CSCI 5535: Homework Assignment 1: Statics and Dynamics for a Simple Language

Fall 2023: Due Friday, September 22, 2023

The purpose of this assignment is to formalize language constructs using a big-step operational semantics.

Recall the evaluation guideline from the course syllabus.

> *Both your ideas and also the clarity with which they are expressed matter—both in your English prose and your code!*
>
> *We will consider the following criteria in our grading:*
>
> - How well does your submission answer the questions? *For example, a common mistake is to give an example when a question asks for an explanation. An example may be useful in your explanation, but it should not take the place of the explanation.*
> - How clear is your submission? *If we cannot understand what you are trying to say, then we cannot give you points for it. Try reading your answer aloud to yourself or a friend; this technique is often a great way to identify holes in your reasoning. For code, not every program that "works" deserves full credit. We must be able to read and understand your intent. Make sure you state any preconditions or invariants for your functions.*

Submission Instructions. Typesetting is preferred but scanned, clearly legible handwritten write-ups are acceptable. Please no other formats—no `.doc` or `.docx`. You may use whatever tool you wish (e.g., LaTeX, Word, markdown, plain text, pencil+paper) as long as it is legibly converted into a `pdf`.

1. **Feedback**. Complete the survey on the linked from the moodle after completing this assignment. Any non-empty answer will receive full credit.

2. **Induction Practice on Cards**. In this question, we will do some induction practice on "cards" from the previous assignment. Recall our model of a deck of cards:

$$\frac{}{\heartsuit \text{ card}} \qquad \frac{}{\spadesuit \text{ card}} \qquad \frac{}{\clubsuit \text{ card}} \qquad \frac{}{\diamondsuit \text{ card}} \qquad \frac{}{\mathsf{nil}\ \text{deck}} \qquad \frac{c\ \text{card} \qquad s\ \text{deck}}{\mathsf{cons}(c, s)\ \text{deck}}$$

These rules are an iterated inductive definition for s deck and lead to the following induction principle:

> *In order to show $P(s)$ for some property P, for any s* deck*, it is enough to show*
>
> 1. $P(\mathsf{nil})$.
> 2. $P(\mathsf{cons}(c, s'))$*, assuming c* card *and $P(s')$*. This $P(s')$ is called the induction hypothesis.

Also recall that definition of the $\mathsf{unshuffle}(s_1, s_2, s_3)$ judgment.

$$\frac{}{\mathsf{unshuffle}(\mathsf{nil}, \mathsf{nil}, \mathsf{nil})} \qquad \frac{c\ \text{card} \qquad \mathsf{unshuffle}(s_1, s_2, s_3)}{\mathsf{unshuffle}(\mathsf{cons}(c, s_1), s_2, \mathsf{cons}(c, s_3))}$$

$$\frac{c\ \text{card} \qquad \mathsf{unshuffle}(s_1, s_2, s_3)}{\mathsf{unshuffle}(\mathsf{cons}(c, s_1), \mathsf{cons}(c, s_2), s_3)}$$

(a) Prove the following about the $\mathsf{unshuffle}(s_1, s_2, s_3)$ judgment:

> *If s_1* deck*, then* $\mathsf{unshuffle}(s_1, s_2, s_3)$ *for some decks s_2 and s_3.*

Note that there are a number of different ways of proving this! For this exercise, be explicit about the induction principle you use and identify what property P you are proving with that induction principle.

(b) For this next part, we will define, using simultaneous inductive definition, decks of cards with even or odd numbers of cards in them.

$$\frac{}{\mathsf{nil}\ \text{even}} \qquad \frac{c\ \text{card} \qquad s\ \text{odd}}{\mathsf{cons}(c, s)\ \text{even}} \qquad \frac{c\ \text{card} \qquad s\ \text{even}}{\mathsf{cons}(c, s)\ \text{odd}}$$

This inductive definition is *simultaneous* (because it simultaneously defines s even and s odd), as well as *iterated* (because it relies on the previously-defined definition of c card).

