Equivalence of System F and $\lambda 2$: A Case Study of Context Morphisms

Jonas Kaiser Tobias Tebbi Gert Smolka

HOR 2016, Porto

June 25, 2016

Contributions

SAARLAND UNIVERSITY
COMPUTER SCIENCE

A: Strong equivalence of two variants of System F [Girard '72, Reynolds '74]

- F with explicit, separate context for type variables, e.g. [Harper '13]
- $\lambda 2$, a pure type system (PTS) [Barendregt '91]
- Notion of Equivalence: *reduction of type checking* in both directions.

Contributions

SAARLAND UNIVERSITY
COMPUTER SCIENCE

A: Strong equivalence of two variants of System F [Girard '72, Reynolds '74]

- F with explicit, separate context for type variables, e.g. [Harper '13]
- $\lambda 2$, a pure type system (PTS) [Barendregt '91]
- Notion of Equivalence: *reduction of type checking* in both directions.

B: Methodology & Best Practices

- Take syntax with binders seriously - else it will bite you!
- Pursue a *big-step approach*:
  - de Bruijn syntax with parallel substitutions [Schäfer et al. CPP'15/ITP'15]
  - context morphisms [Goguen and McKinna '97, Adams '06]
  - CMs extend to syntax translations
- Considerably shorter proofs (here: 3000 loc $\rightsquigarrow$ 700 loc)

Overview

SAARLAND UNIVERSITY
COMPUTER SCIENCE

1 Setup

- de Bruijn Syntax & Parallel Substitutions
- F
- $\lambda 2$
- Equivalence Statement

2 Reductions

- Challenges
- Problem Decomposition
- Context Morphisms – an Example
- Translating Syntax
- Preservation of Typing
- Cancellation Laws

Syntax and Substitutions

SAARLAND UNIVERSITY
COMPUTER SCIENCE

- de Bruijn Syntax: $s, t := n \mid \lambda.s \mid s\ t$
- Parallel Substitutions: $\sigma, \tau : \mathbb{N} \to \mathcal{T} \qquad \sigma = \sigma(0), \sigma(1), \ldots$
  - $t[\sigma]$ applies σ to all free variables of t simultaneously.
  - Cons operation: $s \cdot \sigma := s, \sigma(0), \sigma(1), \ldots$
  - Application $t[\sigma]$ and composition $\sigma \circ \tau$ are mutually recursive:

$$x[\sigma] = \sigma(x) \qquad (\sigma \circ \tau)(x) = \sigma(x)[\tau]$$
$$(s\ t)[\sigma] = s[\sigma]\ t[\sigma]$$
$$(\lambda.s)[\sigma] = \lambda.s[\Uparrow \sigma] \qquad \textbf{where } \Uparrow \sigma := 0 \cdot \sigma \circ +1$$

  - No separate shifting operation, no unnatural lemma statements.
  - Enables the use of context morphisms.
  - Underlying theory: σ-calculus [Abadi et al. '91, Schäfer et al. CPP'15]
    $\Rightarrow$ algebra with computable, unique normal forms.
- Coq library: Autosubst [Schäfer et al. ITP'15].

Variant: F [Harper '13]

Syntax

Separate syntactic sorts for types and terms, $x : \mathbb{N}$:

$$\begin{array}{lrcl} \mathsf{Ty}_F & A, B, C & := & x_{\mathsf{ty}} \mid A \to B \mid \forall.\, A \\ \mathsf{Ter}_F & s, t & := & x_{\mathsf{ter}} \mid s\, t \mid \lambda A.\, s \mid s\, A \mid \Lambda.\, s \end{array}$$

Note: $t[\tau, \sigma]$ denotes the parallel application of both a type substitution $\tau : \mathbb{N} \to \mathsf{Ty}_F$ and a term substitution $\sigma : \mathbb{N} \to \mathsf{Ter}_F$ to the term t.

Variant: F [Harper '13]

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Syntax

Separate syntactic sorts for types and terms, $x : \mathbb{N}$:

$$\begin{aligned} \mathsf{Ty}_F \qquad A, B, C &:= x_{\mathsf{ty}} \mid A \to B \mid \forall.\, A \\ \mathsf{Ter}_F \qquad s, t &:= x_{\mathsf{ter}} \mid s\, t \mid \lambda A.\, s \mid s\, A \mid \Lambda.\, s \end{aligned}$$

Note: $t[\tau, \sigma]$ denotes the parallel application of both a type substitution $\tau : \mathbb{N} \to \mathsf{Ty}_F$ and a term substitution $\sigma : \mathbb{N} \to \mathsf{Ter}_F$ to the term t.

Type System

Separate judgements for type formation and typing:

$$N \vdash_{\mathsf{F}}^{\mathsf{ty}} A \qquad\qquad N; \Gamma \vdash_{\mathsf{F}}^{\mathsf{ter}} s : A$$

Variant: $\lambda 2$ [Barendregt '91]

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Syntax

Single syntactic sort, $x : \mathbb{N}$:

$$\mathsf{Ter}_P \qquad a, b, c, d \;:=\; u \mid x \mid a\,b \mid \lambda a.\, b \mid \Pi a.\, b \qquad\qquad u \in \{*, \Box\}$$

Variant: $\lambda 2$ [Barendregt '91]

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Syntax

Single syntactic sort, $x : \mathbb{N}$:

$\mathrm{Ter}_P \qquad a, b, c, d \;:=\; u \mid x \mid a\,b \mid \lambda a.\, b \mid \Pi a.\, b \qquad u \in \{*, \Box\}$

Type System

Single judgement:

$$\Gamma \vdash_2 a : b$$

Note: $\Gamma \vdash_2 b : *$ represents type formation.

"To show that the two representations of these systems are in fact the same requires some technical but not difficult work."

Herman Geuvers, '93, doctoral thesis

"To show that the two representations of these systems are in fact the same requires some technical but not difficult work."

Herman Geuvers, '93, doctoral thesis

Theorem (Reduction of Typing[1])

There are syntax translations $\lfloor\cdot\rfloor$ *and* $\lceil\cdot\rceil$*, such that*

$$\vdash_{\mathsf{F}} s : A \iff \vdash_{2} \lfloor s\rfloor : \lfloor A\rfloor$$
$$\vdash_{2} a : b \iff \vdash_{\mathsf{F}} \lceil a\rceil : \lceil b\rceil$$

[1] A similar result holds for Type Formation.

"To show that the two representations of these systems are in fact the same requires some technical but not difficult work."

Herman Geuvers, '93, doctoral thesis

Theorem (Reduction of Typing[1])

There are syntax translations $\lfloor\cdot\rfloor$ *and* $\lceil\cdot\rceil$*, such that*

$$\vdash_F s : A \iff \vdash_2 \lfloor s\rfloor : \lfloor A\rfloor$$
$$\vdash_2 a : b \iff \vdash_F \lceil a\rceil : \lceil b\rceil$$

- Requires 2 preservation laws and 2 cancellation laws.
- Contexts can be internalised – so empty context is sufficiently general.

