Ejercicio 2.16.- Construlle una secuencia de derivación para

$S \equiv z:=0;\ \underbrace{\text{while } y \le x \text{ do } z:=z+1;\ x:=x-y}_{S_1}$ al ejecutarla

en un estado s tal que $s\,x = 17$ y $s\,y = 5$.

Por la regla [asign] $\langle z:=0, s\rangle \Rightarrow \overbrace{s[z \mapsto 0]}^{s_1}$

Por la regla [comp²] $\dfrac{\langle z:=0, s\rangle \Rightarrow s_1}{\langle S, s\rangle \overset{1}{\Rightarrow} \langle S_1, s_1\rangle}$

Por la regla [while] $\langle S_1, s_1\rangle \overset{2}{\Rightarrow} \langle \text{if } y \le x \text{ then } (z:=z+1; x:=x-y); S_1 \text{ else skip}, s_1\rangle$

Como $\mathcal{B}[\![y \le x]\!]s_1 = tt$, podemos aplicar la regla [if^{tt}] y obtener

$\langle \text{if } y \le x \text{ then } (z:=z+1; x:=x-y); S_1 \text{ else skip}, s_1\rangle \overset{3}{\Rightarrow} \langle (z:=z+1, x:=x-y); S_1, s_1\rangle$

Aplicando la regla [asign] $\langle z:=z+1, s_1\rangle \Rightarrow \underbrace{s_1[z \mapsto 1]}_{s_2}$ y por la regla [comp²] $\langle z:=z+1, x:=x-y, s_1\rangle \Rightarrow \langle x:=x-y, s_2\rangle$. Aplicando [comp¹]

$\langle (z:=z+1; x:=x-y); S_1, s_1\rangle \overset{4}{\Rightarrow} \langle x:=x-y; S_1, s_2\rangle$

Por la regla [asign] $\langle x:=x-y, s_2\rangle \Rightarrow \underbrace{s_2[x \mapsto 12]}_{s_3}$ así que aplicando [comp²] $\langle x:=x-y; S_1, s_2\rangle \overset{5}{\Rightarrow} \langle S_1, s_3\rangle$.

Si volvemos a aplicar [while], $\langle S_1, s_3\rangle \overset{6}{\Rightarrow} \langle \text{if } y \le x \text{ then } (z:=z+1; x:=x-y); S_1 \text{ else skip}, s_3\rangle$

y como $\mathcal{B}[\![y \le x]\!]s_3 = tt$ aplicamos [if^{tt}] y llegamos a:

$\langle \text{if } y \le x \text{ then} (z:=z+1); (x:=x-y)); S_1 \text{ else skip}, s_3\rangle \overset{7}{\Rightarrow} \langle (z:=z+1; x:=x-y); S_1, s_3\rangle$

Aplicando [asign] $\langle z:=z+1, s_3\rangle \Rightarrow \underbrace{s_3[z\mapsto 2]}_{s_4}$ y por las reglas [$comp_1$] y [$comp_2$]

se llega a $\langle (z:=z+1; x:=x-y); S_1, s_3\rangle \overset{⑧}{\Rightarrow} \langle x:=x-y; S_1, s_4\rangle$

Nuevamente por la regla [asign] $\langle x:=x-y, s_4\rangle \Rightarrow s_4[x\mapsto 7]$ y por la regla [$comp^2$] $\langle x:=x-y; S_1, s_4\rangle \overset{⑨}{\Rightarrow} \langle S_1, s_5\rangle$.

Desplegando el while, aplicando if^{tt} y las reglas de asignación y composición oportunas se llega a:

$\langle S_1, s_5\rangle \overset{⑩}{\Rightarrow} \langle$if $y\le x$ then $(z:=z+1; x:=x-y); S_1$ else skip, $s_5\rangle \overset{⑪}{\Rightarrow} \langle (z:=z+1; x:=x-y) S_1, s_5\rangle \overset{⑫}{\Rightarrow}$

$\Rightarrow \langle x:=x-y; S_1, s_6\rangle \overset{⑬}{\Rightarrow} \langle S_1, s_7\rangle$ donde $s_6 = s_5[z\to 3]$ y

$s_7 = s_6[x\to 2]$

Aplicando la regla while

$\langle S_1, s_7\rangle \overset{⑭}{\Rightarrow} \langle$if $y\le x$ then $z:=z+1, x:=x-y; S_1$ else skip, $s_7\rangle$ y como finalmente $\mathcal{A}[\![y\le x]\!] s_7 = ff$ ya que $s x = 2 < s y$ 5 entonces aplicamos la regla [if ff] y se tiene que

$\langle$if $y\le x$ then $(z:=z+1, x:=x-y); S_1$ else skip, $s_7\rangle \overset{⑮}{\Rightarrow} \langle$skip, $s_7\rangle \overset{⑯}{\Rightarrow} s_7$

donde lo último es por la regla [skip]. Por tanto $\langle S, s\rangle \Rightarrow^{16} s_7$ donde

$s_7\, x = 2$, $s_7\, y = 5$ y $s_7\, z = 3$ y $s_7 = s$ para el resto de variables

Se pide determinar un estado s tal que la secuencia de derivación para S en ese estado sea infinita. Consideramos s_0 tal que $s_0\, x = 1$ y $s_0\, y = 0$

Como las variables x e y no cambian de valor en una ejecución del bucle y son las únicas que intervienen en la condición de parada (que es cierta siempre) intuitivamente esta secuencia de derivación será infinita.

