国防科技大学
NATIONAL UNIVERSITY
OF DEFENSE TECHNOLOGY

AAAI-25 / IAAI-25 / EAAI-25
FEBRUARY 25 – MARCH 4, 2025 | PHILADELPHIA, USA

MRBTP: Efficient Multi-Robot Behavior Tree Planning and Collaboration

Session: Robotics (2/4)

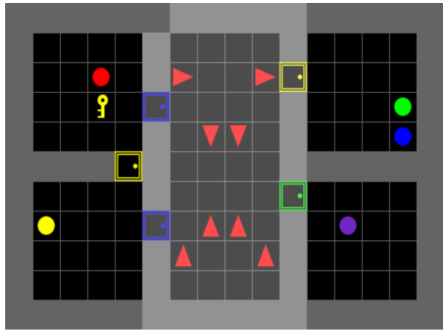
Yishuai Cai*, Xinglin Chen*, Zhongxuan Cai†, Yunxin Mao
Minglong Li†, Wenjing Yang, Ji Wang

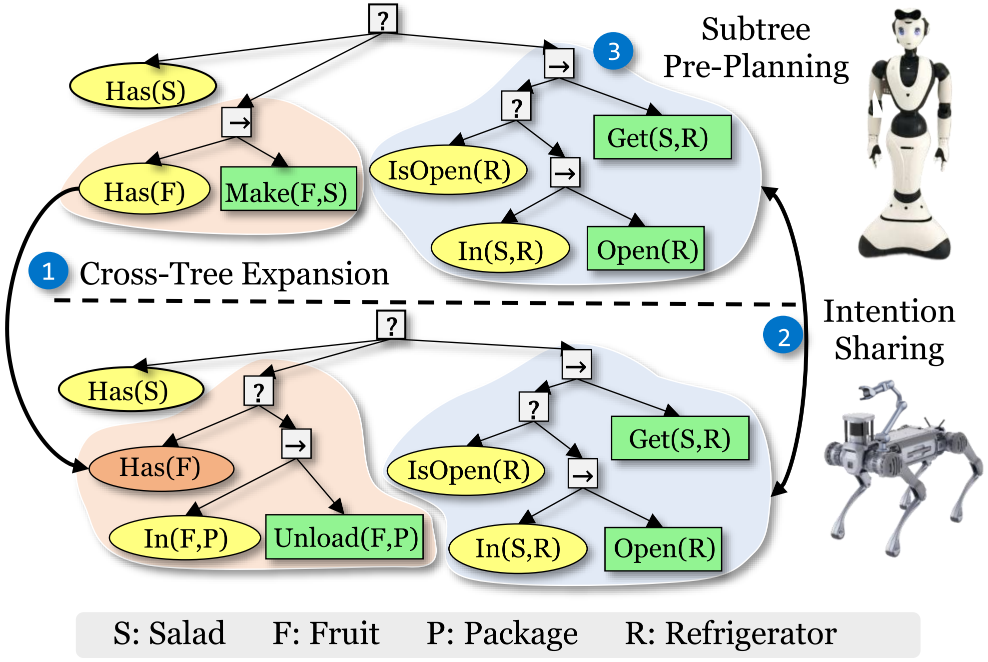
National University of Defense Technology

2025-03-02

Introduction

- Multi-robot systems (MRS)
  - robots with diverse capabilities, improved performance and fault tolerance
  - requires an efficient and robust control architecture
- Behavior Trees (BTs)
  - a popular control architecture for robot behaviors
  - modularity, interpretability, reactivity, and robustness
- BT in MRS
  - BT Planning for a Single Robot
    - **Sound and Complete** (Proven by BT Expansion[1])
  - **Our Contribution in BT Planning for Multi-Robots**
    - **MRBTP**: theoretical guarantees of both **soundness and completeness**
    - **Cross-Tree Expansion** and **Intention Sharing**: Maintain robustness while enhancing execution performance
    - **Optional Plugin** (LLM-based subtrees pre-planning): Improves planning and execution efficiency





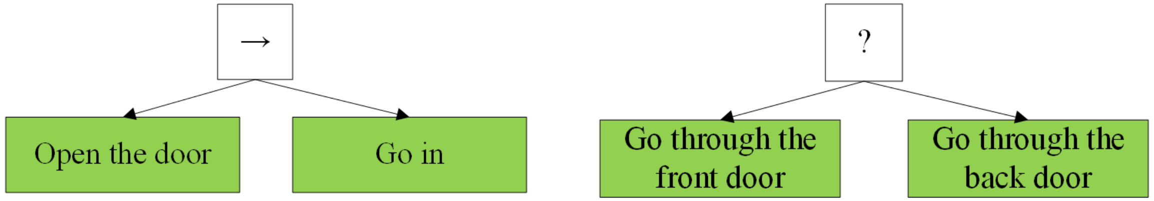
[1] Cai Z, Li M, Huang W, et al. BT Expansion: a sound and complete algorithm for behavior planning of intelligent robots with behavior trees[C] AAAI. 2021.

