7 A Typed Version of The Candidates of Reducibility

We now describe a typed version of the “candidats de reductibilité”, as originally defined by Girard (Girard 1970 [9], Girard 1972 [10]). We will present two sets of conditions for the typed candidates. The first set consists of conditions similar to those used by Stenlund [35], (basically the typed version of Tait’s conditions, Tait 1973 [37]), and the second set consists of Girard’s original conditions (Girard [10], [11]). We also compare these conditions, and prove that Girard’s conditions are stronger than Tait’s conditions.

Rather than using explicitly typed polymorphic terms as in Girard [10], we work with provable typing judgments.

Definition 7.1 For every type σ, let $\mathcal{PT}_\sigma$ be the set of all provable typing judgments of type σ (the provable typing judgments of the form $\Delta \triangleright M\colon \sigma$ for arbitrary Δ and M).

In order to reduce the amount of notation, if S is a set of provable typing judgments of type σ, rather than writing $\Delta \triangleright M\colon \sigma \in S$, we will write $\Delta \triangleright M \in S$.

Given any two types $\sigma, \tau \in \mathcal{T}$ and any two sets $S \subseteq \mathcal{PT}_\sigma$ and $T \subseteq \mathcal{PT}_\tau$, we let $[S \rightarrow T]$ be the subset of $\mathcal{PT}_{\sigma\rightarrow\tau}$ defined as follows:

$$[S \rightarrow T] = \{\Delta \triangleright M \in \mathcal{PT}_{\sigma\rightarrow\tau} \mid \forall \Delta' \triangleright N,\ \text{if } \Delta \subseteq \Delta' \text{ and } \Delta' \triangleright N \in S,\ \text{then } \Delta' \triangleright MN \in T\}.$$

We also refer to the operation $\rightarrow$ on sets of provable typing judgments defined above as the *function space constructor*.

Definition 7.2 Let $\mathcal{C} = (\mathcal{C}_\sigma)_{\sigma\in\mathcal{T}}$ be a $\mathcal{T}$-indexed family where each $\mathcal{C}_\sigma$ is a nonempty set of subsets of $\mathcal{PT}_\sigma$, and the following properties hold:

(1) For every $\sigma \in \mathcal{T}$, every $C \in \mathcal{C}_\sigma$ is a nonempty subset of $\mathcal{PT}_\sigma$.

(2) For every $\sigma, \tau \in \mathcal{T}$, for every $C \in \mathcal{C}_\sigma$ and every $D \in \mathcal{C}_\tau$, we have $[C \rightarrow D] \in \mathcal{C}_{\sigma\rightarrow\tau}$.

(3) For every $\forall t.\, \sigma$, $\tau \in \mathcal{T}$, for every family $(A_{\tau,C})_{\tau\in\mathcal{T}, C\in\mathcal{C}_\tau}$, where each set $A_{\tau,C}$ is in $\mathcal{C}_{\sigma[\tau/t]}$, we have

$$\{\Delta \triangleright M \in \mathcal{PT}_{\forall t.\,\sigma} \mid \forall \tau \in \mathcal{T},\ \Delta \triangleright M\tau \in \bigcap_{C\in\mathcal{C}_\tau} A_{\tau,C}\} \in \mathcal{C}_{\forall t.\,\sigma}.$$

A family satisfying the above conditions is called a *$\mathcal{T}$-closed family*.

Definition 7.3 Let $\mathcal{C}$ be a $\mathcal{T}$-closed family. A pair $\langle \theta, \eta \rangle$ where $\theta \colon \mathcal{V} \to \mathcal{T}$ is a type substitution and $\eta \colon \mathcal{B} \cup \mathcal{V} \to \bigcup \mathcal{C}$ is a *candidate assignment* iff $\eta(t) \in \mathcal{C}_{\theta(t)}$ for every $t \in \mathcal{V}$ and $\eta(\sigma) \in \mathcal{C}_\sigma$ for every $\sigma \in \mathcal{B}$.

We can associate certain sets of provable typing judgments to the types inductively as explained below.

Definition 7.4 Given any candidate assignment $\langle \theta, \eta \rangle$, for every type σ, the set $[\![\sigma]\!]\theta\eta$ is a subset of $\mathcal{PT}_{\theta(\sigma)}$ defined as follows:

$$\begin{aligned}
[\![t]\!]\theta\eta &= \eta(t), \text{ whenever } t \in \mathcal{B} \cup \mathcal{V}, \\
[\![(\sigma \to \tau)]\!]\theta\eta &= [[\![\sigma]\!]\theta\eta \to [\![\tau]\!]\theta\eta], \\
[\![\forall t.\, \sigma]\!]\theta\eta &= \{\Delta \rhd M \in \mathcal{PT}_{\theta(\forall t.\, \sigma)} \mid \forall \tau \in \mathcal{T}, \\
&\qquad \Delta \rhd M\tau \in \bigcap_{C \in \mathcal{C}_\tau} [\![\sigma]\!]\theta[t := \tau]\eta[t := C]\}.
\end{aligned}$$

Strictly speaking, $[\![\sigma]\!]\theta\eta$ is defined for the $\equiv_\alpha$-class $[\sigma]$ of σ. Thus, it can be assumed that σ is safe for θ. For a type $\forall t.\, \sigma$, this implies that $t \notin \mathcal{FV}(\theta(v))$ for every $v \in dom(\theta)$, and consequently that $\theta[t := \tau](\sigma) = \theta(\sigma)[\tau/t]$. This shows that the sets of terms involved in the intersection are indeed of the right type $\theta(\sigma)[\tau/t]$.

The following technical lemma will be useful later.

Lemma 7.5 Given any two candidate assignments $\langle \theta_1, \eta_1 \rangle$ and $\langle \theta_2, \eta_2 \rangle$, for every type σ, if θ_1, θ_2 agree on $\mathcal{FV}(\sigma)$, and η_1, η_2 agree on $\mathcal{FV}(\sigma)$ and $\mathcal{B}$, then $[\![\sigma]\!]\theta_1\eta_1 = [\![\sigma]\!]\theta_2\eta_2$.

Proof. Easy induction on the structure of types. □

We now have a typed version of "Girard's trick".

Lemma 7.6 (Girard) If $\mathcal{C}$ is a $\mathcal{T}$-closed family, for every candidate assignment $\langle \theta, \eta \rangle$, for every type σ, then $[\![\sigma]\!]\theta\eta \in \mathcal{C}_{\theta(\sigma)}$.

Proof. The lemma is proved by induction on the structure of types. The proof is similar to the proof of lemma 6.5. The only case worth mentioning is the case of a universal type. Given a type $\forall t.\, \sigma$ and a candidate assignment $\langle \theta, \eta \rangle$, using α-renaming, it can be assumed that $\forall t.\, \sigma$ is safe for θ. By the induction hypothesis, $[\![\sigma]\!]\theta\eta \in \mathcal{C}_{\theta(\sigma)}$. Thus, for every $\tau \in \mathcal{T}$ and for every $C \in \mathcal{C}_{\theta[t:=\tau](\sigma)}$, we also have $[\![\sigma]\!]\theta[t := \tau]\eta[t := C] \in \mathcal{C}_{\theta[t:=\tau](\sigma)}$. However, $\theta[t := \tau](\sigma) = \theta(\sigma)[\tau/t]$ as we observed earlier since $\forall t.\, \sigma$ is safe for θ. Thus, $[\![\sigma]\!]\theta[t := \tau]\eta[t := C] \in \mathcal{C}_{\theta(\sigma)[\tau/t]}$, and we conclude by condition (3) of $\mathcal{T}$-closed families. □

The following technical lemma will be needed later.

