Masaryk University
Faculty of Informatics

«title»

Bachelor's Thesis

«author»

Brno, Fall 2020

Replace this page with a copy of the official signed thesis assignment and a copy of the Statement of an Author.

Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

«author»

Advisor: «advisor»

Abstract

«abstract»

Keywords

«keywords»

Contents

List of Tables

List of Figures

1 Introduction

2 Preliminaries

2.1 Büchi Automaton

A nondeterministic Büchi automaton (BA) is a tuple $A = (\Sigma, Q, q_0, \Delta, \Gamma)$, where

- Σ is a finite alphabet
- Q is finite set of states
- $q_0 \in Q$ is the initial state
- $\Delta \subseteq Q \times \Sigma \times Q$ are transitions
- $\Gamma \subseteq \Delta$ is the transition-based acceptance condition

run A run r of A on $w \in \Sigma^\omega$ is an ω-word $r_0, w_0, r_1, w_1, ...$ in $(Q \times \Sigma)^\omega$ such that $r_0 = q_0 \wedge \forall i > 0, (r_{i-1}, w_{i-1}, r_i) \in \Delta$

inf(r) We write $inf(r) \subseteq \Delta$ for the set of transitions that appear infinitely often in the run r.

accepting run A run r is accepting if $inf(r) \cap \Gamma \neq \emptyset$

language The language $L_A \subseteq \Sigma^\omega$ is recognized by A.
$\forall w \in L_A \exists r$ on w such that r is accepting.

ω-regular language A language is ω-regular if it is accepted by BA.

deterministic automaton $A = (\Sigma, Q, q_0, \Delta, \Gamma)$ is deterministic if $(q, \rho, q^{'}), (q, \rho, q^{''}) \in \Delta \implies q^{'} = q^{''}$

complete automaton A is complete if, $\forall w \in \Sigma, \forall q \in Q, \exists (q, w, q^{'}) \in \Delta$. A word in Σ^ω has exactly one run in a deterministic, complete automaton.

2.2 Markov Decision Processes

2.2.1 xd

′ ′

GF MDP, model checking

2.3 Algorithms

BP + both slim

3 Implementation

3.1 Technologies

3.2 Implementation inside Seminator

4 Evaluation

4.1 Alternative Algorithm

4.2 Different Implementation - ePMC

4.3 Semi-deterministic Automata

5 Conclusion