# Latest Trends in Abstraction Heuristics for Classical Planning

## 2. Cartesian Abstractions

Malte Helmert  Jendrik Seipp  Silvan Sievers

ICAPS 2015 Tutorial

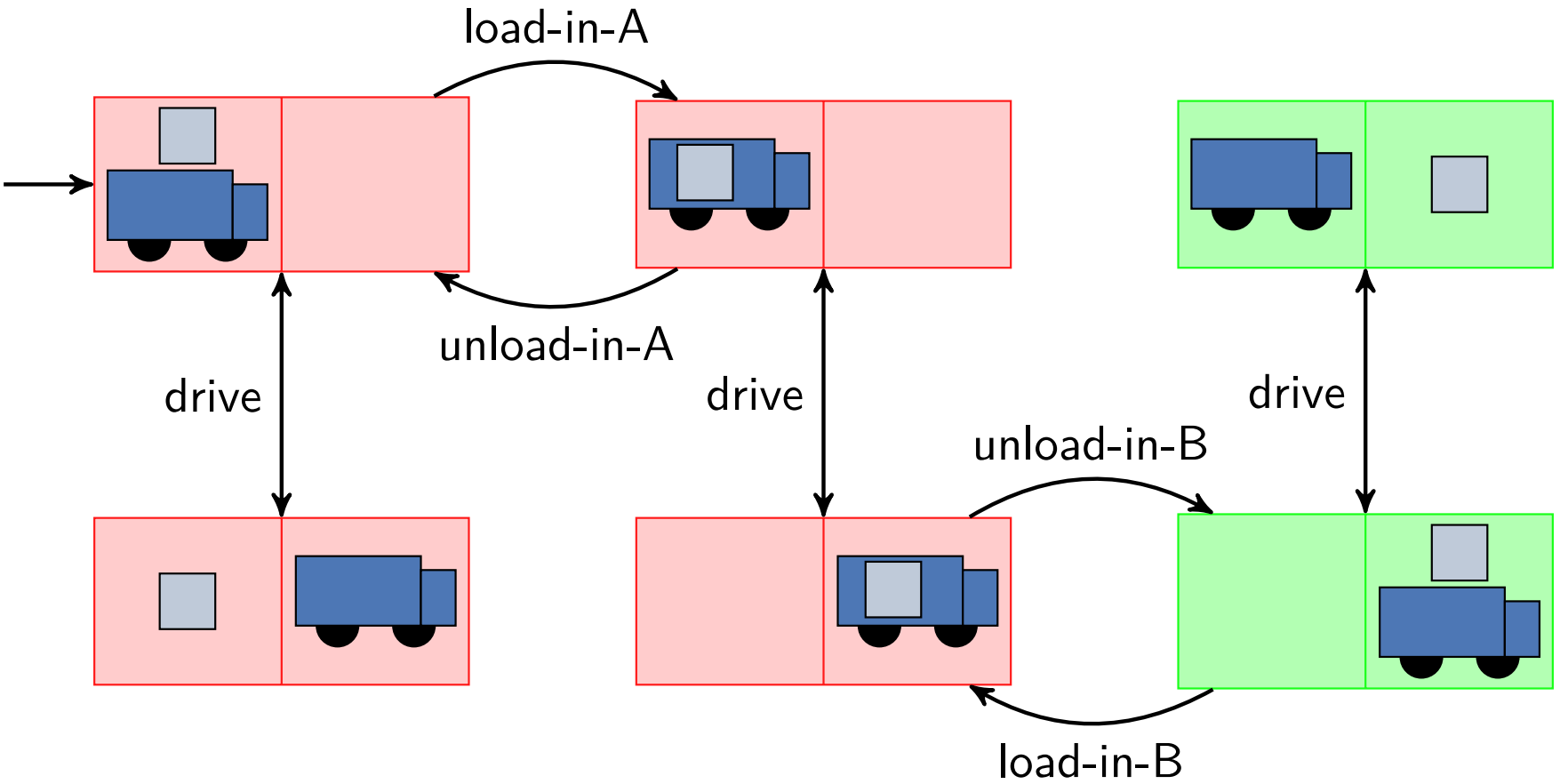
June 7, 2015

# Overview

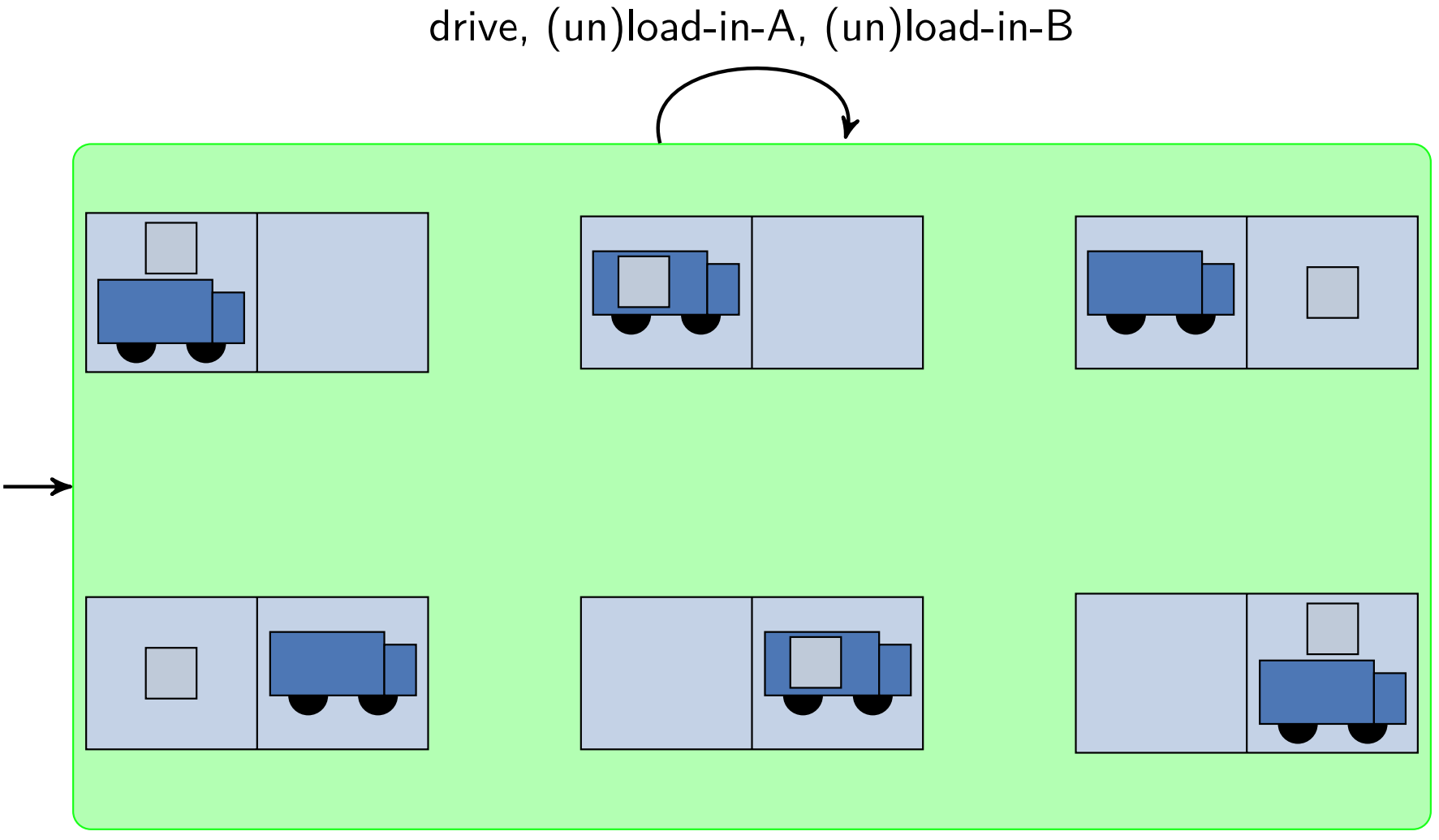
- Cartesian Abstraction Refinement
- Additive Abstractions
- Diversification Strategies

