HW11: Types.v, Stlc.v

Your presentation's title and abstract

Due Thursday November 15, 11:59pm EST
HW12: StlcProp.v

Due Wednesday November 21, 11:59pm EST
Objectives for today

- Look at a larger-scale formalization of a programming language
- Prove two properties about this language
STLC Properties

1. **Type preservation** (the type of a well-typed term is preserved by reduction):
   If $\{\} \vdash t \in T$ and $t \Rightarrow t'$, then $\{\} \vdash t' \in T$.

2. **Progress** (a well-typed term is either a value or it reduces):
   $\{\} \vdash t \in T$, then either $t$ is a value, or $t \Rightarrow t'$ for some $t'$. 
Type preservation

The interesting case of type preservation is:

\[
\begin{align*}
HT2 & : \text{empty} \mid - t2 \in T11 \\
HT1 & : \text{empty} \mid - \text{tabs} x \cdot T \cdot t12 \in \text{TArrow} T11 T12 \quad (* \{\} \vdash \lambda x : T. \, t12 \in T11 \to T12 *) \\
\text{empty} & \mid - [x := t2] t12 \in T12
\end{align*}
\]

We can simplify HT1 and get:

\[
\begin{align*}
HT2 & : \text{empty} \mid - t2 \in T11 \\
H1 & : \text{empty} \land \{x \to T11\} \mid - t12 \in T12 \\
\text{empty} & \mid - [x := t2] t12 \in T12
\end{align*}
\]
Type preservation

Restating the previous proof state:

\[
\begin{align*}
HT2 : \text{empty} & \mid \text{t}_2 \in T_{11} \\
H1 : \text{empty} & \& \{\{x \rightarrow T_{11}\}\} \mid \text{t}_{12} \in T_{12} \\
\text{empty} & \mid [x := \text{t}_2] t_{12} \in T_{12}
\end{align*}
\]

- We are saying that \text{t}_2 has a type \text{T}_{11}.
- We are saying that if \text{x} has type \text{T}_{11}, then \text{t}_{12} has type \text{T}_{12}.
- We want to show that \text{t}_{12} has a type \text{T}_{12}, by replacing \text{x} by \text{t}_2 in \text{t}_{12}.

Before, we prove type-preservation, we need to show that substitution preserves the type of the expression.
Substitution type-preservation

Restating the previous proof state:

\[
\begin{align*}
\text{HT2} & : \text{empty} \mid - \ t_2 \ \text{in} \ T_{11} \\
\text{H1} & : \text{empty} \& \{\{x \rightarrow T_{11}\}\} \mid - \ t_{12} \ \text{in} \ T_{12} \\
\text{empty} & \mid [x := t_2] \ t_{12} \ \text{in} \ T_{12}
\end{align*}
\]

Notice, in order to know that \( t_{12} \) has type \( T_{12} \) we must know that \( x \) has a type \( T_{11} \), however our goal has no \( x \). The typing context in the goal is stronger than that of \( \text{H1} \). So, how can this be provable?
Substitution type-preservation

Restating the previous proof state:

\[\begin{align*}
HT2 &: \text{empty} \ |- \ t_2 \ \text{in} \ T_{11} \\
H1 &: \text{empty} \ &\& \{\{x \rightarrow T_{11}\}\} \ |- \ t_{12} \ \text{in} \ T_{12} \\
\hline
\text{empty} \ |- \ [x := t_2] \ t_{12} \ \text{in} \ T_{12}
\end{align*}\]

Notice, in order to know that \(t_{12}\) has type \(T_{12}\) we must know that \(x\) has a type \(T_{11}\), however our goal has no \(x\). The typing context in the goal is \textit{stronger} than that of \(H1\). So, how can this be provable?

The reason is that \(t_2\) is well typed with an \textit{empty} context, it doesn't need any typing information to be well typed. Which means, it does not need to know the type of \(x\) and, therefore, we can \textit{strengthen} the typing context of \(H1\) and get that of the goal.
Type preservation
↓
Substitution lemma
Substitution Lemma

Lemma substitution_preserves_typing_try0. If \( \{ \} \vdash v \in V \) and \( \{ x \mapsto V \} \vdash t \in T \), then \( \{ \} \vdash [x := v]t \in T \).

