15-814 Homework 1 Solutions

September 25, 2017

1 Arithmetic

Task 1 *Prove the following* inversion lemma*:*

(If Inversion) If $\Gamma \vdash \mathtt{if}(e_1, e_2, e_3) : \tau$*, then* $\Gamma \vdash e_1 : \mathtt{bool}$*,* $\Gamma \vdash e_2 : \tau$*, 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 Γ*,* e*,* τ *such that* $\Gamma \vdash e : \tau$*, if* $e = \mathtt{if}(e_1, e_2, e_3)$ *for some* e_1*,* e_2*,* e_3*, then* $\Gamma \vdash e_1 : \mathtt{bool}$*,* $\Gamma \vdash e_2 : \tau$*, and* $\Gamma \vdash e_3 : \tau$*.)*

Solution: We show that for all Γ, e, τ such that $\Gamma \vdash e : \tau$, if $e = \mathtt{if}(e_1, e_2, e_3)$ for some e_1, e_2, e_3, then $\Gamma \vdash e_1 : \mathtt{bool}$, $\Gamma \vdash e_2 : \tau$, 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 $\Gamma \vdash \mathtt{if}(e_1, e_2, e_3) : \tau$, we need to show $\Gamma \vdash e_1 : \mathtt{bool}$, $\Gamma \vdash e_2 : \tau$, 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 $e = \mathtt{if}(e_1, e_2, e_3)$. Therefore, the required property is trivially true.

Task 2 *Prove unicity of typing for this language.*

(Unicity of Typing) *For any* Γ*,* e*,* τ*,* τ' *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 $e = \overline{n}$ (Num): By assumption, we have $\Gamma \vdash \overline{n} : \mathtt{nat}$ and $\Gamma \vdash \overline{n} : \tau'$, i.e. $\tau = \mathtt{nat}$. By inversion on the latter judgment, we have that $\tau' = \mathtt{nat} = \tau$.

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

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

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

Case $e = \mathtt{if}(e_1, e_2, e_3)$ (If): By assumption, we have $\Gamma \vdash \mathtt{if}(e_1, e_2, e_3) : \tau$, $\Gamma \vdash e_2 : \tau$, and $\Gamma \vdash \mathtt{if}(e_1, e_2, e_3) : \tau'$. 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* $\texttt{next}(d)\ \texttt{is}\ d'$ *judgement which takes a day* d *and returns the next day,* d'*. You should assume that the next day from Sunday is Friday.*

Solution:

$$\frac{}{\texttt{next(Fri) is Sat}}\ (\textsc{Fri}) \qquad \frac{}{\texttt{next(Sat) is Sun}}\ (\textsc{Sat}) \qquad \frac{}{\texttt{next(Sun) is Fri}}\ (\textsc{Sun})$$

Task 4 *Define the* $\texttt{nextn}(n, d)\ \texttt{is}\ d'$ *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* $\texttt{nat}$*.*

Solution:

$$\frac{}{\texttt{nextn}(\texttt{Z}, d)\ \texttt{is}\ d}\ (\textsc{NextZ}) \qquad \frac{\texttt{nextn}(n, d)\ \texttt{is}\ d' \quad \texttt{next}(d')\ \texttt{is}\ d''}{\texttt{nextn}(\texttt{S}(n), d)\ \texttt{is}\ d''}\ (\textsc{NextSucc})$$

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

Solution:

- Statics

$$\frac{}{\Gamma \vdash \overline{d} : \texttt{day}}\ (\textsc{Day}) \qquad \frac{\Gamma \vdash e_1 : \texttt{nat} \quad \Gamma \vdash e_2 : \texttt{day}}{\Gamma \vdash \overline{\texttt{nextn}}(e_1, e_2) : \texttt{day}}\ (\textsc{Nextn})$$

- Dynamics

$$\frac{}{\overline{d}\ \mathsf{val}}\ (\textsc{Day-V})$$

$$\frac{e_1 \mapsto e_1'}{\overline{\texttt{nextn}}(e_1, e_2) \mapsto \overline{\texttt{nextn}}(e_1', e_2)}\ (\textsc{Nextn-S1}) \qquad \frac{e_1\ \mathsf{val} \quad e_2 \mapsto e_2'}{\overline{\texttt{nextn}}(e_1, e_2) \mapsto \overline{\texttt{nextn}}(e_1, e_2')}\ (\textsc{Nextn-S2})$$

$$\frac{\texttt{nextn}(n, d)\ \texttt{is}\ d'}{\overline{\texttt{nextn}}(\overline{n}, \overline{d}) \mapsto \overline{d'}}\ (\textsc{Nextn-I})$$

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* $e\ \mathsf{val}$ *and* $\vdash e : \tau$*, then*

- *if* $\tau = \texttt{nat}$ *then* $e = \overline{n}$ *for some natural number* n*,*
- *if* $\tau = \texttt{bool}$ *then* $e = \overline{\texttt{tt}}$ *or* $e = \overline{\texttt{ff}}$*,*
- *if* $\tau = \texttt{day}$ *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 val *or there exists* e' *such that* $e \mapsto e'$*.*

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 $\vdash \overline{n} : \texttt{nat}$ (Num): We have $\overline{n}$ val by (Num-V) so we are done.

Cases (True), (False), (Day): Similar to (Num).

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

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

Subcase e_1 val, $e_2 \mapsto e_2'$: Then we have $e_1 + e_2 \mapsto e_1 + e_2'$ by (Plus-S2).

Subcase $e_1 \mapsto e_1'$: Then we have $e_1 + e_2 \mapsto e_1' + e_2$ by (Plus-S1).

Cases (Times) (Leq): Similar to (Plus)[1]

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

Case $\vdash \overline{\texttt{nextn}}(e_1, e_2) : \texttt{day}$ (Nextn) [2] The premises of the rule are: $\vdash e_1 : \texttt{nat}$ and $\vdash e_2 : \texttt{day}$. By I.H. on the first premise, we have that either e_1 val or $e_1 \mapsto e_1'$. In the first case, we may further apply the I.H. on the second premise to get that either e_2 val or $e_2 \mapsto e_2'$.

Subcase e_1 val, e_2 val: Since $\vdash e_1 : \texttt{nat}$ and $\vdash e_2 : \texttt{day}$, by canonical forms, we have that $e_1 = \overline{n}, e_2 = \overline{d}$ for some n, d. Moreover, we have that $\texttt{nextn}(n, d)$ $\texttt{is}$ d' for some d'[3]. Thus, $\overline{\texttt{nextn}}(\overline{n}, \overline{d}) \mapsto \overline{d'}$ by (Nextn-I).

Subcase e_1 val, $e_2 \mapsto e_2'$: Then we have $\overline{\texttt{nextn}}(e_1, e_2) \mapsto \overline{\texttt{nextn}}(e_1, e_2')$ by (Nextn-S2).

Subcase $e_1 \mapsto e_1'$: Then we have $\overline{\texttt{nextn}}(e_1, e_2) \mapsto \overline{\texttt{nextn}}(e_1', e_2)$ by (Nextn-S1).

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

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

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

[1] In the (Leq) case, you will have an additional case split on whether (Leq-I1) or (Leq-I2) applies.

[2] This case is actually similar to (Plus), but it is good practice to show it again to check that nothing was missed in Task 5.

[3] Technically, 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 $\vdash e_1 : \texttt{nat}$, i.e. $\tau = \texttt{nat}$. Therefore, by I.H., we have that $\vdash e_1' : \texttt{nat}$, and therefore $\vdash e_1' + e_2 : \texttt{nat}$ 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)[4].

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

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

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

[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.