Ejercicio 2.3.- Considera la sentencia

$$S_1 = z := 0;\ \underbrace{\text{while } y \le x \text{ do } \overbrace{(z := z+1;\ x := x-y)}^{S_3}}_{S_2}$$

Construye un árbol de derivación de la configuración $\langle S_1, s_0 \rangle$ con s_0 tal que $s_0\,x = 17$ y $s_0\,y = 5$

Si consideramos los estados s_i con $i \in \{1, 2, \ldots 8\}$ verificando que

	x	y	z
s_0	17	5	-
s_1	2	5	3
s_2	17	5	0
s_3	12	5	1
s_4	17	5	1
s_5	7	5	2
s_6	12	5	2
s_7	7	5	3

1. Como $s_2 = s_0[z \mapsto 0]$, por la regla [acc] se tiene $\langle z := 0, s_0 \rangle \to s_2$
2. Como $s_4 = s_2[z \mapsto 1]$, por la regla [acc] se tiene $\langle z := z+1, s_2 \rangle \to s_4$
3. Como $s_3 = s_4[x \mapsto 12]$, por la regla [acc] se tiene $\langle x := x-y, s_4 \rangle \to s_3$
4. Como $s_6 = s_3[z \mapsto 2]$, por la regla [acc] se tiene $\langle z := z+1, s_3 \rangle \to s_6$
5. Como $s_5 = s_6[x \mapsto 7]$, por la regla [acc] se tiene $\langle x := x-y, s_6 \rangle \to s_5$
6. Como $s_7 = s_5[z \mapsto 3]$, por la regla [acc] se tiene $\langle z := z+1, s_5 \rangle \to s_7$
7. Como $s_1 = s_7[x \mapsto 2]$, por la regla [acc] se tiene $\langle x := x-y, s_7 \rangle \to s_1$

Comprobando que (8) $\mathcal{B}[\![y \le x]\!]s_1 = ff$ ya que $s_1 y = 5 > 2 = s_1 x$,
(9) $\mathcal{B}[\![y \le x]\!]s_5 = tt$ ($s_5 y = 5 \le 7 = s_5 x$), (10) $\mathcal{B}[\![y \le x]\!]s_3 = tt$ ($s_3 y = 5 \le 12 = s_3 x$) y (11) $\mathcal{B}[\![y \le x]\!]s_2 = tt$
($s_2 y = 5 \le 17 = s_2 x$)

Podemos construir el árbol.

[comp] (6) $\langle z := z+1, s_5 \rangle \to s_7$, (7) $\langle x := x-y, s_7 \rangle \to s_1$ — (8) $\mathcal{B}[\![y \le x]\!]s_1 = ff$ [while^{ff}]

[comp] (4) $\langle z := z+1, s_3 \rangle \to s_6$, (5) $\langle x := x-y, s_6 \rangle \to s_5$ — $\langle S_3, s_5 \rangle \to s_1$, $\langle S_2, s_1 \rangle \to s_1$, (9) $\mathcal{B}[\![x \le y]\!]s_5 = tt$ [while^{tt}]

[comp] (2) $\langle z := z+1, s_2 \rangle \to s_4$, (3) $\langle x := x-y, s_4 \rangle \to s_3$ — $\langle S_3, s_3 \rangle \to s_5$, $\langle S_2, s_5 \rangle \to s_1$, (10) $\mathcal{B}[\![y \le x]\!]s_3 = tt$ [while^{tt}]

$\langle S_3, s_2 \rangle \to s_3$, $\langle S_2, s_3 \rangle \to s_1$ [while^{tt}] si (11) $\mathcal{B}[\![y \le x]\!]s_2 = tt$

(1) $\langle z := 0, s_0 \rangle \to s_2$, $\langle S_2, s_2 \rangle \to s_1$ [comp]

$\langle S_1, s_0 \rangle \to s_1$

Ejercicio 24.- Considerar las siguientes instrucciones:

a) while $\neg(x=1)$ do $(y:=y*x\ ;\ x:=x-1) \equiv S_a$

b) while $1 \leq x$ do $(y:=y*x\ ;\ x:=x-1) \equiv S_b$

c) while true do skip; $\equiv S_c$

Para cada una de ellas, argumenta si siempre termina o si siempre cicla.

a) Veamos que la instrucción while $\neg(x=1)$ do $(y:=y*x\ ;\ x:=x-1)$ no siempre termina y no siempre cicla, es decir, que existen $s_0, s_1 \in$ States tales que la ejecución de S_a sobre el estado s_0 no termina (lo que prueba que no siempre termina) y la ejecución de S_a sobre el estado s_1 no cicla (lo que prueba que no siempre cicla).

Para lo primero, basta considerar s_0 tal que $s_0\, x \leq 0$. Se puede demostrar que $\forall s$ con esta propiedad se verifica que si s' es tal que $\langle y:=y*x\ ;\ x:=x-1, s\rangle \to s'$ entonces $s'\, x \leq 0$. Si suponemos, por reducción al absurdo, que la configuración $\langle S_a, s_0\rangle$ termina en un estado s_1 entonces existe un árbol de derivación finito. Como todo se genera a partir de las reglas de derivación entonces se tiene que haber aplicado las reglas $[\text{while}^{ff}]$ o $[\text{while}^{tt}]$.

