Strong Normalisation for System F

Thomas Waring

June 30, 2021

Contents

1 System F **1**

2 Reducibility Candidates **2**

3 Strong Normalisation **5**

1 System F

We mostly presume familiarity with System F, or polymorphic lambda calculus: the primary reference is Girard's (who else's?) [GLT93], chapter 11 (for definitions) and chapter 14 (for normalisation).

A quick refresher. In what follows, the substitution $a[b/c]$ denotes replacing all free occurences of the variable c in a with b. Also, note that terms are considered up to α-equivalence: changing the names of bound variables. For more detail see [Sel08] chapter 8, noting slight differences of notation.

Types are defined inductively, starting from an infinite sequence $X, Y, Z, \ldots$ of *type variables*, and with three rules.

1. Type variables are types (which are *free* in the resulting type).
2. If U and V are types, then $U \to V$ is a type.
3. If V is a type, and X is a type variable, then $\Pi X.V$ is a type. Any previously free occurence of X in V is now bound.

From this, there are five ways to form terms.

1. Variables: an infinite sequence $x^T, y^T, z^T, \ldots$ for each type T.
2. Application: if t and u are terms of type $U \to V$ and U, then tu is a term of type V.
3. λ-abstraction: if x^U is a variable of type U, and v is a term of type V then $\lambda x^U.v$ is a term of type $U \to V$. As before, occurences of x^U in v are bound in $\lambda x^U.v$.
4. Universal application (or extraction): if t is a term of type $\Pi X.V$ and U is type, then tU is a term of type $V[U/X]$.
5. Universal abstraction: if v is a term of type V, then $\Lambda X.v$ is a term of type $\Pi X.V$, so long as X is not free in the type of any free variable of v.

There are two "reduction" operations on terms. The first familiar is familiar (as β-reduction) from simply typed lambda calculus:

$$(\lambda x^U.v)u \rightsquigarrow v[u/x].$$

The second is its equivalent for universal abstraction / application:

$$(\Lambda X.v)U \rightsquigarrow v[U/X].$$

We note without proof that these reduction rules satisfy the *Church-Rosser* property, also known as confluence. Loosely, if $u \rightsquigarrow u'$ and $u \rightsquigarrow u''$ then there is a term v and sequences of reduction steps, starting at u' and u'', and ending at v. See [Gal90], §10 for a proof.

Finally some terminology. For a given term u, define $\nu(u)$ to be the longest sequence of reductions starting with u. For example:

$$(\Lambda X.\lambda x^X.x)Vv^V \rightsquigarrow (\lambda x^V.x)v^V \rightsquigarrow v^V$$

so $\nu((\Lambda X.\lambda x^X.x)Vv^V) = 2$ (if v is a variable, so atomic), as in each case there was a single possible reduction (a single *redex*). The primary goal of these notes is to show that $\nu(u)$ is finite for every term u. This property is identified with "strongly normalising", as by confluence if a normal form (a reduct with no redexes) exists, it is unique.

A term is called *neutral* if it is of the form x, tu or tU. That is, if it does not start with an abstraction of either type.

2 Reducibility Candidates

Definition 2.1. A *reducibility candidate* of type U is a set $\mathcal{R}$ of terms of type U, such that:

- (CR1) If $t \in \mathcal{R}$, then t is strongly normalising.
- (CR2) $t \in \mathcal{R}$ and $t \rightsquigarrow t'$, then $t' \in \mathcal{R}$.
- (CR3) t is neutral, and whenever we convert a redex in t we obtain a term $t' \in \mathcal{R}$, then $t \in \mathcal{R}$ also.

By (CR3), any term which is neutal and normal belongs to every reducibility candidate of the appropriate type.

Lemma 2.2. *The set $\mathcal{SN}_U$ of strongly normalising terms of type U is a reducibility candidate.*

Proof. (CR1) is tautological. If $t \rightsquigarrow t'$, then $\nu(t') < \nu(t)$, so t' is also strongly normalising. If there were an infinite path of reductions starting from t, then the t' in the second step would also not be strong normalising, so $t' \notin \mathcal{SN}_U$. □

Lemma 2.3. *Given reducibility candidates $\mathcal{R}$ and $\mathcal{S}$ of types U and V, the set $\mathcal{R} \to \mathcal{S}$ of terms of type $U \to V$ is defined by:*

$$t \in \mathcal{R} \to \mathcal{S} \iff \forall u(u \in \mathcal{R} \implies tu \in \mathcal{S})$$

is a reducibility candidate.

