Plan Merging in the asprilo Framework

Adrian Salewsky

University of Potsdam

24.03.2021

Table of Contents

Introduction

Introduction

- Combining plans for single robots

Introduction

- Combining plans for single robots
- Used the asprilo framework and ASP

Introduction

- Combining plans for single robots
- Used the asprilo framework and ASP
- M-Domain of asprilo

Introduction

- Combining plans for single robots
- Used the asprilo framework and ASP
- M-Domain of asprilo
- Visualizer used for testing

Introduction

Introduction

- Input: $occurs(object(robot, R), action(move, D), T)$

Introduction

- Input: $occurs(object(robot, R), action(move, D), T)$
- Move predicates: $move(R, D, T)$

Introduction

- Input: $occurs(object(robot, R), action(move, D), T)$
- Move predicates: $move(R, D, T)$
- Position predicates: $position(R, C, T)$

Renaming of Predicates

Renaming of Predicates

- New names needed for every plan

Renaming of Predicates

- New names needed for every plan
- $\rightarrow$ New argument for every predicate: *conflict_nr*

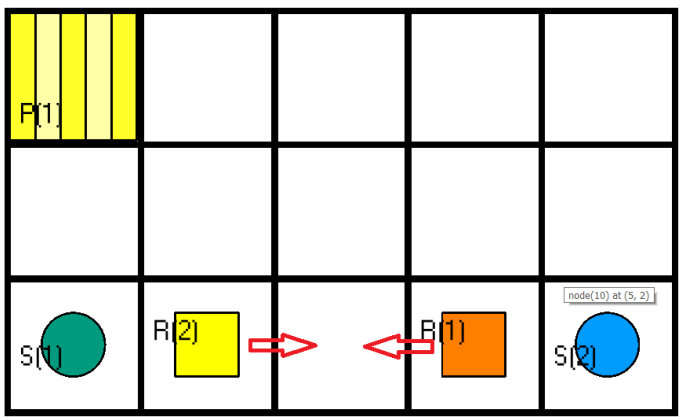
Renaming of Predicates

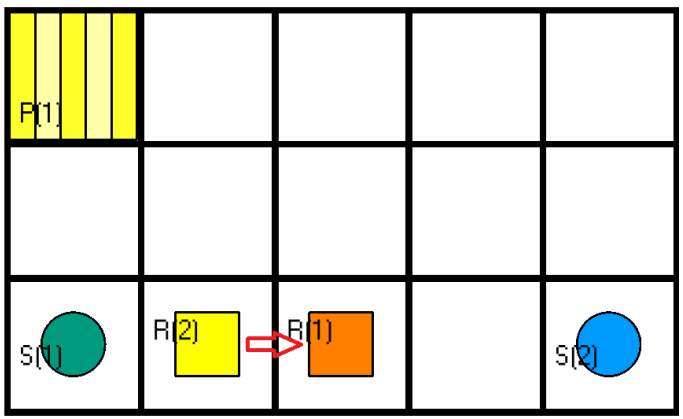
- New names needed for every plan
- $\rightarrow$ New argument for every predicate: *conflict_nr*
- Higher *conflict_nr* $\rightarrow$ newer plan

Renaming of Predicates

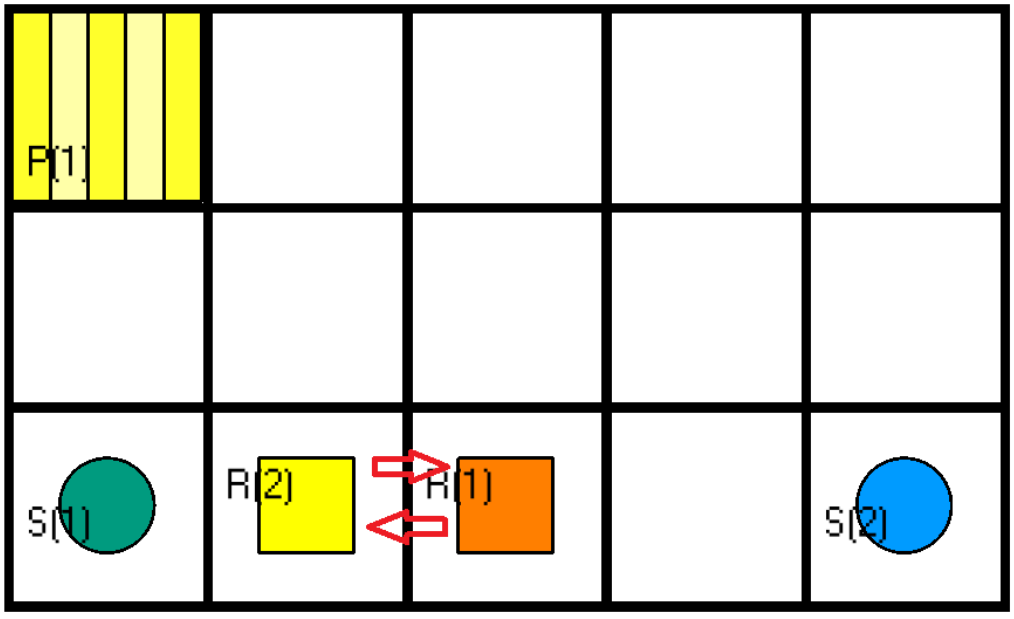
- New names needed for every plan
- $\rightarrow$ New argument for every predicate: *conflict_nr*
- Higher *conflict_nr* $\rightarrow$ newer plan
- $move(R, D, T) \rightarrow move(R, D, T, A)$

