Task & Motion Planning

Fabien Lagriffoul

Centre for Applied Autonomous Sensor Systems (AASS)
Örebro University, Sweden
MRPC Lab

Outline

- Problem definition(s)
- What makes it difficult?
- 3 approaches to TAMP
- Future directions

Outline

- Problem definition(s)
- What makes it difficult?
- 3 approaches to TAMP
- Future directions

Problem definition(s)

Problem definition(s)

TAMP: computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

Problem definition(s)

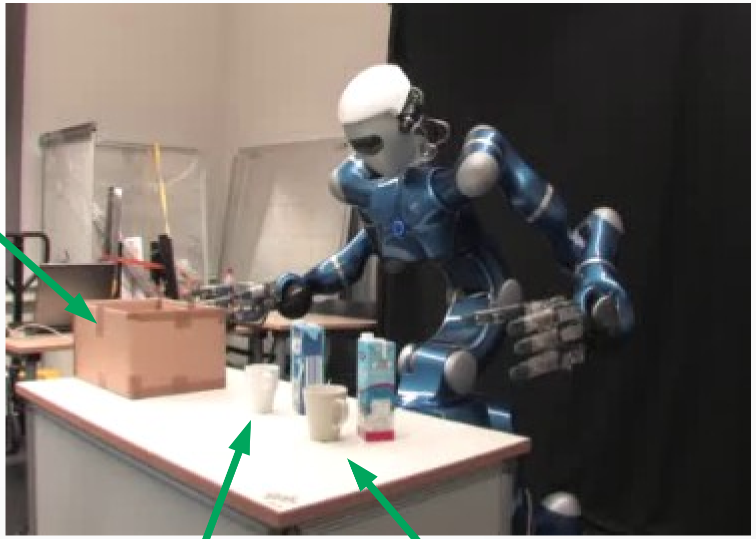
TAMP: computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

Problem definition(s)

TAMP: computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

in (box1, cup1)
in (box1, cup2)

box1



cup1

cup2

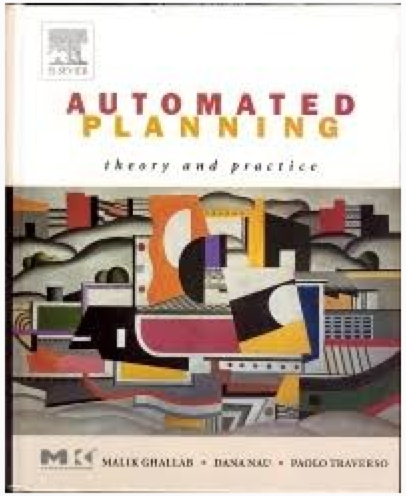
Problem definition(s)

TAMP: computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

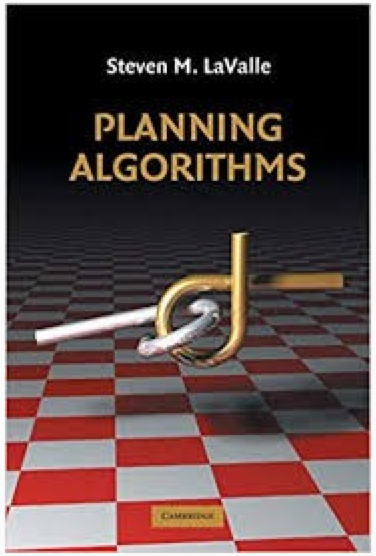
in (box1, cup1)
in (box1, cup2)

Problem definition(s)

TAMP: computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.



Task Planning



Motion Planning

Problem definition(s)

TAMP: computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

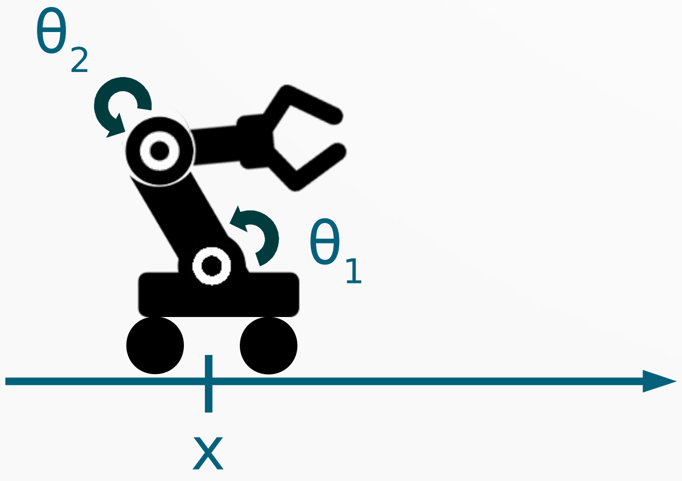
Configuration Space:

Problem definition(s)

TAMP: computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

Configuration Space:

$$q = (x, \theta_1, \theta_2)$$

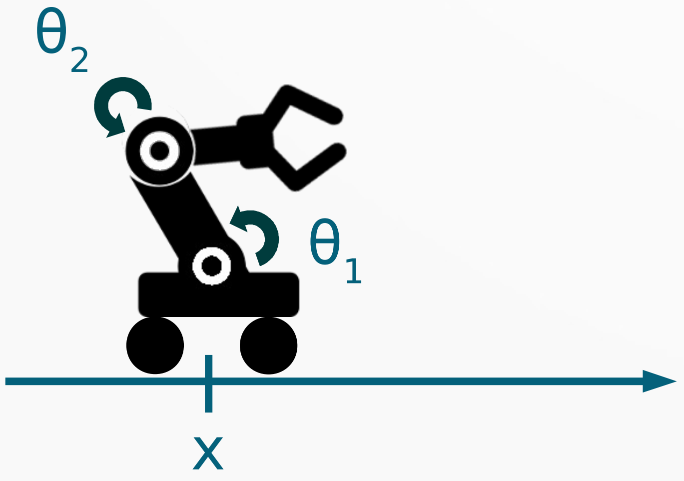


Problem definition(s)

<u>TAMP:</u> computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

<u>Configuration Space:</u>

$q = (x, \theta_1, \theta_2)$ $q = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$

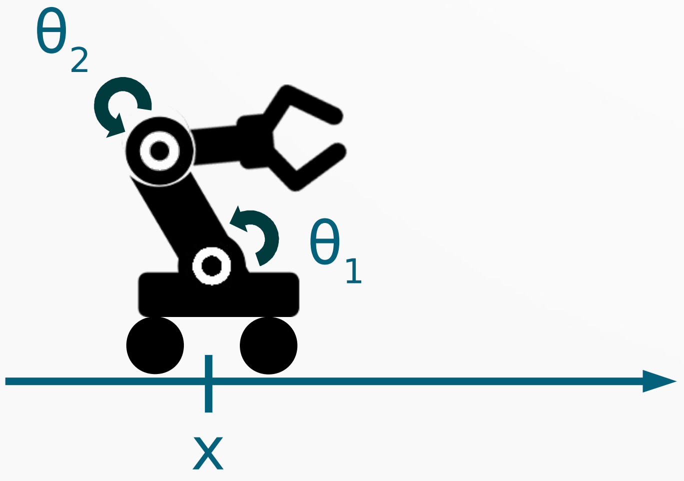


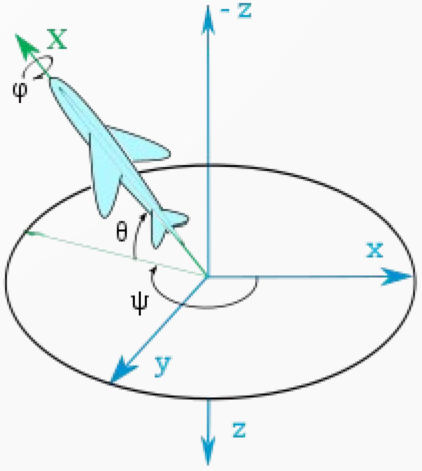
Problem definition(s)

<u>TAMP:</u> computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

<u>Configuration Space:</u>

$q = (x, \theta_1, \theta_2)$ $q = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ $q = (x, y, z, \theta, \varphi, \psi)$



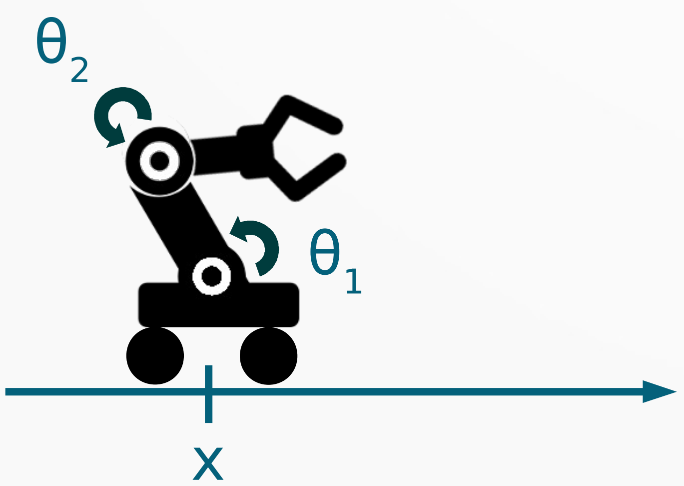


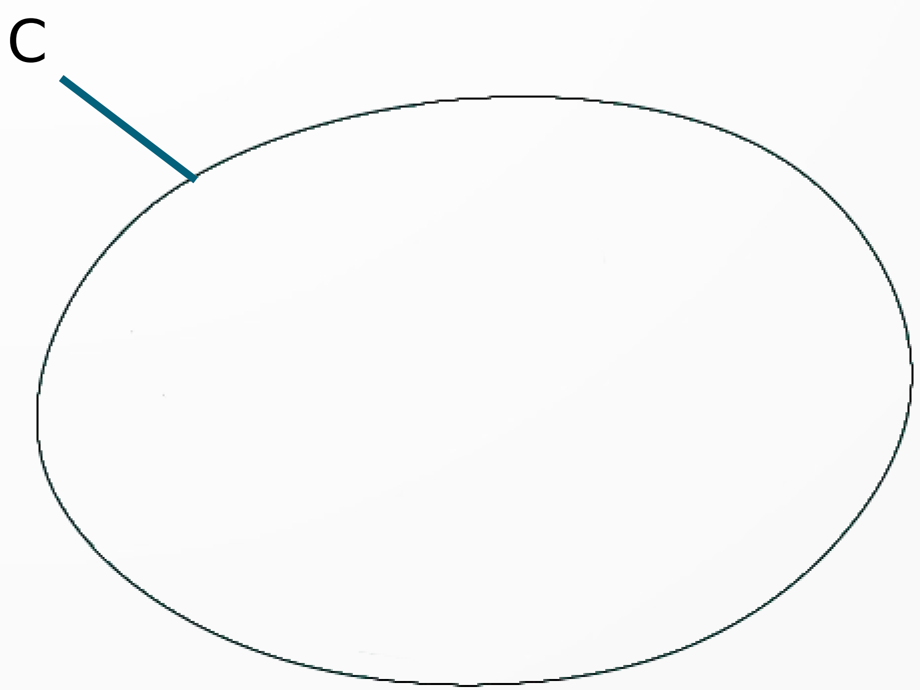
Problem definition(s)

TAMP: computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

Configuration Space:

$$q = (x, \theta_1, \theta_2)$$



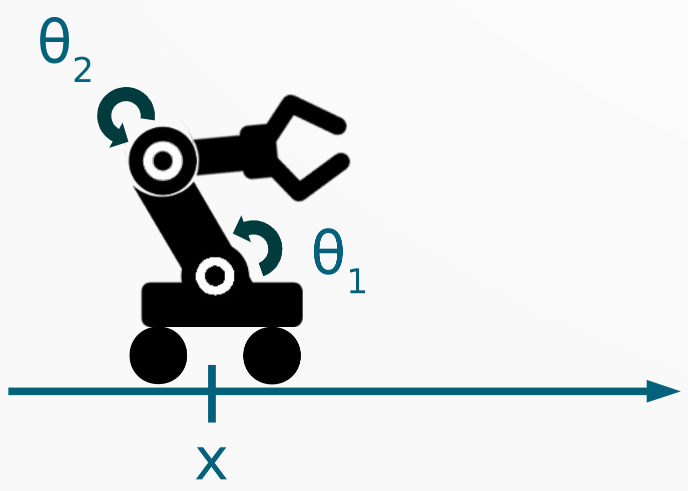


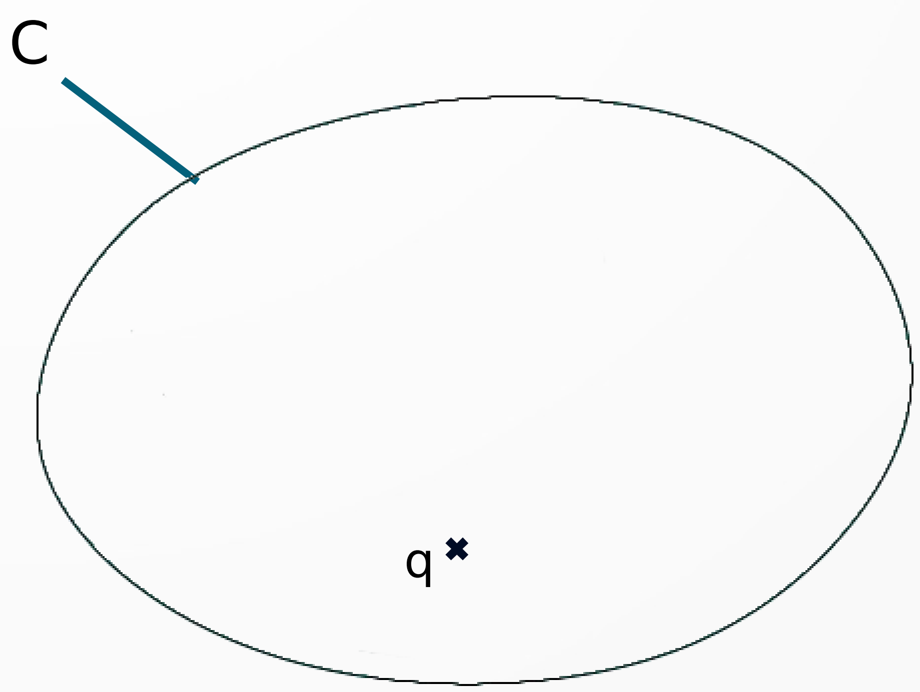
Problem definition(s)

TAMP: computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

Configuration Space:

$$q = (x, \theta_1, \theta_2)$$





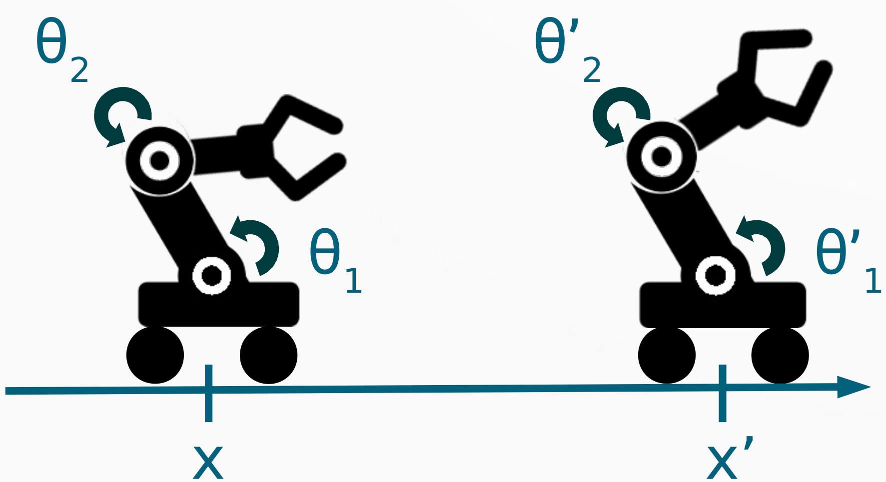
Problem definition(s)

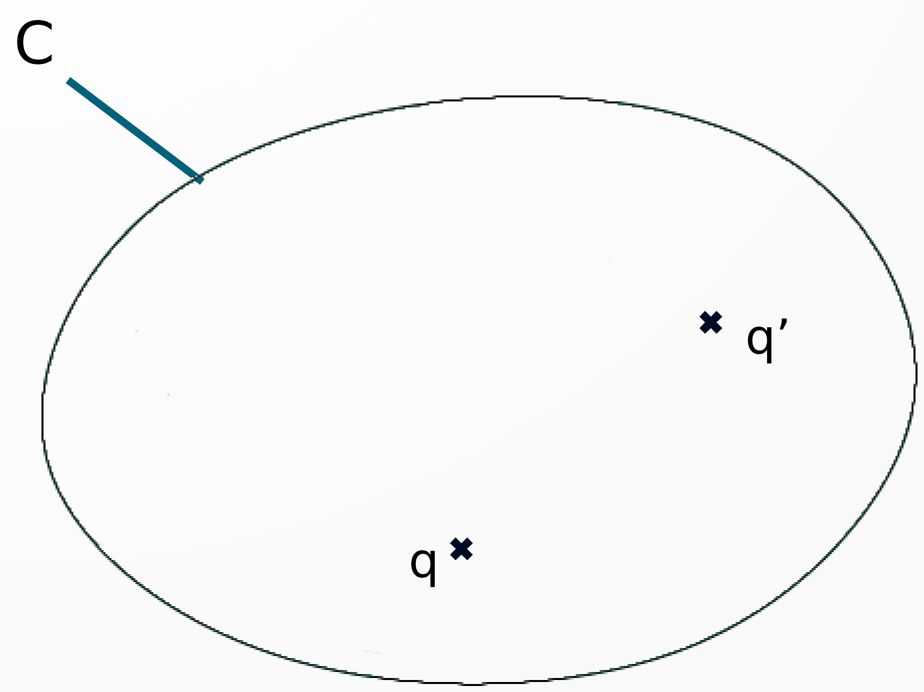
<u>TAMP:</u> computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

<u>Configuration Space:</u>

$q = (x, \theta_1, \theta_2)$

$q' = (x', \theta'_1, \theta'_2)$





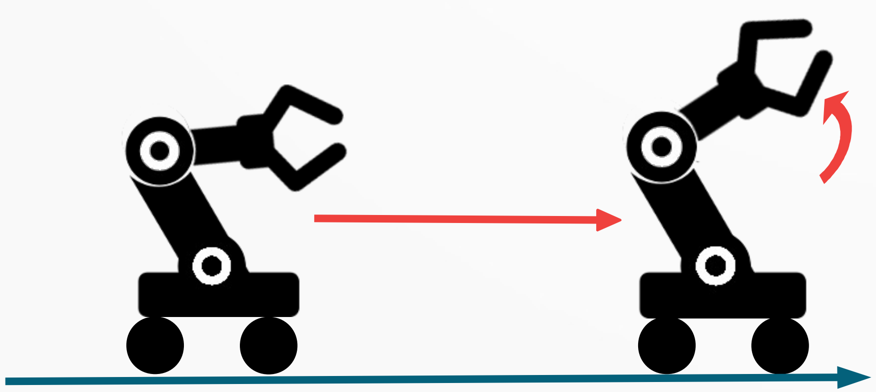
Problem definition(s)

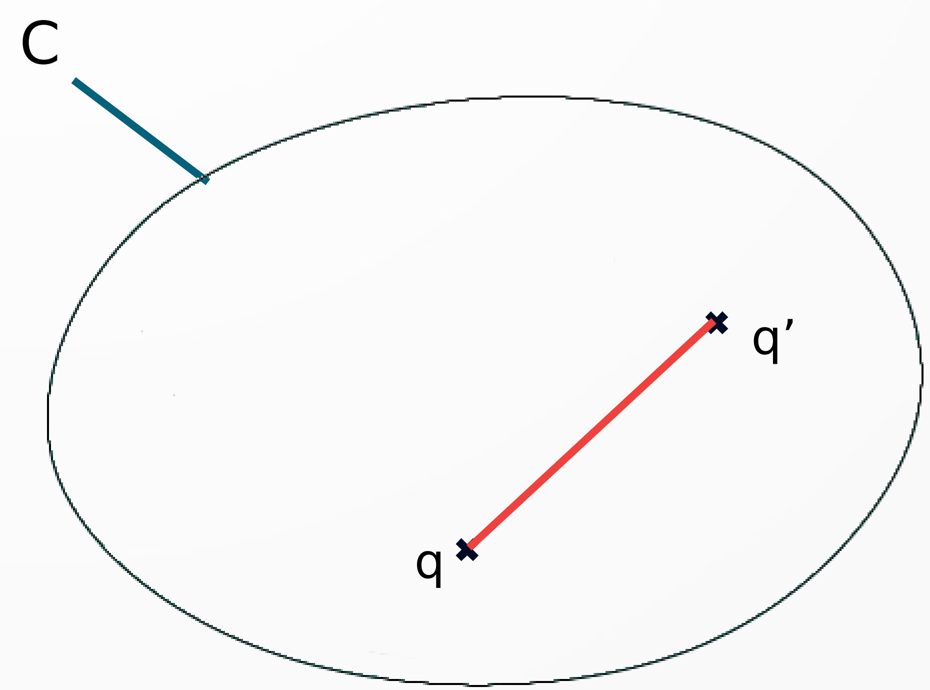
<u>TAMP:</u> computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

<u>Configuration Space:</u>

$$q = (x, \theta_1, \theta_2)$$

$$q' = (x', \theta'_1, \theta'_2)$$





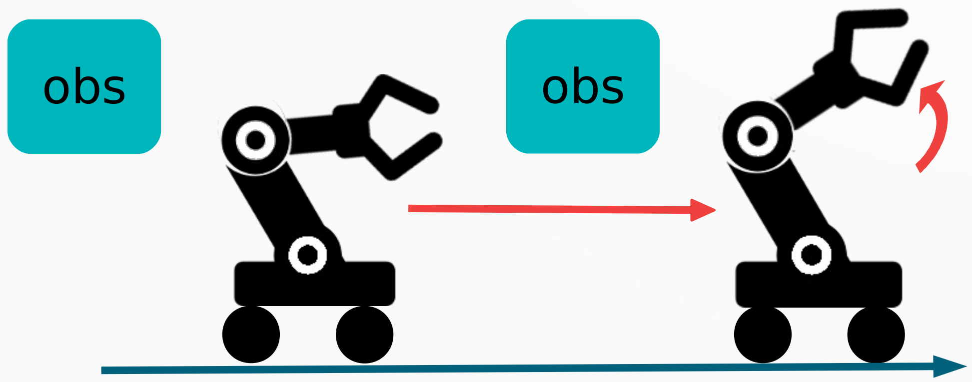
Problem definition(s)

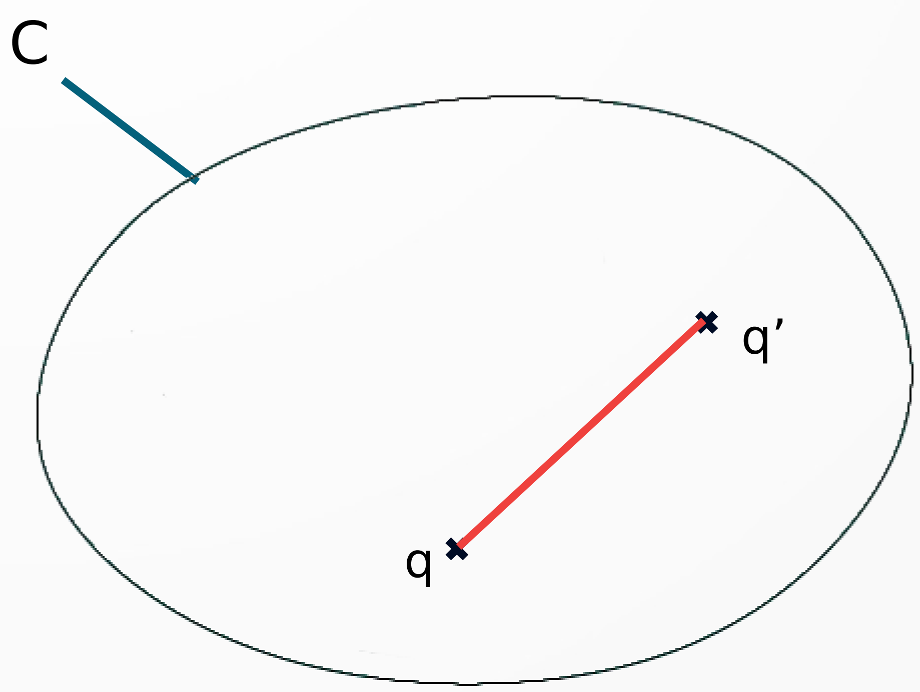
TAMP: computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

Configuration Space:

$q = (x, \theta_1, \theta_2)$

$q' = (x', \theta'_1, \theta'_2)$





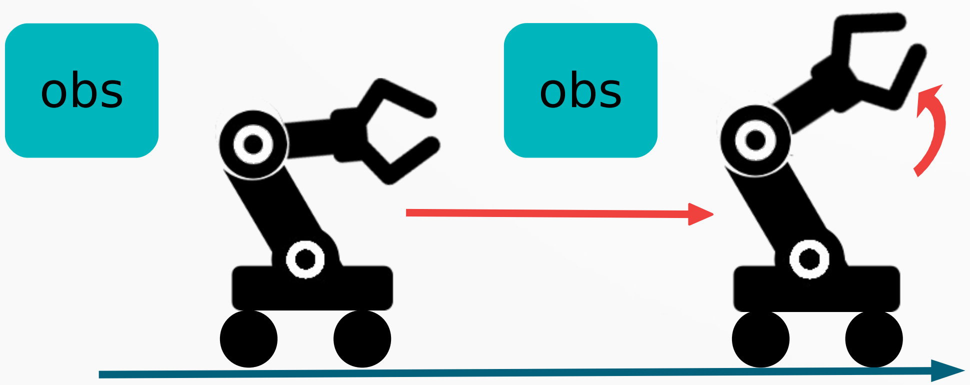
Problem definition(s)

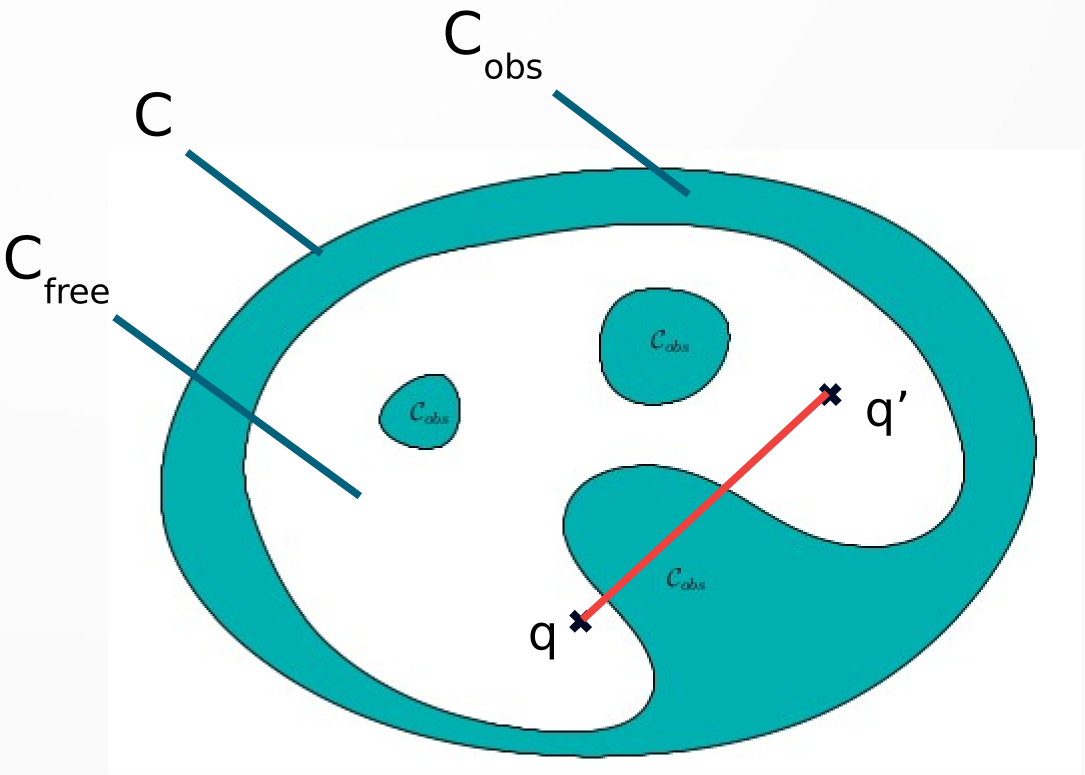
<u>TAMP:</u> computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

<u>Configuration Space:</u>

$$q = (x, \theta_1, \theta_2)$$

$$q' = (x', \theta'_1, \theta'_2)$$





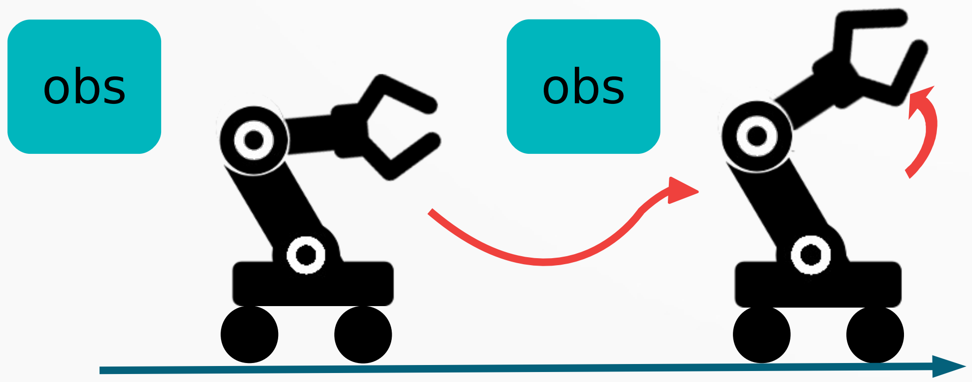
Problem definition(s)

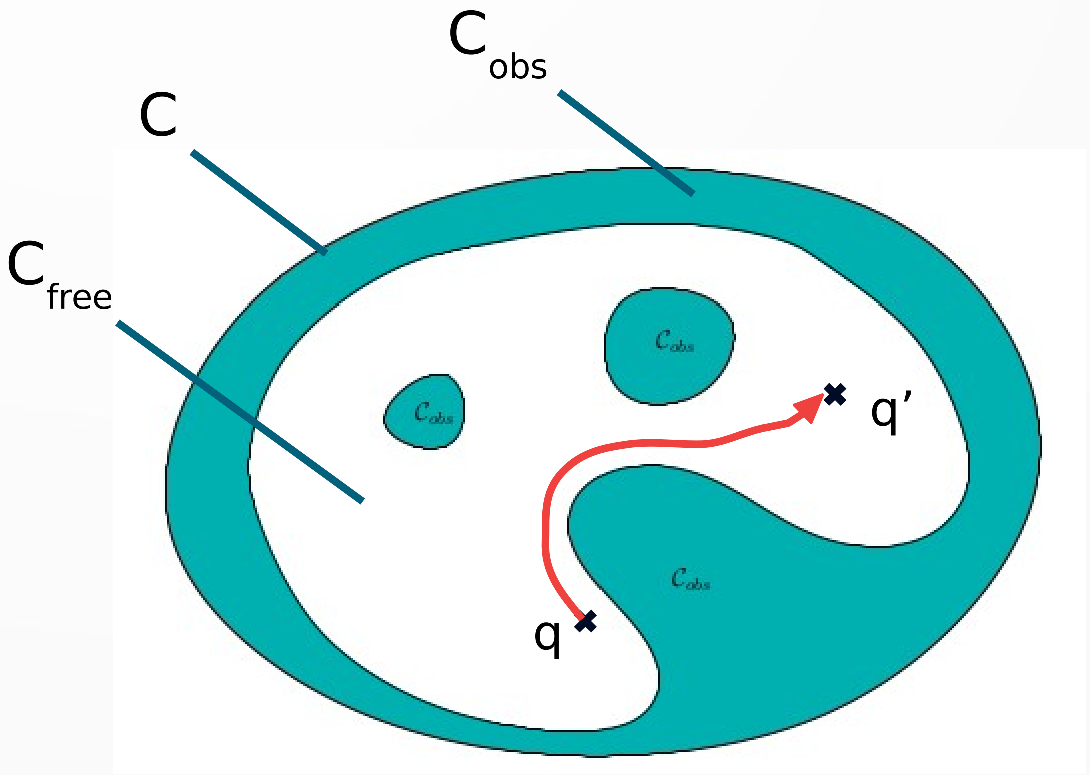
<u>TAMP:</u> computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

<u>Configuration Space:</u>

$$q = (x, \theta_1, \theta_2)$$

$$q' = (x', \theta'_1, \theta'_2)$$



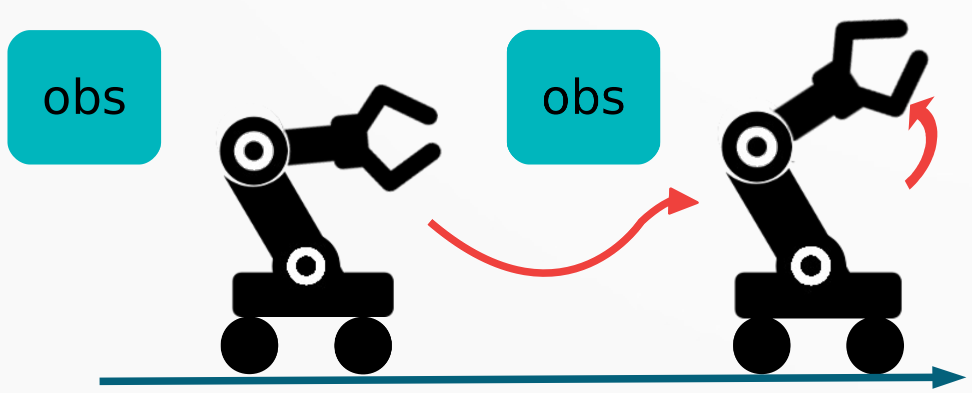


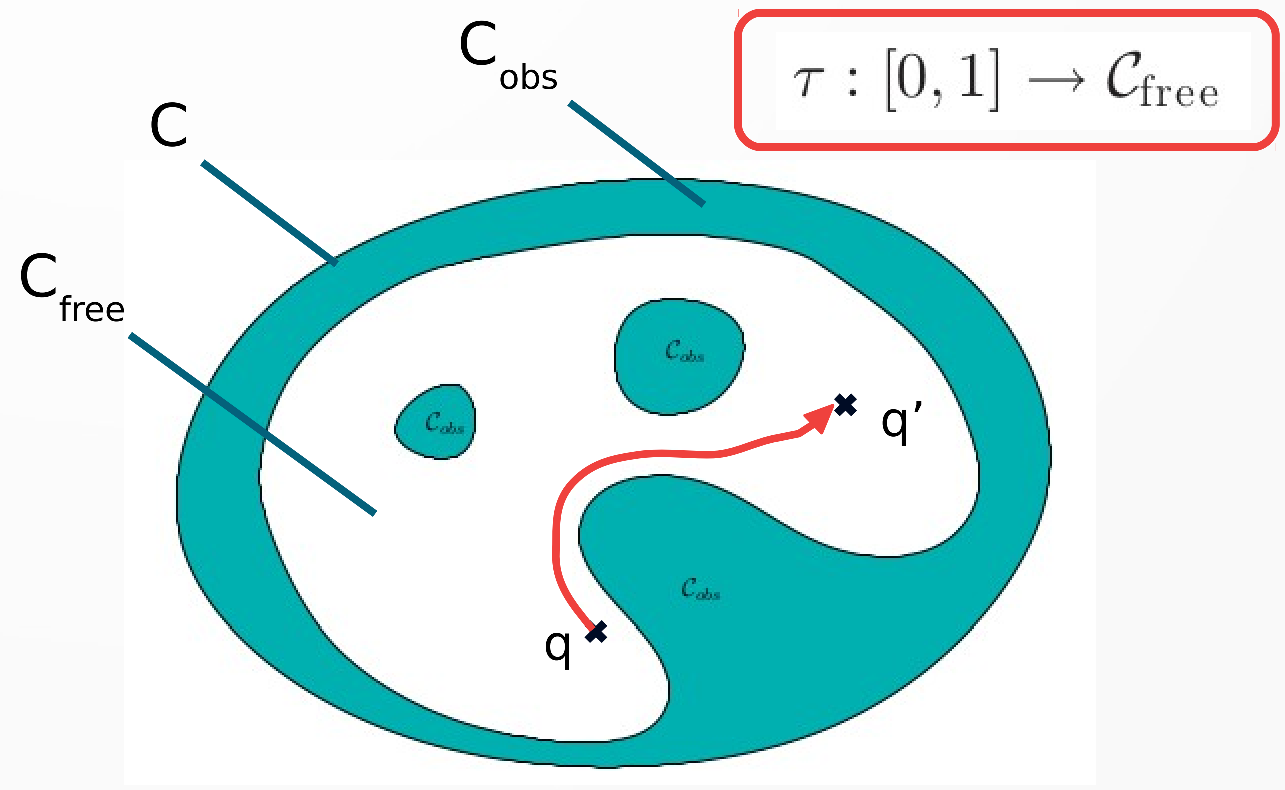
Problem definition(s)

TAMP: computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

Configuration Space:

$$q = (x, \theta_1, \theta_2)$$
$$q' = (x', \theta'_1, \theta'_2)$$





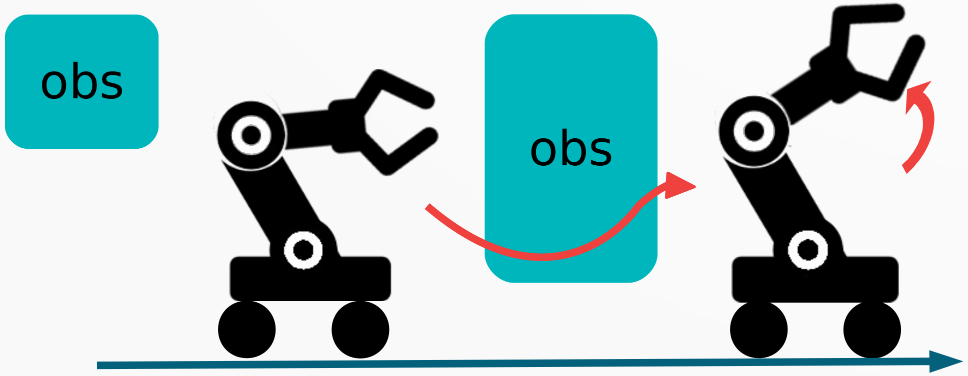
Problem definition(s)

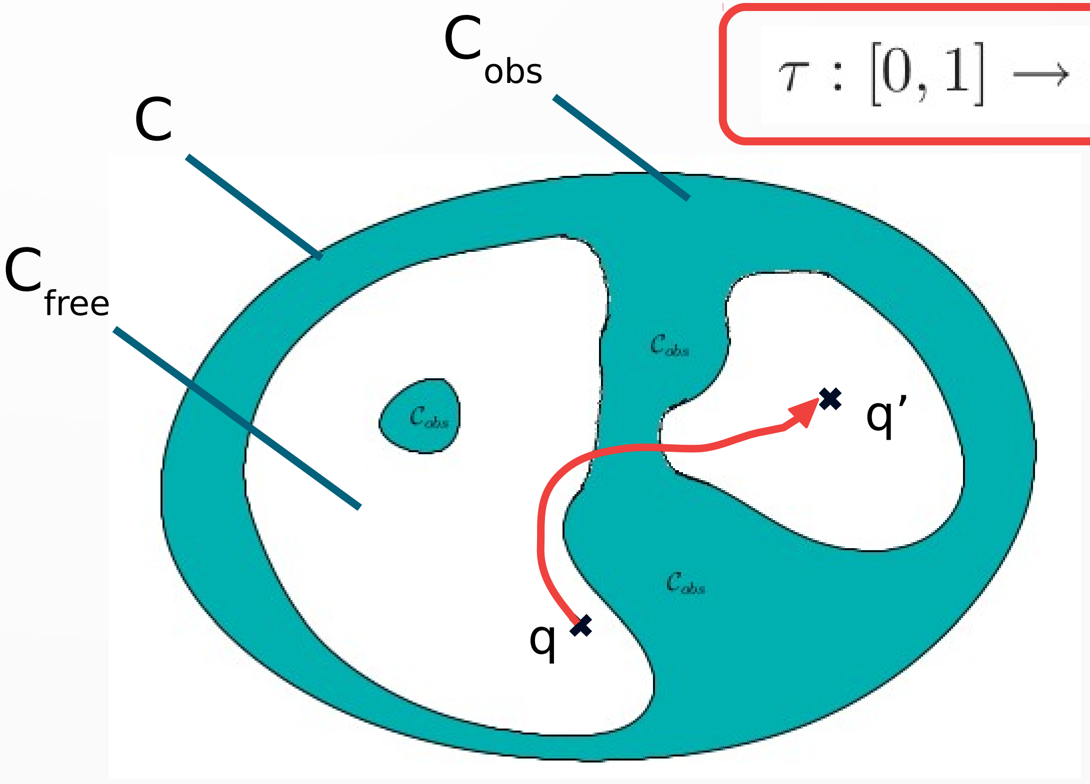
<u>TAMP:</u> computing, given a symbolic goal description, a sequence of symbolic actions **and** motion paths for one or several robots to achieve that goal.