Probamos lo anterior con algo más de rigor. Afirmamos que si partimos de una configuración

$$\gamma_i = \langle \overbrace{\text{while } y \le x \text{ do } (z:=z+1); x:=x-y}^{S_1}), s_i \rangle \quad \text{donde } s_i\, z = i,\ s_i\, x = 1 \text{ y } s_i\, y = 0$$

entonces en k pasos llegamos a una configuración

$$\gamma_{i+1} = \langle \text{while } y \le x \text{ do } (z:=z+1); x:=x-y, s_{i+1} \rangle \quad \text{donde } s_{i+1}\, z = i+1,\ s_{i+1}\, x = 1 \text{ y } s_{i+1}\, y = 0.$$

Efectivamente, por la regla [while] primero y por la regla [if tt] después

$$\langle S_1; s_i \rangle \overset{1}{\Rightarrow} \langle \text{if } y \le x \text{ then } (z:=z+1); x:=x-y); S_1 \text{ else skip}, s_i \rangle \overset{2}{\Rightarrow}$$

$$\Rightarrow \langle (z:=z+1; x:=x-y); S_1, s_i \rangle \quad \text{ya que } \mathcal{B}[\![y \le x]\!]\, s_i = tt \text{ por ser } s_i\, y = 0 \le s_i\, x = 1.$$

Por las reglas [asign], [comp²] y [comp¹]

$$\cfrac{\cfrac{\langle z:=z+1; s_i \rangle \Rightarrow s_{i+1}}{\langle z:=z+1; x:=x-y, s_i \rangle \Rightarrow \langle x:=x-y, s_{i+1} \rangle}}{\langle (z:=z+1; x:=x-y); S_1, s_i \rangle \overset{3}{\Rightarrow} \langle x:=x-y; S_1, s_{i+1} \rangle}.$$

Por la regla [comp²] y [asign]

$$\frac{\langle x:=x-y; s_{i+1} \rangle \Rightarrow s_{i+1}}{\langle x:=x-y; S_1, s_{i+1} \rangle \overset{4}{\Rightarrow} \langle S_1, s_{i+1} \rangle}$$

Por tanto $\langle S_1; s_i \rangle \Rightarrow^4 \langle S_1, s_{i+1} \rangle$ $\forall i \in \mathbb{N}$, es decir, $\gamma_i \Rightarrow^4 \gamma_{i+1}$ $\forall i \in \mathbb{N}$.

Además $\langle S_1, s_0 \rangle \Rightarrow \langle S_1, s_1 \rangle$ por simple aplicación de [asign] y [comp²].

Queda probado que $\langle S, s_0 \rangle$ tiene una secuencia de derivación infinita

Ejercicio 2.17 Extiende el lenguaje While con una construcción repeat S until b y especifica una semántica operacional de paso corto para dicha instrucción (No uses while).

Proponemos el despliegue del repeat con un if. De manera similar a como se ha hecho con el while (para el que se demostró en la semántica de paso largo que era equivalente a su desplegado con if y después se definió la semántica operacional de paso corto como $\langle$ while b do S, s$\rangle \Rightarrow$ if b then S; else skip, s$\rangle$), proponemos que la semántica para el repeat sea:

$$\langle \underbrace{\text{repeat } S \text{ until } b}_{S_1}, s\rangle \Rightarrow \langle \underbrace{S; \text{ if } b \text{ then skip else (repeat } S \text{ until } b)}_{S_2}, s\rangle$$

Recordamos que, según la definición de equivalencia semántica de paso largo, S_1 y S_2 eran equivalentes. En el ejercicio 2.4 veremos que, con esta semántica, S_1 es semánticamente equivalente (en el sentido de semántica de paso corto) a S; while $\neg b$ do S, igual que ya probamos para la semántica de paso largo.

Ejercicio 2.20.- Supongamos que $\langle S_1; S_2, s\rangle \Rightarrow^* \langle S_2, s'\rangle$. Demuestra que no necesariamente se tiene que cumplir que $\langle S_1, s\rangle \Rightarrow^* s'$

Sea $S_1 = x:=1$ y $S_2 = (\text{while true do } x:=x+1)$ y $s \in State$.

Definimos $s' = s[x \mapsto 1]$ y $s'' = s[x \mapsto 2]$.

Entonces

$$\langle x:=1, s\rangle \overset{[asign]}{\Longrightarrow} s'$$

$$\langle S_1; S_2, s\rangle \xrightarrow{[comp]} \langle S_2, s'\rangle \xrightarrow{[while]} \langle \text{if true then } (x:=x+1; S_2) \text{ else skip}, s'\rangle$$

$$\xrightarrow{[if^{tt}]} \langle x:=x+1; S_2, s'\rangle \xrightarrow{[comp]} \langle S_2, s''\rangle.$$

$$\langle x:=x+1, s'\rangle \overset{[asign]}{\Longrightarrow} s''$$

Por tanto $\langle S_1; S_2, s\rangle \Rightarrow^* \langle S_2, s''\rangle$.

Sin embargo, $\langle S_1, s\rangle \Rightarrow^* s''$ es falso ya que la única regla que se ha podido aplicar ha sido [asign] y $\langle x:=1, s\rangle \Rightarrow s'$ y $s' \neq s''$.

Ejercicio 2.21.- Probar que si $\langle S_1, s\rangle \Rightarrow^k s'$ entonces $\langle S_1; S_2, s\rangle \Rightarrow^k \langle S_2, s'\rangle$

Procedemos por inducción sobre la longitud de la cadena de derivación (k).

Caso base| $k=0$. Se verifica trivialmente porque $\langle S_1, s\rangle \not\Rightarrow^0 s'$

ya que $\langle S_1, s\rangle \neq s'$.

Paso inductivo| Sea $k \geq 0$ y supongamos cierta la propiedad para

k y veamos que se verifica para $k+1$.

Supongamos por tanto que $\langle S_1, s\rangle \Rightarrow^{k+1} s'$ y queremos probar que

$\langle S_1; S_2, s\rangle \Rightarrow^{k+1} \langle S_2, s'\rangle$.

Como $\langle S_1, s\rangle \Rightarrow^{k+1} s'$, entonces $\langle S_1, s\rangle \Rightarrow^1 \gamma \Rightarrow^k s'$

Distinguimos los casos en los que γ es una configuración o un estado.

