Simply Typed λ-calculus

Lecture 2

Jeremy Dawson

The Australian National University

Semester 2, 2017
Outline

Properties of Curry type system:

- **Type preservation** (also known as *subject reduction*): is typeability preserved by $\beta$-reduction?
  - Context lemmas: how does typing depend on contexts, what changes to contexts that preserve typing.
  - Substitution lemma: is a typeable term (in a given context) also typeable after substitution?

- Subterm lemma: is a subterm of a typeable term also typeable?
- Church-Rosser property for typed $\lambda$-calculus.
Type preservation

- Type preservation, in general (given any type system), relates a program execution to its type.

Why is it important?

- If $f$ is, say, a factorial function, taking a natural number to to a natural number, then we would like $f$ to satisfy:

  \[ f(n) \text{ evaluates to a natural number, for any natural number } n. \]

- If the type system has the type preservation property, then one needs to check the type of $f$ once, and be convinced that $f(n)$ always produces integer.

This is the essence of static analysis in programming language.

Check programs at compile time and don’t need to worry about its runtime behaviours.
Curry type system

\[
\begin{align*}
\Gamma & \vdash \lambda_{Cu} \, x : \tau & \text{axiom, if } (x : \tau) \in \Gamma \\
\Gamma & \vdash \lambda_{Cu} (\lambda x. M) : \tau_1 \rightarrow \tau_2 & \rightarrow \text{intro} \\
\Gamma & \vdash \lambda_{Cu} \, M : \tau' \rightarrow \tau & \text{elim} \\
\Gamma & \vdash \lambda_{Cu} \, N : \tau' \\
\Gamma & \vdash \lambda_{Cu} (M \, N) : \tau
\end{align*}
\]
Substitution (review)

\[
\begin{align*}
  x[x := N] & \equiv N \\
y[x := N] & \equiv y; \text{ provided } x \not\equiv y \\
(P \ Q)[x := N] & \equiv (P[x := N]) (Q[x := N]) \\
(\lambda y. P)[x := N] & \equiv \lambda y.(P[x := N]) \text{ provided } y \not\equiv x \\
(\lambda x. P)[x := N] & \equiv (\lambda x. P)
\end{align*}
\]

Note: assume **Barendregt’s variable convention**: bound names are always chosen to be distinct from free names.
\(\beta\)-reduction (review)

Axiom: \((\lambda x.M) \, N \rightarrow_{\beta} M[x := N]\)

\[
\begin{align*}
M & \rightarrow_{\beta} N \\
M \, P & \rightarrow_{\beta} N \, P \\
\lambda x.M & \rightarrow_{\beta} \lambda x.N
\end{align*}
\]

Note that \(\beta\)-reduction can be applied in any subterm.

\(M \rightarrow_{\beta^*} N\) means \(M\) can be reduced to \(N\) using zero or more \(\beta\)-reduction steps.
Type preservation for Curry type system

Type preservation (subject reduction)

Suppose $M \rightarrow_\beta M'$. Then $\Gamma \vdash_{\lambda_Cu} M : \tau$ implies $\Gamma \vdash_{\lambda_Cu} M' : \tau$.

Problems:

- Need to consider reduction in subterms.
- Need also to prove that substitution preserves typing, because $\beta$-reduction uses substitutions.
- To prove the substitution lemma, we need a series of lemmas about properties of typing context in typing derivation.
- Dealing with bound variables, freshness constraints etc in inductive proofs is messy (we’ll see one example).
Context lemmas: weakening

**Context weakening**

Let $\Gamma$ be a context and let $\Gamma'$ be another context such that $\Gamma \subseteq \Gamma'$. Then $\Gamma \vdash \lambda_{Cu} M : \tau$ implies $\Gamma' \vdash \lambda_{Cu} M : \tau$.

Proof: By induction on the structure of the derivation of $\Gamma \vdash \lambda_{Cu} M : \tau$.

A problem: suppose $M = \lambda x. N$, $\tau \equiv \tau_1 \rightarrow \tau_2$, and $x$ appears in $\Gamma'$, say $(x : \sigma) \in \Gamma'$ and $\sigma \not\equiv \tau_1$.

\[
\frac{\Gamma, x : \tau_1 \vdash N : \tau_2}{\Gamma \vdash (\lambda x. N) : \tau_1 \rightarrow \tau_2} \quad \rightarrow \text{intro} \quad \not\rightarrow \quad \frac{\Gamma', x : \tau_1 \vdash N : \tau_2}{\Gamma' \vdash (\lambda x. N) : \tau_1 \rightarrow \tau_2} \quad \rightarrow \text{intro}
\]

This is not valid because a variable $x$ has two distinct declarations in $\Gamma'$, $x : \tau_1$. 