# Cartesian Abstraction Refinement

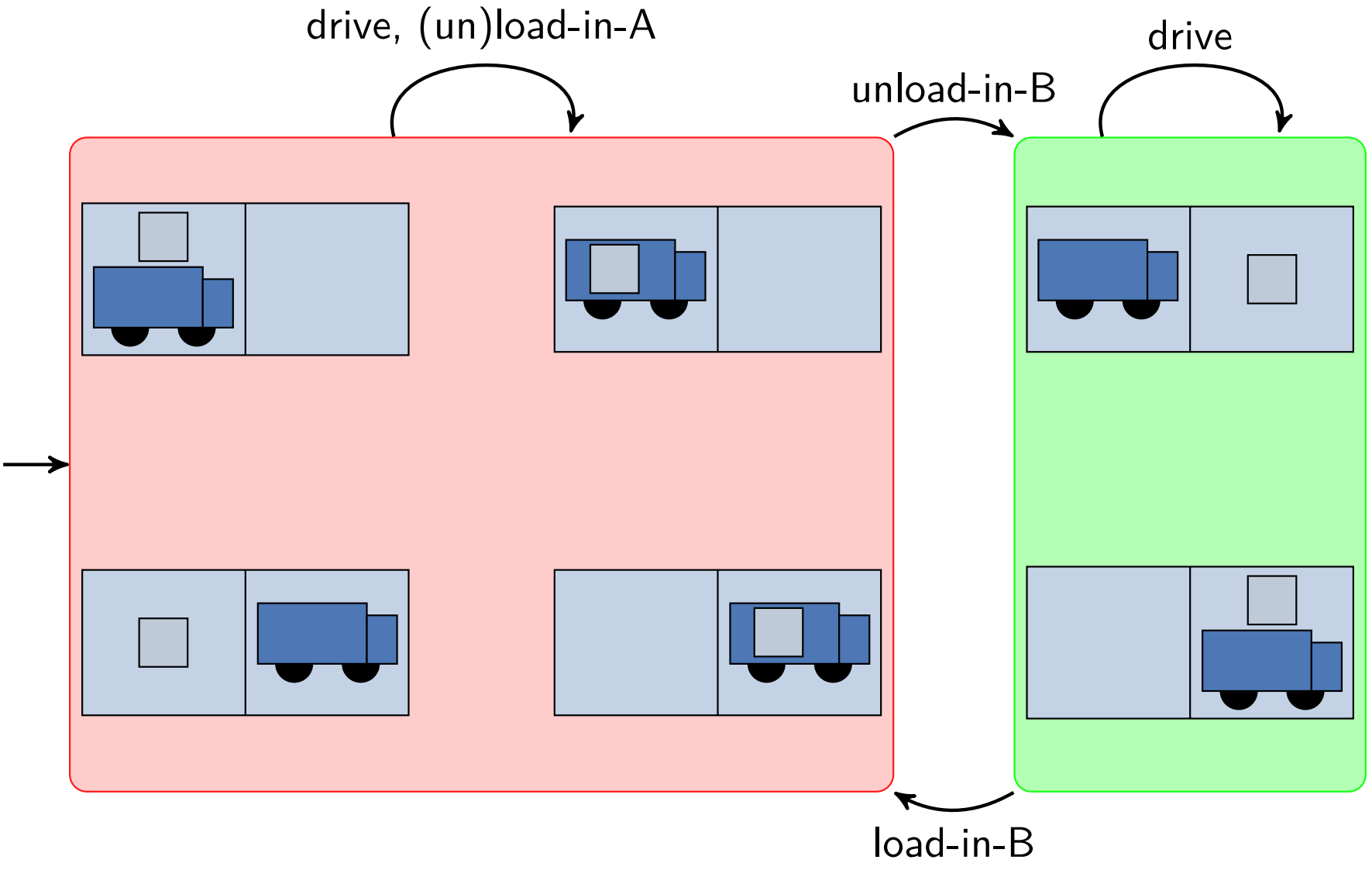
# Example Cartesian Abstraction Refinement



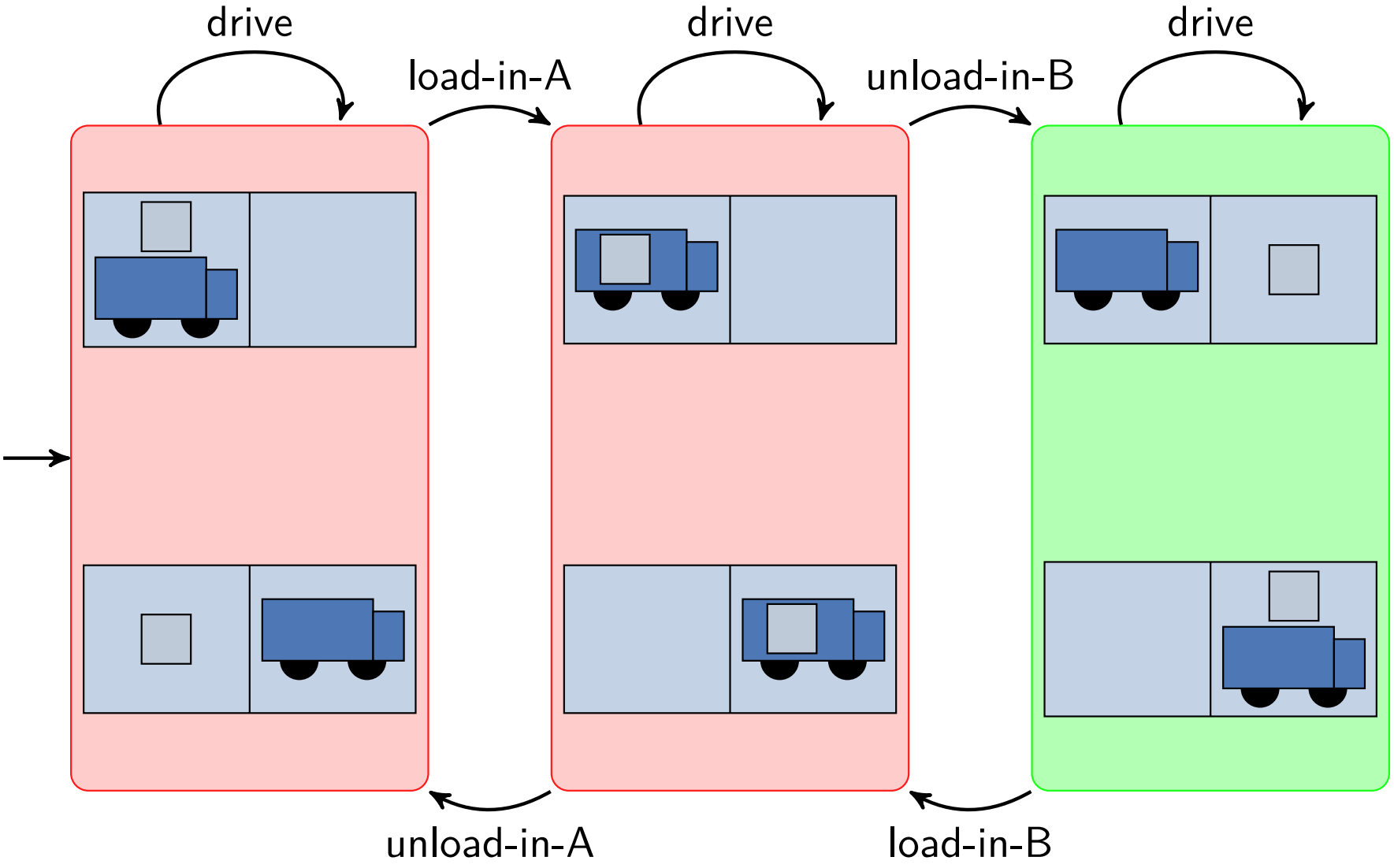
# Example Cartesian Abstraction Refinement



