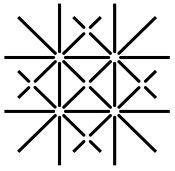
Bachelor Thesis

Abstraction Heuristics for Rubik's Cube

Clemens Büchner

Departement of Mathematics and Computer Science
University of Basel

June 8, 2018



University of Basel

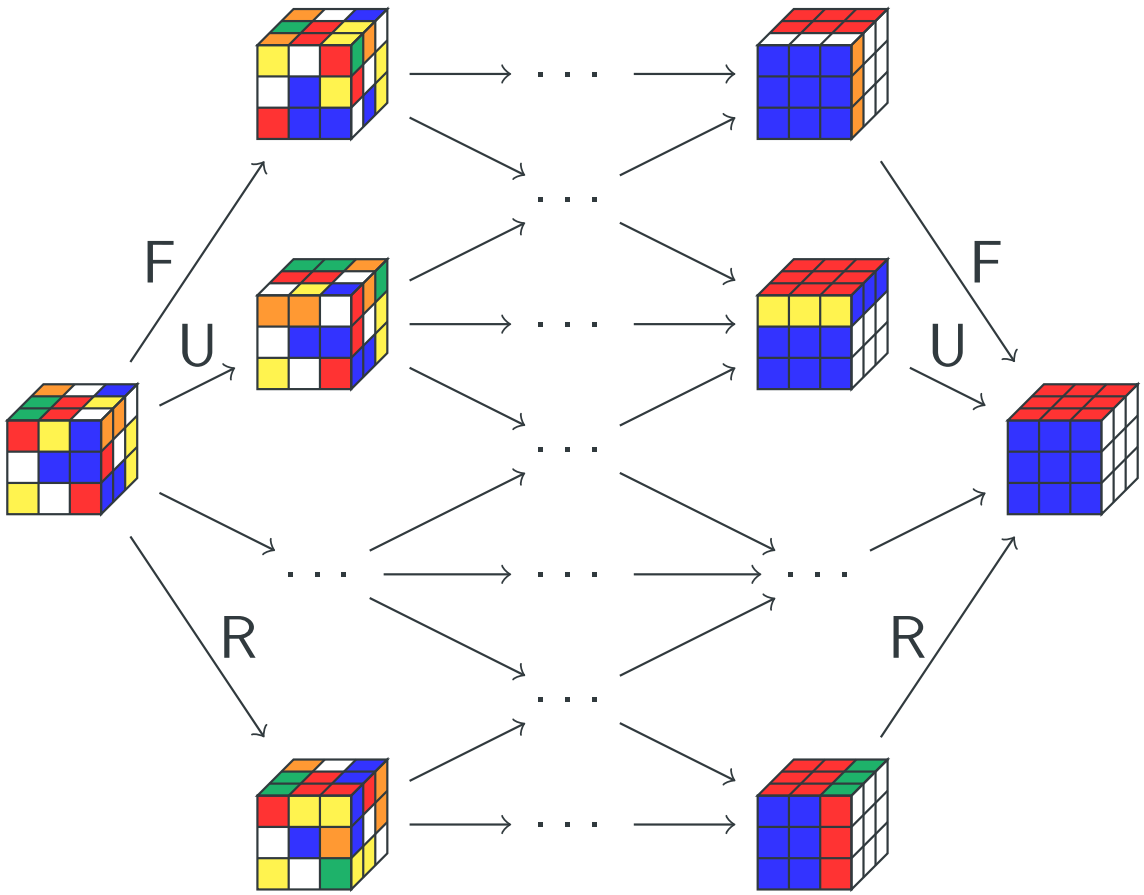

San Francisco Chronicle
A death-penalt ruling to watch
STILL ON TRACK AS COACH IS BACK


San Francisco Chronicle
A death-penalt
ruling to watch


San Francisco Chronicle
A death-penalt
ruling to watch

Planning: Figure out a sequence of actions that leads to a goal.

State Space

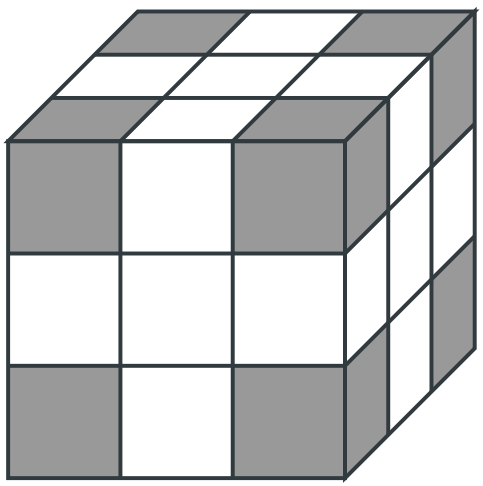


Terminology

- Cubies: small pieces on Rubik's Cube

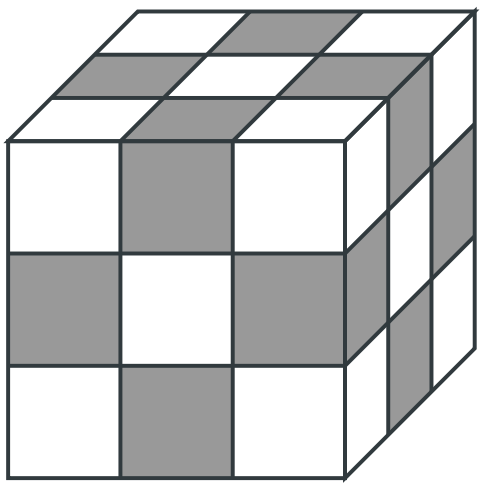
Terminology

- Cubies: small pieces on Rubik's Cube
  - corner cubies



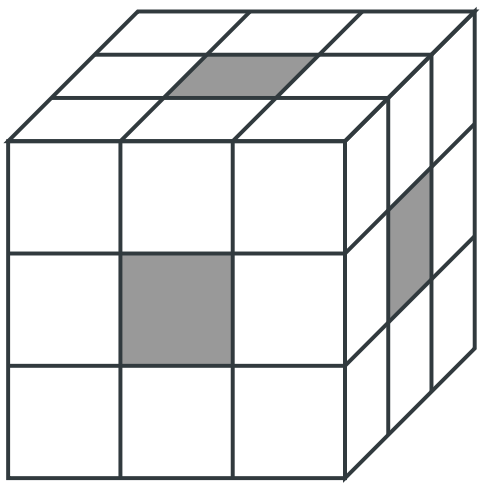
Terminology

- Cubies: small pieces on Rubik's Cube
  - corner cubies
  - edge cubies



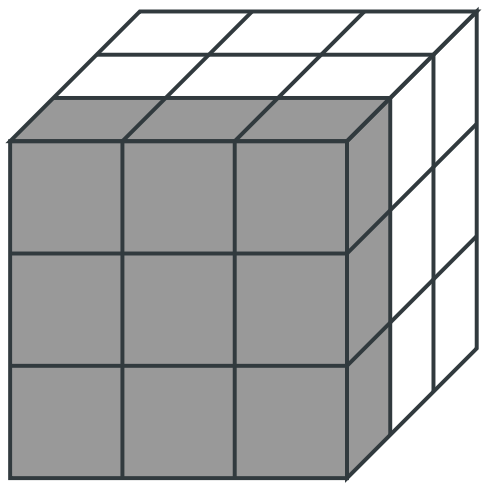
Terminology

- Cubies: small pieces on Rubik's Cube
  - corner cubies
  - edge cubies
  - center cubies



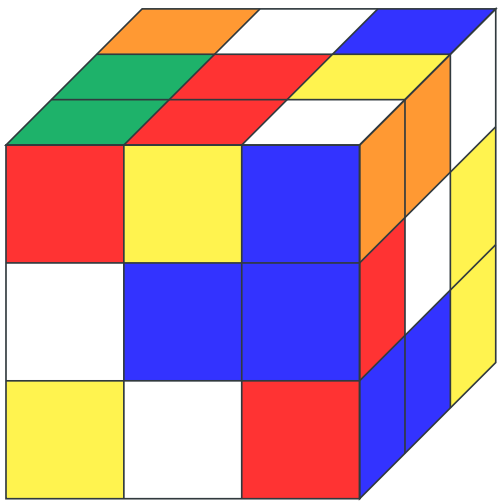
Terminology

- Cubies: small pieces on Rubik's Cube
  - corner cubies
  - edge cubies
  - center cubies
- Faces: sets of cubies
  - F, B, L, R, U, D



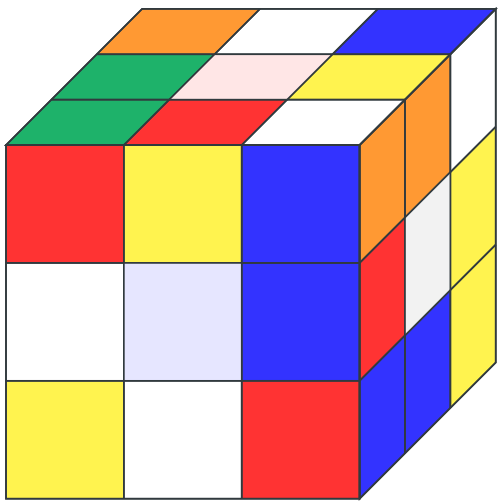
Model

- 20 variables, one for each relevant cubie
  - center cubies and cubie inside never change
  - value describes location and rotation



