Termination of system F-bounded (Note)

Giorgio Ghelli[1]

1 Introduction

System F-bounded is a second order lambda calculus with subtyping. It extends system $F_{\leq}$ (see [Cardelli Wegner 85, Curien Ghelli 92, Ghelli 90, Cardelli et al. 91]) in that bounds of typed quantification may contain the bounded variable. In object-oriented terms, this feature allows one to write functions which accept parameters belonging to all the classes which inherit from one class. F-bounded quantification was introduced in [Canning et al. 89] and is currently included in many proposals for strongly typed object-oriented languages (e.g., [Bruce 93, Mitchell 90, Katiyar et al. 94]).

A typed λ-calculus is *strongly normalizing* (or *terminating*) when no infinite reduction chain starts from a typed term of that calculus. Termination is related to the possibility of solving some recursive type equations. For example, in a system with subtyping, if the disequation system $\{\alpha \leq \alpha\rightarrow\beta,\ \alpha\rightarrow\beta \leq \gamma,\ \alpha\rightarrow\beta \leq \gamma\rightarrow\beta'\}$[2] has a solution, then the non-terminating term $(\lambda x{:}\alpha.x(x))(\lambda x{:}\alpha.x(x)){:}\beta'$ is well typed. Some recursive disequations (e.g. $\alpha \leq \alpha\rightarrow\beta$) can already be solved in $F_{\leq}$ by exploiting the Top type (e.g. $\alpha=Top\rightarrow\beta$). F-bounded quantification introduces a new class of solutions, involving type variables (e.g. $\alpha=t$ when $t \leq t\rightarrow\beta$). This raises the question of whether the addition of F-bounded quantification may allow non-terminating terms to be written. In this note we prove that such terms cannot, in fact, be written.

Our proof is based on the computability method [Girard 72], and our approach is similar to the one used in [Mitchell 86] to prove termination for F, and, more closely, to the proof of termination of $F_{\leq}$ given in [Ghelli 90]. We first show that the type erasure of each well-typed F-bounded term is strongly normalizing. To this aim, we define an interpretation of F-bounded types such that each type is interpreted with a superset of the type erasures of all F-bounded terms with that type, and this superset is small enough to be contained in the set SN of strongly normalizing λ-terms: $\Gamma \vdash a{:}\ T \Rightarrow typeErasure(a) \in [\![\Gamma \vdash T]\!]$ and $[\![\Gamma \vdash T]\!] \subseteq SN$. From this, we derive strong normalization for system F-bounded.

System F-bounded is introduced in Section 2. Strong normalization is proved in Section 3.

2 System F-bounded

We adopt the following syntax for F-bounded types, terms, environments, and judgements.

PreTypes	$A ::= t \mid Top \mid A\rightarrow A \mid \forall t{\leq}A.\ A$
PreTermes	$a ::= x \mid \lambda x{:}A.\ a \mid a(a) \mid \Lambda t{\leq}A.\ a \mid a\{A\}$
Pre-Type Environments	$\Gamma ::= () \mid \Gamma, t{\leq}A$
Pre-Value Environments	$\Delta ::= () \mid \Delta, x{:}A$
Pre-Name Environments	$E ::= () \mid E, t$
Pre-Judgments	$J ::= E \vdash \lozenge \mid \Gamma \vdash \lozenge \mid \Gamma, \Delta \vdash \lozenge \mid E \vdash A \mid \Gamma \vdash A \leq A \mid \Gamma, \Delta \vdash a{:}\ A$

To give a good formalization of system F-bounded, some care is needed to prevent the environment formation problem $\Gamma, t{\leq}A \vdash \lozenge$ from being reduced by the rules to the type formation problem $\Gamma, t{\leq}A \vdash A$, which in turn may need a proof of $\Gamma, t{\leq}A \vdash \lozenge$. We deal with this problem by putting in the environment of each judgement the minimum amount of information needed by that judgement. Specifically, well formation of a type only depends on the type variables which have been defined, and does not depend on their bound. Hence, in our formalization $\Gamma, t{\leq}A \vdash \lozenge$ (read: $\Gamma, t{\leq}A$ is a well-formed environment) is reduced to $vars(\Gamma), t \vdash A$ (read: A is a

[1] Dipartimento di Informatica, Università di Pisa, Corso Italia 40, I-56125, Pisa, Italy, ghelli@di.unipi.it.
This work was carried out with the partial support of E.C., Esprit Basic Research Action 6309 FIDE2, of the Italian National Research Council, "Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo" grant No. 92.01561.PF69, and of the Ministero dell'Università e della Ricerca Scientifica e Tecnologica.

[2] Which, in most systems, is equivalent to the more familiar $\alpha \leq \alpha\rightarrow\beta \leq \alpha$.

well-formed type), where vars(Γ) is the set of variables defined in Γ. This approach is very well suited to our termination proof.

Free type variables:

Types: $FTV(t) = \{t\}$; $FTV(Top) = \{\}$; $FTV(A \rightarrow A') = FTV(A) \cup FTV(A')$; $FTV(\forall t \leq A.\ A') = (FTV(A) \cup FTV(A')) \setminus \{t\}$

Value Environments: $FTV(()) = \{\}$; $FTV(\Delta, x{:}A) = FTV(\Delta) \cup FTV(A)$

Type Environments: $FTV(()) = \{\}$; $FTV(\Gamma, t \leq A) = FTV(\Gamma) \cup FTV(A)$

Name environment, type environment, and value environment formation

(∅ NameEnv) $() \vdash \diamond$

(NameEnv) $$\frac{E \vdash \diamond \quad t \notin E\text{[3]}}{E, t \vdash \diamond}$$

(∅ TypeEnv) $() \vdash \diamond$

(TypeEnv) $$\frac{\Gamma \vdash \diamond \quad vars(\Gamma), t \vdash A \quad t \notin \Gamma}{\Gamma, t \leq A \vdash \diamond}$$

(∅ ValueEnv) $$\frac{\Gamma \vdash \diamond}{\Gamma, () \vdash \diamond}$$

(ValueEnv) $$\frac{\Gamma, \Delta \vdash \diamond \quad vars(\Gamma) \vdash A \quad x \notin \Delta}{\Gamma, \Delta, x{:}A \vdash \diamond}$$

Type formation

(Var Form) $$\frac{E, t, E' \vdash \diamond}{E, t, E' \vdash t}$$

(Top Form) $$\frac{E \vdash \diamond}{E \vdash Top}$$

(→ Form) $$\frac{E \vdash A \quad E \vdash B}{E \vdash A \rightarrow B}$$

(∀ Form) $$\frac{E, t \vdash A \quad E, t \vdash B}{E \vdash \forall t \leq A.B}$$

Subtypes

(Id ≤) $$\frac{\Gamma \vdash \diamond \quad vars(\Gamma) \vdash A}{\Gamma \vdash A \leq A}$$

(Trans ≤) $$\frac{\Gamma \vdash A \leq B \quad \Gamma \vdash B \leq C}{\Gamma \vdash A \leq C}$$

(Var ≤) $$\frac{\Gamma, t \leq A, \Gamma' \vdash \diamond}{\Gamma, t \leq A, \Gamma' \vdash t \leq A}$$

(Top ≤) $$\frac{\Gamma \vdash \diamond \quad vars(\Gamma) \vdash A}{\Gamma \vdash A \leq Top}$$

(→ ≤) $$\frac{\Gamma \vdash A \leq A' \quad \Gamma \vdash B \leq B'}{\Gamma \vdash A' \rightarrow B \leq A \rightarrow B'}$$

(∀ ≤) $$\frac{\Gamma, t \leq A \vdash t \leq A' \quad \Gamma, t \leq A \vdash B \leq B'\text{[4]}}{\Gamma \vdash \forall t \leq A'.B \leq \forall t \leq A.B'}$$

Term formation

(Var) $$\frac{\Gamma, \Delta, x{:}A, \Delta' \vdash \diamond}{\Gamma, \Delta, x{:}A, \Delta' \vdash x{:}\ A}$$

(Subsump) $$\frac{\Gamma, \Delta \vdash a{:}\ A \quad \Gamma \vdash A \leq B}{\Gamma, \Delta \vdash a{:}\ B}$$

(→ Intro) $$\frac{\Gamma, \Delta, x{:}A \vdash b{:}\ B}{\Gamma, \Delta \vdash \lambda x{:}A.b{:}\ A \rightarrow B}$$

