Journal of Symbolic Computation 90 (2019) 89–123

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

Aligning concepts across proof assistant libraries

Thibault Gauthier, Cezary Kaliszyk

Department of Computer Science, University of Innsbruck, 6020 Innsbruck, Austria

A R T I C L E   I N F O

Article history:
Received 6 March 2017
Accepted 25 June 2017

Keywords:
Proof assistant libraries
Library alignment
Higher-order logic
Type theory
Set theory
Dynamical systems

A B S T R A C T

As the knowledge available in the computer understandable proof corpora grows, recognizing repeating patterns becomes a necessary requirement in order to organize, synthesize, share, and transmit ideas. In this work, we automatically discover patterns in the libraries of interactive theorem provers and thus provide the basis for such applications for proof assistants. This involves detecting close properties, inducing the presence of matching concepts, as well as dynamically evaluating the quality of matches from the similarity of the environment of each concept. We further propose a classification process, which involves a disambiguation mechanism to decide which concepts actually represent the same mathematical ideas.

We evaluate the approach on the libraries of six proof assistants based on different logical foundations: HOL4, HOL Light, and Isabelle/HOL for higher-order logic, Coq and Matita for intuitionistic type theory, and the Mizar Mathematical Library for set theory. Comparing the structures available in these libraries our algorithm automatically discovers hundreds of isomorphic concepts and thousands of highly similar ones.

E-mail addresses: thibault.gauthier@uibk.ac.at (T. Gauthier), cezary.kali1szyk@uibk.ac.at (C. Kaliszyk).
URLs: http://cl-informatik.uibk.ac.at/users/tgauthier/ (T. Gauthier), http://cl-informatik.uibk.ac.at/cek/ (C. Kaliszyk).

https://doi.org/10.1016/j.jsc.2018.04.005

1. Introduction

1.1. Context

With the diversity of interactive theorems provers (Harrison et al., 2014), the lack of interoperability is a growing issue. Formalized proofs originating from one prover are hardly reusable in a different one. Discovering and identifying the structures that occur in multiple libraries becomes an important step to better interoperability as the libraries of theorem provers grow.

The benefits of links between different structures have since long been known by mathematicians (Corry, 2012). Algebraic structures such as fields (Rotman, 2010) enable mathematicians to transport properties from real to complex numbers. Moreover the whole field of category theory has been about generalization (Awodey, 2006) with recent techniques such as classifying a topos of a theory as very powerful transfer mechanism (Univalent Foundations Program, 2013). In computer programming, oriented-object languages (Meyer, 1988) can share a method across many object instances using inheritance. Both examples shows how an interconnected structure is beneficial for better insights and faster development.

To this end, we develop an algorithm that automatically evaluates the similarity between formalized concepts (units of thought). This is achieved by inferring the mathematical properties they possess, which is a reflection of the structure they describe or belong to.

1.2. Challenges

Aligning libraries comes with a set of challenges. The mere fact that common mathematical structures have been (re-)formalized in each proof assistant makes this initiative conceivable.

The first difficulty is to express the mathematical properties uniformly. The multiplicity of the logics of the studied provers make this step quite complicated. Indeed, they have often different degree of support for lambda-abstractions, polymorphism, type classes, type hierarchies, algebraic hierarchies, etc. Those features produce some idiosyncratic constructions in the formal developments in each prover.

The next step is to define and recognize which mathematical concepts appear in the library. There may be for instance types, constants, subterms, formula subtrees or even proof tactics. Our goal will be to define what are the unit concepts and which ones are a combination of those concepts. Another issue is that some concepts are defined many times inside one library. Indeed different integer representations can be more suitable for some applications (like code extraction Haftmann et al., 2013). Conversely, a concept can belong to many different structures. It is especially common in the traditional set theoretic approach, where the empty set $\emptyset$ also stands for the natural number 0. This is realized by most formalizations of set theory, for example in the foundations of Mizar (Grabowski et al., 2015) and Isabelle/ZF (Paulson, 2016).

Having delimited our notion of "concepts", we wish to derive their similarities. A uniform representation for the properties makes it easy to infer which concepts share the same properties. We would like to emphasize here that the approach is more effective and more comprehensive than looking only at their definitions. Already for minimally different definitions, recognizing that they represent the same concept is not straightforward. This becomes very hard when definitions are foundationally different, for instance the real numbers may be defined through Dedekind cuts or Cauchy sequences. Moreover, the similarity measure may indicate for example the discovery of the underlying ring structures of integers and real numbers, which would not be possible if we restrict to the discovery of perfect matches only. Furthermore, the context in which the concept is expressed can be essential. To capture its influence, we also study the interconnections between properties inside a library that allow finding similar relations between concepts in different libraries.

We hope that solving these issues will create libraries of alignments suitable for the different types of applications envisioned.