COMP523 - A1

Gulliver Häger - ID: 260907422

January 2022

1 Exercise 1

1.1 Replacing E-PRED-SUCC

Replacing this rule is a **bad idea**. We break both determinancy and uniqueness. This can be seen using the two sample derivations below, where we evaluate the term $succ(pred(0))$

D:

$$\dfrac{}{\texttt{succ}(\texttt{pred}(0)) \to 0}\ \text{E-PREDD-SUCC}$$

E:

$$\dfrac{\dfrac{}{\texttt{pred}(0) \to 0}\ \text{E-PRED-ZERO}}{\texttt{succ}(\texttt{pred}(0)) \to \texttt{succ}(0)}\ \text{E-SUCC}$$

We have broken **determinancy** since $t \to 0$ and $t \to \texttt{succ}(0)$ but clearly $0 \neq \texttt{succ}(0)$.
We have broken **uniqueness** since $t \to^* 0$ and $t \to^* \texttt{succ}(0)$ but clearly $0 \neq \texttt{succ}(0)$.

1.2 Replacing E-ISZERO-SUCC

In this case, terms we normally would want to get stuck can be evaluated. An example of this can be seen if we let $t = \texttt{pred}(\texttt{true})$.

D:

$$\dfrac{}{\texttt{iszero}(\texttt{succ}(\texttt{pred}(\texttt{true}))) \to \texttt{false}}\ \text{E-ISZERO-SUCC}$$

Clearly, this derivation does not behave as desired, since there does not exist a notion of a successor or predecessor to a Boolean value, and should not be evaluated. If we extend our notion of uniqueness to normal forms, which a stuck expression is, this would then break **uniqueness**, since in some circumstances the term would get stuck and in some it would be evaluated to false.

2 Exercise 2

Here we will argue by **induction of the derivation**. Based on the multi step relation and the given rules, we'll have 2 base cases and 1 induction step case.

Suppose that v is a numerical value and that the last inference of the derivation **D** is: $v \to^* v'$.

2.1 Base case 1: Last inference of D is REFL

Clearly we have that: **D:**

$$\frac{}{v \to^* v}\ \text{REFL}$$

meaning that $v = v'$ and we are done.

2.2 Base case 2: Last inference of D is SINGLE

We have that: **D:**

$$\frac{v \to v'}{v \to^* v'}\ \text{SINGLE}$$

meaning that the relation $v \to v'$ is true. But as seen in class on January 11, we have proven a lemma saying that numerical values do not step. Since v is a numerical value, we have $\perp$, from which anything follows, meaning that $v = v'$ and we are done with this case.

2.3 Induction step: Last inference of D is TRANS

We have that:

D:

$$\frac{v \to^* s \qquad s \to^* v'}{v \to^* v'}\ \text{TRANS}$$

Since the premises of this derivation come from shorter derivations, we can apply our IH to get that $v = s$ and $s = v'$. By transitivity of equality, we then have that $v = v'$, and we are done with the induction step and therefore also with the inductive proof.

■

3 Exercise 3

3.1 Small Step Evaluation for `leq`

Below are the small step evaluation rules for `leq`. There are 2 "base cases" to produce Boolean values, 2 congruence rules, avoiding both parallel evaluation and preserving determinancy through asymmetry, and a rule through which we can reduce complex terms involving `leq` into simpler terms.

$$\frac{}{\texttt{leq}(0, nv) \to \texttt{true}}\ \text{E-LEQ-T} \qquad \frac{}{\texttt{leq}(\texttt{succ}(nv), 0) \to \texttt{false}}\ \text{E-LEQ-F}$$

$$\frac{t \to t''}{\texttt{leq}(t, t') \to \texttt{leq}(t'', t')}\ \text{E-LEQ-CONG1} \qquad \frac{t' \to t''}{\texttt{leq}(nv, t') \to \texttt{leq}(nv, t'')}\ \text{E-LEQ-CONG2}$$

$$\frac{\texttt{leq}(nv, nw) \to bv}{\texttt{leq}(\texttt{succ}(nv), \texttt{succ}(nw)) \to bv}\ \text{E-LEQ-RED}$$

3.2 Proving Determinancy

We want to show that if $t \to t'$ and $t \to t''$, then $t' = t''$ when applying any of the `leq`-rules. Let **D** be the derivation of $t \to t'$ and **E** the derivation of $t \to t''$.

We will proceed by **structural induction on D**.

