TEMA-5-APUNTES.pdf

DEYORS

Teoria de la Programacion

4º Grado en Matemáticas

Facultad de Ciencias Matemáticas
Universidad Complutense de Madrid

TPRO

21.04.2021

TEMA 5 - SEMÁNTICA DENOTACIONAL

Estamos interesados en el efecto de la ejecución del programa, es decir, la asociación entre el estado inicial y final.

La idea será definir una FUNCIÓN SEMÁNTICA para cada CATEGORÍA SINTÁCTICA.

Las funciones estarán definidas COMPOSITIVAMENTE:

- **Existe una cláusula semántica para cada elemento básico.**
- **Para cada método de construir un elemento compuesto existe una cláusula semántica definida en términos de elementos más pequeños.**

5.1 SEMÁNTICAS DE ESTILO DIRECTO: ESPECIFICACIONES

El efecto de ejecutar S es cambiar el estado, osea que necesitamos definir el SIGNIFICADO DE S:

$$\mathcal{S}_{\mathrm{ds}}\colon \mathbf{Stm} \to (\mathbf{State} \hookrightarrow \mathbf{State})$$

$$\begin{aligned}
\mathcal{S}_{\mathrm{ds}}[\![x := a]\!]s &= s[x \mapsto \mathcal{A}[\![a]\!]s] \\
\mathcal{S}_{\mathrm{ds}}[\![\mathtt{skip}]\!] &= \mathrm{id} \\
\mathcal{S}_{\mathrm{ds}}[\![S_1 \ ; \ S_2]\!] &= \mathcal{S}_{\mathrm{ds}}[\![S_2]\!] \circ \mathcal{S}_{\mathrm{ds}}[\![S_1]\!] \\
\mathcal{S}_{\mathrm{ds}}[\![\mathtt{if}\ b\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2]\!] &= \mathsf{cond}(\mathcal{B}[\![b]\!], \mathcal{S}_{\mathrm{ds}}[\![S_1]\!], \mathcal{S}_{\mathrm{ds}}[\![S_2]\!]) \\
\mathcal{S}_{\mathrm{ds}}[\![\mathtt{while}\ b\ \mathtt{do}\ S]\!] &= \mathrm{FIX}\ F \\
\text{where } F\ g &= \mathsf{cond}(\mathcal{B}[\![b]\!], g \circ \mathcal{S}_{\mathrm{ds}}[\![S]\!], \mathrm{id})
\end{aligned}$$

$$\begin{array}{c}\mathcal{S}_{\mathrm{ds}}[\![S_1 \ ; \ S_2]\!]s \\ = \\ (\mathcal{S}_{\mathrm{ds}}[\![S_2]\!] \circ \mathcal{S}_{\mathrm{ds}}[\![S_1]\!])s\end{array} = \begin{cases} s'' & \text{if there exists } s' \text{ such that } \mathcal{S}_{\mathrm{ds}}[\![S_1]\!]s = s' \\ & \text{and } \mathcal{S}_{\mathrm{ds}}[\![S_2]\!]s' = s'' \\ \underline{\mathrm{undef}} & \text{if } \mathcal{S}_{\mathrm{ds}}[\![S_1]\!]s = \underline{\mathrm{undef}} \\ & \text{or if there exists } s' \text{ such that } \mathcal{S}_{\mathrm{ds}}[\![S_1]\!]s = s' \\ & \text{but } \mathcal{S}_{\mathrm{ds}}[\![S_2]\!]s' = \underline{\mathrm{undef}} \end{cases}$$

$$\mathsf{cond}\colon (\mathbf{State} \to \mathbf{T}) \times (\mathbf{State} \hookrightarrow \mathbf{State}) \times (\mathbf{State} \hookrightarrow \mathbf{State}) \to (\mathbf{State} \hookrightarrow \mathbf{State})$$

$$\mathsf{cond}(p, g_1, g_2)\ s = \begin{cases} g_1\ s & \text{if } p\ s = \mathbf{tt} \\ g_2\ s & \text{if } p\ s = \mathbf{ff} \end{cases}$$

$$\begin{array}{c}\mathcal{S}_{\mathrm{ds}}[\![\mathtt{if}\ b\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2]\!]\ s \\ = \\ \mathsf{cond}(\mathcal{B}[\![b]\!], \mathcal{S}_{\mathrm{ds}}[\![S_1]\!], \mathcal{S}_{\mathrm{ds}}[\![S_2]\!])\ s\end{array} = \begin{cases} s' & \text{if } \mathcal{B}[\![b]\!]s = \mathbf{tt} \text{ and } \mathcal{S}_{\mathrm{ds}}[\![S_1]\!]s = s' \\ & \text{or if } \mathcal{B}[\![b]\!]s = \mathbf{ff} \text{ and } \mathcal{S}_{\mathrm{ds}}[\![S_2]\!]s = s' \\ \underline{\mathrm{undef}} & \text{if } \mathcal{B}[\![b]\!]s = \mathbf{tt} \text{ and } \mathcal{S}_{\mathrm{ds}}[\![S_1]\!]s = \underline{\mathrm{undef}} \\ & \text{or if } \mathcal{B}[\![b]\!]s = \mathbf{ff} \text{ and } \mathcal{S}_{\mathrm{ds}}[\![S_2]\!]s = \underline{\mathrm{undef}} \end{cases}$$

Para $while\ b\ do\ S$ **la tarea es más complicada, pero tenemos la ventaja de que es igual a** $if\ b\ then\ (S\ ; while\ b\ do\ S)\ else\ skip$**, luego:**

$$\mathcal{S}_{\rm ds}[\![\texttt{while}\ b\ \texttt{do}\ S]\!] = \textsf{cond}(\mathcal{B}[\![b]\!],\ \mathcal{S}_{\rm ds}[\![\texttt{while}\ b\ \texttt{do}\ S]\!] \circ \mathcal{S}_{\rm ds}[\![S]\!],\ \text{id}) \quad (*)$$

Pero no podríamos usar esta definición porque no sería COMPOSICIONAL (aparece $S_{ds}[[while\ b\ do\ S]]$**), pero sí nos dice que** $S_{ds}[[while\ b\ do\ S]]$ **debería ser UN PUNTO FIJO de una función F definida como:**

$$\boxed{F\ g = \textsf{cond}(\mathcal{B}[\![b]\!],\ g \circ \mathcal{S}_{\rm ds}[\![S]\!],\ \text{id})}$$

$$\mathcal{S}_{\rm ds}[\![\texttt{while}\ b\ \texttt{do}\ S]\!] = F\ (\mathcal{S}_{\rm ds}[\![\texttt{while}\ b\ \texttt{do}\ S]\!])$$