[1] A similar result holds for Type Formation.

Challenges

SAARLAND UNIVERSITY
COMPUTER SCIENCE

1. The inference systems do not match up.

Challenges

1. The inference systems do not match up.
2. Different Syntaxes:
   - single-sorted (x) vs. two-sorted (x_{ty} and x_{ter})
   - uniform ($\Pi a.\, b$) vs. non-uniform ($A \to B$ and $\forall.\, A$)
   - different expressivity (prior to typing), e.g. $*, * \vdash_{\mathbf{2}} 1\,0 : 0$.

Challenges

1. The inference systems do not match up.
2. Different Syntaxes:
   - single-sorted (x) vs. two-sorted (x_{ty} and x_{ter})
   - uniform ($\Pi a.\, b$) vs. non-uniform ($A \to B$ and $\forall.\, A$)
   - different expressivity (prior to typing), e.g. $*, * \vdash_{\mathbf{2}} 1\,0 : 0$.
3. Contexts cannot be put into 1-1 correspondence:

Challenges

1. The inference systems do not match up.
2. Different Syntaxes:
   - single-sorted (x) vs. two-sorted (x_{ty} and x_{ter})
   - uniform ($\Pi a.\, b$) vs. non-uniform ($A \to B$ and $\forall.\, A$)
   - different expressivity (prior to typing), e.g. $*, * \vdash_{\mathbf{2}} 1\,0 : 0$.
3. Contexts cannot be put into 1-1 correspondence:

$$1; (\forall.\, 0_t\, y) \vdash_{\mathsf{F}}^{\mathsf{ter}} 0_{\mathsf{ter}}\, 0_{\mathsf{ty}} : 0_{\mathsf{ty}}$$

Challenges

SAARLAND UNIVERSITY
COMPUTER SCIENCE

1. The inference systems do not match up.
2. Different Syntaxes:
   - single-sorted (x) vs. two-sorted (x_{ty} and x_{ter})
   - uniform ($\Pi a.\, b$) vs. non-uniform ($A \to B$ and $\forall.\, A$)
   - different expressivity (prior to typing), e.g. $*, * \vdash_{\mathsf{2}} 1\,0 : 0$.
3. Contexts cannot be put into 1-1 correspondence:

$$*, (\Pi *.\, 0) \vdash_{\mathsf{2}} 0\,1 : 1$$

$$1; (\forall.\, 0_t y) \vdash_{\mathsf{F}}^{\mathsf{ter}} 0_{\mathsf{ter}}\, 0_{\mathsf{ty}} : 0_{\mathsf{ty}}$$

$$(\Pi *.\, 0), * \vdash_{\mathsf{2}} 1\,0 : 0$$

Challenges

1. The inference systems do not match up.
2. Different Syntaxes:
   - single-sorted (x) vs. two-sorted (x_{ty} and x_{ter})
   - uniform ($\Pi a.\, b$) vs. non-uniform ($A \to B$ and $\forall.\, A$)
   - different expressivity (prior to typing), e.g. $*, * \vdash_{\mathsf{2}} 1\,0 : 0$.
3. Contexts cannot be put into 1-1 correspondence:

$$*, (\Pi *.\, 0) \vdash_{\mathsf{2}} 0\,1 : 1$$

$$(\Pi *.\, 0), * \vdash_{\mathsf{2}} 1\,0 : 0$$

??

$$1; (\forall.\, 0_t\, y) \vdash_{\mathsf{F}}^{\mathsf{ter}} 0_{\mathsf{ter}}\, 0_{\mathsf{ty}} : 0_{\mathsf{ty}}$$

Challenges

1. The inference systems do not match up.
2. Different Syntaxes:
   - single-sorted (x) vs. two-sorted (x_{ty} and x_{ter})
   - uniform ($\Pi a.\, b$) vs. non-uniform ($A \to B$ and $\forall.\, A$)
   - different expressivity (prior to typing), e.g. $*, * \vdash_2 1\,0 : 0$.
3. Contexts cannot be put into 1-1 correspondence:

$$*, (\Pi *.\, 0) \vdash_2 0\,1 : 1 \qquad (\Pi *.\, 0), * \vdash_2 1\,0 : 0 \qquad ?? \qquad 1; (\forall.\, 0_t y) \vdash_{\mathsf{F}}^{\mathsf{ter}} 0_{\mathsf{ter}}\, 0_{\mathsf{ty}} : 0_{\mathsf{ty}}$$

This is infeasible: $N; \Gamma \vdash_{\mathsf{F}}^{\mathsf{ter}} s : A \;\to\; \lfloor N; \Gamma \rfloor \vdash_2 \lfloor s \rfloor : \lfloor A \rfloor$

Challenges

SAARLAND UNIVERSITY
COMPUTER SCIENCE

1. The inference systems do not match up.
2. Different Syntaxes:
   - single-sorted (x) vs. two-sorted (x_{ty} and x_{ter})
   - uniform ($\Pi a.\, b$) vs. non-uniform ($A \to B$ and $\forall.\, A$)
   - different expressivity (prior to typing), e.g. $*, * \vdash_{\mathsf{2}} 1\,0 : 0$.
3. Contexts cannot be put into 1-1 correspondence:

$$*, (\Pi *.\, 0) \vdash_2 0\,1 : 1 \qquad \overset{??}{\longleftrightarrow} \qquad 1; (\forall.\, 0_t y) \vdash_{\mathsf{F}}^{\mathsf{ter}} 0_{\mathsf{ter}}\, 0_{\mathsf{ty}} : 0_{\mathsf{ty}}$$

$$(\Pi *.\, 0), * \vdash_2 1\,0 : 0$$

This is infeasible: $N; \Gamma \vdash_{\mathsf{F}}^{\mathsf{ter}} s : A \;\to\; \lfloor N; \Gamma \rfloor \vdash_2 \lfloor s \rfloor : \lfloor A \rfloor$

4. Minimal lemma generalisations are often unwieldy.

Challenges

1. The inference systems do not match up.
2. Different Syntaxes:
   - single-sorted (x) vs. two-sorted (x_{ty} and x_{ter})
   - uniform ($\Pi a.\, b$) vs. non-uniform ($A \to B$ and $\forall.\, A$)
   - different expressivity (prior to typing), e.g. $*, * \vdash_2 1\,0 : 0$.
3. Contexts cannot be put into 1-1 correspondence:

$$*, (\Pi *.\, 0) \vdash_2 0\,1 : 1 \quad \overset{??}{\longleftrightarrow} \quad 1; (\forall.\, 0_t y) \vdash_{\mathsf{F}}^{\mathsf{ter}} 0_{\mathsf{ter}}\, 0_{\mathsf{ty}} : 0_{\mathsf{ty}}$$

$$(\Pi *.\, 0), * \vdash_2 1\,0 : 0$$

This is infeasible: $N; \Gamma \vdash_{\mathsf{F}}^{\mathsf{ter}} s : A \;\to\; \lfloor N; \Gamma \rfloor \vdash_2 \lfloor s \rfloor : \lfloor A \rfloor$

4. Minimal lemma generalisations are often unwieldy.

Context Morphisms to the rescue!

