Harvard School of Engineering and Applied Sciences — CS 152: Programming Languages

Induction; Small-step operational semantics; Large-step operational semantics; IMP Section and Practice Problems

Week 3: Tue Feb 6–Fri Feb 9, 2018

1 Induction

Let's inductively define a set of integers **Quux** with the following inference rules.

$$\textsc{Rule1}\ \frac{}{8 \in \textbf{Quux}} \qquad \textsc{Rule2}\ \frac{}{5 \in \textbf{Quux}} \qquad \textsc{Rule3}\ \frac{a \in \textbf{Quux} \quad b \in \textbf{Quux}}{c \in \textbf{Quux}}\ c = a + b + 1$$

(a) Of the rules above (i.e., RULE1, RULE2, and RULE3), which are axioms and which are inductive rules?

> **Answer:** *The rules* RULE1 *and* RULE2 *are axioms: they have no premises. Rule* RULE3 *is an inductive rule: it has one or more premises.*

(b) Give a derivation showing that 11 is in the set **Quux**.

> **Answer:**
>
> $$\textsc{Rule3}\ \frac{\textsc{Rule2}\ \dfrac{}{5 \in \textit{Quux}} \qquad \textsc{Rule2}\ \dfrac{}{5 \in \textit{Quux}}}{11 \in \textit{Quux}}$$

(c) Give a derivation showing that 20 is in the set **Quux**.

> **Answer:**
>
> $$\textsc{Rule3}\ \frac{\textsc{Rule3}\ \dfrac{\textsc{Rule2}\ \dfrac{}{5 \in \textit{Quux}} \qquad \textsc{Rule2}\ \dfrac{}{5 \in \textit{Quux}}}{11 \in \textit{Quux}} \qquad \textsc{Rule1}\ \dfrac{}{8 \in \textit{Quux}}}{20 \in \textit{Quux}}$$

(d) Write down the inductive reasoning principle for **Quux**. That is, if you wanted to prove that for some property P, for all $a \in$ **Quux** we have $P(a)$, what would you need to show? (See Lecture 3 §2.2 and §2.3.)

> **Answer:** *For any property P,*
> ***If***
>
> - RULE1*: $P(8)$ holds.*
> - RULE2*: $P(5)$ holds.*

- RULE3: *For all $a \in$ **Quux** and all $4 \in$ **Quux**, if $P(a)$ and $P(b)$ then $P(c)$ where $c = a + b + 1$.*

then

*for all $a \in$ **Quux**, $P(a)$ holds.*

(e) Prove that for all $a \in$ **Quux**, there exists $i \in \mathbb{Z}$ such that $a = 3 \times i - 1$.

Make sure that you follow the Recipe for Inductive Proofs! See Lecture 3 §2.5. What set are you inducting on? What is the property you are trying to prove? Go through each case.

Answer: *The property we will prove for all $a \in$ **Quux** is $P(a) = \exists i \in \mathbb{Z}.\ a = 3 \times i - 1$. We proceed by induction on the derivation of $a \in$ **Quux**.*

- RULE1. *Here, $a = 8$. Note that $8 = 3 \times 3 - 1$, and so $P(a)$ holds, as required.*
- RULE2. *Here, $a = 5$. Note that $5 = 3 \times 2 - 1$, and so $P(a)$ holds, as required.*
- RULE3. *Here, $a = b + c + 1$ where $b \in$ **Quux** and $c \in$ **Quux**. Assume that $P(b)$ and $P(c)$. That is, there exists some i and j such that $b = 3 \times i - 1$ and $c = 3 \times j - 1$.*
  We have

$$\begin{aligned} a &= b + c + 1 \\ &= (3 \times i - 1) + (3 \times j - 1) + 1 \\ &= 3 \times (i + j) - 1 \end{aligned}$$

  So there exists an integer k (namely, $k = i + j$) such that $a = 3 \times k - 1$, and so $P(a)$ holds, as required.

(f) Is 2 in the set **Quux**? If so, give a derivation proving it.