Caso A| Si $\gamma = \hat{s}$, entonces, por las reglas de construcción de las que disponimos solo puede pasar que $k=0$ y $\hat{s} = s'$. Pero entonces

$\langle S_1, S_2, s\rangle \Rightarrow^{k+1} \langle S_2, s'\rangle \iff \langle S_1; S_2, s\rangle \Rightarrow^1 \langle S_2, s'\rangle$ lo cuál

es cierto por la aplicación de la regla $[comp^2]$ ya que $\langle S_1, s\rangle \Rightarrow s'$

Caso B| Si $\gamma = \langle S_3, s''\rangle$ $\left(\langle S_1, s\rangle \Rightarrow^1 \langle S_3, s''\rangle \Rightarrow^k s'\right)$.

Por hipótesis de inducción aplicada a $\langle S_3, s''\rangle \Rightarrow^k s'$ se tiene que

$\langle S_3; S_2, s''\rangle \Rightarrow^k \langle S_2, s'\rangle$. Aplicando $[comp^1]$

$$\frac{\langle S_1, s\rangle \Rightarrow \langle S_3, s''\rangle}{\langle S_1; S_2, s\rangle \Rightarrow \langle S_3; S_2, s''\rangle}$$

y juntando todo

$\langle S_1; S_2, s\rangle \Rightarrow \langle S_3; S_2, s''\rangle \Rightarrow^k \langle S_2, s'\rangle$, es decir, $\langle S_1; S_2, s\rangle \Rightarrow^{k+1} \langle S_2, s'\rangle$

Ejercicio 2.22.- Demuestra que la semántica operacional del While es determinista (en el paso). Deduce que existe una una secuencia de derivación para cada configuración $<S, s>$. Argumenta que una instrucción S de While no puede terminar y ciclar para un mismo estado s y que, por tanto, no puede siempre terminar y siempre ciclar.

Para lo primero, sea $s \in$ State y S una instrucción. Razonando por casos sobre S e inducción estructural.

Si $S = skip$ y $<skip, s> \Rightarrow \gamma$, $<skip, s> \Rightarrow \gamma'$, entonces, en ambos casos la única regla que se podía aplicar era [skip] y $\gamma = \gamma' = s$

Si $S = x := a$ y $<x := a, s> \Rightarrow \gamma$, $<x := a, s> \Rightarrow \gamma'$, como la única regla que se ha podido aplicar es [asign] $\Rightarrow \gamma = \gamma' = s[x \mapsto \mathcal{A}[\![a]\!]s]$

Si $S = $ if b then S_1 else S_2 y $<S, s> \Rightarrow \gamma$, $<S, s> = \gamma'$, distinguimos los casos en los que $\mathcal{B}[\![b]\!]s = tt$ y $\mathcal{B}[\![b]\!]s = ff$. En el primero de ellos, la única regla que se podía aplicar para ambas configuraciones era [iftt] por lo que $\gamma = \gamma' = <S_1, s>$. En caso de que $\mathcal{B}[\![b]\!]s$ la única regla que se podía usar era [ifff] y $\gamma = \gamma' = <S_2, s>$.

Si $S = $ while b do S_1 y $<S, s> = \gamma$ y $<S, s> \Rightarrow \gamma'$ entonces, como en ambos casos la única regla que se podía usar era [while] se tiene que $\gamma = \gamma' = <$if b then $S_1; S$ else skip$, s>$

Por último, si $S = (S_1; S_2)$ y $\langle S, s\rangle \Rightarrow \gamma$ y $\langle S, s\rangle \Rightarrow \gamma'$, podemos utilizar la hipotesis de inducción sobre S_1 con lo que sabemos que si $\langle S_1, s\rangle \Rightarrow \bar{\gamma}$ y $\langle S_1, s\rangle \Rightarrow \bar{\bar{\gamma}} \Rightarrow \bar{\gamma} = \bar{\bar{\gamma}}$.

En ambos casos ($\langle S, s\rangle \Rightarrow \gamma$ y $\langle S, s\rangle \Rightarrow \gamma'$), las únicas reglas aplicables han sido [comp^1] o [comp^2].

Si para $\langle S, s\rangle \Rightarrow \gamma$ se ha aplicado [comp$_1$] se tiene que $\langle S_1, s\rangle \Rightarrow \langle S_1', s'\rangle$ para ciertos S_1' y s'. No puede haber sucedido que $\langle S, s\rangle \Rightarrow \gamma'$ se haya obtenido aplicando [comp$_2$] ya que en dicho caso $\langle S_1, s\rangle \Rightarrow s''$ y esto entraría en contradicción con la HI. Por tanto $\langle S, s\rangle \Rightarrow \gamma'$ también se ha obtenido aplicando [comp$_1$] luego se verificaba $\langle S_1, s\rangle \Rightarrow \langle \hat{S}_1, \hat{s}\rangle$ para cierto $\hat{S}$ y $\hat{s}$. Por HI, $\hat{S}_1 = S_1'$ y $s' = \hat{s}$, luego $\gamma = \gamma' = \langle S_1'; S_2, s'\rangle$.

Si para $\langle S, s\rangle \Rightarrow \gamma$ se ha aplicado [comp$_2$] se tiene que $\langle S_1, s\rangle \Rightarrow s'$. Como por HI no se puede dar que $\langle S_1, s\rangle \Rightarrow \langle S_1', s'\rangle$, la única regla aplicable en el paso de derivación $\langle S, s\rangle \Rightarrow \gamma'$ ha sido [comp$_2$] y $\langle S_1, s\rangle \Rightarrow s''$. Por H $s' = s''$ y se deduce que $\gamma = \gamma' = \langle S_2; s'\rangle$

Para lo segundo veamos por inducción sobre k(= pasos de derivación), que $\langle S, s\rangle \Rightarrow^k \gamma$ y $\langle S, s\rangle \Rightarrow^k \gamma' \Rightarrow \gamma = \gamma'$.

<u>Caso base</u>| $k=0$ Trivialmente cierto

<u>Paso inductivo</u>| Supuesto cierto para $k \geq 0$ lo probamos para $k+1$.

$$\langle S, s\rangle \Rightarrow^{k+1} \gamma \overset{k\geq 0}{\Longleftrightarrow} \langle S, s\rangle \Rightarrow \bar{\gamma} \Rightarrow^k \gamma$$

$$\langle S, s\rangle \Rightarrow^{k+1} \gamma' \overset{k\geq 0}{\Longleftrightarrow} \langle S, s\rangle \Rightarrow \bar{\bar{\gamma}} \Rightarrow^k \gamma'.$$

Por lo probado en la primera parte del ejercicio $\bar{\gamma} = \bar{\bar{\gamma}}$.

