PARIS: Planning Algorithms for Reconfiguring Independent Sets

Remo Christen Salomé Eriksson Michael Katz
Christian Muise Alice Petrov Florian Pommerening
Jendrik Seipp Silvan Sievers David Speck

University of Basel IBM T.J. Watson Research Center Queen's University Linköping University

July 10, 2023

Context

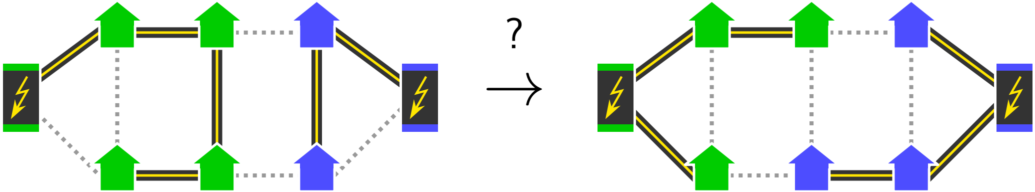
- CoRe Challenge 2022 (Combinatorial Reconfiguration)
- First iteration
- Submission **PARIS** based on planning

What's the problem?

Combinatorial Reconfiguration – Examples

Power Distribution

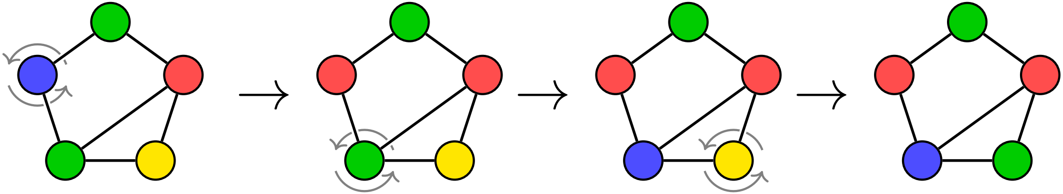
Reconfigure network while keeping all households connected.



Combinatorial Reconfiguration – Examples

Graph Coloring

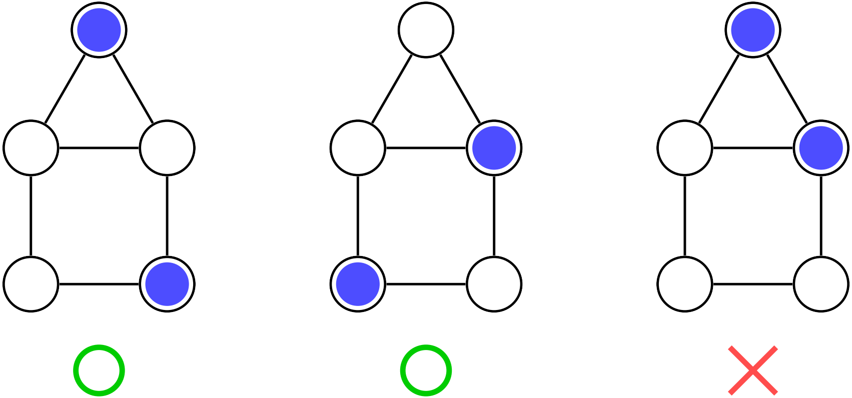
Change from one coloring to another via colorings.



Independent Sets

Independent Set

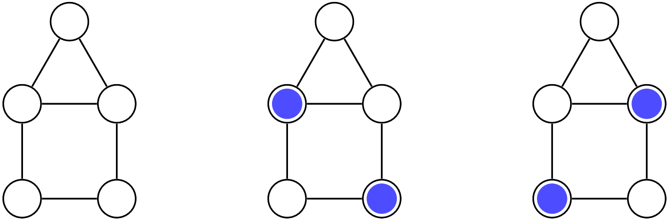
A set of vertices such that **no two are adjacent**.