Mismatch of Inference Systems

SAARLAND UNIVERSITY
COMPUTER SCIENCE

$$\frac{N; \Gamma, A \vdash_{\mathsf{F}}^{\mathsf{ter}} s : B \qquad N \vdash_{\mathsf{F}}^{\mathsf{ty}} A}{N; \Gamma \vdash_{\mathsf{F}}^{\mathsf{ter}} \lambda A.\, s : A \to B} \qquad \frac{(N+1); \Gamma[+1] \vdash_{\mathsf{F}}^{\mathsf{ter}} s : A}{N; \Gamma \vdash_{\mathsf{F}}^{\mathsf{ter}} \Lambda.\, s : \forall.\, A}$$

$$\frac{\Gamma, a \vdash_2 b : c \qquad \Gamma \vdash_2 a : u \qquad \Gamma, a \vdash_2 c : *}{\Gamma \vdash_2 \lambda a.\, b : \Pi a.\, c}$$

Mismatch of Inference Systems

SAARLAND UNIVERSITY
COMPUTER SCIENCE

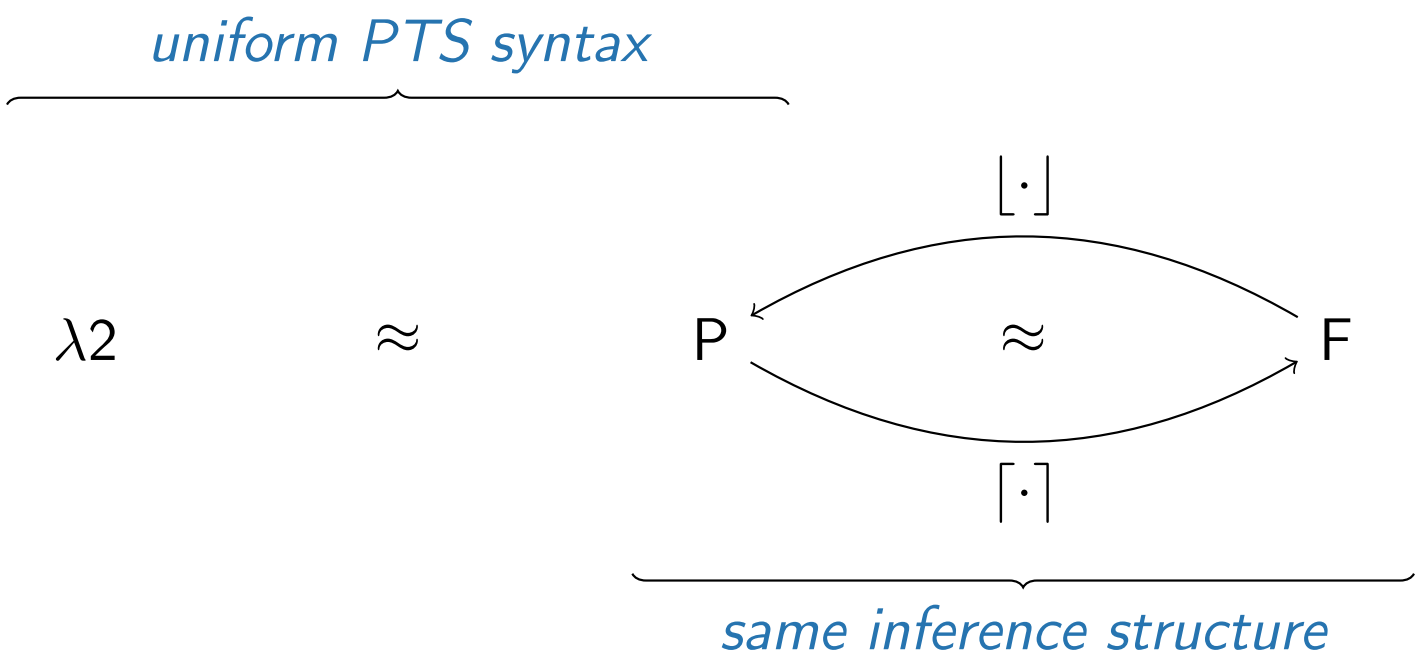
$$\frac{N; \Gamma, A \vdash_{\mathsf{F}}^{\mathsf{ter}} s : B \qquad N \vdash_{\mathsf{F}}^{\mathsf{ty}} A}{N; \Gamma \vdash_{\mathsf{F}}^{\mathsf{ter}} \lambda A.\, s : A \to B} \qquad \frac{(N+1); \Gamma[+1] \vdash_{\mathsf{F}}^{\mathsf{ter}} s : A}{N; \Gamma \vdash_{\mathsf{F}}^{\mathsf{ter}} \Lambda.\, s : \forall.\, A}$$

$$\frac{\Gamma, a \vdash_{\mathsf{P}}^{\mathsf{ter}} b : c \qquad \Gamma \vdash_{\mathsf{P}}^{\mathsf{ty}} a}{\Gamma \vdash_{\mathsf{P}}^{\mathsf{ter}} \lambda a.\, b : \Pi a.\, c} \qquad \frac{\Gamma, * \vdash_{\mathsf{P}}^{\mathsf{ter}} a : b}{\Gamma \vdash_{\mathsf{P}}^{\mathsf{ter}} \lambda *.\, a : \Pi *.\, b}$$

