Computer Science 131, Spring 2001
Assignment 4: Type Safety

Out: Wednesday, February 14

Due: Wednesday, February 21, 11:00am

This assignment involves no programming. Your written answers must be given directly to the professor (or put under his office door, 1253 Olin). *Your work must be clearly legible.* If you cannot write neatly, use LaTeX or otherwise typeset the proofs.

If you submit your solution late, be sure to mark it with the date and time that it was handed in.

1 Adding pairs to Mini-ML

Consider the following extension of Mini-ML of adding pairs and projections from pairs. The abstract syntax is extended as follows:

$$
\begin{array}{rcl}
v & ::= & \cdots \\
& | & \langle v_1, v_2 \rangle \\
\\
e & ::= & \cdots \\
& | & \langle e_1, e_2 \rangle \\
& | & \texttt{fst}\ e \\
& | & \texttt{snd}\ e \\
\\
t, u & ::= & \cdots \\
& | & t_1 \times t_2
\end{array}
$$

A pair of values is considered a value. The new syntax allows creation of a pair, and operations to project out the first or second component of a pair. Unlike SML, there is no pattern-matching for pairs, so in this abstract syntax a function to raise an integer to an positive integer power would look like:

```
fix p(arg) is
  let x be fst(arg) in
     let n be snd(arg) in
       if (n <= 1) then 1 else p(⟨x,n+(-1)⟩)
```

which is a value of type `int×int->int`.

The new typing rules are:

$$\frac{\Gamma \vdash e_1 : t_1 \qquad \Gamma \vdash e_2 : t_2}{\Gamma \vdash \langle e_1, e_2 \rangle : t_1 \times t_2} \tag{26}$$

$$\frac{\Gamma \vdash e : t_1 \times t_2}{\Gamma \vdash \texttt{fst}\ e : t_1} \tag{27}$$

$$\frac{\Gamma \vdash e : t_1 \times t_2}{\Gamma \vdash \texttt{snd}\ e : t_2} \tag{28}$$

and the new evaluation rules are:

$$\frac{e_1 \ \rightarrow \ e_1'}{\langle e_1, e_2 \rangle \ \rightarrow \ \langle e_1', e_2 \rangle} \tag{29}$$

$$\frac{e_2 \ \rightarrow \ e_2'}{\langle v_1, e_2 \rangle \ \rightarrow \ \langle v_1, e_2' \rangle} \tag{30}$$

$$\frac{e \ \rightarrow \ e'}{\texttt{fst}\ e \ \rightarrow \ \texttt{fst}\ e'} \tag{31}$$

$$\frac{}{\texttt{fst}\ \langle v_1, v_2 \rangle \ \rightarrow \ v_1} \tag{32}$$

$$\frac{e \ \rightarrow \ e'}{\texttt{snd}\ e \ \rightarrow \ \texttt{snd}\ e'} \tag{33}$$

$$\frac{}{\texttt{snd}\ \langle v_1, v_2 \rangle \ \rightarrow \ v_2} \tag{34}$$

One can prove the Type Preservation and Progress properties for this extended language by taking the proof for the original system and simply adding new cases corresponding to the new rules. (There will be new cases in the Type Preservation proof corresponding to the new dynamic semantics rules, and new cases in the Progress proof corresponding to the new static semantics rules.) State the new cases required and give the proofs for these cases.

You will need to add new Inversion properties and to extend the Canonical Forms lemma. You should state these extensions, but need not give proofs. Remember that inversion is used only in the proof of type preservation, and canonical forms is used only in the proof of progress; if this is not true of your proof then there's probably something wrong.