Independent Set Reconfiguration

Input

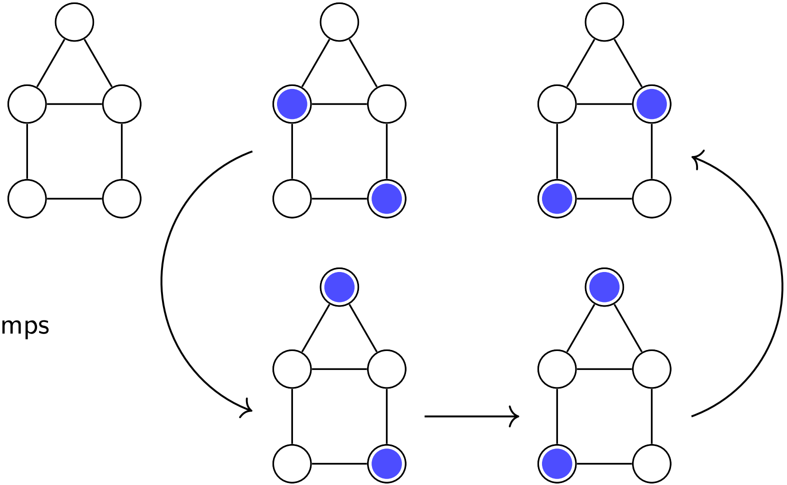
- graph
- initial set
- goal set



Independent Set Reconfiguration

Input

- graph
- initial set
- goal set



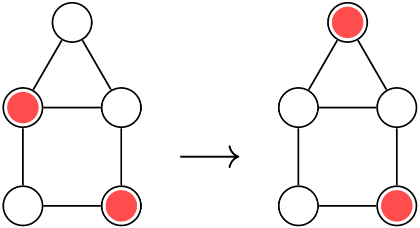
Output

- sequence of token jumps

Planning Encoding

Single action

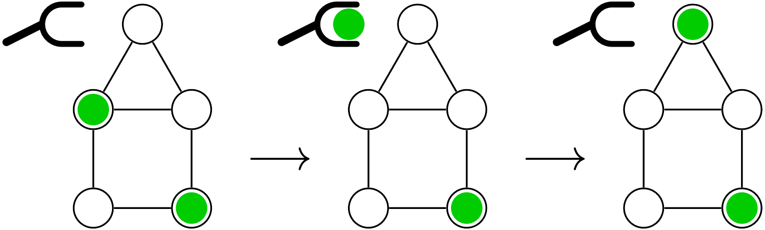
- move



Split action

- pick
- place

→ SAS$^+$



Competition

Solver Tracks

- Existent
- Shortest
- Longest

$\times$

- Single-engine
- Portfolio

Graph Tracks

- 10
- 50
- 100

No resource limits; solutions are submitted

Competition Results – Solver Tracks

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Existent Track

- Any solution
- similar to *agile* IPC track

Competition Results – Solver Tracks

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Existent Track

- Any solution
- similar to *agile* IPC track

PARIS

- GBFS + Landmarks (70min)

Competitors

2 Answer Set Programming

3 Greedy heuristic search + Bounded Model Checking

Competition Results – Solver Tracks

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Existent Track

- Any solution
- similar to *agile* IPC track

PARIS

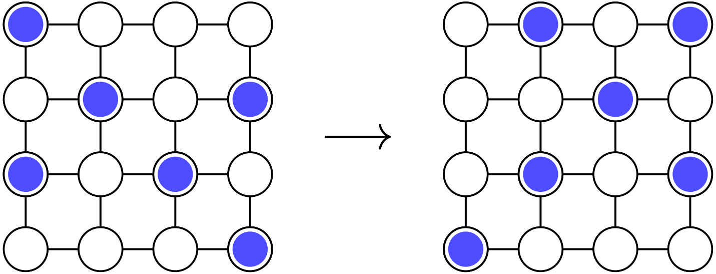
1. **Counter abstraction** (10s)
2. Symbolic search (70min)
3. A^* + Landmarks (70min)
4. GBFS + Landmarks (70min)
5. **Counter abstraction** (14h)