$$\tau : [0,1] \rightarrow \mathcal{C}_{\text{free}}$$

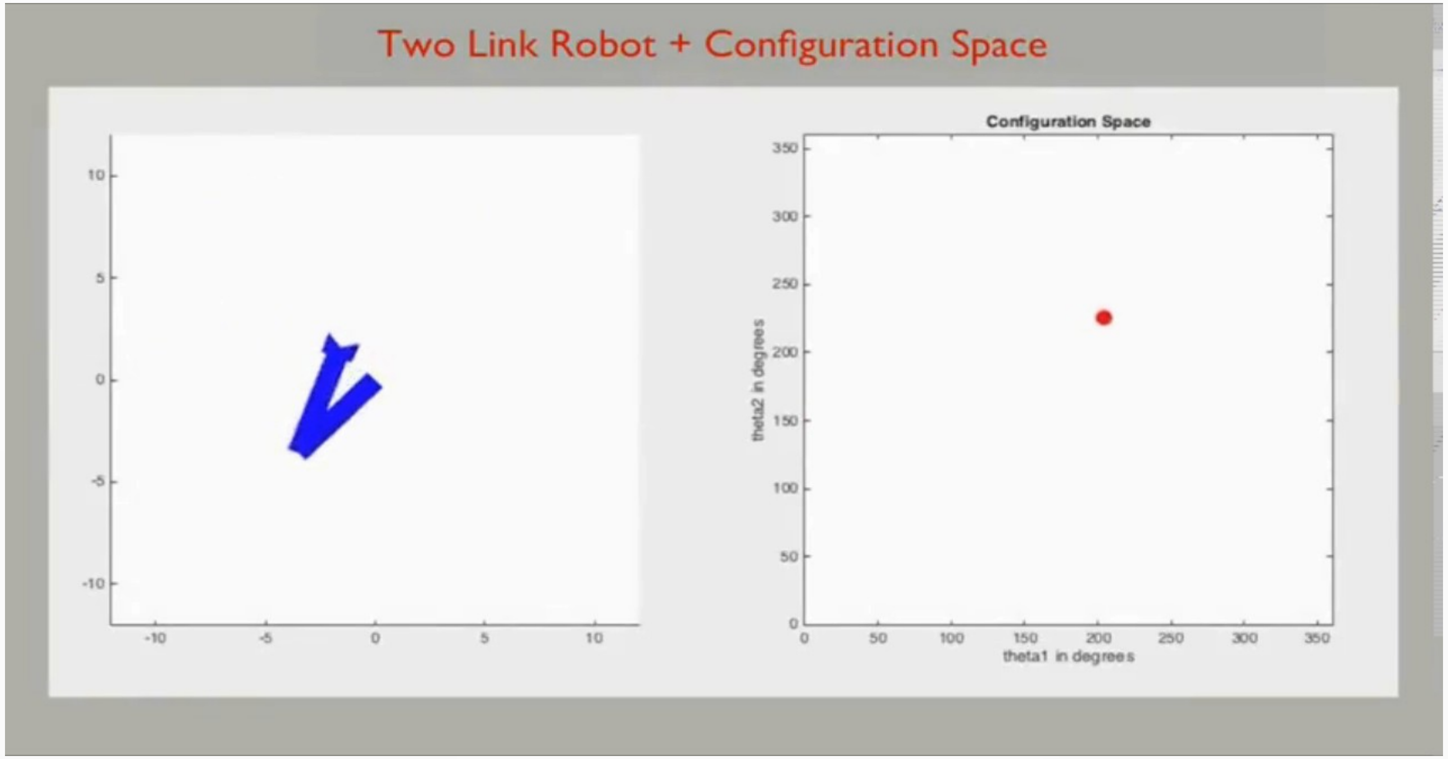
<u>Configuration Space:</u>

$$q = (x, \theta_1, \theta_2)$$
$$q' = (x', \theta'_1, \theta'_2)$$



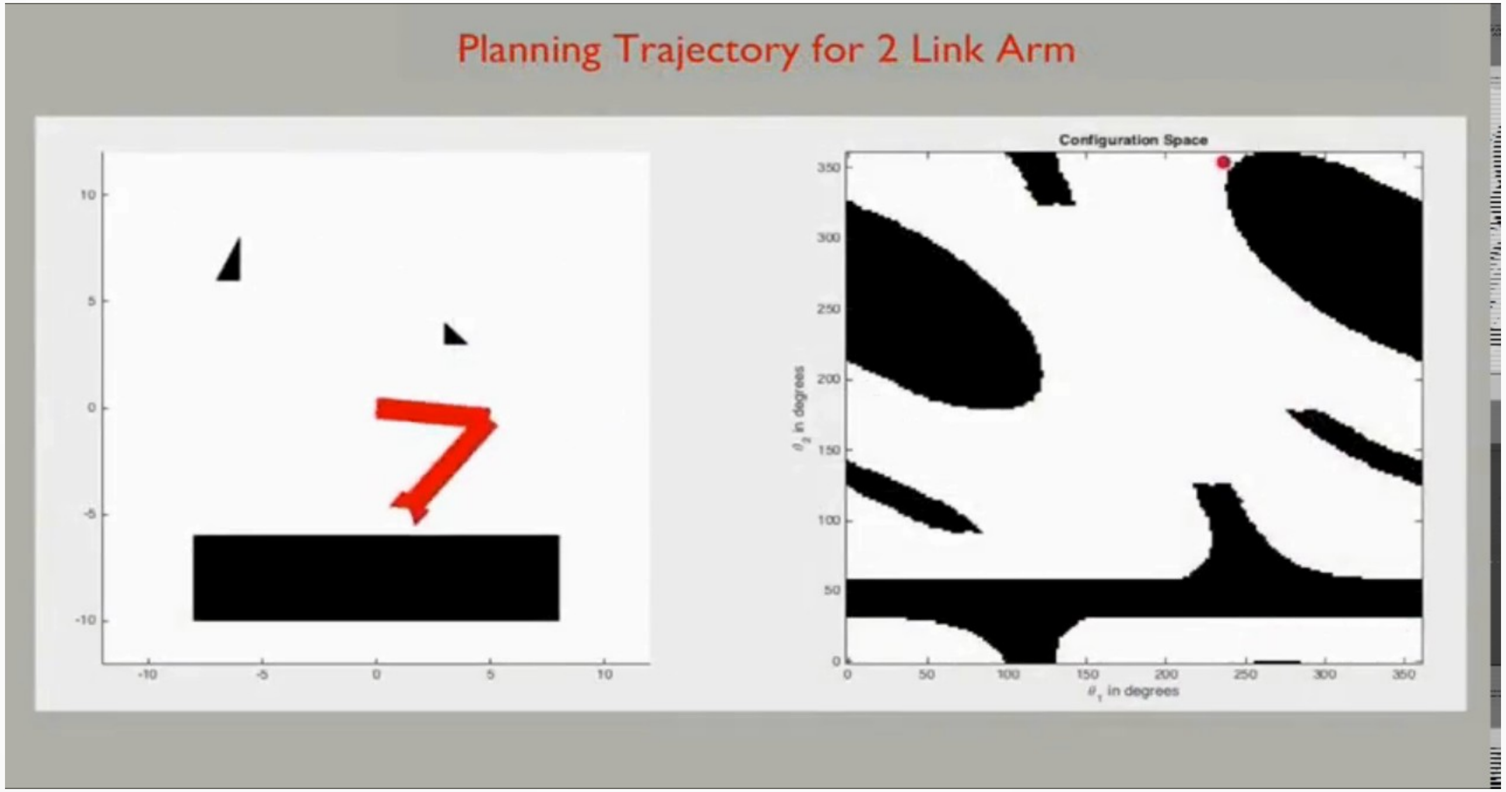


Problem definition(s)



Courtesy Bob Trenwith

Problem definition(s)



Courtesy Bob Trenwith

Problem definition(s)

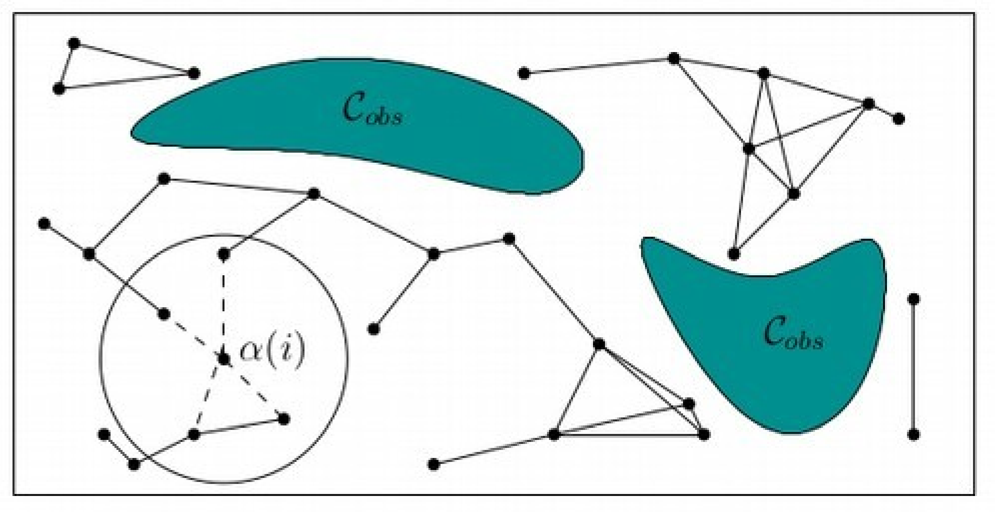
Algorithms for Motion planning:

Problem definition(s)

Algorithms for Motion planning:

- PRM (Probabilistic Roadmaps)

L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*. IEEE Transactions on Robotics & Automation, 12(4):566–580, June 1996.

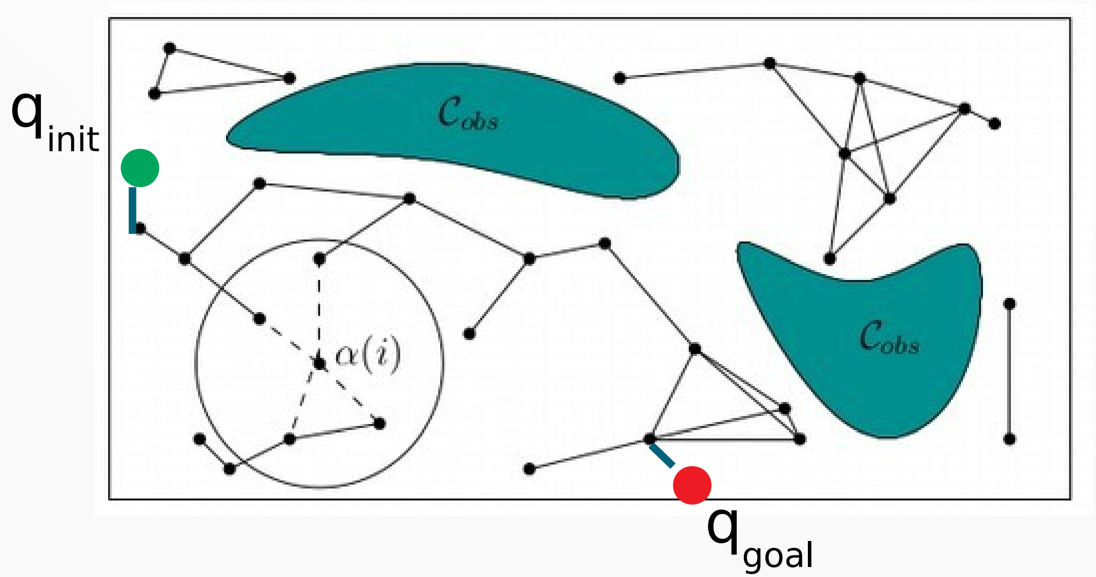


Problem definition(s)

Algorithms for Motion planning:

- PRM (Probabilistic Roadmaps)

L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*. IEEE Transactions on Robotics & Automation, 12(4):566–580, June 1996.



Problem definition(s)

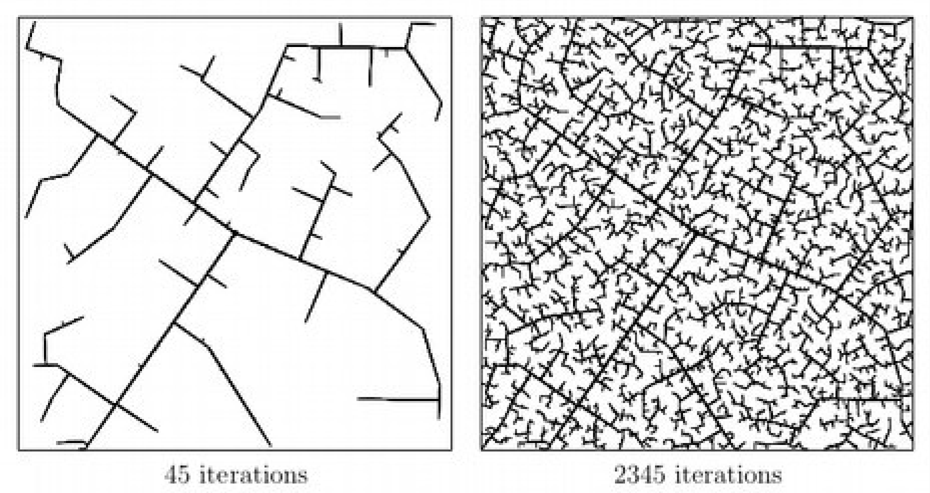
Algorithms for Motion planning:

- PRM (Probabilistic Roadmaps)

  L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*. IEEE Transactions on Robotics & Automation, 12(4):566–580, June 1996.

- RRT (Rapidly exploring Random Trees)

  S. M. LaValle and J. J. Kuffner. *Randomized kinodynamic planning*. In Proceedings IEEE International Conference on Robotics and Automation, pages 473–479, 1999.



Problem definition(s)

Algorithms for Motion planning:

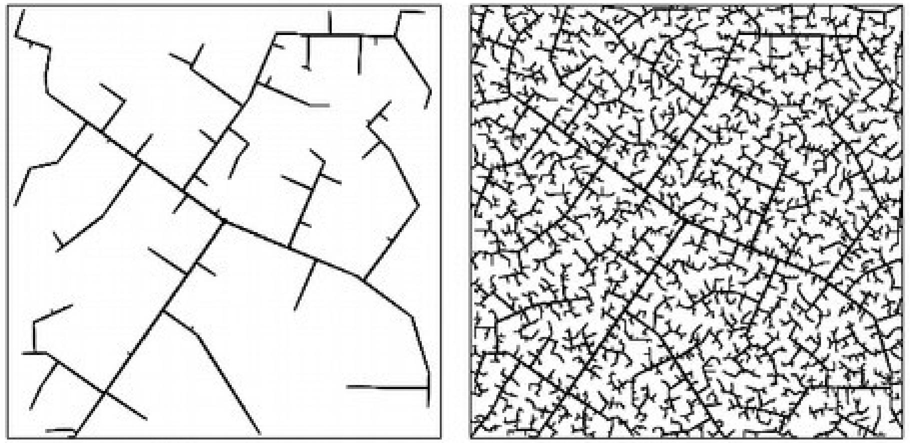
probabilistically complete

- PRM (Probabilistic Roadmaps)

  L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*. IEEE Transactions on Robotics & Automation, 12(4):566–580, June 1996.

- RRT (Rapidly exploring Random Trees)

  S. M. LaValle and J. J. Kuffner. *Randomized kinodynamic planning*. In Proceedings IEEE International Conference on Robotics and Automation, pages 473–479, 1999.



45 iterations 2345 iterations

Problem definition(s)

Problem definition(s)

Motion Planning

Given free configuration space $\mathcal{C}_{\text{free}}$,
initial configuration $q_I \in \mathcal{C}_{\text{free}}$, and
goal configurations $G \subseteq \mathcal{C}_{\text{free}}$,
a motion plan is defined as

$$\tau : [0,1] \rightarrow \mathcal{C}_{\text{free}}$$

$\tau(0) = q_I$ and $\tau(1) \in G$.

Problem definition(s)

Motion Planning

Given free configuration space $\mathcal{C}_{\text{free}}$,
initial configuration $q_I \in \mathcal{C}_{\text{free}}$, and
goal configurations $G \subseteq \mathcal{C}_{\text{free}}$,
a motion plan is defined as

$\tau : [0, 1] \rightarrow \mathcal{C}_{\text{free}}$

$\tau(0) = q_I$ and $\tau(1) \in G$.

Task Planning

$\Sigma = (S, A, \gamma, s_0, S_g)$ where,

- S is a finite set of states
- A is a finite set of actions
- $\gamma : S \times A \rightarrow S$ is a deterministic state-transition function, $\gamma(s, a) = s'$.
- $s_0 \in S$ is the start state
- $S_g \subseteq S$ is the set of goal states

Problem definition(s)

Motion Planning

Given free configuration space $\mathcal{C}_{\text{free}}$,
initial configuration $q_I \in \mathcal{C}_{\text{free}}$, and
goal configurations $G \subseteq \mathcal{C}_{\text{free}}$,
a motion plan is defined as

$\tau : [0, 1] \to \mathcal{C}_{\text{free}}$

$\tau(0) = q_I$ and $\tau(1) \in G$.

TAMP

$\phi : S \to 2^C$ which maps states to configurations;
$\xi : A \to 2^C$ which maps actions to motion plans.

Task Planning

$\Sigma = (S, A, \gamma, s_0, S_g)$ where,

- S is a finite set of states
- A is a finite set of actions
- $\gamma : S \times A \to S$ is a deterministic state-transition function, $\gamma(s, a) = s'$.
- $s_0 \in S$ is the start state
- $S_g \subseteq S$ is the set of goal states

Problem definition(s)

Motion Planning

Given free configuration space $\mathcal{C}_{\text{free}}$, initial configuration $q_I \in \mathcal{C}_{\text{free}}$, and goal configurations $G \subseteq \mathcal{C}_{\text{free}}$, a motion plan is defined as

$\tau : [0, 1] \to \mathcal{C}_{\text{free}}$

$\tau(0) = q_I$ and $\tau(1) \in G$.

Task Planning

$\Sigma = (S, A, \gamma, s_0, S_g)$ where,

- S is a finite set of states
- A is a finite set of actions
- $\gamma : S \times A \to S$ is a deterministic state-transition function, $\gamma(s, a) = s'$.
- $s_0 \in S$ is the start state
- $S_g \subseteq S$ is the set of goal states

TAMP

Space of grasps and placements

$\phi : S \to 2^C$ which maps states to configurations;
$\xi : A \to 2^C$ which maps actions to motion plans.

SSSP [edit]

The surfaces supporting stable placements are specified by 6 values (*xmin, xmax, ymin, ymax, zmin, zmax*) representing a rectangular region **in the reference frame of the object**.

```
<sssp>
   <xmin>-0.25<xmax><xmin>0.25<xmax>
   <ymin>-0.25<ymax><ymin>0.25<ymax>
   <zmin>0.5<zmax><zmin>0.5<zmax>
</sssp>
```

Alternatively, SSSP can be represented by polygonal regions. Then, the tag **<pssp>** (polygon supporting stable placements) is used and a polygonal region is specified as a list of 3D coordinates **in the reference frame of the object**, assumed to lie in a plane:

```
<pssp>
   <point><x>-0.15</x><y>0.33</y><z>0.5</z></point>
   <point><x>-0.22</x><y>0.25</y><z>0.5</z></point>
   <point><x>0.25</x><y>0.12</y><z>0.5</z></point>
   ...
</pssp>
```

Remarks:

- In stacking problems, the SSSP is defined by a single point centred on top of the object (i.e., *xmin=xmax, ymin=ymax*) therefore objects will be aligned on top of each other (e.g., *disc2* and *disc3* in the example below).
- The number of SSSP per object is currently limited to one. This assumes that if an object is rotated about the x or y axis, it is no longer possible to place something on it. This assumption is sufficient for the current set of benchmark problems, but it may be relaxed later, i.e., by allowing lists of SSOP.

SOP [edit]

The stable object poses are defined similarly to continuous grasps: a template rotation and a rotation axis (both expressed **in the world frame of reference**, plus the *distance* parameter, which is the distance between the centre of the object and the SSSP.

```
<sop>
   <template>1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0</template>
   <axis>0.0 0.0 1.0</axis>
   <distance>0.08</distance>
</sop>
```

MP

Space of grasps and placements

states to configurations;
actions to motion plans.

- $s_0 \in S$ is the start state
- $S_g \subseteq S$ is the set of goal states

Problem definition(s)

Motion Planning

Given free configuration space $\mathcal{C}_{\text{free}}$, initial configuration $q_I \in \mathcal{C}_{\text{free}}$, and goal configurations $G \subseteq \mathcal{C}_{\text{free}}$, a motion plan is defined as

$\tau : [0,1] \to \mathcal{C}_{\text{free}}$

$\tau(0) = q_I$ and $\tau(1) \in G$.

Task Planning

$\Sigma = (S, A, \gamma, s_0, S_g)$ where,

- S is a finite set of states
- A is a finite set of actions
- $\gamma : S \times A \to S$ is a deterministic state-transition function, $\gamma(s, a) = s'$.
- $s_0 \in S$ is the start state
- $S_g \subseteq S$ is the set of goal states

TAMP

$\phi : S \to 2^C$ which maps states to configurations;
$\xi : A \to 2^C$ which maps actions to motion plans.

find a sequence of actions $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ following a sequence of task states $\langle s_0, s_1, \ldots, s_n \rangle$ such that

$$s_n \in S_g \text{ and}$$

$$s_{i+1} = \gamma(s_i, a_i)$$

and to find a sequence of motion plans $\langle \tau_0, \tau_1, \ldots, \tau_{n-1} \rangle$ such that $\forall i = 0 \ldots n-1$:

$$\tau_i(0) \in \phi(s_i) \text{ and } \tau_i(1) \in \phi(s_{i+1})$$

$$\tau_i \in \xi(a_i), \text{ and}$$

$$\tau_i(1) = \tau_{i+1}(0)$$

Problem definition(s)

Initial state

On(B, Table)
On(A, Table)
On(C, A)

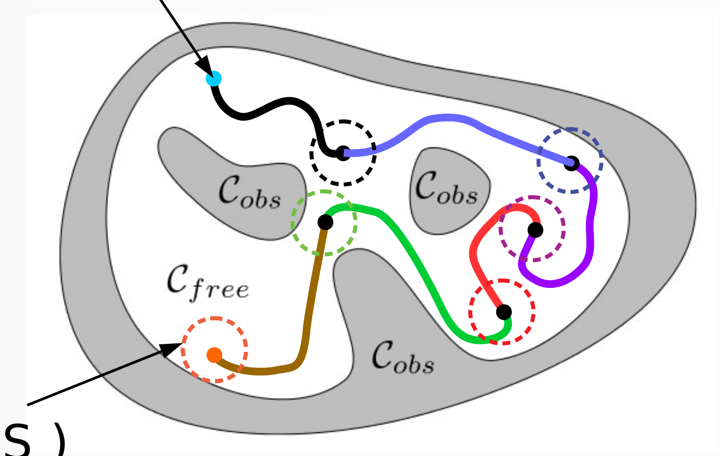
q_{init}

Goal state

On(C, Table)
On(B, C)
On(A, B)

Plan

Pick(C)
Place(C, Table)
Pick(B)
Place(B, C)
Pick(A)
Place(A, B)



$\Phi(S_g)$

TAMP

$\phi : S \rightarrow 2^C$ which maps states to configurations;
$\xi : A \rightarrow 2^C$ which maps actions to motion plans.

find a sequence of actions $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ following a sequence of task states $\langle s_0, s_1, \ldots, s_n \rangle$ such that

$$s_n \in S_g \text{ and}$$

$$s_{i+1} = \gamma(s_i, a_i)$$

and to find a sequence of motion plans $\langle \tau_0, \tau_1, \ldots, \tau_{n-1} \rangle$ such that $\forall i = 0 \ldots n-1$:

$$\tau_i(0) \in \phi(s_i) \text{ and } \tau_i(1) \in \phi(s_{i+1})$$

$$\tau_i \in \xi(a_i), \text{ and}$$

$$\tau_i(1) = \tau_{i+1}(0)$$

Problem definition(s)

TAMP

Initial state

On(B, Table)
On(A, Table)
On(C, A)

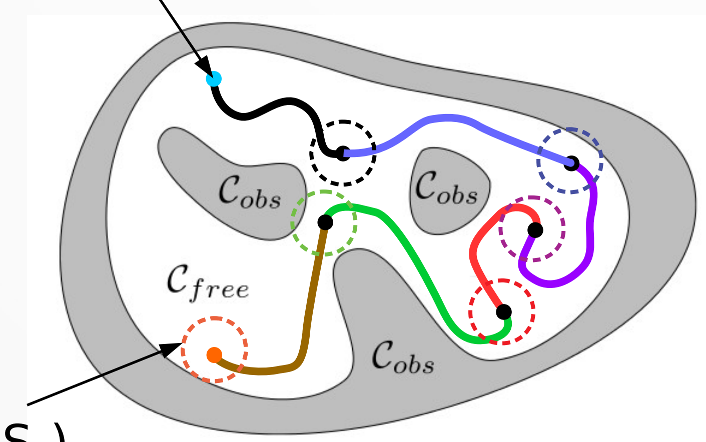
q_{init}

Goal state

On(C, Table)
On(B, C)
On(A, B)

Plan

Pick(C)
Place(C, Table)
Pick(B)
Place(B, C)
Pick(A)
Place(A, B)



$\Phi(S_g)$

$\phi : S \to 2^C$ which maps states to configurations;
$\xi : A \to 2^C$ which maps actions to motion plans.

find a sequence of actions $\langle a_0, a_1, \ldots, a_{n-1} \rangle$ following a sequence of task states $\langle s_0, s_1, \ldots, s_n \rangle$ such that

$$s_n \in S_g \text{ and}$$

$$s_{i+1} = \gamma(s_i, a_i)$$

and to find a sequence of motion plans $\langle \tau_0, \tau_1, \ldots, \tau_{n-1} \rangle$ such that $\forall i = 0 \ldots n-1$:

$$\tau_i(0) \in \phi(s_i) \text{ and } \tau_i(1) \in \phi(s_{i+1})$$

$$\tau_i \in \xi(a_i), \text{ and}$$

$$\tau_i(1) = \tau_{i+1}(0)$$

- Fully observable
- Deterministic

Problem definition(s)

TAMP's siblings:

- **Manipulation Planning**

  Siméon T, Laumond J-P, Cortés J, Sahbani A. *Manipulation Planning with Probabilistic Roadmaps*. The International Journal of Robotics Research. 2004;23(7-8):729-746.

- **NAMO** (Navigation Among Movable Obstacles)

  Mike Stilman. 2007. *Navigation among movable obstacles*. Ph.D. Dissertation. Carnegie Mellon University, USA. Advisor(s) James J. Kuffner.

- **Rearrangement Planning**

  G. Havur, et al., *Geometric rearrangement of multiple movable objects on cluttered surfaces: A hybrid reasoning approach,* 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014, pp

- **Multimodal Motion Planning**

  Hauser K, Latombe J-C. *Multi-modal Motion Planning in Non-expansive Spaces*. The International Journal of Robotics Research. 2010;29(7):897-915.

Problem definition(s)

TAMP's siblings:

- **Manipulation Planning**

  Siméon T, Laumond J-P, Cortés J, Sahbani A. *Manipulation Planning with Probabilistic Roadmaps*. The International Journal of Robotics Research. 2004;23(7-8):729-746.

- **NAMO** (Navigation Among Movable Obstacles)

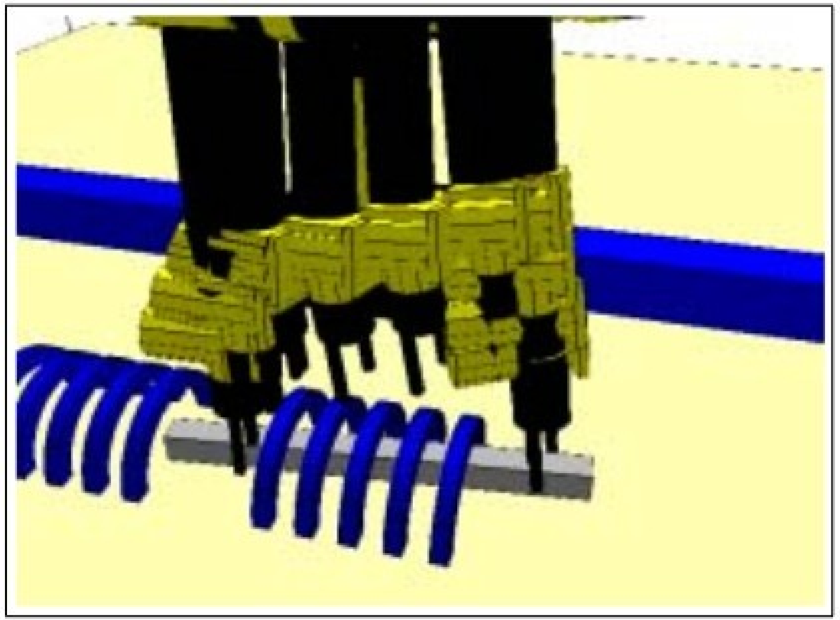
  Mike Stilman. 2007. *Navigation among movable obstacles*. Ph.D. Dissertati ıes J. Kuffner.

- **Rearrangement Planning**

  G. Havur, et al., *Geometric rearrangement of multiple movable objects on* 014 IEEE International Conference on Robotics and Automation (ICRA), Hong Ko

- **Multimodal Motion Planning**

  Hauser K, Latombe J-C. *Multi-modal Motion Planning in Non-expansive Spac* 2010;29(7):897-915.



Problem definition(s)

TAMP's siblings:

- **Manipulation Planning**

  Siméon T, Laumond J-P, Cortés J, Sahbani A. *Manipulation Planning with Probabilistic Roadmaps*. The International Journal of Robotics Research. 2004;23(7-8):729-746.

- **NAMO** (Navigation Among Movable Obstacles)

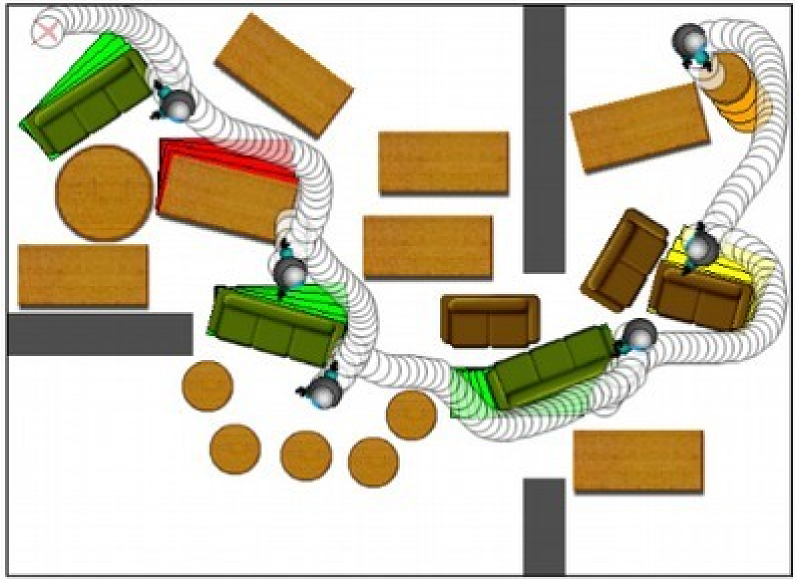
  Mike Stilman. 2007. *Navigation among movable obstacles*. Ph.D. Dissertation. Carnegie Mellon University, USA. Advisor(s) James J. Kuffner.

- **Rearrangement Planning**

  G. Havur, et al., *Geometric rearrangement of multiple movable objects on cluttered su*
  IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014, p

- **Multimodal Motion Planning**

  Hauser K, Latombe J-C. *Multi-modal Motion Planning in Non-expansive Spaces*. The Int
  2010;29(7):897-915.



Problem definition(s)

TAMP's siblings:

- **Manipulation Planning**

  Siméon T, Laumond J-P, Cortés J, Sahbani A. *M*
  Robotics Research. 2004;23(7-8):729-746.

- **NAMO** (Navigation Among Mov

  Mike Stilman. 2007. *Navigation among mova*
  J. Kuffner.

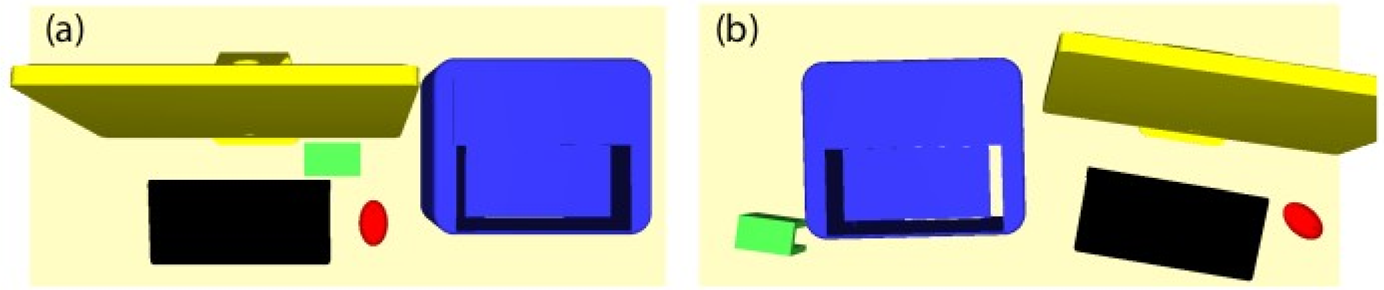


Figure 4. (a) Initial configuration, (b) Tentative final placement

- **Rearrangement Planning**

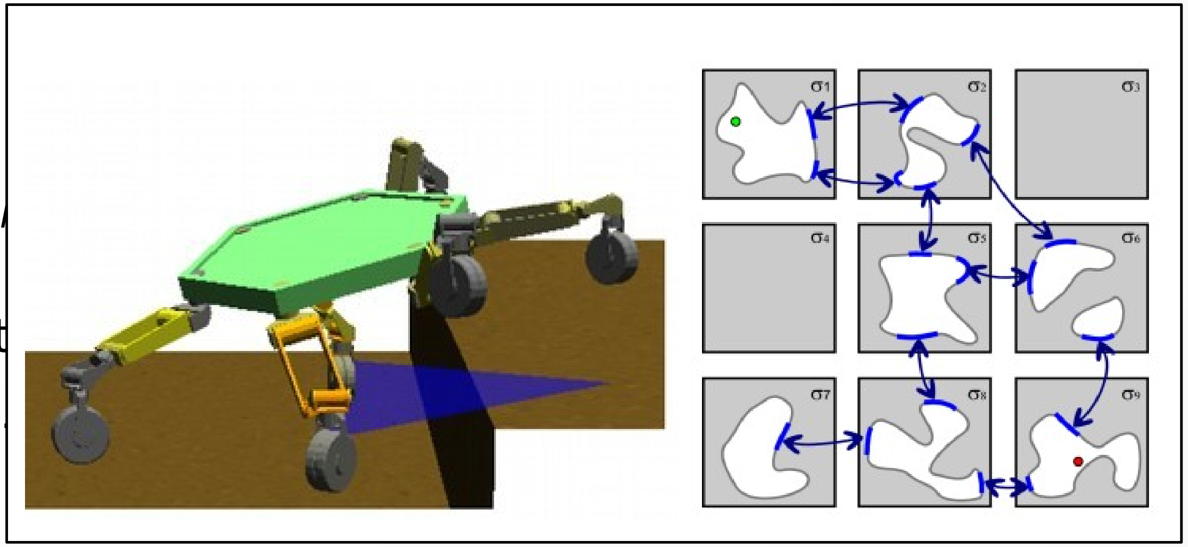
  G. Havur, et al., *Geometric rearrangement of multiple movable objects on cluttered surfaces: A hybrid reasoning approach,* 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014, pp

- **Multimodal Motion Planning**

  Hauser K, Latombe J-C. *Multi-modal Motion Planning in Non-expansive Spaces*. The International Journal of Robotics Research. 2010;29(7):897-915.

Problem definition(s)

TAMP’s siblings:



- **Manipulation Planning**

  Siméon T, Laumond J-P, Cortés J, Sahbani A. *Manipulation I*
  Robotics Research. 2004;23(7-8):729-746.

- **NAMO** (Navigation Among Movable Obst

  Mike Stilman. 2007. *Navigation among movable obstacles.*
  J. Kuffner.

- **Rearrangement Planning**

  G. Havur, et al., *Geometric rearrangement of multiple movable objects on cluttered surfaces: A hybrid reasoning approach,* 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014, pp

- **Multimodal Motion Planning**

  Hauser K, Latombe J-C. *Multi-modal Motion Planning in Non-expansive Spaces*. The International Journal of Robotics Research. 2010;29(7):897-915.

Outline

- Problem definition(s)
- What makes it difficult?
- 3 approaches to TAMP
- Future directions

Outline

- Problem definition(s)
- What makes it difficult?
- 3 approaches to TAMP
- Future directions

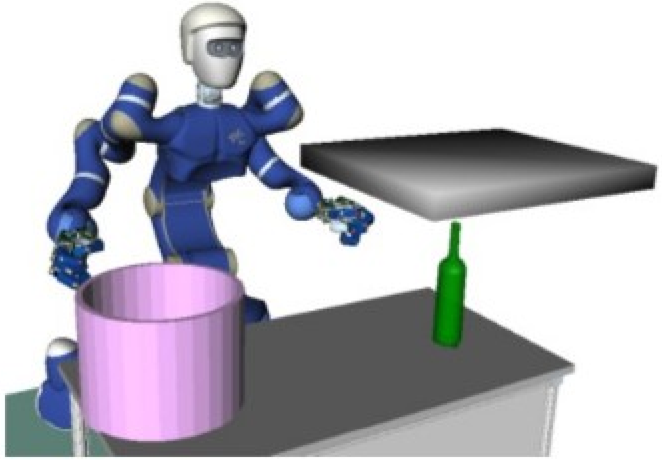
What makes it difficult?

What makes it difficult?

- Coupling between S and C
- Geometric Backtracking
- Large search space, different metrics

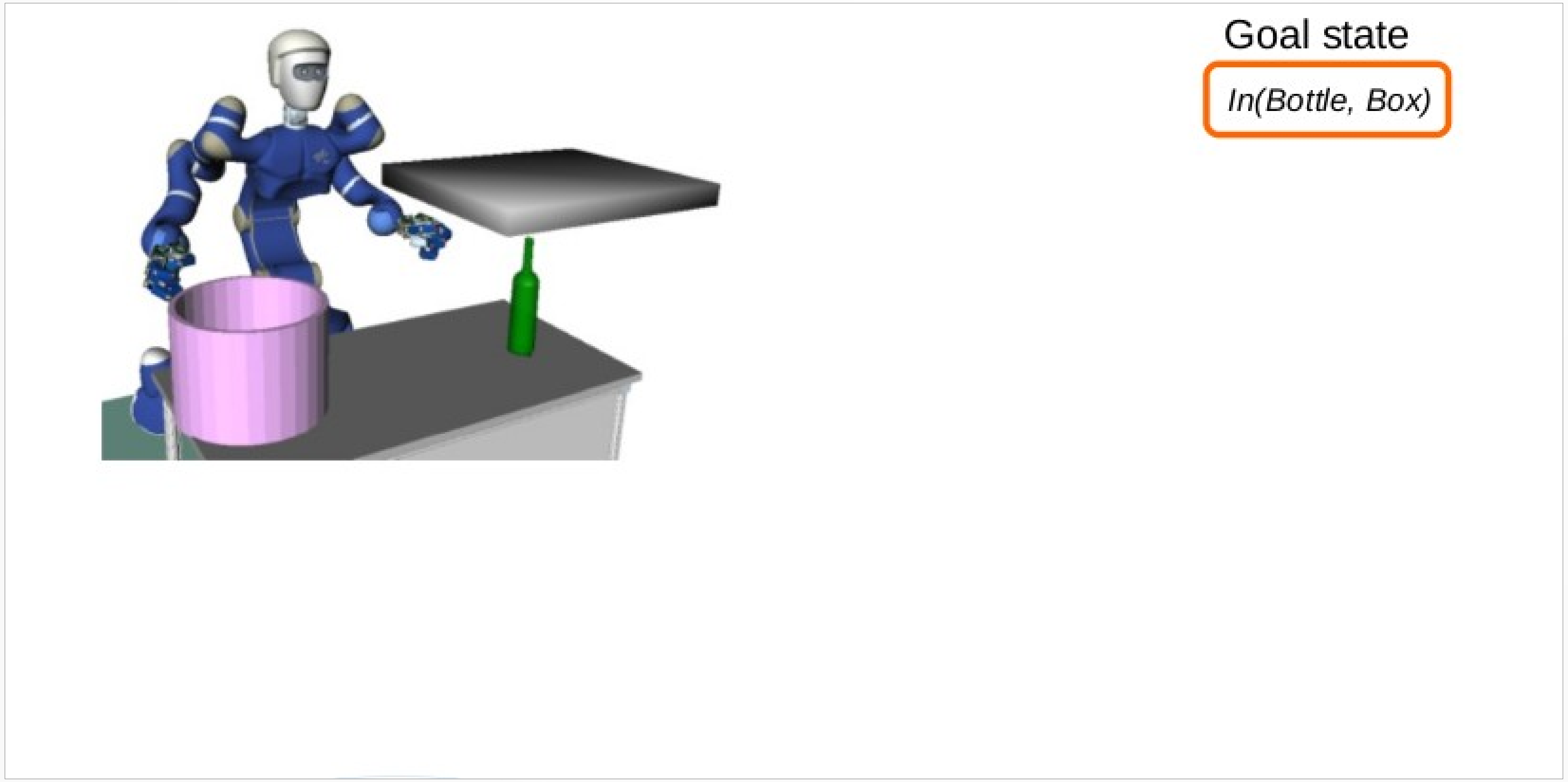
What makes it difficult?