Jeremy Dawson (ANU)  COMP4630,Lecture 2  Semester 2, 2017  8 / 19
Context lemmas: weakening (continued)

Need to rename $x$ in $(\lambda x. N)$ to a fresh variable, say $z$.

\[
\frac{\Gamma, z : \tau_1 \vdash N[x := z] : \tau_2}{\Gamma \vdash (\lambda z. N[x := z]) : \tau_1 \rightarrow \tau_2} \rightarrow \text{intro}
\]

But how do we know $\Gamma, z : \tau_1 \vdash N[x := z] : \tau_2$ is derivable?

Need to prove this separately as a lemma

Lemma (Renaming)

If $\Gamma, x : \sigma \vdash M : \tau$ is derivable in $\lambda_{Cu}$ and $y$ is a fresh variable not appearing anywhere in $\Gamma$ and $M$, then $\Gamma, y : \sigma \vdash M[x := y] : \tau$ is derivable in $\lambda_{Cu}$ with exactly the same deduction steps.

Proof: left as an exercise.
Context lemmas: free variables

- Recall the definition of free variables in a term:

\[
\begin{align*}
FV(x) &= \{x\} \\
FV(MN) &= FV(M) \cup FV(N) \\
FV(\lambda x.M) &= FV(M) \setminus \{x\}
\end{align*}
\]

- Given a context \( \Gamma = \{x_1 : \tau_1, \ldots, x_n : \tau_n\} \), we denote by \( \text{dom}(\Gamma) \) the **domain** of \( \Gamma \), i.e., the set

\[
\text{dom}(\Gamma) = \{x_1, \ldots, x_n\}.
\]

- Let \( \mathcal{V} \) be a set of variables. The **restriction** of \( \Gamma \) to \( \mathcal{V} \) is defined as:

\[
\Gamma \upharpoonright \mathcal{V} = \{x : \sigma \mid x \in \mathcal{V} \text{ and } (x : \sigma) \in \Gamma\}
\]
If $\Gamma \vdash M : \tau$ is derivable, then the only relevant declarations in $\Gamma$ are those that occur free in $M$. 

Free-variables lemma

1. If $\Gamma \vdash \lambda_{Cu} M : \sigma$ then $FV(M) \subseteq \text{dom}(\Gamma)$.
2. If $\Gamma \vdash \lambda_{Cu} M : \sigma$ then $\Gamma \upharpoonright FV(M) \vdash \lambda_{Cu} M : \sigma$. 
Substitution lemma

We are now ready to prove the following substitution lemma:

Lemma (Substitution)

Suppose \( \Gamma, x : \sigma \vdash_{\lambda_{Cu}} M : \tau \) and \( \Gamma \vdash_{\lambda_{Cu}} N : \sigma \). Then \( \Gamma \vdash_{\lambda_{Cu}} M[x := N] : \tau \).

Notice that \( M[x := N] \) is derivable without \( x : \sigma \).

This says that replacing a free variable \( x \) with a term of the same type preserves typeability.

Proof by induction on the structure of the derivation (OR, on the size or structure of the term \( M \))

Base cases:

- \( M = x \) and \( \tau = \sigma \), then \( M[x := N] \) is \( N \)
- \( M = y \), \( y \neq x \), and \( y : \tau \in \Gamma \), then \( M[x := N] \) is \( y \)
Substitution lemma: one inductive case

Inductive hypothesis: assume it is true for all smaller terms $M$
(OR, assume it is true for all earlier steps in showing that $\Gamma, x : \sigma \vdash M : \tau$)

Example case: $M$ is an application $M \ M'$,
so the last rule in its typing deduction must be

$$
\frac{
\Gamma, x : \sigma \vdash \lambda_{Cu} M : \tau' \rightarrow \tau \\
\Gamma, x : \sigma \vdash \lambda_{Cu} M' : \tau'
}{
\Gamma, x : \sigma \vdash \lambda_{Cu} (M \ M') : \tau}
\rightarrow \text{elim}
$$

By induction, $\Gamma \vdash \lambda_{Cu} M[x := N] : \tau' \rightarrow \tau$ and $\Gamma \vdash \lambda_{Cu} M'[x := N] : \tau'$.
Then

$$
\frac{
\Gamma \vdash \lambda_{Cu} M[x := N] : \tau' \rightarrow \tau \\
\Gamma \vdash \lambda_{Cu} M'[x := N] : \tau'
}{
\Gamma \vdash \lambda_{Cu} (M[x := N]) (M'[x := N]) : \tau}
\rightarrow \text{elim}
$$

And as $(M \ M')[x := N] \equiv (M[x := N]) (M'[x := N])$
(by definition of ... $[x := N]$) we have $(M \ M')[x := N] : \tau$, as required
Type preservation (subject reduction)

Finally, the main property we want to prove (let’s give it a ‘theorem’ status to say it’s important):

**Theorem (Subject reduction)**

Suppose $M \rightarrow_{\beta} M'$. Then $\Gamma \vdash_{\lambda_{cu}} M : \tau$ implies $\Gamma \vdash_{\lambda_{cu}} M' : \tau$.