The proof follows by induction on the structure of \( t \). We quickly get stuck on the case for \( T_{\text{Abs}} \) when \( t = \lambda y : U . t' \) and \( x \neq y \).

\[
\text{IHt : } \forall x \ U \ v \ T, \quad \emptyset \land \{ x \mapsto U \} \vdash t \ \text{in} \ T \rightarrow \emptyset \land v \ \text{in} \ U \rightarrow \emptyset \land [x := v] t \ \text{in} \ T
\]

\[
\text{H0 : } \emptyset \land v \ \text{in} \ V
\]

\[
\text{H6 : } \emptyset \land \{ x \mapsto V; y \mapsto U \} \vdash t \ \text{in} \ T
\]

\[
\text{Heq : } x \neq y
\]

\[
\left\|\begin{array}{c}
\text{empty} & \{ y \mapsto U \} \vdash [x := v] t \ \text{in} \ T
\end{array}\right\| 
\]

(1/1)
Substitution Lemma

Lemma \texttt{substitution\_preserves\_typing\_try0}. If $\{\} \vdash v \in V$ and $\{x \mapsto V\} \vdash t \in T$, then $\{\} \vdash [x := v]t \in T$.

The proof follows by induction on the structure of $t$. We quickly get stuck on the case for $T_{\text{Abs}}$ when $t = \lambda y: U.t'$ and $x \neq y$.

IHt : \forall x \ U \ v \ T,
\quad \text{empty} \ & \ \{x \mapsto U\} \vdash t \mid T \rightarrow \text{empty} \quad \neg \ v \mid U \rightarrow \text{empty} \quad \neg \ [x := v] \ t \mid T

H0 : \text{empty} \vdash v \mid V

H6 : \text{empty} \ & \ \{x \mapsto V; y \mapsto U\} \vdash t \mid T

Heq : x \neq y

\[\text{-------------------------------------}(1/1)\]

\text{empty} \ & \ \{y \mapsto U\} \vdash [x := v] \ t \mid T

\textbf{We need to prove a stronger result! We need to generalize the context.}

Lemma. If $\{\} \vdash v \in V$ and $\Gamma \& \{x \mapsto V\} \vdash t \in T$, then $\Gamma \vdash [x := v]t \in T$. 
Lemma. If \( \{ \} \vdash v \in V \) and \( \Gamma \& \{ x \mapsto V \} \vdash t \in T \), then \( \Gamma \vdash [x := v]t \in T \).

Proof. There are two interesting cases to consider: \( T_{\text{Var}} \) and \( T_{\text{Abs}} \). Case \( T_{\text{Var}} \):

\[
\begin{align*}
\text{Ht'}: \text{empty} & \vdash v \ \text{in} \ U \\
\text{H2:} & (\Gamma \& \{ \{ x \mapsto U \} \}) \ s = \text{Some} \ T \\
\end{align*}
\]

\[\text{-------------}(1/1)\]

\[\Gamma \vdash \text{if beq_string} \ x \ s \ \text{then} \ v \ \text{else} \ \text{tvar} \ s \ \text{in} \ T \]

After doing a case analysis on whether \( x = s \) (see goal), we get:

\[
\begin{align*}
\text{Ht'}: \text{empty} & \vdash v \ \text{in} \ T \\
\text{-------------}(1/1)\]

\[\Gamma \vdash v \ \text{in} \ T \]

Let us prove the above in a new lemma: context weakening.
Substitution Lemma (2/3)

Case $T_{Abs}$ when $t = \lambda y : T.t_0$ and $x \neq y$.

Let us prove the above in a new lemma: context rearrange.
Substitution Lemma (3/3)

To be able to prove the substitution lemma we need the auxiliary lemmas:

1. **Context weakening:**
   If \( \{ \} \vdash v \in T \), then \( \Gamma \vdash v \in T \) for any context \( \Gamma \).

2. **Context rearrange:**
   If \( \Gamma \& \{ x \mapsto U; y \mapsto T \} \vdash t \in V \) and \( x \neq y \), then \( \Gamma \& \{ y \mapsto T; x \mapsto U \} \vdash t \in V \).
Type preservation
↓
Substitution lemma
↓
Context weakening
Context weakening

**Theorem.** If \( \{ \} \vdash v \in T \), then \( \Gamma \vdash v \in T \) for any context \( \Gamma \).

**Lemma** context\_weakening:

\[
\text{forall } v, T, \\
\text{empty } \vdash v \ \in \ T \\
\text{forall } \Gamma, \ Gamma \vdash v \ \in \ T.
\]

By induction on \( v \) we get the following when \( v \) is \text{tabs} \ s \ t \ v' \ (\text{after renaming } v' \text{ to } v):

\[
\text{IH}v : \text{forall } T : \text{ty}, \text{empty } \vdash v \ \in \ T \rightarrow \text{forall } \Gamma : \text{context}, \ Gamma \vdash v \ \in \ T
\]

\[\text{H5 : empty } & \ {\{s \rightarrow t\}} \vdash v \ \in \ T_{12}\]

\[\text{------------------------------(1/1)}\]

\[\text{Gamma } & \ {\{s \rightarrow t\}} \vdash v \ \in \ T_{12}\]

