15-814 Homework 1 Solutions

September 25, 2017

1 Arithmetic

Task 1 Prove the following inversion lemma:

(If Inversion) If \( \Gamma \vdash \text{if}(e_1, e_2, e_3) : \tau \), then \( \Gamma \vdash e_1 : \text{bool}, \Gamma \vdash e_2 : \tau \), and \( \Gamma \vdash e_3 : \tau \).

This seems immediate, but really follows from the induction principle for the typing judgment. (Tip: Prove that for all \( \Gamma, e, \tau \) such that \( \Gamma \vdash e : \tau \), if \( e = \text{if}(e_1, e_2, e_3) \) for some \( e_1, e_2, e_3 \), then \( \Gamma \vdash e_1 : \text{bool}, \Gamma \vdash e_2 : \tau \), and \( \Gamma \vdash e_3 : \tau \).)

Solution: We show that for all \( \Gamma, e, \tau \) such that \( \Gamma \vdash e : \tau \), if \( e = \text{if}(e_1, e_2, e_3) \) for some \( e_1, e_2, e_3 \), then \( \Gamma \vdash e_1 : \text{bool}, \Gamma \vdash e_2 : \tau \), and \( \Gamma \vdash e_3 : \tau \).

The proof proceeds by induction on typing judgments (note that we are considering induction where the premises of the judgments are available as assumptions).

Case (If): Suppose \( \Gamma \vdash \text{if}(e_1, e_2, e_3) : \tau \), we need to show \( \Gamma \vdash e_1 : \text{bool}, \Gamma \vdash e_2 : \tau \), and \( \Gamma \vdash e_3 : \tau \). However, these follow directly from the premises of the judgment.

Cases (Hyp), (Num), (True), (False), (Plus), (Times) and (Leq): For each of these rules, we note that the conclusion is not syntactically of the form \( e = \text{if}(e_1, e_2, e_3) \). Therefore, the required property is trivially true.

Task 2 Prove unicity of typing for this language.

(Unicity of Typing) For any \( \Gamma, e, \tau, \tau' \) such that \( \Gamma \vdash e : \tau \) and \( \Gamma \vdash e : \tau' \), we have \( \tau = \tau' \).

You may assume that any variable appears at most once in a given context.

Solution: We proceed by induction on the typing judgement \( \Gamma \vdash e : \tau \).

Case \( e = x \) (Hyp): By assumption, we have \( \Gamma, x : \tau \vdash x : \tau \) and \( \Gamma, x : \tau \vdash x : \tau' \). By inversion on the latter judgment and noting that only one instance of \( x \) occurs in the context, we have \( \tau = \tau' \).

Case \( e = \pi \) (Num): By assumption, we have \( \Gamma \vdash \pi : \text{nat} \) and \( \Gamma \vdash \pi : \tau' \), i.e. \( \tau = \text{nat} \). By inversion on the latter judgment, we have that \( \tau' = \text{nat} = \tau \).

Cases (True) and (False): Similar to the (Num) case.

Case \( e = e_1 + e_2 \) (Plus): By assumption, we have \( \Gamma \vdash e_1 + e_2 : \text{nat} \) and \( \Gamma \vdash e_1 + e_2 : \tau' \), i.e. \( \tau = \text{nat} \). By inversion on the latter judgment, we have that \( \tau' = \text{nat} = \tau \).

Cases (Times) and (Leq): Similar to the (Plus) case.

Case \( e = \text{if}(e_1, e_2, e_3) \) (If): By assumption, we have \( \Gamma \vdash \text{if}(e_1, e_2, e_3) : \tau \), \( \Gamma \vdash e_2 : \tau \), and \( \Gamma \vdash \text{if}(e_1, e_2, e_3) : \tau' \). By inversion on the last typing judgment, we have \( \Gamma \vdash e_2 : \tau' \). Hence, by I.H. on \( \Gamma \vdash e_2 : \tau \), we have \( \tau = \tau' \).
2 Days of the Week

Task 3 Define the next\((d)\) is \(d'\) judgement which takes a day \(d\) and returns the next day, \(d'\). You should assume that the next day from Sunday is Friday.