Lemma 7.7 Given any two types σ, τ, for every candidate assignment $\langle \theta, \eta \rangle$,

$$[\![\sigma[\tau/t]]\!]\theta\eta = [\![\sigma]\!]\theta[t := \theta(\tau)]\eta[t := [\![\tau]\!]\theta\eta].$$

Proof. Straightforward induction on the structure of σ. □

In order to use lemma 7.6 in proving properties of polymorphic lambda calculi, we need to define $\mathcal{T}$-closed families satisfying some additional properties.

Definition 7.8 We say that a $\mathcal{T}$-indexed family $\mathcal{C}$ is a *family of sets of candidates of reducibility* iff it is $\mathcal{T}$-closed and satisfies the conditions listed below:[14]

R0. Whenever $\Delta \rhd M \in C$ and $\Delta \subseteq \Delta'$, then $\Delta' \rhd M \in C$.

R1. For every $\sigma \in \mathcal{T}$, for every set $C \in \mathcal{C}_\sigma$, $\Delta \rhd x \in C$, for every $x{:}\,\sigma \in \Delta$,
For every $\sigma = Type(f)$, for every set $C \in \mathcal{C}_\sigma$, $\Delta \rhd f \in C$, for every $f \in \Sigma$.

R2. (i) For all $\sigma, \tau \in \mathcal{T}$, for every $C \in \mathcal{C}_\tau$, for all Δ, Δ', if

$$\begin{aligned}&\Delta \rhd M \in \bigcup \mathcal{C}_\tau,\\ &\Delta' \rhd N \in \bigcup \mathcal{C}_\sigma, \text{ and}\\ &\Delta' \rhd M[N/x] \in C, \text{ then}\\ &\Delta' \rhd (\lambda x{:}\,\sigma.\ M)N \in C.\end{aligned}$$

(ii) For all $\sigma, \tau \in \mathcal{T}$, for every $C \in \mathcal{C}_{\sigma[\tau/t]}$, if

$$\begin{aligned}&\Delta \rhd M \in \bigcup \mathcal{C}_\sigma \text{ and}\\ &\Delta \rhd M[\tau/t] \in C, \text{ then}\\ &\Delta \rhd (\Lambda t.\ M)\tau \in C.\end{aligned}$$

As in the untyped case, (R0), (R1), and (R2), are all we need to prove the following fundamental result.

Lemma 7.9 (Girard) Let $\mathcal{C} = (\mathcal{C}_\sigma)_{\sigma \in \mathcal{T}}$ be a $\mathcal{T}$-indexed family of sets of candidates of reducibility. For every $\Gamma \rhd M \in \mathcal{PT}_\sigma$, for every candidate assignment $\langle \theta, \eta \rangle$, for every substitution $\varphi{:}\, \Gamma \to \Delta$, if $\theta(\Delta) \rhd \varphi(x) \in [\![\Gamma(x)]\!]\theta\eta$ for $x \in FV(M)$, then $\theta(\Delta) \rhd \varphi(\theta(M)) \in [\![\sigma]\!]\theta\eta$.

Proof. It is similar to the proof of lemma 6.8 and proceeds by induction on the depth of the proof tree for $\Gamma \rhd M{:}\, \sigma$. The only cases worth considering are type abstraction and type

[14] Again, we also have to assume that every $C \in \mathcal{C}$ is closed under α-equivalence.

application, the other two being essentially unchanged, except that (R0) is needed in the case of λ-abstraction.

Case 3.

$$\frac{\Gamma \triangleright M\colon \forall t.\, \sigma}{\Gamma \triangleright M\tau\colon \sigma[\tau/t]} \qquad (\textit{type application})$$

First, by suitable α-renaming, it can be assumed that $M\tau$ is safe for φ and θ, and that $\forall t.\, \sigma$ is safe for θ. By the induction hypothesis, $\theta(\Delta) \triangleright \varphi(\theta(M)) \in [\![\forall t.\, \sigma]\!]\theta\eta$. By the definition of $[\![\forall t.\, \sigma]\!]\theta\eta$, we have

$$\theta(\Delta) \triangleright \varphi(\theta(M))\delta \in [\![\sigma]\!]\theta[t := \delta]\eta[t := C],$$

for every $\delta \in \mathcal{T}$ and $C \in \mathcal{C}_\delta$. Since $\forall t.\, \sigma$ is safe for θ, as observed before, we have $\theta(\sigma)[\delta/t] = \theta[t := \delta](\sigma)$. Since by lemma 7.6, $[\![\tau]\!]\theta\eta \in \mathcal{C}_{\theta(\tau)}$, by setting $\delta = \theta(\tau)$ and $C = [\![\tau]\!]\theta\eta$, we have

$$\theta(\Delta) \triangleright \varphi(\theta(M))\theta(\tau) \in [\![\sigma]\!]\theta[t := \theta(\tau)]\eta[t := [\![\tau]\!]\theta\eta].$$

Since $\forall t.\sigma$ is safe for θ and φ and θ have disjoint domains, we have $\varphi(\theta(M))\theta(\tau) = \varphi(\theta(M\tau))$ and $\theta(\sigma)[\theta(\tau)/t] = \theta(\sigma[\tau/t])$, and so

$$\theta(\Delta) \triangleright \varphi(\theta(M\tau)) \in [\![\sigma]\!]\theta[t := \theta(\tau)]\eta[t := [\![\tau]\!]\theta\eta].$$

However, by lemma 7.7, $[\![\sigma[\tau/t]]\!]\theta\eta = [\![\sigma]\!]\theta[t := \theta(\tau)]\eta[t := [\![\tau]\!]\theta\eta]$, and so

$$\theta(\Delta) \triangleright \varphi(\theta(M\tau)) \in [\![\sigma[\tau/t]]\!]\theta\eta.$$

Case 4.

$$\frac{\Gamma \triangleright M\colon \sigma}{\Gamma \triangleright \Lambda t.\, M\colon \forall t.\, \sigma} \qquad (\textit{type abstraction})$$

where in this rule, $t \notin \mathcal{FV}(\Gamma(x))$ for every $x \in \mathit{dom}(\Gamma) \cap FV(M)$.

