Type Soundness in an Intensional theory with Type in Type and recursion

December 31, 2020

Examples

logical unsoundness:

$\mathsf{fun}\, f : (x.x)\, .\, x : \star . f\, x \qquad : \Pi x : \star . x$

some constructs

while logically unsound the language is extremely expressive. The following CC constructs are expressible,

$a_1 =_A a_2 \coloneqq \lambda A : \star . \lambda a_1 : A . \lambda a_2 : A . \Pi C : (A \to \star)\, . C\, a_1 \to C\, a_2$

$Unit \coloneqq \Pi A : \star . A \to A$

$tt \coloneqq \lambda A : \star . \lambda a : A . a$

$\bot \coloneqq \Pi x : \star . x$

$\neg A \coloneqq \Pi A : \star .. A \to \bot$

church nats:

$\mathbb{N}_c \coloneqq \Pi A : \star . (A \to A) \to A \to A$

$0_c \coloneqq \lambda A : \star . \lambda s : (A \to A) . \lambda z : A . z$

$1_c \coloneqq \lambda A : \star . \lambda s : (A \to A) . \lambda z : A . s\, z$

$2_c \coloneqq \lambda A : \star . \lambda s : (A \to A) . \lambda z : A . s\, (s\, z)$

...

since there is type in type, a kind of large elimination is possible

$\lambda n : \mathbb{N}_c . n\, \star\, (\lambda - . U)\, \bot$

thus $\neg 1_c =_{\mathbb{N}_c} 0_c$ is provable (in a non trivial way):

$\lambda pr : (\Pi C : (\mathbb{N}_c \to \star)\, . C\, 1_c \to C\, 0_c)\, . pr\, (\lambda n : \mathbb{N}_c . n\, \star\, (\lambda - . U)\, \bot)\, tt \qquad : \neg 1_c =_{\mathbb{N}_c} 0_c$

Properties

Sub-Contexts are well formed

The following rules are admissible:

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

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

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

by mutual induction on the derivations.

Weakening

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

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

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

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

by mutual induction on the derivations.

Substitution

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

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

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

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

by 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 $\Gamma, \Gamma' [x \coloneqq N] \vdash N : \tau$.

Type Preservation for Definitional Equalities

The following rule is admissible:

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

By induction on the Definitional Equality derivation with the help of the substitution lemma.

Explicitly:

- Π-C
  - $(\mathsf{fun}\, f : (x.\tau)\,.\, x : \sigma.M)\ N\ :\ \tau\,[x \coloneqq N]$ by Π-Eand the inductive hypotheses
  - $M\,[x \coloneqq N]\ :\ \tau\,[x \coloneqq N]$ by the substitution lemma used on the inductive hypotheses

Inversion

If $\Gamma \vdash M : \sigma$

if M is	$\star$	then		$\Gamma \vdash$
if M is	$\Pi x : \sigma'.\tau$	then there exists	$\Gamma \vdash \sigma' : \star$	$\Gamma, x : \sigma' \vdash \tau : \star$
if M is	$M'\ N$	then there exists	$\Gamma \vdash M' : \Pi x : \sigma'.\tau$	$\Gamma \vdash N : \sigma'$
if M is	$(\mathsf{fun}\, f : (x.\tau')\,.\, x : \sigma'.M')$	then there exists	$\Gamma, x : \sigma' \vdash \tau' : \star$	$\Gamma, x : \sigma', f : \Pi x : \sigma'.\tau \vdash M' :$

Each case follows from inspection on the typing rules. There are only 2 possibilities, the original typing rule and the conversion rule. The conversion rule must eventually refer to the original typing rule. In any sequence of conversion rules the Definitional Equality is preserved.

Small Steps are definitionally equal

For any derivation of $\Diamond \vdash M : \sigma$, the following rules are admissible:

$$\frac{M \rightsquigarrow M'}{\Diamond \vdash M \equiv M' : \sigma}$$

by induction on the small step rules.

