Type Soundness in an Intensional Dependent Type Theory with Type-in-Type and Recursion

February 25, 2021

1 Type Soundness

1.1 Contexts

1.1.1 Sub-Contexts are well formed

The following rules are admissible:

$$\frac{\Gamma, \Gamma' \vdash}{\Gamma \vdash}$$

$$\frac{\Gamma, \Gamma' \vdash M : \sigma}{\Gamma \vdash}$$

$$\frac{\Gamma, \Gamma' \vdash M \Rrightarrow M' : \sigma}{\Gamma \vdash}$$

$$\frac{\Gamma, \Gamma' \vdash M \Rrightarrow_* M' : \sigma}{\Gamma \vdash}$$

$$\frac{\Gamma, \Gamma' \vdash M \equiv M' : \sigma}{\Gamma \vdash}$$

by mutual induction on the derivations.

1.1.2 Context weakening

For any derivation of $\Gamma \vdash \sigma : \star$, the following rules are admissible:

$$\frac{\Gamma, \Gamma' \vdash}{\Gamma, x : \sigma, \Gamma' \vdash}$$

$$\frac{\Gamma, \Gamma' \vdash M : \tau}{\Gamma, x : \sigma, \Gamma' \vdash M : \tau}$$

$$\frac{\Gamma, \Gamma' \vdash M \Rrightarrow M' : \sigma}{\Gamma, x : \sigma, \Gamma' \vdash M \Rrightarrow M' : \sigma}$$

$$\frac{\Gamma, \Gamma' \vdash M \Rrightarrow_* M' : \sigma}{\Gamma, x : \sigma, \Gamma' \vdash M \Rrightarrow_* M' : \sigma}$$

$$\frac{\Gamma, \Gamma' \vdash M \equiv M' : \tau}{\Gamma, x : \sigma, \Gamma' \vdash M \equiv M' : \tau}$$

by mutual induction on the derivations.

1.1.3 $\Rrightarrow$ is reflexive

The following rule is admissible:

$$\frac{\Gamma \vdash M : \sigma}{\Gamma \vdash M \Rrightarrow\ M : \sigma} \Rrightarrow\text{-refl}$$

by induction

1.1.4 $\equiv$ is reflexive

The following rule is admissible:

$$\frac{\Gamma \vdash M : \sigma}{\Gamma \vdash M \equiv M : \sigma} \equiv\text{-refl}$$

by $\Rrightarrow *$-refl

1.1.5 Context substitution

For any derivation of $\Gamma \vdash N : \tau$ the following rules are admissible:

$$\frac{\Gamma, x : \tau, \Gamma' \vdash}{\Gamma, \Gamma' [x \coloneqq N] \vdash}$$

$$\frac{\Gamma, x : \tau, \Gamma' \vdash M : \sigma}{\Gamma, \Gamma' [x \coloneqq N] \vdash M [x \coloneqq N]\ :\ \sigma [x \coloneqq N]}$$

$$\frac{\Gamma, x : \tau, \Gamma' \vdash M \Rrightarrow M' : \sigma}{\Gamma, \Gamma' [x \coloneqq N] \vdash M [x \coloneqq N] \Rrightarrow M' [x \coloneqq N] : \sigma [x \coloneqq N]}$$

$$\frac{\Gamma, x : \tau, \Gamma' \vdash M \Rrightarrow_* M' : \sigma}{\Gamma, \Gamma' [x \coloneqq N] \vdash M [x \coloneqq N] \Rrightarrow_* M' [x \coloneqq N] : \sigma}$$

$$\frac{\Gamma, x : \tau, \Gamma' \vdash M \equiv M' : \sigma}{\Gamma, \Gamma' [x \coloneqq N] \vdash M [x \coloneqq N]\ \equiv\ M' [x \coloneqq N] : \sigma [x \coloneqq N]}$$

by mutual induction on the derivations. Specifically, at every usage of x from the var rule in the original derivation, replace the usage of the var rule with the derivation of $\Gamma \vdash N : \tau$ weakened to the context of $\Gamma, \Gamma' [x \coloneqq N] \vdash N : \tau$, and apply $\Rrightarrow$-refl , $\Rrightarrow_*$-refl or $\equiv$-refl when needed.