Proof. (CR1) Given $t \in \mathcal{R} \to \mathcal{S}$ and any u of type U, $\nu(t) \leqslant \nu(tu)$, so t is strongly normalising (noting that $\mathcal{R}$ is nonempty).

(CR2) Let some $t \in \mathcal{R} \to \mathcal{S}$ be given, and t' such that $t \rightsquigarrow t'$. For any $u \in \mathcal{R}$, $tu \rightsquigarrow t'u$, so $t'u \in \mathcal{S}$ (by CR2). This implies $t' \in \mathcal{R} \to \mathcal{S}$.

(CR3) Let some $u \in \mathcal{R}$ and neutral t as in (CR3) be given. As t does not begin with an abstraction, the only possible one-step reductions beginning with tu are $tu \rightsquigarrow t'u$ and $tu \rightsquigarrow tu'$, where $t \rightsquigarrow t'$ and $u \rightsquigarrow u'$ are one-step reductions. By assumption, $t' \in \mathcal{R} \to \mathcal{S}$, which means that $t'u \in \mathcal{S}$. For the other case, we induct on $\nu(u)$, which is finite. $u' \in \mathcal{R}$ by (CR2), and $\nu(u') < \nu(u)$, which implies, by induction, that $tu' \in \mathcal{S}$. Therefore, by (CR3) applied to $\mathcal{S}$, $tu \in \mathcal{S}$, and so $t \in \mathcal{R} \to \mathcal{S}$. □

The following proposition lays out the key definition. Its proof can be skipped (especially on first reading), but is included in case anyone is suspicious of it.

Proposition 2.4. *Let T be a type, and suppose the sequence $\underset{\sim}{X} = X_1, X_2, \dots$ is assumed to contain all free (type) variables of T. With a sequence $\underset{\sim}{U}$ of types, we may define a type $T[\underset{\sim}{U}/\underset{\sim}{X}]$ by simultaneous substitution. Let $\underset{\sim}{\mathcal{R}}$ be a sequence of reducibility candidates, with $\mathcal{R}_i$ of type U_i. Then we can define a set $RED_T[\underset{\sim}{\mathcal{R}}/\underset{\sim}{X}]$ of terms of type $T[\underset{\sim}{U}/\underset{\sim}{X}]$ inductively by the following.*

- *If $T = X_i$ then $RED_T[\underset{\sim}{\mathcal{R}}/\underset{\sim}{X}] = \mathcal{R}_i$.*
- *If $T = V \to W$, then $RED_T[\underset{\sim}{\mathcal{R}}/\underset{\sim}{X}] = RED_V[\underset{\sim}{\mathcal{R}}/\underset{\sim}{X}] \to RED_W[\underset{\sim}{\mathcal{R}}/\underset{\sim}{X}]$.*
- *If $T = \Pi Y.W$, then $RED_T[\underset{\sim}{\mathcal{R}}/\underset{\sim}{X}]$ is the set of terms t of type $T[\underset{\sim}{U}/\underset{\sim}{X}]$ such that, for every type V and reducibility candidate $\mathcal{S}$ of this type, $tV \in RED_W[\underset{\sim}{\mathcal{R}}/\underset{\sim}{X}, \mathcal{S}/Y]$.*

Proof. As we will see later, this definition is remarkably circular as $\mathrm{RED}_T[\underset{\sim}{\mathcal{R}}/\underset{\sim}{X}]$ is itself a reducibility candidate. As such, we make the definition extra precise. We conceive of this definition as a function, assigning to a type T, and substitution as defined, a set of terms of type $T[\underset{\sim}{U}/\underset{\sim}{X}]$. Note that, entirely separate from this definition, we have for each type U a family $\mathcal{C}_U$ of reducibility candidates of this type: this is defined by comprehension on definition 2.1. The complexity $c(T)$ of a type T is defined in the obvious way, counting the number of Λ or $\to$ symbols.

