Lecture 8
Semantics of STLC terms in a ccc

Given a cartesian closed category $\mathcal{C}$, given any function $M$ mapping

- ground types $G$ to $\mathcal{C}$-objects $M(G)$
  (which extends to a function mapping all types to objects, $A \mapsto M[A]$, as we have seen)
Semantics of STLC terms in a ccc

Given a cartesian closed category $\mathcal{C}$,
given any function $M$ mapping

- ground types $G$ to $\mathcal{C}$-objects $M(G)$
- constants $c^A$ to $\mathcal{C}$-morphisms $M(c^A) : 1 \to M[A]$

(In a category with a terminal object $1$, given an object $X \in \mathcal{C}$, morphisms $1 \to X$ are sometimes called global elements of $X$.)
Semantics of STLC terms in a ccc

Given a cartesian closed category $\mathbf{C}$, given any function $M$ mapping

- ground types $\mathbf{G}$ to $\mathbf{C}$-objects $M(\mathbf{G})$
- constants $c^A$ to $\mathbf{C}$-morphisms $M(c^A) : 1 \rightarrow M[A]$

we get a function mapping provable instances of the typing relation $\Gamma \vdash t : A$ to $\mathbf{C}$-morphisms

$$M[\Gamma \vdash t : A] : M[\Gamma] \rightarrow M[A]$$

defined by recursing over the proof of $\Gamma \vdash t : A$ from the typing rules (which follows the structure of $t$):
Semantics of STLC terms in a ccc

Variables:

\[ M[\Gamma, x : A \vdash x : A] = M[\Gamma] \times M[A] \xrightarrow{\pi_2} M[A] \]

\[ M[\Gamma, x' : A' \vdash x : A] = \]

\[ M[\Gamma] \times M[A'] \xrightarrow{\pi_1} M[\Gamma] \xrightarrow{M[\Gamma \vdash x : A]} M[A] \]

Constants:

\[ M[\Gamma \vdash c^A : A] = M[\Gamma] \xrightarrow{\langle \rangle} 1 \xrightarrow{M(c^A)} M[A] \]

Unit value:

\[ M[\Gamma \vdash () : \text{unit}] = M[\Gamma] \xrightarrow{\langle \rangle} 1 \]
Semantics of STLC terms in a ccc

Pairing:

\[
M[\Gamma \vdash (s, t) : A \times B] = \\
M[\Gamma] \xrightarrow{\langle M[\Gamma \vdash s : A], M[\Gamma \vdash t : B]\rangle} M[A] \times M[B]
\]

Projections:

\[
M[\Gamma \vdash \text{fst } t : A] = \\
M[\Gamma] \xrightarrow{M[\Gamma \vdash t : A \times B]} M[A] \times M[B] \xrightarrow{\pi_1} M[A]
\]
Semantics of STLC terms in a ccc

Pairing:

\[ M[\Gamma \vdash (s, t) : A \times B] = \]
\[ M[\Gamma] \xrightarrow{\langle M[\Gamma \vdash s : A], M[\Gamma \vdash t : B] \rangle} M[A] \times M[B] \]

Projections:

\[ M[\Gamma \vdash \text{fst } t : A] = \]
\[ M[\Gamma] \xrightarrow{M[\Gamma \vdash t : A \times B]} M[A] \times M[B] \xrightarrow{\pi_1} M[A] \]

Given that \( \Gamma \vdash \text{fst } t : A \) holds, there is a unique type \( B \) such that \( \Gamma \vdash t : A \times B \) already holds.

Lemma. If \( \Gamma \vdash t : A \) and \( \Gamma \vdash t : B \) are provable, then \( A = B \).
Semantics of STLC terms in a ccc

Pairing:

\[ M[\Gamma \vdash (s, t) : A \times B] = \]

\[ M[\Gamma] \xrightarrow{\langle M[\Gamma \vdash s : A], M[\Gamma \vdash t : B] \rangle} M[A] \times M[B] \]

Projections:

\[ M[\Gamma \vdash \text{snd } t : B] = \]

\[ M[\Gamma] \xrightarrow{M[\Gamma \vdash t : A \times B]} M[A] \times M[B] \xrightarrow{\pi_2} M[B] \]

(As for the case of \texttt{fst}, if \( \Gamma \vdash \text{snd } t : B \), then \( \Gamma \vdash t : A \times B \) already holds for a unique type \( A \).)
Semantics of STLC terms in a ccc

Function abstraction:

\[ M[\Gamma \vdash \lambda x : A.t : A \to B] = \]

\[ \text{cur } f : M[\Gamma] \to (M[A] \to M[B]) \]

where

\[ f = M[\Gamma, x : A \vdash t : B] : M[\Gamma] \times M[A] \to M[B] \]
Semantics of STLC terms in a ccc