Es decir:

$$\boxed{\begin{array}{c}\mathcal{S}_{\rm ds}[\![\texttt{while}\ b\ \texttt{do}\ S]\!] = \textsf{FIX}\ F \\ \text{where } F\ g = \textsf{cond}(\mathcal{B}[\![b]\!],\ g \circ \mathcal{S}_{\rm ds}[\![S]\!],\ \text{id}) \\ \textsf{FIX}\text{: } ((\textbf{State} \hookrightarrow \textbf{State}) \to (\textbf{State} \hookrightarrow \textbf{State})) \to (\textbf{State} \hookrightarrow \textbf{State})\end{array}}$$

Los problemas que pueden ocurrir son que:

- **Hay funcionales que tienen más de un punto fijo.**
- **Hay funcionales que no tienen puntos fijos.**

REQUERIMIENTOS EN EL PUNTO FIJO:

Hay tres tipos de salidas para un algoritmo $while\ b\ do\ S$ **:**

A: Termina

B: Loopea LOCALMENTE (existe un constructo en S que loopea)

C: Loopea GLOBALMENTE (el mismo *while* **loopea)**

Caso A: Existen estados $s_1, \dots, s_n$ tales que:

$$\mathcal{B}[\![b]\!]\ s_i = \begin{cases} tt & \text{if } i<n \\ ff & \text{if } i=n \end{cases} \quad \text{and} \quad \mathcal{S}_{ds}[\![S]\!]\ s_i = s_{i+1} \quad \text{for } i<n$$

Sea g_0 cualquier punto fijo de F ($F\ g_0 = g_0$):

Si $i \leq n$: $g_0\ s_i = (F\ g_0)\ s_i = \text{cond}(\mathcal{B}[\![b]\!], g_0 \circ \mathcal{S}_{ds}[\![S]\!], \text{id})\ s_i = g_0\ (\mathcal{S}_{ds}[\![S]\!]\ s_i) = g_0\ s_{i+1}$

Si $i = n$: $g_0\ s_n = (F\ g_0)\ s_n = \text{cond}(\mathcal{B}[\![b]\!], g_0 \circ \mathcal{S}_{ds}[\![S]\!], \text{id})\ s_n = \text{id}\ s_n = s_n$

Luego CUALQUIER PUNTO FIJO g_0 cumplirá: $g_0\ s_0 = s_n$ y NO HABRÁ UN PUNTO FIJO PREFERIDO.

Caso B: Existen estados $s_1, \dots, s_n$ tales que:

$$\mathcal{B}[\![b]\!]s_i = tt \text{ for } i\leq n \quad \text{and} \quad \mathcal{S}_{ds}[\![S]\!]s_i = \begin{cases} s_{i+1} & \text{for } i<n \\ \underline{\text{undef}} & \text{for } i=n \end{cases}$$

Sea g_0 cualquier punto fijo de F ($F\ g_0 = g_0$):

Si $i \leq n$: $g_0\ s_i = (F\ g_0)\ s_i = \text{cond}(\mathcal{B}[\![b]\!], g_0 \circ \mathcal{S}_{ds}[\![S]\!], \text{id})\ s_i = g_0\ (\mathcal{S}_{ds}[\![S]\!]\ s_i) = g_0\ s_{i+1}$

Si $i = n$: $g_0\ s_n = (F\ g_0)\ s_n = \text{cond}(\mathcal{B}[\![b]\!], g_0 \circ \mathcal{S}_{ds}[\![S]\!], \text{id})\ s_n = (g_0 \circ \mathcal{S}_{ds}[\![S]\!])\ s_n = \underline{\text{undef}}$

Luego CUALQUIER PUNTO FIJO g_0 cumplirá: $g_0\ s_0 = undef$ y NO HABRÁ UN PUNTO FIJO PREFERIDO.

Caso C: Existen INFINITOS estados $s_1, \dots$ donde:

$$\mathcal{B}[\![b]\!]s_i = tt \text{ for all i} \quad \text{and} \quad \mathcal{S}_{ds}[\![S]\!]s_i = s_{i+1} \text{ for all i.}$$

Sea g_0 cualquier punto fijo de F ($F\ g_0 = g_0$):

Para TODO $i \geq 0$: $g_0\ s_i = (F\ g_0)\ s_i = \text{cond}(\mathcal{B}[\![b]\!], g_0 \circ \mathcal{S}_{ds}[\![S]\!], \text{id})\ s_i = g_0\ (\mathcal{S}_{ds}[\![S]\!]\ s_i) = g_0\ s_{i+1}$

luego $g_0\ s_0 = s_i$ PARA TODO i y no podemos determinar el valor de $g_0\ s_0$ de esta manera.

Pero de acuerdo a la EXPERIENCIA COMPUTACIONAL, es preferible un PUNTO FIJO que CONTENGA A LOS OTROS. Necesitaremos que:

- **g_0 sea un punto fijo de F ($F\ g_0 = g_0$)**
- **Si g es otro punto fijo de F ($F\ g = g$) entonces $g_0\ s = s'$ implica que $g\ s = s'$**

Si $g_0\ s = undef$ entonces no hay requerimientos para $g\ s$

5.2 TEORÍA DE PUNTOS FIJOS:

Primero es necesario REFORMULAR los REQUERIMIENTOS PARA FIX F.
Para eso se define:

ORDERING: $\sqsubseteq$

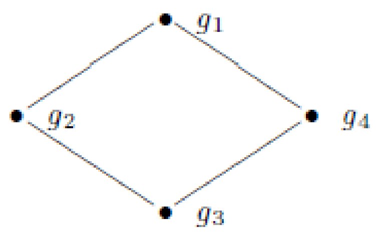
$$g_1 : \mathbf{State} \hookrightarrow \mathbf{State}$$
$$g_2 : \mathbf{State} \hookrightarrow \mathbf{State}$$

$$g_1 \sqsubseteq g_2 \qquad \boxed{\begin{array}{c} \text{if} \\ g_1\ s = s' \\ \text{then} \\ g_2\ s = s' \end{array}} \qquad \text{for all } s \text{ and } s'$$

Ejemplo:

(Diagrama de Hesse)

$$g_1 \sqsubseteq g_1, \quad g_2 \sqsubseteq g_1, \quad g_3 \sqsubseteq g_1, \quad g_4 \sqsubseteq g_1,$$
$$g_2 \sqsubseteq g_2, \quad g_3 \sqsubseteq g_2, \quad g_4 \sqsubseteq g_4.$$
$$g_3 \sqsubseteq g_3,$$
$$g_3 \sqsubseteq g_4,$$



La idea es que los elementos "más pequeños" son los que están más abajo. No es necesario dibujar líneas donde ya se puede acceder por otro camino (por ejemplo $g_3 \sqsubseteq g_1$ pero no es necesario representarlo.

