|)C(|\psi|) \qquad \vdash \psi \to \psi'}{(|\varphi'|)C(|\psi'|)}$$

Esta regra é importante para completar provas usando lógica de primeira ordem e aritmética de inteiros.

Cálculo para Prova de Correção

Implicação

$$\frac{\vdash \varphi' \to \varphi \qquad (|\varphi|)\,C\,(|\psi|) \qquad \vdash \psi \to \psi'}{(|\varphi'|)\,C\,(|\psi'|)}$$

Esta regra é importante para completar provas usando lógica de primeira ordem e aritmética de inteiros.

Ela permite escrever

$$(|\varphi|)$$
$$(|\varphi'|)$$

quando $\vdash \varphi \to \varphi'$.

Exemplos

$$\vdash_{par} (|y < 3|) y = y + 1 (|y < 4|)$$

```
y = y + 1;
```

Exemplos

$$\vdash_{par} (|y < 3|) y = y + 1 (|y < 4|)$$

```
y = y + 1;
```

$(|y < 4|)$

Exemplos

$$\vdash_{par} (|y < 3|) y = y + 1 (|y < 4|)$$

$(|y + 1 < 4|)$

```
y = y + 1;
```

$(|y < 4|)$ Atribuição

Exemplos

$$\vdash_{par} (|y < 3|) y = y + 1 (|y < 4|)$$

$$\begin{array}{ll} \quad (|y < 3|) & \\ \quad (|y + 1 < 4|) & \text{Implicação} \\ \texttt{y = y + 1;} & \\ \quad (|y < 4|) & \text{Atribuição} \end{array}$$

Exemplos

$$\vdash_{par} (|\top|)P(|z = x + y|)$$

```
z = x;

z = z + y;
```

Exemplos

$$\vdash_{par} (|\top|) P (|z = x + y|)$$

```
z = x;

z = z + y;
```

$(|z = x + y|)$

Exemplos

$$\vdash_{par} (|\top|)P(|z = x + y|)$$

```
z = x;
```
$(|z + y = x + y|)$
```
z = z + y;
```
$(|z = x + y|)$ Atribuição

Exemplos

$$\vdash_{par} (|\top|)P(|z = x + y|)$$

$(|x + y = x + y|)$

```
z = x;
```

$(|z + y = x + y|)$ Atribuição

```
z = z + y;
```

$(|z = x + y|)$ Atribuição

Exemplos

$$\vdash_{par} (|\top|)P(|z = x + y|)$$

$(|\top|)$

$(|x + y = x + y|)$ Implicação

```
z = x;
```

$(|z + y = x + y|)$ Atribuição

```
z = z + y;
```

$(|z = x + y|)$ Atribuição

Cálculo para Prova de Correção

If

$$\frac{(|\varphi \wedge B|)C_1(|\psi|) \qquad (|\varphi \wedge \neg B|)C_2(|\psi|)}{(|(\varphi)|)\texttt{if}\ \ B\ \{C_1\}\ \texttt{else}\ \ \{C_2\}(|\psi|)}$$

Cálculo para Prova de Correção

If

$$\frac{(|\varphi \wedge B|)C_1(|\psi|) \qquad (|\varphi \wedge \neg B|)C_2(|\psi|)}{(|(\varphi)|)\texttt{if}\ \ B\ \{C_1\}\ \texttt{else}\ \ \{C_2\}(|\psi|)}$$

If'

$$\frac{(|\varphi_1|)C_1(|\psi|) \qquad (|\varphi_2|)C_2(|\psi|)}{(|(B \rightarrow \varphi_1) \wedge (\neg B \rightarrow \varphi_2)|)\texttt{if}\ \ B\ \{C_1\}\ \texttt{else}\ \ \{C_2\}(|\psi|)}$$

Exemplo

$(|\top|)$

```
a = x + 1;
if (a - 1 == 0) {
  y = 1;
} else {
  y = a;
}
```

$(|y = x + 1|)$

Exemplo

$(|\top|)$

```
a = x + 1;
```

```
if (a - 1 == 0) {
```

```
  y = 1;
```

$(|y = x + 1|)$

```
} else {
```

```
  y = a;
```

$(|y = x + 1|)$

```
}
```

$(|y = x + 1|)$ If'

Exemplo

$(|\top|)$

```
a = x + 1;

if (a - 1 == 0) {

  y = 1;
```
$(|y = x + 1|)$
```
} else {
```
$(|a = x + 1|)$
```
  y = a;
```
$(|y = x + 1|)$ Atribuição
```
}
```
$(|y = x + 1|)$ If'

Exemplo

$(|\top|)$

```
a = x + 1;

if (a - 1 == 0) {

  y = 1;
```
  $(|y = x + 1|)$
```
} else {
```
  $(|a = x + 1|)$ If' (φ_2)
```
  y = a;
```
  $(|y = x + 1|)$ Atribuição
```
}
```
  $(|y = x + 1|)$ If'

