1 Simply typed λ-calculus (ala Church)

Definition 1.1. A **simple type** is a propositional formula built from atoms and the connective $\to$. The set of all simple types is denoted $\Phi_{\to}$.

Definition 1.2. We write x^τ to mean that x is a **variable** of type τ. An **environment** Γ is a finite set of variables $\{x_1^{\tau_1}, \ldots, x_n^{\tau_n}\}$.

Definition 1.3. We say M is a **term** of type τ in Γ, written $\Gamma \vdash M^\tau$, when M can be derived using the following rules.

$$\text{(Var)} \qquad \Gamma, x^\tau \vdash x^\tau$$

$$\text{(Abs)} \qquad \frac{\Gamma, x^\sigma \vdash M^\tau}{\Gamma \vdash (\lambda x^\sigma.M^\tau)^{\sigma\to\tau}}$$

$$\text{(App)} \qquad \frac{\Gamma \vdash M^{\sigma\to\tau} \qquad \Gamma \vdash N^\sigma}{\Gamma \vdash (MN)^\tau}$$

Note in particular that it is only sensible to talk about applications when the types are compatible. There exist λ-terms which cannot be assigned types according to the above rules. One example is $(\lambda x.xx)$.

Example 1.4. Some well-typed terms are:

- $\vdash \lambda x^\sigma.x$ of type $\sigma \to \sigma$.
- $\vdash \lambda x^\sigma y^\tau.x$ of type $\sigma \to \tau \to \sigma$. Note $\to$ is right-associative; i.e. $\sigma \to \tau \to \sigma = \sigma \to (\tau \to \sigma)$.
- $\vdash \lambda x^{\sigma\to\tau\to\rho} y^{\sigma\to\tau} z^\sigma.xz(yz)$ of type $(\sigma \to \tau \to \rho) \to (\sigma \to \tau) \to \sigma \to \rho$.

Some useful properties of the simply typed λ-calculus include:

Theorem 1.5. (Subject reduction) If $\Gamma \vdash M^\sigma$ and $M \to_\beta N$ then $\Gamma \vdash N^\sigma$. Hence well-typed expressions remain well-typed after reductions.

Theorem 1.6. If $\Gamma \vdash M^\sigma$ and $\Gamma \vdash M^\tau$ then $\sigma = \tau$.

Aside: There exists a variant of the simply typed λ-calculus in which bound variables are not given types, in which case this theorem does not hold; for instance, $\lambda x.x$ has both the type $\sigma \to \sigma$ and the type $\tau \to \tau$.

2 Normalisation properties

Definition 2.1. A term M is said to be in **normal form** if M contains no β-redex; that is, no subterm of the form $(\lambda x.P)Q$.

Definition 2.2. Let M be a term. We say that M is **weakly normalisable** if there exists some M' in normal form such that $M \twoheadrightarrow_\beta M'$. We say that M is **strongly normalisable** if there does not exist an infinite β-reduction path starting from M.

There exist terms in the untyped λ-calculus which are neither strongly nor weakly normalisable; for instance $(\lambda x.xx)(\lambda x.xx)$, as well as terms which are weakly but not strongly normalisable such as $(\lambda xy.y)((\lambda x.xx)(\lambda x.xx))$. However, we will see that in the simply typed λ-calculus, every term is strongly normalisable.

Definition 2.3. Suppose that M is strongly normalisable. Then $\nu(M)$ denotes the maximum possible length of a normalisation sequence beginning with N.

The fact that ν is always finite if M is strongly normalisable deserves justification; perhaps M could contain infinitely many reduction paths each of finite length, but unbounded overall. To see that this cannot happen, we note that any given λ-term is finite and hence contains only finitely many β-redexes. Hence at any given point in a β-reduction path, we only have finitely many choices as to which β-redex to reduce next. Hence we can apply König's Lemma; a finitely branching tree with no infinite branch is finite. Specifically, our nodes correspond to λ-terms, and our edges correspond to the choices we can make at a given node.

Definition 2.4. Let τ be a type. The **degree** of τ is

$$\deg(\tau) = \begin{cases} 0 & \text{if } \tau \text{ is atomic} \\ 1 + \max\{\deg(\sigma), \deg(\rho)\} & \text{if } \tau = \sigma \to \rho \end{cases}$$

Essentially, one should think of the degree of a type as its "complexity", a measure of the maximum amount of recursion it contains.

This notion of degree of type allows us to quite readily prove the **weak normalisation theorem** for simply typed lambda calculus; that is, the statement that every term has a normal form. As a proof sketch, the idea is to come up with a β-reduction strategy that is guaranteed to terminate in a normal form. Given a term M containing one or more β-redexes, the strategy will be to reduce the rightmost redex within M with type of maximum degree n. This may produce new redexes, or copy existing ones, but any such redexes are guaranteed to have type of degree less than n. Hence each reduction decreases the number of redexes of maximal degree; repeating this process is therefore guaranteed to terminate in a normal form.