Proof by induction on the size of $M$, OR on the structure of the proof that $M \rightarrow_{\beta} M'$

base case in definition of $\rightarrow_{\beta}$: $(\lambda x. M) \ N \rightarrow_{\beta} M[x := N]$, the typing deduction of $(\lambda x. M) \ N : \tau$ must look like this;

\[
\frac{
\Gamma, x : \sigma \vdash M : \tau
}{
\Gamma \vdash \lambda x. M : \sigma \rightarrow \tau
} \quad \rightarrow_{\text{intro}}
\frac{
\Gamma \vdash N : \sigma
}{
\Gamma \vdash (\lambda x. M) \ N : \tau
} \quad \rightarrow_{\text{elim}}
\]

then $\Gamma \vdash M[x := N] : \tau$, by the Substitution Lemma.
Proof of Subject reduction: one inductive case

Inductive hypothesis: assume it is true for all smaller terms \( M \) (OR, assume it is true for all earlier steps in showing that \( M \rightarrow^\beta M' \))

One example case (one rule in definition of \( \rightarrow^\beta \)):

\[
\frac{M \rightarrow^\beta N}{M P \rightarrow^\beta N P}
\]

Given \( M P : \tau \), we want to show \( N P : \tau \)

The derivation of \( M P : \tau \) must conclude with

\[
\frac{\Gamma \vdash_{\lambda Cu} M : \sigma \rightarrow \tau \quad \Gamma \vdash_{\lambda Cu} P : \sigma}{\Gamma \vdash_{\lambda Cu} (M P) : \tau} \quad \rightarrow \text{elim}
\]

so, by induction, have \( N : \sigma \rightarrow \tau \), and

\[
\frac{\Gamma \vdash_{\lambda Cu} N : \sigma \rightarrow \tau \quad \Gamma \vdash_{\lambda Cu} P : \sigma}{\Gamma \vdash_{\lambda Cu} (N P) : \tau} \quad \rightarrow \text{elim}
\]
Church-Rosser property

- Recall the Church-Rosser property for untyped $\lambda$-calculus:

  If $M \xrightarrow{\beta^*} N_1$ and $M \xrightarrow{\beta^*} N_2$ then for some $N_3$, one has $N_1 \xrightarrow{\beta^*} N_3$ and $N_2 \xrightarrow{\beta^*} N_3$.

- In picture:
Church-Rosser for typeable terms

Suppose $M$ is a closed term of type $\tau$ (typeable in $\lambda_{Cu}$) and $M \rightarrow^{\beta^*} N_1$ and $M \rightarrow^{\beta^*} N_2$.

Does there exist a term $N_3$ such that $N_3$ has type $\tau$ and $N_1 \rightarrow^{\beta^*} N_3$ and $N_2 \rightarrow^{\beta^*} N_3$?

Thanks to subject reduction, the answer is yes.
Subterm typeability

If a term $M$ is typeable, does it follow that each subterm of $M$ is typeable?

Let $M'$ be a subterm of $M$. If $\Gamma \vdash_{\lambda_Cu} M : \sigma$ then $\Gamma' \vdash_{\lambda_Cu} M' : \sigma'$ for some $\Gamma'$ and $\sigma'$.

- Note that $\sigma'$ may not be equal to $\sigma$.
- Also $\Gamma'$ need not be related to $\Gamma$.
- Useful to show untypeability of a term.
  - Example: to show $(\lambda x.x \ x) (\lambda y.y \ y)$ untypeable it is enough to show $(\lambda x.x \ x)$ untypeable.

Is the converse true? That is, if every strict subterm of $M$ is typeable, does it follow that $M$ is typeable? (Exercise)
Subject expansion

- Consider the reverse of subject reduction.

  Suppose \( N \) has type \( \tau \) and \( M \rightarrow_\beta N \). Does it follow that \( M \) also has type \( \tau \)?

- Unfortunately, this is not always true.

  Consider this:

  \[
  N = \lambda x \lambda y. y \quad M = (\lambda z \lambda x \lambda y. y) (\lambda p. p p)
  \]

  We have \( M \rightarrow_\beta N \) but \( M \) is untypeable (why?)

- A consequence of this: typeability is not preserved by \( \beta \)-equality \( =_\beta \).