17 Appendix 1: Product Types in F_ω

In this section, we extend the system F_ω by allowing product types. An even more general system with disjunctive and existential types was investigated by Girard [10].

In the formula–as–type analogy, product types correspond to conjunctions, and the typing rules correspond to the introduction and elimination rules for conjunction. We will be dealing with product types with surjective pairing.

The definition of the kinds given in definition 12.1 remains unchanged. However, it is assumed that every set $\mathcal{TC}$ of type constructors contains the special symbols $\times$, $\Rightarrow$ and Π_K for every $K \in \mathcal{K}$. The type constructor $\times$ is the *product type* constructor. The type constructors are assigned kinds by a kind signature.

Definition 17.1 A *kind signature* is a function $\Xi\colon \mathcal{TC} \to \mathcal{K}$ assigning a kind to every type constructor in $\mathcal{TC}$, and such that $\Xi(\Rightarrow) = \star \to (\star \to \star)$, $\Xi(\times) = \star \to (\star \to \star)$, and $\Xi(\Pi_K) = (K \to \star) \to \star$.

The definition of raw types remains unchanged, but since $\times$ belongs to every set $\mathcal{TC}$ of type constructors, more raw types are allowed. The definition is repeated for the reader's convenience.

Definition 17.2 The set $\mathcal{T}$ of *raw type expressions* (for short, *raw types*) is defined inductively as follows:

$t \in \mathcal{T}$, whenever $t \in \mathcal{V}$,
$\sigma \in \mathcal{T}$, whenever $\sigma \in \mathcal{TC}$,
$(\lambda t\colon K.\, \sigma) \in \mathcal{T}$, whenever $t \in \mathcal{V}$, $\sigma \in \mathcal{T}$, and $K \in \mathcal{K}$, and
$(\sigma\tau) \in \mathcal{T}$, whenever $\sigma, \tau \in \mathcal{T}$.

Since $\times$ belong to $\mathcal{TC}$, by the last clause, $((\times\sigma)\tau)$ is a raw type for all $\sigma, \tau \in \mathcal{T}$, For simplicity of notation, $((\times\sigma)\tau)$ is denoted as $(\sigma \times \tau)$. The subset of $\mathcal{T}$ consisting of the raw types of kind $\star$ is the set of types that can actually be the types of terms.

Next, we define the polymorphic raw terms. There is no change in the definition of a type signature.

Definition 17.3 The set $\mathcal{P}\Lambda$ of *polymorphic lambda raw Σ-terms* (for short, *raw terms*) is defined inductively as follows:

$c \in \mathcal{P}\Lambda$, whenever $c \in \Sigma$,
$x \in \mathcal{P}\Lambda$, whenever $x \in \mathcal{X}$,

$(MN) \in \mathcal{P}\Lambda$, whenever $M, N \in \mathcal{P}\Lambda$,
$\langle M, N\rangle \in \mathcal{P}\Lambda$, whenever $M, N \in \mathcal{P}\Lambda$,
$\pi_1(M), \pi_2(M) \in \mathcal{P}\Lambda$, whenever $M \in \mathcal{P}\Lambda$,
$(\lambda x{:}\, \sigma.\, M) \in \mathcal{P}\Lambda$, whenever $x \in \mathcal{X}$, $\sigma \in \mathcal{T}$, and $M \in \mathcal{P}\Lambda$,
$(M\sigma) \in \mathcal{P}\Lambda$, whenever $\sigma \in \mathcal{T}$ and $M \in \mathcal{P}\Lambda$,
$(\Lambda t{:}\, K.\, M) \in \mathcal{P}\Lambda$, whenever $t \in \mathcal{V}$, $K \in \mathcal{K}$, and $M \in \mathcal{P}\Lambda$.

The notions of substitution and α-equivalence are extended in the obvious way. In order to deal with product types, it is necessary to add the following kind-checking rule:

$$\frac{\Delta \rhd \sigma{:}\, \star \qquad \Delta \rhd \tau{:}\, \star}{\Delta \rhd \sigma \times \tau{:}\, \star} \tag{$\times$}$$

The definition of the relation $\longrightarrow_{\lambda^{\rightarrow}}$ does not have to be changed, since the congruence rule takes care of $\Rightarrow$, $\times$, and Π_K.

It is easy to see that corollary 6.18 and corollary 6.19 hold for the new class of types. Thus, every ($\equiv_\alpha$-equivalence class of) type σ that kind-checks has a unique $\beta\eta$-normal form.

The following inference rules need to be added to the proof system used for type-checking terms.

$$\frac{\Delta \rhd M{:}\, \sigma \qquad \Delta \rhd N{:}\, \tau}{\Delta \rhd \langle M, N\rangle{:}\, \sigma \times \tau} \tag{$product$}$$

$$\frac{\Delta \rhd M{:}\, \sigma \times \tau}{\Delta \rhd \pi_1(M){:}\, \sigma} \qquad \frac{\Delta \rhd M{:}\, \sigma \times \tau}{\Delta \rhd \pi_2(M){:}\, \tau} \tag{$projection$}$$

The notion of reduction in F_ω is defined by adding the following axioms and rules to definition 15.1.

Axioms:

$$\pi_1(\langle M, N\rangle) \longrightarrow_\pi M, \tag{π}$$

$$\pi_2(\langle M, N\rangle) \longrightarrow_\pi N, \tag{π}$$

$$\langle \pi_1(M), \pi_2(M)\rangle \longrightarrow_{\langle\rangle} M, \tag{$\langle\rangle$}$$

Inference Rules: For each kind of reduction $\longrightarrow_r$ where $r \in \{\beta, \eta, \pi, \langle\rangle, \tau\beta, \tau\eta, \lambda^{\rightarrow}\}$,

$$\frac{M \longrightarrow_r N}{\langle M, Q\rangle \longrightarrow_r \langle N, Q\rangle} \qquad \frac{M \longrightarrow_r N}{\langle P, M\rangle \longrightarrow_r \langle P, N\rangle} \qquad \text{for all } P, Q \in \mathcal{P}\Lambda$$

$$\frac{M \longrightarrow_r N}{\pi_1(M) \longrightarrow_r \pi_1(N)} \qquad \frac{M \longrightarrow_r N}{\pi_2(M) \longrightarrow_r \pi_2(N)} \qquad \text{for all } P, Q \in \mathcal{P}\Lambda$$

We now generalize the method of candidates to F_ω with product types (with surjective pairing). As the proof given in section 16, the proof presented next is modelled after Girard's original proof, and only differs in the notation and in the fact that we present it in a slightly more general setting, using $\mathcal{T}$-closed families, and families of closed Girard sets. It should be noted that in the case of the simply-typed lambda calculus, a very similar method (but simpler, since only simple types need to be handled) has been used to give proofs of strong normalization, by Lambek and Scott [19], and de Vrijer [38,39].

