Weak $\beta\eta$-Normalization and Normalization by Evaluation for System F

Andreas Abel

Department of Computer Science
Ludwig-Maximilians-University Munich

Abstract. A general version of the fundamental theorem for System F is presented which can be instantiated to obtain proofs of weak β- and $\beta\eta$-normalization and normalization by evaluation.

1 Introduction and Related Work

Dependently typed lambda-calculi have been successfully used as proof languages in the proof assistants Agda [Nor07], Coq [INR07], LEGO [Pol94], and NuPrl [Ct86]. Since types may depend on values in these type theories, checking equality of types, which is crucial for type and, thus, proof checking, is non-trivial for these languages, and undecidable in the general case. In extensional type theories, such as the one underlying NuPrl, extensional, hence undecidable, type equality has been kept with the consequence that type checking is undecidable and requires user interaction. In intensional type theories, which are the basis of Agda, Coq, and LEGO, type equality has been restricted to a decidable fragment, called *definitional equality*, hence, type checking is decidable. However, the choice of this fragment strongly influences the comfort in using these systems: the more equal types are recognized as equal automatically, the fewer equality proofs the user has to construct manually.

Definitional equality encompasses at least computational equality, i. e., β, expanding of definitions, and recursion, and exactly like that it is currently implemented in Coq. However, there are suggestions to strengthen definitional equality by rewriting rules [Bla05,CWC07] and decision procedures [BJS07]. On another line, it is strongly desirable to include η, which does not fit well under the rewriting paradigm. For incorporating η, normalization-by-evaluation has proven to be successful. Besides being explored for simple types [BS91,Dan99,ADHS01] it has been extended to polymorphic types [AHS96] and predicative dependent type theories [ACD07]. Still open is its application to impredicative type theories such as the Calculus of Constructions (CoC), probably due to the difficult meta theory of CoC. Our long term goal is to formulate a verified normalization-by-evaluation (NbE) algorithm for impredicative type theories. This work is an important step towards this goal: we construct a generic model for System F whose instances are soundness and completeness proofs for NbE.

Altenkirch, Hofmann, and Streicher [AHS96] already developed a NbE algorithm for System F and proved it correct. However, their work concerns only

a combinatory (λ-free) version of System F. Moreover, they construct an internal model of System F in category theory; their work is only accessible to experts in categories, and the structure of the algorithm is a bit lost among the category-related technical details. They provide a SML-implementation of the algorithm in the appendix, but it is not formally related to the mathematical algorithm in the main text. In an unpublished article [AHS97] they later extend their result to full System F (with λ-abstraction). Deep knowledge of category theory is a preliminary also for this paper, in the words of the authors, a "certain acquaintance with categories of presheaves" is assumed.

In this article, we try to give a more conventional presentation of NbE for System F, presuming only basic knowledge of λ-calculus and System F, domain theory, and inductive definitions in set theory. This way, we hope to make NbE for System F accessible to a wider audience, and to pave the way for an adaption of NbE to impredicative dependent type theories.

Overview. In Sec. 2, we introduce syntax and typing and computation rules for System F. A generic type interpretation is given in Sec. 3, and a generic formulation of the fundamental theorem for System F in Sec. 4. It is then instantiated to yield weak normalization proofs for β (Sec. 5) and $\beta\eta$ (Sec. 6). As main results we obtain soundness and completeness of a NbE algorithm for System F in Sec. 7.

2 Church-style System F

In this section, we briefly recapitulate the syntax and static and dynamic semantics of System F. A more gentle introduction to System F can be found in Pierce's book [Pie02, Ch. 23].

Syntax. Type variables X and term variables x are drawn from two distinct, countable supplies TyVar and Var.

$$
\begin{array}{lll}
\mathsf{Ty} \ni A, B, C ::= X \mid A \to B \mid \forall X A & & \text{types} \\
\mathsf{Tm} \ni r, s, t \quad ::= x \mid \lambda x\!:\!A.\, t \mid r\, s \mid \Lambda X t \mid r\, A & & \text{terms} \\
\mathsf{Cxt} \ni \Gamma, \Delta \quad ::= \diamond \mid \Gamma, x\!:\!A & & \text{typing contexts}
\end{array}
$$

We say context Γ' *extends* Γ, written $\Gamma' \leq \Gamma$, if $\Gamma'(x) = \Gamma(x)$ for all $x \in \mathsf{dom}(\Gamma)$. For instance, assuming $y \notin \mathsf{dom}(\Gamma)$, we have $(\Gamma, y\!:\!A) \leq \Gamma$, but not the other way round. Extending a context extends the set of terms typeable in that context; this theorem is called *weakening*.

Remark 1. The direction of $\leq$ is chosen to be compatible with subtyping. There, we let $\Gamma' \leq \Gamma$ if $\Gamma'(x) \leq \Gamma(x)$ for all $x \in \mathsf{dom}(\Gamma)$. Then $\Gamma \vdash t : A$, $\Gamma' \leq \Gamma$, and $A \leq A'$ imply $\Gamma' \vdash t : A'$ (contravariance!).

A substitution σ is a map from type variables to type expressions and from term variables to term expressions. We write $A\sigma$, $t\sigma$ for the simultaneous execution of substitution σ in A, t. As usual, $\mathsf{FV}(t)$ denotes the set of free type and term variables of t, and $\mathsf{FV}(A)$ the set of free type variables of A. Let $\mathsf{FV}(\Gamma) = \bigcup\{\mathsf{FV}(A) \mid (x\!:\!A) \in \Gamma\}$.

Typing (static semantics) $\Gamma \vdash t : A$.

$$\frac{}{\Gamma \vdash x : \Gamma(x)} \qquad \frac{\Gamma, x{:}A \vdash t : B}{\Gamma \vdash \lambda x{:}A.\, t : A \to B} \qquad \frac{\Gamma \vdash r : A \to B \qquad \Gamma \vdash s : A}{\Gamma \vdash r\, s : B}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \Lambda X t : \forall X A}\ X \notin \mathsf{FV}(\Gamma) \qquad \frac{\Gamma \vdash t : \forall X A}{\Gamma \vdash t\, B : A[B/X]}$$

Operational (dynamic) semantics. One step $\beta\eta$-reduction is the closure of the following axioms under all term constructors:

$$\begin{array}{rlrl} (\lambda x{:}A.\, t)\, s & \longrightarrow_{\beta\eta} t[s/x] & \lambda x{:}A.\, t\, x & \longrightarrow_{\beta\eta} t \quad \text{if } x \notin \mathsf{FV}(t) \\ (\Lambda X t)\, A & \longrightarrow_{\beta\eta} t[A/X] & \Lambda X.\, t\, X & \longrightarrow_{\beta\eta} t \quad \text{if } X \notin \mathsf{FV}(t) \end{array}$$

We denote its reflexive-transitive closure by $\longrightarrow^*_{\beta\eta}$ and its reflexive-transitive-symmetric closure by $=_{\beta\eta}$.