Answer: *2 is not in the set **Quux**. How would you go about proving that this is the case? (Hint: could you prove some property that holds true of all elements of **Quux**, and that property isn't true of 2?) Turn page around for an answer... (Whoa, answers inside answers; it's answers all the way down...)*

*Prove that $\forall n \in$ **Quux**. $n > 3$. Since it is not the case that $2 > 3$, we have that $2 \notin$ **Quux**.*

2 Small-step operational semantics

Consider the small-step operational semantics for the language of arithmetic expressions (Lectures 1 and 2). Let σ_0 be a store that maps all program variables to zero.

(a) Show a derivation that $\langle 3 + (5 \times \mathsf{bar}), \sigma_0 \rangle \longrightarrow \langle 3 + (5 \times 0), \sigma_0 \rangle$.

Answer:

$$\textsc{RAdd}\ \dfrac{\textsc{RMul}\ \dfrac{\textsc{Var}\ \dfrac{}{\langle \mathsf{bar}, \sigma_0 \rangle \longrightarrow \langle 0, \sigma_0 \rangle}}{\langle 5 \times \mathsf{bar}, \sigma_0 \rangle \longrightarrow \langle 5 \times 0, \sigma_0 \rangle}}{\langle 3 + (5 \times \mathsf{bar}), \sigma_0 \rangle \longrightarrow \langle 3 + (5 \times 0), \sigma_0 \rangle}$$

(b) What is the sequence of configurations that $\langle \mathsf{foo} := 5;\ (\mathsf{foo} + 2) \times 7, \sigma_0\rangle$ steps to? (You don't need to show the derivations for each step, just show what configuration $\langle \mathsf{foo} := 5;\ (\mathsf{foo} + 2) \times 7, \sigma_0\rangle$ steps to in one step, then two steps, then three steps, and so on, until you reach a final configuration.)

Answer:

$$\begin{array}{lll} & \langle \mathsf{foo} := 5;\ (\mathsf{foo}+2) \times 7 & , \sigma_0 \rangle \\ \longrightarrow & \langle (\mathsf{foo}+2) \times 7 & , \sigma_0[\mathsf{foo} \mapsto 5] \rangle \\ \longrightarrow & \langle (5+2) \times 7 & , \sigma_0[\mathsf{foo} \mapsto 5] \rangle \\ \longrightarrow & \langle 7 \times 7 & , \sigma_0[\mathsf{foo} \mapsto 5] \rangle \\ \longrightarrow & \langle 49 & , \sigma_0[\mathsf{foo} \mapsto 5] \rangle \end{array}$$

(c) Find an integer n and store σ' such that $\langle ((6+(\mathsf{foo} := (\mathsf{bar} := 3; 5); 1+\mathsf{bar}))+\mathsf{bar})\times\mathsf{foo}, \sigma_0\rangle \longrightarrow^* \langle n, \sigma'\rangle$.

Answer: *Let's step through the execution of the configuration, to find a final configuration.*

$$\begin{array}{lll} & \langle ((6 + (\mathsf{foo} := (\mathsf{bar} := 3; 5); 1 + \mathsf{bar})) + \mathsf{bar}) \times \mathsf{foo} & , \sigma_0 \rangle \\ \longrightarrow & \langle ((6 + (\mathsf{foo} := 5; 1 + \mathsf{bar})) + \mathsf{bar}) \times \mathsf{foo} & , \sigma_0[\mathsf{bar} \mapsto 3] \rangle \\ \longrightarrow & \langle ((6 + (1 + \mathsf{bar})) + \mathsf{bar}) \times \mathsf{foo} & , \sigma_0[\mathsf{bar} \mapsto 3, \mathsf{foo} \mapsto 5] \rangle \\ \longrightarrow & \langle ((6 + (1 + 3)) + \mathsf{bar}) \times \mathsf{foo} & , \sigma_0[\mathsf{bar} \mapsto 3, \mathsf{foo} \mapsto 5] \rangle \\ \longrightarrow & \langle ((6 + 4) + \mathsf{bar}) \times \mathsf{foo} & , \sigma_0[\mathsf{bar} \mapsto 3, \mathsf{foo} \mapsto 5] \rangle \\ \longrightarrow & \langle (10 + \mathsf{bar}) \times \mathsf{foo} & , \sigma_0[\mathsf{bar} \mapsto 3, \mathsf{foo} \mapsto 5] \rangle \\ \longrightarrow & \langle (10 + 3) \times \mathsf{foo} & , \sigma_0[\mathsf{bar} \mapsto 3, \mathsf{foo} \mapsto 5] \rangle \\ \longrightarrow & \langle 13 \times \mathsf{foo} & , \sigma_0[\mathsf{bar} \mapsto 3, \mathsf{foo} \mapsto 5] \rangle \\ \longrightarrow & \langle 13 \times 5 & , \sigma_0[\mathsf{bar} \mapsto 3, \mathsf{foo} \mapsto 5] \rangle \\ \longrightarrow & \langle 65 & , \sigma_0[\mathsf{bar} \mapsto 3, \mathsf{foo} \mapsto 5] \rangle \end{array}$$