1.2 Computation

1.2.1 $\Rrightarrow$ preserves type of source

The following rule is admissible:

$$\frac{\Gamma \vdash N \Rrightarrow N' : \tau}{\Gamma \vdash N : \tau}$$

by induction

1.2.2 $\Rrightarrow$-substitution

The following rule is admissible:

$$\frac{\Gamma, x : \sigma, \Gamma' \vdash M \Rrightarrow M' : \tau \quad \Gamma \vdash N \Rrightarrow N' : \sigma}{\Gamma, \Gamma' [x \coloneqq N] \vdash M [x \coloneqq N] \Rrightarrow M' [x \coloneqq N'] : \tau [x \coloneqq N]}$$

by induction on the $\Rrightarrow$ derivations

1.2.3 $\Rrightarrow$ is confluent

if $\Gamma \vdash M \Rrightarrow N : \sigma$ and $\Gamma \vdash M \Rrightarrow N' : \sigma$ then there exists P such that
$\Gamma \vdash N \Rrightarrow P : \sigma$ and $\Gamma \vdash N' \Rrightarrow P : \sigma$
by standard techniques

1.3 $\Rrightarrow_*$

1.3.1 $\Rrightarrow_*$ is transitive

The following rule is admissible:

$$\frac{\Gamma \vdash M \Rrightarrow_* M' : \sigma \quad \Gamma \vdash M' \Rrightarrow_* M'' : \sigma}{\Gamma \vdash M \Rrightarrow_* M' : \sigma} \Rrightarrow *\text{-trans}$$

by induction

1.3.2 $\Rrightarrow$ preserves type in destination

$$\frac{\Gamma \vdash N \Rrightarrow N' : \tau}{\Gamma \vdash N' : \tau}$$

By induction on the $\Rrightarrow$ derivation with the help of the substitution lemma.

- Π-$\Rrightarrow$
  - $M' [x \coloneqq N', f \coloneqq (\mathsf{fun}\, f : (x.\tau') \,.\, x : \sigma'.M')] \;:\; \tau' [x \coloneqq N']$ by the substitution lemma used on the inductive hypotheses
  - $\tau [x \coloneqq N] \Rrightarrow \tau' [x \coloneqq N']$ by $\Rrightarrow$-substitution , so $\tau [x \coloneqq N] \equiv \tau' [x \coloneqq N']$
  - by the conversion rule $M' [x \coloneqq N', f \coloneqq (\mathsf{fun}\, f : (x.\tau') \,.\, x : \sigma'.M')] \;:\; \tau [x \coloneqq N]$

- Π-E-$\Rrightarrow$
  - $M' \, N' \; : \; \tau \, [x \coloneqq N']$, by $\Rrightarrow$-substitution and reflexivity, $\tau \, [x \coloneqq N] \Rrightarrow \tau \, [x \coloneqq N']$, so $\tau \, [x \coloneqq N] \equiv \tau \, [x \coloneqq N']$
  - by the conversion rule $M' \, N' \; : \; \tau \, [x \coloneqq N]$
- Π-I-$\Rrightarrow$
  - $\mathsf{fun}\, f \; : \; (x.\tau') \, . \, x \; : \; \sigma'.M' \; : \; \Pi x \; : \; \sigma'.\tau'$, $\Pi x \; : \; \sigma.\tau \; \Rrightarrow \; \Pi x \; : \; \sigma'.\tau'$, so $\Pi x : \sigma.\tau \equiv \Pi x : \sigma'.\tau'$
  - by the conversion rule $\mathsf{fun}\, f : (x.\tau') \, . \, x : \sigma'.M' \; : \; \Pi x : \sigma.\tau$
- all other cases are trivial

1.3.3 $\Rrightarrow_*$ preserves type

The following rule is admissible:

$$\frac{\Gamma \vdash M \Rrightarrow_* \; M' : \sigma}{\Gamma \vdash M : \sigma}$$

by induction

$$\frac{\Gamma \vdash M \Rrightarrow_* \; M' : \sigma}{\Gamma \vdash M' : \sigma}$$

by induction

1.3.4 $\Rrightarrow_*$ is confluent

if $\Gamma \vdash M \Rrightarrow_* \; N : \sigma$ and $\Gamma \vdash M \Rrightarrow_* \; N' : \sigma$ then there exists P such that
$\Gamma \vdash N \Rrightarrow_* \; P : \sigma$ and $\Gamma \vdash N' \Rrightarrow_* \; P : \sigma$
Follows from $\Rrightarrow *$-trans and the confluence of $\Rrightarrow$ using standard techniques

1.3.5 $\equiv$ is symmetric

The following rule is admissible:

$$\frac{\Gamma \vdash M \equiv N : \sigma}{\Gamma \vdash N \equiv M : \sigma} \equiv\text{-sym}$$

trivial

1.3.6 $\equiv$ is transitive

$$\frac{\Gamma \vdash M \equiv N : \sigma \qquad \Gamma \vdash N \equiv P : \sigma}{\Gamma \vdash M \equiv P : \sigma} \equiv\text{-trans}$$

by the confluence of $\Rrightarrow_*$

1.3.7 $\equiv$ preserves type

The following rules are admissible:

$$\frac{\Gamma \vdash M \equiv M' : \sigma}{\Gamma \vdash M : \sigma}$$

$$\frac{\Gamma \vdash M \equiv M' : \sigma}{\Gamma \vdash M' : \sigma}$$

by the def of $\Rrightarrow_*$

1.3.8 Regularity

The following rule is admissible:

$$\frac{\Gamma \vdash M : \sigma}{\Gamma \vdash \sigma : \star}$$

by induction with $\equiv$-preservation for the Conv case

1.3.9 $\rightsquigarrow$ implies $\Rrightarrow$

For any derivations of $\Gamma \vdash M : \sigma$, $M \rightsquigarrow M'$

$$\Gamma \vdash M \Rrightarrow M' : \sigma$$

by induction on $\rightsquigarrow$

1.3.10 $\rightsquigarrow$ preserves type

For any derivations of $\Gamma \vdash M : \sigma$, $M \rightsquigarrow M'$,

$$\Gamma \vdash M' : \sigma$$

since $\rightsquigarrow$ implies $\Rrightarrow$ and $\Rrightarrow$ preserves types

