Definition of Terms

Define the set of terms as

$$
\begin{aligned}
S_0 &= \emptyset \\
S_{i+1} &= \{\texttt{true}, \texttt{false}, 0\} \cup \{\texttt{succ}\ \texttt{t}_1, \texttt{pred}\ \texttt{t}_1, \texttt{iszero}\ \texttt{t}_1 \mid \texttt{t}_1 \in S_i\} \cup \{\texttt{if}\ \texttt{t}_1\ \texttt{then}\ \texttt{t}_2\ \texttt{else}\ \texttt{t}_3 \mid \texttt{t}_1, \texttt{t}_2, \texttt{t}_3 \in S_i\} \\
S &= \bigcup_i S_i
\end{aligned}
$$

Exercise 3.2.4

How many elements does S_3 have?

Solution: $S_0 = \emptyset$, so $S_1 = \{\texttt{true}, \texttt{false}, 0\}$. S_2 contains the atomic terms from S_1 and also the compound terms built from atomic terms. The functions `suc`, `pred`, and `iszero` are all unary, so for each term `t` they produce a distinct term τ `t` where τ is one of `succ`, `pred`, and `iszero`. Each function produces 3 terms from the atomic terms, so in total $3 \cdot 3$ terms are produced by the unary functions.

The $\texttt{if}\ \texttt{t}_1\ \texttt{then}\ \texttt{t}_2\ \texttt{else}\ \texttt{t}_3$ function is ternary, and produces a distinct term for any choice of three terms $\texttt{t}_1, \texttt{t}_2, \texttt{t}_3$. In the case of S_2 there are three choices (`true`, `false`, 0) for each of $\texttt{t}_1, \texttt{t}_2, \texttt{t}_3$, so $\texttt{if}\ \texttt{t}_1\ \texttt{then}\ \texttt{t}_2\ \texttt{else}\ \texttt{t}_3$ produces $3 \cdot 3 \cdot 3 = 27$ distinct terms.

In total, S_2 has $3 + 9 + 27 = 39$ terms. Following the same approach, for S_3, the unary functions produce $39 \cdot 3 = 117$ terms, and the ternary function produces $39^3 = 59,319$ terms. Combining with the three atomic terms gives $|S_3| = 3 + 117 + 59,319 = 59,439$.

Exercise 3.2.5

Show that the sets S_i are cumulative.

Proof. We will show this by induction on i that $S_i \subseteq S_{i+1}$.

Base case: $S_0 = \emptyset$ so trivially $S_0 \subseteq S_1$.

Inductive step: Assume that we have shown that $S_{i-1} \subseteq S_i$. We will show that $S_i \subseteq S_{i+1}$. Let $\texttt{t} \in S_i$ be some term. Then, either `t` is and atomic term, or it is a compound term containing functions. If `t` is atomic, then by construction, $\texttt{t} \in S_{i+1}$. So, suppose `t` is a term containing functions.

If the outermost function of `t` is a unary function then $\texttt{t} = \tau\ \texttt{s}$ where τ is a unary function and $\texttt{s} \in S_{i-1}$. By the IH, $\texttt{s} \in S_i$ and it follows that $\texttt{t} = \tau\ \texttt{s} \in S_{i+1}$.

If the outermost function of `t` is the ternary conditional function, then a completely analogous argument with `s` replaced by three terms $\texttt{s}_1, \texttt{s}_2, \texttt{s}_3$ shows that $\texttt{t} \in S_{i+1}$.

By induction, it follows that $S_i \subseteq S_{i+1}$ for all $i \in \mathbb{N}$ which shows that the sets S_i are cumulative. □

Exercise 3.3.4

Suppose P is a predicate on terms. Show that Structural Induction holds:
If, for each term `s`, given $P(\texttt{r})$ for all immediate subterms `r` of `s` we can show $P(\texttt{s})$, then $P(\texttt{s})$ holds for all `s`.