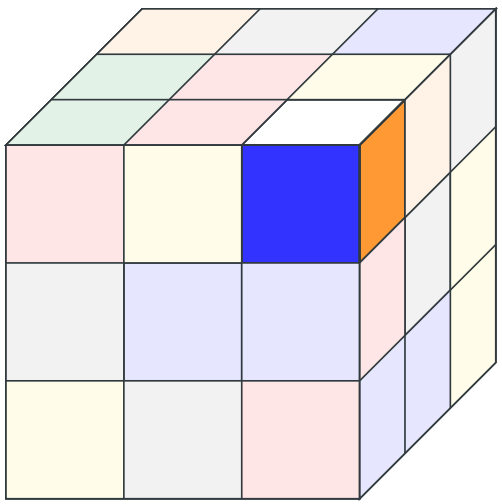
Model

- 20 variables, one for each relevant cubie
  - center cubies and cubie inside never change
  - value describes location and rotation



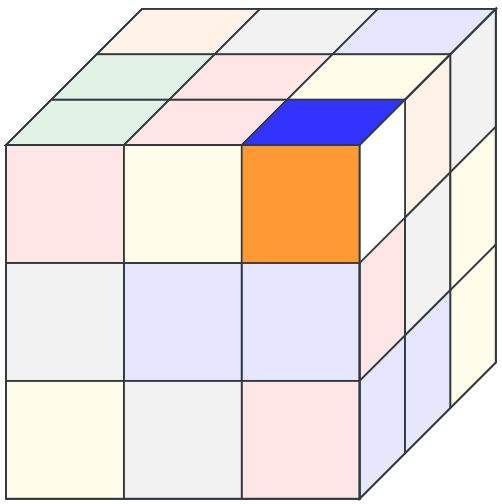
Model

- 20 variables, one for each relevant cubie
  - center cubies and cubie inside never change
  - value describes location and rotation



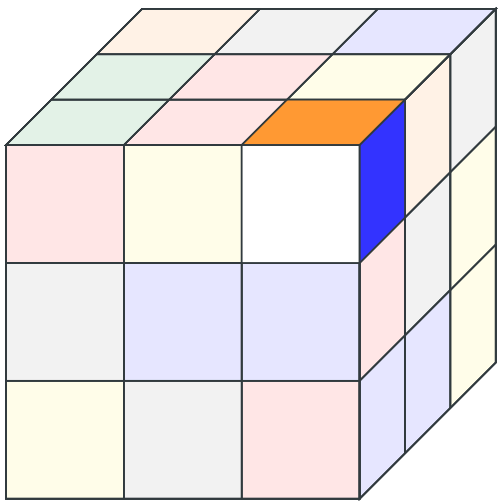
Model

- 20 variables, one for each relevant cubie
  - center cubies and cubie inside never change
  - value describes location and rotation



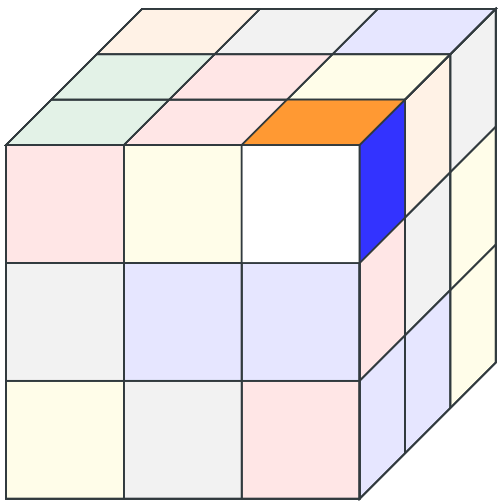
Model

- 20 variables, one for each relevant cubie
  - center cubies and cubie inside never change
  - value describes location and rotation



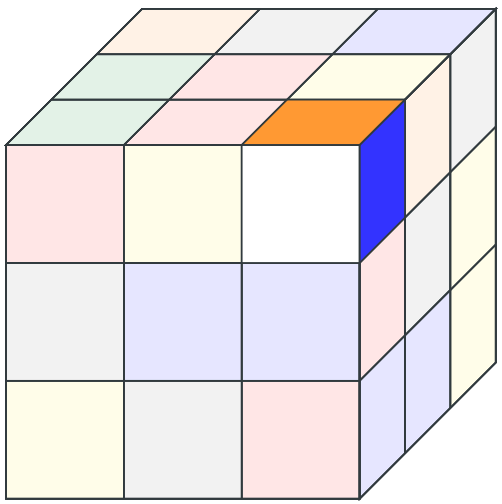
Model

- 20 variables, one for each relevant cubie
  - center cubies and cubie inside never change
  - value describes location and rotation
- size of domain = #locations · #rotations



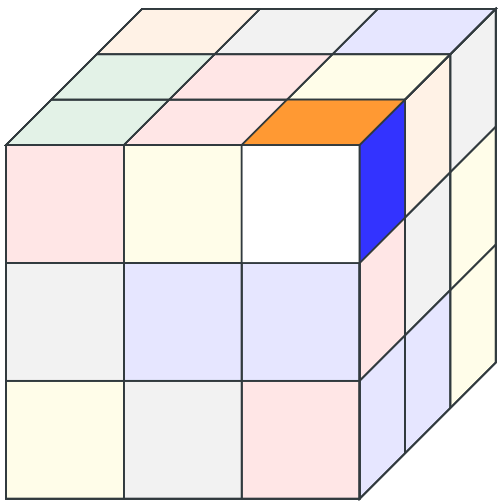
Model

- 20 variables, one for each relevant cubie
  - center cubies and cubie inside never change
  - value describes location and rotation
- size of domain = #locations · #rotations
- 18 operators, three per face
  - turn 90° clockwise, turn 90° counterclockwise and turn 180°
  - no preconditions



Model

- 20 variables, one for each relevant cubie
  - center cubies and cubie inside never change
  - value describes location and rotation
- size of domain = #locations · #rotations
- 18 operators, three per face
  - turn 90° clockwise, turn 90° counterclockwise and turn 180°
  - no preconditions
- $43,252 \cdot 10^{18}$ reachable states