Let $k+1$ be the depth of the proof tree for $\Gamma \triangleright \Lambda t.\, M\colon \forall t.\, \sigma$. Since $t \notin \mathcal{FV}(\Gamma(x))$ for every $x \in \mathit{dom}(\Gamma) \cap FV(M)$, by lemma 7.5, we have $[\![\Gamma(x)]\!]\theta\eta = [\![\Gamma(x)]\!]\theta[t := \tau]\eta[t := C]$, for every $\tau \in \mathcal{T}$ and every $C \in \mathcal{C}_\tau$. By the induction hypothesis,

$$\theta(\Delta) \triangleright \varphi(\theta[t := \tau](M)) \in [\![\sigma]\!]\theta[t := \tau]\eta[t := C],$$

for every $\tau \in \mathcal{T}$ and every $C \in \mathcal{C}_\tau$. By suitable α-renaming, it can be assumed that $\Lambda t.\, M$ is safe for φ and θ, that $\forall t.\, \sigma$ is safe for θ, and that $t \notin \mathcal{FV}(\Delta(x))$ for every $x \in \mathit{dom}(\Delta)$. Then, $\theta[t := \tau](\Delta) = \theta(\Delta)$, and as observed before, $\theta[t := \tau](M) = \theta(M)[\tau/t]$, $\theta[t := \tau](\sigma) = \theta(\sigma)[\tau/t]$, and since φ and $\theta[t := \tau]$ have disjoint domains, we have

$$\theta(\Delta) \triangleright \varphi(\theta(M))[\tau/t] \in [\![\sigma]\!]\theta[t := \tau]\eta[t := C],$$

for every $\tau \in \mathcal{T}$ and every $C \in \mathcal{C}_\tau$. In particular, since $t \in \mathcal{T}$, by choosing $\tau = t$, we have $\theta(\Delta) \rhd \varphi(\theta(M)) \in [\![\sigma]\!]\theta\eta[t := C]$. Since by lemma 7.6, $[\![\sigma]\!]\theta\eta[t := C] \in \mathcal{C}_{\theta(\sigma)}$, we have $\varphi(\theta(M)) \in \bigcup \mathcal{C}_{\theta(\sigma)}$. Thus, by (R2)(ii), we have

$$\theta(\Delta) \rhd (\Lambda t.\, \varphi(\theta(M)))\tau \in [\![\sigma]\!]\theta[t := \tau]\eta[t := C],$$

that is

$$\theta(\Delta) \rhd \varphi(\theta(\Lambda t.\, M))\tau \in [\![\sigma]\!]\theta[t := \tau]\eta[t := C],$$

for every $\tau \in \mathcal{T}$ and every $C \in \mathcal{C}_\tau$. By definition 7.4, this means that

$$\theta(\Delta) \rhd \varphi(\theta(\Lambda t.\, M)) \in [\![\forall t.\, \sigma]\!]\theta\eta.$$

□

Remark: As in the remark just after lemma 6.8, it should be observed that lemma 7.9 still holds if condition (R2) of definition 7.8 is relaxed so that in (R2)(i), $\bigcup \mathcal{C}_\tau$ is replaced by $\mathcal{PT}_\tau$, $\bigcup \mathcal{C}_\sigma$ by $\mathcal{PT}_\sigma$, and in (R2)(ii), $\bigcup \mathcal{C}_\sigma$ is replaced by $\mathcal{PT}_\sigma$. The relaxed conditions will be called (R2′)(i) and and (R2′)(ii).

In order to prove that families of sets of candidates of reducibility exist, one needs conditions stronger that (R1) and (R2). First, we give conditions adapted from Tait and Mitchell.

8 Families of Sets of Saturated Sets

The definition of the saturated sets given in the untyped case (definition 6.9) is adapted as follows.

Definition 8.1 Let $S = (S_\sigma)_{\sigma \in \mathcal{T}}$ be a $\mathcal{T}$-indexed family such that each S_σ is a nonempty subset of $\mathcal{PT}_\sigma$. We say that S is *closed* iff for all $\sigma, \tau \in \mathcal{T}$, for every $x \in \mathcal{X}$, if $\Delta \rhd M \in \mathcal{PT}_{\sigma \to \tau}$ and $\Delta, x{:}\,\sigma \rhd Mx \in S_\tau$, then $\Delta \rhd M \in S_{\sigma \to \tau}$, and for every $t \in \mathcal{V}$, if $\Delta \rhd M \in \mathcal{PT}_{\forall t.\, \sigma}$ and $\Delta \rhd Mt \in S_\sigma$, then $\Delta \rhd M \in S_{\forall t.\, \sigma}$. The family $Sat(S) = (Sat(S)_\sigma)_{\sigma \in \mathcal{T}}$ of sets of *saturated subsets of* S is defined such that for every $\sigma \in \mathcal{T}$, $Sat(S)_\sigma$ consists of those subsets $C \subseteq S_\sigma$ such that the following conditions hold:[15]

S0. Whenever $\Delta \rhd M \in C$ and $\Delta \subseteq \Delta'$, then $\Delta' \rhd M \in C$.

[15] We also have to assume that every saturated subset of S is closed under α-equivalence.

S1. For every type $\sigma \in \mathcal{T}$, for every $C \in Sat(S)_\sigma$, for all $n \geq 0$, for all $N_1, \ldots, N_n \in \mathcal{T} \cup \mathcal{P}\Lambda$, for every $u \in \mathcal{X} \cup \Sigma$, if

$$\begin{aligned}
&\Delta \triangleright uN_1 \ldots N_n \in \mathcal{PT}_\sigma, \text{ and} \\
&\Delta \triangleright N_i \in S_{\xi_i} \text{ for some } \xi_i \text{ whenever } N_i \text{ is a term } (1 \leq i \leq n), \text{ then} \\
&\Delta \triangleright uN_1 \ldots N_n \in C.
\end{aligned}$$

S2. (i) For all $\sigma, \tau \in \mathcal{T}$, for every $C \in Sat(S)_\tau$, for all $n \geq 0$, for all $N_1, \ldots, N_n \in \mathcal{T} \cup \mathcal{P}\Lambda$, for all Δ, Δ', if

$$\begin{aligned}
&\Delta \triangleright M \in S_\xi \text{ for some } \xi, \\
&\Delta' \triangleright M[N/x]N_1 \ldots N_n \in C, \\
&\Delta' \triangleright N \in S_\sigma, \text{ and} \\
&\Delta' \triangleright N_i \in S_{\xi_i} \text{ for some } \xi_i \text{ whenever } N_i \text{ is a term } (1 \leq i \leq n), \text{ then} \\
&\Delta' \triangleright (\lambda x{:}\, \sigma.\, M)NN_1 \ldots N_n \in C.
\end{aligned}$$

(ii) For all $\sigma, \tau \in \mathcal{T}$, for every $C \in Sat(S)_\sigma$, for all $n \geq 0$, for all $N_1, \ldots, N_n \in \mathcal{T} \cup \mathcal{P}\Lambda$, if

$$\begin{aligned}
&\Delta \triangleright M[\tau/t]N_1 \ldots N_n \in C, \\
&\Delta \triangleright M \in S_\xi \text{ for some } \xi, \text{ and} \\
&\Delta \triangleright N_i \in S_{\xi_i} \text{ for some } \xi_i \text{ whenever } N_i \text{ is a term } (1 \leq i \leq n), \quad \text{then} \\
&\Delta \triangleright (\Lambda t.\, M)\tau N_1 \ldots N_n \in C.
\end{aligned}$$

We have the following typed version of lemma 6.10.

Lemma 8.2 (Girard, Tait, Mitchell) Let $S = (S_\sigma)_{\sigma \in \mathcal{T}}$ be a closed family where each S_σ is a nonempty subset of $\mathcal{PT}_\sigma$, let $\mathcal{C}$ be the $\mathcal{T}$-indexed family of sets of saturated subsets of S, and assume that $S_\sigma \in \mathcal{C}_\sigma$ for every $\sigma \in \mathcal{T}$ (i.e. S_σ is a saturated subset of itself). Then $\mathcal{C}$ is a family of sets of candidates of reducibility.

Proof. It is similar to the proof of lemma 6.10. One point worth mentioning is the necessity of allowing $N_1 \ldots N_n$ to be terms *or types*, in order to prove closure condition (3) of definition 7.2. $\square$

We now consider the conditions used by Girard in [10] and [11], and their relationship to Tait and Mitchell's conditions.