(→ Elim) $$\frac{\Gamma, \Delta \vdash f{:}\ A \rightarrow B \quad \Gamma, \Delta \vdash a{:}\ A}{\Gamma, \Delta \vdash f(a){:}\ B}$$

(∀ Intro) $$\frac{\Gamma, t \leq A, \Delta \vdash b{:}\ B \quad t \notin FTV(\Delta)}{\Gamma, \Delta \vdash \Lambda t \leq A.b{:}\ \forall t \leq A.B}$$

(∀ Elim) $$\frac{\Gamma, \Delta \vdash f{:}\ \forall t \leq A.B \quad \Gamma \vdash A' \leq A[t \leftarrow A']}{\Gamma, \Delta \vdash f\{A'\}{:}\ B[t \leftarrow A']}$$

Reduction rules:

(β)	$(\lambda x{:}A.b)(a)$	—▷	$b[x \leftarrow a]$	
(η)	$\lambda x{:}A.b(x)$	—▷	b	$x \notin FV(b)$
(β2)	$(\Lambda t \leq A.b)\{A'\}$	—▷	$b[t \leftarrow A']$	
(η2)	$\Lambda t \leq A.b\{t\}$	—▷	b	$t \notin FV(b)$

Remark: By the rules, once a variable x or t is defined in an environment, it can neither be defined in the same environment once more, nor be defined by a λ, Λ or ∀ in the right hand side of the judgment.

[3] We write $t \in \Gamma$ if $t \leq A$ is a component of Γ, for some A; similarly for $x \in \Delta$ and $t \in E$.

[4] See [Katiyar 92] for a discussion of this rule.

3 Strong normalization of F-bounded terms

3.1 Saturated sets

Before giving the strong normalization proof, the notion of saturated sets must be introduced.

Notation (Λ, SN, $\mathcal{P}$**)**: Λ is the set of all (untyped) λ-terms (defined, as usual, as a ::= x | λx.a | aa).
SN is the set of all $\beta\eta$ strongly normalizing λ-terms.
$\mathcal{P}$(A) is the set of all subsets of A.

Definition (Saturated set**)**: A set $R \subseteq \Lambda$ is saturated when:

Sat_0	$R \subseteq SN$		
Sat_1	$a \in SN$, $(b[x\backslash a]b_1\ldots b_n) \in R$	$\Rightarrow$	$(\lambda x.b)ab_1\ldots b_n \in R$
Sat_2	$b_1,\ldots,b_n \in SN$	$\Rightarrow$	$xb_1\ldots b_n \in R$.

Notation (SAT**)**: SAT is the set of all saturated sets. Note that SAT $\subseteq \mathcal{P}(SN) \subseteq \mathcal{P}(\Lambda)$.

Remark NotEmpty: By Sat_2, if $R \in SAT$, then, for any variable x, $x \in R$, hence $R \neq \varnothing$.

Lemma SN: $SN \in SAT$.

Proof: We prove that Sat_1 and Sat_2 hold for SN.

Sat_1: $a \in SN$, $b[x\leftarrow a]b_1\ldots b_n \in SN \;\Rightarrow\; (\lambda x.b)ab_1\ldots b_n \in SN$

Proof of Sat_1: Let depth(a) be the maximum length of a $\beta\eta$ reduction chain starting from a term $a \in SN$. We prove Sat_1 by induction on $depth(b[x\leftarrow a]b_1\ldots b_n)+depth(a)$. Let $b[x\leftarrow a]b_1\ldots b_n \in SN$ and consider any reduction chain starting from $(\lambda x.b)ab_1\ldots b_n$. The first step of this chain is one of the following:

1)	$(\lambda x.b)ab_1\ldots b_n$	$-\!\!\rhd_\beta$	$b[x\leftarrow a]b_1\ldots b_n$	
2)	$(\lambda x.b)ab_1\ldots b_n$	$-\!\!\rhd_\eta$	$cab_1\ldots b_n$	with b=cx and $x\notin FV(c)$
3)	$(\lambda x.b)ab_1\ldots b_n$	$-\!\!\rhd$	$(\lambda x.b')ab_1\ldots b_n$	with $b-\!\!\rhd b'$
4)	$(\lambda x.b)ab_1\ldots b_i\ldots b_n$	$-\!\!\rhd$	$(\lambda x.b)ab_1\ldots b'_i\ldots b_n$	$1\leq i\leq n$, $b_i-\!\!\rhd b'_i$
5)	$(\lambda x.b)ab_1\ldots b_n$	$-\!\!\rhd$	$(\lambda x.b)a'b_1\ldots b_n$	with $a-\!\!\rhd a'$

We show that in any case the reduced term is in SN. If $depth(b[x\leftarrow a]b_1\ldots b_n)+depth(a)=0$, only the first two cases are possible, and in those cases the inductive hypothesis is not needed.

1) $b[x\leftarrow a]b_1\ldots b_n \in SN$ by hypothesis.
2) By b=cx and $x\notin FV(b)$, $cab_1\ldots b_n = b[x\leftarrow a]b_1\ldots b_n$. $b[x\leftarrow a]b_1\ldots b_n \in SN$ by hypothesis.
3) $b-\!\!\rhd b'$ implies $b[x\leftarrow a]b_1\ldots b_n -\!\!\rhd b'[x\leftarrow a]b_1\ldots b_n$. Hence $b'[x\leftarrow a]b_1\ldots b_n \in SN$.
   Since $depth(b'[x\leftarrow a]b_1\ldots b_n)+depth(a) < depth(b[x\leftarrow a]b_1\ldots b_n)+depth(a)$, then, by induction:
   $b'[x\leftarrow a]b_1\ldots b_n \in SN \Rightarrow (\lambda x.b')ab_1\ldots b_n \in SN$.
4) The same as 3), but substitute b' with b and b_i with b_i'.
5) $a-\!\!\rhd a'$ implies that $b[x\leftarrow a]b_1\ldots b_n$ reduces to $b[x\leftarrow a']b_1\ldots b_n$ in 0-n steps. Hence, $b[x\leftarrow a']b_1\ldots b_n \in SN$.
   Since $depth(b[x\leftarrow a']b_1\ldots b_n)+depth(a') < depth(b[x\leftarrow a]b_1\ldots b_n)+depth(a)$, then, by induction:
   $b[x\leftarrow a']b_1\ldots b_n \in SN \Rightarrow (\lambda x.b)a'b_1\ldots b_n \in SN$.
   The addendum depth(a) is important when x is not free in b.

Sat_2: $b_1,\ldots,b_n \in SN \;\Rightarrow\; xb_1\ldots b_n \in SN$

Proof of Sat_2: By induction on $\sum_{i=1..n} depth(b_i)$. If $\sum_{i=1..n} depth(b_i)=0$, $xb_1\ldots b_n \in SN$ since it is in normal form. Otherwise, consider any reduction chain starting from $xb_1\ldots b_n$: $xb_1\ldots b_i\ldots b_n -\!\!\rhd xb_1\ldots b'_i\ldots b_n -\!\!\rhd \ldots$: b'_i is a reduct of b_i, hence $b'_i \in SN$, and $depth(b'_i) < depth(b_i)$. We can now apply induction to $b_1,\ldots,b'_i,\ldots,b_n$ to obtain that $xb_1\ldots b'_i\ldots b_n \in SN$, hence $xb_1\ldots b_i\ldots b_n \in SN$. □

Lemma Intersect: $(I \neq \emptyset \text{ and } \forall i \in I.\ S_i \in SAT) \Rightarrow (\bigcap_{i\in I} S_i) \in SAT$

Proof: Sat_0: $\bigcap_{i\in I} S_i \subseteq SN$: let j be an element of I: $\bigcap_{i\in I} S_i \subseteq S_j \subseteq SN$.

Sat_1: $a \in SN,\ b[x\leftarrow a]b_1\ldots b_n \in \bigcap_{i\in I} S_i \Rightarrow (\lambda x.b)ab_1\ldots b_n \in \bigcap_{i\in I} S_i$:

let $a \in SN,\ b[x\leftarrow a]b_1\ldots b_n \in \bigcap_{i\in I} S_i$; by def. of $\bigcap$: $\forall i\in I.\ b[x\leftarrow a]b_1\ldots b_n \in S_i$
by Sat_1 of S_i: $\forall i\in I.\ (\lambda x.b)ab_1\ldots b_n \in S_i$
by def. of $\bigcap$: $(\lambda x.b)ab_1\ldots b_n \in \bigcap_{i\in I} S_i$.

