Lecture 5

Simply typed lambda calculus

The syntax is now

$$e \ ::= \ x \mid e\ e \mid \lambda(x:T)\ e \mid \mathsf{zero} \mid \mathsf{succ}\ e$$

where

$$T, A \ ::= \ \mathsf{Nat} \mid T \to T$$

The typing rule are of the form $\Gamma \vdash t : T$ where Γ is a *context* i.e. a list of typing declaration $x : T$.

$$\frac{}{\Gamma, x : T \vdash x : T} \qquad \frac{\Gamma \vdash x : T}{\Gamma, y : A \vdash x : T} x \neq y \qquad \frac{}{\Gamma \vdash \mathsf{zero} : \mathsf{Nat}} \qquad \frac{\Gamma \vdash e : \mathsf{Nat}}{\Gamma \vdash \mathsf{succ}\ e : \mathsf{Nat}}$$

$$\frac{\Gamma \vdash t_0 : A \to B \qquad \Gamma \vdash t_1 : A}{\Gamma \vdash t_0\ t_1 : B} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda(x : A)\ t : A \to B}$$

A term of the form $\lambda(x : A)\ x\ x$ will *not* be well-typed.

Lemma 0.1 *If* $\vdash t : A$ *and* $\Gamma, x : A \vdash e : B$ *then* $\Gamma \vdash e(t/x) : B$

From this Lemma we can prove

Theorem 0.2 *(preservation) If* $t : A$ *and* $t \to t'$ *then* $t' : A$.

Theorem 0.3 *(progress) If* $\vdash t : A$ *then* t *is a value or* $\exists t'\ t \to t'$

Closures and evaluation

We define:

Closures $c \ ::= \ (\lambda\ A\ t, \rho) \mid c\ c \mid \mathsf{zero} \mid \mathsf{succ}\ c$

Environment $\rho \ ::= \ () \mid \rho, c$

Values $v \ ::= \ nv \mid (\lambda\ A\ t, \rho) \qquad nv \ ::= \ \mathsf{zero} \mid \mathsf{succ}\ nv$

Susbstitution

$$0(\rho, c) = c \qquad (n+1)(\rho, c) = n\rho \qquad (e_0\ e_1)\rho = e_0\rho\ (e_1\rho) \qquad (\lambda e)\rho = (\lambda e, \rho)$$

$$\mathsf{zero}\rho = \mathsf{zero} \qquad (\mathsf{succ}\ e)\rho = \mathsf{succ}\ (e\rho)$$

Evaluation

$$\frac{}{(\lambda\ A\ t)\rho\ c \to t(\rho, c)} \qquad \frac{c_0 \to c_0'}{c_0\ c_1 \to c_0'\ c_1}$$

$$\frac{c \to c'}{\mathsf{succ}\ c \to \mathsf{succ}\ c'}$$

Logical relations/predicates

A logical predicate is a predicate $P_A(c)$ on terms of type A such that

$$\frac{P_{\mathsf{Nat}}(c') \qquad c \to c'}{P_{\mathsf{Nat}}(c)} \qquad \frac{}{P_{\mathsf{Nat}}(\mathsf{zero})} \qquad \frac{P_{\mathsf{Nat}}(c)}{P_{\mathsf{Nat}}(\mathsf{succ}\ c)}$$

and $P_{A \to B}(c_0) \Leftrightarrow \forall c_1\ (P_A(c_1) \to P_B(c_0\ c_1))$.

Theorem 0.4 *We have for* all *type A*

$$\frac{P_A(c') \qquad c \to c'}{P_A(c)}$$

We define $P_\Gamma(\rho)$ by $P_{()}()$ and $P_{\Gamma.A}(\rho, c)$ is $P_\Gamma(\rho)$ and $P_A(c)$.

