

# Decidable Subtyping for Path Dependent Types

JULIAN MACKAY, ALEX POTANIN, and LINDSAY GROVES\*, Victoria University of Wellington,  
New Zealand

JONATHAN ALDRICH†, Carnegie Mellon University, USA

Additional Key Words and Phrases: Decidability, Language Design, Object Oriented Languages, Functional Languages

Path dependent types have long served as an expressive component of the Scala programming language. They allow for the modelling of both bounded polymorphism and a degree of nominal subtyping. Nominality in turn provides the ability to capture first class modules. Thus a single language feature gives rise to a rich array of expressiveness. Recent work, on the Dependent Object Calculus (DOT), has proven path dependent types sound in the presence of both intersection and recursive types, providing a valuable foundation for language designers.

Unfortunately the combination of such a rich array of language features comes at a cost. Typing remains undecidable in DOT, posing problems for programmers who rely on the results of type checkers. The undecidability of typing in DOT arises from the subtyping of three of its' core features: path dependent types, recursive types, and dependent function types. A proof of subtype undecidability in the presence of path dependent types and dependent function types is easy to derive, by encoding the subtyping of System  $F_{<}$ : (a language in which subtyping is known to be undecidable). This is something that has been noted in past work on DOT. We demonstrate how even in the absence of dependent function types, the combination of path dependent types and recursive types allows for an encoding of not only System  $F_{<}$ , but an undecidable fragment of Java Generics.

The Wyvern programming language is an object oriented language with path dependent types, recursive types, dependent function types, and first class modules. We have developed three variants of Wyvern that feature all of these features, in restricted form, that also feature decidable subtyping. Despite the restrictions, our approaches retain the ability to encode the parametric polymorphism of Java generics along with many idioms of the Scala module system.

For further details and references see Mackay et al. [Mackay et al. 2020] and Mackay [Mackay 2020].

## REFERENCES

- Julian Mackay. 2020. *Decidable Subtyping for Path Dependent Types*. Ph.D. Dissertation. School of Engineering and Computer Science, Victoria University of Wellington.
- Julian Mackay, Potanin Alex, Jonathan Aldrich, and Lindsay Groves. 2020. Decidable Subtyping for Path Dependent Types. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2020)*. ACM, New York, NY, USA.

---

\*  
†

---

Authors' addresses: Julian Mackay; Alex Potanin; Lindsay Groves, School of Engineering and Computer Science, Victoria University of Wellington, New Zealand, julian@ecs.vuw.ac.nz, alex@ecs.vuw.ac.nz, lindsay@ecs.vuw.ac.nz; Jonathan Aldrich, School of Computer Science, Carnegie Mellon University, USA, jonathan.aldrich@cs.cmu.edu.

---

2019. 2475-1421/2019/11-ART1 \$15.00  
<https://doi.org/>