9 Families of Sets of Girard Sets

Girard's conditions basically assert that complete induction holds for certain simple terms w.r.t. $\longrightarrow_{\lambda^\forall}$.

Definition 9.1 A *simple term* is a term that is not an abstraction. Thus, a term M is simple iff it is either a variable x, a constant $f \in \Sigma$, an application MN, or a type application $M\tau$.

The idea behind this definition is that for a simple term M, for every term N, if $MN \longrightarrow_{\lambda^\forall} Q$, then either $M \longrightarrow_{\lambda^\forall} M'$ and $Q = M'N$, or $N \longrightarrow_{\lambda^\forall} N'$ and $Q = MN'$.

Definition 9.2 Let $S = (S_\sigma)_{\sigma\in\mathcal{T}}$ be a $\mathcal{T}$-indexed family such that each S_σ is a nonempty subset of $\mathcal{PT}_\sigma$. A subset C of S_σ is a *Girard set* of type σ iff the following conditions hold:[16]

CR0. Whenever $\Delta \triangleright M \in C$ and $\Delta \subseteq \Delta'$, then $\Delta' \triangleright M \in C$.

CR1. If $\Delta \triangleright M \in C$, then M is SN w.r.t $\longrightarrow_{\lambda^\forall}$;

CR2. If $\Delta \triangleright M \in C$ and $M \longrightarrow_{\lambda^\forall} N$, then $\Delta \triangleright N \in C$;

CR3. For every simple term $\Delta \triangleright M \in \mathcal{PT}_\sigma$, if $\Delta \triangleright N \in C$ for every N such that $M \longrightarrow_{\lambda^\forall} N$, then $\Delta \triangleright M \in C$.

Note that (CR3) implies that all simple irreducible terms are in C. We shall prove a lemma analogous to lemma 8.2 for families of sets of Girard subsets, but first, we establish a precise connection between Girard sets and saturated sets. We prove that conditions (CR1), (CR2) and (CR3) imply conditions (S1) and (S2).

Lemma 9.3 Let $S = (S_\sigma)_{\sigma\in\mathcal{T}}$ be a family where each S_σ is a Girard subset of $\mathcal{PT}_\sigma$. Every Girard subset of S is a saturated set, i.e., satisfies conditions (S1), (S2).

Proof. We first make the following observation. For every term M, it is clear that there are only finitely many terms N such that $M \longrightarrow_{\lambda^\forall} N$. Thus, for every SN term M, by König's lemma, the tree of reduction sequences from M is finite. Thus, for every SN term M, the depth of the reduction tree from M is well defined, and we denote it as $\delta(M)$.

We now prove (S1). For every $u \in \mathcal{X} \cup \Sigma$, assume that

$$\begin{aligned} &\Delta \triangleright uN_1 \ldots N_n \in \mathcal{PT}_\sigma, \text{ and} \\ &\Delta \triangleright N_i \in S_{\xi_i} \text{ for some } \xi_i \text{ whenever } N_i \text{ is a term } (1 \leq i \leq n). \end{aligned}$$

[16] We also have to assume that every Girard subset of S is closed under α-equivalence.

We prove by complete induction on $\delta = \delta(N_1) + \ldots + \delta(N_n)$ that $\Delta \triangleright uN_1 \ldots N_n \in C$. Since $uN_1 \ldots N_n$ is a simple term, we can do this by using (CR3).

The base case $\delta = 0$ holds, since then $uN_1 \ldots N_n$ is simple and irreducible. For the induction step, note that $uN_1 \ldots N_n \longrightarrow_{\lambda^\forall} Q$ implies that $Q = uN_1 \ldots N_i' \ldots N_n$, where $N_i \longrightarrow_{\lambda^\forall} N_i'$. Also, if $N_i \longrightarrow_{\lambda^\forall} N_i'$, by (CR2) we have $\Delta \triangleright N_i' \in S_{\xi_i}$, and since $\delta(N_i') < \delta(N_i)$, the induction hypothesis implies that $\Delta \triangleright uN_1 \ldots N_i' \ldots N_n \in C$. Using (CR3), we conclude that $\Delta \triangleright uN_1 \ldots N_n \in C$.

We also prove (S2). Assume that for some Δ, Δ',

$\Delta \triangleright M \in S_\xi$ for some ξ,

$\Delta' \triangleright M[N/x]N_1 \ldots N_n \in C$,

$\Delta' \triangleright N \in S_\sigma$, and

$\Delta' \triangleright N_i \in S_{\xi_i}$ for some ξ_i whenever N_i is a term $(1 \leq i \leq n)$.

We want to prove that $\Delta' \triangleright (\lambda x{:}\,\sigma.\ M)NN_1 \ldots N_n \in C$. Since $(\lambda x{:}\,\sigma.\ M)NN_1 \ldots N_n$ is a simple term, we can do this by using (CR3). Since the terms $M, N, N_1, \ldots, N_n$ are SN, we prove by complete induction on $\delta = \delta(M) + \delta(N) + \delta(N_1) + \ldots + \delta(N_n)$ that if $\Delta' \triangleright M[N/x]N_1 \ldots N_n \in C$ then $\Delta' \triangleright (\lambda x{:}\,\sigma.\ M)NN_1 \ldots N_n \in C$.

The base case $\delta = 0$ holds by (CR3), since then the only possible reduction is $(\lambda x{:}\,\sigma.\ M)NN_1 \ldots N_n \longrightarrow_{\lambda^\forall} Q$ where $Q = M[N/x]N_1 \ldots N_n$, but $\Delta' \triangleright Q \in C$ by hypothesis. Otherwise, we just prove that $\Delta' \triangleright Q \in C$ whenever $(\lambda x{:}\,\sigma.\ M)NN_1 \ldots N_n \longrightarrow_{\lambda^\forall} Q$. If $Q = M[N/x]N_1 \ldots N_n$, then we know that $\Delta' \triangleright M[N/x]N_1 \ldots N_n \in C$, by the hypothesis. Otherwise, either $Q = (\lambda x{:}\,\sigma.\ M')NN_1 \ldots N_n$ where $M \longrightarrow_{\lambda^\forall} M'$, or $Q = (\lambda x{:}\,\sigma.\ M)N'N_1 \ldots N_n$ where $N \longrightarrow_{\lambda^\forall} N'$, or $Q = (\lambda x{:}\,\sigma.\ M)NN_1 \ldots N_i' \ldots N_n$ where $N_i \longrightarrow_{\lambda^\forall} N_i'$, or $Q = M'NN_1 \ldots N_n$ where $M = M'x$ and where $x \notin FV(M')$.

In the last case, note that $Q = M'NN_1 \ldots N_n = (M'x)[N/x]N_1 \ldots N_n$ since $x \notin FV(M')$, and since $M = M'x$, we have $\Delta' \triangleright M[N/x]N_1 \ldots N_n \in C$, by the hypothesis.

