Reducing Acceptance Condition of Emerson-Lei Automata

Tereza Schwarzová and Jan Strejček

Masaryk University, Brno, Czech Republic
{xschwar3,strejcek}@mail.muni.cz

Abstract. We present an QBF-based algorithm that reduces the number of acceptance sets of transition-based Emerson-Lei automata.

1 Introduction

- Emerson-Lei automata jsou v principu staré, ale v poslední době znovuobjevene [?] a je o tom ted spousta clanku [?] a nastroju
- algoritmy jsou obvykle citlive na akceptacni podminku (citace)
- cilem prace je zjednodusit akceptacni podminku (zejmnena redukovat pocet akceptacnich znacek) bez zmeny struktury automatu

2 Preliminaries

2.1 TELA

A *transition-based Emerson-Lei automaton* over alphabet Σ is a tuple $\mathcal{A} = (\mathcal{Q}, M, \Sigma, q_0, \delta, \varphi)$, where

- $\mathcal{Q}$ is a finite set of *states*,
- M is a finite set of *acceptance marks*,
- Σ is a finite *alphabet*,
- q_0 is an *initial state*,
- $\delta \subseteq \mathcal{Q} \times \Sigma \times 2^M \times \mathcal{Q}$ is a transition relation,
- φ is the acceptance condition constructed by the following abstract syntax equation, where $m \in M$:

$$\varphi ::= \mathit{true} \mid \mathit{false} \mid \mathsf{Inf}(m) \mid \mathsf{Fin}(m) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi).$$

A run π of $\mathcal{A}$ on word $u = u_0u_1u_2 \cdots \in \Sigma^\omega$ is an infinite sequence of transitions $\pi = (q_0, u_0, M_0, q_1)(q_2, u_1, M_1, q_2) \cdots \in \delta^\omega$.

Let $M(\pi)$ denote set of all acceptance marks, that occur in run π in infinitely many transitions. The run satisfies $\mathsf{Inf}(m)$ *iff* $m \in M(\pi)$ and it satisfies $\mathsf{Fin}(m)$ *iff* $m \notin M(\pi)$. Run π is *accepting* if it satisfies the formula φ. A set of words, which are accepted by $\mathcal{A}$, represent a language $\mathcal{L}(\mathcal{A})$.

Sometimes the set of transitions with the same acceptance mark $m \in M$ is referred to as *acceptance set* $Z(m)$.

- define DNF
- define SCC

2.2 Quantified Boolean formula

Quantified Boolean formulae (QBFs) is an extension of propositional logic – some Boolean variables in the formula can be quantified with *universal* or *existential* quantifiers. Formula Φ is a QBF formula in *prenex normal form* $\iff \Phi = Q_1 x_1 ... Q_n x_n . \varphi$, where $Q_i \in \{\forall, \exists\}$, x_i is a quantified Boolean variable and φ is quantifier-free Boolean formula.

Let Φ be a QBF. Semantically we define the *universal* quantifier as $\forall x . \Phi = \Phi[x \to \mathit{true}] \land \Phi[x \to \mathit{false}]$ and the *existential* quantifier as $\exists x . \Phi = \Phi[x \to \mathit{true}] \lor \Phi[x \to \mathit{false}]$.

Let X be a finite set of variables. Assignment $\mathcal{A}_X : X \to \{\mathit{false}, \mathit{true}\}$ is a total mapping of variables defined on set X to Boolean values. Let Φ be a QBF. If there is an assignment $\mathcal{A}_X$ that evaluates formula Φ as *true*, then Φ is *satisfiable* (SAT). A set $S \subseteq X$ is a *model* of Φ if the assignment $\mathcal{A}_X$ maps *true* to every element of S. If there is no such assignment that evaluates the formula as *true*, we say the formula is *unsatisfiable* (UNSAT).

3 Reduction based on acceptance sets relations / SCC-based reduction

In this section, the simplification strategy is to find relations between acceptance sets and reduce the redundant ones. SCC-based simplification is the coarsest simplification we propose.

Let $\mathcal{A}$ be an original automaton and let ψ be an acceptance condition of $\mathcal{A}$. Every run of $\mathcal{A}$ has an infinite suffix that takes place within one SCC S_i. Thus the evaluation of ψ depends purely on the SCC S_i and we can optimize the acceptance condition for each SCC separately. :w This optimization consists of removing redundant terms from acceptance condition and relabeling of acceptance marks on the transitions. The state and transition structure of the SCC does not change. This way we obtain set of simplified acceptance conditions that we merge into new acceptance formula ψ'. The automaton with relabeled transitons and new acceptance condition ψ' we denote $\mathcal{A}'$. Finally, we ensure that automaton $\mathcal{A}'$ is equivalent to the original automaton $\mathcal{A}$.

3.1 S_i simplification

At first, we remove the unused acceptance marks and transform the SCC's acceptance condition ψ_i to *disjunctive normal form* (DNF), so we have a unified shape of the condition. If TELA acceptance condition is in DNF, two distinct terms in one disjunct can only appear in three possible forms:

– Inf❷$i \land$ Inf❹j
– Fin⑧$i \land$ Fin❹j
– Fin⑧$i \land$ Inf❹j

Since formula ψ_i is in DNF, we can represent it as a set of disjuncts $\overline{\psi_i} = \{D_1, D_2, \ldots D_k\}$.

Let $D_k \in \overline{\psi_i}$ be a disjunct of formula $\overline{\psi_i}$ and $C_j \in D_k$ a conjunct of disjunct D_k. Furthermore let ⓿i, ❹j, ❽k, ❷l $\in M$ be distinct acceptance marks that occur in $\mathcal{A}$.

In the next section, some properties of an automaton allow us to substitute a Boolean value *true* or *false* for a particular term of the formula ψ_i. The consequences of this substitution are divided into a number of cases. We represent it on the set-format of formula we just defined.

- If C_j is substituted by *true*, the conjunct is omitted from D_k. Thus $D_k = D_k \setminus \{C_j\}$.
- If C_j is substituted by *false*, the conjunct causes that the evaluates of the whole D_k is *false*. Thus $\overline{\psi_i} = \overline{\psi_i} \setminus \{D_k\}$.

That being said, we can define the reduction techniques.

4 QBF-based reduction

- obecny popis konstrukce formule
- (level 3)
- level 4

5 Implementation and experimental evaluation

6 Conclusions