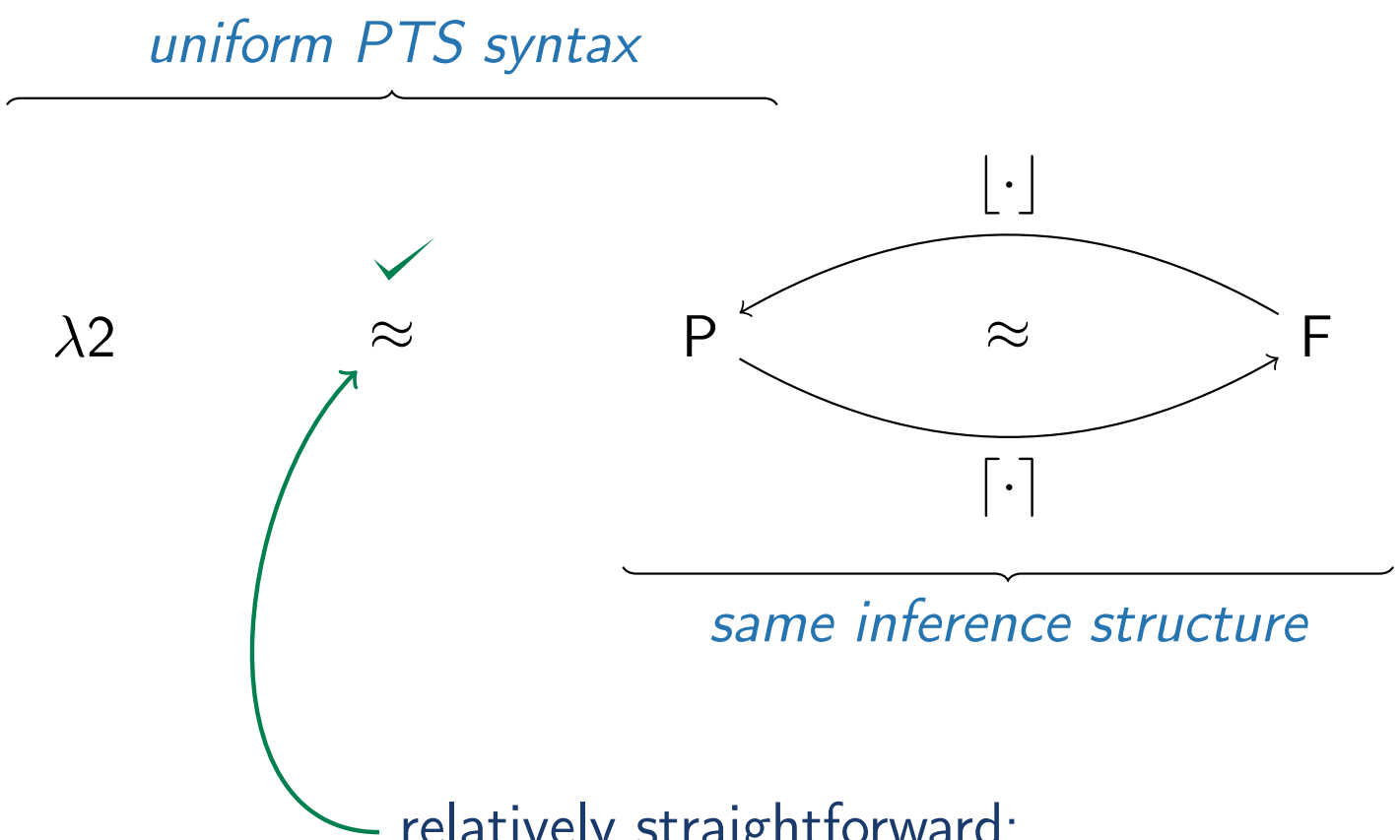
$$\frac{\Gamma, a \vdash_2 b : c \qquad \Gamma \vdash_2 a : u \qquad \Gamma, a \vdash_2 c : *}{\Gamma \vdash_2 \lambda a.\, b : \Pi a.\, c}$$

We introduce an auxiliary type system P on the PTS syntax of $\lambda 2$ that matches F.

Problem Decomposition



Problem Decomposition



Problem Decomposition

SAARLAND UNIVERSITY
COMPUTER SCIENCE

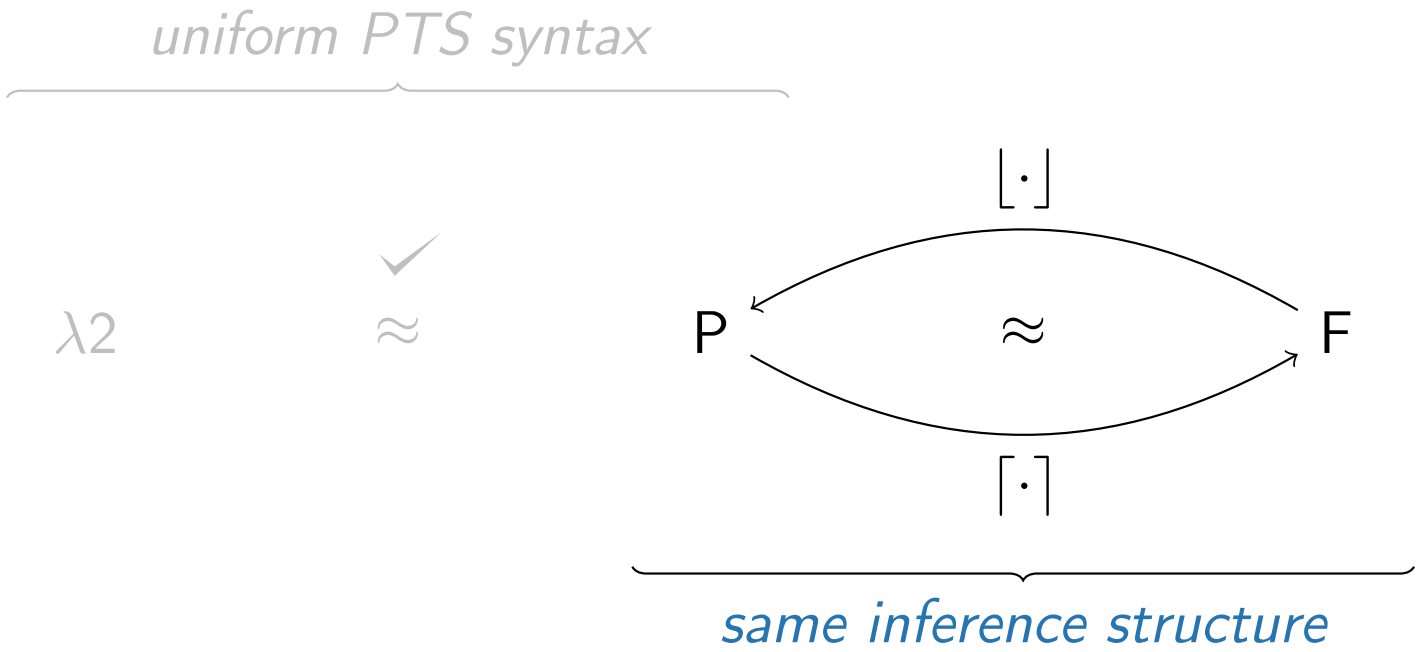
uniform PTS syntax

$\lambda 2$ $\quad \checkmark \approx$ $\quad$ P $\lfloor \cdot \rfloor$ $\approx$ $\lceil \cdot \rceil$ F

same inference structure

Problem Decomposition

SAARLAND UNIVERSITY
COMPUTER SCIENCE



Theorem (PF-Reduction of Typing Problem under Translations)

