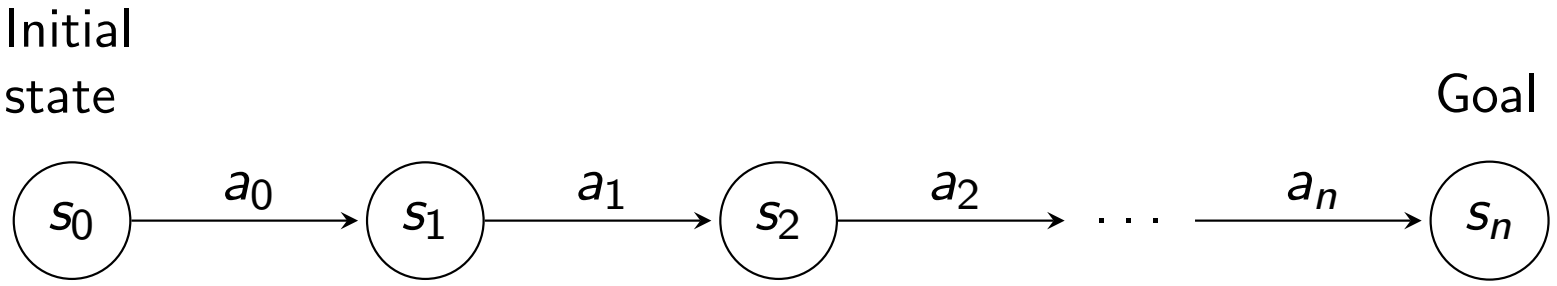
Heuristic Planning with Single Action Goal Expansion

