

On linear layouts with SAT

Mirco Haug

July 11, 2019

Eberhard Karls Universität Tübingen

Contents

Motivation

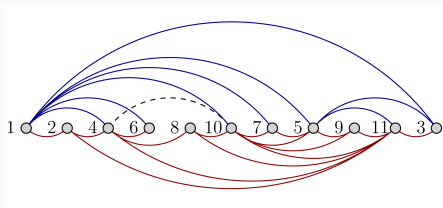
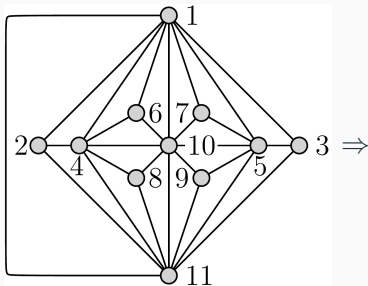
Theory

Implementation

Demo

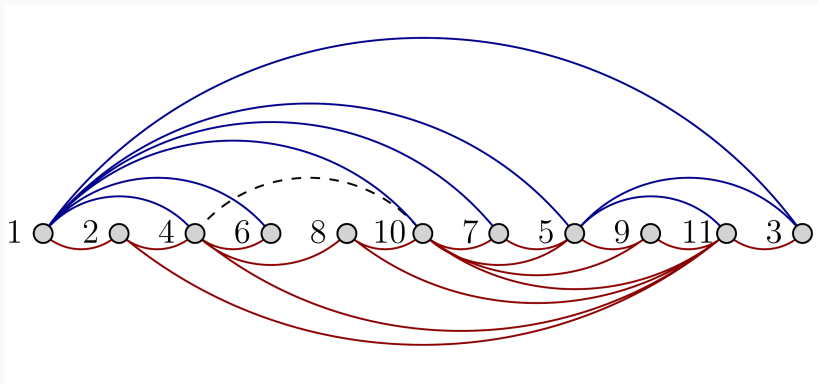
Motivation

What to do?



The bookembedding problem

- Multiple pages / Colors
- No crossings
- Node order
- Edge assignments



Approach

- Combinatoric explosion
- SAT solvers
- <http://be.cs.arizona.edu>
- Rather static
- Flexibility needed
- Old proof sketch of Yannakakis[Yan86]

Theory

- Node order $\sigma(n_1, n_2)$
- Edge assignment $\phi_{P1}(e_1)$

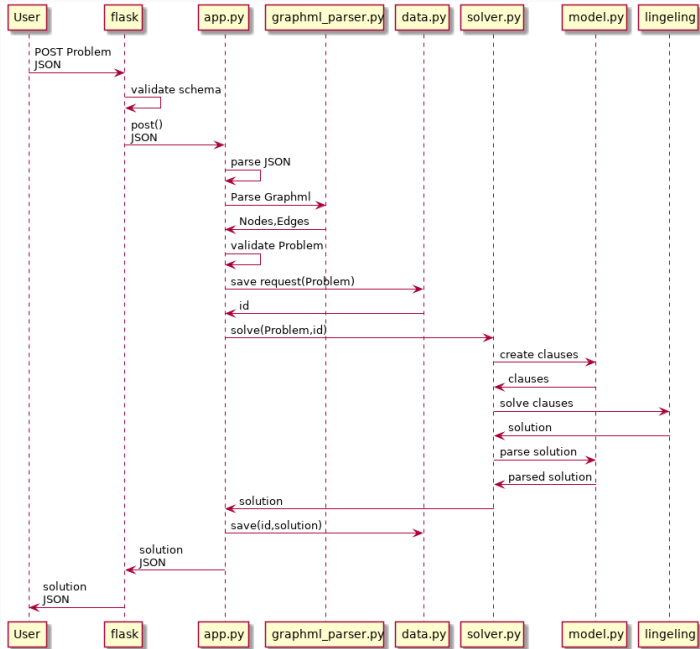
Constraints

- EDGES_ON_PAGES
- EDGES_SAME_PAGES
- EDGES_DIFFERENT_PAGES
- EDGES_TO_SUB_ARC_ON_PAGES
- EDGES_FROM_NODES_ON_PAGES
- NODES_PREDECESSOR
- NODES_REQUIRE_ABSOLUTE_ORDER
- NODES_REQUIRE_PARTIAL_ORDER
- NODES_FORBID_PARTIAL_ORDER
- NODES_CONSECUTIVE

Constraints

- EDGES_ON_PAGES
- EDGES_SAME_PAGES
- EDGES_DIFFERENT_PAGES
- EDGES_TO_SUB_ARC_ON_PAGES
- EDGES_FROM_NODES_ON_PAGES
- NODES_PREDECESSOR
- NODES_REQUIRE_ABSOLUTE_ORDER
- NODES_REQUIRE_PARTIAL_ORDER
- NODES_FORBID_PARTIAL_ORDER
- NODES_CONSECUTIVE

Implementation



Linear layout API ^{1.0}

[Base URL: /]

<http://127.0.0.1:5000/swagger.json>

Through this API one can request for a linear layout of a graph in graphml format.

The actual computation of the linear layout is done using SAT solving. The instances are solved using [lingeling](#)

default Default namespace



POST /embeddings Create a new embedding

GET /embeddings List all embeddings

GET /embeddings/{id} Get an embedding by id

Models



Main classes

- `app.py`
- `model.py`
- `solver.py`

- Schema definition
- Deserialization
- Validate

- Glue
- Service interface
- Coordination
- Calls lingeling

- Clause generation
- DIMACS Generation
- Parser for lingeling result

Demo

`http://algo.inf.uni-tuebingen.de/linearlayouts/index.
html#or174`



Michael A. Bekos, Michael Kaufmann, and Christian Zielke.
The book embedding problem from a SAT -solving perspective.

In *Lecture Notes in Computer Science*, pages 125–138.
Springer International Publishing, 2015.



Mihalis Yannakakis.

Four pages are necessary and sufficient for planar graphs (extended abstract).

In Juris Hartmanis, editor, *ACM Symposium on Theory of Computing*, pages 104–108. ACM , 1986.