Given any two types $\sigma, \tau \in \mathcal{T}|_\kappa$ of kind $\star$ and any two sets $S \subseteq \mathcal{PT}_\sigma$ and $T \subseteq \mathcal{PT}_\tau$, we let $S \times T$ be the subset of $\mathcal{PT}_{\sigma\times\tau}$ defined as before:

$$S \times T = \{\Delta \triangleright M \in \mathcal{PT}_{\sigma\times\tau} |\ \Delta \triangleright \pi_1(M) \in S \quad \text{and} \quad \Delta \triangleright \pi_2(M) \in T\}.$$

Given a kind assignment $\kappa\colon \mathcal{V} \to \mathcal{K}$, a $\mathcal{T}|_\kappa$-closed family is defined as follows.

Definition 17.4 Let $\mathcal{C} = (\mathcal{C}_\sigma)_{\sigma\in\mathcal{T}|_\kappa}$ be a $\mathcal{T}|_\kappa$-indexed family where for each σ, if σ is of kind $\star$ then $\mathcal{C}_\sigma$ is a nonempty set of subsets of $\mathcal{PT}_\sigma$, else if σ is of kind $K_1 \to K_2$ then $\mathcal{C}_\sigma$ is a nonempty set of functions from $\bigcup_{\tau:K_1}\{\tau\} \times \mathcal{C}_\tau$ to $\bigcup_{\tau:K_1} \mathcal{C}_{\sigma\tau}$, and the following properties hold:

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

(2) For every $\sigma, \tau \in \mathcal{T}|_\kappa$ of kind $\star$, for every $C \in \mathcal{C}_\sigma$ and $D \in \mathcal{C}_\tau$, we have $[C \Rightarrow D] \in \mathcal{C}_{\sigma\Rightarrow\tau}$.

(3) For every $\sigma \in \mathcal{T}|_\kappa$ of kind $K \to \star$, for every $\tau \in \mathcal{T}|_\kappa$ of kind K, for every family $(A_{\tau,C})_{\tau\in\mathcal{T}|_\kappa, C\in\mathcal{C}_\tau}$, where each set $A_{\tau,C}$ is in $\mathcal{C}_{\sigma\tau}$, we have

$$\{\Delta \triangleright M \in \mathcal{PT}_{\Pi_K\sigma} \mid \forall(\tau\colon K) \in \mathcal{T}|_\kappa,\ \Delta, \kappa \triangleright M\tau \in \bigcap_{C\in\mathcal{C}_\tau} A_{\tau,C}\} \in \mathcal{C}_{\Pi_K\sigma}.$$

(4) For every $\sigma \in \mathcal{T}|_\kappa$ of kind $K_1 \to K_2$,

$$\begin{aligned}\mathcal{C}_\sigma = \{f\colon \bigcup_{\tau:K_1\in\mathcal{T}|_\kappa} \{\tau\} \times \mathcal{C}_\tau \to \bigcup_{\tau:K_1\in\mathcal{T}|_\kappa} \mathcal{C}_{\sigma\tau} \\ \text{such that } f(\tau, C) \in \mathcal{C}_{\sigma\tau} \text{ for every } C \in \mathcal{C}_\tau, \text{ and} \\ f(\tau_1, C) = f(\tau_2, C) \text{ whenever } \tau_1 \overset{*}{\longleftrightarrow}_{\lambda\to} \tau_2\}.\end{aligned}$$

(5) For every $\sigma, \tau \in \mathcal{T}|_\kappa$ of kind $\star$, for every $C \in \mathcal{C}_\sigma$ and $D \in \mathcal{C}_\tau$, we have $C \times D \in \mathcal{C}_{\sigma\times\tau}$.

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

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

Definition 17.5 Given any candidate assignment $\langle\theta,\eta\rangle$, for every type $\sigma \in \mathcal{T}|_\kappa$, we define $[\![\sigma]\!]\theta\eta$ as follows:

$$
\begin{aligned}
[\![t]\!]\theta\eta &= \eta(t), \text{ whenever } t \in \mathcal{TC} \cup \mathcal{V};\\
[\![(\sigma \Rightarrow \tau)]\!]\theta\eta &= [[\![\sigma]\!]\theta\eta \Rightarrow [\![\tau]\!]\theta\eta];\\
[\![\sigma \times \tau]\!]\theta\eta &= [\![\sigma]\!]\theta\eta \times [\![\tau]\!]\theta\eta;\\
[\![\Pi_K\sigma]\!]\theta\eta &= \{\Delta \rhd M \in \mathcal{PT}_{\theta(\Pi_K\sigma)} \mid \forall(\tau\colon K) \in \mathcal{T}|_\kappa,\\
&\qquad \Delta,\kappa \rhd M\tau \in \bigcap_{C\in\mathcal{C}_\tau} [\![\sigma]\!]\theta\eta(\tau,C)\};\\
[\![\sigma\tau]\!]\theta\eta &= [\![\sigma]\!]\theta\eta(\theta(\tau),[\![\tau]\!]\theta\eta);\\
[\![\lambda t\colon K.\,\sigma]\!]\theta\eta &= \lambda\tau\lambda C \in \mathcal{C}_{\tau:K}.\,[\![\sigma]\!]\theta[t:=\tau]\eta[t:=C].
\end{aligned}
$$

In the last clause of this definition, $\lambda\tau\lambda C \in \mathcal{C}_{\tau:K}.\,[\![\sigma]\!]\theta[t:=\tau]\eta[t:=C]$ denotes the function f such that $f(\tau,C) = [\![\sigma]\!]\theta[t:=\tau]\eta[t:=C]$ for every $C \in \mathcal{C}_\tau$ such that $\tau \in \mathcal{T}|_\kappa$ is of kind K.

Lemma 16.5, 16.6, and 16.7 are unchanged. It is also easy to prove the following version of "Girard's trick" for F_ω with product types.