3 Type Interpretation by Kripke Relations

We seek an interpretation of System F's types which is general enough to account for different normalization results. In particular, we are interested in normalization by evaluation, which exists in different flavors. To name a few, Berger and Schwichtenberg [BS91] interpret simple types as set-theoretical function spaces over the base type of term families, Filinski [Fil99] uses continuous function spaces instead, and Altenkirch, Hofmann, and Streicher [AHS96] construct a glueing model of System F types. We choose Abel, Coquand, and Dybjer's approach of *contextual reification* [ACD08], where types are modelled as applicative structures with variables, and reification, i. e., converting semantic objects back to syntax, is context- and type-sensitive. This means that we need to interpret types as Kripke relations, i.e., relations indexed by contexts.

Kripke relations. Given a poset $(S, \subseteq)$, we say F is *Kripke* if $F \in \mathsf{Cxt} \to S$ and antitone, i.e., $\Gamma' \leq \Gamma$ implies $F(\Gamma) \subseteq F(\Gamma')$. We usually write F_Γ for $F(\Gamma)$.

We say D is an *applicative System F structure*, if D^A is Kripke for each $A \in \mathsf{Ty}$ and for all $A, B \in \mathsf{Ty}$, $X \in \mathsf{TyVar}$ and $\Gamma \in \mathsf{Cxt}$ there exist operations

$$\begin{array}{l} \mathsf{app}^{A,B}_\Gamma \in \mathsf{D}^A_\Gamma \to \mathsf{D}^B_\Gamma \to \mathsf{D}^{A\to B}_\Gamma, \\ \mathsf{App}^{X.A}_\Gamma \in \mathsf{D}^{\forall XA}_\Gamma \to (B \in \mathsf{Ty}) \to \mathsf{D}^{A[B/X]}_\Gamma. \end{array}$$

These operations need to be independent of type and context indices, i. e., $\mathsf{app}^{A,B}_\Gamma(f,d) = \mathsf{app}^{A',B'}_{\Gamma'}(f,d)$ if $f \in \mathsf{D}^{A\to B}_\Gamma \cap \mathsf{D}^{A'\to B'}_{\Gamma'}$ and $d \in \mathsf{D}^A_\Gamma \cap \mathsf{D}^{A'}_{\Gamma'}$, and similar for App. Thus, we can introduce an overloaded notation $\_ \cdot \_$ for application by $f \cdot d := \mathsf{app}(f,d)$ and $d \cdot B := \mathsf{App}(d,B)$.

Examples for applicative System F structures are $\mathsf{Tm}^A_\Gamma := \{t \mid \Gamma \vdash t : A\}$, as well as $\mathsf{Tm}^A_\Gamma/{=_{\beta\eta}}$.

Let $\mathsf{D}, \hat{\mathsf{D}}$ be applicative System F structures. We define the set of *Kripke relations* of type A by

$$\mathcal{A} \in \mathbb{K}^A \iff \mathcal{A}_\Gamma \subseteq \mathsf{D}_\Gamma^A \times \hat{\mathsf{D}}_\Gamma^A \text{ for all } \Gamma \text{ and } \mathcal{A} \text{ is Kripke.}$$

We use the letters $\mathcal{A}, \mathcal{B}, \mathcal{C}$ for elements of $\mathbb{K}^A$. We write $\Gamma \vdash d \sim d' \in \mathcal{A}$ for $(d, d') \in \mathcal{A}_\Gamma$. $\mathbb{K}^A$ forms a complete lattice with $\subseteq$, $\cap$, and $\cup$ defined pointwise.

A Kripke relation $\mathcal{A}$ is a *Kripke PER* (partial equivalence relation) if $\mathcal{A}_\Gamma$ is symmetric and transitive for any context Γ, i.e., $\Gamma \vdash d \sim d' \in \mathcal{A}$ implies $\Gamma \vdash d' \sim d \in \mathcal{A}$ and $\Gamma \vdash d_1 \sim d_2 \in \mathcal{A}$ and $\Gamma \vdash d_2 \sim d_3 \in \mathcal{A}$ imply $\Gamma \vdash d_1 \sim d_3 \in \mathcal{A}$.

Constructing Kripke relations. In predicative type theories, such as Martin-Löf Type Theory, one can construct semantical types inductively, i. e., from below [ACD07], without reference to syntax. For impredicative systems, like System F or the Calculus of Constructions, this is not possible. Instead, for each type constructor one has to define a matching operation in the interpretation domain of types, in our case, $\mathbb{K}$, and then define the interpretation of a type by *induction on syntax*, e. g., the size of the type expression [GLT89, Sec. 14.2] or its derivation of wellfoundedness [Str91]. In the following, we provide the necessary constructions to interpret function type and universal type.

Kripke relations are closed under arbitrary intersections. Further constructions on Kripke relations are function space and type abstraction:

$$\begin{aligned}(\mathcal{A} \to \mathcal{B})_\Gamma &= \{(f, f') \mid \Gamma' \vdash d \sim d' \in \mathcal{A} \text{ implies } \Gamma' \vdash f \cdot d \sim f' \cdot d' \in \mathcal{B} \\ &\qquad\qquad \text{for all } d, d', \Gamma' \leq \Gamma\} \\ (A.\mathcal{B})_\Gamma &= \{(d, d') \mid \Gamma \vdash d \cdot A \sim d' \cdot A \in \mathcal{B}\}\end{aligned}$$

The function space $\mathcal{A} \to \mathcal{B}$ is monotone (covariant) in $\mathcal{B}$ and antitone (contravariant) in $\mathcal{A}$. The type abstraction operator $A.\mathcal{B}$ is monotone in $\mathcal{B}$.

