Type theories without contexts

Andrej Bauer[1] **Philipp G. Haselwarter[2]**

June 15, TYPES'21

[1]University of Ljubljana [2]Aarhus University

1. Hi everybody, my name is Philipp Haselwarter, thank you for coming to my talk about *type theories without contexts*. This is joint work with Andrej Bauer.

Type theories without contexts

Contribution:

Two presentation of finitary general type theories (BHL'20)

- with (CX) and without (CF) explicit contexts
- prove of translations back and forth
- basis for implementation (Andromeda 2)

1. Finitary type theories are a version of Bauer, Haselwarter, and Lumsdaine's General dependent type theories that is suitable for implementation in an LCF style proof assistant.

   We restrict the arities of rules to be finite and use a locally nameless syntax for variable binding. Finitary type theories encompass a wide variety of known theories, so long as they use intuitionistic contexts and can be phrased with the traditional four judgement forms for types, terms, and type- and term-equality.
2. pause
3. pause

Type theories without contexts

Contribution:

Two presentation of finitary general type theories (BHL'20)

- with (CX) and without (CF) explicit contexts
- prove of translations back and forth
- basis for implementation (Andromeda 2)

1. Examples include tame theories such as simple type theories, or intensional type theory, but also wild theories such as extensional Martin-Löf type theory. Theories with other judgement forms such as cubical type theory or for instance linear type theories are not currently within the scope of general type theories.

   We define two presentations of finitary type theories, one with and one without explicit contexts. We have proven that both presentations derive the same judgements by constructing translations in both directions. The context free presentation serves as the basis of the implementation of the Andromeda 2 prover.
2. pause
3. pause

Type theories without contexts

Contribution:

Two presentation of finitary general type theories (BHL'20)

- with (CX) and without (CF) explicit contexts
- prove of translations back and forth
- basis for implementation (Andromeda 2)

Motivation:

Implementing general type theories is tricky!

- joining contexts (when forward chaining)
- strengthening fails (e.g. equality reflection)
- effectful metalanguage can cause scope extrusion

$$\frac{\Gamma_1 \vdash s : A \qquad \Gamma_2 \vdash t : A \qquad \Gamma_3 \vdash p : \mathrm{Eq}(A, s, t)}{??? \vdash s \equiv t : A}$$

$\lambda(x : A).\ s := x\ ;\ x$

1. pause
2. Implementing general type theories is tricky, because we have to accommodate type theories that are not very well-behaved.

   When we combine separately constructed judgements in forward chaining reasoning, we have to join together different contexts, while respecting the dependency graph of each context.

   Strengthening can fail for certain type theories. For example, the conclusion of the equality reflection rule here on the right does not mention the term p, so we cannot read the variables used in a derivation off the conclusion of a judgement.

   Finally, if we work in an effectful meta-language, variables can escape their scope. For example, while type-checking this definition of the identity function we can smuggle the variable x out of its scope by storing it in the reference s.
3. pause

Type theories without contexts

Contribution:

Two presentation of finitary general type theories (BHL'20)

- with (CX) and without (CF) explicit contexts
- prove of translations back and forth
- basis for implementation (Andromeda 2)

Motivation:

Implementing general type theories is tricky!

- joining contexts (when forward chaining)
- strengthening fails (e.g. equality reflection)
- effectful metalanguage can cause scope extrusion

$$\frac{\Gamma_1 \vdash s : A \qquad \Gamma_2 \vdash t : A \qquad \Gamma_3 \vdash p : \mathrm{Eq}(A, s, t)}{??? \vdash s \equiv t : A}$$

$\lambda(x : A).\ s := x\ ;\ x$

Related work:

Γ^∞ for Pure Type Systems (Geuvers et al. 2010)

1. pause
2. pause
3. The method we use to tackle these difficulties can be seen as an extension of the Gamma-infinity system introduced by Geuvers and co-authors as a presentation of pure type systems without explicit contexts.

   Instead of storing the typing information in the context, variables are annotated with types. The connection with the original calculus is established via translation theorems in both directions.

   Now, the main source of difficulty extending context-free presentations to our setting lies in the kind of proof irrelevance exhibited for instance by the equality reflection rule and the ensuing failure of strengthening, which does not arise in Pure Type Systems.

Context-free type theory

$$a : \mathbb{N} \vdash a + 0 : \mathbb{N} \quad \rightsquigarrow \quad a^{\mathbb{N}} + 0 : \mathbb{N}$$

1. No contexts
2. Every variable is tagged with a type

1. So, how do we do it? Consider the judgement at the top of the slide. We start by deleting the contexts from our judgements. The judgement "in context a of type nat, a plus zero has type nat" becomes instead "variable a *annotated* with the type nat plus zero has type nat".
   This solves two of our problems: we don't have to worry about combining contexts anymore, and if a variable escapes its scope, we can still make sense of it because it carries sufficient information.
2. pause
3. pause
4. pause

Context-free type theory

$$a : \mathbb{N} \vdash a + 0 : \mathbb{N} \quad \rightsquigarrow \quad \mathrm{a}^{\mathbb{N}} + 0 : \mathbb{N}$$

$$\frac{A \text{ type} \qquad s : A \qquad t : A \qquad p : \mathrm{Eq}(A, s, t)}{s \equiv t : A}$$

$$A \rightsquigarrow \mathrm{A}^{\square \text{ type}} \qquad s \rightsquigarrow \mathrm{s}^{\square : A} \qquad t \rightsquigarrow \mathrm{t}^{\square : A} \qquad p \rightsquigarrow \mathrm{p}^{\square : \mathrm{Eq}(A,s,t)}$$