Si se ha generado por $[\text{while}^{ff}]$ se verifica $\mathcal{B}[\![\neg(x=1)]\!]\, s_0 = ff$ pero $s_0\, x \leq 0$ y esto nos lleva a contradicción.

Si se ha generado por $[\text{while}^{tt}]$ se verifica $\mathcal{B}[\![\neg(x=1)]\!]\, s_0 = tt$, $\langle y:=y*x\ ;\ x:=x-1, s_0\rangle \to s_2$ y $\langle S_a, s_2\rangle \to s_1$. Como s_2 también verifica que $s_2\, x \leq 0$ y el árbol de derivación es en tamaño una unidad menor, se puede argumentar por inducción que el estado en todo nivel del árbol verifica que $s\, x \leq 0$ y esto no se puede cumplir porque en las hojas se han tenido que aplicar $[\text{while}^{ff}]$ y se llega a contradicción.

Para lo segundo demostramos algo más general:

Sea $s \in \mathbf{State}$ con $s\,x \geq 1$, entonces la configuración

$$\langle \text{while } \neg(x=1) \text{ do } (y:=y+x;\ x:=x-1), s \rangle \text{ termina.}$$

Hay que demostrar que existe un estado s' tal que

$$\langle \text{while } \neg(x=1) \text{ do } (y:=y+x;\ x:=x-1), s \rangle \rightarrow s'$$

Sea $n = s\,x$ que por hipótesis es mayor o igual que 1.

Vamos a probar por inducción sobre n que

$$\langle \text{while } \neg(x=1) \text{ do } y:=y+x;\ x:=x-1), s_n \rangle \text{ termina}$$

donde $s_n\,x = n$.

Caso base. Si $n=1$, entonces $s_1\,x = 1$.

Se verifica que $\mathcal{B}[\![\neg(x=1)]\!]s_1 = \begin{cases} tt & \text{si } \mathcal{B}[\![x=1]\!]s_1 = ff \\ ff & \text{si } \mathcal{B}[\![x=1]\!]s_1 = tt \end{cases} =$

$= \begin{cases} tt & \text{si } \mathcal{A}[\![x]\!]s_1 \neq \mathcal{A}[\![1]\!]s_1 \\ ff & \text{si } \mathcal{A}[\![x]\!]s_1 = \mathcal{A}[\![1]\!]s_1 \end{cases} = \begin{cases} tt & \text{si } s_1\,x \neq \mathcal{N}[\![1]\!] \\ ff & \text{si } s_1\,x = \mathcal{N}[\![1]\!] \end{cases} =$

$= \begin{cases} tt & \text{si } 1 \neq 1 \\ ff & \text{si } 1 = 1 \end{cases} = ff$, es decir, $\mathcal{B}[\![\neg(x=1)]\!]s_1 = ff$.

Por la regla $[\text{while}_{ns}^{ff}]$ se tiene que la configuración

$\langle \text{while } \neg(x=1) \text{ do } y:=y+x;\ x:=x-1, s_1 \rangle \rightarrow s_1$, es decir,

existe un estado terminal (que es s_1) que es resultado de la ejecución de dicha configuración

Por tanto se verifica el caso base.

Paso inductivo. Sea $n \geq 1$ y supongamos probado que

$\langle \text{while } \neg(x=1) \text{ do } y:=\cancel{y}y+x;\ x:=x-1, s_n\rangle$ termina y queremos probar que

$\langle \text{while } \neg(x=1) \text{ do } y:=y+x;\ x:=x-1, s_{n+1}\rangle$ termina.

En primer lugar vamos a encontrar un estado $\bar{s}$ que verifique que $s' x = n$ (es decir tomaremos $s_n = \bar{s}$ al aplicar ~~la~~ hipótesis de inducción) tal que $\langle y:=y+x;\ x:=x-1, s_{n+1}\rangle \to \bar{s}$

- En primer lugar, por la regla $[ass_{bs}]$ $\langle y:=y+x, s_{n+1}\rangle \to s_{n+1}[y \mapsto \mathcal{A}[\![y+x]\!]s_{n+1}]$ y podemos llamar $\hat{s} = s_{n+1}[y \mapsto \mathcal{A}[\![y+x]\!]s_{n+1}]$, que sigue verificando que $\hat{s}\,x = n+1$ porque $s_{n+1}[y \mapsto \mathcal{A}[\![y+x]\!]s_{n+1}]\,x = \begin{cases} \mathcal{A}[\![y+x]\!]s_{n+1} & \text{si } x=y \\ s_{n+1}\,x & \text{si } x \neq y \end{cases} = s_{n+1}\,x = n+1$
- Aplicando la regla $[ass_{bs}]$ $\langle x:=x-1, \hat{s}\rangle \to \hat{s}[x \mapsto \mathcal{A}[\![x-1]\!]\hat{s}]$.

