29

Midterm Exam

CS231: Types and Programming Languages

Wednesday, February 5, 2020

Rules of the game:

- **Write your name above.**
- The exam is closed-book and closed-notes.
- Please write your answers directly on the exam. Do not turn in anything else. Do not use any other paper during the exam.
- The exam ends promptly at 3:50pm.

A bit of advice:

- Read questions carefully. Understand a question before you start writing.
- Write down thoughts and intermediate steps so you can get partial credit.
- Relax!

1. (3 points each) Consider the language of booleans and integers on the cheat sheet.

   (a) Provide terms t and t′ along with a derivation tree demonstrating that $\texttt{t} \longrightarrow \texttt{t}'$, where the derivation tree has height exactly 3 (i.e., the longest path in the derivation tree from the root to a leaf has exactly 3 solid lines in it, each representing a usage of one of the inference rules), or say None if no such terms exist.

$t = ((1+1)+1)+1$

$t' = (2+1)+1$

$$\dfrac{\dfrac{\dfrac{2 = 1 [\![+]\!] 1}{1+1 \rightarrow 2}\;(\text{E-PlusRed})}{(1+1)+1 \rightarrow 2+1}\;(\text{E-Plus1})}{((1+1)+1)+1 \rightarrow (2+1)+1}\;(\text{E-Plus1})$$

   (b) Provide a term t and type T along with a derivation tree demonstrating that $\texttt{t} : \texttt{T}$, where the derivation tree has height exactly 3, or say None if no such term and type exist.

$t = (1+1)+1$

$T = \text{Int}$

−1

$$\dfrac{\dfrac{\dfrac{}{1\ \text{Int}}\;(\text{T-Num}) \quad \dfrac{}{1\ \text{Int}}\;(\text{T-Num})}{1+1 : \text{Int}}\;(\text{T-Plus}) \quad \dfrac{}{1\ \text{Int}}\;\text{T-Num}}{(1+1)+1\ \ \text{Int}}\;(\text{T-Plus})$$

(c) Provide three terms $\mathtt{t_1}$, $\mathtt{t_2}$, and $\mathtt{t_3}$ such that $\mathtt{t_1} \longrightarrow \mathtt{t_2}$ and $\mathtt{t_2} \longrightarrow \mathtt{t_3}$ and $\mathtt{t_3}$ is stuck. Say None if no such terms exist.

t_1 = ((1+1)+1) + true

t_2 = (2+1) + true

t_3 = 3 + true

✓

(d) Provide three terms $\mathtt{t_1}$, $\mathtt{t_2}$, and $\mathtt{t_3}$ such that $\mathtt{t_1} \longrightarrow \mathtt{t_2}$ and $\mathtt{t_2} \longrightarrow \mathtt{t_3}$ and $\mathtt{t_3}$ is value and there is no type T such that $\mathtt{t_1}$: T. Say None if no such terms exist.

t_1 = (if true then 1 else true) + 1

t_2 = 1+1

t_3 = 2

✓

2. Suppose we augment the grammar for metavariable `t` from the cheat sheet with a new expression form `t;t`, which provides expression *sequencing*. The semantics of an expression $t_1 ; t_2$ is to evaluate t_1, throw its value away, and return the value of t_2 as the value of the whole expression. (Sequencing is only useful in a language that has side effects, but we'll ignore that.)

   (a) (5 points) Add rules to the small-step operational semantics to evaluate the new expression form. Give a name to each new rule.

$$\frac{t_1 \to t_1'}{t_1 ; t_2 \to t_1' ; t_2} \quad (\text{E-SEQ1})$$

$$\frac{}{v_1 ; t_2 \to t_2} \quad (\text{E-SEQ2})$$

   (b) (3 points) Add rules to the type system to support the new expression form. Give a name to each new rule. Note that since the value of t_1 in the expression $t_1 ; t_2$ is thrown away, it can be any kind of value.

$$\frac{t_1 \;\; T_1 \qquad t_2 : T}{t_1 , t_2 : T} \quad (\text{T-SEQ})$$

3. (5 points each) In this question, you will prove (portions of) type soundness for your new expression from the previous question.

   (a) Provide only the cases specific to the new sequencing expression for the proof of the Progress theorem:

   Theorem (Progress): If `t : T`, then either `t` is a value or there exists some term `t'` such that `t ⟶ t'`.

   Assume that the proof is performed by induction on the derivation of `t : T`.

   Clearly state your induction hypothesis.

Induction hypothesis: If $t_0 : T_0$ and $(t_0 : T_0) \leq (t\ T)$, then either t_0 is a value or there exists some term t_0' such that $t_0 \to t_0'$

Case analysis on the root rule in the derivation $t\ T$.

- Case T-SEQ: We know $t = t_1 ; t_2$ and $t_1 : T_1$.

  By the induction hypothesis, either t_1 is a value.

  → By E-SEQ2, there exists $t' = t_2$ such that $t \to t'$.

  Or there exists some term t_1' such that $t_1 \to t_1'$:

  → By E-SEQ1, there exists $t' = t_1' ; t_2$ such that $t \to t'$

(b) Provide only the cases specific to the new sequencing expression for the proof of the Preservation theorem:

Theorem (Preservation): If t : T and t $\longrightarrow$ t′, then t′ : T.

Assume that the proof is performed by induction on the derivation of t : T.

Clearly state your induction hypothesis.

Induction hypothesis: If t_0 T_0 and $t_0 \to t_0'$ where $(t_0 : T_0) \le (t : T)$, then $t_0' : T_0$.

Case analysis on the root rule in the derivation $t : T$

- Case T-SEQ. We know $t = t_1 ; t_2$ and $t_1 : T_1$ and $t_2 : T$

  Case analysis on the root rule in the derivation $t \to t'$:

  - Case E-SEQ1: We know $t_1 \to t_1'$ and $t' = t_1' ; t_2$.

    By the induction hypothesis, $t_1' : T_1$ Then by T-SEQ, $t' : T$.

  - Case E-SEQ2. We know t_1 is a value and $t' = t_2$

    We already know $t_2 : T$, so $t' : T$