Exercise 5.15 The typing rules for Mini-ML in Section 2.5 are not a realistic basis for an implementation, since they require e_1 in an expression of the form **let name** $u = e_1$ **in** e_2 to be re-checked at every occurrence of u in e_2. This is because we may need to assign different types to e_1 for different occurrences of u.

Fortunately, all the different types for an expression e can be seen as instances of a most general *type schema* for e. In this exercise we explore an alternative formulation of Mini-ML which uses explicit type schemas.

$$\begin{array}{rrcl} \text{Types} & \tau & ::= & \mathsf{nat} \mid \tau_1 \times \tau_2 \mid \tau_1 \to \tau_2 \mid \alpha \\ \text{Type Schemas} & \sigma & ::= & \tau \mid \forall\alpha.\, \sigma \end{array}$$

Type schemas σ are related to types τ through *instantiation*, written as $\sigma \preceq \tau$. This judgment is defined by

$$\frac{}{\tau \preceq \tau}\,\mathsf{inst\_tp} \qquad \frac{[\tau'/\alpha]\sigma \preceq \tau}{\forall\alpha.\, \sigma \preceq \tau}\,\mathsf{inst\_all}.$$

We modify the judgment $\Delta \triangleright e : \tau$ and add a second judgment, $\Delta \mathrel{\triangleright\!\!\triangleright} e : \sigma$ stating that e has type schema σ. The typing rule for **let name** now no longer employs substitution, but refers to a schematic type for the definition. It must therefore be possible to assign type schemas to variables which are instantiated when we need an actual type for a variable.

$$\frac{\Delta \mathrel{\triangleright\!\!\triangleright} e_1 : \sigma_1 \qquad \Delta, x{:}\sigma_1 \triangleright e_2 : \tau_2}{\Delta \triangleright \textbf{let name}\ x = e_1\ \textbf{in}\ e_2 : \tau_2}\,\mathsf{tp\_letn} \qquad \frac{\Delta(x) = \sigma \qquad \sigma \preceq \tau}{\Delta \triangleright x : \tau}\,\mathsf{tp\_var}$$

Type schemas can be derived for expressions by means of quantifying over free type variables.

$$\frac{\Delta \triangleright e : \tau}{\Delta \mathrel{\triangleright\!\!\triangleright} e : \tau}\,\mathsf{tpsc\_tp} \qquad \frac{\Delta \mathrel{\triangleright\!\!\triangleright} e : \sigma}{\Delta \mathrel{\triangleright\!\!\triangleright} e : \forall\alpha.\, \sigma}\,\mathsf{tpsc\_all}^{\alpha}$$

Here the premiss of the $\mathsf{tpsc\_all}^{\alpha}$ rule must be parametric in α, that is, α must not occur free in the context Δ.

In the proofs and implementations below you may restrict yourself to the fragment of the language with functions and **let name**, since the changes are orthogonal to the other constructs of the language.

1. Give an example which shows why the restriction on the $\mathsf{tpsc\_all}$ rule is necessary.

2. Prove type preservation for this formulation of Mini-ML. Carefully write out and prove any substitution lemmas you might need, but you may take weakening and exchange for granted.

3. State the theorem which asserts the equivalence of the new typing rules when compared to the formulation in Section 2.5.
4. Prove the easy direction of the theorem in item 3. Can you conjecture the critical lemma for the opposite direction?
5. Implement type schemas, schematic instantiation, and the new typing judgments in Elf.
6. Unlike our first implementation, the new typing rules do not directly provide an implementation of type inference for Mini-ML in Elf. Show the difficulty by means of an example.
7. Implement the proof of type preservation from item 2 in Elf.
8. Implement one direction of the equivalence proof from item 3 in Elf.