Si $\bar{\gamma} = \bar{\bar{\gamma}} \in State$ entonces se tiene que cumplir que $k=0$ (no se ha podido aplicar ninguna regla) y $\bar{\gamma} = \gamma$ y $\bar{\bar{\gamma}} = \gamma'$ $\Longrightarrow \gamma = \gamma'$.

Si $\bar{\gamma} = \bar{\bar{\gamma}} = \langle S_1, s_1 \rangle$ entonces, aplicando la HI se tiene que $\gamma = \gamma'$

Una vez visto que $\langle S, s \rangle \Longrightarrow^k \gamma$ y $\langle S, s \rangle \Longrightarrow^k \gamma' \Rightarrow \gamma = \gamma'$ veamos que $\langle S, s \rangle \Longrightarrow^* \gamma$ y $\langle S, s \rangle \Longrightarrow^* \gamma' \Longrightarrow \gamma = \gamma'$. (Donde γ y γ' son configuraciones terminales)

Si $\langle S, s \rangle \Longrightarrow^* \gamma$ entonces $\exists k_1 \geq 0$ tal que $\langle S, s \rangle \Longrightarrow^{k_1} \gamma$.

Si $\langle S, s \rangle \Longrightarrow^* \gamma'$ entonces $\exists k_2 \geq 0$ tal que $\langle S, s \rangle \Longrightarrow^{k_2} \gamma'$.

Podemos suponer sin pérdida de generalidad que $k_1 \leq k_2$.

Entonces por el determinismo en k_1 pasos

$$\langle S, s \rangle \Longrightarrow^{k_1} \gamma$$

$$\langle S, s \rangle \Longrightarrow^{k_1} \gamma \Longrightarrow^{k_2 - k_1} \gamma'$$

Como γ es terminal, si $\gamma \in State$ entonces $k_2 - k_1 = 0$ y $\gamma' = \gamma$ y si $\gamma = \langle S_1, s_1 \rangle$; como γ es terminal entonces tiene que estar bloqueada y $k_2 - k_1 = 0$ y $\gamma' = \gamma$.

Esto termina de probar que la secuencia de derivación (finita) es única.

Como esto se cumple para cadenas de derivación de tamaño n, para n arbitrariamente grande también se cumple para secuencias de derivación infinitas. El argumento simplemente es que si existen dos cadenas de derivación que no coinciden el la configuración γ_i, como la secuencia de derivación hasta γ_i es finita, es única y llegamos a contradicción.

Por lo probado anteriormente una configuración $\langle S, s \rangle$ no puede terminar y ciclar ya que ~~por la~~ unicidad de la secuencia de derivación esta no puede ser a la vez finita e infinita.

Evidentemente, si siempre cicla, nunca termina y en particular, no termina siempre.

Ejercicio 2.23.- Demostrar que son semánticamente equivalentes:

a) $S;skip$ y S.

Sea $s \in State$ y veamos que

1) $\langle S;skip,s\rangle \Rightarrow^* \gamma \iff \langle S,s\rangle \Rightarrow^* \gamma$ (γ config. terminal)

2) La secuencia de derivación $\langle S;skip,s\rangle$ es infinita $\iff$ $\langle S,s\rangle$ es infinita

1) $\Rightarrow]$ Supongamos que $\langle S;skip,s\rangle \Rightarrow^* \gamma$, es decir, $\exists k \geq 0$ tal que $\langle S;skip,s\rangle \Rightarrow^k \gamma$, donde γ es una configuración bloqueada o un estado final. Asumamos que ya hemos probado que ninguna configuración del lenguaje While se bloquea así que sea $s' \in State$ tal que $\langle S;skip,s\rangle \Rightarrow^k s'$. Por el Lema 2.19 $\exists \hat{s} \in State$ y $\exists k_1,k_2 \geq 0$ tales que $\langle S,s\rangle \Rightarrow^{k_1} \hat{s}$, $\langle skip,\hat{s}\rangle \Rightarrow^{k_2} s'$ y $k_1+k_2=k$

Como la única regla aplicable a la estructura sintáctica skip es [skip] deducimos que $k_2=1$ y $\hat{s}=s'$ Por tanto $\langle S,s\rangle \Rightarrow^{k-1} s'$ luego $\langle S,s\rangle \Rightarrow^* \gamma$.

$\Leftarrow]$ Supongamos que $\langle S,s\rangle \Rightarrow^* \gamma$, es decir, $\exists k \geq 0$ tal que $\langle S,s\rangle \Rightarrow^k \gamma$ donde γ es una configuración bloqueada o estado final. Nuevamente tratamos solo el caso en el que $\gamma = s'$ porque el otro no se puede dar. Por tanto $\langle S,s\rangle \Rightarrow^k s'$.

Vamos a probar un resultado previo antes de concluir la demostración y algo más general de lo que necesitamos:

Si por $\langle S_1, s\rangle \Rightarrow^{k_1} s'$ y $\langle S_2, s'\rangle \Rightarrow^{k_2} s''$ $\Longrightarrow$ $\langle S_1; S_2, s\rangle \Rightarrow^{k_1+k_2} s''$

Basta probar que si $\langle S_1, s\rangle \Rightarrow^{k_1} s'$ entonces $\langle S_1; S_2, s\rangle \Rightarrow^{k_1} \langle S_2; s'\rangle$

ya que visto esto $\langle S_1; S_2, s\rangle \Rightarrow^{k_1} \langle S_2; s'\rangle \Rightarrow^{k_2} s''$, luego

$$\langle S_1; S_2, s\rangle \Longrightarrow^{k_1+k_2} s''.$$

(Ejercicio 2.21)

Veamos por inducción que $\forall k \geq 0$ si $\langle S_1, s\rangle \Rightarrow^{k} s'$ entonces $\langle S_1; S_2, s\rangle \Rightarrow^{k} \langle S_2; s'\rangle$.

