

## Exercícios de Semântica Axiomática

Usando as regras da Semântica Axiomática para IMP e usando o método de *tableaux* verifique se as seguintes afirmações são verdadeiras.

(1)  $\{\top\} \text{ succ } \{y = x + 1\}$  - Resposta nas notas de aula

(2) Resposta:

```

{⊤}
{(x > y → y = min(x, y)) ∧ (¬(x > y) → x = min(x, y))}
if (x > y) then (
  {y = min(x, y)}
  z := y
  {z = min(x, y)}
) else (
  {x = min(x, y)}
  z := x
  {z = min(x, y)}
)
{z = min(x, y)}

```

- Condições de verificação:  
 $\top \rightarrow ((x > y \rightarrow y = \min(x, y)) \wedge (\neg(x > y) \rightarrow x = \min(x, y)))$  (verdadeiro)
- Conclusão:  
 programa parcialmente correto em relação a especificação

(3) Resposta: (o invariante do while abaixo é  $y = z!$ )

```

{x ≥ 0}
{1 = 0!}
y := 1;
{y = 0!}
z := 0;
{y = z!}
while (z <> x) do (
  {y = z! ∧ z <> x}
  {y * (z + 1) = (z + 1)!}
  z := z + 1;
  {y * z = z!}
  y := y * z
  {y = z!}
)
{y = z! ∧ ¬(z <> x)}
{y = x!}

```

- Condições de verificação:  
 $y = z! \wedge \neg(z <> x) \rightarrow y = x!$  (verdadeira)

- $y = z! \wedge z <> x \rightarrow y * (z + 1) = (z + 1)!$  (verdadeira)
- $x \geq 0 \rightarrow 1 = 0!$  (verdadeira)

- Conclusão:

programa é parcialmente correto em relação a especificação

programa é totalmente correto em relação a especificação (variante:  $x - z$ )

(4) Resposta: (o invariante do while abaixo é  $y = z!$ )

```

{⊤}
{1 = 0!}
y := 1;
{y = 0!}
z := 0;
{y = z!}
while (z <> x) do (
    {y = z! ∧ z <> x}
    {y * (z + 1) = (z + 1)!}
    z := z + 1;
    {y * z = z!}
    y := y * z
    {y = z!}
)
{y = z! ∧ ¬(z <> x)}
{y = x!}

```

- Condições de verificação:

- $y = z! \wedge \neg(z <> x) \rightarrow y = x!$  (verdadeira)
- $y = z! \wedge z <> x \rightarrow y * (z + 1) = (z + 1)!$  (verdadeira)
- $\top \rightarrow 1 = 0!$  (verdadeira)

- Conclusão:

programa é parcialmente correto em relação a especificação

programa não é totalmente correto em relação a especificação (não termina para  $x$  negativo por exemplo)

(5)  $\{x \geq 0\}$  fac2  $\{y = x!\}$  onde fac2 é o seguinte programa

```

y := 1;
while (x <> 0) do
    (y := y * x;
     x := x - 1)

```

Resposta: claramente não é correto em relação a especificação pois ao final do laço o valor de  $x$  é sempre zero, logo a pós-condição  $y = x!$  só é verdadeira quando o  $x$  da entrada for igual a zero.

(6) Resposta: (o invariante do while abaixo é  $y * x! = x_0!$ )

```

{ $x = x_0 \wedge x \geq 0$ }
{ $1 * x! = x_0!$ }
 $y := 1;$ 
{ $y * x! = x_0!$ }
while ( $x <> 0$ ) do (
    { $y * x! = x_0! \wedge x <> 0$ }
    { $(y * x) * (x - 1)! = x_0!$ }
     $y := y * x;$ 
    { $y * (x - 1)! = x_0!$ }
     $x := x - 1$ 
    { $y * x! = x_0!$ }
)
{ $y * x! = x_0! \wedge \neg(x <> 0)$ }
{ $y = x_0!$ }

```

- Condições de verificação:

- $y * x! = x_0! \wedge \neg(x <> 0) \rightarrow y = x_0!$  (verdadeira)
- $y * x! = x_0! \wedge x <> 0 \rightarrow (y * x) * (x - 1)! = x_0!$  (verdadeira)
- $x = x_0 \wedge x \geq 0 \rightarrow 1 * x! = x_0$  (verdadeira)

- Conclusão:

programa é parcialmente correto em relação a especificação

programa é totalmente correto em relação a especificação (variante:  $x$ )