- **We can't use the induction hypothesis.** We need a stronger theorem.
Context weakening

**Lemma** context\_weakening:

\[
\forall v, T, \text{empty} \vdash v \in T \Rightarrow \\
\forall \Gamma, \Gamma \vdash v \in T.
\]

**Proof.**

\[
\text{induction } v; \text{intros; inversion } H; \text{subst; clear } H. \\
- \text{inversion } H2. \\
- \text{eapply } T\_\text{App; eauto}. \\
- \text{apply } T\_\text{Abs}. \\
\text{Abort.}
\]
Type preservation

Substitution lemma

Context weakening

Context invariance
Context invariance

Let restricted equivalence of contexts be defined as \( \Gamma \equiv_P \Gamma' := \forall x, P(x) \implies \Gamma(x) = \Gamma'(x) \).

**Theorem.** If \( \Gamma \vdash t \in T \) and \( \Gamma \equiv_{\text{free}(t)} \Gamma' \), then \( \Gamma' \vdash t \in T \).

**Definition (free variables).** We say that \( x \) is free in term \( t \), with the following inductive definition:

\[
\begin{align*}
    x & \in \text{free}(x) \\
    x & \in \text{free}(t_1) \\
    x & \in \text{free}(t_1 t_2) \\
    x & \in \text{free}(\text{if } t_1 \text{ then } t_2 \text{ else } t_3) \\
    x & \in \text{free}(\lambda y : T. t) \\
    x & \in \text{free}(\text{if } t_1 \text{ then } t_2 \text{ else } t_3)
\end{align*}
\]
Context invariance (proof)

**Lemma** context_invariance : \( \forall \Gamma \Gamma' \ t \ T, \Gamma \vdash t \ \text{in} \ T \rightarrow (\forall x, \text{appears_free_in } x \ t \rightarrow \Gamma x = \Gamma' x) \rightarrow \Gamma' \vdash t \ \text{in} \ T. \)

By induction on the derivation of \( \Gamma \vdash t \in T \). The interesting case is that of \( T_{\text{Abs}} \), where after applying \( T_{\text{Abs}} \) and the induction hypothesis, we get the following proof state.

\[ \begin{align*}
H0 & : \forall x : \text{string}, \text{appears_free_in } x \ (\text{tabs } y \ T11 \ t12) \rightarrow \Gamma x = \Gamma' x \\
Hafi & : \text{appears_free_in } x1 \ t12 \\
\hline
(\Gamma \& \{y \rightarrow T11\}) x1 & = (\Gamma' \& \{y \rightarrow T11\}) x1
\end{align*} \]

Which holds by unfolding update and testing whether \( x1 = y \).
Type preservation

\[\downarrow\]

Substitution lemma

\[\downarrow\]

Context weakening
Context weakening (proof)

Lemma context_weakening:
\[ \forall v \ T, \emptyset |- v \ \text{in} \ T \rightarrow \forall \Gamma, \Gamma |- v \ \text{in} \ T. \]

The proof follows by applying lemma context_invariance, which yields the following proof state.

| H : empty |- v \ in T |
|------------------------|
| H\(\theta \) : appears_free_in x v |

\[ \text{empty } x = \Gamma x.(1/1) \]

How do we solve this?
Context weakening (proof)

**Lemma** context_weakening:
\[
\forall v, T, \quad \text{empty} \mid - v \in T \Rightarrow \\
\forall \Gamma, \Gamma \mid - v \in T.
\]

The proof follows by applying lemma **context_invariance**, which yields the following proof state.

\[
H : \text{empty} \mid - v \in T \\
H_0 : \text{appears_free_in} x v \\
\hline
\text{empty} x = \Gamma x
\]

How do we solve this? Notice, we are saying that there is a free variable in \( v \) and that \( v \) is typable with an empty context.
No free names in an empty context

**Lemma `typable_empty_closed`.** If `{}` ⊢ v ∈ T, then x ∉ free(v) for any x.
(Proof is homework.)

A direct proof, by induction on the structure of v, quickly leads us astray. *Proving negative values is generally more complicated.* Instead, show a positive result.

**Lemma `free_in_context`.** If x ∈ free(t) and Γ ⊢ T, then Γ(x) = T' for some type T'.

*Proof.* The proof is trivial and follows induction on the derivation of the first hypothesis.
Progress
Progress

**Theorem** progress : forall t T, 
   empty |- t \in T \rightarrow 
   value t \lor exists t', t \Rightarrow t'.