Sat_2: $b_1,\ldots,b_n \in SN \Rightarrow xb_1\ldots b_n \in \bigcap_{i\in I} S_i$:

let $b_1,\ldots,b_n \in SN$; by Sat_2 on S_i: $\forall i\in I.\ b[x\leftarrow a]b_1\ldots b_n \in S_i$
by def. of $\bigcap$: $b[x\leftarrow a]b_1\ldots b_n \in \bigcap_{i\in I} S_i$. □

Notation: $Min_{SAT} = \bigcap_{\iota\in SAT} \iota$.

Remark MinSAT: Min_{SAT} is a well defined saturated set: it is well defined since, by lemma SN, SAT is not empty. It is saturated by Lemma Intersect.

3.2 The theorem

Definition (type erasure):
typeErasure(x) = x
typeErasure(λx:A. a) = λx. typeErasure(a)
typeErasure(a(a')) = typeErasure(a)(typeErasure(a'))
typeErasure(Λt≤A. a) = typeErasure(a)
typeErasure(a{A}) = typeErasure(a).

Notation (|Γ|, |Δ|, |E|): $|t_1\leq A_1,\ldots,t_n\leq A_n| = |x_1:A_1,\ldots,x_n:A_n| = |t_1,\ldots,t_n| = n$.

Notation (S^n): As usual, for any set S, we define $S^n=(\ldots(S^0\times S_{(1)})\times\ldots\times S_{(n)})$, where S^0 is an arbitrary singleton {s}, to deal smoothly with the nullary case. Similarly, we define $<s_1,\ldots,s_n> = <\ldots<s,s_1>,\ldots,s_n>$.

Notation (Λ^0, SAT^0): Λ^0 is an arbitrary singleton ({x}) used as a unit in products of subsets of Λ. Similarly, SAT^0 is an arbitrary singleton ({MinSAT}) used as a unit for products of subsets of SAT.

Notation ($a_{\Delta\leftarrow\delta}$): If Δ is a value environment $x_1:A_1,\ldots,x_n:A_n$ and δ is a tuple $<a_1,\ldots,a_n>\subseteq\Lambda^n$, then $\Delta\leftarrow\delta$ is the substitution $[x_1\leftarrow a_1,\ldots,x_n\leftarrow a_n]$, and $a_{\Delta\leftarrow\delta}$ is the result of applying $\Delta\leftarrow\delta$ to a.

Lemma: The type erasure of any F-bounded term is a terminating λ term.

Proof: We define five "semantic functions" which interpret any provable type environment, value environment, type, subtype, or term formation judgement. We prove that the interpretation of each provable judgement satisfies an associated "soundness condition". These conditions imply βη strong normalization for any type-erased F-bounded term.

Informally: a type is interpreted by a set of λ-terms; a type environment $t_1\leq T_1,\ldots,t_n\leq T_n$ by a set of n-tuples of sets, where each n-tuple specifies a possible way of associating a set with each type variable; a value environment is interpreted by a set of tuples of λ-terms, where each tuple specifies a well-typed assignments of λ-terms to the value variables; a value is interpreted by its type erasure.

The soundness properties specify, informally, that: any type is a saturated set; each term belongs to its type; no environment interpretation may be empty (empty environments would make the other soundness conditions useless, since those conditions are quantified on variables ranging over environment interpretations).

We give here either the interpretation, or the domain of the interpretation, for each judgement, and all the soundness conditions. The missing interpretations, and the proof of the soundness properties, will be given in Sections 3.3 - 3.7.

(TypeEnv) Domain: $\llbracket\Gamma \vdash \diamond\rrbracket \subseteq (P(\Lambda))^{|\Gamma|}$ (Definition: Section 3.3)
Soundness: $\llbracket\Gamma \vdash \diamond\rrbracket \subseteq SAT^{|\Gamma|}$ and $\llbracket\Gamma \vdash \diamond\rrbracket \neq \emptyset$

(ValueEnv) Definition: $\forall\gamma\in\llbracket\Gamma \vdash \diamond\rrbracket.\ \llbracket\Gamma, x_1:A_1,\ldots,x_n:A_n \vdash \diamond\rrbracket\gamma = \Lambda^0\times\llbracket vars(\Gamma) \vdash A_1\rrbracket\gamma\times\ldots\times\llbracket vars(\Gamma) \vdash A_n\rrbracket\gamma$
Soundness: $\forall\gamma\in\llbracket\Gamma \vdash \diamond\rrbracket.\ \llbracket\Gamma, x_1:A_1,\ldots,x_n:A_n \vdash \diamond\rrbracket\gamma \neq \emptyset$,
$\forall<a_1,\ldots,a_n>\in\llbracket\Gamma, x_1:A_1,\ldots,x_n:A_n \vdash \diamond\rrbracket\gamma.\ a_i \in \llbracket vars(\Gamma) \vdash A_i\rrbracket\gamma$

(Type) Domain: $\forall\gamma\in SAT^{|E|}.\ \llbracket E \vdash A\rrbracket\gamma \in P(\Lambda)$ (Definition: Section 3.5)
Soundness: $\forall\gamma\in SAT^{|E|}.\ \llbracket E \vdash A\rrbracket\gamma \in SAT$

(Subtype) Definition: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \llbracket\Gamma \vdash A \leq B\rrbracket\gamma = <\llbracket vars(\Gamma) \vdash A\rrbracket\gamma,\llbracket vars(\Gamma) \vdash B\rrbracket\gamma>$
Soundness: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \pi_1\llbracket\Gamma \vdash A \leq B\rrbracket\gamma \subseteq \pi_2\llbracket\Gamma \vdash A \leq B\rrbracket\gamma$

(Term) Definition: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \llbracket\Gamma, \Delta \vdash a{:}\ A\rrbracket\gamma\delta = typeErasure(a)_{\Delta\leftarrow\delta}$
Soundness: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \llbracket\Gamma, \Delta \vdash a{:}\ A\rrbracket\gamma\delta \in \llbracket vars(\Gamma) \vdash A\rrbracket\gamma$

We prove that the interpretation of any provable judgement satisfies soundness by induction on its proof tree, by showing that, for each rule, if the interpretation of the premises is sound, the interpretation of the consequences is sound too. This proof will be carried out rule by rule in the next five sections.

Assuming that the soundness of the interpretation will be proved, we can now prove the lemma. Let a be an F-bounded term typed in an environment Γ, Δ. By Remark NotEmpty and definition (ValueEnv), for any $\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket$, the tuple of λ-terms $vars(\Delta)$ belongs to $\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma$. Take a $\gamma_0\in\llbracket\Gamma \vdash \Diamond\rrbracket$ (this exists by soundness condition (TypeEnv)). By soundness condition (Term), $typeErasure(a) = typeErasure(a)_{\Delta\leftarrow vars(\Delta)} \in \llbracket vars(\Gamma) \vdash A\rrbracket\gamma_0$; by soundness condition (Type), $\llbracket vars(\Gamma) \vdash A\rrbracket\gamma_0$ is saturated hence, by Sat_0, typeErasure(a) is strongly normalizing. □

Theorem: F-bounded is strongly normalizing.

Proof: Consider a β-η-$\beta2$-$\eta2$ reduction chain R starting from an F-bounded term and the chain typeErasure(R) consisting of all the type erasures of the elements of R. A β-η step corresponds in typeErasure(R) to each β-η step in R, while two identical terms correspond in typeErasure(R) to each pair of terms related by a $\beta2$-$\eta2$ step in R. Hence, if we collapse all sequences of identical terms in typeErasure(R), we still obtain a β-η reduction chain “collapse(typeErasure(R))” in Λ. Since each $\beta2$-$\eta2$ step deletes a Λ symbol without creating any new Λ, any sequence of $\beta2$-$\eta2$ reductions in R has a finite length. Hence, if R and typeErasure(R) were infinite, then collapse(typeErasure(R)) would be infinite too. Since collapse(typeErasure(R)) is finite by the previous Lemma, then R is finite too. □

In the next section we give the missing interpretations and we prove that each rule, when applied to judgements with a sound interpretation, yields a judgement with a sound interpretation, completing the proof of the theorem.