Conflict Detection and Selection





Conflict Detection and Selection



Conflict Solving

Conflict Solving

- Randomly dodge in any possible direction or wait

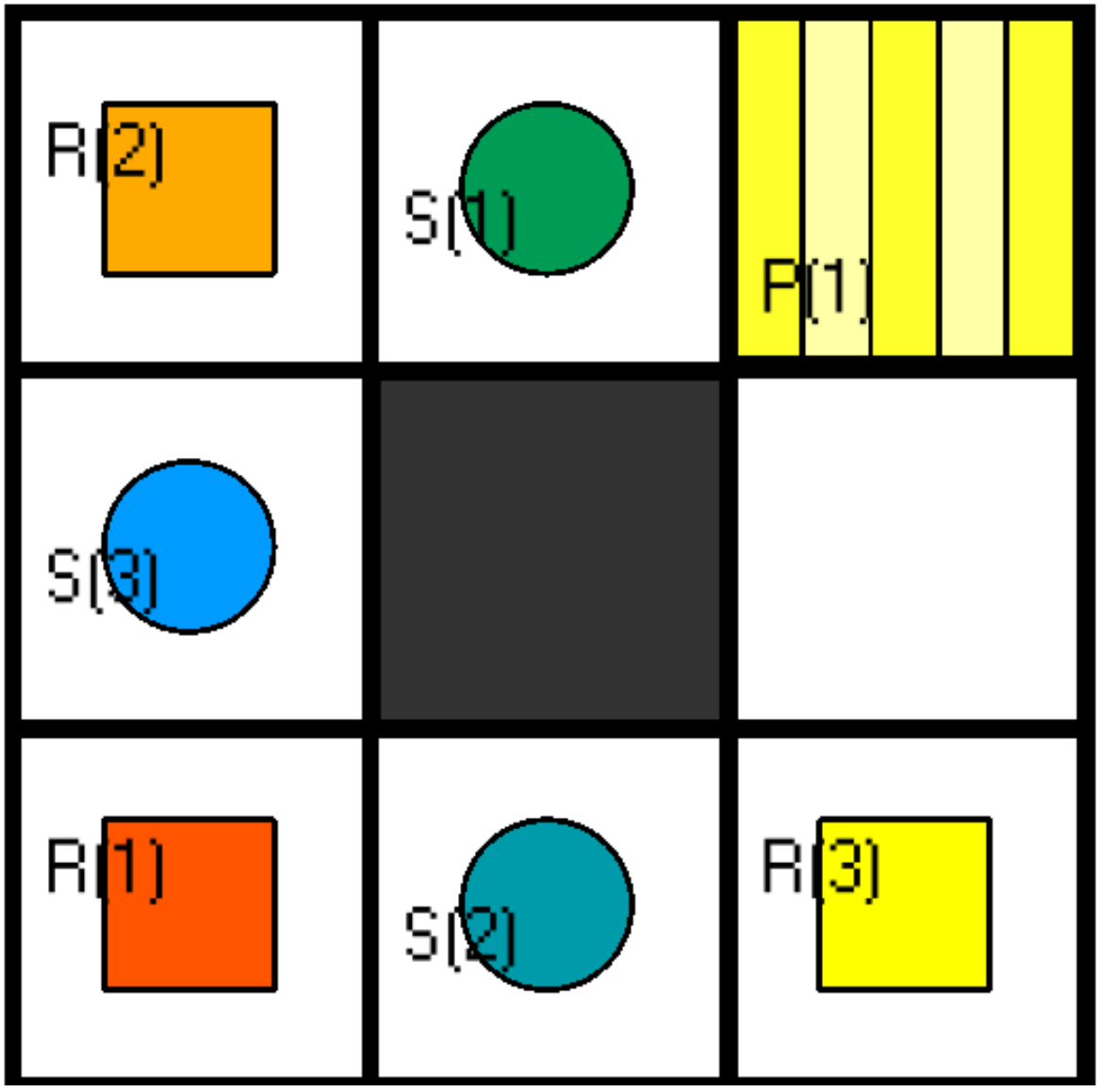
Conflict Solving

- Randomly dodge in any possible direction or wait
- Wait: remaining plan gets pushed back one step

