Structural Induction

CS 6371: Advanced Programming Languages

Kevin W. Hamlen

February 1, 2024

Derivational Proofs

- Formal (rule-based) definitions of programming languages create a foundation for mathematical proofs of correctness for real programs.
  - essential for assuring real-world, mission-critical systems
  - superior to unit testing (which has low coverage for most real systems)
  - basis for machine-checked formal methods verification
- In this class when I ask you to "prove" a property of a SIMPL program:
  - You may assume (without proof) basic facts about math (e.g., $n + 1 > n$).
  - But you **may not assume** basic facts about how SIMPL programs behave (e.g., you must prove using a derivation that $3 + 1$ returns the integer 4).

$$\dfrac{\dfrac{}{\langle 3, \sigma\rangle \Downarrow 3}(13) \qquad \dfrac{}{\langle 1, \sigma\rangle \Downarrow 1}(13)}{\langle 3 + 1, \sigma\rangle \Downarrow 4}(15)$$

Induction over $\mathbb{N}$

- To prove that a property P holds for all natural numbers $n \in \mathbb{N}$, you typically use induction:
  - Weak induction (over $\mathbb{N}$):
    1. Prove that $P(0)$ holds (called the **base case**).
    2. For arbitrary $n \geq 1$, prove $P(n-1) \Rightarrow P(n)$. (Assumption $P(n-1)$ called the **(weak) inductive hypothesis (IH)**.)
  - Strong induction (over $\mathbb{N}$):

    For arbitrary $n \in \mathbb{N}$, prove $(\forall n_0 < n, P(n_0)) \Rightarrow P(n)$. (Assumption $\forall n_0 < n, P(n_0)$ called the **(strong) IH**).

    Usually this divides into two cases:
    1. $n = 0$ (prove $P(0)$ without IH)
    2. $n \geq 1$ (prove $P(n)$ using IH)
- These inductive principles are actually just special cases of a much more general inductive principle called **structural induction**.

Structural Induction

Goal: Prove that a property P holds for all **derivations** $\mathcal{D}$.

Proof by **structural induction** uses two steps:

1. Base case(s): Prove that $P(\mathcal{D})$ holds for "minimal" derivation(s) $\mathcal{D}$.
2. Inductive case(s): Assume (IH) that $P(\mathcal{D}_0)$ holds for all derivations $\mathcal{D}_0$ "smaller than" $\mathcal{D}$, and prove $P(\mathcal{D})$.

What do "minimal" and "smaller than" mean for derivations?

- Any sensible definition will do, but we will use tree height as our metric.
- $\mathcal{D}$ is "minimal" if it has only one rule application.
- $\mathcal{D}_1 < \mathcal{D}_2$ if $\mathcal{D}_1$ has height strictly less than $\mathcal{D}_2$.
- Note: Size of judgments within the rules is irrelevant; only tree height matters.

Generalizing Strong Induction

Why is structural induction a *generalization* of inductions over $\mathbb{N}$?

- Natural numbers are actually primitive linear structures.
  - Zero is primitive (base case).
  - One is the successor of zero $S(0)$.
  - Two is the successor of one $S(S(0))$.
  - Three is the successor of two $S(S(S(0)))$.
  - n is the nth-successor of zero $S^n(0)$.
- The inductions you learned in discrete class work for linear structures, but derivations are trees so we generalize to arbitrary-arity structures.
- (This isn't the end of the story. We'll be learning even more powerful induction principles later in the course!)

Example Proof by Structural Induction

To illustrate, let's prove the following example theorem by structural induction:

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

May seem like a trivial theorem, but this theorem is not true of some programming languages. Can you think of an example of a language for which it's not true?

Setting up the Induction

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Let $\mathcal{D}$ be a derivation of judgment $\langle c, \sigma \rangle \Downarrow \sigma'$. We will prove the theorem by structural induction over $\mathcal{D}$. ...