3.3 Interpretation and soundness of type environment judgements

Interpretation: ($\varnothing$ TypeEnv) $\llbracket () \vdash \Diamond\rrbracket = SAT^0$ (a singleton)
(TypeEnv) $\llbracket\Gamma, t\leq A \vdash \Diamond\rrbracket = \{<\gamma,\iota> \mid \gamma \in \llbracket\Gamma \vdash \Diamond\rrbracket, \iota \in SAT, \iota \subseteq \llbracket vars(\Gamma), t \vdash A\rrbracket<\gamma,\iota>\}$

Note that no circularity is hidden in the condition $\iota \subseteq \llbracket vars(\Gamma), t \vdash A\rrbracket<\gamma,\iota>$, which is just a set inclusion with ι appearing on both sides. $\llbracket vars(\Gamma), t \vdash A\rrbracket<\gamma,\iota>$ is well defined since $\gamma \in \llbracket\Gamma \vdash \Diamond\rrbracket \subseteq (SAT(\Lambda))^{|\Gamma|}$ by induction, and $\iota \in SAT$ by construction.

Soundness: $\llbracket\Gamma \vdash \Diamond\rrbracket \subseteq SAT^{|\Gamma|}$ and $\llbracket\Gamma \vdash \Diamond\rrbracket \neq \varnothing$

($\varnothing$ TypeEnv): $SAT^0 \subseteq SAT^0$ and $SAT^0 \neq \varnothing$ are both true by definition.

(TypeEnv): $\Gamma \vdash \Diamond,\quad vars(\Gamma), t \vdash A,\quad t\notin\Gamma \quad\Rightarrow\quad \Gamma, t\leq A \vdash \Diamond$

Hyp.: $\llbracket\Gamma \vdash \Diamond\rrbracket \subseteq SAT^{|\Gamma|},\ \llbracket\Gamma \vdash \Diamond\rrbracket \neq \varnothing,\ \forall<\gamma,\iota>\in SAT^{|\Gamma|+1}.\ \llbracket vars(\Gamma), t \vdash A\rrbracket<\gamma,\iota> \in SAT$
Th.: $\llbracket\Gamma, t\leq A \vdash \Diamond\rrbracket \subseteq SAT^{|\Gamma|+1}$ (a)
$\llbracket\Gamma, t\leq A \vdash \Diamond\rrbracket \neq \varnothing$ (b)

Proof: (a) $\llbracket\Gamma, t\leq A \vdash \Diamond\rrbracket = \{<\gamma,\iota> \mid \gamma \in \llbracket\Gamma \vdash \Diamond\rrbracket, \iota \in SAT,\ldots\} \subseteq SAT^{|\Gamma|+1|}$ since $\gamma \in SAT^{|\Gamma|}$ and $\iota \in SAT$.
(b) $\llbracket\Gamma, t\leq A \vdash \Diamond\rrbracket \neq \varnothing$: By Hyp. $\exists\gamma_0\in\llbracket\Gamma \vdash \Diamond\rrbracket$. Let $\iota=Min_{SAT}$. By Hyp., $\llbracket vars(\Gamma), t \vdash A\rrbracket<\gamma_0,Min_{SAT}> \in SAT$, by Lemma MinSAT, $Min_{SAT} \subseteq \llbracket vars(\Gamma), t \vdash A\rrbracket<\gamma_0,Min_{SAT}>$, hence $<\gamma_0,Min_{SAT}> \in \llbracket\Gamma, t\leq A \vdash \Diamond\rrbracket$. □

3.4 Interpretation and soundness of value environment judgements

Interpretation: $\llbracket\Gamma, x_1{:}A_1,\ldots,x_n{:}A_n \vdash \Diamond\rrbracket\gamma = \Lambda^0 \times \llbracket\mathrm{vars}(\Gamma) \vdash A_1\rrbracket\gamma \times \ldots \times \llbracket\mathrm{vars}(\Gamma) \vdash A_n\rrbracket\gamma$

In this case the cartesian product is used, rather than the “dependent product” used to interpret type environment, since the bounds in a value environment do not depend on the previous value variables.

Soundness: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \Lambda^0\times\llbracket\mathrm{vars}(\Gamma) \vdash A_1\rrbracket\gamma\times\ldots\times\llbracket\mathrm{vars}(\Gamma) \vdash A_n\rrbracket\gamma\triangle$⬠ $\neq \emptyset$ (a)

$\forall<a_1,\ldots,a_n>\in\Lambda^0\times\llbracket\mathrm{vars}(\Gamma) \vdash A_1\rrbracket\gamma\times\ldots\times\llbracket\mathrm{vars}(\Gamma) \vdash A_n\rrbracket\gamma.\ a_i \in \llbracket\mathrm{vars}(\Gamma) \vdash A_i\rrbracket\gamma$ (b)

Proof: (a): Each factor of the product is not empty, by $\llbracket\mathrm{vars}(\Gamma) \vdash A_i\rrbracket\gamma \in \mathrm{SAT}$ and by Remark NotEmpty.
(b): By definition of cartesian product. □

3.5 Interpretation and soundness of type judgements

Interpretation: (Var Form) $\forall<\iota_1,\ldots,\iota_n>\in\mathrm{SAT}^n.\ \llbracket t_1,\ldots,t_n \vdash t_i\rrbracket<\iota_1,\ldots,\iota_n> = \iota_i$
(Top Form) $\forall\gamma\in\mathrm{SAT}^{|E|}.\ \llbracket E \vdash \mathrm{Top}\rrbracket\gamma = \mathrm{SN}$
(→ Form) $\forall\gamma\in\mathrm{SAT}^{|E|}.\ \llbracket E \vdash A\to B\rrbracket\gamma = \{b\in\Lambda \mid a \in \llbracket E \vdash A\rrbracket\gamma \Rightarrow b(a) \in \llbracket E \vdash B\rrbracket\gamma\}$
(∀ Form) $\forall\gamma\in\mathrm{SAT}^{|E|}.\ \llbracket E \vdash \forall t\leq A.B\rrbracket\gamma = \bigcap_{\iota\in\mathrm{SAT},\iota\subseteq\llbracket E,\, t \vdash A\rrbracket<\gamma,\iota>} \llbracket E, t \vdash B\rrbracket<\gamma,\iota>$

Type variables are interpreted by the type environment. Top is the set of all strongly normalizing λ-terms. A functional type $\llbracket A\to B\rrbracket$ contains all terms which, applied to a term in $\llbracket A\rrbracket$, yeld a term in $\llbracket B\rrbracket$; note that we mean a bare syntactic application, with no evaluation. Quantification is interpreted by intersection.

Soundness: $\forall\gamma\in\mathrm{SAT}^{|E|}.\ \llbracket E \vdash A\rrbracket\gamma \in \mathrm{SAT}$.

(Var Form) $E, t, E' \vdash \Diamond \quad\Rightarrow\quad E, t, E' \vdash t$

$\forall<\iota_1,\ldots,\iota_n>\in \mathrm{SAT}^n.\ \llbracket t_1,\ldots,t_n \vdash t_i\rrbracket<\iota_1,\ldots,\iota_n> = \iota_i$ belongs to SAT by construction. □

(Top Form) $E, t, E' \vdash \Diamond \quad\Rightarrow\quad E, t, E' \vdash \mathrm{Top}$

$\forall\gamma\in\mathrm{SAT}^{|E|}.\ \llbracket E \vdash \mathrm{Top}\rrbracket\gamma = \mathrm{SN}$ belongs to SAT by Lemma SN. □

(→ Form) $E \vdash A,\quad E \vdash B \quad\Rightarrow\quad E \vdash A\to B$

Soundness: Hyp.: $\forall\gamma\in\mathrm{SAT}^{|E|}.\ \llbracket E \vdash A\rrbracket\gamma \in \mathrm{SAT}$ and $\llbracket E \vdash B\rrbracket\gamma \in \mathrm{SAT}$.
Th.: $\forall\gamma\in\mathrm{SAT}^{|E|}.\ \llbracket E \vdash A\to B\rrbracket\gamma \in \mathrm{SAT}$.

Sat_0: $\llbracket E \vdash A\to B\rrbracket\gamma \subseteq \mathrm{SN}$.

Let $f \in \llbracket E \vdash A\to B\rrbracket\gamma$. Consider any variable x; $x \in \llbracket E \vdash A\rrbracket\gamma$ by Sat_2, hence $f(x) \in \llbracket E \vdash B\rrbracket\gamma$ by definition of $\llbracket E \vdash A\to B\rrbracket\gamma$. $f(x) \in \mathrm{SN}$ by Sat_0, hence $f \in \mathrm{SN}$.