Competitors

2 IDA^* + Breadth-first search

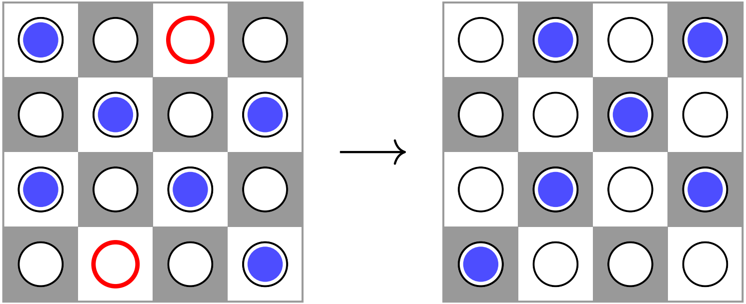
Counter Abstraction – Motivation

- Grid instances
- 4×4 to 200×200



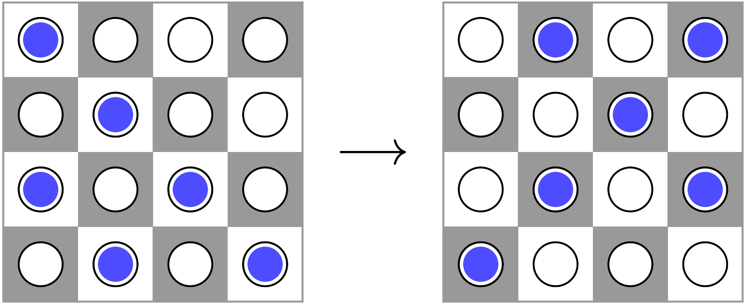
Counter Abstraction – Motivation

- Grid instances
- 4×4 to 200×200
- 1–2 gaps
- $n \times n$ unsolvable if fewer than $n/2$ gaps



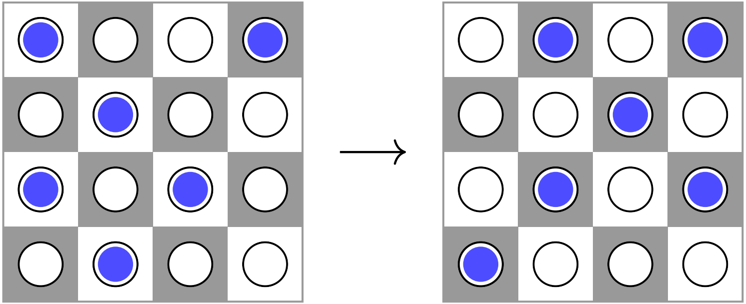
Counter Abstraction – Motivation

- Grid instances
- 4×4 to 200×200
- 1–2 gaps
- $n \times n$ unsolvable if fewer than $n/2$ gaps



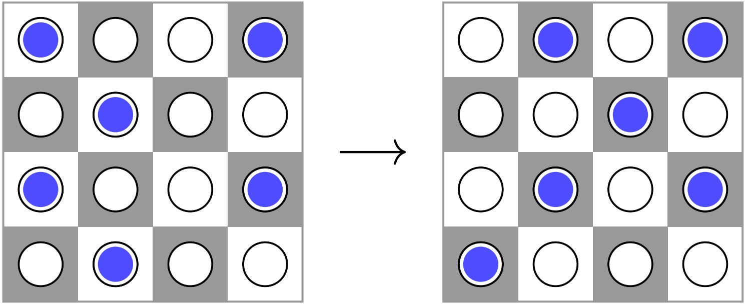
Counter Abstraction – Motivation

- Grid instances
- 4×4 to 200×200
- 1–2 gaps
- $n \times n$ unsolvable if fewer than $n/2$ gaps



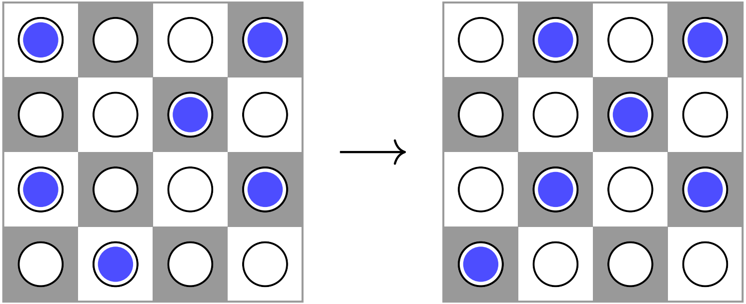
Counter Abstraction – Motivation

- Grid instances
- 4×4 to 200×200
- 1–2 gaps
- $n \times n$ unsolvable if fewer than $n/2$ gaps



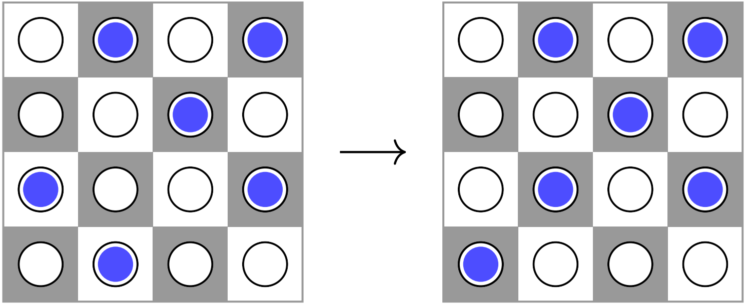
Counter Abstraction – Motivation

- Grid instances
- 4×4 to 200×200
- 1–2 gaps
- $n \times n$ unsolvable if fewer than $n/2$ gaps



