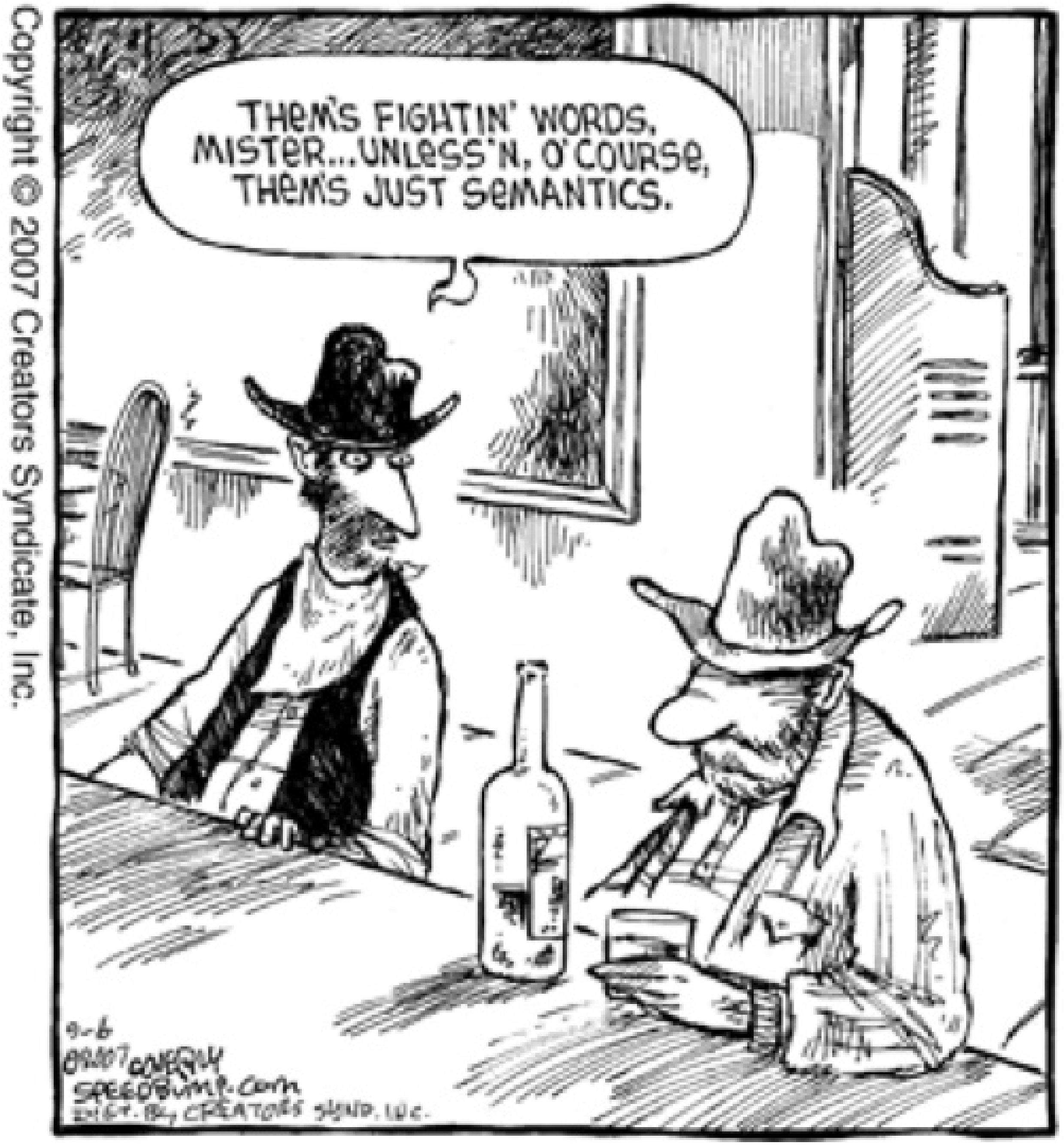
Meeting 5: Statics and dynamics



Announcements

- Schedule updated:
  - More explicit that HW0 = Part I Project and HW1 = Part II Project.
- Homework 1 due next week: Friday at 6:00pm

Submission

- Submit (push) to GitHub
- *And* upload zip to Moodle

Questions

- Eirian
  - Work through Lemma 4.4 (Substitution)
  - Notation: dom; e1/e2/z
  - Concrete versus abstract syntax
  - Lemma 4.1: Unicity of typing
  - ~~Example of not satisfying~~
  - Chapter 3
  - Notation from 3.4
  - "Structural properties" -- big picture