Sat_1: $a \in \mathrm{SN},\ (b[x\backslash a]b_1\ldots b_n) \in \llbracket E \vdash A\to B\rrbracket\gamma \quad\Rightarrow\quad (\lambda x.b)ab_1\ldots b_n \in \llbracket E \vdash A\to B\rrbracket\gamma$

Let:	$a \in \mathrm{SN},\ (b[x\backslash a]b_1\ldots b_n) \in \llbracket E \vdash A\to B\rrbracket\gamma$
By def. of $\llbracket E \vdash A\to B\rrbracket\gamma$:	$\forall a'\in\llbracket E \vdash A\rrbracket\gamma.\ (b[x\backslash a]b_1\ldots b_n)a' \in \llbracket E \vdash B\rrbracket\gamma$
By Sat_1 for $\llbracket E \vdash B\rrbracket\gamma$:	$\forall a'\in\llbracket E \vdash A\rrbracket\gamma.\ ((\lambda x.b)ab_1\ldots b_n)a' \in \llbracket E \vdash B\rrbracket\gamma$
By def. of $\llbracket E \vdash A\to B\rrbracket\gamma$:	$(\lambda x.b)ab_1\ldots b_n \in \llbracket E \vdash A\to B\rrbracket\gamma$

Sat_2: $b_1,\ldots,b_n \in \mathrm{SN} \quad\Rightarrow\quad xb_1\ldots b_n \in \llbracket E \vdash A\to B\rrbracket\gamma$

Let:	$b_1,\ldots,b_n \in \mathrm{SN}$
By Sat_2 for $\llbracket E \vdash B\rrbracket\gamma$:	$\forall a\in\mathrm{SN}.\ xb_1\ldots b_n a \in \llbracket E \vdash B\rrbracket\gamma$
By $\llbracket E \vdash A\rrbracket\gamma \subseteq \mathrm{SN}$:	$\forall a\in\llbracket E \vdash A\rrbracket\gamma.\ xb_1\ldots b_n a \in \llbracket E \vdash B\rrbracket\gamma$
By def. of $\llbracket E \vdash A\to B\rrbracket\gamma$:	$xb_1\ldots b_n \in \llbracket E \vdash A\to B\rrbracket\gamma$. □

(∀ Form) $E, t \vdash A,\quad E, t \vdash B \;\Rightarrow\; E \vdash \forall t{\le}A.B$

Soundness: Hyp.: $\forall{<}\gamma,\iota{>}\in SAT^{|E|+1}.\ \llbracket E, t \vdash A\rrbracket{<}\gamma,\iota{>} \in SAT$ and $\llbracket E, t \vdash B\rrbracket{<}\gamma,\iota{>} \in SAT$
Th.: $\forall\gamma\in SAT^{|E|}.\ \llbracket E \vdash \forall t{\le}A.B\rrbracket\gamma =_{def} \bigcap_{\iota\in SAT,\iota\subseteq\llbracket E,\, t \vdash A\rrbracket<\gamma,\iota>} \llbracket E, t \vdash B\rrbracket{<}\gamma,\iota{>} \in SAT$

Proof: $\{\iota \mid \iota\in SAT,\ \iota\subseteq\llbracket E, t \vdash A\rrbracket{<}\gamma,\iota{>}\}$ is not empty: by Hyp., $\forall\gamma\in SAT^{|E|}.\ \llbracket E, t \vdash A\rrbracket{<}\gamma,Min_{SAT}{>} \in SAT$, hence $Min_{SAT} \subseteq \llbracket E, t \vdash A\rrbracket{<}\gamma,Min_{SAT}{>}$. We can now apply lemma Intersect, by observing that each $\llbracket E, t \vdash B\rrbracket{<}\gamma,\iota{>}$ is saturated by Hyp., to conclude that $\llbracket E \vdash \forall t{\le}A.B\rrbracket\gamma \in SAT$. □

We now present some lemmas about type interpretation which will be used in the next sections.

Lemma Weakening: If $t\notin FTV(A)$: $\forall\gamma\in SAT^{|E|},\ \iota\in SAT,\ \gamma'\in SAT^{|E'|}.\ \llbracket E, t, E' \vdash A\rrbracket{<}\gamma,\iota,\gamma'{>} = \llbracket E, E' \vdash A\rrbracket{<}\gamma,\gamma'{>}$.

Proof: By induction and by cases on the shape of A. Cases A=u (with u≠t) and A=Top are immediate. Cases $A=A'\rightarrow A''$ and $A=\forall u{\le}A'.A''$ are immediate by induction. □

Lemma ValueEnvWeakening: If $t\notin FTV(\Delta)$ and $t\notin FTV(\Gamma')$:

$$\forall\gamma\in SAT^{|E|},\ \iota\in SAT,\ \gamma'\in SAT^{|E'|}.\ \llbracket \Gamma, t{\le}A, \Gamma', \Delta \vdash \diamond\rrbracket{<}\gamma,\iota,\gamma'{>} = \llbracket \Gamma, \Gamma', \Delta \vdash \diamond\rrbracket{<}\gamma,\gamma'{>}.$$

Proof: It is a corollary of Lemma Weakening, since:

$$\llbracket \Gamma, t{\le}A, \Gamma', x_1{:}A_1,\ldots,x_n{:}A_n \vdash \diamond\rrbracket\gamma =_{def} \Lambda^0\times\llbracket vars(\Gamma), t, vars(\Gamma') \vdash A_1\rrbracket\gamma\times\ldots\times\llbracket vars(\Gamma), t, vars(\Gamma') \vdash A_n\rrbracket\gamma.\quad □$$

Lemma TypeSubst: For any type formation judgement $\Gamma'' \vdash B[t\leftarrow A]$ where t does not appear, for any way of splitting Γ'' into two parts Γ, Γ' such that $\llbracket vars(\Gamma) \vdash A\rrbracket$, i.e. such that $\forall t'\in vars(\Gamma').\ t'\notin FTV(A)$:

$$\forall\gamma\in SAT^{|\Gamma|},\gamma'\in SAT^{|\Gamma'|}.\ \llbracket vars(\Gamma), vars(\Gamma') \vdash B[t\leftarrow A]\rrbracket{<}\gamma,\gamma'{>} = \llbracket vars(\Gamma), t, vars(\Gamma') \vdash B\rrbracket{<}\gamma,\llbracket vars(\Gamma) \vdash A\rrbracket\gamma,\gamma'{>}$$

Proof: By induction and by cases on the shape of B. Cases B=u (with u≠t) and B=Top are corollaries of Lemma Weakening.

B=t: $\llbracket vars(\Gamma), vars(\Gamma') \vdash t[t\leftarrow A]\rrbracket{<}\gamma,\gamma'{>}$
$= \llbracket vars(\Gamma), vars(\Gamma') \vdash A\rrbracket{<}\gamma,\gamma'{>}$
$=_{\text{by Lemma Weakening}} \llbracket vars(\Gamma) \vdash A\rrbracket{<}\gamma{>}$
$=_{\text{by def. (Var Form)}} \llbracket vars(\Gamma), t, vars(\Gamma') \vdash t\rrbracket{<}\gamma,\llbracket vars(\Gamma) \vdash A\rrbracket\gamma,\gamma'{>}$.

B=∀t'≤B'.B'': $\llbracket vars(\Gamma), vars(\Gamma') \vdash (\forall t'{\le}B'.B'')[t\leftarrow A]\rrbracket{<}\gamma,\gamma'{>}$
$=_{\text{by def. }(\forall\text{ Form})} \bigcap_{\iota\in SAT,\iota\subseteq\llbracket vars(\Gamma),\, vars(\Gamma'),\, t' \vdash B'[t\leftarrow A]\rrbracket<\gamma,\gamma',\iota>} \llbracket vars(\Gamma), vars(\Gamma'), t' \vdash B''[t\leftarrow A]\rrbracket{<}\gamma,\gamma',\iota{>}$
$=_{\text{by ind.}} \bigcap_{\iota\in SAT,\iota\subseteq\llbracket vars(\Gamma),\, t,\, vars(\Gamma'),\, t' \vdash B'\rrbracket<\gamma,\llbracket vars(\Gamma) \vdash A\rrbracket\gamma,\gamma',\iota>} \llbracket vars(\Gamma), t, vars(\Gamma'), t' \vdash B''\rrbracket{<}\gamma,\llbracket vars(\Gamma) \vdash A\rrbracket\gamma,\gamma',\iota{>}$
$=_{\text{by def. }(\forall\text{ Form})} \llbracket vars(\Gamma), t, vars(\Gamma') \vdash \forall t'{\le}B'.B''\rrbracket{<}\gamma,\llbracket vars(\Gamma) \vdash A\rrbracket\gamma,\gamma'{>}$.