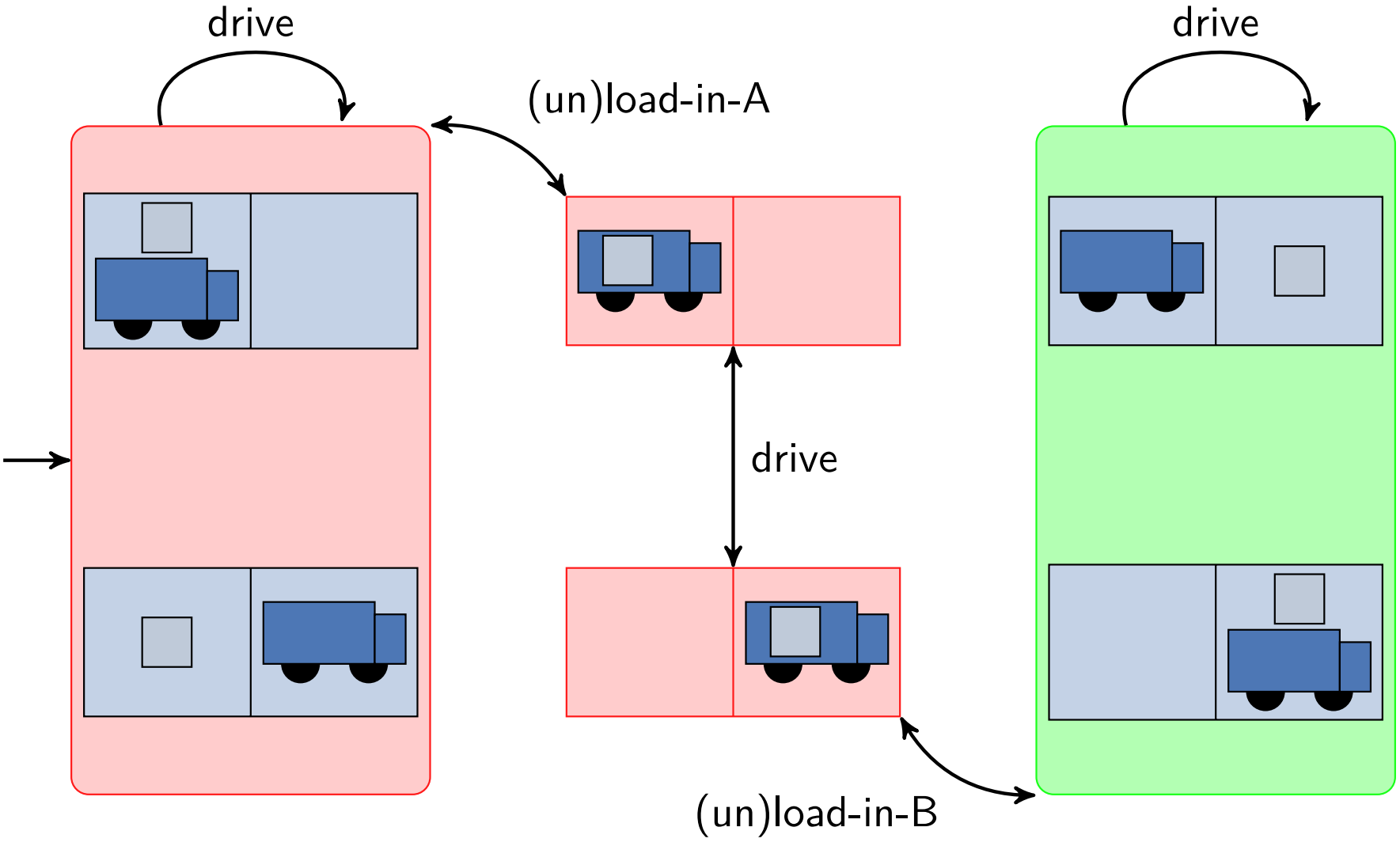
# Example Cartesian Abstraction Refinement