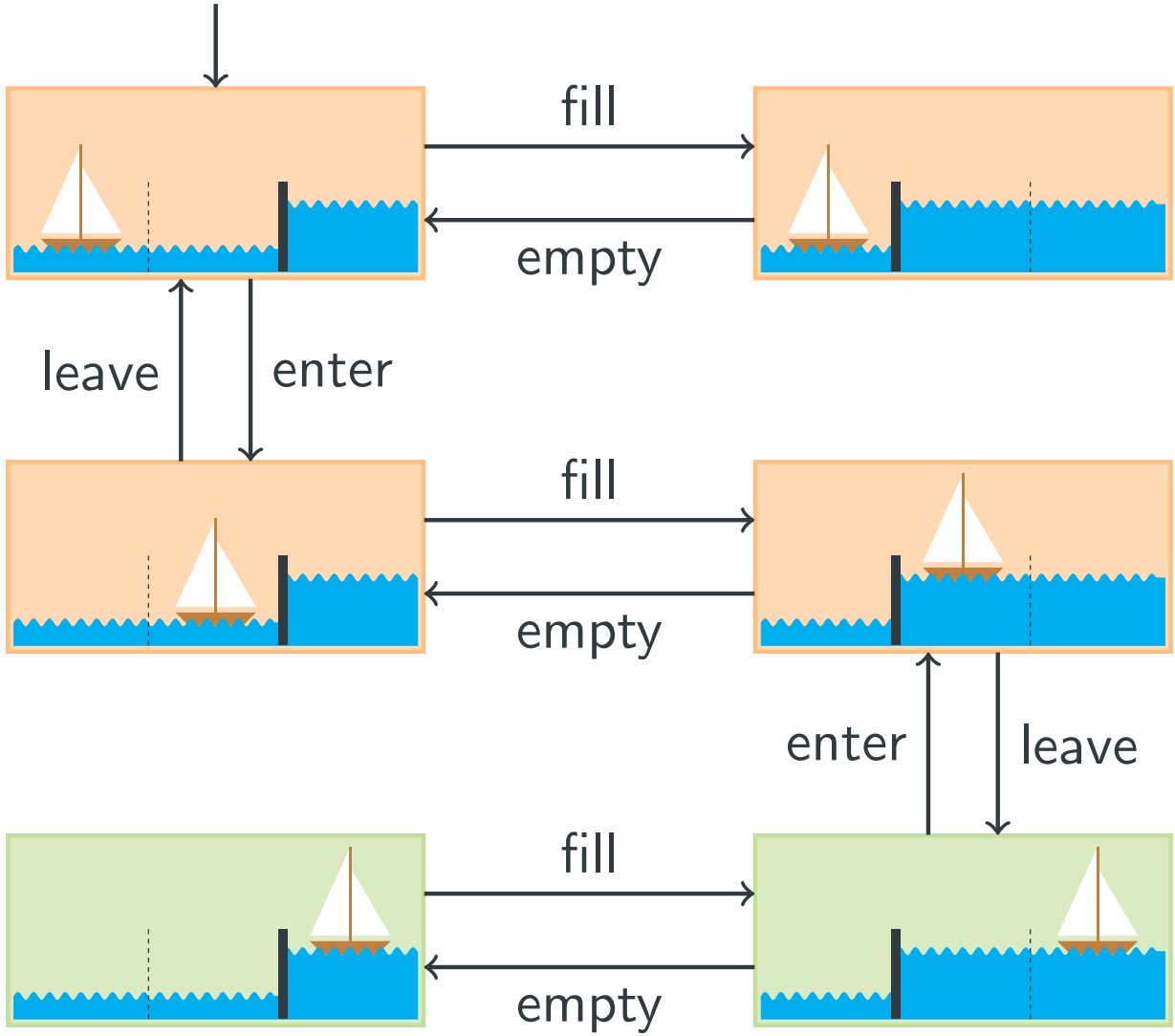


Abstraction Heuristics

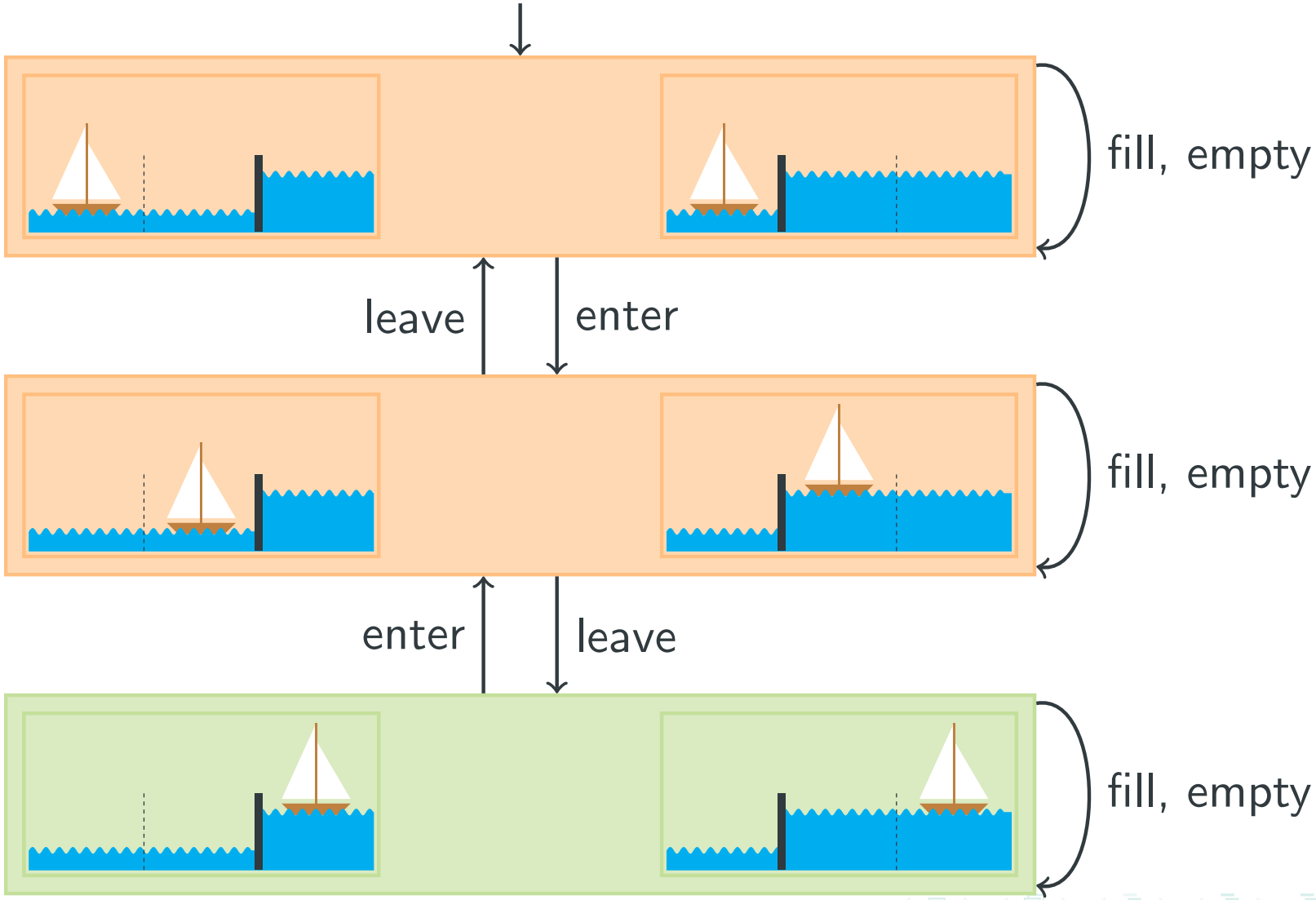
Abstractions

- disregard information on purpose
- combine states with similarities to abstract states
- always admissible $\rightarrow$ A* yields optimal plan