(1) $\vdash^{\mathsf{ter}}_{\mathsf{F}} s : A \iff \vdash^{\mathsf{ter}}_{\mathsf{P}} \lfloor s \rfloor : \lfloor A \rfloor$

(2) $\vdash^{\mathsf{ter}}_{\mathsf{P}} a : b \iff \vdash^{\mathsf{ter}}_{\mathsf{F}} \lceil a \rceil^{\langle\rangle}_{\mathsf{ter}} : \lceil b \rceil^{\langle\rangle}_{\mathsf{ty}}$

Example: Weakening

SAARLAND UNIVERSITY
COMPUTER SCIENCE

the standard/painful way: minimal generalisation

$$\frac{\Gamma \vdash s : t}{\Gamma, u \vdash s[+1] : t[+1]}$$

Example: Weakening

SAARLAND UNIVERSITY
COMPUTER SCIENCE

the standard/painful way: minimal generalisation

$$\frac{\Gamma_1, \Gamma_2 \vdash s : t}{\Gamma_1, u, \Gamma_2 \vdash s[+1] : t[+1]}$$

Example: Weakening

SAARLAND UNIVERSITY
COMPUTER SCIENCE

the standard/painful way: minimal generalisation

$$\frac{\Gamma_1, \Gamma_2 \vdash s : t}{\Gamma_1, u, \Gamma_2[\Uparrow^{?}(+1)] \vdash s[\Uparrow^{|\Gamma_2|}(+1)] : t[\Uparrow^{|\Gamma_2|}(+1)]}$$

Example: Weakening

SAARLAND UNIVERSITY
COMPUTER SCIENCE

the standard/painful way: minimal generalisation

$$\frac{\Gamma_1, \Gamma_2 \vdash s : t}{\Gamma_1, u, \Gamma_2[\Uparrow^{?}(+1)] \vdash s[\Uparrow^{|\Gamma_2|}(+1)] : t[\Uparrow^{|\Gamma_2|}(+1)]}$$

- Issue 1: arithmetic, e.g. $x \overset{?}{<} |\Gamma_2|$
- Issue 2: the operation $\Gamma_2[\Uparrow^{?}(+1)]$

let $\Gamma = u_{|\Gamma|-1}, u_{|\Gamma|-2}, \ldots, u_0$

then $\Gamma[\Uparrow^{?}(+1)] = u_{|\Gamma|-1}[\Uparrow^{0}(+1)], u_{|\Gamma|-2}[\Uparrow^{1}(+1)], \ldots, u_0[\Uparrow^{|\Gamma|-1}(+1)]$

Example: Weakening

SAARLAND UNIVERSITY
COMPUTER SCIENCE

the elegant way: maximal generalisation

$$\frac{\Gamma \vdash s : t}{\Gamma, u \vdash s[+1] : t[+1]}$$

Example: Weakening

SAARLAND UNIVERSITY
COMPUTER SCIENCE

the elegant way: maximal generalisation

$$\frac{\Gamma \vdash s : t}{\Delta \vdash s[\sigma] : t[\sigma]}$$

Example: Weakening

SAARLAND UNIVERSITY
COMPUTER SCIENCE

the elegant way: maximal generalisation

$$\frac{\Gamma \vdash s : t \qquad \forall (x : u) \in \Gamma.\ \ \Delta \vdash x[\sigma] : u[\sigma]}{\Delta \vdash s[\sigma] : t[\sigma]}$$

Example: Weakening

SAARLAND UNIVERSITY
COMPUTER SCIENCE

the elegant way: maximal generalisation

$$\frac{\Gamma \vdash s : t \qquad \forall (x : u) \in \Gamma.\ \ \Delta \vdash x[\sigma] : u[\sigma]}{\Delta \vdash s[\sigma] : t[\sigma]}$$

- Extra premise *quantifies over initial context* Γ.
  $\Rightarrow$ holds vacuously for $\Gamma = \langle\rangle$.
- Fully specifies the behaviour of σ at *relevant variables*.
- Lemma lifts this from variables to *terms*.

Example: Weakening

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Definition (Context Morphism)

$$\vdash \sigma : \Gamma \to \Delta \quad := \quad \forall (x : u) \in \Gamma. \quad \Delta \vdash x[\sigma] : u[\sigma]$$

$$\frac{\Gamma \vdash s : t \qquad \vdash \sigma : \Gamma \to \Delta}{\Delta \vdash s[\sigma] : t[\sigma]}$$

Example: Weakening

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Definition (Context Morphism)

$$\vdash \sigma : \Gamma \to \Delta \quad := \quad \forall (x : u) \in \Gamma. \quad \Delta \vdash x[\sigma] : u[\sigma]$$

$$\frac{\Gamma \vdash s : t \qquad \vdash \sigma : \Gamma \to \Delta}{\Delta \vdash s[\sigma] : t[\sigma]}$$

Concept

Given contexts Γ and Δ and a judgement $\vdash$, we say that a substitution σ is a *context morphism* from Γ to Δ if it maps *variable judgements* under Γ to judgements under Δ, written $\vdash \sigma : \Gamma \to \Delta$.

Example: Weakening

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Definition (Context Morphism)

$$\vdash \sigma : \Gamma \to \Delta \quad := \quad \forall (x : u) \in \Gamma. \quad \Delta \vdash x[\sigma] : u[\sigma]$$

$$\frac{\Gamma \vdash s : t \qquad \vdash \sigma : \Gamma \to \Delta}{\Delta \vdash s[\sigma] : t[\sigma]}$$

Key Properties of CMs

$$\frac{}{\vdash +1 : \Gamma \to \Gamma, u} \qquad \frac{\vdash \sigma : \Gamma \to \Delta}{\vdash \Uparrow \sigma : \Gamma, u \to \Delta, u[\sigma]}$$

Example: Weakening

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Definition (Context Morphism)

$$\vdash \sigma : \Gamma \to \Delta \quad := \quad \forall (x : u) \in \Gamma. \quad \Delta \vdash x[\sigma] : u[\sigma]$$

$$\frac{\Gamma \vdash s : t \qquad \overline{\vdash +1 : \Gamma \to \Gamma, u}}{\Gamma, u \vdash s[+1] : t[+1]}$$

Key Properties of CMs

$$\frac{}{\vdash +1 : \Gamma \to \Gamma, u} \qquad \frac{\vdash \sigma : \Gamma \to \Delta}{\vdash \Uparrow \sigma : \Gamma, u \to \Delta, u[\sigma]}$$

From F to PTS Syntax

SAARLAND UNIVERSITY
COMPUTER SCIENCE

$\lfloor \cdot \rfloor : \mathrm{Ty}_F \to \mathrm{Ter}_P$

$\lfloor \cdot \rfloor : \mathrm{Ter}_F \to \mathrm{Ter}_P$

From F to PTS Syntax

SAARLAND UNIVERSITY
COMPUTER SCIENCE

$\lfloor \cdot \rfloor : \mathrm{Ty}_F \to \mathrm{Ter}_P$

$$\begin{aligned} \lfloor x_{\mathrm{ty}} \rfloor &:= x \\ \lfloor A \to B \rfloor &:= \Pi \lfloor A \rfloor .\ \lfloor B \rfloor [+1] \\ \lfloor \forall .\, A \rfloor &:= \Pi * .\ \lfloor A \rfloor \end{aligned}$$

$\lfloor \cdot \rfloor : \mathrm{Ter}_F \to \mathrm{Ter}_P$

Note: renamings are essential to avoid namespace collision for bound vars.