  Si llamamos $\bar{s} = \hat{s}[x \mapsto \mathcal{A}[\![x-1]\!]\hat{s}]$ entonces veamos que $\bar{s}\,x = n$. En efecto $\bar{s}x = \hat{s}[x \mapsto \mathcal{A}[\![x-1]\!]\hat{s}]\,x =$

  $$= \begin{cases} \mathcal{A}[\![x-1]\!]\hat{s} & \text{si } x=x \\ \hat{s}\,x & \text{si } x \neq x \end{cases} = \mathcal{A}[\![x-1]\!]\hat{s} = \mathcal{A}[\![x]\!]\hat{s} - \mathcal{A}[\![1]\!]\hat{s} =$$

  $$= \hat{s}\,x - \mathcal{N}[\![1]\!] \overset{\text{Ya probado}}{=} n+1-1 = n.$$
- Como se cumplen $\langle y:=y+x, s_{n+1}\rangle \to \hat{s}$ y $\langle x:=x-1, \hat{s}\rangle \to \bar{s}$, aplicando la regla $[comp_{bs}]$ se tiene que $\langle y:=y+x; x:=x-1, s_{n+1}\rangle \to \bar{s}$

Como se verifica que $\bar{s}x = n$, por hipotesis de inducción, la configuración $\langle \text{while } \neg(x=1) \text{ do } y:=y+x;\ x:=x+1, \bar{s}\rangle$ termina, luego $\exists s' \in State$, $\langle \text{while } \neg(x=1) \text{ do } y:=y+x, x:=x+1, \bar{s}\rangle \to s'$

Por último, como se verifica que

$$\langle y:=y+x;\ x:=x-1,\ s_{n+1}\rangle \to \bar{s} \quad y \quad \langle \text{while } \neg(x=1) \text{ do } (y:=y+x;\ x:=x-1),\ \bar{s}\rangle \to s',$$

aplicando la regla $[\text{while}_{bs}^{tt}]$ se tiene que

$$\langle \text{while } \neg(x=1) \text{ do } (y:=y+x;\ x:=x-1),\ s_{n+1}\rangle \to s', \text{ es decir,}$$

la configuración $\langle \text{while } \neg(x=1) \text{ do } y:=y+x;\ x:=x-1,\ s_{n+1}\rangle$. termina.

Antes de aplicar esta regla hay que verificar que $\mathcal{B}[\![\neg(x=1)]\!]s_{n+1}=tt$, pero esto es así ya que

$$\mathcal{B}[\![\neg(x=1)]\!]s_{n+1} = \begin{cases} tt & \text{si } \mathcal{B}[\![x=1]\!]s_{n+1}=ff \\ ff & \text{si } \mathcal{B}[\![x=1]\!]s_{n+1}=tt \end{cases} = \begin{cases} tt & \text{si } \mathcal{A}[\![x]\!]s_{n+1}\neq \mathcal{A}[\![1]\!]s_{n+1} \\ ff & \text{si } \mathcal{A}[\![x]\!]s_{n+1} = \mathcal{A}[\![1]\!]s_{n+1} \end{cases}$$

$$= \begin{cases} tt & \text{si } \cancel{s_{n+1}}\ x \neq \mathcal{N}[\![1]\!] \\ ff & \text{si } s_{n+1}\,x = \mathcal{N}[\![1]\!] \end{cases} = \begin{cases} tt & \text{si } n+1\neq 1 \\ ff & \text{si } n+1 = 1 \end{cases} \underset{n\geq 1}{\uparrow}{=} tt.$$

Esto finaliza el paso inductivo y la demostración.

b) El argumento utilizado para el apartado a) se puede adaptar para probar que la configuración $\langle S_b, s\rangle$ termina si $s\,x \geq 0$.

Además en este caso, si $s\,x<0$ entonces la configuración también termina. En efecto, como $\mathcal{B}[\![1\leq x]\!]s=ff$ podemos aplicar la regla $[\text{while}^{ff}]$ y $\langle \text{While } 1\leq x \text{ do } (y:=y+x;\ x:=x-1),s\rangle \to s$, es decir, la conf. termina.

Como $\langle S_b, s\rangle$ termina si $sx \geq 0$ y $sx<0$ entonces termina siempre.

c) Vamos a argumentar que S_c siempre cicla. En un estado s_2

Supongamos que no, es decir, $\exists s_0 \in State$ tal que $\langle S_c, s_0\rangle$ termina, y

Por lo tanto existe cierto árbol de derivación finito de tamaño n.

Por construcción S_c solo se ha podido generar a partir de las reglas $[while^{ff}]$ y $[while^{tt}]$.

Si se hubiera construido por la primera regla entonces se verificaría $\mathcal{B}[\![true]\!]s_0 = ff$, pero esto es falso.