- Coupling between S and C



What makes it difficult?

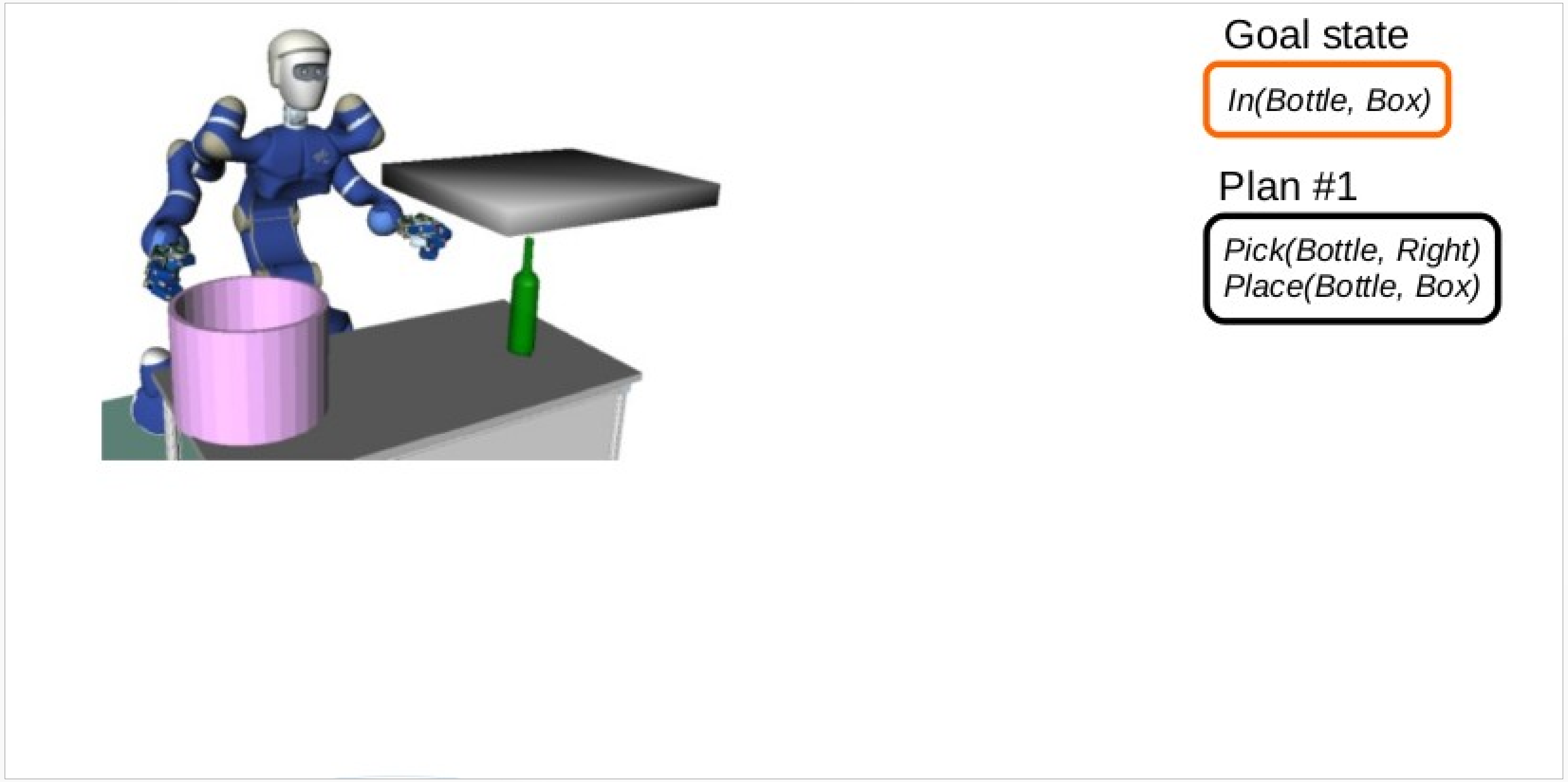
- Coupling between S and C



Naive approach: Task Planning first, Motion Planning then.

What makes it difficult?

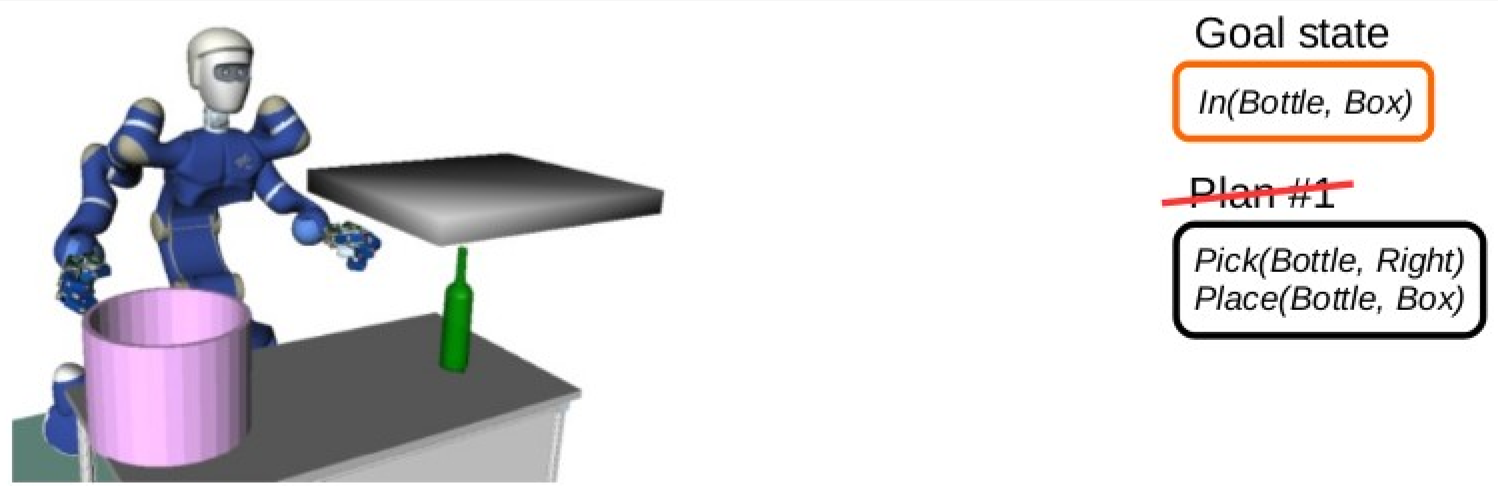
- Coupling between S and C



Naive approach: Task Planning first, Motion Planning then.

What makes it difficult?

- Coupling between S and C

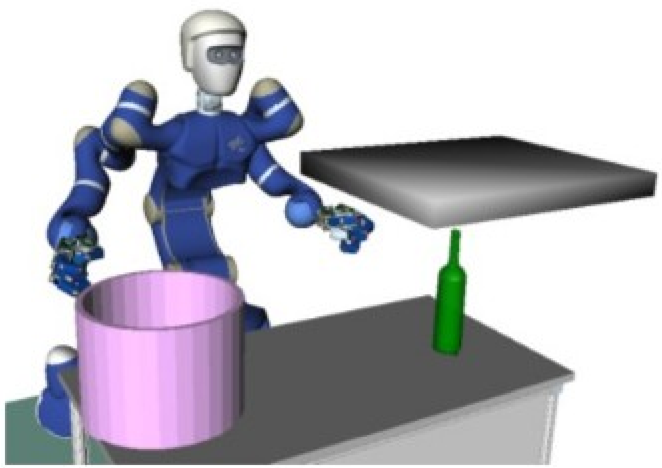


- the **bottle** is not reachable by the **right arm**

Naive approach: Task Planning first, Motion Planning then.

What makes it difficult?

- Coupling between S and C



Goal state

In(Bottle, Box)

~~Plan #1~~

Plan #2

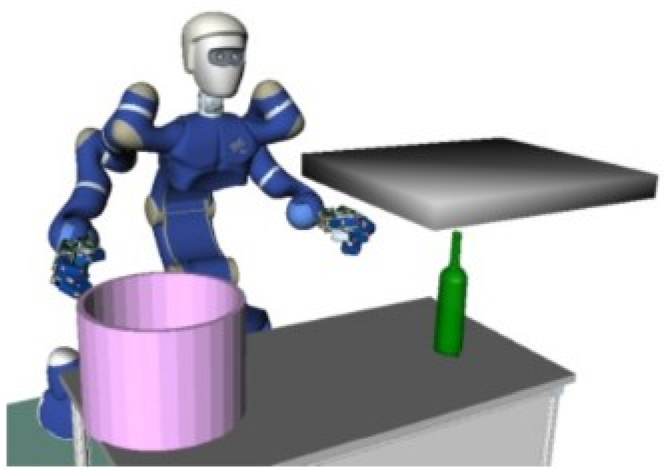
Pick(Bottle, Left)
Place(Bottle, Box)

- the **bottle** is not reachable by the **right arm**

Naive approach: Task Planning first, Motion Planning then.

What makes it difficult?

- Coupling between S and C



Goal state

In(Bottle, Box)

~~Plan #1~~
~~Plan #2~~

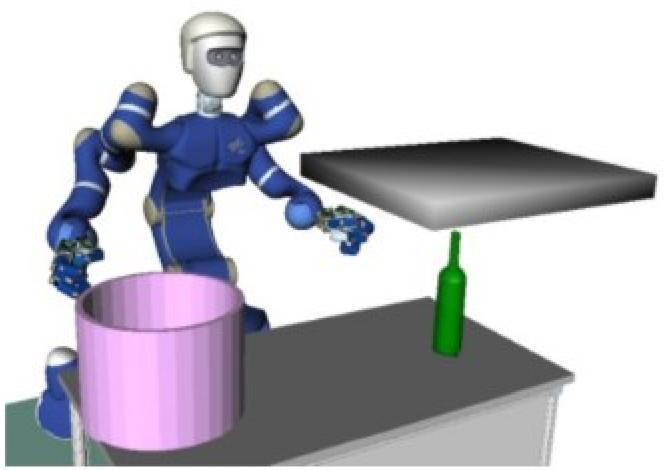
Pick(Bottle, Left)
Place(Bottle, Box)

- the **bottle** is not reachable by the **right arm**
- the **box** is not reachable by the **left arm**

Naive approach: Task Planning first, Motion Planning then.

What makes it difficult?

- Coupling between S and C



Goal state

In(Bottle, Box)

~~Plan #1~~
~~Plan #2~~
Plan #3

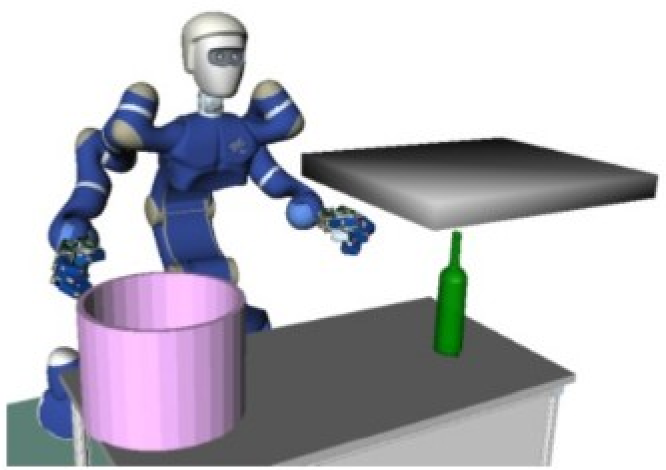
Pick(Bottle, Left)
Place(Bottle, Table)
Pick(Bottle, Right)
Place(Bottle, Box)

- the **bottle** is not reachable by the **right arm**
- the **box** is not reachable by the **left arm**

Naive approach: Task Planning first, Motion Planning then.

What makes it difficult?

- Coupling between S and C



Goal state

In(Bottle, Box)

~~Plan #1~~

~~Plan #2~~

Plan #3 ✓

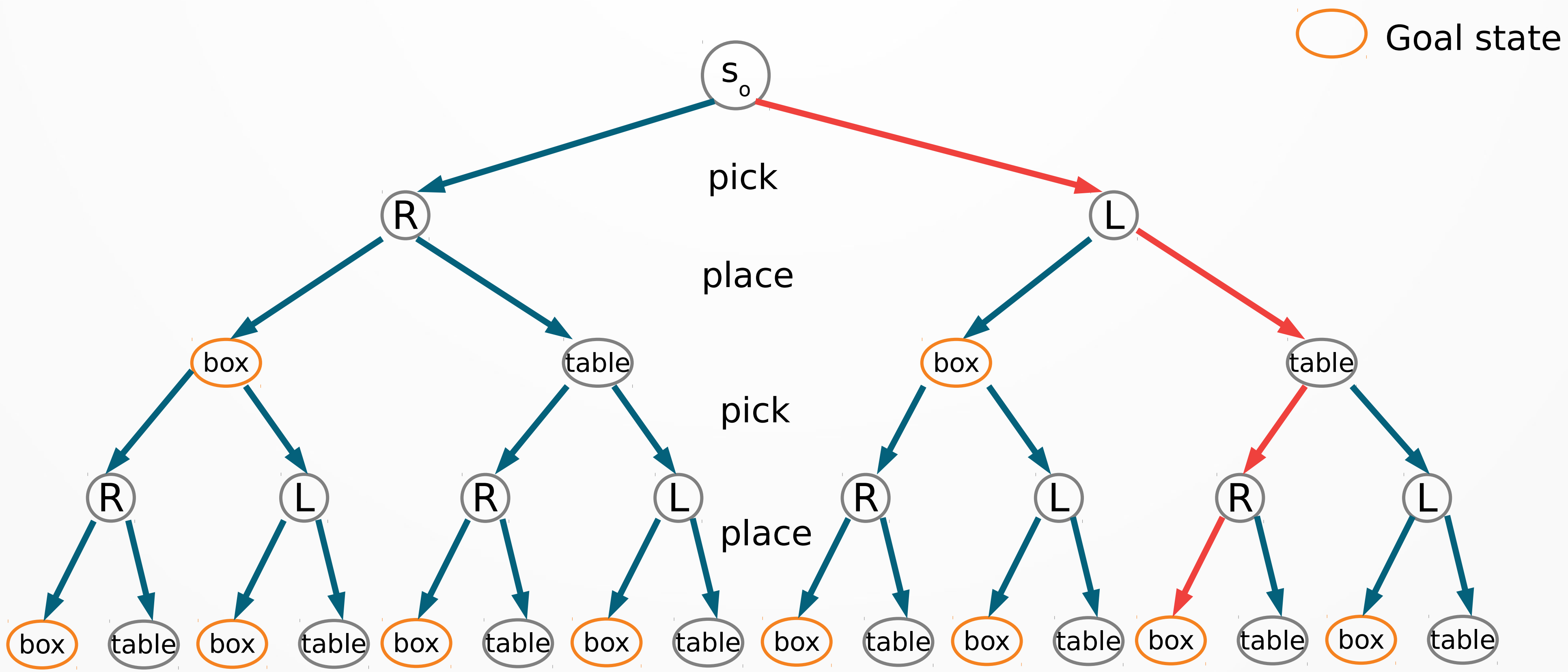
Pick(Bottle, Left)
Place(Bottle, Table)
Pick(Bottle, Right)
Place(Bottle, Box)

- the **bottle** is not reachable by the **right arm**
- the **box** is not reachable by the **left arm**

Naive approach: Task Planning first, Motion Planning then.

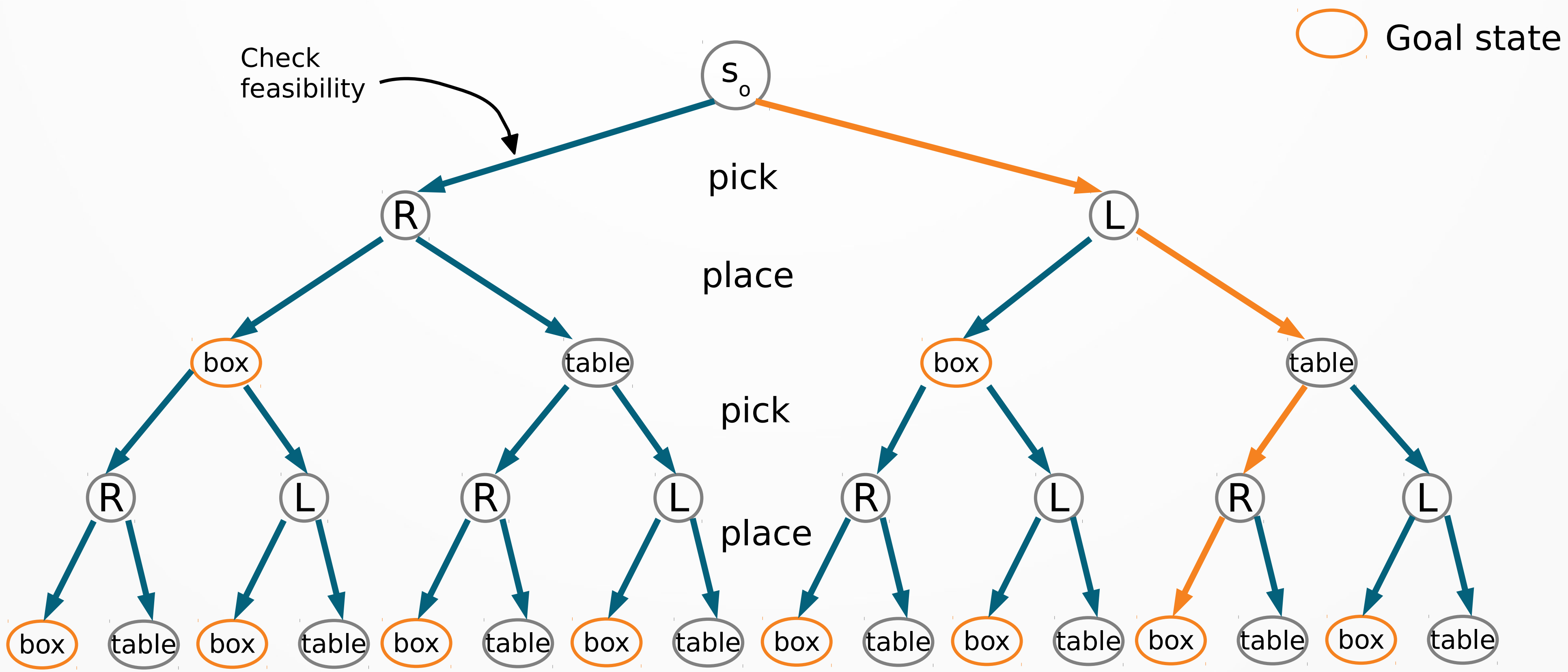
What makes it difficult?

- Coupling between S and C



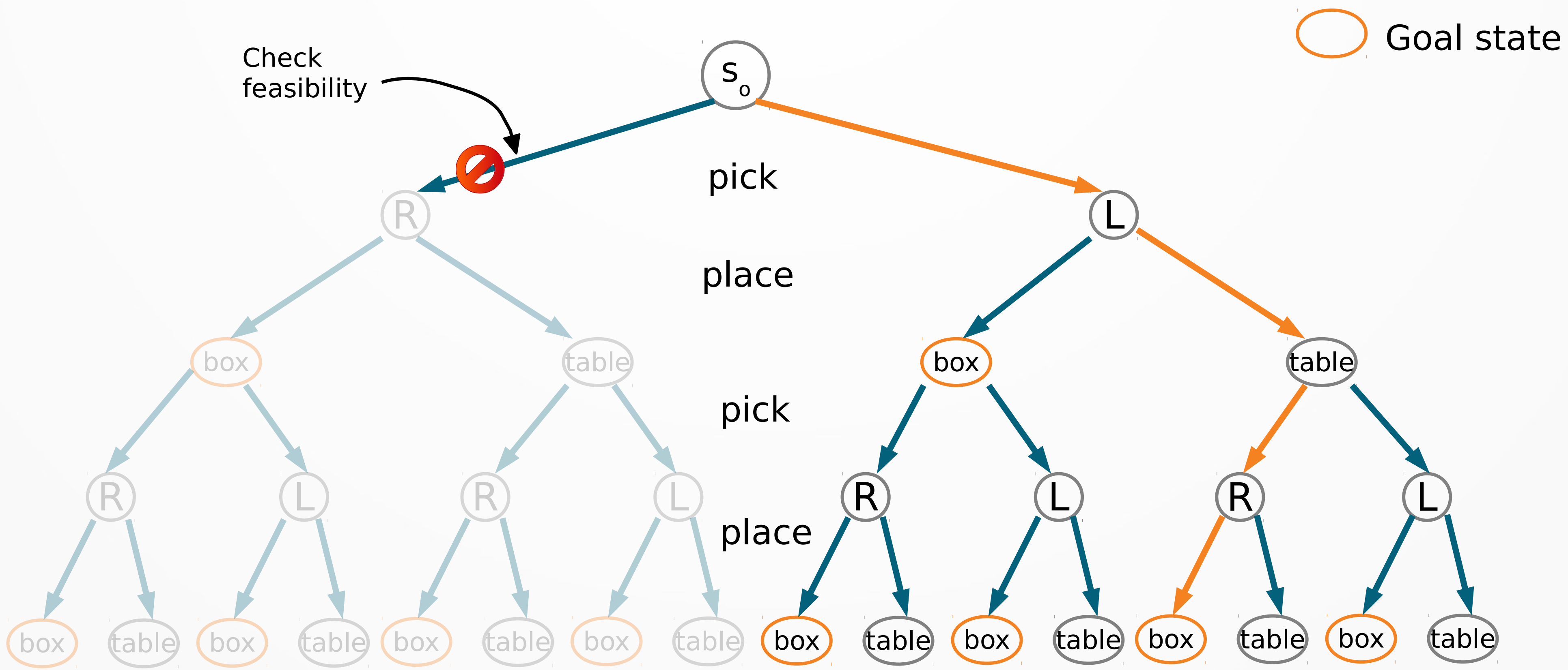
What makes it difficult?

- Coupling between S and C



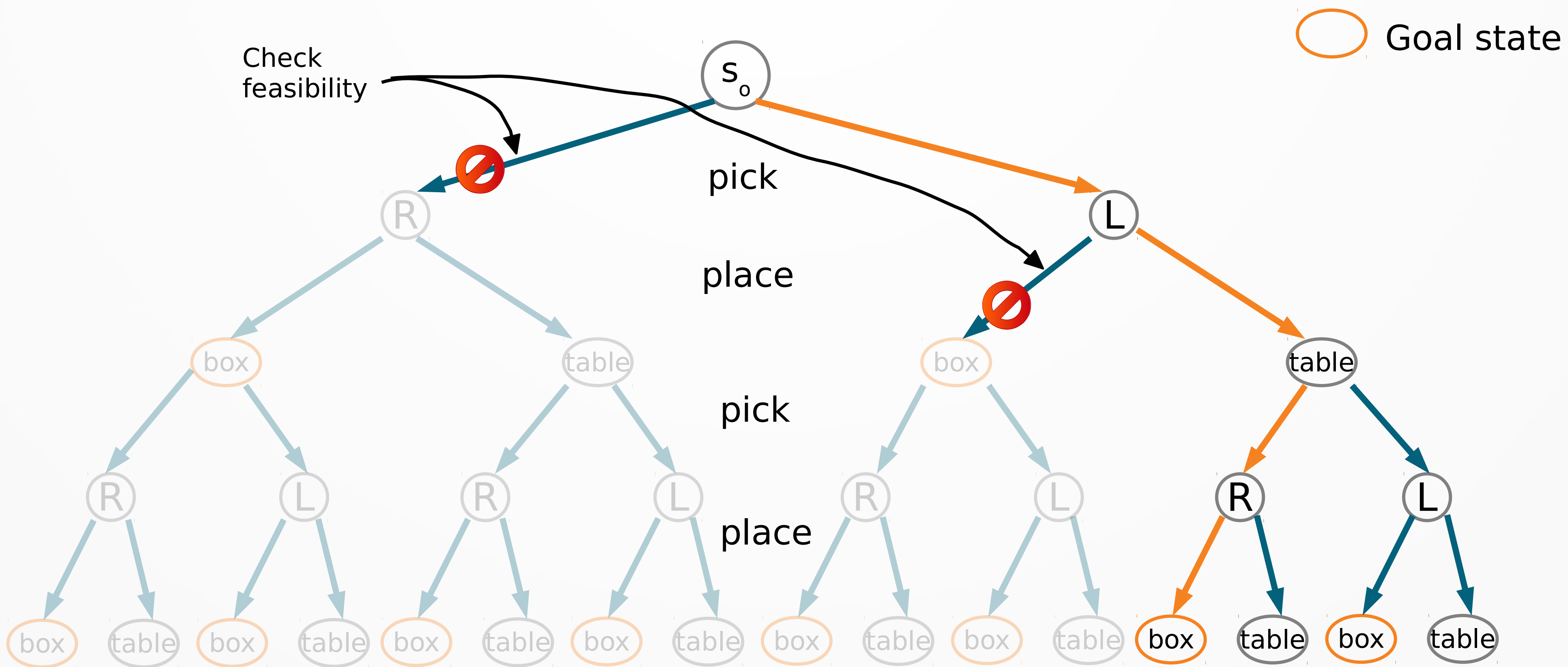
What makes it difficult?

- Coupling between S and C



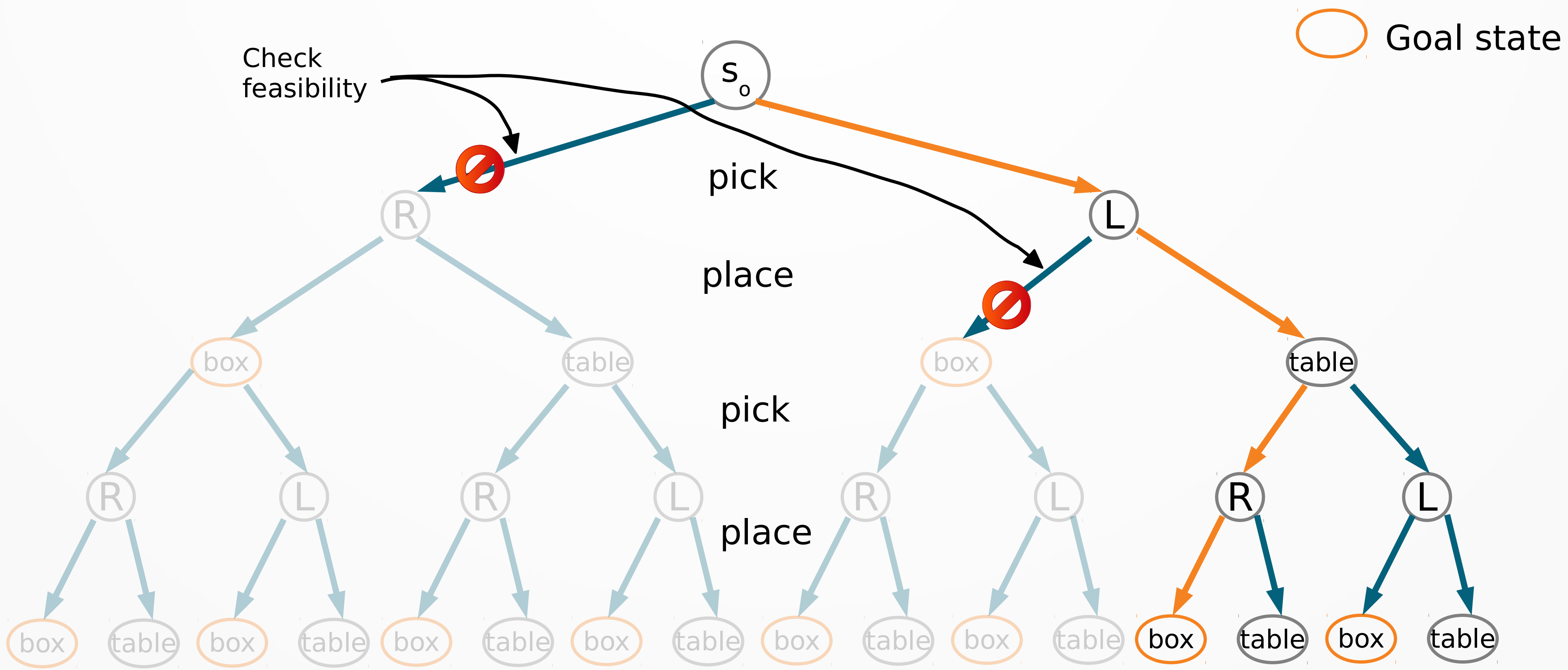
What makes it difficult?

- Coupling between S and C



What makes it difficult?

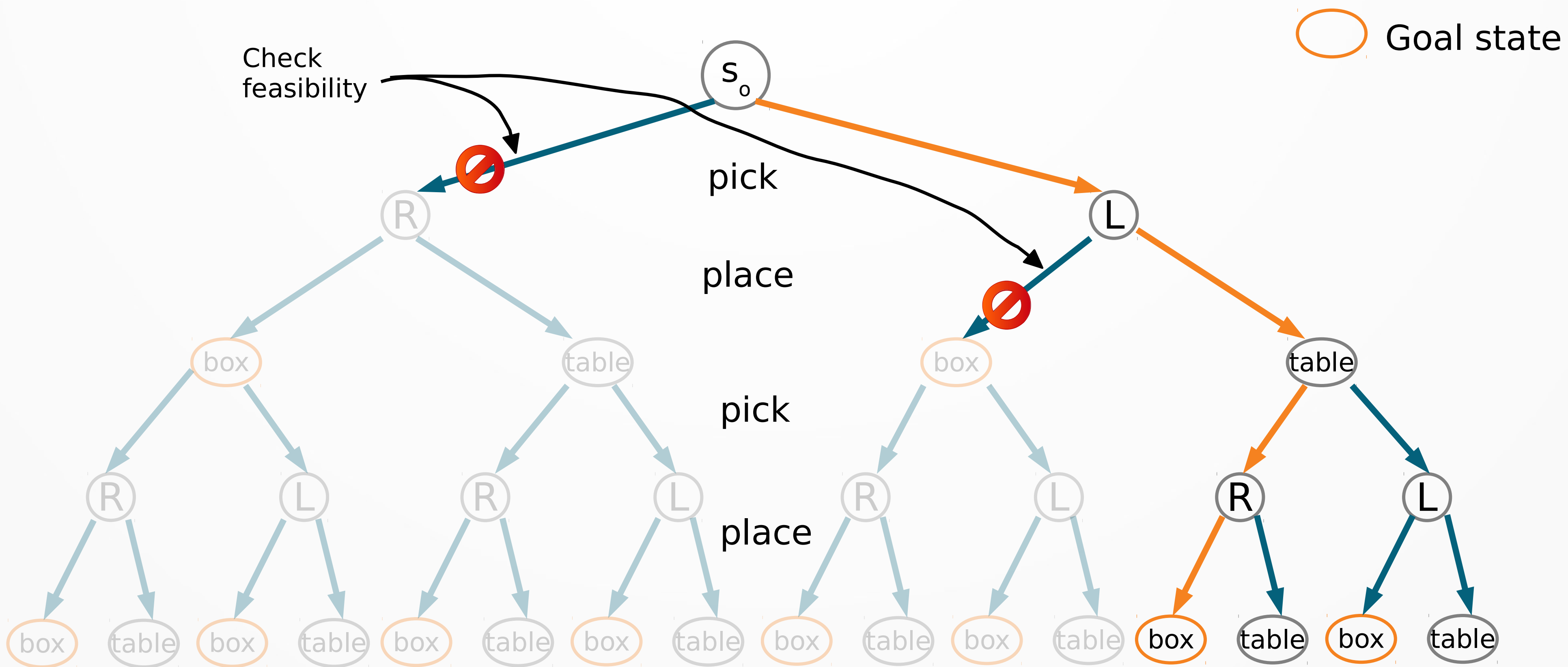
- Coupling between S and C



→ **Interleaving** symbolic and geometric search enables efficient pruning!

What makes it difficult?

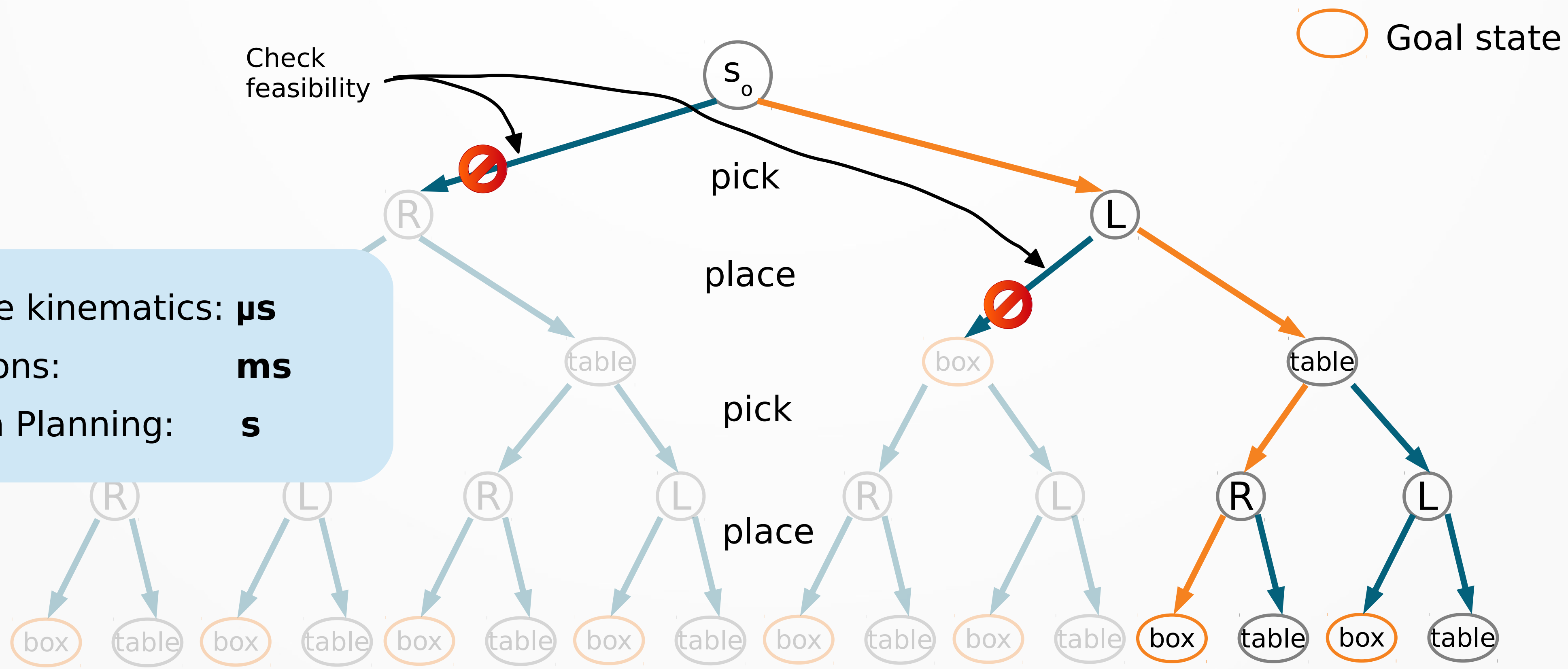
- Coupling between S and C



→ **Interleaving** symbolic and geometric search enables efficient pruning!
But checking feasibility has a cost...

What makes it difficult?

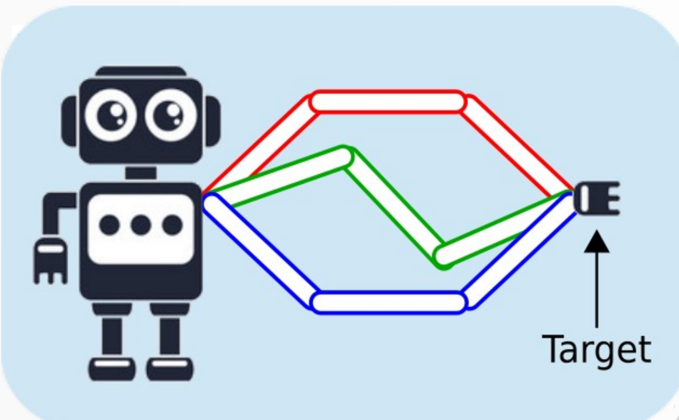
- Coupling between S and C



- Inverse kinematics: **µs**
- Collisions: **ms**
- Motion Planning: **s**

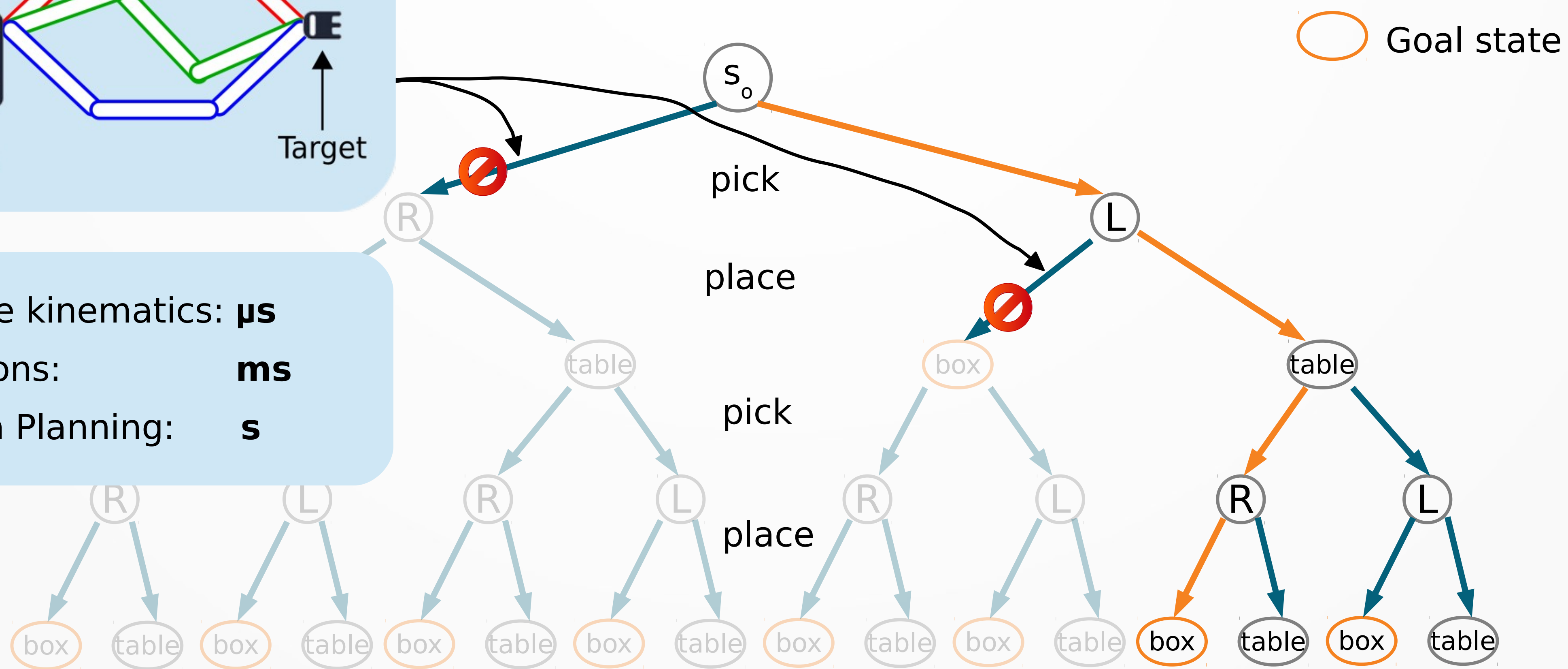
→ **Interleaving** symbolic and geometric search enables efficient pruning!
But checking feasibility has a cost…

What makes it difficult?