(d) Is the relation $\longrightarrow$ reflexive? Is it symmetric? Is it anti-symmetric? Is it transitive?

(For each of these questions, if the answer is "no", what is a suitable counterexample? If any of the answers are "yes", think about how you would prove it.)

Answer: *The relation $\longrightarrow$ is not reflexive. A relation R is reflexive if for all x in the domain of R we have $x\ R\ x$. Consider, for example, $\langle 42, \sigma_0\rangle$. It is not the case that $\langle 42, \sigma_0\rangle \longrightarrow \langle 42, \sigma_0\rangle$, and so $\longrightarrow$ is not reflexive.*

The relation $\longrightarrow$ is not symmetric. A relation R is symmetric if for all x, y such that $x\ R\ y$ we have $y\ R\ x$. Consider, for example, $\langle 39 + 3, \sigma_0\rangle$ and $\langle 42, \sigma_0\rangle$. We have $\langle 39 + 3, \sigma_0\rangle \longrightarrow \langle 42, \sigma_0\rangle$ but we do not have $\langle 42, \sigma_0\rangle \longrightarrow \langle 39 + 3, \sigma_0\rangle$. So $\longrightarrow$ is not symmetric.

The relation $\longrightarrow$ is anti-symmetric. A relation R is anti-symmetric if for all distinct x and y we do not have both $x\ R\ y$ and $y\ R\ x$. In our setting, if we have (distinct) configurations $\langle e, \sigma\rangle$ and $\langle e', \sigma'\rangle$ such that $\langle e, \sigma\rangle \longrightarrow \langle e', \sigma'\rangle$, then we do not have that $\langle e', \sigma'\rangle \longrightarrow \langle e, \sigma\rangle$.

Here is one way to prove this. If we did have distinct configurations $\langle e, \sigma\rangle$ and $\langle e', \sigma'\rangle$ such that $\langle e, \sigma\rangle \longrightarrow \langle e', \sigma'\rangle$ and $\langle e', \sigma'\rangle \longrightarrow \langle e, \sigma\rangle$, then we could construct an infinite sequence of small steps:

$$\langle e, \sigma\rangle \longrightarrow \langle e', \sigma'\rangle \longrightarrow \langle e, \sigma\rangle \longrightarrow \langle e', \sigma'\rangle \longrightarrow \langle e, \sigma\rangle \longrightarrow \langle e', \sigma'\rangle \longrightarrow \dots$$

But this would contradict the property that all programs in our language of arithmetic expressions with assignments terminate!

The relation $\longrightarrow$ is not transitive. A relation R is transitive if for all x, y, z, if $x\ R\ y$ and $y\ R\ z$ then $x\ R\ z$. Consider the configurations $\langle (2+3)\times 7, \sigma_0\rangle$ and $\langle 5\times 7, \sigma_0\rangle$ and $\langle 42, \sigma_0\rangle$. We have $\langle (2+3)\times 7, \sigma_0\rangle \longrightarrow \langle 5\times 7, \sigma_0\rangle$ and $\langle 5\times 7, \sigma_0\rangle \longrightarrow \langle 42, \sigma_0\rangle$ but we do not have $\langle (2+3)\times 7, \sigma_0\rangle \longrightarrow \langle 42, \sigma_0\rangle$.

3 Large-step operational semantics

Consider the large-step operational semantics for the language of arithmetic expressions (Lecture 4). Let σ_0 be a store that maps all program variables to zero.

(a) Show a derivation that $\langle 3 + (5\times \mathsf{bar}), \sigma_0\rangle \Downarrow \langle 3, \sigma_0\rangle$.