B=B'→B'': similar but easier. □

3.6 Interpretation and soundness of subtype judgements

Interpretation: $\forall\gamma\in\llbracket\Gamma \vdash \diamond\rrbracket.\ \llbracket\Gamma \vdash A \le B\rrbracket\gamma = {<}\llbracket vars(\Gamma) \vdash A\rrbracket\gamma,\llbracket vars(\Gamma) \vdash B\rrbracket\gamma{>}$

Soundness: $\forall\gamma\in\llbracket\Gamma \vdash \diamond\rrbracket.\ \pi_1\llbracket\Gamma \vdash A \le B\rrbracket\gamma \subseteq \pi_2\llbracket\Gamma \vdash A \le B\rrbracket\gamma$,
i.e. $\forall\gamma\in\llbracket\Gamma \vdash \diamond\rrbracket.\ \llbracket vars(\Gamma) \vdash A\rrbracket\gamma \subseteq \llbracket vars(\Gamma) \vdash B\rrbracket\gamma$

(Var ≤) $\Gamma, t{\le}A, \Gamma' \vdash \diamond \;\Rightarrow\; \Gamma, t{\le}A, \Gamma' \vdash t \le A$

Soundness: $\forall\gamma\in SAT^{|\Gamma|},\ \iota\in SAT,\ \gamma'\in SAT^{|\Gamma'|}.\ {<}\gamma,\iota,\gamma'{>} \in \llbracket\Gamma, t{\le}A, \Gamma' \vdash \diamond\rrbracket$
$\Rightarrow\; \llbracket vars(\Gamma), t, vars(\Gamma') \vdash t\rrbracket{<}\gamma,\iota,\gamma'{>} \subseteq \llbracket vars(\Gamma), t, vars(\Gamma') \vdash A\rrbracket{<}\gamma,\iota,\gamma'{>}$

Let:	${<}\gamma,\iota,\gamma'{>} \in \llbracket\Gamma, t{\le}A, \Gamma' \vdash \diamond\rrbracket$	(a)
By definition (≤ Env):	${<}\gamma,\iota{>} \in \llbracket\Gamma, t{\le}A \vdash \diamond\rrbracket$	(b)
By the same definition:	$\iota \subseteq \llbracket vars(\Gamma), t \vdash A\rrbracket{<}\gamma,\iota{>}$	(c)
By definition (Var Form):	$\iota = \llbracket vars(\Gamma), t, vars(\Gamma') \vdash t\rrbracket{<}\gamma,\iota,\gamma'{>}$	(d)
By Lemma Weakening:	$\llbracket vars(\Gamma), t \vdash A\rrbracket{<}\gamma,\iota{>} = \llbracket vars(\Gamma), t, vars(\Gamma') \vdash A\rrbracket{<}\gamma,\iota,\gamma'{>}$	(e)
Substituting (d) and (e) in (c):	$\llbracket vars(\Gamma), t, vars(\Gamma') \vdash t\rrbracket{<}\gamma,\iota,\gamma'{>} \subseteq \llbracket vars(\Gamma), t, vars(\Gamma') \vdash A\rrbracket{<}\gamma,\iota,\gamma'{>}$. □	

(Top ≤) $\Gamma \vdash \Diamond,\quad \mathrm{vars}(\Gamma) \vdash A \quad\Rightarrow\quad \Gamma \vdash A \le \mathrm{Top}$

Soundness: Hyp.: $\llbracket \Gamma \vdash \;\rrbracket \subseteq \mathrm{SAT}^{|\Gamma|},\ \llbracket \Gamma \vdash \;\rrbracket \neq \emptyset,\ \forall\gamma\in\mathrm{SAT}^{|\Gamma|}.\ \llbracket \mathrm{vars}(\Gamma) \vdash A\rrbracket\gamma \in \mathrm{SAT}$
Th.: $\forall\gamma\in\llbracket \Gamma \vdash \Diamond\rrbracket.\ \llbracket \mathrm{vars}(\Gamma) \vdash A\rrbracket\gamma \subseteq \llbracket \mathrm{vars}(\Gamma) \vdash \mathrm{Top}\rrbracket\gamma = \mathrm{SN}$

By Hyp., $\llbracket \mathrm{vars}(\Gamma) \vdash A\rrbracket\gamma \in \mathrm{SAT}$; hence, $\llbracket \mathrm{vars}(\Gamma) \vdash A\rrbracket\gamma \subseteq \mathrm{SN}$ by Sat_0. □

(→ ≤) $\Gamma \vdash A \le A',\quad \Gamma \vdash B \le B' \quad\Rightarrow\quad \Gamma \vdash A'\!\to\! B \le A\!\to\! B'$

Soundness: Hyp.: $\forall\gamma\in\llbracket \Gamma \vdash \;\rrbracket.\ \llbracket \mathrm{vars}(\Gamma) \vdash A\rrbracket\gamma \subseteq \llbracket \mathrm{vars}(\Gamma) \vdash A'\rrbracket\gamma,$ (a)
$\forall\gamma\in\llbracket \Gamma \vdash \;\rrbracket.\ \llbracket \mathrm{vars}(\Gamma) \vdash B\rrbracket\gamma \subseteq \llbracket \mathrm{vars}(\Gamma) \vdash B'\rrbracket\gamma$ (b)
Th.: $\forall\gamma\in\llbracket \Gamma \vdash \;\rrbracket.\ \forall f\in\llbracket \mathrm{vars}(\Gamma) \vdash A'\!\to\! B\rrbracket\gamma.\ f \in \llbracket \mathrm{vars}(\Gamma) \vdash A\!\to\! B'\rrbracket\gamma$

Let: $\gamma \in \llbracket \Gamma \vdash \;\rrbracket,\ f \in \llbracket \mathrm{vars}(\Gamma) \vdash A'\!\to\! B\rrbracket\gamma$
By def. (→ Form): $a \in \llbracket \mathrm{vars}(\Gamma) \vdash A'\rrbracket\gamma \Rightarrow f(a) \in \llbracket \mathrm{vars}(\Gamma) \vdash B\rrbracket\gamma$
By (a): $a \in \llbracket \mathrm{vars}(\Gamma) \vdash A\rrbracket\gamma \Rightarrow f(a) \in \llbracket \mathrm{vars}(\Gamma) \vdash B\rrbracket\gamma$
By (b): $a \in \llbracket \mathrm{vars}(\Gamma) \vdash A\rrbracket\gamma \Rightarrow f(a) \in \llbracket \mathrm{vars}(\Gamma) \vdash B'\rrbracket\gamma$
By def. (→ Form): $f \in \llbracket \mathrm{vars}(\Gamma) \vdash A\!\to\! B'\rrbracket\gamma$. □

(∀ ≤) $\Gamma, t{\le}A \vdash t \le A',\quad \Gamma, t{\le}A \vdash B \le B' \quad\to\quad \Gamma \vdash \forall t{\le}A'.B \le \forall t{\le}A.B'$

Soundness: Hyp.: $\forall<\gamma,\iota>\in\llbracket \Gamma, t{\le}A \vdash \Diamond\rrbracket.\ \llbracket \mathrm{vars}(\Gamma), t \vdash t\rrbracket<\gamma,\iota> \subseteq \llbracket \mathrm{vars}(\Gamma), t \vdash A'\rrbracket<\gamma,\iota>$
$\forall<\gamma,\iota>\in\llbracket \Gamma, t{\le}A \vdash \Diamond\rrbracket.\ \llbracket \mathrm{vars}(\Gamma), t \vdash B\rrbracket<\gamma,\iota> \subseteq \llbracket \mathrm{vars}(\Gamma), t \vdash B'\rrbracket<\gamma,\iota>$
Th.: $\forall\gamma\in\llbracket \Gamma \vdash \;\rrbracket.\ \forall f\in\llbracket \mathrm{vars}(\Gamma) \vdash \forall t{\le}A'.B\rrbracket\gamma.\ f \in \llbracket \mathrm{vars}(\Gamma) \vdash \forall t{\le}A.B'\rrbracket\gamma$