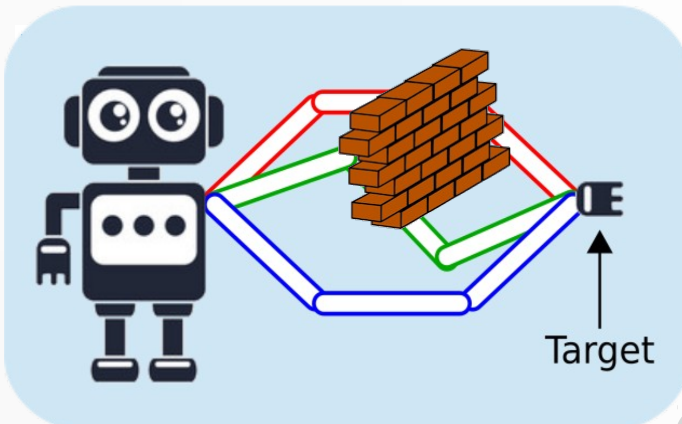
S and C

- Inverse kinematics: **μs**
- Collisions: **ms**
- Motion Planning: **s**

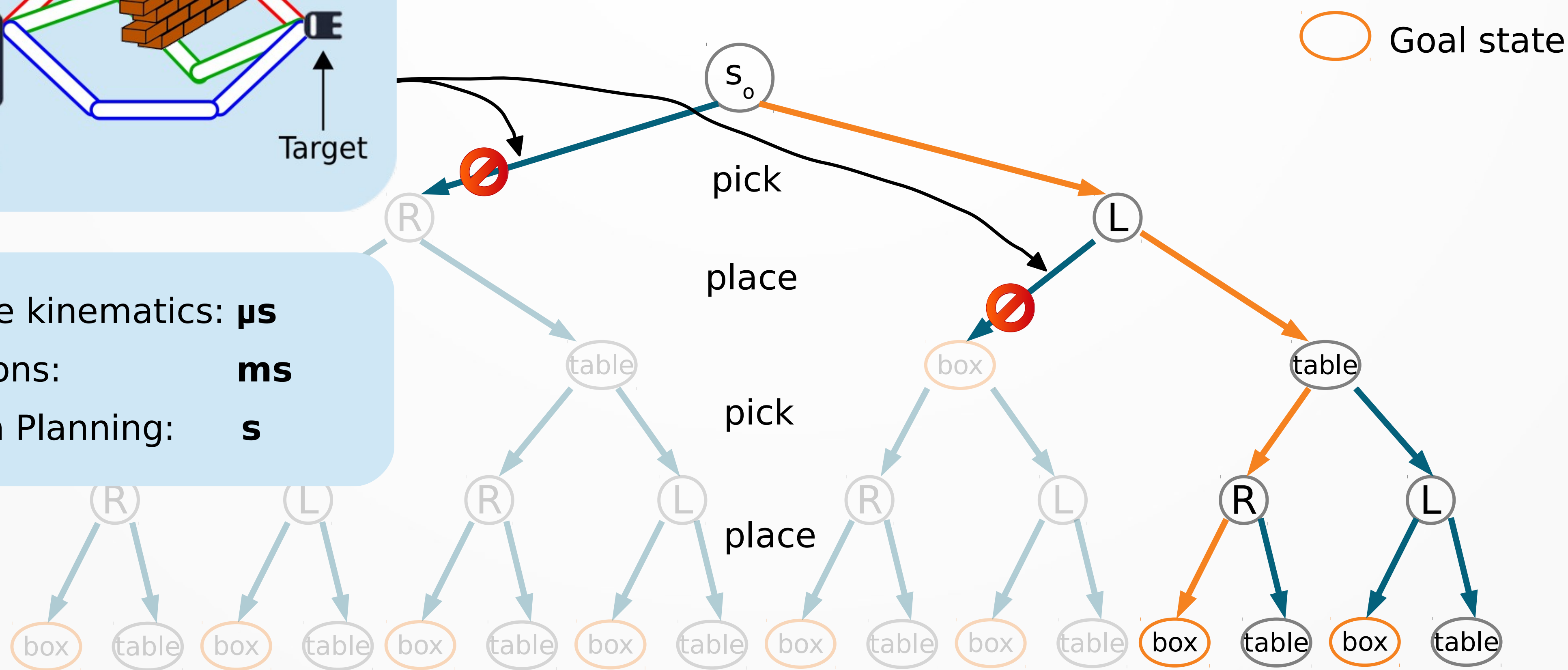


→ **Interleaving** symbolic and geometric search enables efficient pruning!
But checking feasibility has a cost…

What makes it difficult?



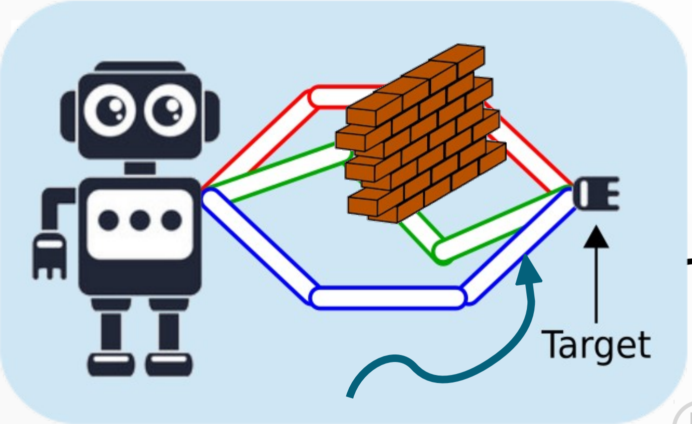
S and C



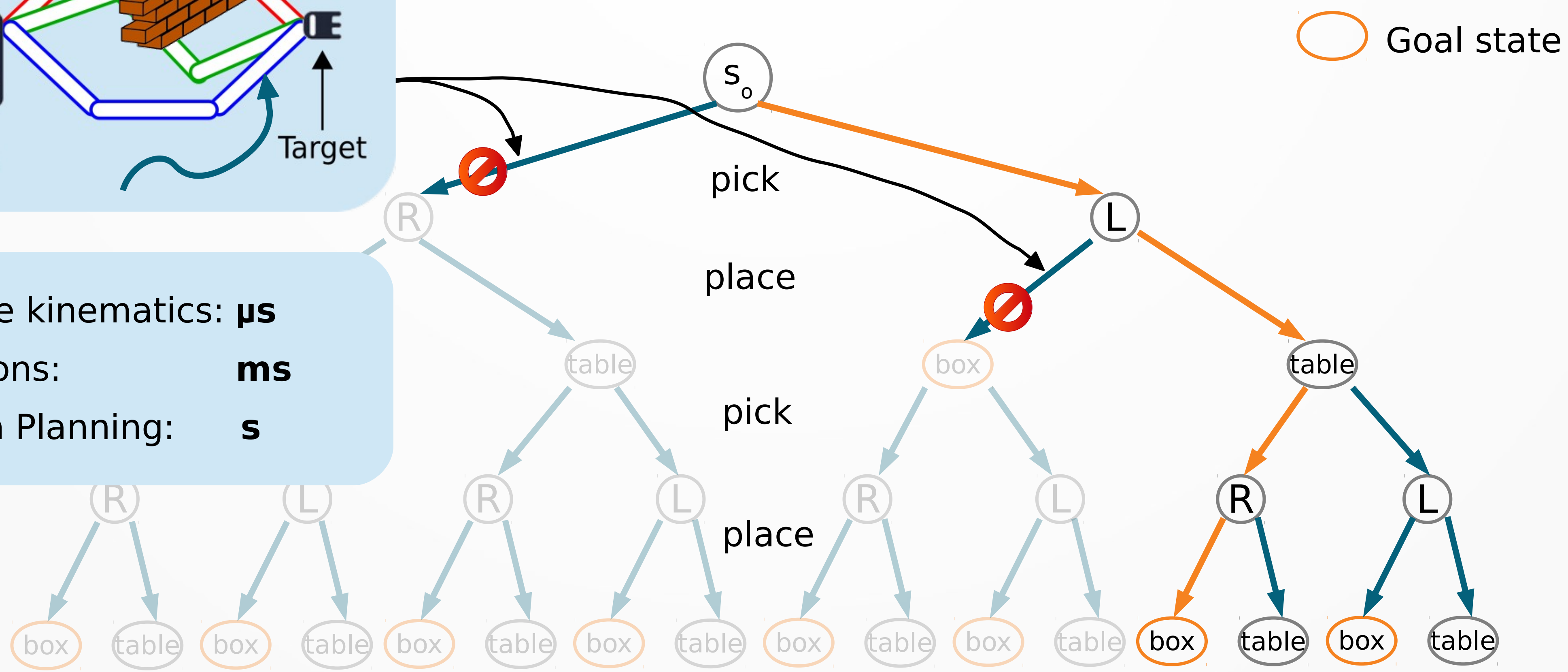
- Inverse kinematics: **μs**
- Collisions: **ms**
- Motion Planning: **s**

→ **Interleaving** symbolic and geometric search enables efficient pruning!
But checking feasibility has a cost…

What makes it difficult?



S and C



- Inverse kinematics: **μs**
- Collisions: **ms**
- Motion Planning: **s**

→ **Interleaving** symbolic and geometric search enables efficient pruning!
But checking feasibility has a cost...

What makes it difficult?

- Coupling between S and C
- Geometric Backtracking
- Large search space, different metrics

What makes it difficult?

- Coupling between S and C
- Geometric Backtracking
- Large search space, different metrics

What makes it difficult?

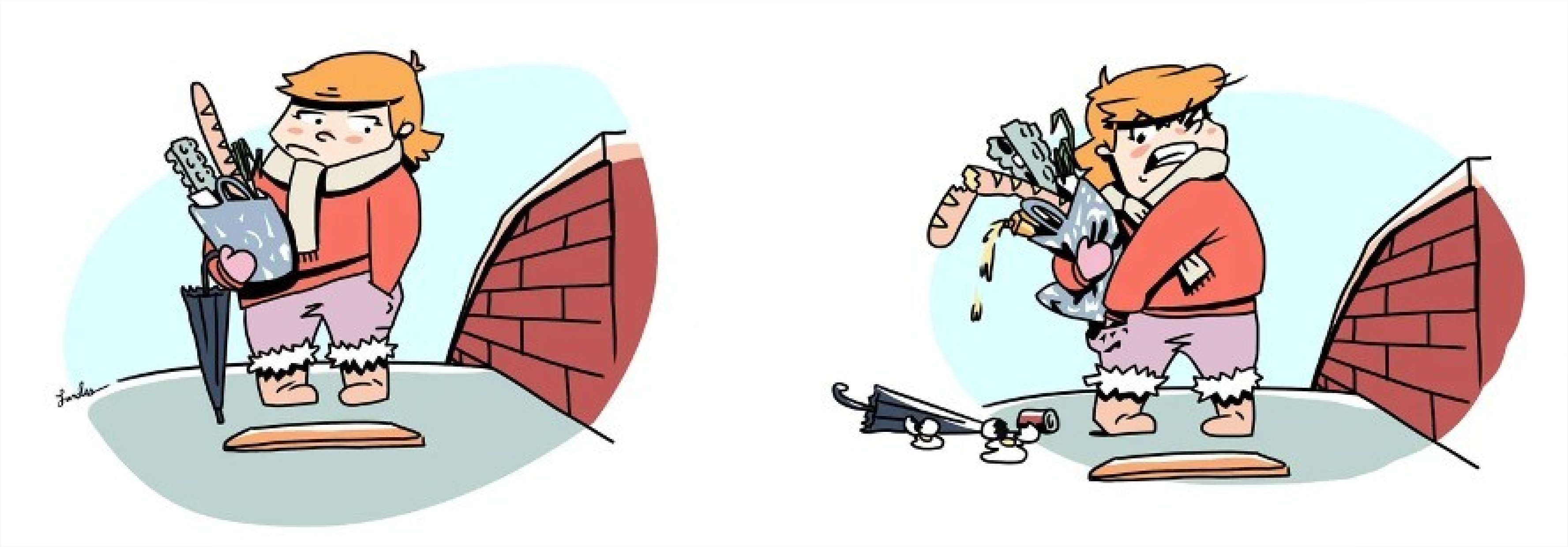
- Geometric Backtracking

  The process of reconsidering choices at the geometric level.

What makes it difficult?

- **Geometric Backtracking**

  The process of reconsidering choices at the geometric level.



https://i.redd.it/p2m17shrhjuz.jpg

What makes it difficult?

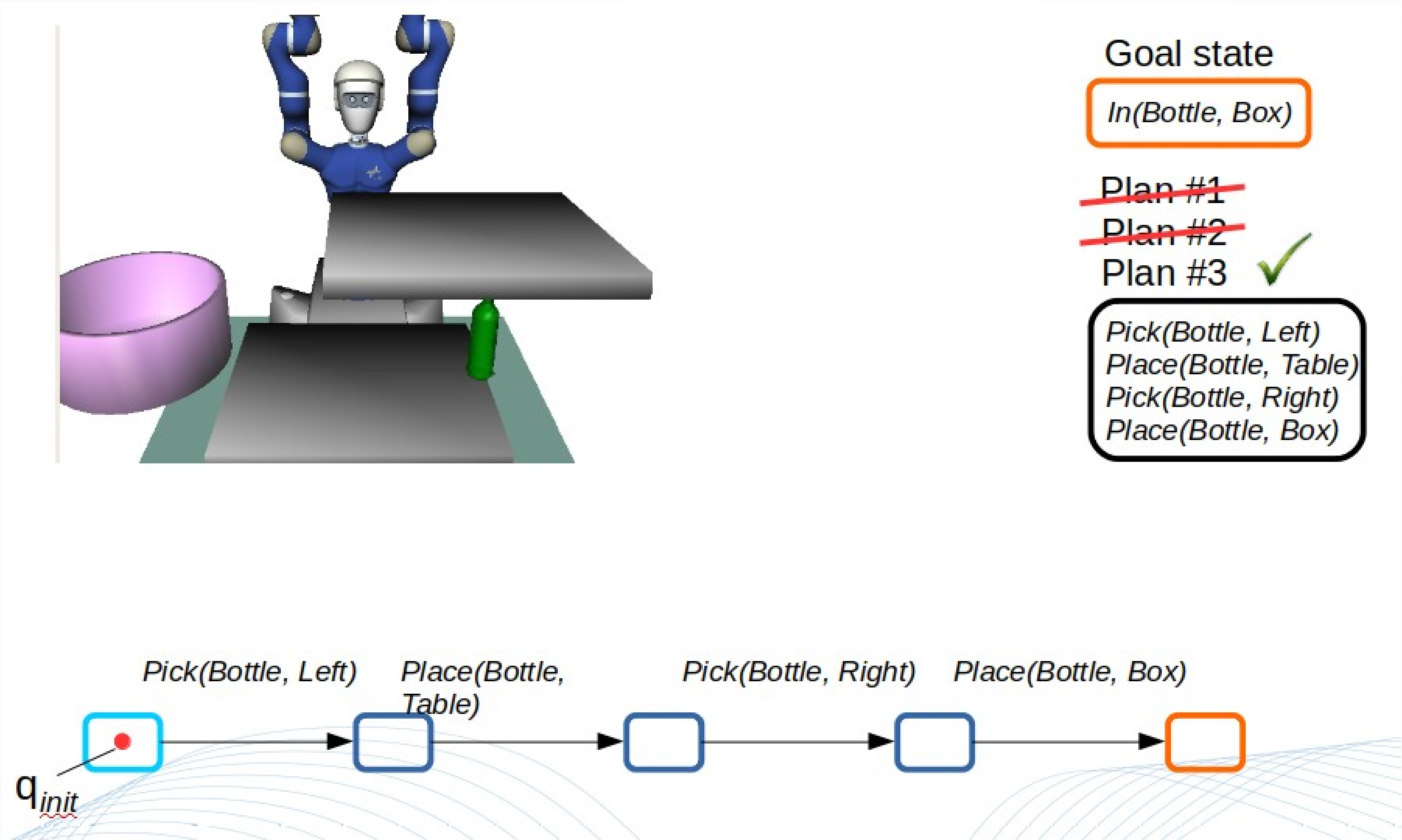
- Geometric Backtracking

  The process of reconsidering choices at the geometric level.

https://i.redd.it/p2m17shrhjuz.jpg

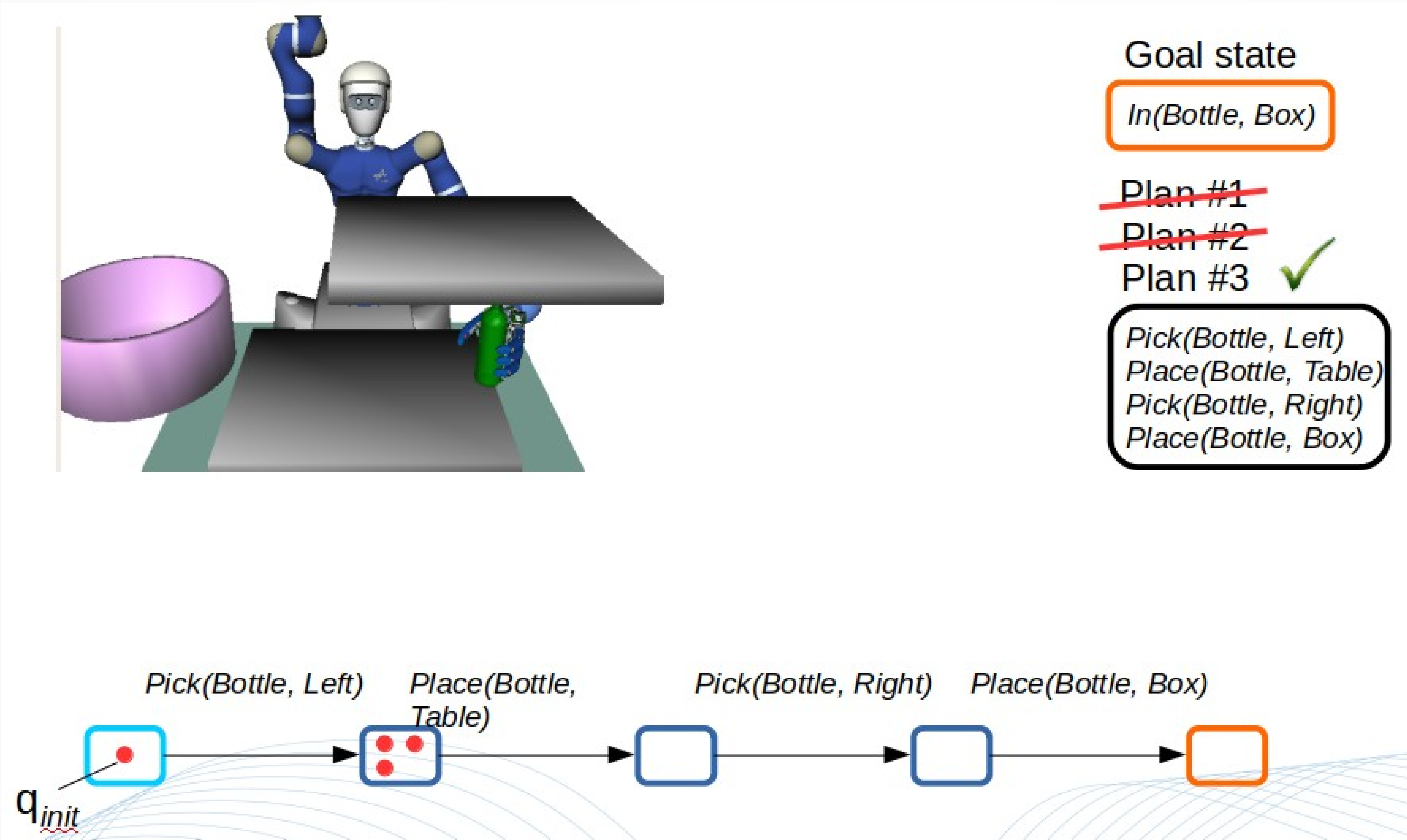
What makes it difficult?

- Geometric Backtracking



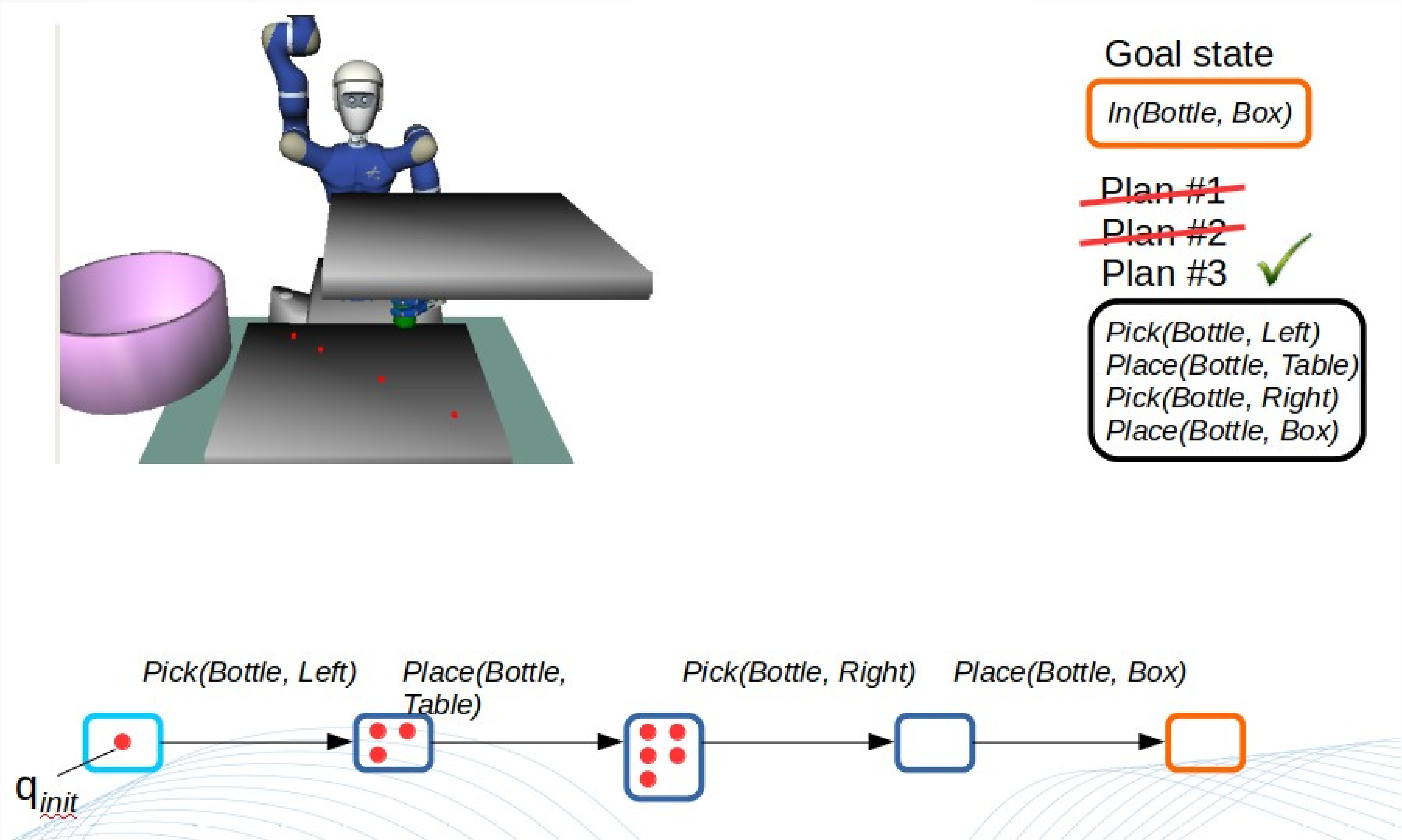
What makes it difficult?

- Geometric Backtracking



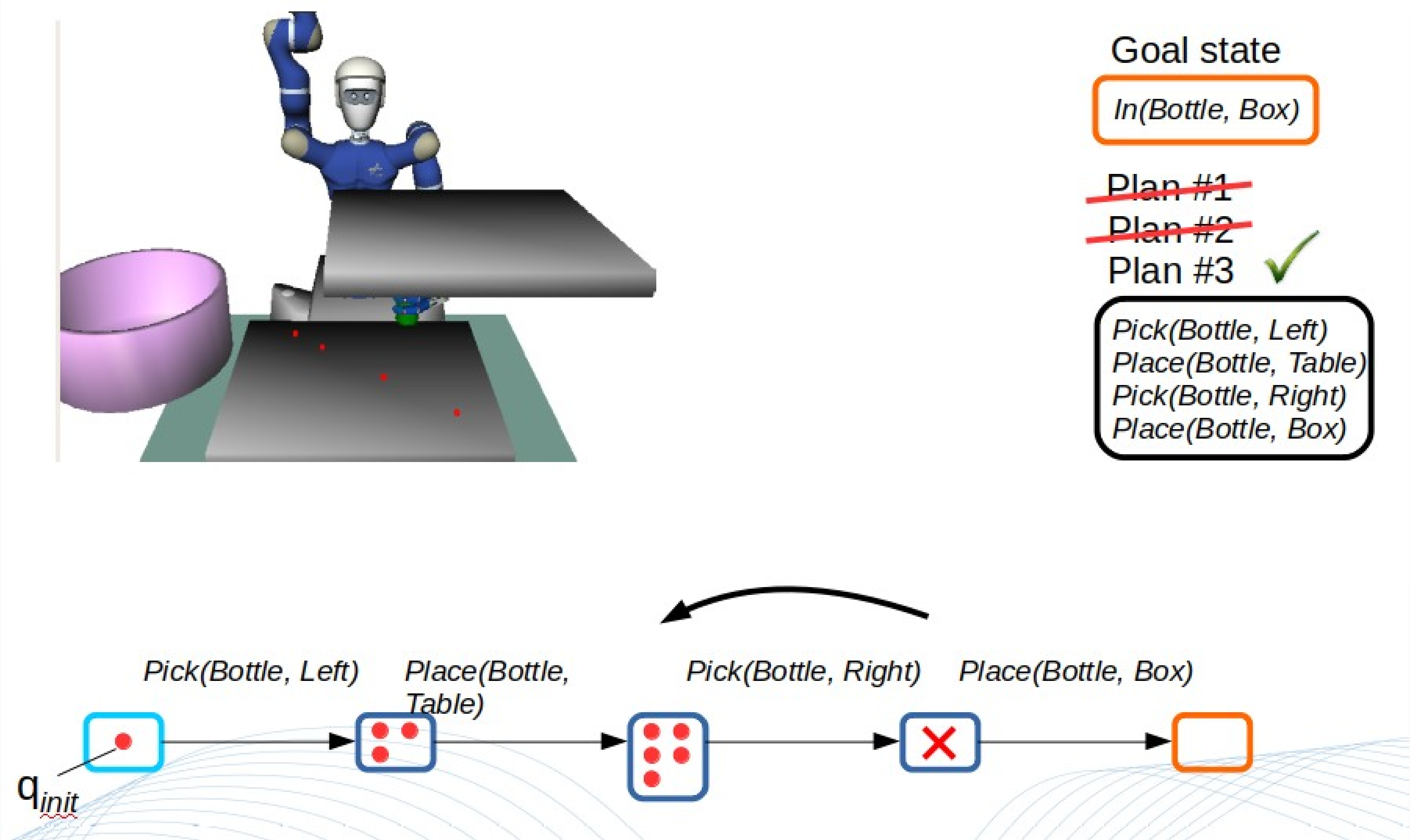
What makes it difficult?

- Geometric Backtracking



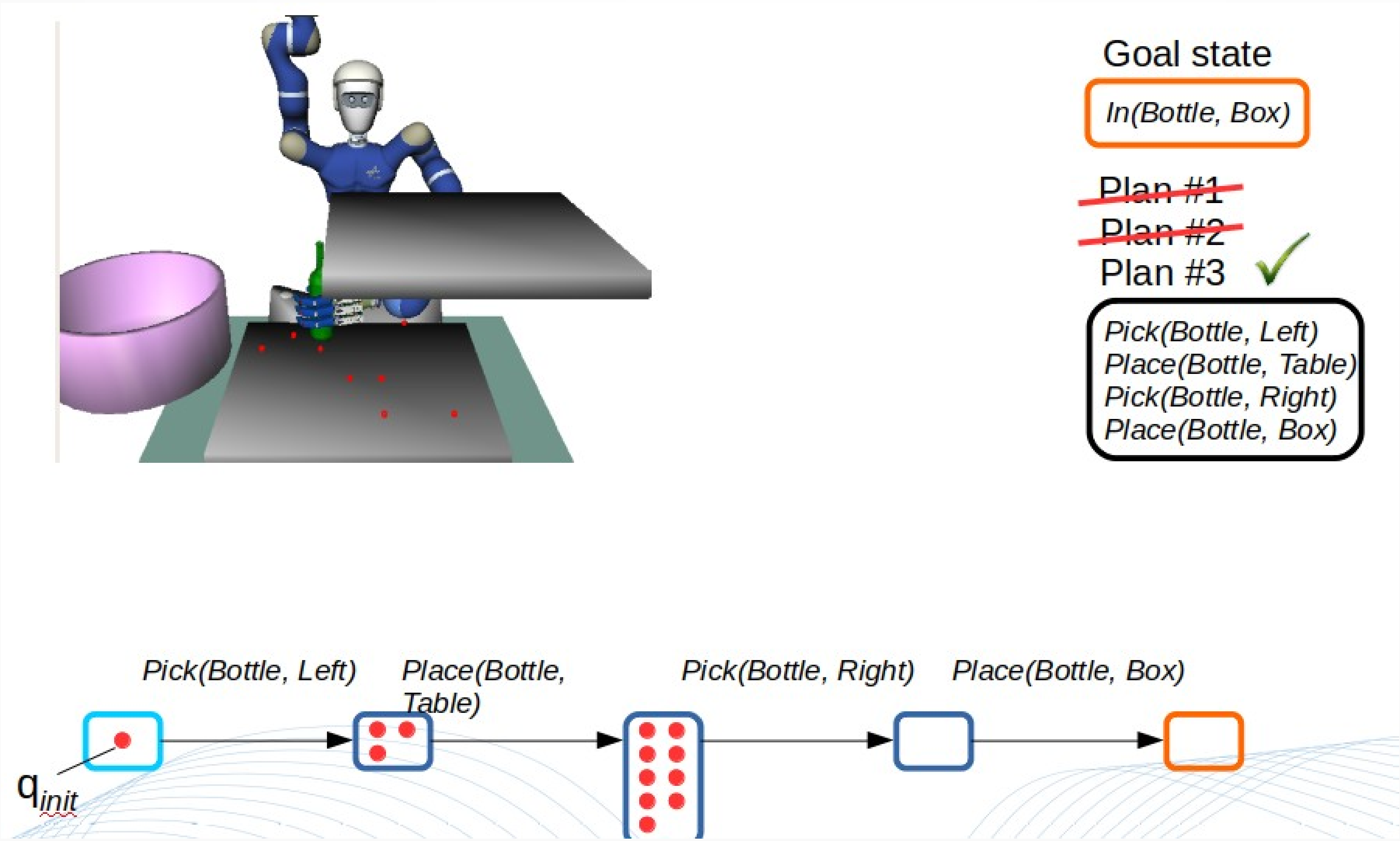
What makes it difficult?

- Geometric Backtracking



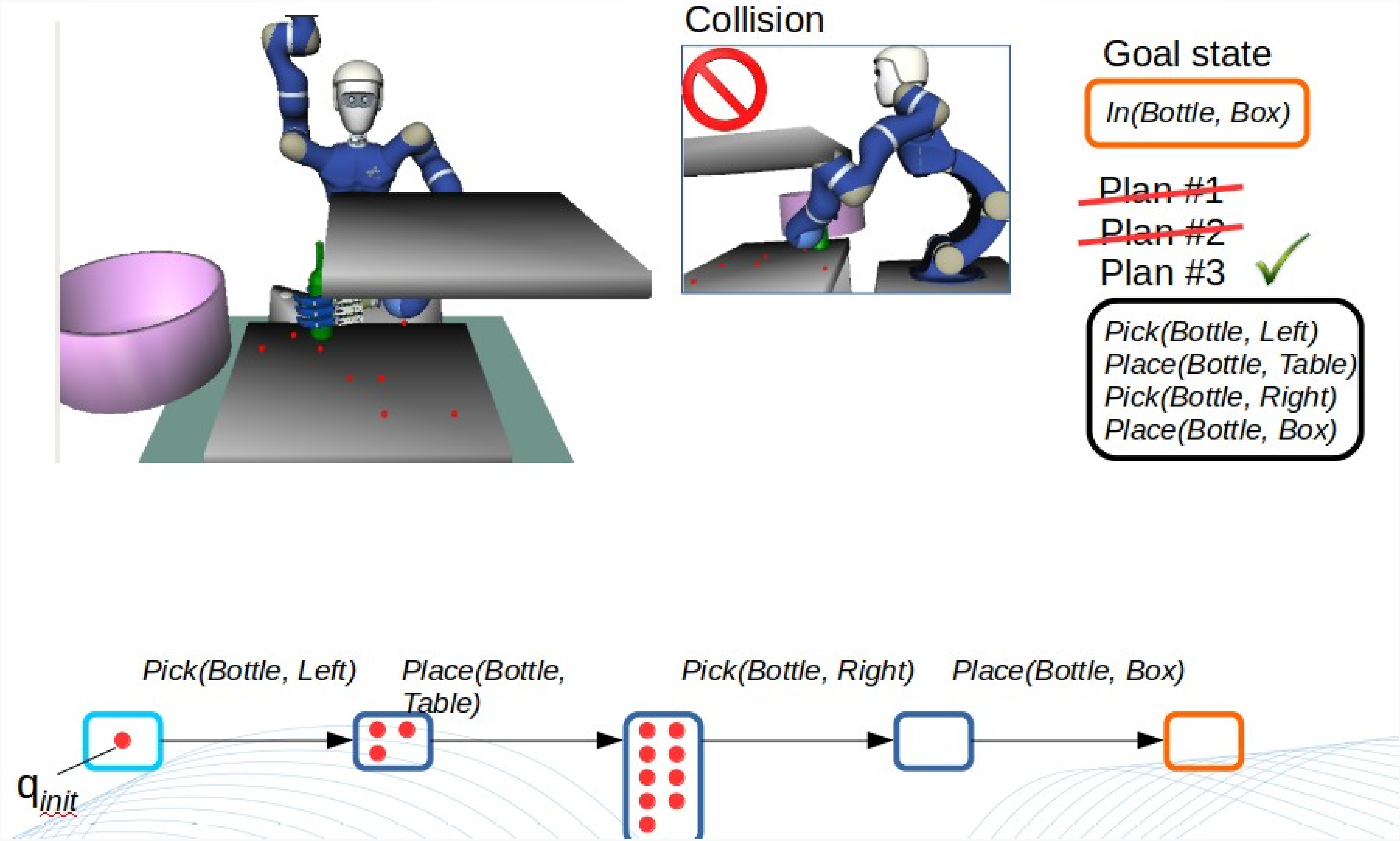
What makes it difficult?

- Geometric Backtracking



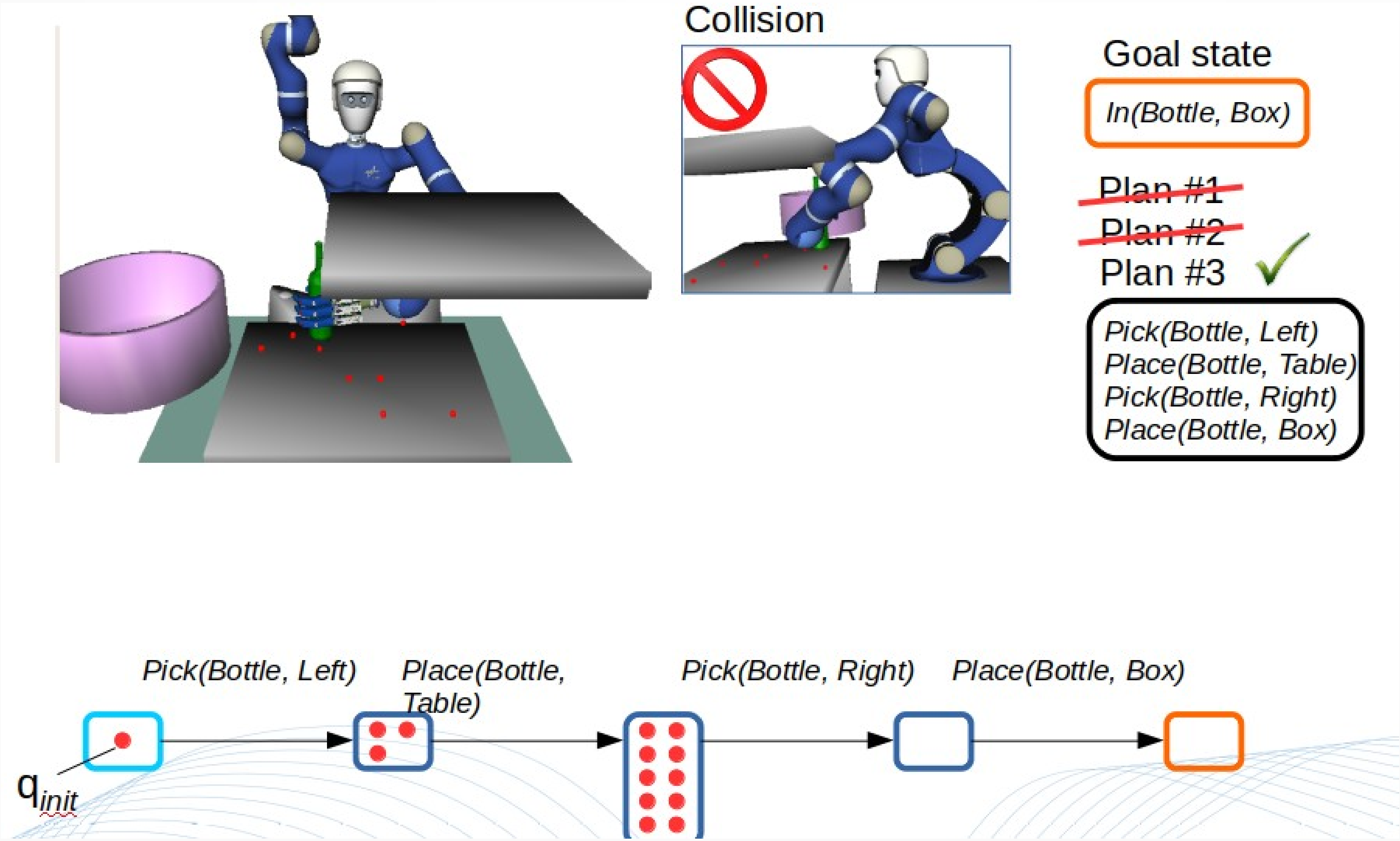
What makes it difficult?

- Geometric Backtracking



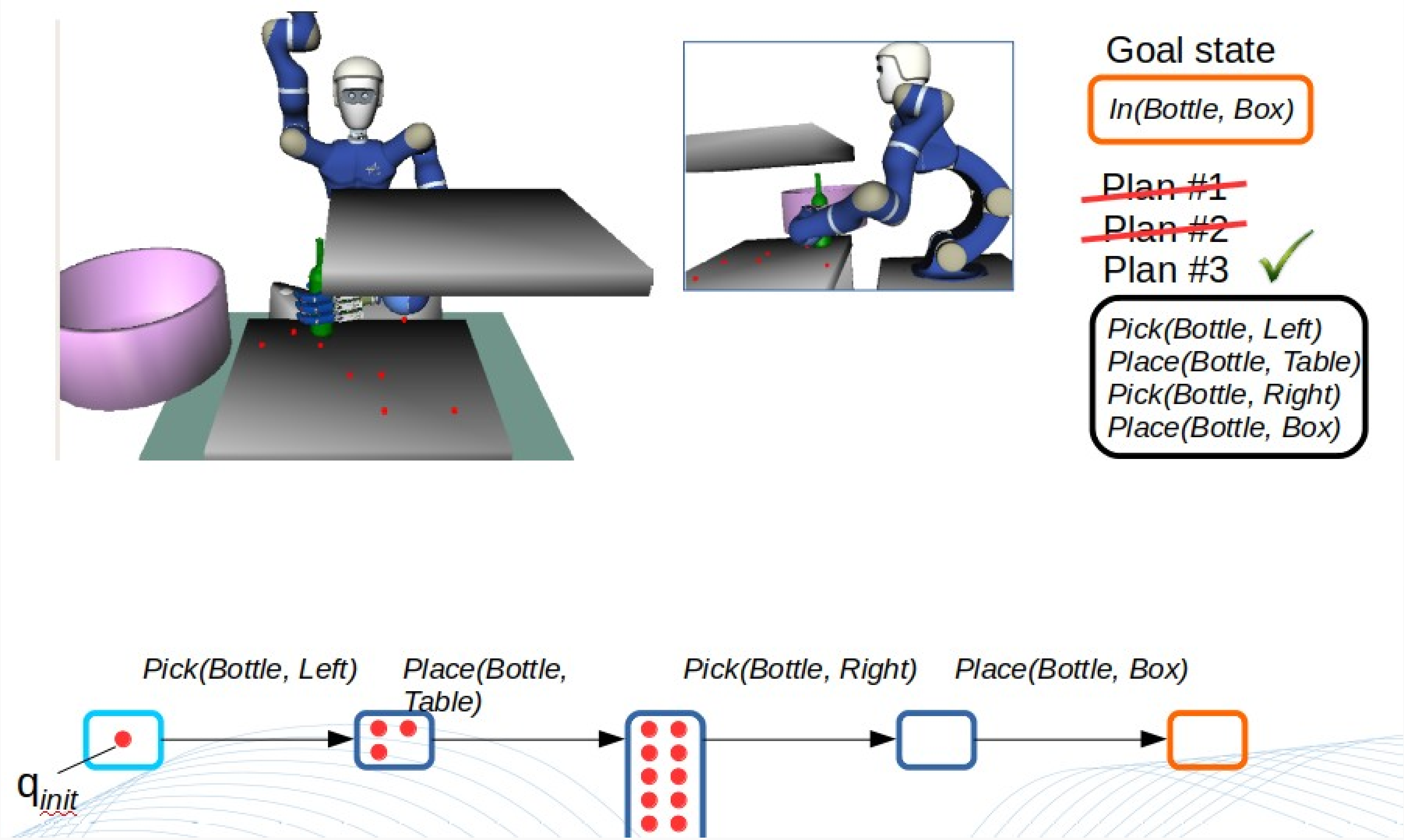
What makes it difficult?

