Constructive Formalization of Regular Languages

Jan-Oliver Kaiser

September 5, 2012

Contents

Chapter 2

Coqand SSReflect

We decided to employ the Small Scale Reflection Extension (**SSReflect**) for the **Coq** proof assistant. The most important factors in this decision were SSReflect's excellent support for finite types, list operations and graphs. SSReflect also introduces an alternative scripting language that can often be used to shorten the bookkeeping overhead of proofs considerably.

2.1 Coq

2.2 SSReflect

2.2.1 Finite Types and Ordinals

The most important feature of SSReflect for our purpose are finite types. SSReflect provides boolean versions of the universal and existential quantifiers on finite types, **forallb** and **existsb**. We can compute the number of elements in a finite type F with #|F|. enum gives a list of all items of a finite type. Finite types also come with enumeration functions which provide a consistent ordering. The corresponding functions are enum_rank and enum_val. The input of enum_val and the result of enum_rank are ordinals, i.e. values in [0, \#|F|−1]. The corresponding type can be written as I_#|F|

2.2.2 Boolean Reflection

SSReflect offers boolean reflections for decidable propostions. This allows us to switch back and forth between equivalent boolean and propositional predicates.

2.2.3 Boolean Predicates

We make use of SSReflect's syntax to specify boolean predicates. This allows us to specify predicates in a way that resembles set-theoretic notation, e.g. [pred x | <boolean expression in x>]. The complement of a predicate can be written as [predC p]. The syntax for combining precidates is [pred? p1 & p2], with ? being replaced with one of U (union), I (intersection) or D (difference). The resulting predicates are called collective predicates. SSReflect allows us to use infix notation, i.e. $x \in p$, for these predicates.

There are also applicative (functional) versions of of predC, predU, predI, predD which are functions that take predicates and return applicative predicates.

Chapter 2

Conclusion

Chapter 3

References