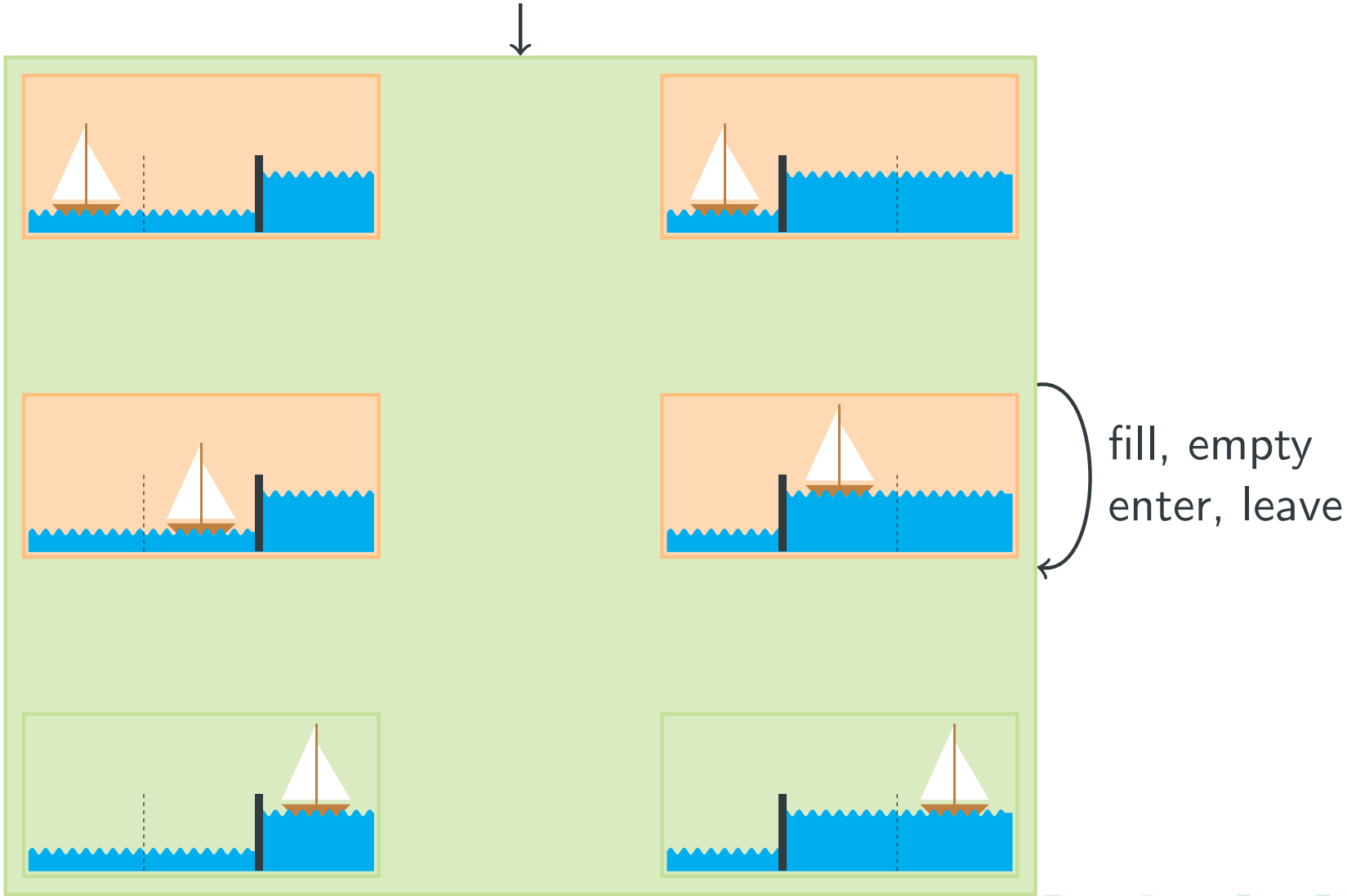
Example: Boat and Sluice



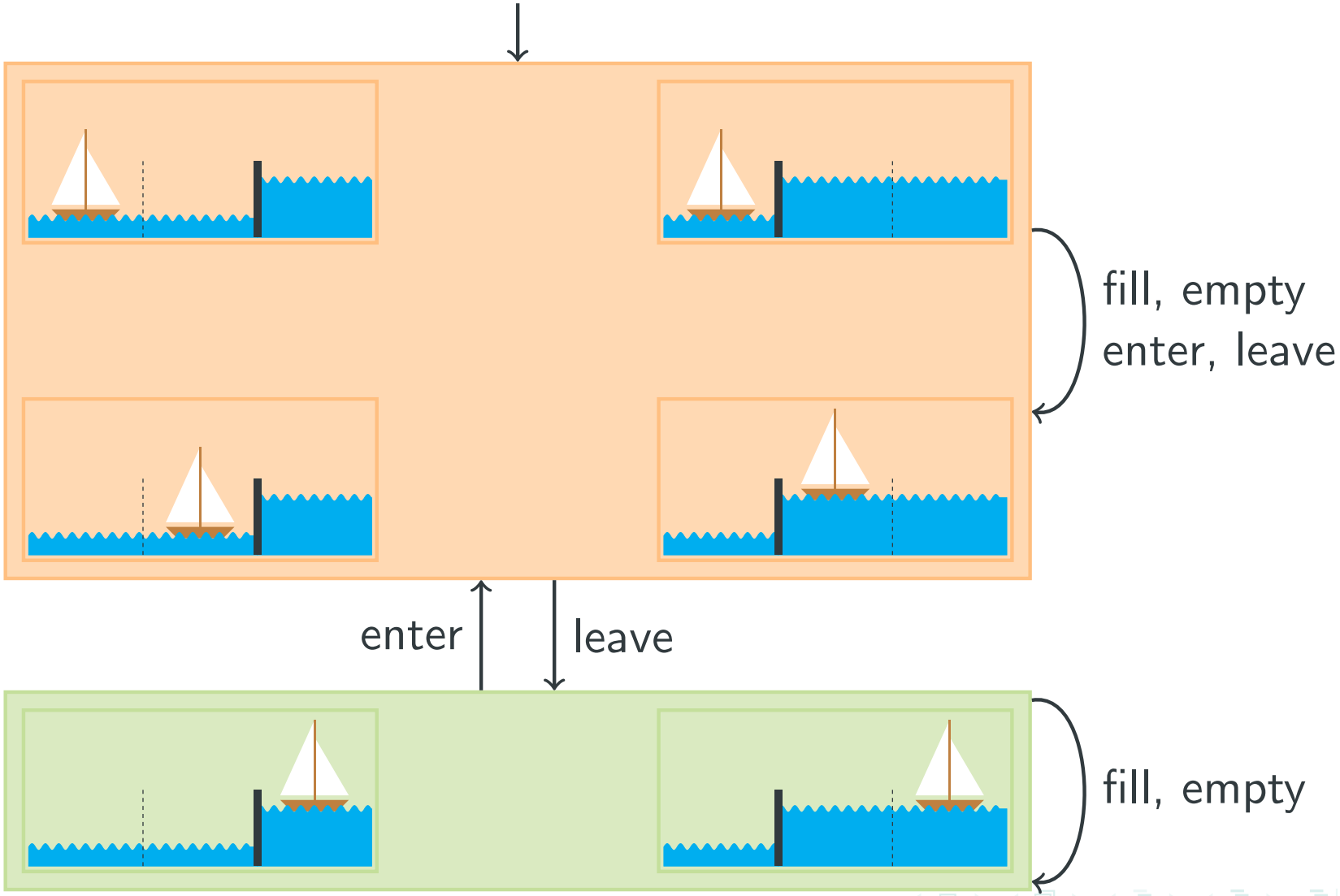
Projection



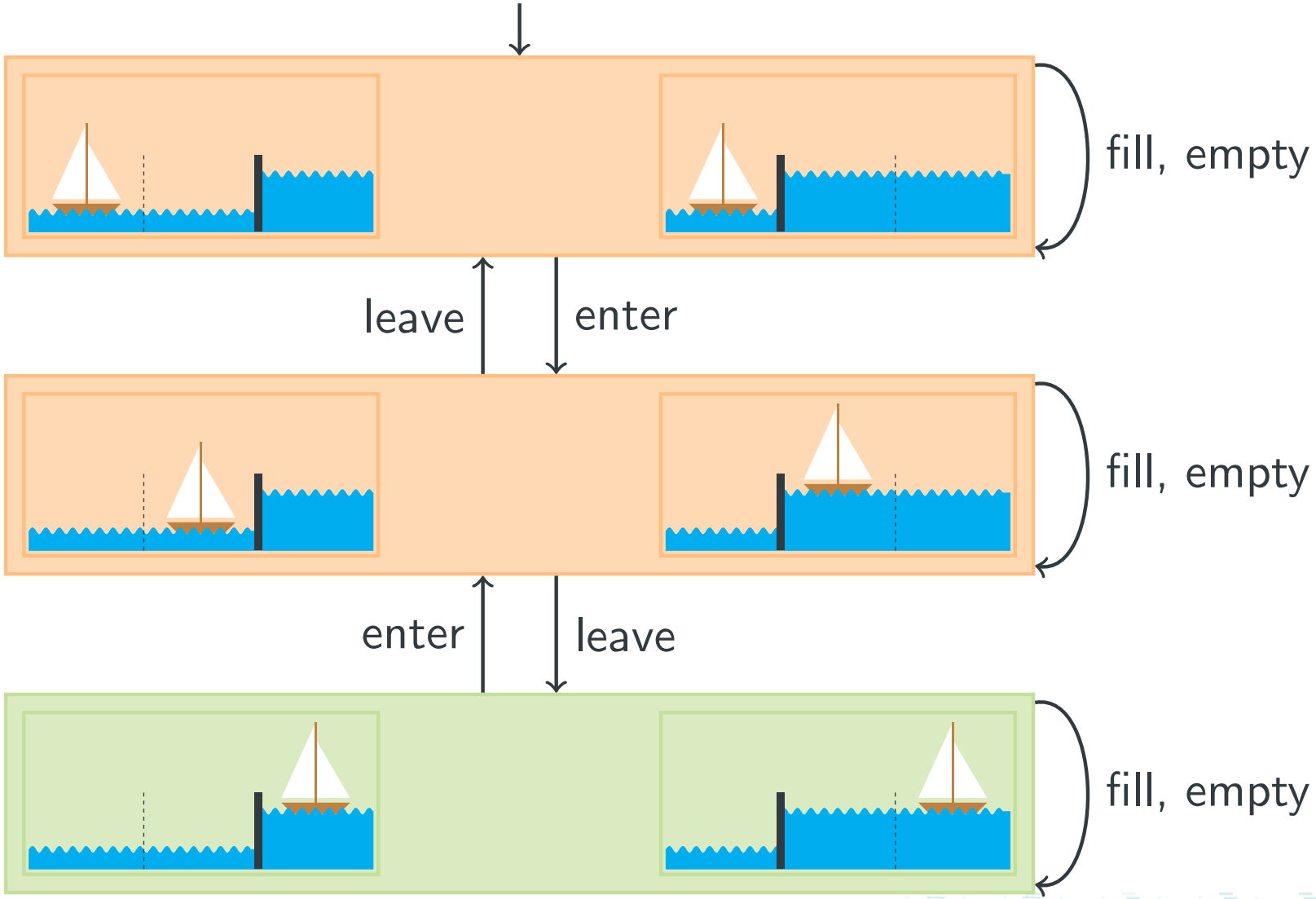
Cartesian Abstraction



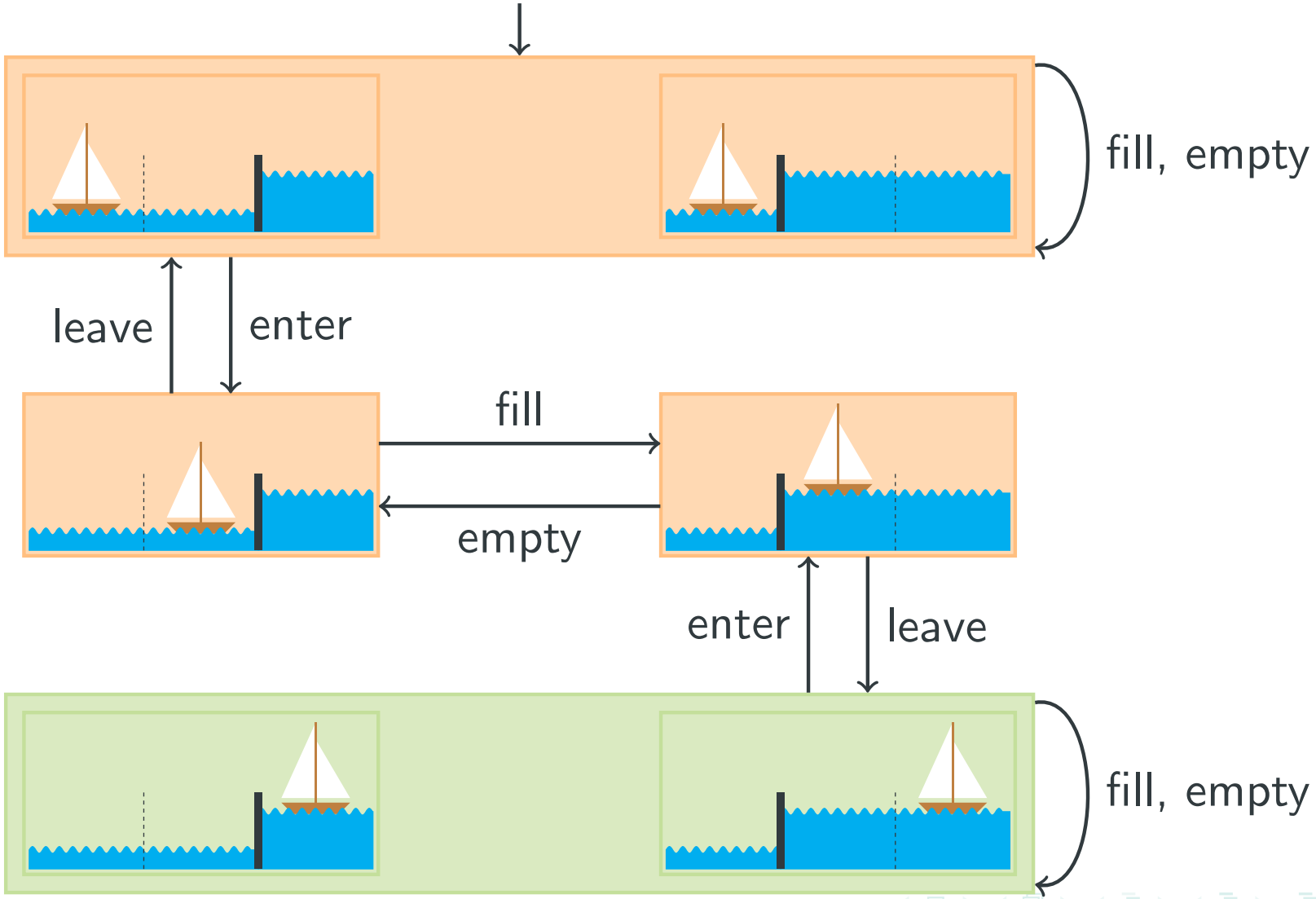
Cartesian Abstraction



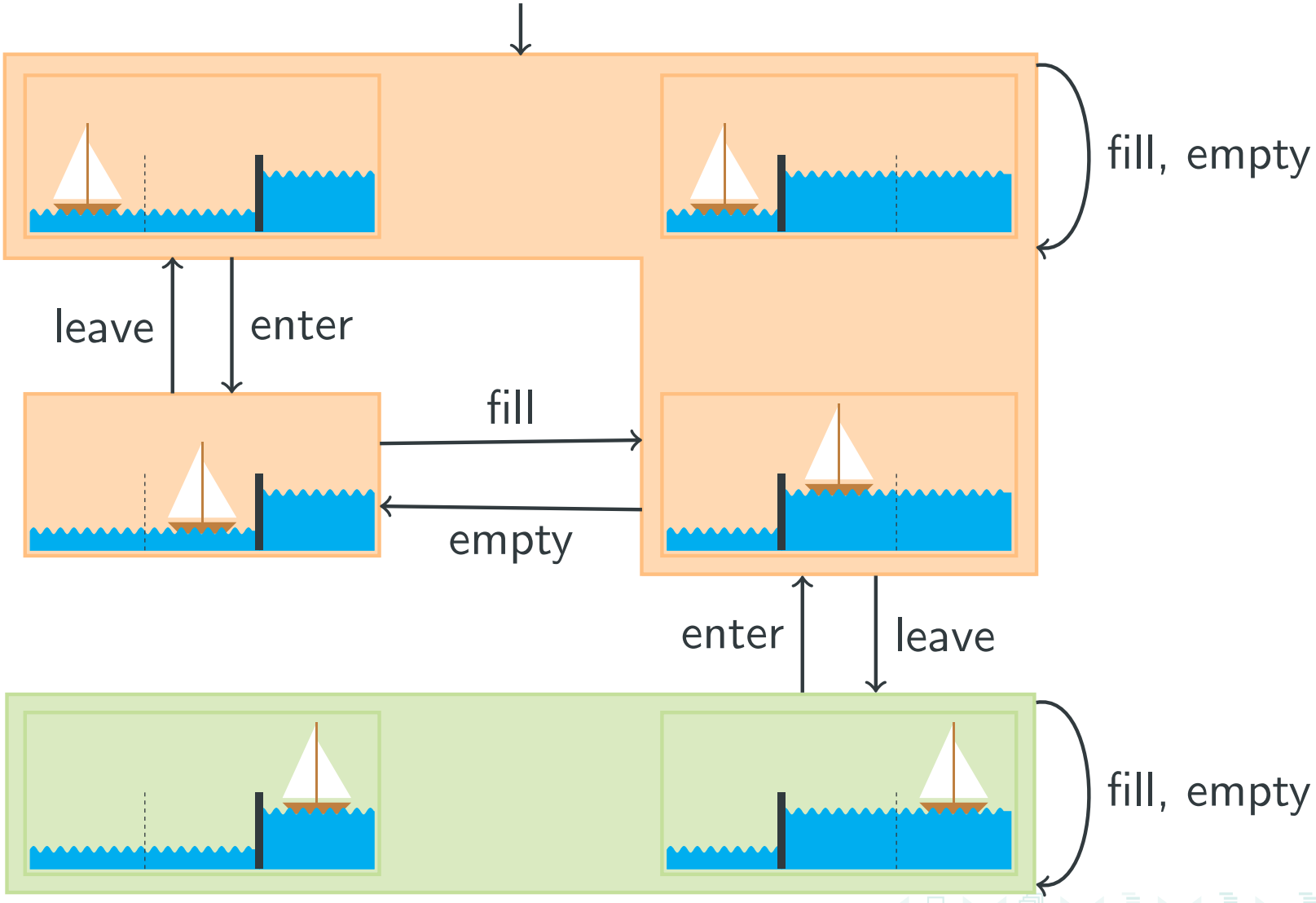
Cartesian Abstraction



Cartesian Abstraction



Merge-and-Shrink Abstraction



Cartesian Abstraction Refinement

Regression

- regression of Cartesian Set not Cartesian
- no support for conditional effects
- Rubik's Cube has only conditional effects

Factored Effect Task

Definition

Operator o is called a factored effect operator if $\mathit{eff}(o)$ has the following form:

$$
\begin{aligned}
X \mapsto x_1 \rhd X \mapsto x_2 \quad & \wedge \\
X \mapsto x_3 \rhd X \mapsto x_4 \quad & \wedge \\
\cdots \quad & \wedge \\
Z \mapsto z_1 \rhd Z \mapsto z_2 \quad &
\end{aligned}
$$

- factored effect tasks only have factored effect operators

Transition Check (before)

1: **function** CheckTransition(a, o, b)
2: **for all** $v \in \mathcal{V}$ **do**
3: **if** $v \in vars(pre(o))$ **and** $pre(o)[v] \notin dom(v, a)$ **then**
4: **return** false
5: **if** $v \in vars(post(o))$ **and** $post(o)[v] \notin dom(v, b)$ **then**
6: **return** false
7: **if** $v \notin vars(post(o))$ **and** $dom(v, a) \cap dom(v, b) = \emptyset$ **then**
8: **return** false
9: **return** true

Transition Check (before)

1: **function** CheckTransition(a, o, b)
2: **for all** $v \in \mathcal{V}$ **do**
3: **if** $v \in vars(pre(o))$ **and** $pre(o)[v] \notin dom(v, a)$ **then**
4: **return** false
5: **if** $v \in vars(\sout{post(o)})$ **and** $\sout{post(o)[v]} \notin dom(v, b)$ **then**
6: **return** false
7: **if** $v \notin vars(\sout{post(o)})$ **and** $dom(v, a) \cap dom(v, b) = \emptyset$ **then**
