Universität Basel

Efficient Implementation of h^2 in the Fast Downward Planning System

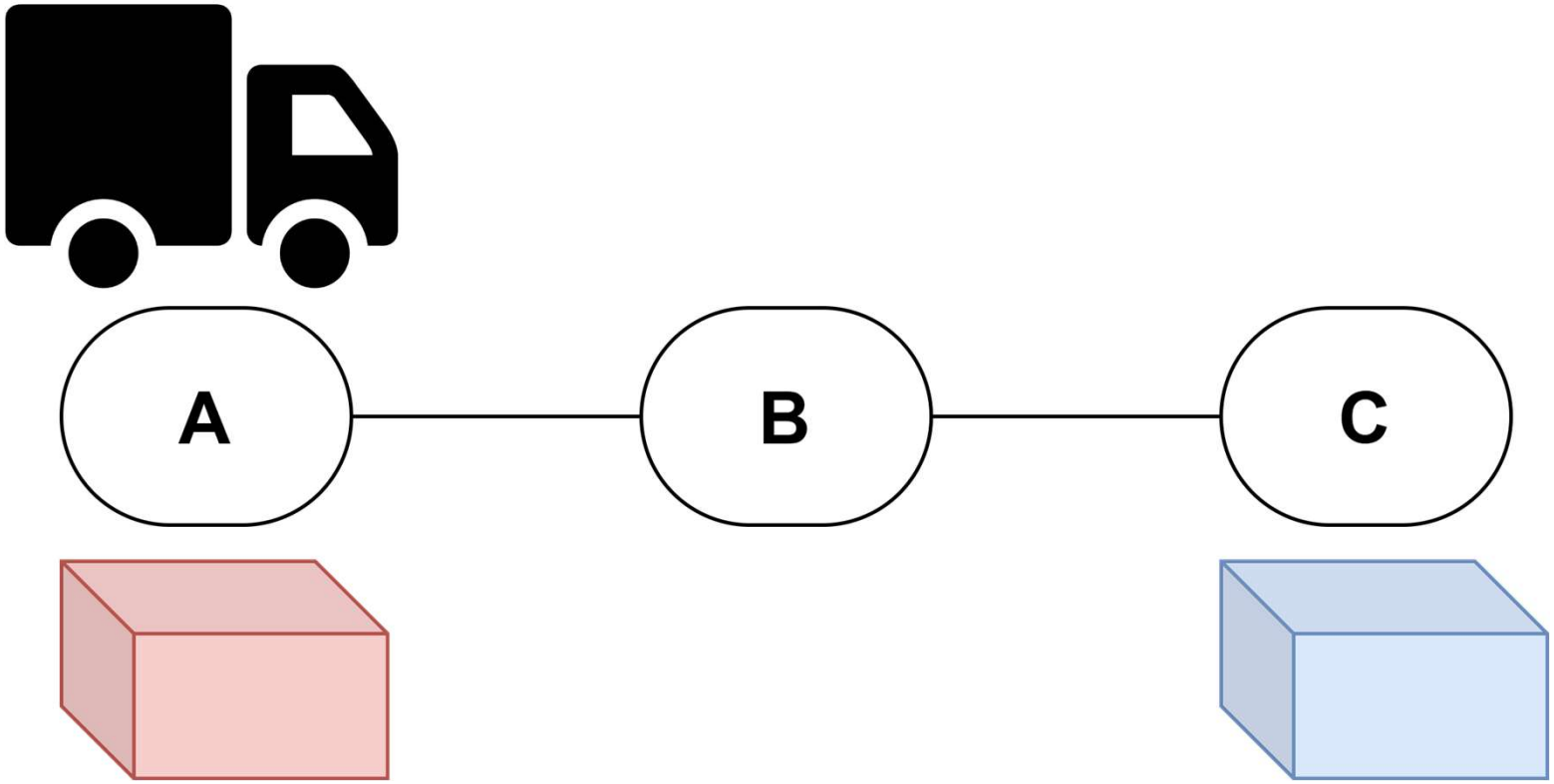
Master's Thesis Presentation

Elia Hänggi

June 11, 2025

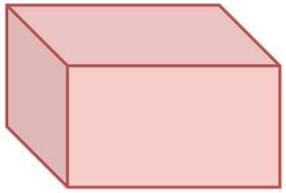
Table of Contents

Example Problem Domain

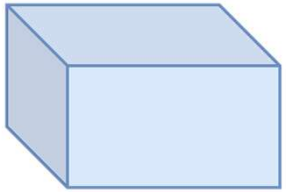


SAS$^+$ Representation

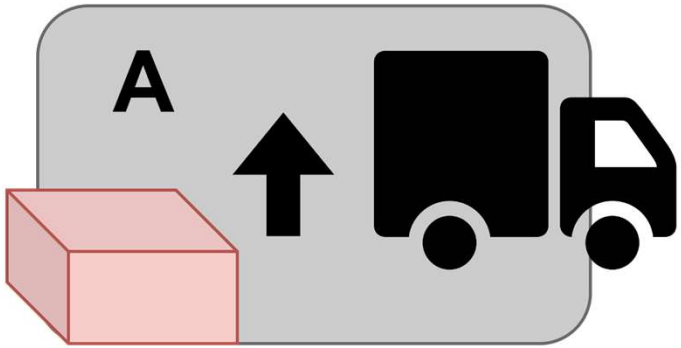
= {A, B, C}



= {A, B, C,  }

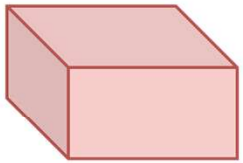


= {A, B, C,  }



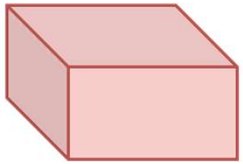
Preconditions

= A



= A

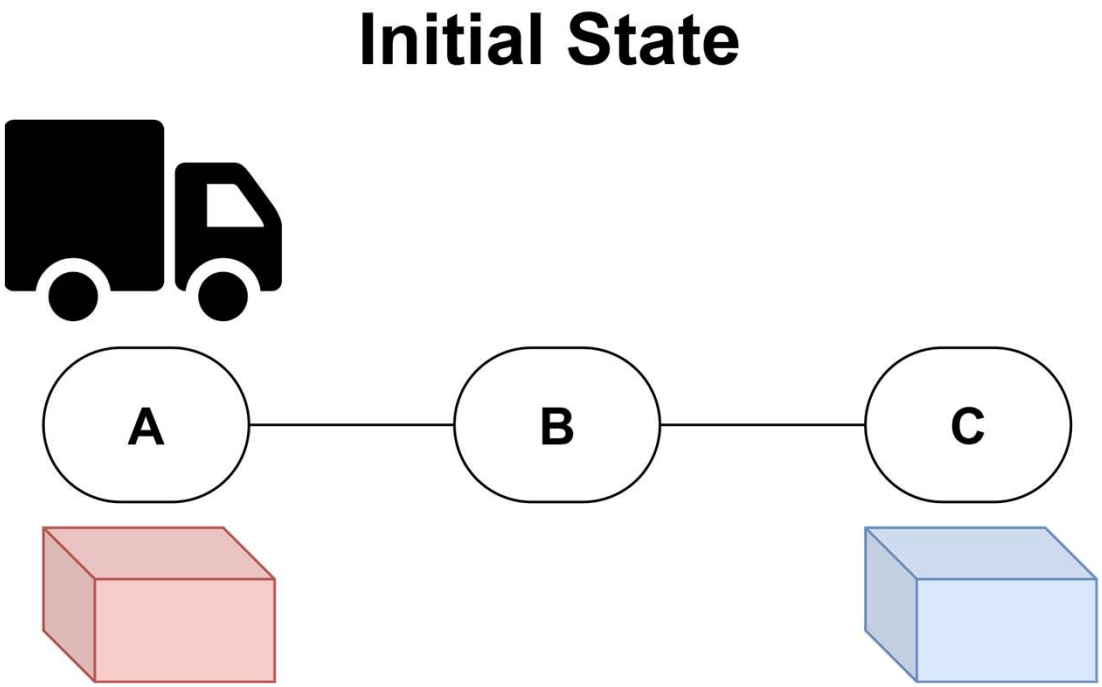
Effects



= 

Heuristic Table Initialization

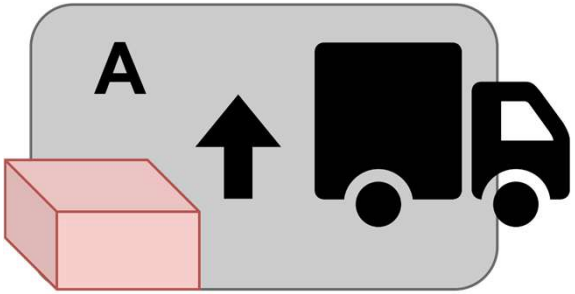
Heuristic Table	
= A	0
= A = A	0
= A = B	∞
