Plan

1. Review of three styles of operational semantics
2. Briefing on Coq, installing & running it, etc.
3. Interactive examples working up to operational semantics
\[ \begin{align*} &\lambda x.\ e \Downarrow \lambda x.\ e \\ \hline &e_1 \Downarrow \lambda x.\ e' \quad e_2 \Downarrow \nu \quad e'_1[\nu/x] \Downarrow \nu' \\ \hline &e_1 \quad e_2 \Downarrow \nu' \end{align*} \]
λ-Calculus Small-Step Semantics

\[
(\lambda x. \ e) \ v \rightarrow e[v/x]
\]

\[
e_1 \rightarrow e'_1
\]

\[
e_1 \ e_2 \rightarrow e'_1 \ e_2
\]

\[
e_2 \rightarrow e'_2
\]

\[
v \ e_2 \rightarrow v \ e'_2
\]
$\lambda$-Calculus Contextual Semantics

\[
C ::= \square \mid C \ e \mid \nu \ C
\]

\[
C[(\lambda x. \ e) \ \nu] \rightarrow C[e[\nu/x]]
\]
What is Coq?

1. Computer software for checking mathematical proofs
2. Also contains a functional programming language much like Haskell/OCaml.
3. A research project from INRIA (a French national lab) with a large international user community in CS and math
Installing Coq

One or more of:

1. Download source or binary packages for a variety of platforms: http://coq.inria.fr/download
2. Install Debian/Ubuntu package: apt-get install coq
3. Run it on MIT Athena machines (should be available by Wednesday or so)

I’ll be using the latest version, 8.4pl2, but a variety of old versions should be usable for everything in this class, too. Let me know if something funny seems to happen when using any version $\geq 8.3$. 
IDEs

Writing formal proofs in an interpreter is no fun at all, when you realize you made a mistake and need to back up and change something. So, I recommend using a graphical interface:

1. CoqIDE (Debian/Ubuntu package coqide) is a specialized IDE that I’ve never used much.
2. Proof General (Debian/Ubuntu package proofgeneral-coq or just proofgeneral in more recent releases) is an Emacs mode.

I’ve only ever used Proof General seriously, and that’s what I’ll use for demos in class, but you’re free to use whatever approach you like.
1. The reference manual isn’t bad:
   http://coq.inria.fr/distrib/current/refman/

2. My draft book *Certified Programming with Dependent Types* is available online (http://adam.chlipala.net/cpdt/), but most of what it covers goes beyond the scope of 6.820.


4. A popular online book that uses Coq to introduce ideas in semantics: *Software Foundations* by Pierce et al. (http://www.cis.upenn.edu/~bcpierce/sf/)
All You Need to Know about Proof General

Colored highlighting indicates which prefix of the current file has been processed already. To run to some point in a file, move the cursor there and press “control-C enter.” You can process new commands with those keystrokes after moving the cursor after the new commands, and you can undo with the same keystrokes after moving before the commands to undo.
...starting from L04-CoqIntro-Template.v.

We will prove theorems by running commands called tactics in one of Coq’s programming languages. Tactics are the building blocks of interactive proof.