Solution:

\[
\begin{align*}
\text{next}(\text{Fri}) & \text{ is } \text{Sat} \\
\text{next}(\text{Sat}) & \text{ is } \text{Sun} \\
\text{next}(\text{Sun}) & \text{ is } \text{Fri}
\end{align*}
\]

Task 4 Define the \(\text{nextn}(n, d)\) is \(d'\) judgement which takes a natural number \(n\), a day \(d\), and returns the \(n^{th}\) day after \(d\). You should make use of the inductive definition of \(\text{nat}\).

Solution:

\[
\begin{align*}
\text{nextn}(Z, d) & \text{ is } d \\
\text{nextn}(S(n), d) & \text{ is } d''
\end{align*}
\]

Task 5 Using your answer to Task 4 extend the static and dynamic semantics of \(e\) with the cases for \(d\) and \(\text{nextn}(e_1, e_2)\). Your definition should satisfy progress and type preservation, which you will need to prove below.

Solution:

- Statics

\[
\begin{align*}
\Gamma & \vdash d : \text{day} \\
\Gamma & \vdash e_1 : \text{nat} \\
\Gamma & \vdash e_2 : \text{day} \\
\Gamma & \vdash \text{nextn}(e_1, e_2) : \text{day}
\end{align*}
\]

- Dynamics

\[
\begin{align*}
\frac{e_1 \mapsto e'_1}{\text{nextn}(e_1, e_2) \mapsto \text{nextn}(e'_1, e_2)} \quad \text{(nextn-S1)} \\
\frac{e_1 \text{ val} \quad e_2 \mapsto e'_2}{\text{nextn}(e_1, e_2) \mapsto \text{nextn}(e_1, e'_2)} \quad \text{(nextn-S2)} \\
\frac{\text{nextn}(n, d) \text{ is } d'}{\text{nextn}(n, d) \mapsto d'} \quad \text{(nextn-I)}
\end{align*}
\]

3 Type Safety

We will now show type safety for the language, including your extension in Task 5 by proving progress and type preservation.

Task 6 Carefully state a canonical forms lemma for your extended semantics. You do not have to prove the lemma, and you may assume it for the rest of your proof.

Solution: Lemma 1 (Canonical Forms) If \(e\) val and \(\vdash e : \tau\), then

- if \(\tau = \text{nat}\) then \(e = \overline{n}\) for some natural number \(n\),
- if \(\tau = \text{bool}\) then \(e = \overline{t}\) or \(e = \overline{f}\),
- if \(\tau = \text{day}\) then \(e = \overline{d}\) for some day \(d\).
Task 7 Prove progress for your extended semantics, i.e.

(Progress) If $\vdash e : \tau$, then either $e \text{ val}$ or there exists $e'$ such that $e \rightarrow e'$.

Solution: We proceed by induction on the typing judgment $\vdash e : \tau$. (I write the form of $\vdash e : \tau$ followed by the rulename for each relevant case).

Case (HYP): This case is vacuous since we are considering closed terms.

Case $\vdash \pi : \text{nat} \, (\text{Num})$: We have $\pi \text{ val}$ by (NUM-V) so we are done.

Cases (TRUE), (FALSE), (DAY): Similar to (NUM).

Case $\vdash e_1 + e_2 : \text{nat} \, (\text{PLUS})$: The premises of the rule are: $\vdash e_1 : \text{nat}$ and $\vdash e_2 : \text{nat}$. By I.H. on the first premise, we have that either $e_1 \text{ val}$ or $e_1 \rightarrow e'_1$. In the first case, we may further apply the I.H. on the second premise to get that either $e_2 \text{ val}$ or $e_2 \rightarrow e'_2$.

Subcase $e_1 \text{ val}, e_2 \text{ val}$: Since $\vdash e_1 : \text{nat}$ and $\vdash e_2 : \text{nat}$, by canonical forms, we have that $e_1 = \overline{n_1}, e_2 = \overline{n_2}$ for some $n_1, n_2$. Thus, $\overline{n_1} + \overline{n_2} \rightarrow \overline{n_1 + n_2}$ by (PLUS-I).

Subcase $e_1 \text{ val}, e_2 \rightarrow e'_2$: Then we have $e_1 + e_2 \rightarrow e_1 + e'_2$ by (PLUS-S2).