Lemma 1. *If $\mathcal{A} \in K^A$ and $\mathcal{B} \in \mathbb{K}^B$ then $\mathcal{A} \to \mathcal{B} \in \mathbb{K}^{A \to B}$, and if $\mathcal{B} \in \mathbb{K}^{B[A/X]}$ then $A.\mathcal{B} \in \mathbb{K}^{\forall XB}$. If $\mathcal{A}, \mathcal{B}$ are Kripke PERs, so are $\mathcal{A} \to \mathcal{B}$ and $A.\mathcal{B}$.*

Interpretation space. Depending on what result one wants to harvest from a model construction for System F, one has to impose restrictions on the interpretation domain of types. For example, in a Tait-style proof of strong normalization using saturated sets, one requires each semantical type to be below the set $\mathcal{S}$ of strongly normalizing terms and above the set $\mathcal{N}$ of neutral strongly normalizing terms. Vaux [Vau04] found an abstraction of $(\mathcal{N}, \mathcal{S})$ which he called *stable pair*. In the following, we present a further generalization which allows the restriction to be dependent on a syntactical type.

Definition 1 (Interpretation space). *An interpretation space consists of two Kripke relations $\underline{A}, \overline{A} \in \mathbb{K}^A$ for each type A such the following conditions hold.*

K-FUN-E	$\underline{A \to B}$	$\subseteq \overline{A} \to \underline{B}$	
K-FUN-I	$\underline{A} \to \overline{B}$	$\subseteq \overline{A \to B}$	
K-ALL-E	$\underline{\forall Y\, A}$	$\subseteq B.\underline{A[B/Y]}$	*for any* B
K-ALL-I	$X.\overline{A[X/Y]}$	$\subseteq \overline{\forall Y\, A}$	*for a new* X

We write $A \Vdash \mathcal{A}$ (pronounced A realizes *$\mathcal{A}$) if $\underline{A} \subseteq \mathcal{A} \subseteq \overline{A}$.*

A trivial, not type-sensitive interpretation space is $\underline{A}_\Gamma = \mathcal{N}$ and $\overline{A}_\Gamma = \mathcal{S}$. In this case, $A \Vdash \mathcal{A}$ just means $\mathcal{A}$ is saturated. An analogy of $A \Vdash \mathcal{A}$ can be found in Matthes' proof of strong normalization of System F [Mat98, Sec. 9.1.2] where it means $\mathcal{A}$ *is A-saturated*. More examples of interpretation spaces can be found in this article.

In the following, we assume an interpretation space. We now introduce the last construction on semantical types, quantification, which is relative to an interpretation space. If $\mathcal{F}(B) \in \mathbb{K}^B \to \mathbb{K}^{A[B/X]}$ for all $B \in \mathsf{Ty}$, we let

$$\bigcap \mathcal{F} = \bigcap \{B.\mathcal{F}(B, \mathcal{B}) \mid B \Vdash \mathcal{B}\} \in \mathbb{K}^{\forall X A}.$$

Intersection is restricted to realizable semantical types. This has an analogue in other normalization proofs of System F, e. g., Girard [GLT89, Ch. 14] restricts quantification to reducibility candidates. The type constructions preserve realizability, thanks to the conditions imposed by Def. 1.

Lemma 2 (Realizability of type constructions).

1. *If $A \Vdash \mathcal{A}$ and $B \Vdash \mathcal{B}$ then $A \to B \Vdash \mathcal{A} \to \mathcal{B}$.*
2. *If $A[B/Y] \Vdash \mathcal{F}(B, \mathcal{B})$ for all $B \Vdash \mathcal{B}$, then $\forall Y\, A \Vdash \bigcap \mathcal{F}$.*

Proof. Directly, using the postulates on $\underline{C}$ and $\overline{C}$.

1. First, we have $\underline{A \to B} \subseteq \overline{A} \to \underline{B}$ by K-FUN-E, and since $\mathcal{A} \subseteq \overline{A}$ and $\underline{B} \subseteq \mathcal{B}$ we obtain by contravariance of the function space $\underline{A \to B} \subseteq \mathcal{A} \to \mathcal{B}$.
   Secondly, since $\underline{A} \subseteq \mathcal{A}$ and $\mathcal{B} \subseteq \overline{B}$, contravariance yields $\mathcal{A} \to \mathcal{B} \subseteq \underline{A} \to \overline{B} \subseteq \overline{A \to B}$ by K-FUN-I.
2. First, for any $B \Vdash \mathcal{B}$ we have $A[B/Y] \subseteq \mathcal{F}(B, \mathcal{B})$, thus by monotonicity of the abstraction operator, $B.\overline{A[B/Y]} \subseteq B.\mathcal{F}(B, \mathcal{B})$. By K-ALL-E it follows $\underline{\forall Y\, A} \subseteq B.\mathcal{F}(B, \mathcal{B})$, and since $B, \mathcal{B}$ were arbitrary, $\underline{\forall Y\, A} \subseteq \bigcap \mathcal{F}$.
   Secondly, let X be a new type variable. We have $X \Vdash \underline{X}$, hence, $A[X/Y] \Vdash \mathcal{F}(X, \underline{X})$ which implies $\mathcal{F}(X, \underline{X}) \subseteq \overline{A[X/Y]}$. Thus, $\bigcap \mathcal{F} \subseteq X.\mathcal{F}(X, \underline{X}) \subseteq X.\overline{A[X/Y]} \subseteq \overline{\forall Y\, A}$ by K-ALL-I.

Type interpretation can now be defined mechanically, mapping the syntactic type constructors to the semantic ones. Let σ be a syntactical type substitution and $\rho(X) \in \mathbb{K}^{\sigma(X)}$ for all type variables X, which we write $\rho \in \mathbb{K}^\sigma$. We define $[\![A]\!]_\rho \in \mathbb{K}^{A\sigma}$ by the following equations.

$$\begin{aligned} [\![X]\!]_\rho &= \rho(X) \\ [\![A \to B]\!]_\rho &= [\![A]\!]_\rho \to [\![B]\!]_\rho \\ [\![\forall X A]\!]_\rho &= \bigcap((B \in \mathsf{Ty}) \mapsto (\mathcal{B} \in \mathbb{K}^B) \mapsto [\![A]\!]_{\rho[X \mapsto \mathcal{B}]}) \end{aligned}$$

Note that $\rho[X \mapsto \mathcal{B}] \in \mathbb{K}^{\sigma[X \mapsto B]}$ in the last line.

Lemma 3. *If $\rho(X)$ is a Kripke PER for all X, so is $[\![A]\!]_\rho$.*

Lemma 4 (Substitution). $[\![A[B/X]]\!]_\rho = [\![A]\!]_{\rho[X \mapsto [\![B]\!]_\rho]}$.

If $\rho \in \mathbb{K}^\sigma$, we let $\sigma \Vdash \rho$ if $\sigma(X) \Vdash \rho(X)$ for all type variables X. It is now easy to show that types realize their own interpretations.