From F to PTS Syntax

SAARLAND UNIVERSITY
COMPUTER SCIENCE

$\lfloor \cdot \rfloor : \mathrm{Ty}_F \to \mathrm{Ter}_P$

$$
\begin{aligned}
\lfloor x_{\mathrm{ty}} \rfloor &:= x \\
\lfloor A \to B \rfloor &:= \Pi \lfloor A \rfloor .\ \lfloor B \rfloor [+1] \\
\lfloor \forall .\, A \rfloor &:= \Pi * .\ \lfloor A \rfloor
\end{aligned}
$$

$\lfloor \cdot \rfloor : \mathrm{Ter}_F \to \mathrm{Ter}_P$

$$
\begin{aligned}
\lfloor x_{\mathrm{ter}} \rfloor &:= x \\
\lfloor s\ t \rfloor &:= \lfloor s \rfloor\ \lfloor t \rfloor \\
\lfloor \lambda A.\, s \rfloor &:= \lambda \lfloor A \rfloor .\ \lfloor s[+1, \mathrm{id}] \rfloor \\
\lfloor s\ A \rfloor &:= \lfloor s \rfloor\ \lfloor A \rfloor \\
\lfloor \Lambda .\, s \rfloor &:= \lambda * .\ \lfloor s[\mathrm{id}, +1] \rfloor
\end{aligned}
$$

Note: renamings are essential to avoid namespace collision for bound vars.

From PTS to F Syntax

SAARLAND UNIVERSITY
COMPUTER SCIENCE

$\lceil \cdot \rceil^{\Gamma}_{\text{ty}} : \text{Ter}_P \to \text{option Ty}_F$

$\lceil \cdot \rceil^{\Gamma}_{\text{ter}} : \text{Ter}_P \to \text{option Ter}_F$

From PTS to F Syntax

SAARLAND UNIVERSITY
COMPUTER SCIENCE

$\lceil\cdot\rceil^{\Gamma}_{\text{ty}} : \text{Ter}_P \to \text{option Ty}_F$

$$\lceil x\rceil^{\Gamma}_{\text{ty}} := x_{\text{ty}} \qquad \text{if } x : * \in \Gamma$$
$$\lceil \Pi *.\, a\rceil^{\Gamma}_{\text{ty}} := \forall.\, \lceil a\rceil^{\Gamma,*}_{\text{ty}}$$
$$\lceil \Pi a.\, b\rceil^{\Gamma}_{\text{ty}} := \lceil a\rceil^{\Gamma}_{\text{ty}} \to \lceil b\rceil^{\Gamma,a}_{\text{ty}}\,[\text{-}1]$$

$\lceil\cdot\rceil^{\Gamma}_{\text{ter}} : \text{Ter}_P \to \text{option Ter}_F$

From PTS to F Syntax

SAARLAND UNIVERSITY
COMPUTER SCIENCE

$\lceil\cdot\rceil^{\Gamma}_{\text{ty}} : \text{Ter}_P \to \text{option Ty}_F$

$$\lceil x\rceil^{\Gamma}_{\text{ty}} := x_{\text{ty}} \qquad \text{if } x : * \in \Gamma$$
$$\lceil \Pi *.\, a\rceil^{\Gamma}_{\text{ty}} := \forall.\, \lceil a\rceil^{\Gamma,*}_{\text{ty}}$$
$$\lceil \Pi a.\, b\rceil^{\Gamma}_{\text{ty}} := \lceil a\rceil^{\Gamma}_{\text{ty}} \to \lceil b\rceil^{\Gamma,a}_{\text{ty}}[\text{-1}]$$

$\lceil\cdot\rceil^{\Gamma}_{\text{ter}} : \text{Ter}_P \to \text{option Ter}_F$

$$\lceil x\rceil^{\Gamma}_{\text{ter}} := x_{\text{ter}} \qquad \text{if } x : a \in \Gamma, a \neq *$$
$$\lceil a\, b\rceil^{\Gamma}_{\text{ter}} := \lceil a\rceil^{\Gamma}_{\text{ter}}\ \lceil b\rceil^{\Gamma}_{\text{ty}}$$
$$\lceil a\, b\rceil^{\Gamma}_{\text{ter}} := \lceil a\rceil^{\Gamma}_{\text{ter}}\ \lceil b\rceil^{\Gamma}_{\text{ter}}$$
$$\lceil \lambda *.\, a\rceil^{\Gamma}_{\text{ter}} := \Lambda.\, \lceil a\rceil^{\Gamma,*}_{\text{ter}}[\text{id}, \text{-1}]$$
$$\lceil \lambda a.\, b\rceil^{\Gamma}_{\text{ter}} := \lambda\, \lceil a\rceil^{\Gamma}_{\text{ty}}.\, \lceil b\rceil^{\Gamma,a}_{\text{ter}}[\text{-1}, \text{id}]$$

Generalised Context Morphisms

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Concept

Given contexts Γ and Δ and two judgements $\vdash_1$ and $\vdash_2$, we say that a (multi-sorted) substitution $\overline{\sigma}$ is a *generalised context morphism* from Γ to Δ if it maps *variable judgements* under $(\,\Gamma \vdash_1)$ to judgements under $(\,\Delta \vdash_2)$, written $\overline{\sigma} : (\,\Gamma \vdash_1) \to (\,\Delta \vdash_2)$.

Generalised Context Morphisms

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Concept

Given contexts Γ and Δ and two judgements $\vdash_1$ and $\vdash_2$, we say that a (multi-sorted) substitution $\overline{\sigma}$ is a *generalised context morphism* from Γ to Δ if it maps *variable judgements* under $(\,\Gamma \vdash_1)$ to judgements under $(\,\Delta \vdash_2)$, written $\overline{\sigma} : (\,\Gamma \vdash_1) \to (\,\Delta \vdash_2)$.