3.2.1 Base Case 1: Last inference of D is E-LEQ-T

We have by assumption that:

$$\mathbf{D}: \frac{}{\texttt{leq}(0, nv) \to \texttt{true}}\ \text{E-LEQ-T} \qquad \mathbf{E}: \frac{\mathbf{E}'}{\texttt{leq}(0, nv) \to t''}$$

Notice that both congruence rules are not possible as the last rule of **E** since these would require values to step, which we have proven in class they cannot. The last rule of **E** cannot be E-LEQ-RED either since this would require 0 to be a successor of a natural value, which it is not. The exact same reason is true for why E-LEQ-F cannot be the last inference. Thus the only possible inference left is E-LEQ-T, which then trivially means that $t' = t''$ holds and we are done.

3.2.2 Base Case 2: Last inference of D is E-LEQ-F

This case is almost entirely analogous to the previous case. Since the Boolean rules are asymmetric, some adjustments have to be made, but these are trivial such as saying that the second argument instead of the first is required to be a successor, etc.

3.2.3 Induction step 1: Last inference of D is E-LEQ-CONG1

We have by assumption that:

$$\mathbf{D}: \frac{\texttt{leq}(nv, nw) \to bv}{\texttt{leq}(\texttt{succ}(nv), \texttt{succ}(nw)) \to bv}\ \text{E-LEQ-RED} \qquad \mathbf{E}: \frac{\mathbf{E}'}{\texttt{leq}(t_0, t_1) \to t''}$$

The cases where the last inference rule of **E** is E-LEQ-T or E-LEQ-F are impossible since this would require values to step, since we have $t_0 \to t_2$ in **D**, which we have shown is impossible.

The case where the last inference of **E** is E-LEQ-CONG2 is also impossible since this would once again require a value, t_0, to step.

The case where the last inference of **E** is E-LEQ-RED would mean that **E** would look like below:

$$\mathbf{E}: \quad \cfrac{\cfrac{\mathbf{E''}}{\texttt{leq}(nv, nw) \to bv}}{\texttt{leq}(\texttt{succ}(nv), \texttt{succ}(nw)) \to bv} \text{ E-LEQ-RED}$$

meaning that $t_0 = \texttt{succ}(nv)$ and $t_1 = \texttt{succ}(nw)$. But from **D** we have that $t_0 \to t_2$, meaning then that $\texttt{succ}(nv) \to t_2$, which once again is impossible - thus meaning that this case is also impossible.

The only remaining case is then when the last inference of **E** is E-LEQ-CONG1 and $t' = t''$ is then trivially true.

3.2.4 Induction step 2: Last inference of D is E-LEQ-CONG2

This case is analogous to the previous case.

3.2.5 Induction step 3: Last inference of D is E-LEQ-RED

We have by assumption that:

$$\mathbf{D}: \quad \cfrac{\cfrac{\mathbf{D'}}{\texttt{leq}(nv, nw) \to bv}}{\texttt{leq}(\texttt{succ}(nv), \texttt{succ}(nw)) \to bv} \text{ E-LEQ-RED} \qquad \mathbf{E}: \quad \cfrac{\mathbf{E'}}{\texttt{leq}(\texttt{succ}(nv), \texttt{succ}(nw)) \to t''}$$

In this case, it is once again impossible that the last inference of **E** is E-LEQ-CONG1 or E-LEQ-CONG2 since this would require a value to step.

In the case when the last inference of **E** is either E-LEQ-T or E-LEQ-F, this would require 0 to be the successor of a natural value, which it is not, and therefore these cases are impossible.

In the case when the last inference of **E** is E-LEQ-RED, $t' = t''$ holds trivially through symmetry. This completes the proof.

■

3.3 Typing Rule for `leq`

Below is a proposed typing rule for $\texttt{leq}(t, t')$.

$$\frac{\texttt{t : Nat} \quad \texttt{t}' \texttt{ : Nat}}{\texttt{leq}(t, t') \texttt{ : Bool}}$$

3.4 Proving Type Preservation

We want to show that if t:T and $t \to t'$, then t':T. We proceed with structural induction on the derivation of $t \to t'$, call it **D**. For this we need a typing inversion lemma, for which the proof is direct from the definition of the typing rule.

Typing Lemma: If $\texttt{leq}(t, t')$: T, then T = Bool, t : Nat and t' : Nat.