Theorem 1 (Type interpretation is realizable). *If $\sigma \Vdash \rho$ then $A\sigma \Vdash [\![A]\!]_\rho$.*

Corollary 1. $\underline{A} \subseteq \overline{A}$.

4 Fundamental Lemma

A general model of System F can be given by interpreting terms in a applicative System F structure D and types as PERs over D. The fundamental theorem does not rely on types being PERs, thus, we can easily take Kripke relations instead.

Let η map type variables to syntactical types and term variables to elements of D such that $\eta \in \mathsf{D}^\Gamma_\Delta$, meaning $\eta(x) \in \mathsf{D}^B_\Delta$ for all $(x\!:\!B) \in \Gamma$. Let $t \in \mathsf{Tm}^A_\Gamma$. We stipulate the existence of an evaluation function $(\!|t|\!)_\eta \in \mathsf{D}^A_\Delta$ with the following properties.

DEN-VAR	$(\!|x|\!)_\eta = \eta(x)$	
DEN-FUN-E	$(\!|r\ s|\!)_\eta = (\!|r|\!)_\eta \cdot (\!|s|\!)_\eta$	
DEN-ALL-E	$(\!|r\ A|\!)_\eta = (\!|r|\!)_\eta \cdot A\eta$	
DEN-FUN-I	$(\!|\lambda x\!:\!A.\, t|\!)_\eta \cdot d = (\!|t|\!)_{\eta[x \mapsto d]}$	if $d \in \mathsf{D}^A_\Delta$
DEN-ALL-I	$(\!|\Lambda X t|\!)_\eta \cdot A = (\!|t|\!)_{\eta[X \mapsto A]}$	

We call $(\mathsf{D}, \_\cdot\_, (\!|\_|\!)\_)$ a *syntactical applicative System F structure* (cf. Barendregt [Bar84, 5.3.1]).

Now we are ready to show the fundamental theorem. We define

$$\Delta \vdash \eta \sim \eta' \in [\![\Gamma]\!]_\rho \iff \Delta \vdash \eta(x) \sim \eta'(x) \in [\![\Gamma(x)]\!]_\rho \text{ for all } x \in \mathsf{dom}(\Gamma).$$

Theorem 2 (Validity of typing). *Let $\eta \Vdash \rho$ and both $\eta \restriction \mathsf{TyVar} = \eta' \restriction \mathsf{TyVar}$ and $\Delta \vdash \eta \sim \eta' \in [\![\Gamma]\!]_\rho$. If $\Gamma \vdash t : A$ then $\Delta \vdash (\!|t|\!)_\eta \sim (\!|t|\!)_{\eta'} \in [\![A]\!]_\rho$.*

Proof. By induction on $\Gamma \vdash t : A$. Interesting are the System F specific cases.

Case

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \Lambda X t : \forall X A}\ X \notin \mathsf{FV}(\Gamma)$$

$B \Vdash \mathcal{B}$	assumption
$\eta[X \mapsto B] \Vdash \rho[X \mapsto \mathcal{B}] =: \rho'$	by def.
$\Delta \vdash \eta[X \mapsto B] \sim \eta'[X \mapsto B] \in [\![\Gamma]\!]_\rho = [\![\Gamma]\!]_{\rho'}$	since $X \notin \mathsf{FV}(\Gamma)$
$\Delta \vdash (\!|t|\!)_{\eta[X\mapsto B]} \sim (\!|t|\!)_{\eta'[X\mapsto B]} \in [\![A]\!]_{\rho'}$	by ind. hyp.
$\Delta \vdash (\!|\Lambda X t|\!)_\eta \cdot B \sim (\!|\Lambda X t|\!)_{\eta'} \cdot B \in [\![A]\!]_{\rho'}$	DEN-ALL-I
$\Delta \vdash (\!|\Lambda X t|\!)_\eta \sim (\!|\Lambda X t|\!)_{\eta'} \in B.[\![A]\!]_{\rho'}$	by def.
$\Delta \vdash (\!|\Lambda X t|\!)_\eta \sim (\!|\Lambda X t|\!)_{\eta'} \in [\![\forall X A]\!]_\rho$	since $B \Vdash \mathcal{B}$ arbitrary

Case

$$\frac{\Gamma \vdash t : \forall X A}{\Gamma \vdash t\, B : A[B/X]}$$

$\Delta \vdash (\!|t|\!)_\eta \sim (\!|t|\!)_{\eta'} \in \bigcap(\mathcal{X} \mapsto [\![A]\!]_{\rho[X\mapsto\mathcal{X}]})$	by ind.hyp.
$B\eta \Vdash [\![B]\!]_\rho$	by Thm. 1
$\Delta \vdash (\!|t|\!)_\eta \cdot B\eta \sim (\!|t|\!)_{\eta'} \cdot B\eta \in [\![A]\!]_{\rho[X\mapsto[\![B]\!]_\rho]}$	by instantiation
$\Delta \vdash (\!|t\, B|\!)_\eta \sim (\!|t\, B|\!)_{\eta'} \in [\![A]\!]_{\rho[X\mapsto[\![B]\!]_\rho]}$	DEN-ALL-E
$\Delta \vdash (\!|t\, B|\!)_\eta \sim (\!|t\, B|\!)_{\eta'} \in [\![A[B/X]]\!]_\rho$	Substitution

5 Weak β-Normalization

From our general fundamental theorem we can recover a proof of β-normalization for System F. In this case, we construct an untyped interpretation space of discrete Kripke relations which ignores types and contexts. The following development is standard, we show here only that it fits into the abstractions we have chosen in sections 3 and 4.

Let $\overline{r}$ denote the β-equivalence class of r. Let $\mathsf{D} = \mathsf{D}' = \mathsf{Tm}/{=_\beta}$ with application defined by $\overline{r} \cdot \overline{s} = \overline{r\, s}$ and $\overline{r} \cdot A = \overline{r\, A}$. Evaluation is defined by $(\!|t|\!)_{\overline{\sigma}} = \overline{t\sigma}$.

Lemma 5. *Application and evaluation are well-defined and* $(\mathsf{D}, \_ \cdot \_, (\!|\_|\!)\_)$ *forms a syntactical applicative System F structure.*

Neutral terms are given by the grammar $n ::= x \mid n\, s \mid n\, A$. The interpretation space for β-normalization is untyped, i. e., we set

$$\overline{A}_\Gamma = \mathcal{W}_\Gamma := \{(\overline{t}, \overline{t}) \mid t \text{ has a } \beta\text{-normal form } \}$$
$$\underline{A}_\Gamma = \mathcal{N}_\Gamma := \{(\overline{n}, \overline{n}) \mid n \text{ has a } \beta\text{-normal form } \}$$

for all types A. To show that these settings really constitute an interpretation space is a standard exercise.

