A Formalised Theory of Turing Machines in Isabelle/HOL

Xu Jian, Xingyuan Zhang
PLA University of Science and Technology Nanjing, China

Christian Urban
King's College London, UK

***Abstract*—Isabelle/HOL is an interactive theorem prover based on classical logic. While classical reasoning allow users to take convenient shortcuts in some proofs, it precludes *direct* reasoning about decidability: every boolean predicate is either true or false because of the law of excluded middle. The only way to reason about decidability in a classical theorem prover, like Isabelle/HOL, is to formalise a concrete model for computation. In this paper we formalise Turing machines and relate them to register machines.**

***Keywords*-Turing Machines, Decidability, Isabelle/HOL;**

I. Introduction

We formalised in earlier work the correctness proofs for two algorithms in Isabelle/HOL, one about type-checking in LF and another about deciding requests in access control [1] [???]: these formalisations uncovered a gap in the informal correctness proof of the former and made us realise that important details were left out in the informal model for the latter. However, in both cases we were unable to formalise computablility arguments for the algorithms. The reason is that both algorithms are formulated in terms of inductive predicates. Suppose P stands for one such predicate. Decidability of P usually amounts to showing whether $P \vee \neg P$ holds. But this does not work in Isabelle/HOL, since it is a theorem prover based on classical logic where the law of excluded midle ensures that $P \vee \neg P$ is always provable.

These algorithms were given as inductively defined predicates. inductively defined predicates, but

Norrish choose the λ-calculus as a starting point for his formalisation, because of its "simplicity" [Norrish]

"Turing machines are an even more daunting prospect" [Norrish]

Our formalisation follows XXX

Contributions:

II. Related Work

The most closely related work is by Norrish. He bases his approach on lambda-terms. For this he introduced a clever rewriting technology based on combinators and de-Bruijn indices for rewriting modulo β-equivalence (to keep it manageable)

References

[1] C. Urban, J. Cheney, and S. Berghofer. Mechanizing the Metatheory of LF. *ACM Transactions on Computational Logic*, 12:15:1–15:42, 2011.