8: **return** false
9: **return** true

Functions

$$possible\,(a, o, X) = \bigcup_{x \in dom(X,a)} \{resulting\_fact\,(X \mapsto x, o)\}$$

Transition Check (after)

1: **function** CheckTransition(a, o, b)
2: **for all** $v \in \mathcal{V}$ **do**
3: **if** $v \in vars(pre(o))$ **and** $pre(o)[v] \notin dom(v, a)$ **then**
4: **return** false
5: **if** $v \in vars(o)$ **and** $possible(a, o, v) \cap dom(v, b) = \emptyset$ **then**
6: **return** false
7: **if** $v \notin vars(o)$ **and** $dom(v, a) \cap dom(v, b) = \emptyset$ **then**
8: **return** false
9: **return** true

Evaluation

Setup

- 200 problem instances of different difficulties
- seven heuristics
  - Blind Search Heuristic as a baseline
  - different Pattern Database heuristics
  - Cartesian Abstraction with conditional effects
  - Merge-and-Shrink

Coverage

Summary	blind	man	syst	corner	edge	cegar	m&s
coverage	68	128	122	108	105	111	95
out-of-memory[a]	132	72	0	92	95	89	105
timeout[b]	0	0	78	0	0	0	0
total time[c]	0.32	16.98	141.36	2.64	2.21	0.61	19.94

[a] memory limit: 3.5 GiB
[b] time limit: 30 minutes
[c] geometric mean of instances solved by all configurations

Coverage

Summary	blind	man	syst	corner	edge	cegar	m&s
coverage	68	**128**	122	108	105	111	95
out-of-memory[a]	132	72	0	92	95	89	105
timeout[b]	0	0	78	0	0	0	0
total time[c]	0.32	16.98	141.36	2.64	2.21	0.61	19.94

[a] memory limit: 3.5 GiB
[b] time limit: 30 minutes
[c] geometric mean of instances solved by all configurations

Coverage

Summary	blind	man	**syst**	corner	edge	cegar	m&s
coverage	68	128	122	108	105	111	95
out-of-memory[a]	132	72	**0**	92	95	89	105
timeout[b]	0	0	**78**	0	0	0	0
total time[c]	0.32	16.98	**141.36**	2.64	2.21	0.61	19.94

