The untyped lambda calculus

CIS 6700, Spring 2023

January 17, 2023

1 Untyped lambda calculus

This document describes the untyped lambda-calculus, with the following grammar. References are to definitions in Barendregt, “The Lambda Calculus: Its Syntax and Semantics”.

tm, t, u	$::=$		terms
	$\mid$	x	variables
	$\mid$	$\lambda x.t$	abstractions
	$\mid$	$t\ u$	function application

In β-reduction, the argument of an application is substituted for the bound variable in an abstraction. We use the $t[x \rightsquigarrow u]$ notation for substituting the term u for the variable x in the term t.

$\boxed{t\ \beta\text{-reduces to}\ u}$ *(Definition 3.1.3)*

$$\textsc{beta-reduct}\quad \frac{}{(\lambda x.t)\ u\ \beta\text{-reduces to}\ t[x \rightsquigarrow u]}$$

We can define a deterministic, small-step evaluation relation by reducing in the heads of applications. This is a *call-by-name* semantics and performs head-reduction.

$\boxed{t \rightsquigarrow t'}$ *(Small-step evaluation)*

$$\textsc{S-app}\quad \frac{t \rightsquigarrow t'}{t\ u \rightsquigarrow t'\ t} \qquad\qquad \textsc{S-beta}\quad \frac{}{(\lambda x.t)\ u \rightsquigarrow t[x \rightsquigarrow u]}$$

We can also define a nondeterministic single-step full-reduction by performing β-reduction in any subterm. Iterating this reduction will convert a term into its β-normal form.

$\boxed{t \longrightarrow_\beta u}$ *(Full β-reduction)*

$$\frac{\textsc{F-beta}}{(\lambda x.t)\ u \longrightarrow_\beta t[x \rightsquigarrow u]} \qquad \textsc{F-abs}\ \frac{t \longrightarrow_\beta t'}{\lambda x.t \longrightarrow_\beta \lambda x.t'} \qquad \textsc{F-app1}\ \frac{t \longrightarrow_\beta t'}{t\ u \longrightarrow_\beta t'\ u}$$

$$\textsc{F-app2}\ \frac{u \longrightarrow_\beta u}{t\ u \longrightarrow_\beta t\ u'}$$

We can define when two terms are equivalent up to β.

$\boxed{t \equiv_\beta u}$ *(β-conversion or β-equality (Definition 2.1.4))*

$$\textsc{eq-beta}\ \frac{}{(\lambda x.t)\ u \equiv_\beta t[x \rightsquigarrow u]} \qquad \textsc{eq-refl}\ \frac{}{t \equiv_\beta t} \qquad \textsc{eq-sym}\ \frac{u \equiv_\beta t}{t \equiv_\beta u} \qquad \textsc{eq-trans}\ \frac{t_1 \equiv_\beta t_2 \qquad t_2 \equiv_\beta t_3}{t_1 \equiv_\beta t_3}$$

$$\textsc{eq-app1}\ \frac{t \equiv_\beta t'}{t\ u \equiv_\beta t'\ u} \qquad \textsc{eq-app2}\ \frac{u \equiv_\beta u'}{t\ u \equiv_\beta t\ u'} \qquad \textsc{eq-abs}\ \frac{t \equiv_\beta t'}{\lambda x.t \equiv_\beta \lambda x.t'}$$

Finally, the Church-Rosser Theorem relies on the definition of parallel reduction. This is a version of reduction that is confluent.

$\boxed{t \Longrightarrow_\beta u}$ *(Parallel reduction (3.2.3))*

$$\textsc{p-beta}\ \frac{t \Longrightarrow_\beta \lambda x.t' \qquad u \Longrightarrow_\beta u'}{t\ u \Longrightarrow_\beta t'[x \rightsquigarrow u']} \qquad \textsc{p-var}\ \frac{}{x \Longrightarrow_\beta x} \qquad \textsc{p-abs}\ \frac{t \Longrightarrow_\beta t'}{\lambda x.t \Longrightarrow_\beta \lambda x.t'} \qquad \textsc{p-app}\ \frac{t \Longrightarrow_\beta t' \qquad u \Longrightarrow_\beta u'}{t\ u \Longrightarrow_\beta t'\ u'}$$

2 Relation operations

Many of the operations above can be generated by applying the following closure operations to the β reduction relation. These operations are parameterized by an arbitrary relation R.

