Constructive Formalization of Regular Languages

Jan-Oliver Kaiser

Advisors: Christian Doczkal, Gert Smolka

Supervisor: Gert Smolka

Contents

1. Quick Recap
2. Motivation
3. Previous work
4. Our development
5. Roadmap

Quick Recap

The regular languages over an alphabet Σ can be defined recursively:

- $\emptyset \in RL_\Sigma$
- $a \in \Sigma \rightarrow \{a\} \in RL_\Sigma$
- $A,B \in RL_\Sigma \rightarrow A \cup B \in RL_\Sigma$
- $A,B \in RL_\Sigma \rightarrow A \bullet B \in RL_\Sigma$
- $A \in RL_\Sigma \rightarrow A^* \in RL_\Sigma$

They can also be defined using regular expressions:

- $\emptyset \in RE_{\Sigma}, \mathcal{L}(\emptyset) := \{\}$
- $\varepsilon \in RE_{\Sigma}, \mathcal{L}(\varepsilon) := \{\varepsilon\}$
- $a \in \Sigma \rightarrow a \in RE_{\Sigma}, \mathcal{L}(a) := \{a\}$
- $r,s \in RL_{\Sigma} \rightarrow (r+s) \in RE_{\Sigma}, \mathcal{L}(r+s) := \mathcal{L}(r) \cup \mathcal{L}(s)$
- $r,s \in RL_{\Sigma} \rightarrow (r \bullet s) \in RE_{\Sigma}, \mathcal{L}(r \bullet s) := \mathcal{L}(r) \bullet \mathcal{L}(s)$
- $r \in RL_{\Sigma} \rightarrow r^* \in RE_{\Sigma}, \mathcal{L}(r^*) := \mathcal{L}(r)^*$

Additionally, regular languages are also exactly those languages accepted by finite automata.

One possible definition of FA over an alphabet Σ is:

- a finite set of states Q
- an initial state $s_0 \in$ Q
- a transition relation Δ $\in$ (Q, Σ, Q)
- a set of finite states F, F $\sqsubseteq$ Q

Let A be a FA.

$$\mathcal{L}(A) := \left\{ w \mid \exists s_1, \dots s_{|w|} \in Q \ s.t. \ \forall \ i : 0 < i \leq n \to (s_{i-1}, w_i, s_i) \in \Delta \right\}$$

Finally, regular languages are also characterized by the Myhill-Nerode theorem.

- First, we define a binary relation on L:
  $R_L xy := \neg \exists z,\ x \bullet z \in L \oplus y \bullet z \in L$
- L is regular if and only if $R_L$divides L into a finite number of equivalence classes.

Motivation

- Strong interest in formalizations in this area.
- No complete and elegant formalization of regular languages in Coq.
- Recent formalizations avoid FA in favor of partial derivatives.

Previous work

- Constructively formalizing automata theory (2000)

  Robert L. Constable, Paul B. Jackson, Pavel Naumov, Juan C. Uribe

  PA: Nuprl

  The first constructive formalization of MH.

  Based on **FA**.

- Proof Pearl: Regular Expression Equivalence and Relation Algebra (2011)

  Alexander Krauss, Tobias Nipkow

  PA: Isabelle

  Based on **partial derivatives of RE**. No proof of termination.

- Deciding Kleene Algebras in Coq (2011)

  Thomas Braibant, Damien Pous

  PA: Coq

  Based on **FA**, matrices. Focus on performance.

- A Decision Procedure for Regular Expression Equivalence in Type Theory (2011)

  Thierry Coquand, Vincent Siles

  PA: Coq

  Based on **partial derivatives of RE**.

- A Formalisation of the Myhill-Nerode Theorem based on Regular Expressions (Proof Pearl) (2011)

  Chunhan Wu, Xingyuan Zhang, Christian Urban

  PA: Isabelle

  The first proof of MH based on **partial derivatives of RE**.

- Deciding Regular Expressions (In-)Equivalence in Coq (2011)

  Nelma Moreira, David Pereira, Simão Melo de Sousa

  PA: Coq

  Based on Krauss, Nipkow. Proof of termination.

Our Development

- We want to focus on elegance, not performance.
- Our main goals are MH and the decidability of RE equivalence.
- We use FA.
  They are not at all impractical. (Partly thanks to Ssreflect's finType)

Ssreflect

- Excellent support for all things boolean.
- Finite types with all necessary operations and closure properties.
  (very useful for alphabets, FA states, etc.)
- Lots and lots of useful lemmas and functions.

Finite automata

DFA and NFA without e-transitions.

- DFA to prove closure under $\cup$, $\cap$, and $\neg$.
- NFA to prove closure under $\bullet$ and $*$.

Also proven: NFA $\Leftrightarrow$ DFA.

This gives us: RE $\Rightarrow$ FA.

Roadmap

1. Emptiness test on FA ($\emptyset(A) := \mathcal{L}(A) = \emptyset$)
2. FA $\Rightarrow$ RE
3. Dedicedability of RE equivalence using RE $\Rightarrow$FA, (2) and (1):

$$\mathcal{L}(r) = \mathcal{L}(s)$$

$$\Leftrightarrow$$

$$\emptyset(\mathcal{A}(r) \cap \overline{\mathcal{A}(s)}) \wedge \emptyset(\overline{\mathcal{A}(r)} \cap \mathcal{A}(s))$$

4. Finally, we want to prove the MH theorem

With this we'll have a complete formalization of regular languages including RE, FA and MH and all corresponding equivalences.