Exemplo

$(|\top|)$

```
a = x + 1;

if (a - 1 == 0) {
```
$(|1 = x + 1|)$
```
  y = 1;
```
$(|y = x + 1|)$ Atribuição
```
} else {
```
$(|a = x + 1|)$ If' (φ_2)
```
  y = a;
```
$(|y = x + 1|)$ Atribuição
```
}
```
$(|y = x + 1|)$ If'

Exemplo

$(|\top|)$

```
a = x + 1;

if (a - 1 == 0) {
```
$(|1 = x + 1|)$ If' (φ_1)
```
  y = 1;
```
$(|y = x + 1|)$ Atribuição
```
} else {
```
$(|a = x + 1|)$ If' (φ_2)
```
  y = a;
```
$(|y = x + 1|)$ Atribuição
```
}
```
$(|y = x + 1|)$ If'

Exemplo

$(|\top|)$

```
a = x + 1;
```

$(|((a-1=0) \to (1=x+1)) \wedge$
$(\neg(a-1=0) \to (a=x+1))|)$

```
if (a - 1 == 0) {
```

$(|1=x+1|)$ If' (φ_1)

```
  y = 1;
```

$(|y=x+1|)$ Atribuição

```
} else {
```

$(|a=x+1|)$ If' (φ_2)

```
  y = a;
```

$(|y=x+1|)$ Atribuição

```
}
```

$(|y=x+1|)$ If'

Exemplo

$(|\top|)$

$(|((x+1-1=0) \to (1=x+1)) \wedge$
$(\neg(x+1-1=0) \to (x+1=x+1))|)$

```
a = x + 1;
```

$(|((a-1=0) \to (1=x+1)) \wedge$
$(\neg(a-1=0) \to (a=x+1))|)$ Atribuição

```
if (a - 1 == 0) {
```

$(|1=x+1|)$ If' (φ_1)

```
y = 1;
```

$(|y=x+1|)$ Atribuição

```
} else {
```

$(|a=x+1|)$ If' (φ_2)

```
y = a;
```

$(|y=x+1|)$ Atribuição

```
}
```

$(|y=x+1|)$ If'

Exemplo

$(|\top|)$
$(|((x+1-1=0) \to (1=x+1)) \wedge$
$(\neg(x+1-1=0) \to (x+1=x+1))|)$ Implicação
```
a = x + 1;
```
$(|((a-1=0) \to (1=x+1)) \wedge$
$(\neg(a-1=0) \to (a=x+1))|)$ Atribuição
```
if (a - 1 == 0) {
```
$(|1=x+1|)$ If' (φ_1)
```
  y = 1;
```
$(|y=x+1|)$ Atribuição
```
} else {
```
$(|a=x+1|)$ If' (φ_2)
```
  y = a;
```
$(|y=x+1|)$ Atribuição
```
}
```
$(|y=x+1|)$ If'

Cálculo para Prova de Correção

While

$$\frac{(|\chi \wedge B|)C(|\chi|)}{(|\chi|)\texttt{while}\ B\ \{C\}\ (|\chi \wedge \neg B|)}$$

Cálculo para Prova de Correção

While

$$\frac{(|\chi \wedge B|)C(|\chi|)}{(|\chi|)\texttt{while}\ B\ \{C\}\ (|\chi \wedge \neg B|)}$$

Normalmente queremos provar

$$(|(\varphi)|)\texttt{while}\ B\ \{C\}\ (|\psi|)$$

Cálculo para Prova de Correção

While

$$\frac{(|\chi \land B|)C(|\chi|)}{(|\chi|)\texttt{while } B\ \{C\}\ (|\chi \land \neg B|)}$$

Normalmente queremos provar

$$(|(\varphi)|)\texttt{while } B\ \{C\}\ (|\psi|)$$

Achar χ tal que:

1. $\vdash \varphi \to \chi$

Cálculo para Prova de Correção

While

$$\frac{(|\chi \wedge B|)C(|\chi|)}{(|\chi|)\texttt{while } B\,\{C\}\,(|\chi \wedge \neg B|)}$$

Normalmente queremos provar

$$(|(\varphi)|)\texttt{while } B\,\{C\}\,(|\psi|)$$

Achar χ tal que:

1. $\vdash \varphi \to \chi$
2. $\vdash \chi \wedge \neg B \to \psi$

Cálculo para Prova de Correção

While

$$\frac{(|\chi \wedge B|)C(|\chi|)}{(|\chi|)\texttt{while } B\,\{C\}\,(|\chi \wedge \neg B|)}$$

Normalmente queremos provar

$$(|(\varphi)|)\texttt{while } B\,\{C\}\,(|\psi|)$$

Achar χ tal que:

1. $\vdash \varphi \to \chi$
2. $\vdash \chi \wedge \neg B \to \psi$
3. $\vdash_{par} (|(\chi)|)\texttt{while } B\,\{C\}\,(|\chi \wedge \neg B|)$