i. What is the induction principle for these judgments—that is, for the judgments s even and s odd?

ii. Prove the following theorem that we can see as a *derived induction principle*:

> *For all properties S, if*
>
> 1. $S(\mathsf{nil})$*; and*
> 2. *for all c_1, c_2, and s, if c_1* card*, c_2* card*, and $S(s)$, then $S(\mathsf{cons}(c_1, \mathsf{cons}(c_2, s)))$;*
>
> *then for all s, if s* even *then $S(s)$.*

You will want to use the induction principle mentioned above in order to prove this. And as always, remember to carefully consider and state the induction hypothesis you are using.

iii. Another "operation" on cards is *cutting*, where a player separates a single deck of cards into two decks of cards by removing some number of cards from the top of the deck. We can define cutting cards using an inductive definition:

$$\frac{s\ \text{deck}}{\mathsf{cut}(s, s, \mathsf{nil})} \qquad \frac{c\ \text{card} \qquad \mathsf{cut}(s_1, s_2, s_3)}{\mathsf{cut}(\mathsf{cons}(c, s_1), s_2, \mathsf{cons}(c, s_3))}$$

Using the derived induction principle from the previous task (you can use the induction principle from the previous task even if you do not do the previous task!), prove the following:

For all s_1, s_2, s_3, *if* s_2 even, s_3 even, *and* $\mathsf{cut}(s_1, s_2, s_3)$, *then* s_1 even.

You are allowed to assume the following lemmas:

- *Inversion for* nil*:* For all s_1 and s_2, if $\mathsf{cut}(s_1, s_2, \mathsf{nil})$, then $s_1 = s_2$ and s_1 deck.
- *Inversion for* cons*:* For all s_1, s_2, and s_3, if $\mathsf{cut}(s_1, s_2, \mathsf{cons}(c, s_3))$, then there exists a s_1' such that $s_1 = \mathsf{cons}(c, s_1')$, c card, and $\mathsf{cut}(s_1', s_2, s_3)$.

3. **Language Extension and Run-Time Errors.**

   (a) Consider the IMP language from FSPL with the **Aexp** sub-language extended with an integer division expression: $e ::= \cdots \mid e_1 \,/\, e_2$ Explain what changes must be made to the big-step operational semantics (i.e., the evaluation judgment form $\langle e, \sigma \rangle \Downarrow n$). Write out formally any new rules of inference you introduce.

   (b) Consider the **E** language from PFPL (in Section 6.3) similarly extended with an integer division expression: $e ::= \cdots \mid \mathtt{div}(e_1; e_2)$ Complete the formalization of the extension of **E** with $\mathtt{div}(e_1; e_2)$. That is, extend the judgment forms e val, $e \longmapsto e'$, e err, $e \Downarrow e'$, and $\Gamma \vdash e : \tau$ from the appendix with rules for expression $\mathtt{div}(e_1; e_2)$ as appropriate.

4. **Type Safety.** Again, consider the **E** language from PFPL.

   The tasks here ask you to prove (meta-theoretical) properties about the language **E**. They are "meta theoretical" in that they give a theory about the theory of **E**and are true about a large set of **E** programs, not specific, individual programs. Language **E** is, by design, tiny to focus on the essence of meta-theory of programming languages.

   There is a lot of error checking going on in the dynamics in Appendix A. But as we will been discussing in class, we can eliminate much or all of this by equipping our language with a static type system (see Chapter 6 of PFPL)!

   The type checking rules for **E** are reproduced in Appendix B for your reference.

   Grading criteria: To receive full credit for any proof, you must *at least* do the following:

   - At the beginning of your proof, specify over what structure or derivation you are performing induction (i.e., which structure's inductive principle are you using?)
   - In the inductive cases of the proof, specify how you are applying the inductive hypothesis, and what result it gives you.

   If you omit these steps and/or do not make them explicit, you will receive zero credit for your proof. If you attempt to do these steps, but you make a mistake, you may still receive some partial credit, depending on your proof.