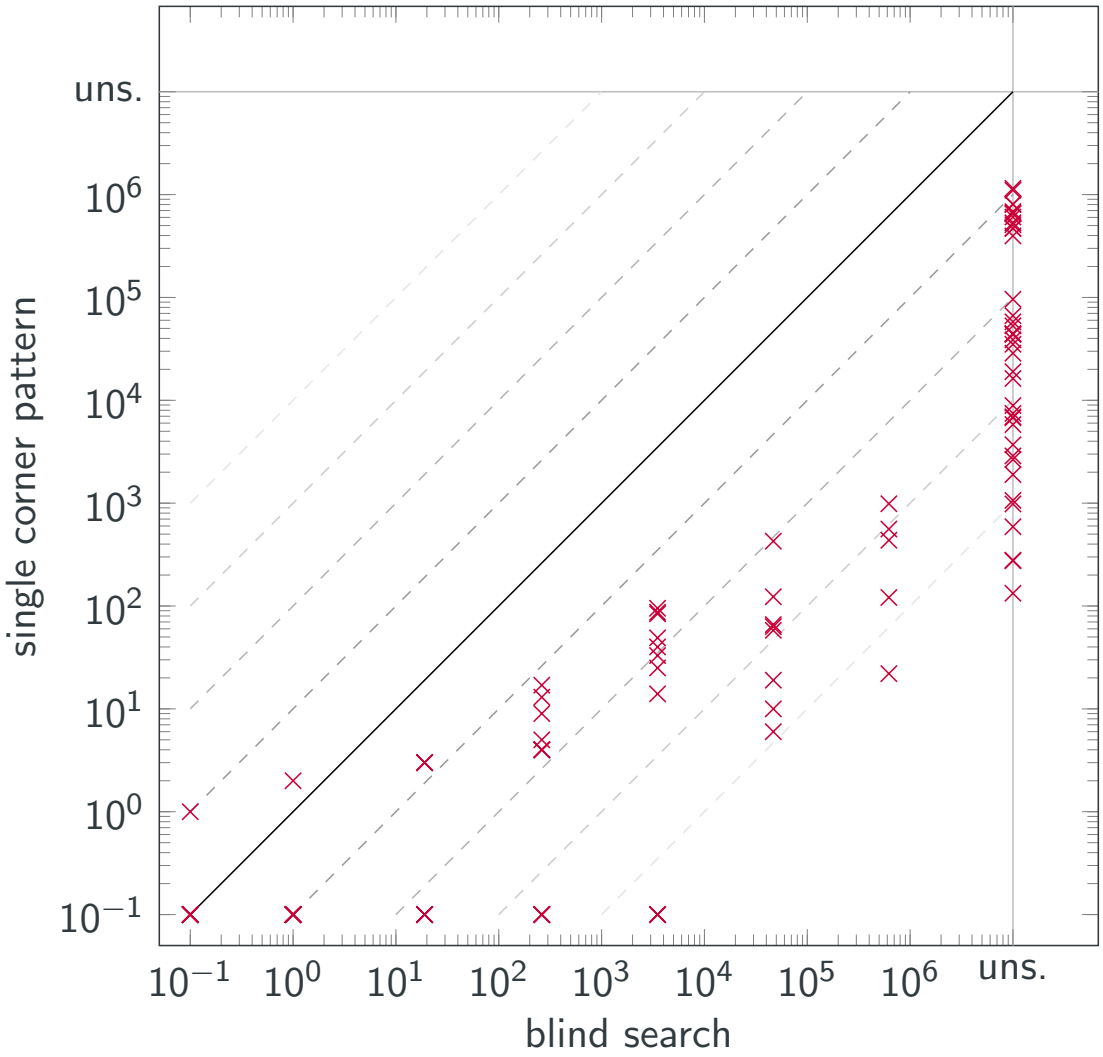
[a] memory limit: 3.5 GiB
[b] time limit: 30 minutes
[c] geometric mean of instances solved by all configurations

Coverage

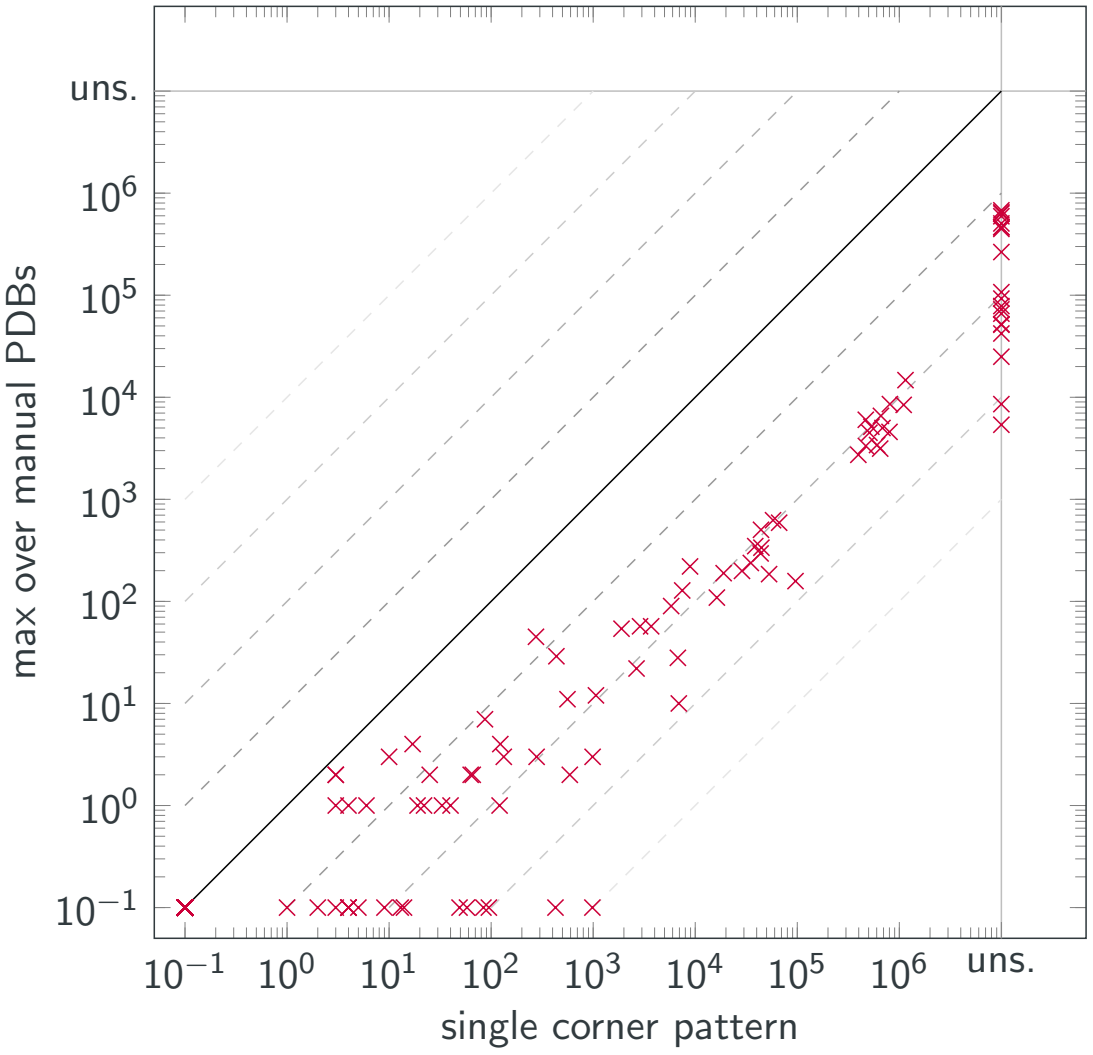
Summary	blind	man	syst	corner	edge	**cegar**	m&s
coverage	68	128	122	108	105	111	95
out-of-memory[a]	132	72	0	92	95	89	105
timeout[b]	0	0	78	0	0	0	0
total time[c]	0.32	16.98	141.36	2.64	2.21	**0.61**	19.94

[a] memory limit: 3.5 GiB
[b] time limit: 30 minutes
[c] geometric mean of instances solved by all configurations

