Fixed-Parameter Algorithms for Graph Constraint Logic

Tatsuhiko Hatanaka*[1], Felix Hommelsheim[2], Takehiro Ito†[1], Yusuke Kobayashi‡[3], Moritz Mühlenthaler[4], and Akira Suzuki§[1]

[1]Graduate School of Information Sciences, Tohoku University, Sendai, Japan
[2]Fakultät für Mathematik, TU Dortmund University, Germany
[3]Research Institute for Mathematical Sciences, Kyoto University, Japan
[4]Laboratoire G-SCOP, Grenoble INP, Université Grenoble Alpes, France

November 23, 2020

What is Graph Constraint Logic?

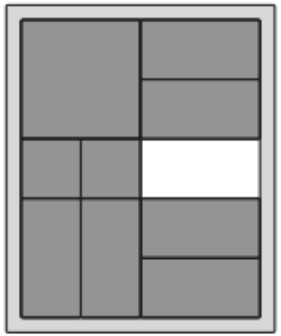
Graph Constraint Logic

Computational models

- Whole (undirected) graphs as machine
- Allows to model logical constraints as graphs

Origin

- (Sliding-block) puzzles tackled by Hearn & Demaine



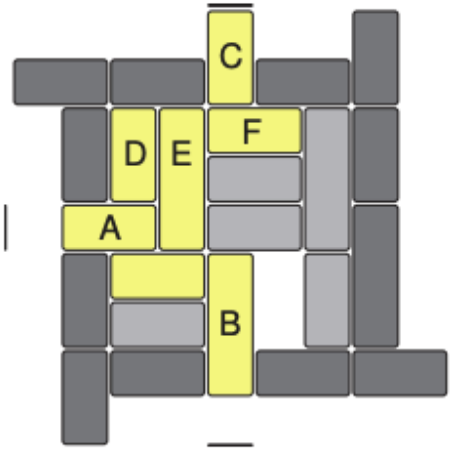
Making Moves

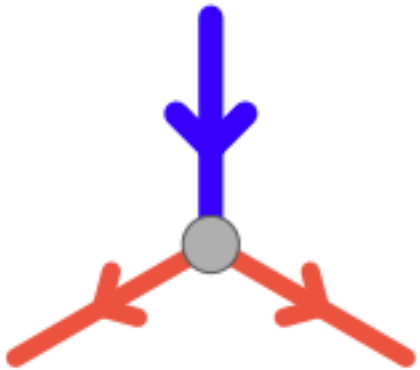
- Reversing an edge (that meets constraints)

Resulting decision problems

- Given the constraint graph:
  - Can I reverse a specific edge by sequence of valid moves?
  - Can I reach a desired configuration by a sequence of valid moves?

(Non-deterministic) Graph Constraint Logic

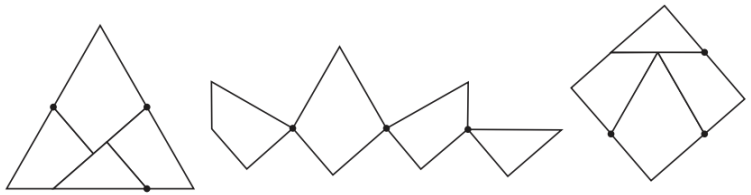




Why should we care (expect for fun) ?

Hardness proves
- Chess
- Checkers
- Go

But also useful for
- Scheduling jobs
- Optimizing workflows
- Resource allocation
- Hinged Polygon Dissection
- or even for protein folding's



and many more
- Visual representation of problem structure (Symmetry detection, ..)
- Great for debugging
- High expressivity
- ....

Goal of the paper

Goal / Intention of the paper

Show FPT for C2C and C2E

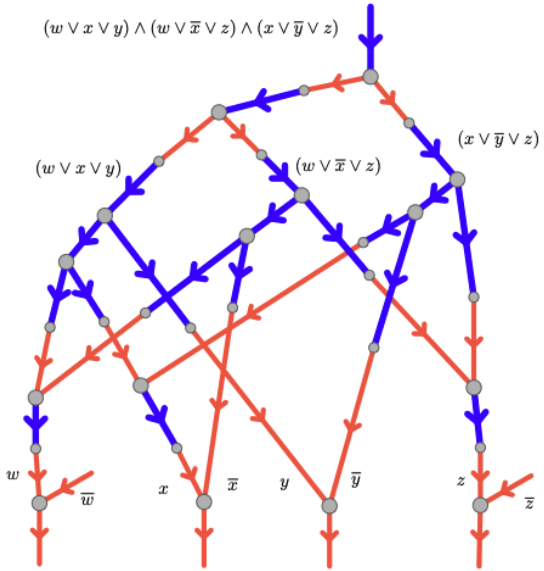
For any of the following parameters:

- Number of weight-two edges **(Blue Edges)**

- Number of weight-one edges **(Red Edges)**

- Number of **OR** vertices of an AND/OR Graph

- Number of **AND** vertices of an AND/OR Graph



C2C ... whether a given configuration of a constraint graph is reachable from another given configuration
C2E ... whether we can reach from a given configuration another one such that a given edge is reversed.

But how?

But how?

Logic-Parameters:

- Number of **OR** vertices of a Graph

  1 . Preprocessing

  2. Reduction/Transformation to BCSR*

- Number of **AND** vertices of a Graph

  → via linear kernel with 4 reduction rules

Color-Parameters:

- Number of weight-two edges **(Blue Edges)**

  → problem decomposition and abstraction

- Number of weight-one edges **(Red Edges)**

  → via linear kernel with 4 reduction rules

*Binary Constraint Satisfiability Reconfiguration

Results

Parameter(s)	C2C	C2E
treewidth and max. degree [15]	PSPACE-c	PSPACE-c
transformation length [15]	W[1]-hard	W[1]-hard
transformation length and max. degree [15]	FPT	FPT
# of AND vertices (AND/OR graphs)	FPT† (Cor. 4)	FPT† (Cor. 7)
# of OR vertices (AND/OR graphs)	FPT (Thm. 1)	FPT (Thm. 1)
# of red edges	FPT† (Thm. 3)	FPT† (Cor. 7)
# of blue edges	FPT (Thm. 8)	FPT (Cor. 18)

C2C ... whether a given configuration of a constraint graph is reachable from another given configuration
C2E ... whether we can reach from a given configuration another one such that a given edge is reversed.

Thank you for your attention