- Chapter 3 - General vs. Parametric Judgments
- Proving inversion lemmas

parser

abstract syntax

"trees"

Syntax

Ny

What ("strings" or "terms") = programs

are in the language

concrete syntax — What the human inputs

Semantics

What do programs in the language

mean ~ computation/execution/evaluation

$$\frac{\Gamma \vdash e_1 : num \qquad \Gamma \vdash e_2 : num}{\Gamma \vdash plus(e_1, e_2) : num}$$

$$\Gamma \vdash e : \tau$$

$$typecheck(\underset{: TyEnv}{\Gamma}, \underset{: Exp}{e}) : Type$$

$$\Gamma \vdash plus(\text{"foo"}, \text{"bar"}) : num \qquad = \tau$$

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}$$

$$\frac{}{\Gamma, x : \tau \vdash x : \tau}$$

$$\frac{}{\Gamma \vdash x : \Gamma(x)}$$

Assignment #1:
Dynamics and Statics for a Simple Language

CSCI 5535 / ECEN 5533: Fundamentals of Programming Languages

Spring 2018: due Friday, February 9, 2018

The tasks in this homework ask you to prove ("meta theoretical") properties about the language **E**, defined in Part II of *PFPL*. 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.

Tasks

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 below, 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.**

Task 1 (5 pts). (Canonical Forms) Prove that if e val, then

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

x : num

e expression is a value – "something done computing"

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

$\boxed{e : \tau}$ type $\boxed{\tau \text{ type}}$

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

expression e "should" evaluate to a value that is "described" by type τ

Task 2 (15 pts). (Substitution) State and prove the *substitution lemma* for **E**.

Task 3 (20 pts). (Progress) State and prove the *progress theorem* for **E**. To receive full credit, you must additionally use the canonical forms lemma correctly, when appropriate.

Task 4 (20 pts). (Preservation) State and prove the *preservation theorem* for **E**. To receive full credit, you must additionally use the substitution lemma correctly, when appropriate.

Task 5 (15 pts). (Language Extension and Run-Time Errors) Complete the formalization of the extension of **E** with $\texttt{div}(e_1;e_2)$ from Section 6.3 of PFPL. That is, extend the judgment forms $e\ \mathsf{val}$, $e \longmapsto e'$, $e\ \mathsf{err}$, $e \Downarrow e'$, and $\Gamma \vdash e : \tau$ from the appendix with rules for expression $\texttt{div}(e_1;e_2)$ as appropriate.

Task 6 (25 pts). (Relating Structural and Evaluation Dynamics) Prove that your structural dynamics for **E** with $\texttt{div}(e_1;e_2)$ coincide with your evaluation dynamics. Specifically, prove if $e \longmapsto^* e'$ and $e'\ \mathsf{val}$, then $e \Downarrow e'$. Hint: you will need a nesting of two inductions, that is, you will need to state and prove a separate lemma that you use in the proof of this main theorem.

A Dynamics of E

$\mathtt{plus}(\mathtt{num}[n_1], \mathtt{num}[n_2])$ val

Evan's bizarre E

$\boxed{e \ \mathsf{val}}$

$$\frac{}{\mathtt{num}[n]\ \mathsf{val}} \qquad \frac{}{\mathtt{str}[s]\ \mathsf{val}}$$

$\boxed{e \longmapsto e'}$

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

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

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

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

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

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

$\boxed{e \ \mathsf{err}}$

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

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

$$\frac{}{\mathtt{cat}(\mathtt{num}[n];e_2)\ \mathsf{err}} \qquad \frac{}{\mathtt{cat}(\mathtt{str}[s];\mathtt{num}[n])\ \mathsf{err}} \qquad \frac{e_1\ \mathsf{err}}{\mathtt{cat}(e_1;e_2)\ \mathsf{err}} \qquad \frac{e_2\ \mathsf{err}}{\mathtt{cat}(\mathtt{str}[s];e_2)\ \mathsf{err}}$$

$$\frac{}{\mathtt{len}(\mathtt{num}[n])\ \mathsf{err}} \qquad \frac{e\ \mathsf{err}}{\mathtt{len}(e)\ \mathsf{err}} \qquad \frac{e_1\ \mathsf{err}}{\mathtt{let}(e_1;x.e_2)\ \mathsf{err}}$$