# Example Cartesian Abstraction Refinement



# Example Cartesian Abstraction Refinement



# Counterexample-guided Abstraction Refinement (CEGAR)

## CEGAR algorithm

Start with coarsest abstraction

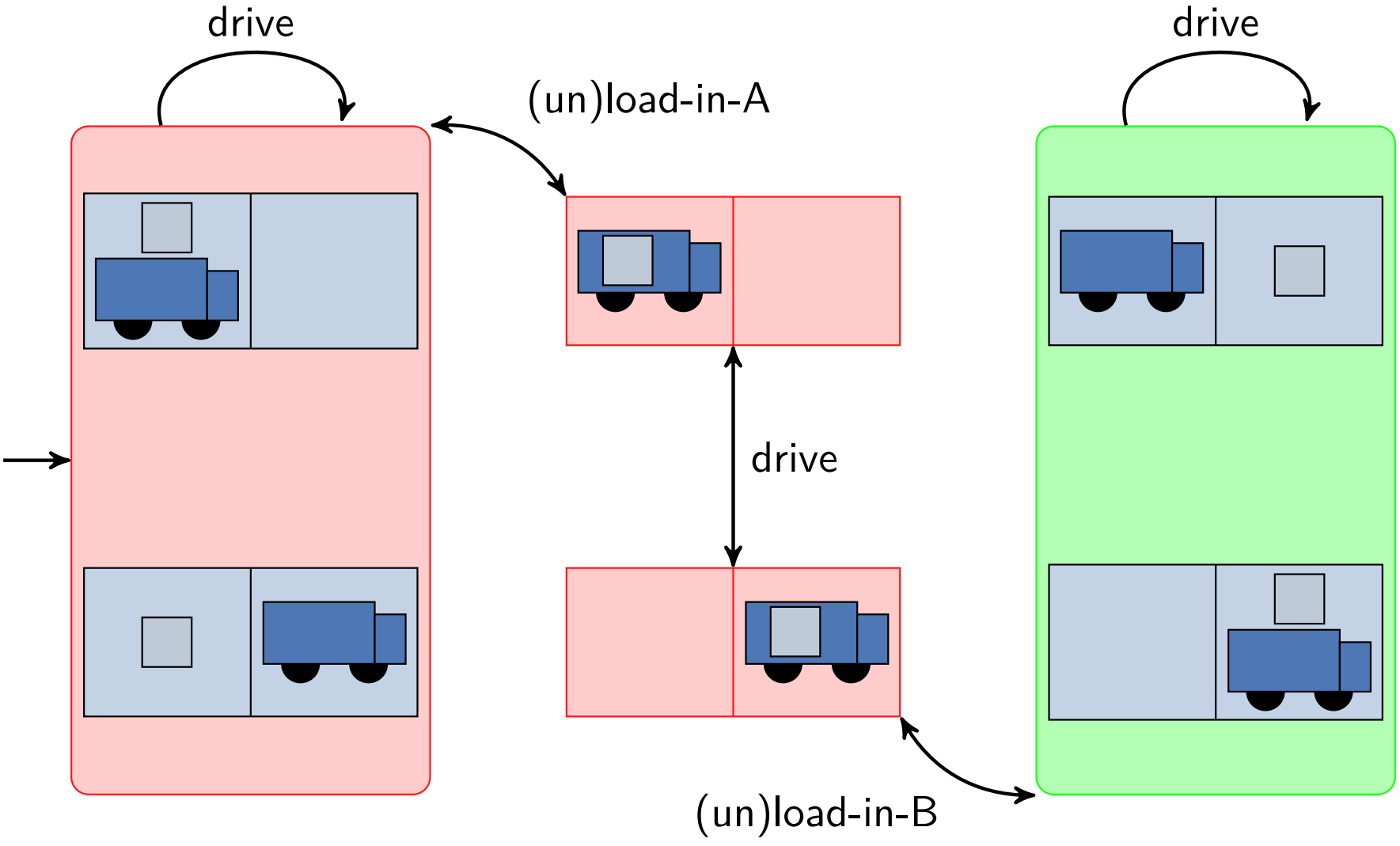
Until concrete solution is found or time runs out:

- Find abstract solution
- Check if and why it fails in the real world
- Refine abstraction

# Cartesian Abstractions

A set of states is called Cartesian if it is of the form $A_1 \times A_2 \times ... \times A_n$, where $A_i \subseteq dom(v_i)$ for all $v_i \in V$.
An abstraction is called Cartesian if all its abstract states are Cartesian sets.

# Example Cartesian Abstraction Refinement

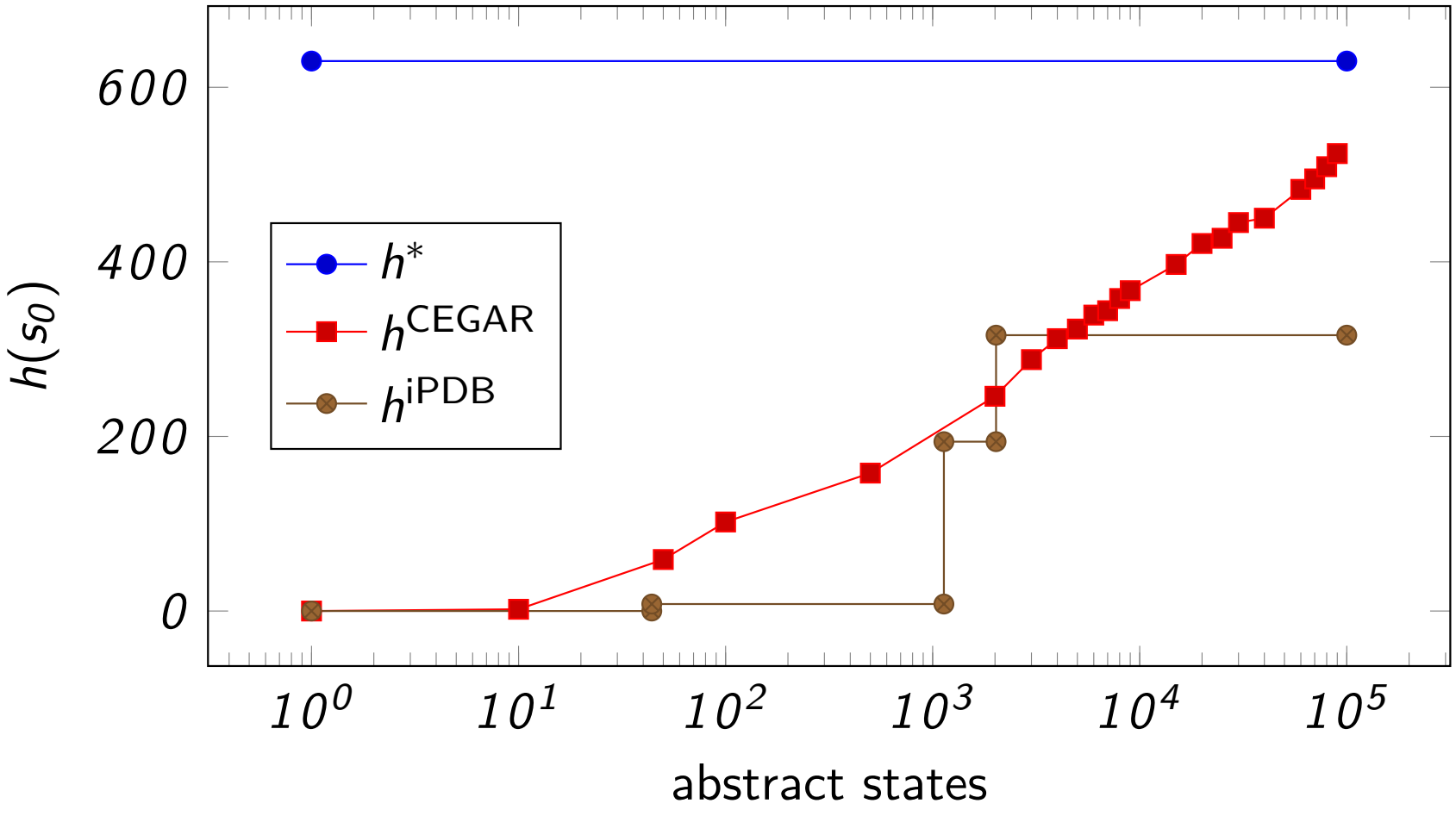


# Classes of Abstractions
## Suitability for CEGAR

- Pattern databases
  Refinement at least doubles number of states
- Domain abstractions
  Don't allow fine-grained refinement
- Cartesian abstractions
  Perform refinement operations quickly
- Merge-and-shrink abstractions
  Preimage of abstract states not efficiently computable

