Scala with Explicit Nulls (Artifact)

Abel Nieto
University of Waterloo, Canada
anietoro@uwaterloo.ca

Yaoyu Zhao
University of Waterloo, Canada
y437zhao@edu.uwaterloo.ca

Ondřej Lhoták
University of Waterloo, Canada
olhotak@uwaterloo.ca

Angela Chang
University of Waterloo, Canada
yue.chang@edu.uwaterloo.ca

Justin Pu
University of Waterloo, Canada
justin.pu@edu.uwaterloo.ca

Abstract

This artifact is a companion to the paper "Scala with Explicit Nulls", where we present a modification to the Scala type system that makes nullability *explicit* in the types. Specifically, we make reference types *non-nullable* by default, while still allowing for nullable types via *union types*.

The artifact contains an implementation of this new type system design as a fork of the Dotty (Scala 3) compiler. Additionally, the artifact contains the source code of multiple Scala libraries that we used to evaluate our design.

2012 ACM Subject Classification Software and its engineering → General programming languages; Theory of computation → Denotational semantics; Theory of computation → Type theory; Software and its engineering → Interoperability

Keywords and phrases Scala, Java, nullability, language interoperability, type systems

Digital Object Identifier 10.4230/DARTS.6.2.14

Funding This research was supported by the Natural Sciences and Engineering Research Council of Canada and by the Waterloo-Huawei Joint Innovation Lab.

Acknowledgements We would like to thank Sébastien Doeraene, Fengyun Liu, Guillaume Martres, and Martin Odersky for their feedback on our explicit nulls design and their help with Dotty.

Related Article Abel Nieto, Yaoyu Zhao, Ondřej Lhoták, Angela Chang, and Justin Pu, "Scala with Explicit Nulls", in 34th European Conference on Object-Oriented Programming (ECOOP 2020), LIPIcs, Vol. 166, pp. 25:1–25:26, 2020. `https://doi.org/10.4230/LIPIcs.ECOOP.2020.25`

Related Conference 34th European Conference on Object-Oriented Programming (ECOOP 2020), November 15–17, 2020, Berlin, Germany (Virtual Conference)

1 Scope

The artifact substantiates the paper's contributions (reproduced verbatim in *italics* below), not including the formalization of type nullification:

- *We retrofitted Scala's type system with a mechanism for tracking nullability, using union types. To improve usability of nullable values in Scala code, we also added a simple form of flow typing to Scala.* **The artifact contains a fork for the Dotty (Scala 3) compiler that implements the type system for explicit nulls described in the paper, including flow typing.**

- *So that Scala programs can interoperate with Java code, where* `nulls` *remain implicit, we present a* type nullification *function that turns Java types into equivalent Scala types.* **The compiler contained in the artifact implements type nullification as described in the paper. This can can be verified through a test suite also present in the artifact**.
- *We evaluate the design by migrating multiple Scala libraries to explicit nulls. The main findings are that most of the effort in migrating Scala code to explicit nulls comes from Java interoperability, and that the effort is significant for some libraries.* **The artifact contains the source code of multiple Scala libraries used to evaluate the effort required to migrate Scala code to the explicit nulls type system. The artifact contains instructions for how to (manually) generate the tables appearing in the paper. Due to mismatching library versions, the results (e.g. number of errors per thousand lines of code) obtained from the artifact do not exactly match those in the paper, but the high-level findings of the paper continue to hold.**

2 Content

The artifact is packaged as a Virtual Box VM (an `.ova` file) containing:
- A detailed guide describing the artifact.
- A modified version of the Dotty compiler containing the new explicit nulls type system.
- A test suite for the new type system.
- Copies of the Dotty Community Build libraries used to evaluate our implementation.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS).

4 Tested platforms

We have tested the VM with VirtualBox Version 6.1.6 r137129 (Qt5.6.3). The OS installed in the VM is Ubuntu 18.04.

5 License

The Dotty compiler is available under an Apache License 2.0 license. The community build libraries are available under their respective licenses.

6 MD5 sum of the artifact

5aa27baba97d324c2e28d5361e5a672e

7 Size of the artifact

5.1 GiB