Semántica Operacional

CI4722 – Lenguajes de Programación III

Ernesto Hernández-Novich
<emhn@usb.ve>

Universidad "Simón Bolívar"

UNIVERSIDAD SIMÓN BOLÍVAR

Semántica Formal

- Modelo matemático que permita comprender y razonar sobre el *comportamiento* de los programas de un lenguaje.

UNIVERSIDAD SIMÓN BOLÍVAR

Semántica Formal

- Modelo matemático que permita comprender y razonar sobre el *comportamiento* de los programas de un lenguaje.
- **Semántica Axiomática** – Floyd, Hoare
  - Enfasis en demostrar la correctitud de inicio a fin.
  - Significado de cada instrucción como una regla de lógica.

UNIVERSIDAD SIMÓN BOLÍVAR

Semántica Formal

- Modelo matemático que permita comprender y razonar sobre el *comportamiento* de los programas de un lenguaje.
- **Semántica Axiomática** – Floyd, Hoare
  - Enfasis en demostrar la correctitud de inicio a fin.
  - Significado de cada instrucción como una regla de lógica.
- **Semántica Operacional** – Plotkin
  - Enfasis en explicar cómo se ejecuta en una máquina abstracta.
  - Especificación de expresiones y comandos dirigidos por sintaxis.

UNIVERSIDAD SIMÓN BOLÍVAR

Semántica Formal

- Modelo matemático que permita comprender y razonar sobre el *comportamiento* de los programas de un lenguaje.
- **Semántica Axiomática** – Floyd, Hoare
  - Enfasis en demostrar la correctitud de inicio a fin.
  - Significado de cada instrucción como una regla de lógica.
- **Semántica Operacional** – Plotkin
  - Enfasis en explicar cómo se ejecuta en una máquina abstracta.
  - Especificación de expresiones y comandos dirigidos por sintaxis.
- **Semántica Denotacional** – Strachey, Scott
  - Enfasis en describir las transformaciones efectivas.
  - Especificación usando funciones sobre órdenes parciales.

UNIVERSIDAD SIMÓN BOLÍVAR

Técnicas complementarias

- Semántica Axiomática
  - Util para el aprendizaje de técnicas algorítmicas.
  - Insuficiente para implantar lenguajes
- Semántica Operacional
  - Util para los implementadores de un lenguaje.
  - Soporte a la construcción de axiomas semánticos.
- Semántica Denotacional
  - Modelo matemático provee más técnicas de análisis.
  - Soporte a la construcción de axiomas semánticos.

UNIVERSIDAD SIMÓN BOLÍVAR

Técnicas complementarias

- Semántica Axiomática
  - Util para el aprendizaje de técnicas algorítmicas.
  - Insuficiente para implantar lenguajes
- Semántica Operacional
  - Util para los implementadores de un lenguaje.
  - Soporte a la construcción de axiomas semánticos.
- Semántica Denotacional
  - Modelo matemático provee más técnicas de análisis.
  - Soporte a la construcción de axiomas semánticos.

La implantación de un lenguaje es "correcta" sólo si las semánticas operacional y denotacional concuerdan.

UNIVERSIDAD SIMÓN BOLÍVAR

Semántica Operacional

- Describe el **comportamiento** de los programas de un lenguaje.
- Separa evaluación de ejecución – modelo de máquina abstracta.
- Cumple como notación formal para la implantación del lenguaje – "alcanzas" el lenguaje si implantas su semántica operacional.
- Permite demostrar algunas propiedades del lenguaje – equivalencia entre comandos suele ser la de interés.

UNIVERSIDAD SIMÓN BOLÍVAR

IMP – un lenguaje imperativo

Definición inductiva de los conjuntos sintácticos

IMP – un lenguaje imperativo

Definición inductiva de los conjuntos sintácticos

$$
\begin{aligned}
E &\leftarrow \mathbf{n} \\
&\leftarrow \mathbf{id} \\
&\leftarrow E_0 + E_1 \\
&\leftarrow E_0 - E_1 \\
&\leftarrow E_0 * E_1
\end{aligned}
$$

UNIVERSIDAD SIMÓN BOLÍVAR

IMP – un lenguaje imperativo

Definición inductiva de los conjuntos sintácticos

$$
\begin{aligned}
E &\leftarrow \mathbf{n} \\
&\leftarrow \mathbf{id} \\
&\leftarrow E_0 + E_1 \\
&\leftarrow E_0 - E_1 \\
&\leftarrow E_0 * E_1
\end{aligned}
\qquad
\begin{aligned}
B &\leftarrow \mathbf{true} \\
&\leftarrow \mathbf{false} \\
&\leftarrow E_0 = E_1 \\
&\leftarrow E_0 \leq E_1 \\
&\leftarrow \neg B_1 \\
&\leftarrow E_0 \wedge E_1 \\
&\leftarrow E_0 \vee E_1
\end{aligned}
$$

UNIVERSIDAD SIMÓN BOLÍVAR

IMP – un lenguaje imperativo

Definición inductiva de los conjuntos sintácticos

$$
\begin{aligned}
E &\leftarrow \mathbf{n} \\
&\leftarrow \mathbf{id} \\
&\leftarrow E_0 + E_1 \\
&\leftarrow E_0 - E_1 \\
&\leftarrow E_0 * E_1
\end{aligned}
\qquad
\begin{aligned}
B &\leftarrow \mathbf{true} \\
&\leftarrow \mathbf{false} \\
&\leftarrow E_0 = E_1 \\
&\leftarrow E_0 \leq E_1 \\
&\leftarrow \neg B_1 \\
&\leftarrow E_0 \wedge E_1 \\
&\leftarrow E_0 \vee E_1
\end{aligned}
\qquad
\begin{aligned}
C &\leftarrow \mathbf{skip} \\
&\leftarrow \mathbf{id} := E \\
&\leftarrow C_0; C_1 \\
&\leftarrow \mathbf{if}\ B\ \mathbf{then}\ C_0\ \mathbf{else}\ C_1 \\
&\leftarrow \mathbf{while}\ B\ \mathbf{do}\ C
\end{aligned}
$$