Todo el conjunto $State \hookrightarrow State$ **con** $\sqsubseteq$ **se llama CONJUNTO PARCIALMENTE ORDENADO, y sus relaciones son:**

$(D, \sqsubseteq_D)$ *partially ordered set* D set $\sqsubseteq_D$ relation on D	(reflexivity)	$d \sqsubseteq_D d$		
	(transitivity)	$d_1 \sqsubseteq_D d_2$ and $d_2 \sqsubseteq_D d_3$	imply	$d_1 \sqsubseteq_D d_3$
	(anti-symmetry)	$d_1 \sqsubseteq_D d_2$ and $d_2 \sqsubseteq_D d_1$	imply	$d_1 = d_2$

OBSERVACIÓN: Si un conjunto parcialmente ordenado tiene un ELEMENTO MÍNIMO d, este es ÚNICO.

Lema 5.13:

$(\mathbf{State} \hookrightarrow \mathbf{State}, \ \sqsubseteq)$ partially ordered set

$\perp: \mathbf{State} \hookrightarrow \mathbf{State}$

$s \rightarrow \perp s = \underline{\text{undef}}$ for all s

$\Rightarrow$

$\perp$ is the least element of $\mathbf{State} \hookrightarrow \mathbf{State}$

Demostración:

$\sqsubseteq$ fulfils the three requirements :

reflexive

$g\ s = s'$ trivially implies that $g\ s = s'$ $\Rightarrow$ $g \sqsubseteq g$ holds

transitive

assume that $g_1 \sqsubseteq g_2$ and $g_2 \sqsubseteq g_3$

assume that $g_1\ s = s'$

From $g_1 \sqsubseteq g_2$, we get $g_2\ s = s'$ and then $g_2 \sqsubseteq g_3$ gives $g_3\ s = s'$.

anti-symmetric

assume that $g_1 \sqsubseteq g_2$ and $g_2 \sqsubseteq g_1$

assume that $g_1\ s = s'$, then we get $g_2\ s = s'$

then $g_1 = g_2$

$\perp$ is the *least element* of $\mathbf{State} \hookrightarrow \mathbf{State}$:

$\perp$ is indeed an element of $\mathbf{State} \hookrightarrow \mathbf{State}$.

it is also obvious that $\perp \sqsubseteq g$ holds for all g

since $\perp s = s'$ vacuously implies that $g\ s = s'$.

$\square$

Con esto ya se pueden definir formalmente los REQUERIMIENTOS PARA EL PUNTO FIJO:

- $FIX\ F$ **debe ser un PUNTO FIJO** ---> $F(\ FIX\ F\ ) = FIX\ F$
- $FIX\ F$ **debe ser EL MINIMO ELEMENTO DE** $(F, \sqsubseteq)$ ---> $si\ \ F\ g = g\ \ entonces\ FIX\ F\ \sqsubseteq g$

El siguiente objetivo es asegurarse de que TODOS LOS FUNCIONALES F TIENEN PUNTOS FIJOS MÍNIMOS.

CONJUNTOS PARCIALMENTE ORDENADOS COMPLTOS:

Dado un CPO $(D, \sqsubseteq)$ con $Y \subset D$, es posible encontrar un ELEMENTO $d \in D$ que RESUME TODA LA INFORMACIÓN DE Y, llamado LÍMITE SUPERIOR DE Y (upper bound).

Un LÍMITE SUPERIOR es el MÍNIMO LÍMITE SUPERIOR (least upper bound ó LUB) si es el menor (con el ordering) de todos.

$(D, \sqsubseteq)$ partially ordered set Y subset of D d element of D	$\Rightarrow$	d *upper bound* of Y if $\forall d' \in Y\colon d' \sqsubseteq d$
d upper bound of Y	$\Rightarrow$	d *least upper bound* of Y if d' is an upper bound of Y implies that $d \sqsubseteq d'$

Si Y tiene un LUB, entonces es ÚNICO y SE DENOTA COMO $\sqcup Y$.

Y es una CADENA (chain) si para todos los elementos cogidos de dos en dos, uno comparte información con el otro y viceversa:

Y is *a chain*	$\Rightarrow$	$\forall d_1, d_2 \in Y\colon$ $d_1 \sqsubseteq d_2$ or $d_2 \sqsubseteq d_1$

Un CPO $(D, \sqsubseteq)$ se llama CONJUNTO PARCIALMENTE ORDENADO DE CADENA COMPLETA (chain complete partially ordered set ó CCPO) si TODAS SUS CADENAS $Y \subset D$ TIENEN UN LUB $\sqcup Y$.

Un CPO $(D, \sqsubseteq)$ se llama CELOSÍA COMPLETA xd (ó complete lattice) si PARA TODO SUBCONJUNTO $Y \subset D$ existe LUB.

OBSERVACIÓN 5.24: Un CCPO siempre tiene LUB y es $\perp = \sqcup \phi$.

EJERCICIO 2.21: Nuestro conjunto $State \hookrightarrow State$ NO ES COMPLETE LATTICE.

Lema 5.25:

(**State** $\hookrightarrow$ **State**, $\sqsubseteq$) cpo Y chain of (**State** $\hookrightarrow$ **State**, $\sqsubseteq$) $\Rightarrow$	**1)** (**State** $\hookrightarrow$ **State**, $\sqsubseteq$) is a ccpo **2)** The least upper bound $\bigsqcup Y$ of Y is $\text{graph}(\bigsqcup Y) = \bigcup\{\ \text{graph}(g) \mid g \in Y\ \}$ that is $(\bigsqcup Y)s = s'$ if and only if for some $g \in Y$: $g\ s = s'$

Demostración:

three parts

First : $\bigcup\{\ \text{graph}(g) \mid g \in Y\ \}$ is a graph of a partial function in **State** $\hookrightarrow$ **State** ?

we only need to show that if $\langle s, s'\rangle$ and $\langle s, s''\rangle$ are elements of $X = \bigcup\{\ \text{graph}(g) \mid g \in Y\ \}$ then $s' = s''$

When $\langle s, s'\rangle \in X$, there will be a partial function $g \in Y$ such that $g\ s = s'$

when $\langle s, s''\rangle \in X$, there will be a partial function $g' \in Y$ such that $g'\ s = s'$