Applying definitions (≤ Env) and (∀ Form) we can rewrite Hyp. and Th., respectively, as:

Hyp.: $\forall\gamma\in\llbracket \Gamma \vdash \;\rrbracket.\ \forall\iota\in\mathrm{SAT}.\ \iota \subseteq \llbracket \mathrm{vars}(\Gamma), t \vdash A\rrbracket<\gamma,\iota> \Rightarrow \llbracket \mathrm{vars}(\Gamma), t \vdash t\rrbracket<\gamma,\iota> \subseteq \llbracket \mathrm{vars}(\Gamma), t \vdash A'\rrbracket<\gamma,\iota>$ (a)
$\forall\gamma\in\llbracket \Gamma \vdash \;\rrbracket.\ \forall\iota\in\mathrm{SAT}.\ \iota \subseteq \llbracket \mathrm{vars}(\Gamma), t \vdash A\rrbracket<\gamma,\iota> \Rightarrow \llbracket \mathrm{vars}(\Gamma), t \vdash B\rrbracket<\gamma,\iota> \subseteq \llbracket \mathrm{vars}(\Gamma), t \vdash B'\rrbracket<\gamma,\iota>$ (b)
Th.: $\forall\gamma\in\llbracket \Gamma \vdash \;\rrbracket.\ \forall f\in\Lambda.\ (\ \forall\iota\in\mathrm{SAT}.\ \iota \subseteq \llbracket \mathrm{vars}(\Gamma), t \vdash A'\rrbracket<\gamma,\iota> \Rightarrow f \in \llbracket \mathrm{vars}(\Gamma), t \vdash B\rrbracket<\gamma,\iota>$ (c)
$\Rightarrow (\forall\iota\in\mathrm{SAT}.\ \iota \subseteq \llbracket \mathrm{vars}(\Gamma), t \vdash A\rrbracket<\gamma,\iota> \Rightarrow f \in \llbracket \mathrm{vars}(\Gamma), t \vdash B'\rrbracket<\gamma,\iota>))$ (d)

Assuming (a), (b), and (c), we prove that (d) holds.

Proof: Let: $f \in \Lambda,\ \gamma \in \llbracket \Gamma \vdash \;\rrbracket,\ \iota \in \mathrm{SAT}$ (e)
Let: $\iota \subseteq \llbracket \mathrm{vars}(\Gamma), t \vdash A\rrbracket<\gamma,\iota>$ (f)
By (e) and (f), and (a): $\llbracket \mathrm{vars}(\Gamma), t \vdash t\rrbracket<\gamma,\iota> \subseteq \llbracket \mathrm{vars}(\Gamma), t \vdash A'\rrbracket<\gamma,\iota>$ (g)
By def. (Var Form) and (g): $\iota \subseteq \llbracket \mathrm{vars}(\Gamma), t \vdash A'\rrbracket<\gamma,\iota>$ (h)
By (e) and (h), and (c): $f \in \llbracket \mathrm{vars}(\Gamma), t \vdash B\rrbracket<\gamma,\iota>$ (l)
By (l), (e) and (f), and (b): $f \in \llbracket \mathrm{vars}(\Gamma), t \vdash B'\rrbracket<\gamma,\iota>$. □

Id and Trans subtyping: soundness of these rules follows from reflexivity and transitivity of set inclusion.

(Id ≤) $\Gamma \vdash \Diamond,\quad \mathrm{vars}(\Gamma) \vdash A \quad\Rightarrow\quad \Gamma \vdash A \le A$

(Trans ≤) $\Gamma \vdash A \le B,\quad \Gamma \vdash B \le C \quad\Rightarrow\quad \Gamma \vdash A \le C.$

3.7 Interpretation and soundness of term judgements

Interpretation: $\forall\gamma\in\llbracket \Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket \Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \llbracket \Gamma, \Delta \vdash a\!: A\rrbracket\gamma\delta = \mathrm{typeErasure}(a)_{\Delta\leftarrow\delta}$

Soundness: $\forall\gamma\in\llbracket \Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket \Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \mathrm{typeErasure}(a)_{\Delta\leftarrow\delta} \in \llbracket \mathrm{vars}(\Gamma) \vdash A\rrbracket\gamma$

(Var) $\Gamma, x_1{:}A_1,\ldots,x_n{:}A_n \vdash \Diamond \quad\Rightarrow\quad \Gamma, x_1{:}A_1,\ldots,x_n{:}A_n \vdash x_i{:}A_i$

Soundness: Hyp.: $\forall\gamma\in\llbracket \Gamma \vdash \Diamond\rrbracket,\ \forall<a_1,\ldots,a_n>\in\llbracket \Gamma, x_1{:}A_1,\ldots,x_n{:}A_n \vdash \Diamond\rrbracket\gamma.\ a_i \in \llbracket \mathrm{vars}(\Gamma) \vdash A_i\rrbracket\gamma$
Th.: $\forall\gamma\in\llbracket \Gamma \vdash \Diamond\rrbracket.\ \forall<a_1,\ldots,a_n>\in\llbracket \Gamma, x_1{:}A_1,\ldots,x_n{:}A_n \vdash \Diamond\rrbracket\gamma.$
$\mathrm{typeErasure}(x_i)[x_1\leftarrow a_1,\ldots,x_n\leftarrow a_n] \in \llbracket \mathrm{vars}(\Gamma) \vdash A_i\rrbracket\gamma$. □

(→ Intro) $\Gamma, \Delta, x{:}A \vdash b{:}\ B \;\Rightarrow\; \Gamma, \Delta \vdash \lambda x{:}A.b{:}\ A\to B$

Here the first saturation condition is used.

Soundness: Hyp.: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket, \forall{<}\delta,a{>}\in\llbracket\Gamma, \Delta, x{:}A \vdash \Diamond\rrbracket\gamma.\ \mathrm{typeErasure}(b)_{(\Delta,\ x:A)\leftarrow<\delta,a>} \in \llbracket\mathrm{vars}(\Gamma) \vdash B\rrbracket\gamma$

Th.: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \mathrm{typeErasure}(\lambda x{:}A.b)_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma) \vdash A\to B\rrbracket\gamma$

By def. (→ Form), Th. may be rewritten as:

Th.: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \forall a\in\llbracket\mathrm{vars}(\Gamma) \vdash A\rrbracket\gamma.\ (\mathrm{typeErasure}(\lambda x{:}A.b)_{\Delta\leftarrow\delta})(a) \in \llbracket\mathrm{vars}(\Gamma) \vdash B\rrbracket\gamma$

Proof: Let: $\gamma \in \llbracket\Gamma \vdash \Diamond\rrbracket,\ \delta \in \llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma,\ a \in \llbracket\mathrm{vars}(\Gamma) \vdash A\rrbracket\gamma$

By def. (ValueEnv): ${<}\delta,a{>} \in \llbracket\Gamma, \Delta, x{:}A \vdash \Diamond\rrbracket\gamma$

By Hyp.: $\mathrm{typeErasure}(b)_{(\Delta,\ x:A)\leftarrow<\delta,a>}\in \llbracket\mathrm{vars}(\Gamma) \vdash B\rrbracket\gamma$

Splitting the substitution: $\mathrm{typeErasure}(b)_{\Delta\leftarrow\delta}[x\leftarrow a] \in \llbracket\mathrm{vars}(\Gamma) \vdash B\rrbracket\gamma$

By cond. Sat_1: $(\lambda x.\mathrm{typeErasure}(b)_{\Delta\leftarrow\delta})(a) = (\mathrm{typeErasure}(\lambda x{:}A.b)_{\Delta\leftarrow\delta})(a) \in \llbracket\mathrm{vars}(\Gamma) \vdash B\rrbracket\gamma$. □

(→ Elim) $\Gamma, \Delta \vdash f{:}\ A\to B,\quad \Gamma, \Delta \vdash a{:}\ A \;\Rightarrow\; \Gamma, \Delta \vdash f(a){:}\ B$

Soundness: Hyp.: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \mathrm{typeErasure}(f)_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma) \vdash A\to B\rrbracket\gamma$

$\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \mathrm{typeErasure}(a)_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma) \vdash A\rrbracket\gamma$

Th.: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \mathrm{typeErasure}(f(a))_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma) \vdash B\rrbracket\gamma$

