Heuristics and Symmetries in Classical Planning

Alexander Shleyfman[1] Michael Katz[2] Malte Helmert[3]
Silvan Sievers[3] Martin Wehrle[3]

[1]Technion
Haifa, Israel

[2]IBM Haifa Research Lab
Israel

[3]University of Basel
Switzerland

Classical Planning in STRIPS

A STRIPS Planning task is 4-tuple $\langle P, O, I, G \rangle$:

- P: finite set of propositions
- O: finite set of actions of form $\langle Pre, Add, Del, c \rangle$
  (Preconditions/Add/Delete; subsets of propositions)
  $c \in \mathbb{R}^{0+}$ captures action cost
- I: initial state (subset of propositions)
- G: goal description (subset of propositions)

Classical Planning as Heuristic Search

Recipe

1. Search algorithm - BFS ($GBFS$ or WA^*)
2. Heuristic(s) function(s)
3. Secret ingredients:
   - Inference–based state/action pruning
   - Action preferences
   - ...

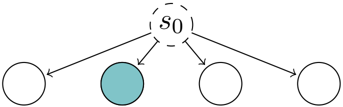
Classical Planning as Heuristic Search

Recipe

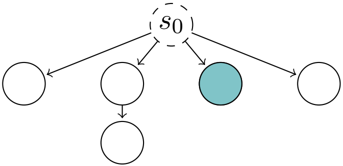
1. Search algorithm - BFS ($GBFS$ or WA^*)
2. Heuristic(s) function(s)
3. Secret ingredients:
   - Inference–based state/action pruning
   - Action preferences
   - ...