Caso base | $k=0$. Se verifica trivialmente ya que $\langle S_1, s\rangle \neq s'$

Paso inductivo | Sea $k \geq 0$ y supongamos que se cumple la propiedad para k y hay que probarla para $k+1$.

Supongamos por tanto que $\langle S_1, s\rangle \Rightarrow^{k+1} s'$ y hay que probar que $\langle S_1; S_2, s\rangle \Rightarrow^{k+1} \langle S_2; s'\rangle$.

Como $\langle S_1, s\rangle \Rightarrow^{k+1} s'$ entonces $\langle S_1, s\rangle \Rightarrow^{1} \gamma \Rightarrow^{k} s'$

Distinguimos si γ es un estado final o configuración.

Caso A | Si $\gamma = \hat{s} \in State$ entonces $\hat{s} = s'$ y $k=0$ así que basta aplicar comp ya que

$$\frac{\langle S_1, s\rangle \Rightarrow s'}{\langle S_1; S_2, s\rangle \Rightarrow \langle S_2; s'\rangle} \Leftrightarrow \langle S_1; S_2, s\rangle \Rightarrow^{\overset{1}{\overset{\shortparallel}{k+1}}} \langle S_2; s'\rangle \checkmark$$

Caso B | Si $\gamma = \langle S_3, \hat{s}\rangle$ $(\langle S_1, s\rangle \Rightarrow^{1} \langle S_3, \hat{s}\rangle \Rightarrow^{k} s')$.

Podemos aplicar la HI a $\langle S_3, \hat{s}\rangle \Rightarrow^{k} s'$ con lo que se tiene

$$\langle S_3; S_2, \hat{s}\rangle \Rightarrow^{k} \langle S_2, s'\rangle$$

Aplicando la regla [comp]

$$\frac{\langle S_1, s\rangle \Rightarrow \langle S_3, \hat{s}\rangle}{\langle S_1, S_2, s\rangle \Rightarrow \langle S_3; S_2, \hat{s}\rangle}.$$

Juntando esto $\langle S_1, S_2, s\rangle \Rightarrow \langle S_3; S_2, \hat{s}\rangle \Rightarrow^k \langle S_2, s'\rangle$, es decir,

$\langle S_1; S_2, s\rangle \Rightarrow^{k+1} \langle S_2, s'\rangle$., lo que termina la demostración del lema.

Ahora podemos aplicar este resultado a nuestro problema inicial

Como $\langle S, s\rangle \Rightarrow^k s'$ y por la regla [skip] $\langle skip, s'\rangle \Rightarrow^1 s'$

entonces, $\langle S; skip, s\rangle \Rightarrow^{k+1} s'$. Por tanto $\langle S; skip, s\rangle \Rightarrow^* \gamma$,

que es lo que queríamos probar.

2) $\Rightarrow$] Supongamos que $\langle S; skip, s\rangle$ tiene una secuencia de derivación infinita y, por reducción al absurdo que la secuencia de derivación de $\langle S, s\rangle$ no es infinita, es decir, $\exists k \geq 0$ tal que $\langle S, s\rangle \Rightarrow^k \gamma$ con $\gamma = s'$ (no puede bloquearse). Utilizando el resultado anteriormente probado $\langle S; skip, s\rangle \Rightarrow^{k+1} \gamma$, lo que contradice la infinitud de la secuencia de derivación.

$\Leftarrow$] Supongamos que $\langle S, s\rangle$ tiene una secuencia de derivación infinita y por reducción al absurdo que $\langle S; skip\rangle \Rightarrow^k s'$. Argumentando como en 1) $\Rightarrow$] se sigue que $\langle S; s\rangle \Rightarrow^{k-1} s'$ lo que contradice la infinitud de la secuencia de derivación.

b) $\overbrace{\text{while } b \text{ do } S}^{S_1}$ y if b then $(S;$ while b do $S)$ else skip.

Sea $s \in$ State y veamos que

1) $\langle S_1, s\rangle \Rightarrow^* \gamma \iff \langle \text{if } b \text{ then } S;S_1 \text{ else skip}, s\rangle \Rightarrow^* \gamma$

2) $\langle S_1, s\rangle$ es infinita $\iff \langle \text{if } b \text{ then } S;S_1 \text{ else skip}, s\rangle$ es infinita

1) $\Rightarrow]$ Como $\exists k \geq 0$ tal que $\langle \text{while } b \text{ do } S, s\rangle \Rightarrow^k \gamma$ y

$\langle \text{while } b \text{ do } S, s\rangle \Rightarrow \langle \text{if } b \text{ then } S;S_1 \text{ else skip}, s\rangle$ por la regla [while], entonces $\langle \text{if } b \text{ then } S;S_1 \text{ else skip}, s\rangle \Rightarrow^{k-1} \gamma$.

Por el determinismo

$\Downarrow$

$\langle \text{if } b \text{ then } S;S_1 \text{ else skip}, s\rangle \Rightarrow^* \gamma$.

$\Leftarrow]$ Como $\exists k \geq 0$ tal que $\langle \text{if } b \text{ do } S;S_1 \text{ else skip}, s\rangle \Rightarrow^k \gamma$ y aplicando la regla [while]

$$\langle S_1, s\rangle \Rightarrow \langle \text{if } b \text{ do } S;S_1 \text{ else skip}, s\rangle (\Rightarrow^k \gamma) \text{ entonces}$$

$$\langle S_1; s\rangle \Rightarrow^* \gamma \text{ porque } \langle S_1; s\rangle \Rightarrow^{k+1} \gamma$$

2) Se razona por reducción al absurdo análogamente al caso a).

c) $S_1;(S_2;S_3)$ y $(S_1;S_2);S_3$.

Sea $s \in$ State y veamos que

$\langle S_1;(S_2;S_3), s\rangle \Rightarrow^* \gamma \iff \langle (S_1;S_2);S_3, s\rangle \Rightarrow^* \gamma$.

