TPC-2

Gabriel Costa, MEFIS
Universidade do Minho, Portugal
Email: pg53828@uminho.pt

Exercício 2

Neste exercício era pedido para provarmos e pensarmos em novas equivalências entre programas. No entanto, para facilitar a interpretação e prova destas, começarei por escrever as regras de semântica adotadas:

$$\frac{\langle t, \sigma \rangle \Downarrow r}{\langle x := t, \sigma \rangle \Downarrow 0, \sigma\,[r/x]}(asg) \qquad \frac{\langle p, \sigma \rangle \Downarrow n, \sigma'}{\langle \mathrm{wait_m}(p), \sigma \rangle \Downarrow n + m, \sigma'}(wait)$$

$$\frac{\langle p, \sigma \rangle \Downarrow n, \sigma' \qquad \langle q, \sigma' \rangle \Downarrow m, \sigma''}{\langle p; q, \sigma \rangle \Downarrow n + m, \sigma''}(seq)$$

$$\frac{\langle b, \sigma \rangle \Downarrow \mathrm{tt} \qquad \langle p, \sigma \rangle \Downarrow n, \sigma'}{\langle \text{if } b \text{ then } p \text{ else } q, \sigma \rangle \Downarrow n, \sigma'}(\mathrm{if}_1) \qquad \frac{\langle b, \sigma \rangle \Downarrow \mathrm{ff} \qquad \langle q, \sigma \rangle \Downarrow n, \sigma'}{\langle \text{if } b \text{ then } p \text{ else } q, \sigma \rangle \Downarrow n, \sigma'}(\mathrm{if}_2)$$

$$\frac{\langle b, \sigma \rangle \Downarrow \mathrm{tt} \qquad \langle p, \sigma \rangle \Downarrow n, \sigma' \qquad \langle \text{while } b \text{ do } \{p\}, \sigma' \rangle \Downarrow m, \sigma''}{\langle \text{while } b \text{ do } \{p\}, \sigma \rangle \Downarrow n + m, \sigma''}(\mathrm{wh}_1)$$

$$\frac{\langle b, \sigma \rangle \Downarrow \mathrm{ff}}{\langle \text{while } b \text{ do } \{p\}, \sigma \rangle \Downarrow 0, \sigma}(\mathrm{wh}_2)$$

Após introduzir a semântica convém ainda referir o que torna dois programa equivalentes. Dois programas p e q são equivalentes ($p \rightsquigarrow q$) se para todos os estados σ :

$$\langle p, \sigma \rangle \Downarrow n, \sigma' \quad \text{iff} \quad \langle q, \sigma \rangle \Downarrow n, \sigma' \tag{1}$$

Com estas noções podemos agora começar a provas equivalências entre programas.

1 $\mathbf{wait}_m\left(\mathbf{wait}_n\left(p\right)\right) \rightsquigarrow \mathbf{wait}_{n+m}\left(p\right)$

Demonstração. Comecemos por admitir que o programa p tem um tempo de execução de k. Desta forma, para o lado esquerdo da igualdade temos:

$$\cfrac{\cfrac{\langle p, \sigma \rangle \Downarrow k, \sigma'}{\langle \mathrm{wait_m}(p), \sigma \rangle \Downarrow k + m, \sigma'} \text{ (wait)}}{\langle \mathrm{wait}_n\left(\mathrm{wait}_m\left(p\right)\right), \sigma \rangle \Downarrow k + m + n, \sigma} \text{ (wait)}$$

Da mesma forma, para o lado direito da equivalência temos:

$$\frac{\langle p, \sigma \rangle \Downarrow k, \sigma'}{\langle \mathrm{wait}_{m+n}\left(p\right), \sigma \rangle \Downarrow k + m + n, \sigma} \text{ (wait)}$$

Podemos então observar, que ambos os programas resultam no mesmo tempo de execução e na mesma output, pelo que provamos que:

$$\mathrm{wait}_m\left(\mathrm{wait}_n\left(p\right)\right) \rightsquigarrow \mathrm{wait}_{n+m}\left(p\right)$$

□

2 $x := t; x := t \rightsquigarrow x := t$

A equivalência acima descrita significa que, associar um valor duas vezes à mesma variável de forma sucessiva, tem o mesmo impacto que associar apenas uma vez (desde que o termo t não dependa da variável x).