UNIVERSIDAD SIMÓN BOLÍVAR

IMP – un lenguaje imperativo

Definición inductiva de los conjuntos sintácticos

$$
\begin{array}{lll}
E \leftarrow \mathbf{n} & B \leftarrow \mathbf{true} & C \leftarrow \mathbf{skip} \\
\quad \leftarrow \mathbf{id} & \quad \leftarrow \mathbf{false} & \quad \leftarrow \mathbf{id} := E \\
\quad \leftarrow E_0 + E_1 & \quad \leftarrow E_0 = E_1 & \quad \leftarrow C_0; C_1 \\
\quad \leftarrow E_0 - E_1 & \quad \leftarrow E_0 \leq E_1 & \quad \leftarrow \mathbf{if}\ B\ \mathbf{then}\ C_0\ \mathbf{else}\ C_1 \\
\quad \leftarrow E_0 * E_1 & \quad \leftarrow \neg B_1 & \quad \leftarrow \mathbf{while}\ B\ \mathbf{do}\ C \\
 & \quad \leftarrow E_0 \wedge E_1 & \\
 & \quad \leftarrow E_0 \vee E_1 &
\end{array}
$$

- $\mathbf{n} \in \mathbf{N}$ – enteros positivos, negativos y cero.
- $\mathbf{id} \in \mathbf{Loc}$ – identificadores son ubicaciones mutables (locations).

UNIVERSIDAD SIMÓN BOLÍVAR

IMP – un lenguaje imperativo

Definición inductiva de los conjuntos sintácticos

$$
\begin{array}{lll}
E \leftarrow \mathbf{n} & B \leftarrow \mathbf{true} & C \leftarrow \mathbf{skip} \\
\quad \leftarrow \mathbf{id} & \quad \leftarrow \mathbf{false} & \quad \leftarrow \mathbf{id} := E \\
\quad \leftarrow E_0 + E_1 & \quad \leftarrow E_0 = E_1 & \quad \leftarrow C_0; C_1 \\
\quad \leftarrow E_0 - E_1 & \quad \leftarrow E_0 \leq E_1 & \quad \leftarrow \mathbf{if}\ B\ \mathbf{then}\ C_0\ \mathbf{else}\ C_1 \\
\quad \leftarrow E_0 * E_1 & \quad \leftarrow \neg B_1 & \quad \leftarrow \mathbf{while}\ B\ \mathbf{do}\ C \\
 & \quad \leftarrow E_0 \wedge E_1 & \\
 & \quad \leftarrow E_0 \vee E_1 &
\end{array}
$$

- $\mathbf{n} \in \mathbf{N}$ – enteros positivos, negativos y cero.
- $\mathbf{id} \in \mathbf{Loc}$ – identificadores son ubicaciones mutables (locations).

¿Cómo *funciona* un programa **IMP**?

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

El modelo intuitivo que hemos desarrollado

- "Hay unas cajitas que cambian su valor"
  - *Estado* definido por los valores de los identificadores.
  - Expresiones aritméticas y booleanas son evaluadas en ese contexto.
  - Resultado de las expresiones influyen en los comandos – pueden conducir a cambios en el estado.
- Eso es suficiente para "explicar" el funcionamiento, pero es insuficiente para razonar o deducir cosas formalmente.

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

El modelo intuitivo que hemos desarrollado

- "Hay unas cajitas que cambian su valor"
  - *Estado* definido por los valores de los identificadores.
  - Expresiones aritméticas y booleanas son evaluadas en ese contexto.
  - Resultado de las expresiones influyen en los comandos – pueden conducir a cambios en el estado.
- Eso es suficiente para "explicar" el funcionamiento, pero es insuficiente para razonar o deducir cosas formalmente.

¿Cómo formalizamos esta intuición?

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

Estado mutable

- Σ es el conjunto de Estados, constituido por *funciones*

$$\sigma : \mathbf{Loc} \to \mathbf{N}$$

- $\sigma(\mathbf{id})$ es el valor de **id** en el estado σ

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

Evaluar una expresión

- Los programas evalúan expresiones en un estado particular – si $e \in E$, denotaremos esta intención como **configuraciones**

$$\langle e, \sigma \rangle$$

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

Evaluar una expresión

- Los programas evalúan expresiones en un estado particular – si $e \in E$, denotaremos esta intención como **configuraciones**

$$\langle e, \sigma \rangle$$

- Podemos definir la relación de **configuraciones** a valores

$$\langle e, \sigma \rangle \to n \in \mathbf{N}$$

indicando que n es el resultado de evaluar e en el estado σ.

Usaremos una definición dirigida por sintaxis, pero con notación de inferencia lógica.

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

Evaluación de expresiones – casos base

- **Evaluar números**

$$\frac{}{\langle \mathbf{n}, \sigma \rangle \to \mathbf{n}}$$

- **Evaluar identificadores**

$$\frac{}{\langle \mathbf{id}, \sigma \rangle \to \sigma(\mathbf{id})}$$

Reglas axiomáticas –
suele omitirse la premisa vacía

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

Evaluación de expresiones – casos recursivos

- **Evaluar sumas**

$$\frac{\langle e_0, \sigma \rangle \to n_0 \qquad \langle e_1, \sigma \rangle \to n_1}{\langle e_0 + e_1, \sigma \rangle \to n}$$

donde $n = n_0 + n_1$ – suma ocurre "fuera" del lenguaje.

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

Evaluación de expresiones – casos recursivos

- **Evaluar sumas**

$$\frac{\langle e_0, \sigma \rangle \to n_0 \qquad \langle e_1, \sigma \rangle \to n_1}{\langle e_0 + e_1, \sigma \rangle \to n}$$

donde $n = n_0 + n_1$ – suma ocurre "fuera" del lenguaje.