Lemma 17.6 (Girard) If $\mathcal{C}$ is a $\mathcal{T}|_\kappa$-closed family, for every candidate assignment $\langle\theta,\eta\rangle$, for every type σ, then $[\![\sigma]\!]\theta\eta \in \mathcal{C}_{\theta(\sigma)}$. $\square$

One more condition needs to be added to the conditions of definition 16.9

Definition 17.7 We say that a $\mathcal{T}|_\kappa$-indexed family $\mathcal{C}$ is a *family of sets of candidates of reducibility* iff it is $\mathcal{T}|_\kappa$-closed and satisfies the conditions listed below.[26]

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

R1. For every $\sigma\colon\star \in \mathcal{T}|_\kappa$, for every set $C \in \mathcal{C}_\sigma$, $\Delta \rhd x \in C$, for every $x\colon\sigma \in \Delta$,
For every $\sigma\colon\star \in \mathcal{T}|_\kappa$ where $\sigma = \Theta(f)$, for every set $C \in \mathcal{C}_\sigma$, $\Delta \rhd f \in C$, for every $f \in \Sigma$.

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

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

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

(ii) For every $\sigma \in \mathcal{T}|_\kappa$ of kind $K \to \star$, every $\tau \in \mathcal{T}|_\kappa$ of kind K, for every $C \in \mathcal{C}_{\sigma\tau}$, for all Δ, Δ', if

$$\begin{aligned} &\Delta \rhd M \in \bigcup \mathcal{C}_{\sigma t} \text{ and} \\ &\Delta' \rhd M[\tau/t] \in C, \text{ then} \\ &\Delta' \rhd (\Lambda t{:}\, K.\, M)\tau \in C. \end{aligned}$$

R3. For all $\sigma{:}\star, \tau{:}\star \in \mathcal{T}|_\kappa$, for every $C \in \mathcal{C}_\sigma$ and $D \in \mathcal{C}_\tau$, if $\Delta \rhd M \in C$ and $\Delta \rhd N \in D$, then $\Delta \rhd \langle M, N\rangle \in C \times D$.

We now have a version of lemma 16.10 for F_ω with product types.

Lemma 17.8 (Girard) Let $\mathcal{C} = (\mathcal{C}_\sigma)_{\sigma \in \mathcal{T}|_\kappa}$ be a $\mathcal{T}|_\kappa$-indexed family of sets of candidates of reducibility. For every $\Gamma \rhd M \in \mathcal{PT}_\sigma$, every candidate assignment $\langle \theta, \eta\rangle$, every substitution $\varphi{:}\, \Gamma \to \Delta$, if $\theta(\Delta), \kappa \rhd \varphi(x) \in [\![\Gamma(x)]\!]\theta\eta$ for $x \in FV(M)$, then $\theta(\Delta), \kappa \rhd \varphi(\theta(M)) \in [\![\sigma]\!]\theta\eta$.

Proof. We only sketch the verification for the new cases.

Case 1.

$$\frac{\Gamma \rhd M{:}\,\sigma \qquad \Gamma \rhd N{:}\,\tau}{\Gamma \rhd \langle M, N\rangle{:}\, \sigma \times \tau} \qquad (\textit{product})$$

By the induction hypothesis,

$$\theta(\Delta), \kappa \rhd \varphi(\theta(M)) \in [\![\sigma]\!]\theta\eta,$$

and

$$\theta(\Delta), \kappa \rhd \varphi(\theta(N)) \in [\![\tau]\!]\theta\eta.$$

By (R3) and the definition of $[\![\sigma \times \tau]\!]\theta\eta$, we have

$$\theta(\Delta), \kappa \rhd \langle \varphi(\theta(M)), \varphi(\theta(N))\rangle \in [\![\sigma \times \tau]\!]\theta\eta.$$

However, this is equivalent to

$$\theta(\Delta), \kappa \rhd \varphi(\theta(\langle M, N\rangle)) \in [\![\sigma \times \tau]\!]\theta\eta.$$

Case 2.

$$\frac{\Gamma \rhd M{:}\,\sigma \times \tau}{\Gamma \rhd \pi_1(M){:}\,\sigma} \qquad \frac{\Gamma \rhd M{:}\,\sigma \times \tau}{\Gamma \rhd \pi_2(M){:}\,\tau} \qquad (\textit{projection})$$

By the induction hypothesis,

$$\theta(\Delta), \kappa \triangleright \varphi(\theta(M)) \in [\![\sigma \times \tau]\!]\theta\eta.$$

Since $[\![\sigma \times \tau]\!]\theta\eta = [\![\sigma]\!]\theta\eta \times [\![\tau]\!]\theta\eta$, this implies that $\theta(\Delta), \kappa \triangleright \varphi(\theta(\pi_1(M))) \in [\![\sigma]\!]\theta\eta$ and $\theta(\Delta), \kappa \triangleright \varphi(\theta(\pi_2(M))) \in [\![\tau]\!]\theta\eta$. $\square$

Remarkably, because Girard's conditions take the reduction relation $\longrightarrow_{F_\omega}$ into account, there is no need to add extra conditions besides (CR0), (CR1), (CR2), and (CR3). We simply need to modify the definition of a simple term, so that lemma 16.13 holds for F_ω with product types. For this, it is enough to preclude a pair $\langle M, N\rangle$ from being simple. Thus, a term M is *simple* iff it is either a variable x, a constant $f \in \Sigma$, an application MN, a projection $\pi_1(M)$ or $\pi_2(M)$, or a type application $M\tau$.

Definition 17.9 Let $S = (S_\sigma)_{\sigma:\star\in\mathcal{T}|_\kappa}$ be a family such that each S_σ is a nonempty subset of $\mathcal{PT}_\sigma$.[27] For every type $\sigma:\star \in \mathcal{T}|_\kappa$, a subset C of S_σ is a *Girard set* of type σ iff the following conditions hold:[28]

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 $\longrightarrow_{F_\omega}$;

CR2. If $\Delta \triangleright M \in C$ and $M \longrightarrow_{F_\omega} N$, 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 $M \longrightarrow_{F_\omega} N$, then $\Delta \triangleright M \in C$.

Note that in the above definition, $\longrightarrow_{F_\omega}$ is the reduction relation for F_ω with product types, and consequently, covers the case of reductions when M is of the form $\pi_1(\langle P, Q\rangle)$ or $\pi_2(\langle P, Q\rangle)$.

We need to add one more clause to definition 16.12 (defining a closed family): for all $\sigma:\star, \tau:\star \in \mathcal{T}|_\kappa$, if $\Delta \triangleright M \in \mathcal{PT}_{\sigma\times\tau}$, $\Delta \triangleright \pi_1(M) \in S_\sigma$, and $\Delta \triangleright \pi_2(M) \in S_\tau$, then $\Delta \triangleright M \in S_{\sigma\times\tau}$.

We have the following generalization of lemma 16.13.