Remo Christen

Department of Mathematics and Computer Science
University of Basel

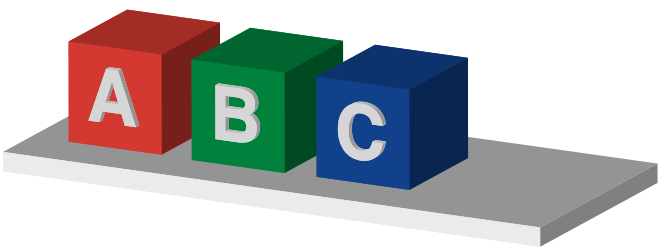
5.7.2019

Classical Planning



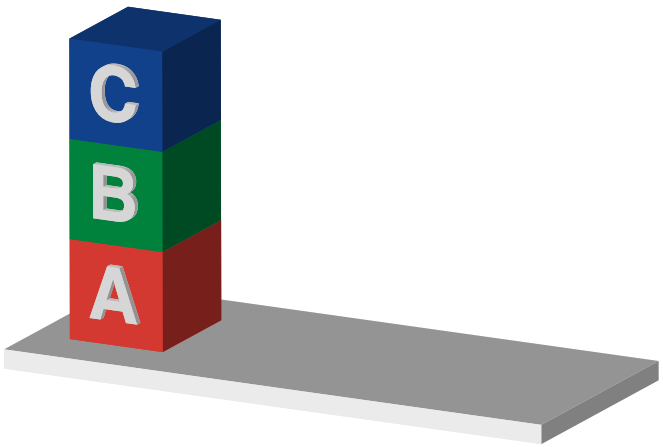
Blocksworld Domain

Initial state

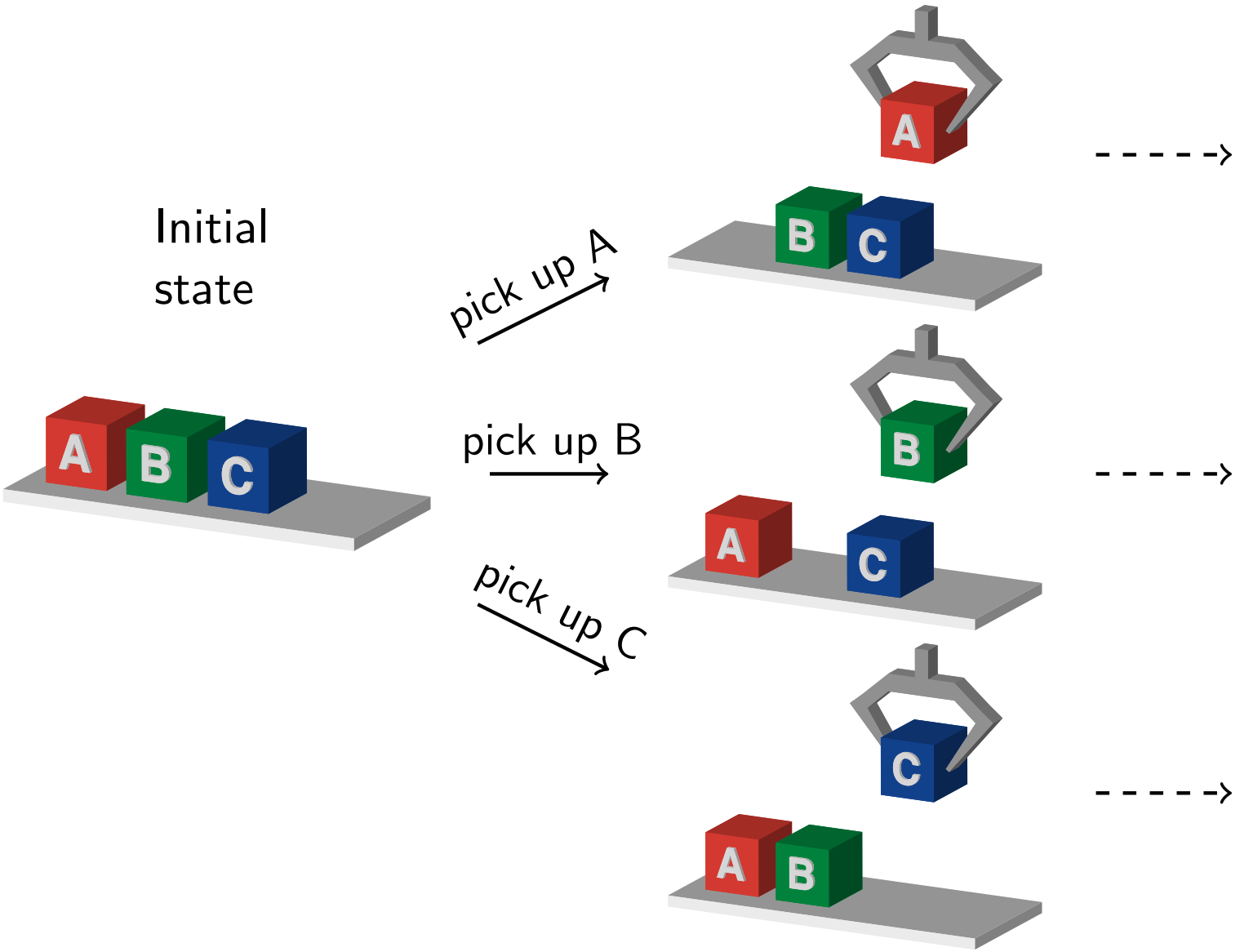


$\xrightarrow{a_0} \cdots \xrightarrow{a_n}$

Goal

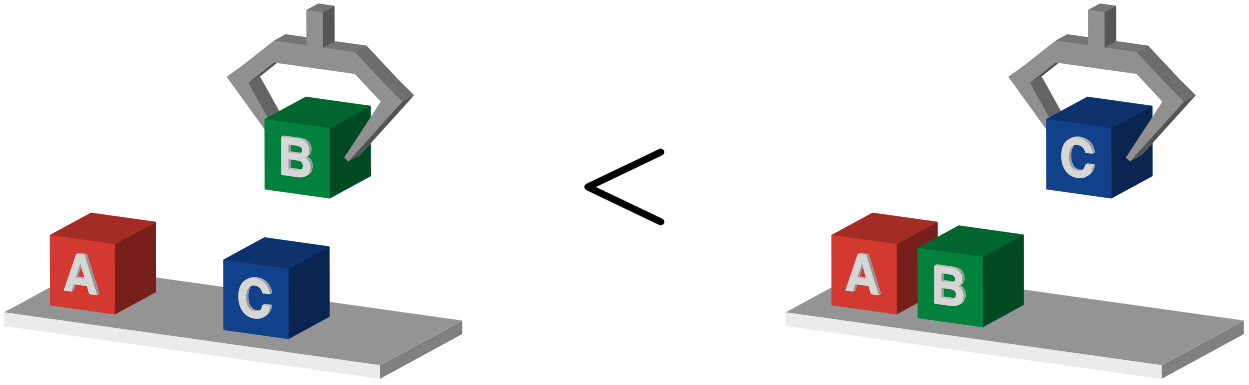


Forward Search



Heuristics

- Estimate the distance from a state to the goal
- Assign every state a number indicating this estimate



Heuristics

- Estimate the distance from a state to the goal
- Assign every state a number indicating this estimate

Delete Relaxation Heuristic

- Find plan of simplified problem
  $\rightarrow$ Relaxed Plan
- Simplify by removing delete effects
- Relaxed plan length is heuristic value

Heuristics

- Estimate the distance from a state to the goal
- Assign every state a number indicating this estimate

Delete Relaxation Heuristic

- Find plan of simplified problem
  $\rightarrow$ Relaxed Plan
- Simplify by removing delete effects
- Relaxed plan length is heuristic value

Pick up C

$pre = \{\texttt{clear C}, \texttt{ontable C}, \texttt{handempty}\}$
$add = \{\texttt{holding C}\}$
$del = \{\texttt{clear C}, \texttt{ontable C}, \texttt{handempty}\}$

Heuristics

- Estimate the distance from a state to the goal
- Assign every state a number indicating this estimate

Delete Relaxation Heuristic

- Find plan of simplified problem
  $\rightarrow$ Relaxed Plan
- Simplify by removing delete effects
- Relaxed plan length is heuristic value

