

# Cocktail II

Michael Franssen  
m.franssen@tue.nl

Eindhoven University of Technology  
Eindhoven, The Netherlands

## 1 Introduction

Cocktail II (a further development of Cocktail) is an interactive tool for deriving programs from specifications. Instead of verifying a program after it was constructed, Cocktail II aids the goal oriented derivation of a program. First, the user provides a pre- and postcondition. Then, the gap between these conditions is filled by manually inserting statements, constructing the program step by step. The tool then computes the sub specifications of the new gaps, which can be manually refined further. Finally, when a precondition of a gap implies its postcondition, the gap can be closed. Cocktail II also provides support for constructing the required proof. The program is complete when all gaps are closed.

## 2 Features

Some of the more prominent features of Cocktail II are:

- Pre- and postconditions are written in multi-sorted first-order predicate logic. Internally, this logic is represented by a typed lambda calculus. Programs too are stored as a kind of lambda-terms. As a result, all proofs and completed annotated programs can be checked by a small, simple program. Therefore, the result is reliable. Even if the (large) tool constructs a faulty proof, this will be detected in the final check.
- Cocktail II provides a built-in interactive theorem prover that uses a graphical Fitch-style natural deduction interface. With this interface, the user can prove theorems mostly by pointing and clicking. If a sub-theorem has become simple enough and no longer requires induction proofs, it can be passed to an automated theorem prover. Cocktail II has a built-in automated theorem prover for first-order logic with equalities, but also supports using external theorem provers.
- Algebraic datatypes like trees and lists can be simply defined in Cocktail II. The interactive theorem prover provides an induction tactic for these datatypes. Also, it can pass (sub)proofs to automated theorem provers at any time. As a result, many interesting programming problems that are hard to tackle using other tools can be solved by Cocktail II.
- Completed proofs are stored in a database, so they never have to be constructed twice.
- Formulas are displayed in a user friendly way using unicode characters. E.g.  $(\forall x : U. \exists y : U. P(y) \Rightarrow \neg P(x) \wedge Q(x))$  is displayed as such.

### **3 Future work**

Currently, we are extending Cocktail II with features to also support post-hoc verification of completed programs. Also, we are extending the programming language to support more complex datatypes, (mutually) recursive procedures and object-like features.