= A = C	∞
= A =	∞
= A = A	∞
= A	0
=	∞



Operator Evaluation

Heuristic Table	
= A	0
= A = A	0
= A = B	∞
= A = C	∞
= A =	∞
= A = A	∞
= A	0
=	∞

. . . . .



Preconditions

= A

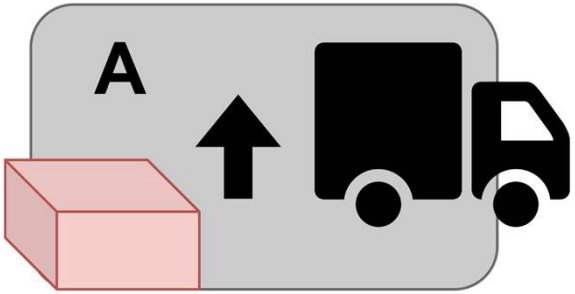
= A

Effects

=

Operator Evaluation

Heuristic Table	
= A	0
= A = A	0
= A = B	∞
= A = C	∞
= A =	∞
= A = A	∞
= A	0
=	1



Preconditions

= A

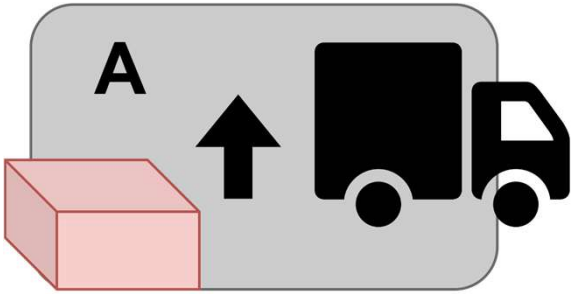
= A

Effects

=

Operator Evaluation

Heuristic Table	
= A	0
= A = A	0
= A = B	∞
= A = C	∞
= A =	1
= A = A	∞
= A	0
=	1



Preconditions

= A

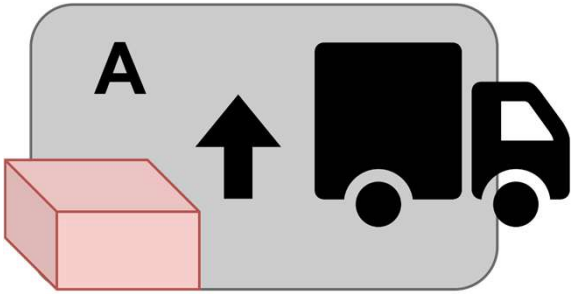
= A

Effects

=

Operator Evaluation

Heuristic Table	
= A	0
= A = A	0
= A = B	∞
= A = C	∞
= A =	1
= A = A	∞
= A	0
=	1
= = C	∞