- Always tell me which kind of induction you're doing!
  - We will learn many, and all are on the table.
  - Sometimes you'll use more than one kind in the same proof.
  - If your proof falls apart, at least telling me up front which kind you're attempting will save you many points!
- Always tell me what your induction is over!
  - If it's a structural induction, what's the structure?
  - There will often be many choices, only one of which works!
- How do we know $\mathcal{D}$ exists?
  - The definition of $\langle c, \sigma \rangle \Downarrow \sigma'$ being "true" (assumed) is that it is derivable.
  - May not assume the derivation is unique! (But at least one exists.)

Base Case

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

...

Base Case: *Suppose $\mathcal{D}$ consists of only one rule. Then it must be Rule 1, so $\mathcal{D}$ must look like this:*

$$\mathcal{D} = \frac{}{\langle \texttt{skip}, \sigma \rangle \Downarrow \sigma}(1)$$

- Extremely important: Each case is defined by the final rule number in derivation $\mathcal{D}$ (in this case Rule 1).
  - **Do not** write "Suppose $c = \texttt{skip}$. ..." That is **not** a case of a structural induction over $\mathcal{D}$. It is instead a case of a structural induction over c!
  - c is also a structure. You could do a structural induction over it. It will not work. (Will show you why later.)
  - If you make this mistake, 83% of your proof will work and then you'll get stuck at the end.

Base Case

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

...

Base Case: *Suppose $\mathcal{D}$ consists of only one rule. Then it must be Rule 1, so $\mathcal{D}$ must look like this:*

$$\mathcal{D} = \frac{}{\langle \texttt{skip}, \sigma \rangle \Downarrow \sigma}(1)$$

We infer that $\sigma' = \sigma$. Since $\sigma(x) = n$ (by assumption), we conclude that $\sigma'(x) = n$.

- Extremely important: Each case is defined by the final rule number in derivation $\mathcal{D}$ (in this case Rule 1).
  - Conclusions like $c = \texttt{skip}$ and $\sigma' = \sigma$ follow from the assumption that $\mathcal{D}$ ends in Rule 1, not the other way around.

Inductive Hypothesis

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if* $\sigma_0(x_0) = n_0$, *and* $\langle c_0, \sigma_0 \rangle \Downarrow \sigma_0'$ *has a derivation* $\mathcal{D}_0 < \mathcal{D}$, *and* x_0 *does not appear in* c_0, *then* $\sigma_0'(x_0) = n_0$.

- Always write out the IH **in full** before proving any inductive cases.
- The IH must be exactly the following:
  - the original theorem statement (verbatim) with all variables renamed (I add a subscript zero)
  - an extra assumption that the structure being inducted over is "smaller" ($\mathcal{D}_0 < \mathcal{D}$)
- Resist the urge to skip this seemingly boring step! Resist the urge to rephrase the theorem! It will get you into trouble!

Inductive Case

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if* $\sigma_0(x_0) = n_0$, *and* $\langle c_0, \sigma_0 \rangle \Downarrow \sigma_0'$ *has a derivation* $\mathcal{D}_0 < \mathcal{D}$, *and* x_0 *does not appear in* c_0, *then* $\sigma_0'(x_0) = n_0$.

Case 2: *Suppose* $\mathcal{D}$ *ends in Rule 2. Then* $\mathcal{D}$ *looks like this:*

$$\mathcal{D} = \dfrac{\dfrac{\mathcal{D}_1}{\langle c_1, \sigma \rangle \Downarrow \sigma_2} \qquad \dfrac{\mathcal{D}_2}{\langle c_2, \sigma_2 \rangle \Downarrow \sigma'}}{\langle c_1 ; c_2, \sigma \rangle \Downarrow \sigma'} (2)$$

Inductive Case

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if* $\sigma_0(x_0) = n_0$, *and* $\langle c_0, \sigma_0 \rangle \Downarrow \sigma_0'$ *has a derivation* $\mathcal{D}_0 < \mathcal{D}$, *and* x_0 *does not appear in* c_0, *then* $\sigma_0'(x_0) = n_0$.

Case 2: *Suppose* $\mathcal{D}$ *ends in Rule 2. Then* $\mathcal{D}$ *looks like this:*

$$\mathcal{D} = \frac{\dfrac{\mathcal{D}_1}{\langle c_1, \sigma \rangle \Downarrow \sigma_2} \qquad \dfrac{\mathcal{D}_2}{\langle c_2, \sigma_2 \rangle \Downarrow \sigma'}}{\langle c_1 ; c_2, \sigma \rangle \Downarrow \sigma'} (2)$$

So $c = c_1 ; c_2$.