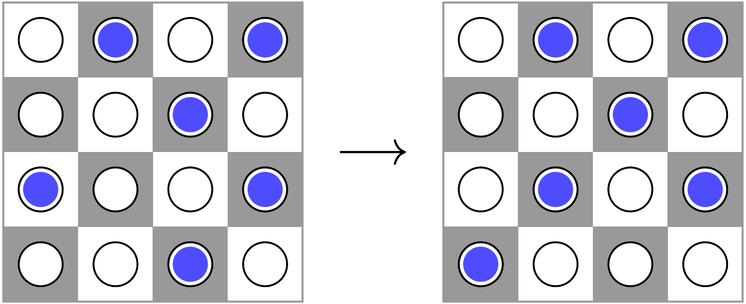
Counter Abstraction – Motivation

- Grid instances
- 4×4 to 200×200
- 1–2 gaps
- $n \times n$ unsolvable if fewer than $n/2$ gaps



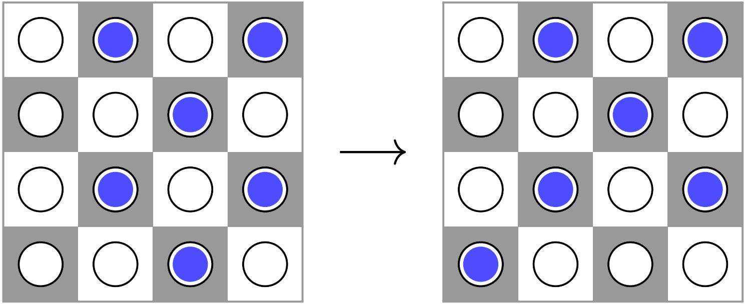
Counter Abstraction – Motivation

- Grid instances
- 4×4 to 200×200
- 1–2 gaps
- $n \times n$ unsolvable if fewer than $n/2$ gaps



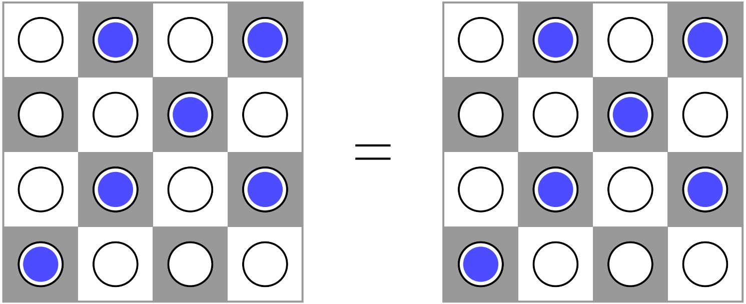
Counter Abstraction – Motivation

- Grid instances
- 4×4 to 200×200
- 1–2 gaps
- $n \times n$ unsolvable if fewer than $n/2$ gaps



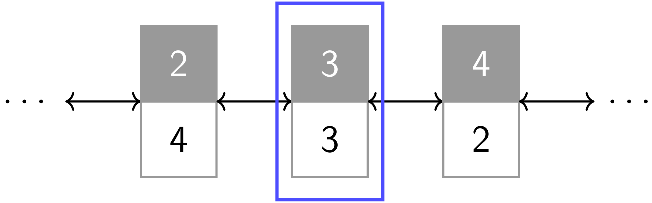
Counter Abstraction – Motivation

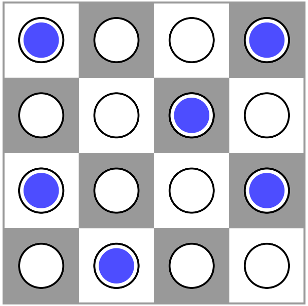
- Grid instances
- 4×4 to 200×200
- 1–2 gaps
- $n \times n$ unsolvable if fewer than $n/2$ gaps



Counter Abstraction

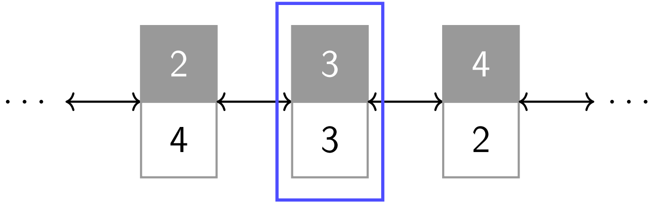
- Color the graph
- Count number of tokens on each color
- Abstract states:



