Automating alignments of HOL Light inductive types and recursive functions in Rocq

Antoine Gontard, LMF, Inria

1 Introduction

hol2dk [1] is a tool that allows to translate HOL Light [2] proofs to obtain dedukti/lambdapi [3] [4] proofs which can in turn be translated to Rocq. But for the translated theorems to be usable, one has to "align" translated HOL Light types and definitions with their Rocq defined counterparts by proving they are equal.

In this work I present tactics that I have written in Ltac to automate these alignments in the case of inductive types and recursive functions. I will focus on the example of the length function on lists, but for that, one needs to first align lists.

With that goal in mind, I will describe how inductive types are defined in HOL Light, which implies talking about subtypes and inductive propositions. Indeed, I should start by introducing two important concepts in HOL Light that are very relevant to my work : Hilbert's ε choice operator [5] and native subtypes.

2 Alignments

I will first present the way inductive types are defined in HOL Light and compare their induction principles with those in Rocq, then show how their alignments are simplified with my tactic `ind_align` automating the proofs by double induction.

I will then describe the Rocq aligned version of HOL Light's `recspace` type that uses infinitely branching trees to represent elements of any inductive type, and showcase how it works with the definition of lists in HOL Light.

The next step will then be to talk about how to align this type with Rocq's lists, which is (not yet fully) automated by my `_mk_dest_rec` and `_dest_mk_rec` tactics.

As was the goal, I will be able to talk about the alignment of the length function, and the general alignment of recursive functions with the `total_align` tactic using previous work by Theo Winterhalter.

Through all of it, I will showcase examples of how I simplified previous work in the coq-hol-light [6] [1] Library as well as some examples of use from HOL Light's Logic library that I have been translating

3 More and Conclusion

I will also briefly talk about more complex alignments that I have worked on such as partial functions, inductive types with list recursive arguments and their recursive functions.

Finally, I will discuss the up-to-date progress of my work and what more can be done, including the potential for even more automation by generating the correct inductive type or recursive function.

References

[1] Frédéric Blanqui. "Translating HOL-Light proofs to Coq". In: *Proceedings of 25th Conference on Logic for Programming, Artificial Intelligence and Reasoning*. Ed. by Nikolaj Bjørner, Marijn Heule, and Andrei Voronkov. Vol. 100. EPiC Series in Computing. EasyChair, 2024, pp. 1–18. DOI: 10.29007/6k4x. URL: /publications/paper/mtFT.

[2] John Harrison. *The HOL Light theorem prover*. URL: https://hol-light.github.io/.

[3] Ali Assaf et al. "Dedukti: a Logical Framework based on the λΠ-Calculus Modulo Theory". working paper or preprint. 2016. URL: https://inria.hal.science/hal-04281492.

[4] *lambdapi*. URL: https://github.com/Deducteam/lambdapi.

[5] John Harrison and Juneyoung Lee. *HOL Light tutorial*. URL: https://hol-light.github.io/tutorial.pdf.

[6] Frédéric Blanqui et al. *coq-hol-light*. URL: https://github.com/Deducteam/coq-hol-light.