Background

☐ Classic Behavior Tree Representation

■ BT: a directed rooted tree

- Root at the top (usually omitted in graphics)
- Internal: control-flow nodes
- Leaf: condition/action nodes
- Returns to the parent: R={Success, Running, Failure}*



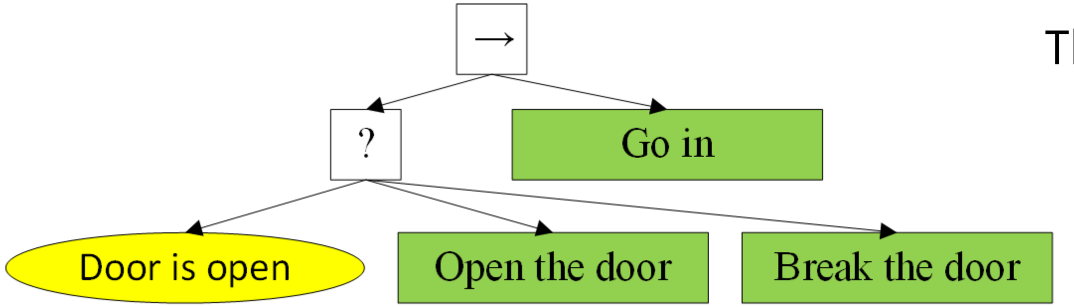
Sequence nodes
S: All children Succeed
F: One child Fails
R: Others

Fallback nodes
S: One child succeeds
F: All children fail
R: Others

Other control-flow nodes: Parallel nodes and decorator nodes.

☐ Behavior Trees: a running example

■ Robot entering a room



The BT can react and adapt to different run-time situations:

Door is open → Directly go in
Door not open → Open the door
Opening fails → Break the door
Someone helps open it → Skip breaking and go in

*Condition nodes only return success or failure.
For more details, we refer the readers to the book *Behavior Trees in Robotics and AI: An Introduction*, Michele Colledanchise, Petter Ögren, CRC Press.

Background

- BT Expansion[1]： A sound and complete algorithm for single-robot BT planning
  - STRIPS-style planning states and actions
    - An action a is a three tuple **<pre(a),add(a),del(a)>**, consisting of the precondition, add effects and delete effects of the action.

$$add(a) \cap del(a) = \emptyset$$

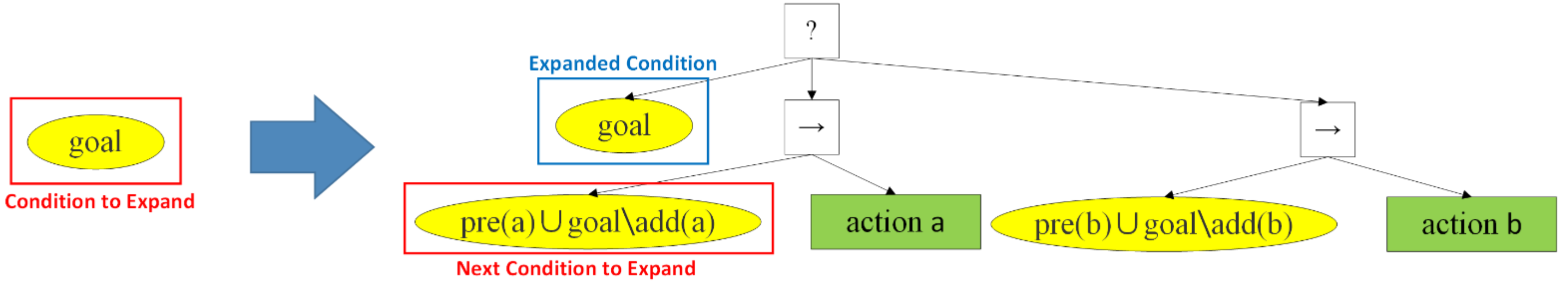
$$add(a) \cap pre(a) = \emptyset$$

Properties for well-defined actions*

$$s_{t+k} = f_a(s_t) = s_t \cup add(a) \setminus del(a)$$

State transition of a BT executing a*

  - Input & Output
    - **Input**: *Goal, Initial State, the finite set of actions*
    - **Output**: *Goal-Oriented BT*
  - The one-step expansion