# Evolution of $h(s_0)$

## Transport #23



# Drawbacks of Single Cartesian Abstractions

- Diminishing returns
- Goal facts are considered one after another
- Abstraction more refined in regions around abstract solutions

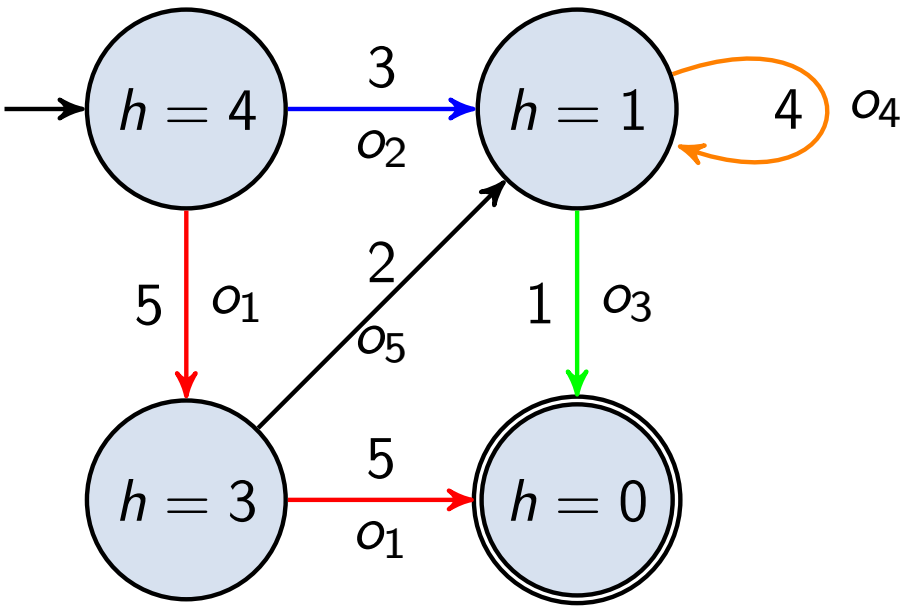
# Drawbacks of Single Cartesian Abstractions

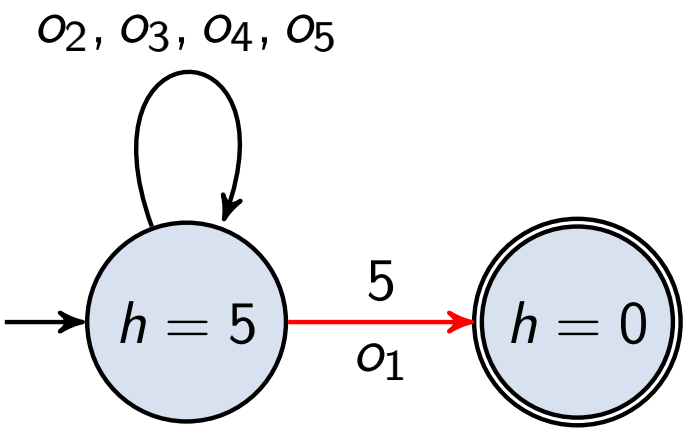
- Diminishing returns
- Goal facts are considered one after another
- Abstraction more refined in regions around abstract solutions

$\rightarrow$ Multiple abstractions

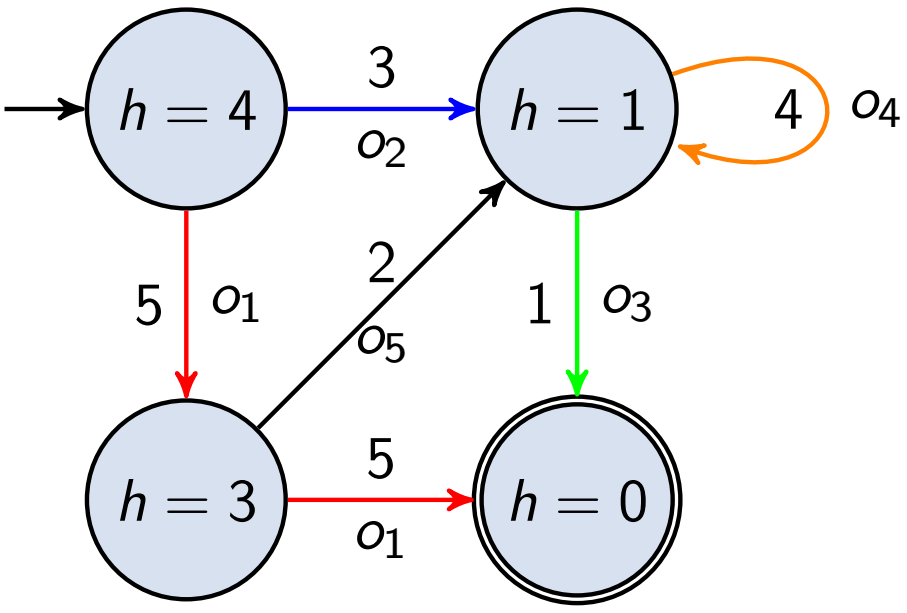
# Additive Abstractions

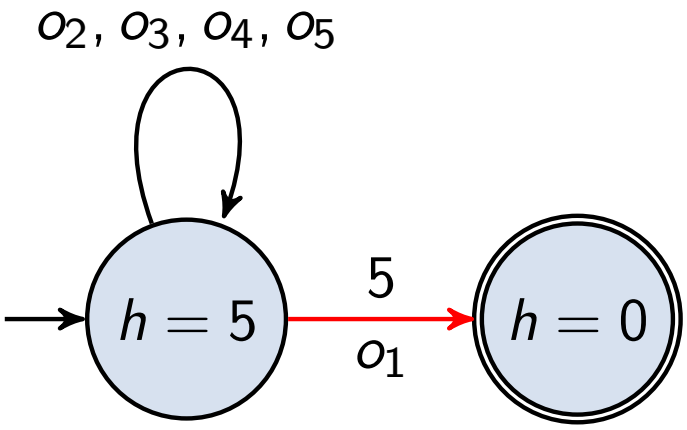
# Multiple Abstractions



# Multiple Abstractions




How to combine heuristic estimates?