Lemma 17.10 (Girard) Let $S = (S_\sigma)_{\sigma:\star\in\mathcal{T}|_\kappa}$ be a closed family where each S_σ is a nonempty subset of $\mathcal{PT}_\sigma$, and let $\mathcal{C}$ be the $\mathcal{T}|_\kappa$-indexed family such that for each $\sigma:\star \in \mathcal{T}|_\kappa$, $\mathcal{C}_\sigma$ is the set of Girard subsets of S_σ, and for $\sigma: K_1 \to K_2 \in \mathcal{T}|_\kappa$, $\mathcal{C}_\sigma$ is defined as in clause

[27] Note that $S = (S_\sigma)_{\sigma:\star\in\mathcal{T}|_\kappa}$ is not a $\mathcal{T}|_\kappa$-indexed family. It is indexed by the set of types of kind $\star$.

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

(4) of definition 17.4. If $S_\sigma \in \mathcal{C}_\sigma$ for every $\sigma\colon\star \in \mathcal{T}|_\kappa$ (i.e. S_σ is a Girard subset of itself), then $\mathcal{C}$ is a family of sets of candidates of reducibility.

Proof. It is similar to the proof of lemmas 9.3 and 9.4. Property (R3) is shown as follows. Assume that $\Delta \rhd M \in C$ and $\Delta \rhd N \in D$. We show by induction on $\delta(M) + \delta(N)$ that $\Delta \rhd Q \in C$ whenever $\pi_1(\langle M, N\rangle) \longrightarrow_{F_\omega} Q$, and that $\Delta \rhd Q \in D$ whenever $\pi_2(\langle M, N\rangle) \longrightarrow_{F_\omega} Q$. Then, by (CR3), we have $\Delta \rhd \pi_1(\langle M, N\rangle) \in C$ and $\Delta \rhd \pi_2(\langle M, N\rangle) \in D$, which, by the definition of $C \times D$, shows that $\Delta \rhd \langle M, N\rangle \in C \times D$.

We prove that $\Delta \rhd Q \in C$ whenever $\pi_1(\langle M, N\rangle) \longrightarrow_{F_\omega} Q$, the other case being similar. The point is that either $Q = M$, or $Q = \pi_1(\langle M', N\rangle)$ where $M \longrightarrow_{F_\omega} M'$, or $Q = \pi_1(\langle M, N'\rangle)$ where $N \longrightarrow_{F_\omega} N'$. Note that the case where $M = \pi_1(U)$ and $N = \pi_2(U)$ must be considered, since in this case $\langle \pi_1(U), \pi_2(U)\rangle \longrightarrow_{F_\omega} U$. But then,

$$\pi_1(\langle M, N\rangle) = \pi_1(\langle \pi_1(U), \pi_2(U)\rangle) \longrightarrow_{F_\omega} \pi_1(U) = M.$$

In the first case, the hypothesis yields $\Delta \rhd M \in C$. In the other two cases, by (CR2) we have $\Delta \rhd M' \in C$ and $\Delta \rhd N' \in D$, and since $\delta(M') < \delta(M)$ and $\delta(N') < \delta(N)$, we use the induction hypothesis and (CR3). The base case where M and N are irreducible follows from (CR3).

It is also necessary to verify conditions (1), (2), (3), (4), (5) of definition 17.4. This is done by induction on kinds, and for the kind $\star$ by induction on types. The only new case is case (5). Given that $C \subseteq S_\sigma$ and $D \subseteq S_\tau$, the fact that $C \times D \subseteq S_{\sigma\times\tau}$ follows from the new closure condition (on $\times$). We need to prove that (CR1), (CR2), and (CR3) hold for $C \times D$, given that they hold for C and D. Let $\Delta \rhd M\colon \sigma \times \tau$ be a term in $C \times D$, where $C \in \mathcal{C}_\sigma$ and $D \in \mathcal{C}_\tau$. By the definition of $C \times D$, $\Delta \rhd \pi_1(M) \in C$ (and $\Delta \rhd \pi_2(M) \in D$). Since all terms in C are SN, $\pi_1(M)$ is SN. But then, M itself is necessary SN since any infinite reduction from M yields an infinite reduction from $\pi_1(M)$.

Assume that $M \longrightarrow_{F_\omega} M'$. Then, $\pi_1(M) \longrightarrow_{F_\omega} \pi_1(M')$ and $\pi_2(M) \longrightarrow_{F_\omega} \pi_2(M')$. Since $\pi_1(M) \in C$ and $\pi_2(M) \in D$, by (CR2) applied to C and D, we have $\pi_1(M') \in C$ and $\pi_2(M') \in D$. By the definition of $C \times D$, we have $\Delta \rhd M' \in C \times D$, and (CR2) holds.

Now, assume that M is simple, and that whenever $M \longrightarrow_{F_\omega} Q$, then $\Delta \rhd Q \in C \times D$. We want to prove that $\Delta \rhd M \in C \times D$. Note that $\pi_1(M)$ and $\pi_2(M)$ are also simple, and that because M is simple,

$(*)$ $\pi_1(M) \longrightarrow_{F_\omega} R$ implies that $R = \pi_1(Q)$ where $M \longrightarrow_{F_\omega} Q$, and similarly for $\pi_2(M)$.

Since we assumed that $\Delta \rhd Q \in C \times D$, we have $\Delta \rhd \pi_1(Q) \in C$ and $\Delta \rhd \pi_2(Q) \in D$. By (CR3) applied to C and D and $(*)$, since $\pi_1(M)$ and $\pi_2(M)$ are simple, we have

$\Delta \triangleright \pi_1(M) \in C$ and $\Delta \triangleright \pi_2(M) \in D$. By the definition of $C \times D$, we have $\Delta \triangleright M \in C \times D$, and (CR3) holds. $\square$

We now have a version of Girard's fundamental theorem for F_ω with product types.

Theorem 17.11 (Girard) Let $S = (S_\sigma)_{\sigma:\star\in\mathcal{T}|_\kappa}$ be a closed family where each S_σ is a nonempty subset of $\mathcal{PT}_\sigma$, let $\mathcal{C}$ be the $\mathcal{T}|_\kappa$-indexed family of sets defined in lemma 17.10, and assume that $S_\sigma \in \mathcal{C}_\sigma$ for every $\sigma\colon\star \in \mathcal{T}|_\kappa$. For every $\Delta \triangleright M \in \mathcal{PT}_\sigma$, we have $\Delta \triangleright M \in S_\sigma$.

Proof. By lemma 17.10, $\mathcal{C}$ is a family of sets of candidates of reducibility. We now apply lemma 17.8 to any assignment (for example, the assignment with value $\eta(t) = can_t$), the identity type substitution, and the identity term substitution, which is legitimate since by (CR3), every variable belongs to every Girard set.[29] $\square$

The next lemma shows that F_ω with product types is strongly normalizing and confluent under both β-reduction and $\beta\eta$-reduction.