Preconditions

= A

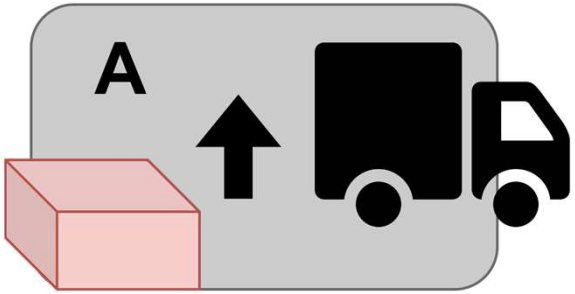
= A

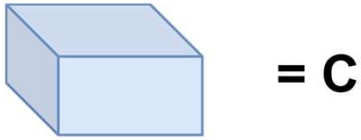
Effects

=

Operator Evaluation

Heuristic Table	
= A	0
= A = A	0
= A = B	∞
= A = C	∞
= A =	1
= A = A	∞
= A	0
=	1
= = C	1





= C

Preconditions

= A

= A

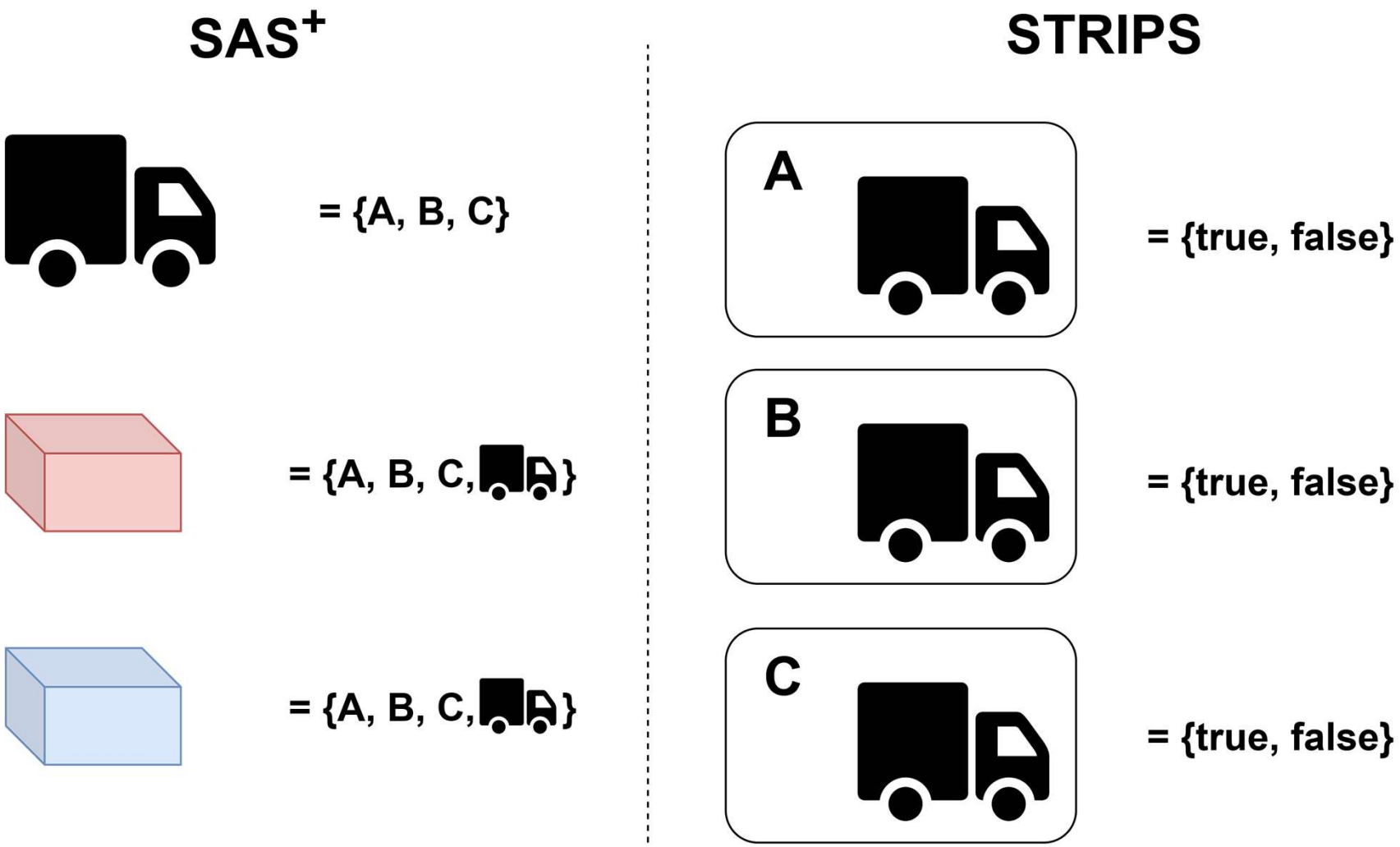
Effects

=

Optimizations to Existing Implementation

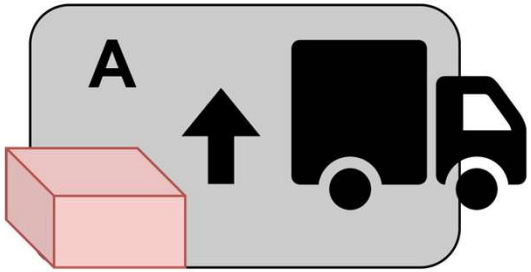
- Store Evaluation of Operator Preconditions
- Optimize Data Structures (std::unordered_map)
- Introduce Operator Queue
- Cache Operator Information in Sorted Order
- Track List of Changed Entries