- Geometric Backtracking



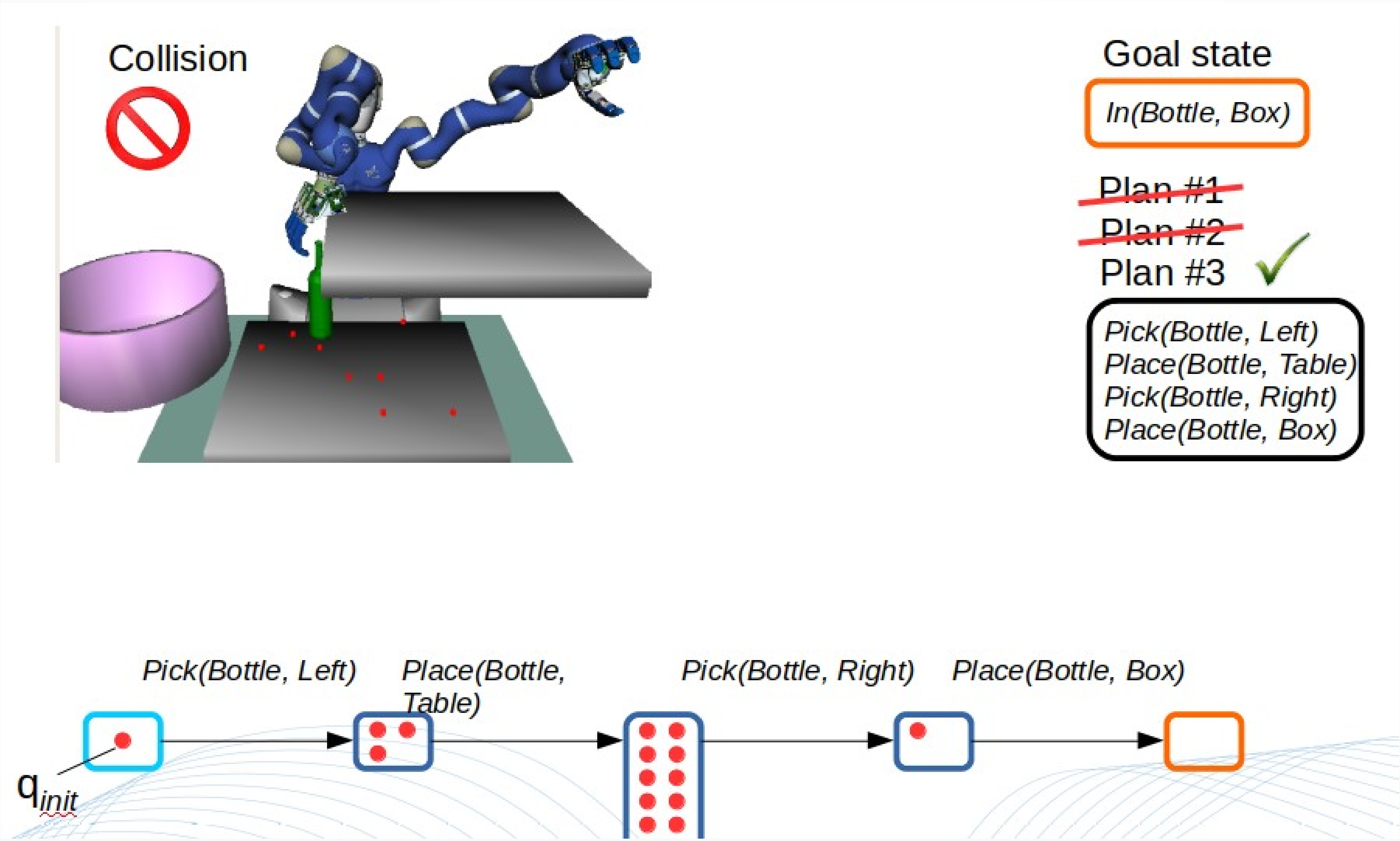
What makes it difficult?

- Geometric Backtracking



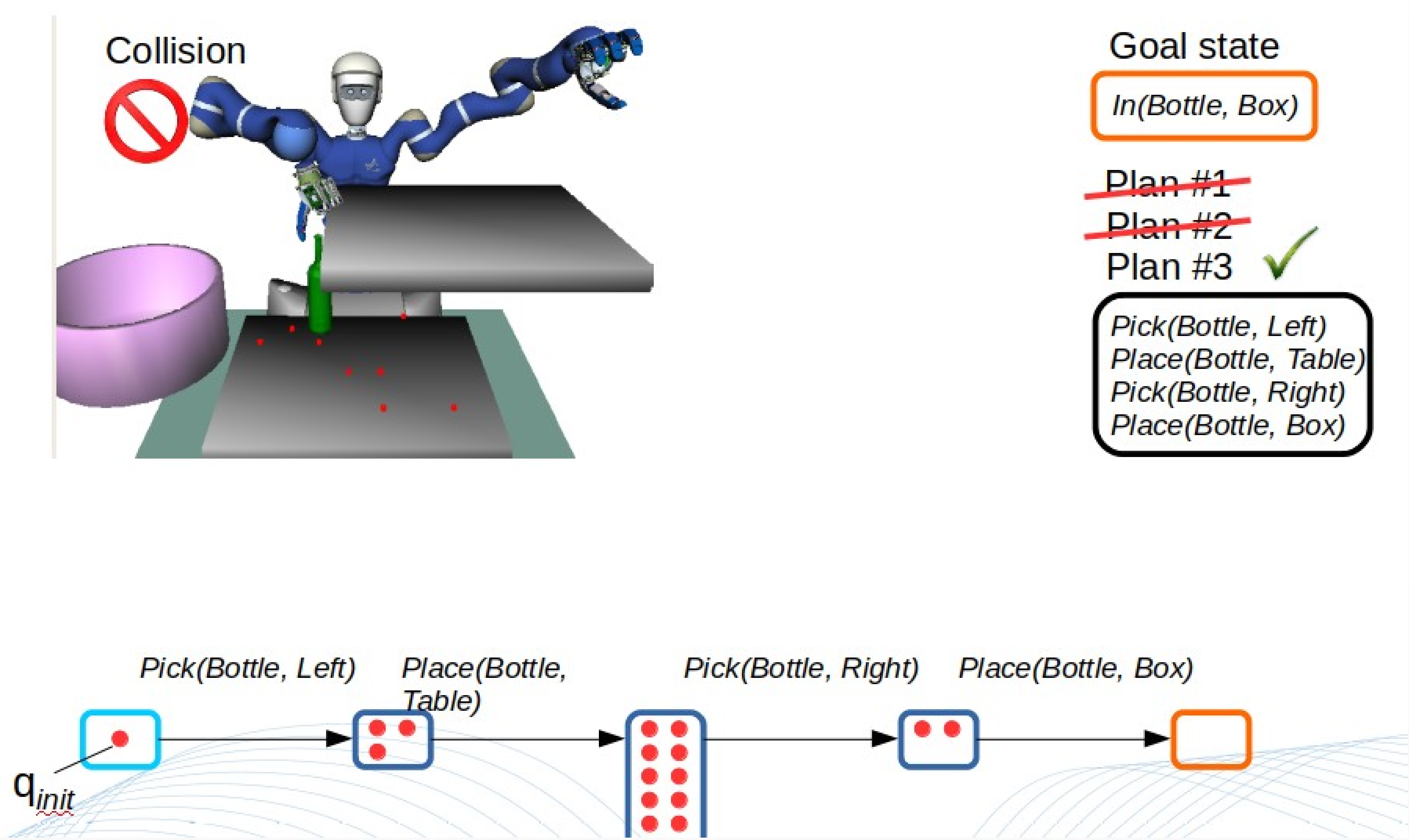
What makes it difficult?

- Geometric Backtracking



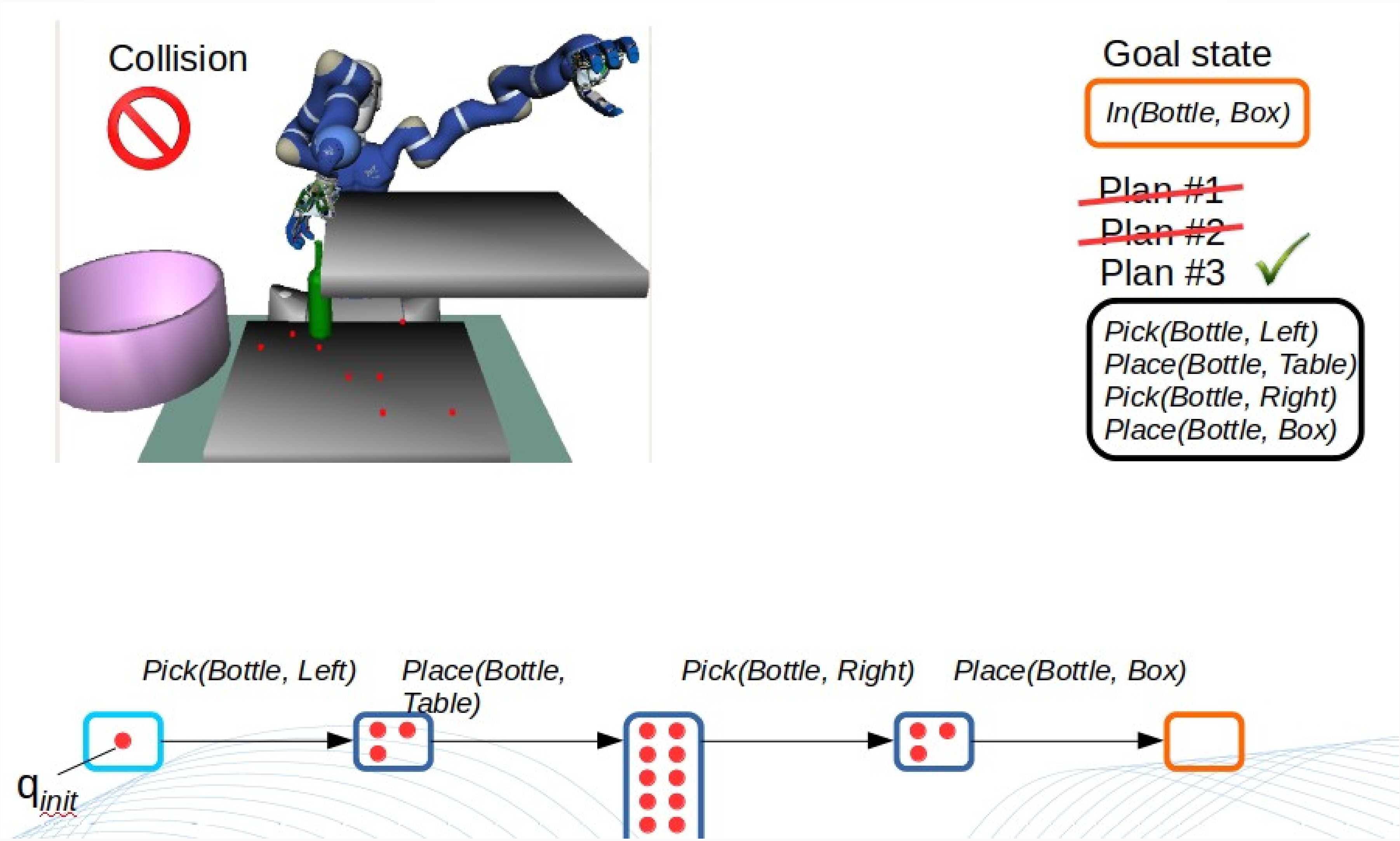
What makes it difficult?

- Geometric Backtracking



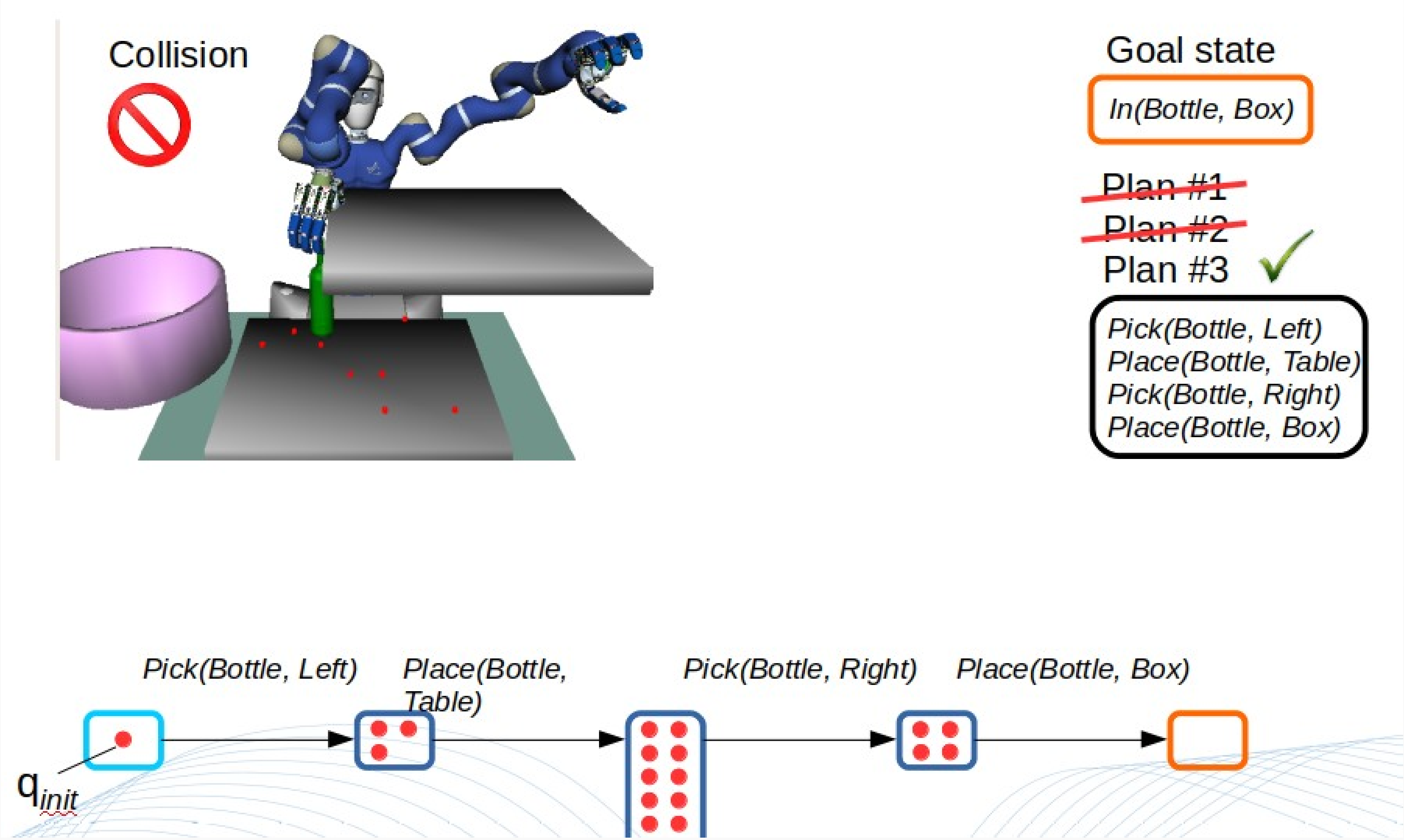
What makes it difficult?

- Geometric Backtracking



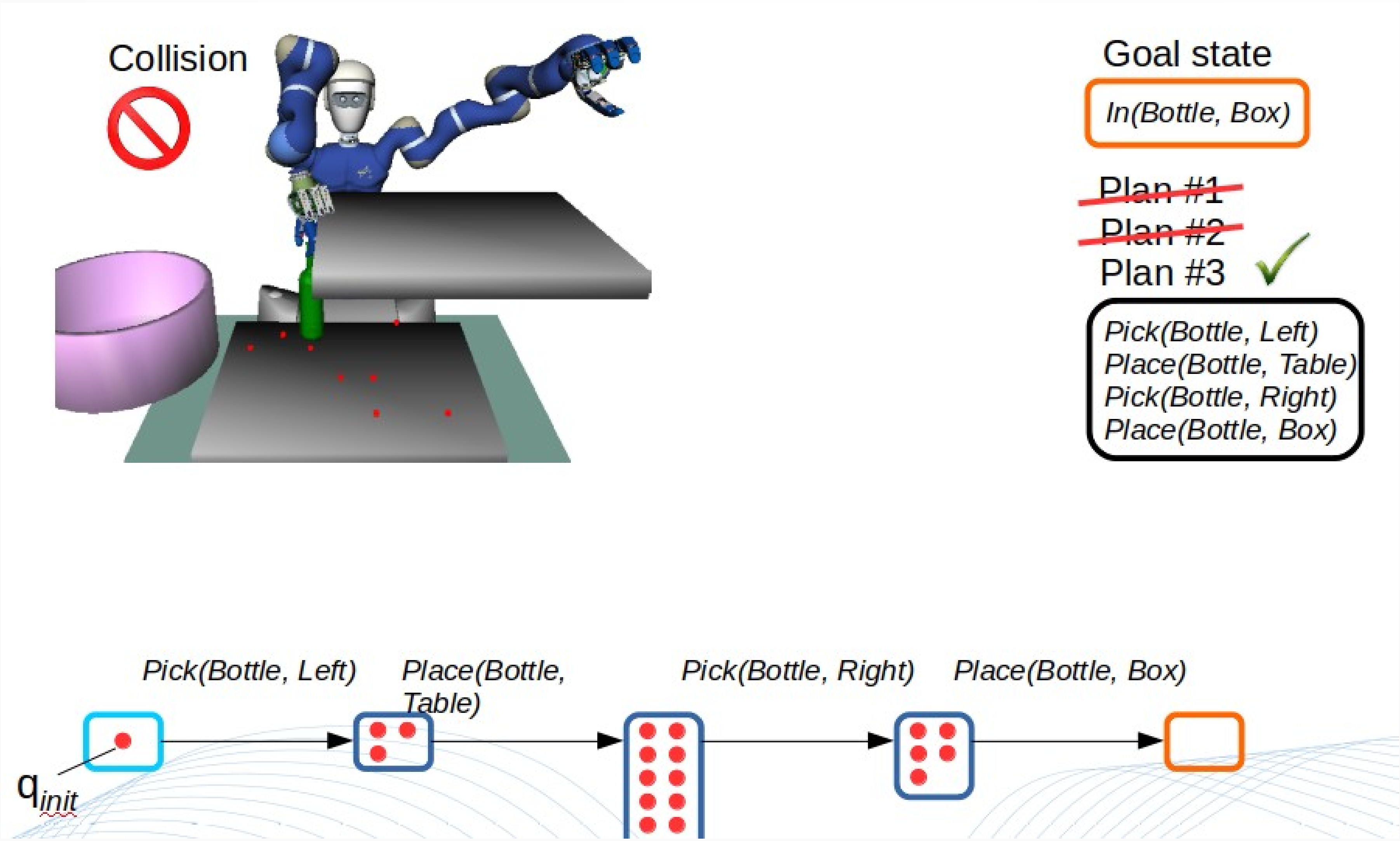
What makes it difficult?

- Geometric Backtracking



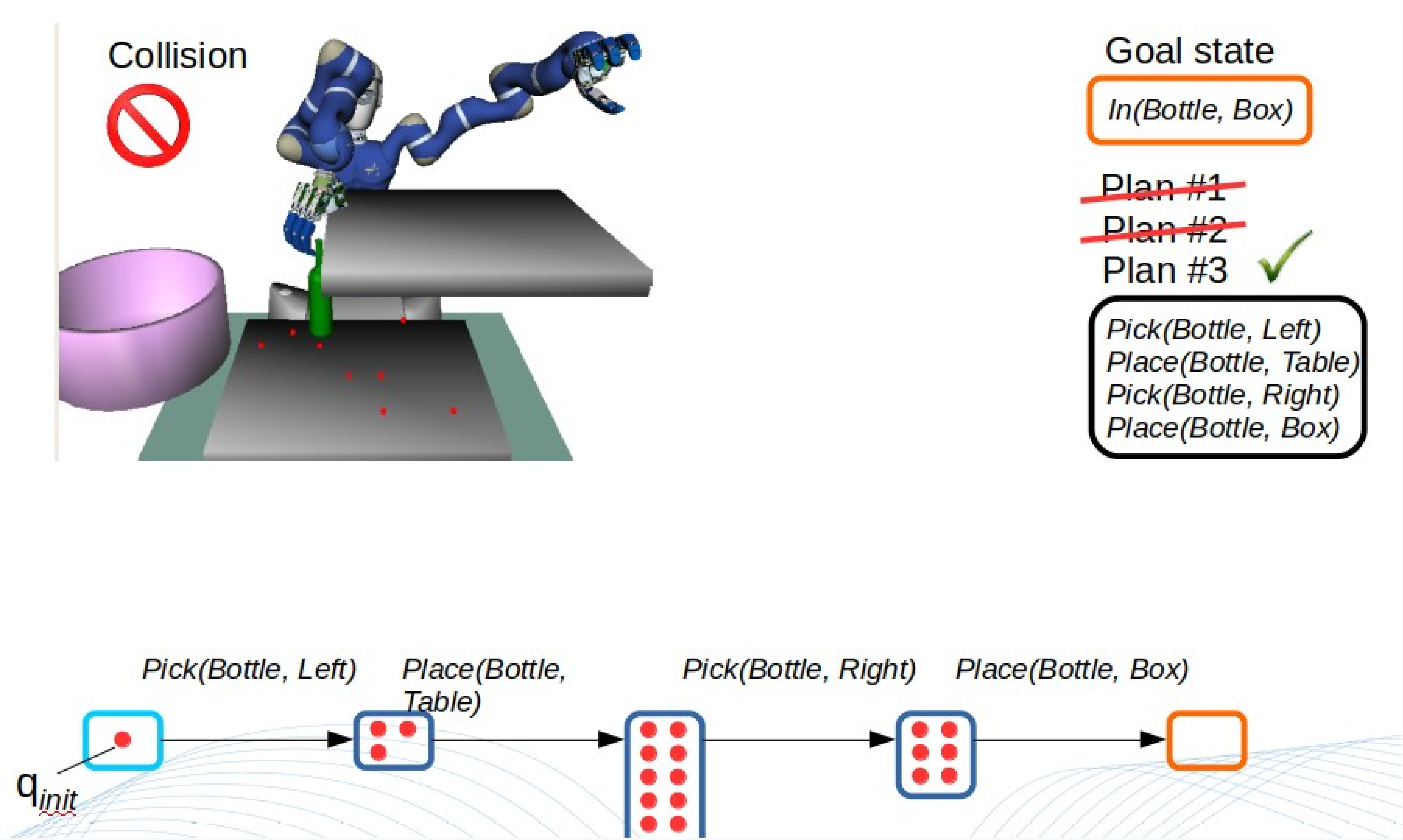
What makes it difficult?

- Geometric Backtracking



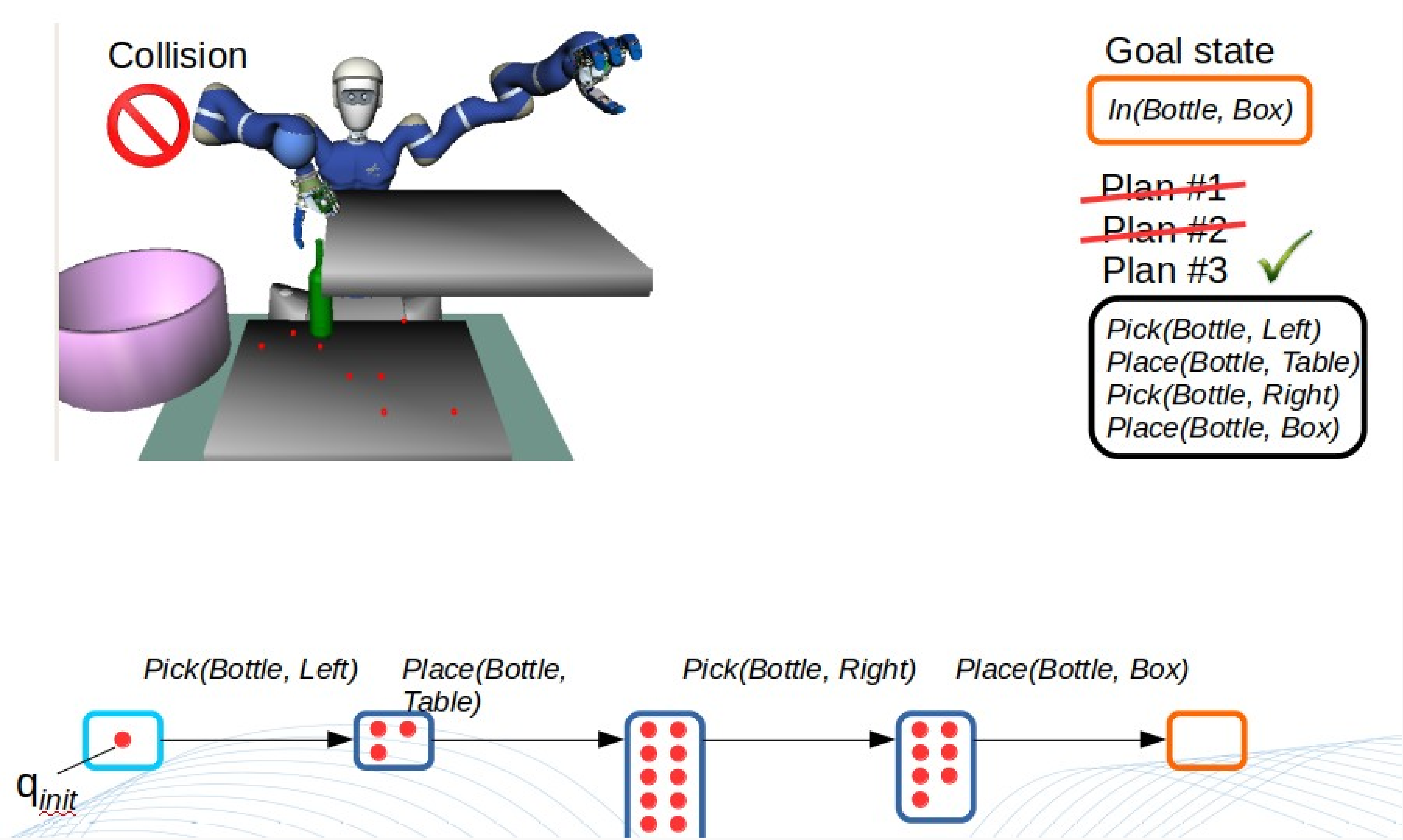
What makes it difficult?

- Geometric Backtracking



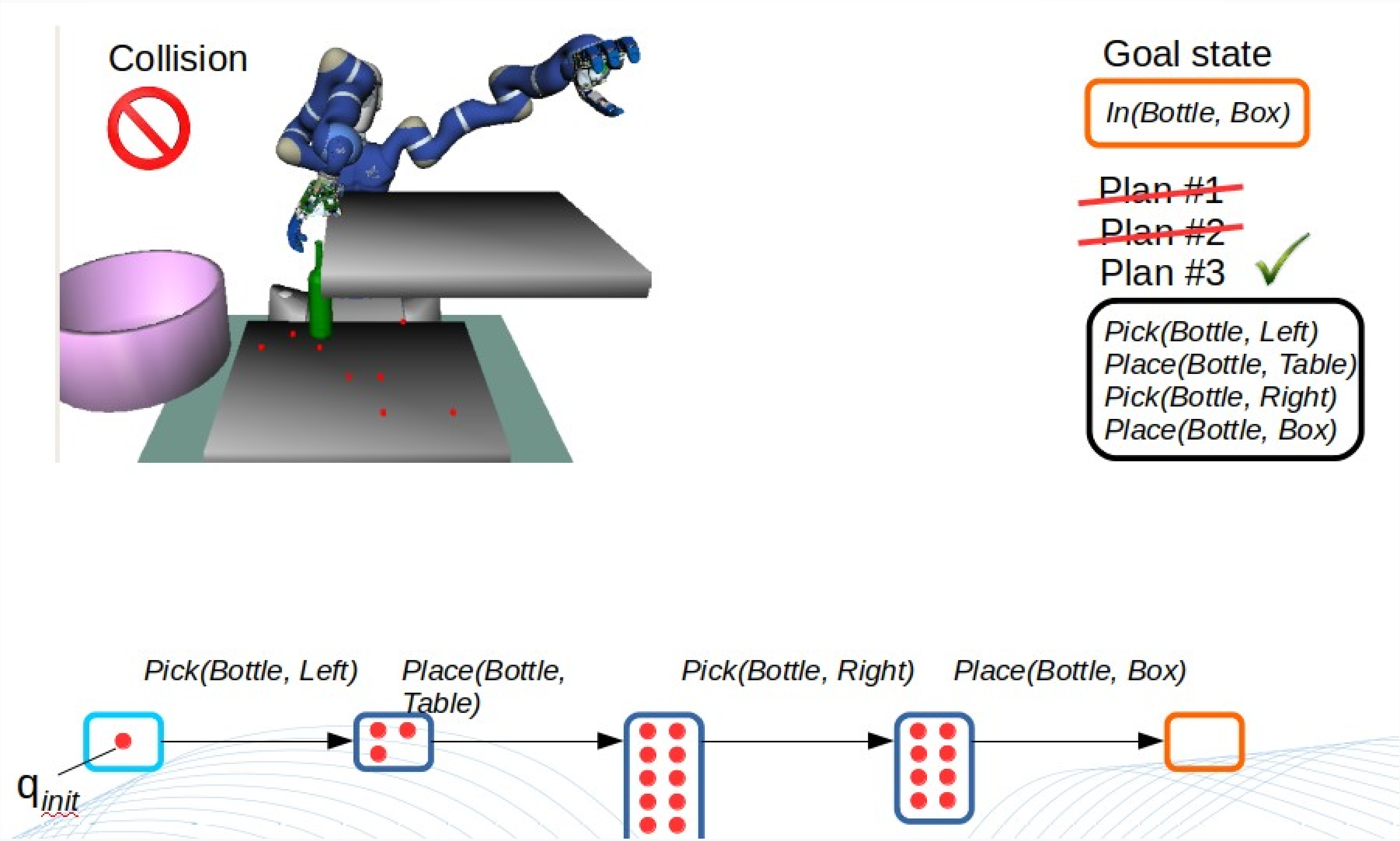
What makes it difficult?

- Geometric Backtracking



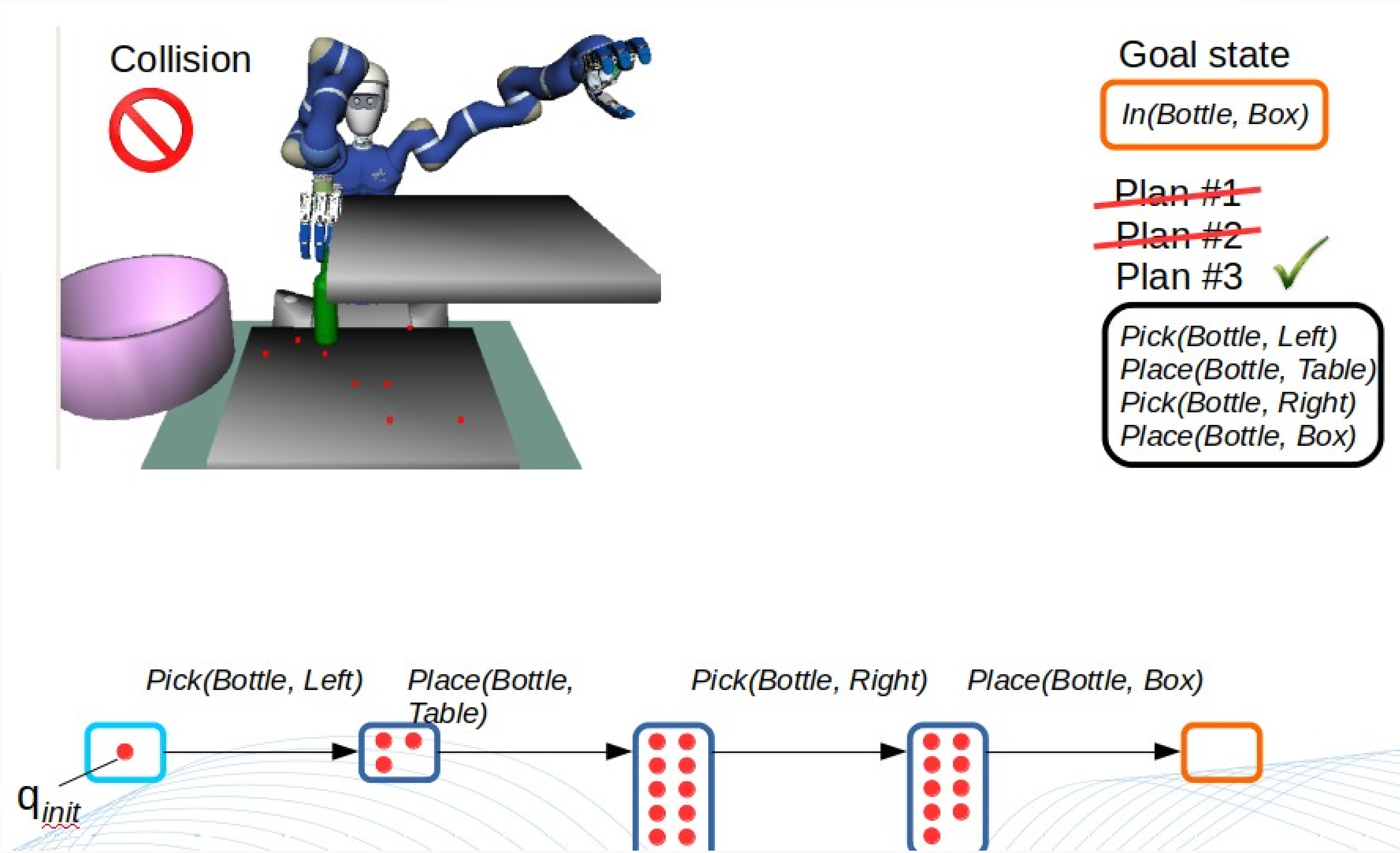
What makes it difficult?

- Geometric Backtracking



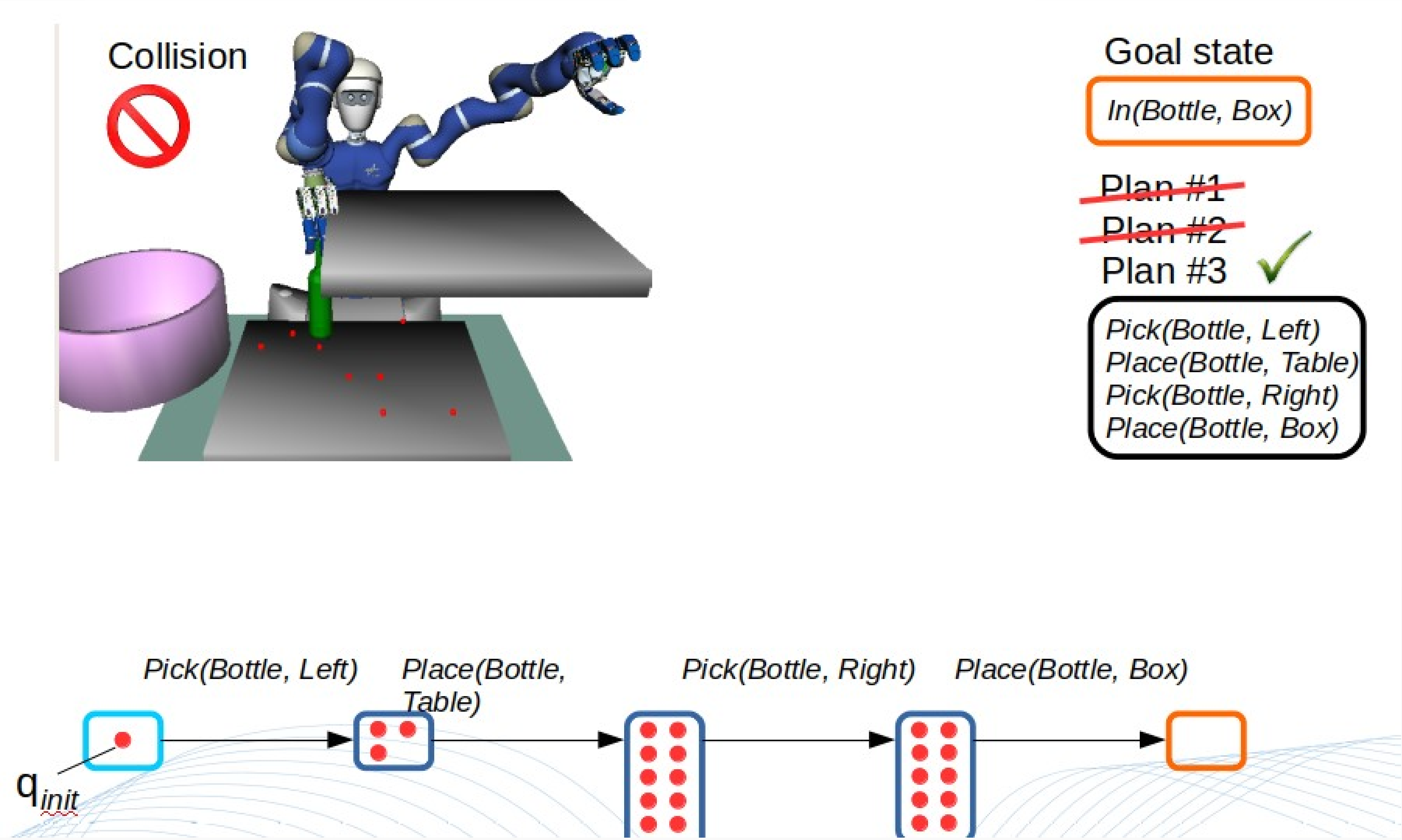
What makes it difficult?

- Geometric Backtracking



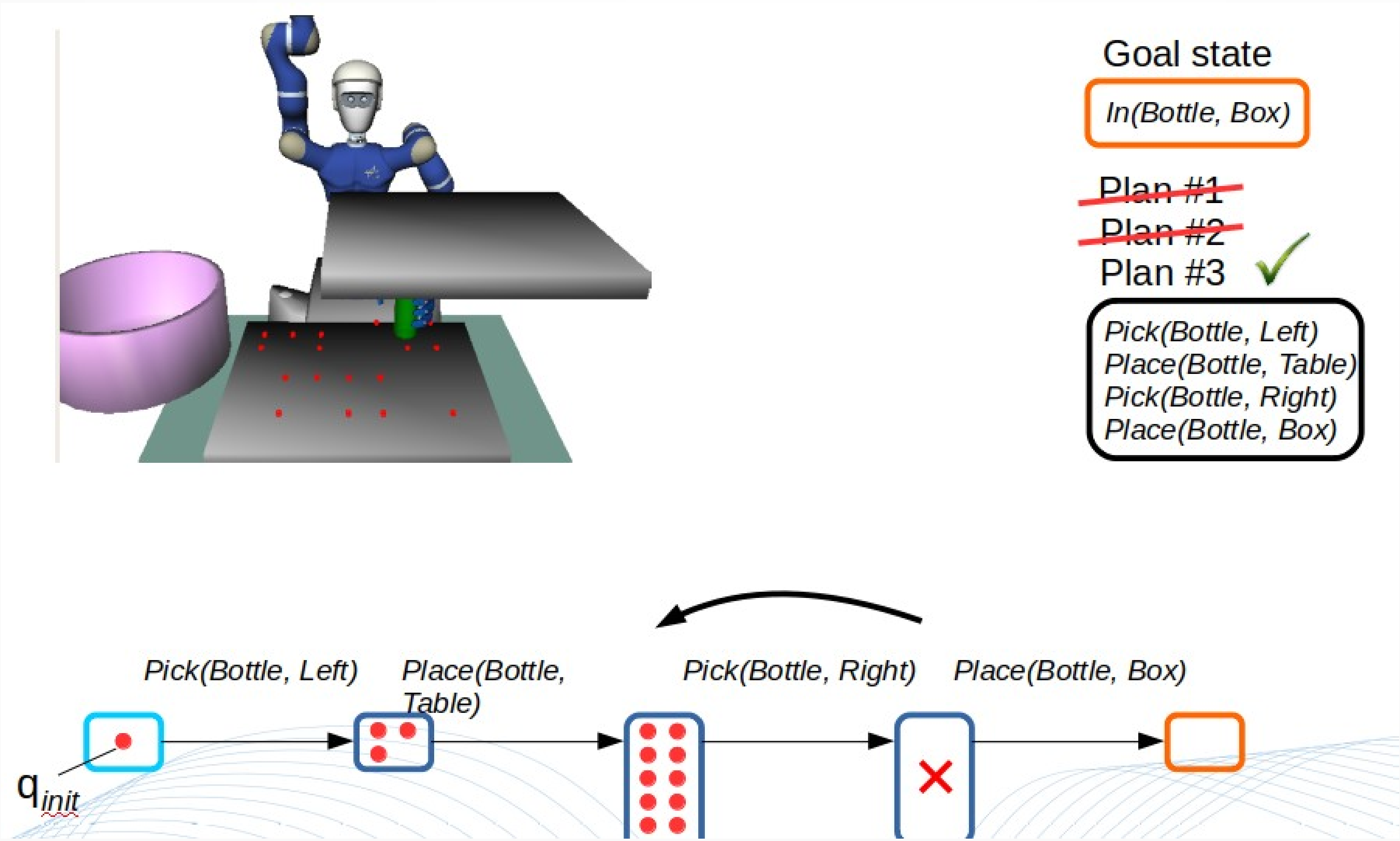
What makes it difficult?