Pick up C

pre = {`clear C`, `ontable C`, `handempty`}

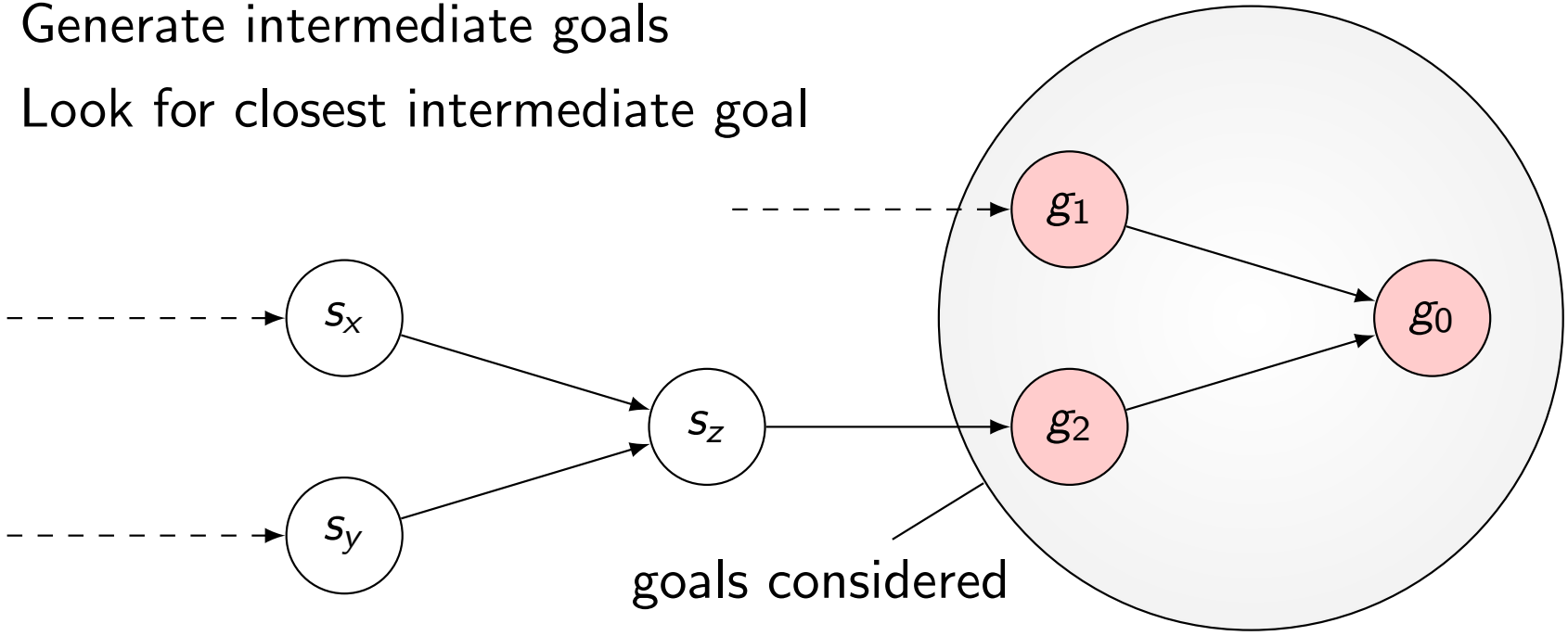
add = {`holding C`}

~~*del* = {`clear C`, `ontable C`, `handempty`}~~

Starting Point

Based on **"Using Backwards Generated Goals for Heuristic Planning"** by Alcázar et al. (2010):

- Use information from relaxed plan
- Generate intermediate goals
- Look for closest intermediate goal



Starting Point

Based on **"Using Backwards Generated Goals for Heuristic Planning"** by Alcázar et al. (2010):

- Use information from relaxed plan
- Generate intermediate goals
- Look for closest intermediate goal

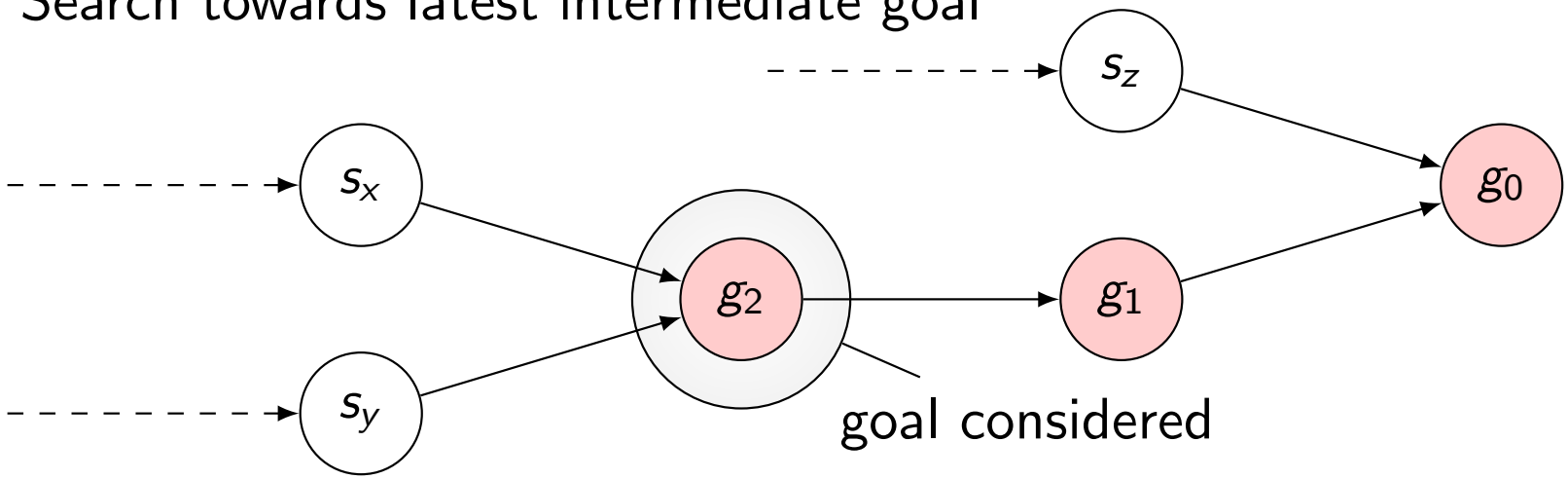
Goals:

- Reduce depth of Heuristic Computation
- Reduce depth of Forward Search
- Handle difficulties close to the goal

Single Action Goal Expansion

Modify approach of Alcázar et al. by only expanding along a **single sequence of actions**:

- Introduce decision strategies about when to expand
- Limit generation of intermediate goals
- Search towards latest intermediate goal



Single Action Goal Expansion

Modify approach of Alcázar et al. by only expanding along a **single sequence of actions**:

- Introduce decision strategies about when to expand
- Limit generation of intermediate goals
- Search towards latest intermediate goal

Goals:

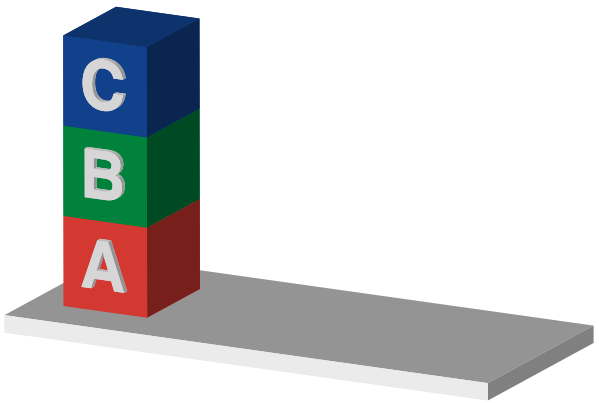
- Improve efficiency
- Preserve advantages of goal expansion

Overview

Structure of the Goal Expansion Process

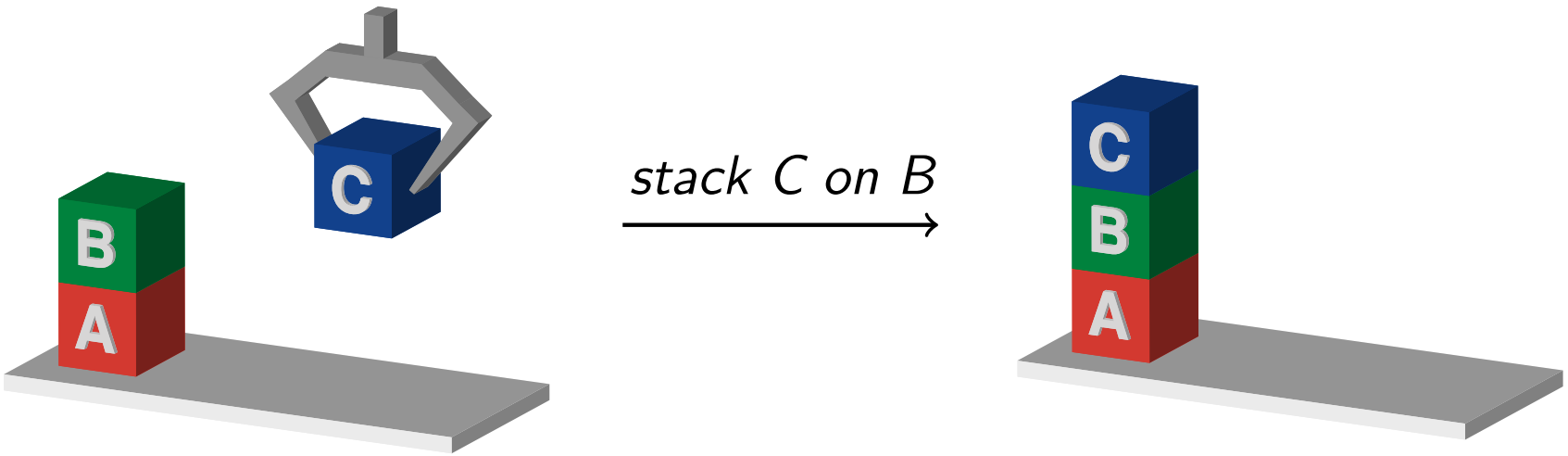
1. State Decision
2. Operator Ordering
3. Check Operator for Legality
4. Expand Goal with Legal Operator