Y is a chain $\Rightarrow$ either $g \sqsubseteq g'$ or $g' \sqsubseteq g$ $\Rightarrow$ In any case, we get $g\ s = g'\ s$ $\Rightarrow$ $s' = s''$

Second : will this function be an upper bound of Y ?

we define the partial function g_0 by $\text{graph}(g_0) = \bigcup\{\ \text{graph}(g) \mid g \in Y\ \}$

let g be an element of Y $\Rightarrow$ $\text{graph}(g) \subseteq \text{graph}(g_0)$ $\xRightarrow[\text{Ex. 5.8}]{}$ $g \sqsubseteq g_0$

Third : is it less than any other upper bound of Y ?

let g_1 be some upper bound of Y

Using the definition of an upper bound : $g \sqsubseteq g_1$ must hold for all $g \in Y$

Exercise 5.8 gives that $\text{graph}(g) \subseteq \text{graph}(g_1)$ $\Rightarrow$ it must be $\bigcup\{\ \text{graph}(g) \mid g \in Y\ \} \subseteq \text{graph}(g_1)$

But this is the same as $\text{graph}(g_0) \subseteq \text{graph}(g_1)$

Exercise 5.8 gives that $g_0 \sqsubseteq g_1$ $\Rightarrow$ g_0 is the least upper bound of Y

FUNCIONES CONTINUAS:

Sean $(D, \sqsubseteq)$ y $(D', \sqsubseteq')$ dos CCPO y $f : D \rightarrow D'$ aplicación total. Si $d_1 \sqsubseteq d_2$ entonces d_1 comparte su información con d_2, así que cuando f es aplicado a los dos elementos esperamos que LAS IMAGENES TENGAN COMPORTAMIENTO SIMILAR, es decir, $f\ d_1 \sqsubseteq' f\ d_2$.

Una función es MONÓTONA si:

for all
d_1 and d_2 :

$d_1 \sqsubseteq d_2$ implies $f\ d_1 \sqsubseteq' f\ d_2$

OBSERVACIÓN 5.29:

$(D, \sqsubseteq), (D', \sqsubseteq')$, and $(D'', \sqsubseteq'')$	ccpo	Then	$f' \circ f \colon D \rightarrow D''$ is a monotone function.
$f\colon D \rightarrow D'$ $f'\colon D' \rightarrow D''$	monotone functions		

Demostración:

Assume $d_1 \sqsubseteq d_2$
monotonicity of f gives $f\ d_1 \sqsubseteq' f\ d_2$
monotonicity of f' then gives $f'\ (f\ d_1) \sqsubseteq'' f'\ (f\ d_2)$ $\square$

LEMA 5.30:

$(D, \sqsubseteq), (D', \sqsubseteq')$ ccpo $f\colon D \rightarrow D'$ monotone function Y chain in D	then	1)	$\{ f\ d \mid d \in Y \}$ is a chain in D'
		2)	$\bigsqcup'\{ f\ d \mid d \in Y \} \sqsubseteq' f(\bigsqcup Y)$

Demostración:

$Y = \emptyset$ $\Longrightarrow$ $\bot' \sqsubseteq' f\ \bot$ $\Longrightarrow$ the result holds immediately

$Y \neq \emptyset$:

$\{ f\ d \mid d \in Y \}$ is a chain in D' ?

let d'_1 and d'_2 be two elements of $\{ f\ d \mid d \in Y \}$
there are elements d_1 and d_2 in Y such that
$d'_1 = f\ d_1$ and $d'_2 = f\ d_2$.

Y is a chain $\Longrightarrow$ either $d_1 \sqsubseteq d_2$ or $d_2 \sqsubseteq d_1$
In either case, we get that the same order holds between d'_1 and d'_2 because of the monotonicity of f.
$\{ f\ d \mid d \in Y \}$ is a chain.

$Y \neq \emptyset$: second part :
consider an arbitrary element d of Y
it is the case that $d \sqsubseteq \bigsqcup Y$.
The monotonicity of f gives that $f\ d \sqsubseteq' f(\bigsqcup Y)$
Since this holds for all $d \in Y$,
we get that $f(\bigsqcup Y)$ is an upper bound on $\{ f\ d \mid d \in Y \}$
that is, $\bigsqcup' \{ f\ d \mid d \in Y \} \sqsubseteq' f(\bigsqcup Y)$.

$\square$

Son interesantes las funciones QUE PRESERVEN los LUB de las CADENAS, es decir, funciones f que satisfagan:

$$\boxed{\bigsqcup'\{\, f\ d \mid d \in Y \,\} = f(\bigsqcup Y)}$$

Esto es que podamos obtener la MISMA INFORMACIÓN sin tener que determinar el LUB obligatoriamente antes o después de ejecutar la función.

FUNCIÓN CONTINUA:

$(D, \sqsubseteq)$ and $(D', \sqsubseteq')$ **ccpo**

$f\colon D \to D'$

f monotone

$\bigsqcup'\{\, f\ d \mid d \in Y \,\} = f(\bigsqcup Y)$

for all *non-empty* chains Y

$\Longrightarrow$ f *continuous*

FUNCIÓN ESTRICTA:

If $\bigsqcup\{\, f\ d \mid d \in Y \,\} = f(\bigsqcup Y)$ holds for the empty chain (that is, $\perp = f\ \perp$), $\Longrightarrow$ f is *strict.*

LEMA 5.35:

$(D, \sqsubseteq)$, $(D', \sqsubseteq')$ and $(D'', \sqsubseteq'')$ ccpo

$f\colon D \to D'$
$f'\colon D' \to D''$ continuous

$\Longrightarrow$ $f' \circ f\colon D \to D''$ is continuous

Demostración:

From Fact 5.29 we get that $f' \circ f$ is monotone

let Y be a non-empty chain in D

continuity of f gives $\bigsqcup'\{\, f\ d \mid d \in Y \,\} = f\ (\bigsqcup Y)$

$\{\, f\ d \mid d \in Y \,\}$ is a (non-empty) chain in D'

use the continuity of f' $\Longrightarrow$ $\bigsqcup''\{\, f'\ d' \mid d' \in \{\, f\ d \mid d \in Y \,\} \,\} = f'\ (\bigsqcup'\{\, f\ d \mid d \in Y \,\})$

equivalent to

$$\bigsqcup''\{\, f'\ (f\ d) \mid d \in Y \,\} = f'\ (f\ (\bigsqcup Y))$$

$\square$

TEOREMA 5.37: (MÍNIMO PUNTO FIJO DE f)