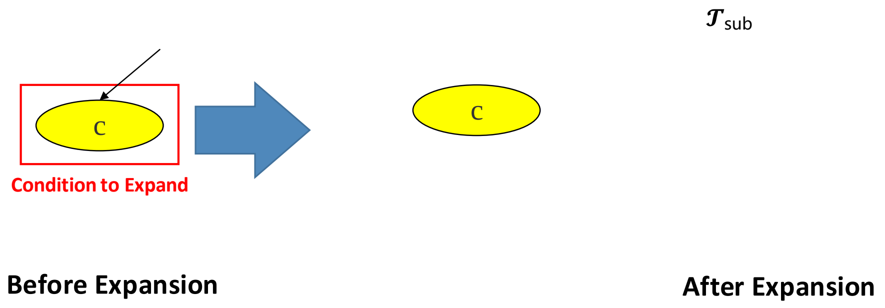
The Primary BT **BT Expanded after One Iteration**

[1] Cai Z, Li M, Huang W, et al. BT expansion: a sound and complete algorithm for behavior planning of intelligent robots with behavior trees[C] AAAI. 2021.

Background

- BT Expansion[1]： A sound and complete algorithm for single-robot BT planning
  - Input & Output
    - Input: *Goal, Initial State, the finite set of actions*
    - Output: *Goal-Oriented BT*
  - The one-step expansion

1. Preserve condition c

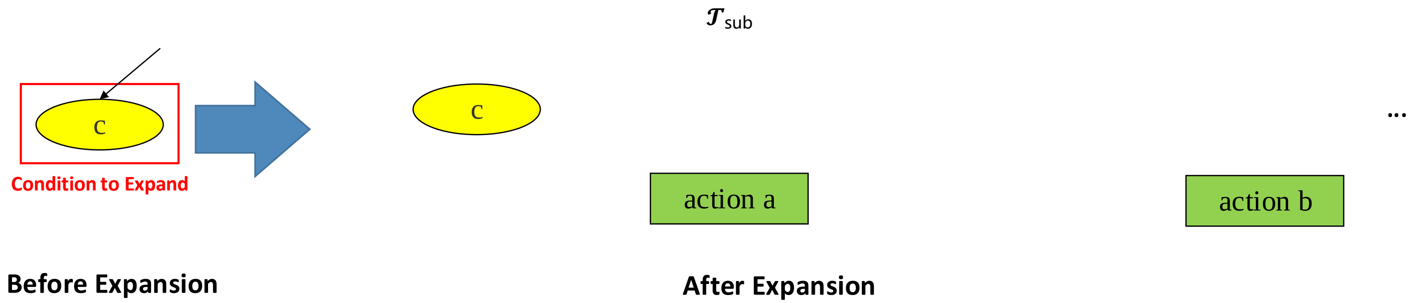


[1] Cai Z, Li M, Huang W, et al. BT expansion: a sound and complete algorithm for behavior planning of intelligent robots with behavior trees[C] AAAI. 2021.

Background

- BT Expansion[1]： A sound and complete algorithm for single-robot BT planning
  - Input & Output
    - Input: *Goal, Initial State, the finite set of actions*
    - Output: *Goal-Oriented BT*
  - The one-step expansion