Answer:

$$\dfrac{\dfrac{}{\langle 3,\sigma_0\rangle \Downarrow \langle 3,\sigma_0\rangle} \qquad \dfrac{\dfrac{}{\langle 5,\sigma_0\rangle\Downarrow\langle 5,\sigma_0\rangle}\qquad \dfrac{}{\langle \mathsf{bar},\sigma_0\rangle\Downarrow\langle 0,\sigma_0\rangle}}{\langle 5\times \mathsf{bar},\sigma_0\rangle \Downarrow \langle 0,\sigma_0\rangle}}{\langle 3+(5\times\mathsf{bar}),\sigma_0\rangle\Downarrow\langle 3,\sigma_0\rangle}$$

(b) Find an integer n and store σ' such that $\langle \mathsf{foo} := 5;\ (\mathsf{foo}+2)\times 7, \sigma_0\rangle \Downarrow \langle n, \sigma'\rangle$.

If you have time and a big piece of paper, give the derivation of $\langle \mathsf{foo} := 5;\ (\mathsf{foo}+2)\times 7, \sigma_0\rangle \Downarrow \langle n, \sigma'\rangle$.

Answer: *We have $\langle \mathsf{foo} := 5;\ (\mathsf{foo}+2)\times 7, \sigma_0\rangle \Downarrow \langle 49, \sigma_0[\mathsf{foo}\mapsto 5]\rangle$.*

In the following derivation, let $\sigma' = \sigma_0[\mathsf{foo}\mapsto 5]$.

$$\dfrac{\dfrac{}{\langle 5,\sigma_0\rangle\Downarrow\langle 5,\sigma_0\rangle}\qquad \dfrac{\dfrac{\dfrac{}{\langle \mathsf{foo},\sigma'\rangle\Downarrow\langle 5,\sigma'\rangle}\qquad\dfrac{}{\langle 2,\sigma'\rangle\Downarrow\langle 2,\sigma'\rangle}}{\langle \mathsf{foo}+2,\sigma'\rangle\Downarrow\langle 7,\sigma'\rangle}\qquad \dfrac{}{\langle 7,\sigma'\rangle\Downarrow\langle 7,\sigma'\rangle}}{\langle (\mathsf{foo}+2)\times 7,\sigma'\rangle\Downarrow\langle 49,\sigma'\rangle}}{\langle \mathsf{foo} := 5;\ (\mathsf{foo}+2)\times 7,\sigma_0\rangle\Downarrow\langle 49,\sigma'\rangle}$$

(c) Is the relation $\Downarrow$ reflexive? Is it symmetric? Is it anti-symmetric? Is it transitive?

(For each of these questions, if the answer is "no", what is a suitable counterexample? If any of the answers are "yes", think about how you would prove it.)

Answer: *The relation $\Downarrow$ is not reflexive. A relation R is reflexive if for all x in the domain of R we have $x\ R\ x$. Consider, for example, $\langle 3+4, \sigma_0\rangle$. It is not the case that $\langle 3+4, \sigma_0\rangle \Downarrow \langle 3+4, \sigma_0\rangle$, and so $\Downarrow$ is not reflexive.*

The relation $\Downarrow$ is not symmetric. A relation R is symmetric if for all x, y such that $x\ R\ y$ we have $y\ R\ x$. Consider, for example, $\langle 39+3, \sigma_0\rangle$ and $\langle 42, \sigma_0\rangle$. We have $\langle 39+3, \sigma_0\rangle \Downarrow \langle 42, \sigma_0\rangle$ but we do not have $\langle 42, \sigma_0\rangle \Downarrow \langle 39+3, \sigma_0\rangle$. So $\Downarrow$ is not symmetric.

The relation $\Downarrow$ is not anti-symmetric. A relation R is anti-symmetric if for all distinct x and y we do not have both $x\ R\ y$ and $y\ R\ x$. In our setting, if we have (distinct) configurations $\langle e, \sigma\rangle$ and $\langle n, \sigma'\rangle$ such that $\langle e, \sigma\rangle \Downarrow \langle n, \sigma'\rangle$ and e' is not an integer, then we do not have that $\langle n, \sigma'\rangle \Downarrow \langle e, \sigma\rangle$.