BFS tree step-by-step

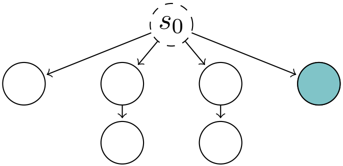
BFS tree step-by-step



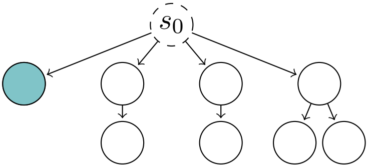
BFS tree step-by-step



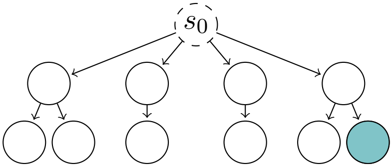
BFS tree step-by-step



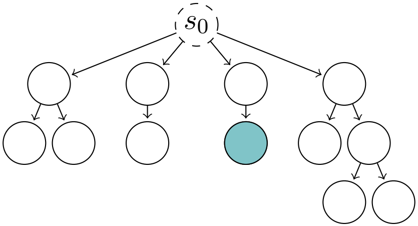
BFS tree step-by-step



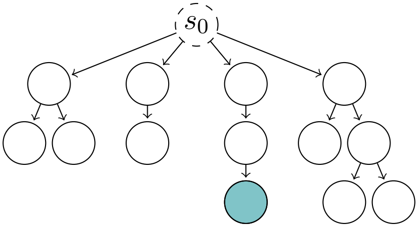
BFS tree step-by-step



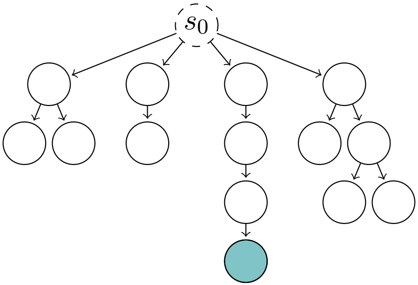
BFS tree step-by-step



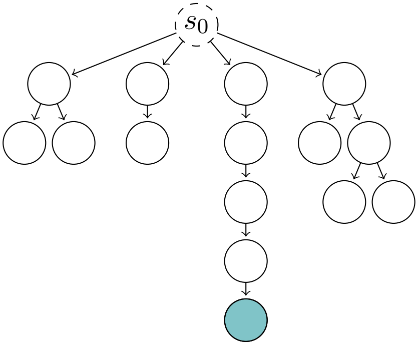
BFS tree step-by-step



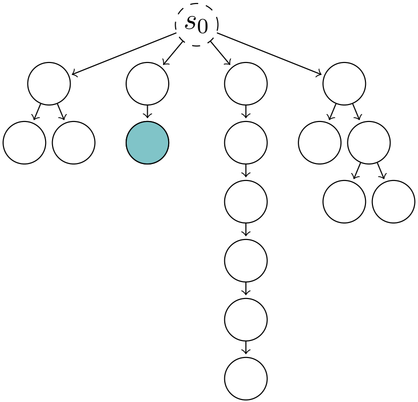
BFS tree step-by-step



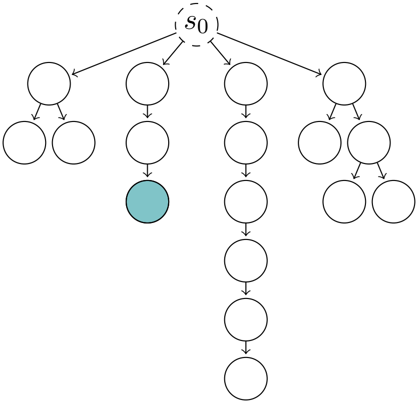
BFS tree step-by-step



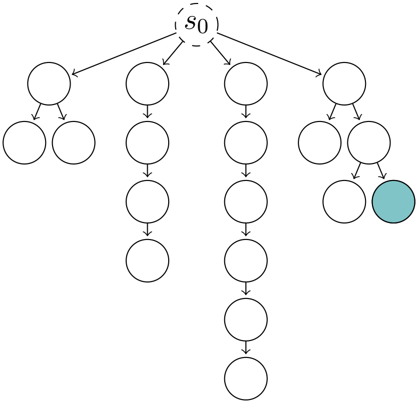
BFS tree step-by-step



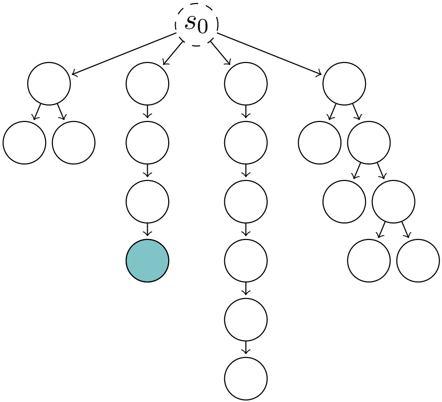
BFS tree step-by-step



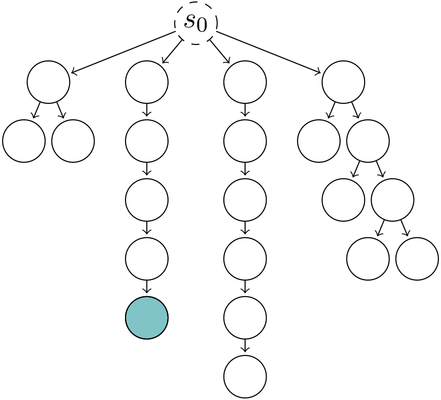
BFS tree step-by-step



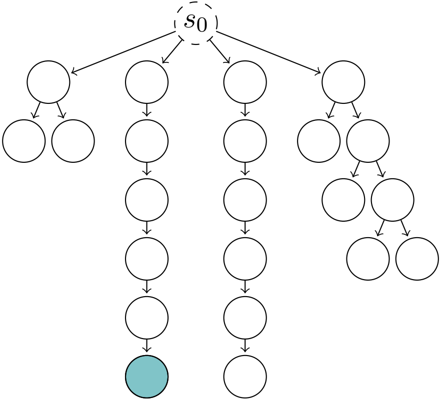
BFS tree step-by-step



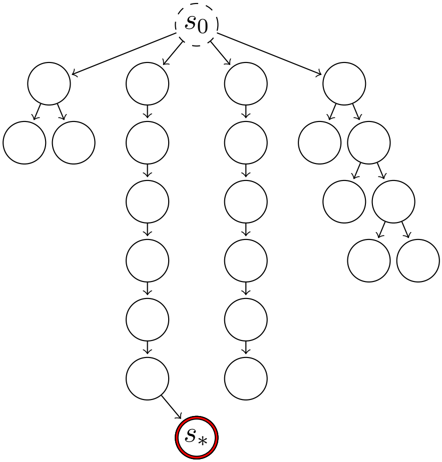
BFS tree step-by-step



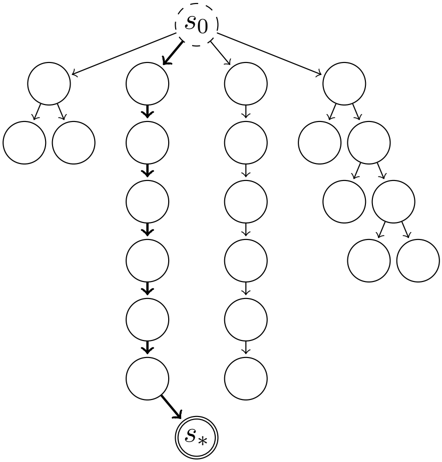
BFS tree step-by-step



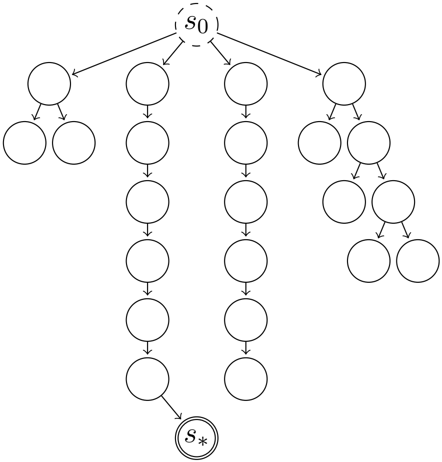
BFS tree step-by-step



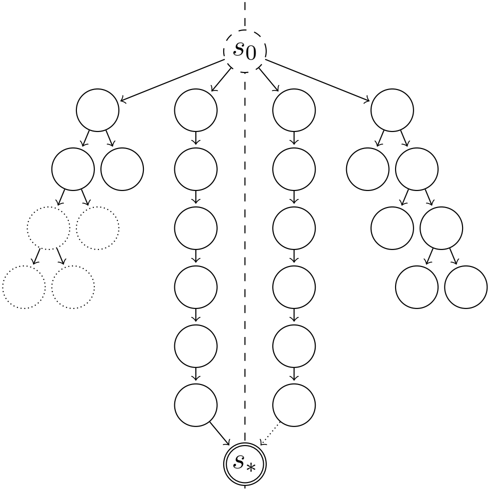
BFS tree step-by-step