Let η_{id} be the identity map on term and type variables, and let $\rho_{\mathsf{id}}(X) = \mathcal{N}$ for all $X \in \mathsf{TyVar}$. Clearly, $\eta_{\mathsf{id}} \Vdash \rho_{\mathsf{id}}$.

Lemma 6 (Identity environment). $\Gamma \vdash \overline{\eta_{\mathsf{id}}} \sim \overline{\eta_{\mathsf{id}}} \in [\![\Gamma]\!]_{\rho_{\mathsf{id}}}$.

Theorem 3 (Weak β-normalization of System F). *If $\Gamma \vdash t : A$ then t has a β-normal form.*

Proof. By the fundamental theorem $\Gamma \vdash (\!|t|\!)_{\overline{\eta_{\mathsf{id}}}} \sim (\!|t|\!)_{\overline{\eta_{\mathsf{id}}}} \in [\![A]\!]_{\rho_{\mathsf{id}}}$ which implies $\Gamma \vdash \bar{t} \sim \bar{t} \in \mathcal{W}$, hence, t has a β-normal form.

6 Weak $\beta\eta$-Normalization

In this section, we instantiate Thm. 2 to prove weak $\beta\eta$-normalization for System F. In particular, we show that each well-typed term has a η-long β-normal form. In this, we will require the Kripke aspect of our type interpretation, since *being η-long* for open terms can only be defined in the presence of a typing context.

Let now $\overline{r}$ denote the $\beta\eta$-equivalence class of r and set $\mathsf{D} = \mathsf{D}' = \mathsf{Tm}/=_{\beta\eta}$. Again, D, with application and evaluation defined as in the last section, constitutes a syntactical applicative System F structure.

Long normal forms are characterized by the two mutually defined judgments

$$\begin{array}{ll} \Gamma \vdash t \Uparrow A & t \text{ is a long normal form of type } A \\ \Gamma \vdash t \Downarrow A & t \text{ is a neutral long normal form of type } A \end{array}$$

given by the following rules:

$$\frac{}{\Gamma \vdash x \Downarrow \Gamma(x)} \qquad \frac{\Gamma \vdash r \Downarrow A \to B \qquad \Gamma \vdash s \Uparrow A}{\Gamma \vdash r\, s \Downarrow B} \qquad \frac{\Gamma \vdash r \Downarrow \forall X A}{\Gamma \vdash rB \Downarrow A[B/X]}$$

$$\frac{\Gamma \vdash r \Downarrow X}{\Gamma \vdash r \Uparrow X} \qquad \frac{\Gamma, x\!:\!A \vdash t \Uparrow B}{\Gamma \vdash \lambda x\!:\!A.\, t \Uparrow A \to B} \qquad \frac{\Gamma \vdash t \Uparrow A}{\Gamma \vdash \Lambda X t \Uparrow \forall X A}\ X \notin \mathsf{FV}(\Gamma)$$

From the model construction of System F we want to harvest that each well-typed term has a η-long β-normal form. Thus, we define an interpretation space by setting

$$\begin{array}{ll} \Gamma \vdash d \sim d' \in \overline{A} \iff & \text{exists } r \text{ with } d = d' = \overline{r} \text{ and } \Gamma \vdash r \Uparrow A, \\ \Gamma \vdash d \sim d' \in \underline{A} \iff & \text{exists } r \text{ with } d = d' = \overline{r} \text{ and } \Gamma \vdash r \Downarrow A. \end{array}$$

Lemma 7 (Weakening). $\underline{A}, \overline{A} \in \mathbb{K}^A$.

Indeed, $\underline{A}$ and $\overline{A}$ span an interpretation space, which we will prove in detail in the following.

Lemma 8 (Interpretation space).

K-FUN-E	$\underline{A \to B}$	$\subseteq \overline{A} \to \underline{B}$	
K-FUN-I	$\underline{A} \to \overline{B}$	$\subseteq \overline{A \to B}$	
K-ALL-E	$\underline{\forall X A}$	$\subseteq B.\underline{A[B/X]}$	
K-ALL-I	$X.\overline{A[X/Y]}$	$\subseteq \overline{\forall Y A}$	*for a new* X

Proof. K-FUN-E $\underline{A \to B} \subseteq \overline{A} \to \underline{B}$

$\Gamma \vdash f \sim f' \in \underline{A \to B}$	by hyp.
$f = f' = \overline{r}$ and $\Gamma \vdash r \Downarrow A \to B$	by def.
$\Gamma' \leq \Gamma$ and $\Gamma' \vdash d \sim d' \in \overline{A}$	assumption
$d = d' = \overline{s}$ and $\Gamma' \vdash s \Uparrow A$	by def.
$f \cdot d = f' \cdot d' = \overline{r\,s}$	def. of application
$\Gamma' \vdash r\,s \Downarrow B$	rule, Lemma 7
$\Gamma' \vdash f \cdot d \sim f' \cdot d' \in \underline{B}$	by def.
$\Gamma \vdash f \sim f' \in \overline{A} \to \underline{B}$	since Γ', d, d' arb.

K-FUN-I $\underline{A} \to \overline{B} \subseteq \overline{A \to B}$

$\Gamma \vdash \overline{r} \sim \overline{r'} \in \underline{A} \to \overline{B}$	by hyp.
$\Gamma, x\!:\!A \vdash x \Downarrow A$	rule
$\Gamma, x\!:\!A \vdash \overline{x} \sim \overline{x} \in \underline{A}$	by def.
$\Gamma, x\!:\!A \vdash \overline{r} \cdot \overline{x} \sim \overline{r'} \cdot \overline{x} \in \overline{B}$	by def. $\to$
$\overline{r\,x} = \overline{r'\,x} = \overline{t}$ and $\Gamma, x\!:\!A \vdash t \Uparrow B$	by def.
$\overline{r} = \overline{\lambda x\!:\!A.\, r\,x} = \overline{r'} = \overline{\lambda x\!:\!A.\, r'\,x} = \overline{\lambda x\!:\!A.\, t}$	η
$\Gamma \vdash \lambda x\!:\!A.\, t \Uparrow A \to B$	rule
$\Gamma \vdash \overline{r} \sim \overline{r'} \in \overline{A \to B}$	by def.