Proof. First, note that the relation $\prec$ defined on terms by

$$\texttt{r} \prec \texttt{s} \text{ iff } \texttt{r} \text{ is a subterm of } \texttt{s}$$

is well-founded. This can be seen by considering the concrete definition of terms using the sets S_i. We showed in Exercise 3.2.5 that these sets are cumulative, and moreover by their definition, it is easy to see that they are all finite. Since any term `t` appears in some set S_i, it follows that any `t` can only have finitely many $\prec$-predecessors. Hence, there are no infinite $\prec$-descending chains.

Now we can prove the desired statement by contradiction. Suppose we have a predicate P satisfying the hypothesis of Structural Induction. We will show that $P(\texttt{s})$ holds for all `s`. Suppose this was not the case, so for some term `s`, $P(\texttt{s})$ does not hold. Using the well-foundedness of $\prec$ choose such an `s` which is $\prec$-minimal. That is, $P(\texttt{s})$ does not hold, but $P(\texttt{r})$ does hold for all terms `r` such that $\texttt{r} \prec \texttt{s}$, i.e. all subterms of `s`. In particular, P holds for all immediate subterms of `s`. By our assumption that P satisfies the hypothesis of structural induction, it follows that $P(\texttt{s})$ does hold. Contradiction. Hence, there are no terms `s` such that $P(\texttt{s})$ does not hold. □

Exercise 3.5.5

Spell out the induction principle used in the proof of Theorem 3.5.4.

Solution: Suppose P is a predicate evalaution statement derivations. If D is a derivation and assuming $P(\overline{D})$ holds for all subderivations $\overline{D}$ of D we can prove $P(D)$, then $P(D)$ holds for all derivations D.

Exercise 3.5.10

Rephrase Definition 3.5.9 of the *multi-step evaluation* relation $\to^*$ as a set of inference rules.

Solution:

$$\frac{\mathtt{t} \to \mathtt{t}'}{\mathtt{t} \to^* \mathtt{t}'} \quad \mathtt{t} \to^* \mathtt{t} \quad \frac{\mathtt{t} \to^* \mathtt{t}' \text{ and } \mathtt{t}' \to^* \mathtt{t}''}{\mathtt{t} \to^* \mathtt{t}''}$$

Exercise 3.5.13

1. Adding the rule

$$\mathtt{if\ true\ then\ t_2\ else\ t_3} \to \mathtt{t_3}$$

   would make

   - Theorem 3.5.4 (Determinacy of one-step evalaution) fail since there are two posibilities for transitioning with `if true then` $\cdot$ `else` $\cdot$.
   - Theorem 3.5.7 (every value is in normal form) will still hold
   - Theorem 3.5.8 (normal forms are values) will still hold
   - Theorem 3.5.11 (uniqueness of normal forms) will fail like Theorem 3.5.4

1. Adding the rule

$$\frac{\mathtt{t_2} \to \mathtt{t_2'}}{\mathtt{if\ t_1\ then\ t_2\ else\ t_3} \to \mathtt{if\ t_1\ then\ t_2'\ else\ t_3}}$$

   would make

   - Theorem 3.5.4 (Determinacy of one-step evalaution) fail since there are two posibilities for transitioning with `if t1 then t2 else` $\cdot$.
   - Theorem 3.5.7 (every value is in normal form) will still hold
   - Theorem 3.5.8 (normal forms are values) will still hold
   - Theorem 3.5.11 (uniqueness of normal forms) will still hold. Suppose that $\mathtt{if\ t_1\ then\ t_2\ else\ t_3} \to^* \mathtt{v}$ where $\mathtt{t_2} \to^* \mathtt{v}$. In the original semantics, this would only be possible by first evaluating $\mathtt{t_1} \to^* \mathtt{true}$ producing $\mathtt{t_2}$ then evaluating $\mathtt{t_2} \to^* \mathtt{v}$. With the new rule, it is possible that some steps of the evalaution $\mathtt{t_2} \to^* \mathtt{v}$ are interleaved with the evaluation of $\mathtt{t_1} \to^* \mathtt{true}$. Intuitively, this reorganization will not change the resulting value, but a complete proof of this would require proving the diamond property. See the book for a full argument.

Exercise 3.5.14

Proof Theorem 3.5.4 on arithmetic expressions: if $\mathtt{t} \to \mathtt{t}'$ and $\mathtt{t} \to \mathtt{t}''$ then $\mathtt{t}' = \mathtt{t}''$.