- Geometric Backtracking



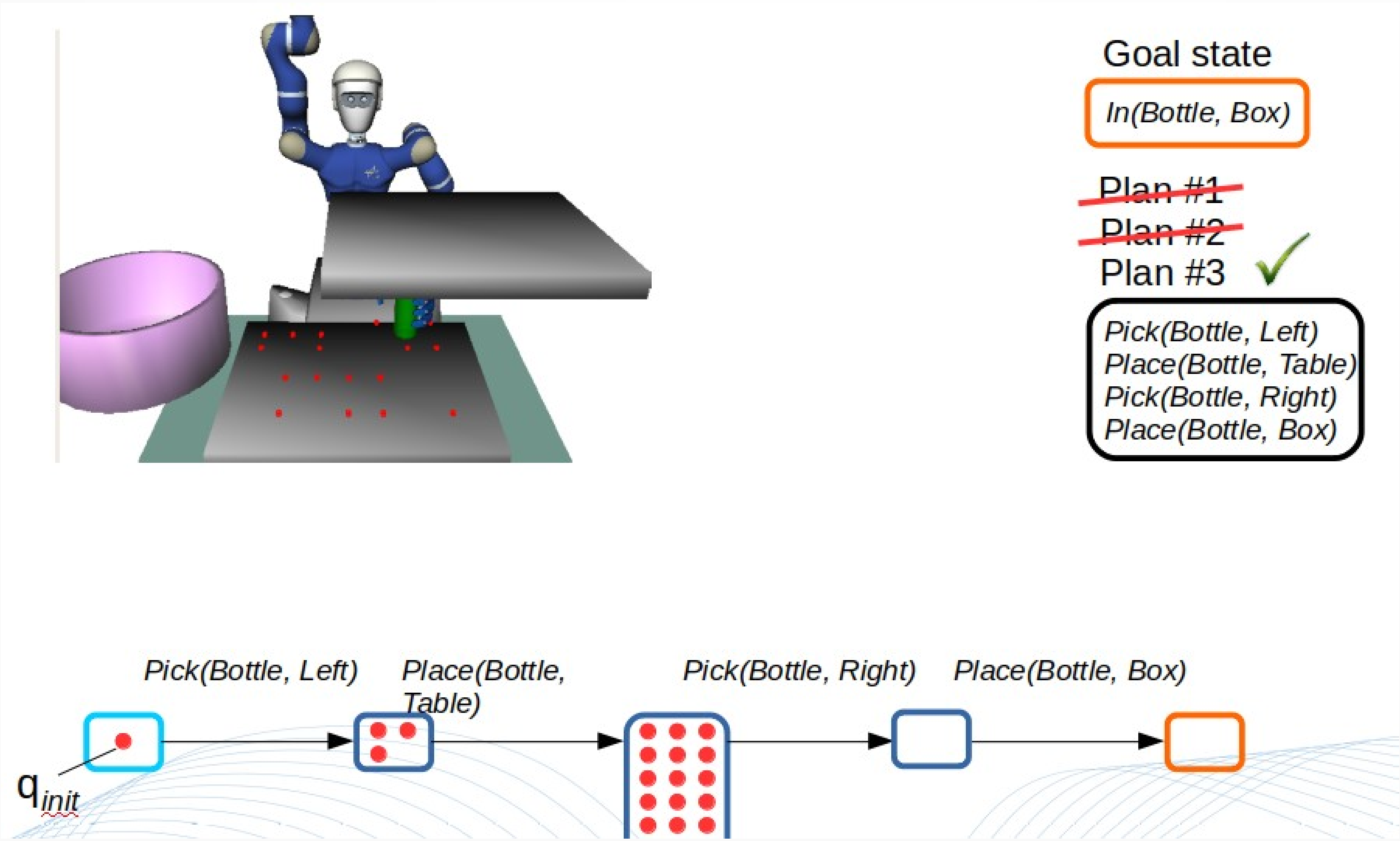
What makes it difficult?

- Geometric Backtracking



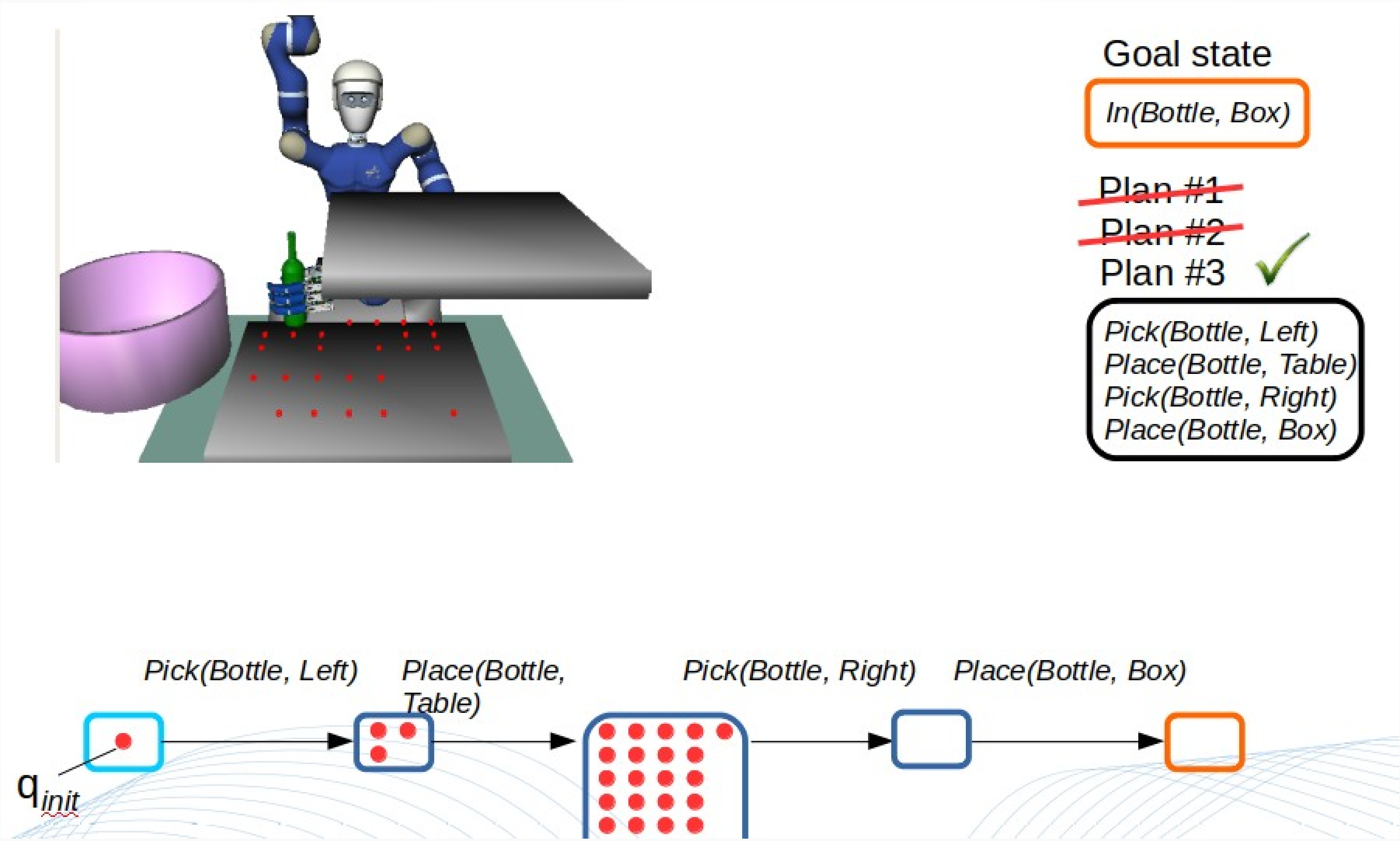
What makes it difficult?

- Geometric Backtracking



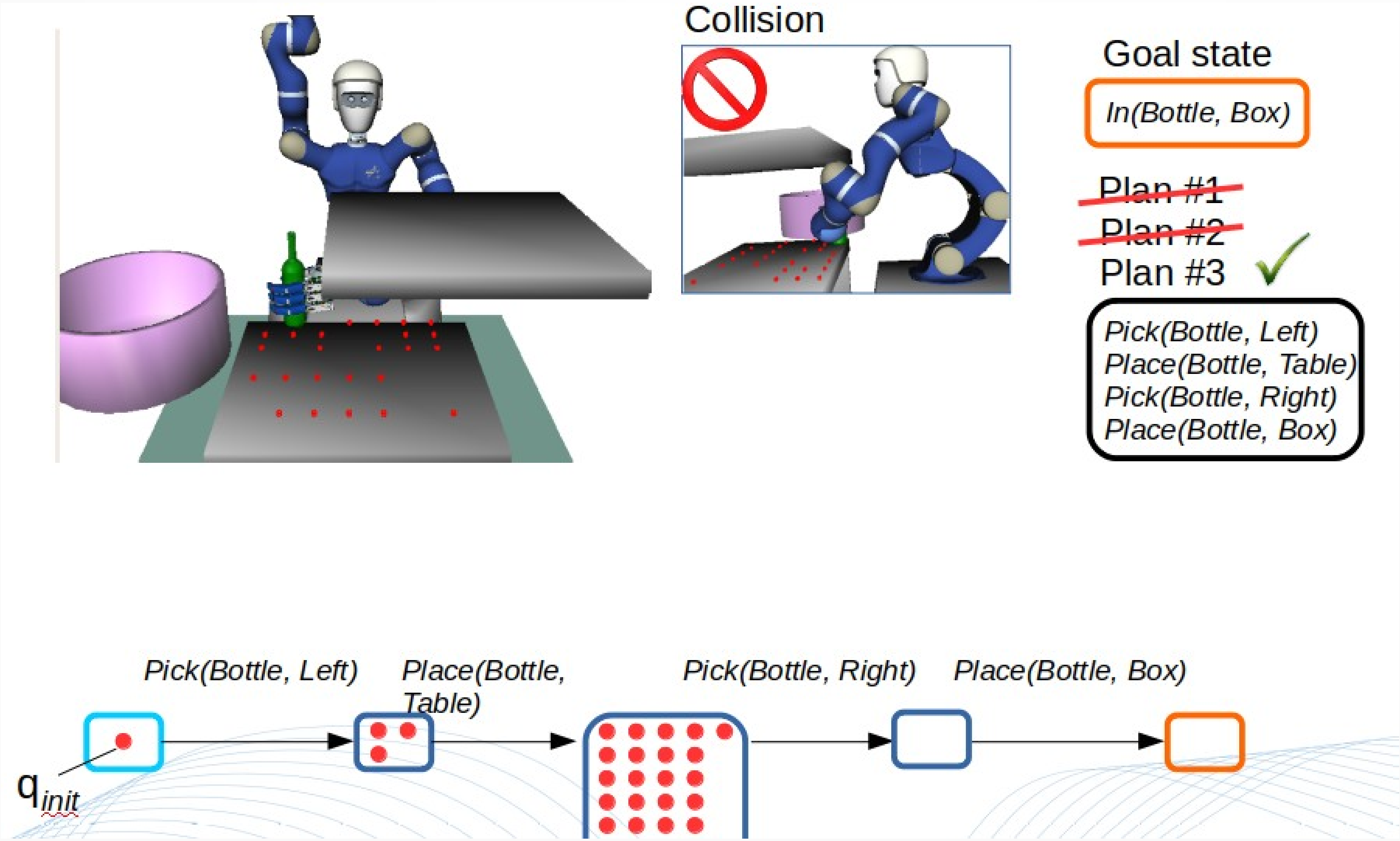
What makes it difficult?

- Geometric Backtracking



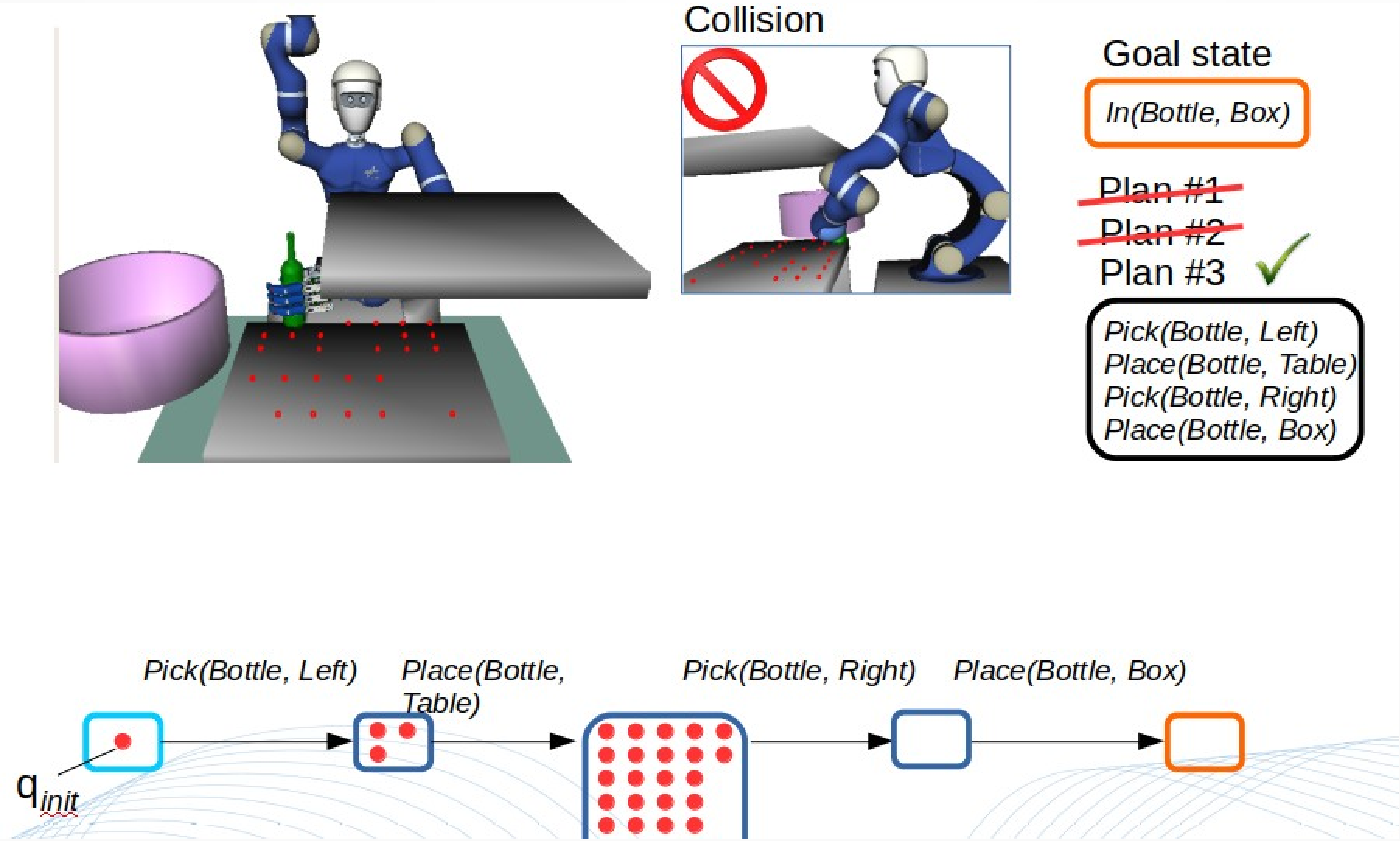
What makes it difficult?

- Geometric Backtracking



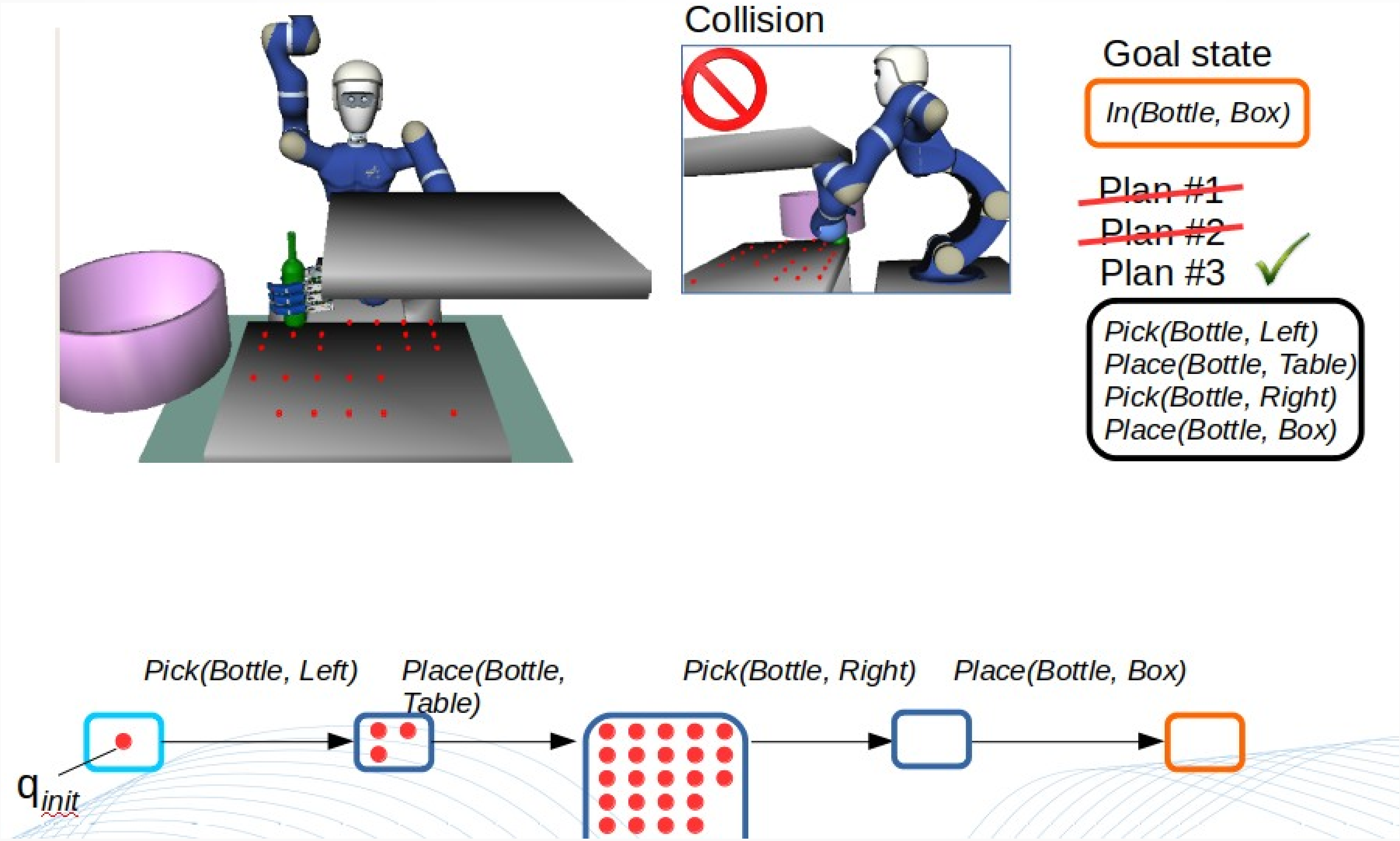
What makes it difficult?

- Geometric Backtracking



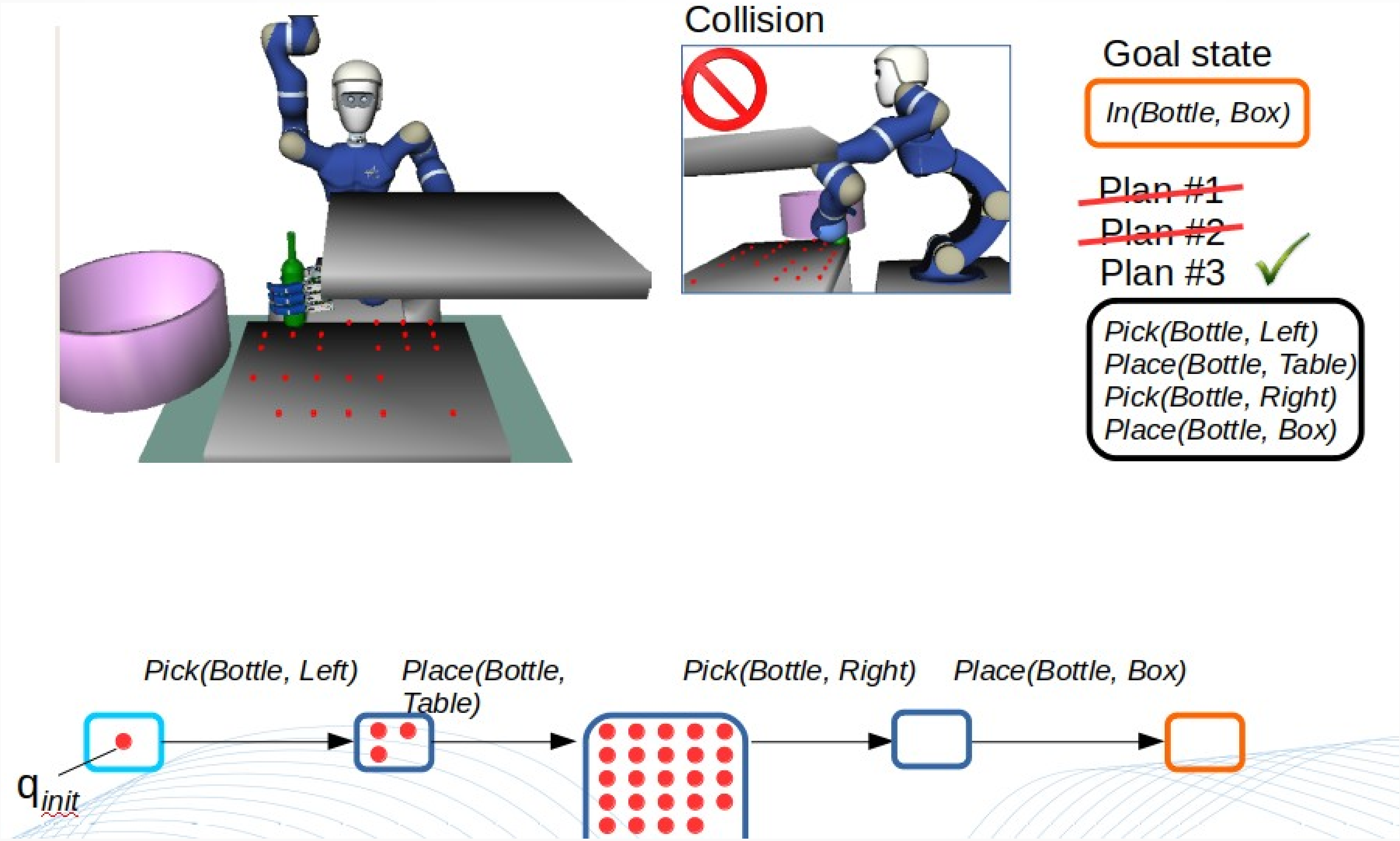
What makes it difficult?

- Geometric Backtracking



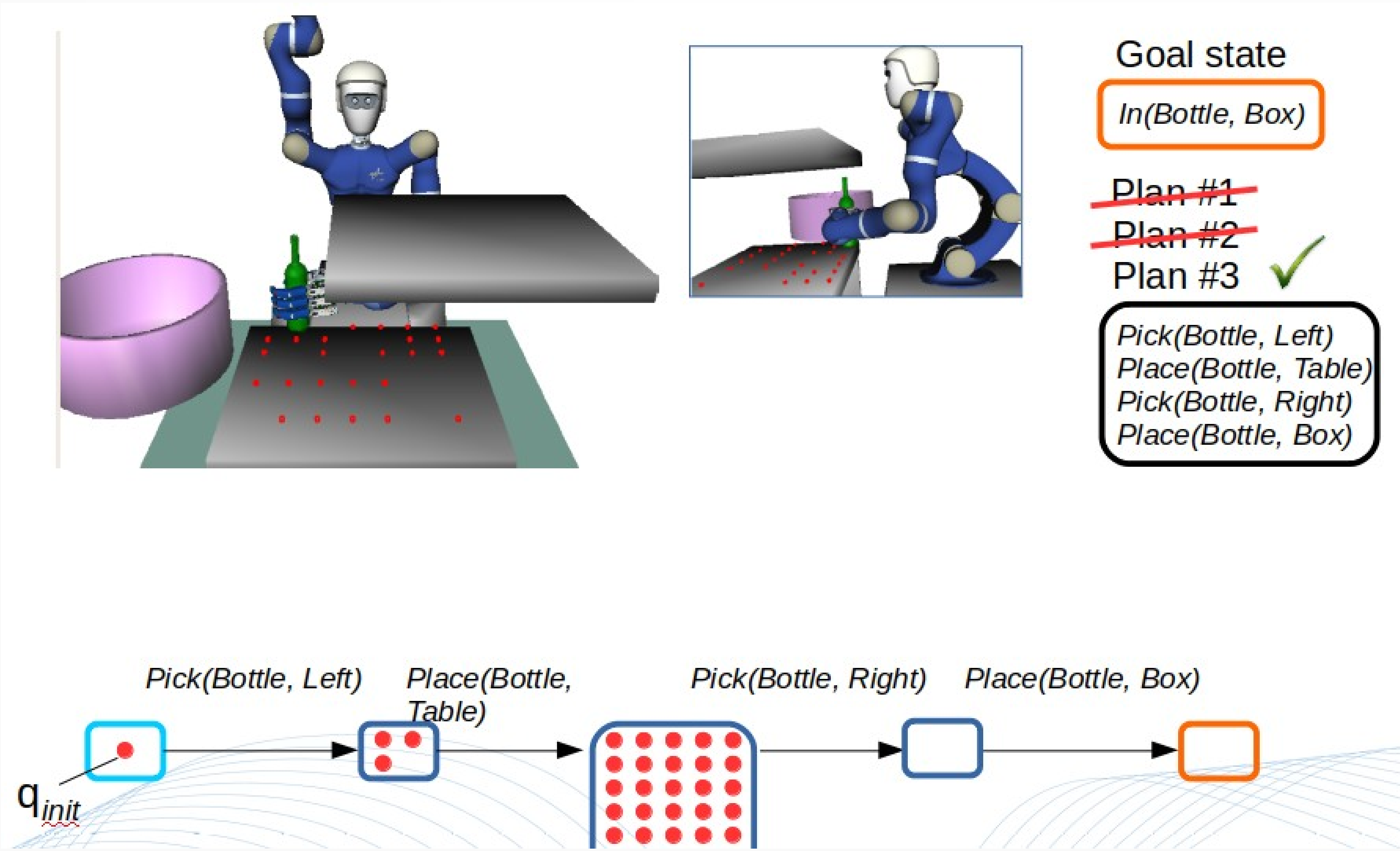
What makes it difficult?

- Geometric Backtracking



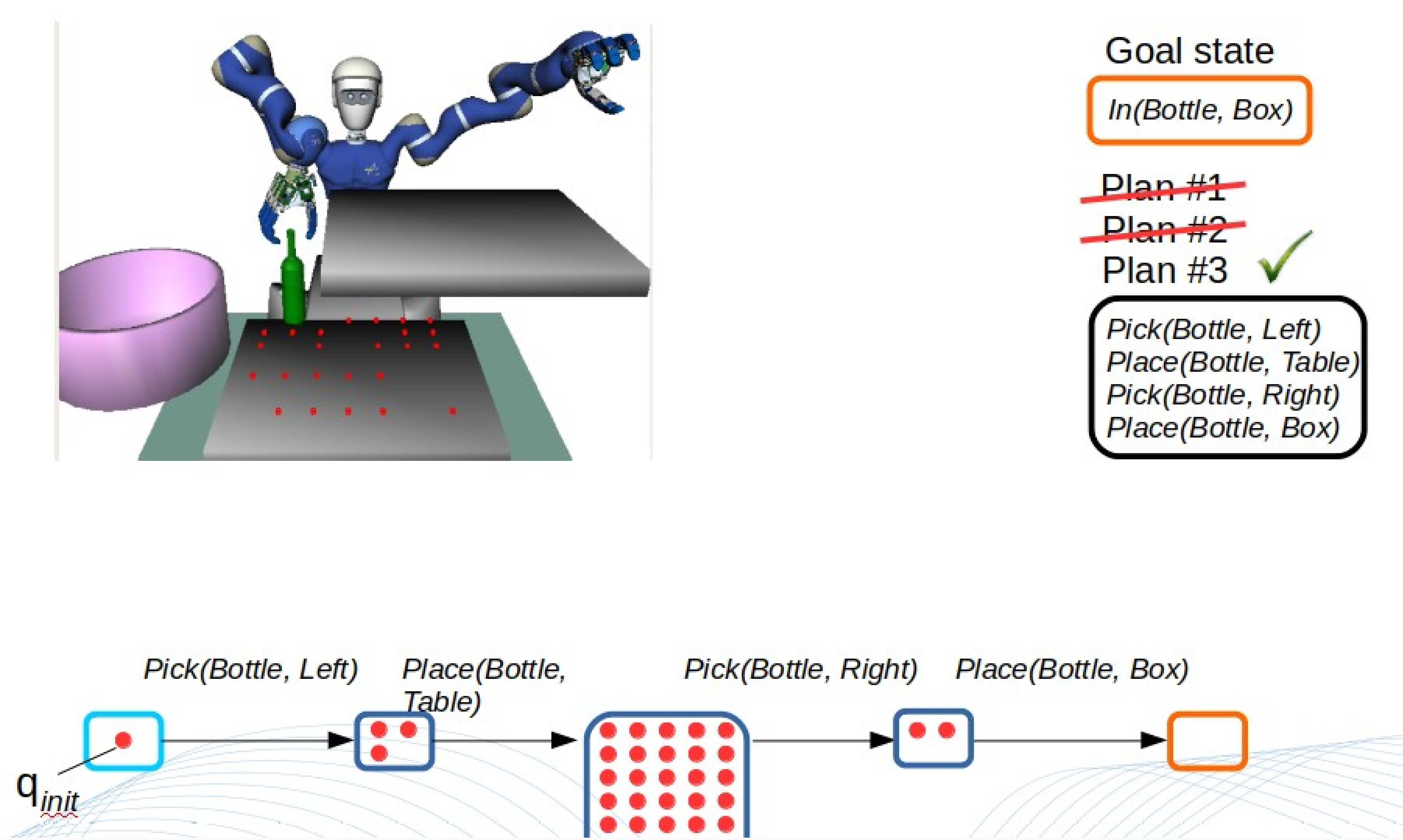
What makes it difficult?

- Geometric Backtracking



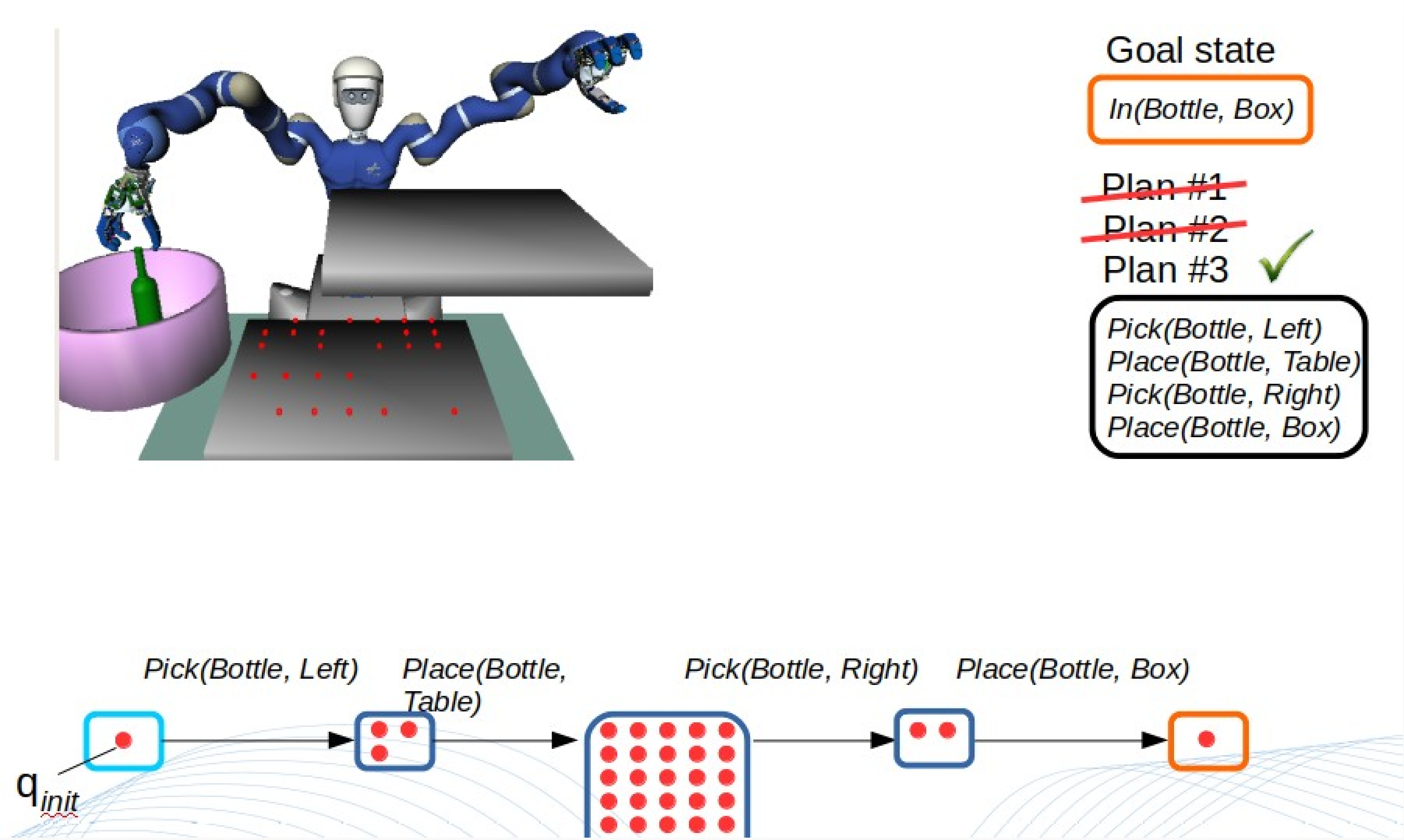
What makes it difficult?

- Geometric Backtracking



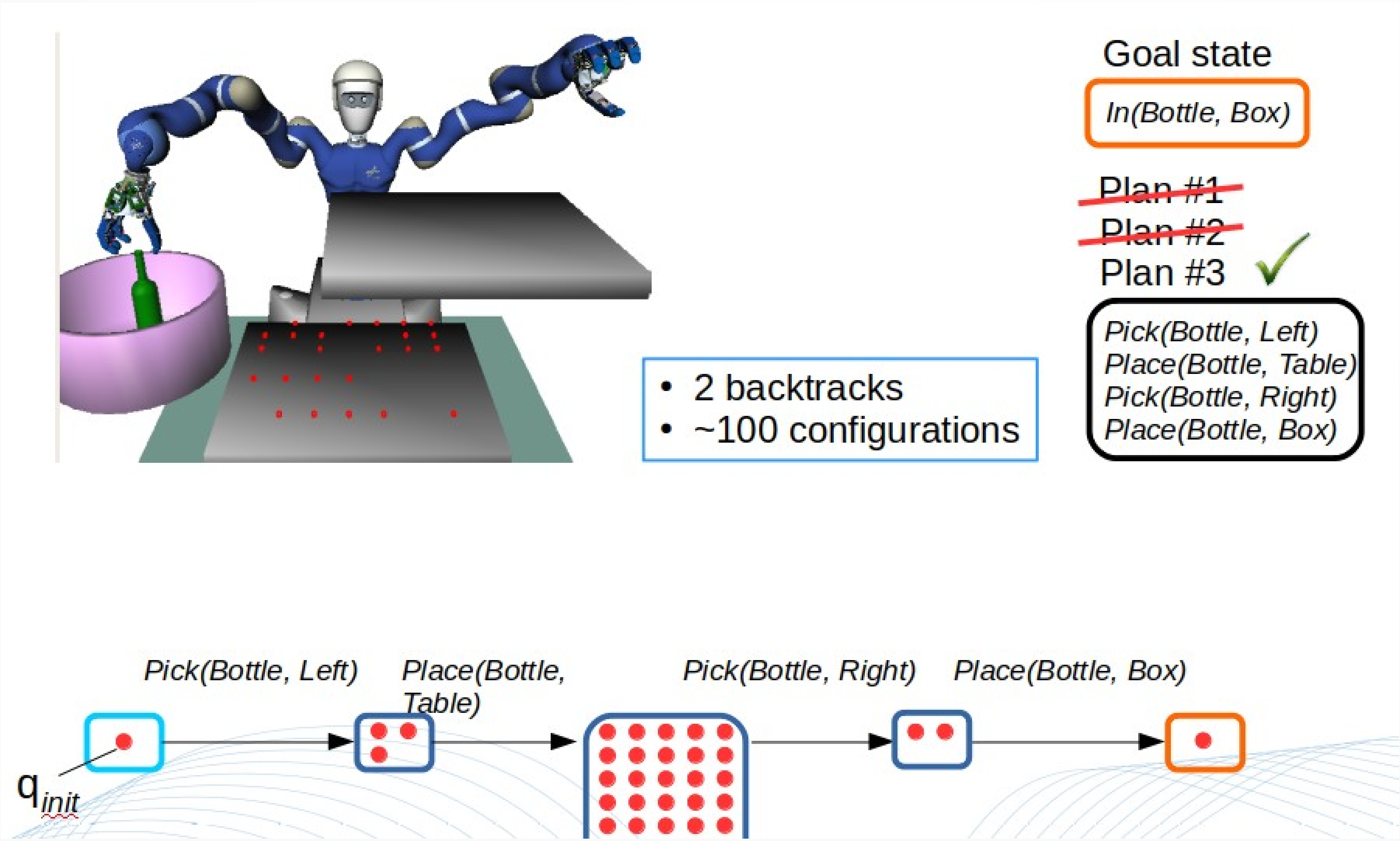
What makes it difficult?

- Geometric Backtracking



What makes it difficult?

- Geometric Backtracking



What makes it difficult?

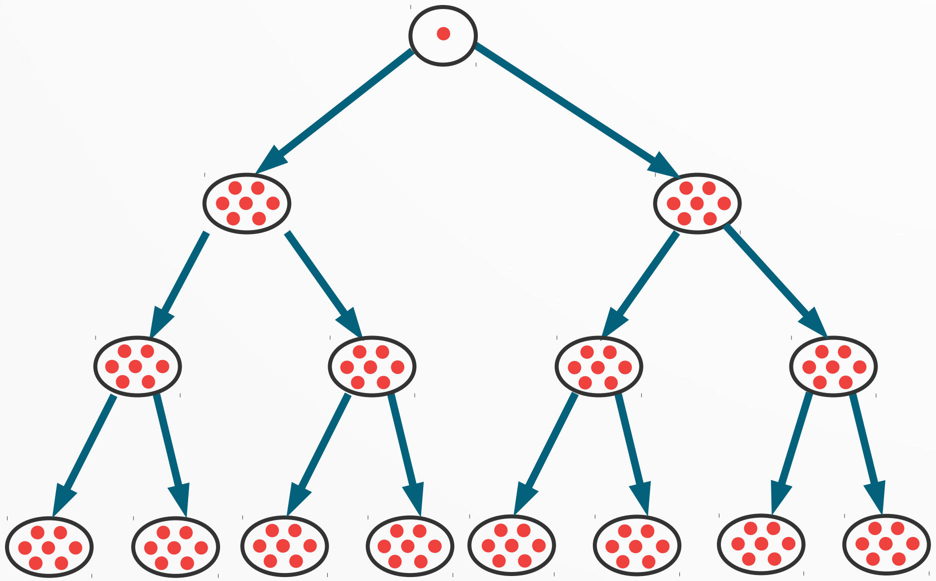
- Coupling between S and C
- Geometric Backtracking
- Large search space, different metrics

What makes it difficult?

- Coupling between S and C
- Geometric Backtracking
- Large search space, different metrics

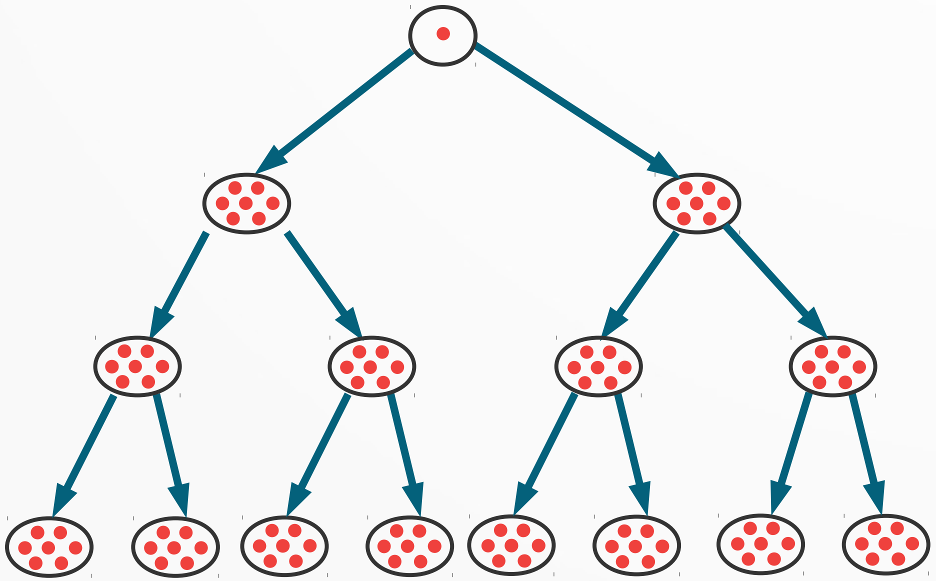
What makes it difficult?

- Coupling between S and C
- Geometric Backtracking
- Large search space, different metrics



What makes it difficult?

