Local Type Inference for Polarised System F with Existentials

ANONYMOUS AUTHOR(S)

A clear and well-documented LaTeX document is presented as an article formatted for publication by ACM in a conference proceedings or journal publication. Based on the “acmart” document class, this article presents and explains many of the common variations, as well as many of the formatting elements an author may use in the preparation of the documentation of their work.

CCS Concepts: • **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

Additional Key Words and Phrases: Type Inference, System F, Call-by-Push-Value, Polarized Typing, Focalisation, Subtyping

ACM Reference Format:
Anonymous Author(s). 2018. Local Type Inference for Polarised System F with Existentials. *J. ACM* 37, 4, Article 111 (August 2018), 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

$N = M$

ACM’s consolidated article template, introduced in 2017, provides a consistent LaTeX style for use across ACM publications, and incorporates accessibility and metadata-extraction functionality necessary for future Digital Library endeavors. Numerous ACM and SIG-specific LaTeX templates have been examined, and their unique features incorporated into this single new template.

If you are new to publishing with ACM, this document is a valuable guide to the process of preparing your work for publication. If you have published with ACM before, this document provides insight and instruction into more recent changes to the article template.

The “`acmart`” document class can be used to prepare articles for any ACM publication — conference or journal, and for any stage of publication, from review to final “camera-ready” copy, to the author’s own version, with *very* few changes to the source.

2 OVERVIEW

3 DECLARATIVE SYSTEM

3.1 The Language

The type syntax of $\mathrm{F}^{\pm}_{\exists}$ is given in fig. 1. The types of $\mathrm{F}^{\pm}_{\exists}$ are stratified into two syntactic categories (polarities): positive and negative, similarly to the Call-By-Push-Value system [Levy 2006]. The negative types represent computations, and the positive types represent values:

- α^- is a negative type variable, which can be taken from a context or introduced by $\exists$.
- a function $P \to N$ takes a value as input and returns a computation;
- a polymorphic abstraction $\forall \overrightarrow{\alpha^+}.N$ quantifies a computation over a list of positive type variables $\overrightarrow{\alpha^+}$. The polarities are chosen to follow the definition of functions.

0004-5411/2018/8-ART111 $15.00
https://doi.org/XXXXXXX.XXXXXXX

– a shift $\uparrow P$ allows a value to be used as a computation, which at the term-level corresponds to a pure computation **return** v.

+ α^+ is a positive type variable, taken from a context or introduced by $\forall$.
+ $\exists\overrightarrow{\alpha^-}.P$, symmetrically to $\forall$, binds negative variables in a positive type P.
+ a shift $\downarrow N$, symmetrically to the up-shift, thunks a computation, which at the term-level corresponds to $\{c\}$.

Negative declarative types

$$
\begin{array}{lcl}
N,\ M,\ K & ::= & \\
 & | & \alpha^- \\
 & | & \uparrow P \\
 & | & P \rightarrow N \\
 & | & \forall\overrightarrow{\alpha^+}.N
\end{array}
$$

Positive declarative types

$$
\begin{array}{lcl}
P,\ Q,\ R & ::= & \\
 & | & \alpha^+ \\
 & | & \downarrow N \\
 & | & \exists\overrightarrow{\alpha^-}.P
\end{array}
$$

Fig. 1. Declarative Types of $\mathsf{F}^{\pm}_{\exists}$

4 ALGORITHM

5 PROOF

6 EXTENSIONS

7 CONCLUSION

[Botlan et al. 2003] [Dunfield et al. 2020]

REFERENCES

Didier Le Botlan and Didier Rémy (Aug. 2003). “MLF Raising ML to the Power of System F.” In: *ICFP ’03*. Uppsala, Sweden: ACM Press, pp. 52–63.

Jana Dunfield and Neel Krishnaswami (Nov. 2020). “Bidirectional Typing.” In: arXiv: 1908.05839.

Paul Blain Levy (Dec. 2006). “Call-by-Push-Value: Decomposing Call-by-Value and Call-by-Name.” In: *Higher-Order and Symbolic Computation* 19.4, pp. 377–414. DOI: 10.1007/s10990-006-0480-6.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009