Function application:

\[ M[\Gamma \vdash s \ t : B] = \]

\[ M[\Gamma] \xrightarrow{(f,g)} (M[A] \rightarrow M[B]) \times M[A] \xrightarrow{\text{app}} M[B] \]

where

\[ A = \text{unique type such that } \Gamma \vdash s : A \rightarrow B \text{ and } \Gamma \vdash t : A \text{ already holds (exists because } \Gamma \vdash s \ t : B \text{ holds)} \]

\[ f = M[\Gamma \vdash s : A \rightarrow B] : M[\Gamma] \rightarrow (M[A] \rightarrow M[B]) \]

\[ g = M[\Gamma \vdash t : A] : M[\Gamma] \rightarrow M[A] \]
Example

Consider $t \triangleq \lambda x : A. g(f \, x)$ so that $\Gamma \vdash t : A \to C$ when $\Gamma \triangleq \circ, f : A \to B, g : B \to C$.


$$M[\Gamma] = (1 \times Y^X) \times Z^Y$$

$$M[\Gamma, x : A] = ((1 \times Y^X) \times Z^Y) \times X$$

$$M[\Gamma, x : A \vdash x : A] = \pi_2$$

$$M[\Gamma, x : A \vdash g : B \to C] = \pi_2 \circ \pi_1$$

$$M[\Gamma, x : A \vdash f : A \to B] = \pi_2 \circ \pi_1 \circ \pi_1$$

$$M[\Gamma, x : A \vdash f \, x : B] = \text{app} \circ \langle \pi_2 \circ \pi_1 \circ \pi_1 , \pi_2 \rangle$$

$$M[\Gamma, x : A \vdash g \, (f \, x) : C] = \text{app} \circ \langle \pi_2 \circ \pi_1 , \text{app} \circ \langle \pi_2 \circ \pi_1 \circ \pi_1 , \pi_2 \rangle \rangle$$

$$M[\Gamma \vdash t : A \to C] = \text{cur}(\text{app} \circ \langle \pi_2 \circ \pi_1 , \text{app} \circ \langle \pi_2 \circ \pi_1 \circ \pi_1 , \pi_2 \rangle \rangle)$$
STLC equations

take the form $\Gamma \vdash s = t : A$ where $\Gamma \vdash s : A$ and $\Gamma \vdash t : A$ are provable.

Such an equation is satisfied by the semantics in a ccc if $M[\Gamma \vdash s : A]$ and $M[\Gamma \vdash t : A]$ are equal $\mathcal{C}$-morphisms $M[\Gamma] \to M[A]$.

Qu: which equations are always satisfied in any ccc?
STLC equations

take the form \( \Gamma \vdash s = t : A \) where \( \Gamma \vdash s : A \) and \( \Gamma \vdash t : A \) are provable.

Such an equation is satisfied by the semantics in a ccc if \( M[\Gamma \vdash s : A] \) and \( M[\Gamma \vdash t : A] \) are equal \( C \)-morphisms \( M[\Gamma] \to M[A] \).

**Qu:** which equations are always satisfied in any ccc?

**Ans:** \((\alpha)\beta\eta\)-equivalence — to define this, first have to define alpha-equivalence, substitution and its semantics.
Alpha equivalence of STLC terms

The names of $\lambda$-bound variables should not affect meaning.

E.g. $\lambda f : A \rightarrow B. \lambda x : A. f \ x$ should have the same meaning as $\lambda x : A \rightarrow B. \lambda y : A. x \ y$
Alpha equivalence of STLC terms

The names of $\lambda$-bound variables should not affect meaning.

E.g. $\lambda f : A \rightarrow B. \lambda x : A. f x$ should have the same meaning as $\lambda x : A \rightarrow B. \lambda y : A. x y$

This issue is best dealt with at the level of syntax rather than semantics: from now on we re-define “STLC term” to mean not an abstract syntax tree (generated as described before), but rather an equivalence class of such trees with respect to alpha-equivalence $s =_{\alpha} t$, defined as follows...

(Alternatively, one can use a “nameless” (de Bruijn) representation of terms.)
Alpha equivalence of STLC terms

\[
\begin{array}{llll}
\text{c}^A =_{\alpha} \text{c}^A & \quad x =_{\alpha} x & \quad () =_{\alpha} () & \quad \text{fst } t =_{\alpha} \text{fst } t' \\
\quad t =_{\alpha} t' & \quad \text{snd } t =_{\alpha} \text{snd } t' & \quad \phi & \quad (y \ x) \cdot t =_{\alpha} (y \ x') \cdot t' & \quad y \text{ does not occur in } \{x, x', t, t'\} \\
\quad s =_{\alpha} s' & \quad t =_{\alpha} t' & \quad \text{snd } t =_{\alpha} \text{snd } t' \quad \phi & \quad s t =_{\alpha} s' t' & \quad \lambda x : A. t =_{\alpha} \lambda x' : A. t' \\
\end{array}
\]
Alpha equivalence of STLC terms

\[ c^A =_\alpha c^A \]
\[ x =_\alpha x \]
\[ () =_\alpha () \]
\[ s =_\alpha s' \quad t =_\alpha t' \]
\[ (s, t) =_\alpha (s', t') \]
\[ \text{fst } t =_\alpha \text{fst } t' \]
\[ t =_\alpha t' \]
\[ \text{snd } t =_\alpha \text{snd } t' \]
\[ s =_\alpha s' \quad t =_\alpha t' \]
\[ st =_\alpha s't' \]