Proof. By induction on the derivation of $\mathtt{t} \to \mathtt{t}'$. If the last rule in the derivation is E-SUCC, then $\mathtt{t}$ has the form $\mathtt{succ\ t_1}$ and $\mathtt{t_1} \to \mathtt{t_1'}$. It follows that the only rule that can apply to $\mathtt{t}$ is E-SUCC, so the last rule of the derivation of $\mathtt{t} \to \mathtt{t}''$ is E-SUCC as well and $\mathtt{t_1} \to \mathtt{t_1''}$. By the IH, $\mathtt{t_1'} = \mathtt{t_1''}$, and it follows that $\mathtt{t}' = \mathtt{t}''$.

Now, suppose the last rule of the derivation is E-PRED. So, $\mathtt{t}$ has the form $\mathtt{pred\ t_1}$ and $\mathtt{t_1} \to \mathtt{t_1'}$. Note that the rules E-PREDZERO and E-PREDSUCC cannot apply to $\mathtt{t}$ because $\mathtt{t_1}$ is not a value. It follows that the last rule applied in the derivation of $\mathtt{t} \to \mathtt{t}''$ must be E-PRED and $\mathtt{t_1} \to \mathtt{t_1'}$. By the IH, $\mathtt{t_1'} = \mathtt{t_1''}$, and it follows that $\mathtt{t}' = \mathtt{t}''$.

The argument if the last rule of the derivation is E-ISZERO follow in a completely analogous manner, except that we need to rule out the possibilites the possibilities of E-ISZEROZERO and E-ISZEROSUCC instead of E-PREDZERO and E-PREDSUCC.

The arguments for the remaining rules are even simpler. For example, suppose the final rule used in the derivation is E-PREDZERO. Then, $\mathtt{t}$ is $\mathtt{pred\ 0}$ and trivially, $\mathtt{t}' = 0 = \mathtt{t}''$. Similarly, if the final rule is E-ISZEROZERO we can show that $\mathtt{t}' = \mathtt{true} = \mathtt{t}''$. Analogous reasoning works for E-PREDSUCC and E-ISZEROSUCC. In the case of E-PREDSUCC we show that $\mathtt{t}' = \mathtt{nv1} = \mathtt{t}''$ and in the case of E-ISZEROSUCC we show that $\mathtt{t}' = \mathtt{false} = \mathtt{t}''$. □

Exercise 3.5.16

Show that the two treatments of runtime errors agree.

Claim 1. *Let* $\mathtt{t}$ *be a term in the original syntax and let* $\to, \to_1^*$ *denote the transition relations for the original semantics and* $\to, \to_2^*$ *denote the transition relation for the augmented semantics. Then,* $\mathtt{t} \to_1^* \mathtt{t}'$ *where* $\mathtt{t}'$ *is stuck iff* $\mathtt{t} \to_2^* \mathtt{wrong}$*.*

Lemma 1. *If* $\mathtt{t} \to_1^* \mathtt{t}'$ *then* $\mathtt{t} \to_2^* \mathtt{t}'$ *and follows the same steps.*

Proof. If $\mathtt{t}'$ can be transitioned to from $\mathtt{t}$, then $\mathtt{badnat}$ or $\mathtt{badbool}$ could never appear in a term that would allow one of the WRONG evaluations to apply. Hence, the transition $\mathtt{t} \to_2^* \mathtt{t}'$ would follow exactly the same steps as $\mathtt{t} \to_1^* \mathtt{t}'$. □

Lemma 2. *If* $\mathtt{t}$ *is stuck in the original semantics then* $\mathtt{t} \to_2^* \mathtt{wrong}$*.*

Proof. Suppose $\mathtt{t}$ is stuck. We show this by induction on the structure of $\mathtt{t}$. There are a few cases to consider.