$\Rightarrow]$ Supongamos que $\exists K \geq 0$ tal que $\langle S_1;(S_2;S_3), s\rangle \Rightarrow^k \gamma$.

Por el Lema 2.19, $\exists s'$ y $k_1, k_2 \geq 0$ tales que $\langle S_1, s\rangle \Rightarrow^{k_1} s'$

$\langle (S_2;S_3), s'\rangle \Rightarrow^{k_2} \gamma$

Nuevamente por el Lema 2.19. $\exists s''$ y $k_3, k_4 \geq 0$ tales que $k_1+k_2=k$

$\langle S_2, s'\rangle \Rightarrow^{k_3} s''$, $\langle S_3, s''\rangle \Rightarrow^{k_4} \gamma$ y $k_3+k_4=k_2$.

Aplicando el Lema demostrado en el apartado a), como

$<S_1, s> \Rightarrow^{k_1} s'$ y $<S_2, s'> \Rightarrow^{k_3} s'' \Rightarrow <S_1, S_2, s> \Rightarrow^{k_1+k_3} s''$

Nuevamente aplicando este lema, como

$<S_1; S_2, s> \Rightarrow^{k_1+k_3} s''$ y $<S_3, s''> \Rightarrow^{k_4} \gamma$ entonces

$<(S_1; S_2); S_3, s> \Rightarrow^{k_1+k_3+k_4} \gamma$. Utilizando que $k_3+k_4=k_2$ y que $k_1+k_2=k$

$\Rightarrow <(S_1; S_2); S_3, s> \Rightarrow^{k} \gamma$

$\boxed{\Leftarrow}$ Análogo.

Ejercicio 2.24.- Probar que $\overbrace{\text{repeat } S \text{ until } b}^{S_1}$ (definido en el ejercicio 2.17) es semánticamente equivalente a $\underbrace{S; \text{while } \neg b \text{ do } S}_{S_2}$.

Sea $s \in State$ y veamos que

$<S_1, s> \Rightarrow^{*} s' \Leftrightarrow <S_2, s> \Rightarrow^{*} s'$ (Ya sabemos que no hay config. bloq).

$\boxed{\Rightarrow}$ Supongamos que $\exists k \geq 0$ tal que $<\text{repeat } S \text{ until } b, s> \Rightarrow^{k} s'$ y hay que ver que $<S; \text{while } \neg b \text{ do } S, s> \Rightarrow^{k'} s'$ para cierto k'.

De hecho vamos a probar que $\forall k \geq 0$ si $<\text{repeat } S \text{ until } b, s> \Rightarrow^{k} s'$, entonces $<S; \text{while } \neg b \text{ do } S, s> \Rightarrow^{k} s'$

Por inducción sobre k.

Si $k=0$ el resultado se cumple trivialmente.

Sea $k \geq 0$ para el que se cumple la propiedad y se cumple $\forall n \geq 0$ y $n \leq k$ y queremos verlo para $k+1$.

Sabemos entonces que $\langle \text{repeat } S \text{ until } b, s \rangle \Rightarrow^{k+1} s'$. Por tanto, como $k \geq 0$,

$\langle \text{repeat } S \text{ until } b, s \rangle \Rightarrow \gamma$ y $\gamma \Rightarrow^{k} s'$. Por determinismo, se tiene que

$\langle \text{repeat } S \text{ until } b, s \rangle \Rightarrow \langle S; \text{if } b \text{ then skip else (repeat } S \text{ until } b), s \rangle \Rightarrow^{k} s'$

Por el Lema 2,19, se tiene que $\exists k_1, k_2, s''$ tales que

$\langle S, s \rangle \Rightarrow^{k_1} s''$, $\langle \text{if } b \text{ then skip else (repeat } S \text{ until } b), s'' \rangle \Rightarrow^{k_2} s'$ y

$k_1 + k_2 = k$.

En primer lugar, $k_2 > 0$ y las reglas que se han podido aplicar para dar el primer paso han podido ser $[\text{if}^{tt}]$ o $[\text{if}^{ff}]$.

Si ha sido $[\text{if}^{tt}]$ es porque $\mathcal{B}[\![b]\!]s'' = tt$ y, por determinismo

$\langle \text{if } b \text{ then skip else repeat } S \text{ until } b, s'' \rangle \Rightarrow \langle \text{skip}, s'' \rangle \Rightarrow^{k_2 - 1} s'$

Nuevamente, $k_2 - 1 > 0$ y la primera regla que se ha aplicado ha sido [skip] por lo que, por determinismo, $k_2 - 1 = 1$ y $s'' = s'$.

En este caso $\langle S; \text{while } \neg b \text{ do } S, s \rangle \Rightarrow^{k_1} \langle \text{while } \neg b \text{ do } S, s' \rangle$

¿Ejercicio 2.21?
$(\langle S, s \rangle \Rightarrow^{k_1} s'' = s')$

Por las regla $[\text{while}]$, $[\text{if}^{ff}]$ y [skip]

$\langle \text{while } \neg b \text{ do } S, s' \rangle \Rightarrow \langle \text{if } \neg b \text{ then } S; \text{while } \neg b \text{ do } S \text{ else skip}, s' \rangle \Rightarrow$

$\Rightarrow \langle \text{skip}, s' \rangle \Rightarrow s'$

donde se ha podido aplicar $[\text{if}^{ff}]$ porque $\mathcal{B}[\![b]\!]s'' = \mathcal{B}[\![b]\!]s = tt$

$\Updownarrow$

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

Por tanto $\langle S; \text{while } \neg b \text{ do } S, s \rangle \Rightarrow^{k_1 + 3} s'$ y teníamos que $\langle \text{repeat } S \text{ until } b, s\rangle \Rightarrow^{k+1} s'$ donde $k = k_1 + k_2$ y $k_2 = 2$,

por lo que se cumple la propiedad.

