A Foundational Formalization of Grounded Deduction In Isabelle/Pure

Sascha Kehrli	Prof. Dr. Roger Wattenhofer	Prof. Dr. Bryan Ford
ETH Zürich	ETH Zürich	EPFL
Author	Supervisor	Co-Supervisor

Introduction

Modern theorem provers such as Coq and Lean rely on strong foundational logics to support inductive definitions and recursive functions. These powerful definitional mechanisms can only remain consistency by being tightly constrained, typically requiring termination, well-foundedness, or other syntactic criteria on recursion.

Grounded Deduction (GD) is a novel logical framework developed to invert this model [1]. Rather than constraining what users may define, GD permits arbitrary definitional recursion, even non-terminating or circular definitions such as $L \equiv \neg L$.

To maintain consistency in the presence of such unconstrained recursion, expansions of definitions in proofs are syntactically restricted and generate proof obligations. This principle is the foundational promise of GD: definitional freedom without compromising consistency.

Goal

GD is not well studied yet and in particular remains to be proven consistent relative to any logic. Ongoing work formalizes GD in Isabelle/HOL to study such meta-logical properties.

However, it is desirable to formalize GD within a minimal base logic, as this essentially removes a layer of abstraction in the form of the usual object logic used as meta-theory. With such a formalization, reasoning within the embedded GD logic is simpler and closer to the intended foundational principles, without being clouded by the language of a rich meta-logic. Such a development would also make GD more usable as a tool for automated reasoning itself.

This thesis seeks to realize such an implementation in Isabelle's minimal base logic, *Pure*, which is designed for precisely such a task [2].

The thesis aims to:

- Formalize the syntax of GD.
- Implement the deductive rules of GD: Axiomatize the inference system.
- Develop Grounded Arithmetic (GA): Build a basic arithmetic theory within GD.
- Construct encoded data structures: Implement finite datatypes such as lists and trees using purely arithmetical encodings over GA.
- Prototype definitional tooling: Create user-facing infrastructure for defining functions, data structures, and interpretations within GD, while reducing all such features to GA representations.

Bibliography

[1] B. Ford, "Reasoning Around Paradox with Grounded Deduction." [Online]. Available: https://arxiv.org/abs/2409.08243

[2] M. Wenzel, "The Isabelle/Pure Reference Manual." 2024. [Online]. Available: https://isabelle.in.tum.de/