$\mathtt{t}$ is of the form $\mathtt{pred}\ \mathtt{t_1}$ If $\mathtt{t_1}$ is not a value, then $\mathtt{pred}\ \mathtt{t_1}$ is stuck iff $\mathtt{t_1}$ is stuck. By the IH, $\mathtt{t_1} \to_2^* \mathtt{wrong}$, so $\mathtt{t} \to_2^* \mathtt{wrong}$ via E-PRED-WRONG. If $\mathtt{t_1}$ is a value $\mathtt{v}$, then since $\mathtt{t}$ is stuck $\mathtt{v}$ must not be a numeric value. Hence, $\mathtt{v}$ is $\mathtt{true}$ or $\mathtt{false}$. Thus, $\mathtt{v}$ is a $\mathtt{badnat}$ and using E-PRED-WRONG $\mathtt{t} \to_2^* \mathtt{wrong}$.

$\mathtt{t}$ is of the form $\mathtt{iszero}\ \mathtt{t_1}$ By the same reasoning as the previous case, $\mathtt{t} \to_2^* \mathtt{wrong}$ using E-ISZERO-WRONG.

$\mathtt{t}$ is of the form $\mathtt{succ}\ \mathtt{t_1}$ Similar to the previous cases, but we need not consider the case when $\mathtt{t_1}$ is a value.

$\mathtt{t}$ is of the form $\mathtt{if}\ \mathtt{t_1}\ \mathtt{then}\ \mathtt{t_2}\ \mathtt{else}\ \mathtt{t_3}$ If $\mathtt{t_1}$ is not a value, then $\mathtt{if}\ \mathtt{t_1}\ \mathtt{then}\ \mathtt{t_2}\ \mathtt{else}\ \mathtt{t_3}$ is stuck iff $\mathtt{t_1}$ is stuck. By the IH, $\mathtt{t_1} \to_2^* \mathtt{wrong}$, so $\mathtt{t} \to_2^* \mathtt{wrong}$ via E-IF-WRONG. If $\mathtt{t_1}$ is a value $\mathtt{v}$, then since $\mathtt{t}$ is stuck $\mathtt{v}$ must be a numeric value. Hence, $\mathtt{v}$ is a $\mathtt{badbool}$ and using E-IF-WRONG $\mathtt{t} \to_2^* \mathtt{wrong}$.

□

The two previous lemmas show that if $\mathtt{t} \to_1^* \mathtt{t}'$ where $\mathtt{t}'$ is stuck, then $\mathtt{t} \to_2^* \mathtt{wrong}$. For the reverse direction, we show

Lemma 3. *Suppose* $\mathtt{t} \to_1^* \mathtt{t}'$ *and* $\mathtt{t}' \to \mathtt{t}''$ *where* $\mathtt{t}''$ *has* $\mathtt{wrong}$ *as a subterm, then* $\mathtt{t}'$ *is stuck.*

Proof. By induction on a derivation of $\mathtt{t}' \to \mathtt{t}''$. Suppose that the final rule in the derivation is E-IF-WRONG. Then, $\mathtt{t}'$ has the form $\mathtt{if}\ \mathtt{badbool}\ \mathtt{then}\ \mathtt{t_2}\ \mathtt{else}\ \mathtt{t_3}$. Because $\mathtt{t}'$ was obtained through evaluation in the original semantics, it follows that $\mathtt{badbool}$ is a numeric value $\mathtt{nv}$ and not $\mathtt{wrong}$. Hence, $\mathtt{t}'$ is stuck. We can argue similarly if the final rule is E-SUCC-WRONG, E-PRED-WRONG, or E-ISZERO-WRONG.

If the final rule in the derivation is E-PRED, then $\mathtt{t}'$ is of the form $\mathtt{pred}\ \mathtt{t_1}$ and $\mathtt{t_1} \to \mathtt{t_1'}$. By our assumption that $\mathtt{t}''$ has $\mathtt{wrong}$ as a subterm it follows that the evaluation $\mathtt{t_1} \to \mathtt{t_1'}$ must use one of the $\mathtt{wrong}$ producing rules E-IF-WRONG, E-SUCC-WRONG, E-PRED-WRONG, E-ISZERO-WRONG. From the induction hypothesis, we get that $\mathtt{t_1}$ is stuck which implies that $\mathtt{t}' = \mathtt{pred}\ \mathtt{t_1}$ is stuck. We can argue in a similar manner if the final rule in the derivation is E-SUCC, E-ISZERO, or E-IF.