We seek to define a function for each type T, assigning a valid substitution (one including all free variables) to the set $\mathrm{RED}_T[\underset{\sim}{\mathcal{R}}/\underset{\sim}{X}]$. In excruiciating detail, let $\mathcal{X}$ be the set of all type variables, and Sub the set of partial functions:

$$\mathcal{X} \to \coprod_{U \in \mathcal{U}} \mathcal{C}_U$$

with finite domain. Then for a given type T the domain $\Delta(T)$ of our function is the subset:

$$\Delta(T) = \{\eta \in \mathrm{Sub} \mid \mathrm{dom}(\eta) \supset \mathrm{FV}_{\mathrm{type}}(T)\}$$

Let Σ be the set of System F terms. To induct, we need to prove that for any $n \in \mathbb{N}$, if we are given the set:

$$\{\mathrm{RED}_T : \Delta(T) \to \mathcal{P}\Sigma \ \mid c(T) < n\} \tag{1}$$

then there is a unique choice of set:

$$\{\mathrm{RED}_T : \Delta(T) \to \mathcal{P}\Sigma \ \mid c(T) < n+1\}$$

corresponding to the above definition. From this perspective, the fact that $\mathrm{RED}_T[\underset{\sim}{\mathcal{R}}/\underset{\sim}{X}]$ is a set of terms of a particular type has been glossed over, so this must be part of the induction.

By the disjoint union, each η determines an assignment $\mathcal{X} \to \mathcal{U}$, for each free varibale. Abusing our notation slightly we denote $T[\eta] = T[\underline{U}/\underline{X}]$, where $\eta(X_i)$ is a reducibility candidate of type U_i.

For $n = 0$, T must be a variable X, so we assign $\mathrm{RED}_T[\eta] = \eta(X)$. If $\eta(X)$ is a reducibility candidate of type U, then $\mathrm{RED}_T[\eta]$ is a set of terms of type $U = T[\eta]$.

For $n > 0$, T is either an arrow or universal abstraction. If $T = V \to W$ then for any given η we may construct the set as usual, noting that the free type variables of V and W are each at most those of T. Also, by the inductive hypothesis, the members of $\mathrm{RED}_T[\eta]$ will terms of type:

$$V[\eta] \to W[\eta] = T[\eta]$$

Finally, suppose $T = \Pi Y.W$. For any $\eta : \Delta(W) \to \coprod_{U \in \mathcal{U}} \mathcal{C}_U$ and reducibility candidate $\mathcal{S}$ of type V, define:

$$(\eta + \mathcal{S}/Y)(X) = \begin{cases} \mathcal{S} & X = Y \\ \eta(X) & \text{else} \end{cases}$$

Then we define $\mathrm{RED}_T[\eta]$ to be the set of terms of type $\Pi Y.W[\eta]$, such that for any type V and reducibility candidate $\mathcal{S}$ of that type, $tV \in \mathrm{RED}_W[\eta + \mathcal{S}/Y]$.

This constructs $\mathrm{RED}_T[\eta]$ for any T of complexity n, and $\eta \in \Delta(T)$, so by induction our construction uniquely determines the sets as claimed. □

Remark 2.5. Observe that the notation $\mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}]$ does not explicitly include the substitutions U_i/X_i, which are nonetheless necessary to choose the right $\mathcal{R}_i$ (see [Gal90] p.38).

Example 2.6. If $T = \Pi X.X \to X$, then (with $\underline{X}$ empty), $\mathrm{RED}_T[-]$ is the set of terms t with type T, such that for every type V and reducibility candidate $\mathcal{S}$:

$$tV \in \mathrm{RED}_{X \to X}[\mathcal{S}/X] = \mathcal{S} \to \mathcal{S}.$$

We need a couple of facts about these sets.

Lemma 2.7. *$RED_T[\underline{\mathcal{R}}/\underline{X}]$ is a reducibility candidate of type $T[\underline{U}/\underline{X}]$.*

Proof. By induction on T. The only case we need verify is $T = \Pi Y.W$.

(CR1) Let some $t \in \mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}]$ be given. With an arbitrary type V, and arbitrary reducibility candidate $\mathcal{S}$, tV is strongly normalising, by inductively applying (CR1) to $\mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}, \mathcal{S}/Y]$. As $\nu(t) \leqslant \nu(tV)$, t is also strongly normalising.

(CR2) If $t \rightsquigarrow t'$, then for any type V, $tV \rightsquigarrow t'V$. Given a reducibility candidate $\mathcal{S}$ of this type, by induction:

$$t'V \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}, \mathcal{S}/Y]$$

so $t' \in \mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}]$.