Explicitly:

- Π-C-Step since typing is unique up to Definitional Equality, Π-C matches at the correct type
- Id-C-Step since typing is unique up to Definitional Equality $P \equiv M \equiv N$, Id-C matches at the correct type
- all other computation steps follow like Π-C-Step
- other step rules follow from induction and and the standard Definitional Equalities rules

Type Preservation for small steps

Type preservation at Definitional Equalities and that small steps match definition equalities establish the preservation of types over small steps.

Definitional equality Distinguishes Type Constructors (informally)

for all σ, τ then $\Diamond \nvdash \star \equiv \Pi x : \sigma.\tau \ : \star$

Informally: assume there exists $\Rrightarrow$ such that

$\Gamma \vdash A_1 \equiv A_2 \ : \sigma$

holds iff

$\Gamma \vdash A : \sigma$, $A_1 \Rrightarrow A$ and $A_2 \Rrightarrow A$

We need to presume that the relation $\Rrightarrow$ is confluent, s.t. $B \Rrightarrow A_1$ and $B \Rrightarrow A_2$ then $A_1 \Rrightarrow C$ and $A_2 \Rrightarrow C$, so that refl, sym, trans holds (?)

$\Rrightarrow$ preserves the outermost type former

- $\mathit{\Pi} x : \sigma.\tau \Rrightarrow M$, M is $\mathit{\Pi} x : \sigma'.\tau'$
- $\star \Rrightarrow M$, M is $\star$

thus $\mathit{\Pi} x : \sigma.\tau$ and $\star$ don't share a reduct and $\Diamond \nvdash \star \equiv \Pi x : \sigma.\tau \ : \star$

Canonical forms lemma

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

- 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 $\Diamond \vdash \mathsf{fun}\, f : (x.\tau)\,.\, x : \sigma.M \ : \ \mathit{\Pi} x : \sigma.\tau$
- conv,
  - if σ is $\star$ then eventually, it was typed with type-in-type, or Π-F
  - if σ is $\Pi x : \sigma'.\tau$ then eventually, it was typed with Π-I
- no other typing rules are applicable

Progress

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

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 $M \rightsquigarrow N$
- 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 there exist some σ, τ for $\Diamond \vdash P : \Pi x : \sigma.\tau$ and $\Diamond \vdash N : \sigma$. By the inductive hypothesis (on the P branch of the derivation) P is a value or there exists P' such that $P \rightsquigarrow P'$. By the inductive hypothesis (on the N branch of the derivation) N is a value or there exists N' such that $N \rightsquigarrow N'$
  - if P is a value then by the canonical forms lemma, P is$\mathsf{fun}\, f : (x.\tau)\,.\, x : \sigma.P'$ and
    - if N is a value then the one step reduction is $(\mathsf{fun}\, f : (x.\tau)\,.\, x : \sigma.P')\ N \rightsquigarrow P'\,[x := N, f := \mathsf{fun}\, f : (x.\tau)\,.\, x : \sigma.M]$
    - otherwise there exists N' such that $N \rightsquigarrow N'$, and the one step reduction is $(\mathsf{fun}\, f : (x.\tau)\,.\, x : \sigma.P')\ N \rightsquigarrow (\mathsf{fun}\, f : (x.\tau)\,.\, x : \sigma.P')\ N'$
  - otherwise, there exists P' such that $P \rightsquigarrow P'$ and the one step reduction is $P\,N \rightsquigarrow P'\,N$

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.

Conjectured Properties

- regularity, $\Gamma \vdash M : \sigma$ implies $\Gamma \vdash \sigma : \star$
- $\frac{\Gamma, x:\sigma, \Delta \vdash}{\Gamma \vdash \sigma : \star}$

Non-Properties

- decidable type checking
- normalization/logical soundness

Definitional Equality does not preserve type constructors on the nose

If $\Gamma \vdash \sigma \equiv \sigma' : \star$ then
if σ is $\Pi x : \sigma''.\tau$ for some σ'', τ then σ' is $\Pi x : \sigma'''.\tau'$ for some σ''', τ'
counter example $\vdash \Pi x : \star.\star \equiv (\lambda x : \star.x)(\Pi x : \star.\star) : \star$
this implies the additional work in the Canonical forms lemma