3.4.1 Last inference of D is E-LEQ-T

We have that:

$$\mathbf{D}:\ \dfrac{}{\texttt{leq}(0, nv) \to \texttt{true}}\ \text{E-LEQ-T}$$

$\texttt{leq}(0, nv) : \texttt{T}$	By assumption
$\texttt{leq}(0, nv) : \texttt{Bool}$ and $\texttt{T} = \texttt{Bool}$	By inversion lemma
$\texttt{true} : \texttt{Bool}$	By inversion on $\texttt{true}$

Thus we have shown that both t and t' have the same type and we are done.

3.4.2 Last inference of D is E-LEQ-F

Analogous to the previous case.

3.4.3 Last inference of D is E-LEQ-CONG1

We have that:

$$\mathbf{D}:\ \dfrac{\dfrac{\mathbf{D'}}{t_0 \to t_2}}{\texttt{leq}(t_0, t_1) \to \texttt{leq}(t_2, t_1)}\ \text{E-LEQ-CONG1}$$

$\texttt{leq}(t_0, t_1) : \texttt{T}$	By assumption
$\texttt{leq}(t_0, t_1) : \texttt{Bool}$, $\texttt{T} = \texttt{Bool}$, $t_0 : \texttt{Nat}$ and $t_1 : \texttt{Nat}$	By inversion lemma
$t_2 : \texttt{Nat}$	By IH on **D'** since $t_0 : \texttt{Nat}$
$\texttt{leq}(t_2, t_1) : \texttt{Bool}$	By typing rule for $\texttt{leq}$

Thus we have shown that both t and t' have the same type and we are done.

3.4.4 Last inference of D is E-LEQ-CONG2

Almost entirely analogous to the previous case.

3.4.5 Last inference of D is E-LEQ-RED

We have that:

$$\mathbf{D}:\ \dfrac{\dfrac{\mathbf{D'}}{\texttt{leq}(nv, nw) \to bv}}{\texttt{leq}(\texttt{succ}(nv), \texttt{succ}(nw)) \to bv}\ \text{E-LEQ-RED}$$

$\texttt{leq}(\texttt{succ}(nv), \texttt{succ}(nw)) : \texttt{T}$	By assumption
$\texttt{leq}(\texttt{succ}(nv), \texttt{succ}(nw)) : \texttt{Bool}$, $\texttt{T} = \texttt{Bool}$	By inversion lemma
$bv : \texttt{Bool}$	Since $bv \in \{\texttt{true}, \texttt{false}\}$ which both invert to $\texttt{Bool}$

Thus we have shown that both t and t' have the same type and we are done. This completes the proof.

Note: To be more rigorous it is possible to show that $bv : \texttt{Bool}$ by appealing to the IH and adding a couple of steps.

4 Exercise 4

4.1 Proving: If $e \to^* v$ then $e \Downarrow v$

First we start proving the necessary lemmas for the proof.

Lemma 1: For any value v, we have: $v \Downarrow v$
Proof of Lemma 1: By structural induction on the structure of v.

- Base case: $v = z$. True from the evaluation rule B-Z.
- Induction step: $v = \texttt{iszero}(v')$ OR $v = \texttt{pred}(v')$. Not a possible case since these are not values.
- Induction step: $v = \texttt{succ}(v')$. As our IH, we have that $v' \Downarrow v'$. Applying B-SUCC, we get the desired result, that is: $\texttt{succ}(v') \Downarrow \texttt{succ}(v')$

■

Lemma 2: If $\texttt{pred}(e) \to^* v$, then the derivation of $e \to^* v'$ is shorter (where we either have that $z = v' = v$ OR $v' = \texttt{succ}(v)$.
Proof of Lemma 2: By mathematical induction on the length of the derivation, n, of $\texttt{pred}(e) \to^* v$, call it **D**. Notice that there are three different rules so that we get the conclusion $\texttt{pred}(e) \to^* v$. These are shown below and naturally require that $n \geq 1$. In the case when $n = 0$, we have that $v = \texttt{pred}(e)$, implying that $\texttt{pred}(e)$ is a value, which is incorrect, and therefore this case is impossible.

- Base case: last inference of D is **E-PRED-ZERO**. We have that **D** looks like below:

$$\frac{}{\texttt{pred}(e) \to z}\ \text{E-PRED-ZERO}$$

  Since $e = z$ in this case, we have that e is a value and therefore that the derivation of $e \to^* v'$ must be of length 0, meaning that $z = v = e$, meaning that we have $e \to^* v'$ in less than n steps and we are done.