Lemma 17.12 (i) The family SN_β such that for every $\sigma\colon\star \in \mathcal{T}|_\kappa$, $SN_{\beta,\sigma}$ is the set of typing judgments $\Delta \triangleright M\colon\sigma$ provable in F_ω such that M is strongly normalizing under β-reduction, is a closed family of Girard sets. (ii) The family $SN_{\beta\eta}$ such that for every $\sigma\colon\star \in \mathcal{T}|_\kappa$, $SN_{\beta\eta,\sigma}$ is the set of typing judgments $\Delta \triangleright M\colon\sigma$ provable in F_ω such that M is strongly normalizing under $\beta\eta$-reduction, is a closed family of Girard sets. (iii) The family consisting for every $\sigma\colon\star \in \mathcal{T}|_\kappa$ of the set of typing judgments $\Delta \triangleright M\colon\sigma$ provable in F_ω such that confluence under β-reduction holds from M and all of its subterms, is a closed family of Girard sets. (iv) The family consisting for every $\sigma\colon\star \in \mathcal{T}|_\kappa$ of the set of typing judgments $\Delta \triangleright M\colon\sigma$ provable in F_ω such that confluence under $\beta\eta$-reduction holds from M and all of its subterms, is a closed family of Girard sets.

Proof. (i)-(ii) It is trivial to verify that closure and (CR0)–(CR3) hold. (iii)-(iv) The proof is similar to the one given in appendix 2, with a few more cases involving pairs and projections. $\square$

18 Appendix 2

This appendix contains the details that were omitted in the proof of lemma 10.2.

Proof of lemma 10.2. We need to prove that the family of sets in (iii) and (iv) is a closed family of saturated sets.

[29] 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.

Closure is obvious since if confluence holds from Mx and all of its subterms, then it holds from M.

Verifying (S1) is easy and uses the fact that if confluence holds from $N_1, \ldots, N_n$ and all of their subterms, then confluence holds from each $uN_1 \ldots N_k$ and all of its subterms, where $u \in \mathcal{X} \cup \Sigma$ and $1 \leq k \leq n$. This is because reductions must apply withing the N_i's. Since confluence from u is trivial, confluence from each subterm of $uN_1 \ldots N_n$ follows from the assumption on $N_1, \ldots, N_n$.

Proving (S2) is very tedious. Assume that confluence holds from $M[N/x]N_1 \ldots N_n$ and all of its subterms (and that confluence holds from N, which is implied by (S2)). Thus, confluence holds from $M, N, N_1, \ldots, N_n$. We need to show that confluence holds from $(\lambda x\colon \sigma.\, M)NN_1 \ldots N_n$ and all of its subterms.

First, we show confluence from every subterm of $\lambda x\colon \sigma. M$. Since confluence holds from every subterm of M, the only nontrivial case is the case where $M \xrightarrow{*}_{\lambda^\forall} (M_1'x)$, for some term M_1' such that $x \notin FV(M_1')$. In this case, we can have reductions of the form

$$\lambda x\colon \sigma.\, M \xrightarrow{*}_{\lambda^\forall} \lambda x\colon \sigma.\, (M_1'x) \longrightarrow_\eta M_1' \xrightarrow{*}_{\lambda^\forall} P_1,$$

where $M \xrightarrow{*}_{\lambda^\forall} M_1'x$. There are four cases to consider.

Case 1.

$$\lambda x\colon \sigma.\, M \xrightarrow{*}_{\lambda^\forall} \lambda x\colon \sigma.\, (M_1'x) \longrightarrow_\eta M_1' \xrightarrow{*}_{\lambda^\forall} P_1$$

and

$$\lambda x\colon \sigma.\, M \xrightarrow{*}_{\lambda^\forall} \lambda x\colon \sigma.\, (M_1''x) \longrightarrow_\eta M_1'' \xrightarrow{*}_{\lambda^\forall} P_2,$$

where $M \xrightarrow{*}_{\lambda^\forall} M_1'x$ and $M \xrightarrow{*}_{\lambda^\forall} M_1''x$. Since confluence holds from M, there are reductions $P_1x \xrightarrow{*}_{\lambda^\forall} Q$ and $P_2x \xrightarrow{*}_{\lambda^\forall} Q$ for some Q.

If both reductions are of the form $P_1x \xrightarrow{*}_{\lambda^\forall} P_3x$ and $P_2x \xrightarrow{*}_{\lambda^\forall} P_3x$, for some P_3 such that $Q = P_3x$, $P_1 \xrightarrow{*}_{\lambda^\forall} P_3$, and $P_2 \xrightarrow{*}_{\lambda^\forall} P_3$, then confluence holds.

If $P_2 \xrightarrow{*}_{\lambda^\forall} P_3$ and $P_1x \xrightarrow{*}_{\lambda^\forall} P_3x$ is of the form

$$P_1x \xrightarrow{*}_{\lambda^\forall} (\lambda y\colon \sigma.\, Q_1)x \longrightarrow_\beta Q_1[x/y] \xrightarrow{*}_{\lambda^\forall} P_3x,$$

since $x \notin FV(M_1'')$ due to the η-step, and since $\lambda y\colon \sigma.\, Q_1 \equiv_\alpha \lambda x\colon \sigma.\, Q_1[x/y]$, we have $x \notin FV(P_3)$ and

$$P_1 \xrightarrow{*}_{\lambda^\forall} \lambda y\colon \sigma.\, Q_1 \xrightarrow{*}_{\lambda^\forall} \lambda x\colon \sigma.\, (P_3x) \longrightarrow_\eta P_3.$$

Confluence holds, since $P_2 \xrightarrow{*}_{\lambda^\forall} P_3$.

The subcase where $P_1 \xrightarrow{*}_{\lambda^\forall} P_3$ and

$$P_2x \xrightarrow{*}_{\lambda^\forall} (\lambda y\colon \sigma.\, Q_2)x \longrightarrow_\beta Q_2[x/y] \xrightarrow{*}_{\lambda^\forall} P_3x,$$

is symmetric.