1. Preserve condition c
2. Select actions

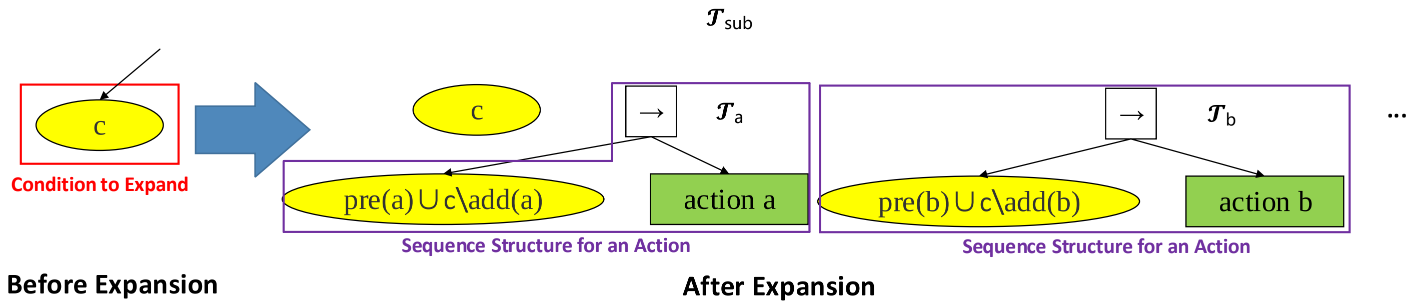


[1] Cai Z, Li M, Huang W, et al. BT expansion: a sound and complete algorithm for behavior planning of intelligent robots with behavior trees[C] AAAI. 2021.

Background

- BT Expansion[1]： A sound and complete algorithm for single-robot BT planning
  - Input & Output
    - Input: *Goal, Initial State, the finite set of actions*
    - Output: *Goal-Oriented BT*
  - The one-step expansion

1. Preserve condition c
2. Select actions
3. Construct sequence structures

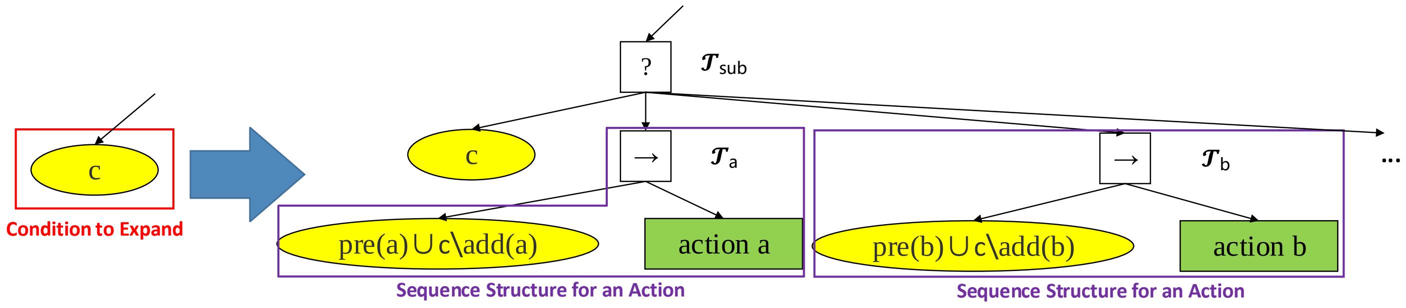


[1] Cai Z, Li M, Huang W, et al. BT expansion: a sound and complete algorithm for behavior planning of intelligent robots with behavior trees[C] AAAI. 2021.

Background

- BT Expansion[1]: A sound and complete algorithm for single-robot BT planning
  - Input & Output
    - Input: *Goal, Initial State, the finite set of actions*
    - Output: *Goal-Oriented BT*
  - The one-step expansion

1. Preserve condition c
2. Select actions
3. Construct sequence structures
4. Link to the top fallback node



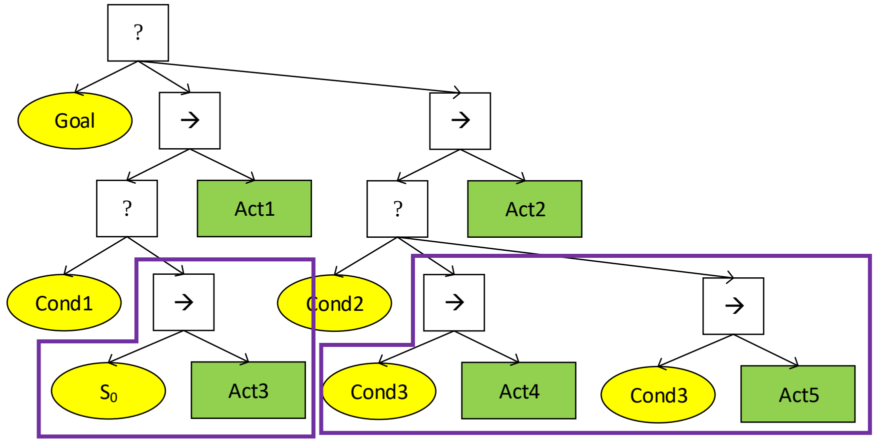
Before Expansion　　After Expansion

[1] Cai Z, Li M, Huang W, et al. BT expansion: a sound and complete algorithm for behavior planning of intelligent robots with behavior trees[C] AAAI. 2021.