This can be proven by inspection of the rules, or by induction on the derivation of $\langle e, \sigma\rangle \Downarrow \langle n, \sigma'\rangle$.

The relation $\Downarrow$ is transitive. A relation R is transitive if for all x, y, z, if $x\ R\ y$ and $y\ R\ z$ then $x\ R\ z$. To prove this, suppose that $\langle e, \sigma\rangle \Downarrow \langle e', \sigma'\rangle$ and $\langle e', \sigma'\rangle \Downarrow \langle e'', \sigma''\rangle$. By examination of the rules, we have that e' is an integer. Thus, by the rule INT *we have $\langle e', \sigma'\rangle \Downarrow \langle e', \sigma'\rangle$. Moreover, by the determinism of the arithmetic language (which we discussed in Lecture 2), we have that $e' = e''$ and $\sigma' = \sigma''$. Thus we have that $\langle e, \sigma\rangle \Downarrow \langle e'', \sigma''\rangle$ as required.*

4 IMP

Consider the small-step operational semantics for IMP given in Lecture 5. Let σ_0 be a store that maps all program variables to zero.

(a) Find a configuration $\langle c, \sigma'\rangle$ such that $\langle \textbf{if } 8 < 6 \textbf{ then } \mathsf{foo} := 2 \textbf{ else } \mathsf{bar} := 8, \sigma_0\rangle \longrightarrow \langle c, \sigma'\rangle$ and give a derivation showing that $\langle \textbf{if } 8 < 6 \textbf{ then } \mathsf{foo} := 2 \textbf{ else } \mathsf{bar} := 8, \sigma_0\rangle \longrightarrow \langle c, \sigma'\rangle$.

Answer:

$$\dfrac{\dfrac{}{\langle 8 < 6, \sigma_0\rangle \longrightarrow \langle \textbf{false}, \sigma_0\rangle}}{\langle \textbf{if } 8 < 6 \textbf{ then } \mathsf{foo} := 2 \textbf{ else } \mathsf{bar} := 8, \sigma_0\rangle \longrightarrow \langle \textbf{if false then } \mathsf{foo} := 2 \textbf{ else } \mathsf{bar} := 8, \sigma_0\rangle}$$

(b) What is the sequence of configurations that

$$\langle \mathsf{foo} := \mathsf{bar} + 3; \textbf{if } \mathsf{foo} < \mathsf{bar} \textbf{ then skip else } \mathsf{bar} := 1, \sigma_0\rangle$$

steps to? (You don't need to show the derivations for each step, just show what configuration $\langle \mathsf{foo} := \mathsf{bar} + 3; \textbf{if } \mathsf{foo} < \mathsf{bar} \textbf{ then skip else } \mathsf{bar} := 1, \sigma_0\rangle$ steps to in one step, then two steps, then three steps, and so on, until you reach a final configuration.)

Answer:

$$\begin{array}{ll}
 & \langle \mathsf{foo} := \mathsf{bar} + 3; \textbf{if } \mathsf{foo} < \mathsf{bar} \textbf{ then skip else } \mathsf{bar} := 1, \sigma_0\rangle \\
\longrightarrow & \langle \mathsf{foo} := 0 + 3; \textbf{if } \mathsf{foo} < \mathsf{bar} \textbf{ then skip else } \mathsf{bar} := 1, \sigma_0\rangle \\
\longrightarrow & \langle \mathsf{foo} := 3; \textbf{if } \mathsf{foo} < \mathsf{bar} \textbf{ then skip else } \mathsf{bar} := 1, \sigma_0\rangle \\
\longrightarrow & \langle \textbf{if } \mathsf{foo} < \mathsf{bar} \textbf{ then skip else } \mathsf{bar} := 1, \sigma_0[\mathsf{foo} \mapsto 3]\rangle \\
\longrightarrow & \langle \textbf{if } 3 < \mathsf{bar} \textbf{ then skip else } \mathsf{bar} := 1, \sigma_0[\mathsf{foo} \mapsto 3]\rangle \\
\longrightarrow & \langle \textbf{if } 3 < 0 \textbf{ then skip else } \mathsf{bar} := 1, \sigma_0[\mathsf{foo} \mapsto 3]\rangle \\
\longrightarrow & \langle \textbf{if false then skip else } \mathsf{bar} := 1, \sigma_0[\mathsf{foo} \mapsto 3]\rangle \\
\longrightarrow & \langle \mathsf{bar} := 1, \sigma_0[\mathsf{foo} \mapsto 3]\rangle \\
\longrightarrow & \langle \textbf{skip}, \sigma_0[\mathsf{foo} \mapsto 3, \mathsf{bar} \mapsto 1]\rangle
\end{array}$$

Now consider the large-step operational semantics for IMP given in Lecture 5. Let σ_0 be a store that maps all program variables to zero.