- **Evaluar productos**

$$\frac{\langle e_0, \sigma \rangle \to n_0 \qquad \langle e_1, \sigma \rangle \to n_1}{\langle e_0 * e_1, \sigma \rangle \to n}$$

donde $n = n_0 \times n_1$ – producto ocurre "fuera" del lenguaje.

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

Evaluación de expresiones – casos recursivos

- **Evaluar sumas**

$$\frac{\langle e_0, \sigma \rangle \to n_0 \qquad \langle e_1, \sigma \rangle \to n_1}{\langle e_0 + e_1, \sigma \rangle \to n}$$

donde $n = n_0 + n_1$ – suma ocurre "fuera" del lenguaje.

- **Evaluar productos**

$$\frac{\langle e_0, \sigma \rangle \to n_0 \qquad \langle e_1, \sigma \rangle \to n_1}{\langle e_0 * e_1, \sigma \rangle \to n}$$

donde $n = n_0 \times n_1$ – producto ocurre "fuera" del lenguaje.

- **Evaluar diferencias**

$$\frac{\langle e_0, \sigma \rangle \to n_0 \qquad \langle e_1, \sigma \rangle \to n_1}{\langle e_0 - e_1, \sigma \rangle \to n}$$

donde $n = n_0 - n_1$ – resta ocurre "fuera" del lenguaje.

UNIVERSIDAD SIMÓN BOLÍVAR

Uso de las reglas

- *Instanciar* las reglas con números, identificadores y estados específicos.
- Se construyen *árboles de derivación* usando los axiomas y premisas, para deducir consecuencias.
  - De abajo (conclusión) hacia arriba (axioma o premisa).
  - Más de un axioma o premisa podría aplicar en algún caso – es necesario considerarlas todas.
- Es el algoritmo que permite describir el proceso de la evaluación, o sea la **semántica operacional estructural** del lenguaje.

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

Evaluación de expresiones booleanas – casos base

- **Evaluar literales**

$$\frac{}{\langle \textbf{true}, \sigma \rangle \rightarrow \textbf{true}}$$

$$\frac{}{\langle \textbf{false}, \sigma \rangle \rightarrow \textbf{false}}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

Evaluación de expresiones booleanas – casos recursivos

- **Igualdad de expresiones**

$$\frac{\langle b_0, \sigma\rangle \to n \qquad \langle b_1, \sigma\rangle \to m}{\langle b_0 = b_1, \sigma\rangle \to \textbf{true}}$$

si $n = m$

$$\frac{\langle b_0, \sigma\rangle \to n \qquad \langle b_1, \sigma\rangle \to m}{\langle b_0 = b_1, \sigma\rangle \to \textbf{false}}$$

si $n \neq m$

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

Evaluación de expresiones booleanas – casos recursivos

- **Orden numérico de expresiones**

$$\frac{\langle b_0, \sigma\rangle \to n \qquad \langle b_1, \sigma\rangle \to m}{\langle b_0 \leq b_1, \sigma\rangle \to \textbf{true}}$$

si $n \leq m$

$$\frac{\langle b_0, \sigma\rangle \to n \qquad \langle b_1, \sigma\rangle \to m}{\langle b_0 \leq b_1, \sigma\rangle \to \textbf{false}}$$

si $n \not\leq m$

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

Evaluación de expresiones boolenas – casos recursivos

- **Negación booleana**

$$\frac{\langle b, \sigma \rangle \to \textbf{true}}{\langle \neg b, \sigma \rangle \to \textbf{false}}$$

$$\frac{\langle b, \sigma \rangle \to \textbf{false}}{\langle \neg b, \sigma \rangle \to \textbf{true}}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

Evaluación de expresiones booleanas – casos recursivos

- **Conjunción booleana**

$$\frac{\langle b_0, \sigma \rangle \to t_0 \qquad \langle b_1, \sigma \rangle \to t_1}{\langle b_0 \wedge b_1, \sigma \rangle \to t}$$

donde $t = \textbf{true}$ si $t_0 \equiv \textbf{true}$ y $t_1 \equiv \textbf{true}$,
y $t = \textbf{false}$ en cualquier otro caso.

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

Evaluación de expresiones booleanas – casos recursivos

- **Disyunción booleana**

$$\frac{\langle b_0, \sigma \rangle \to t_0 \qquad \langle b_1, \sigma \rangle \to t_1}{\langle b_0 \vee b_1, \sigma \rangle \to t}$$

donde $t = \textbf{true}$ si $t_0 \equiv \textbf{true}$ o $t_1 \equiv \textbf{true}$,
y $t = \textbf{false}$ en cualquier otro caso.

UNIVERSIDAD SIMÓN BOLÍVAR

Comportamiento de un programa

Evaluación de expresiones booleanas – cortocircuito

- **Conjunción booleana** – *left-first-sequential*

$$\frac{\langle b_0, \sigma \rangle \to \textbf{false}}{\langle b_0 \wedge b_1, \sigma \rangle \to \textbf{false}}$$

$$\frac{\langle b_0, \sigma \rangle \to \textbf{true} \qquad \langle b_1, \sigma \rangle \to \textbf{false}}{\langle b_0 \wedge b_1, \sigma \rangle \to \textbf{false}}$$

$$\frac{\langle b_0, \sigma \rangle \to \textbf{true} \qquad \langle b_1, \sigma \rangle \to \textbf{true}}{\langle b_0 \wedge b_1, \sigma \rangle \to \textbf{true}}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Ejecución de comandos

- Expresiones – calcular sobre un estado particular sin cambiarlo.
- Comandos – efectuar cambios sobre el estado.