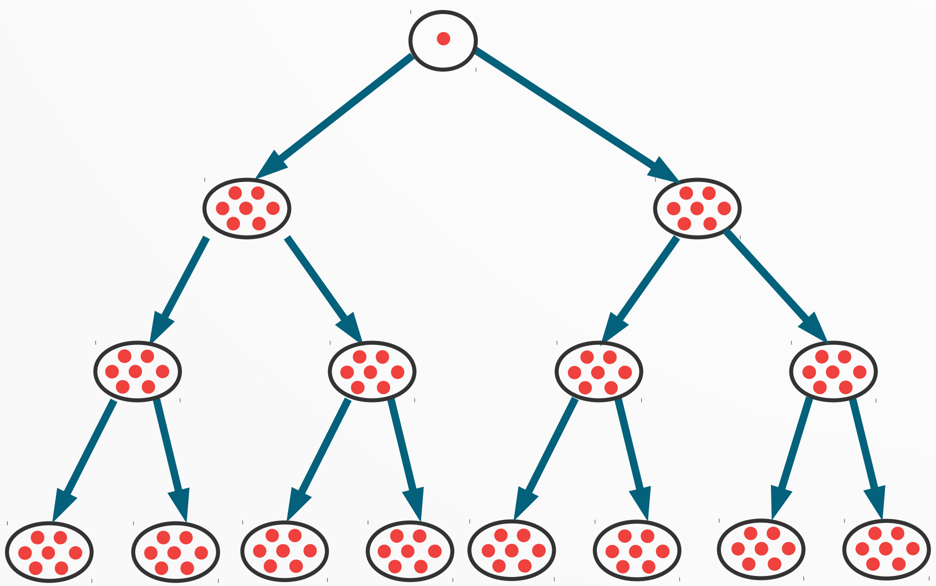
- Coupling between S and C
- Geometric Backtracking
- Large search space, different metrics



- 2 equal symbolic states may have very different geometric configurations
- 2 different symbolic states may have very similar geometric configurations

What makes it difficult?

- Coupling between S and C
- Geometric Backtracking
- Large search space, different metrics



- 2 equal symbolic states may have very different geometric configurations
- 2 different symbolic states may have very similar geometric configurations

→ no easy "distance to goal" heuristics

Outline

- Problem definition(s)
- What makes it difficult?
- 3 approaches to TAMP
- Future directions

Outline

- Problem definition(s)
- What makes it difficult?
- 3 approaches to TAMP
- Future directions

3 approaches to TAMP

- AsyMov
- FFRob
- ASP + failure explanation

3 approaches to TAMP

- AsyMov

3 approaches to TAMP

- AsyMov

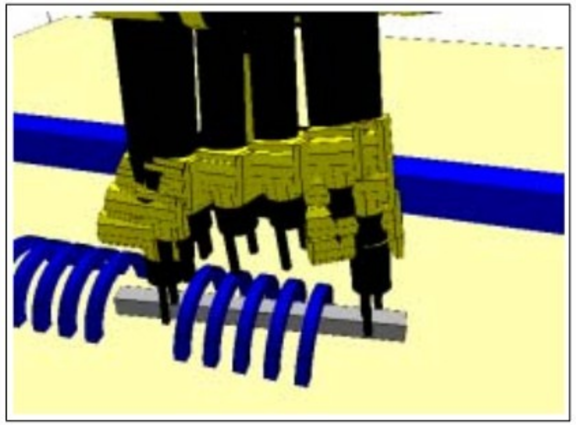
Grasp
Planning

- *A Geometrical Approach to Planning Manipulation Tasks. The Case of Discrete Placements and Grasps*
R. Alami, T. Siméon, and J.-P. Laumond. in International Symposium on Robotics Research, 1989.

3 approaches to TAMP

AsyMov

Grasp Planning → Manipulation Planning



A Geometrical Approach to Planning Manipulation Tasks. The Case of Discrete Placements and Grasps
R. Alami, T. Siméon, and J.-P. Laumond. in International Symposium on Robotics Research, 1989.

- Siméon T, Laumond J-P, Cortés J, Sahbani A. *Manipulation Planning with Probabilistic Roadmaps*. The International Journal of Robotics Research. 2004;23(7-8):729-746.

3 approaches to TAMP

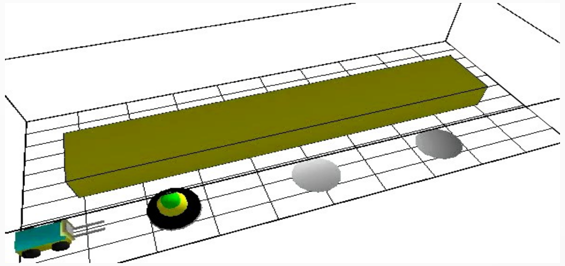
AsyMov

Grasp Planning → Manipulation Planning → TAMP

A Geometrical Approach to Planning Manipulation Tasks. The Case of Discrete Placements and Grasps
R. Alami, T. Siméon, and J.-P. Laumond. in International Symposium on Robotics Research, 1989.

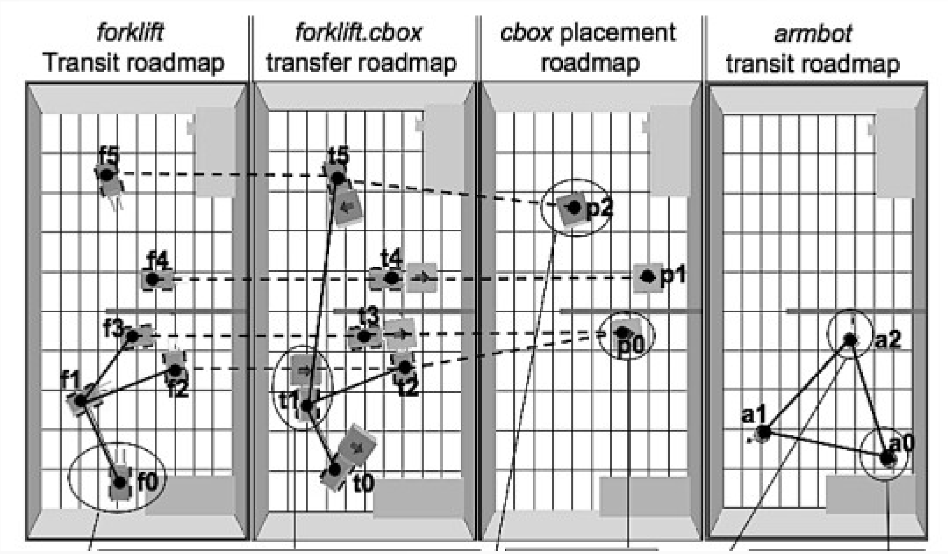
Siméon T, Laumond J-P, Cortés J, Sahbani A. *Manipulation Planning with Probabilistic Roadmaps*. The International Journal of Robotics Research. 2004;23(7-8):729-746.

- *A Hybrid Approach to Intricate Motion, Manipulation and Task Planning*. S. Cambon, R. Alami, and F. Gravot, International Journal of Robotics Research, 2009.



3 approaches to TAMP

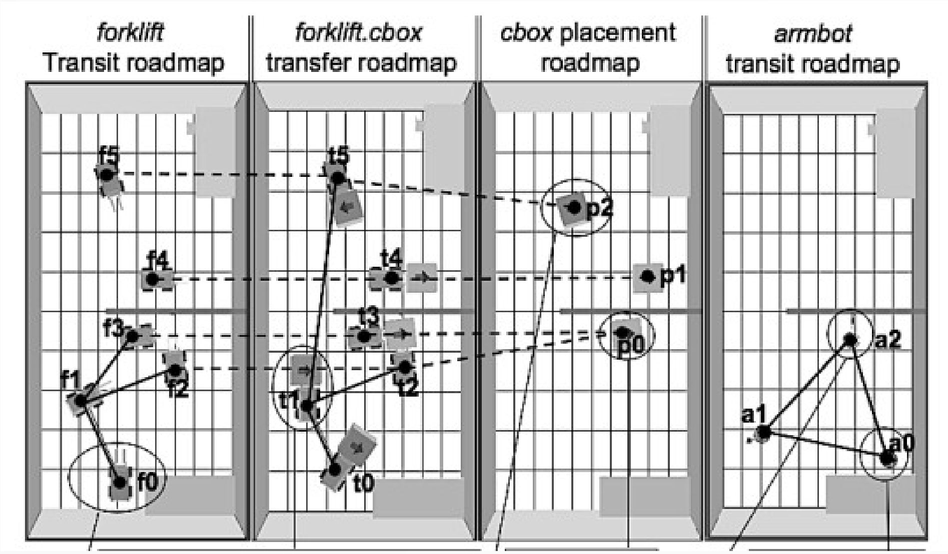
- AsyMov



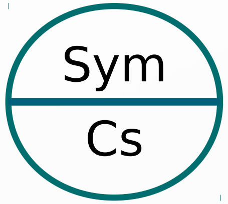
Roadmaps (manipulation planning)

3 approaches to TAMP

- AsyMov

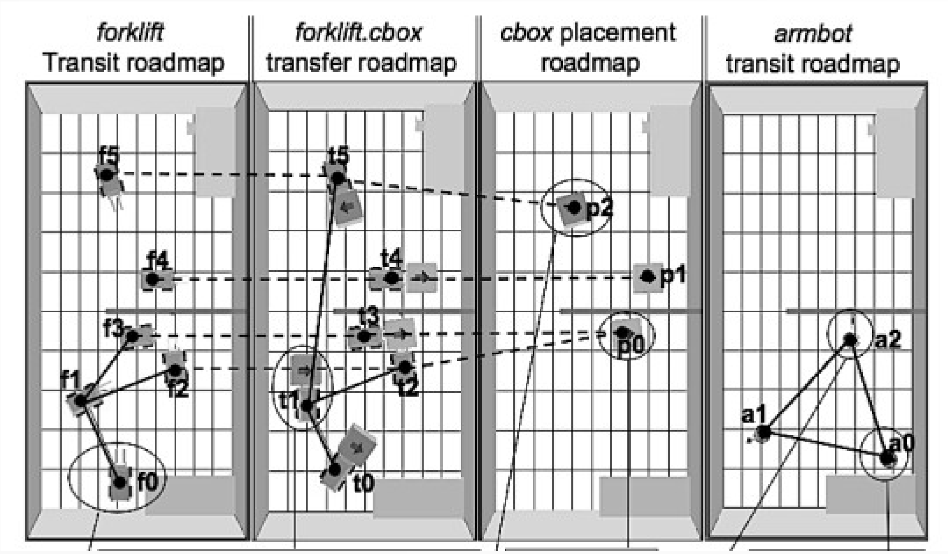


Roadmaps (manipulation planning)



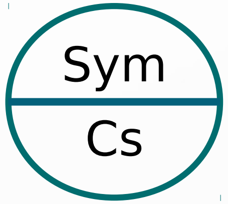
3 approaches to TAMP

AsyMov



Roadmaps (manipulation planning)

A*

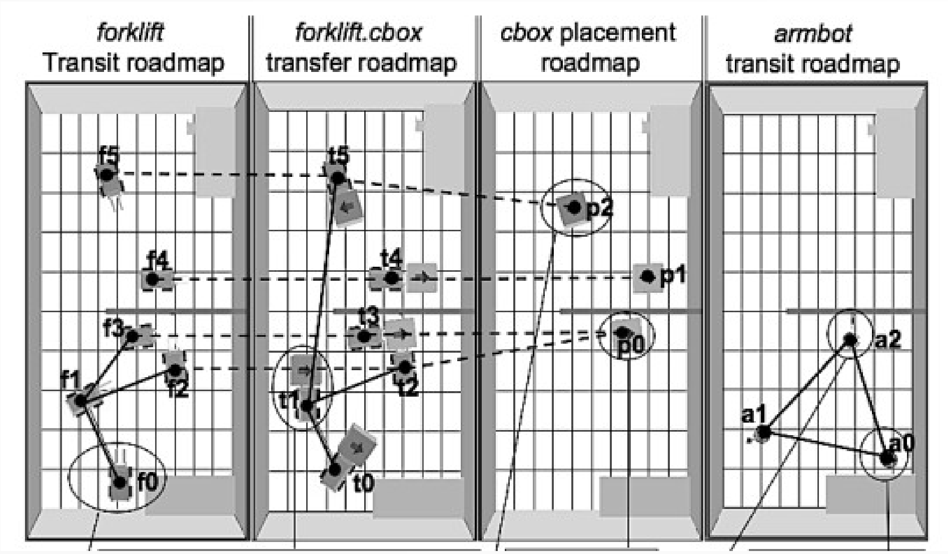


Cost:

Accumulated cost (h)
+ Heuristic cost
+ Cost of the number of failures

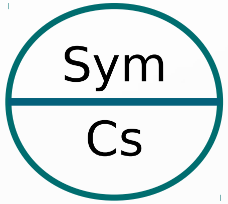
3 approaches to TAMP

- AsyMov



Roadmaps (manipulation planning)

A*



Cost =

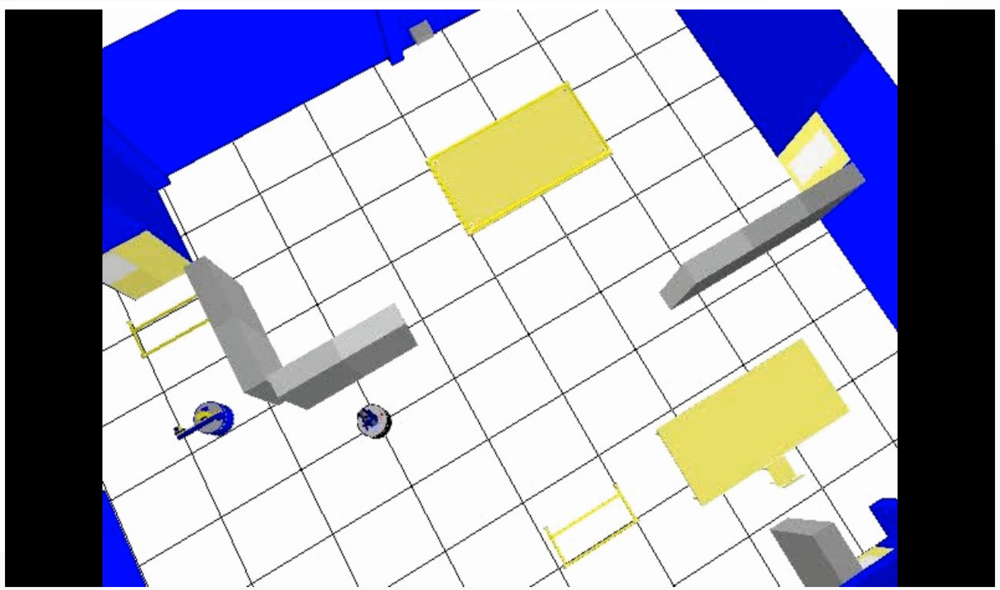
Accumulated cost (h)
+ Heuristic cost
+ Cost of the number of failures

Length of the symbolic plan from current state to goal state (computed with FF planner)

Trying to geometrically validate the state

3 approaches to TAMP

- AsyMov



A*

Sym

Cs

Cost =

Accumulated cost (h)

+ Heuristic cost

+ Cost of the number of failures

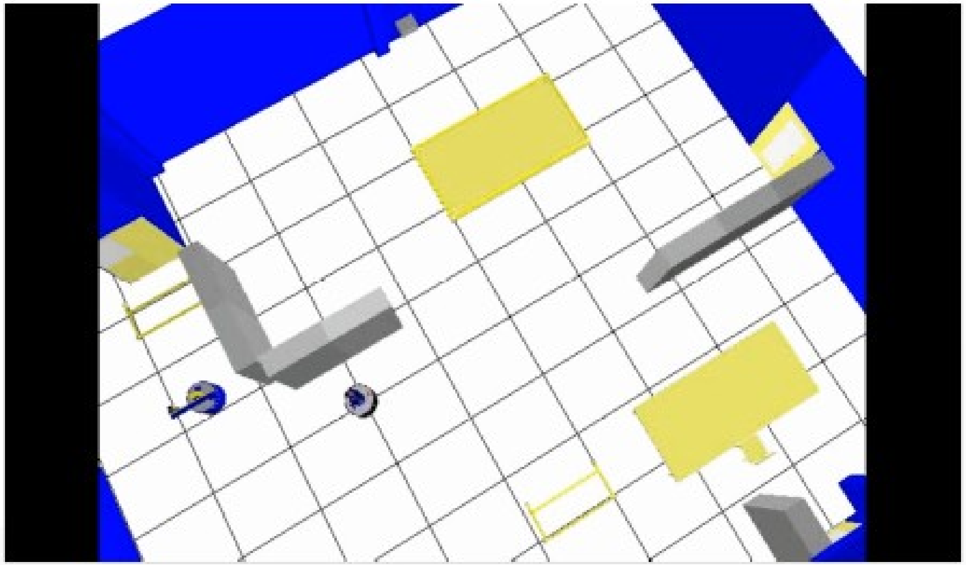
Length of the symbolic plan from current state to goal state (computed with FF planner)

Trying to geometrically validate the state

3 approaches to TAMP

Erion Plaku and Gregory D. Hager, *Sampling-based motion and symbolic action planning with geometric and differential constraints*, in ICRA, 2010, p. 5002--5008

AsyMov



A*

Sym
Cs

Cost =

Accumulated cost (h)
+ Heuristic cost
+ Cost of the number of failures

Length of the symbolic plan from current state to goal state (computed with FF planner)

Trying to geometrically validate the state

3 approaches to TAMP

- AsyMov
- FFRob
- ASP + failure explanation

3 approaches to TAMP

- AsyMov
- FFRob
- ASP + failure explanation

3 approaches to TAMP

- FFRob

3 approaches to TAMP

- FFRob

$\textsc{Pick}(C_1, O, G, P, C_2)$:

pre: $\mathit{HandEmpty}$, $\mathit{Pose}(O, P)$, $\mathit{Robotconf}(C_1)$, $\mathit{CanGrasp}(O, P, G, C_2)$, $\mathit{Reachable}(C_1, C_2)$
add: $\mathit{Holding}(O, G)$, $\mathit{RobotConf}(C_2)$
delete: $\mathit{HandEmpty}$, $\mathit{RobotConf}(C_1)$

$\textsc{Place}(C_1, O, G, P, C_2)$:

pre: $\mathit{Holding}(O, G)$, $\mathit{Robotconf}(C_1)$, $\mathit{CanGrasp}(O, P, G, C_2)$, $\mathit{Reachable}(C_1, C_2)$
add: $\mathit{HandEmpty}$, $\mathit{Pose}(O, P)$, $\mathit{RobotConf}(C_2)$
delete: $\mathit{Holding}(O, G)$, $\mathit{RobotConf}(C_1)$

3 approaches to TAMP

- FFRob

$\textsc{Pick}(C_1, O, G, P, C_2)$:

pre: *HandEmpty*, *Pose*(O, P), *Robotconf*(C_1), *CanGrasp*(O, P, G, C_2), $\boxed{\textit{Reachable}(C_1, C_2)}$
add: *Holding*(O, G), *RobotConf*(C_2)
delete: *HandEmpty*, *RobotConf*(C_1)

$\textsc{Place}(C_1, O, G, P, C_2)$:

pre: *Holding*(O, G), *Robotconf*(C_1), *CanGrasp*(O, P, G, C_2) $\boxed{\textit{Reachable}(C_1, C_2)}$
add: *HandEmpty*, *Pose*(O, P), *RobotConf*(C_2)
delete: *Holding*(O, G), *RobotConf*(C_1)

3 approaches to TAMP

- FFRob

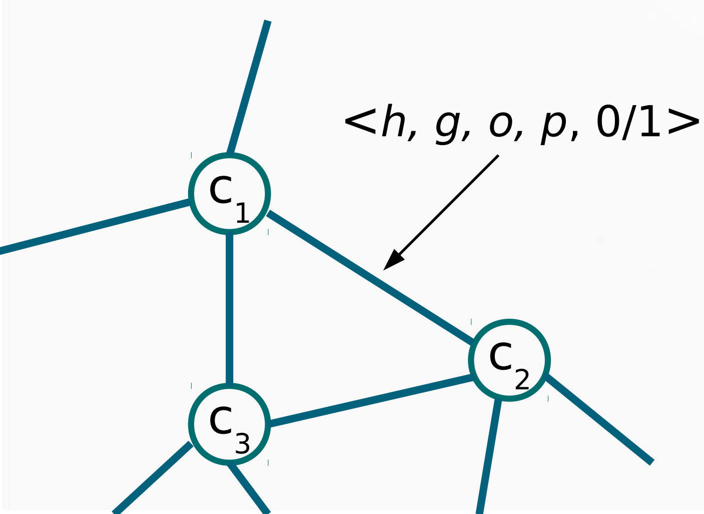
$\textsc{Pick}(C_1, O, G, P, C_2)$:

pre: $HandEmpty$, $Pose(O,P)$, $Robotconf(C_1)$, $CanGrasp(O,P,G,C_2)$, $Reachable(C_1,C_2)$
add: $Holding(O,G)$, $RobotConf(C_2)$
delete: $HandEmpty$, $RobotConf(C_1)$

$\textsc{Place}(C_1, O, G, P, C_2)$:

pre: $Holding(O,G)$, $Robotconf(C_1)$, $CanGrasp(O,P,G,C_2)$ $Reachable(C_1,C_2)$
add: $HandEmpty$, $Pose(O,P)$, $RobotConf(C_2)$
delete: $Holding(O,G)$, $RobotConf(C_1)$

CRG: Conditional Reachability Graph



3 approaches to TAMP

- FFRob

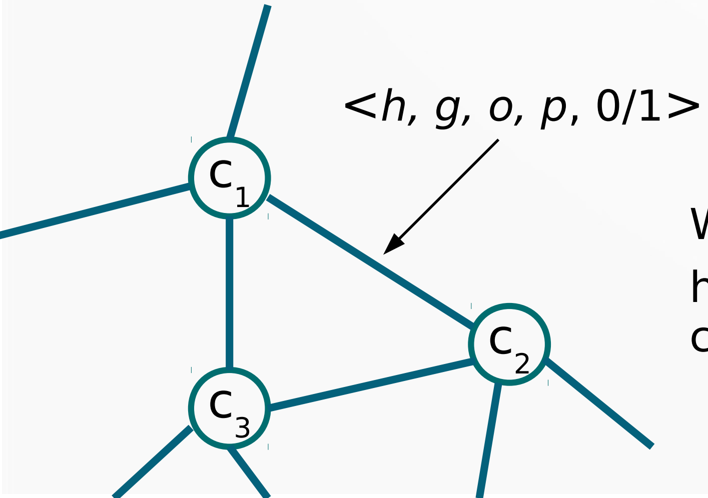
$\text{PICK}(C_1, O, G, P, C_2)$:

pre: $\mathit{HandEmpty},\ \mathit{Pose}(O,P),\ \mathit{Robotconf}(C_1),\ \mathit{CanGrasp}(O,P,G,C_2),\ \boxed{\mathit{Reachable}(C_1,C_2)}$
add: $\mathit{Holding}(O,G),\ \mathit{RobotConf}(C_2)$
delete: $\mathit{HandEmpty},\ \mathit{RobotConf}(C_1)$

$\text{PLACE}(C_1, O, G, P, C_2)$:

pre: $\mathit{Holding}(O,G),\ \mathit{Robotconf}(C_1),\ \mathit{CanGrasp}(O,P,G,C_2)\ \boxed{\mathit{Reachable}(C_1,C_2)}$
add: $\mathit{HandEmpty},\ \mathit{Pose}(O,P),\ \mathit{RobotConf}(C_2)$
delete: $\mathit{Holding}(O,G),\ \mathit{RobotConf}(C_1)$

CRG: Conditional Reachability Graph



When, moving from c_1 to c_2,
holding object *h* with grasp *g*,
collision with object *o* at pose *p*

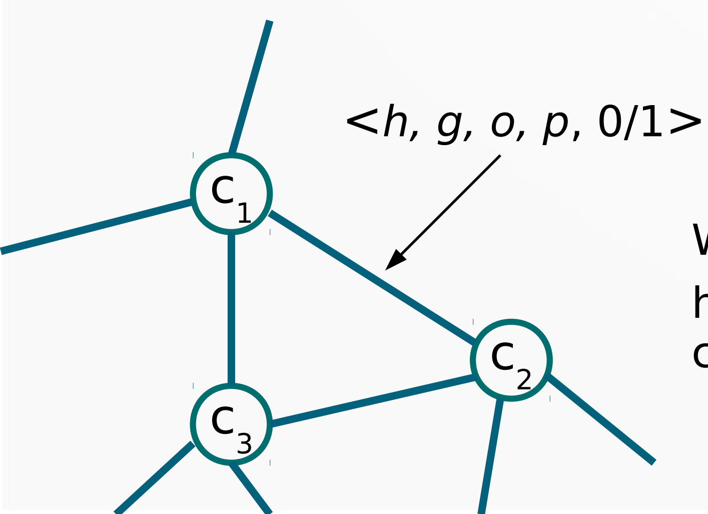
3 approaches to TAMP

FFRob

PICK(C_1, O, G, P, C_2):
pre: $HandEmpty$, $Pose(O, P)$, $Robotconf(C_1)$, $CanGrasp(O, P, G, C_2)$, $\boxed{Reachable(C_1, C_2)}$
add: $Holding(O, G)$, $RobotConf(C_2)$
delete: $HandEmpty$, $RobotConf(C_1)$

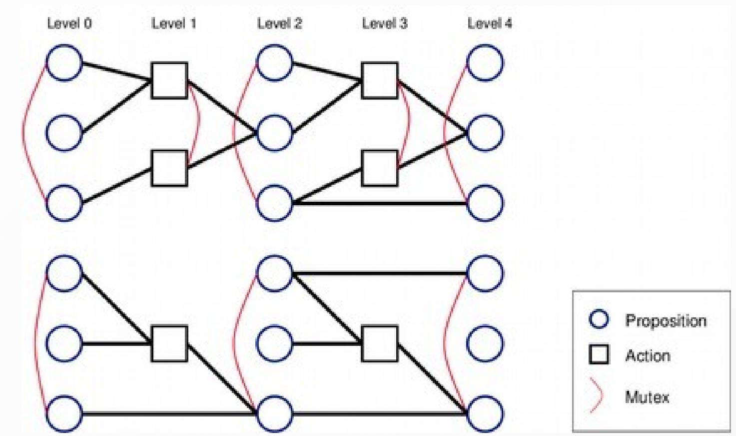
PLACE(C_1, O, G, P, C_2):
pre: $Holding(O, G)$, $Robotconf(C_1)$, $CanGrasp(O, P, G, C_2)$ $\boxed{Reachable(C_1, C_2)}$
add: $HandEmpty$, $Pose(O, P)$, $RobotConf(C_2)$
delete: $Holding(O, G)$, $RobotConf(C_1)$

CRG: Conditional Reachability Graph



When, moving from c_1 to c_2, holding object *h* with grasp *g*, collision with object *o* at pose *p*

Modified FF heuristic



H_{FF} = #actions to achieve the goal, ignoring deletions

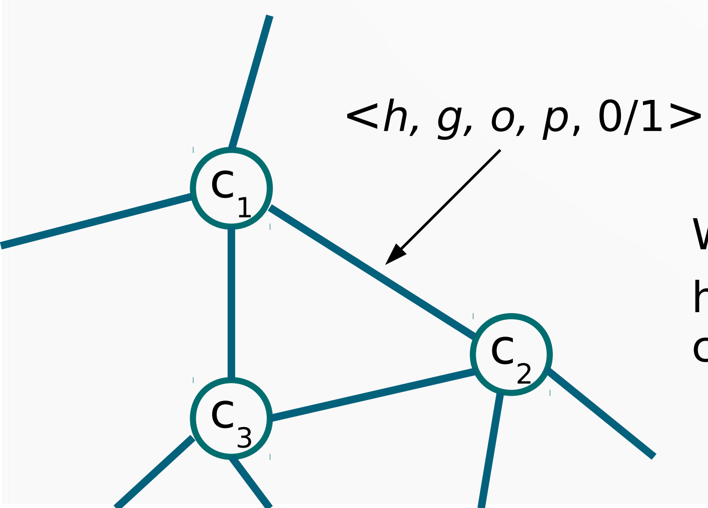
3 approaches to TAMP

FFRob

$\textsc{Pick}(C_1, O, G, P, C_2)$:
  pre: $\mathit{HandEmpty}$, $\mathit{Pose}(O,P)$, $\mathit{Robotconf}(C_1)$, $\mathit{CanGrasp}(O,P,G,C_2)$, $\boxed{\mathit{Reachable}(C_1, C_2)}$
  add: $\mathit{Holding}(O,G)$, $\mathit{RobotConf}(C_2)$
  delete: $\mathit{HandEmpty}$, $\mathit{RobotConf}(C_1)$

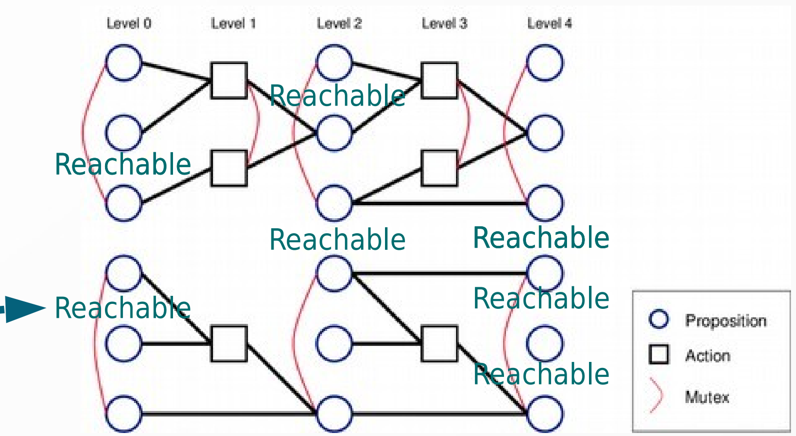
$\textsc{Place}(C_1, O, G, P, C_2)$:
  pre: $\mathit{Holding}(O,G)$, $\mathit{Robotconf}(C_1)$, $\mathit{CanGrasp}(O,P,G,C_2)$ $\boxed{\mathit{Reachable}(C_1, C_2)}$
  add: $\mathit{HandEmpty}$, $\mathit{Pose}(O,P)$, $\mathit{RobotConf}(C_2)$
  delete: $\mathit{Holding}(O,G)$, $\mathit{RobotConf}(C_1)$

CRG: Conditional Reachability Graph



When, moving from c_1 to c_2, holding object *h* with grasp *g*, collision with object *o* at pose *p*

Modified FF heuristic



H_{FF} = #actions to achieve the goal, ignoring deletions

→ Give a better heuristic value to states from which occluding objects can be moved towards non-occluding positions.

3 approaches to TAMP

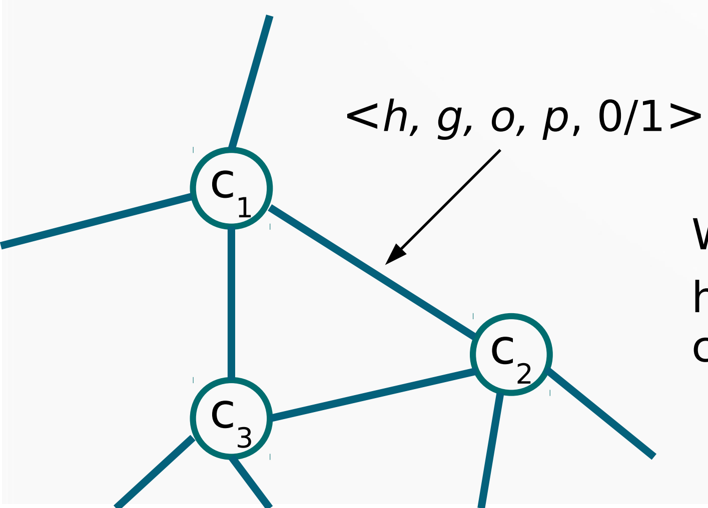
FFRob

PICK(C_1, O, G, P, C_2):
pre: $HandEmpty$, $Pose(O,P)$, $Robotconf(C_1)$, $CanGrasp(O,P,G,C_2)$, $\boxed{Reachable(C_1,C_2)}$
add: $Holding(O,G)$, $RobotC$
delete: $HandEmpty$, $RobotC$

PLACE(C_1, O, G, P, C_2):
pre: $Holding(O,G)$, $Robotco$
add: $HandEmpty$, $Pose(O,P)$, $RobotConf(C_2)$
delete: $Holding(O,G)$, $RobotConf(C_1)$

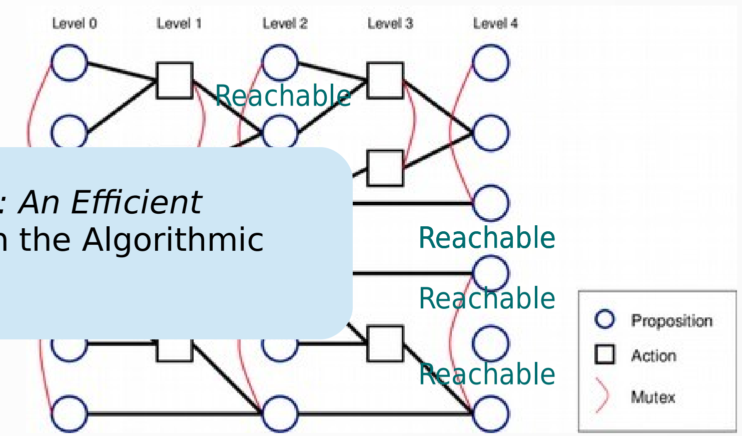
Caelan R. Garrett, Tomas Lozano-Perez, Leslie P. Kaelbling. *FFRob: An Efficient Heuristic for Task and Motion Planning*, International Workshop on the Algorithmic Foundations of Robotics (WAFR), 2014.

CRG: Conditional Reachability Graph



When, moving from c_1 to c_2, holding object *h* with grasp *g*, collision with object *o* at pose *p*

Modified FF heuristic



H_{FF} = #actions to achieve the goal, ignoring deletions

→ Give a better heuristic value to states from which occluding objects can be moved towards non-occluding positions.

3 approaches to TAMP

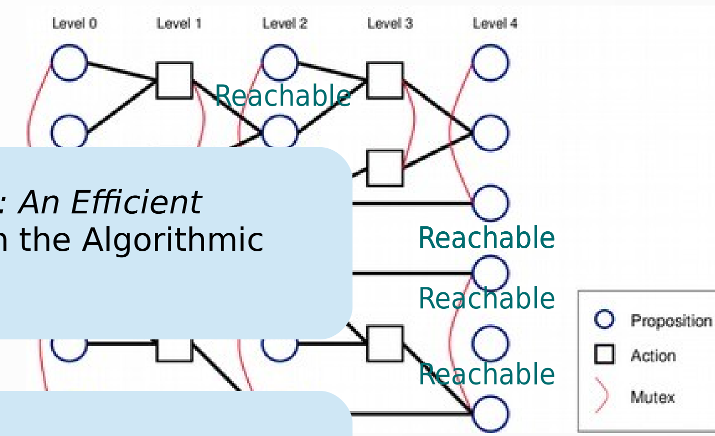
FFRob

PICK(C_1, O, G, P, C_2):
pre: *HandEmpty*, *Pose*(O,P), *Robotconf*(C_1), *CanGrasp*(O,P,G,C_2), *Reachable*(C_1,C_2)
add: *Holding*(O,G), *RobotC*
delete: *HandEmpty*, *RobotC*

PLACE(C_1, O, G, P, C_2):
pre: *Holding*(O,G), *Robotco*
add: *HandEmpty*, *Pose*(O,P), *RobotConf*(C_2)
delete: *Holding*(O,G), *RobotConf*(C_1)

Caelan R. Garrett, Tomas Lozano-Perez, Leslie P. Kaelbling. *FFRob: An Efficient Heuristic for Task and Motion Planning*, International Workshop on the Algorithmic Foundations of Robotics (WAFR), 2014.

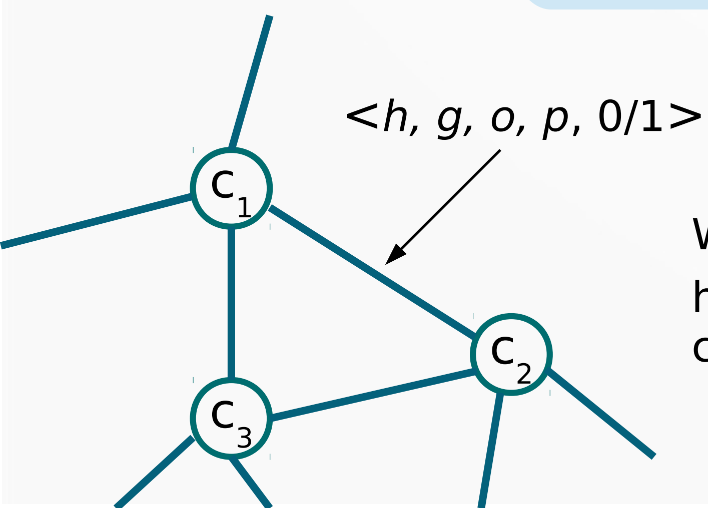
Modified FF heuristic



...o achieve the goal, ignoring deletions

CRG: Conditi...

Akbari, A., Lagriffoul, F. & Rosell, J. *Combined heuristic task and motion planning for bi-manual robots*. Autonomous Robot 43, 1575–1590 (2019).



When, moving from c_1 to c_2, holding object *h* with grasp *g*, collision with object *o* at pose *p*

→ Give a better heuristic value to states from which occluding objects can be moved towards non-occluding positions.