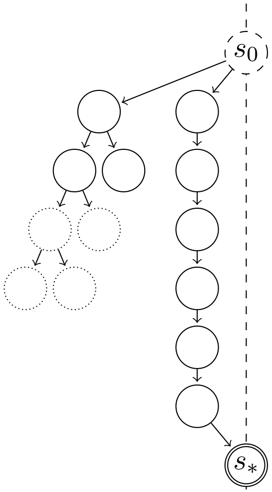
Motivation to use symmetries



Motivation to use symmetries



Motivation to use symmetries



Exploiting Symmetries for Pruning

General Recipe

1. Efficiently generate a subgroup of the automorphism group of the problem's transition graph
   - efficiently = empirically efficiently
2. Use that subgroup to prune *some* symmetric states

- Emerson & Sistla (1996) *[model checking]*
- Rintanen (1993) *[planning as SAT]*
- Fox & Long (1999, 2002) *[Graphplan-style]*
- Pochter, Zohar, and Rosenschein (2011) *[heuristic search]*
- Domshlak, Katz, Shleyfman (2012, 2013) *[heuristic search]*

Exploiting Symmetries for Pruning

General Recipe

1. Efficiently generate a subgroup of the automorphism group of the problem's transition graph
   - efficiently = empirically efficiently
2. Use that subgroup to prune *some* symmetric states

- Emerson & Sistla (1996) *[model checking]*
- Rintanen (1993) *[planning as SAT]*
- Fox & Long (1999, 2002) *[Graphplan-style]*
- Pochter, Zohar, and Rosenschein (2011) *[heuristic search]*
- Domshlak, Katz, Shleyfman (2012, 2013) *[heuristic search]*