(c) Find a store σ' such that $\langle \textbf{while}\ \mathsf{foo} < 3\ \textbf{do}\ \mathsf{foo} := \mathsf{foo} + 2, \sigma_0\rangle \Downarrow \sigma'$ and give a derivation showing that $\langle \textbf{while}\ \mathsf{foo} < 3\ \textbf{do}\ \mathsf{foo} := \mathsf{foo} + 2, \sigma_0\rangle \Downarrow \sigma'$.

Answer:

In the following, let $\sigma_2 = \sigma_0[\mathsf{foo} \mapsto 2]$ *and* $\sigma_4 = \sigma_0[\mathsf{foo} \mapsto 4]$.

$$\dfrac{\dfrac{\overline{\langle \mathsf{foo}, \sigma_0\rangle \Downarrow 0} \quad \overline{\langle 3, \sigma_0\rangle \Downarrow 3}}{\langle \mathsf{foo} < 3, \sigma_0\rangle \Downarrow \textbf{true}} \qquad \dfrac{\dfrac{\overline{\langle \mathsf{foo}, \sigma_0\rangle \Downarrow 0} \quad \overline{\langle 2, \sigma_0\rangle \Downarrow 2}}{\langle \mathsf{foo} + 2, \sigma_0\rangle \Downarrow 2}}{\langle \mathsf{foo} := \mathsf{foo} + 2, \sigma_0\rangle \Downarrow \sigma_2} \qquad D_1}{\langle \textbf{while}\ \mathsf{foo} < 3\ \textbf{do}\ \mathsf{foo} := \mathsf{foo} + 2, \sigma_0\rangle \Downarrow \sigma_4}$$

where D_1 *is the following derivation*

$$\dfrac{\dfrac{\overline{\langle \mathsf{foo}, \sigma_2\rangle \Downarrow 2} \quad \overline{\langle 3, \sigma_2\rangle \Downarrow 3}}{\langle \mathsf{foo} < 3, \sigma_2\rangle \Downarrow \textbf{true}} \qquad \dfrac{\dfrac{\overline{\langle \mathsf{foo}, \sigma_2\rangle \Downarrow 2} \quad \overline{\langle 2, \sigma_2\rangle \Downarrow 2}}{\langle \mathsf{foo} + 2, \sigma_2\rangle \Downarrow 4}}{\langle \mathsf{foo} := \mathsf{foo} + 2, \sigma_2\rangle \Downarrow \sigma_4} \qquad D_2}{\langle \textbf{while}\ \mathsf{foo} < 3\ \textbf{do}\ \mathsf{foo} := \mathsf{foo} + 2, \sigma_2\rangle \Downarrow \sigma_4}$$

where D_2 *is the following derivation*

$$\dfrac{\dfrac{\overline{\langle \mathsf{foo}, \sigma_2\rangle \Downarrow 4} \quad \overline{\langle 3, \sigma_2\rangle \Downarrow 3}}{\langle \mathsf{foo} < 3, \sigma_2\rangle \Downarrow \textbf{false}}}{\langle \textbf{while}\ \mathsf{foo} < 3\ \textbf{do}\ \mathsf{foo} := \mathsf{foo} + 2, \sigma_4\rangle \Downarrow \sigma_4}$$

(d) Suppose we extend boolean expressions with negation.

$$b ::= \cdots \mid \textbf{not}\ b$$

(i) Give an inference rule or inference rules that show the (large step) evaluation of **not** b.