(CR3) Suppose that t is neutral, and every term t' one step from t belongs to $\mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}]$. Then for any type V, the only one-step reductions of tV are of the form $t'V$, as t is neutral. Since $t' \in \mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}]$, $t'V \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}, \mathcal{S}/Y]$ for every candidate $\mathcal{S}$. By (CR3), this means $t \in \mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}]$. □

Lemma 2.8. *$RED_{T[V/Y]}[\underline{\mathcal{R}}/\underline{X}] = RED_T[\underline{\mathcal{R}}/\underline{X}, RED_V[\underline{\mathcal{R}}/\underline{X}]/Y]$*

Proof. Again, induction on T. First, if T is a variable, then $T = X_i$ or $T = Y$. In the first case, $T[V/Y] = T$, and both sides are $\mathcal{R}_i$ by definition. In the latter case, both sides are $\mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}]$.

If $T = W_1 \to W_2$, then the left-hand side is:

$$\mathrm{RED}_{W_1[V/Y] \to W_2[V/Y]}[\underline{\mathcal{R}}/\underline{X}] = \mathrm{RED}_{W_1[V/Y]}[\underline{\mathcal{R}}/\underline{X}] \to \mathrm{RED}_{W_2[V/Y]}[\underline{\mathcal{R}}/\underline{X}]$$

The right-hand side is

$$\mathrm{RED}_{W_1}[\underline{\mathcal{R}}/\underline{X}, \mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}]/Y] \to \mathrm{RED}_{W_2}[\underline{\mathcal{R}}/\underline{X}, \mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}]/Y]$$

By induction $\mathrm{RED}_{W_i[V/Y]}[\underline{\mathcal{R}}/\underline{X}] = \mathrm{RED}_{W_i}[\underline{\mathcal{R}}/\underline{X}, \mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}]/Y]$, for $i = 1, 2$, so the two expressions agree.

Finally, let $T = \Pi Z.W$. Then $T[V/Y] = \Pi Z.(W[V/Y])$, so the equality is clear by applying the inductive hypothesis to W. □

3 Strong Normalisation

We can now state the statement we will prove; general strong normalisation will pop out as a corollary.

Theorem 3.1. *Let t be any term of type T, with free variables among $x_1, \ldots, x_n$, of types $U_1, \ldots, U_n$. Suppose also that the free type variables of $T, U_1, \ldots, U_n$ are among $X_1, \ldots, X_m$. Let $\mathcal{R}_1, \ldots, \mathcal{R}_m$ be reducibility candidates of types $V_1, \ldots, V_m$, and $u_1, \ldots, u_n$ terms of types $U_1[\underline{V}/\underline{X}], \ldots, U_n[\underline{V}/\underline{X}]$, each in $RED_{U_1}[\underline{\mathcal{R}}/\underline{X}], \ldots, RED_{U_n}[\underline{\mathcal{R}}/\underline{X}]$. Then:*

$$t[\underline{V}/\underline{X}][\underline{u}/\underline{x}] \in RED_T[\underline{\mathcal{R}}/\underline{X}]$$

The variable and application cases are (by Girard's standards) fairly straightfoward, and for the other three the facts we need are the following.

Lemma 3.2 (λ-abstraction)**.** *If for every $v \in RED_V[\underline{\mathcal{R}}/\underline{X}]$ the term $w[v/y] \in RED_W[\underline{\mathcal{R}}/\underline{X}]$, then $\lambda y^V.w \in RED_{V \to W}[\underline{\mathcal{R}}/\underline{X}]$.*

Proof. We need to show that $(\lambda y^V.w)v \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}]$ for every $v \in \mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}]$. Let such a v be given. Noting that by assumption (with $v = y$), w is strongly normalising, we induct on $\nu(v) + \nu(w)$. Considering one-step reductions from $(\lambda y^V.w)v$, there are three cases. In each, they belong to $\mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}]$.

- $(\lambda y^V.w)v'$ with $v \rightsquigarrow v'$ in one step. Then $\nu(v') < \nu(v)$.
- $(\lambda y^V.w')v$ with $w \rightsquigarrow w'$ in one step. Then $\nu(w') < \nu(w)$.
- $w[v/y] \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}]$ by assumption.