Theorem 0.5 *If $\Gamma \vdash t : A$ and $P_\Gamma(\rho)$ then $P_A(t\rho)$. In particular if $\vdash t : A$ then $P_A(t())$.*

$$\frac{}{\mathsf{zero} : \mathsf{Nat}} \qquad \frac{c : \mathsf{Nat}}{\mathsf{succ}\ c : \mathsf{Nat}}$$

$$\frac{c_0 : A \to B \qquad c_1 : A}{c_0\ c_1 : B} \qquad \frac{\Gamma \vdash t : A \qquad \rho : \Gamma}{t\rho : A}$$

$$\frac{}{() : ()} \qquad \frac{\rho : \Gamma \qquad c : A}{(\rho, c) : \Gamma.A}$$

Theorem 1: *If $c : A$ then c is a value or $\exists c'\ (c \to c')$*

Theorem 2: *If $c : A$ and $c \to c'$ then $c' : A$*

Normalization Theorem

We define $R_A(c)$ by induction on A

$R_{\mathsf{Nat}}(c)$ is $\exists v\ (c \to^* v)$
$R_{A \to B}(c)$ is $\forall c' : A\ (R_A(c') \to R_B(c\ c'))$

Lemma 1: *If $c \to c'$ and $c : A$ and $R_A(c')$ then $R_A(c)$*

So R_A is a logical predicate. It follows that we have.

Theorem: *If $c : \mathsf{Nat}$ then $\exists v\ (c \to^* v)$.*

A small term with a large value

We can define $\mathsf{exp}\ A = A \to A$ and the term $\mathsf{twice}\ A : \mathsf{exp}\ (\mathsf{exp}\ A) = \lambda(\mathsf{exp}\ A)\lambda A\ 1\ (1\ 0)$

It is possible then to define $\mathsf{twice}_n = \mathsf{twice}\ (\mathsf{exp}^n\ \mathsf{Nat})$ and the term

$$t = (((\ldots((\mathsf{twice}_n\ \mathsf{twice}_{n-1})\ \mathsf{twice}_{n-2})\ldots)\ \mathsf{twice}_0)\ \mathsf{succ})\ \mathsf{zero}$$

is then of type $t : \mathsf{Nat}$. By the Theorem, there exists v such that $t \to^* v$. However v is of the form $\mathsf{succ}^k\ \mathsf{zero}$ where k is a tower of n exponentials $k = 2^{2^{2^{\cdots}}}$.

Denotational semantics

For $\Gamma \vdash t : A$ and ρ in $[\![\Gamma]\!]$ we define $[\![t]\!]\rho$ in $[\![A]\!]$ where

- $[\![\mathsf{Nat}]\!]$ is the set of natural numbers
- $[\![A \to B]\!]$ is the set of functions from the set $[\![A]\!]$ to the set $[\![B]\!]$
- $[\![()]\!]$ is the singleton $\{0\}$ and $[\![\Gamma.A]\!]$ is the product $[\![\Gamma]\!] \times [\![A]\!]$

The definition is by induction on t

$$[\![0]\!](\rho, u) = u \qquad [\![n+1]\!](\rho, u) = [\![n]\!]\rho$$

$$[\![\mathsf{zero}]\!]\rho = 0 \qquad [\![\mathsf{succ}\ e]\!]\rho = 1 + [\![e]\!]\rho$$

$$[\![t_0\ t_1]\!](\rho) = [\![t_0]\!]\rho([\![t_1]\!]\rho) \qquad [\![\lambda\ A\ t]\!]\rho(u) = [\![t]\!](\rho, u)$$

If n is a natural number, we define $q(n)$ of type Nat by $q(0) = \mathsf{zero}$ and $q(n+1) = \mathsf{succ}\ q(n)$. We prove the following result by the technique of logical relation

Theorem 0.6 *If* $\vdash t : \mathsf{Nat}$ *then* $t() \to^* q([\![t]\!])$

Abstract data type and representation independence

We consider two different implemenations of the context

$$test : X \to \mathsf{Bool},\ rev : X \to X,\ init : X$$

One is $X = \mathsf{Bool},\ test = \lambda(x : \mathsf{Bool})x,\ rev = \neg,\ init = \mathsf{true}$ and the other is $X = \mathbb{Z},\ test = \lambda(n : \mathsf{Nat})n > 0,\ init = 1$ and $rev\ x = -x$.

Given two such implementations A_0, f_0, g_0 and A_1, f_1, g_1 we say that they are *related* by a relation R if we have $R(u_0, u_1) \Rightarrow f_0(u_0) = f_1(u_1)$ and $R(u_0, u_1) \Rightarrow R(g_0(u_0), g_1(u_1))$.

Theorem 0.7 *If* $\vdash t(X, test, rev) : \mathsf{Bool}$ *and the two implementations are related then* $[\![t]\!](A_0, f_0, g_0) = [\![t]\!](A_1, f_1, g_1)$.

An example of such a term is $test\ (rev\ (rev\ init))$.