Lema 1
a) Si $(c_0,s) \to^* s'$ entonces $(c_0;c_1,s) \to^* (c_1,s')$
b) Si $(c , [s|v:[|e|]s]) \to^* s'$ entonces $(\text{newvar } v:=e \text{ in } c , s) \to^* [s'|v:sv]$
c) Si $(c , [s|v:[|e|]s]) \to^* (c',s')$ entonces

$$(\text{newvar } v:=e \text{ in } c , s) \to^* (\text{newvar } v:=s'v \text{ in } c' , [s'|v:sv])$$

Demostración: a) Supongamos $G0 = (c_0,s)$, y que la ejecución $G0 \to^* s'$ tiene n pasos:

$$G0 \to G1 \to \dots \to Gn = s'$$

Hacemos inducción en n.

<u>Caso n=1</u>: Se tiene que $(c_0,s) \to s'$, entonces la tesis surge inmediatamente de la regla

$$\frac{(c_0,s) \to s'}{(c_0;c_1,s) \to (c_1,s')}$$

<u>Caso recursivo</u>: Suponemos ahora que a) es válido para ejecuciones de tamaño menor que n. Supongamos que tenemos la ejecución $G0 \to G1 \to \dots \to Gn = s'$ (de n pasos). Sea $G1 = (c_0^1,s^1)$. Dado que $G1 \to^* s'$, y tal ejecución tiene n-1 pasos, por HI se tiene $(c_0^1;c_1,s^1) \to^* (c^1,s')$. Luego la conclusión se obtiene desde la segunda regla para el (;):

$$\frac{(c_0,s) \to (c_0^1,s^1)}{(c_0;c_1,s) \to (c_0^1;c_1,s^1) \to^* (c_1,s')}$$

b) Nuevamente por inducción en la longitud de la derivación de $(c , [s|v:[|e|]s]) \to^* s'$. Si la longitud es 1, al igual que el caso a), la conclusión surge de la primera regla del newvar:

$$\frac{(c , [s|v:[|e|]s]) \to s'}{(\text{newvar } v:=e \text{ in } c , s) \to [s'|v:sv]}$$

Supongamos que la longitud de la de derivación de $(c , [s|v:[|e|]s]) \to^* s'$ es mayor que 1, y que $(c , [s|v:[|e|]s]) \to (c^1,s^1)$. Por la segunda regla del newvar se tiene que:

$$(\text{newvar } v:=e \text{ in } c , s) \to (\text{newvar } v:=s^1v \text{ in } c^1 , [s^1|v:sv])$$

Queremos aplicar la hipótesis inductiva para deducir que

$$(c^1,s^1) \to^* s' \quad \text{implica} \quad (\text{newvar } v:=s^1v \text{ in } c^1 , [s^1|v:sv]) \to [s'|v:sv]$$

Para esto debemos verificar que si $s_0 = [s^1|v:sv]$, entonces $s^1 = [s_0|v:s^1v]$. Esto se prueba a continuación:

Si $w = v$ entonces $s^1\ v = [s_0|v:s^1v]\ v$
Si w /= v entonces $s^1\ v = [s^1|v:sv]\ v = s_0\ v = [s_0|v:s^1v]\ v$

Lema 2

(1) (c,s) -> s' implica [| c |]s = s'
(2) (c,s) -> (c',s') implica [| c |]s = [| c' |]s'

Demostración: Tanto (1) como (2) se prueban recurriendo a una inducción sobre la derivación de la relación "->". Para esto debemos recurrir a verificar la tesis para cada una de las reglas cuya conclusión tiene la forma (c,s) -> s' (para la prueba de (1)), y para cada una de las reglas cuya conclusión tiene la forma (c,s) -> (c',s') (para la prueba de (2)). Algunos casos no triviales como ejemplo.

(1) <u>Caso</u>: ------------------------------ (¬[| b |])
(while b do c , s) -> s

Surge inmediatamente de la propiedad: si ¬[| b |] entonces [| while b do c |]s = s. Para probar esto basta observar que $F^1\bot\ s = s$, siempre que ¬[| b |]. Como el dominio $\Sigma_\bot$ es llano, entonces [| while b do c |]s = $F^1\bot\ s = s$.

(2) <u>Caso</u>: (c , [s|v:[|e|]s]) -> (c^1,s^1)

(newvar v:=e in c , s) -> (newvar v:=s^1v in c^1 , [s^1|v:sv])

[| newvar v:=s^1v in c^1 |][s^1|v:sv] = (\s'.[s'|v:[s^1|v:sv]v]) $_\bot$ ([|c^1|][[s^1|v:sv]|v:s^1v])

= (\s'.[s'|v:sv]) $_\bot$ ([|c^1|]s^1)

= {por hipótesis inductiva}
(\s'.[s'|v:sv]) $_\bot$ ([|c|][s|v:[|e|]s])

= [| newvar v:=e in c |]s

Lema 3 [| c |]s = s' implica (c,s) -> s'

Demostración: Se utiliza inducción en la estructura de c. Probaremos algunos casos no triviales.

<u>Caso</u> $c = c_0;c_1$: Por la hipótesis [| $c_0;c_1$ |]s = s', podemos suponer que [| c_0 |]s = s_0, y que [| c_1 |]s_0 = s'. Por hipótesis inductivas se tiene (c_0,s) ->* s_0, y que (c_1,s_0) ->* s'. Por lema 1, dado que (c_0,s) ->* s_0, tenemos que ($c_0;c_1$,s) ->* ($c_1 s_0$). Luego, por transitividad de ->*, obtenemos ($c_0;c_1$,s) ->* s'.

Teorema Corrección de la semántica operacional respecto de la denotacional.

Si definimos:

$$\{| c |\}s = \perp \quad \text{si} \quad (c,s)\uparrow$$
$$\{| c |\}s = s' \quad \text{si} \quad \text{existe } s' \text{ tal que } (c,s) \to^* s'$$

Entonces {| c |} = [| c |].

Demostración: Sea s tal que [| c |]s es un estado. Entonces {| c |}s = [| c |]s por lema 3. Supongamos ahora que [| c |]s = $\perp$. Entonce demostramos por el absurdo que $(c,s)\uparrow$. Supongamos lo contrario, entonces existe s' tal que (c,s) ->* s'. Pero en este caso, aplicando sucesivamente el lema 2 a la ejecución (c,s) ->* s' tendríamos [| c |]s = s'.