The remaining numeric expression produce values, so could not be the final rule in the derivation. The rules E-IFTRUE and E-IFFALSE extract a subterm from $\mathtt{t}'$, so they could not have $\mathtt{wrong}$ as a subterm since the derivation $\mathtt{t} \to_1^* \mathtt{t}'$ happens in the original semantics. Hence, they could not be the final rule of the derivation either. □

Combining the previous lemma and Lemma 1, we can prove the reverse direction. Suppose that $\mathtt{t} \to_2^* \mathtt{wrong}$. By Lemma 1 an initial segment of the multi-step evaluation is done purely in the original semantics. Let $\mathtt{t}'$ be the final term produced using the original evaluation rules. It follows that the next term in the evaluation of $\mathtt{t} \to_2^* \mathtt{wrong}$ must have $\mathtt{wrong}$ as a subterm, so using the previous lemma, $\mathtt{t}'$ is stuck.

Exercise 3.5.17

Show that the small-step and big-step semantics agree, i.e. $\mathtt{t} \to^* \mathtt{v}$ iff $\mathtt{t} \Downarrow \mathtt{v}$.

Proof. We show this by induction on the structure of $\mathtt{t}$.

$\mathtt{t}$ is a value Trivially, $\mathtt{t} \to^* \mathtt{v}$ iff $\mathtt{t} \Downarrow \mathtt{v}$.

$\mathtt{t}$ is $\mathtt{if}\ \mathtt{t_1}\ \mathtt{then}\ \mathtt{t_2}\ \mathtt{else}\ \mathtt{t_3}$: By the induction hypothesis, $\mathtt{t_i} \to^* \mathtt{v}$ iff $\mathtt{t_i} \Downarrow \mathtt{v}$ for $i = 1, 2, 3$. For the forward direction, assume that $\mathtt{t} \to^* \mathtt{v}$. Necessarily, $\mathtt{t_i} \to^* \mathtt{v_i}$ for $i = 1, 2, 3$. Consider the following evaluation of $\mathtt{t}$. First, use the evaluation $\mathtt{t_1} \to^* \mathtt{v_1}$ and E-IF to obtain $\mathtt{t} \to^* \mathtt{if}\ \mathtt{v_1}\ \mathtt{then}\ \mathtt{t_2}\ \mathtt{else}\ \mathtt{t_3}$. If $\mathtt{v_1} = \mathtt{true}$ we next apply E-IFTRUE to obtain $\mathtt{t_2}$ and proceed with $\mathtt{t_2} \to^* \mathtt{v_2}$ to obtain $\mathtt{t} \to^* \mathtt{v_2}$. Otherwise, $\mathtt{v_1} = \mathtt{false}$ and we use E-IFFALSE and $\mathtt{t_3} \to^* \mathtt{v_3}$ to obtain $\mathtt{t} \to^* \mathtt{v_3}$. By the uniqueness of normal forms (Theorem 3.5.11), this proposed evaluation is valid and results in the correct value. Suppose the former case $\mathtt{t_1} \to^* \mathtt{true}$ holds. Then, by the IH, $\mathtt{t_1} \Downarrow \mathtt{true}$ and $\mathtt{t_2} \Downarrow \mathtt{v_2}$, so B-IFTRUE gives $\mathtt{t} \Downarrow \mathtt{v_2}$. In the latter case $\mathtt{t_1} \to^* \mathtt{false}$, and B-IFFALSE gives $\mathtt{t} \Downarrow \mathtt{v_3}$.

For the reverse direction, assume that $\mathtt{t} \Downarrow \mathtt{v}$. So, either B-IFTRUE or B-IFFALSE were used to evaluate $\mathtt{t}$. If B-IFTRUE was used, then $\mathtt{t_1} \Downarrow \mathtt{true}$, $\mathtt{t_2} \Downarrow \mathtt{v_2}$, and $\mathtt{t} \Downarrow \mathtt{v_2}$ for some $\mathtt{v_2}$. By the induction hypothesis, $\mathtt{t_1} \to^* \mathtt{true}$ and $\mathtt{t_2} \to^* \mathtt{v_2}$. We can again appeal to the uniqueness of normal forms theorem to construct a valid evaluation of $\mathtt{t} \to^* \mathtt{v_2}$ following the same approach as above. If B-IFFALSE was used, then $\mathtt{t_3} \Downarrow \mathtt{v_3}$ for some $\mathtt{v_3}$ so that $\mathtt{t} \Downarrow \mathtt{v_3}$, and we again construct a valid evaluation producing giving $\mathtt{t} \to^* \mathtt{v_3}$.