In the other cases, by (CR2), we have $\Delta \triangleright M' \in S_\xi$, $\Delta' \triangleright N' \in S_\sigma$, and $\Delta' \triangleright N_i' \in S_{\xi_i}$. Using (CR2) (and a simple induction on the number of reduction steps when $N \longrightarrow_{\lambda^\forall} N'$), since $\Delta' \triangleright M[N/x]N_1 \ldots N_n \in C$, we also have $\Delta' \triangleright M'[N/x]N_1 \ldots N_n \in C$ when $M \longrightarrow_{\lambda^\forall} M'$, $\Delta' \triangleright M[N'/x]N_1 \ldots N_n \in C$ when $N \longrightarrow_{\lambda^\forall} N'$, and $\Delta' \triangleright M[N/x]N_1 \ldots N_i' \ldots N_n \in C$ when $N_i \longrightarrow_{\lambda^\forall} N_i'$ (Note that this step of the proof seems to have been overlooked in other published proofs, as pointed out to us by Pierre Louis Curien and Roberto Di Cosmo).

Since $\delta(M') < \delta(M)$, $\delta(N') < \delta(N)$, and $\delta(N_i') < \delta(N_i)$, the induction hypothesis implies that $\Delta' \triangleright (\lambda x{:}\,\sigma.\ M')NN_1 \ldots N_n \in C$, $\Delta' \triangleright (\lambda x{:}\,\sigma.\ M)N'N_1 \ldots N_n \in C$, and $\Delta' \triangleright$

$(\lambda x{:}\,\sigma.\, M)NN_1 \ldots N_i' \ldots N_n \in C$. Using (CR3), it follows that $\Delta' \triangleright (\lambda x{:}\,\sigma.\, M)NN_1 \ldots N_n \in C$.

The case where

$$\Delta \triangleright M[\tau/t]N_1 \ldots N_n \in C,$$
$$\Delta \triangleright M \in S_\xi \text{ for some } \xi, \text{ and}$$
$$\Delta \triangleright N_i \in S_{\xi_i} \text{ for some } \xi_i \text{ whenever } N_i \text{ is a term } (1 \le i \le n)$$

is handled similarly. We show that if $\Delta \triangleright M[\tau/t]N_1 \ldots N_n \in C$, then $\Delta \triangleright Q \in C$ whenever $\Delta \triangleright (\Lambda t.\, M)\tau N_1 \ldots N_n \longrightarrow_{\lambda^\forall} Q$. $\square$

We now prove the analogous to lemma 8.2 for Girard sets.

Lemma 9.4 Let $S = (S_\sigma)_{\sigma \in \mathcal{T}}$ be a closed family where each S_σ is a nonempty subset of $\mathcal{PT}_\sigma$, let $\mathcal{C}$ be the $\mathcal{T}$-indexed family of sets of Girard subsets of S, and assume that $S_\sigma \in \mathcal{C}_\sigma$ for every $\sigma \in \mathcal{T}$ (i.e. S_σ is a Girard subset of itself). Then $\mathcal{C}$ is a family of candidates of reducibility.

Proof. We need to check that $\mathcal{C}$ satisfies the conditions of definition 7.8. By lemma 9.3, the sets in each $\mathcal{C}_\sigma$ satisfy conditions (S1) and (S2), which obviously imply (R1) and (R2). It remains to show that $\mathcal{C}$ is $\mathcal{T}$-closed. We need to prove properties (1), (2), (3), of definition 7.2. This is done by induction on types.

Property (1) is an immediate consequence of (S1). We will verify condition (2), leaving (3) as an exercise.

Let C and D be any two Girard sets. We need to show that $[C \to D]$ is a Girard set. Assume that the type of the terms in $[C \to D]$ is $\sigma \to \tau$.

For every $\Delta \triangleright M \in [C \to D]$, since $\Delta, x{:}\,\sigma \triangleright x \in C$ by (S1), by the definition of $[C \to D]$, we have $\Delta, x{:}\,\sigma \triangleright Mx \in D$, and since S is closed, we have $\Delta \triangleright M \in S_{\sigma\to\tau}$. Thus, $[C \to D]$ is a subset of $S_{\sigma\to\tau}$. Since x is obviously SN, $\Delta, x{:}\,\sigma \triangleright Mx \in D$, and (CR1) holds for D because it is a Girard set, it follows that M is SN. Thus, $[C \to D]$ satisfies (CR1).

Assume that $M \longrightarrow_{\lambda^\forall} M'$, where $\Delta \triangleright M \in [C \to D]$. For every Δ' such that $\Delta \subseteq \Delta'$ and $\Delta' \triangleright N \in C$, we have $\Delta' \triangleright MN \in D$ and $MN \longrightarrow_{\lambda^\forall} M'N$. By (CR2) applied to D, we have $\Delta' \triangleright M'N \in D$. Thus, by the definition of $[C \to D]$, we have $\Delta \triangleright M' \in [C \to D]$.

It remains to verify (CR3). Let $\Delta \triangleright M$ be any simple term of type σ, and assume that $\Delta \triangleright Q \in [C \to D]$ whenever $M \longrightarrow_{\lambda^\forall} Q$. We want to prove that $\Delta \triangleright M \in [C \to D]$. By the definition of $[C \to D]$, this will be the case if we can show that for every Δ' such that $\Delta \subseteq \Delta'$ and $\Delta' \triangleright N \in C$, then $\Delta' \triangleright MN \in D$.

We prove by complete induction on $\delta(N)$ that $\Delta' \triangleright MN \in D$. Since MN is simple, we can do this by using (CR3). Because M is simple, observe that $MN \longrightarrow_{\lambda^\forall} U$ implies that either

(a) $U = M'N$ and $M \longrightarrow_{\lambda^\forall} M'$, or

(b) $U = MN'$ and $N \longrightarrow_{\lambda^\forall} N'$.

In case (a), since we have assumed that $\Delta \triangleright Q \in [C \to D]$ whenever $M \longrightarrow_{\lambda^\forall} Q$, we have $\Delta \triangleright M' \in [C \to D]$, and thus $\Delta' \triangleright M'N \in D$ since $\Delta' \triangleright N \in C$.

In case (b), since $\Delta' \triangleright N \in C$ and $N \longrightarrow_{\lambda^\forall} N'$, by (CR2) applied to C we have $\Delta' \triangleright N' \in C$, we also have $\delta(N') < \delta(N)$, and by the induction hypothesis, this yields $\Delta' \triangleright MN' \in D$. We can conclude that $\Delta' \triangleright MN \in D$ by application of (CR3) to D. Hence, we have shown that $[C \to D]$ also satisfies (CR3), and consequently it is a Girard set. □

Having shown that both closed families of saturated sets and closed families of Girard sets yield families of sets of candidates of reducibility, we can prove the typed version of Girard's fundamental theorem.

10 Girard's Fundamental Theorem

The fundamental theorem holds for both saturated and Girard sets.

Theorem 10.1 (Girard) Let $S = (S_\sigma)_{\sigma \in \mathcal{T}}$ be a closed family where each S_σ is a nonempty subset of $\mathcal{PT}_\sigma$. Let $\mathcal{C}$ be either the $\mathcal{T}$-indexed family of sets of saturated subsets of S, or the family of Girard subsets of S, and assume that $S_\sigma \in \mathcal{C}_\sigma$ for every $\sigma \in \mathcal{T}$. For every $\Delta \triangleright M \in \mathcal{PT}_\sigma$, we have $\Delta \triangleright M \in S_\sigma$.