3 approaches to TAMP

- AsyMov
- FFRob
- ASP + failure explanation

3 approaches to TAMP

- AsyMov
- FFRob
- ASP + failure explanation

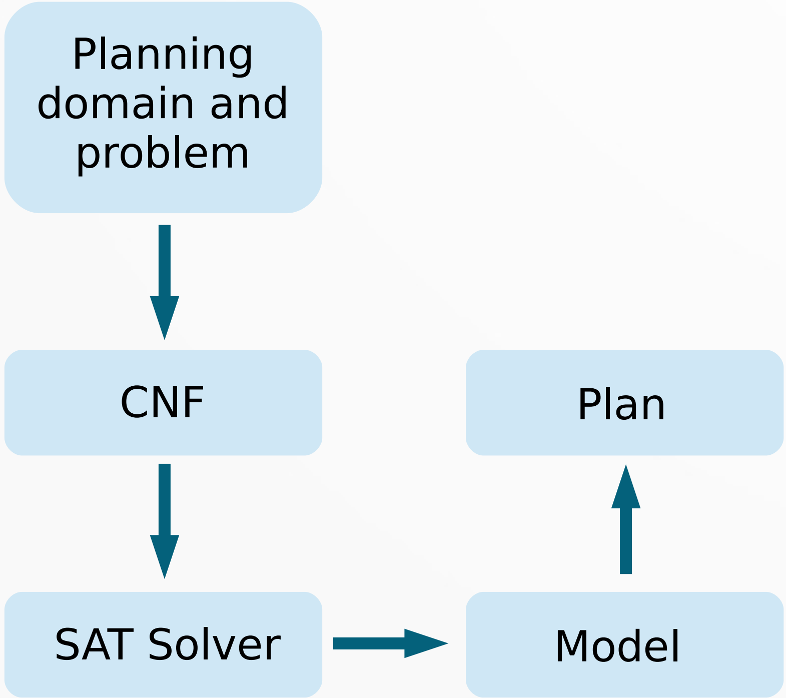
3 approaches to TAMP

- ASP + failure explanation

3 approaches to TAMP

- ASP + failure explanation

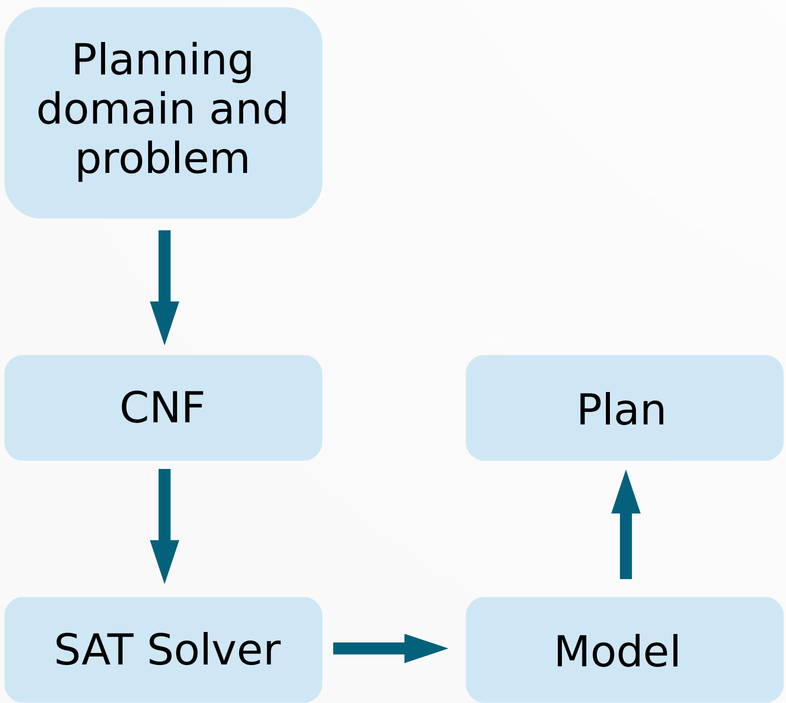
SAT Planning:

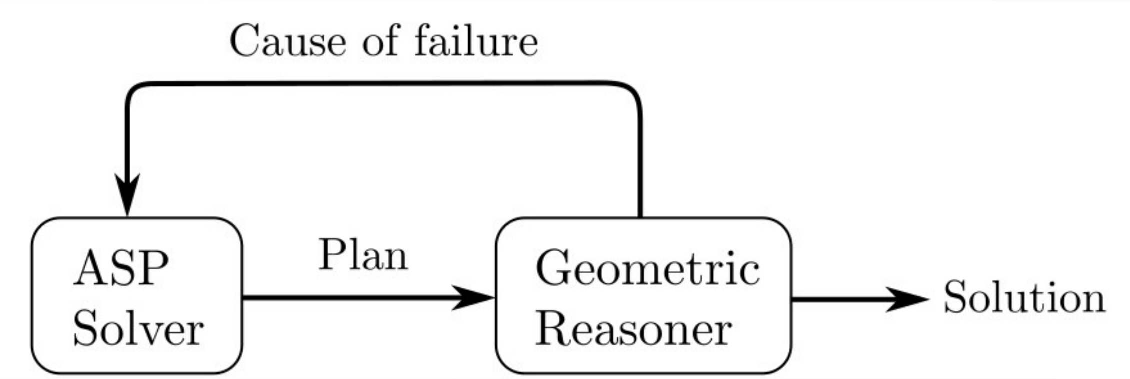


3 approaches to TAMP

- ASP + failure explanation

SAT Planning:

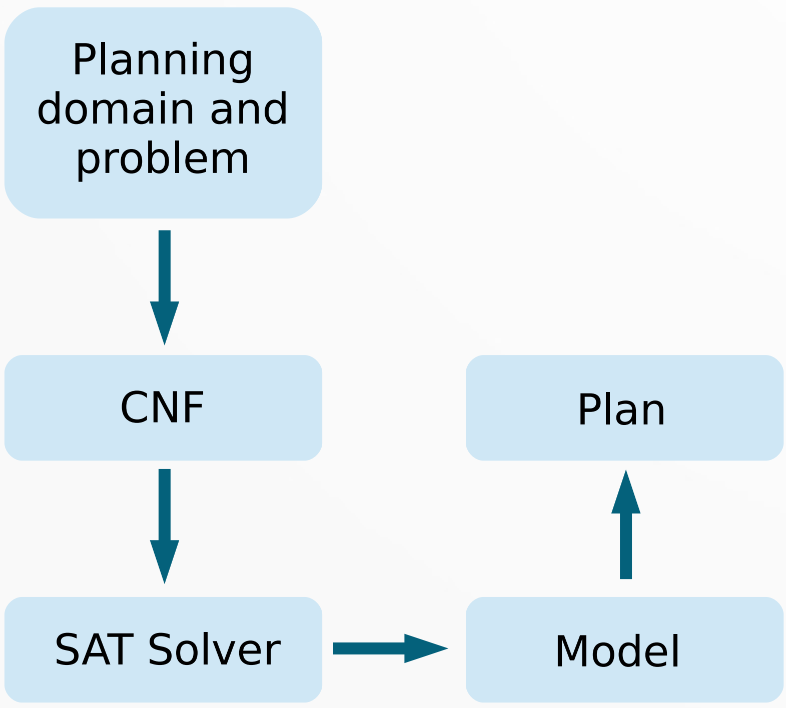


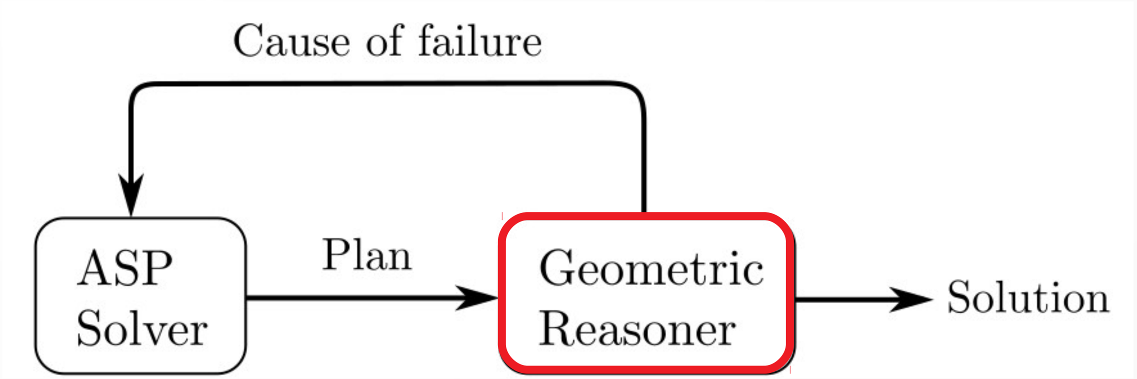


3 approaches to TAMP

- ASP + failure explanation

SAT Planning:

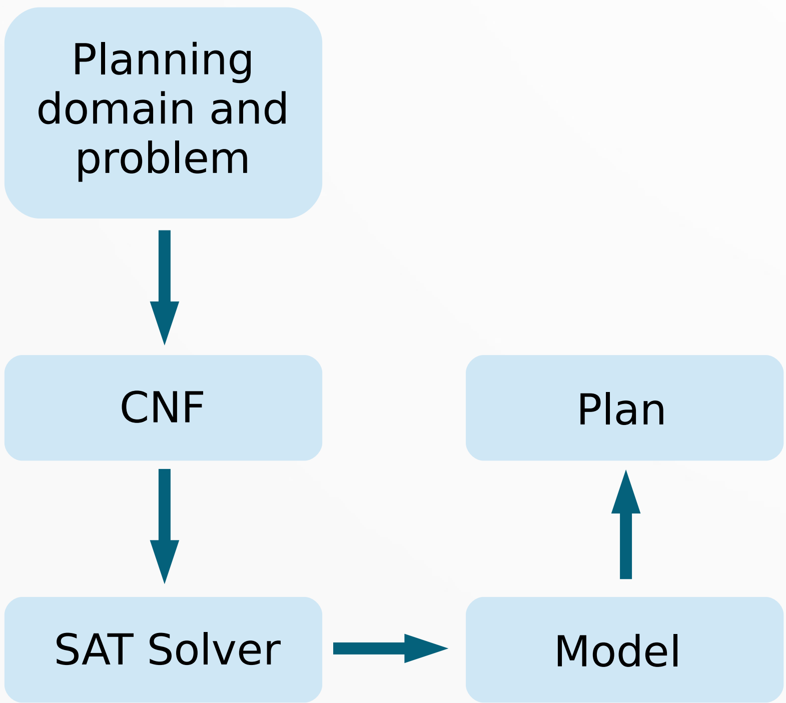


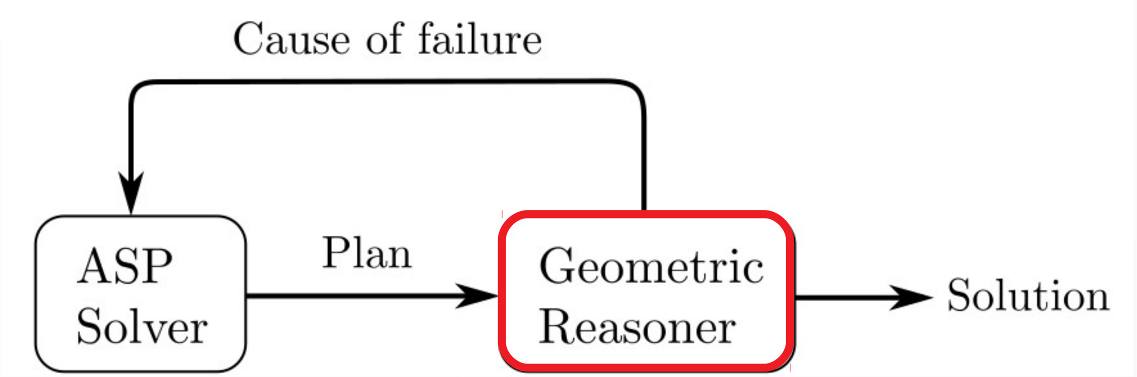


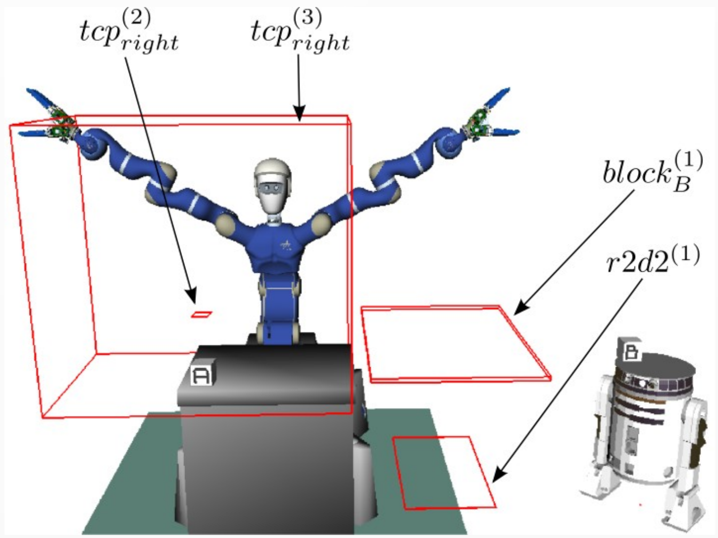
3 approaches to TAMP

- ASP + failure explanation

SAT Planning:



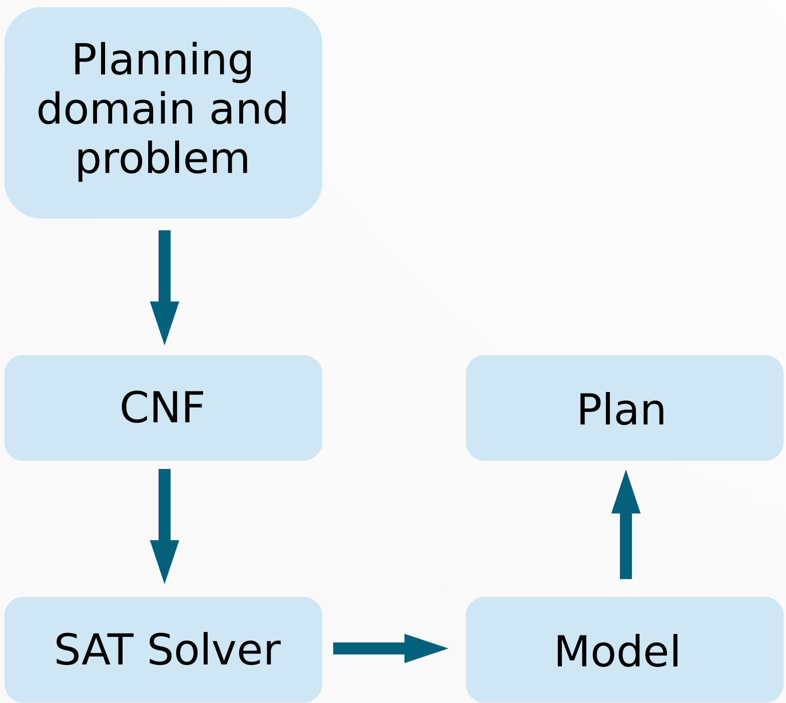


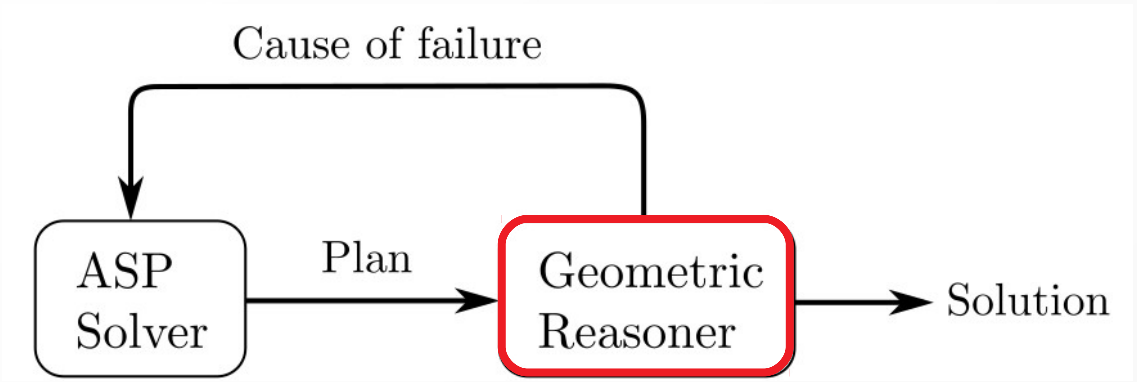


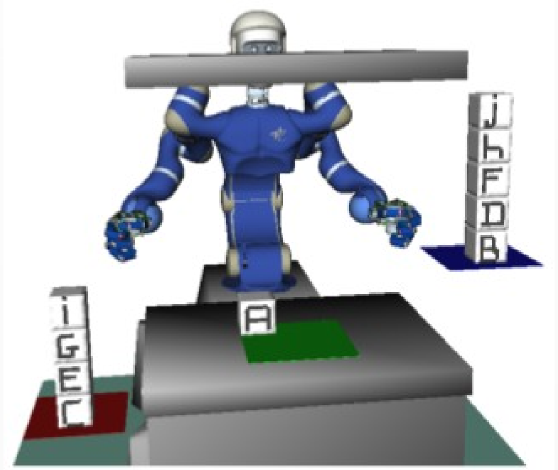
3 approaches to TAMP

- ASP + failure explanation

SAT Planning:





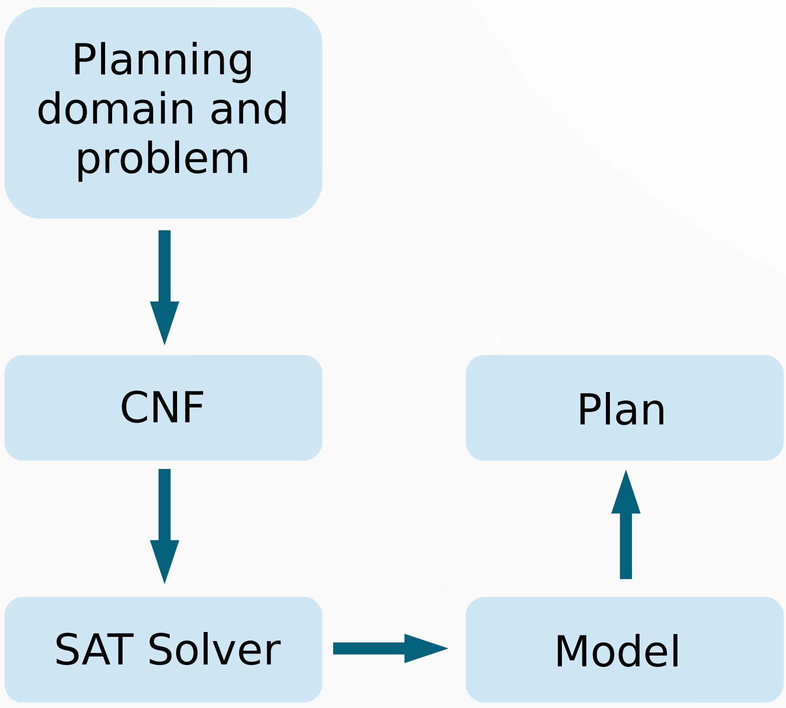


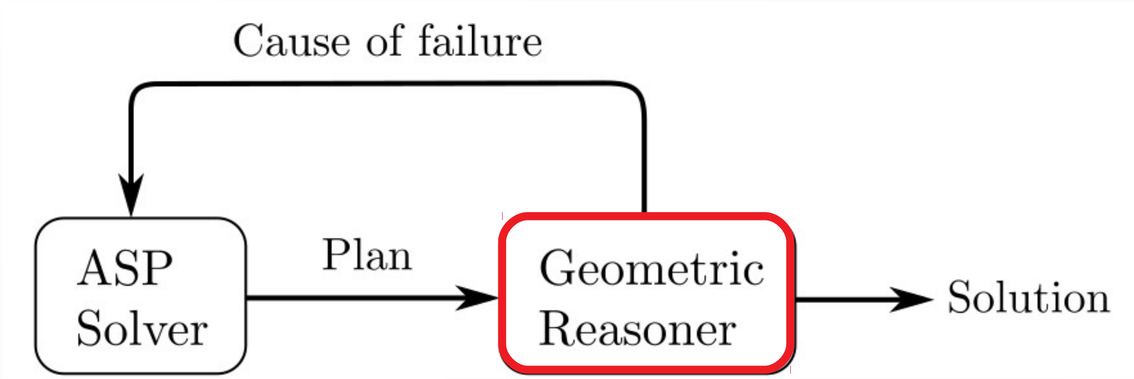
Blocks world 3D

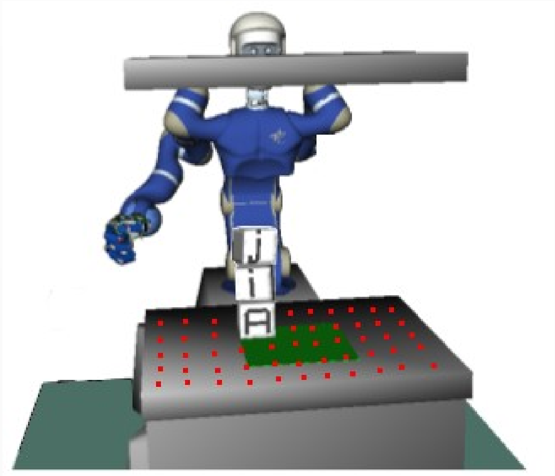
3 approaches to TAMP

- ASP + failure explanation

SAT Planning:





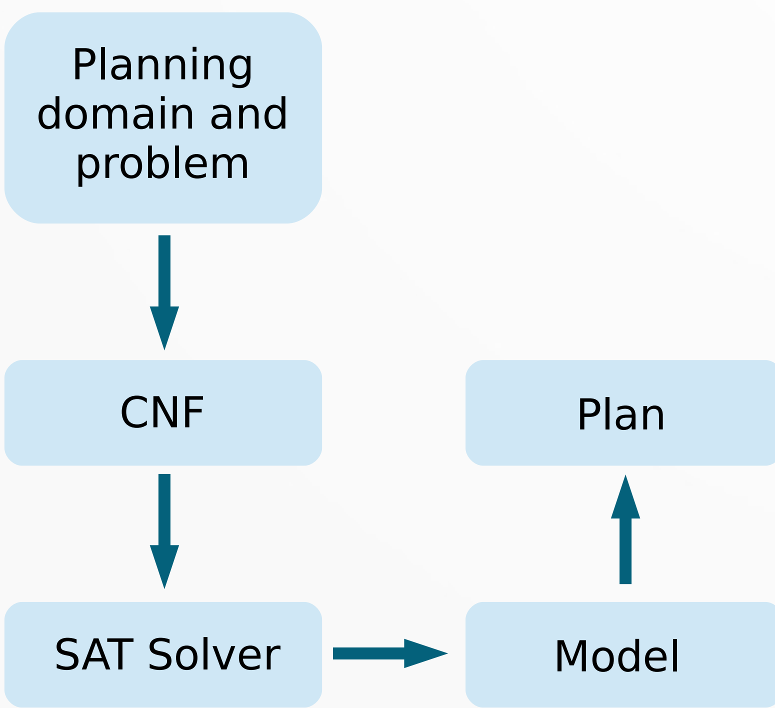


Blocks world 3D

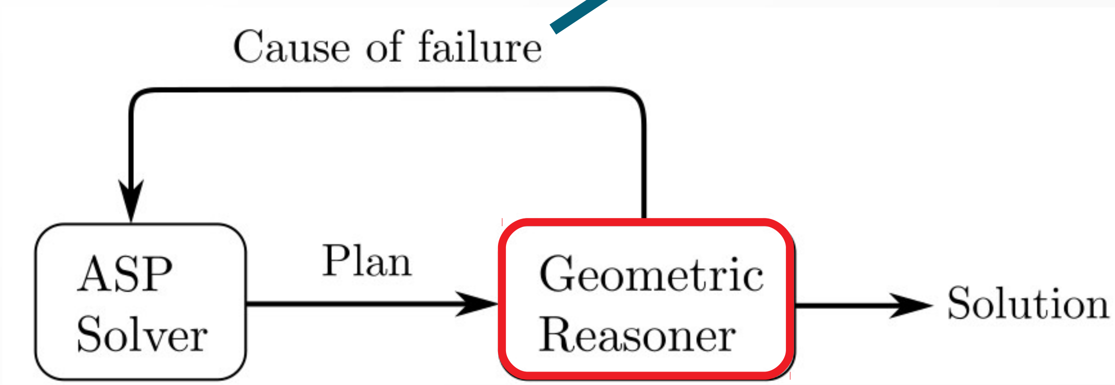
3 approaches to TAMP

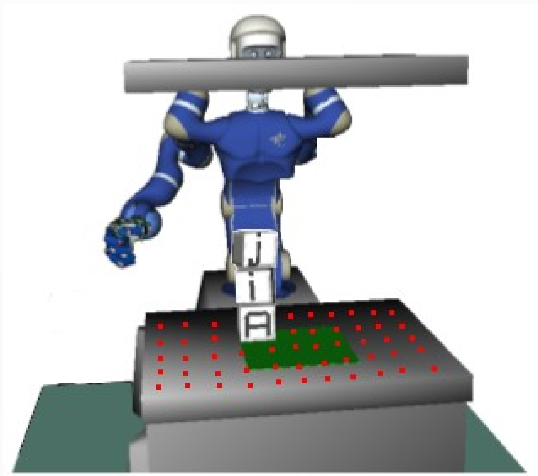
ASP + failure explanation

SAT Planning:



NOT (on (J, I)
AND on (I, A)
AND on (A, table))



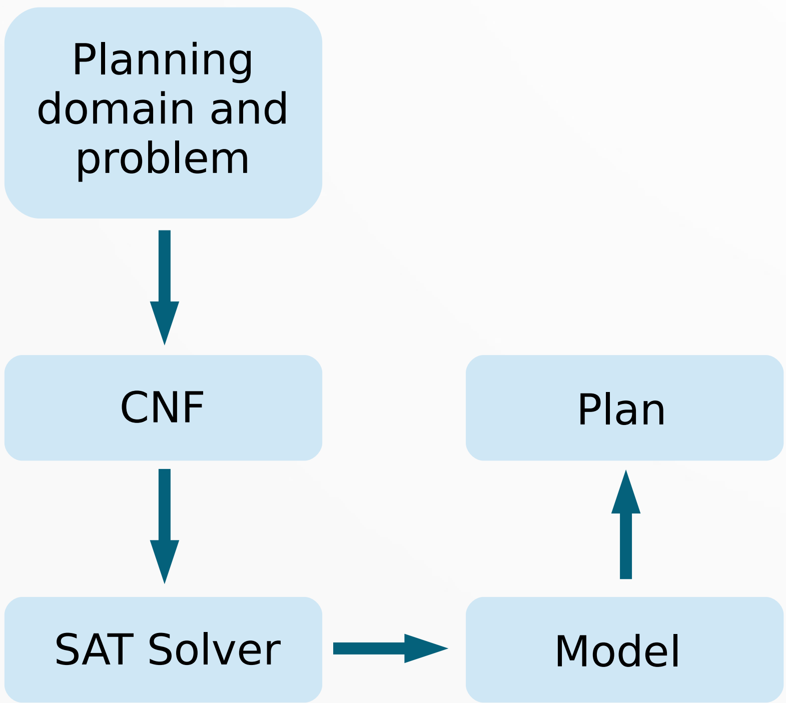


Blocks world 3D

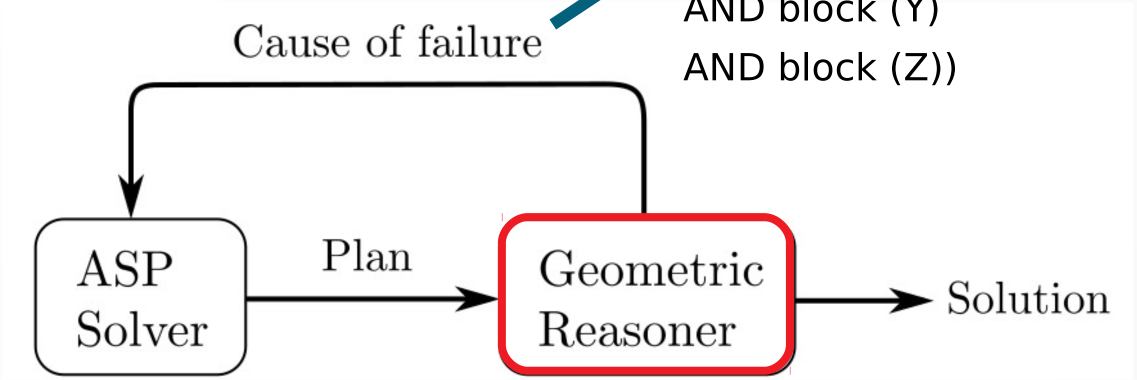
3 approaches to TAMP

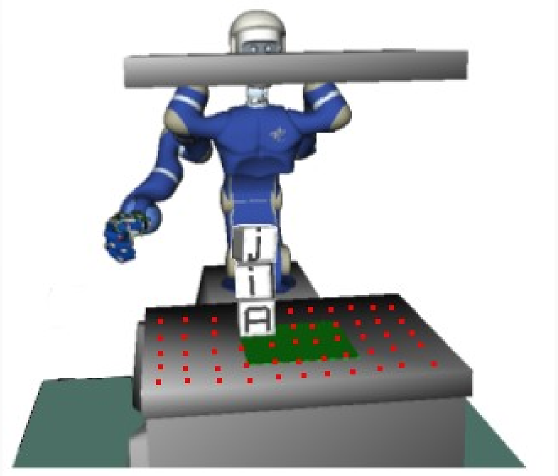
ASP + failure explanation

SAT Planning:



NOT (on (X, Y)
AND on (Y, Z)
AND on (Z, table)
AND block (X)
AND block (Y)
AND block (Z))



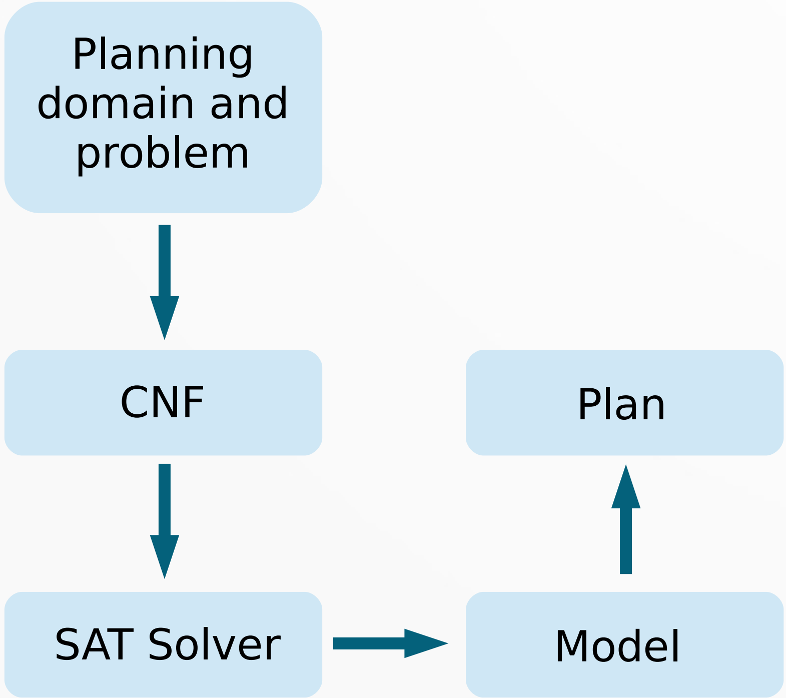


Blocks world 3D

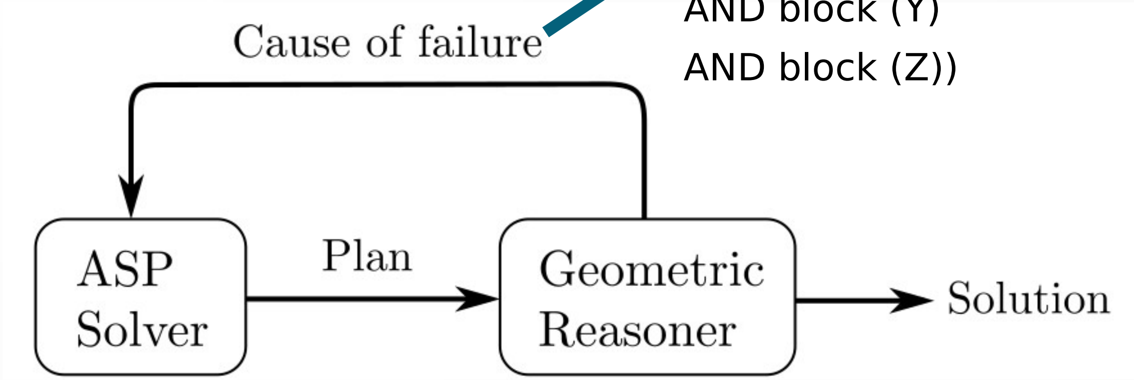
3 approaches to TAMP

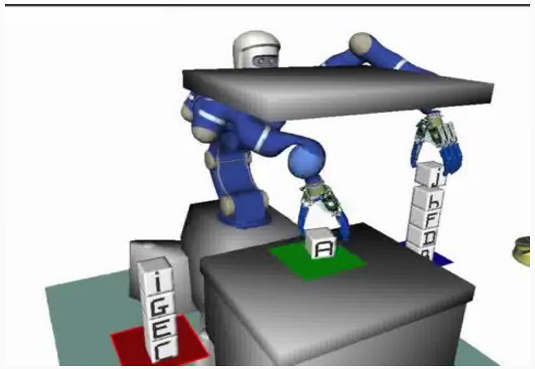
ASP + failure explanation

SAT Planning:



NOT (on (X, Y)
AND on (Y, Z)
AND on (Z, table)
AND block (X)
AND block (Y)
AND block (Z))





Blocks world 3D

3 approaches to TAMP

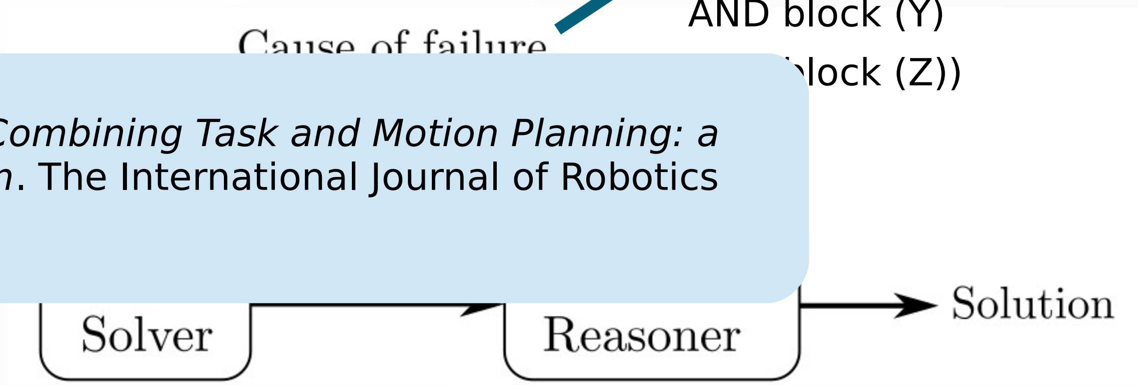
ASP + failure explanation

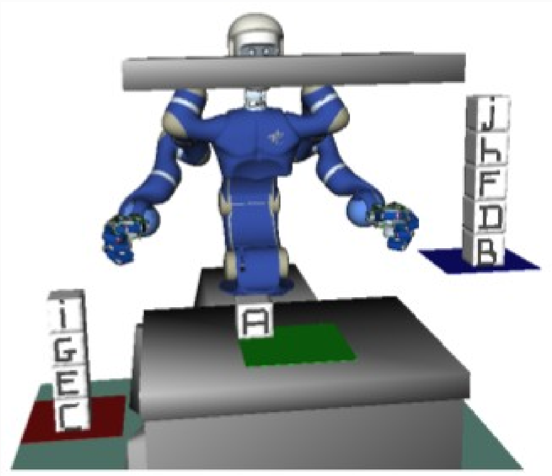
NOT (on (X, Y)
AND on (Y, Z)
AND on (Z, table)
AND block (X)
AND block (Y)
lock (Z))

SAT Planning:

Planning domain and problem → CNF → SAT Solver → Model → Plan

F. Lagriffoul, B. Andres, *Combining Task and Motion Planning: a culprit detection problem*. The International Journal of Robotics Research, 2016.





Blocks world 3D

3 approaches to TAMP

- AsyMov
- FFRob
- ASP + failure explanation

3 approaches to TAMP

- AsyMov — Roadmaps + task planner as heuristic
- FFRob — Task planner + CRG as heuristic
- ASP + failure explanation — Geometric reasoner pruning the search space of the task planner

Outline

Outline

- Problem definition(s)
- What makes it difficult?
- 3 approaches to TAMP
- Future directions

Outline

- Problem definition(s)
- What makes it difficult?
- 3 approaches to TAMP
- Future directions

Future directions

- Learning

A. M. Wells, N. T. Dantam, A. Shrivastava, and L. E. Kavraki, *Learning Feasibility for Task and Motion Planning in Tabletop Environments,* IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1255–1262, Apr. 2019.

Kim B, Wang Z, Kaelbling LP, Lozano-Pérez T. *Learning to guide task and motion planning using score-space representation*. The International Journal of Robotics Research. 2019;38(7):793-812.

Future directions

- Learning

  A. M. Wells, N. T. Dantam, A. Shrivastava, and L. E. Kavraki, *Learning Feasibility for Task and Motion Planning in Tabletop Environments,* IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1255–1262, Apr. 2019.

  Kim B, Wang Z, Kaelbling LP, Lozano-Pérez T. *Learning to guide task and motion planning using score-space representation*. The International Journal of Robotics Research. 2019;38(7):793-812.

- Theory

  William Vega-Brown, Nicholas Roy, *Task and Motion Planning Is PSPACE-Complete*. AAAI 2020: 10385-10392.

Future directions

- Learning

  A. M. Wells, N. T. Dantam, A. Shrivastava, and L. E. Kavraki, *Learning Feasibility for Task and Motion Planning in Tabletop Environments,* IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1255–1262, Apr. 2019.

  Kim B, Wang Z, Kaelbling LP, Lozano-Pérez T. *Learning to guide task and motion planning using score-space representation*. The International Journal of Robotics Research. 2019;38(7):793-812.

- Theory

  William Vega-Brown, Nicholas Roy, *Task and Motion Planning Is PSPACE-Complete*. AAAI 2020: 10385-10392.

- Culprit detection

  Hauser K. *The minimum constraint removal problem with three robotics applications*. The International Journal of Robotics Research. 2014;33(1):5-17.

Future directions

- Learning

  A. M. Wells, N. T. Dantam, A. Shrivastava, and L. E. Kavraki, *Learning Feasibility for Task and Motion Planning in Tabletop Environments,* IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1255–1262, Apr. 2019.

  Kim B, Wang Z, Kaelbling LP, Lozano-Pérez T. *Learning to guide task and motion planning using score-space representation*. The International Journal of Robotics Research. 2019;38(7):793-812.

- Theory

  William Vega-Brown, Nicholas Roy, *Task and Motion Planning Is PSPACE-Complete*. AAAI 2020: 10385-10392.

- Culprit detection

  Hauser K. *The minimum constraint removal problem with three robotics applications*. The International Journal of Robotics Research. 2014;33(1):5-17.

- You guys!

Bibliography

- L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE, Transactions on Robotics & Automation, 12(4):566–580, June 1996.
- S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. In Proceedings IEEE International Conference on Robotics and Automation, pages 473–479, 1999
- Siméon T, Laumond J-P, Cortés J, Sahbani A. Manipulation Planning with Probabilistic Roadmaps. The International Journal of Robotics Research. 2004;23(7-8):729-746.
- Mike Stilman. 2007. Navigation among movable obstacles. Ph.D. Dissertation. Carnegie Mellon University, USA. Advisor(s) James J. Kuffner.
- G. Havur, et al., Geometric rearrangement of multiple movable objects on cluttered surfaces: A hybrid reasoning approach, 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014, pp
- Hauser K, Latombe J-C. Multi-modal Motion Planning in Non-expansive Spaces. The International Journal of Robotics Research. 2010;29(7):897-915.
- R. Alami, T. Siméon, and J.-P. Laumond. A Geometrical Approach to Planning Manipulation Tasks. The Case of Discrete Placements and Grasps. in International Symposium on Robotics Research, 1989.

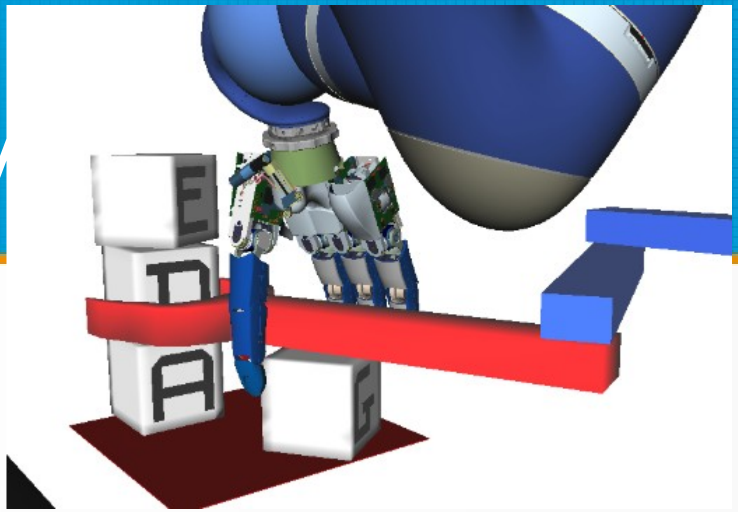
- Siméon T, Laumond J-P, Cortés J, Sahbani A. Manipulation Planning with Probabilistic Roadmaps. The International Journal of Robotics Research. 2004;23(7-8):729-746.

- S. Cambon, R. Alami, and F. Gravot, A Hybrid Approach to Intricate Motion, Manipulation and Task Planning. International Journal of Robotics Research, 2009.
- Caelan R. Garrett, Tomas Lozano-Perez, Leslie P. Kaelbling. FFRob: An Efficient Heuristic for Task and Motion Planning, International Workshop on the Algorithmic Foundations of Robotics (WAFR), 2014.
- Akbari, A., Lagriffoul, F. & Rosell, J. Combined heuristic task and motion planning for bi-manual robots. Autonomous Robot 43, 1575–1590 (2019).
- F. Lagriffoul, B. Andres, Combining Task and Motion Planning: a culprit detection problem. The International Journal of Robotics Research, 2016.
- Hauser K. The minimum constraint removal problem with three robotics applications. The International Journal of Robotics Research. 2014;33(1):5-17.
- A. M. Wells, N. T. Dantam, A. Shrivastava, and L. E. Kavraki, Learning Feasibility for Task and Motion Planning in Tabletop Environments, IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1255–1262, Apr. 2019.
- Kim B, Wang Z, Kaelbling LP, Lozano-Pérez T. Learning to guide task and motion planning using score-space representation. The International Journal of Robotics Research. 2019;38(7):793-812.
- William Vega-Brown, Nicholas Roy, Task and Motion Planning Is PSPACE-Complete. AAAI 2020: 10385-10392.

- Hauser K. The minimum constraint removal problem with three robotics applications. The International Journal of Robotics Research. 2014;33(1):5-17.

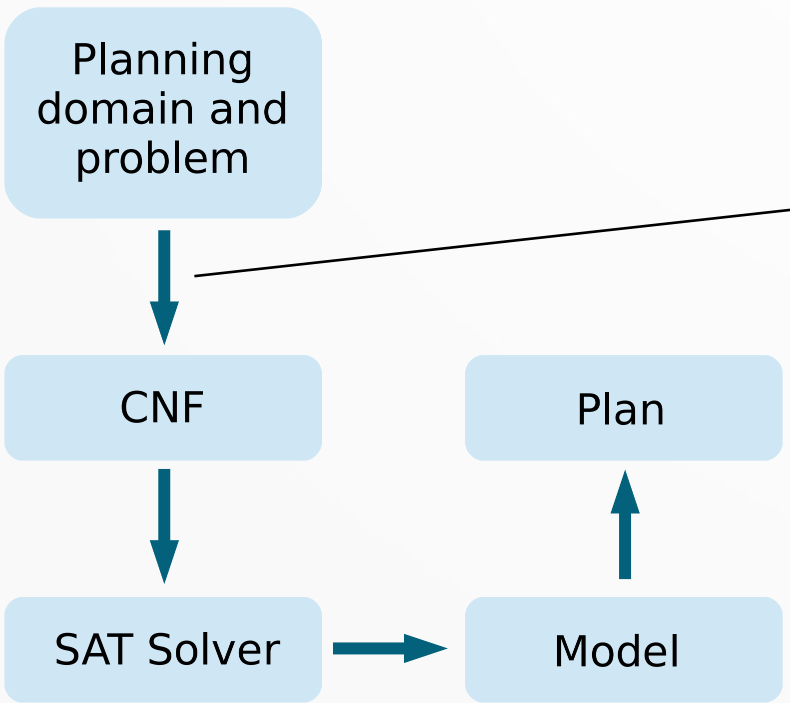
- WIKI: http://tampbenchmark.aass.oru.se/

3 approaches to TAM



- ASP + failure explanation

SAT Planning:



Formula describing the initial state:

$$\bigwedge\{l_0 \mid l \in s_0\} \wedge \bigwedge\{\neg l_0 \mid l \in L - s_0\}$$

Formula describing the goal state:

$$\bigwedge\{l_n \mid l \in g^+\} \wedge \bigwedge\{\neg l_n \mid l \in g^-\}$$

Formulas describing the preconditions and effects of actions:

For every action a in A, formulas describing what changes a would make if it were the i'th step of the plan:

- $a_i \Rightarrow \bigwedge\{p_i \mid p \in \text{Precond}(a)\} \wedge \bigwedge\{e_{i+1} \mid e \in \text{Effects}(a)\}$

Formulas describing *Complete exclusion*:

- For all actions a and b, formulas saying they cannot occur at the same time
  $\neg a_i \vee \neg b_i$
- this guarantees there can be only one action at a time