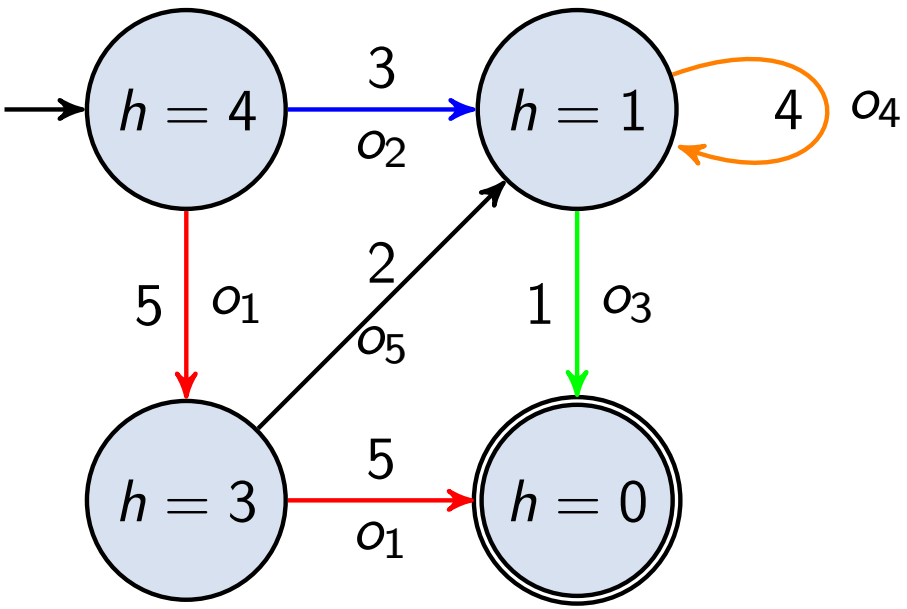
- Maximum: $h(s_0) = \max(4, 5) = 5$

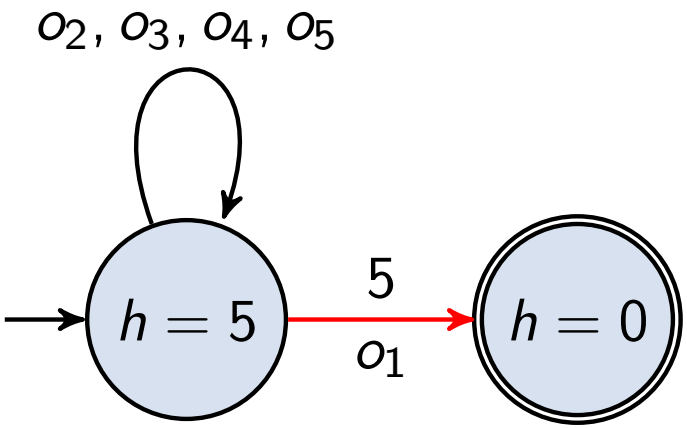
# Cost Partitioning

A cost partitioning for a planning task with operator set $O$ and cost function $c$ is a sequence $c_1, \ldots, c_n$ of cost functions $c_i : O \to \mathbb{R}$ that assign costs to operators $o \in O$ such that $\sum_{1 \leq i \leq n} c_i(o) \leq c(o)$ for all $o \in O$.

# Cost Partitioning

A cost partitioning for a planning task with operator set $O$ and cost function $c$ is a sequence $c_1, \ldots, c_n$ of cost functions $c_i : O \to \mathbb{R}$ that assign costs to operators $o \in O$ such that $\sum_{1 \leq i \leq n} c_i(o) \leq c(o)$ for all $o \in O$.



