COMPSCI 3MI3

Assignment 8

1 Solution Set

1.1 Q1

Using the definition of simply typed λ-Calculus presented in the slides from topic 9, we define $\lambda^{\mathcal{E}}$ as our external calculus composed of simply typed λ-Calculus enriched with the Unity type and term, as well as the ; term, evaluation rules E-Seq, E-SeqNext and typing rule T-Seq. Similarly we define $\lambda^{\mathcal{I}}$ as our internal calculus composed of simply typed λ-Calculus and the Unit type and term *only*.
We then define an elboration function $e : \lambda^{\mathcal{E}} \rightarrow \lambda^{\mathcal{I}}$, which translates terms from $\lambda^{\mathcal{E}}$ to $\lambda^{\mathcal{I}}$. We define e with the following inference rule:

$$\frac{x \notin FV(t_2') \qquad e(t_1) = t_1' \qquad e(t_2) = t_2'}{e(t_1 \,;\, t_2) = (\lambda x : \text{Unit}.\, t_2')\; t_1'}$$

For all other cases where the argument passed to e does not use the sequencing operator (;) we define e to simply return its argument such that $e(t) = t$ for the term t.

(a) We will prove the property $P(t) = t \xrightarrow{\mathcal{E}} t' \implies e(t) \xrightarrow{\mathcal{I}} e(t')$ for all terms t in $\lambda^{\mathcal{E}}$ by structural induction on t. We proceed by case analysis on t. We can group the cases on t into two groups: those for which an inference rule on e exists and those for which a rule does not exist.

For cases on t in which $e(t) = t$ the proof is trivial. By our assumption we know $t \xrightarrow{\mathcal{E}} t'$ which is equivalent to $t \xrightarrow{\mathcal{I}} t'$ by $e(t) = t$. Since the only the evaluations rules in $\lambda^{\mathcal{E}}$ that are not in $\lambda^{\mathcal{I}}$ involve sequencing, and t does not contain sequencing, we can say that t also evaluates to t' in our internal calculus. Thus our property holds. This case includes all our base cases on t.

For rest of the cases on t, there exists an inference rule for t on e. The term must then be of the form $t = t_1 \,;\, t_2$ As part of our induction step we assume our property holds for subterms of t, t_1 and t_2. We will prove our property holds for t. From our assumption we know $t \xrightarrow{\mathcal{E}} t'$, thus the evaluation derivation must be one of the following:

Case: E-Seq
If t evaluates to t' by E-Seq then we know $t = t_1 \,;t_2$, $t' = t_1' \,;t_2$ and have the premise $t_1 \xrightarrow{\mathcal{E}} t_1'$. By the definition of e we know $e(t) = (\lambda x : \text{Unit}.\, t_2'')\, t_1''$ with the premises $x \notin FV(t_2')$, $e(t_1) = t_1''$ and $e(t_2) = t_2''$. By our induction hypothesis on t_1 we know if $t_1 \xrightarrow{\mathcal{E}} t_1'$ then $e(t_1) \xrightarrow{\mathcal{I}} e(t_1')$, and by our premise $e(t_1) = t_1''$ we know $e(t_1) \xrightarrow{\mathcal{I}} e(t_1')$ is equivalent to $t_1'' \xrightarrow{\mathcal{I}} e(t_1')$. Thus, by E-App2 and the premise $t_1'' \xrightarrow{\mathcal{I}} e(t_1')$ we have $(\lambda x : \text{Unit}.\, t_2'')\, t_1'' \xrightarrow{\mathcal{I}} (\lambda x : \text{Unit}.\, t_2'')\, y$ where $y = e(t_1')$. By our inference rule on e with the premises $x \notin FV(t_2')$, $e(t_1') = y$ and $e(t_2) = t_2''$ we have $(\lambda x : \text{Unit}.\, t_2'')\, y = e(t_1' \,;t_2)$. Therefore we have $e(t) \xrightarrow{\mathcal{I}} e(t')$ and our property holds for the E-Seq case.

Case: E-SeqNext
If t evaluates to t' by E-SeqNext then we know $t = \texttt{unit} \,;\, t_2$ and $t' = t_2$. By the definition of e we know $e(t) = (\lambda x : \text{Unit}.\, t_2')\ \texttt{unit}$ and have the premises $x \notin FV(t_2')$ and $e(t_2) = t_2'$. By E-AppAbs we know $(\lambda x : \text{Unit}.\, t_2')\ \texttt{unit} \xrightarrow{\mathcal{I}} t_2'$, we throw away the argument x since we know $x \notin FV(t_2')$. This is equivalent to $e(t) \xrightarrow{\mathcal{I}} t_2'$, and by premises $e(t') = e(t_2) = t_2'$. Therefore we've shown $e(t) \xrightarrow{\mathcal{I}} e(t')$ and our property holds.

Thus we've shown that our property holds for terms for which there exists an inference rule on e as part of our induction step. Thus our induction step holds for all cases.