Now what?

Inductive Case

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if* $\sigma_0(x_0) = n_0$, *and* $\langle c_0, \sigma_0 \rangle \Downarrow \sigma_0'$ *has a derivation* $\mathcal{D}_0 < \mathcal{D}$, *and* x_0 *does not appear in* c_0, *then* $\sigma_0'(x_0) = n_0$.

Case 2: *Suppose* $\mathcal{D}$ *ends in Rule 2. Then* $\mathcal{D}$ *looks like this:*

$$\mathcal{D} = \frac{\dfrac{\mathcal{D}_1}{\langle c_1, \sigma \rangle \Downarrow \sigma_2} \qquad \dfrac{\mathcal{D}_2}{\langle c_2, \sigma_2 \rangle \Downarrow \sigma'}}{\langle c_1 ; c_2, \sigma \rangle \Downarrow \sigma'} (2)$$

So $c = c_1 ; c_2$. *Apply IH with* $\mathcal{D}_0 = \mathcal{D}_1$, $c_0 = c_1$, $\sigma_0 = \sigma$, $\sigma_0' = \sigma_2$, $x_0 = x$, *and* $n_0 = n$.

- $\sigma_0(x_0) = n_0$ *because* $\sigma(x) = n$ *(assumption)*
- $\mathcal{D}_0 < \mathcal{D}$ *because* $\mathcal{D}_1$ *is inside* $\mathcal{D}$
- x_0 *not in* c_0 *because* c_1 *is part of* c *and* x *not in* c *(assumption)*

From IH, we conclude that $\sigma_2(x) = n$.

- Give explicit instantiations of IH variables.
- Explicitly prove all IH assumptions.

Inductive Case

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if* $\sigma_0(x_0) = n_0$, *and* $\langle c_0, \sigma_0 \rangle \Downarrow \sigma_0'$ *has a derivation* $\mathcal{D}_0 < \mathcal{D}$, *and* x_0 *does not appear in* c_0, *then* $\sigma_0'(x_0) = n_0$.

Case 2: *Suppose* $\mathcal{D}$ *ends in Rule 2. Then* $\mathcal{D}$ *looks like this:*

$$\mathcal{D} = \frac{\dfrac{\mathcal{D}_1}{\langle c_1, \sigma \rangle \Downarrow \sigma_2} \qquad \dfrac{\mathcal{D}_2}{\langle c_2, \sigma_2 \rangle \Downarrow \sigma'}}{\langle c_1 ; c_2, \sigma \rangle \Downarrow \sigma'} (2)$$

So $c = c_1 ; c_2$. *Apply IH with* $\mathcal{D}_0 = \mathcal{D}_1$ *... concluding that* $\sigma_2(x) = n$.

Apply IH with $\mathcal{D}_0 = \mathcal{D}_2$, $c_0 = c_2$, $\sigma_0 = \sigma_2$, $\sigma_0' = \sigma'$, $x_0 = x$, *and* $n_0 = n$.

- $\sigma_0(x_0) = n_0$ *because* $\sigma_2(x) = n$ *(proved above)*
- $\mathcal{D}_0 < \mathcal{D}$ *because* $\mathcal{D}_2$ *is inside* $\mathcal{D}$
- x_0 *not in* c_0 *because* c_2 *is part of* c *and* x *not in* c *(assumption)*

From IH, we conclude that $\sigma'(x) = n$.

Assignment Case

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if* $\sigma_0(x_0) = n_0$, *and* $\langle c_0, \sigma_0 \rangle \Downarrow \sigma_0'$ *has a derivation* $\mathcal{D}_0 < \mathcal{D}$, *and* x_0 *does not appear in* c_0, *then* $\sigma_0'(x_0) = n_0$.

Case 3: *Suppose* $\mathcal{D}$ *ends in Rule 3. Then* $\mathcal{D}$ *looks like this:*

$$\mathcal{D} = \frac{\dfrac{\mathcal{D}_1}{\langle a, \sigma \rangle \Downarrow i}}{\langle v := a, \sigma \rangle \Downarrow \sigma[v \mapsto i]} (3)$$

So $c = (v := a)$ and $\sigma' = \sigma[v \mapsto i]$.