Proof. By lemma 8.2 or lemma 9.4, $\mathcal{C}$ is a family of sets of candidates of reducibility. We now apply lemma 7.9 to any assignment (for example, the assignment with value $\eta(t) = S_t$), the identity type substitution, and the identity term substitution, which is legitimate since by (S1), every variable belongs to every saturated set.[17]

Remark: As in the untyped case, if we are interested in the version of the candidates using conditions (R2′)(i) and (R2′)(ii), then lemma 8.2 holds if conditions (S2)(i) and (S2)(ii) of definition 8.1 are relaxed in the obvious way (for example, S_ξ is replaced by $\mathcal{PT}_\xi$).

The next lemma is a typed version of lemma 6.16 and gives interesting examples of closed families of Girard sets and saturated sets.

[17] Actually, some α-renaming may have to be performed on M and σ so that they are both safe for the type and term identity substitution.

Lemma 10.2 (Girard, Tait, Mitchell) (i) The $\mathcal{T}$-indexed family SN_{β} such that for every $\sigma \in \mathcal{T}$, $SN_{\beta,\sigma}$ is the set of provable typing judgments $\Delta \rhd M{:}\,\sigma$ such that M is strongly normalizing under β-reduction, is a closed family of Girard and saturated sets. (ii) The $\mathcal{T}$-indexed family $SN_{\beta\eta}$ such that for every $\sigma \in \mathcal{T}$, $SN_{\beta\eta,\sigma}$ is the set of provable typing judgments $\Delta \rhd M{:}\,\sigma$ such that M is strongly normalizing under $\beta\eta$-reduction, is a closed family of Girard and saturated sets. (iii) The $\mathcal{T}$-indexed family consisting for every $\sigma \in \mathcal{T}$ of the set of provable typing judgments $\Delta \rhd M{:}\,\sigma$ such that confluence under β-reduction holds from M and all of its subterms, is a closed family of Girard and saturated sets. (iv) The $\mathcal{T}$-indexed family consisting for every $\sigma \in \mathcal{T}$ of the set of provable typing judgments $\Delta \rhd M{:}\,\sigma$ such that confluence under $\beta\eta$-reduction holds from M and all of its subterms, is a closed family of Girard and saturated sets.

Proof. (i)-(ii) The verification that SN_{β} and $SN_{\beta\eta}$ are closed families of Girard sets is obvious. The verification that they are closed families of saturated sets is essentially identical to the proof of lemma 6.16. Verifying (S2)(ii) in the case of type abstraction is similar to the other case (S2)(i) but a bit simpler, since types cannot be $\beta\eta$-reduced.

(iii)-(iv) The verification that these sets are Girard sets is similar to the verification that they are saturated sets and is omitted. The verification that they are saturated sets is done in appendix 2. $\square$

One should note that because the conditions for being a Girard set are stronger than the conditions for being a saturated set, it is trivial to show that $SN_{\beta\eta}$ is a closed family of Girard sets, whereas, showing that it is a closed family of saturated sets requires more work (namely, part of lemma 6.16). Applying theorem 10.1 to the set $SN_{\beta\eta}$, which, by lemma 9.4 is a closed family of Girard sets (or by lemma 8.2, a closed family of saturated sets), we obtain the following corollary to theorem 10.1.

Lemma 10.3 Every term M that type-checks is strongly normalizing under $\beta\eta$-reduction.

Interestingly, using parts (iii)-(iv) of lemma 10.2, we obtain a new proof of the fact that $\longrightarrow_{\lambda^{\forall}}$ is confluent on terms that type-check. Girard (Girard [10]) proved this result (for β-reduction) using an adaptation of Tait and Matin Löf's proof of confluence for the untyped lambda calculus.

Lemma 10.4 Confluence holds under $\beta\eta$-reduction for terms M that type-check. Confluence also holds under β-reduction for terms that type-check.

It is interesting to note that Lemma 10.4 **fails** for raw terms. The following example shows what goes wrong.

Consider the term $M = \lambda x{:}\, \sigma. (\lambda y{:}\, \tau. y)x$ where $\sigma \neq \tau$. Clearly, M does not type-check, and we have two reductions

$$\lambda x{:}\, \sigma.\, (\lambda y{:}\, \tau.\, y)x \longrightarrow_\beta \lambda x{:}\, \sigma.\, x \quad \text{and} \quad \lambda x{:}\, \sigma.\, (\lambda y{:}\, \tau.\, y)x \longrightarrow_\eta \lambda y{:}\, \tau.\, y,$$

and there is no way to achieve confluence since $\sigma \neq \tau$. Thus, $\longrightarrow_{\lambda^\forall}$ is not confluent on all raw terms, only those that type-check.

However, the polymorphic lambda calculus $\lambda^\forall$ is Church-Rosser and strongly normalizing on terms that type-check. What is interesting about this new proof of the Church-Rosser property is that it makes an essential use of the type structure. This observation was made by Statman in the context of logical relations [34] (for the simply typed lambda calculus).

11 A Comparison of Proofs

The purpose of this section is to compare various proofs that have appeared in the literature. These proofs are considered in chronological order.

1. Girard's proof(s) (1970, 1972).

The method of candidates of reducibility was invented by Girard in order to settle entirely proof theoretically a famous open problem in proof theory known as Takeuti's conjecture. Girard's "tour de force", settling positively Takeuti's conjecture for higher-order intuitionistic logic by purely proof theoretic means, is first accomplished in Girard [9]. Takeuti's conjecture is the generalization of Gentzen's cut elimination theorem to (classical) higher-order logic (for details on Takeuti's conjecture, the reader should consult Girard [12]). Girard's proof of Takeuti's conjecture (in [9]) consists in exploiting the "formulae as types" analogy, first observed by Curry and Howard. Roughly speaking, a *proof* (in a Prawitz-style deduction system) is coded as a certain kind of *lambda term*, and the *formula* occurring as the conclusion of the proof is considered to be the *type* of the term. What is remarkable about this correspondence proof – lambda term, formula – type, is that the process of normalizing a proof (eliminating certain redundancies having to do with an introduction rule followed by an elimination rule for the same logical connective) corresponds to β-reduction applied to the term representing the proof. Thus, if one succeeds in defining a typed lambda calculus in which terms represent proofs in a natural deduction system, and β-conversion corresponds to proof normalization, if in addition one is able to prove strong normalization for this typed lambda calculus, then one has shown strong normalization for proofs in the natural deduction system.

Girard's achievement was to define a typed lambda calculus, system F, which corresponds to second-order propositional intuitionistic logic, and to prove strong normalization for system F. In order to prove strong normalization, Girard invented the method of candidates of reducibility. Girard also used the method of candidates of reducibility to prove Takeuti's conjecture for higher-order intuitionistic logic.

System F is actually more general than the calculus that we have presented, since it includes product types and existentially quantified types. In [9], candidates of reducibility are sets of typed terms satisfying certain conditions technically rather different from conditions (R1) and (R2) given in definition 7.8. We will not list these conditions here, but instead present the conditions given in Girard's thesis [10] and his class notes [11] later in this section.

Reading [9] is quite challenging, because a lot of extremely original material is presented in a short space, and also because the notations used are not the most illuminating. Nevertheless, the method of the candidates emerges very clearly and with great power.

In his thesis [10], Girard extends system F to a typed lambda calculus named F_ω. The system F_ω encodes proofs in higher-order intuitionistic logic, whereas system F only encodes the second-order fragment of this logic. The system F_ω also includes product types, existential types, and even disjunctive types. The method of candidates of reducibility is extended to F_ω, and strong normalization is shown, as well as the Church-Rosser theorem (by the method of Tait and Martin Löf). Much more is done in the thesis, but we are primarily focusing on the method of candidates of reducibility. A simpler (and more readable) version of this proof for system F is given in Girard [11].