$(D, \sqsubseteq)$ ccpo $f: D \to D$ continuous $\perp$ least element	$\Rightarrow$	$\mathsf{FIX}\ f = \bigsqcup\{\ f^n \perp \mid n \geq 0\ \}$ defines an element of D and this element is the least fixed point of f

$$(\quad f^0 = id \qquad y \qquad f^{n+1} = f \circ f^n \quad)$$

Demostración:

well-definedness of FIX f :

$f^0 \perp = \perp$ and $\perp \sqsubseteq d$ for all $d \in D$

induction on n
f is monotone $\Rightarrow$ $f^n \perp \sqsubseteq f^n\ d$ for all $d \in D$ $\Rightarrow$ $f^n \perp \sqsubseteq f^m \perp$ whenever n≤m

$\Rightarrow$ $\{\ f^n \perp \mid n \geq 0\ \}$ is a (non-empty) chain in D

and FIX f exists because D is a ccpo

FIX f is a *fixed point* (that is, f (FIX f) = FIX f) :

$$\begin{aligned}
f\ (\mathsf{FIX}\ f) &= f\ (\bigsqcup\{\ f^n \perp \mid n \geq 0\ \}) && \text{(definition of FIX } f\text{)}\\
&= \bigsqcup\{\ f(f^n \perp) \mid n \geq 0\ \} && \text{(continuity of } f\text{)}\\
&= \bigsqcup\{\ f^n \perp \mid n \geq 1\ \} && \\
&= \bigsqcup(\{\ f^n \perp \mid n \geq 1\ \} \cup \{\perp\}) && (\textstyle\bigsqcup(Y \cup \{\perp\}) = \bigsqcup Y\\
& && \text{for all chains } Y)\\
&= \bigsqcup\{\ f^n \perp \mid n \geq 0\ \} && (f^0 \perp = \perp)\\
&= \mathsf{FIX}\ f && \text{(definition of FIX } f\text{)}
\end{aligned}$$

FIX f is the *least* fixed point :

assume that d is some other fixed point $\Rightarrow$ Clearly $\perp \sqsubseteq d$

f monotone $\Rightarrow$ $f^n \perp \sqsubseteq f^n\ d$ for n≥0

d was a fixed point $\Rightarrow$ $f^n \perp \sqsubseteq d$ for all n≥0

d is an upper bound of the chain $\{\ f^n \perp \mid n \geq 0\ \}$
FIX f is the least upper bound $\Rightarrow$ FIX $f \sqsubseteq d$

□

Fixed Point Theory	
1:	We restrict ourselves to *chain complete partially ordered sets* (abbreviated ccpo).
2:	We restrict ourselves to *continuous functions* on chain complete partially ordered sets.
3:	We show that continuous functions on chain complete partially ordered sets always have *least fixed points* (Theorem 5.37).

5.3. SEMÁNTICAS DE ESTILO DIRECTO: EXISTENCIA

Volvemos a considerar $\boxed{\mathcal{S}_{ds}[\![\texttt{while } b \texttt{ do } S]\!] = \mathsf{FIX}\ F \quad \text{where} \quad F\ g = \mathsf{cond}(\mathcal{B}[\![b]\!],\ g \circ \mathcal{S}_{ds}[\![S]\!],\ \mathrm{id})}$

Necesitamos demostrar que F es CONTINUA: $F\ g = F_1\ (F_2\ g)$ **donde** $F_1\ g = \mathsf{cond}(\mathcal{B}[\![b]\!],\ g,\ \mathrm{id})$, $F_2\ g = g \circ \mathcal{S}_{ds}[\![S]\!]$

Por el Lema 5.35 es suficiente demostrar que F_1 y F_2 son CONTINUAS:

LEMA 5.43:

g_0: **State** $\hookrightarrow$ **State**
p: **State** $\rightarrow$ **T**
$F\ g = \mathsf{cond}(p,\ g,\ g_0)$

Then F is continuous

Demostración:

F is *monotone*?

assume that $g_1 \sqsubseteq g_2$ and we shall show that $F\ g_1 \sqsubseteq F\ g_2$

consider state s and show that $(F\ g_1)\ s = s'$ implies $(F\ g_2)\ s = s'$

If $p\ s = \mathbf{tt}$:

$(F\ g_1)\ s = g_1\ s$

from $g_1 \sqsubseteq g_2$ we get $g_1\ s = s'$ implies $g_2\ s = s'$

Since $(F\ g_2)\ s = g_2\ s$, we have proved the result

If $p\ s = \mathbf{ff}$:

$(F\ g_1)\ s = g_0\ s$ and $(F\ g_2)\ s = g_0\ s$ $\Rightarrow$ the result is immediate

F is *continuous*?

let Y be a non-empty chain in **State** $\hookrightarrow$ **State**

We must show that $F\ (\bigsqcup Y) \sqsubseteq \bigsqcup\{\ F\ g \mid g \in Y\ \}$

F *monotone* $\xRightarrow[\text{Lema 5.30}]{}$ $F\ (\bigsqcup Y) \sqsupseteq \bigsqcup\{\ F\ g \mid g \in Y\ \}$

Thus we have to show that $\mathrm{graph}(F(\bigsqcup Y)) \subseteq \bigcup\{\ \mathrm{graph}(F\ g) \mid g \in Y\ \}$

using the characterization of least upper bounds
of chains in **State** $\hookrightarrow$ **State** given in Lemma 5.25.

assume that $(F\ (\bigsqcup Y))\ s = s'$ $\Rightarrow$ let us determine $g \in Y$ such that $(F\ g)\ s = s'$.

If $p\ s = \mathbf{ff}$:

$F\ (\bigsqcup Y)\ s = g_0\ s = s'$

for every element g of the non-empty set Y we have $(F\ g)\ s = g_0\ s = s'$

If $p\ s = \mathbf{tt}$:

$(F\ (\bigsqcup Y))\ s = (\bigsqcup Y)\ s = s'$ $\Rightarrow$ $\langle s, s' \rangle \in \mathrm{graph}(\bigsqcup Y)$

$\underbrace{\mathrm{graph}(\bigsqcup Y) = \bigcup\{\ \mathrm{graph}(g) \mid g \in Y\ \}}_{\text{(Lema 5.25)}}$ $\Rightarrow$ we have $g \in Y$ such that $g\ s = s'$ $\Rightarrow$ $(F\ g)\ s = s'$ $\square$

LEMA 5.45: g_0: **State** $\hookrightarrow$ **State**, $F\ g = g \circ g_0$ Then F is continuous.

Demostración:

F is monotone ?