Propositional Variables



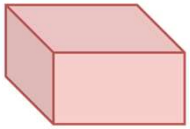
STRIPS Operators

SAS$^+$



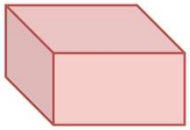
Preconditions

= A



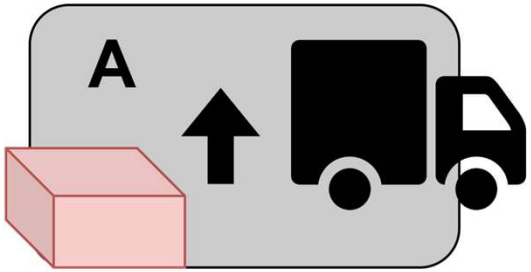
= A

Effects

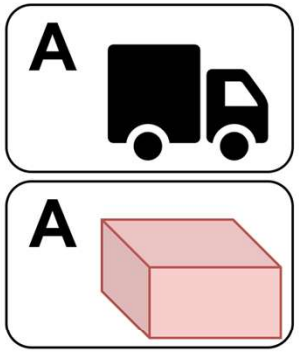


=

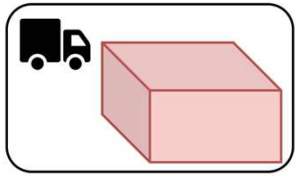
STRIPS



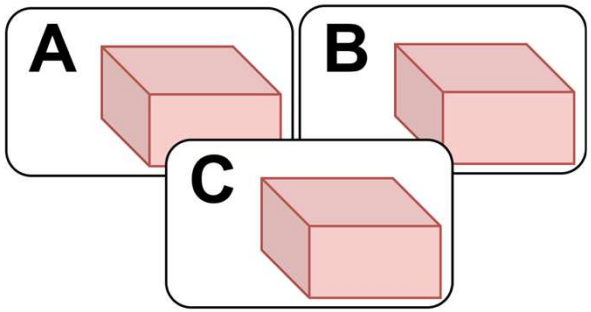
Preconditions



Add Effects

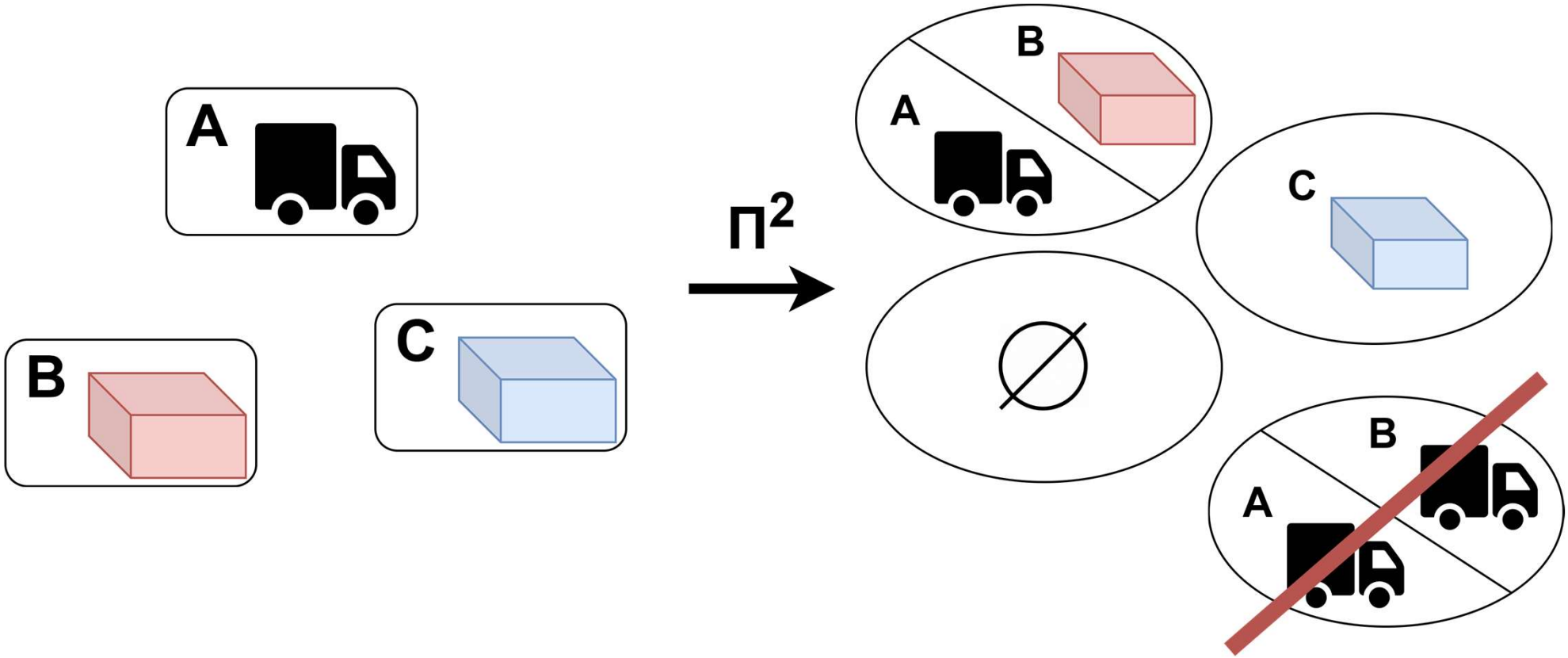


