SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis, Master's Thesis, ... in Informatics

Thesis title

Author

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis, Master's Thesis, ... in Informatics

Thesis title

Titel der Abschlussarbeit

Author:	Author
Supervisor:	Supervisor
Advisor:	Advisor
Submission Date:	Submission date

I confirm that this bachelor's thesis, master's thesis, ... is my own work and I have documented all sources and material used.

Munich, Submission date Author

Acknowledgments

Abstract

Contents

1 Introduction

The main purpose of the thesis is to use a very recently studied class of formal languages, the weakly acyclic languages, to address the coverability problem in Petri nets.

jonas war hier

2 Preliminaries

This chapter aims to define a standardized formal notation of the theoretical models used in thesis.

2.1 TODOS

- include definition of a transducer
- def of upward closed sets / markings

2.2 Formal Languages

2.3 Petri Nets

A petri net is tuple (S, T, F, M_0) where

- S is a finite set of places
- T is a finite set of transitions disjoint from S
- $F \subseteq (S \times T) \cup (T \times S)$ is a flow relation
- $M_0 : S$ is an initial marking

A marking of a net is a mapping $M : S -> nat$, assigning each place a number of tokens. We represent these markings by using vectors. We fix a total order on the places of the net

2.4 Coverability?

3 Weakly Acyclic Languages

3.1 Definition for DFAs

Definition 1 ([MB]). *Let $A = (Q, \Sigma, q_0, \delta, F)$ be a DFA and let $\alpha(w)$ be the set of letter, which occur within in word w. DFA A is weakly acyclic, if $\delta(q, w) = q$ implies $\delta(q, a) = q$ for every $a \in \alpha(w)$*

The following definitions are equivalently expressing, that a DFA $(Q, \Sigma, q_0, \delta, F)$ is *weakly acyclic*:

- the binary relation $\precsim$ over $Q \times Q$ with $q \precsim q'$ if $\delta(q, w) = q'$ is a partial order
- each strongly connected component of underlying directed graph contains a single state
- the underlying directed graph does not contain any simple cycle except from self-loops

Lemma 2. *Let A be a weakly acyclic DFA. The minimal DFA that accepts $L(A)$ is also weakly acyclic.*

Proof. See [[MB, Proposition 4]] □

3.2 Other Representations

As with regular languages, there are equivalent ways to represent weakly acyclic languages. Let $\alpha(w)$ be the set of letter, which occur within in word w. Blondin et al. [MB] have shown, that weakly acyclic DFAs and weakly acyclic NFAs represent the same class of weakly acyclic languages.

3.2.1 Weakly Acyclic NFA

An NFA $(Q, \Sigma, q_0, \delta, F)$ is *weakly acyclic* if

- $q \in \delta(q, w)$ implies $\delta(q, a) = q$ for every $a \in \alpha(w)$

- the underlying directed graph does not contain any simple cycle except from self-loops and nondeterminism with a letter a can only appear from a state with no self-loop of a.

3.2.2 Weakly Acyclic Expressions

The class of weakly acyclic languages can also be characterized by *weakly acyclic expressions* of the following form

$$r ::= \emptyset \mid \Gamma^* \mid \Lambda^* a r \mid r + r \quad \text{where} \quad \Gamma, \Lambda \subseteq \Sigma \quad \text{and} \quad a \in \Sigma \setminus \Lambda$$

3.3 Properties

3.3.1 Position in Language Hierarchy

The weakly acyclic language lie strictly in between finite and regular languages. They are within the regular languages, as they can be characterized with DFAs, NFAs and regular expressions. Furthermore, every finite language is weakly acyclic, as only a self loop in the trap state is necessary.