Ejecución de comandos

- Expresiones – calcular sobre un estado particular sin cambiarlo.
- Comandos – efectuar cambios sobre el estado.
- Estado inicial – $\forall id, \sigma_0(\mathbf{id}) = 0$

Ejecución de comandos

- Expresiones – calcular sobre un estado particular sin cambiarlo.
- Comandos – efectuar cambios sobre el estado.
- Estado inicial – $\forall id, \sigma_0(\mathbf{id}) = 0$
- $\langle c, \sigma \rangle$ – **configuración** indicando que c está por ejecutar sobre σ.
- Definiremos la relación

$$\langle c, \sigma \rangle \to \sigma'$$

para indicar que la ejecución de c sobre σ *termina* en el estado σ'

Transformar el estado requiere cambiar valores.

UNIVERSIDAD SIMÓN BOLÍVAR

Ejecución de comandos

Notación para las transformaciones

Sea σ un estado, $m \in \mathbf{N}$ e $\mathbf{id} \in \mathbf{Loc}$. Escribimos

$$\sigma[m/\mathbf{id}](Y) = \begin{cases} m & : Y = \mathbf{id} \\ \sigma(Y) & : Y \neq \mathbf{id} \end{cases}$$

para denotar como $\sigma[m/\mathbf{id}]$, al estado σ luego de reemplazar los contenidos de **id** con el valor m.

UNIVERSIDAD SIMÓN BOLÍVAR

Ejecución de comandos

Comandos atómicos

- **Comando vacío**

$$\frac{}{\langle \textbf{skip}, \sigma \rangle \to \sigma}$$

- **Asignación**

$$\frac{\langle e, \sigma \rangle \to m}{\langle \textbf{id} := e, \sigma \rangle \to \sigma[m/\textbf{id}]}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Ejecución de comandos

Comandos compuestos

- **Secuenciación**

$$\frac{\langle c_0, \sigma \rangle \to \sigma' \qquad \langle c_1, \sigma' \rangle \to \sigma''}{\langle c_0; c_1, \sigma \rangle \to \sigma''}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Ejecución de comandos

Comandos compuestos

- **Secuenciación**

$$\frac{\langle c_0, \sigma \rangle \to \sigma' \qquad \langle c_1, \sigma' \rangle \to \sigma''}{\langle c_0; c_1, \sigma \rangle \to \sigma''}$$

- **Condicionales**

$$\frac{\langle b, \sigma \rangle \to \textbf{true} \qquad \langle c_0, \sigma \rangle \to \sigma'}{\langle \textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1, \sigma \rangle \to \sigma'}$$

$$\frac{\langle b, \sigma \rangle \to \textbf{false} \qquad \langle c_1, \sigma \rangle \to \sigma'}{\langle \textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1, \sigma \rangle \to \sigma'}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Ejecución de comandos

Comandos compuestos

- **Iteración condicional**

$$\frac{\langle b, \sigma \rangle \to \textbf{false}}{\langle \textbf{while } b \textbf{ do } c, \sigma \rangle \to \sigma}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Ejecución de comandos

Comandos compuestos

- **Iteración condicional**

$$\frac{\langle b, \sigma \rangle \to \textbf{false}}{\langle \textbf{while } b \textbf{ do } c, \sigma \rangle \to \sigma}$$

$$\frac{\langle b, \sigma \rangle \to \textbf{true} \qquad \langle c, \sigma \rangle \to \sigma' \qquad \langle \textbf{while } b \textbf{ do } c, \sigma' \rangle \to \sigma''}{\langle \textbf{while } b \textbf{ do } c, \sigma \rangle \to \sigma''}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Relaciones de Equivalencia

- Dos expresiones aritméticas son equivalentes

$$e_0 \equiv e_1 \text{ si y sólo si } \forall n \in \mathbf{N}, \forall \sigma \in \Sigma : \langle e_0, \sigma \rangle \to n \Leftrightarrow \langle e_1, \sigma \rangle \to n$$

Relaciones de Equivalencia

- Dos expresiones aritméticas son equivalentes

$$e_0 \equiv e_1 \text{ si y sólo si } \forall n \in \mathbf{N}, \forall \sigma \in \Sigma : \langle e_0, \sigma \rangle \to n \Leftrightarrow \langle e_1, \sigma \rangle \to n$$

- Dos expresiones booleanas son equivalentes

$$b_0 \equiv b_1 \text{ si y sólo si } \forall t, \forall \sigma \in \Sigma : \langle b_0, \sigma \rangle \to t \Leftrightarrow \langle b_1, \sigma \rangle \to t$$

UNIVERSIDAD SIMÓN BOLÍVAR

Relaciones de Equivalencia

- Dos expresiones aritméticas son equivalentes

$$e_0 \equiv e_1 \text{ si y sólo si } \forall n \in \mathbf{N}, \forall \sigma \in \Sigma : \langle e_0, \sigma \rangle \to n \Leftrightarrow \langle e_1, \sigma \rangle \to n$$

- Dos expresiones booleanas son equivalentes

$$b_0 \equiv b_1 \text{ si y sólo si } \forall t, \forall \sigma \in \Sigma : \langle b_0, \sigma \rangle \to t \Leftrightarrow \langle b_1, \sigma \rangle \to t$$

- Dos comandos son equivalentes

$$c_0 \equiv c_1 \text{ si y sólo si } \forall \sigma, \sigma' \in \Sigma : \langle c_0, \sigma \rangle \to \sigma' \Leftrightarrow \langle c_1, \sigma \rangle \to \sigma'$$

UNIVERSIDAD SIMÓN BOLÍVAR

Relaciones de Equivalencia

- Dos expresiones aritméticas son equivalentes

$$e_0 \equiv e_1 \text{ si y sólo si } \forall n \in \mathbf{N}, \forall \sigma \in \Sigma : \langle e_0, \sigma \rangle \to n \Leftrightarrow \langle e_1, \sigma \rangle \to n$$

- Dos expresiones booleanas son equivalentes

$$b_0 \equiv b_1 \text{ si y sólo si } \forall t, \forall \sigma \in \Sigma : \langle b_0, \sigma \rangle \to t \Leftrightarrow \langle b_1, \sigma \rangle \to t$$

- Dos comandos son equivalentes

$$c_0 \equiv c_1 \text{ si y sólo si } \forall \sigma, \sigma' \in \Sigma : \langle c_0, \sigma \rangle \to \sigma' \Leftrightarrow \langle c_1, \sigma \rangle \to \sigma'$$

Relaciones de Transición
fundamento de los Sistemas de Transiciones

Demostrando propiedades

Iteración como secuencia de condicionales

Sea $w \equiv \textbf{while}\ b\ \textbf{do}\ c$ con $b \in B$ y $c \in C$, queremos demostrar

$$w \equiv \textbf{if}\ b\ \textbf{then}\ c; w\ \textbf{else skip}$$

esto es, "desenrollar" el ciclo condicional.