Delete Effects

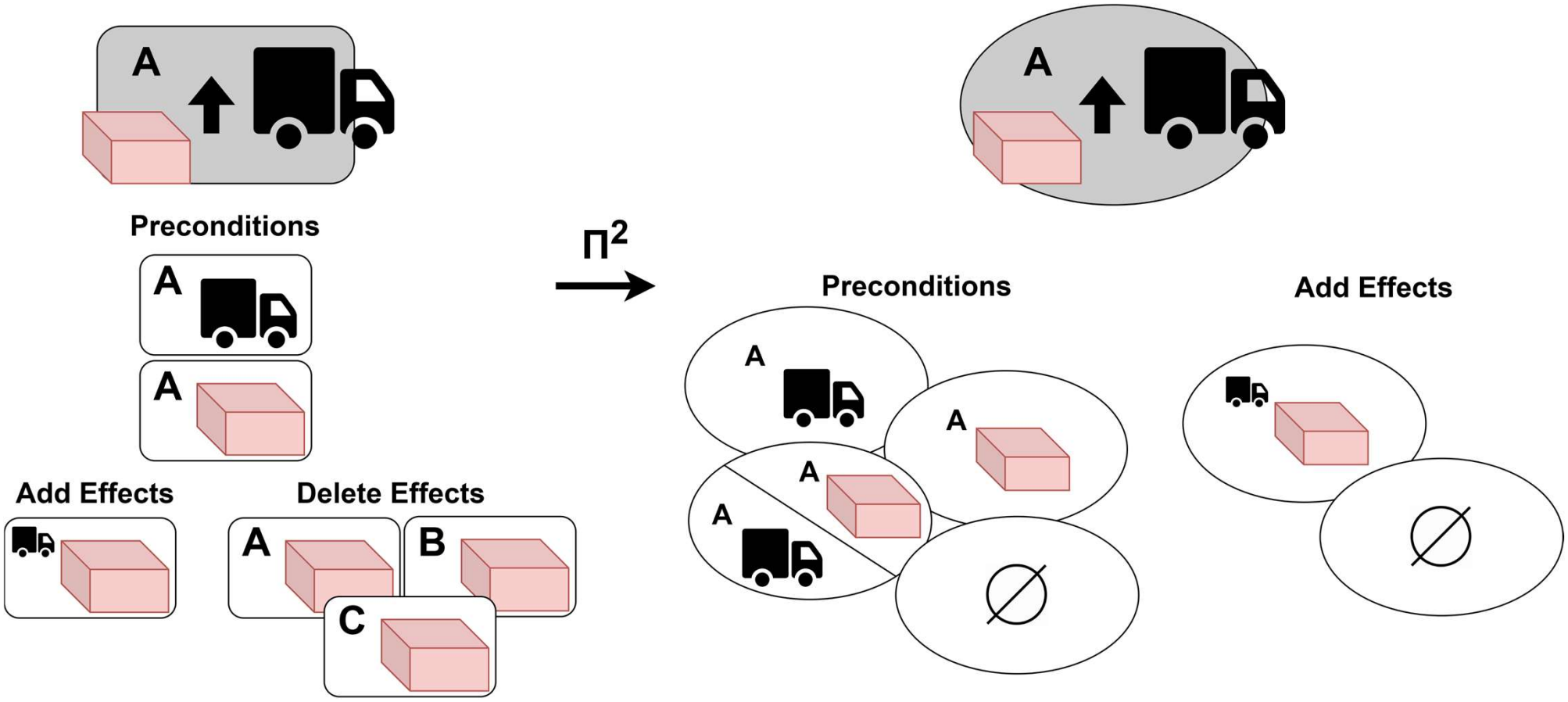


Introducing Meta-Variables

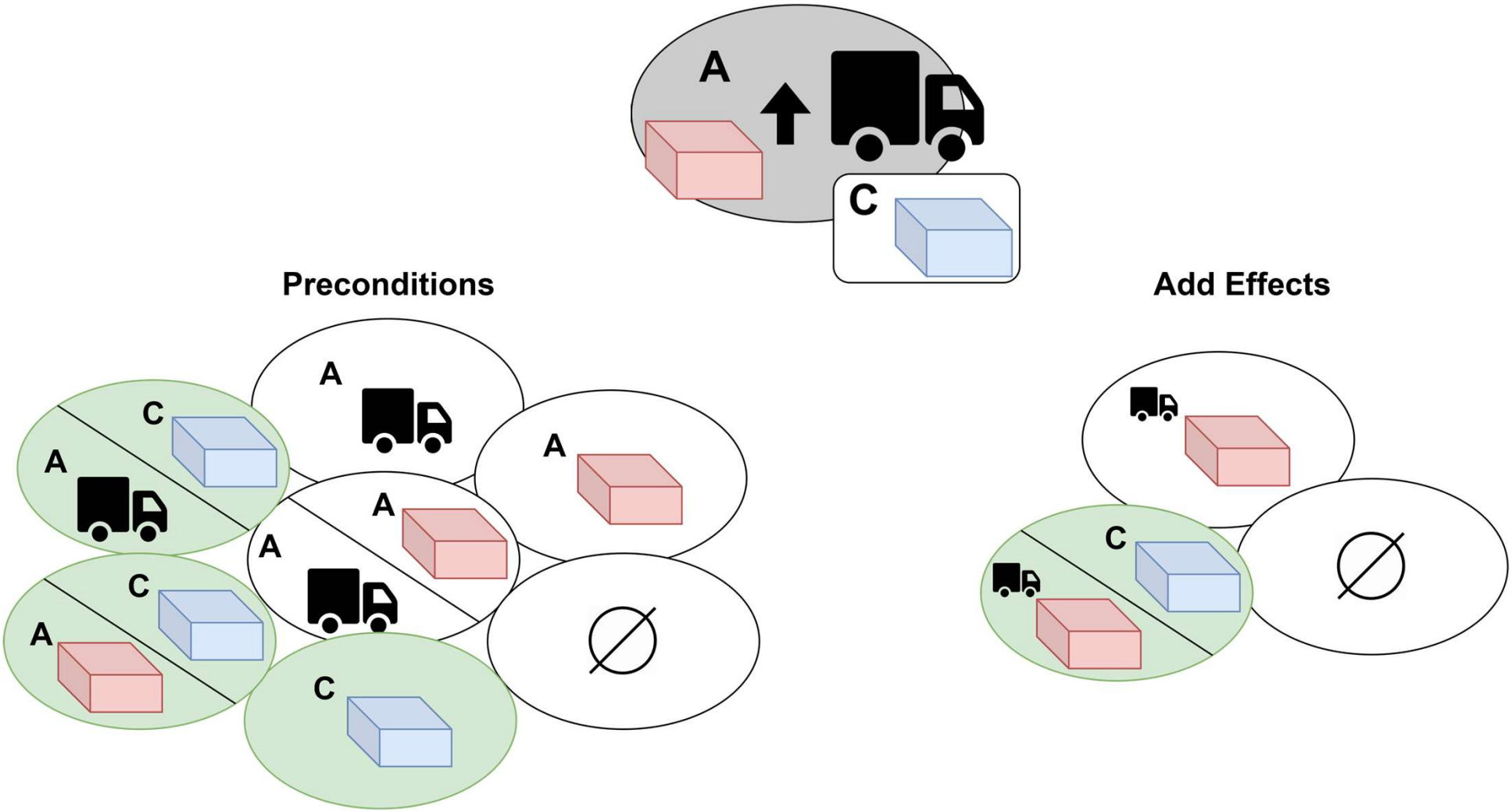
- $h^{\max}(\Pi^2) = h^2(\Pi)$



Introducing Meta-Operators

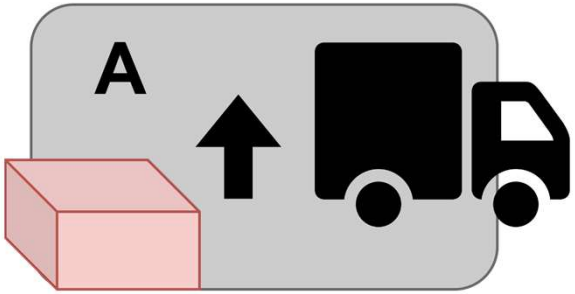


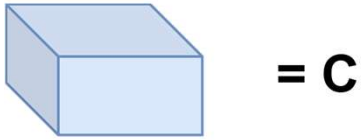
Introducing Meta-Operators



Comparison Operator Evaluation

Heuristic Table	
= A	0
= A = A	0
= A = B	∞
= A = C	∞
= A =	1
= A = A	∞
= A	0
=	1
= = C	1