Goal Expansion



Original Goal

Goal Expansion

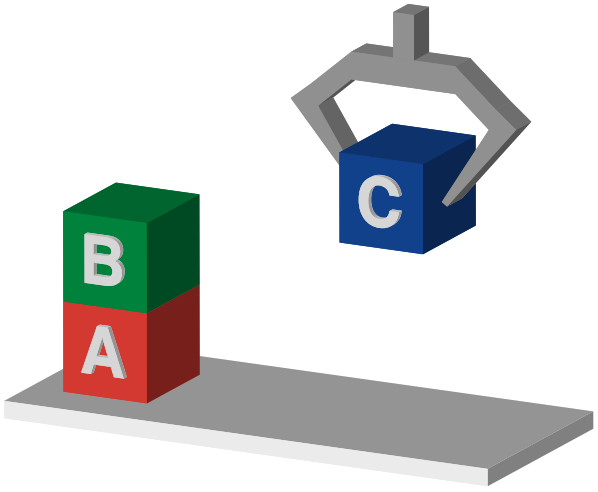


New Goal

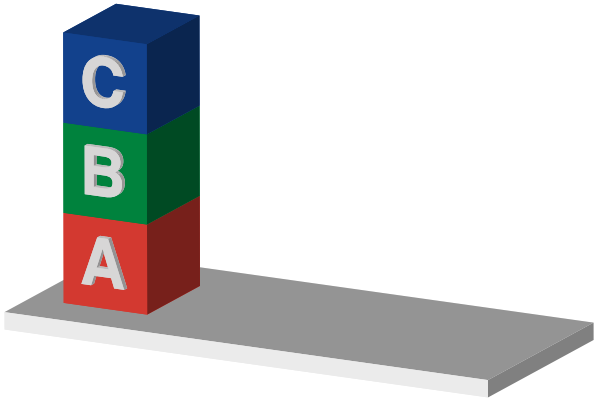
Original Goal

Goal Expansion

```
B on A
C on B
```



stack C on B

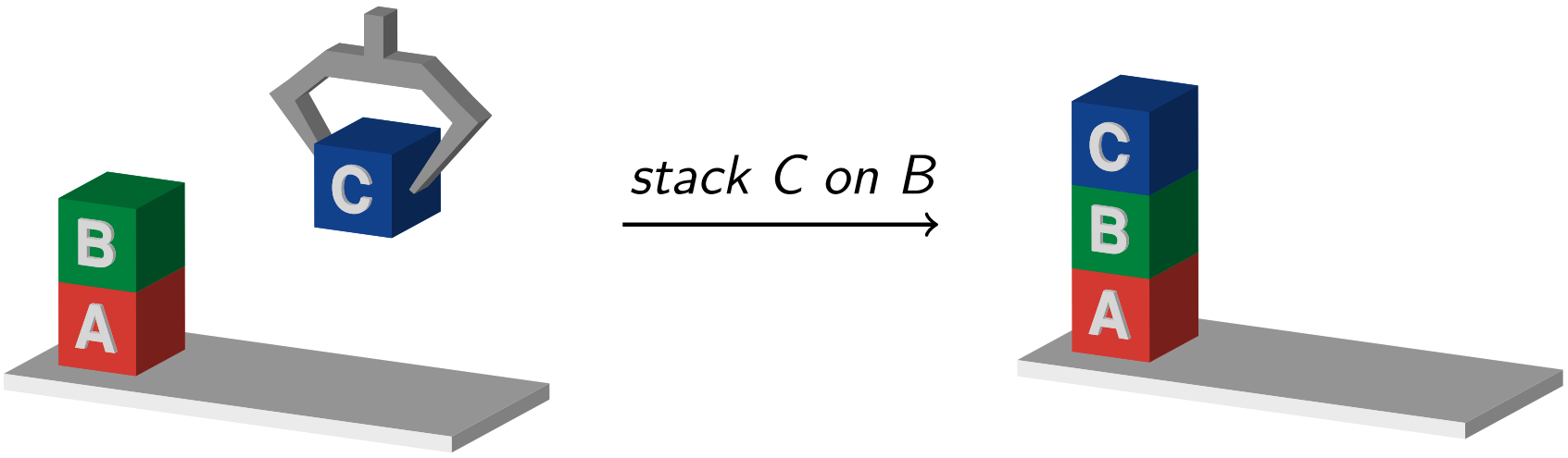


New Goal

Original Goal

Goal Expansion

$$
\begin{aligned}
\textit{pre} &= \{\texttt{clear B}, \texttt{holding C}\} && \texttt{B on A}\\
\textit{add} &= \{\texttt{C on B}, \texttt{clear C}, \texttt{hand empty}\} && \texttt{C on B}\\
\textit{del} &= \{\texttt{clear B}, \texttt{holding C}\}
\end{aligned}
$$

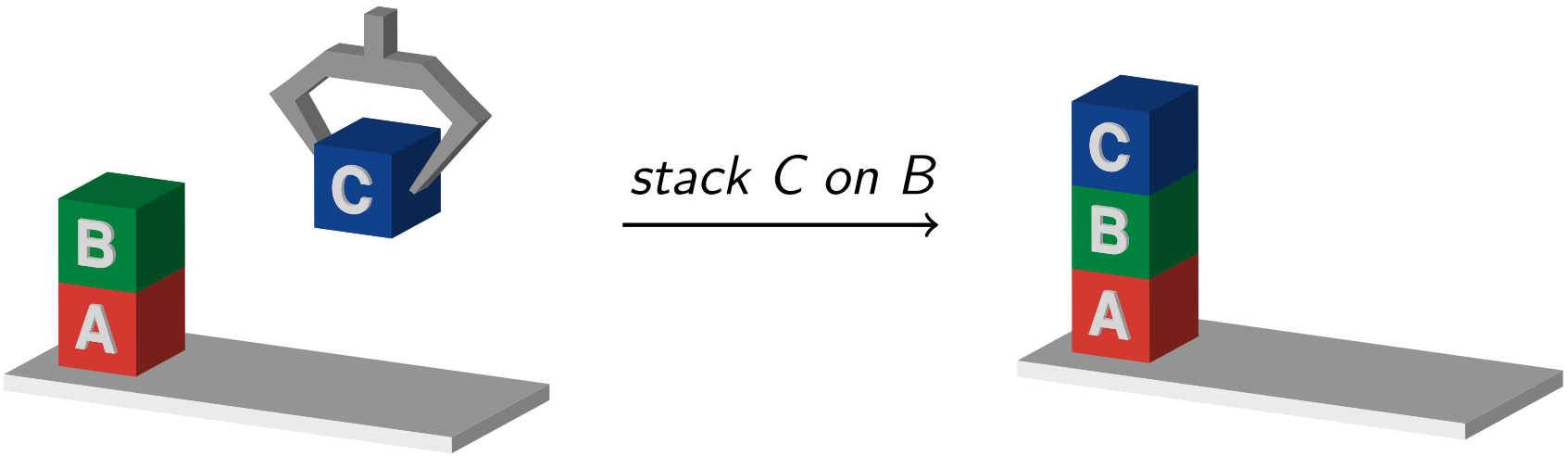


New Goal

Original Goal

Goal Expansion

$$
\begin{aligned}
\textit{pre} &= \{\texttt{clear B}, \texttt{holding C}\} \\
\textit{add} &= \{\texttt{C on B}, \texttt{clear C}, \texttt{hand empty}\} \\
\textit{del} &= \{\texttt{clear B}, \texttt{holding C}\}
\end{aligned}
\qquad
\begin{aligned}
&\texttt{B on A} \\
&\texttt{C on B}
\end{aligned}
$$