UNIVERSIDAD SIMÓN BOLÍVAR

Demostrando propiedades

Iteración como secuencia de condicionales

Sea $w \equiv \textbf{while } b \textbf{ do } c$ con $b \in B$ y $c \in C$, queremos demostrar

$$w \equiv \textbf{if } b \textbf{ then } c; w \textbf{ else skip}$$

esto es, "desenrollar" el ciclo condicional.
Operacionalmente, queremos demostrar la veracidad de

$$\forall \sigma, \sigma' : \langle w, \sigma \rangle \to \sigma' \Leftrightarrow \langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma'$$

UNIVERSIDAD SIMÓN BOLÍVAR

Demostrando propiedades

Iteración como secuencia de condicionales

$(\Rightarrow)$ i.e. $\forall \sigma, \sigma' : \langle w, \sigma \rangle \to \sigma' \Rightarrow \langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma'$

Demostrando propiedades

Iteración como secuencia de condicionales

($\Rightarrow$) i.e. $\forall \sigma, \sigma' : \langle w, \sigma \rangle \to \sigma' \Rightarrow \langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma'$

1. Suponemos cierto $\langle w, \sigma \rangle \to \sigma'$

UNIVERSIDAD SIMÓN BOLÍVAR

Demostrando propiedades

Iteración como secuencia de condicionales

($\Rightarrow$) i.e. $\forall \sigma, \sigma' : \langle w, \sigma \rangle \to \sigma' \Rightarrow \langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma'$

1. Suponemos cierto $\langle w, \sigma \rangle \to \sigma'$
2. Debe haber una derivación para $\langle w, \sigma \rangle \to \sigma'$

UNIVERSIDAD SIMÓN BOLÍVAR

Demostrando propiedades

Iteración como secuencia de condicionales

($\Rightarrow$) i.e. $\forall \sigma, \sigma' : \langle w, \sigma \rangle \to \sigma' \Rightarrow \langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma'$

1. Suponemos cierto $\langle w, \sigma \rangle \to \sigma'$
2. Debe haber una derivación para $\langle w, \sigma \rangle \to \sigma'$
3. Según las reglas para el **while**, la *última* regla debe ser
   - Caso $(1 \Rightarrow)$

$$\frac{\langle b, \sigma \rangle \to \textbf{false}}{\langle w, \sigma \rangle \to \sigma}$$

   - Caso $(2 \Rightarrow)$

$$\frac{\langle b, \sigma \rangle \to \textbf{true} \qquad \langle c, \sigma \rangle \to \sigma'' \qquad \langle w, \sigma'' \rangle \to \sigma'}{\langle w, \sigma \rangle \to \sigma'}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Demostrando propiedades

Caso $(1 \Rightarrow)$

La derivación de $\langle w, \sigma \rangle \to \sigma'$ debe tener la forma

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \textbf{false}}}{\langle w, \sigma \rangle \to \sigma}$$

Demostrando propiedades

Caso $(1 \Rightarrow)$

La derivación de $\langle w, \sigma \rangle \to \sigma'$ debe tener la forma

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \textbf{false}}}{\langle w, \sigma \rangle \to \sigma}$$

y entonces podemos construir, agregando el axioma de **skip**

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \textbf{false}} \qquad \dfrac{}{\langle \textbf{skip}, \sigma \rangle \to \sigma}}{\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Demostrando propiedades

Caso $(2 \Rightarrow)$

La derivación de $\langle w, \sigma \rangle \to \sigma'$ debe tener la forma

$$\frac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \textbf{true}} \qquad \dfrac{\vdots}{\langle c, \sigma \rangle \to \sigma''} \qquad \dfrac{\vdots}{\langle w, \sigma'' \rangle \to \sigma'}}{\langle w, \sigma \rangle \to \sigma'}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Demostrando propiedades

Caso $(2 \Rightarrow)$

La derivación de $\langle w, \sigma \rangle \to \sigma'$ debe tener la forma

$$\frac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \textbf{true}} \qquad \dfrac{\vdots}{\langle c, \sigma \rangle \to \sigma''} \qquad \dfrac{\vdots}{\langle w, \sigma'' \rangle \to \sigma'}}{\langle w, \sigma \rangle \to \sigma'}$$

y entonces podemos construir, agregando la premisa de secuenciación

$$\frac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \textbf{true}} \qquad \dfrac{\dfrac{\vdots}{\langle c, \sigma \rangle \to \sigma''} \qquad \dfrac{\vdots}{\langle w, \sigma'' \rangle \to \sigma'}}{\langle c; w, \sigma \rangle \to \sigma'}}{\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma'}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Demostrando propiedades

Iteración como secuencia de condicionales

($\Rightarrow$) i.e. $\forall \sigma, \sigma' : \langle w, \sigma \rangle \to \sigma' \Rightarrow \langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip }, \sigma \rangle \to \sigma'$

1. Suponemos cierto $\langle w, \sigma \rangle \to \sigma'$
2. Debe haber una derivación para $\langle w, \sigma \rangle \to \sigma'$
3. La *última* regla debe ser
   - Caso (1 $\Rightarrow$)

$$\frac{\langle b, \sigma \rangle \to \textbf{false}}{\langle w, \sigma \rangle \to \sigma}$$

   - Caso (2 $\Rightarrow$)

$$\frac{\langle b, \sigma \rangle \to \textbf{true} \qquad \langle c, \sigma \rangle \to \sigma'' \qquad \langle w, \sigma'' \rangle \to \sigma'}{\langle w, \sigma \rangle \to \sigma'}$$

4. Para ambos casos construimos la derivación a partir de la premisa, así que la implicación ($\Rightarrow$) se sostiene.