1.4 Type constructors

1.4.1 Type constructors are stable

- if $\Gamma \vdash * \Rrightarrow M : \sigma$ then M is $*$
- if $\Gamma \vdash * \Rrightarrow_* M : \sigma$ then M is $*$
- if $\Gamma \vdash \Pi x : \sigma.\tau \Rrightarrow M : \sigma$ then M is $\Pi x : \sigma'.\tau'$ for some σ', τ'
- if $\Gamma \vdash \Pi x : \sigma.\tau \Rrightarrow_* M : \sigma$ then M is $\Pi x : \sigma'.\tau'$ for some σ', τ'

by induction on the respective relations

1.4.2 Type constructors definitionaly unique

There is no derivation of $\Gamma \vdash * \equiv \Pi x : \sigma.\tau \ : \sigma'$ for any $\Gamma, \sigma, \tau, \sigma'$

from $\equiv$ -Def and constructor stability

1.5 Canonical forms

If $\Diamond \vdash v : \sigma$ then

- if σ is $\star$ then v is $\star$ or $\Pi x : \sigma.\tau$
- if σ is $\Pi x : \sigma'.\tau$ for some σ', τ then v is $\mathsf{fun}\, f : (x.\tau')\, .\, x : \sigma''.P'$ for some τ', σ'', P'

By induction on the typing derivation

- Conv,
  - if σ is $\star$ then eventually, it was typed with type-in-type, or Π-F. it could not have been typed by Π-I since constructors are definitionaly unique
  - if σ is $\Pi x : \sigma'.\tau$ then eventually, it was typed with Π-I. it could not have been typed by type-in-type, or Π-F since constructors are definitionaly unique
- type-in-type, $\Diamond \vdash v : \sigma$ is $\Diamond \vdash \star : \star$
- Π-F, $\Diamond \vdash v : \sigma$ is $\Diamond \vdash \Pi x : \sigma.\tau \ : \star$
- Π-I, $\Diamond \vdash v : \sigma$ is $\Diamond \vdash \mathsf{fun}\, f : (x.\tau)\, .\, x : \sigma.M \ : \ \mathit{\Pi} x : \sigma.\tau$
- no other typing rules are applicable

1.6 Progress

$\Diamond \vdash M : \sigma$ implies that M is a value or there exists N such that $M \leadsto N$.

By direct induction on the typing derivation with the help of the canonical forms lemma

Explicitly:

- M is typed by the conversion rule, then by **induction**, M is a value or there exists N such that $M \leadsto N$
- M cannot be typed by the variable rule in the empty context
- M is typed by type-in-type. M is $\star$, a value
- M is typed by Π-F. M is $\Pi x : \sigma.\tau$, a value
- M is typed by Π-I. M is $\mathsf{fun}\, f : (x.\tau)\, .\, x : \sigma.M'$, a value

- M is typed by Π-E. M is $P\,N$ then exist some σ, τ for $\Diamond \vdash P : \Pi x : \sigma.\tau$ and $\Diamond \vdash N : \sigma$. By **induction** (on the P branch of the derivation) P is a value or there exists P' such that $P \leadsto P'$. By **induction** (on the N branch of the derivation) N is a value or there exists N' such that $N \leadsto N'$
  - if P is a value then by **canonical forms**, P is$\mathsf{fun}\, f : (x.\tau) \,.\, x : \sigma.P'$ and
    - if N is a value then the one step reduction is $(\mathsf{fun}\, f : (x.\tau) \,.\, x : \sigma.P')\ N \leadsto P' \left[x \coloneqq N, f \coloneqq \mathsf{fun}\, f : (x.\tau) \,.\, x : \sigma.M\right]$
    - otherwise there exists N' such that $N \leadsto N'$, and the one step reduction is $(\mathsf{fun}\, f : (x.\tau) \,.\, x : \sigma.P')\ N \leadsto (\mathsf{fun}\, f : (x.\tau) \,.\, x : \sigma.P')\ N'$
  - otherwise, there exists P' such that $P \leadsto P'$ and the one step reduction is $P\,N \leadsto P'\,N$

1.7 Type Soundness

For any well typed term in an empty context, no sequence of small step reductions will cause result in a computation to “get stuck”. Either a final value will be reached or further reductions can be taken. This follows by iterating the progress and preservation lemmas.