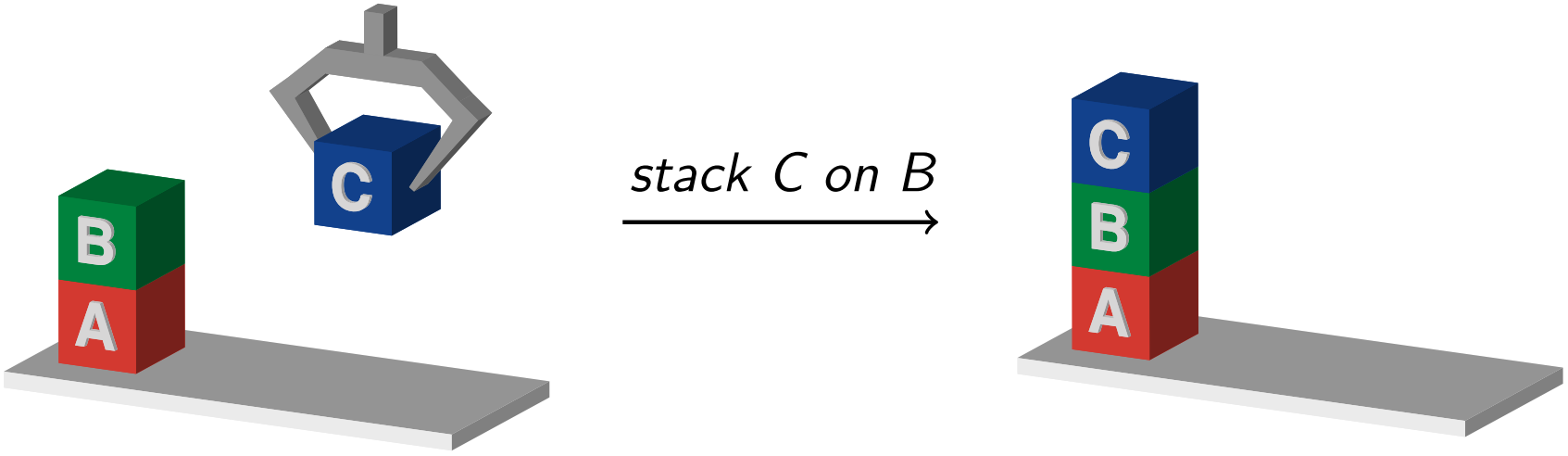


New Goal

Original Goal

Goal Expansion

`B on A`	$pre = \{$`clear B, holding C`$\}$	`B on A`
`clear B`	$add = \{$`C on B, clear C, hand empty`$\}$	`C on B`
`holding C`	$del = \{$`clear B, holding C`$\}$	

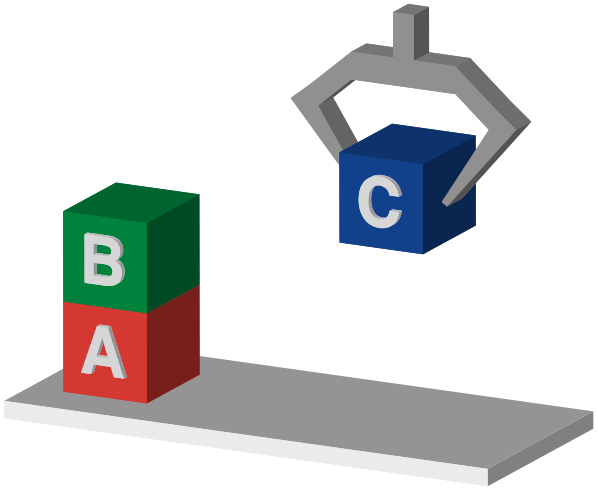


New Goal

Original Goal

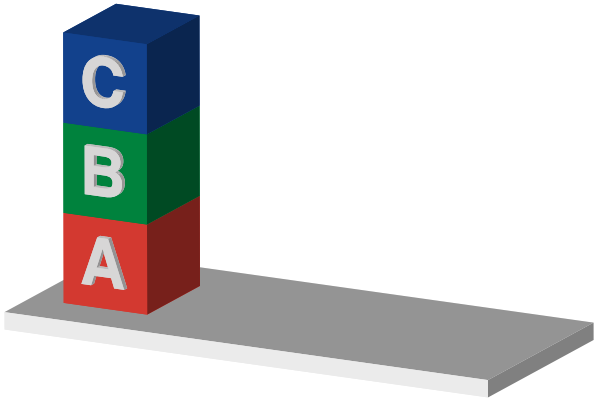
Goal Expansion

B on A	$pre = \{$clear B, holding C$\}$	B on A
clear B	$add = \{$C on B, clear C, hand empty$\}$	C on B
holding C	$del = \{$clear B, holding C$\}$	



New Goal

stack C on B →



Original Goal

Overview

Structure of the Goal Expansion Process

1. State Decision
2. Operator Ordering
3. Check Operator for Legality
4. ~~Expand Goal with Legal Operator~~

State decision

Expand the goal when ...

NewMinimum
...a state has the lowest heuristic value of all evaluated states.

State decision

Expand the goal when ...

NewMinimum
... a state has the lowest heuristic value of all evaluated states.

Accuracy
... the difference of a state's heuristic value and the heuristic value of the initial state is equal to the cost of reaching the state.

State decision

Expand the goal when ...

NewMinimum
... a state has the lowest heuristic value of all evaluated states.

Accuracy
... the difference of a state's heuristic value and the heuristic value of the initial state is equal to the cost of reaching the state.

Counter
... an operator appeared in the relaxed plan of a set percentage of previous evaluations, given that a minimum number of evaluations has been reached.