Answer:

$$\dfrac{\langle b, \sigma\rangle \Downarrow \textbf{false}}{\langle \textbf{not}\ b, \sigma\rangle \Downarrow \textbf{true}} \qquad\qquad \dfrac{\langle b, \sigma\rangle \Downarrow \textbf{true}}{\langle \textbf{not}\ b, \sigma\rangle \Downarrow \textbf{false}}$$

(ii) Show that **if** b **then** c_1 **else** c_2 is equivalent to **if not** b **then** c_2 **else** c_1. (See Lecture 5.)

Answer: **if** b **then** c_1 **else** c_2 *is equivalent to* **if not** b **then** c_2 **else** c_1 *if for all stores* σ *and* σ', *we have*

$$\langle \textbf{if}\ b\ \textbf{then}\ c_1\ \textbf{else}\ c_2, \sigma\rangle \Downarrow \sigma' \text{ if and only if } \langle \textbf{if not}\ b\ \textbf{then}\ c_2\ \textbf{else}\ c_1, \sigma\rangle \Downarrow \sigma'$$

Let's show the forward direction. Suppose we have σ and σ' and $\langle \textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2, \sigma\rangle \Downarrow \sigma'$. We need to show that $\langle \textbf{if not } b \textbf{ then } c_2 \textbf{ else } c_1, \sigma\rangle \Downarrow \sigma'$.
Because $\langle \textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2, \sigma\rangle \Downarrow \sigma'$, there is a finite derivation whose conclusion is $\langle \textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2, \sigma\rangle \Downarrow \sigma'$. Let's think about what inference rules could have been used to conclude $\langle \textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2, \sigma\rangle \Downarrow \sigma'$. There are only two possibilities: the rule for conditionals where the boolean expression b evaluates to **true***, and the rule for conditionals where the boolean condition b evaluates to* **false***. That is, the derivation of $\langle \textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2, \sigma\rangle \Downarrow \sigma'$ has one of the following two forms.*

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma\rangle \Downarrow \textbf{true}} \qquad \dfrac{\vdots}{\langle c_1, \sigma\rangle \Downarrow \sigma'}}{\langle \textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2, \sigma\rangle \Downarrow \sigma'} \qquad \dfrac{\dfrac{\vdots}{\langle b, \sigma\rangle \Downarrow \textbf{false}} \qquad \dfrac{\vdots}{\langle c_2, \sigma\rangle \Downarrow \sigma'}}{\langle \textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2, \sigma\rangle \Downarrow \sigma'}$$

Let's consider these two cases in turn. Suppose that $\langle b, \sigma\rangle \Downarrow$ **true***. Then we can reuse the derivations $\frac{\vdots}{\langle b, \sigma\rangle \Downarrow \textbf{true}}$ and $\frac{\vdots}{\langle c_1, \sigma\rangle \Downarrow \sigma'}$ to construct the following proof tree, showing that $\langle \textbf{if not } b \textbf{ then } c_2 \textbf{ else } c_1, \sigma\rangle \Downarrow \sigma'$.*

$$\dfrac{\dfrac{\dfrac{\vdots}{\langle b, \sigma\rangle \Downarrow \textbf{true}}}{\langle \textbf{not } b, \sigma\rangle \Downarrow \textbf{false}} \qquad \dfrac{\vdots}{\langle c_1, \sigma\rangle \Downarrow \sigma'}}{\langle \textbf{if not } b \textbf{ then } c_2 \textbf{ else } c_1, \sigma\rangle \Downarrow \sigma'}$$

Now consider the other case, where $\langle b, \sigma\rangle \Downarrow$ **false***. Then we can reuse the derivations $\frac{\vdots}{\langle b, \sigma\rangle \Downarrow \textbf{false}}$ and $\frac{\vdots}{\langle c_2, \sigma\rangle \Downarrow \sigma'}$ to construct the following proof tree, showing that $\langle \textbf{if not } b \textbf{ then } c_2 \textbf{ else } c_1, \sigma\rangle \Downarrow \sigma'$.*

$$\dfrac{\dfrac{\dfrac{\vdots}{\langle b, \sigma\rangle \Downarrow \textbf{false}}}{\langle \textbf{not } b, \sigma\rangle \Downarrow \textbf{true}} \qquad \dfrac{\vdots}{\langle c_2, \sigma\rangle \Downarrow \sigma'}}{\langle \textbf{if not } b \textbf{ then } c_2 \textbf{ else } c_1, \sigma\rangle \Downarrow \sigma'}$$