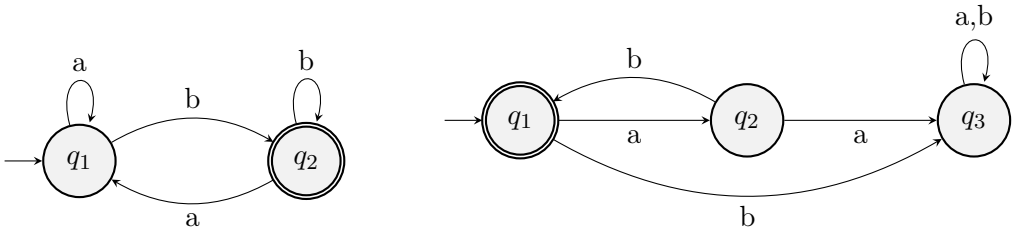
3.3.2 Closure Properties

Weakly acyclic languages are closed under union, intersection and complementation, but not under concatenation or Kleene star.

As weakly acyclic languages can be described by DFAs as in **??**, complementing a DFA still preserves its weakly acyclic structure. Union is already present in the definition of weakly acyclic expressions in Subsection 3.2.2. Therefore, the languages are also closed under intersection, which can be expressed with the combination of union and complementation.

Weakly acyclic languages are not closed under concatenation. The expressions $(\boldsymbol{a+b})^*$ and $\boldsymbol{b}$ are weakly acyclic by themselves. The language of their concatenation $(\boldsymbol{a+b})^*\boldsymbol{b}$ is depicted in a minimal DFA in Figure 3.1a.

As this automaton contains a cycle of length 2, this language is not weakly acyclic. The same argumentation can be made for the Kleene star closure, with $\boldsymbol{ab}$ being a weakly acyclic language, but $(\boldsymbol{ab})^*$ represented in Figure 3.1b is not weakly acyclic.



(a) Minimal DFA of $L((\boldsymbol{a}+\boldsymbol{b})^*\boldsymbol{b})$ (b) Minimal DFA of $L((\boldsymbol{ab})^*)$

Figure 3.1: Not weakly acyclic DFA

4 A Data Structure for Weakly Acyclic DFAs

In order to use weakly acyclic languages efficiently, a data structure for storing the automata efficiently and offering binary set operations is required.

4.1 Master Automaton

A theoretical model for representing multiple dfas.

4.2 Table of Nodes

An implementation of the idea of master automaton with successors

4.2.1 make

Adding new nodes to table

4.2.2 create

creating nodes based on star words

4.3 Recursive Algorithms on the Table: union

- recursive structure of algorithms with recursion end with emptyset or sigmastar

4.4 Transducer

4.5 Pre Algorithm

4.5.1 proof of correctness

5 Coverability Problem in Petri Nets

Algorithm 1 Backwards Reachability Algorithm

Require: N, M_0, M
  $\mathcal{M} \leftarrow \{M^{'} : M^{'} \geq M\}$
  $\mathcal{M}_{old} \leftarrow \emptyset$
  while true **do**
    $\mathcal{M}_{old} \leftarrow \mathcal{M}$
    $\mathcal{M}_{pre} \leftarrow \mathcal{M} \cup pre(\mathcal{M})$
    if $M_0 \in \mathcal{M}$ **then**
      return true
    end if
    if $\mathcal{M} = \mathcal{M}_{old}$ **then**
      return false
    end if
  end while

6 Implementation

6.1 The High-Level Idea

The main purpose of the thesis is to use weakly acyclic languages to address the coverability problem in Petri nets. Therefore, the concepts from Chapter 3 are used to adapt the Backwards Reachability Algorithm in Algorithm 1.

The Algorithm 1 is working with infinite upward-closed sets of markings, like $\mathcal{M}$ and $\mathcal{M}_{old}$. Since automata are characterizing languages, which are a set of words, they are a way to describe these infinite sets in a compact finite representation. In fact, for the representation of the upward closed markings, the subset of weakly acyclic languages are sufficient.

Therefore, the idea is to represent each of the upward-closed sets, which are used in Algorithm 1, by a weakly acyclic DFA. To manage all those DFAs efficiently, a data structure *table* is introduced. The *table* stores a set of weakly acyclic DFAs using the concept of the master automaton in Chapter 3. It further provides the required set operations *union* and *intersection*, which calculate the union/intersection of two weakly acyclic languages. Furthermore, there are encoding functions, which take a marking, convert it into a weakly acyclic DFA, and adds it to the *table*.

For the main operation *pre*, to calculate all possible predecessor markings, the structure of the petri net N comes into use.