Por tanto se ha generado por la regla $[while^{tt}]$ y se verifica bien.

$$\langle skip, s_0\rangle \to s_1 \quad y \quad \langle S_c, s_1\rangle \to s_2$$

Por construcción del $\{skip\}$, se ha construido por la regla $[skip]$, luego $s_0 = s_1$, así que $\langle S_c, s_0\rangle \to s_2$. Pero esto implica que el árbol original de tamaño n tiene un subárbol de tamaño n, lo cual es absurdo y llegamos a contradicción.

(El propio árbol.)

Ejercicio 26: Probar que $S_1;(S_2;S_3)$ es semánticamente equivalente a $(S_1;S_2);S_3$.

Construir una instrucción para probar que, en general, $S_1;S_2$ no es sintácticamente equivalente a $S_2;S_1$.

Para lo primero, sean $s,s' \in State$ y hay que probar que

$$\langle S_1;(S_2;S_3),s\rangle \to s' \iff \langle (S_1;S_2);S_3,s\rangle \to s'$$

$\Rightarrow$] Supongamos que $\langle S_1;(S_2;S_3),s\rangle \to s'$. Por construcción, solo se ha podido generar aplicando la regla [comp] por lo que $\exists \bar{s} \in State$ tales que $\langle S_1,s\rangle \to \bar{s}$ y $\langle S_2;S_3,\bar{s}\rangle \to s'$. De la segunda derivación, como solo se ha podido generar por la regla [comp] obtenemos que $\exists \hat{s} \in State$ tal que $\langle S_2,\bar{s}\rangle \to \hat{s}$ y $\langle S_3,\hat{s}\rangle \to s'$. Con esto y aplicando las reglas [comp] podemos construir el árbol:

$$\dfrac{\dfrac{\langle S_1,s\rangle \to \bar{s},\ \langle S_2,\bar{s}\rangle \to \hat{s}}{\langle S_1;S_2,s\rangle \to \hat{s}}[comp] \qquad \langle S_3,\hat{s}\rangle \to s'}{\langle (S_1;S_2);S_3,s\rangle \to s'}[comp].$$

$\Leftarrow$] Totalmente análogo.

Para lo segundo consideramos $S_1 = x:=1$, $S_2 \equiv x:=2$, y veamos que $x:=1;x:=2$ no es sem. equ. a $x:=2;x:=1$.

Basta tomar los estados s_0, s_1 y s_2 tales que

$s_0\,x = 0$, $s_1\,x = 1$ y $s_2\,x = 2$ y comprobar que

$\langle x:=2; x:=1, s_0\rangle \rightarrow s_1$ y $\langle x:=1; x:=2, s_0\rangle \rightarrow s_2$.

Efectivamente, por la regla [asig] se tiene

1. $\langle x:=2, s_0\rangle \rightarrow s_2$ (Porque $s_2 = s_0[x \mapsto 2]$)
2. $\langle x:=1, s_2\rangle \rightarrow s_1$ (porque $s_1 = s_2[x \mapsto 1]$)
3. $\langle x:=1, s_0\rangle \rightarrow s_1$ (Porque $s_1 = s_0[x \mapsto 1]$
4. $\langle x:=2, s_1\rangle \rightarrow s_2$ (porque $s_2 = s_0[x \mapsto 2]$).

y aplicando la regla [comp]:

$$\frac{\overset{1}{\langle x:=2, s_0\rangle \rightarrow s_2} \quad \overset{2}{\langle x:=1, s_2\rangle \rightarrow s_1}}{\langle x:=2; x:=1, s_0\rangle \rightarrow s_1} \qquad \frac{\overset{3}{\langle x:=1, s_0\rangle \rightarrow s_1}, \ \overset{4}{\langle x:=2, s_1\rangle \rightarrow s_2}}{\langle x:=1; x:=2, s_0\rangle \rightarrow s_2}.$$

Esto prueba que, en general, $S_1; S_2$ no es sem. equ. a $S_2; S_1$

Ejercicio 2.7.- Extender el lenguaje While con la sentencia $S_1 := \text{repeat } S \text{ until } b$. y demostrar que es semánticamente equivalente a $S_2 := S; \underbrace{\text{if } b \text{ then skip else } (\text{repeat } S \text{ until } b)}_{S_3}$.

$$[\text{repeat}^{tt}] \quad \frac{\langle S, s\rangle \to s'}{\langle \text{repeat } S \text{ until } b, s\rangle \to s'} \quad \text{si } \mathcal{B}[\![b]\!]s' = tt$$

$$[\text{repeat}^{ff}] \quad \frac{\langle S, s\rangle \to s', \ \langle \text{repeat } S \text{ until } b, s'\rangle \to s''}{\langle \text{repeat } S \text{ until } b, s\rangle \to s''} \quad \text{si } \mathcal{B}[\![b]\!]s' = ff$$

Veamos que son semánticamente equivalentes:

Sea $s \in State$ y hay que probar que $\langle S_1, s\rangle \to s' \Leftrightarrow \langle S_2, s\rangle \to s'$

$\Rightarrow)$ Supongamos que $\langle S_1, s\rangle \to s'$. Por construcción, esta sólo se ha podido generar de dos formas:

CASO A: Habiendo aplicado $[\text{repeat}^{ff}]$ en cuyo caso se verificaban

$\langle S, s\rangle \to \bar{s}$, $\langle \text{repeat } S \text{ until } b, \bar{s}\rangle \to s'$ y $\mathcal{B}[\![b]\!]\bar{s} = ff$.