UNIVERSIDAD SIMÓN BOLÍVAR

Demostrando propiedades

Iteración como secuencia de condicionales

($\Leftarrow$) i.e. $\forall \sigma, \sigma' : \langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma' \Rightarrow \langle w, \sigma \rangle \to \sigma'$

Demostrando propiedades

Iteración como secuencia de condicionales

($\Leftarrow$) i.e. $\forall \sigma, \sigma' : \langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma' \Rightarrow \langle w, \sigma \rangle \to \sigma'$

1. Suponemos cierto $\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma'$

UNIVERSIDAD SIMÓN BOLÍVAR

Demostrando propiedades

Iteración como secuencia de condicionales

($\Leftarrow$) i.e. $\forall \sigma, \sigma' : \langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma' \Rightarrow \langle w, \sigma \rangle \to \sigma'$

1. Suponemos cierto $\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma'$
2. Entonces, hay una derivación con dos formas posibles
   - Caso ($1 \Leftarrow$) – noten que $\sigma = \sigma'$

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \textbf{false}} \qquad \langle \textbf{skip}, \sigma \rangle \to \sigma}{\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma}$$

   - Caso ($2 \Leftarrow$)

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \textbf{true}} \qquad \dfrac{\vdots}{\langle c; w, \sigma \rangle \to \sigma'}}{\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma'}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Demostrando propiedades

Caso $(2 \Leftarrow)$

La derivación de $\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma'$ debe tener la forma

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \textbf{true}} \qquad \dfrac{\vdots}{\langle c; w, \sigma \rangle \to \sigma'}}{\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma'}$$

Demostrando propiedades

Caso $(2 \Leftarrow)$

La derivación de $\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma'$ debe tener la forma

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \textbf{true}} \qquad \dfrac{\vdots}{\langle c; w, \sigma \rangle \to \sigma'}}{\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip}, \sigma \rangle \to \sigma'}$$

y desmantelando la secuenciación podemos construir

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \textbf{true}} \qquad \dfrac{\vdots}{\langle c, \sigma \rangle \to \sigma''} \qquad \dfrac{\vdots}{\langle w, \sigma'' \rangle \to \sigma'}}{\langle w, \sigma \rangle \to \sigma'}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Demostrando propiedades

Iteración como secuencia de condicionales

($\Leftarrow$) i.e. $\forall \sigma, \sigma' : \langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip }, \sigma \rangle \to \sigma' \Rightarrow \langle w, \sigma \rangle \to \sigma'$

1. Suponemos cierto $\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip }, \sigma \rangle \to \sigma'$
2. Entonces, hay una derivación con dos formas posibles
   - Caso (1 $\Leftarrow$)

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \textbf{false}} \qquad \langle \textbf{skip}, \sigma \rangle \to \sigma}{\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip }, \sigma \rangle \to \sigma}$$

   - Caso (2 $\Leftarrow$)

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma \rangle \to \textbf{true}} \qquad \dfrac{\vdots}{\langle c; w, \sigma \rangle \to \sigma'}}{\langle \textbf{if } b \textbf{ then } c; w \textbf{ else skip }, \sigma \rangle \to \sigma'}$$

3. Para ambos casos construimos la derivación a partir de la premisa, así que la implicación ($\Leftarrow$) se sostiene.

UNIVERSIDAD SIMÓN BOLÍVAR

Semántica alternativa

Expresiones "paso a paso"

- $\langle e, \sigma \rangle \to n$ y $\langle b, \sigma \rangle \to t$ especifican la evaluación en pasos "largos" – pasan directamente desde la expresión hasta el valor resultante.

Semántica alternativa

Expresiones "paso a paso"

- $\langle e, \sigma \rangle \to n$ y $\langle b, \sigma \rangle \to t$ especifican la evaluación en pasos "largos" – pasan directamente desde la expresión hasta el valor resultante.
- Es posible expresar reglas que pongan en evidencia el "paso a paso" de la evaluación, e.g. para la suma

$$\frac{\langle e_0, \sigma \rangle \to_1 \langle e_0', \sigma \rangle}{\langle e_0 + e_1, \sigma \rangle \to_1 \langle e_0' + e_1, \sigma \rangle}$$

$$\frac{\langle e_1, \sigma \rangle \to_1 \langle e_1', \sigma \rangle}{\langle n + e_1, \sigma \rangle \to_1 \langle n + e_1', \sigma \rangle}$$

$$\frac{}{\langle n + m, \sigma \rangle \to_1 \langle p, \sigma \rangle}$$

UNIVERSIDAD SIMÓN BOLÍVAR

Semántica alternativa

Expresiones "paso a paso"

- $\langle e, \sigma \rangle \to n$ y $\langle b, \sigma \rangle \to t$ especifican la evaluación en pasos "largos" – pasan directamente desde la expresión hasta el valor resultante.
- Es posible expresar reglas que pongan en evidencia el "paso a paso" de la evaluación, e.g. para la suma

$$\frac{\langle e_0, \sigma \rangle \to_1 \langle e_0', \sigma \rangle}{\langle e_0 + e_1, \sigma \rangle \to_1 \langle e_0' + e_1, \sigma \rangle}$$

$$\frac{\langle e_1, \sigma \rangle \to_1 \langle e_1', \sigma \rangle}{\langle n + e_1, \sigma \rangle \to_1 \langle n + e_1', \sigma \rangle}$$

$$\frac{}{\langle n + m, \sigma \rangle \to_1 \langle p, \sigma \rangle}$$

Formalización del orden de evaluación "izquierda a derecha"