Overview

Structure of the Goal Expansion Process

1. ~~State Decision~~
2. Operator Ordering
3. Check Operator for Legality
4. ~~Expand Goal with Legal Operator~~

Operator Ordering

Collect all operators from the relaxed plan that satisfy a goal proposition. Order them according to one of two criteria:

MostSatisfied

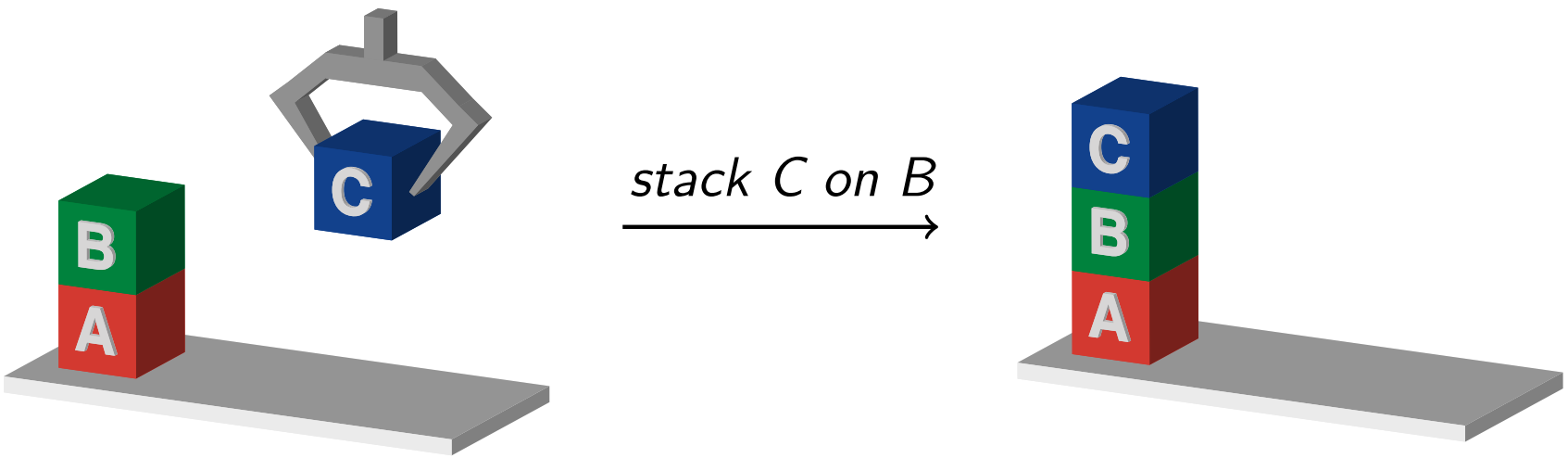
or

LowestLayer

Operator Ordering

MostSatisfied

`B on A`	$pre = \{$`clear B, holding C`$\}$	`B on A`
`clear B`	$add = \{$`C on B, clear C, hand empty`$\}$	`C on B`
`holding C`	$del = \{$`clear B, holding C`$\}$	



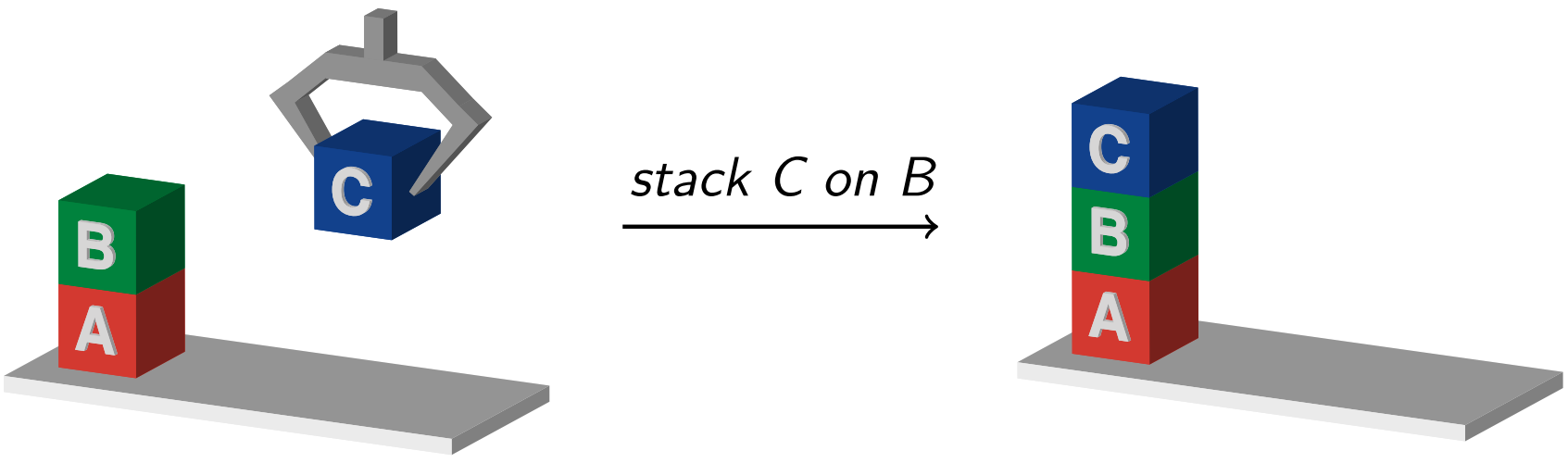
New Goal

Original Goal

Operator Ordering

LowestLayer

`B on A`	$pre = \{$`clear B, holding C`$\}$	`B on A`
`clear B`	$add = \{$`C on B, clear C, hand empty`$\}$	`C on B`
`holding C`	$del = \{$`clear B, holding C`$\}$	



New Goal

Original Goal

Overview

Structure of the Goal Expansion Process

1. ~~State Decision~~
2. ~~Operator Ordering~~
3. Check Operator for Legality
4. ~~Expand Goal with Legal Operator~~

Legality Check

An operator is not legal if it meets one of these three conditions:

- Deletes goal proposition
- Mutual exclusion between goal propositions
- Dominated by previous goal

