

Chapter 1

Proof Support for General Type Classes

Ron van Kesteren¹, Marko van Eekelen¹, Maarten de Mol¹

Abstract: We present a proof rule and an effective tactic for proving properties about HASKELL type classes by proving them for the available instance definitions. This is not straightforward, because instance definitions may depend on each other. The proof assistant ISABELLE handles this problem for single parameter type classes by structural induction on types. However, this does not suffice for an effective tactic for more complex forms of overloading. We solve this using an induction scheme derived from the instance definitions. The tactic based on this rule is implemented in the proof assistant SPARKLE.

1.1 INTRODUCTION

It is often stated that formulating properties about programs increases robustness and safety, especially when formal reasoning is used to prove these properties. Robustness and safety are becoming increasingly important considering the current dependence of society on technology. Research on formal reasoning has spawned many general purpose proof assistants, such as COQ [dt04], ISABELLE [NPW02], and Pvs [OSRSC99]. Unfortunately, these general purpose tools are geared towards mathematicians and are hard to use when applied to more practical domains such as actual programming languages.