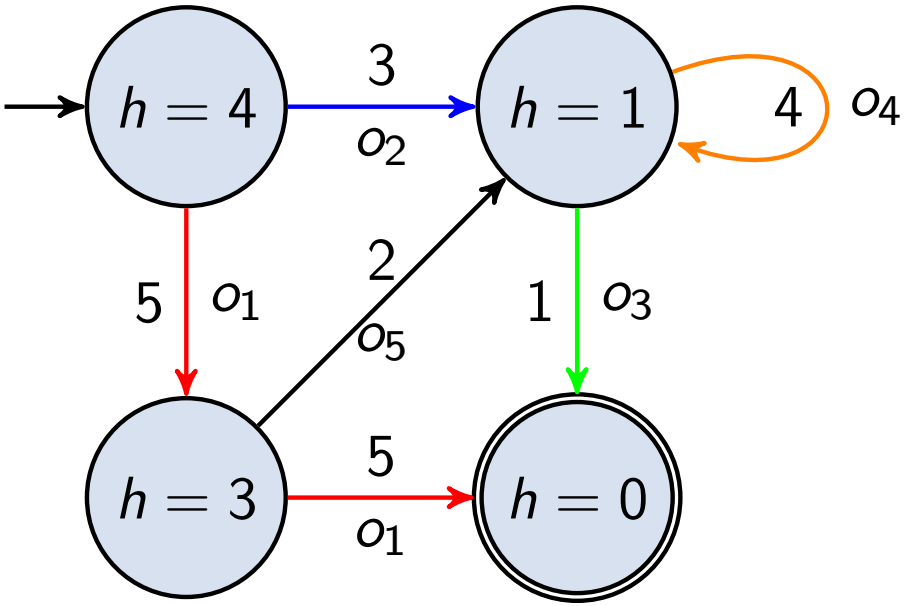



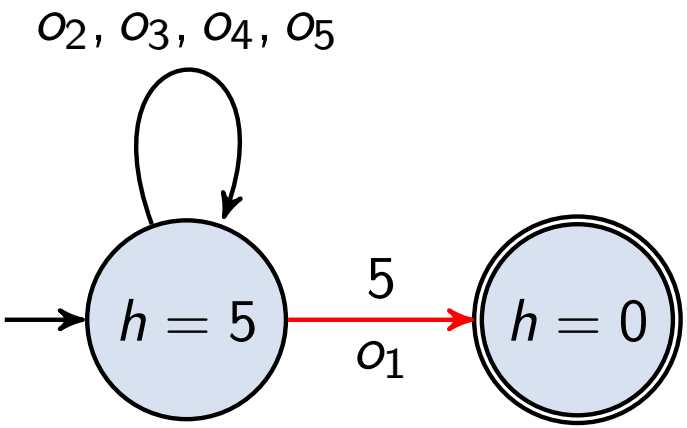

- Cost partitioning: $h(s_0) = 0 + 5 = 5$

# Saturated Cost Partitioning

## Saturated cost function

$$\hat{c}(o) = \max_{a \xrightarrow{o} b \in T} \max\{0, h(a) - h(b)\}$$

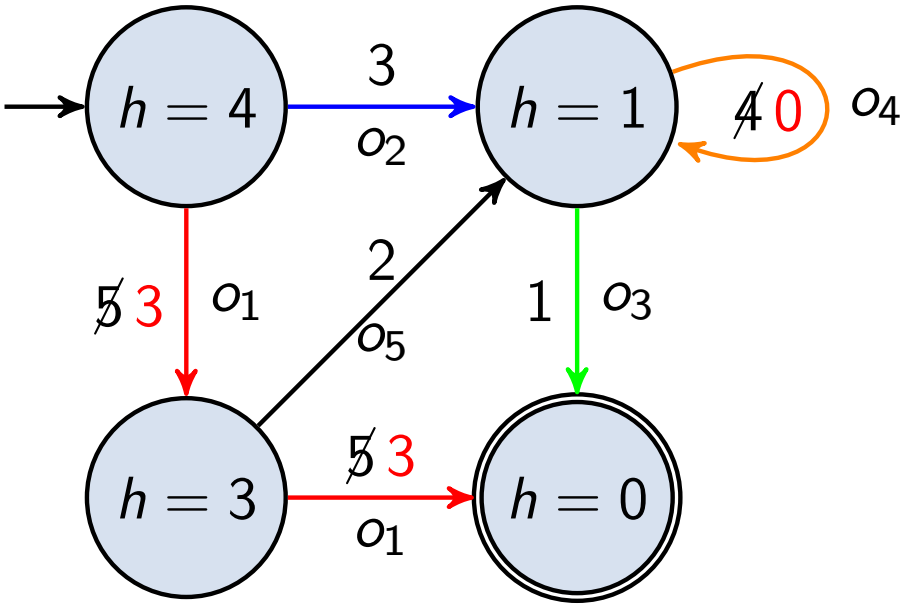


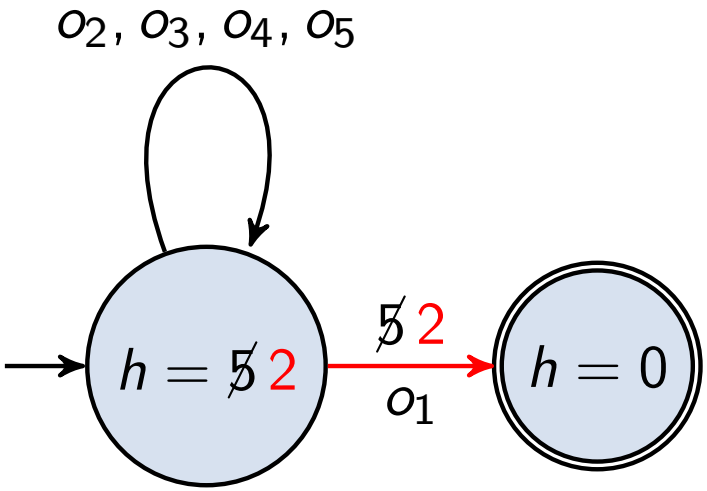



# Saturated Cost Partitioning

## Saturated cost function

$$\hat{c}(o) = \max_{a \xrightarrow{o} b \in T} \max\{0, h(a) - h(b)\}$$

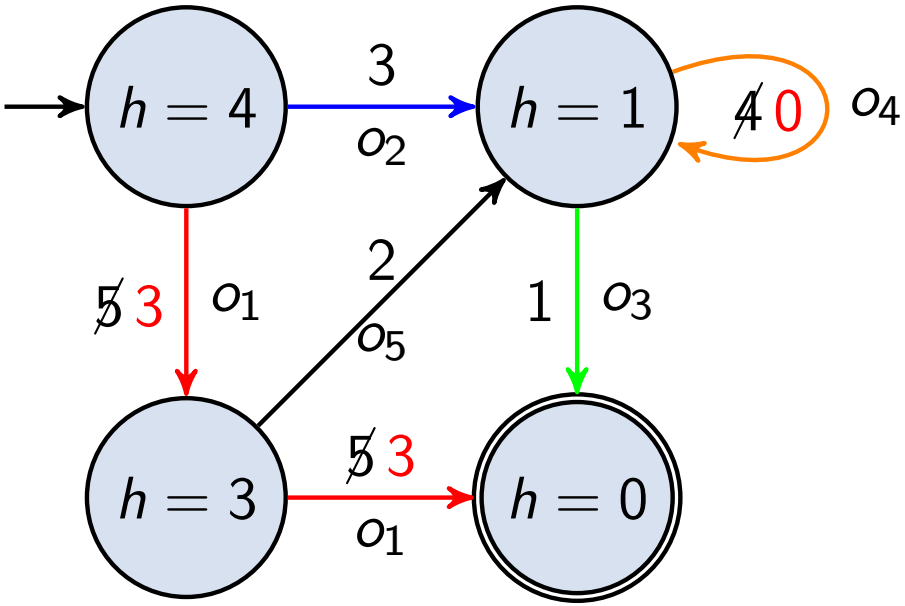


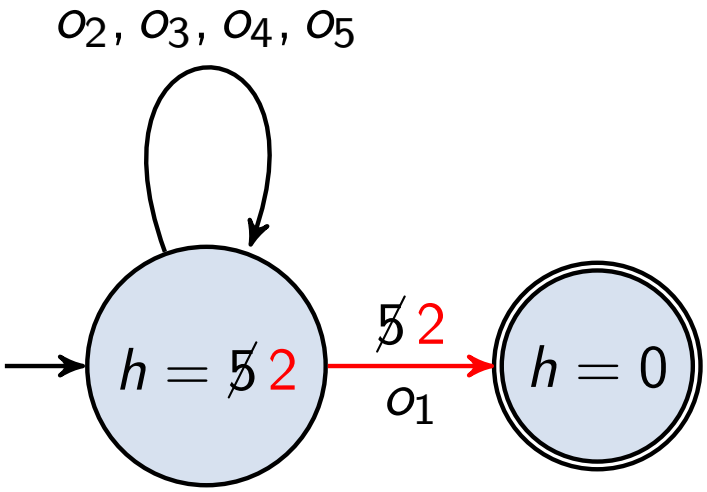



# Saturated Cost Partitioning

## Saturated cost function

$$\hat{c}(o) = \max_{a \xrightarrow{o} b \in T} \max\{0, h(a) - h(b)\}$$



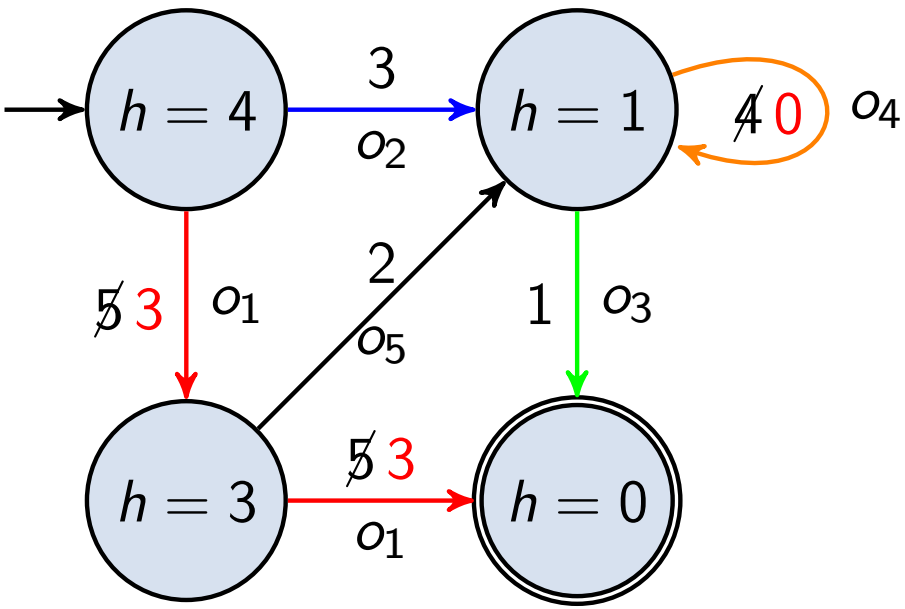



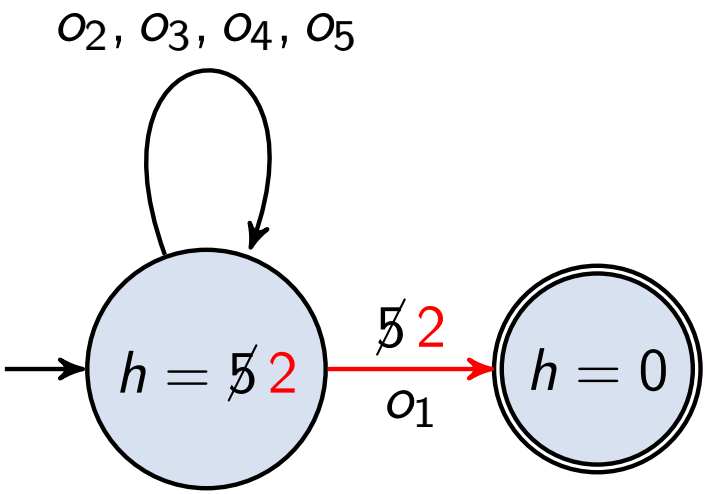

- Saturated cost partitioning: $h(s_0) = 4 + 2 = 6$

# Saturated Cost Partitioning

## Saturated cost function

$$\hat{c}(o) = \max_{a \xrightarrow{o} b \in T} \max\{0, h(a) - h(b)\}$$




- Saturated cost partitioning: $h(s_0) = 4 + 2 = 6$
- $\hat{c}$: minimum distance-preserving cost function

# Additive CEGAR Abstractions

- Build $n$ abstractions
- No changes to the CEGAR algorithm

# Additive CEGAR Abstractions

- Build $n$ abstractions
- No changes to the CEGAR algorithm
- Problem: abstractions too similar $\rightarrow$ no improvement

# Diversification Strategies

# Abstraction by Goals

- Build an abstraction for each goal fact
- Focus on different subproblems

# Abstraction by Goals

- Build an abstraction for each goal fact
- Focus on different subproblems