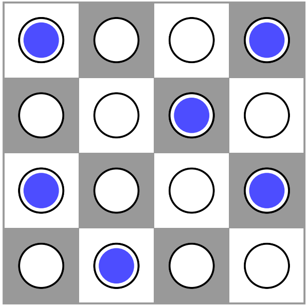


Counter Abstraction

- Color the graph
- Count number of tokens on each color
- Abstract states:



- Encode independent set + count constraints as **MIP**
- If constraints unsatisfiable for abstract state → **prune**
- Fully explored abstract state space → **unsolvable**



Competition Results – Solver Tracks

	existent	shortest	longest
single-engine	1	3	1
portfolio	1, 3	2	1, 3

Shortest Track

- Shortest solution among competitors
- similar to *satisficing* IPC track

Competition Results – Solver Tracks

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Shortest Track

- Shortest solution among competitors
- similar to *satisficing* IPC track

PARIS

- GBFS + Landmarks (70min)

Competitors

1 Answer Set Programming

2 Reinforcement Learning

Competition Results – Solver Tracks

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Shortest Track

- Shortest solution among competitors
- similar to *satisficing* IPC track

PARIS

1. Counter abstraction (10s)
2. Symbolic search (70min)
3. A^* + Landmarks (70min)
4. GBFS + Landmarks (70min)
5. Counter abstraction (14h)

Competitors

1 Greedy heuristic search + Bounded Model Checking

3 Answer Set Programming

Competition Results – Solver Tracks

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Longest Track

- Longest loopless solution among competitors
- no IPC equivalent

Competition Results – Solver Tracks

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Longest Track

- Longest loopless solution among competitors
- no IPC equivalent

PARIS

- Symbolic top-k search (70min)

Competitors

2 Answer Set Programming

3 Bounded Model Checking

Competition Results – Solver Tracks

	existent	shortest	longest
single-engine	1	3	1
portfolio	1 3	2	1 3

Longest Track

- Longest loopless solution among competitors
- no IPC equivalent

PARIS

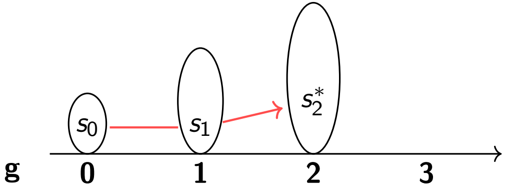
1. GBFS + Landmarks (5min)
2. **Symbolic top-k search** (65min)

Competitors

2 Answer Set Programming

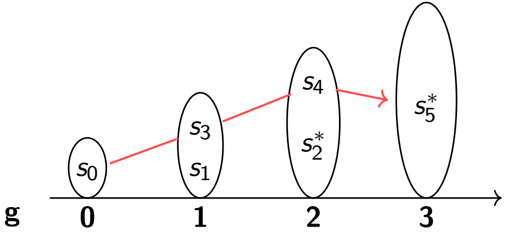
Symbolic Top-k Search

- Run loopless symbolic top-k search
- Reconstruct **one** plan per cost
- **Iteratively** find longer plans



Symbolic Top-k Search

- Run loopless symbolic top-k search
- Reconstruct **one** plan per cost
- **Iteratively** find longer plans



Competition Results – Graph Track

	10	50	100
PARIS	3	2	2

Graph

- Find difficult graphs
- Fixed number of nodes
- Longest optimal sequence

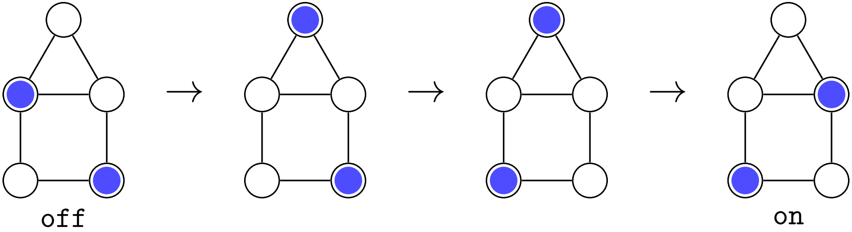
PARIS

- Great graphs (pretty)