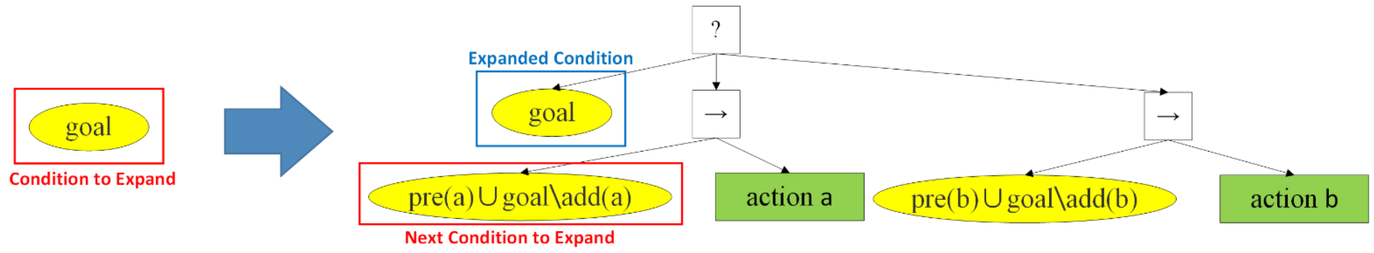
Background

- BT Expansion[1]: A sound and complete algorithm for single-robot BT planning
  - Input & Output
    - Input: *Goal, Initial State, the finite set of actions*
    - Output: *Goal-Oriented BT*
  - The one-step expansion -> BT Expansion

1. Preserve condition c
2. Select actions
3. Construct sequence structures
4. Link to the top fallback node
5. Repeat 2-4 until the BT can run in *Initial State*, or all conditions expanded.



