Cyber-Physical Programming

TPC-2

Melânia Pereira

pg47520@alunos.uminho.pt

May 29, 2022

Exercise 1. Prove that `wait`$_{n}$(`wait`$_{m}$(p)) $\sim$ `wait`$_{n+m}$(p).

To prove this equivalence, using the semantic rules provided, we need to do one of two things:

1. we can begin with one program and try to calculate our way to the other one, or
2. we can calculate the output and execution time for both of the programs using the semantic rules and check if they are equal

Let's do both.

Following the first way, let's start with `wait`$_{n}$(`wait`$_{m}$(p)) and try to end up with `wait`$_{n+m}$(p):

$$
\begin{aligned}
& \langle wait_n(wait_m(p)), \sigma\rangle \Downarrow n + x, \sigma' \\
\equiv\ & \{\text{wait rule}\} \\
& \langle wait_m(p), \sigma\rangle \Downarrow m + y, \sigma' \wedge x = m + y \\
\equiv\ & \{\text{wait rule, substitution}\} \\
& \langle p, \sigma\rangle \Downarrow y, \sigma' \wedge n + x = n + m + y \\
\equiv\ & \{\text{wait rule}\} \\
& \langle wait_{n+m}(p), \sigma\rangle \Downarrow (n + m) + y, \sigma' \wedge n + x = n + m + y \\
\equiv\ & \{\text{substitution}\} \\
& \langle wait_{n+m}(p), \sigma\rangle \Downarrow n + x, \sigma'
\end{aligned}
$$

So, we have now proved that the two programs have equivalent output and execution time and, therefore, are equivalent.

Now following the second way, let's simplify each program using the language's semantic rules and finding out the output and execution time for each one and, then, compare and check if they are equal.

$$
\cfrac{\cfrac{\langle p, \sigma\rangle \Downarrow y^1, \sigma'}{\langle wait_m(p), \sigma\rangle \Downarrow m + y, \sigma'}\ (\text{wait})}{\langle wait_n(wait_m(p)), \sigma\rangle \Downarrow n + m + y, \sigma'}\ (\text{wait})
$$

$$
\cfrac{\langle p, \sigma\rangle \Downarrow y^1, \sigma'}{\langle wait_{n+m}(p), \sigma\rangle \Downarrow n + m + y, \sigma'}\ (\text{wait})
$$

As we can see, the output and execution time of the programs are the same: $n + m + y, \sigma'$. So we can infer that the programs are equivalent.

[1] Assuming y as the execution time of p