- Base case: last inference of D is **E-PRED-SUCC**. We have that **D** looks like below:

$$\frac{}{\texttt{pred}(\texttt{succ}(e)) \to nv}\ \text{E-PRED-SUCC}$$

  Since $e = nv$ in this case, we have that e cannot step further since it is a value. The only way we can have that $e \to^* v$ then is if $e = v$ and therefore that we have a derivation in less than n steps and we are done.

- Induction step: last inference of D is **E-PRED**. We have that **D** looks like below:

$$\frac{e \to e'}{\texttt{pred}(e) \to \texttt{pred}(e')}\ \text{E-PRED}$$

  By our IH, we have that the derivation for $e' \to^* v'$ is at most $n-2$ steps, and that either $v' = \texttt{succ}(v)$ or $z = v = v'$. Thus, since $e \to^* e'$ and $e' \to^* v'$, by transativity of the multistep-relation, we have that $e \to^* v'$ in at most $(n-2)+1 = (n-1)$ steps, which completes the proof.

■

Main result: If $e \to^* v$ then $e \Downarrow v$
Proof of main result: By induction on the n, the length of the derivation $e \to^* v$.

- **Case 1**: $e = z$. Since z is a value, we have that **lemma 1** proves this case directly.
- **Case 2**: $e = \texttt{pred}(e')$. By **lemma 2**, we have that $e \to^* v'$ in fewer steps than $\texttt{pred}(e) \to^* v$ and that either $z = v' = v$ OR $v' = \texttt{succ}(v)$. If $z = v' = v$, we have by our IH that $e' \Downarrow z$, meaning that by B-PRED-ZERO, $\texttt{pred}(e') \Downarrow z$, which is equivalent to $e \Downarrow v$.

  In the other case, where $v' = \texttt{succ}(v)$, then by out IH, we have that $e' \Downarrow v'$, which by substitution is equivalent to $e' \Downarrow \texttt{succ}(v)$ and applying B-PRED-SUCC we get that $\texttt{pred}(e') \Downarrow v$ which again by substitution means that $e \Downarrow v$ and we are done.
- **Case 3-4**: in the case where $e = \texttt{iszero}(e')$ or $e = \texttt{succ}(nv)$ we can construct similar lemmas as **lemma 2** (by induction on the length of the derivation) and do a similar case analysis.

This completes the proof of this direction.

■

4.2 Proving: If $e \Downarrow v$ then $e \to^* v$

Lemma 1: if $e \to^* e'$ then $\texttt{pred}(e) \to^* \texttt{pred}(e')$
Proof of Lemma 1: By mathematical induction on the length, n, of the derivation for $e \to^* e'$.

- **Base case:** $n = 0$. In this case we have that $e = e'$, meaning that $\texttt{pred}(e) = \texttt{pred}(e')$ and by the REFL of the multi-step relation we are done.
- **Induction step:** we have that $e \to^* e''$ and $e'' \to e$ (by the possible rules for the multi-step relation) for some expression e''. Applying E-PRED to $e'' \to e$, we get that $\texttt{pred}(e'') \to \texttt{pred}(e')$. Notice that $e \to^* e''$ has length $(n-1)$, meaning that by our IH $\texttt{pred}(e) \to^* \texttt{pred}(e'')$. Applying TRANS of the multi-step relation, we then have that $\texttt{pred}(e) \to^* \texttt{pred}(e')$, completing the proof.

■

Main result: If $e \Downarrow v$ then $e \to^* v$
Proof of main result: By structural induction on the derivation of $e \Downarrow v$.

- **Case 1: Last rule applied is B-V**: this case is trivial since $e = v$ and by TRANS of the multi-step relation we then have $v \to^* v$, equivalent to $e \to^* v$.
- **Case 2: Last rule applied is B-PRED-SUCC**: in this case we have:

$$\mathbf{D:}\ \dfrac{\dfrac{\mathbf{D'}}{e' \Downarrow \texttt{succ}(v)}}{\texttt{pred}(e') \Downarrow v}$$