What now?

Assignment Case

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if* $\sigma_0(x_0) = n_0$, *and* $\langle c_0, \sigma_0 \rangle \Downarrow \sigma_0'$ *has a derivation* $\mathcal{D}_0 < \mathcal{D}$, *and* x_0 *does not appear in* c_0, *then* $\sigma_0'(x_0) = n_0$.

Case 3: *Suppose* $\mathcal{D}$ *ends in Rule 3. Then* $\mathcal{D}$ *looks like this:*

$$\mathcal{D} = \frac{\dfrac{\mathcal{D}_1}{\langle a, \sigma \rangle \Downarrow i}}{\langle v := a, \sigma \rangle \Downarrow \sigma[v \mapsto i]} (3)$$

So $c = (v := a)$ and $\sigma' = \sigma[v \mapsto i]$. Apply IH with $\mathcal{D}_0 = \mathcal{D}_1$, $c_0 = a$, $\sigma_0 = \sigma$, $\sigma_0' = i$, $x_0 = x$, and $n_0 = n$.

Assignment Case

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if* $\sigma_0(x_0) = n_0$, *and* $\langle c_0, \sigma_0 \rangle \Downarrow \sigma_0'$ *has a derivation* $\mathcal{D}_0 < \mathcal{D}$, *and* x_0 *does not appear in* c_0, *then* $\sigma_0'(x_0) = n_0$.

Case 3: *Suppose* $\mathcal{D}$ *ends in Rule 3. Then* $\mathcal{D}$ *looks like this:*

$$\mathcal{D} = \frac{\dfrac{\mathcal{D}_1}{\langle a, \sigma \rangle \Downarrow i}}{\langle v := a, \sigma \rangle \Downarrow \sigma[v \mapsto i]}\,(3)$$

So $c = (v := a)$ and $\sigma' = \sigma[v \mapsto i]$. Apply IH with $\mathcal{D}_0 = \mathcal{D}_1$, $c_0 = a$, $\sigma_0 = \sigma$, $\sigma_0' = i$, $x_0 = x$, and $n_0 = n$.

What? A command equals an arithmetic expression? A store equals an integer? This makes no sense!

Don't blindly copy a template. Need to understand *why* each step makes mathematical sense and is needed.

Assignment Case

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if* $\sigma_0(x_0) = n_0$, *and* $\langle c_0, \sigma_0 \rangle \Downarrow \sigma_0'$ *has a derivation* $\mathcal{D}_0 < \mathcal{D}$, *and* x_0 *does not appear in* c_0, *then* $\sigma_0'(x_0) = n_0$.

Case 3: *Suppose* $\mathcal{D}$ *ends in Rule 3. Then* $\mathcal{D}$ *looks like this:*

$$\mathcal{D} = \frac{\dfrac{\mathcal{D}_1}{\langle a, \sigma \rangle \Downarrow i}}{\langle v := a, \sigma \rangle \Downarrow \sigma[v \mapsto i]} (3)$$

So $c = (v := a)$ and $\sigma' = \sigma[v \mapsto i]$.

Let's try this again...

Assignment Case

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if* $\sigma_0(x_0) = n_0$, *and* $\langle c_0, \sigma_0 \rangle \Downarrow \sigma_0'$ *has a derivation* $\mathcal{D}_0 < \mathcal{D}$, *and* x_0 *does not appear in* c_0, *then* $\sigma_0'(x_0) = n_0$.

Case 3: *Suppose* $\mathcal{D}$ *ends in Rule 3. Then* $\mathcal{D}$ *looks like this:*

$$\mathcal{D} = \frac{\dfrac{\mathcal{D}_1}{\langle a, \sigma \rangle \Downarrow i}}{\langle v := a, \sigma \rangle \Downarrow \sigma[v \mapsto i]} \quad (3)$$

So $c = (v := a)$ and $\sigma' = \sigma[v \mapsto i]$. Since x not in c (by assumption), v is not x. Therefore $\sigma[v \mapsto i](x) = \sigma(x)$. Since $\sigma(x) = n$ (by assumption), we conclude that $\sigma'(x) = n$.