(7) Resposta: (o invariante do while abaixo é  $z = \frac{x_0(x_0+1)}{2} - \frac{x(x+1)}{2} \wedge x \geq 0$ )

```

{ $x = x_0 \wedge x \geq 0$ }
{ $0 = \frac{x_0(x_0+1)}{2} - \frac{x(x+1)}{2} \wedge x \geq 0$ }
 $z := 0;$ 
{ $z = \frac{x_0(x_0+1)}{2} - \frac{x(x+1)}{2} \wedge x \geq 0$ }
while ( $x > 0$ ) do (
    { $z = \frac{x_0(x_0+1)}{2} - \frac{x(x+1)}{2} \wedge x \geq 0 \wedge x > 0$ }
    { $z + x = \frac{x_0(x_0+1)}{2} - \frac{(x-1)((x-1)+1)}{2} \wedge (x-1) \geq 0$ }
     $z := z + x;$ 
    { $z = \frac{x_0(x_0+1)}{2} - \frac{(x-1)((x-1)+1)}{2} \wedge (x-1) \geq 0$ }
     $x := x - 1$ 
    { $z = \frac{x_0(x_0+1)}{2} - \frac{x(x+1)}{2} \wedge x \geq 0$ }
)
{ $z = \frac{x_0(x_0+1)}{2} - \frac{x(x+1)}{2} \wedge x \geq 0 \wedge \neg(x > 0)$ }
{ $z = \frac{x_0(x_0+1)}{2}$ }

```

- Condições de verificação:

- $z = \frac{x_0(x_0+1)}{2} - \frac{x(x+1)}{2} \wedge x \geq 0 \wedge \neg(x > 0) \rightarrow z = \frac{x_0(x_0+1)}{2}$  (verdadeira)
- $z = \frac{x_0(x_0+1)}{2} - \frac{x(x+1)}{2} \wedge x \geq 0 \wedge x > 0 \rightarrow z + x = \frac{x_0(x_0+1)}{2} - \frac{(x-1)((x-1)+1)}{2}$  (verdadeira)

$$- x = x_0 \wedge x \geq 0 \rightarrow 0 = \frac{x_0(x_0+1)}{2} - \frac{x(x+1)}{2} \wedge x \geq 0 \text{ (verdadeira)}$$

- Conclusão:

programa é parcialmente correto em relação a especificação

programa é totalmente correto em relação a especificação (variante:  $x$ )

Resposta: (o invariante do while abaixo é  $y + a = x$ )

```

{ $x \geq 0$ }
{ $0 + x = x$ }
a := x;
{ $0 + a = x$ }
y := 0;
{ $y + a = x$ }
while (a <> 0) do (
    { $y + a = x \wedge a <> 0$ }
    { $(y + 1) + (a - 1) = x$ }
    y := y + 1;
    { $y + (a - 1) = x$ }
    a := a - 1
    { $y + a = x$ }
)
{ $y + a = x \wedge \neg(a <> 0)$ }
{ $y = x$ }

```

(8) • Condições de verificação:

$$- y + a = x \wedge \neg(a <> 0) \rightarrow y = x \text{ (verdadeira)}$$

$$- y + a = x \wedge a <> 0 \rightarrow (y + 1) + (a - 1) = x \text{ (verdadeira)}$$

$$- x \geq 0 \rightarrow 0 + x = x \text{ (verdadeira)}$$

- Conclusão:

programa é parcialmente correto em relação a especificação

programa é totalmente correto em relação a especificação (variante:  $x - y$ )

(9) Resposta: (o invariante do while abaixo é  $z = a.x$ )

```

{ $y \geq 0$ }
{ $0 = 0.x$ }
a := 0;
{ $0 = a.x$ }
z := 0;
{ $z = a.x$ }
while (a <> y) do (
    { $z = a.x \wedge a <> y$ }
    { $z + x = (a + 1).x$ }
    z := z + x;
    { $z = (a + 1).x$ }
    a := a + 1
    { $z = a.x$ }
)

```

)  
 $\{z = a.x \wedge \neg(a <> y)\}$   
 $\{z = x.y\}$

- Condições de verificação:

- $z = a.x \wedge \neg(a <> y) \rightarrow z = x.y$  (verdadeira)
- $z = a.x \wedge a <> y \rightarrow z + x = (a + 1).x$  (verdadeira)
- $y \geq 0 \rightarrow 0 = 0.x$  (verdadeira)

- Conclusão:

programa é parcialmente correto em relação a especificação

programa é totalmente correto em relação a especificação (variante:  $y - a$ )

(10) Resposta: (o invariante do while abaixo é  $z = x.(y_0 - y)$ )

```

 $\{y = y_0 \wedge y \geq 0\}$ 
 $\{0 = x.(y_0 - y)\}$ 
 $z := 0;$ 
 $\{z = x.(y_0 - y)\}$ 
while ( $y <> 0$ ) do (
     $\{z = x.(y_0 - y) \wedge y <> 0\}$ 
     $\{z + x = x.(y_0 - (y - 1))\}$ 
     $z := z + x;$ 
     $\{z = x.(y_0 - (y - 1))\}$ 
     $y := y - 1$ 
     $\{z = x.(y_0 - y)\}$ 
)
 $\{z = x.(y_0 - y) \wedge \neg(y <> 0)\}$ 
 $\{z = x.y_0\}$ 