6.2 Table of Nodes

6.3 Transducer

6.4 Petri Net

6.5 Parser

6.6 CorrectnessCheck

Algorithm 2 Backwards Reachability Algorithm using Weakly Acyclic Automata

Require: N, M_0, M
  $table \leftarrow EmptyTableOfNodes$
  $transducer \leftarrow \mathsf{net2transducer}(N)$
  $\mathcal{M}_0 \leftarrow table.\mathsf{encode}(M_0)$
  $\mathcal{M} \leftarrow table.\mathsf{encodeUpwardClosed}(M)$
  $\mathcal{M}_{old} \leftarrow q_\emptyset$
  while true **do**
    $\mathcal{M}_{old} \leftarrow \mathcal{M}$
    $\mathcal{M} \leftarrow table.\mathsf{pre}(transducer, \mathcal{M})$
    $\mathcal{M} \leftarrow table.\mathsf{union}(\mathcal{M}_0, \mathcal{M}_{pre})$
    if $table.\mathsf{intersection}(\mathcal{M}_0, \mathcal{M}) \neq q_\emptyset$ **then**
      return true
    end if
    if $\mathcal{M} = \mathcal{M}_{old}$ **then**
      return false
    end if
  end while

7 Evaluation

8 Conclusion

bfdfbdufbqdfdffdffff

9 Introduction

9.1 Section

Citation test [Lam94].

Acronyms must be added in `main.tex` and are referenced using macros. The first occurrence is automatically replaced with the long version of the acronym, while all subsequent usages use the abbreviation.

E.g. `\ac{TUM}, \ac{TUM}` ⇒ Technical University of Munich (TUM), TUM

For more details, see the documentation of the `acronym` package[1].

9.1.1 Subsection

See Table 9.1, Figure 9.1, Figure 9.2, Figure 9.3.

Table 9.1: An example for a simple table.

A	B	C	D
1	2	1	2
2	3	2	3

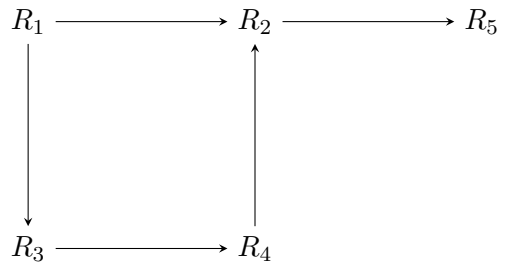


Figure 9.1: An example for a simple drawing.

[1] `https://ctan.org/pkg/acronym`

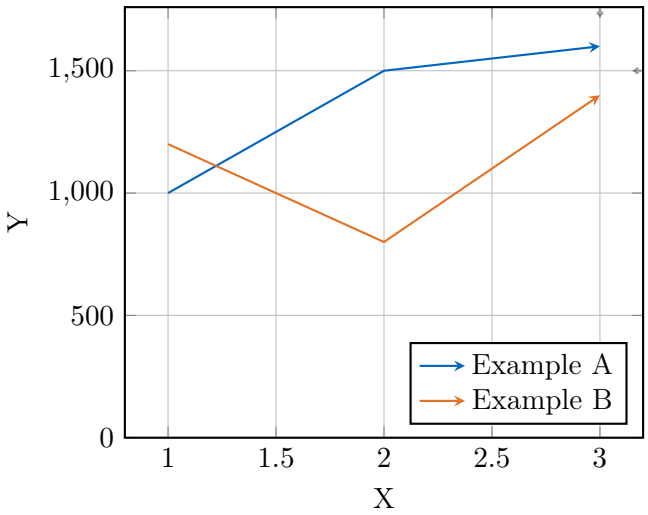


Figure 9.2: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 9.3: An example for a source code listing.

Abbreviations

TUM Technical University of Munich

List of Figures

List of Tables

Bibliography

[Lam94] L. Lamport. *LaTeX : A Documentation Preparation System User's Guide and Reference Manual*. Addison-Wesley Professional, 1994.

[MB] J. E. M. Blondin M. Cadilhac. "Symbolic representation of weakly acyclic sets."