Results

Coverage (Total 1827)	NewMin	Accuracy	Counter	Eager Greedy (FF)
MostSatisfied	1051	1165	727	1503
LowestLayer	1027	1184	726	

Results

Coverage (Total 1827)	NewMin	Accuracy	Counter	Eager Greedy (FF)
MostSatisfied	1051	1165	727	1503
LowestLayer	1027	1184	726	

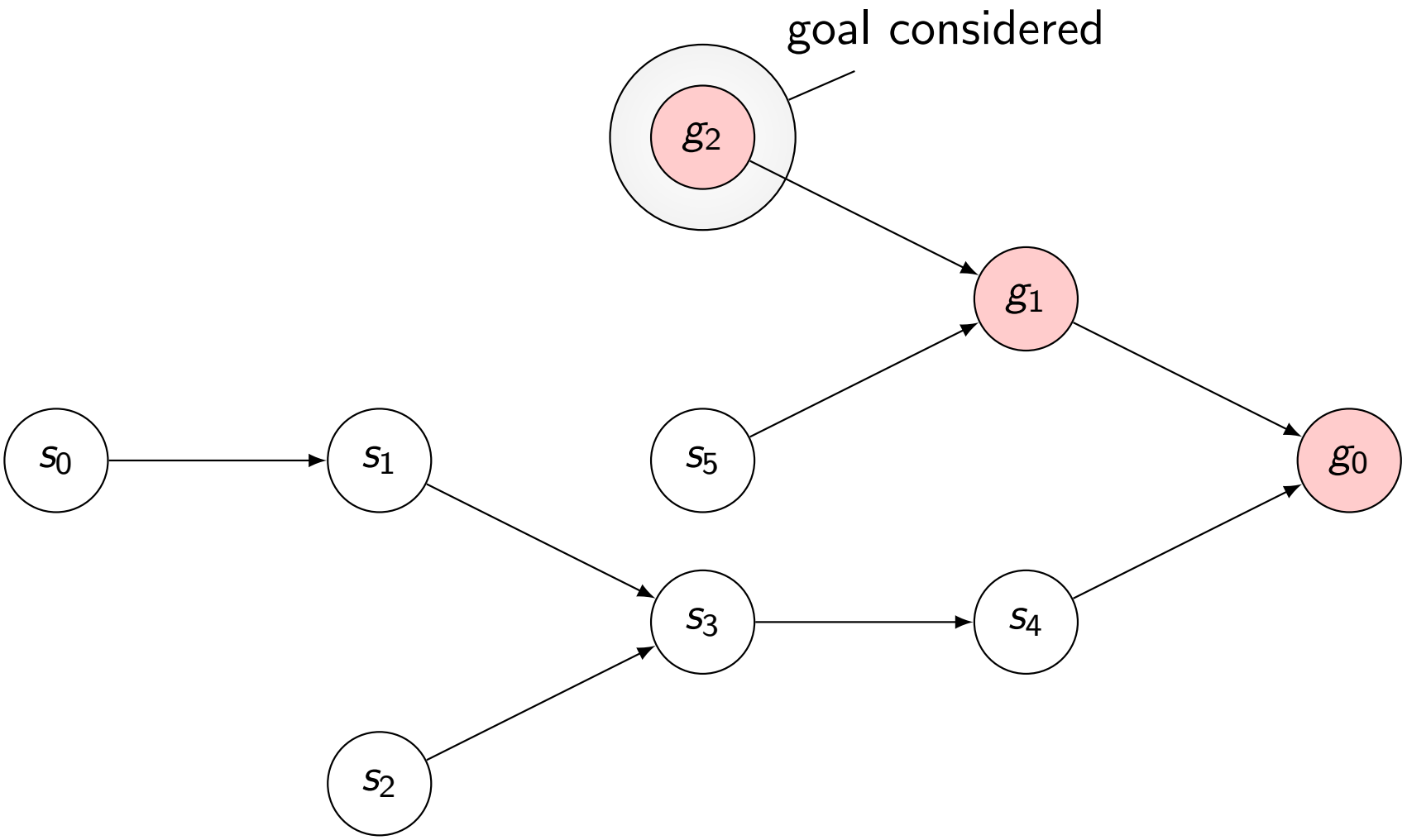
Goal Expansions Total	NewMin	Accuracy	Counter	Eager Greedy (FF)
MostSatisfied	11699	7338	108636	0
LowestLayer	11986	6488	189694	

Results

Forward Expansions	NewMin	Accuracy	Counter	Eager Greedy (FF)
blocks (32)	**112.70**	444.76	525.74	464.76
driverlog (14)	207.41	63.29	**40.21**	115.88
elevator-opt08-strips (19)	**1379.44**	2691.59	2154.87	3467.17
elevator-opt11-strips (12)	1462.71	2691.59	**925.79**	3516.75
gripper (17)	134.88	192.99	**125.87**	380.28
logistics00 (26)	153.92	34.04	**25.35**	42.20
miconic (130)	56.14	50.79	**28.32**	68.54
rovers (16)	111.78	**104.46**	153.03	225.05
scanalyzer-08-strips (23)	65.04	84.06	**29.25**	89.42
scanalyzer-opt11-strips (15)	71.71	45.73	**19.00**	48.49

Geometric mean of forward expansions for problems solved by all variants per domain. The operator order is LowestLayer.

Unreachable Intermediate Goals



Conclusion

- Negligable difference between operator orderings
- NewMinimum and Accuracy outperform Counter
- Visible potential in suitable domains
- Unreachable intermediate goals pose a problem

Future Work

- Evaluate COUNTER with different settings
- What makes a domain suitable
- Find a strategy to avoid unreachable intermediate goals
- Improve implementation

Questions