  Applying **Lemma 1**, we get that $e' \to^* \texttt{pred}(\texttt{succ}(v))$, since in this case, $\texttt{pred}(e') = e$. We also have by our IH that $e' \to^* \texttt{succ}(v)$. Finally, we can use our small-step evaluation rule E-PRED-SUCC to get $\texttt{pred}(\texttt{succ}(v)) \to v$, which together with TRANS and SINGLE of the multi-step evaluation gets us $\texttt{pred}(e') \to^* v$ - equivalent to $e \to^* v$.
- **Case 3: Last rule applied is B-PRED-ZERO**: in this case we have:

$$\mathbf{D:}\ \dfrac{\dfrac{\mathbf{D'}}{e' \Downarrow z}}{\texttt{pred}(e') \Downarrow z}$$

Applying **Lemma 1**, we get that $\texttt{pred}(e') \to^* \texttt{pred}(z)$ and by our IH applied to **D'** we get that $e' \to^* z$. By the single-step relation rule E-PRED-ZERO, we have that $\texttt{pred}(z) \to z$, meaning that by the SINGLE and TRANS multi-step relation rules means that $\texttt{pred}(e') \to^* z$, which is equivalent to $e \to^* v$ (since $e = \texttt{pred}(e')$.

- **Case 4-5: Last rule applied is B-SUCC or B-ISSUCC**: in these cases, we can construct a similar lemma to **lemma 1** for $\texttt{succ}(e) \to^* \texttt{succ}(e')$.

- **Case 6: Last rule applied is B-ISZERO**: in this case we can also construct a similar lemma to **lemma 1**.

■

A4 - COMP523 - Winter 2022

Unkown author

March 24, 2022

Q1. Weak Normalization

Q1.1) Reducibility Relation

Definition: $t \in R_{\texttt{Bool}}$ iff t halts and $t : \texttt{Bool}$

Q1.2) Backwards Close Lemma for Booleans

Lemma: if $t \to t'$ and $t' \in R_{\texttt{Bool}}$ then $t \in R_{\texttt{Bool}}$.

Proof: By induction on the type T.

The only case we will visit is $T = \texttt{Bool}$, since all other relevant cases was proven in class.

Case 1: $T = \texttt{Bool}$

(1) $t' \in R_{\texttt{Bool}}$	By assumption
(2) $\exists v.t' \to^* v$	Def. of halting and using (1)
(3) $t \to t'$	By assumption
(4) $t \to^* v$	By transativity of multistep using (2) and (3)
(5) t halts	By def. of halting and (4)
(6) $t' : \texttt{Bool}$	By def. of $R_{\texttt{Bool}}$ using (1)
(7) $t : \texttt{Bool}$	By type-preservation using (3) and (6)
(8) $t \in R_{\texttt{Bool}}$	By def. of $R_{\texttt{Bool}}$ using (5) and (7)

■

Q1.3) Proving the Main Lemma

To prove the main lemma, we first introduce some sub-lemmas. The first two are given without proof since similar lemmas have been proven before[1]. The third lemma is given with a proof. In this setting, it is assumed that the definition of R_T has been changed so that all terms in R_T have type T, which was touched upon on the discussion board and is used in the definition in Q1.1.

Lemma 1: if $t \to^*$ true then (if t then t_1 else t_2)$\to^*$ t_1.

Lemma 2: if $t \to^*$ false then (if t then t_1 else t_2)$\to^*$ t_2.

Lemma 3:[2] if $\sigma \in R_\Gamma$ then $\cdot \vdash \sigma : \Gamma$

Proof: By induction on the derivation of $\sigma \in R_\Gamma$, call it D.

Case 1: $D = \dfrac{}{\cdot \in R_{\cdot}}$

[1] I confirmed with Prof. Pientka in her OH that this was indeed OK.

[2] I ended up not needing this, but left it here for good measure.

(1) $\sigma = \cdot$ and $\Gamma = \cdot$	From case
(2) $\cdot \vdash (\cdot) : (\cdot)$	Trivially true

Which completes this case since (2) was what to be shown and is trivially true.

Case 2: $D = \dfrac{\sigma \in R_\Gamma \qquad s \in R_T}{\sigma, (s/x) \in R_{\Gamma,(x:T)}}$

(1) $\cdot \vdash (\sigma : \Gamma)$	By IH on D_1, the left subderiv.
(2) $s \in R_T$	From D_2, the right suberiv.
(3) $\cdot \vdash (s : T)$	By def. of R_T using (2)
(4) $\cdot \vdash (s/x) : (x : T)$	From (3)
(5) $\cdot \vdash (\sigma, s/x) : (\Gamma, x : T)$	Combining (1) and (4)

This completes the proof, since all cases have been shown.

