



Elaboration on $\text{HM}(X)$: Type Inference with Constraint Types

Marius Weidner

Chair of Programming Languages, University of Freiburg
`weidner@cs.uni-freiburg.de`

Abstract. We discuss $\text{HM}(X)$, a family of type systems that supports polymorphism, full type inference and constraint types. $\text{HM}(X)$ is an extension to the Hindley-Milner type system, that itself restricts System F such that full type inference is decidable and unambiguous. The constraint system X used in $\text{HM}(X)$ is left abstract and can be instantiated with arbitrary constraint systems that fulfill a set of conditions. Because of this abstraction $\text{HM}(X)$ can be used to model and reason about many commonly used constraint-like type features. Examples for constraint-like type features include subtyping, substructural types and type classes. $\text{HM}(X)$ comes with a complete and sound type inference algorithm, as well as a soundness proof, that both are independent of the actual constraint system X . Thus, the work for proving theoretical properties and constructing an inference algorithm for new constraint-like type features in a HM setting is reduced.

Table of Contents

1	Introduction.....	3
1.1	Hindley Milner: Polymorphism with Full Type Inference.....	3
1.2	Example: A Program with Constraint Types.....	3
2	HM(X).....	3
2.1	Syntax.....	3
2.2	Typing.....	3
3	Instantiating HM(X).....	3
3.1	HM(\mathcal{R}): Extension with Polymorphic Records.....	3
3.2	HM(\mathcal{O}): Extension with Overloading.....	3
4	Metatheory.....	3
4.1	Soundness.....	3
4.2	Type Inference.....	3
5	Related Work & Conclusion.....	3
5.1	Related Work.....	3
5.2	Conclusion.....	3

Fig. 1. Syntax

Fig. 2. Typing $(C, \Gamma \vdash e : \sigma)$

1 Introduction

1.1 Hindley Milner: Polymorphism with Full Type Inference

1.2 Example: A Program with Constraint Types

2 HM(X)

2.1 Syntax

2.2 Typing

3 Instantiating HM(X)

3.1 HM(\mathcal{R}): Extension with Polymorphic Records

Extensions

Example

3.2 HM(\mathcal{O}): Extension with Overloading

Extensions

Example

4 Metatheory

4.1 Soundness

4.2 Type Inference

5 Related Work & Conclusion

5.1 Related Work

5.2 Conclusion

References

- [1] Martin Odersky, Martin Sulzmann, and Martin Wehr. “Type Inference with Constrained Types”. In: *TAPoS* 5 (Jan. 1999), pp. 35–55. DOI: 10.1002/(SICI)1096-9942(199901/03)5:1<35::AID-TAP04>3.0.CO;2-4.

Fig. 3. Syntax

Fig. 4. Constraints

- [2] Benjamin C. Pierce. *Advanced Topics in Types and Programming Languages*. The MIT Press, 2004. ISBN: 0262162288.
- [3] Christian Skalka and François Pottier. “Syntactic Type Soundness for HM(X)”. In: *Electronic Notes in Theoretical Computer Science* 75 (2003). TIP’02, International Workshop in Types in Programming, pp. 61–74. ISSN: 1571-0661. DOI: [https://doi.org/10.1016/S1571-0661\(04\)80779-5](https://doi.org/10.1016/S1571-0661(04)80779-5). URL: <https://www.sciencedirect.com/science/article/pii/S1571066104807795>.

Fig. 5. Syntax

Fig. 6. Constraints