Competitors

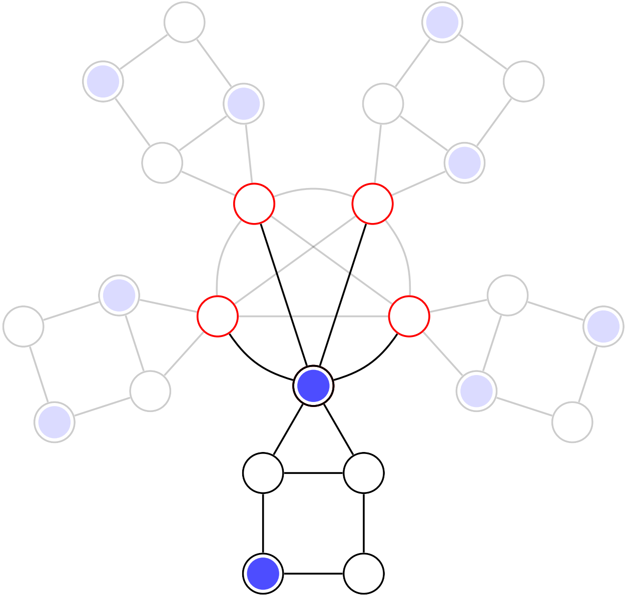
1 Slightly “better” graphs (not pretty)

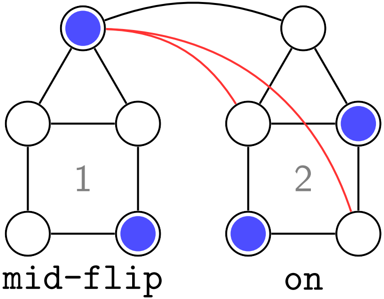
The House Widget



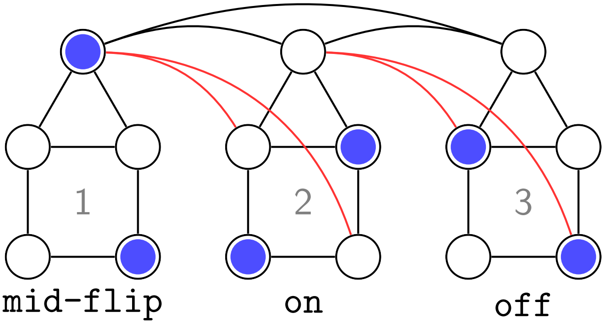
- Optimal for $n = 5$
- Cannot fit more than 2 tokens
- “Anchor” is occupied throughout flip

- Anchors fully connected and occupied during flip
- One flip at a time

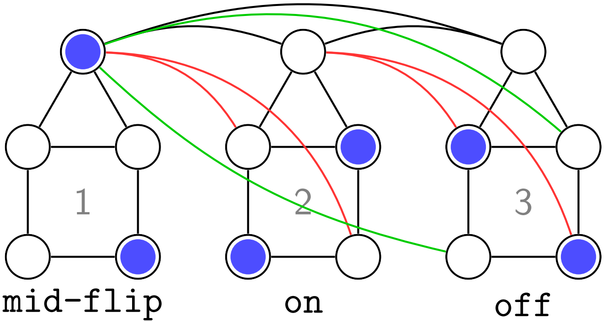




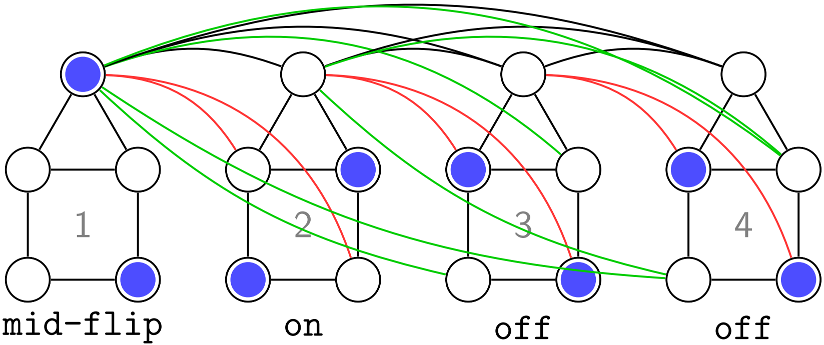
- Rule 1: house $k+1$ must be on



- Rule 1: house $k+1$ must be on



- Rule 1: house $k+1$ must be on
- Rule 2: houses $\geq k+2$ must be off



- Rule 1: house $k+1$ must be on
- Rule 2: houses $\geq k+2$ must be off
- Start: $\langle$off, off, ..., off$\rangle$
- Goal: $\langle$on, off, ..., off$\rangle$