Therefore, we've shown that our property $P(t)$ holds for all terms t in $\lambda^{\mathcal{E}}$ by structural induction on t. $\square$

(b) We will prove the property $P(t) = e(t) \xrightarrow{\mathcal{I}} e(t') \implies t \xrightarrow{\mathcal{E}} t'$ for all terms t in $\lambda^{\mathcal{E}}$ by structural induction on t. We proceed by case analysis on t. We can group the cases on t into two groups: those for which an inference rule on e exists and those for which a rule does not exist.

For cases on t in which $e(t) = t$ the proof is trivial. By our assumption we know $e(t) \xrightarrow{\mathcal{I}} e(t')$, which is equivalent to $t \xrightarrow{\mathcal{I}} t'$ by $e(t) = t$. We know all evaluation rules in $\lambda^{\mathcal{I}}$ are present in $\lambda^{\mathcal{E}}$, thus we also have $t \xrightarrow{\mathcal{E}} t'$ and our property holds. This case includes all our base cases on t.

For rest of the cases on t, there exists an inference rule for t on e. The term must then be of the form $t = t_1 \,;t_2$, with $e(t) = (\lambda x : \text{Unit}.\, t_2')\, t_1'$ and the premises $x \notin FV(t_2')$, $e(t_1) = t_1'$ and $e(t_2) = t_2'$. As part of our induction step we assume our property holds for subterms of t: t_1 and t_2. We will prove our property holds for t. From our assumption we know $e(t) \xrightarrow{\mathcal{I}} e(t')$, thus the evaluation derivation must be one of the following:

Case: E-App2
If $e(t)$ evaluates to $e(t')$ by E-App2 then we know $e(t') = (\lambda x : \text{Unit}.\, t_2')\ t_1''$ and have the premise $t_1' \xrightarrow{\mathcal{I}} t_1''$. Setting $e(y) = t_1''$, by the definition of e with the premises $x \notin FV(t_2')$ and $e(t_2) = t_2'$ we have $t' = y\,; t_2$. By our induction hypothesis on t_1 with the premises $e(t_1) = t_1'$ and $t_1' \xrightarrow{\mathcal{I}} t_1''$ we know $t_1 \xrightarrow{\mathcal{E}} y$. Since $t_1 \xrightarrow{\mathcal{E}} y$, by E-Seq we know $t_1\,; t_2 \xrightarrow{\mathcal{E}} y\,; t_2$, which is $t \xrightarrow{\mathcal{E}} t'$. Thus our property holds for the E-App2 case.

Case: E-AppAbs
If $e(t)$ evaluates to $e(t')$ by E-AppAbs then we know $e(t') = t_2'$ since $x \notin FV(t_2')$ by our premises. By our premises $e(t_2) = t_2'$ thus $t' = t_2$. If $e(t)$ is well typed then we know t_1 must be a value of type Unit. The only value of type Unit is `unit`. Thus by setting $t_1 =$ `unit` and using the E-SeqNext evaluation rule we have $t \xrightarrow{\mathcal{E}} t'$ and our property holds for the E-AppAbs case.

Thus we've shown that our property holds for terms for which there exists an inference rule on e as part of our induction step. Thus our induction step holds for all cases.

Therefore, we've shown that our property $P(t)$ holds for all terms t in $\lambda^{\mathcal{E}}$ by structural induction on t. □

(c) We will prove the property $P(t) = \Gamma \vdash^{\mathcal{E}} t : T \implies \Gamma \vdash^{\mathcal{I}} e(t) : T$ for all terms t in $\lambda^{\mathcal{E}}$ by structural induction on t. We proceed by case analysis on t. We can group the cases on t into two groups: those for which an inference rule on e exists and those for which a rule does not exist.

For cases on t in which $e(t) = t$ the proof is trivial. By our assumption we know t is well-typed as type T in context Γ in the external calculus, thus there exists an typing rule for $\Gamma \vdash^{\mathcal{E}} t : T$. Since we know there does not exist an inference rule on e for t then t does not contain the sequencing operator and $e(t) = t$. Thus we know $\Gamma \vdash^{\mathcal{I}} e(t) : T$ is equivalent to $\Gamma \vdash^{\mathcal{I}} t : T$. Since the only the typing rules in $\lambda^{\mathcal{E}}$ that are not in $\lambda^{\mathcal{I}}$ involve sequencing, and t does not contain sequencing, we can say that t is also well-typed as type T in our internal calculus. Thus our property holds. This case includes all our base cases on t.

For rest of the cases on t, there exists an inference rule for t on e. The term must then be of the form $t = t_1\,; t_2$ As part of our induction step we assume our property holds for subterms of t: t_1 and t_2. We will prove our property holds for t. From our assumption we know $\Gamma \vdash^{\mathcal{E}} t : T$, thus the typing derivation must be T-Seq.