   Note on omitting redundant proof cases: In the proofs below, some cases are very similar to other cases, e.g., the cases for `plus` and `times` in the proofs below are likely to be analogous, in that (nearly) the same proof steps are used in each. When this happens, you can omit the redundant cases as follows: If you do one case, say for `plus`, you may (optionally) write in the other case for `times` that it is "analogous to the case above, for `plus`". You must

make this omission explicit, to show that you have thought about it. Further, this shortcut is only applicable when the cases really are analogous, and (nearly) the same steps apply in the proof. *When in doubt, do not omit the proof case.*

(a) **Canonical Forms.** Prove that if e val, then

1. if $\Gamma \vdash e : \texttt{num}$ then $e = \texttt{num}[n]$ for some number n.
2. if $\Gamma \vdash e : \texttt{str}$ then $e = \texttt{str}[s]$ for some string s.

(b) **Substitution.** State and prove the *substitution lemma* for E.

(c) **Progress.** State and prove the *progress theorem* for E. To receive full credit, you must additionally use the canonical forms lemma correctly, when appropriate.

(d) **Preservation.** State and prove the *preservation theorem* for E. To receive full credit, you must additionally use the substitution lemma correctly, when appropriate.

A Dynamics of E

$\boxed{e \text{ val}}$

$$\frac{}{\texttt{num}[n] \text{ val}} \qquad \frac{}{\texttt{str}[s] \text{ val}}$$

$\boxed{e \longmapsto e'}$

$$\frac{}{\texttt{plus}(\texttt{num}[n_1];\texttt{num}[n_2]) \longmapsto \texttt{num}[n_1 + n_2]} \qquad \frac{e_1 \longmapsto e_1'}{\texttt{plus}(e_1;e_2) \longmapsto \texttt{plus}(e_1';e_2)}$$

$$\frac{e_2 \longmapsto e_2'}{\texttt{plus}(\texttt{num}[n_1];e_2) \longmapsto \texttt{plus}(\texttt{num}[n_1];e_2')} \qquad \frac{}{\texttt{times}(\texttt{num}[n_1];\texttt{num}[n_2]) \longmapsto \texttt{num}[n_1 \cdot n_2]}$$

$$\frac{e_1 \longmapsto e_1'}{\texttt{times}(e_1;e_2) \longmapsto \texttt{times}(e_1';e_2)} \qquad \frac{e_2 \longmapsto e_2'}{\texttt{times}(\texttt{num}[n_1];e_2) \longmapsto \texttt{times}(\texttt{num}[n_1];e_2')}$$

$$\frac{}{\texttt{cat}(\texttt{str}[s_1];\texttt{str}[s_2]) \longmapsto \texttt{str}[s_1 \hat{} s_2]} \qquad \frac{e_1 \longmapsto e_1'}{\texttt{cat}(e_1;e_2) \longmapsto \texttt{cat}(e_1';e_2)}$$

$$\frac{e_2 \longmapsto e_2'}{\texttt{cat}(\texttt{str}[s_1];e_2) \longmapsto \texttt{cat}(\texttt{str}[s_1];e_2')} \qquad \frac{}{\texttt{len}(\texttt{str}[s]) \longmapsto \texttt{num}[|s|]} \qquad \frac{e \longmapsto e'}{\texttt{len}(e) \longmapsto \texttt{len}(e')}$$

$$\frac{e_1 \text{ val}}{\texttt{let}(e_1;x.e_2) \longmapsto [e_1/x]e_2} \qquad \frac{e_1 \longmapsto e_1'}{\texttt{let}(e_1;x.e_2) \longmapsto \texttt{let}(e_1';x.e_2)}$$