Both in [10] and [11], Girard uses a typed version of the candidates satisfying some interesting conditions. Girard defines a *simple term* as a term that is not an abstraction. Thus, a term M is simple iff it is either a variable x, an application MN, or a type application $M\tau$. A candidate of reducibility of type σ is a set C of terms of type σ such that:

CR1. If $M \in C$, then M is SN;

CR2. If $M \in C$ and $M \longrightarrow_{\lambda^\forall} N$, then $N \in C$;

CR3. If M is a simple term and if $N \in C$ for every N such that $M \longrightarrow_{\lambda^\forall} N$, then $M \in C$.

Note that (CR3) implies that all variables of type σ are in C (what we call (R1) in definition 7.8). Girard defines what he calls *reducibility with parameters*, as we do in definition 7.4, except that he uses a notation that we find a little confusing. Given a type σ, if $\mathcal{FV}(\sigma) = \{t_1, \ldots, t_n\}$ is the set of free type variables in σ, instead of our type substitution $\theta\colon \mathcal{V} \to \mathcal{T}$ he uses a sequence $U = \langle U_1, \ldots, U_n \rangle$ of types, and instead of our assignment $\eta\colon \mathcal{V} \to \bigcup \mathcal{C}$, he uses a sequence $C = \langle C_1, \ldots, C_n \rangle$ of candidates, each C_i being of type

U_i. Then, what we denote as $[\![\sigma]\!]\theta\eta$ is denoted by Girard as $RED(\sigma[C_1/t_1, \ldots, C_n/t_n])$. $RED(\sigma[C_1/t_1, \ldots, C_n/t_n])$ is a certain set of terms of type $\theta(\sigma)$, in Girard's notation of type $\sigma[U_1/t_1, \ldots, U_n/t_n]$.

One of the problems that we have with this notation is that it is difficult to distinguish between actual substitution, as in $\sigma[U_1/t_1, \ldots, U_n/t_n]$, and assigning candidates to the type variables, as in $RED(\sigma[C_1/t_1, \ldots, C_n/t_n])$. The notation $RED(\sigma[C_1/t_1, \ldots, C_n/t_n])$ is also slightly ambiguous since it does not refer to the type substitution $[U_1/t_1, \ldots, U_n/t_n]$, which is nevertheless indispensable to know where to pick the C_i's from. We prefer the notation $[\![\sigma]\!]\theta\eta$.

It is interesting to note that the intriguing condition (CR2) is needed to show that (CR3) holds for sets of the form $[C \to D]$, and to show that Girard's conditions are stronger than conditions (S1) and (S2). Also, Girard does not need (S2) (from definition 8.1) in his proof of lemma 7.9, because he uses (CR1), (CR2), (CR3) and the fact that if a term M is SN, then there is an upper bound on the length of reduction sequences from M, as discussed in section 9. In effect, Girard uses (CR1), (CR2), (CR3) as a substitute for what we call (R2) in definition 7.8, in his proof of lemma 7.9. As we showed in section 9, Girard's conditions are stronger than conditions (S1) and (S2). We also remark that formulating lemma 7.9 in Girard's notation is rather cumbersome.

2. Stenlund's version (1972).

In [35], Stenlund presents a version of the proof of strong normalization for second-order intuitionistic logic using the method of candidates of reducibility (the theory of species). His proof is basically a typed version of Tait's proof discussed next. Stenlund eliminates condition (i) on page 247 of Tait [37] because it is redundant, and uses essentially our (S1) and (S2) of definition 7.8. The sets $[\![\sigma]\!]\theta\eta$ are defined basically as we do in definition 7.4. Stenlund's notation is easier to follow that Tait (and Girard), but lemma 7.7 is also not mentioned. The fact that in conditions (S1) and (S2), the expressions $N_1, \ldots, N_n$ must be allowed to be types as well as terms seems to have been overlooked. Nevertheless, this proof is fairly readable.

3. Tait's version (1973).

In [37], Tait proves a realizability result analogous to our lemma 6.8 for second-order intuitionistic logic (what he calls the theory of species) using the method of candidates of reducibility. As a consequence, Tait obtains a version of the proof of strong normalization for second-order intuitionistic logic (this is slightly more general than strong normalization for second-order propositional intuitionistic logic, which corresponds to system F). Tait takes advantage of the erasing trick, and he defines the candidates as sets of untyped lambda terms. Actually, Tait's erasing function is not quite the one we use, because Tait uses conditions slightly different from our (S1), (S2) of definition 6.7. Tait assume that the untyped lambda calculus has a special constant K. Let SN denote the set of untyped lambda terms that are strongly normalizable. Then, a candidate of reducibility C is defined as a subset of SN satisfying the following properties:

(i) If $M \in C$ and $M \longrightarrow_{\lambda} N$, then $N \in C$;

(ii) For all $n \geq 0$ and all $M, N_1, \ldots, N_n \in \Lambda$, for every $N \in SN$, if $M[N/x]N_1 \ldots N_n \in C$, then $(\lambda x.\, M)NN_1 \ldots N_n \in C$.

(iii) For all $n \geq 0$ and all $N_1, \ldots, N_n \in SN$, $KN_1 \ldots N_n \in C$.

Condition (i) is Girard's (CR2), (ii) is basically our (S2), and (iii) is basically our (S1). Tait then proves Girard's trick (lemma 6.5) and lemma 6.8. He concludes by using the erasing trick that if M type-checks, then it is SN. As we see it, condition (i) is never used anywhere and appears to be redundant.

There are other technical differences with Girard's proof. First, Tait expands the language of types by adding a base type $\overline{C}$ for every candidate of reducibility C in $\mathcal{C}$. Then, Tait defines $[\![\sigma]\!]$ for all *closed* types over this extended language. There is no need for an explicit assignment η, since the new base types correspond to candidates in $\mathcal{C}$. The definition of $[\![\forall t.\, \sigma]\!]$ is worth noting:

$$[\![\forall t.\, \sigma]\!] = \{M \in \Lambda \mid \forall C \in \mathcal{C},\ MK \in [\![\sigma[\overline{C}/t]]\!]\},$$

where K is the special constant added to the untyped lambda calculus. Tait uses the following erasing function:

$Erase'(c) = c$, whenever $c \in \Sigma$,
$Erase'(x) = x$, whenever $x \in \mathcal{X}$,
$Erase'(MN) = Erase'(M)Erase'(N)$,
$Erase'(\lambda x{:}\sigma.\, M) = \lambda x.\, Erase'(M)$,
$Erase'(M\sigma) = Erase'(M)K$,
$Erase'(\Lambda t.\, M) = \lambda t.\, Erase'(M)$.

The difference between this erase function and ours is that we have $Erase(\Lambda t.\, M) = Erase(M)$, that is, the type abstraction is deleted, and we have $Erase(M\sigma) = Erase(M)$, whereas Tait uses the special constant K. Finally, in order to formulate and prove lemma 6.8, even though Tait was able to get away from using an explicit type assignment η, he is now forced to consider such assignments in the form of substitutions, which, in our opinion, is rather confusing. Given a raw term M that type-checks with proof $\Delta \triangleright M{:}\,\sigma$, if

$\mathcal{FV}(M) = \{t_1, \ldots, t_n\}$, Tait considers substitutions $[T_1/t_1, \ldots, T_n/t_n]$ of closed types for the free type variables in M, and forms $M_0 = M[T_1/t_1, \ldots, T_n/t_n]$ and $\sigma_0 = \sigma[T_1/t_1, \ldots, T_n/t_n]$. In effect, the substitution $[T_1/t_1, \ldots, T_n/t_n]$ plays the role of our assignment η.[18] Tait then proceeds basically as we do.

