$$[assert_{pe}]\ \frac{\langle S, s\rangle \to s'}{\langle assert\ b\ before\ S, s\rangle \to s'} \quad si\ \mathcal{B}[\![b]\!]s = tt$$

$$[assert_{pe}]\ \frac{si\ \mathcal{B}[\![b]\!]s = tt}{\langle assert\ b\ before\ S, s\rangle \Rightarrow \langle S, s\rangle}$$

---

assert true before S es equiv a S

Paso largo

$\Rightarrow$ Sea $s \in State$. y supongamos que $\langle assert\ true\ before\ S, s\rangle \to s'$

$\Rightarrow$ Solo se ha podido aplicar la regla $[assert_{pe}]$ y $\langle S, s\rangle \to s'$ OK

$\Leftarrow$ Sea $s \in State$ y supongamos que $\langle S, s\rangle \to s'$. Como

$\mathcal{B}[\![true]\!]s = tt \Rightarrow$ regla $[assert_{pe}]$ $\langle assert\ true\ before\ S, s\rangle \to s'$

Paso corto

$\Rightarrow$ Sea $s \in State$ y supongamos que $\langle assert\ true\ before\ S, s\rangle \Rightarrow^* \gamma$ con γ terminal.

Si γ es un estado final $\Rightarrow \langle assert\ true\ before\ S, s\rangle \Rightarrow^k s'$ con $k \geq 1$ (no puede ser $k=0$)

La primera regla que se ha tenido que aplicar ha sido

$[assert_{pc}] \Rightarrow \mathcal{B}[\![true]\!]s = tt$ y $\langle assert\ b\ before\ S, s\rangle \Rightarrow \langle S, s\rangle \Rightarrow^{k-1} s'$

$\Rightarrow \langle S, s\rangle \Rightarrow^* s'$ OK

Si γ es una configuración bloqueada vemos que $\langle S, s\rangle \Rightarrow^* \gamma$ también se bloquea.

$\Rightarrow$ $\langle$assert true before $S, s\rangle \Rightarrow^k \langle S', s'\rangle$ con $\langle S', s'\rangle$ bloqueada

Como se puede aplicar la respuesta [assert], entonces $k \neq 0$.

Se ha tenido que aplicar la regla [assert] y.

$\langle$assert true before $S, s\rangle \Rightarrow \langle S, s\rangle \Rightarrow^{k-1} \langle S', s'\rangle$

$\Rightarrow \langle S, s\rangle \Rightarrow^* \gamma$ con γ configuración bloqueada.

$\Leftarrow$ Supongamos que $\langle S, s\rangle \Rightarrow^* \gamma$ con γ config terminal.

Como $\mathcal{A}[\![\text{true}]\!]s = tt$ entonces se puede aplicar la regla [assert]

y $\langle$assert true before $S, s\rangle \Rightarrow \langle S, s\rangle \Rightarrow^* \gamma$

$\Rightarrow \langle$assert true before $S, s\rangle \Rightarrow^* \gamma$. ✓

assert false before S no es equivalente a skip

Paso largo

Sea s_0 un estado cualquiera.

No existe árbol de derivación para assert false before S ya que la única regla que sintácticamente encaja es [assert] pero exige que

$\mathcal{A}[\![\text{false}]\!]s_0 = tt$ y esto es falso. $\Rightarrow \langle$assert false before $S, s_0\rangle$ cicla.

Sin embargo $\langle$skip $s_0\rangle \longrightarrow s_0$ por la regla skip.

Paso corto

Sea s_0 un estado cualquiera

$\langle$assert false before $S, s_0\rangle$ es una configuración bloqueada, es decir

$\langle$assert false before $S, s_0\rangle \Rightarrow^{(0)*} \langle$assert false before $S, s_0\rangle$.

Sin embargo, por la regla [skip]

$\langle$skip, $s_0\rangle \Rightarrow s_0$ es decir $\langle$skip, $s_0\rangle \Rightarrow^{(1)*} s_0$ y las configuraciones terminales no son iguales.

assert false before S no es equivalente a abort en semántica de paso corto pero sí lo es en paso largo.

Para paso largo es inmediato porque para ninguna de los dos hay árbol de derivación.

Sin embargo. $\langle \text{assert false before } S, s\rangle \Rightarrow^{(0)*} \langle \text{assert false before } S, s\rangle$ ← Configuración terminada pero no igual

$\langle \text{abort}, s\rangle \Rightarrow^{(0)*} \langle \text{abort}, s\rangle$