Por la regla $[\text{if}^{ff}]$ se tiene que:

$$\frac{\langle \text{repeat } S \text{ until } b, \bar{s}\rangle \to s'}{\langle \text{if } b \text{ then skip else } (\text{repeat } S \text{ until } b), \bar{s}\rangle \to s'} \quad \text{ya que}$$

$\mathcal{B}[\![b]\!]\bar{s} = ff.$, es decir, $\langle S_3, \bar{s}\rangle \to s'$

Aplicando la regla [comp] se llega a lo buscado ya que

$$\frac{<S,s> \rightarrow \bar{s} \quad , \quad <S_3,\bar{s}> \rightarrow s'}{<S;S_3,s> \rightarrow s'}$$

, es decir, $<S_2,s> \rightarrow s'$ lo que buscábamos.

CASO B: Habiendo aplicado $[\text{repeat}^{tt}]$ en cuyo caso se verificaban $<S,s> \rightarrow s'$ y $\mathcal{B}[\![b]\!]s' = tt$.

Aplicando la regla [skip] se tiene que $<\text{skip}, s'> \rightarrow s'$ y por la regla $[\text{if}^{tt}]$ se tiene

$$\frac{<\text{skip}, s'> \rightarrow s'}{<\text{if } b \text{ then skip; else repeat } S \text{ until } b, s'> \rightarrow s'}$$

ya que $\mathcal{B}[\![b]\!]s' = tt$

Por la regla [comp] se tiene

$$\frac{<S,s> \rightarrow s' \quad <\text{if } b \text{ then skip; else repeat } S \text{ until } b, s'> \rightarrow s'}{<S; \text{if } b \text{ then skip; else repeat } S \text{ until } b, s> \rightarrow s'}$$

$\Leftarrow$ Supongamos que $<S_2,s> \rightarrow s'$. Por construcción esta solo se ha podido obtener de la regla [comp] por lo que $\exists \bar{s}$ tal que $<S,s> \rightarrow \bar{s}$ y $<\text{if } b \text{ then skip else repeat } S \text{ until } b, \bar{s}> \rightarrow s'$.

Por construcción, esta segunda configuración solo ha podido generarse con las reglas $[\text{if}^{tt}]$ o $[\text{if}^{ff}]$.

CASO A: Por la regla $[\text{if}^{tt}]$, entonces se verificaba $\mathcal{B}[\![b]\!]\bar{s} = tt$ y $<\text{skip}, \bar{s}> \rightarrow s'$. Veamos que con esta información podemos probar que $<S_1,s> \rightarrow s'$.

Basta aplicar la regla [repeat^{tt}] ya que

$$\frac{\langle S, s\rangle \to \bar{s}}{\langle \text{repeat } S \text{ until } b, s\rangle \to \bar{s}} \quad \text{porque } \mathcal{A}[\![b]\!]\bar{s} = tt$$

Por tanto $\langle S_1, s\rangle \to \bar{s}$. Lo que queríamos probar es que $\langle S_1, s\rangle \to s'$ luego hay que ver que $\bar{s} = s'$, pero esto es cierto porque sabemos que $\langle skip, \bar{s}\rangle \to s'$
como la regla por lo que se ha construido es
y la regla [skip] ($\forall s \ \langle skip, s\rangle \to s$) entonces $\bar{s} = s'$

CASO B: Por la regla [if^{ff}], entonces se verifica $\mathcal{A}[\![b]\!]\bar{s} = ff$ y $\langle \text{repeat } S \text{ until } b, \bar{s}\rangle \to s'$

Podemos aplicar directamente la regla [repeat^{ff}] ya que

$$\frac{\langle S, s\rangle \to \bar{s}, \ \langle \text{repeat } S \text{ until } b, \bar{s}\rangle \to s', \ \mathcal{A}[\![b]\!]\bar{s} = ff}{\langle \text{repeat } S \text{ until } b, s\rangle \to s'}$$

es decir, $\langle S_1, s\rangle \to s'$

Ejercicio 2.10

Demostrar que

repeat S until b es equivalente a S; while ¬b do S

Para probarlo, sea $s \in$ State y veamos que

$$\underbrace{\langle \text{repeat } S \text{ until } b, s\rangle}\rightarrow s' \iff \langle \underbrace{S; \text{while } \neg b \text{ do } S}_{S_2}, s\rangle \rightarrow s'$$

$\Leftarrow$] Supongamos que $\langle S_2, s\rangle \rightarrow s'$

Por construcción S_2 solo se ha podido generar como la regla [comp] por lo que se ha partido de $\langle S; s\rangle \rightarrow \bar{s}$ y $\langle \text{while } \neg b \text{ do } S; \bar{s}\rangle \rightarrow s'$

Por construcción while ¬b do S sólo se ha podido generar aplicando las reglas $[\text{while}^{tt}]$ o $[\text{while}^{ff}]$.