Assignment Case

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma \rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if $\sigma_0(x_0) = n_0$, and $\langle c_0, \sigma_0 \rangle \Downarrow \sigma_0'$ has a derivation $\mathcal{D}_0 < \mathcal{D}$, and x_0 does not appear in c_0, then $\sigma_0'(x_0) = n_0$.*

Case 3: *Suppose $\mathcal{D}$ ends in Rule 3. Then $\mathcal{D}$ looks like this:*

$$\mathcal{D} = \frac{\dfrac{\mathcal{D}_1}{\langle a, \sigma \rangle \Downarrow i}}{\langle v := a, \sigma \rangle \Downarrow \sigma[v \mapsto i]} (3)$$

So $c = (v := a)$ and $\sigma' = \sigma[v \mapsto i]$. Since x not in c (by assumption), v is not x. Therefore $\sigma[v \mapsto i](x) = \sigma(x)$. Since $\sigma(x) = n$ (by assumption), we conclude that $\sigma'(x) = n$.

Note: We never used the IH in this case. This case is actually a base case!

In general, don't worry about which cases are base cases and which cases are inductive. Just state the IH before proving any cases and use it as needed.

Cases for Conditionals

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma\rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Case 4: *Suppose $\mathcal{D}$ ends in Rule 4. Then $\mathcal{D}$ looks like this:*

$$\mathcal{D} = \frac{\dfrac{\mathcal{D}_1}{\langle b, \sigma\rangle \Downarrow T} \qquad \dfrac{\mathcal{D}_2}{\langle c_1, \sigma\rangle \Downarrow \sigma'}}{\langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow \sigma'}\,(4)$$

...

Case 5: *Suppose $\mathcal{D}$ ends in Rule 5. Then $\mathcal{D}$ looks like this:*

$$\mathcal{D} = \frac{\dfrac{\mathcal{D}_1}{\langle b, \sigma\rangle \Downarrow F} \qquad \dfrac{\mathcal{D}_2}{\langle c_2, \sigma\rangle \Downarrow \sigma'}}{\langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow \sigma'}\,(5)$$

...

Exercise: See if you can solve these cases on your own. (Online lecture notes give solutions.)

Case for While-loop

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma\rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if* $\sigma_0(x_0) = n_0$, *and* $\langle c_0, \sigma_0\rangle \Downarrow \sigma_0'$ *has a derivation* $\mathcal{D}_0 < \mathcal{D}$, *and* x_0 *does not appear in* c_0, *then* $\sigma_0'(x_0) = n_0$.

Case 6: *Suppose* $\mathcal{D}$ *ends in Rule 6. Then* $\mathcal{D}$ *looks like this:*

$$\mathcal{D} = \frac{\dfrac{\mathcal{D}_1}{\langle \texttt{if } b \texttt{ then } (c_1 \texttt{; while } b \texttt{ do } c_1) \texttt{ else skip}, \sigma\rangle \Downarrow \sigma'}}{\langle \texttt{while } b \texttt{ do } c_1, \sigma\rangle \Downarrow \sigma'} \quad (6)$$

So $c = \texttt{while } b \texttt{ do } c_1$.

Case for While-loop

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma\rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if* $\sigma_0(x_0) = n_0$, *and* $\langle c_0, \sigma_0\rangle \Downarrow \sigma_0'$ *has a derivation* $\mathcal{D}_0 < \mathcal{D}$, *and* x_0 *does not appear in* c_0, *then* $\sigma_0'(x_0) = n_0$.

Case 6: *Suppose* $\mathcal{D}$ *ends in Rule 6. Then* $\mathcal{D}$ *looks like this:*

$$\mathcal{D} = \frac{\dfrac{\mathcal{D}_1}{\langle \texttt{if } b \texttt{ then } (c_1 \texttt{;while } b \texttt{ do } c_1) \texttt{ else skip}, \sigma\rangle \Downarrow \sigma'}}{\langle \texttt{while } b \texttt{ do } c_1, \sigma\rangle \Downarrow \sigma'} \quad (6)$$

So $c = \texttt{while } b \texttt{ do } c_1$. Apply IH with $\mathcal{D}_0 = \mathcal{D}_1$, $c_0 = \texttt{if } b \texttt{ then } (c_1 \texttt{;while } b \texttt{ do } c_1) \texttt{ else skip}$, $\sigma_0 = \sigma$, $\sigma_0' = \sigma'$, $x_0 = x$, and $n_0 = n$.