$\boxed{e \Downarrow e'}$

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

$$\frac{e_1 \Downarrow \texttt{str}[s_1] \qquad e_2 \Downarrow \texttt{str}[s_2]}{\texttt{cat}(e_1;e_2) \Downarrow \texttt{str}[s_1 s_2]} \qquad \frac{e \Downarrow \texttt{str}[s] \qquad |s| = n}{\texttt{len}(e) \Downarrow \texttt{num}[n]} \qquad \frac{e_1 \Downarrow e_1' \qquad [e_1'/x]e_2 \Downarrow e_2'}{\texttt{let}(e_1;x.e_2) \Downarrow e_2'}$$

B Statics of E

expression

$e : \tau$ type

Γ $x : \tau, \ldots$

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

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

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

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

$e ::= \ldots$ $\boxed{e \text{ val}}$ $x \in \{ a123\_,\ baz, \ldots \}$

$v ::= \ldots$ (values) (let a123 be 10 in a123) $=_\alpha$ (let baz be 10 in baz)

trated by example. The following chart summarizes the abstract and concrete syntax of **E**.

Typ	τ	::=	`num`	`num`	numbers
			`str`	`str`	strings
Exp	e	::=	x	x	variable
			$\texttt{num}[n]$	n	numeral
			$\texttt{str}[s]$	$''s''$	literal
			$\texttt{plus}(e_1;e_2)$	$e_1 + e_2$	addition
			$\texttt{times}(e_1;e_2)$	$e_1 * e_2$	multiplication
			$\texttt{cat}(e_1;e_2)$	$e_1 \hat{}\ e_2$	concatenation
			$\texttt{len}(e)$	$\lvert e\rvert$	length
			$\texttt{let}(e_1;x.e_2)$	$\texttt{let}\ x\ \texttt{be}\ e_1\ \texttt{in}\ e_2$	definition

$\tau \in Typ = \{num, str\}$

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

$\Gamma ::= \cdot \mid \Gamma, x : \tau$

This chart defines two sorts, Typ, ranged over by τ, and Exp, ranged over by e. The chart defines a set of operators and their arities. For example, it specifies that the operator `let` has arity $(\text{Exp}, \text{Exp}.\text{Exp})\text{Exp}$, which specifies that it has two arguments of sort Exp, and binds a variable of sort Exp in the second argument.

4.2 Type System

The role of a type system is to impose constraints on the formations of phrases that are sensitive to the context in which they occur. For example, whether the expression $\texttt{plus}(x;\texttt{num}[n])$ is sensible depends on whether the variable x is restricted to have type `num` in the surrounding context of the expression. This example is, in fact, illustrative of the general case, in that the *only* information required about the context of an expression is the type of the variables within whose scope the expression lies. Consequently, the statics of **E** consists of an inductive definition of generic hypothetical judgments of the form

$$\mathcal{X} \mid \Gamma \vdash e : \tau,$$

where $\mathcal{X}$ is a finite set of variables, and Γ is a *typing context* consisting of hypotheses of the form $x : \tau$, one for each $x \in \mathcal{X}$. We rely on typographical conventions to determine the set of variables, using the letters x and y to stand for them. We write $x \notin dom(\Gamma)$ to say that there is no assumption in Γ of the form $x : \tau$ for any type τ, in which case we say that the variable x is *fresh* for Γ.

The rules defining the statics of **E** are as follows:

$$\frac{}{\Gamma, x : \tau \vdash x : \tau} \tag{4.1a}$$

$$\frac{}{\Gamma \vdash \texttt{str}[s] : \texttt{str}} \tag{4.1b}$$

$$\frac{}{\Gamma \vdash \texttt{num}[n] : \texttt{num}} \tag{4.1c}$$

$$\frac{\Gamma \vdash e_1 : \texttt{num} \quad \Gamma \vdash e_2 : \texttt{num}}{\Gamma \vdash \texttt{plus}(e_1;e_2) : \texttt{num}} \tag{4.1d}$$

$$\frac{\Gamma \vdash e_1 : \texttt{num} \quad \Gamma \vdash e_2 : \texttt{num}}{\Gamma \vdash \texttt{times}(e_1;e_2) : \texttt{num}} \tag{4.1e}$$