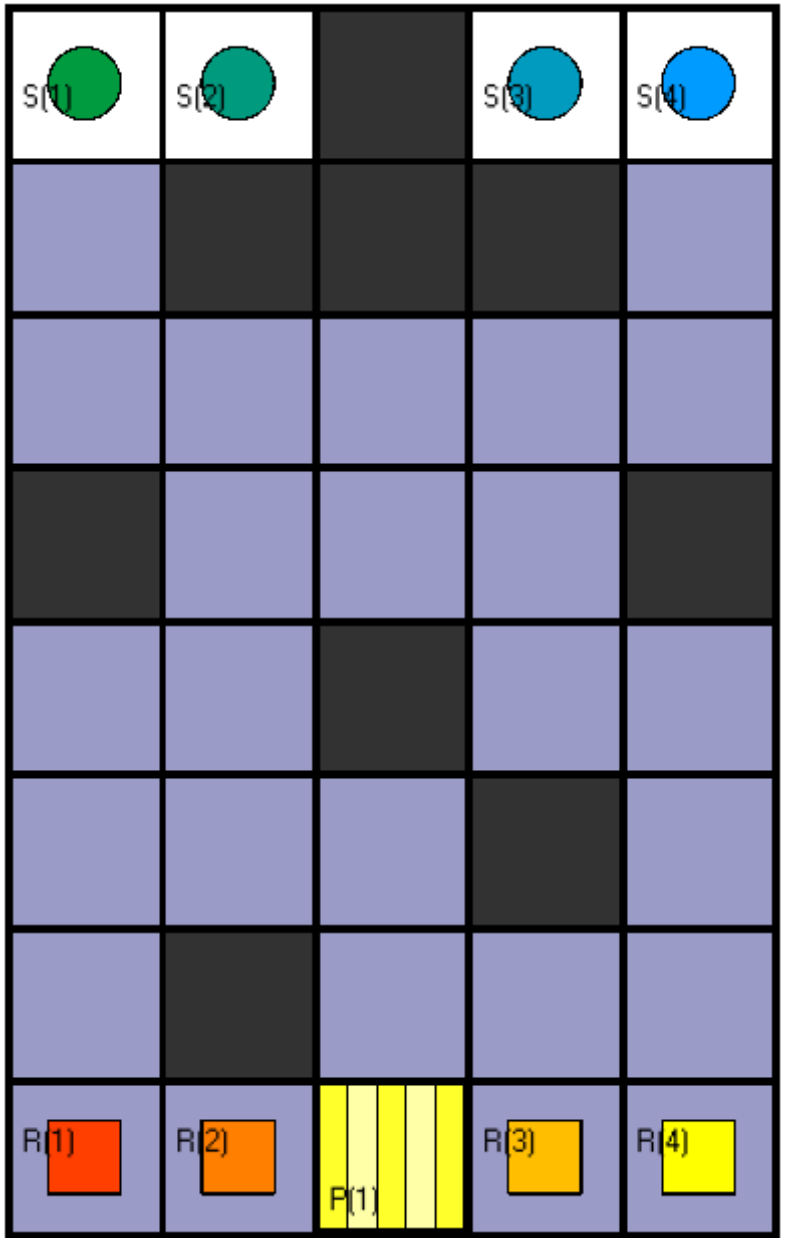
Conflict Solving

- Randomly dodge in any possible direction or wait
- Wait: remaining plan gets pushed back one step
- Dodge: go back at random time step

Unsolvable Benchmarks



Unsolvable Benchmarks



Unsolvable Benchmarks

Comparison with other Approaches

Comparison with other Approaches

- Very time inefficient

Comparison with other Approaches

- Very time inefficient
- Most other approaches found more efficient solutions

Comparison with other Approaches

- Very time inefficient
- Most other approaches found more efficient solutions
- Solves most instances

Comparison with other Approaches

- Very time inefficient
- Most other approaches found more efficient solutions
- Solves most instances
- One other approach better in every way

Conclusion

Conclusion

- Combining plans of single robots

Conclusion

- Combining plans of single robots
- Randomly solve arising conflicts

Conclusion

- Combining plans of single robots
- Randomly solve arising conflicts
- Time inefficient

Conclusion

- Combining plans of single robots
- Randomly solve arising conflicts
- Time inefficient
- Solves most instances