BT Expansion



The Primary BT

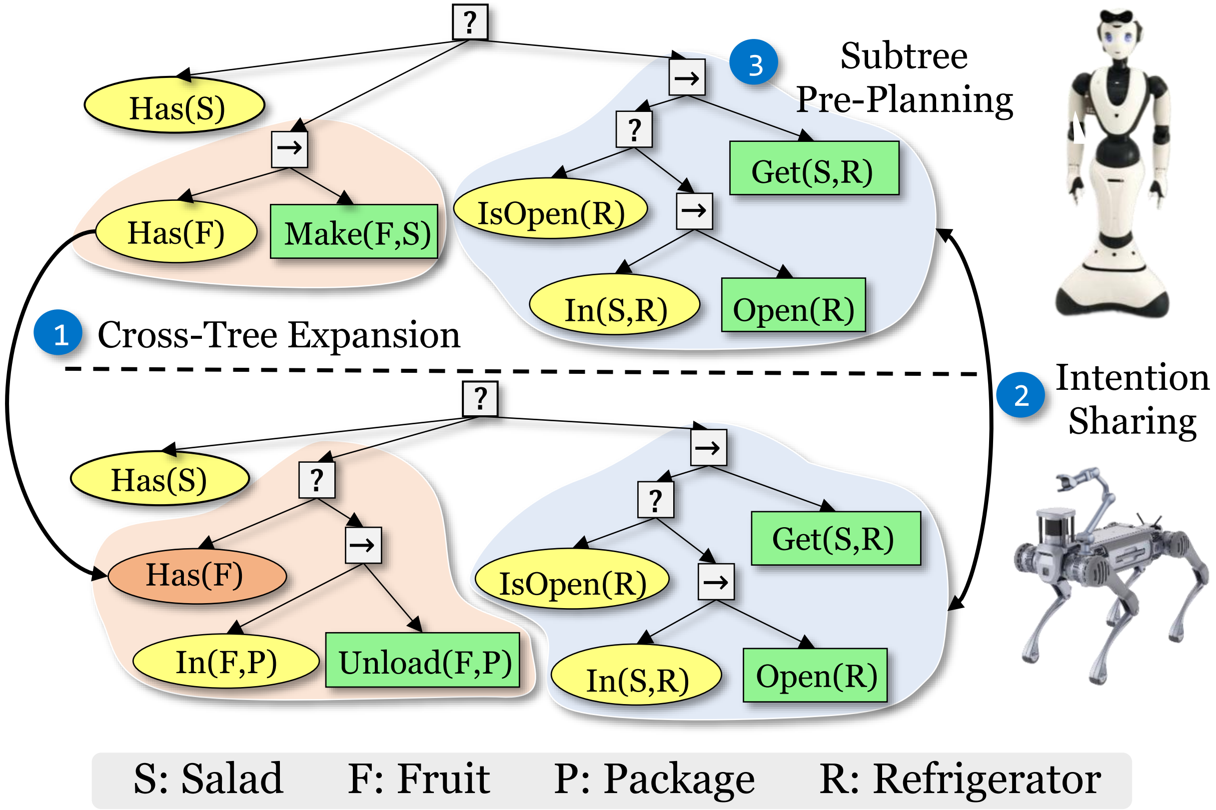
BT Expanded after One Iteration

[1] Cai Z, Li M, Huang W, et al. BT expansion: a sound and complete algorithm for behavior planning of intelligent robots with behavior trees[C] AAAI. 2021.

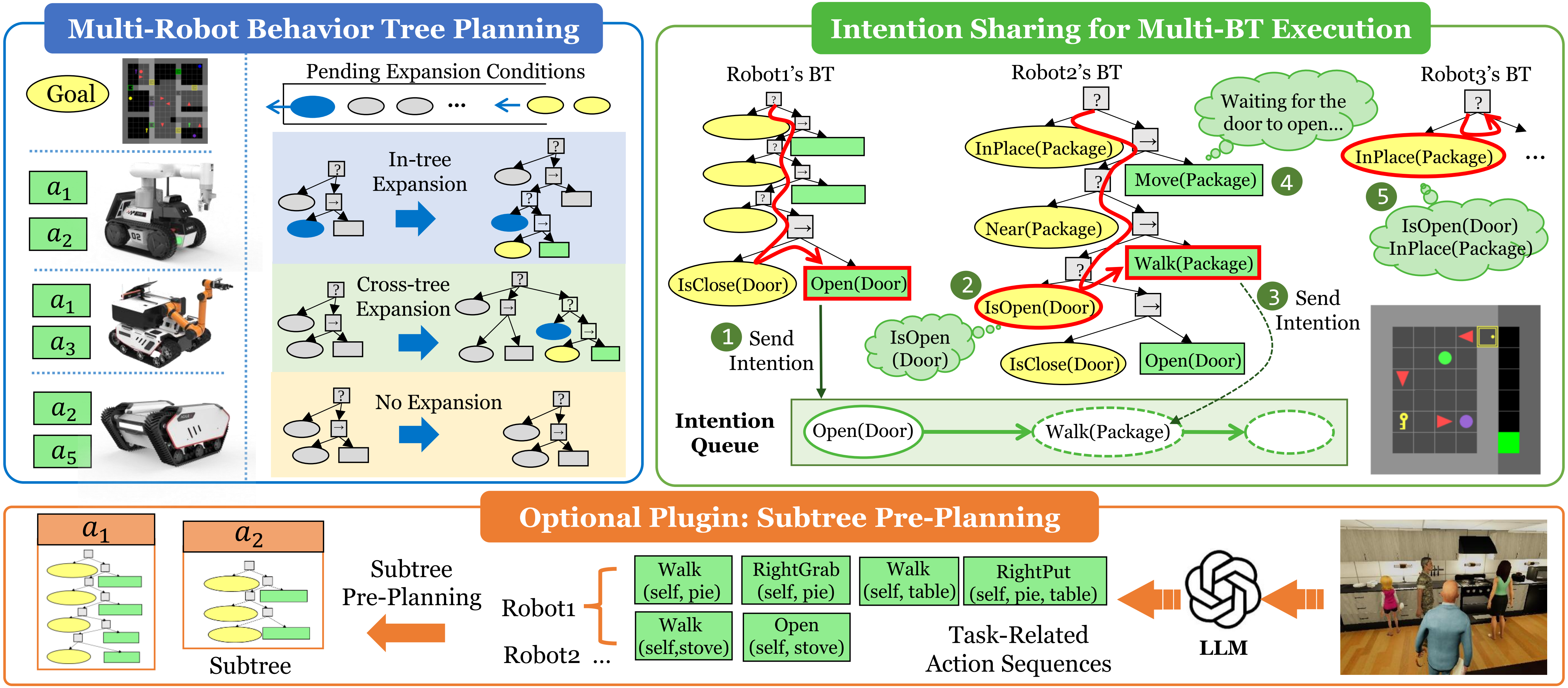
Proposed Algorithm: MRBTP

- BT Planning for Multi-Robots
  - Input & Output
    - Input: *Team Goal, Initial State,* ***Finite Action Set for each Robot***
    - Output: ***Goal-Oriented BT for each Robot***