The reverse direction is almost exactly the same. Suppose we have σ and σ' and $\langle \textbf{if not } b \textbf{ then } c_2 \textbf{ else } c_1, \sigma\rangle \Downarrow \sigma'$. We need to show that $\langle \textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2, \sigma\rangle \Downarrow \sigma'$.
Because $\langle \textbf{if not } b \textbf{ then } c_2 \textbf{ else } c_1, \sigma\rangle \Downarrow \sigma'$, there is a finite derivation whose conclusion is $\langle \textbf{if not } b \textbf{ then } c_2 \textbf{ else } c_1, \sigma\rangle \Downarrow \sigma'$. Let's think about what inference rules could have been used to conclude $\langle \textbf{if not } b \textbf{ then } c_2 \textbf{ else } c_1, \sigma\rangle \Downarrow \sigma'$. There are only two possibilities: the rule for conditionals where the boolean expression **not** *b evaluates to* **true***, and the rule for conditionals where the boolean condition* **not** *b evaluates to* **false***. That is, the derivation of $\langle \textbf{if not } b \textbf{ then } c_2 \textbf{ else } c_1, \sigma\rangle \Downarrow \sigma'$ has one of the following two forms.*

$$\dfrac{\dfrac{\dfrac{\vdots}{\langle b, \sigma\rangle \Downarrow \textbf{true}}}{\langle \textbf{not } b, \sigma\rangle \Downarrow \textbf{false}} \qquad \dfrac{\vdots}{\langle c_1, \sigma\rangle \Downarrow \sigma'}}{\langle \textbf{if not } b \textbf{ then } c_2 \textbf{ else } c_1, \sigma\rangle \Downarrow \sigma'} \qquad \dfrac{\dfrac{\dfrac{\vdots}{\langle b, \sigma\rangle \Downarrow \textbf{false}}}{\langle \textbf{not } b, \sigma\rangle \Downarrow \textbf{true}} \qquad \dfrac{\vdots}{\langle c_2, \sigma\rangle \Downarrow \sigma'}}{\langle \textbf{if not } b \textbf{ then } c_2 \textbf{ else } c_1, \sigma\rangle \Downarrow \sigma'}$$

Let's consider these two cases in turn. Suppose that $\langle \textbf{not } b, \sigma\rangle \Downarrow$ **false***. Then we can reuse the derivations $\frac{\vdots}{\langle b, \sigma\rangle \Downarrow \textbf{true}}$ and $\frac{\vdots}{\langle c_1, \sigma\rangle \Downarrow \sigma'}$ to construct the following proof tree, showing that $\langle \textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2, \sigma\rangle \Downarrow \sigma'$.*

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma\rangle \Downarrow \textsf{\textbf{true}}} \qquad \dfrac{\vdots}{\langle c_1, \sigma\rangle \Downarrow \sigma'}}{\langle \textsf{\textbf{if}}\ b\ \textsf{\textbf{then}}\ c_1\ \textsf{\textbf{else}}\ c_2, \sigma\rangle \Downarrow \sigma'}$$

Now consider the other case, where $\langle \textsf{\textbf{not}}\ b, \sigma\rangle \Downarrow \textsf{\textbf{true}}$. *Then we can reuse the derivations* $\dfrac{\vdots}{\langle b, \sigma\rangle \Downarrow \textsf{\textbf{false}}}$ *and* $\dfrac{\vdots}{\langle c_2, \sigma\rangle \Downarrow \sigma'}$ *to construct the following proof tree, showing that* $\langle \textsf{\textbf{if}}\ b\ \textsf{\textbf{then}}\ c_1\ \textsf{\textbf{else}}\ c_2, \sigma\rangle \Downarrow \sigma'$.

$$\dfrac{\dfrac{\vdots}{\langle b, \sigma\rangle \Downarrow \textsf{\textbf{false}}} \qquad \dfrac{\vdots}{\langle c_2, \sigma\rangle \Downarrow \sigma'}}{\langle \textsf{\textbf{if}}\ b\ \textsf{\textbf{then}}\ c_1\ \textsf{\textbf{else}}\ c_2, \sigma\rangle \Downarrow \sigma'}$$