Proof: By Hyp. and def. (→ Form), $\mathrm{typeErasure}(f)_{\Delta\leftarrow\delta}(\mathrm{typeErasure}(a)_{\Delta\leftarrow\delta}) \in \llbracket\mathrm{vars}(\Gamma) \vdash B\rrbracket\gamma$.
The thesis follows, since $\mathrm{typeErasure}(f(a))_{\Delta\leftarrow\delta} = \mathrm{typeErasure}(f)_{\Delta\leftarrow\delta}(\mathrm{typeErasure}(a)_{\Delta\leftarrow\delta})$. □

(∀ Intro) $\Gamma, t{\le}A, \Delta \vdash b{:}\ B,\quad t\notin FTV(\Delta) \;\Rightarrow\; \Gamma, \Delta \vdash \Lambda t{\le}A.b{:}\ \forall t{\le}A.B$

Soundness: Hyp.: $\forall{<}\gamma,\iota{>}\in\llbracket\Gamma, t{\le}A \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, t{\le}A, \Delta \vdash \Diamond\rrbracket{<}\gamma,\iota{>}.\ \mathrm{typeErasure}(b)_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma), t \vdash B\rrbracket{<}\gamma,\iota{>}$

Th.: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \mathrm{typeErasure}(\Lambda t{\le}A.b)_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma) \vdash \forall t{\le}A.B\rrbracket\gamma$

Applying definitions (≤ Env) and (∀ Form) we can rewrite Hyp. and Th., respectively, as:

Hyp.: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\iota\in SAT.\ \iota \subseteq \llbracket\mathrm{vars}(\Gamma), t \vdash A\rrbracket{<}\gamma,\iota{>}$
$\Rightarrow \forall\delta\in\llbracket\Gamma, t{\le}A, \Delta \vdash \Diamond\rrbracket{<}\gamma,\iota{>}.\ \mathrm{typeErasure}(b)_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma), t \vdash B\rrbracket{<}\gamma,\iota{>}$

Th.: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \forall\iota\in SAT.\ \iota \subseteq \llbracket\mathrm{vars}(\Gamma), t \vdash A\rrbracket{<}\gamma,\iota{>}$
$\Rightarrow \mathrm{typeErasure}(\Lambda t{\le}A.b)_{\Delta\leftarrow\delta} = \mathrm{typeErasure}(b)_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma), t \vdash B\rrbracket{<}\gamma,\iota{>}$

Proof: Let $\gamma \in \llbracket\Gamma \vdash \Diamond\rrbracket,\ \delta \in \llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma,\ \iota \in SAT, \iota \subseteq \llbracket\mathrm{vars}(\Gamma), t \vdash A\rrbracket{<}\gamma,\iota{>}$.
By Lemma ValueEnvWeakening, since $t\notin FTV(\Delta)$, $\delta \in \llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma \Rightarrow \delta \in \llbracket\Gamma, t{\le}A, \Delta \vdash \Diamond\rrbracket{<}\gamma,\iota{>}$.
We obtain $\mathrm{typeErasure}(b)_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma), t \vdash B\rrbracket{<}\gamma,\iota{>}$ by applying Hyp. to γ, ι, and δ. □

(∀ Elim) $\Gamma, \Delta \vdash f{:}\ \forall t{\le}A.B,\quad \Gamma \vdash A' \le A[t\leftarrow A'] \;\Rightarrow\; \Gamma, \Delta \vdash f\{A'\}{:}\ B[t\leftarrow A']$

Soundness: Hyp.: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \mathrm{typeErasure}(f)_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma) \vdash \forall t{\le}A.B\rrbracket\gamma$

$\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \llbracket\mathrm{vars}(\Gamma) \vdash A'\rrbracket\gamma \subseteq \llbracket\mathrm{vars}(\Gamma) \vdash A[t\leftarrow A']\rrbracket\gamma$

Th.: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \mathrm{typeErasure}(\ f\{A'\})_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma) \vdash B[t\leftarrow A']\rrbracket\gamma$

We apply definition (∀ Form) to the first line, and Lemma TypeSubst to the second and third lines:

Hyp.: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \forall\iota\in SAT.$
$\iota \subseteq \llbracket\mathrm{vars}(\Gamma), t \vdash A\rrbracket{<}\gamma,\iota{>} \Rightarrow \mathrm{typeErasure}(f)_{\Delta\leftarrow\delta} = \mathrm{typeErasure}(f\{A'\})_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma), t \vdash B\rrbracket{<}\gamma,\iota{>}$ (a)
$\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \llbracket\mathrm{vars}(\Gamma) \vdash A'\rrbracket\gamma \subseteq \llbracket\mathrm{vars}(\Gamma), t \vdash A\rrbracket{<}\gamma,\llbracket\mathrm{vars}(\Gamma) \vdash A'\rrbracket\gamma{>}$ (b)

Th.: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \mathrm{typeErasure}(f\{A'\})_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma), t \vdash B\rrbracket{<}\gamma,\llbracket\mathrm{vars}(\Gamma) \vdash A'\rrbracket\gamma{>}$

By (b), $\llbracket\mathrm{vars}(\Gamma) \vdash A'\rrbracket\gamma$ can be ι in (a), yielding: $\mathrm{typeErasure}(f\{A'\})_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma), t \vdash B\rrbracket{<}\gamma,\llbracket\mathrm{vars}(\Gamma) \vdash A'\rrbracket\gamma{>}$. □

(Subsump) $\Gamma, \Delta \vdash a{:}\ A,\quad \Gamma \vdash A \le B \;\Rightarrow\; \Gamma, \Delta \vdash\ a{:}\ B$

Hyp.: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \mathrm{typeErasure}(a)_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma) \vdash A\rrbracket\gamma$
$\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \llbracket\mathrm{vars}(\Gamma) \vdash A\rrbracket\gamma \subseteq \llbracket\mathrm{vars}(\Gamma) \vdash B\rrbracket\gamma$

Th.: $\forall\gamma\in\llbracket\Gamma \vdash \Diamond\rrbracket.\ \forall\delta\in\llbracket\Gamma, \Delta \vdash \Diamond\rrbracket\gamma.\ \mathrm{typeErasure}(a)_{\Delta\leftarrow\delta} \in \llbracket\mathrm{vars}(\Gamma) \vdash B\rrbracket\gamma$. □

References

[Bruce 93] K. B. Bruce, “Safe type checking in a statically typed object-oriented programming language”, in *POPL ’93*, 1993.

[Canning et al. 89] P. Canning, W. Cook, W. Hill, J.C. Mitchell, and W. Olthoff, “F-bounded quantification for object-oriented programming”, in *Functional Programming and Computer Architecture*, 273-280, 1989.

[Cardelli et al. 91] L. Cardelli, S. Martini, J.C. Mitchell, and A. Scedrov, “An extension of system F with subtyping”, in *Intl. Conference on Theoretical Aspects of Computer Software*, Sendai, Japan, LNCS 526, 1991. To appear in Information & Computation.

[Cardelli Wegner 85] L. Cardelli and P. Wegner, “On understanding types, data abstraction and polymorphism”, *ACM Computing Surveys*, 17 (4), 1985.

[Curien Ghelli 92] P.-L. Curien and G. Ghelli, “Coherence of Subsumption in $F_{\leq}$, Minimum Typing and Type Checking”, *Mathematical Structures in Computer Science*, 2(1), 1992.

[Ghelli 90] G. Ghelli, “*Proof Theoretic Studies about a Minimal Type System Integrating Inclusion and Parametric Polymorphism*”, PhD Thesis, TD-6/90, Dipartimento di Informatica dell’Università di Pisa, Italy, 1990.

[Girard 72] J.Y. Girard, “*Interprétation fonctionnelle et élimination des coupures dans l’arithmétique d’ordre supérieur*”, Thèse de Doctorat d’Etat, Paris, 1972.

[Katiyar 92] D. Katiyar, “Subtyping F-bounded types”, in *ANSA Workshop on F-bounded quantification, Cambridge*, 1992. Position paper.

[Katiyar et al. 94] D. Katiyar, D. Luckham, and J. Mitchell, “A type system for prototyping languages”, in *POPL ’94*, 1994.

[Mitchell 86] J.C. Mitchell, “A type-inference approach to reduction properties and semantics of polymorphic expressions (summary)”, in *11th ACM Conf. on Lisp and Functional Programming*, 1986.

[Mitchell 90] J.C. Mitchell, “Towards a typed foundation for method specialization and inheritance”, in *POPL ’90*, 1990.