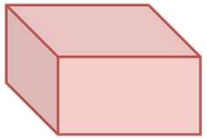




= C

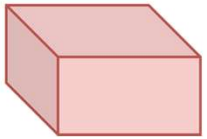
Preconditions

= A



= A

Effects



= 

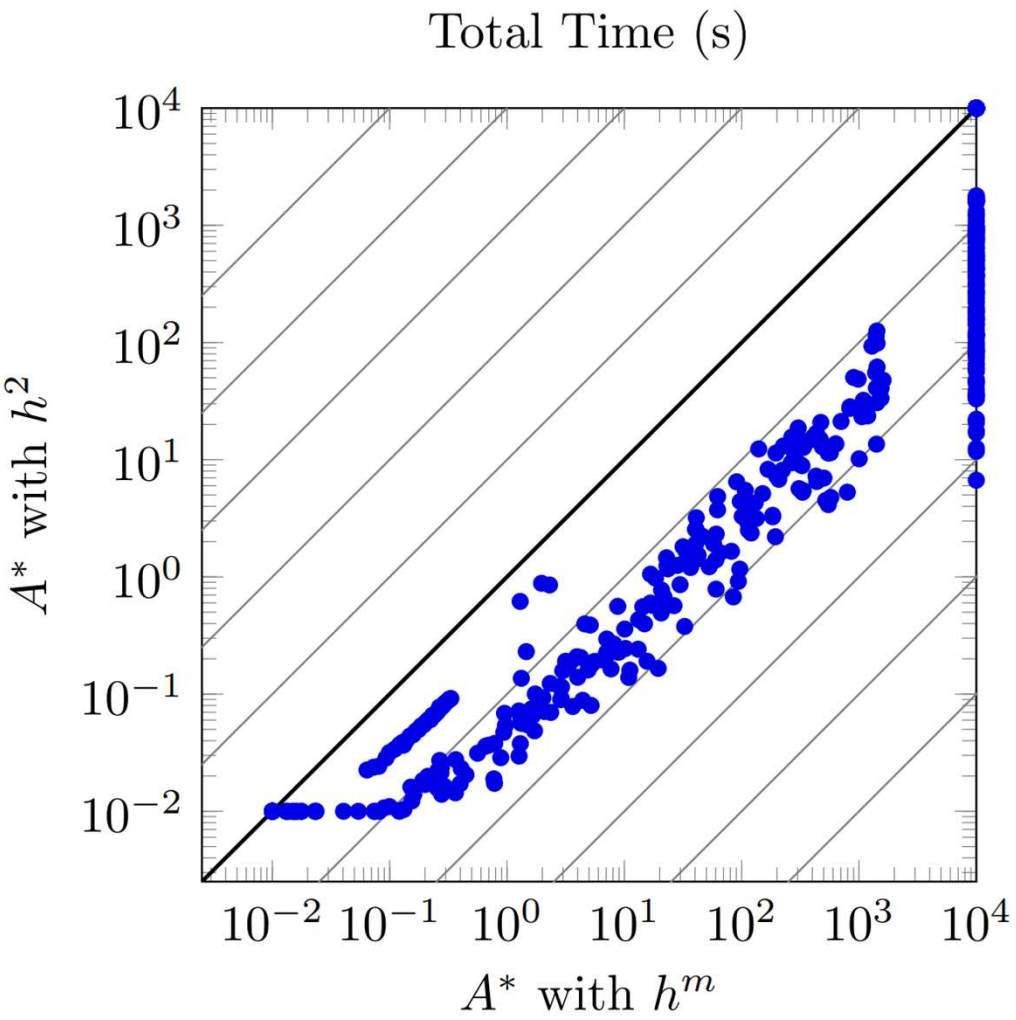
Π^2 Compilation

- Combine STRIPS Transformation with Π^2 Compilation
- Compile Task once in Preprocessing Step
- Use Existing $h^{\max}$ Implementation
- Unary Operators Require Duplicate Detection

1	Introduction
2	Implementation of h^2 Heuristic
3	SAS^+ to STRIPS Transformation
4	Π^2 Compilation
5	Results & Conclusions