While the simply typed λ-calculus is strongly normalisable, this important property does not come for free. In particular, untyped λ-calculus is well known to be Turing complete, which is not true of simply typed systems. One hint that this is the case is

that the strong normalisation theorem essentially tells us that any function which is able to be encoded by the simply typed system must be calculable by a program which halts.

In fact the expressive power of the simply typed system is even weaker than we would hope. The exact class of functions it can define are the **extended polynomials**; the smallest class of functions closed under compositions, and containing the constants 0 and 1, projections, addition, multiplication, and the conditional function ($cond(a, b, c) = b$ if $a = 0$, and $= c$ otherwise). In particular, we do not have primitive recursion or minimisation in general.

One possible modification of the simply typed calculus which increases the expressive power while still maintaining strong normalisation is System F, which introduces polymorphism.

3 Proof of the strong normalisation theorem

Definition 3.1. We first define RED_τ, the set of **reducible terms** type τ, by induction on type.

- If τ is atomic, then a term of type τ is reducible if it is strongly normalisable.
- Otherwise $\tau = \sigma \to \rho$. A term M of type τ is reducible if, for every reducible term N of type σ, MN is reducible of type ρ.

Definition 3.2. We say a term M of type τ is **prereducible** if $M' \in \mathrm{RED}_\tau$ for every term M' with $M \to_\beta M'$.

Lemma 3.3. Let M be a term of type τ.

1. If $M \in \mathrm{RED}_\tau$ then M is strongly normalisable.
2. If $M \in \mathrm{RED}_\tau$ and $M \twoheadrightarrow_\beta M'$ then $M' \in \mathrm{RED}_\tau$.
3. If M is prereducible and is not an abstraction then $M \in \mathrm{RED}_\tau$.

Proof. Induction on degree of type. Suppose that τ is atomic. (1) is immediate by definition. (2) is the statement that if M is strongly normalisable then so is every term to which M reduces. (3) follows from the fact that if M is prereducible, then any β-reduction path starting from M contains a strongly normalisable term, and hence M is strongly normalisable. Now let $\tau = \sigma \to \rho$, and suppose that the lemma holds for each type of degree less than $\deg(\tau)$.

(1) Suppose that $M \in \mathrm{RED}_\tau$, and let x be a variable of type σ. Since x is prereducible and not an abstraction, by the induction hypothesis $x \in \mathrm{RED}_\sigma$. Since $M \in \mathrm{RED}_\tau$, it follows that $Mx \in \mathrm{RED}_\rho$ and thus Mx is strongly normalisable by the induction hypothesis. So M must be strongly normalisable as well, for if there existed an infinite reduction path within M, there would also exist one in Mx as well as we

could simply expand within M in exactly the same way, ignoring the presence of x entirely.

(2) Suppose that $M \in \mathrm{RED}_\tau$ and $M \twoheadrightarrow_\beta M'$, and let $N \in \mathrm{RED}_\sigma$. Since $M \in \mathrm{RED}_\tau$ we therefore have $MN \in \mathrm{RED}_\sigma$. In addition, $MN \twoheadrightarrow_\beta M'N$, as the same sequence of β-reductions within M works. As $\deg(\rho) < \deg(\tau)$, by the induction hypothesis we have $M'N \in \mathrm{RED}_\rho$. Since this holds for any $N \in \mathrm{RED}_\sigma$, we conclude that $M' \in \mathrm{RED}_\rho$.

(3) Suppose M of type τ is prereducible and not an abstraction. We wish to show that $MN \in \mathrm{RED}_\rho$ for each $N \in \mathrm{RED}_\sigma$; by our induction hypothesis it suffices to show that MN is prereducible. By the induction hypothesis N is strongly normalisable, and so $\nu(N) < \infty$. We will reason by a secondary induction on $\nu(N)$.

For the base case, suppose that $\nu(N) = 0$ and that $MN \to_\beta P$. Since M is not an abstraction and N contains no β-redex, we must have $P = M'N$ for some term M' with $M \to_\beta M'$. Since M is prereducible, $M' \in \mathrm{RED}_\tau$ and hence $P = M'N \in \mathrm{RED}_\rho$, so MN is prereducible.