CASO A: Si se ha aplicado la regla $[\text{while}^{tt}]$ es porque se verificaba.

$$\langle S; \bar{s}\rangle \rightarrow \hat{s}, \langle \text{while } \neg b \text{ do } S; \hat{s}\rangle \rightarrow s' \text{ y } \mathcal{B}[\![\neg b]\!]\bar{s} = tt$$

$\Updownarrow$ es prueba

$$\mathcal{B}[\![b]\!]\bar{s} = ff.$$

Por la regla [comp.] se tiene que

$$\frac{\langle S; \bar{s}\rangle \rightarrow \hat{s}; \langle \text{while } \neg b \text{ do } S; \hat{s}\rangle \rightarrow s'}{\langle S; \text{while } \neg b \text{ do } S; \bar{s}\rangle \rightarrow s'}$$

Por inducción sobre el árbol de derivación

$$\langle S; \text{while } \neg b \text{ do } S; \bar{s}\rangle \rightarrow s' \Rightarrow \langle \text{repeat } S \text{ until } b, \bar{s}\rangle \rightarrow s'$$

Aplicando la regla $[\text{repeat}^{ff}]$, como se tiene

$$\frac{\langle S, s\rangle \rightarrow \bar{s}, \langle \text{repeat } S \text{ until } b, \bar{s}\rangle \rightarrow s'}{\langle \text{repeat } S \text{ until } b, s\rangle \rightarrow s'} \quad \text{ya que } \mathcal{B}[\![b]\!]\bar{s} = ff$$

CASO B:- Si se ha aplicado la regla [while^{ff}] es porque

$\mathcal{B}[\![\neg b]\!]\bar{s} = ff \iff \mathcal{B}[\![b]\!]\bar{s} = tt$. y se deduce que $\bar{s} = s'$

Con esto podemos aplicar la regla [repeat^{tt}] y

$$\frac{\langle S, s\rangle \to \bar{s}}{\langle \text{repeat } S \text{ until } b, s\rangle \to \bar{s}}$$ ya que $\mathcal{B}[\![b]\!]\bar{s} = tt$

Como $\bar{s} = s'$ podemos concluir que

$\langle \text{repeat } S \text{ until } b, s\rangle \to s'$, como queríamos demostrar.

$\Longrightarrow$) Supongamos que $\langle S_1, s\rangle \to s'$

Por construcción, S_1 solo se ha podido generar por las reglas [repeat^{tt}] o [repeat^{ff}].

CASO A.- Se ha generado a partir de la regla [repeat^{tt}] por lo que se garantizaba $\langle S, s\rangle \to s'$ y ~~porque~~ $\mathcal{B}[\![b]\!]s' = tt$

A partir de $\mathcal{B}[\![b]\!]s' = tt$ podemos concluir que $\mathcal{B}[\![\neg b]\!]s' = ff$ y podemos aplicar la regla [while^{ff}] para llegar a que

$\langle \text{while } \neg b \text{ do } (S), s'\rangle \to s'$

Aplicando la regla [comp] se tiene

$$\frac{\langle S, s\rangle \to s', \quad \langle \text{while } \neg b \text{ do } S, s'\rangle \to s'}{\langle S; \text{while } \neg b \text{ do } S, s\rangle \to s'}$$ es decir,

$\langle S_2, s\rangle \to s'$

CASO B.- Se ha generado a partir de la regla [repeat^{ff}] por lo que se cumplían $\langle S, s\rangle \to \bar{s}$, $\langle \text{repeat } S \text{ until } b, \bar{s}\rangle \to s'$ y $\mathcal{B}[\![b]\!]\bar{s} = ff$

Por inducción sobre el tamaño del árbol

se cumple que $\langle \text{repeat } S \text{ until } b, \bar{s}\rangle \to s' \Rightarrow \langle S; \text{while } \neg b \text{ do } S, \bar{s}\rangle \to s'$

Por construcción $\langle S; \text{while } \neg b \text{ do } S, \bar{s}\rangle \to s'$ solo se ha podido generar por $\langle S, \bar{s}\rangle \to \bar{\bar{s}}$, $\langle \text{while } \neg b \text{ do } S, \bar{\bar{s}}\rangle \to s'$

Aplicando la regla [while^{tt}] $$\frac{\langle S, \bar{s}\rangle \to \bar{\bar{s}}, \quad \langle \text{while } \neg b \text{ do } S, \bar{\bar{s}}\rangle \to s'}{\langle \text{while } \neg b \text{ do } S, \bar{s}\rangle \to s'}$$

porque $\mathcal{B}[\![\neg b]\!]\bar{s} = tt$

Juntando lo anterior y aplicando la regla [comp]

$$\frac{\langle S, s\rangle \to \bar{s},\ \langle \text{while } \neg b \text{ do } S, \bar{s}\rangle \to s'}{\langle S; \text{while } \neg b \text{ do } S, s\rangle \to s'}\quad , \text{ es decir, } \langle S_2, s\rangle \to s'$$

Ejercicio 2.11.- Propón una semántica operacional para las Aexp y prueba una cierta equivalencia entre la semántica denotacional y la semántica operacional.

