

Bachelor Thesis

Visualizing Dynamic Programming on Tree Decompositions

MARTIN RÖBKE

born: 04.03.1995 in Dresden, Germany

matriculation number: 3949819

martin.roebke@tu-dresden.de

Technische Universität Dresden

Faculty of Computer Science

International Center For Computational Logic

Supervisor: Dr. Johannes Fichte

Second evaluator: Prof. Dr. rer. nat. Stefan Gumhold

Dresden, May 28, 2020

Abstract

The thesis is about a practical and lightweight implementation for visualizing dynamic programming on tree decompositions. I created the python-package tdvisu for the purpose of visualizing, teaching and analyzing the solving process of MSOL-problems using tree decomposition. As two reference implementations of dynamic programming on tree decompositions the projects GPUSAT and dpdb were chosen.

???????Who benefits from using

Contents

1	Introduction	4
2	Background	5
2.1	MSOL	5
2.2	Tree Decomposition	5
2.3	Courcelle's Theorem	5
3	Concept	7
4	My Visualization Project	8
4.1	Integration in GPUSAT	9
4.2	Integration in dpdb	10
5	Application and Images	11
6	Summary and Outline	12

1 Introduction

intro. mit motivation und related work, state of the art, advancements.

Idee für Projekt: Johannes

Visualization Pipeline

Stand Umsetzung, Tools: Slack, Trello, GitHub, Presentations

2 Background

2.1 MSOL

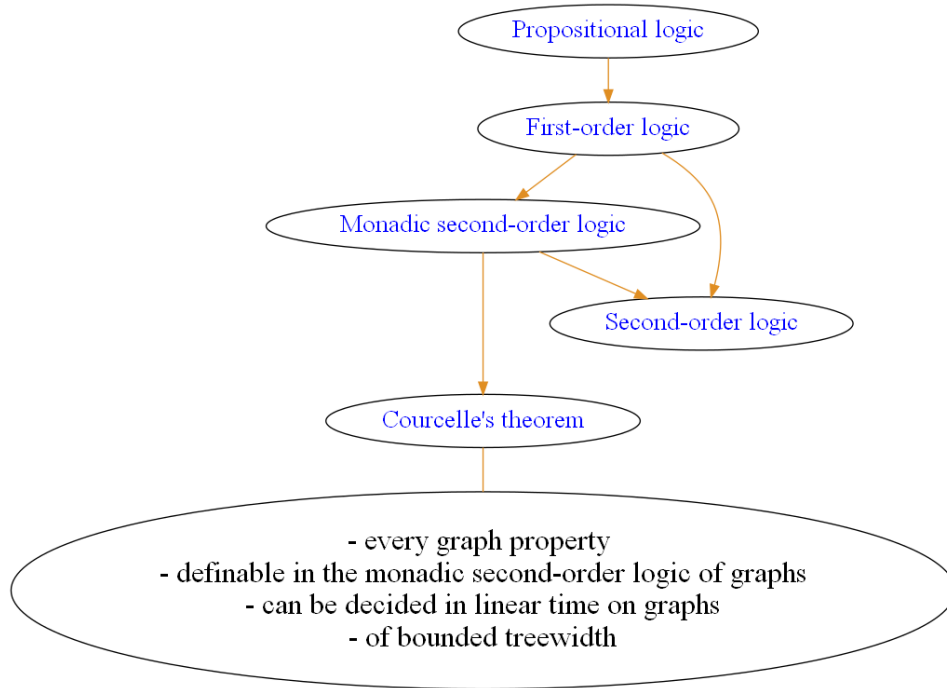


Figure 1: From propositional logic to monadic second order logic

2.2 Tree Decomposition

2.3 Courcelle's Theorem

Every graph property definable in monadic second-order logic (MSO) is decidable in linear time on graphs of bounded treewidth.

Courcelle, Bruno (1990)¹

For all $k \in \mathbb{N}$ and MSO-sentences F is the decision problem for a given graph G , whether $G \models F$ is true, in time $2^{p(tw(G))} \cdot |G|$ with a polynom p decidable.

- *drawback*: still expensive ($2^{p(tw(G))}$, $2^{2^{(\#Q)}}$, large constants)

The workflow then looks like we see in figure 2.

¹Courcelle, Bruno "The monadic second-order logic of graphs. I. Recognizable sets of finite graphs", Information and Computation, 85 (1990) no. 1: 12-75

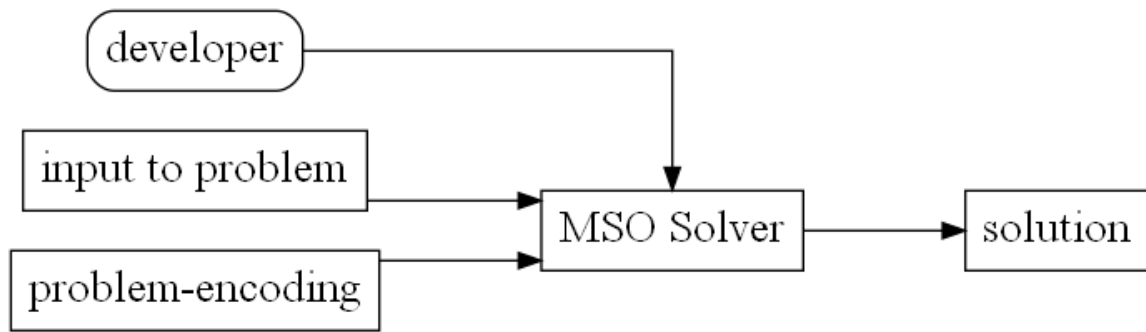


Figure 2: Implementation of the theorem

3 Concept

What I do and why I did it Research: language (python - explain) graph-construction (graphviz vs networkX), examples (diploma at first).

4 My Visualization Project

Github Objectives htd hier oder auslassen? Files / Classes / Methods Current perspective

4.1 Integration in GPUSAT

Programm Umsetzung Beispiel

4.2 Integration in dpdb

Programm Umsetzung Beispiel

5 Application and Images

beispiele und ergebnisse das vertex cover eignet sich dafür hervorragend Z.B. Fehler in VC - Visualization and htd error with seed=0 in windows

6 Summary and Outline

What is achieved? What worked good, what bad?

References