Demonstração. Podemos provar tal equivalência fazendo:

$$\begin{aligned}
&\langle x := t; x := t, \sigma\rangle \Downarrow 0, \sigma' \\
\Leftrightarrow&\langle x := t, \sigma\rangle \Downarrow 0, \sigma'' \quad \wedge \quad \langle x := t, \sigma''\rangle \Downarrow 0, \sigma' \\
\Leftrightarrow&\sigma'' = x \mapsto t \quad \wedge \quad \langle x := t, x \mapsto t\rangle \Downarrow 0, \sigma' \\
\Leftrightarrow&\sigma' = x \mapsto t \quad \wedge \quad \sigma'' = x \mapsto t \quad \wedge \quad \langle x := t, x \mapsto t\rangle \Downarrow 0, \sigma' \\
\Leftrightarrow&\langle x := t; x := t, \sigma\rangle \Downarrow 0, x \mapsto t \quad \wedge \quad \langle x := t, \sigma\rangle \Downarrow 0, x \mapsto t \quad \wedge \quad \langle x := t; x := t, \sigma\rangle \Downarrow 0, \sigma' \\
\Leftrightarrow&\langle x := t, \sigma\rangle \Downarrow 0, \sigma'
\end{aligned}$$

Pela demonstração acima escrita, e tendo em conta a definição de equivalência(1), podemos então afirmar que:

$$x := t \ ; \ x := t \rightsquigarrow x := t$$

□

3 $x := t; s := t \rightsquigarrow x := t; s := x$

Esta equivalência permite que o valor do termo t seja calculado apenas uma vez, tornando a computação mais eficiente. Tal como na equivalência de cima, esta equivalência apenas é possível para casos em que o termo t não depende das variáveis x e s.

Demonstração. Esta equivalência pode então ser provada da seguinte maneira:

$$\begin{aligned}
&\langle x := t; s := t, \sigma\rangle \Downarrow 0, \sigma' \\
\Leftrightarrow&\langle x := t, \sigma\rangle \Downarrow 0, \sigma'' \quad \wedge \quad \langle s := t, \sigma''\rangle \Downarrow 0, \sigma' \\
\Leftrightarrow&\sigma'' = x \mapsto t \quad \wedge \quad \langle s := t, x \mapsto t\rangle \Downarrow 0, \sigma' \\
\Leftrightarrow&\sigma' = s \mapsto t, x \mapsto t \quad \wedge \quad \sigma'' = x \mapsto t \quad \wedge \quad \langle s := t, x \mapsto t\rangle \Downarrow 0, \sigma' \\
\Leftrightarrow&\sigma' = s \mapsto \sigma''(x), x \mapsto t \quad \wedge \quad \sigma'' = x \mapsto t \quad \wedge \quad \langle s := t, x \mapsto t\rangle \Downarrow 0, \sigma' \\
\Leftrightarrow&\langle x := t; s := t, \sigma\rangle \Downarrow 0, s \mapsto t, x \mapsto t \quad \wedge \quad \langle x := t, s := x, \sigma\rangle \Downarrow 0, s \mapsto x, x \mapsto t \quad \wedge \quad \langle x := t; s := t, \sigma\rangle \Downarrow 0, \sigma' \\
\Leftrightarrow&\langle x := t; s := x, \sigma\rangle \Downarrow 0, \sigma'
\end{aligned}$$

Com a demonstração acima e tendo em conta a definição de equivalência, provamos então que:

$$x := t; s := t \rightsquigarrow x := t; s := x$$

□

4 $(p; q) ; s \rightsquigarrow p; (q; s)$

Demonstração. Para provar a equivalência acima escrita comecemos por assumir que:

$$\begin{aligned}
\langle p, \sigma\rangle \Downarrow a, \sigma' \\
\langle q, \sigma\rangle \Downarrow b, \sigma' \\
\langle s, \sigma\rangle \Downarrow c, \sigma'
\end{aligned}$$

Podemos então averiguar a veracidade da equivalência acima escrita:

$$\begin{aligned}
&\langle (p; q) ; s, \sigma\rangle \Downarrow a + b + c, \sigma' \\
\Leftrightarrow&\langle p; q, \sigma\rangle \Downarrow a + b, \sigma'' \qquad \langle s, \sigma''\rangle \Downarrow c, \sigma' \\
\Leftrightarrow&\langle p, \sigma\rangle \Downarrow a, \sigma''' \qquad \langle q, \sigma'''\rangle \Downarrow b, \sigma'' \qquad \langle s, \sigma''\rangle \Downarrow c, \sigma' \\
\Leftrightarrow&\langle p, \sigma\rangle \Downarrow a, \sigma''' \qquad \langle p; q, \sigma'''\rangle \Downarrow b + c, \sigma' \\
\Leftrightarrow&\langle p; (q; s) , \sigma\rangle \Downarrow a + b + c, \sigma'
\end{aligned}$$

Como é possível observar, o programa à esquerda da equivalência pode ser reescrito de maneira a ficar igual ao programa à direita, pelo que, tendo em conta a definição de equivalência(1), provamos que:

$$(p; q) \,; s \rightsquigarrow p; (q; s)$$

$\square$

5 $p; \textbf{wait}_n\,(q) \rightsquigarrow \textbf{wait}_n\,(p)\,; q$

Demonstração. Esta equivalência diz-nos que executar um programa e esperar n segundos para executar o programa seguinte tem o mesmo efeito que esperar n segundos antes de executar o primeiro programa, e executar o segundo programa logo de seguida.