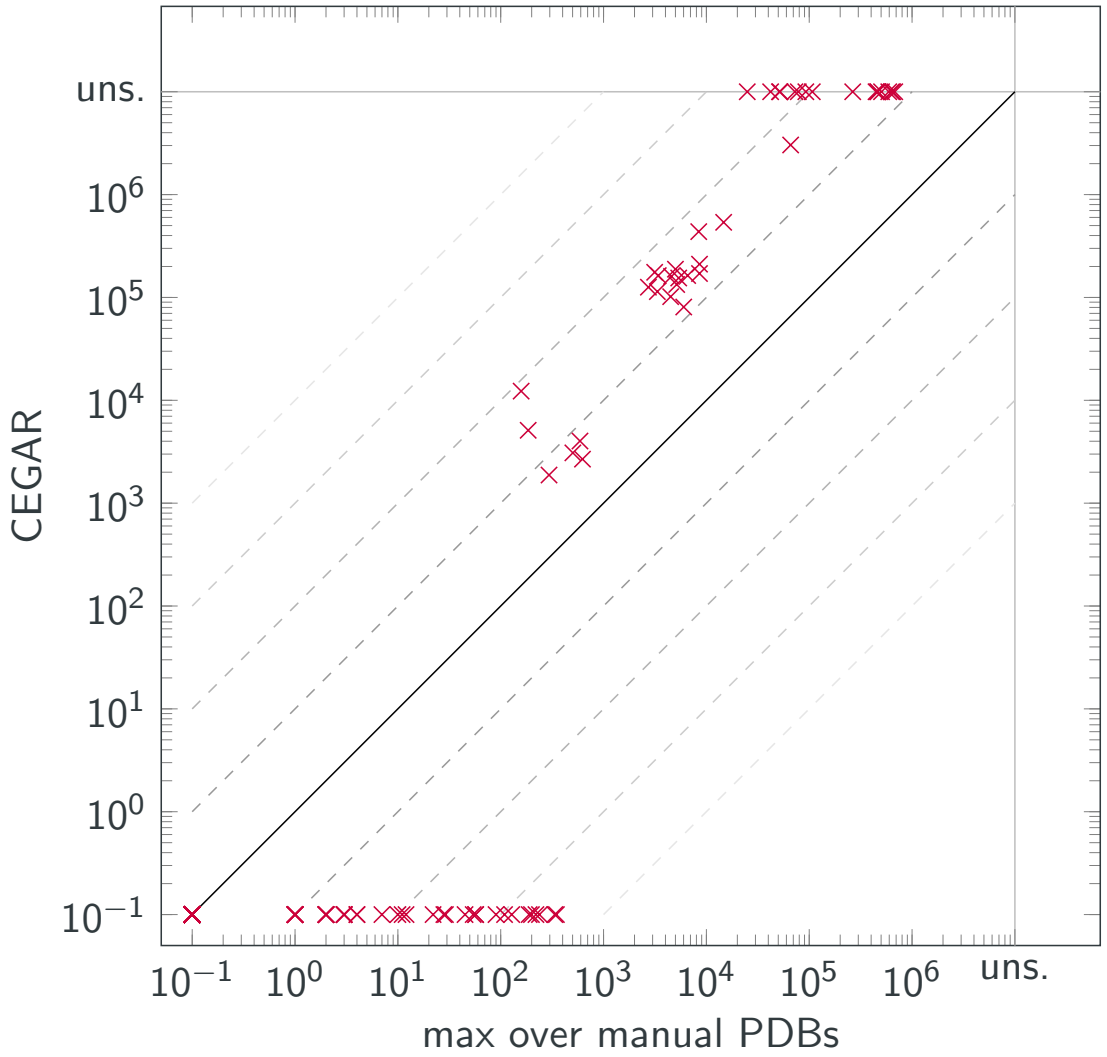
Expansions Until the Last f-layer



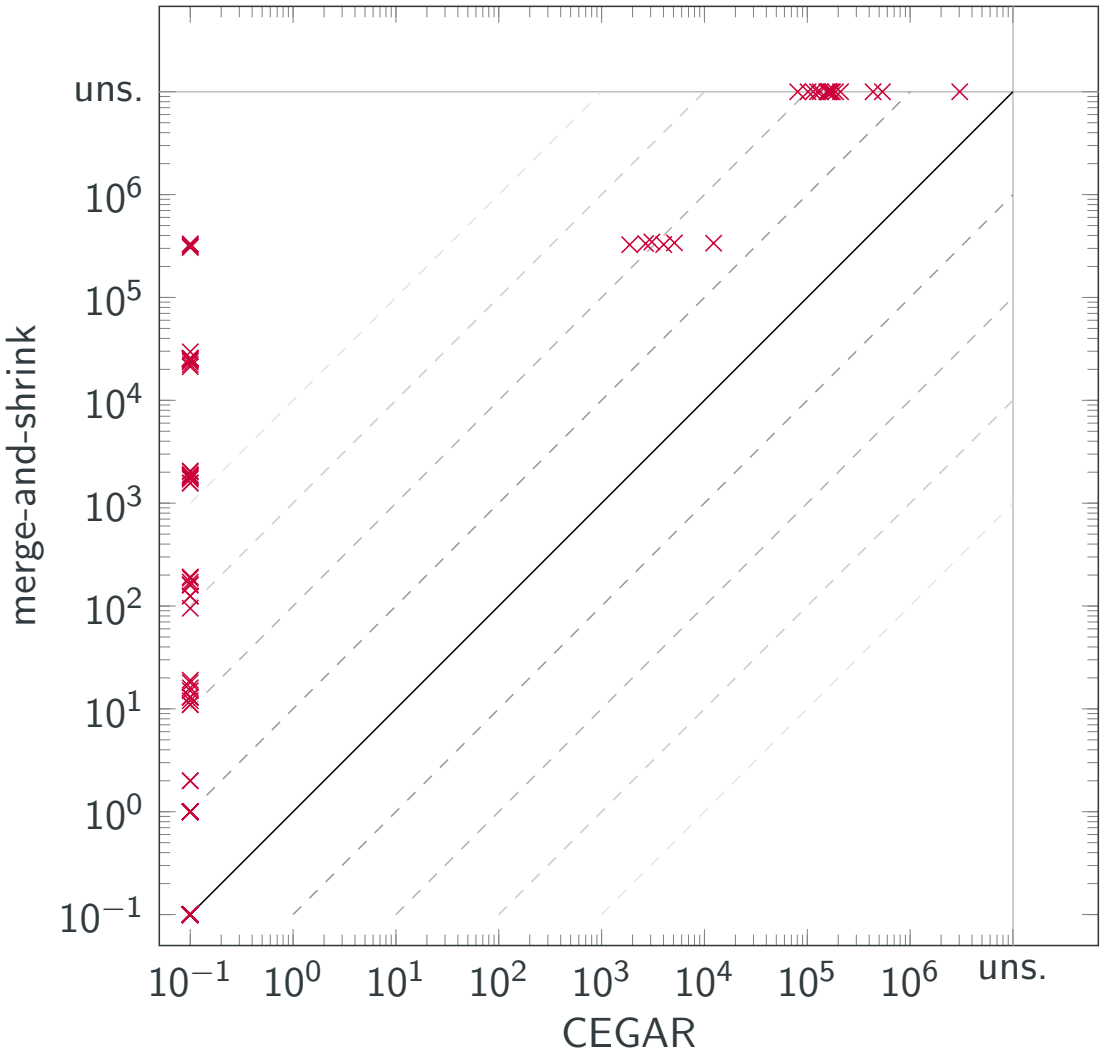
Expansions Until the Last f-layer



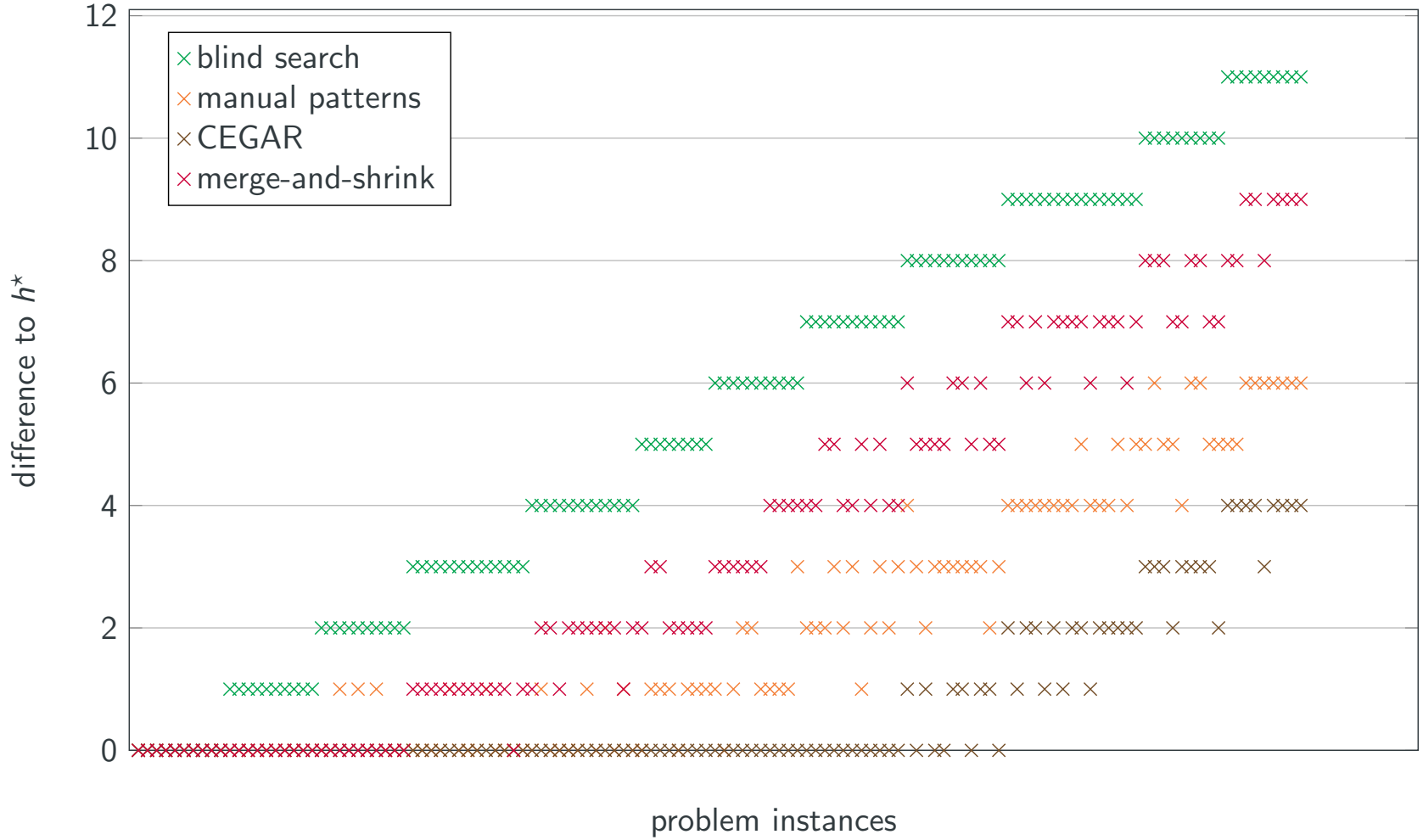
Expansions Until the Last f-layer



Expansions Until the Last f-layer



Initial h-values



Summary

- factored effect tasks can be abstracted with CEGAR
- PDBs find most solutions with least expansions
- CEGAR is fast and yields perfect heuristic values

Thank you for your attention.