Definition (CM for PF Term-Translation)

$$\xi, \zeta : (\,\Gamma \vdash_{\mathsf{P}}^{\mathsf{ter}}) \to (\,N; \Delta \vdash_{\mathsf{F}}^{\mathsf{ter}}) :=$$
$$\forall x a.\quad \Gamma \vdash_{\mathsf{P}}^{\mathsf{ter}} x : a \;\to\; N; \Delta \vdash_{\mathsf{F}}^{\mathsf{ter}} \zeta(x) : \lceil a \rceil_{\mathsf{ty}}^{\Gamma}[\xi]$$

Generalised Context Morphisms

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Concept

Given contexts Γ and Δ and two judgements $\vdash_1$ and $\vdash_2$, we say that a (multi-sorted) substitution $\overline{\sigma}$ is a *generalised context morphism* from Γ to Δ if it maps *variable judgements* under $(\,\Gamma \vdash_1\,)$ to judgements under $(\,\Delta \vdash_2\,)$, written $\overline{\sigma} : (\,\Gamma \vdash_1\,) \to (\,\Delta \vdash_2\,)$.

Definition (CM for PF Term-Translation)

$$\xi, \zeta : (\,\Gamma \vdash_{\mathsf{P}}^{\mathsf{ter}}\,) \to (\,N; \Delta \vdash_{\mathsf{F}}^{\mathsf{ter}}\,) :=$$
$$\forall x a.\quad \Gamma \vdash_{\mathsf{P}}^{\mathsf{ter}} x : a \;\to\; N; \Delta \vdash_{\mathsf{F}}^{\mathsf{ter}} \zeta(x) : \lceil a \rceil_{\mathsf{ty}}^{\Gamma}[\xi]$$

Key Properties have to be generalised accordingly, e.g.

$$\frac{\xi, \zeta : (\,\Gamma \vdash_{\mathsf{P}}^{\mathsf{ter}}\,) \to (\,N; \Delta \vdash_{\mathsf{F}}^{\mathsf{ter}}\,)}{\Uparrow\xi, \zeta(0) \cdot \zeta : (\,\Gamma, * \vdash_{\mathsf{P}}^{\mathsf{ter}}\,) \to (\,N+1; \Delta[+1] \vdash_{\mathsf{F}}^{\mathsf{ter}}\,)}$$

Preservation of Typing

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Lemma (Preservation of Typing under Translations)

$$\frac{N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}} s : A \qquad \xi : (N \vdash^{\mathsf{ty}}_{\mathsf{F}}) \to (\Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}}) \qquad \xi, \zeta : (N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}}) \to (\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}})}{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} \lfloor s[\xi, \zeta] \rfloor : \lfloor A \rfloor [\xi]}$$

$$\frac{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} a : b \qquad \xi : (\Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}}) \to (N \vdash^{\mathsf{ty}}_{\mathsf{F}}) \qquad \xi, \zeta : (\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}}) \to (N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}})}{N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}} \lceil a \rceil^{\Gamma}_{\mathsf{ter}} [\xi, \zeta] : \lceil b \rceil^{\Gamma}_{\mathsf{ty}} [\xi]}$$

Preservation of Typing

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Lemma (Preservation of Typing under Translations)

$$\frac{N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}} s : A \qquad \xi : (N \vdash^{\mathsf{ty}}_{\mathsf{F}}) \to (\Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}}) \qquad \xi, \zeta : (N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}}) \to (\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}})}{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} \lfloor s[\xi, \zeta] \rfloor : \lfloor A \rfloor [\xi]}$$

$$\frac{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} a : b \qquad \xi : (\Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}}) \to (N \vdash^{\mathsf{ty}}_{\mathsf{F}}) \qquad \xi, \zeta : (\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}}) \to (N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}})}{N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}} \lceil a \rceil^{\Gamma}_{\mathsf{ter}} [\xi, \zeta] : \lceil b \rceil^{\Gamma}_{\mathsf{ty}} [\xi]}$$

- Proof by structural induction on the respective first premise.

Preservation of Typing

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Lemma (Preservation of Typing under Translations)

$$\frac{N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}} s : A \qquad \xi : (N \vdash^{\mathsf{ty}}_{\mathsf{F}}) \to (\Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}}) \qquad \xi, \zeta : (N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}}) \to (\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}})}{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} \lfloor s[\xi, \zeta] \rfloor : \lfloor A \rfloor [\xi]}$$

$$\frac{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} a : b \qquad \xi : (\Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}}) \to (N \vdash^{\mathsf{ty}}_{\mathsf{F}}) \qquad \xi, \zeta : (\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}}) \to (N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}})}{N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}} \lceil a \rceil^{\Gamma}_{\mathsf{ter}} [\xi, \zeta] : \lceil b \rceil^{\Gamma}_{\mathsf{ty}} [\xi]}$$

- Proof by structural induction on the respective first premise.
- Requires a total of 8 morph. rules.

Preservation of Typing

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Lemma (Preservation of Typing under Translations)

$$\frac{N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}} s : A \qquad \xi : (N \vdash^{\mathsf{ty}}_{\mathsf{F}}) \to (\Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}}) \qquad \xi, \zeta : (N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}}) \to (\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}})}{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} \lfloor s[\xi, \zeta] \rfloor : \lfloor A \rfloor [\xi]}$$

$$\frac{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} a : b \qquad \xi : (\Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}}) \to (N \vdash^{\mathsf{ty}}_{\mathsf{F}}) \qquad \xi, \zeta : (\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}}) \to (N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}})}{N; \Delta \vdash^{\mathsf{ter}}_{\mathsf{F}} \lceil a \rceil^{\Gamma}_{\mathsf{ter}} [\xi, \zeta] : \lceil b \rceil^{\Gamma}_{\mathsf{ty}} [\xi]}$$

- Proof by structural induction on the respective first premise.
- Requires a total of 8 morph. rules.
- These settle the forward implications of our main theorem.

Cancellation Laws

$$\frac{\lceil a[\xi] \rceil^{\Gamma}_{\mathsf{ter}} = s}{\lfloor s \rfloor = a[\xi]} \qquad \frac{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} \lfloor s[\xi, \zeta] \rfloor : c \qquad \Gamma \vdash_{\mathsf{P}} \xi \parallel \zeta}{\lceil \lfloor s[\xi, \zeta] \rfloor \rceil^{\Gamma}_{\mathsf{ter}} = s[\xi, \zeta]}$$

- where $\Gamma \vdash_{\mathsf{P}} \xi \parallel \zeta$ expresses that, according to Γ, ξ only yields type variables and ζ only yields term variables.

Cancellation Laws

SAARLAND UNIVERSITY
COMPUTER SCIENCE

$$\frac{\lceil a[\xi] \rceil^{\Gamma}_{\mathsf{ter}} = s}{\lfloor s \rfloor = a[\xi]} \qquad \frac{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} \lfloor s[\xi, \zeta] \rfloor : c \qquad \Gamma \vdash_{\mathsf{P}} \xi \parallel \zeta}{\lceil \lfloor s[\xi, \zeta] \rfloor \rceil^{\Gamma}_{\mathsf{ter}} = s[\xi, \zeta]}$$

- where $\Gamma \vdash_{\mathsf{P}} \xi \parallel \zeta$ expresses that, according to Γ, ξ only yields type variables and ζ only yields term variables.
- Proof by induction on a and s respectively.