$\blacksquare$

Main Lemma: if $\Gamma \vdash t : T$ and $\sigma \in R_\Gamma$ then $[\sigma]t \in R_T$

Proof: By induction on the typing derivation $\Gamma \vdash t : T$, call it D.

Case 1: $D = \dfrac{}{\Gamma \vdash \text{true} : \texttt{Bool}}\ \text{T-TRUE}$

(1) true $\to^*$ true	By reflexivity of multistep
(2) true halts	By def of halting using (1) and since true is a value
(3) $\Gamma \vdash \text{true} : \texttt{Bool}$	From D
(4) true $\in R_{\texttt{Bool}}$	By def of $R_{\texttt{Bool}}$ using (2) and (3)
(5) $[\sigma]\text{true} = \text{true}$	By def. of substitution
(6) $[\sigma]\text{true} \in R_{\texttt{Bool}}$	Rewriting (4) using (5)

Which is what to be shown in this case.

Case 2: $D = \dfrac{}{\Gamma \vdash \text{false} : \texttt{Bool}}\ \text{T-FALSE}$

This case is entirely analogous to case 1 and is left for brevity.

Case 3: $D = \dfrac{\Gamma \vdash t_1 : S \to T \qquad \Gamma \vdash t_2 : S}{t_1\ \ t_2 : T}\ \text{T-APP}$

This case was shown in class and is therefore left out for brevity.

Case 4: $D = \dfrac{\Gamma(x) = T}{\Gamma \vdash x : T}\ \text{T-VAR}$

This case was shown in class and is therefore left out for brevity.

Case 5: $D = \dfrac{\Gamma, x : T \vdash t : S}{\Gamma \vdash \lambda x : T.t : (T \to S)}\ \text{T-ABS}$

This case was shown in class and is therefore left out for brevity.

Case 6: $D = \quad \dfrac{\Gamma \vdash t : \texttt{Bool} \qquad \Gamma \vdash t_1 : T \qquad \Gamma \vdash t_2 : T}{\Gamma \vdash (\text{if } t \text{ then } t_1 \text{ else } t_2) : T}$ T-IF

(1) $\sigma \in R_\Gamma$	By assumption
(2) $[\sigma]t \in R_{\texttt{Bool}}$	By IH on D_1, the left subderiv., using (1)
(3) $\exists v.[\sigma]t \rightarrow^* v$	By def of $R_{\texttt{Bool}}$
(4) $v = \text{true}$ or $v = \text{false}$	By Canonical forms lemma using (3)

Subcase 6.1: $v = \text{true}$

(5) $[\sigma]t \rightarrow^* \text{true}$	Rewriting (3) using Subcase
(6) $[\sigma]t_1 \in R_T$	By IH on D_2, the middle subderiv.
(7) $[\sigma](\text{if } t \text{ then } t_1 \text{ else } t_2) = (\text{if } [\sigma]t \text{ then } [\sigma]t_1 \text{ else } [\sigma]t_2)$	By def. of subst.
(8) $(\text{if } [\sigma]t \text{ then } [\sigma]t_1 \text{ else } [\sigma]t_2) \rightarrow^* [\sigma]t_1$	By Lemma 1 using (5)
(9) $(\text{if } [\sigma]t \text{ then } [\sigma]t_1 \text{ else } [\sigma]t_2) \in R_T$	By backw. close lemma using (6) and (8)

Subcase 6.2: $v = \text{false}$

This subcase is entirely analogous to 6.1 except branching in the if-statement will be done using lemma 2. Therefore, the proof of this subcase is left out for brevity.

Thus the proof is complete since all cases have been shown.

■

Q2. Type Safety Using Logical Relations

Q2.1) Normal Type Safety Proof

Theorem (Type Safety): if $\vdash t : T$ then $\forall s\,[(t \rightarrow^* s) \rightarrow (s \text{ is a value } \vee \exists s'.s \rightarrow s')]$

Proof: Suppose that $\vdash t : T$. From progress, there are 2 cases.