K-ALL-E $\underline{\forall X A} \subseteq B.\underline{A[B/X]}$

$\Gamma \vdash d \sim d' \in \underline{\forall X A}$	assumption
$d = d' = \overline{r}$ and $\Gamma \vdash r \Downarrow \forall X A$	by def.
$d \cdot B = d' \cdot B = \overline{r\,B}$ and $\Gamma \vdash r\,B \Downarrow A[B/X]$	rule, app.
$\Gamma \vdash d \cdot B \sim d' \cdot B \in \underline{A[B/X]}$	by def.
$\Gamma \vdash d \sim d' \in B.\underline{A[B/X]}$	by def. $B.\mathcal{A}$

K-ALL-I $X.\overline{A[X/Y]} \subseteq \overline{\forall Y\, A}$ for a new X.

$\Gamma \vdash \overline{r} \sim \overline{r'} \in X.\overline{A[X/Y]}$	assumption
$\Gamma \vdash \overline{r} \cdot X \sim \overline{r'} \cdot X \in \overline{A[X/Y]}$	by def. $X.\mathcal{A}$
$\overline{r\,X} = \overline{r'\,X} = \overline{t}$ and $\Gamma \vdash t \Uparrow A[X/Y]$	by def.
$\overline{r} = \overline{\Lambda X.\, r\,X} = \overline{r'} = \overline{\Lambda X.\, r'\,X} = \overline{\Lambda X t}$	η
$X \notin \mathsf{FV}(\Gamma)$	since X new
$\Gamma \vdash \Lambda X t \Uparrow \forall X.\, A[X/Y] = \forall Y\, A$	rule
$\Gamma \vdash \overline{r} \sim \overline{r'} \in \overline{\forall Y\, A}$	by def.

The rest is just an application of the fundamental theorem. Recall that η_{id} is the identity map, and let this time $\rho_{\mathsf{id}}(X) = \underline{X}$ for all $X \in \mathsf{TyVar}$. Again, $\eta_{\mathsf{id}} \Vdash \rho_{\mathsf{id}}$, and $\Gamma \vdash \overline{\eta_{\mathsf{id}}} \sim \overline{\eta_{\mathsf{id}}} \in [\![\Gamma]\!]_{\rho_{\mathsf{id}}}$.

Theorem 4 (Weak $\beta\eta$-normalization of System F). *If $\Gamma \vdash t : A$ then t β-reduces η-expands to a long normal form t'.*

Proof. Clearly, $A \Vdash [\![A]\!]_{\rho_{\mathsf{id}}}$. By Thm. 2, $\Gamma \vdash (\!|t|\!)_{\overline{\eta_{\mathsf{id}}}} \sim (\!|t|\!)_{\overline{\eta_{\mathsf{id}}}} \in [\![A]\!]_{\rho_{\mathsf{id}}}$, meaning $t =_{\beta\eta} t'$ with $\Gamma \vdash t' \Uparrow A$. We conclude by Church-Rosser for β-reduction η-expansion.

7 Normalization by Evaluation

In this section we now define a normalization function which maps exactly the $\beta\eta$-equal terms of the same type to the same η-long β-normal form. To this end, we define judgmental $\beta\eta$-equality $\Gamma \vdash t = t' : A$ and the function $\mathsf{nf}(\Gamma \vdash t : A)$ such that it is

1. complete for judgmental equality, i. e., $\Gamma \vdash t = t' : A$ implies $\mathsf{nf}(\Gamma \vdash t : A) \equiv \mathsf{nf}(\Gamma \vdash t' : A)$, and
2. sound, i. e., if $\Gamma \vdash t : A$ then $\Gamma \vdash t = \mathsf{nf}(\Gamma \vdash t : A) : A$.

7.1 Judgmental Equality

Judgmental $\beta\eta$-equality $\Gamma \vdash t = t' : A$ is defined inductively by the following axiom, plus congruence rules and equivalence rules (reflexivity, symmetry, and transitivity).

$$\frac{\Gamma, x : A \vdash t : B \qquad \Gamma \vdash s : A}{\Gamma \vdash (\lambda x : A.\, t)\, s = t[s/x] : B} \qquad \frac{\Gamma \vdash t : A \to B}{\Gamma \vdash \lambda x : A.\, t\, x = t : A \to B}\; x \notin \mathsf{FV}(t)$$

$$\frac{\Gamma \vdash t : A \qquad X \notin \mathsf{FV}(\Gamma)}{\Gamma \vdash (\Lambda X t)\, B = t[B/X] : A[B/X]} \qquad \frac{\Gamma \vdash t : \forall X A}{\Gamma \vdash \Lambda X.\, t\, X = t : \forall X A}\; X \notin \mathsf{FV}(t)$$

A fundamental theorem for judgmental equality is of course not valid for arbitrary *Kripke relations*, it can only be shown for *Kripke PERs*, since symmetry and transitivity have to be modeled. Also, to model the above equations, the evaluation function must satisfy additional laws:

DEN-SUBST	$(\!\|t[s/x]\|\!)_\eta = (\!\|t\|\!)_{\eta[x\mapsto(\!\|s\|\!)_\eta]}$	if $\Gamma, x\!:\!A \vdash t : B$ and $\Gamma \vdash s : A$
DEN-TY-SUBST	$(\!\|t[A/X]\|\!)_\eta = (\!\|t\|\!)_{\eta[X\mapsto A\eta]}$	if $\Gamma \vdash t : B$ and $X \notin \mathsf{FV}(\Gamma)$
DEN-IRR	$(\!\|t\|\!)_\eta = (\!\|t\|\!)_{\eta'}$	if $\eta(x) = \eta'(x)$ for all $x \in \mathsf{FV}(t)$

If these laws are satisfied, $(\mathsf{D}, \_\cdot\_, (\!|\_|\!)\_)$ is called a *syntactical combinatorial System F algebra* (cf. [ACD07]).

For the following theorem, consider an interpretation space of Kripke-PERs over a syntactical combinatorial algebra.

Theorem 5 (Validity of equality). *Let $\sigma \Vdash \rho$ and $\Delta \vdash \eta \sim \eta' \in [\![\Gamma]\!]_\rho$. If $\Gamma \vdash t = t' : A$ then $\Delta \vdash (\!|t|\!)_\eta \sim (\!|t'|\!)_{\eta'} \in [\![A]\!]_\rho$.*

Proof. By induction on $\Gamma \vdash t = t' : A$.

7.2 The normalization algorithm

Normalization by evaluation consist of two steps: first, evaluate the term in the identity environment, obtaining a semantic object, and then, reify the semantic object back to syntax, yielding a long normal form.