$\boxed{e \text{ err}}$

$$\frac{}{\texttt{plus}(\texttt{str}[s];e_2) \text{ err}} \qquad \frac{}{\texttt{plus}(\texttt{num}[n];\texttt{str}[s]) \text{ err}} \qquad \frac{e_1 \text{ err}}{\texttt{plus}(e_1;e_2) \text{ err}} \qquad \frac{e_2 \text{ err}}{\texttt{plus}(\texttt{num}[n];e_2) \text{ err}}$$

$$\frac{}{\texttt{times}(\texttt{str}[s];e_2) \text{ err}} \qquad \frac{}{\texttt{times}(\texttt{num}[n];\texttt{str}[s]) \text{ err}} \qquad \frac{e_1 \text{ err}}{\texttt{times}(e_1;e_2) \text{ err}}$$

$$\frac{e_2 \text{ err}}{\texttt{times}(\texttt{num}[n];e_2) \text{ err}} \qquad \frac{}{\texttt{cat}(\texttt{num}[n];e_2) \text{ err}} \qquad \frac{}{\texttt{cat}(\texttt{str}[s];\texttt{num}[n]) \text{ err}} \qquad \frac{e_1 \text{ err}}{\texttt{cat}(e_1;e_2) \text{ err}}$$

$$\frac{e_2 \text{ err}}{\texttt{cat}(\texttt{str}[s];e_2) \text{ err}} \qquad \frac{}{\texttt{len}(\texttt{num}[n]) \text{ err}} \qquad \frac{e \text{ err}}{\texttt{len}(e) \text{ err}} \qquad \frac{e_1 \text{ err}}{\texttt{let}(e_1;x.e_2) \text{ err}}$$

$\boxed{e \Downarrow e'}$

$$\frac{}{\mathtt{num}[n] \Downarrow \mathtt{num}[n]} \qquad \frac{}{\mathtt{str}[s] \Downarrow \mathtt{str}[s]} \qquad \frac{e_1 \Downarrow \mathtt{num}[n_1] \qquad e_2 \Downarrow \mathtt{num}[n_2]}{\mathtt{plus}(e_1; e_2) \Downarrow \mathtt{num}[n_1 + n_2]}$$

$$\frac{e_1 \Downarrow \mathtt{num}[n_1] \qquad e_2 \Downarrow \mathtt{num}[n_2]}{\mathtt{times}(e_1; e_2) \Downarrow \mathtt{num}[n_1 \cdot n_2]} \qquad \frac{e_1 \Downarrow \mathtt{str}[s_1] \qquad e_2 \Downarrow \mathtt{str}[s_2]}{\mathtt{cat}(e_1; e_2) \Downarrow \mathtt{str}[s_1 s_2]} \qquad \frac{e \Downarrow \mathtt{str}[s] \qquad |s| = n}{\mathtt{len}(e) \Downarrow \mathtt{num}[n]}$$

$$\frac{e_1 \Downarrow e_1' \qquad [e_1'/x]e_2 \Downarrow e_2'}{\mathtt{let}(e_1; x.e_2) \Downarrow e_2'}$$

B Statics of E

$\boxed{\Gamma \vdash e : \tau}$

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau} \qquad \frac{}{\Gamma \vdash \mathtt{num}[n] : \mathtt{num}} \qquad \frac{}{\Gamma \vdash \mathtt{str}[s] : \mathtt{str}} \qquad \frac{\Gamma \vdash e_1 : \mathtt{num} \qquad \Gamma \vdash e_2 : \mathtt{num}}{\Gamma \vdash \mathtt{plus}(e_1; e_2) : \mathtt{num}}$$

$$\frac{\Gamma \vdash e_1 : \mathtt{num} \qquad \Gamma \vdash e_2 : \mathtt{num}}{\Gamma \vdash \mathtt{times}(e_1; e_2) : \mathtt{num}} \qquad \frac{\Gamma \vdash e_1 : \mathtt{str} \qquad \Gamma \vdash e_2 : \mathtt{str}}{\Gamma \vdash \mathtt{cat}(e_1; e_2) : \mathtt{str}} \qquad \frac{\Gamma \vdash e : \mathtt{str}}{\Gamma \vdash \mathtt{len}(e) : \mathtt{num}}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma, x : \tau_1 \vdash e_2 : \tau_2}{\Gamma \vdash \mathtt{let}(e_1; x.e_2) : \tau_2}$$