Cancellation Laws

$$\frac{\lceil a[\xi] \rceil^{\Gamma}_{\mathsf{ter}} = s}{\lfloor s \rfloor = a[\xi]} \qquad \frac{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} \lfloor s[\xi, \zeta] \rfloor : c \qquad \Gamma \vdash_{\mathsf{P}} \xi \parallel \zeta}{\lceil \lfloor s[\xi, \zeta] \rfloor \rceil^{\Gamma}_{\mathsf{ter}} = s[\xi, \zeta]}$$

- where $\Gamma \vdash_{\mathsf{P}} \xi \parallel \zeta$ expresses that, according to Γ, ξ only yields type variables and ζ only yields term variables.
- Proof by induction on a and s respectively.
- *Important:* neither require well-typing for the initial term.

Summary

- Full Reduction of the type formation and typing problems in both directions, formalised in Coq.
- Lessons learned:
  - The standard approach of minimally generalising statements tends to introduce a lot of unnecessary complexity.
  - *De Bruijn syntax*, paired with *parallel substitutions* and *context morphisms* enables clean formalisations of syntax with binders.
  - *Context morphisms* scale to translation scenarios.

Thank you for your attention.

`http://www.ps.uni-saarland.de/extras/hor16`

System F – full definition

$$A, B, C \;:=\; x_{\text{ty}} \mid A \to B \mid \forall.\, A \qquad s, t \;:=\; x_{\text{ter}} \mid s\, t \mid \lambda A.\, s \mid s\, A \mid \Lambda.\, s \qquad x \in \mathbb{N}$$

$$\frac{x < N}{N \vdash_{\mathbf{F}}^{\mathbf{ty}} x_{\text{ty}}} \qquad \frac{N \vdash_{\mathbf{F}}^{\mathbf{ty}} A \qquad N \vdash_{\mathbf{F}}^{\mathbf{ty}} B}{N \vdash_{\mathbf{F}}^{\mathbf{ty}} A \to B} \qquad \frac{(N+1) \vdash_{\mathbf{F}}^{\mathbf{ty}} A}{N \vdash_{\mathbf{F}}^{\mathbf{ty}} \forall.\, A}$$

$$\frac{A_x = A \qquad N \vdash_{\mathbf{F}}^{\mathbf{ty}} A}{N; A_n, \ldots, A_0 \vdash_{\mathbf{F}}^{\mathbf{ter}} x_{\text{ter}} : A}$$

$$\frac{N; \Gamma \vdash_{\mathbf{F}}^{\mathbf{ter}} s : A \to B \qquad N; \Gamma \vdash_{\mathbf{F}}^{\mathbf{ter}} t : A}{N; \Gamma \vdash_{\mathbf{F}}^{\mathbf{ter}} s\, t : B} \qquad \frac{N; \Gamma, A \vdash_{\mathbf{F}}^{\mathbf{ter}} s : B \qquad N \vdash_{\mathbf{F}}^{\mathbf{ty}} A}{N; \Gamma \vdash_{\mathbf{F}}^{\mathbf{ter}} \lambda A.\, s : A \to B}$$

$$\frac{N; \Gamma \vdash_{\mathbf{F}}^{\mathbf{ter}} s : \forall.\, A \qquad N \vdash_{\mathbf{F}}^{\mathbf{ty}} B}{N; \Gamma \vdash_{\mathbf{F}}^{\mathbf{ter}} s\, B : A[B \cdot \mathrm{id}]} \qquad \frac{(N+1); \Gamma[+1] \vdash_{\mathbf{F}}^{\mathbf{ter}} s : A}{N; \Gamma \vdash_{\mathbf{F}}^{\mathbf{ter}} \Lambda.\, s : \forall.\, A}$$

$\lambda 2$ – full definition

SAARLAND UNIVERSITY
COMPUTER SCIENCE

$$a, b, c, d \;:=\; u \mid x \mid a\,b \mid \lambda a.\, b \mid \Pi a.\, b \qquad u \in \{*, \Box\} \qquad x \in \mathbb{N}$$

$$\frac{}{0 : s[+1] \;\in\; \Gamma, s} \qquad \frac{x : s \;\in\; \Gamma}{(x+1) : s[+1] \;\in\; \Gamma, t}$$

$$\frac{}{\Gamma \vdash_2 * : \Box} \qquad \frac{x : a \in \Gamma \qquad \Gamma \vdash_2 a : u}{\Gamma \vdash_2 x : a} \qquad \frac{\Gamma \vdash_2 a : u \qquad \Gamma, a \vdash_2 b : *}{\Gamma \vdash_2 \Pi a.\, b : *}$$

$$\frac{\Gamma \vdash_2 a : \Pi c.\, d \qquad \Gamma \vdash_2 b : c}{\Gamma \vdash_2 a\,b : d[b \cdot \mathrm{id}]} \qquad \frac{\Gamma, a \vdash_2 b : c \qquad \Gamma \vdash_2 a : u \qquad \Gamma, a \vdash_2 c : *}{\Gamma \vdash_2 \lambda a.\, b : \Pi a.\, c}$$

Auxiliary type system P – full definition

SAARLAND UNIVERSITY
COMPUTER SCIENCE

$$\frac{x : * \in \Gamma}{\Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}} x} \qquad \frac{\Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}} a \qquad \Gamma, a \vdash^{\mathsf{ty}}_{\mathsf{P}} b}{\Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}} \Pi a.\, b} \qquad \frac{\Gamma, * \vdash^{\mathsf{ty}}_{\mathsf{P}} a}{\Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}} \Pi *.\, a}$$

$$\frac{x : a \in \Gamma \qquad \Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}} a}{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} x : a}$$

$$\frac{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} a : \Pi c.\, d \qquad \Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} b : c}{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} a\, b : d[b \cdot \mathrm{id}]} \qquad \frac{\Gamma, a \vdash^{\mathsf{ter}}_{\mathsf{P}} b : c \qquad \Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}} a}{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} \lambda a.\, b : \Pi a.\, c}$$

$$\frac{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} a : \Pi *.\, b \qquad \Gamma \vdash^{\mathsf{ty}}_{\mathsf{P}} c}{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} a\, c : b[c \cdot \mathrm{id}]} \qquad \frac{\Gamma, * \vdash^{\mathsf{ter}}_{\mathsf{P}} a : b}{\Gamma \vdash^{\mathsf{ter}}_{\mathsf{P}} \lambda *.\, a : \Pi *.\, b}$$