As we are dealing with an application, (CR3) implies that $(\lambda y^V.w)v \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}]$, which implies the result. □

Lemma 3.3 (Universal application)**.** *If $t \in RED_{\Pi Y.W}[\underline{\mathcal{R}}/\underline{X}]$, then $tV \in RED_{W[V/Y]}[\underline{\mathcal{R}}/\underline{X}]$.*

Proof. By assumption, for any reducibility candidate $\mathcal{S}$ of type V, $tV \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}, \mathcal{S}/Y]$. Taking $\mathcal{S} = \mathrm{RED}_V[\underline{\mathcal{R}}/\underline{X}]$ and using Lemma 2.8 the result is immediate. □

Lemma 3.4 (Universal abstraction)**.** *If for every type V and candidate $\mathcal{S}$ of that type, $w[V/Y] \in RED_W[\underline{\mathcal{R}}/\underline{X}, \mathcal{S}/Y]$, then $\Lambda Y.w \in RED_{\Pi Y.W}[\underline{\mathcal{R}}/\underline{X}]$.*

Proof. Given a type V and candidate $\mathcal{S}$, we must show that $(\Lambda Y.w)V \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}, \mathcal{S}/Y]$. This is entirely analogous to the λ-abstraction case, now we induct on $\nu(w)$. Converting a redex in $(\Lambda Y.w)V$ gives two cases:

- $(\Lambda Y.w)V \rightsquigarrow (\Lambda Y.w')V$, where $\nu(w') < \nu(w)$.
- $(\Lambda Y.w)V \rightsquigarrow w[V/Y] \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}, \mathcal{S}/Y]$ by assumption.

Applying (CR3) and the definition of $\mathrm{RED}_{\Pi Y.W}[\underline{\mathcal{R}}/\underline{X}]$ the result follows. □

Right then, in we jump.

Proof of Theorem 3.1. We induct on the construction of t. If t is a variable, say x_i, then $T[\underline{V}/\underline{X}] = U_i[\underline{V}/\underline{X}]$, and $t[\underline{V}/\underline{X}][\underline{u}/\underline{x}] = u_i \in \mathrm{RED}_{U_i}[\underline{\mathcal{R}}/\underline{X}] = \mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}]$.

If $t = vw$, then both $v[\underline{V}/\underline{X}][\underline{u}/\underline{x}]$ and $w[\underline{V}/\underline{X}][\underline{u}/\underline{x}]$ belong to the appropriate set by induction. By definition, this implies that:

$$v[\underline{V}/\underline{X}][\underline{u}/\underline{x}](w[\underline{V}/\underline{X}][\underline{u}/\underline{x}]) = t[\underline{V}/\underline{X}][\underline{u}/\underline{x}]$$

belongs to $\mathrm{RED}_T[\underline{\mathcal{R}}/\underline{X}]$.

Let $t = \lambda y^V.w$ of type $V \to W$. By the inductive hypothesis

$$w[\underline{V}/\underline{X}][\underline{u}/\underline{x}, v/y] \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}]$$

for every v of type $V[\underline{V}/\underline{X}]$. Then by Lemma 3.2 we have that:

$$\lambda y^{V[\underline{V}/\underline{X}]}.w[\underline{V}/\underline{X}][\underline{u}/\underline{x}] = t[\underline{V}/\underline{X}][\underline{u}/\underline{x}]$$

belongs to our reducible set.

If $t = t'V$, with t' of type $\Pi Y.T'$, making $T = T'[V/Y]$. By the inductive hypothesis,

$$t'[\underline{V}/\underline{X}][\underline{u}/\underline{x}] \in \mathrm{RED}_{\Pi Y.T'}[\underline{\mathcal{R}}/\underline{X}]$$

Applying Lemma 3.3 implies the result.

The final case is $t = \Lambda Y.w$. Again, using the inductive hypothesis, for any type V and reducibility candidate $\mathcal{S}$ of this type:

$$w[\underline{V}/\underline{X}, V/Y][\underline{u}/\underline{x}] \in \mathrm{RED}_W[\underline{\mathcal{R}}/\underline{X}, \mathcal{S}/Y]$$

We apply Lemma 3.4 which implies the result. □

Corollary 3.5. *Every term of System F is strongly normalising.*

Proof. Apply the above, with $V_i = X_i$ and $u_j = x_j$, making each substitution the identity. Any sequence $\mathcal{R}_i$ of reducibility candidates works, for example the sets $\mathcal{SN}_i$ of strongly normalising terms of type X_i. Then (CR1) implies that every term is strongly normalising. □

References

[Gal90] Jean Gallier. On girards candidats de reductibilite. 01 1990.

[GLT93] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and types.* Cambridge Univ. Press, 1993.

[Sel08] Peter Selinger. Lecture notes on the lambda calculus. *CoRR*, abs/0804.3434, 2008.