If $g_1 \sqsubseteq g_2 \Longrightarrow$ graph$(g_1) \subseteq$ graph(g_2)

Exercise 5.8

so that graph$(g_0) \diamond$ graph(g_1) satisfies graph$(g_0) \diamond$ graph$(g_1) \subseteq$ graph$(g_0) \diamond$ graph(g_2)

(relational composition of graph(g_0) and graph(g_1)) see Appendix A

this shows that $F\ g_1 \sqsubseteq F\ g_2$

F is continuous ?

If Y is a non-empty chain :

$$\begin{aligned} \text{graph}(F(\bigsqcup Y)) &= \text{graph}((\bigsqcup Y) \circ g_0) = \text{graph}(g_0) \diamond \text{graph}(\bigsqcup Y) \\ &= \text{graph}(g_0) \diamond \bigcup\{\text{graph}(g) \mid g \in Y\} = \bigcup\{\text{graph}(g_0) \diamond \text{graph}(g) \mid g \in Y\} \\ &= \text{graph}(\bigsqcup\{F\ g \mid g \in Y\}) \end{aligned}$$

where we have used Lemma 5.25 twice.

Thus $F\ (\bigsqcup Y) = \bigsqcup\{F\ g \mid g \in Y\}$ $\square$

PROPOSICIÓN 5.47:

Las semánticas de la Tabla 5.1 definen una función TOTAL S_{ds} **en** $Stm \rightarrow (State \hookrightarrow State)$.

Demostración:

structural induction on S :

The case $x := a$: function that maps a state s to the state $s[x \mapsto \mathcal{A}[\![a]\!]s]$ is well-defined.

The case `skip`: Clearly the function id is well-defined.

HI : $\mathcal{S}_{ds}[\![S_1]\!]$ and $\mathcal{S}_{ds}[\![S_2]\!]$ are well-defined

The case $S_1;S_2$: **with HI** clearly their composition will be well-defined.

The case `if` b `then` S_1 `else` S_2: **with HI** clearly this property is preserved by the function cond.

The case `while` b `do` S:

with HI $\mathcal{S}_{ds}[\![S]\!]$ is well-defined

F_1 and F_2 defined by $F_1\ g = \text{cond}(\mathcal{B}[\![b]\!], g, \text{id})$ and $F_2\ g = g \circ \mathcal{S}_{ds}[\![S]\!]$

F_1 and F_2 are continuous according to Lemmas 5.43 and 5.45.

Lemma 5.35 gives that $F\ g = F_1\ (F_2\ g)$ is continuous.

From Theorem 5.37 we have that FIX F is well-defined

thereby that $\mathcal{S}_{ds}[\![\texttt{while}\ b\ \texttt{do}\ S]\!]$ is well-defined $\square$

Proof Summary for While: **Well-definedness of Denotational Semantics**	
1:	The set **State** $\hookrightarrow$ **State** equipped with an appropriate order $\sqsubseteq$ is a ccpo (Lemmas 5.13 and 5.25).
2:	Certain functions Ψ: (**State** $\hookrightarrow$ **State**) $\to$ (**State** $\hookrightarrow$ **State**) are continuous (Lemmas 5.43 and 5.45).
3:	In the definition of $\mathcal{S}_{ds}$, we only apply the fixed point operation to continuous functions (Proposition 5.47).

PROPIEDADES DE LAS SEMÁNTICAS:

Se pueden definir EQUIVALENCIAS ENTRE SEMÁNTICAS también en DENOTACIONAL:

S_1 and S_2 are *semantically equivalent*	if and only if	$\mathcal{S}_{ds}[\![S_1]\!] = \mathcal{S}_{ds}[\![S_2]\!]$

5.4 UN RESULTADO DE EQUIVALENCIA:

Veremos si hay relación entre SOS y la DENOTACIONAL.

TEOREMA 5.55: For every statement S of **While**, we have $\mathcal{S}_{\mathrm{sos}}[\![S]\!] = \mathcal{S}_{\mathrm{ds}}[\![S]\!]$

Demostración:

$\mathcal{S}_{\mathrm{ds}}[\![S]\!]$ and $\mathcal{S}_{\mathrm{sos}}[\![S]\!]$ are functions in **State** $\hookrightarrow$ **State**

they are elements of a partially ordered set

. To prove that two elements d_1 and d_2 of a partially ordered set are equal

it is sufficient to prove that $d_1 \sqsubseteq d_2$ and that $d_2 \sqsubseteq d_1$

to prove Theorem 5.55, we shall show that

- $\mathcal{S}_{\mathrm{sos}}[\![S]\!] \sqsubseteq \mathcal{S}_{\mathrm{ds}}[\![S]\!]$ (LEMA 5.56)
- $\mathcal{S}_{\mathrm{ds}}[\![S]\!] \sqsubseteq \mathcal{S}_{\mathrm{sos}}[\![S]\!]$ (LEMA 5.57)

Proof Summary for While:

Equivalence of Operational and Denotational Semantics

1: Prove that $\mathcal{S}_{\mathrm{sos}}[\![S]\!] \sqsubseteq \mathcal{S}_{\mathrm{ds}}[\![S]\!]$ by first using *induction on the shape of derivation trees* to show that

- if a statement is executed *one step* in the structural operational semantics and does not terminate, then this does not change the meaning in the denotational semantics, and
- if a statement is executed *one step* in the structural operational semantics and does terminate, then the same result is obtained in the denotational semantics

and secondly by using *induction on the length of derivation sequences.*

2: Prove that $\mathcal{S}_{\mathrm{ds}}[\![S]\!] \sqsubseteq \mathcal{S}_{\mathrm{sos}}[\![S]\!]$ by showing that

- $\mathcal{S}_{\mathrm{sos}}$ fulfils slightly weaker versions of the clauses defining $\mathcal{S}_{\mathrm{ds}}$ in Table 5.1, that is, if

$$\mathcal{S}_{\mathrm{ds}}[\![S]\!] = \Psi(\cdots \mathcal{S}_{\mathrm{ds}}[\![S']\!] \cdots)$$

then $\mathcal{S}_{\mathrm{sos}}[\![S]\!] \sqsupseteq \Psi(\cdots \mathcal{S}_{\mathrm{sos}}[\![S']\!] \cdots)$

A proof by *structural induction* then gives that $\mathcal{S}_{\mathrm{ds}}[\![S]\!] \sqsubseteq \mathcal{S}_{\mathrm{sos}}[\![S]\!]$.

LEMA 5.56: For every statement S of **While**, we have $\mathcal{S}_{\mathrm{sos}}[\![S]\!] \sqsubseteq \mathcal{S}_{\mathrm{ds}}[\![S]\!]$.