If both

$$P_1x \xrightarrow{*}_{\lambda^\forall} (\lambda y\colon \sigma.\, Q_1)x \longrightarrow_\beta Q_1[x/y] \xrightarrow{*}_{\lambda^\forall} Q,$$

and

$$P_2x \xrightarrow{*}_{\lambda^\forall} (\lambda y\colon \sigma.\, Q_2)x \longrightarrow_\beta Q_2[x/y] \xrightarrow{*}_{\lambda^\forall} Q,$$

since $\lambda y\colon \sigma.\, Q_1 \equiv_\alpha \lambda x\colon \sigma.\, Q_1[x/y]$ and $\lambda y\colon \sigma.\, Q_2 \equiv_\alpha \lambda x\colon \sigma.\, Q_2[x/y]$, we have

$$P_1 \xrightarrow{*}_{\lambda^\forall} \lambda y\colon \sigma.\, Q_1 \xrightarrow{*}_{\lambda^\forall} \lambda x\colon \sigma.\, Q$$

and

$$P_2 \xrightarrow{*}_{\lambda^\forall} \lambda y\colon \sigma.\, Q_2 \xrightarrow{*}_{\lambda^\forall} \lambda x\colon \sigma.\, Q.$$

Case 2.

$$\lambda x\colon \sigma.\, M \xrightarrow{*}_{\lambda^\forall} \lambda x\colon \sigma.\, (M_1'x) \longrightarrow_\eta M_1' \xrightarrow{*}_{\lambda^\forall} P_1$$

and

$$\lambda x\colon \sigma.\, M \xrightarrow{*}_{\lambda^\forall} \lambda x\colon \sigma.\, M_1'',$$

where $M \xrightarrow{*}_{\lambda^\forall} M_1'x$ and $M \xrightarrow{*}_{\lambda^\forall} M_1''$. This is quite similar to case 1. If $P_1x \xrightarrow{*}_{\lambda^\forall} P_3x$ and $M_1'' \xrightarrow{*}_{\lambda^\forall} P_3x$, since $x \notin FV(M_1')$ due to the η-step, we have $x \notin FV(P_3)$, $P_1 \xrightarrow{*}_{\lambda^\forall} P_3$, and

$$\lambda x\colon \sigma.\, M_1'' \xrightarrow{*}_{\lambda^\forall} \lambda x\colon \sigma.\, (P_3x) \longrightarrow_\eta P_3.$$

Confluence holds since $P_1 \xrightarrow{*}_{\lambda^\forall} P_3$.

If

$$P_1x \xrightarrow{*}_{\lambda^\forall} (\lambda y\colon \sigma.\, Q_1)x \longrightarrow_\beta Q_1[x/y] \xrightarrow{*}_{\lambda^\forall} Q,$$

and

$$M_1'' \xrightarrow{*}_{\lambda^\forall} Q,$$

since $\lambda x\colon \sigma.\, Q_1[x/y] \equiv_\alpha \lambda y\colon \sigma.\, Q_1$, we have

$$P_1 \xrightarrow{*}_{\lambda^\forall} \lambda y\colon \sigma.\, Q_1 \xrightarrow{*}_{\lambda^\forall} \lambda x\colon \sigma.\, Q$$

and

$$\lambda x\colon \sigma.\, M_1'' \xrightarrow{*}_{\lambda^\forall} \lambda x\colon \sigma.\, Q.$$

Case 3. Symmetric to case 2.

Case 4.

$$\lambda x{:}\,\sigma.\ M \xrightarrow{*}_{\lambda\forall} \lambda x{:}\,\sigma.\ M_1'$$

and

$$\lambda x{:}\,\sigma.\ M \xrightarrow{*}_{\lambda\forall} \lambda x{:}\,\sigma.\ M_1'',$$

where $M \xrightarrow{*}_{\lambda\forall} M_1'$ and $M \xrightarrow{*}_{\lambda\forall} M_1''$. Since confluence holds from M, we conclude immediately.

We now prove that confluence holds from every term $(\lambda x{:}\,\sigma.\ M)NN_1 \ldots N_i$, where $1 \leq i \leq n$, and from $(\lambda x{:}\,\sigma.\ M)N$. Without loss of generality, we can assume that $i = n$. The proof of lemma 6.16 showed that every reduction sequence from $(\lambda x{:}\,\sigma.\ M)NN_1 \ldots N_n$ is of the form

$$(\lambda x{:}\,\sigma.\ M)NN_1 \ldots N_n \xrightarrow{*}_{\lambda\forall} (\lambda x{:}\,\sigma.\ M')N'N_1' \ldots N_n',$$

or

$$(\lambda x{:}\,\sigma.\ M)NN_1 \ldots N_n \xrightarrow{*}_{\lambda\forall} (\lambda x{:}\,\sigma.\ M')N'N_1' \ldots N_n' \longrightarrow_{\beta} M'[N'/x]N_1' \ldots N_n' \xrightarrow{*}_{\lambda\forall} Q,$$

or

$$(\lambda x{:}\,\sigma.\ M)NN_1 \ldots N_n \xrightarrow{*}_{\lambda\forall} (\lambda x{:}\,\sigma.\ (M_1'x))N'N_1' \ldots N_n' \longrightarrow_{\eta} M_1'N'N_1' \ldots N_n' \xrightarrow{*}_{\lambda\forall} Q,$$

where in (1)-(2) $M \xrightarrow{*}_{\lambda\forall} M'$, in (3) $M \xrightarrow{*}_{\lambda\forall} M_1'x$, and in all cases $N \xrightarrow{*}_{\lambda\forall} N'$, and $N_i \xrightarrow{*}_{\lambda\forall} N_i'$, for every i, $1 \leq i \leq n$. We have seven main cases.

Case 1.

$$(\lambda x{:}\,\sigma.\ M)NN_1 \ldots N_n \xrightarrow{*}_{\lambda\forall} (\lambda x{:}\,\sigma.\ M')N'N_1' \ldots N_n'$$

and

$$(\lambda x{:}\,\sigma.\ M)NN_1 \ldots N_n \xrightarrow{*}_{\lambda\forall} (\lambda x{:}\,\sigma.\ M'')N''N_1'' \ldots N_n''.$$

Since confluence holds from $M, N, N_1 \ldots N_n$ (for N, this is implied by (S2)), we have reduction sequences $M' \xrightarrow{*}_{\lambda\forall} M'''$, $M'' \xrightarrow{*}_{\lambda\forall} M'''$, $N' \xrightarrow{*}_{\lambda\forall} N'''$, $N'' \xrightarrow{*}_{\lambda\forall} N'''$, and $N_i' \xrightarrow{*}_{\lambda\forall} N_i'''$, $N_i'' \xrightarrow{*}_{\lambda\forall} N_i'''$, for every i, $1 \leq i \leq n$, and thus reductions

$$(\lambda x{:}\,\sigma.\ M')N'N_1' \ldots N_n' \xrightarrow{*}_{\lambda\forall} (\lambda x{:}\,\sigma.\ M''')N'''N_1''' \ldots N_n'''$$

and

$$(\lambda x{:}\,\sigma.\ M'')N''N_1'' \ldots N_n'' \xrightarrow{*}_{\lambda\forall} (\lambda x{:}\,\sigma.\ M''')N'''N_1''' \ldots N_n'''.$$

Case 2.