Evaluation. For the evaluation we need a combinatorial algebra Val with computable application and evaluation and with variables. One possibility is to let Val be the solution of the recursive domain equation:

$$\mathsf{Val} = (\mathsf{Var} \times (\mathsf{Val} \cup \mathsf{Ty})^{<\omega}) \oplus [\mathsf{Val} \to \mathsf{Val}] \oplus (\mathsf{Ty} \to \mathsf{Val}).$$

Then a semantic object $d \in \mathsf{Val}$ is either a neutral object e of the shape $e ::= x \mid e\,d \mid e\,A$, a continuous function $f \in [\mathsf{Val} \to \mathsf{Val}]$ on semantic objects or a function $F \in \mathsf{Ty} \to \mathsf{Val}$ from types to semantic objects.

Let $\mathsf{D}^A_\Gamma = \mathsf{Val}$ for all Γ, A. Application is defined by

$$\begin{array}{ll} e \cdot d = e\,d & e \cdot A = e\,A \\ f \cdot d = f(d) & F \cdot A = F(A) \end{array}$$

and evaluation by DEN-VAR, DEN-FUN-E, DEN-ALL-E and

$$\begin{array}{ll} (\!|\lambda x\!:\!A.\, t|\!)_\eta(d) = (\!|t|\!)_{\eta[x\mapsto d]} \\ (\!|\Lambda X t|\!)_\eta(A) \quad = (\!|t|\!)_{\eta[X\mapsto A]}. \end{array}$$

It is easy to check that $(\mathsf{D}, \_\cdot\_, (\!|\_|\!)\_)$ forms a syntactical combinatorial System F algebra.

Reification converts semantic object back to expressions. Functions are reified by applying them to fresh variables. This is of course only possible if Val contains the variables.

We adapt *contextual reification* [ACD08] to System F and define inductively the mutual judgements

$$\begin{array}{ll} \Gamma \vdash d \searrow t \Uparrow A & d \text{ reifies to } t \text{ at type } A, \\ \Gamma \vdash d \searrow t \Downarrow A & d \text{ reifies to } t, \text{ inferring type } A. \end{array}$$

by the following rules

$$\frac{}{\Gamma \vdash x \searrow x \Downarrow \Gamma(x)}$$

$$\frac{\Gamma \vdash e \searrow r \Downarrow A \to B \qquad \Gamma \vdash d \searrow s \Uparrow A}{\Gamma \vdash e\, d \searrow r\, s \Downarrow B} \qquad]\frac{\Gamma \vdash e \searrow r \Downarrow \forall X A}{\Gamma \vdash e\, B \searrow r\, B \Downarrow A[B/X]}$$

$$\frac{\Gamma \vdash e \searrow r \Downarrow X}{\Gamma \vdash e \searrow r \Uparrow X}$$

$$\frac{\Gamma, x{:}A \vdash f \cdot x \searrow t \Uparrow B}{\Gamma \vdash f \searrow \lambda x{:}A.\, t \Uparrow A \to B} \qquad \frac{\Gamma \vdash F \cdot X \searrow t \Uparrow A}{\Gamma \vdash F \searrow \Lambda X t \Uparrow \forall X A}\ X \notin \mathsf{FV}(\Gamma)$$

These rules can be interpreted computationally by considering them clauses of a logic program. Both judgements take Γ and d as input and return t. In the type-directed mode $\Uparrow$, type A is input, and in the inference mode $\Downarrow$, type A is output. It is easy to check that the associated logic program is well-moded and returns a long normal form. Termination, however, will follow from the fundamental theorem.

Lemma 9 (Weakening). *Let $\Gamma' \leq \Gamma$.*

1. *If $\Gamma \vdash e \searrow r \Downarrow A$ then $\Gamma' \vdash e \searrow r \Downarrow A$.*
2. *If $\Gamma \vdash d \searrow t \Uparrow A$ then $\Gamma' \vdash d \searrow t \Uparrow A$.*

The normalization function $\mathsf{nf}(\Gamma \vdash t : A)$ can now be defined to yield the t' such that $\Gamma \vdash (\!|t|\!)_{\eta_{\mathsf{id}}} \searrow t' \Uparrow A$.

7.3 Completeness of NbE

We obtain completeness (and termination) of the normalization function as instance of the fundamental theorem for judgmental equality. Let an interpretation space be defined by

$$\begin{array}{ll} \Gamma \vdash d \sim d' \in \overline{A} \iff & \text{exists } t \text{ with } \Gamma \vdash d \searrow t \Uparrow A \text{ and } \Gamma \vdash d' \searrow t \Uparrow A, \\ \Gamma \vdash d \sim d' \in \underline{A} \iff & \text{exists } t \text{ with } \Gamma \vdash d \searrow t \Downarrow A \text{ and } \Gamma \vdash d' \searrow t \Downarrow A. \end{array}$$

Lemma 10 (Interpretation space). *$\underline{A}, \overline{A}$ form an interpretation space of Kripke PERs.*

Proof. Analogous to Lemma 8.

Theorem 6 (Completeness of NbE). *If* $\Gamma \vdash t = t' : A$ *then* $\Gamma \vdash (\!|t|\!)_{\eta_{\mathsf{id}}} \searrow r \Uparrow A$ *and* $\Gamma \vdash (\!|t'|\!)_{\eta_{\mathsf{id}}} \searrow r \Uparrow A$ *for some long normal form* r*.*

Proof. Since $\Gamma \vdash \eta_{\mathsf{id}} \sim \eta_{\mathsf{id}} \in [\![\Gamma]\!]_{\rho_{\mathsf{id}}}$, by Thm. 5 $\Gamma \vdash (\!|t|\!)_{\eta_{\mathsf{id}}} \sim (\!|t'|\!)_{\eta_{\mathsf{id}}} \in \overline{A}$.

7.4 Soundness of NbE

Soundness (and termination) of the normalization function is a consequence of the fundamental theorem for typing. We set $\hat{\mathsf{D}}^A_\Gamma = \mathsf{Tm}^A_\Gamma / \Gamma \vdash \_ = \_ : A$, i.e., terms modulo judgmental equality, and let

$$\begin{aligned}
\Gamma \vdash d \sim \bar{t} \in \overline{A} &\iff \text{exists } t' \text{ with } \Gamma \vdash d \searrow t' \Uparrow A \text{ and } \Gamma \vdash t = t' : A,\\
\Gamma \vdash d \sim \bar{t} \in \underline{A} &\iff \text{exists } t' \text{ with } \Gamma \vdash d \searrow t' \Downarrow A \text{ and } \Gamma \vdash t = t' : A.
\end{aligned}$$

Lemma 11 (Interpretation space). $\underline{A}, \overline{A}$ *form an interpretation space.*

Theorem 7 (Soundness of NbE). *If* $\Gamma \vdash t : A$ *then* $\Gamma \vdash (\!|t|\!)_{\eta_{\mathsf{id}}} \searrow t' \Uparrow A$ *and* $\Gamma \vdash t = t' : A$*.*

Proof. For all $(x : A) \in \Gamma$, it holds that $\Gamma \vdash x \sim \overline{x} \in \underline{A}$, hence, $\Gamma \vdash \eta_{\mathsf{id}}(x) \sim \overline{\eta_{\mathsf{id}}}(x) \in [\![A]\!]_{\rho_{\mathsf{id}}}$, thus, $\Gamma \vdash \eta_{\mathsf{id}} \sim \overline{\eta_{\mathsf{id}}} \in [\![\Gamma]\!]_{\rho_{\mathsf{id}}}$. By Thm. 2, $\Gamma \vdash (\!|t|\!)_{\eta_{\mathsf{id}}} \sim \bar{t} \in A$.