Demostración:

for all states s and s' $\langle S, s\rangle \Rightarrow^* s'$ implies $\mathcal{S}_{\mathrm{ds}}[\![S]\!]s = s'$ (*) :

we shall need to establish the following property

$$\begin{array}{lll} \langle S, s\rangle \Rightarrow s' & \text{implies} & \mathcal{S}_{\mathrm{ds}}[\![S]\!]s = s' \\ \langle S, s\rangle \Rightarrow \langle S', s'\rangle & \text{implies} & \mathcal{S}_{\mathrm{ds}}[\![S]\!]s = \mathcal{S}_{\mathrm{ds}}[\![S']\!]s' \end{array} \qquad (**)$$

Assuming that (**) holds the proof of (*) is induction on the length k of the derivation sequence $\langle S, s\rangle \Rightarrow^k s'$

We now turn to the proof of (**), and for this we shall use induction on the shape of the derivation tree for $\langle S, s\rangle \Rightarrow s'$ or $\langle S, s\rangle \Rightarrow \langle S', s'\rangle$.

The case [ass$_{\mathrm{sos}}$]: We have $\langle x := a, s\rangle \Rightarrow s[x \mapsto \mathcal{A}[\![a]\!]s]$ and since $\mathcal{S}_{\mathrm{ds}}[\![x := a]\!]s = s[x \mapsto \mathcal{A}[\![a]\!]s]$, the result follows.

The case [skip$_{\mathrm{sos}}$]: Analogous.

The case [comp$^1_{\mathrm{sos}}$]: $\langle S_1;S_2, s\rangle \Rightarrow \langle S_1';S_2, s'\rangle$ because $\langle S_1, s\rangle \Rightarrow \langle S_1', s'\rangle$

Then the induction hypothesis applied to the latter transition gives

$\mathcal{S}_{\mathrm{ds}}[\![S_1]\!]s = \mathcal{S}_{\mathrm{ds}}[\![S_1']\!]s'$ and we get $\mathcal{S}_{\mathrm{ds}}[\![S_1;S_2]\!]\ s = \mathcal{S}_{\mathrm{ds}}[\![S_2]\!](\mathcal{S}_{\mathrm{ds}}[\![S_1]\!]s) = \mathcal{S}_{\mathrm{ds}}[\![S_2]\!](\mathcal{S}_{\mathrm{ds}}[\![S_1']\!]s') = \mathcal{S}_{\mathrm{ds}}[\![S_1';S_2]\!]s'$

The case [comp$^2_{\mathrm{sos}}$]: $\langle S_1;S_2, s\rangle \Rightarrow \langle S_2, s'\rangle$ because $\langle S_1, s\rangle \Rightarrow s'$

Then the induction hypothesis applied to that transition gives

$\mathcal{S}_{\mathrm{ds}}[\![S_1]\!]s = s'$ and get $\mathcal{S}_{\mathrm{ds}}[\![S_1;S_2]\!]s = \mathcal{S}_{\mathrm{ds}}[\![S_2]\!](\mathcal{S}_{\mathrm{ds}}[\![S_1]\!]s) = \mathcal{S}_{\mathrm{ds}}[\![S_2]\!]s'$

where the first equality comes from the definition of $\mathcal{S}_{\mathrm{ds}}$ and we just argued for the second equality.

The case [if$^{\mathrm{tt}}_{\mathrm{sos}}$]: $\langle \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2, s\rangle \Rightarrow \langle S_1, s\rangle$ because $\mathcal{B}[\![b]\!]\ s = \mathbf{tt}$

Then $\mathcal{S}_{\mathrm{ds}}[\![\texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2]\!]s = \mathsf{cond}(\mathcal{B}[\![b]\!], \mathcal{S}_{\mathrm{ds}}[\![S_1]\!], \mathcal{S}_{\mathrm{ds}}[\![S_2]\!])s = \mathcal{S}_{\mathrm{ds}}[\![S_1]\!]s$ as required.

The case [if$^{\mathrm{ff}}_{\mathrm{sos}}$]: Analogous.

The case [while$_{\mathrm{sos}}$]:

$\langle \texttt{while}\ b\ \texttt{do}\ S, s\rangle \Rightarrow \langle \texttt{if}\ b\ \texttt{then}\ (S; \texttt{while}\ b\ \texttt{do}\ S)\ \texttt{else skip}, s\rangle$

From the definition of $\mathcal{S}_{\mathrm{ds}}$, we have

$\mathcal{S}_{\mathrm{ds}}[\![\texttt{while}\ b\ \texttt{do}\ S]\!] = \mathsf{FIX}\ F$, where $F\ g = \mathsf{cond}(\mathcal{B}[\![b]\!], g \circ \mathcal{S}_{\mathrm{ds}}[\![S]\!], \mathrm{id})$.

We therefore get

$$\begin{aligned} \mathcal{S}_{\mathrm{ds}}[\![\texttt{while}\ b\ \texttt{do}\ S]\!] &= (\mathsf{FIX}\ F) = F\ (\mathsf{FIX}\ F) = \mathsf{cond}(\mathcal{B}[\![b]\!], (\mathsf{FIX}\ F) \circ \mathcal{S}_{\mathrm{ds}}[\![S]\!], \mathrm{id}) \\ &= \mathsf{cond}(\mathcal{B}[\![b]\!], \mathcal{S}_{\mathrm{ds}}[\![\texttt{while}\ b\ \texttt{do}\ S]\!] \circ \mathcal{S}_{\mathrm{ds}}[\![S]\!], \mathrm{id}) \\ &= \mathsf{cond}(\mathcal{B}[\![b]\!], \mathcal{S}_{\mathrm{ds}}[\![S; \texttt{while}\ b\ \texttt{do}\ S]\!], \mathcal{S}_{\mathrm{ds}}[\![\texttt{skip}]\!]) \\ &= \mathcal{S}_{\mathrm{ds}}[\![\texttt{if}\ b\ \texttt{then}\ (S; \texttt{while}\ b\ \texttt{do}\ S)\ \texttt{else skip}]\!] \end{aligned}$$

as required.

□

LEMA 5.57: For every statement S of **While**, we have $\mathcal{S}_{\rm ds}[\![S]\!] \sqsubseteq \mathcal{S}_{\rm sos}[\![S]\!]$.

Demostración:

structural induction on S.

The case $x := a$: Clearly $\mathcal{S}_{\rm ds}[\![x := a]\!]s = \mathcal{S}_{\rm sos}[\![x := a]\!]s$.