- Problem: tasks with single goal fact

# Abstraction by Landmarks

- Compute fact landmarks
- Build an abstraction for each fact landmark $L$

# Abstraction by Landmarks

- Compute fact landmarks
- Build an abstraction for each fact landmark $L$
- Problem: landmarks as goals not admissible
- Solution: $h_L(s) = 0$ if $L$ might have been achieved
- Path-dependent landmark heuristics $\rightarrow$ state-based criterion

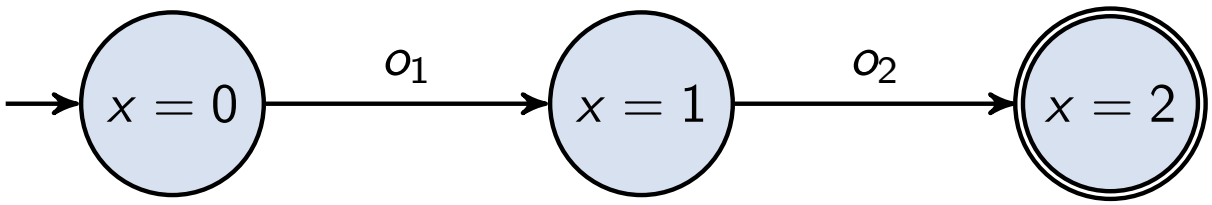
# Abstraction by Landmarks

Modified task for landmark $L$:

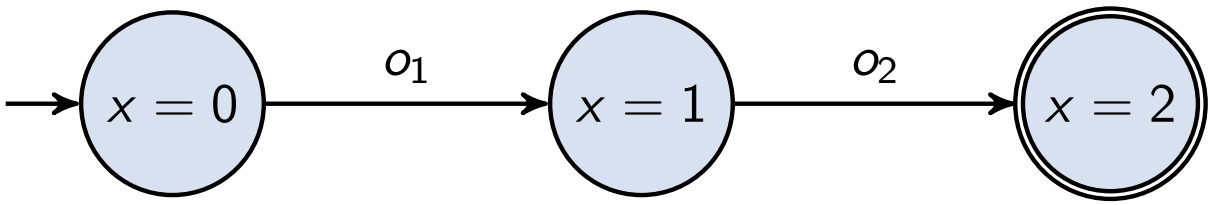
- Compute *possibly-before* set $pb(L)$
- Facts: $pb(L) \cup \{L\}$
- Goal: $L$
- Operators:
  - discard operators with preconditions not in $pb(L)$
  - let operators achieving $L$ achieve only $L$
- Initial state: unmodified

$h_L(s) = 0$ if $s \not\subseteq pb(L) \cup \{L\}$

# Abstraction by Landmarks: Improved



# Abstraction by Landmarks: Improved

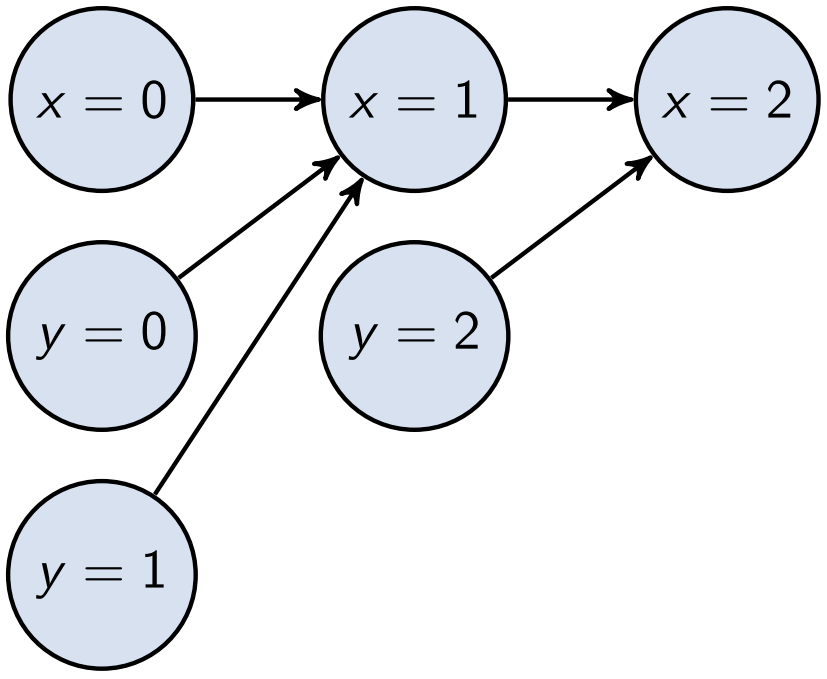



Solution:

- Compute landmark orderings
- Combine facts that have probably already been achieved

# Abstraction by Landmarks: Improved

## Example




- $x = 1$: $\{y = 0, y = 1\}$
- $x = 2$: $\{y = 0, y = 1, y = 2\}, \{x = 0, x = 1\}$

# Conclusion

# Literature

- Clarke et al., CAV 2000: CEGAR for model checking
- Seipp and Helmert, ICAPS 2013: Cartesian CEGAR for planning
- Seipp and Helmert, ICAPS 2014: Diverse and additive Cartesian abstractions

# Summary

- Cartesian abstractions: useful class of abstractions
- Saturated cost partitioning: preserves distances
- Diversification strategies: focus on different subtasks