If t is typed T in context Γ by T-Seq then we know $t = t_1\,; t_2$ and have the premises $\Gamma \vdash^{\mathcal{E}} t_1 : \text{Unit}$ and $\Gamma \vdash^{\mathcal{E}} t_2 : T$. By the definition of e we know $e(t) = (\lambda x : \text{Unit}.\, t_2')\, t_1'$,

with premises $x \notin FV(t'_2)$, $e(t_1) = t'_1$ and $e(t_2) = t'_2$. By our induction hypothesis on t_1 and with the premises $\Gamma \vdash^{\mathcal{E}} t_1 : \text{Unit}$ and $e(t_1) = t'_1$, we know $\Gamma \vdash^{\mathcal{I}} t'_1 : \text{Unit}$. Similarly, by our induction hypothesis on t_2 with the premises $\Gamma \vdash^{\mathcal{E}} t_2 : T$ and $e(t_2) = t'_2$ we know $\Gamma \vdash^{\mathcal{T}} t'_2 : T$.
Since $x \notin FV(t'_2)$ we can assume there exists no typing information about x, which implies $x \notin dom(\Gamma)$. Thus, we have $\Gamma \vdash^{\mathcal{T}} t'_2 : T$ and $x \notin dom(\Gamma)$, by the weakening lemma we have $\Gamma, x : \text{Unit} \vdash^{\mathcal{T}} t'_2 : T$. Therefore by T-Abs we have $\Gamma \vdash^{\mathcal{I}} (\lambda x : \text{Unit}.\, t'_2) : \text{Unit} \Rightarrow T$.
Since we know $\Gamma \vdash^{\mathcal{I}} t'_1 : \text{Unit}$ and $\Gamma \vdash^{\mathcal{I}} (\lambda x : \text{Unit}.\, t'_2) : \text{Unit} \Rightarrow T$, then by T-App we can conclude $\Gamma \vdash^{\mathcal{I}} (\lambda x : \text{Unit}.\, t'_2)\ t'_1 : T$. Thus $\Gamma \vdash^{\mathcal{I}} e(t) : T$ and our property holds.

Thus we've shown that our property holds for terms for which there exists an inference rule on e as part of our induction step. Thus our induction step holds for all cases.

Therefore, we've shown that our property $P(t)$ holds for all terms t in $\lambda^{\mathcal{E}}$ by structural induction on t. □

(d) We will prove the property $P(t) = \Gamma \vdash^{\mathcal{I}} e(t) : T \implies \Gamma \vdash^{\mathcal{E}} t : T$ for all terms t in $\lambda^{\mathcal{E}}$ by structural induction on t. We proceed by case analysis on t. We can group the cases on t into two groups: those for which an inference rule on e exists and those for which a rule does not exist.

For cases on t in which $e(t) = t$ the proof is trivial. By our assumption we know $\Gamma \vdash^{\mathcal{I}} e(t) : T$, which is equivalent to $\Gamma \vdash^{\mathcal{I}} t : T$ by $e(t) = t$. We know all typing rules in $\lambda^{\mathcal{I}}$ are present in $\lambda^{\mathcal{E}}$, thus we also have $\Gamma \vdash^{\mathcal{E}} t : T$ and our property holds. This case includes all our base cases on t.

For rest of the cases on t, there exists an inference rule for t on e. The term must then be of the form $t = t_1\,;t_2$, with $e(t) = (\lambda x : \text{Unit}.\, t'_2)\ t'_1$ and the premises $x \notin FV(t'_2)$, $e(t_1) = t'_1$ and $e(t_2) = t'_2$. As part of our induction step we assume our property holds for subterms of t: t_1 and t_2. We will prove our property holds for t. From our assumption we know $\Gamma \vdash^{\mathcal{I}} e(t) : T$, thus the typing derivation must be T-App.

If $e(t)$ is typed T in context Γ by T-App then we know $T = T_2$ and have the premises $\Gamma \vdash^{\mathcal{I}} (\lambda x : \text{Unit}.\, t'_2) : T_1 \Rightarrow T_2$ and $\Gamma \vdash^{\mathcal{I}} t'_1 : T_1$. Since we have $\Gamma \vdash^{\mathcal{I}} (\lambda x : \text{Unit}.\, t'_2) : T_1 \Rightarrow T_2$, by T-Abs we have the premise $\Gamma, x : T_1 \vdash^{\mathcal{I}} t'_2 : T_2$, with $T_1 = \text{Unit}$. Since $x \notin FV(t'_2)$ we can discard x from our typing context without losing information about t'_2 such that $\Gamma \vdash^{\mathcal{I}} t'_2 : T_2$, and since $T_1 = \text{Unit}$ we have $\Gamma \vdash^{\mathcal{I}} t'_1 : \text{Unit}$. By applying these premises to our induction hypothesis for t_1 and t_2 since we know $e(t_1) = t'_1$ and $e(t_2) = t'_2$ we can show $\Gamma \vdash^{\mathcal{E}} t_1 : \text{Unit}$ and $\Gamma \vdash^{\mathcal{E}} t_2 : T_2$. Therefore, by T-Seq we know $\Gamma \vdash^{\mathcal{E}} (t_1\,;t_2) : T_2$ which is $\Gamma \vdash^{\mathcal{E}} t : T$. Thus our property holds.

Thus we've shown that our property holds for terms for which there exists an inference rule on e as part of our induction step. Thus our induction step holds for all cases.

Therefore, we've shown that our property $P(t)$ holds for all terms t in $\lambda^{\mathcal{E}}$ by structural induction on t. $\square$