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} (\lambda x{:}\,\sigma.\,M')N'N'_1\ldots N'_n$$

and

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} (\lambda x{:}\,\sigma.\,M'')N''N''_1\ldots N''_n \longrightarrow_\beta M''[N''/x]N''_1\ldots N''_n \xrightarrow{*}_{\lambda^\forall} P_2.$$

This time, we also have reduction sequences

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \longrightarrow_\beta M[N/x]N_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} M'[N'/x]N'_1\ldots N'_n$$

and

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \longrightarrow_\beta M[N/x]N_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} M''[N''/x]N''_1\ldots N''_n \xrightarrow{*}_{\lambda^\forall} P_2.$$

Using the confluence from $M[N/x]N_1\ldots N_n$, we have $M'[N'/x]N'_1\ldots N'_n \xrightarrow{*}_{\lambda^\forall} P_3$ and $P_2 \xrightarrow{*}_{\lambda^\forall} P_3$ for some P_3. Thus, we have reductions

$$(\lambda x{:}\,\sigma.\,M')N'N'_1\ldots N'_n \longrightarrow_\beta M'[N'/x]N'_1\ldots N'_n \xrightarrow{*}_{\lambda^\forall} P_3$$

and

$$P_2 \xrightarrow{*}_{\lambda^\forall} P_3.$$

Case 3.

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} (\lambda x{:}\,\sigma.\,M')N'N'_1\ldots N'_n \longrightarrow_\beta M'[N'/x]N'_1\ldots N'_n \xrightarrow{*}_{\lambda^\forall} P_1$$

and

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} (\lambda x{:}\,\sigma.\,M'')N''N''_1\ldots N''_n.$$

Symmetric to case 2.

Case 4.

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} (\lambda x{:}\,\sigma.\,M')N'N'_1\ldots N'_n \longrightarrow_\beta M'[N'/x]N'_1\ldots N'_n \xrightarrow{*}_{\lambda^\forall} P_1$$

and

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} (\lambda x{:}\,\sigma.\,M'')N''N''_1\ldots N''_n \longrightarrow_\beta M''[N''/x]N''_1\ldots N''_n \xrightarrow{*}_{\lambda^\forall} P_2.$$

As in case 2, we have reductions

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \longrightarrow_\beta M[N/x]N_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} M'[N'/x]N'_1\ldots N'_n \xrightarrow{*}_{\lambda^\forall} P_1$$

and

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \longrightarrow_\beta M[N/x]N_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} M''[N''/x]N''_1\ldots N''_n \xrightarrow{*}_{\lambda^\forall} P_2,$$

and we conclude using the confluence from $M[N/x]N_1\ldots N_n$.

Case 5.

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} (\lambda x{:}\,\sigma.\,(M'_1x))N'N'_1\ldots N'_n \longrightarrow_\eta M'_1N'N'_1\ldots N'_n \xrightarrow{*}_{\lambda^\forall} P_1,$$

and

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} (\lambda x{:}\,\sigma.\,(M''_1x))N''N''_1\ldots N''_n$$
$$\longrightarrow_\eta M''_1N''N''_1\ldots N''_n \xrightarrow{*}_{\lambda^\forall} P_2,$$

where $M \xrightarrow{*}_{\lambda^\forall} M'_1x$, $M \xrightarrow{*}_{\lambda^\forall} M''_1x$. Because of the η-steps, $x \notin FV(M'_1)$ and $x \notin FV(M''_1)$, and thus $M'_1[N'/x] = M'_1$ and $M''_1[N''/x] = M''_1$, and we have reductions

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \longrightarrow_\beta M[N/x]N_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} M'_1N'N'_1\ldots N'_n \xrightarrow{*}_{\lambda^\forall} P_1,$$

and

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \longrightarrow_\beta M[N/x]N_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} M''_1N''N''_1\ldots N''_n \xrightarrow{*}_{\lambda^\forall} P_2,$$

and we conclude using the confluence from $M[N/x]N_1\ldots N_n$.

Case 6.

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} (\lambda x{:}\,\sigma.\,(M'_1x))N'N'_1\ldots N'_n \longrightarrow_\eta M'_1N'N'_1\ldots N'_n \xrightarrow{*}_{\lambda^\forall} P_1,$$

and

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} (\lambda x{:}\,\sigma.\,M''_1N'')N''_1\ldots N''_n$$
$$\longrightarrow_\beta M''_1[N''/x]N''_1\ldots N''_n \xrightarrow{*}_{\lambda^\forall} P_2,$$

where $M \xrightarrow{*}_{\lambda^\forall} M'_1x$, $M \xrightarrow{*}_{\lambda^\forall} M''_1$. Since we have an η-reduction step, $x \notin FV(M'_1)$, which implies that $M'_1[N'/x] = M'_1$, and we have

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \longrightarrow_\beta M[N/x]N_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} M'_1N'N'_1\ldots N'_n \xrightarrow{*}_{\lambda^\forall} P_1,$$

and

$$(\lambda x{:}\,\sigma.\,M)NN_1\ldots N_n \xrightarrow{*}_\beta M[N/x]N_1\ldots N_n \xrightarrow{*}_{\lambda^\forall} M''_1[N''/x]N''_1\ldots N''_n \xrightarrow{*}_{\lambda^\forall} P_2.$$

We conclude using the confluence from $M[N/x]N_1\ldots N_n$.

Case 7. Symmetric to case 6.

Proving that confluence holds from $(\Lambda t.\,M)\tau N_1\ldots N_n$ and all its subterms assuming that confluence holds from $M[\tau/t]N_1\ldots N_n$ and all its subterms is similar to the previous proof, except that there are no reductions from τ. $\square$

19 References

[1] Barendregt, H. *The Lambda Calculus. Its Syntax and Semantics*, Revised Edition, North-Holland, Amsterdam, 1984.

[2] Breazu-Tannen, V., and Coquand, Th., Extensional models for polymorphism, *Theor. Comput. Sci.* 59(1,2), 1988, 85-114.

[3] Breazu-Tannen, V., Combining Algebra and Higher-Order Types, In *3rd Annual Symposium on Logic In Computer Science*, IEEE Computer Society, Edinburgh, Scotland, July 1988, 82-90.