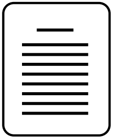
Exploiting Symmetries for Pruning

General Recipe

1. Efficiently generate a subgroup of the automorphism group of the problem's transition graph
   - efficiently = empirically efficiently
2. Use that subgroup to prune *some* symmetric states

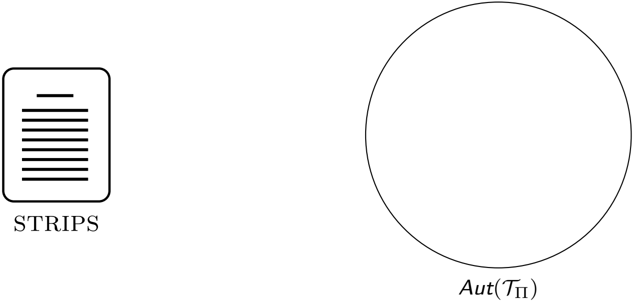
- Emerson & Sistla (1996) *[model checking]*
- Rintanen (1993) *[planning as SAT]*
- Fox & Long (1999, 2002) *[Graphplan-style]*
- Pochter, Zohar, and Rosenschein (2011) *[heuristic search]*
- Domshlak, Katz, Shleyfman (2012, 2013) *[heuristic search]*

Symmetry Groups

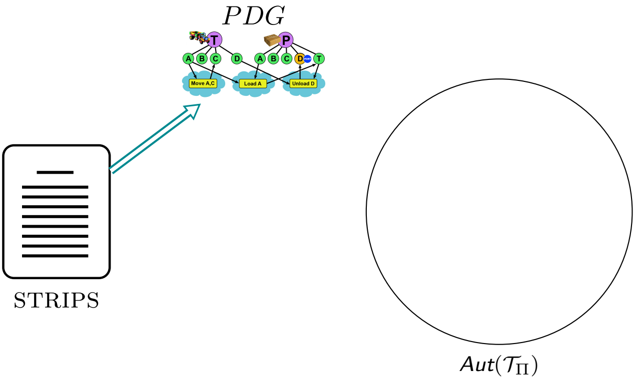


STRIPS

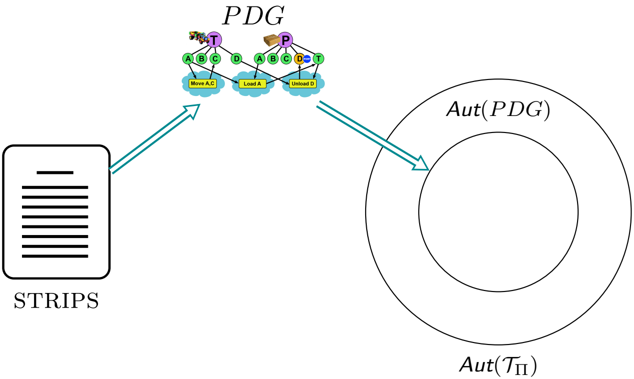
Symmetry Groups



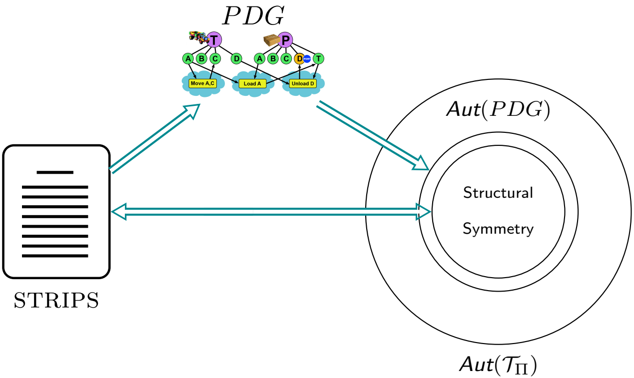
Symmetry Groups



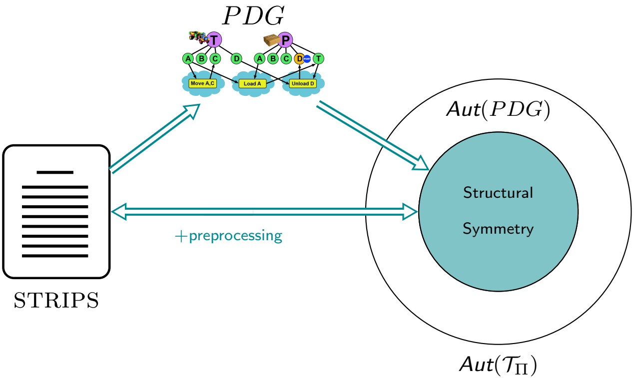
Symmetry Groups



Symmetry Groups



Symmetry Groups



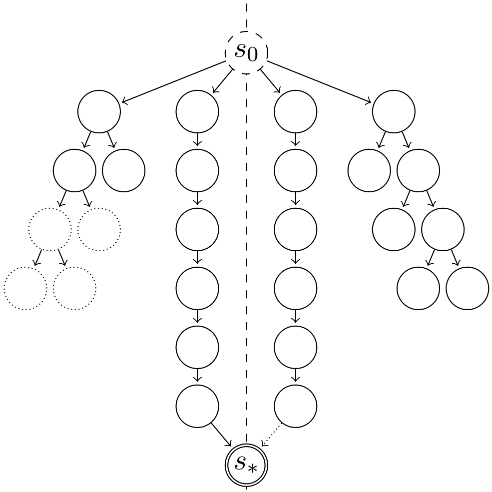
Structural Symmetry

Definition

Let $\langle P, O, I, G \rangle$ be a STRIPS planning task.
A permutation σ is a structural symmetry if

- $\sigma(P) = P$
- $\sigma(O) = O$, and for all $o \in O$:
  - $Pre(\sigma(o)) = \sigma(Pre(o))$
  - $Add(\sigma(o)) = \sigma(Add(o))$
  - $Del(\sigma(o)) = \sigma(Del(o))$
  - $C(\sigma(o)) = C(o)$
- $\sigma(G) = G$

Born equal?



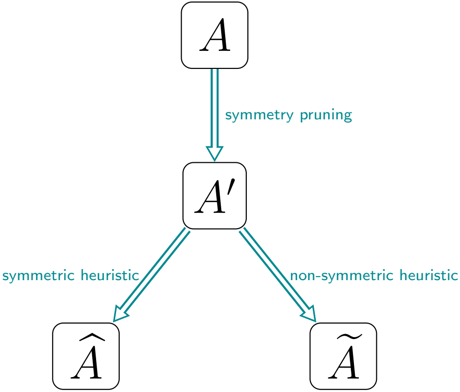
Why do we want to know?

$$\boxed{A}$$

Why do we want to know?

$$\begin{tikzcd} \boxed{A} \arrow[d, Rightarrow, "\text{symmetry pruning}"] \\ \boxed{A'} \end{tikzcd}$$

Why do we want to know?



Heuristics Invariance Under Structural Symmetries

Non-symmetric

Symmetric

Heuristics Invariance Under Structural Symmetries

Non-symmetric

Symmetric

h^+ Hoffmann & Nebel

Heuristics Invariance Under Structural Symmetries

Non-symmetric

Symmetric

h^+ Hoffmann & Nebel

$h_{\max}$ Bonet & Geffner

Heuristics Invariance Under Structural Symmetries

Non-symmetric

Symmetric

h^+ Hoffmann & Nebel

$h_{\max}$ Bonet & Geffner

h_{add} Bonet & Geffner

Heuristics Invariance Under Structural Symmetries

Non-symmetric	Symmetric
	h^+ Hoffmann & Nebel
	$h_{\max}$ Bonet & Geffner
	h_{add} Bonet & Geffner
h_{FF} Hoffmann & Nebel	$\mathbb{E}h_{\text{FF}}$ Hoffmann & Nebel