- El otro caso es que se haya aplicado $[\text{if}^{ff}]$, en cuyo caso es porque se verificaba $\mathcal{A}[\![b]\!]s'' = ff$ y, por determinismo,

$\langle \text{if } b \text{ then skip else repeat } S \text{ until } b, s''\rangle \Rightarrow \langle \text{repeat } S \text{ until } b, s''\rangle \Rightarrow^{k_2-1} s'$

Por hipótesis de inducción completa, como $k_2 - 1 \geq 0$ y $k_2 - 1 \leq k$ ...

entonces $\langle S; \text{while } \neg b \text{ do } S, s''\rangle \Rightarrow^{k_2-1} s'$

Ahora bien, se tiene que

$[\text{if}^{tt}]$ ya que $\mathcal{A}[\![\neg b]\!]s'' = tt$ por ser $\mathcal{A}[\![b]\!]s'' = ff$

$$\langle S; \text{while } \neg b \text{ do } S, s\rangle \underset{\substack{\text{¿Ejercicio 2.21?}\\ \langle S, s\rangle \Rightarrow^{k_1} s''}}{\Rightarrow^{k_1}} \langle \text{while } \neg b \text{ do } S, s''\rangle \underset{[\text{while}]}{\Rightarrow}$$

$$\Rightarrow \langle \text{if } \neg b \text{ then } S; \text{while } \neg b \text{ do } S \text{ else skip}, s''\rangle \Rightarrow$$

$$\Rightarrow \langle S; \text{while } \neg b \text{ do } S, s''\rangle \Rightarrow^{k_2-1} s'$$

Es decir $\langle S; \text{while } \neg b \text{ do } S, s\rangle \Rightarrow^{k_1+1+1+k_2-1} s'$

Como $k_1 + k_2 = k \Rightarrow k_1 + 1 + 1 + k_2 - 1 = k+1$ y se sigue el resultado

$[\Leftarrow]$ Supongamos ahora que $\exists k \geq 0$ tal que $\langle S; \text{while } \neg b \text{ do } S, s\rangle \Rightarrow^k s'$ y hay que ver que $\langle \text{repeat } S \text{ until } b, s\rangle \Rightarrow^{k'} s'$ para cierto k'.

De hecho vamos a probar que $\forall k \geq 0$ si $\langle S; \text{while } \neg b \text{ do } S, s\rangle \Rightarrow^k s'$, entonces $\langle \text{repeat } S \text{ until } b, s\rangle \Rightarrow^k s'$.

Por inducción sobre k.

Si $k=0$ el resultado se cumple trivialmente

Sea $k \geq 0$, para el que se cumple la propiedad y se cumple $\forall n \geq 0$ y $n \leq k$ y veamos que la propiedad también se cumple para $k+1$.

Sabemos entonces que $<S; \text{while} \neg b \text{ do } S, s> \Rightarrow^{k+1} s'$. Por el Lema 2.19 $\exists k_1, k_2 \geq 0$ y s'' tales que $<S, s> \Rightarrow^{k_1} s''$, $<\text{while} \neg b \text{ do } S, s''> \Rightarrow^{k_2} s'$ y $k_1 + k_2 = k+1$. Se tiene que $k_2 > 0$ y por determinismo.

$<\text{while } \neg b \text{ do } S, s''> \Rightarrow <\text{if } \neg b \text{ then } S; \text{while} \neg b \text{ do } S, \text{ else skip}, s''> \Rightarrow^{k_2 - 1} s'$

Nuevamente $k_2 - 1 > 0$ y solo se han podido aplicar las reglas $[if^{tt}]$ o $[if^{ff}]$ para el primer paso.

- Si se ha aplicado $[if^{ff}]$ entonces se cumplía $\mathcal{B}[\![\neg b]\!]s'' = ff$ y se tiene (por determinismo)

$<\text{if } \neg b \text{ then } S; \text{while} \neg b \text{ do } S, \text{ else skip}, s''> \Rightarrow <\text{skip}, s''> \Rightarrow^{k_2 - 2} s'$

Como $<\text{skip}, s''> \Rightarrow s''$ por la regla $[skip]$, por determinismo, $k_2 - 2 = 1$ y $s'' = s'$.

Con esta información

$<S, s> \Rightarrow^{k_1} s'' = s'$ y el Ejercicio 2.21

$<\text{repeat } S \text{ until } b, s> \Rightarrow <S; \text{if } b \text{ then skip else (repeat } S \text{ until } b), s> \Rightarrow^{k_1}$

$\Rightarrow^{k_1} <\text{if } b \text{ then skip else (repeat } S \text{ until } b), s'> \underset{[if^{tt}]}{\Rightarrow} <\text{skip}, s'> \underset{[skip]}{\Rightarrow} s'$

$[if^{tt}]$ ya que $\mathcal{B}[\![b]\!]s' = tt$ por ser $\mathcal{B}[\![\neg b]\!]s'' = ff$, $s'' = s'$. $[skip]$.

En resumen $<\text{repeat } S \text{ until } b, s> \Rightarrow^{k_1 + 3} s'$ y $<S; \text{while} \neg b \text{ do } S, s> \Rightarrow^{k+1} s'$ con $k+1 = k_1 + k_2$ y $k_2 = 3$. $\Rightarrow k+1 = k_1 + 3$ y se tiene lo buscado.

- Si se ha aplicado $[if^{tt}]$ entonces se cumplía $\mathcal{B}[\![\neg b]\!]s'' = tt$ y se tiene (por determinismo).

$<\text{if } \neg b \text{ then } S; \text{while} \neg b \text{ do } S \text{ else skip}, s''> \Rightarrow <S; \text{while} \neg b \text{ do } S, s''> \Rightarrow^{k_2 - 2} s'$

Por hipótesis de inducción completa, como $0 \leq k_2 - 2 \leq k$

$\Rightarrow$ $\langle \text{repeat } S \text{ until } b, s'' \rangle \Rightarrow^{k_2-2} s'$

Ejercicio 2.21 y $\langle S, s \rangle \Rightarrow^{k_1} s''$

Juntando toda esta información.

