1 Reading

Please read (the rest of) Chapter 3 in your textbook.

2 Problems

Please do problem 3.5.13 (Funny rules). Explain your answers, but no proof (natural language or SASyLF) is required.

Please also answer the following question:

Which of the theorems 3.5.4, 3.5.7, 3.5.8, 3.5.11, 3.5.12 remain true after adding the rules for arithmetic expressions (Figure 3–2) to the original system (Figure 3–1)? Explain!

(Ex. 3.5.14 and its solution should help, as should definition 3.5.15.)

3 Proofs

Do problem 3.5.17 and write the proof in SASyLF (only for the “if” sublanguage!). More precisely, prove that if \( t \xrightarrow{*} t' \) and \( t' \Downarrow v \), then \( t \Downarrow v \). (I am not requiring you to prove that \( t' \) is a value.) You may use the solution in the back of the book (p. 498 in my edition) to help you write the proof.

When I solved this problem, I noticed I needed the following lemmas:

1. If \( t \rightarrow t' \) and \( t' \Downarrow v \), then \( t \Downarrow v \).
2. If \( t \xrightarrow{*} t' \) and \( t' \Downarrow v \), then \( t \Downarrow v \).
3. If \( t_1 \xrightarrow{*} t'_1 \), then for any \( t_2 \) and \( t_3 \), if \( t_1 \) then \( t_2 \) else \( t_3 \xrightarrow{*} \) if \( t'_1 \) then \( t_2 \) else \( t_3 \). (Make sure that \( t_2 \) and \( t_3 \) are inputs, not outputs of the lemma.)

Adding \texttt{succ}, \texttt{pred}, \texttt{iszero} doubles the size of the proof—be thankful you don’t need to handle them!

4 Graduate Students

Find three papers from journals or academic conferences in programming languages (TOPLAS, POPL, OOPSLA, ECOOP) from the last 5 years which define an operational semantics (evaluation) for a programming language. For each one, determine whether the evaluation relation is

\texttt{small-step} as with the book (with errors getting the program “stuck”);

\texttt{small-step with errors} as in Exercise 3.5.16

\texttt{big-step} as in Exercise 3.5.17

Cite each paper and explain your categorization.
5 Submission

Please turn in your paper homework problems on paper at the beginning of lecture. The SASSyLF proof should be done by updating (and committing and pushing) your `homework3` repository.