Planning Tasks

Definition

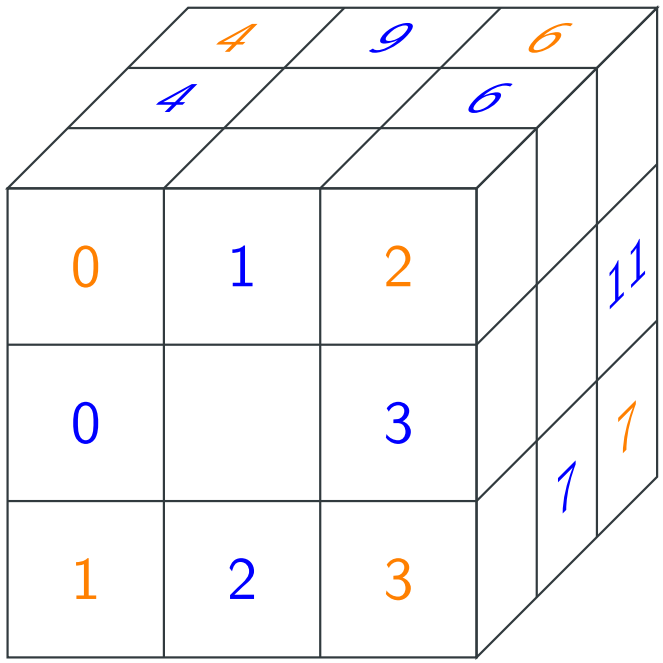
A planning task is a 4-tuple $\Pi = \langle \mathcal{V}, I, \mathcal{O}, \gamma \rangle$ where

- $\mathcal{V}$ is a finite set of finite-domain state variables,
- I is a state over $\mathcal{V}$ called the initial state,
- $\mathcal{O}$ is a finite set of finite-domain operators over $\mathcal{V}$, and
- γ is a formula over $\mathcal{V}$ called the goal.

- operators $o \in \mathcal{O}$ with precondition $pre(o)$, effect $eff(o)$, and cost $cost(o)$

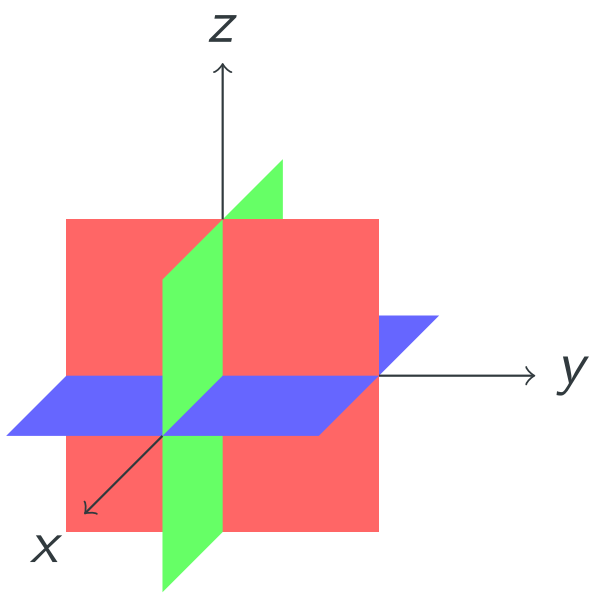
Locations

- coordinate system
- distinguish categories
  - corner cubies and edge cubies



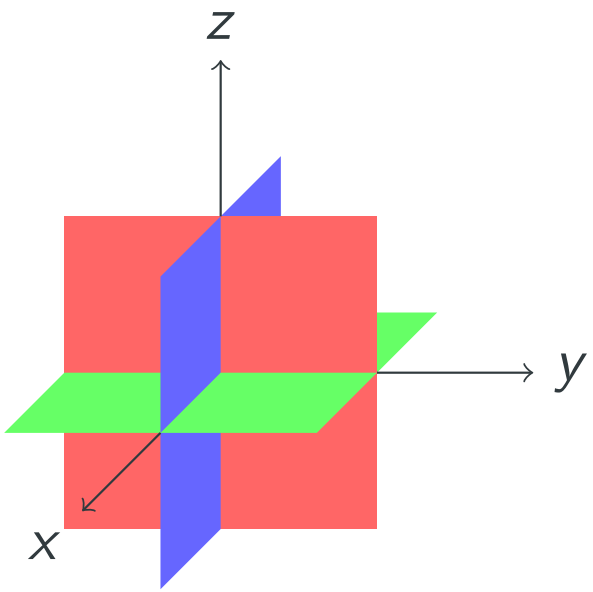
Rotations

- number of facelets corresponds to possible rotations
- triple representation $\langle x, y, z \rangle$
  - use blanks (#) where less than 3 facelets
  - rotation is first non-blank triple element



Rotations

- number of facelets corresponds to possible rotations
- triple representation $\langle x, y, z \rangle$
  - use blanks (#) where less than 3 facelets
  - rotation is first non-blank triple element



Regression (1)

Definition

Let $X \mapsto x_1$ be an atomic effect and let o be an operator in a (general) planning task. Then the regression of X through o is defined as follows:

$$\begin{aligned} \mathit{regr}\,(X \mapsto x_1, \mathit{eff}\,(o)) \quad = \quad & \mathit{pre}\,(o)\,[X] \wedge (\mathit{effcond}\,(X \mapsto x_1, \mathit{eff}\,(o)) \vee \\ & (X \mapsto x_1 \wedge \neg \mathit{effcond}\,(X \neq x_1, \mathit{eff}\,(o)))) \end{aligned}$$

Regression (2)

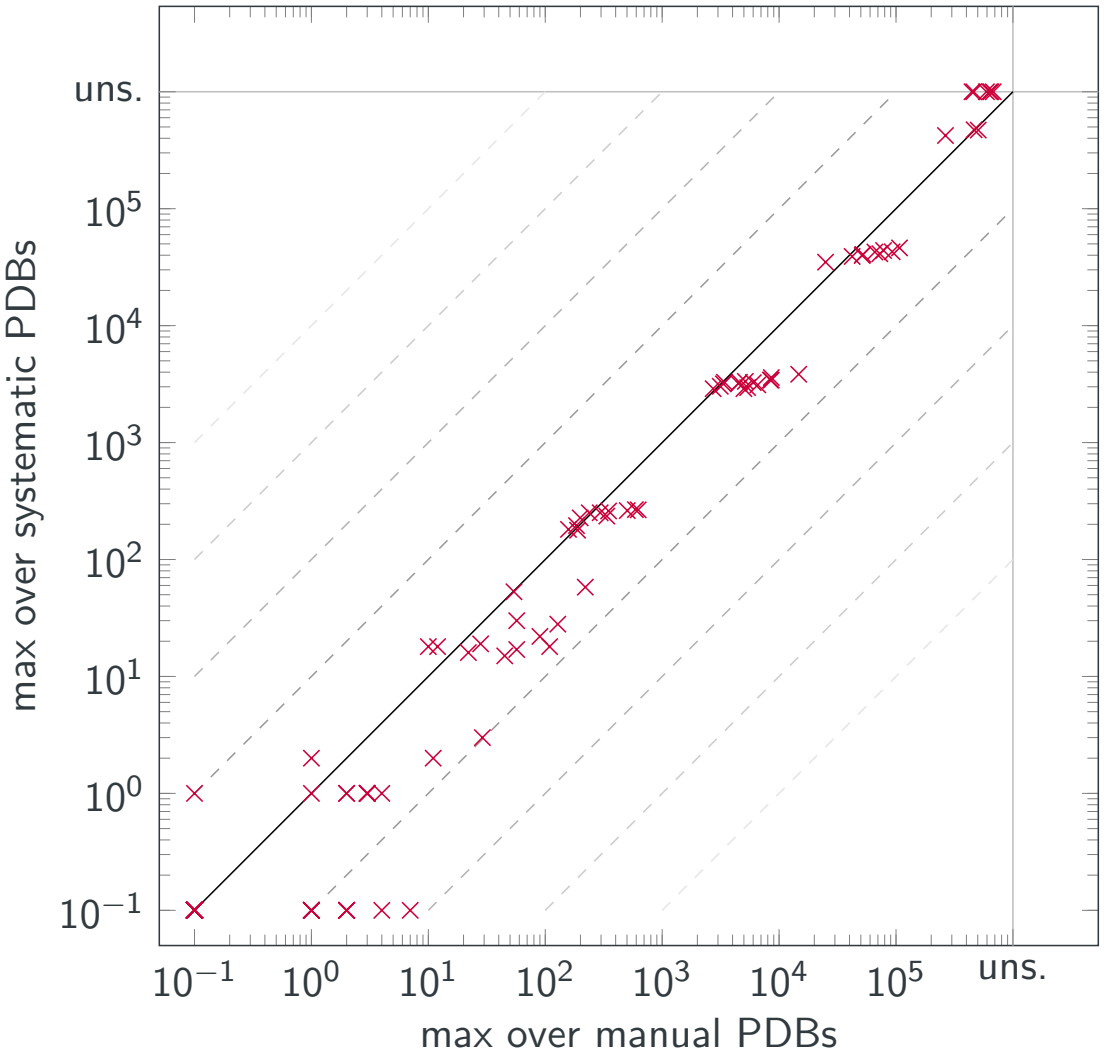
$$regr\,(X \mapsto x_2, o) \quad = \quad pre\,(o)\,[X] \wedge \left(\bigvee_{\substack{\langle X \mapsto x_1, X \mapsto x_2 \rangle \in \\ effects(o)}} X \mapsto x_1 \vee \right.$$

$$\left\{ \begin{array}{ll} X \mapsto x_2 & \text{if } X \mapsto x_2 \text{ does not occur as an effect condition in } o \\ \bot & \text{otherwise} \end{array} \right)$$

$$regr\,(\mathcal{X}, o) = \bigcup_{x \in \mathcal{X}} regr\,(X \mapsto x, o)$$

$$regr\,(a, o) = A_1 \times \cdots \times A_n \text{ where } A_i = regr\,(dom(v_i, a), o)$$

Expansions Until the Last f-layer



Expansions Until the Last f-layer