1 Reading

Please read Chapter 18 and 19 in the textbook.

2 Programming (5 points)

Extend the counter class with a new “class” of variable counter class, with a new method setinc that sets the step size which is used by a new definition for inc that increments by the current step size. The code should use ascription so that the record types are given useful names. Use the fullref checker to check your code. Here is a simple test case:

```java
vc1 = newVarCounter(2);
vc2 = newVarCounter(3);
inc3 vc1;
inc3 vc2;
vc1.get unit;
vc2.get unit;
vc1.setinc(1);
inc3 vc1;
inc3 vc2;
vc1.get unit;
vc2.get unit;
```

The last two numbers printed should be 10 and 19 (if you start your counters at 1).

Leave your code (including the definitions of Counter, CounterRep and related functions) in a file varinc.f in your homework10 repository. Note that open recursion is not needed for this program.

3 Reading a SASyLF Proof (5 points)

The soundness proof for FJ (to be finished in the following section) is the largest proof we have worked with or that we will work with this semester. (My solution is over 3000 lines.) Mechanizing a proof often shows how details were ignored in the natural language proof.

Ignoring the natural numbers module, go over the helper lemmas (that is, all lemmas up to but not including “progress-loophole”) and classify each according to the following partition (KRONUM):

**Known Fact** Something that didn’t need to be proved in the natural language proof, because it was known already to be true apart from FJ. These fall into the same class as “gt-transitive” in the natural number module.
Repetition SASyLF has no concept of repetition or sets, and thus many things needs to be re-proved for sequences that are also true for individual items. Often this involves mutual induction.

Obvious The lemma is so obvious that Pierce apparently didn’t feel it needed to be stated.

Named The lemma is mentioned in the text and given a name or number (e.g., A.14). The lemma will probably be stated somewhat differently because of the way in which we formalized FJ.

Unused The lemma is not actually necessary to prove progress or preservation. It could have been omitted. (Of course, you can’t be sure about this classification until you complete the proofs.)

Missing The lemma is not obvious, but is necessary for certain steps in the proof of progress or preservation.

Please explain your answers, especially since a lemma could be classified (correctly!) in different ways, depending on how the classes are interpreted.

4 Soundness of FJ (10 points)

Complete the SASyLF proofs in the skeleton file (homework11.slf) to prove that FJ is sound. FJ uses algorithmic typing rather than T-Sub which requires that the preservation theorem be permitted to compute a subtype for the result. On the other hand, the canonical forms lemma doesn’t need to worry about subtyping any more.

5 Graduate Students (10 points)

Find a mechanized proof of the soundness of FJ (in Coq or Isabelle or Twelf) and go back through the KRONUM classification of lemmas and see if they have analogues in this other proof. If the formalization is very different (such as not permitting methods to take multiple parameters), you may find fewer analogues. Try to avoid using mechanized proofs of extensions of FJ unless the proof includes the soundness of FJ as a subsystem. Otherwise the details of the extension can confuse the proof of the basic FJ system.