Note that this means that $\mathcal{S}_{\rm sos}$ satisfies the clause defining $\mathcal{S}_{\rm ds}$ in Table 5.1.

The case skip: Clearly $\mathcal{S}_{\rm ds}[\![\texttt{skip}]\!]s = \mathcal{S}_{\rm sos}[\![\texttt{skip}]\!]s$.

The case S_1 ; S_2: Recall that $\circ$ is monotone in both arguments (Lemma 5.45 and Exercise 5.46).

$$\mathcal{S}_{\rm ds}[\![S_1 ; S_2]\!] = \mathcal{S}_{\rm ds}[\![S_2]\!] \circ \mathcal{S}_{\rm ds}[\![S_1]\!] \sqsubseteq \mathcal{S}_{\rm sos}[\![S_2]\!] \circ \mathcal{S}_{\rm sos}[\![S_1]\!]$$

because the induction hypothesis applied to S_1 and S_2 gives $\mathcal{S}_{\rm ds}[\![S_1]\!] \sqsubseteq \mathcal{S}_{\rm sos}[\![S_1]\!]$ and $\mathcal{S}_{\rm ds}[\![S_2]\!] \sqsubseteq \mathcal{S}_{\rm sos}[\![S_2]\!]$.

Exercise 2.21 : if $\langle S_1, s\rangle \Rightarrow^* s'$ then $\langle S_1 ; S_2, s\rangle \Rightarrow^* \langle S_2, s'\rangle$ $\Rightarrow$ $\mathcal{S}_{\rm sos}[\![S_2]\!] \circ \mathcal{S}_{\rm sos}[\![S_1]\!] \sqsubseteq \mathcal{S}_{\rm sos}[\![S_1 ; S_2]\!]$

Note that in this case $\mathcal{S}_{\rm sos}$ fulfils a weaker version of the clause defining $\mathcal{S}_{\rm ds}$ in Table 5.1.

The case if b then S_1 else S_2:

Recall that cond is monotone in its second and third arguments (Lemma 5.43 and Exercise 5.44).

$$\mathcal{S}_{\rm ds}[\![\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2]\!] = \mathsf{cond}(\mathcal{B}[\![b]\!], \mathcal{S}_{\rm ds}[\![S_1]\!], \mathcal{S}_{\rm ds}[\![S_2]\!]) \sqsubseteq \mathsf{cond}(\mathcal{B}[\![b]\!], \mathcal{S}_{\rm sos}[\![S_1]\!], \mathcal{S}_{\rm sos}[\![S_2]\!])$$

because the induction hypothesis applied to S_1 and S_2 gives $\mathcal{S}_{\rm ds}[\![S_1]\!] \sqsubseteq \mathcal{S}_{\rm sos}[\![S_1]\!]$ and $\mathcal{S}_{\rm ds}[\![S_2]\!] \sqsubseteq \mathcal{S}_{\rm sos}[\![S_2]\!]$.

Furthermore, it follows from $[\text{if}^{\rm tt}_{\rm sos}]$ and $[\text{if}^{\rm ff}_{\rm sos}]$ that

$$\mathcal{S}_{\rm sos}[\![\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2]\!]s = \mathcal{S}_{\rm sos}[\![S_1]\!]s \quad \text{if } \mathcal{B}[\![b]\!]s = \mathbf{tt}$$

$$\mathcal{S}_{\rm sos}[\![\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2]\!]s = \mathcal{S}_{\rm sos}[\![S_2]\!]s \quad \text{if } \mathcal{B}[\![b]\!]s = \mathbf{ff}$$

so that

$\mathsf{cond}(\mathcal{B}[\![b]\!], \mathcal{S}_{\rm sos}[\![S_1]\!], \mathcal{S}_{\rm sos}[\![S_2]\!]) = \mathcal{S}_{\rm sos}[\![\texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2]\!]$ and this proves the result.

Note that in this case $\mathcal{S}_{\rm sos}$ fulfils the clause defining $\mathcal{S}_{\rm ds}$ in Table 5.1.

The case while b do S: We have $\mathcal{S}_{\rm ds}[\![\texttt{while } b \texttt{ do } S]\!] = \mathsf{FIX}\ F$ where $F\ g = \mathsf{cond}(\mathcal{B}[\![b]\!], g \circ \mathcal{S}_{\rm ds}[\![S]\!], \text{id})$

we recall that F is continuous.

It is sufficient to prove that $F(\mathcal{S}_{\rm sos}[\![\texttt{while } b \texttt{ do } S]\!]) \sqsubseteq \mathcal{S}_{\rm sos}[\![\texttt{while } b \texttt{ do } S]\!]$

because then Exercise 5.40 gives $\mathsf{FIX}\ F \sqsubseteq \mathcal{S}_{\rm sos}[\![\texttt{while } b \texttt{ do } S]\!]$ as required.

Exercise 2.21:

$$\mathcal{S}_{\rm sos}[\![\texttt{while } b \texttt{ do } S]\!] = \mathsf{cond}(\mathcal{B}[\![b]\!], \mathcal{S}_{\rm sos}[\![S ; \texttt{while } b \texttt{ do } S]\!], \text{id}) \sqsupseteq$$

$$\sqsupseteq \mathsf{cond}(\mathcal{B}[\![b]\!], \mathcal{S}_{\rm sos}[\![\texttt{while } b \texttt{ do } S]\!] \circ \mathcal{S}_{\rm sos}[\![S]\!], \text{id})$$

The induction hypothesis applied to S gives $\mathcal{S}_{\rm ds}[\![S]\!] \sqsubseteq \mathcal{S}_{\rm sos}[\![S]\!]$, so using the monotonicity of $\circ$ and cond:

$$\begin{aligned} \mathcal{S}_{\rm sos}[\![\texttt{while } b \texttt{ do } S]\!] &\sqsupseteq \mathsf{cond}(\mathcal{B}[\![b]\!], \mathcal{S}_{\rm sos}[\![\texttt{while } b \texttt{ do } S]\!] \circ \mathcal{S}_{\rm sos}[\![S]\!], \text{id}) \\ &\sqsupseteq \mathsf{cond}(\mathcal{B}[\![b]\!], \mathcal{S}_{\rm sos}[\![\texttt{while } b \texttt{ do } S]\!] \circ \mathcal{S}_{\rm ds}[\![S]\!], \text{id}) = F(\mathcal{S}_{\rm sos}[\![\texttt{while } b \texttt{ do } S]\!]) \end{aligned}$$

Note that in this case $\mathcal{S}_{\rm sos}$ also fulfils a weaker version of the clause defining $\mathcal{S}_{\rm ds}$ in Table 5.1. $\square$