1. No contexts
2. Every variable is tagged with a type
3. Every metavariable is tagged with the boundary of its judgement ("meta-type")

1. pause
2. A general definition of type theories must include a definition of rules, which means we need a formal theory of metavariables. The metavariables in this rule are A, s, t, and p. Just like we do with variables, we annotate each metavariable with its "meta-type".
3. pause
4. pause

Context-free type theory

$$a : \mathbb{N} \vdash a + 0 : \mathbb{N} \quad \rightsquigarrow \quad \mathrm{a}^{\mathbb{N}} + 0 : \mathbb{N}$$

$$\frac{A \text{ type} \qquad s : A \qquad t : A \qquad p : \mathrm{Eq}(A, s, t)}{s \equiv t : A \text{ by } \{\!|p|\!\}}$$

$$A \rightsquigarrow \mathrm{A}^{\square \text{ type}} \qquad s \rightsquigarrow \mathrm{s}^{\square : A} \qquad t \rightsquigarrow \mathrm{t}^{\square : A} \qquad p \rightsquigarrow \mathrm{p}^{\square : \mathrm{Eq}(A,s,t)}$$

1. No contexts
2. Every variable is tagged with a type
3. Every metavariable is tagged with the boundary of its judgement ("meta-type")
4. Proof-irrelevant rules (e.g. equations) must record all variables in assumption sets

1. pause
2. pause
3. Finally we address the failure of certain rules to record all of the variables that were used to derive their premises by introducing *assumption sets*. Instead of storing the entire term p, the context-free equality reflection rule traverses p and collects the variables it contains. This information is then stored in the conclusion as curly braces p.
4. pause

Context-free type theory

$$a : \mathbb{N} \vdash a + 0 : \mathbb{N} \quad \rightsquigarrow \quad a^{\mathbb{N}} + 0 : \mathbb{N}$$

$$\frac{A \text{ type} \qquad s : A \qquad t : A \qquad p : \mathrm{Eq}(A, s, t)}{s \equiv t : A \text{ by } \{\!|p|\!\}}$$

$$\frac{s : A \qquad A \equiv B \text{ by } \alpha}{\kappa(s, \beta) : B} \quad \text{with } \beta = \alpha \cup \{\!|A|\!\}$$

$$A \rightsquigarrow A^{\square \text{ type}} \qquad s \rightsquigarrow s^{\square : A} \qquad t \rightsquigarrow t^{\square : A} \qquad p \rightsquigarrow p^{\square : \mathrm{Eq}(A,s,t)}$$

1. No contexts
2. Every variable is tagged with a type
3. Every metavariable is tagged with the boundary of its judgement ("meta-type")
4. Proof-irrelevant rules (e.g. equations) must record all variables in assumption sets
5. Conversion records assumption sets

1. pause
2. pause
3. pause
4. This change to the structure of equality judgements means that we also have to change the conversion rule. We record the assumption set alpha used in the derivation of the equation A equals B and the variables used to derive A as the conversion term kappa s beta.

Equivalence of contextual & context-free FTTs

Theorem

There is a judgement-level translation from CF to contextual FTT.

Proof idea: erase typing annotations and replace them with a suitable context.

$$\mathcal{J} \quad \mapsto \quad \Gamma_{\{\mathcal{J}\}} \vdash \lfloor \mathcal{J} \rfloor$$

$$\frac{\mathcal{D}_{CF}}{\mathcal{J}} \quad \Longrightarrow \quad \frac{\mathcal{D}_{CX}}{\Gamma_{\{\mathcal{J}\}} \vdash \lfloor \mathcal{J} \rfloor}$$

1. Our first translation theorem tells us that we can safely use context free type theories to derive FTT judgements. Crucially, the *construction* of the concluding judgement does not depend on derivability. This is important for an implementation, because we want to be able to translate a CF judgement to an FTT judgement without storing its entire derivation. The theorem then *only* relies on the existence of a derivation to show that the translated judgement is *derivable* as well.
2. pause

Equivalence of contextual & context-free FTTs

Theorem

There is a judgement-level translation from CF to contextual FTT.

Proof idea: erase typing annotations and replace them with a suitable context.

$$\mathcal{J} \quad \mapsto \quad \Gamma_{\{\mathcal{J}\}} \vdash \lfloor \mathcal{J} \rfloor$$

$$\frac{\mathcal{D}_{CF}}{\mathcal{J}} \implies \frac{\mathcal{D}_{CX}}{\Gamma_{\{\mathcal{J}\}} \vdash \lfloor \mathcal{J} \rfloor}$$

Theorem

There is a derivation-level translation from contextual FTT to CF.

Proof idea: induction on derivation, collecting assumptions.

$$\frac{\mathcal{D}_{CX}}{\Gamma \vdash \mathcal{J}} \implies \frac{\mathcal{D}_{CF}}{\mathcal{J}^+}$$

$$\lfloor \mathcal{J}^+ \rfloor \;=\; \mathcal{J}$$

1. pause
2. We also have a completeness theorem: Any judgement derivable in a finitary type theory with contexts can be translated to a derivable judgement in the corresponding context-free theory.
3. This proof proceeds by induction on the derivation, and here the construction of the judgement *does* depend on the derivation.
4. Finally, let me say that these translations are *robust*. They work not only for nice finitary type theories, but also for raw theories. If we do start with a type theory satisfying certain good properties such as tightness, these properties are *preserved* by the translations.
5. And that concludes my talk, thank you for listening.