Case 1: t is a value

(1) t is a value	From Subcase
(2) $t \rightarrow^* t$	By m-step reflexivity
(3) $\forall s.t \rightarrow^* s$, s is a value	Since (2) is the only possible m-step since t is a value and (1)
(4) t is safe	By def. and (3)

Case 2: $\exists t'.t \rightarrow t'$

(1) $\vdash t : T$	By assumption
(2) $t \rightarrow t'$	Restating Subcase
(3) $\vdash t' : T$	By type-preservation using (1) and (2)
(4) $(t' \text{ is a value}) \vee (\exists t''.t' \rightarrow t'')$	By progress using (3)

Subcase 2.1: t' is a value

(5) $t' \rightarrow^* t'$	By m-step reflexivity
(6) $t \rightarrow^* t'$	By m-step transativity using (2) and (5)
(7) t is safe	By def. and (6) since t' is a value

Subcase 2.2: $\exists t''.t' \rightarrow t''$

(8) $t' \rightarrow^* t'$	By m-step reflexivity
(9) $t \rightarrow^* t'$	By m-step transativity
(10) $\exists t''.t' \rightarrow t''$	Restating Subcase
(11) t is safe	By def using (9) and (10)

■

Q2.2) S_T Implies Type Safety

To start off, I will correct the typos made in the assignment so that it is clear what we are working with. I am using '⇒' for logical implication.

$$\begin{aligned} v \in V_{\texttt{Bool}} &\iff (v = \text{true}) \vee (v = \text{false}) \\ (\lambda x : S.t) \in V_{S \rightarrow T} &\iff \forall s\,(s \in S_S \Rightarrow [s/x]t \in S_T) \\ t \in S_T &\iff \forall s\left[(t \rightarrow^* s \wedge \neg\exists s'.s \rightarrow s') \Rightarrow s \in V_T\right] \end{aligned}$$

We now proceed with stating the desired lemma and the proof.

Lemma 1: if $t \in S_T$ then $\forall s \, [(t \rightarrow^* s) \Rightarrow (s \text{ is a value} \vee \exists s'.s \rightarrow s')]$

Proof: Suppose that $t \in S_T$, which is equivalent to: $\forall s \, [(t \rightarrow^* s \wedge \neg\exists s'.s \rightarrow s') \Rightarrow s \in V_T]$

In order to prove the desired implication, we suppose the antecedent. So suppose that $t \rightarrow^* s$.

We proceed with cases on $\exists s'.s \rightarrow s'$.

Case 1: $\exists s'.s \rightarrow s'$

In this case, t is_safe follows immediately by definition of is_safe since it is true that $\exists s'.s \rightarrow s'$ by the subcase.

Case 2: $\neg\exists s'.s \rightarrow s'$

We then have that $(t \rightarrow^* s \wedge \neg\exists s'.s \rightarrow s')$, which by our original supposition means that $s \in V_T$. But by definition of V_T, then s is a value.

So t is_safe by definition since $t \rightarrow^* s$ and s is a value.

■

Q2.3) Well-Typedness Implies Semantic Safety

First off, the notion of safe substitution is defined as:

$$\frac{}{\cdot \in R_{\cdot}}\text{ SUB-EMPT} \qquad\qquad \frac{\sigma \in R_\Gamma \quad s \in S_T}{\sigma, (s/x) \in R_{\Gamma, x:T}}\text{ SUB-ADD}$$

It is now possible to generalize the statement to the following lemma.

Lemma 2.3: if $\Gamma \vdash t : T$ and $\sigma \in R_\Gamma$ then $[\sigma]t \in S_T$

Proof: By induction on the typing derivation of $\Gamma \vdash t : T$, call it D.

Case 1: $D = \dfrac{}{\Gamma \vdash \text{true} : \texttt{Bool}}\text{ T-TRUE}$

Since true is ground, by definition of substitution, we have that $[\sigma]$true=true. So what is to be shown is true $\in S_{\texttt{Bool}}$.

By definition, this is equivalent to showing that $\forall s\,[(\text{true} \rightarrow^* s \wedge \neg\exists s'.s \rightarrow s') \Rightarrow s \in V_{\texttt{Bool}}]$.
So to prove the implication, suppose the antecedent:
$(\text{true} \rightarrow^* s \wedge \neg\exists s'.s \rightarrow s')$.

But by definition of our small-step semantics, $\neg\exists s'.\text{true} \rightarrow s'$.

So by definition of the m-step relation, it must be the case that s=true.

By definition of $V_{\texttt{Bool}}$ then, $s \in V_{\texttt{Bool}}$, which completes this case.

Case 2: $D = \dfrac{}{\Gamma \vdash \text{false} : \texttt{Bool}}\text{ T-FALSE}$

This case is exactly analogous to case 1 and is left out for brevity.