[nat] $\langle n, s\rangle \longrightarrow \mathcal{N}[\![n]\!]$

[var] $\langle x, s\rangle \longrightarrow s\,x$

[sum] $\dfrac{\langle a_1, s\rangle \to z_1,\ \langle a_2, s\rangle \to z_2}{\langle a_1 + a_2, s\rangle \to z_1 + z_2}$

[rest] $\dfrac{\langle a_1, s\rangle \to z_1,\ \langle a_2, s\rangle \to z_2}{\langle a_1 - a_2, s\rangle \to z_1 - z_2}$

[prod] $\dfrac{\langle a_1, s\rangle \to z_1,\ \langle a_2, s\rangle \to z_2}{\langle a_1 * a_2, s\rangle \to z_1 * z_2}$

Veamos que $\forall a \in \text{Aexp}$ se verifica que $\forall s \in \text{State}$

$$\langle a, s\rangle \longrightarrow \mathcal{A}[\![a]\!]\, s.$$

Por casos sobre las Aexp y razonando por inducción estructural:

Si $a = n \Rightarrow \langle n, s\rangle \xrightarrow{[nat]} \mathcal{N}[\![n]\!] = \mathcal{A}[\![n]\!]\, s$ ✓

Si $a = x \Rightarrow \langle x, s\rangle \xrightarrow{[var]} s\,x = \mathcal{A}[\![x]\!]\, s$ ✓

Si $a = a_1 \; op \; a_2$, por hipótesis de inducción se tiene que

$$\langle a_1, s\rangle \to \mathcal{A}[\![a_1]\!]s \quad y \quad \langle a_2, s\rangle \to \mathcal{A}[\![a_2]\!]s$$

Aplicando la regla [op], como $\dfrac{\langle a_1, s\rangle \to \mathcal{A}[\![a_1]\!]s \;,\; \langle a_2, s\rangle \to \mathcal{A}[\![a_2]\!]s}{\langle a_1 \; op \; a_2, s\rangle \to \mathcal{A}[\![a_1]\!]s \; op \; \mathcal{A}[\![a_2]\!]s}$.

Por tanto $\langle a_1 \; op \; a_2, s\rangle \longrightarrow \mathcal{A}[\![a_1]\!]s \; op \; \mathcal{A}[\![a_2]\!]s = \mathcal{A}[\![a_1 \, op \, a_2]\!]s$.

Ejercicio 2.12.- Lo mismo para los booleanos.

[true] $\langle true, s\rangle \longrightarrow tt$

[false] $\langle false, s\rangle \longrightarrow ff$

$[eq^{tt}]$ $\dfrac{\langle a_1, s\rangle \to z_1 \,,\, \langle a_2, s\rangle \to z_2 \,;\, z_1 = z_2}{\langle a_1 = a_2, s\rangle \to tt}$

$[eq^{ff}]$ $\dfrac{\langle a_1, s\rangle \to z_1 \,,\, \langle a_2, s\rangle \to z_2 \,;\, z_1 \neq z_2}{\langle a_1 = a_2, s\rangle \to ff}$.

$[leq^{tt}]$ $\dfrac{\langle a_1, s\rangle \to z_1 \,,\, \langle a_2, s\rangle \to z_2 \,,\, z_1 \leq z_2}{\langle a_1 \leq a_2, s\rangle \to tt}$

$[leq^{ff}]$ $\dfrac{\langle a_1, s\rangle \to z_1 \,,\, \langle a_2, s\rangle \to z_2 \,,\, z_1 > z_2}{\langle a_1 \leq a_2, s\rangle \to ff}$

$[not^{tt}]$ $\dfrac{\langle b, s\rangle \to ff}{\langle \neg b, s\rangle \to tt}$

$[not^{ff}]$ $\dfrac{\langle b, s\rangle \to tt}{\langle \neg b, s\rangle \to ff}$

$[and^{tt}]$ $\dfrac{\langle b_1, s\rangle \to tt \,,\, \langle b_2, s\rangle \to tt}{\langle b_1 \wedge b_2, s\rangle \to tt}$

$[and_1^{ff}]$ $\dfrac{\langle b_1, s\rangle \to ff}{\langle b_1 \wedge b_2, s\rangle \to ff}$

$[and_2^{ff}]$ $\dfrac{\langle b_2, s\rangle \to ff}{\langle b_1 \wedge b_2, s\rangle \to ff}$

Veamos que $\forall b \in Bexp$ se verifica que $\forall s \in State$

$$\langle b, s \rangle \to \mathcal{B}[\![b]\!] s$$

Por casos sobre las Bexp y razonando por inducción estructural:

Si $b = true$ $\quad \langle true, s \rangle \xrightarrow{[true]} tt = \mathcal{B}[\![true]\!] s$

Si $b = false$ $\quad \langle false, s \rangle \xrightarrow{[false]} ff = \mathcal{B}[\![false]\!] s$.