Semántica alternativa

Comandos "paso a paso"

- $\langle c, \sigma \rangle \to \sigma'$ especifica la ejecución *completa* en pasos "largos"

Semántica alternativa

Comandos "paso a paso"

- $\langle c, \sigma \rangle \to \sigma'$ especifica la ejecución *completa* en pasos "largos"
- Es posible expresar reglas que pongan en evidencia el "paso a paso" de la ejecución, e.g.

$$\langle x := 42; y := 69, \sigma \rangle \to_1 \langle y := 69, \sigma[42/x] \rangle \to_1 \sigma[42/x][69/y]$$

UNIVERSIDAD SIMÓN BOLÍVAR

Semántica alternativa

Comandos "paso a paso"

- $\langle c, \sigma \rangle \to \sigma'$ especifica la ejecución *completa* en pasos "largos"
- Es posible expresar reglas que pongan en evidencia el "paso a paso" de la ejecución, e.g.

$$\langle x := 42; y := 69, \sigma \rangle \to_1 \langle y := 69, \sigma[42/x] \rangle \to_1 \sigma[42/x][69/y]$$

- ¿Cuán granular? – si tenemos $\langle b, \sigma \rangle \to \mathbf{true}$, puede ser

$$\langle \mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1, \sigma \rangle \to_1 \langle c_0, \sigma \rangle$$

o bien

$$\langle \mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1, \sigma \rangle \to_1 \langle \mathbf{if}\ \mathbf{true}\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1,$$

Ambas modelan lo mismo

Theorem ("pasos largos" equivalentes a "pasos cortos")

$\langle I, \sigma \rangle \to \sigma' \Leftrightarrow \langle I, \sigma \rangle \overset{*}{\to}_1 \sigma'$

Demostración.

Inducción estructural sobre I – Casos Base

UNIVERSIDAD SIMÓN BOLÍVAR

Ambas modelan lo mismo

Theorem ("pasos largos" equivalentes a "pasos cortos")

$\langle l, \sigma \rangle \to \sigma' \Leftrightarrow \langle l, \sigma \rangle \overset{*}{\to}_1 \sigma'$

Demostración.

Inducción estructural sobre l – Casos Base

1. Suponemos $\langle \textbf{skip}, \sigma \rangle \to \sigma'$

UNIVERSIDAD SIMÓN BOLÍVAR

Ambas modelan lo mismo

Theorem ("pasos largos" equivalentes a "pasos cortos")

$\langle l, \sigma \rangle \to \sigma' \Leftrightarrow \langle l, \sigma \rangle \overset{*}{\to}_1 \sigma'$

Demostración.

Inducción estructural sobre l – Casos Base

1. Suponemos $\langle \textbf{skip}, \sigma \rangle \to \sigma'$
2. Pero $\sigma = \sigma'$ por la definición de **skip**

UNIVERSIDAD SIMÓN BOLÍVAR

Ambas modelan lo mismo

Theorem ("pasos largos" equivalentes a "pasos cortos")

$\langle l, \sigma \rangle \to \sigma' \Leftrightarrow \langle l, \sigma \rangle \overset{*}{\to}_1 \sigma'$

Demostración.

Inducción estructural sobre l – Casos Base

1. Suponemos $\langle \textbf{skip}, \sigma \rangle \to \sigma'$
2. Pero $\sigma = \sigma'$ por la definición de **skip**
3. $\langle \textbf{skip}, \sigma \rangle \to_1 \sigma'$

UNIVERSIDAD SIMÓN BOLÍVAR

Ambas modelan lo mismo

Theorem ("pasos largos" equivalentes a "pasos cortos")

$\langle l, \sigma \rangle \to \sigma' \Leftrightarrow \langle l, \sigma \rangle \overset{*}{\to}_1 \sigma'$

Demostración.

Inducción estructural sobre l – Casos Base

1. Suponemos $\langle \textbf{skip}, \sigma \rangle \to \sigma'$
2. Pero $\sigma = \sigma'$ por la definición de **skip**
3. $\langle \textbf{skip}, \sigma \rangle \to_1 \sigma'$
4. $\langle \textbf{skip}, \sigma \rangle \overset{*}{\to}_1 \sigma'$

□

Completen el caso base **id** := E

UNIVERSIDAD SIMÓN BOLÍVAR

Ambas modelan lo mismo

Theorem ("pasos largos" equivalentes a "pasos cortos")

$\langle I, \sigma \rangle \to \sigma' \Leftrightarrow \langle I, \sigma \rangle \overset{*}{\to}_1 \sigma'$

Demostración.

Inducción estructural sobre I – Casos Inductivos

UNIVERSIDAD SIMÓN BOLÍVAR

Ambas modelan lo mismo

Theorem ("pasos largos" equivalentes a "pasos cortos")

$\langle I, \sigma \rangle \to \sigma' \Leftrightarrow \langle I, \sigma \rangle \overset{*}{\to}_1 \sigma'$

Demostración.

Inducción estructural sobre I – Casos Inductivos

1. Suponemos $\forall \sigma, \sigma' : \langle I_0, \sigma \rangle \to \sigma' \Rightarrow \langle I_0, \sigma \rangle \overset{*}{\to}_1 \sigma'$
   y también $\forall \sigma, \sigma' : \langle I_1, \sigma \rangle \to \sigma' \Rightarrow \langle I_1, \sigma \rangle \overset{*}{\to}_1 \sigma'$

UNIVERSIDAD SIMÓN BOLÍVAR

Ambas modelan lo mismo

Theorem ("pasos largos" equivalentes a "pasos cortos")

$\langle I, \sigma \rangle \to \sigma' \Leftrightarrow \langle I, \sigma \rangle \overset{*}{\to}_1 \sigma'$

Demostración.