Summarizing this section, we have obtained a $\beta\eta$-normalization function for System F which is complete and sound for judgmental equality.

8 Conclusion

We have introduced the concept of type *interpretation space* and proven generic fundamental theorems for typing and judgmental equality in System F. As instances, we obtained proofs of weak normalization for β and $\beta\eta$, and proofs of soundness and completeness for a normalization-by-evaluation algorithm based on contextual reification.

Further work. We seek to extend this work to type theories with non-trivial equality on the type level, like System F^ω and the Calculus of Constructions.

Acknowledgments. This work was carried out during a visit to Frédéric Blanqui and Cody Roux at LORIA, Nancy, France, financed by the *Bayrisch-Französisches Hochschulzentrum*. My gratitude extends also to Thierry Coquand and Peter Dybjer for discussions on the topic, and to the anonymous referees for their suggestions which helped to improve this paper.

References

[ACD07] Andreas Abel, Thierry Coquand, and Peter Dybjer. Normalization by evaluation for Martin-Löf Type Theory with typed equality judgements. In *Proc. of the 22nd IEEE Symp. on Logic in Computer Science (LICS 2007)*, pages 3–12. IEEE Computer Soc. Press, 2007.

[ACD08] Andreas Abel, Thierry Coquand, and Peter Dybjer. Verifying a semantic $\beta\eta$-conversion test for Martin-Löf type theory. In *Mathematics of Program Construction, MPC'08*, volume 5133 of *Lect. Notes in Comput. Sci.* Springer-Verlag, 2008.

[ADHS01] Thorsten Altenkirch, Peter Dybjer, Martin Hofmann, and Philip J. Scott. Normalization by evaluation for typed lambda calculus with coproducts. In *Proc. of the 16th IEEE Symp. on Logic in Computer Science (LICS 2001)*, pages 303–310. IEEE Computer Soc. Press, 2001.

[AHS96] Thorsten Altenkirch, Martin Hofmann, and Thomas Streicher. Reduction-free normalisation for a polymorphic system. In *Proc. of the 11th IEEE Symp. on Logic in Computer Science (LICS'96)*, pages 98–106. IEEE Computer Soc. Press, 1996.

[AHS97] Thorsten Altenkirch, Martin Hofmann, and Thomas Streicher. Reduction-free normalisation for System F. available from `http://www.cs.nott.ac.uk/~txa/publ/f97.pdf`, 1997.

[Bar84] Henk Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. North Holland, Amsterdam, 1984.

[BJS07] Frédéric Blanqui, Jean-Pierre Jouannaud, and Pierre-Yves Strub. Building decision procedures in the Calculus of Inductive Constructions. In Jacques Duparc and Thomas A. Henzinger, editors, *Computer Science Logic, 21th Int. Wksh., CSL 2007, 16th Annual Conf. of the EACSL*, volume 4646 of *Lect. Notes in Comput. Sci.*, pages 328–342. Springer-Verlag, 2007.

[Bla05] Frédéric Blanqui. Definitions by rewriting in the calculus of constructions. *Mathematical Structures in Computer Science*, 15(1):37–92, 2005.

[BS91] Ulrich Berger and Helmut Schwichtenberg. An inverse to the evaluation functional for typed λ-calculus. In *Proc. of the 6th IEEE Symp. on Logic in Computer Science (LICS'91)*, pages 203–211. IEEE Computer Soc. Press, 1991.

[Ct86] Robert Constable and team. *Implementing Mathematics with the Nuprl Proof Development System*. Prentice Hall, 1986.

[CWC07] Jacek Chrzaszcz and Daria Walukiewicz-Chrzaszcz. Towards rewriting in Coq. In Hubert Comon-Lundh, Claude Kirchner, and Hélène Kirchner, editors, *Rewriting, Computation and Proof, Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of His 60th Birthday*, volume 4600 of *Lect. Notes in Comput. Sci.*, pages 113–131. Springer-Verlag, 2007.

[Dan99] Olivier Danvy. Type-directed partial evaluation. In John Hatcliff, Torben Æ. Mogensen, and Peter Thiemann, editors, *Partial Evaluation – Practice and Theory, DIKU 1998 International Summer School, Copenhagen, Denmark, June 29 - July 10, 1998*, volume 1706 of *Lect. Notes in Comput. Sci.*, pages 367–411. Springer-Verlag, 1999.

[Fil99] Andrzej Filinski. A semantic account of type-directed partial evaluation. In Gopalan Nadathur, editor, *Proc. of the Int. Conf. on Principles and Practice of Declarative Programming, PPDP'99*, volume 1702 of *Lect. Notes in Comput. Sci.*, pages 378–395. Springer-Verlag, 1999.

[GLT89] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989.

[INR07] INRIA. *The Coq Proof Assistant, Version 8.1*. INRIA, 2007. http://coq.inria.fr/.

[Mat98] Ralph Matthes. *Extensions of System F by Iteration and Primitive Recursion on Monotone Inductive Types*. PhD thesis, Ludwig-Maximilians-University, May 1998.

[Nor07] Ulf Norell. *Towards a practical programming language based on dependent type theory*. PhD thesis, Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, Sweden, September 2007.

[Pie02] Benjamin C. Pierce. *Types and Programming Languages*. MIT Press, 2002.

[Pol94] Randy Pollack. *The Theory of LEGO*. PhD thesis, University of Edinburgh, 1994.

[Str91] Thomas Streicher. *Semantics of Type Theory*. Progress in Theoretical Computer Science. Birkhaeuser Verlag, Basel, 1991.

[Vau04] Lionel Vaux. A type system with implicit types. English version of his mémoire de maîtrise, June 2004.