Lógica
Aula 23

Renata Wassermann

renata@ime.usp.br

2020

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

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

$$\Updownarrow$$

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

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

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

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

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'

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

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

Aplicando a regra do while

1. Adivinhar χ
2. Provar $\vdash \chi \land \neg B \to \psi$ e $\vdash \varphi \to \chi$ (se falhar, volta para o primeiro passo)
3. Subir χ por C, obtendo χ'
4. Provar que $\vdash \chi \land B \to \chi'$: χ é invariante (se falhar, volta para o primeiro passo)
5. Colocar χ acima do while e φ acima do χ.

Correção Total

Variante: Expressão inteira E que diminui e nunca é negativa.

Correção Total

Variante: Expressão inteira E que diminui e nunca é negativa.

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

Correção Total

Variante: Expressão inteira E que diminui e nunca é negativa.

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

$E = x$

Correção Total

Variante: Expressão inteira E que diminui e nunca é negativa.

```
y = 1;
z = 0;
while (z != x) {
   z = z + 1;
   y = y * z;
}
```

Correção Total

Variante: Expressão inteira E que diminui e nunca é negativa.

```
y = 1;
z = 0;
while (z != x) {
   z = z + 1;
   y = y * z;
}
```

$E = x - z$

Correção Total

While - total

$$\frac{(|\chi \land B \land 0 \leq E = E_0|)\,C\,(|\chi \land 0 \leq E < E_0|)}{(|\chi \land 0 \leq E|)\,\texttt{while}\ B\ \{C\}\ (|\chi \land \neg B|)}$$

Correção Total

While - total

$$\frac{(|\chi \wedge B \wedge 0 \leq E = E_0|)C(|\chi \wedge 0 \leq E < E_0|)}{(|\chi \wedge 0 \leq E|)\texttt{while}\ \ B\ \{C\}\ (|\chi \wedge \neg B|)}$$

While - parcial

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

While - total

$(|\chi \wedge 0 \leq E|)$
```
while B
```
$\quad (|\chi \wedge B \wedge 0 \leq E = E_0|)$
```
    C
```
$\quad (|\chi \wedge 0 \leq E < E_0|)$

$(|\chi \wedge \neg B|)$

Exemplos

Fatorial:

```
y = 1;
z = 0;
while (z != x) {
   z = z + 1;
   y = y * z;
}
```

$$\vdash_{tot} (|x \geq 0|) \text{ Fatorial } (|y = x!|)$$

Exemplos

Fatorial:

```
y = 1;
z = 0;
while (z != x) {
   z = z + 1;
   y = y * z;
}
```

$$\vdash_{tot} (|x \geq 0|) \text{ Fatorial } (|y = x!|)$$

Invariante:

Exemplos

Fatorial:

```
y = 1;
z = 0;
while (z != x) {
   z = z + 1;
   y = y * z;
}
```

$$\vdash_{tot} (\!|x \geq 0|\!)\ \text{Fatorial}\ (\!|y = x!|\!)$$

Invariante: $y = z!$

Exemplos

Fatorial:

```
y = 1;
z = 0;
while (z != x) {
   z = z + 1;
   y = y * z;
}
```

$$\vdash_{tot} (|x \geq 0|) \text{ Fatorial } (|y = x!|)$$

Invariante: $y = z!$

Variante:

Exemplos

Fatorial:

```
y = 1;
z = 0;
while (z != x) {
   z = z + 1;
   y = y * z;
}
```

$$\vdash_{tot} (|x \geq 0|)\ \text{Fatorial}\ (|y = x!|)$$

Invariante: $y = z!$

Variante: $x - z$

Exemplos

Fatorial 2:

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

$$\vdash_{tot} (|x \geq 0|) \text{ Fatorial 2 } (|y = x!|) \textbf{ X}$$

Exemplos

Fatorial 2:

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

$$\vdash_{tot} (|x \geq 0|) \text{ Fatorial 2 } (|y = x!|) \text{ } \mathbf{X}$$

$$\vdash_{tot} (|x = x_0 \land x \geq 0|) \text{ Fatorial 2 } (|y = x_0!|)$$

Exemplos

Fatorial 2:

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

$$\vdash_{tot} (|x \geq 0|) \text{ Fatorial 2 } (|y = x!|) \ \textbf{X}$$

$$\vdash_{tot} (|x = x_0 \wedge x \geq 0|) \text{ Fatorial 2 } (|y = x_0!|)$$

Invariante:

Exemplos

Fatorial 2:

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

$$\vdash_{tot} (|x \geq 0|) \text{ Fatorial 2 } (|y = x!|) \text{ X}$$

$$\vdash_{tot} (|x = x_0 \wedge x \geq 0|) \text{ Fatorial 2 } (|y = x_0!|)$$

Invariante: $y = x_0!/x!$

Exemplos

Fatorial 2:

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

$$\vdash_{tot} (|x \geq 0|) \text{ Fatorial 2 } (|y = x!|) \text{ } \mathbf{X}$$

$$\vdash_{tot} (|x = x_0 \wedge x \geq 0|) \text{ Fatorial 2 } (|y = x_0!|)$$

Invariante: $y = x_0!/x!$

Variante:

Exemplos

Fatorial 2:

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

$$\vdash_{tot} (|x \geq 0|) \text{ Fatorial 2 } (|y = x!|) \ \textbf{X}$$

$$\vdash_{tot} (|x = x_0 \wedge x \geq 0|) \text{ Fatorial 2 } (|y = x_0!|)$$

Invariante: $y = x_0!/x!$

Variante: x

Exemplos

Potência:

```
y = 1;
z = 0;
while (z != n) {
   z = z + 1;
   y = y * x;
}
```

$$\vdash_{tot} (|n \geq 0|) \text{ Potência } (|y = x^n|)$$

Exemplos

Potência:

```
y = 1;
z = 0;
while (z != n) {
   z = z + 1;
   y = y * x;
}
```

$$\vdash_{tot} (|n \geq 0|) \text{ Potência } (|y = x^n|)$$

Invariante:

Exemplos

Potência:

```
y = 1;
z = 0;
while (z != n) {
   z = z + 1;
   y = y * x;
}
```

$$\vdash_{tot} (|n \geq 0|) \text{ Potência } (|y = x^n|)$$

Invariante: $y = x^z$

Exemplos

Potência:

```
y = 1;
z = 0;
while (z != n) {
   z = z + 1;
   y = y * x;
}
```

$$\vdash_{tot} (|n \geq 0|) \text{ Potência } (|y = x^n|)$$

Invariante: $y = x^z$

Variante:

Exemplos

Potência:

```
y = 1;
z = 0;
while (z != n) {
   z = z + 1;
   y = y * x;
}
```

$$\vdash_{tot} (|n \geq 0|) \text{ Potência } (|y = x^n|)$$

Invariante: $y = x^z$

Variante: $n - z$

Exemplos

Collatz:

```
c = x;
while (c != 1){
   if (c%2 == 0){
      c = c/2;
      }
    else{
       c = 3*c + 1;
       }
  }
```

$$\vdash_{tot} (|0 < x|)\ \text{Collatz}\ (|\top|)$$

Exemplos

Collatz:

```
c = x;
while (c != 1){
   if (c%2 == 0){
      c = c/2;
      }
    else{
       c = 3*c + 1;
       }
  }
```

$$\vdash_{tot} (|0 < x|) \text{ Collatz } (|\top|)$$

Variante:

Exemplos

Collatz:

```
c = x;
while (c != 1){
   if (c%2 == 0){
      c = c/2;
      }
    else{
       c = 3*c + 1;
       }
  }
```

$$\vdash_{tot} (|0 < x|) \text{ Collatz } (|\top|)$$

Variante: ????

Exemplo: Busca Binária

Dado um inteiro x e um vetor crescente v[0..n-1] de inteiros, encontrar um índice j tal que $v[j] = x$.

Exemplo: Busca Binária

Dado um inteiro x e um vetor crescente v[0..n-1] de inteiros, encontrar um índice j tal que $v[j] = x$.

Ou melhor:

... encontrar um índice j tal que $v[j-1] < x \leq v[j]$ (funciona se o elemento x não estiver no vetor).

Exemplo: Busca Binária

```
int  buscaBinaria (int x, int n, int v[]) {
   int e, m, d;
   e = -1; d = n;
   while (e < d-1) {
      m = (e + d)/2;
      if (v[m] < x) e = m;
      else d = m;
   }
   return d;
}
```

Exemplo: Busca Binária

```
int  buscaBinaria (int x, int n, int v[]) {
   int e, m, d;
   e = -1; d = n;
   while (e < d-1) {
      m = (e + d)/2;
      if (v[m] < x) e = m;
      else d = m;
   }
   return d;
}
```

Invariante:

Exemplo: Busca Binária

```
int  buscaBinaria (int x, int n, int v[]) {
   int e, m, d;
   e = -1; d = n;
   while (e < d-1) {
      m = (e + d)/2;
      if (v[m] < x) e = m;
      else d = m;
   }
   return d;
}
```

Invariante: $v[e] < x \leq v[d]$

Exemplo: Busca Binária

```
int  buscaBinaria (int x, int n, int v[]) {
   int e, m, d;
   e = -1; d = n;
   while (e < d-1) {
      m = (e + d)/2;
      if (v[m] < x) e = m;
      else d = m;
   }
   return d;
}
```

Invariante: $v[e] < x \leq v[d]$

Variante:

Exemplo: Busca Binária

```
int  buscaBinaria (int x, int n, int v[]) {
   int e, m, d;
   e = -1; d = n;
   while (e < d-1) {
      m = (e + d)/2;
      if (v[m] < x) e = m;
      else d = m;
   }
   return d;
}
```

Invariante: v[e] $<$ x $\leq$ v[d]

Variante: $d - e$