Inducción estructural sobre I – Casos Inductivos

1. Suponemos $\forall \sigma, \sigma' : \langle I_0, \sigma \rangle \to \sigma' \Rightarrow \langle I_0, \sigma \rangle \overset{*}{\to}_1 \sigma'$
   y también $\forall \sigma, \sigma' : \langle I_1, \sigma \rangle \to \sigma' \Rightarrow \langle I_1, \sigma \rangle \overset{*}{\to}_1 \sigma'$
2. Suponemos $\langle I_0; I_1, \sigma \rangle \to \sigma'$

UNIVERSIDAD SIMÓN BOLÍVAR

Ambas modelan lo mismo

Theorem ("pasos largos" equivalentes a "pasos cortos")

$\langle I, \sigma \rangle \to \sigma' \Leftrightarrow \langle I, \sigma \rangle \xrightarrow{*}_1 \sigma'$

Demostración.

Inducción estructural sobre I – Casos Inductivos

1. Suponemos $\forall \sigma, \sigma' : \langle I_0, \sigma \rangle \to \sigma' \Rightarrow \langle I_0, \sigma \rangle \xrightarrow{*}_1 \sigma'$
   y también $\forall \sigma, \sigma' : \langle I_1, \sigma \rangle \to \sigma' \Rightarrow \langle I_1, \sigma \rangle \xrightarrow{*}_1 \sigma'$
2. Suponemos $\langle I_0; I_1, \sigma \rangle \to \sigma'$
3. Debe haber derivaciones $\langle I_0, \sigma \rangle \to \sigma''$ e $\langle I_1, \sigma'' \rangle \to \sigma'$ para algún σ''.

UNIVERSIDAD SIMÓN BOLÍVAR

Ambas modelan lo mismo

Theorem ("pasos largos" equivalentes a "pasos cortos")

$\langle I, \sigma \rangle \to \sigma' \Leftrightarrow \langle I, \sigma \rangle \xrightarrow{*}_1 \sigma'$

Demostración.

Inducción estructural sobre I – Casos Inductivos

1. Suponemos $\forall \sigma, \sigma' : \langle I_0, \sigma \rangle \to \sigma' \Rightarrow \langle I_0, \sigma \rangle \xrightarrow{*}_1 \sigma'$
   y también $\forall \sigma, \sigma' : \langle I_1, \sigma \rangle \to \sigma' \Rightarrow \langle I_1, \sigma \rangle \xrightarrow{*}_1 \sigma'$
2. Suponemos $\langle I_0; I_1, \sigma \rangle \to \sigma'$
3. Debe haber derivaciones $\langle I_0, \sigma \rangle \to \sigma''$ e $\langle I_1, \sigma'' \rangle \to \sigma'$ para algún σ''.
4. Por la hipótesis concluimos $\langle I_0, \sigma \rangle \xrightarrow{*}_1 \sigma''$ y $\langle I_1, \sigma'' \rangle \xrightarrow{*}_1 \sigma'$

UNIVERSIDAD SIMÓN BOLÍVAR

Ambas modelan lo mismo

Theorem ("pasos largos" equivalentes a "pasos cortos")

$\langle I, \sigma \rangle \to \sigma' \Leftrightarrow \langle I, \sigma \rangle \overset{*}{\to}_1 \sigma'$

Demostración.

Inducción estructural sobre I – Casos Inductivos

1. Suponemos $\forall \sigma, \sigma' : \langle I_0, \sigma \rangle \to \sigma' \Rightarrow \langle I_0, \sigma \rangle \overset{*}{\to}_1 \sigma'$
   y también $\forall \sigma, \sigma' : \langle I_1, \sigma \rangle \to \sigma' \Rightarrow \langle I_1, \sigma \rangle \overset{*}{\to}_1 \sigma'$
2. Suponemos $\langle I_0; I_1, \sigma \rangle \to \sigma'$
3. Debe haber derivaciones $\langle I_0, \sigma \rangle \to \sigma''$ e $\langle I_1, \sigma'' \rangle \to \sigma'$ para algún σ''.
4. Por la hipótesis concluimos $\langle I_0, \sigma \rangle \overset{*}{\to}_1 \sigma''$ y $\langle I_1, \sigma'' \rangle \overset{*}{\to}_1 \sigma'$
5. Por el Lema de la siguiente lámina $\langle I_0; I_1, \sigma \rangle \overset{*}{\to}_1 \sigma'$

□

Completen los casos inductivos restantes.

UNIVERSIDAD SIMÓN BOLÍVAR

Combinación de Pasos Cortos

Lemma

$\langle l_0; l_1, \sigma \rangle \xrightarrow{*}_1 \sigma'$ *si y sólo si* $\langle l_0, \sigma \rangle \xrightarrow{*}_1 \sigma''$ *y* $\langle l_1, \sigma'' \rangle \xrightarrow{*}_1 \sigma'$

Demostración.

- $(\Rightarrow)$ – por inducción sobre la clausura de la izquierda.
- $(\Leftarrow)$ – por inducción sobre la primera clausura de la derecha.

□

UNIVERSIDAD SIMÓN BOLÍVAR

Consideraciones

- ¿Qué reglas hay que agregar para incorporar expresiones con efectos sobre el estado, como el `i++` de C?
- ¿Qué es necesario agregar para permitir *múltiples* ubicaciones asociadas a un mismo nombre y modelar alcance?
- ¿Qué es necesario agregar para permitir subrutinas?

UNIVERSIDAD SIMÓN BOLÍVAR

Bibliografía

- [*Winskel*]
  - Capítulo 2
  - Ejercicios 2.1 a 2.11

UNIVERSIDAD SIMÓN BOLÍVAR