[4] Breazu-Tannen, V., and Gallier, J.H., Polymorphic Rewriting Conserves Algebraic Strong Normalization and Confluence, *16th International Colloquium on Automata, Languages and Programming*, ed. by G. Ausiello, M. Dezani-Ciancaglini, and S. Ronchi Della Rocca, LNCS No. 372, Springer Verlag, July 1989, 137-150.

[5] Coquand, Th., Une Théorie des Constructions, Thèse de 3^{eme} Cycle, Université de Paris VII (1985).

[6] Coquand, Th., and Huet, G., The Calculus of Constructions, *Information and Computation* 76(2/3), 1988, 95-120.

[7] Coquand, Th. Mathematical investigation of a calculus of constructions, In: *Logic and Computer Science*, P. Odifreddi, editor, 1990.

[8] Fortune, S., Leivant, D., and O'Donnell, M., The expressiveness of simple and second-order type structures, *J.ACM* 30(1), 1983, 151-185.

[9] Girard, J.Y., Une extension de l'interprétation de Gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types, *Proc. 2nd Scand. Log. Symp., 1970*, ed. J.E. Fenstad, North-Holland, Amsterdam, 1971, 63-92.

[10] Girard, J.Y., Interprétation functionelle et élimination des coupures de l'arithmétique d'ordre supérieur, Thèse d'Etat, Université de Paris VII (1972).

[11] Girard, J.Y., Lafont, Y., and Taylor, P., *Proofs and Types*, Cambridge University Press, 1989.

[12] Girard, J.Y., *Proof Theory and Logical Complexity*, Studies in proof theory, Bibliopolis, Napoli, 1987.

[13] Harper, R., Honsell, F., and Plotkin, G., A Framework for Defining Logics, In *2nd Annual Symposium on Logic In Computer Science*, IEEE Computer Society, Ithaca, New York, June 1987, 194-204.

[14] Harper, R., Honsell, F., and Plotkin, G., A Framework for Defining Logics, draft submitted for publication, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, 1989.

[15] Hindley, J.R., and Seldin, J.P., *Introduction to Combinators and Lambda Calculus*, Cambridge University Press, 1986.

[16] Huet, G., *Résolution d'Equations dans les Langages d'Ordre* $1, 2, \ldots, \omega$, Thèse d'Etat, Université de Paris VII (1976).

[17] Huet, G., "Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems," *J.ACM* 27:4, 1980, 797-821.

[18] Huet, G., *Deduction and Computation*, Lecture Notes, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, 1987.

[19] Lambek, J., and Scott, P.J., *Introduction to Higher Order Categorical Logic*, Cambridge University Press, 1986.

[20] Leivant, D., Typing and Computational Properties of Lambda Expressions, *Theoretical Computer Science* 44, 1986, 51-68.

[21] Martin-Löf, P., Hauptsatz for the Intuitionistic Theory of Iterated Inductive Definitions, *Proc. 2nd Scand. Log. Symp., 1970*, ed. J.E. Fenstad, North-Holland, Amsterdam, 1971, 179-216.

[22] Martin-Löf, P., Hauptsatz for the Theory of Species, *Proc. 2nd Scand. Log. Symp., 1970*, ed. J.E. Fenstad, North-Holland, Amsterdam, 1971, 217-233.

[23] Martin-Löf, P., Hauptsatz for Intuitionistic Simple Type Theory, In *Logic, Methodology and Philosophy of Science IV*, ed. by P. Suppes, L. Henkin, A. Joja, and Gr.C. Moisil, North-Holland, Amsterdam, 1973, 279-290.

[24] Mitchell, J.C., A type-inference approach to reduction properties and semantics of polymorphic expressions, In *Proc. 1986 ACM Symposium on Lisp and Functional Programming*, ACM, New York, August 1986, 308-319.

[25] Mitchell, J.C., Type Systems for Programming Languages, to appear in *Handbook of Theoretical Computer Science*, ed. by J. van Leeuwen, A. Meyer, and D. Perrin, North-Holland, 1990.

[26] Paulin-Morhing, C., Extraction de programmes dans le calcul des constructions, Thèse d'Etat, Université de Paris VII, 1989.

[27] Pfenning, F., Partial Polymorphic Type Inference and Higher-Order Unification, In *1988 ACM Conference on Lisp and Functional Programming*, ACM, Snowbird, Utah, July 25-27, 1988, 153-163.

[28] Pfenning, F., Elf, A language for logic definition and verified metaprogramming, In *4th Annual Symposium on Logic In Computer Science*, IEEE Computer Society, Asilomar, California, June 1989, 313-322.

[29] Prawitz, D., Ideas and Results in Proof Theory, *Proc. 2nd Scand. Log. Symp., 1970*, ed. J.E. Fenstad, North-Holland, Amsterdam, 1971, 235-307.

[30] Reynolds, J., Towards a theory of type structure. In *Proc. sur la Programmation*, LCNS No. 19, Springer Verlag, NY, 1974, 408-425.

[31] Scedrov, A, Normalization revisited, In: *Categories in Logic and Computer Science, Proceedings of a Summer Research Conference, Boulder, Colorado, June 1987*, ed. by J.W. Gray and A. Scedrov, Contemporary Mathematics, Vol. 92, Amer. Math. Soc., Providence, RI, 1989, 357-369.

[32] Scedrov, A., A guide to polymorphic types, In: *Logic and Computer Science*, P. Odifreddi, editor, 1990.

[33] Seldin, J.P., Mathesis: The Mathematical Foundations of Ulysses, Technical Report RADC-TR-87-223, Odyssey Research Associates, 1987.

[34] Statman, R., Logical Relations and the Typed λ-Calculus, *Information and Control* 65, 3-2, 1985, 85-97.

[35] Stenlund, S., *Combinators, Lambda Terms, and Proof Theory*, D. Reidel, Dordrecht, Holland, 1972.

[36] Tait, W.W., Intensional interpretation of functionals of finite type I, *J. Symbolic Logic* 32, 1967, 198-212.

[37] Tait, W.W., A realizability interpretation of the theory of species, In *Logic Colloquium*, ed. R. Parikh, LNCS No. 453, Springer Verlag, 1975, 240-251.

[38] de Vrijer, R.C., Surjective Pairing and Strong Normalization: Two Themes in Lambda Calculus, Doctoral thesis, Amsterdam, the Netherlands, 1987.

[39] de Vrijer, R.C., Strong Normalization in $N\text{-}HA_p^\omega$, Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen, Series A, Volume 90, No. 4, December 18, 1987, 473-478.

[40] de Vrijer, R.C., Exactly estimating functionals and strong normalization, Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen, Series A, Volume 90, No. 4, December 18, 1987, 479-493.