It should be noted that Tait does not actually justify the validity of the use of his erasing trick, and our lemma 6.6 is hidden in (2) on page 248 (however, Girard does prove explicitly a lemma analogous to our lemma 7.7). We also believe that case (3) in the proof of 4.2 on page 250 is erroneous, but it can be fixed easily (along the line of our proof).

4. Fortune, Leivant, O'Donnell's version (1983).

This version of the proof [8] can be considered as a typed version of Tait's proof for system F. Although it is not stated explicitly that the candidates are typed, this is the case since the conditions (G1), (G2), (G2'), (G3) on page 174 of [8] involve types and type abstraction. These conditions are basically our conditions (S1), (S2) of definition 8.1. As in Tait [37], the language of types is expanded by adding a base type $\overline{C}$ for every candidate of reducibility C in $\mathcal{C}$. The sets $[\![\sigma]\!]$ are only defined for *closed* types over this extended language (As far as we can see, definition 6.3.2 has no provision for assigning anything to the type variables). The definition of $[\![\forall t.\, \sigma]\!]$ is almost as in Tait:

$$[\![\forall t.\, \sigma]\!] = \{M \in \mathcal{P}\Lambda \mid \forall C \in \mathcal{C},\ M\overline{C} \in [\![\sigma[\overline{C}/t]]\!]\}.$$

However, we believe that there is a subtle problem with this clause of definition 6.3.2 (page 174) which invalidates the subsequent results. The problem is that in clause (S4) of definition 6.3.2, the sets $\{M \in \mathcal{P}\Lambda \mid \forall C \in \mathcal{C},\ M\overline{C} \in [\![\sigma[\overline{C}/t]]\!]\}$ are always **empty**, since the constants $\overline{C}$ do **not** belong to the original language, but yet the *grounds* (definition 6.3.1, page 173-174) are defined over the original language. Tait's argument does not suffer from this problem because $[\![\forall t.\, \sigma]\!]$ is a set of (untyped) terms over the original language ($\{M \in \Lambda \mid \forall C \in \mathcal{C},\ MK \in [\![\sigma[\overline{C}/t]]\!]\}$). Unfortunately, in Fortune, Leivant, O'Donnell, since $[\![\forall t.\, \sigma]\!] = \emptyset$, the subsequent lemmas are invalidated. Another minor problem arises from definition 6.4.1. Given a term M that type-checks, a *type instance* of M is a term M' obtained by substituting in M *base types* for the free type variables. This substitution essentially plays the role of our η. An *instance* of M is a substitution instance $\varphi(M')$ of M', where φ is a substitution of terms for the free variables (similar to our φ). But then, according to these definitions, it is not true that every term M is an instance of itself, because M' cannot contain any type variables. In particular, if M contains free type variables, since M' does not, $\varphi(M')$ will never be equal to M for any term substitution φ.

[18] It would be sufficient to consider substitutions of constants corresponding to the candidates.

Thus, as is, theorem 6.4.2 only holds for terms with no free type variables (but it is also false, due to the problem with definition 6.3.2). It seems difficult to fix the problem with definition 6.3.2 other than by using Girard's original definition of $[\![\forall t.\ \sigma]\!]$. Indeed, "easy" attempts to fix definition 6.3.2 seem to spoil the proof of lemma 6.3.1.

The next three versions are actually sketches of proofs, and they do not go through the various induction steps (and there are quite a few!).

5. Mitchell's version (1986).

In [24], Mitchell sketches a proof of strong normalization for system F, in which he introduces the erasing function *Erase* of definition 2.3, and an untyped version of the candidates of reducibility. Mitchell also introduces conditions (S1), (S2') (essentially our definition 6.9) and makes the clever observation that the range of applicability of the method can be broadened by relativizing the definition of saturated sets to a closed set S which is not necessarily SN (the set of lambda terms that are strongly normalizing). The definition of $[\![\sigma]\!]\eta$ given in definition 6.3 comes from lemma 4 of [24], and definition 6.9 is basically a reformulation of Mitchell's conditions. Mitchell states theorem 6.15, but does not state explicitly lemma 6.8, nor lemma 6.6.[19] Although lacking proofs, this paper is quite readable.

6. Huet's version (1987).

In [18], Huet sketches a proof of strong normalization for system F, using an untyped version of the candidates and the erasing trick, but using a special constant in condition (S1), as in Tait [37]. This proof was found by Coquand and Huet independently of Mitchell.[20] In our opinion, the role of the assignment η should be made more explicit in the definition of $[\![\sigma]\!]\eta$, and lemma 6.8 should be stated. Lemma 6.6 is not mentioned.

7. Scedrov's versions (1987, 1988).

In Scedrov [31], a very elegant and simple proof of normalization for $\lambda^{\forall}$ is presented. This proof is obtained by noticing two interesting facts. The first fact is that if one is simply interested in normalization (as opposed to strong normalization), then one can drop condition (S2) in the definition of the saturated sets, and instead require closure under β-conversion. Then, one can prove a version of lemma 6.8 stating that for every term M that type-checks, $Erase(M)$ is normalizable. The second fact already noted by Girard in his thesis ([10]), is that if $Erase(M) \xrightarrow{*}_{\lambda} Q$, then there is some term $P \in \mathcal{P}\Lambda$ such that $Erase(P) = Q$ and $M \xrightarrow{*}_{\lambda^{\forall}} P$. These two facts together yield normalization for all terms that type-check.

[19] However, Val Breazu-Tannen is in possession of some notes by Mitchell in which such a lemma is stated.

[20] Private communication from Thierry Coquand.

In [32], Scedrov gives an informal exposition of the proof of strong normalization for $\lambda^{\forall}$. His version is basically an expanded version of Mitchell's sketch, using the erasing trick. In our opinion, the role of the assignment η should be made more explicit in the definition of $[\![\sigma]\!]\eta$, and lemma 3.1 from [32] (our lemma 6.8) should be stated more clearly.

8. Related Proofs.

Other proofs of normalization or strong normalization for various natural deduction systems based on Girard's method have been published. Since they are not specifically about polymorphic lambda calculi, we will simply list them without further comments. Prawitz [29] proves strong normalization for classical first-order logic (natural deduction), and intuitionistic second-order logic (natural deduction). It interesting to observe that the notion of strong validity introduced in section 3.2 of Appendix A, and in section B.2 of Appendix B of [29], is essentially equivalent to the definition of the sets $[\![\sigma]\!]\eta$. Martin-Löf proves normalization results for various proof systems, including the Theory of Intuitionistic Iterated Inductive Definitions, Second-Order Intuitionistic Logic, and Intuitionistic Simple Type Theory [21, 22, 23]. The result in [23] is significantly strengthened in Girard [10]. We should also mention that Leivant [20] has given an interesting semantic generalization of Girard's technique. This allows him to prove various properties of terms in $\lambda^{\forall}$, including normalization, strong normalization, and solvability. Finally, in the case of the simply-typed lambda calculus, a radically different proof of strong normalization has been given by de Vrijer [40].