$$\dfrac{\langle p, \sigma\rangle \Downarrow k, \sigma'' \qquad \dfrac{\langle q, \sigma''\rangle \Downarrow m, \sigma'}{\langle \text{wait}_n(q), \sigma''\rangle \Downarrow m+n, \sigma'}\,(wait)}{\langle p; \text{wait}_n(q), \sigma\rangle \Downarrow k+m+n, \sigma'}\,(seq)$$

Fazendo o mesmo para o programa à direita da equivalência obtemos:

$$\dfrac{\dfrac{\langle p, \sigma\rangle \Downarrow k, \sigma''}{\langle \text{wait}_n(p), \sigma\rangle \Downarrow k+n, \sigma''}\,(wait) \qquad \langle q, \sigma''\rangle \Downarrow m, \sigma'}{\langle \text{wait}_n(p); q, \sigma\rangle \Downarrow k+n+m, \sigma'}\,(seq)$$

Como se pode observar, ambos os programas são semanticamente equivalentes, pelo que, pela definição de equivalência(1), provamos que:

$$p; \text{wait}_n\,(q) \rightsquigarrow \text{wait}_n\,(p)\,; q$$

$\square$

6 if b then p else $p \rightsquigarrow p$

Demonstração. Esta equivalência diz-nos que ter um *if statement* redundante (os dois possíveis caminhos levam ambos à execução do programa p) é o mesmo que simplesmente executar o programa p.

$$\begin{aligned} &\langle \text{if } b \text{ then } p \text{ else } p, \sigma\rangle \Downarrow n, \sigma' \\ \Leftrightarrow &\left(\langle b, \sigma\rangle \Downarrow \text{tt} \quad \vee \quad \langle b, \sigma\rangle \Downarrow \text{ff}\right) \quad \wedge \quad \langle p, \sigma\rangle \Downarrow n, \sigma' \\ \Leftrightarrow &\langle p, \sigma\rangle \Downarrow n, \sigma' \end{aligned}$$

Como podemos ver, o programa $\langle \text{if } b \text{ then } p \text{ else } p, \sigma\rangle \Downarrow n, \sigma'$ pode ser convertido em $\langle p, \sigma\rangle \Downarrow n, \sigma'$, e vice-versa, pelo que os programas são equivalentes.

$\square$

7 if b then wait$_n(p)$ else wait$_n(q) \rightsquigarrow$ wait$_n$ (if b then p else q)

Esta equivalência permite ao compilador remover uma instrução na árvore. Se os dois caminhos possíveis do *if* forem programas precedidos por um *wait*, então a instrução de *wait* pode simplesmente ser colocada fora da *if clause.*

Demonstração. O programa da esquerda tem como avaliações possíveis os dois seguintes casos:

$$\begin{cases} \dfrac{\langle \text{if } b \text{ then wait}_n(p) \text{ else wait}_n(q), \sigma\rangle \Downarrow n+m, \sigma'}{\langle \text{wait}_n\,(p)\,, \sigma\rangle \Downarrow n+m, \sigma'} & \langle b, \sigma\rangle \Downarrow \text{tt} \\ \dfrac{\langle \text{if } b \text{ then wait}_n(p) \text{ else wait}_n(q), \sigma\rangle \Downarrow n+m, \sigma'}{\langle \text{wait}_n\,(q)\,, \sigma\rangle \Downarrow n+m, \sigma'} & \langle b, \sigma\rangle \Downarrow \text{ff} \end{cases}$$

Já o programa à direita da equivalência tem como possíveis casos:

$$\begin{cases} \dfrac{\langle \text{wait}_n\,(\text{if } b \text{ then wait}_n(p) \text{ else wait}_n(q))\,, \sigma\rangle \Downarrow n+m, \sigma'}{\langle \text{wait}_n\,(p)\,, \sigma\rangle \Downarrow n+m, \sigma'} & \langle b, \sigma\rangle \Downarrow \text{tt} \\ \dfrac{\langle \text{wait}_n\,(\text{if } b \text{ then wait}_n(p) \text{ else wait}_n(q))\,, \sigma\rangle \Downarrow n+m, \sigma'}{\langle \text{wait}_n\,(q)\,, \sigma\rangle \Downarrow n+m, \sigma'} & \langle b, \sigma\rangle \Downarrow \text{ff} \end{cases}$$

Como podemos observar, a avaliação de ambos os programas têm o mesmo estado final e o mesmo tempo de execução. Desta forma, voltando a recorrer à definição de equivalência(1), provamos que:

$$\text{if } b \text{ then } \text{wait}_n(p) \text{ else } \text{wait}_n(q) \rightsquigarrow \text{wait}_n\,(\text{if } b \text{ then } p \text{ else } q)$$

$\square$