**Exercise 3.5.14:**

**Proof:** by induction on the structure of t

**Case 1:** t is a value

this case is not applicable because t is in normal form.

**Case 2:** t = succ t1

in the evaluation rules the only rule that has succ on the left hand side is the E-SUCC rule, so for deriving t → t’ and t→ t’’ we use the same rule. Using induction hypothesis for smaller derivations t1→ t1’ and t1 → t1”, we can see that t1’ = t1”, so succ t1’ = succ t1’’.

**Case 3:** t = pred t1

in the evaluation rules we have 3 rules that has pred on the left hand side (E-PRED, E-PREDZERO, E-PREDSUCC). But the different is the arguments of these rules (E-PRED get a non value, E-PREDZERO gets 0, E-PREDSUCC gets a succ nv) so for derivation t→ t’ and t→ t’’ we use the same rule. If the rule is E-PRED we use induction hypothesis for smaller derivations t1 → t1’ and t1 → t1’’, we see that t1’ = t1’’, so pred t1’ = pred t1’’. Else if the rule is E-PREDZERO or E-PREDSUCC we can immediately see the result.

**Case 4:** t = iszero t1

in the evaluation rules we have 3 rules that has iszero on the left hand side (E-ISZERO, E-ISZEROZERO, E-ISZEROSUCC). But the different is the arguments of these rules (E- ISZERO get a non value, E-P ISZEROZERO gets 0, E-ISZEROSUCC gets a succ nv) so for derivation t→ t’ and t→ t’’ we use the same rule. If the rule is E- ISZERO we use induction hypothesis for smaller derivations t1 → t1’ and t1 → t1’’, we see that t1’ = t1’’, so iszero t1’ = iszero t1’’. Else if the rule is E-ISZEROZERO or E-ISZEROSUCC we can immediately see the result.

**Exercise 3.5.16:**

**Theorem: t →\* t' (in the previous treatment) S.T. t' is stuck iff t→\* wrong (in the new treatment)**

***Proof:We will prove it in both directions***

***Direction left to right:***

we will prove this part by proving the following Lemma:

**Lemma 1:** if t is stuck then t→\* wrong (in the new treatment)

**Proof:** by induction on structure of t

**Case 1:** t is either true or false or 0

since the assumption of Lemma 1 is that t is stuck this case is not applicable.

**Case 2:** t = if t1 then t2 else t3

since t is stuck, t1 cannot be true or false because the rule would be E-IFTRUE or E-IFFALSE and then t would not be stuck. So t1 has to be in normal form.

**Case 2.1:** t1 is either true or false or 0

this case is not applicable.

**Case 2.2:** t1 = if t11 then t21 else t31

we found out that t1 is in normal form, so it is stuck. We use induction hypothesis for smaller part of Lemma 1, so t1 →\* wrong. The rule would be like t →\* if wrong then t2 else t3 and the rule E-IFWRONG goes to wrong.

**Case 2.3:** t1 = succ t11

if t11 is a numeric value by using the rule E-IF-WRONG we will have t → wrong. If t1 is not a numeric value then t1 is stuck and using induction hypothesis t1→\* wrong, we see that t →\* if wrong then t2 else t3. So the rule E-IF-WRONG goes to wrong.

**Case 2.4:** t1 = pred t11

if t11 is a numeric value by using the rule E-IF-WRONG we will have t → wrong. If t1 is not a numeric value then t1 is stuck and using induction hypothesis t1→\* wrong, we see that t →\* if wrong then t2 else t3. So the rule E-IF-WRONG goes to wrong.

**Case 2.5:** t1 = iszero t11

if t11 is a numeric value by using the rule E-IF-WRONG we will have t → wrong. If t1 is not a numeric value then t1 is stuck and using induction hypothesis t1→\* wrong, we see that t →\* if wrong then t2 else t3. So the rule E-IF-WRONG goes to wrong.

**Case 3:** t = succ t1

since t is stuck, t1 cannot be a value so it has to be in normal form. For t11, we can think of being true/false or it is not a value; if t1 is true/false we use rule E-SUCC-WRONG and it goes to wrong in new treatment, and if t1 is not a value so it is stuck and this is a smaller part of the Lemma 1 and we use induction hypothesis for t1→\* wrong.

**Case 4:** t = pred t1

since t is stuck t1 has to be in normal form. For t1, we can think of being true/false or it is not a value; if t1 is true/false we use rule E-PRED-WRONG and it goes to wrong in new treatment, and if t1 is not a value so it is stuck and this is a smaller part of the Lemma 1 and we use induction hypothesis for t1→\* wrong.

**Case 5:** t = iszero t1

since t is stuck t1 has to be in normal form. For t1, we can think of being true/false or it is not a value; if t1 is true/false we use rule E-ISZERO-WRONG and it goes to wrong in new treatment, and if t1 is not a value so it is stuck and this is a smaller part of the Lemma 1 and we use induction hypothesis for t1→\* wrong.

***Direction right to left:***

we will prove this part by proving the following Lemma:

**Lemma 2:** if t → t’ in the new treatment and t’ has wrong in it, then t is stuck in the previous treatment

**Proof:** by induction on derivation in new treatment

**Case 1:** the rule for deriving t → t’ is E-IF-WRONG

the rule will be if t then t1 else t2 → wrong, so t is not true or false and the rule goes to stuck.

**Case 2:** the rule for deriving t → t’ is E-SUCC-WRONG

the rule will be succ t → wrong, so t is not a numeric value and the rule goes to stuck.

**Case 3:** the rule for deriving t → t’ is E-PRED-WRONG

the rule will be pred t → wrong, so t is not a numeric value and the rule goes to stuck.

**Case 4:** the rule for deriving t → t’ is E-ISZERO-WRONG

the rule will be iszero t → wrong, so t is not a numeric value and the rule goes to stuck.

**Exercise 9.3.9:**

**Theorem: If Γ t : T and t → t’, then Γ t’ : T.**

**Proof:**

**Case T-VAR:**

This cannot happen since there is no evaluation rule to evaluate a variable

**Case T-ABS :**

This also cannot happen for the same reason

**Case T-APP:**

Case 3.1: is a value and is also a value

Then by E-APPABS

By Inversion Lemma 2, we can know deduct

and by using Lemma “Preservation of types under substitution” we can know

Case 3.2:

Since we have , then by induction hypothesis we will know

Then, according to T-APP

Case 3.3:

The proof of this case is similar to the previous one.

**Case T-True:**

This cannot happen since there is no evaluation rule to evaluate true

**Case T-False:**

This also cannot happen for the same reason

**Case T-IF:**

Case 6.1:

Then by E-IFTrue,

Case 6.2:

Proof just like case 6.1

Case 6.3:

By induction hypothesis, we will know

Then by T-IF