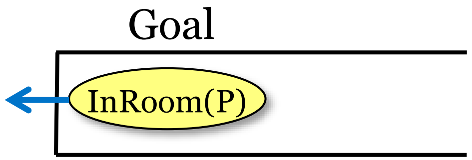
- Challenges
  - How to coordinating **heterogeneous actions** across multiple BTs to achieve team goals?
  - How to enhancing fault tolerance for **homogeneous actions** without redundant execution?

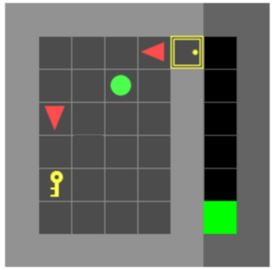


Proposed Algorithm: MRBTP



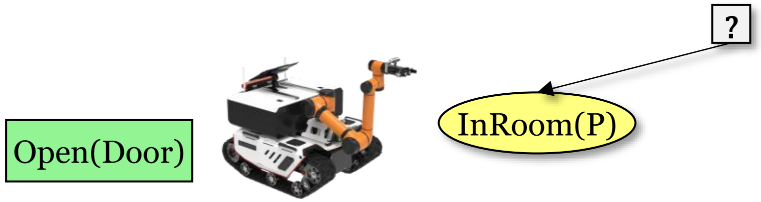
Proposed Algorithm: MRBTP





① Multi-Robot Behavior Tree Planning

1. Preserve condition g



Algorithm 2: MABTP

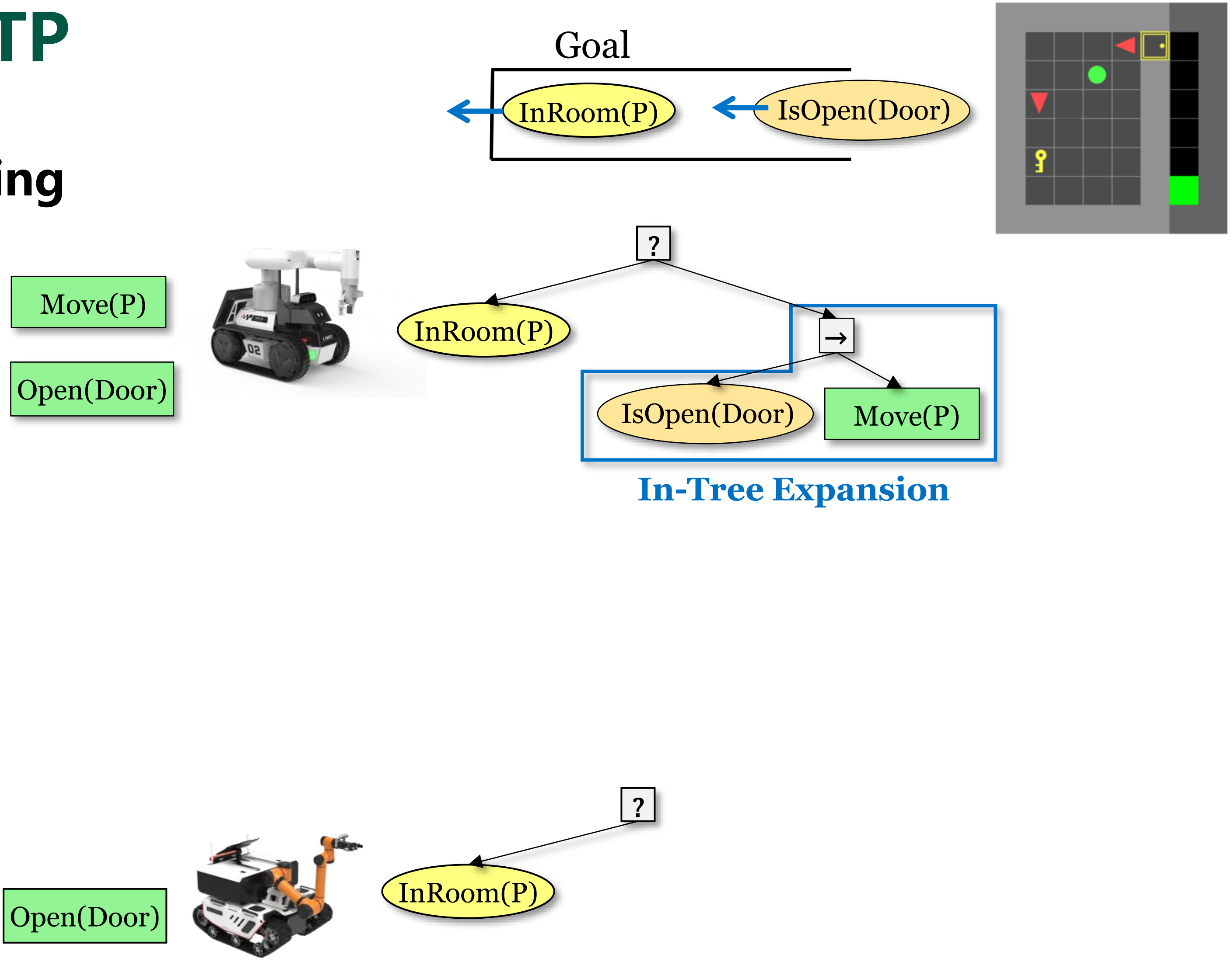
Input: problem $\langle \mathcal{S}, \mathcal{L}, \{\mathcal{A}_i\}_{i=1}^{n}, \mathcal{M}, s_0, c \rangle$
Output: solution $\Phi = \{\mathcal{T}_i\}_{i=1}^{n}$