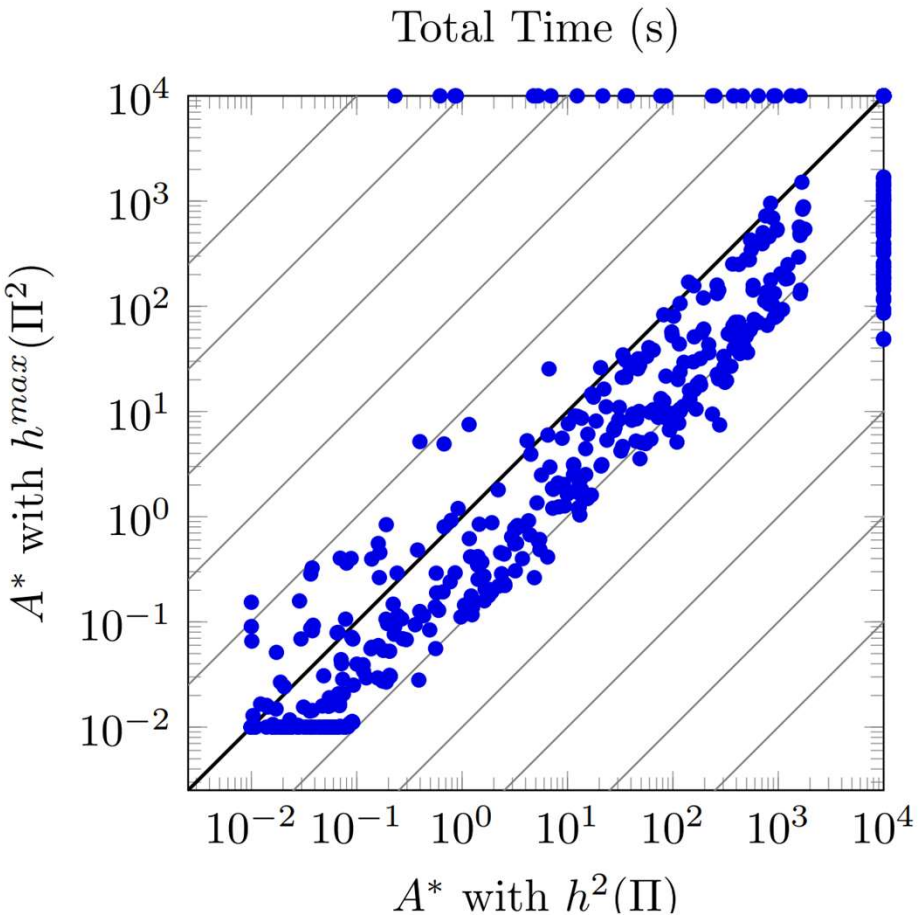
Results & Conclusions

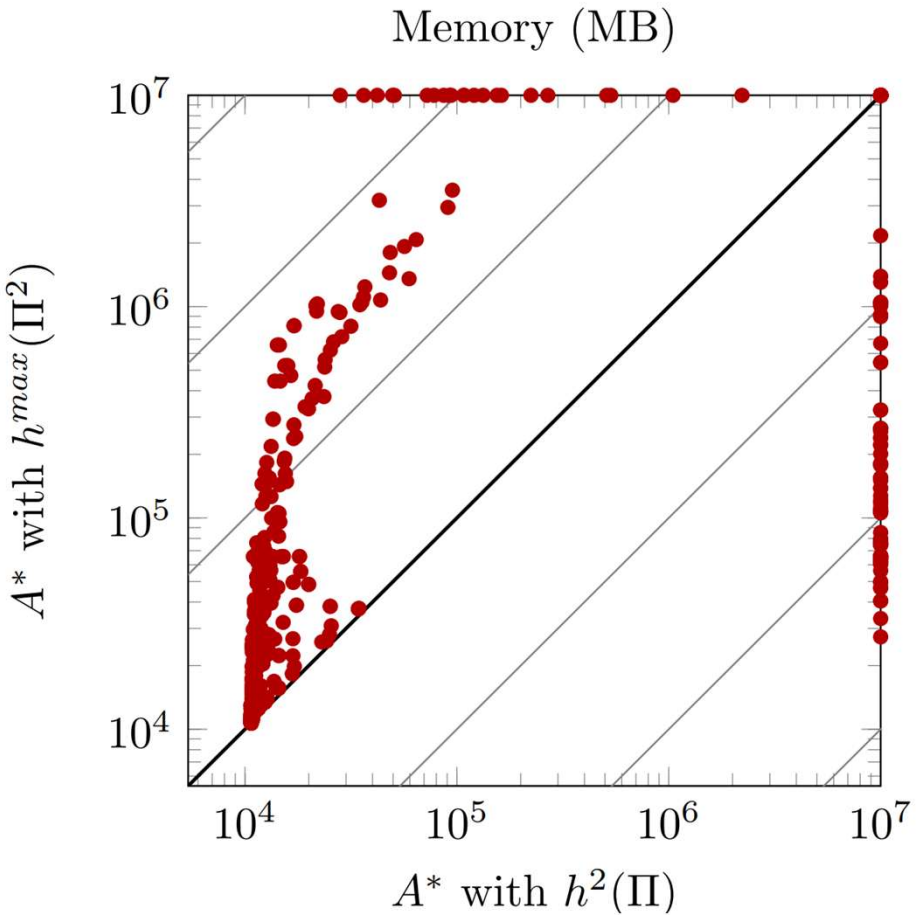
- Optimized h^2 Clearly Outperforms Existing Implementation



Results & Conclusions

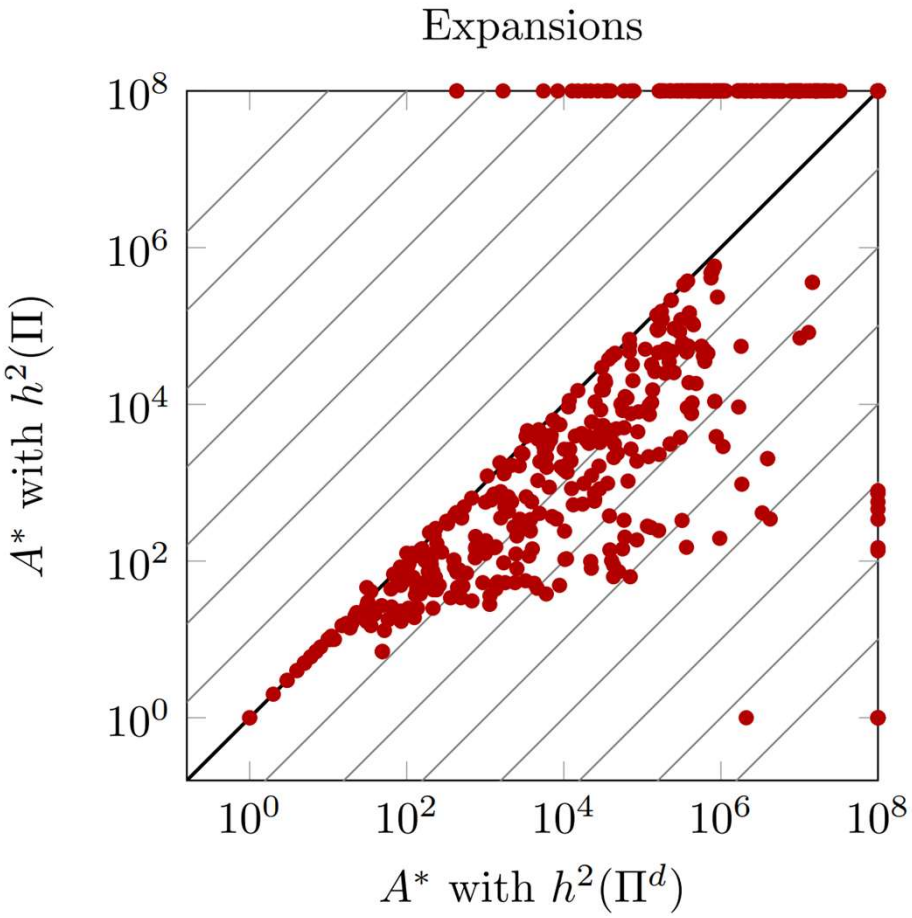
- Optimized h^2 Clearly Outperforms Existing Implementation
- Π^2 Compilation Yields Best Results Despite High Memory Consumption





Results & Conclusions

- Optimized h^2 Clearly Outperforms Existing Implementation
- Π^2 Compilation Yields Best Results Despite High Memory Consumption
- Regression Simulation With Duality Yields Uninformed Heuristic



Results & Conclusions

- Optimized h^2 Clearly Outperforms Existing Implementation
- Π^2 Compilation Yields Best Results Despite High Memory Consumption
- Regression Simulation With Duality Yields Uninformed Heuristic
- Majority of Benchmark Problems Unsolved (1827 Total Tasks)

276 Tasks	$h^m(\Pi)$ for $m = 2$	Existing Implementation
431 Tasks	$h^2(\Pi)$	Optimized h^2
493 Tasks	$h^{\max}(\Pi^2)$	Π^2 Compilation

Results & Conclusions

- Optimized h^2 Clearly Outperforms Existing Implementation
- Π^2 Compilation Yields Best Results Despite High Memory Consumption
- Regression Simulation With Duality Yields Uninformed Heuristic
- Majority of Benchmark Problems Unsolved (1827 Total Tasks)
- Regression Search Theoretically Better Suited for h^m

Progression Search: $h^m(s, G)$

Regression Search: $h^m(I, S)$

Universität Basel

Thank You for Listening.