Heuristics Invariance Under Structural Symmetries

Non-symmetric

h_{FF} Hoffmann & Nebel

$h_{\mathrm{FF}}/h_{\mathrm{add}}$ Keyder & Geffner

Symmetric

h^+ Hoffmann & Nebel

$h_{\max}$ Bonet & Geffner

h_{add} Bonet & Geffner

$\mathbb{E}h_{\mathrm{FF}}$ Hoffmann & Nebel

Heuristics Invariance Under Structural Symmetries

Non-symmetric

h_{FF} Hoffmann & Nebel

$h_{\mathrm{FF}}/h_{\mathrm{add}}$ Keyder & Geffner

$h_{\mathrm{FF}}/h_{\max}$ Keyder & Geffner

Symmetric

h^{+} Hoffmann & Nebel

$h_{\max}$ Bonet & Geffner

h_{add} Bonet & Geffner

$\mathbb{E}h_{\mathrm{FF}}$ Hoffmann & Nebel

Heuristics Invariance Under Structural Symmetries

Non-symmetric

h_{FF} Hoffmann & Nebel

$h_{\mathrm{FF}}/h_{\mathrm{add}}$ Keyder & Geffner

$h_{\mathrm{FF}}/h_{\max}$ Keyder & Geffner

Symmetric

h^+ Hoffmann & Nebel

$h_{\max}$ Bonet & Geffner

h_{add} Bonet & Geffner

$\mathbb{E}h_{\mathrm{FF}}$ Hoffmann & Nebel

h^m Haslum & Geffner

Heuristics Invariance Under Structural Symmetries

Non-symmetric	Symmetric
	h^+ Hoffmann & Nebel
	$h_{\max}$ Bonet & Geffner
	h_{add} Bonet & Geffner
h_{FF} Hoffmann & Nebel	$\mathbb{E}h_{\text{FF}}$ Hoffmann & Nebel
$h_{\text{FF}}/h_{\text{add}}$ Keyder & Geffner	
$h_{\text{FF}}/h_{\max}$ Keyder & Geffner	
	h^m Haslum & Geffner
$h^{\text{LM-cut}}$ Helmert & Domshlak	$\mathbb{E}h^{\text{LM-cut}}$ Helmert & Domshlak

Landmarks and generation procedures

Non-symmetric

Symmetric

Landmarks and generation procedures

Non-symmetric

Symmetric

ZG Zhu & Givan 2003

Landmarks and generation procedures

Non-symmetric	Symmetric
	ZG Zhu & Givan 2003
	KRH Keyder, Richter, & Helmert 2010

Landmarks and generation procedures

Non-symmetric

RHW Richter, Helmert, & Westphal 2008

Symmetric

ZG Zhu & Givan 2003

KRH Keyder, Richter, & Helmert 2010

Landmarks and generation procedures

Non-symmetric

RHW Richter, Helmert, & Westphal 2008

Symmetric

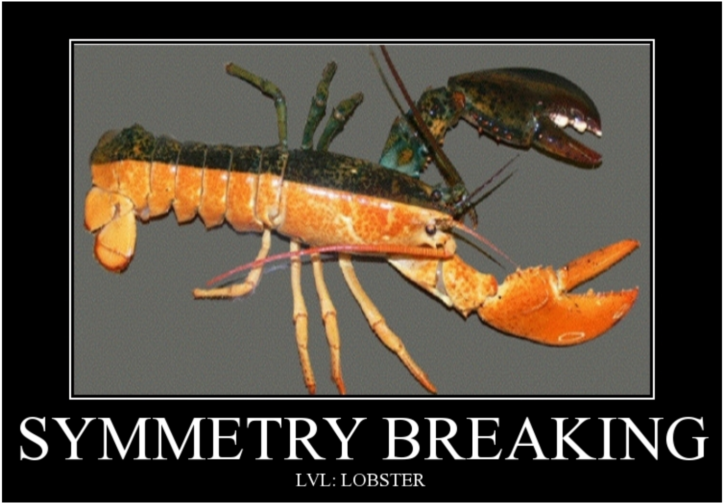
ZG Zhu & Givan 2003

KRH Keyder, Richter, & Helmert 2010

Given that generation method is invariant under structural symmetry the heuristics below are symmetric

- counting landmarks (Richter, Helmert, & Westphal)
- optimal/uniform cost partitioning (Karpas & Domshlak)
- hitting sets (Bonet & Helmert)

The END



Lobster is taken from: http://www.biology.ualberta.ca/palmer.hp/asym/axes/splitlobster.GIF