Subcase $e_1 \rightarrow e'_1$: Then we have $e_1 + e_2 \rightarrow e'_1 + e_2$ by (PLUS-S1).

Cases (TIMES) (LEQ): Similar to (PLUS)\(^1\)

Case $\vdash \text{if}(e_1, e_2, e_3) : \tau \, (\text{IF})$ (abbreviated): From the premise of the rule, we have $\vdash e_1 : \text{bool}$. By I.H. on $e_1$, we have that either $e_1 \text{ val}$ or $e_1 \rightarrow e'_1$. In the first case, canonical forms gives us that $e_1 = \overline{\ttt}$ or $e_1 = \overline{\www}$, and we may respectively apply (IF-I1) or (IF-I2). In the latter case, we may apply (IF-S).

Case $\vdash \text{next}(e_1, e_2) : \text{day} \, (\text{NEXTN})$\(^2\): The premises of the rule are: $\vdash e_1 : \text{nat}$ and $\vdash e_2 : \text{day}$. By I.H. on the first premise, we have that either $e_1 \text{ val}$ or $e_1 \rightarrow e'_1$. In the first case, we may further apply the I.H. on the second premise to get that either $e_2 \text{ val}$ or $e_2 \rightarrow e'_2$.

Subcase $e_1 \text{ val}, e_2 \text{ val}$: Since $\vdash e_1 : \text{nat}$ and $\vdash e_2 : \text{day}$, by canonical forms, we have that $e_1 = \overline{n}, e_2 = \overline{d}$ for some $n, d$. Moreover, we have that $\text{next}(n, d)$ is $d'$ for some $d'$. Thus, $\overline{n} \text{ next}(\overline{n}, \overline{d}) \rightarrow \overline{d'}$ by (NEXTN-I).

Subcase $e_1 \text{ val}, e_2 \rightarrow e'_2$: Then we have $\overline{n} \text{ next}(e_1, e_2) \rightarrow \overline{n} \text{ next}(e_1, e'_2)$ by (NEXTN-S2).

Subcase $e_1 \rightarrow e'_1$: Then we have $\overline{n} \text{ next}(e_1, e_2) \rightarrow \overline{n} \text{ next}(e'_1, e_2)$ by (NEXTN-S1).

Task 8 Prove preservation for your extended semantics, i.e.

(Preservation) If $\vdash e : \tau$ and $e \rightarrow e'$, then $\vdash e' : \tau$.

Solution: We proceed by induction on the dynamics $e \rightarrow e'$.

---

1In the (LEQ) case, you will have an additional case split on whether (LEQ-I1) or (LEQ-I2) applies.
2This case is actually similar to (PLUS), but it is good practice to show it again to check that nothing was missed in Task 5.
3Technically, this needs to be shown by induction on the judgments defined in Tasks 3 and 4.
Case $e_1 + e_2 \mapsto e'_1 + e_2$ (Plus-S1): The premise of the rule is $e_1 \mapsto e'_1$. By inversion on the typing judgement, we have that $\vdash e_1 : \text{nat}$, i.e. $\tau = \text{nat}$. Therefore, by I.H., we have that $\vdash e'_1 : \text{nat}$, and therefore $\vdash e'_1 + e_2 : \text{nat}$ by (Plus).

Cases (Plus-S2), (Times-S1), (Times-S2), (Leq-S1), (Leq-S2), (If-S), (Nextn-S1) and (Nextn-S2): All of these are congruence cases similar to (Plus-S1).

Case $\overline{m} + \overline{n} \mapsto \overline{m+n}$ (Plus-I): By inversion, on the typing judgment, we have that $\tau = \text{nat}$. By (Num), $\vdash m+n : \text{nat}$.

Cases (Times-I), (Leq-I1), (Leq-I2), (Nextn-I): These are reduction cases similar to (Plus-I).

Case $\text{if}(\overline{e}, e_2, e_3) \mapsto e_2$ (If-I1): By inversion on the typing judgment, we have that $\vdash e_2 : \tau$ and we are done. The remaining case for (If-I2) is similar.

\[4^4\]I have collapsed all the congruence cases here, but you should be a bit more careful in your proofs. Again, it might be useful to check the cases for (Nextn-S1) and (Nextn-S2) explicitly to make sure they are correctly defined.