```

- Condições de verificação:

- $z = x.(y_0 - y) \wedge \neg(y <> 0) \rightarrow z = x.y_0$  (verdadeira)
- $z = x.(y_0 - y) \wedge y <> 0 \rightarrow z + x = x.(y_0 - (y - 1))$  (verdadeira)
- $y = y_0 \wedge y \geq 0 \rightarrow 0 = x.(y_0 - y)$  (verdadeira)

- Conclusão:

programa é parcialmente correto em relação a especificação

programa é totalmente correto em relação a especificação (variante:  $(x.y_0) - z$ )

(11) Resposta: (o invariante do while abaixo é  $y.a! = x! \wedge a \geq 0$ )

```

 $\{x \geq 0\}$ 
 $\{1.x! = x! \wedge x \geq 0\}$ 
 $a := x;$ 
 $\{1.a! = x! \wedge a \geq 0\}$ 
 $y := 1;$ 
 $\{y.a! = x! \wedge a \geq 0\}$ 
while ( $a > 0$ ) do (

```

$$\begin{array}{l}
\{y.a! = x! \wedge a \geq 0 \wedge a > 0\} \\
\{y.a.(a-1)! = x! \wedge (a-1) \geq 0\} \\
y := y * a; \\
\{y.(a-1)! = x! \wedge (a-1) \geq 0\} \\
a := a - 1 \\
\{y.a! = x! \wedge a \geq 0\} \\
) \\
\{y.a! = x! \wedge a \geq 0 \wedge \neg(a > 0)\} \\
\{y = x!\}
\end{array}$$

- Condições de verificação:
  - $y.a! = x! \wedge a \geq 0 \wedge \neg(a > 0) \rightarrow y = x!$  (verdadeira)
  - $y.a! = x! \wedge a \geq 0 \wedge a > 0 \rightarrow y.a.(a-1)! = x! \wedge (a-1) \geq 0$  (verdadeira)
  - $x \geq 0 \rightarrow 1.x! = x! \wedge x \geq 0$  (verdadeira)
- Conclusão:
  - programa é parcialmente correto em relação a especificação
  - programa é totalmente correto em relação a especificação (variante:  $a$ )

(12) Resposta: (o invariante de ambos os while é  $m^n = z.x^y$ )

$$\begin{array}{l}
\{x = m \wedge y = n \wedge z = 1\} \\
\{m^n = z.x^y\} \\
\textbf{while } (y <> 0) \textbf{ do } ( \\
\quad \{m^n = z.x^y \wedge y <> 0\} \\
\quad \{m^n = z.x^y\} \\
\quad \textbf{while } (\textbf{par}(y)) \textbf{ do } ( \\
\quad \quad \{m^n = z.x^y \wedge \textbf{par}(y)\} \\
\quad \quad \{m^n = z.(x.x)^{y/2}\} \\
\quad \quad x := x * x; \\
\quad \quad \{m^n = z.x^{y/2}\} \\
\quad \quad y := y / 2 \\
\quad \quad \{m^n = z.x^y\} \\
\quad ); \\
\quad \{m^n = z.x^y \wedge \neg(\textbf{par}(y))\} \\
\quad \{m^n = z.x.x^{(y-1)}\} \\
\quad z := z * x; \\
\quad \{m^n = z.x^{(y-1)}\} \\
\quad y := y - 1 \\
\quad \{m^n = z.x^y\} \\
) \\
\{m^n = z.x^y \wedge \neg(y <> 0)\} \\
\{z = m^n\}
\end{array}$$

- Condições de verificação:
  - $m^n = z.x^y \wedge \neg(y <> 0) \rightarrow z = m^n$  (verdadeira)
  - $m^n = z.x^y \wedge \neg(\textbf{par}(y)) \rightarrow m^n = z.x.x^{(y-1)}$  (verdadeira)

- $m^n = z.x^y \wedge \text{par}(y) \rightarrow m^n = z.(x.x)^{y/2}$  (verdadeira)
- $m^n = z.x^y \wedge y <> 0 \rightarrow m^n = z.x^y$  (verdadeira)
- $x = m \wedge y = n \wedge z = 1 \rightarrow m^n = z.x^y$  (verdadeira)

- Conclusão:

programa é parcialmente correto em relação a especificação

programa não é totalmente correto em relação a especificação (para  $y$  negativo, por exemplo, o programa entra em *loop*)