$\langle \text{repeat } S \text{ until } b, s \rangle \overset{[\text{repeat}]}{\Rightarrow} \langle S; \text{if } b \text{ then skip else (repeat } S \text{ until } b), s \rangle \Rightarrow^{k_1}$

$\langle \text{if } b \text{ then skip else (repeat } S \text{ until } b), s'' \rangle \Rightarrow$ $[\text{if}^{ff}]$ porque $\mathcal{A}[\![b]\!]s'' = ff$ ya que $\mathcal{A}[\![\neg b]\!]s'' = tt$

$\Rightarrow \langle \text{repeat } S \text{ until } b, s'' \rangle \Rightarrow^{k_2-2} s'$.

En resumen $\langle S; \text{while } \neg b \text{ do } S, s \rangle \Rightarrow^{k+1} s'$ y

$\langle \text{repeat } S \text{ until } b \rangle \Rightarrow^{1+k_1+1+k_2-2} s'$

Pero como $k_1 + k_2 = k+1$ se tiene lo buscado.

Para concluir justificamos que una cicla si y solo si cicla la otra.
Si una de ellas ciclara y la otra no, como la segunda de ellas no cicla entonces termina, y aplicando lo demostrado anteriormente, la primera debería terminar. Contradicción.

Ejercicio 2.25.- Determinar si la equivalencia semántica de S_1 y S_2 implica $S_{sos}[\![S_1]\!] = S_{sos}[\![S_2]\!]$

Sea $s \in State$.

Si $\langle S_1, s \rangle$ termina en un estado s', por la equivalencia semántica $\langle S_2, s \rangle \Rightarrow^* s'$, luego $S_{sos}[\![S_1]\!]s = S_{sos}[\![S_2]\!]s$. Si $\langle S_1, s \rangle$ termina en una configuración bloqueada $\langle S_1', s_1' \rangle$, por la equivalencia semántica $\langle S_2', s \rangle \Rightarrow^* \langle S_1', s_1' \rangle$ y

$S_{sos}[\![S_1]\!]s$ "=" "undefined" "=" "undefined" "=" $S_{sos}[\![S_2]\!]$.

Por último, si $\langle S_1, s\rangle$ cicla entonces, por la equivalencia semántica, $\langle S_2, s\rangle$ cicla y $\mathcal{S}_{sos}[\![S_1]\!]s =$ "undefined" $= \mathcal{S}_{sos}[\![S_2]\!]s$

Ejercicio 2.29.- Considera la extensión del lenguaje While con repeat S until b (Ejercicios 2.7 y 2.17). Amplía la demostración del Teorema 2.26 para esta extensión.

Recordamos que las definiciones de las reglas eran:

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

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

$[\text{repeat}_{sos}]$

$$\langle \text{repeat } S \text{ until } b, s\rangle \Rightarrow \langle S; \text{if } b \text{ then skip else } (\text{repeat } S \text{ until } b), s\rangle$$

Ampliación del Lema 2.27

Para el caso $[\text{repeat}^{tt}_{ns}]$, supongamos que

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

Entonces $\langle \text{repeat } S \text{ until } b, s\rangle \overset{[\text{repeat}_{sos}]}{\Rightarrow} \langle S; \text{if } b \text{ then skip else } (\text{repeat } S \text{ until } b, s\rangle$

Como $\langle S, s\rangle \to s'$, por hipótesis de inducción $\langle S, s\rangle \Rightarrow^* s'$.

Aplicando el Ejercicio 2.21,

$$\langle S; \text{if } b \text{ then skip else } (\text{repeat } S \text{ until } b, s\rangle \Rightarrow^* \langle \text{if } b \text{ then skip else repeat } S \text{ until } b, s'\rangle.$$

Como $\mathcal{B}[\![b]\!]s' = tt$, por la regla $[\text{if}^{tt}_{sos}]$,

$$\langle \text{if } b \text{ then skip else } (\text{repeat } S \text{ until } b), s'\rangle \Rightarrow \langle \text{skip}, s'\rangle \overset{[\text{skip}]}{\Rightarrow} s'$$

Por tanto $\langle \text{repeat } S \text{ until } b, s\rangle \Rightarrow^* s'$

Para el caso $[\text{repeat}^{ff}_{ns}]$, si pongamos que

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

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

Por hipótesis de inducción $\langle S, s\rangle \Rightarrow^* s'$ y $\langle \text{repeat } S \text{ until } b, s'\rangle \Rightarrow^* s''$

(Se puede aplicar porque estos son subárboles de la expresión original).

Entonces $\langle \text{repeat } S \text{ until } b, s\rangle \overset{[\text{repeat}_{sos}]}{\Rightarrow} \langle S; \text{if } b \text{ then skip else (repeat } S \text{ until } b), s\rangle$

y por el Ejercicio 2.21, como $\langle S, s\rangle \Rightarrow^* s'$, entonces

$\langle S; \text{if } b \text{ then skip else (repeat } S \text{ until } b), s\rangle \Rightarrow^*$

$\langle \text{if } b \text{ then skip else (repeat } S \text{ until } b), s'\rangle$

Como $\mathcal{B}[\![b]\!]s' = ff$, aplicando $[\text{if}^{ff}_{sos}]$ se tiene

$\langle \text{if } b \text{ then skip else (repeat } S \text{ until } b), s'\rangle \Rightarrow^* \langle \text{repeat } S \text{ until } b, s'\rangle \Rightarrow^*$

$\Rightarrow^* s''$, luego $\langle \text{repeat } S \text{ until } b, s\rangle \Rightarrow^* s''$.

Esto finaliza la ampliación del Lema 2.27.

Ampliación del Lema 2.28.

Para el caso $[\text{repeat}_{sos}]$ se tiene que

$\langle \text{repeat } S \text{ until } b, s\rangle \Rightarrow \langle S; \text{if } b \text{ then skip else (repeat } S \text{ until } b), s\rangle \Rightarrow^{k_0} s''$

Por hipótesis de inducción $\langle S; \text{if } b \text{ then skip else (repeat } S \text{ until } b), s\rangle \to s''$

Ya vimos en el ejercicio 2.7 que con esta semántica, repeat S until b es sintácticamente equivalente a S; if b then skip else (repeat S until b)

luego $\langle \text{repeat } S \text{ until } b, s\rangle \to s''$, lo que termina la ampliación del Lema 2.28.