(y x) \cdot t =_\alpha (y x') \cdot t'

\[ \lambda x : A. t =_\alpha \lambda x' : A. t' \]

result of replacing all occurrences of x with y in t
Alpha equivalence of STLC terms

\[ c^A =_\alpha c^A \]
\[ x =_\alpha x \]
\[ () =_\alpha () \]
\[ s =_\alpha s' \quad t =_\alpha t' \]
\[ (s, t) =_\alpha (s', t') \]
\[ \text{fst } t =_\alpha \text{fst } t' \]
\[ t =_\alpha t' \]
\[ \text{snd } t =_\alpha \text{snd } t' \]
\[ s =_\alpha s' \quad t =_\alpha t' \]
\[ s t =_\alpha s't' \]
\[ (y x) \cdot t =_\alpha (y x') \cdot t' \quad y \text{ does not occur in } \{x, x', t, t'\} \]
\[ \lambda x : A. t =_\alpha \lambda x' : A. t' \]

E.g.

\[ \lambda x : A. x x =_\alpha \lambda y : A. y y \neq_\alpha \lambda x : A. x y \]
\[ (\lambda y : A. y) x =_\alpha (\lambda x : A. x) x \neq_\alpha (\lambda x : A. x) y \]
Substitution

\[ t[s/x] \] = result of replacing all free occurrences of variable \( x \) in term \( t \) (i.e. those not occurring within the scope of a \( \lambda x : A. \_ \) binder) by the term \( s \), alpha-converting \( \lambda \)-bound variables in \( t \) to avoid them “capturing” any free variables of \( t \).

E.g. \( (\lambda y : A. (y, x))[y/x] \) is \( \lambda z : A. (z, y) \) and is not \( \lambda y : A. (y, y) \)
Substitution

\[ t[s/x] \] = result of replacing all free occurrences of variable \( x \) in term \( t \) (i.e. those not occurring within the scope of a \( \lambda x : A. \) binder) by the term \( s \), alpha-converting \( \lambda \)-bound variables in \( t \) to avoid them “capturing” any free variables of \( t \).

E.g. \( (\lambda y : A. (y, x))[y/x] \) is \( \lambda z : A. (z, y) \) and is not \( \lambda y : A. (y, y) \)

The relation \( t[s/x] = t' \) can be inductively defined by the following rules...
Substitution

<table>
<thead>
<tr>
<th>Rule</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$c^A[s/x] = c^A$</td>
</tr>
<tr>
<td>2</td>
<td>$x[s/x] = s$</td>
</tr>
<tr>
<td>3</td>
<td>$y \neq x$</td>
</tr>
<tr>
<td>4</td>
<td>$y[s/x] = y$</td>
</tr>
<tr>
<td>5</td>
<td>$()[s/x] = ()$</td>
</tr>
<tr>
<td>6</td>
<td>$t_1[s/x] = t'_1$</td>
</tr>
<tr>
<td>7</td>
<td>$t_2[s/x] = t'_2$</td>
</tr>
<tr>
<td>8</td>
<td>$(t_1, t_2)[s/x] = (t'_1, t'_2)$</td>
</tr>
<tr>
<td>9</td>
<td>$(\text{fst } t)[s/x] = \text{fst } t'$</td>
</tr>
<tr>
<td>10</td>
<td>$(\text{snd } t)[s/x] = \text{snd } t'$</td>
</tr>
<tr>
<td>11</td>
<td>$t[s/x] = t'$</td>
</tr>
<tr>
<td>12</td>
<td>$(t_1 \cdot t_2)[s/x] = t'_1 \cdot t'_2$</td>
</tr>
<tr>
<td>13</td>
<td>$(\lambda y : A. \ t)[s/x] = \lambda y : A. \ t'$</td>
</tr>
</tbody>
</table>

$t \neq x$ and $y$ does not occur in $s$
Semantics of substitution in a ccc

**Substitution Lemma** If $\Gamma \vdash s : A$ and $\Gamma, x : A \vdash t : B$ are provable, then so is $\Gamma \vdash t[s/x] : B$.

**Substitution Theorem** If $\Gamma \vdash s : A$ and $\Gamma, x : A \vdash t : B$ are provable, then in any ccc the following diagram commutes:

$$
\begin{array}{ccc}
M[\Gamma] & \xrightarrow{\langle \text{id}, M[\Gamma \vdash s : A] \rangle} & M[\Gamma] \times M[A] \\
\downarrow M[\Gamma \vdash t[s/x] : B] & & \downarrow M[\Gamma, x : A \vdash t : B] \\
M[B] & & M[B]
\end{array}
$$