



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

*Distributed
Computing*



A Foundational Formalization of GD in Isabelle/Pure

BSc Thesis

Sascha Kehrli
skehrli@ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Prof. Dr. Bryan Ford
Prof. Dr. Roger Wattenhofer

04.07.2025

Abstract

Contents

1	Introduction	1
2	Background	2
2.1	Grounded Deduction (GD)	2
2.2	Isabelle/Pure	2
2.2.1	Syntax of Pure	2
2.2.2	Deduction Rules	3
A	References	4

Introduction

1

Background

2

2.1 Grounded Deduction (GD)

2.2 Isabelle/Pure

Pure is the minimal meta-logic in Isabelle. The popular Isabelle/HOL fragment is formalized atop Pure, as are all other object logics in Isabelle. Aside from a bare-bones logic, Pure provides infrastructure to define datatypes and axioms to facilitate the formalization of object logics in it.

2.2.1 Syntax of Pure

The core syntax of Pure is a typed lambda calculus with type variables, universal quantification, equality, and implication. Propositions are terms of the distinct type `prop`. Propositions in Pure are thus terms and not types, like they are in type-theory based provers like Rocq or Lean.

Type Syntax

$$\begin{aligned} \tau &::= \alpha \text{ (type variable)} \\ &| \tau \Rightarrow \tau \text{ (function type)} \\ &| \text{prop (type of propositions)} \end{aligned}$$

Term Syntax

$$\begin{aligned} t &::= x \text{ (variable)} \\ &| c \text{ (constant)} \\ &| t \ t \text{ (application)} \\ &| \lambda x : \tau. t \text{ (lambda abstraction)} \\ &| t \Longrightarrow t \text{ (implication)} \\ &| t \equiv t \text{ (equality)} \\ &| \Lambda x : \tau. t \text{ (universal quantification)} \end{aligned}$$

The symbols used for implication, equality, and universal quantification are non-standard to leave the standard symbols free for object logics atop Pure.

Even though Pure has type variables, it provides no construct to capture them as an argument, and thus also has no for-all type like the polymorphic lambda calculus System F.

2.2.2 Deduction Rules

The implication operator in Pure expresses logical entailment. The deduction rules for the implication and universal quantifier are unsurprising:

Implication Deduction Rules

$$\frac{[\varphi] \quad \psi}{\varphi \Rightarrow \psi} \quad (\Rightarrow I)$$

$$\frac{\varphi \Rightarrow \psi \quad \varphi}{\psi} \quad (\Rightarrow E)$$

Universal Quantification Deduction Rules

$$\frac{\varphi[x]}{\wedge x. \varphi[x]} * \quad (\forall I)$$

$$\frac{\wedge x. \varphi[x]}{\varphi[t]} \quad (\forall E)$$

* The variable x must not be free in the assumptions.

An object logic in Pure is created by adding new axioms, constants, and types.

A References