Si $b = a_1 = a_2$ $\quad$ Distinguimos los casos $\mathcal{A}[\![a_1]\!] s = \mathcal{A}[\![a_2]\!] s$ y $\mathcal{A}[\![a_1]\!] s \neq \mathcal{A}[\![a_2]\!] s$.

A] $\mathcal{A}[\![a_1]\!] s = \mathcal{A}[\![a_2]\!] s$.

Utilizando el ejercicio anterior $\langle a_1, s \rangle \to \mathcal{A}[\![a_1]\!] s$ y $\langle a_2, s \rangle \to \mathcal{A}[\![a_2]\!] s$.

Como $\mathcal{A}[\![a_1]\!] s = \mathcal{A}[\![a_2]\!] s$ se deduce por la regla $[eq^{tt}]$.

$$\frac{\langle a_1, s \rangle \to \mathcal{A}[\![a_1]\!] s, \ \langle a_2, s \rangle \to \mathcal{A}[\![a_2]\!] s, \ \mathcal{A}[\![a_1]\!] s = \mathcal{A}[\![a_2]\!] s}{\langle a_1 = a_2, s \rangle \to tt \overset{def}{=} \mathcal{B}[\![a_1 = a_2]\!] s}$$

B] Si $\mathcal{A}[\![a_1]\!] s \neq \mathcal{A}[\![a_2]\!] s$.

Utilizando el ejercicio anterior $\langle a_1, s \rangle \to \mathcal{A}[\![a_1]\!] s$ y $\langle a_2, s \rangle \to \mathcal{A}[\![a_2]\!] s$.

Como $\mathcal{A}[\![a_1]\!] s \neq \mathcal{A}[\![a_2]\!] s$ se deduce por la regla $[eq^{ff}]$

$$\frac{\langle a_1, s \rangle \to \mathcal{A}[\![a_1]\!] s, \ \langle a_2, s \rangle \to \mathcal{A}[\![a_2]\!] s, \ \mathcal{A}[\![a_1]\!] s \neq \mathcal{A}[\![a_2]\!] s}{\langle a_1 = a_2, s \rangle \to ff \overset{def}{=} \mathcal{B}[\![a_1 = a_2]\!] s}$$

En cualquiera de los casos. $\langle a_1 = a_2, s \rangle \to \mathcal{B}[\![a_1 = a_2]\!] s$.

Si $b = a_1 \leq a_2$ se prueba de manera análoga al caso $b = a_1 = a_2$

Si: $b = \neg b_1$ Por hipótesis de inducción sabemos que

$$\langle b_1, s\rangle \longrightarrow \mathcal{B}[\![b_1]\!]\, s.$$

Distinguimos:

A] $\mathcal{B}[\![b_1]\!]\, s = tt$ Aplicando la regla $[not^{ff}]$

$$\frac{\langle b_1, s\rangle \to tt}{\langle \neg b_1, s\rangle \to ff = \mathcal{B}[\![\neg b_1]\!]\, s.}$$

B] $\mathcal{B}[\![b_1]\!]\, s = ff$. Aplicando la regla $[not^{tt}]$

$$\frac{\langle b_1, s\rangle \to ff}{\langle \neg b_1, s\rangle \to tt = \mathcal{B}[\![\neg b_1]\!]\, s.}$$

En cualquiera de los casos $\langle \neg b_1, s\rangle \to \mathcal{B}[\![\neg b_1]\!]\, s.$

Si $b = b_1 \wedge b_2$. Por hipótesis de inducción sabemos que

$$\langle b_1, s\rangle \longrightarrow \mathcal{B}[\![b_1]\!]\, s$$
$$\langle b_2, s\rangle \longrightarrow \mathcal{B}[\![b_2]\!]\, s.$$

Distinguimos:

A] $\mathcal{B}[\![b_1]\!]\, s = tt$ y $\mathcal{B}[\![b_2]\!]\, s = tt$. Aplicando la regla $[and^{tt}]$

$$\frac{\langle b_1, s\rangle \to tt, \ \langle b_2, s\rangle \to tt}{\langle b_1 \wedge b_2, s\rangle \to tt = \mathcal{B}[\![b_1 \wedge b_2]\!]\, s.}$$

B] $\mathcal{B}[\![b_1]\!]\, s = ff$. Aplicando la regla $[and_1^{ff}]$

$$\frac{\langle b_1, s\rangle \to ff}{\langle b_1 \wedge b_2, s\rangle \to ff = \mathcal{B}[\![b_1 \wedge b_2]\!]\, s.}$$

C] $\mathcal{B}[\![b_2]\!]\, s = ff$. Aplicando la regla $[and_2^{ff}]$

$$\frac{\langle b_2, s\rangle \to ff}{\langle b_1 \wedge b_2, s\rangle \to ff = \mathcal{B}[\![b_1 \wedge b_2]\!]\, s}$$

En todos los casos $\langle b_1 \wedge b_2, s\rangle \to \mathcal{B}[\![b_1 \wedge b_2]\!]\, s$