- $\sigma_0(x_0) = n_0$ because $\sigma(x) = n$ (by assumption)
- $\mathcal{D}_0 < \mathcal{D}$ because $\mathcal{D}_1$ is inside $\mathcal{D}$
- x_0 not in c_0 because the only variables in c_0 are in b and c_1, which are parts of c, and x not in c by assumption

Case for While-loop

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma\rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if* $\sigma_0(x_0) = n_0$, *and* $\langle c_0, \sigma_0\rangle \Downarrow \sigma_0'$ *has a derivation* $\mathcal{D}_0 < \mathcal{D}$, *and* x_0 *does not appear in* c_0, *then* $\sigma_0'(x_0) = n_0$.

Case 6: *Suppose* $\mathcal{D}$ *ends in Rule 6. Then* $\mathcal{D}$ *looks like this:*

$$\mathcal{D} = \frac{\begin{array}{c}\mathcal{D}_1\\ \langle \texttt{if } b \texttt{ then } (c_1 \texttt{;while } b \texttt{ do } c_1) \texttt{ else skip}, \sigma\rangle \Downarrow \sigma'\end{array}}{\langle \texttt{while } b \texttt{ do } c_1, \sigma\rangle \Downarrow \sigma'} \quad (6)$$

So $c = \texttt{while } b \texttt{ do } c_1$. Apply IH with $\mathcal{D}_0 = \mathcal{D}_1$, $c_0 = \texttt{if } b \texttt{ then } (c_1 \texttt{;while } b \texttt{ do } c_1) \texttt{ else skip}$, $\sigma_0 = \sigma$, $\sigma_0' = \sigma'$, $x_0 = x$, and $n_0 = n$.

- $\sigma_0(x_0) = n_0$ because $\sigma(x) = n$ (by assumption)
- $\mathcal{D}_0 < \mathcal{D}$ because $\mathcal{D}_1$ is inside $\mathcal{D}$
- x_0 not in c_0 because the only variables in c_0 are in b and c_1, which are parts of c, and x not in c by assumption

From IH, we conclude that $\sigma'(x) = n$. □

Case for While-loop

Theorem:

If $\sigma(x) = n$ and $\langle c, \sigma\rangle \Downarrow \sigma'$ and x does not appear in c, then $\sigma'(x) = n$.

Proof:

Inductive Hypothesis: *Assume that if* $\sigma_0(x_0) = n_0$, *and* $\langle c_0, \sigma_0\rangle \Downarrow \sigma_0'$ *has a derivation* $\mathcal{D}_0 < \mathcal{D}$, *and* x_0 *does not appear in* c_0, *then* $\sigma_0'(x_0) = n_0$.

Case 6: *Suppose* $\mathcal{D}$ *ends in Rule 6. Then* $\mathcal{D}$ *looks like this:*

$$\mathcal{D} = \frac{\dfrac{\mathcal{D}_1}{\langle \texttt{if } b \texttt{ then } (c_1 \texttt{;while } b \texttt{ do } c_1) \texttt{ else skip}, \sigma\rangle \Downarrow \sigma'}}{\langle \texttt{while } b \texttt{ do } c_1, \sigma\rangle \Downarrow \sigma'} \quad (6)$$

So $c = \texttt{while } b \texttt{ do } c_1$. Apply IH with $\mathcal{D}_0 = \mathcal{D}_1$, $c_0 = \texttt{if } b \texttt{ then } (c_1 \texttt{;while } b \texttt{ do } c_1) \texttt{ else skip}$, $\sigma_0 = \sigma$, $\sigma_0' = \sigma'$, $x_0 = x$, and $n_0 = n$.