For the inductive step, suppose that $\nu(N) = n$, and that MN_0 is prereducible for any N_0 with $\nu(N_0) < n$. If $MN \to_\beta P$, we either have $P = M'N$ for some $M \to_\beta M'$, or $P = MN'$ for some $N \to_\beta N'$. If $P = M'N$, the proof that $P \in \mathrm{RED}_\rho$ is the same as in the base case. If instead $P = MN'$, then since N has type σ and $\deg(\sigma) < \deg(\tau)$, by the first induction hypothesis $N' \in \mathrm{RED}_\sigma$. Since $\nu(N') < \nu(N) = n$, by our secondary induction hypothesis we have $P = MN' \in \mathrm{RED}_\rho$. Hence MN is prereducible.

We have shown that MN is prereducible for each $N \in \mathrm{RED}_\sigma$. Since $\deg(\rho) < \deg(\tau)$, it follows that $MN \in \mathrm{RED}_\rho$, and hence $M \in \mathrm{RED}_\tau$. □

Lemma 3.4. If M is a term such that $M[x^\sigma := N] \in \mathrm{RED}_\rho$ for each $N \in \mathrm{RED}_\sigma$, then $\lambda x^\sigma.M \in \mathrm{RED}_{\sigma\to\rho}$.

Proof. Let $N \in \mathrm{RED}_\sigma$. Since $(\lambda x^\sigma.M)N$ is not an abstraction, by Lemma 3.3 it suffices to show that $(\lambda x^\sigma.M)N$ is prereducible.

Note that M and N are both reducible; M being reducible follows from the fact that $M[x := x]$ is reducible by hypothesis. Hence they are both strongly normalisable by Lemma 3.3. We will reason by induction on $\nu(M) + \nu(N)$.

For the base case, suppose that $\nu(M) + \nu(N) = 0$ and that $(\lambda x^\sigma.M)N \to_\beta P$. Since M and N are in normal form, we must have $P = M[x := N]$ which is reducible by hypothesis. So $(\lambda x^\sigma.M)N$ is prereducible, hence reducible by Lemma 3.3.

For the inductive step, suppose that $\nu(M) + \nu(N) = n$, and that $(\lambda x^\sigma.M_0)N_0$ is reducible for any $M_0 \in \mathrm{RED}_\rho, N_0 \in \mathrm{RED}_\sigma$ satisfying $M_0[x := N_0] \in \mathrm{RED}_\rho$ and $\nu(M) + \nu(N) < n$. If $(\lambda x^\sigma.M)N \to_\beta P$, then either

(i) $P = M[x := N]$ which is reducible by hypothesis.

(ii) $P = (\lambda x^\sigma.M')N$, where $M \to_\beta M'$. Then $\nu(M') < \nu(M)$, and since $M[x := N]$ is reducible so is $M'[x := N]$ by Lemma 3.3. It follows from the induction hypothesis that P is reducible.

(iii) $P = (\lambda x^\sigma.M)N'$, where $N \to_\beta N'$. Then $\nu(N') < \nu(N)$, and $M[x := N']$ is reducible again by Lemma 3.3, so by the induction hypothesis P is reducible.

In each case P is reducible. So $(\lambda x^\sigma.M)N$ is prereducible, hence reducible. Since $N \in \mathrm{RED}_\sigma$ was arbitrary, it follows that $\lambda x^\sigma.M \in \mathrm{RED}_{\sigma\to\rho}$. □

Lemma 3.5. Let M be a term (not necessarily reducible). Suppose that the free variables of M are $\vec{x} = (x_1, ..., x_k)$ of types $\tau_1, ..., \tau_k$. If $\vec{N} = (N_1, ..., N_k)$ are reducible terms of types $\tau_1, ..., \tau_k$, then $M[\vec{x} := \vec{N}]$ is reducible.

Proof. Induction on the structure of M.

If M is a variable, trivial.

If $M = P\,Q$, then by then induction hypothesis $P[\vec{x} := \vec{N}]$ and $Q[\vec{x} := \vec{N}]$ are reducible. Hence $P[\vec{x} := \vec{N}]Q[\vec{x} := \vec{N}]$ is reducible by definition. This is exactly $(P\,Q)[\vec{x} := \vec{N}]$.

Lastly, suppose $M = (\lambda y.P)$ is of type $\sigma \to \rho$. By the induction hypothesis $P[\vec{x} := \vec{N}][y := N']$ is reducible for all $N' \in \mathrm{RED}_\sigma$. Hence by Lemma 3.4, $(\lambda y.P[\vec{x} := \vec{N}])$ is reducible. □

From Lemma 3.5, by setting $N_i = x_i$ we immediately obtain that all terms are reducible. Hence by applying Lemma 3.3:

Theorem 3.6. All terms are strongly normalisable.

4 References

1. Sørensen, Urzycz; *Lectures on the Curry-Howard Isomorphism*, Ch.3.
2. Girard; *Proofs and Types*, Ch.4,6
3. W. Gunther; *Strong Normalisation of Propositional Types*, retrieved from http://www.math.cmu.edu/~wgunther/doc.html