$\mathtt{t}$ is $\mathtt{pred}\ \mathtt{t_1}$: By the induction hypothesis, $\mathtt{t_i} \to^* \mathtt{v}$ iff $\mathtt{t_i} \Downarrow \mathtt{v}$ for $i = 1, 2, 3$. For the forward direction, assume that $\mathtt{t} \to^* \mathtt{v}$. Hence, $\mathtt{t_1} \to^* \mathtt{v_1}$ for some value $\mathtt{v_1}$. Following the approach for if-then-else term, we will construct a valid evaluation of $\mathtt{t}$. If $\mathtt{v_1} = 0$, then we can evaluate $\mathtt{t}$ by first using $\mathtt{t_1} \to^* 0$ and E-PRED to obtain $\mathtt{t} \to^* \mathtt{pred}\ \mathtt{0}$. Then, we can apply E-PREDZERO to obtain $\mathtt{t} \to^* 0$. On the other hand, if $\mathtt{v_1} = \mathtt{succ}\ \mathtt{nv}$, then we would use E-PREDSUCC in the final step to obtain $\mathtt{t} \to^* \mathtt{nv}$. In the former case, the induction hypothesis gives $\mathtt{t_1} \Downarrow 0$ so applying B-PREDZERO to $\mathtt{t}$ gives $\mathtt{t} \Downarrow 0$. In the later case, the induction hypothesis gives $\mathtt{t_1} \Downarrow \mathtt{succ}\ \mathtt{nv}$ and applying B-PREDSUCC gives $\mathtt{t} \Downarrow \mathtt{nv}$ as required.

For the reverse direction, assume that $\mathtt{t} \Downarrow \mathtt{v}$. If B-PREDZERO is applied then $\mathtt{t_1} \Downarrow 0$ so by the IH, $\mathtt{t_1} \to^* 0$. Hence, we can obtain $\mathtt{t} \to^* 0$ using E-PRED with $\mathtt{t_1} \to^* 0$ and applying E-PREDZERO in the final step. If B-PREDSUCC is applied then $\mathtt{t_1} \Downarrow \mathtt{nv}$ and $\mathtt{t_1} \to^* \mathtt{nv}$ by the IH. As in the B-PREDZERO case, we can use this to obtain $\mathtt{t} \to^* \mathtt{nv}$.

$\mathtt{t}$ is $\mathtt{succ}\ \mathtt{t_1}$ or $\mathtt{iszero}\ \mathtt{t_1}$: Similar to the `pred` case.

□

Exercise 3.5.18

Give evaluation rules so that the `then` and `else` branches of an `if` expression are evaluated before the guard is evaluated.

Solution: Remove E-IFTRUE, E-IFFALSE, and E-IF and replace with

$$\mathtt{if\ true\ then\ v_2\ else\ v_3} \to \mathtt{v_2} \qquad \mathtt{if\ false\ then\ v_2\ else\ v_3} \to \mathtt{v_3} \qquad \frac{\mathtt{t_2} \to \mathtt{t_2'}}{\mathtt{if\ t_1\ then\ t_2\ else\ t_3} \to \mathtt{if\ t_1\ then\ t_2'\ else\ t_3}}$$

$$\frac{\mathtt{t_3} \to \mathtt{t_3'}}{\mathtt{if\ t_1\ then\ v_2\ else\ t_3} \to \mathtt{if\ t_1\ then\ v_2\ else\ t_3'}} \qquad \frac{\mathtt{t_1} \to \mathtt{t_1'}}{\mathtt{if\ t_1\ then\ v_2\ else\ v_3} \to \mathtt{if\ t_1'\ then\ v_2\ else\ v_3}}$$