- $\sigma_0(x_0) = n_0$ because $\sigma(x) = n$ (by assumption)
- $\mathcal{D}_0 < \mathcal{D}$ because $\mathcal{D}_1$ is inside $\mathcal{D}$
- x_0 not in c_0 because the only variables in c_0 are in b and c_1, which are parts of c, and x not in c by assumption

From IH, we conclude that $\sigma'(x) = n$. □

What would have failed if we'd done a structural induction on c instead of $\mathcal{D}$?

Induction Failures

- Induction is a principled way to perform loop-like reasoning *without accidentally making a circular argument*.
  - Inductive case argues $\left(\forall \mathcal{D}_0 < \mathcal{D} \,.\, P(\mathcal{D}_0)\right) \Rightarrow \mathcal{D}$
  - Never argues $P(\mathcal{D}) \Rightarrow P(\mathcal{D})$!
- Signs that your induction is incorrect:
  - You find yourself arguing that some computation "keeps going until..." or "eventually" does something.
  - The proof doesn't use the inductive hypothesis literally.
  - A case is argued correct because some other case is proved. (Invalid reasoning since all cases start with mutually exclusive assumptions!)
- Proof failures are better than invalid proof successes!
  - If your approach to getting stuck is to force the proof to succeed through hand-waving, then your "proofs" are not supplying additional assurance.
  - Inability to prove something is not a disaster. "Proving" something that's false is a disaster!

Generalizing the Theorem

- Inductive proofs sometimes fail when the IH cannot be applied.
  - Simple example: Theorem has the form $P_1(\mathcal{D}) \Rightarrow P_2(\mathcal{D})$, but assumption $P_1(\mathcal{D})$ is unnecessary.
  - IH will look like: $\forall \mathcal{D}_0 < \mathcal{D} \,.\, P_1(\mathcal{D}_0) \Rightarrow P_2(\mathcal{D}_0)$
  - At some point you need to apply the IH, but $P_1(\mathcal{D}_0)$ is false. Proof fails.
- Solution is often to "generalize" the theorem:
  - Prove a different theorem P' that generalizes (i.e., implies) the original theorem P.
  - Example: Drop assumption P_1 from the theorem above.
  - Even though P' seems harder to prove (e.g., fewer assumptions), you get a stronger IH (e.g., fewer prerequisites).
  - After proving P' by induction, separately prove that $P' \Rightarrow P$.
- Mathematicians colloquially refer to this as "turning the crank of the induction."
  - Finding the right generalization of the theorem is often the central challenge for formal verification of real-world software in practice.

Proof Without Induction

Sometimes it's possible to avoid structural induction entirely by doing some creative derivation "copy and pasting". Here's an example:

Theorem:

The judgment $\langle \texttt{if } !b \texttt{ then } c_1 \texttt{ else } c_2, \sigma \rangle \Downarrow \sigma'$ *holds if and only if the judgment* $\langle \texttt{if } b \texttt{ then } c_2 \texttt{ else } c_1, \sigma \rangle \Downarrow \sigma'$ *holds.*

(This is the sort of theorem one must prove in order to implement safe compiler optimizations for a language.)

Proof Without Induction

Theorem:

The judgment $\langle \texttt{if } !b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow \sigma'$ *holds if and only if the judgment* $\langle \texttt{if } b \texttt{ then } c_2 \texttt{ else } c_1, \sigma\rangle \Downarrow \sigma'$ *holds.*

Proof:

We first prove the forward implication. Assume judgment $\langle \texttt{if } !b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow \sigma'$ *holds. Then there exists some derivation* $\mathcal{D}$ *of this judgment. Derivation* $\mathcal{D}$ *can only end in Rule 4 or 5.*

Case 1: *Suppose* $\mathcal{D}$ *ends in Rule 4:*

$$\mathcal{D} = \frac{\dfrac{?}{\langle !b, \sigma\rangle \Downarrow T} \qquad \dfrac{\mathcal{D}_2}{\langle c_1, \sigma\rangle \Downarrow \sigma'}}{\langle \texttt{if } !b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow \sigma'}\,(4)$$

Proof Without Induction

Theorem:

The judgment $\langle \texttt{if } !b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow \sigma'$ *holds if and only if the judgment* $\langle \texttt{if } b \texttt{ then } c_2 \texttt{ else } c_1, \sigma\rangle \Downarrow \sigma'$ *holds.*

Proof:

We first prove the forward implication. Assume judgment $\langle \texttt{if } !b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow \sigma'$ *holds. Then there exists some derivation* $\mathcal{D}$ *of this judgment. Derivation* $\mathcal{D}$ *can only end in Rule 4 or 5.*

Case 1: *Suppose* $\mathcal{D}$ *ends in Rule 4:*

$$\mathcal{D} = \dfrac{\dfrac{\dfrac{\mathcal{D}_1}{\langle b, \sigma\rangle \Downarrow F}(12)}{\langle !b, \sigma\rangle \Downarrow T} \qquad \dfrac{\mathcal{D}_2}{\langle c_1, \sigma\rangle \Downarrow \sigma'}}{\langle \texttt{if } !b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow \sigma'}(4)$$

Proof Without Induction

Theorem:

The judgment $\langle \texttt{if } !b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow \sigma'$ *holds if and only if the judgment* $\langle \texttt{if } b \texttt{ then } c_2 \texttt{ else } c_1, \sigma\rangle \Downarrow \sigma'$ *holds.*

Proof:

We first prove the forward implication. Assume judgment $\langle \texttt{if } !b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow \sigma'$ *holds. Then there exists some derivation* $\mathcal{D}$ *of this judgment. Derivation* $\mathcal{D}$ *can only end in Rule 4 or 5.*

Case 1: *Suppose* $\mathcal{D}$ *ends in Rule 4:*

$$\mathcal{D} = \dfrac{\dfrac{\dfrac{\mathcal{D}_1}{\langle b, \sigma\rangle \Downarrow F}(12)}{\langle !b, \sigma\rangle \Downarrow T} \qquad \dfrac{\mathcal{D}_2}{\langle c_1, \sigma\rangle \Downarrow \sigma'}}{\langle \texttt{if } !b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow \sigma'}(4)$$

Using subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$ we can derive:

$$\dfrac{\dfrac{\mathcal{D}_1}{\langle b, \sigma\rangle \Downarrow F} \qquad \dfrac{\mathcal{D}_2}{\langle c_1, \sigma\rangle \Downarrow \sigma'}}{\langle \texttt{if } b \texttt{ then } c_2 \texttt{ else } c_1, \sigma\rangle \Downarrow \sigma'}(5)$$

Proof Without Induction

Theorem:

The judgment $\langle \texttt{if } !b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow \sigma'$ *holds if and only if the judgment* $\langle \texttt{if } b \texttt{ then } c_2 \texttt{ else } c_1, \sigma\rangle \Downarrow \sigma'$ *holds.*

Proof:

We first prove the forward implication. Assume judgment $\langle \texttt{if } !b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow \sigma'$ *holds. Then there exists some derivation* $\mathcal{D}$ *of this judgment. Derivation* $\mathcal{D}$ *can only end in Rule 4 or 5.*

Case 1: *Suppose* $\mathcal{D}$ *ends in Rule 4:*

$$\mathcal{D} = \dfrac{\dfrac{\dfrac{\mathcal{D}_1}{\langle b, \sigma\rangle \Downarrow F}}{\langle !b, \sigma\rangle \Downarrow T}(12) \qquad \dfrac{\mathcal{D}_2}{\langle c_1, \sigma\rangle \Downarrow \sigma'}}{\langle \texttt{if } !b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \Downarrow \sigma'}(4)$$

Using subderivations $\mathcal{D}_1$ and $\mathcal{D}_2$ we can derive:

$$\dfrac{\dfrac{\mathcal{D}_1}{\langle b, \sigma\rangle \Downarrow F} \qquad \dfrac{\mathcal{D}_2}{\langle c_1, \sigma\rangle \Downarrow \sigma'}}{\langle \texttt{if } b \texttt{ then } c_2 \texttt{ else } c_1, \sigma\rangle \Downarrow \sigma'}(5)$$

Exercise: Complete Case 2 and the proof of the reverse implication (same basic idea). Answers given in online notes.