Case 3: $D = \dfrac{\Gamma \vdash t : \texttt{Bool} \quad \Gamma \vdash t_1 : T \quad \Gamma \vdash t_2 : T}{\Gamma \vdash (\text{if } t \text{ then } t_1 \text{ else } t_2) : T}\text{ T-IF}$

By definition of substitution, what is to be shown is:
$(\text{if } [\sigma]t \text{ then } [\sigma]t_1 \text{ else } [\sigma]t_2) \in S_T$
which by definition can be written as:

$$\forall s\,\big[((\text{if } [\sigma]t \text{ then } [\sigma]t_1 \text{ else } [\sigma]t_2) \rightarrow^* s \wedge \neg\exists s'.s \rightarrow s') \Rightarrow s \in V_T\big]$$

To prove this implication, we start off by supposing the antecedent.

(1) $(\text{if } [\sigma]t \text{ then } [\sigma]t_1 \text{ else } [\sigma]t_2) \rightarrow^* s$ Supposing the antecedent
(2) $\neg\exists s'.s \rightarrow s'$ Supposing the antecedent
(3) $\sigma \in R_\Gamma$ By assumption
(4) $[\sigma]t \in S_{\texttt{Bool}}$ By IH on D_1 the left subderiv. using (3)
(5) $\forall s\,[([\sigma]t \rightarrow^* s \wedge \neg\exists s'.s \rightarrow s') \Rightarrow s \in V_{\texttt{Bool}}]$ By def of S_T using (4)
(6) $[\sigma]t \rightarrow^* \text{true} \vee [\sigma]t \rightarrow^* \text{false}$ By def of $V_{\texttt{Bool}}$ using (5)

Subcase 3.1: $[\sigma]t \rightarrow^* \text{true}$

(7) $(\text{if } [\sigma]t \text{ then } [\sigma]t_1 \text{ else } [\sigma]t_2) \rightarrow^* [\sigma]t_1$ By lemma 1 introduced in Q1 and subcase
(8) $[\sigma]t_1 \in S_T$ By IH on D_2, the mid subderiv, using (3)

(9) $[\sigma]t_1 \to^* [\sigma]t_1$ By transativity of m-step
(10) $[\sigma]t_1 \in V_T$ By def of S_T using (8), (2)
(11) (if $[\sigma]t$ then $[\sigma]t_1$ else $[\sigma]t_2$) $\in S_T$ By def. of S_T using (7), (2) and (10)

Subcase 3.2: $[\sigma]t \to^*$ false

This subcase is entirely analogous to subcase 3.1 and is therefore left out for brevity.

This completes the proof, since all relevant cases (as said on the discussion board) have been shown.

■

Q2.4) Extension to Cross Products

The semantic interpretation of values is extended to cross products as:

$$\langle t_1, t_2 \rangle \in S_{T_1 \times T_2} \iff (t_1 \in S_{T_1}) \land (t_2 \in S_{T_2})$$

Q2.5) Extension of Proof for Cross-Products

Since there is only one case of the proof added, I will state it as a lemma and prove it directly.

Lemma 2.5: if $\Gamma \vdash t : (T_1 \times T_2)$ and $\sigma \in R_\Gamma$ then $[\sigma]t \in S_{T_1 \times T_2}$

Proof: By induction on the typing derivation of $\Gamma \vdash t : (T_1 \times T_2)$. Clearly, $t = \langle t_1, t_2 \rangle$ for some t_1 and t_2. So:

$$D = \quad \frac{\Gamma \vdash t_1 : T_1 \qquad \Gamma \vdash t_2 : T_2}{\Gamma \vdash \langle t_1, t_2 \rangle : (T_1 \times T_2)}$$

What we want to show is: $[\sigma]t \in S_{T_1 \times T_2}$, which since $t = \langle t_1, t_2 \rangle$ means that it is equivalent to prove $\langle [\sigma]t_1, [\sigma]t_2 \rangle \in S_{T_1 \times T_2}$.

But by definition of $S_{T_1 \times T_2}$ is equivalent to having to prove: $([\sigma]t_1 \in S_{T_1}) \land ([\sigma]t_2 \in S_{T_2})$.

(1) $\sigma \in R_\Gamma$	By assumption
(2) $[\sigma]t_1 \in S_{T_1}$	By IF on the left subderiv. of D using (1)
(3) $[\sigma]t_2 \in S_{T_2}$	By IF on the right subderiv. of D using (1)

Which completes the proof of the theorem since (2) and (3) are what was to be shown.

■