Note that $t \rightarrow_R u$ with R equal to β is the same relation as $t \longrightarrow_\beta u$. And, the compatible, reflexive, symmetic and transitive closure of β is the same relation as $t \equiv_\beta u$.

$\boxed{t \rightarrow_R u}$ *(Compatible closure of R (aka one-step reduction (3.1.5))*

$$\textsc{cc-rel}\ \frac{R\ t\ u}{t \rightarrow_R u} \qquad \textsc{cc-abs}\ \frac{t \rightarrow_R u}{\lambda x.t \rightarrow_R \lambda x.u} \qquad \textsc{cc-app1}\ \frac{t \rightarrow_R t'}{t\ u \rightarrow_R t'\ u} \qquad \textsc{cc-app2}\ \frac{u \rightarrow_R u'}{t\ u \rightarrow_R t\ u'}$$

$\boxed{t \rightarrow_R^{=} u}$ *(reflexive closure of R (3.1.4))*

$$\frac{R\ t\ u}{t \rightarrow_R^{=} u}\ \text{R-REL} \qquad \frac{}{t \rightarrow_R^{=} t}\ \text{R-REFL}$$

$\boxed{t \rightarrow_R^{+} u}$ *(transitive closure of R (3.1.4))*

$$\frac{R\ t\ u}{t \rightarrow_R^{+} u}\ \text{T-REL} \qquad \frac{t_1 \rightarrow_R^{+} t_2 \qquad t_2 \rightarrow_R^{+} t_3}{t_1 \rightarrow_R^{+} t_3}\ \text{T-TRANS}$$

$\boxed{t \rightarrow_R^{*} u}$ *(reflexive-transitive closure of R)*

$$\frac{R\ t\ u}{t \rightarrow_R^{*} u}\ \text{RT-REL} \qquad \frac{}{t \rightarrow_R^{*} t}\ \text{RT-REFL} \qquad \frac{t_1 \rightarrow_R^{*} t_2 \qquad t_2 \rightarrow_R^{*} t_3}{t_1 \rightarrow_R^{*} t_3}\ \text{RT-TRANS}$$

$\boxed{t \leftrightarrow_R u}$ *(symmetric-transitive closure of R)*

$$\frac{R\ t\ u}{t \leftrightarrow_R u}\ \text{ST-REL} \qquad \frac{u \leftrightarrow_R t}{t \leftrightarrow_R u}\ \text{ST-SYM} \qquad \frac{t_1 \leftrightarrow_R t_2 \qquad t_2 \leftrightarrow_R t_3}{t_1 \leftrightarrow_R t_3}\ \text{ST-TRANS}$$

3 Eta-reduction

By changing the definition of the underlying primitive reduction, we can also reason about η-reduction and $\beta\eta$-equivalence. Note that the rule for η-reduction has been stated in a way that generates the approporiate output for the locally-nameless representation in Coq. In this output, the fact that $x \notin \mathsf{fv}t$ is implicit and does not need to be added as a precondition to the rule.

$\boxed{t\ \eta\text{-reduces to}\ u}$ *(Definition 3.1.1 (i))*

$$\frac{t' = t\ x}{\lambda x.t'\ \eta\text{-reduces to}\ t}\ \text{ETA-REDUCT}$$

$\boxed{t\ \beta\eta\text{-reduces to}\ u}$ *(Either β or η reduction (Definition 3.1.1 (ii)))*

$$\frac{t\ \beta\text{-reduces to}\ u}{t\ \beta\eta\text{-reduces to}\ u}\ \text{ETA-BETA} \qquad \frac{t\ \eta\text{-reduces to}\ u}{t\ \beta\eta\text{-reduces to}\ u}\ \text{ETA-ETA}$$