$\mathcal{C}_U \leftarrow \{g\}$ ▷ conditions to be explored
$\mathcal{C}_E \leftarrow \emptyset$ ▷ expanded conditions
for $i = 1$ **to** n **do**
  $\mathcal{T}_i \leftarrow Fallback(g)$ ▷ init the BTs
while $\mathcal{C}_U \neq \emptyset$ **do**
  $c \leftarrow \texttt{Pop}(\mathcal{C}_U)$ ▷ explore c
  if $\texttt{HasSubSet}(c, \mathcal{C}_E)$ **then continue** ▷ prune
  for $i = 1$ **to** n **do**
    $\mathcal{T}_i, \mathcal{C}_{new} \leftarrow \texttt{ExpandOneRobot}(\mathcal{T}_i, \mathcal{A}_i, c)$
    if $\texttt{HasSubSet}(s_0, \mathcal{C}_{new})$ **then**
      return $\Phi = \{\mathcal{T}_i\}_{i=1}^{n}$ ▷ return a solution
    $\mathcal{C}_E \leftarrow \mathcal{C}_E \cup \mathcal{C}_{new}$
    $\mathcal{C}_U \leftarrow \mathcal{C}_U \cup \mathcal{C}_{new}$ ▷ add new conditions
return $\texttt{Unsolvable}$

Proposed Algorithm: MRBTP

① Multi-Robot Behavior Tree Planning

1. Preserve condition g
2. One-step expansion of each robot's BT



Algorithm 2: MABTP

Input: problem $\langle \mathcal{S}, \mathcal{L}, \{\mathcal{A}_i\}_{i=1}^{n}, \mathcal{M}, s_0, c \rangle$
Output: solution $\Phi = \{\mathcal{T}_i\}_{i=1}^{n}$

$\mathcal{C}_U \leftarrow \{g\}$ ▷ conditions to be explored
$\mathcal{C}_E \leftarrow \emptyset$ ▷ expanded conditions
for $i = 1$ **to** n **do**
  $\mathcal{T}_i \leftarrow Fallback(g)$ ▷ init the BTs
while $\mathcal{C}_U \neq \emptyset$ **do**
  $c \leftarrow$ Pop $(\mathcal{C}_U)$ ▷ explore c
  if HasSubSet $(c, \mathcal{C}_E)$ **then continue** ▷ prune
  for $i = 1$ **to** n **do**
    $\mathcal{T}_i, \mathcal{C}_{new} \leftarrow$ ExpandOneRobot $(\mathcal{T}_i, \mathcal{A}_i, c)$
    if HasSubSet $(s_0, \mathcal{C}_{new})$ **then**
      return $\Phi = \{\mathcal{T}_i\}_{i=1}^{n}$ ▷ return a solution
    $\mathcal{C}_E \leftarrow \mathcal{C}_E \cup \mathcal{C}_{new}$
    $\mathcal{C}_U \leftarrow \mathcal{C}_U \cup \mathcal{C}_{new}$ ▷ add new conditions
return Unsolvable

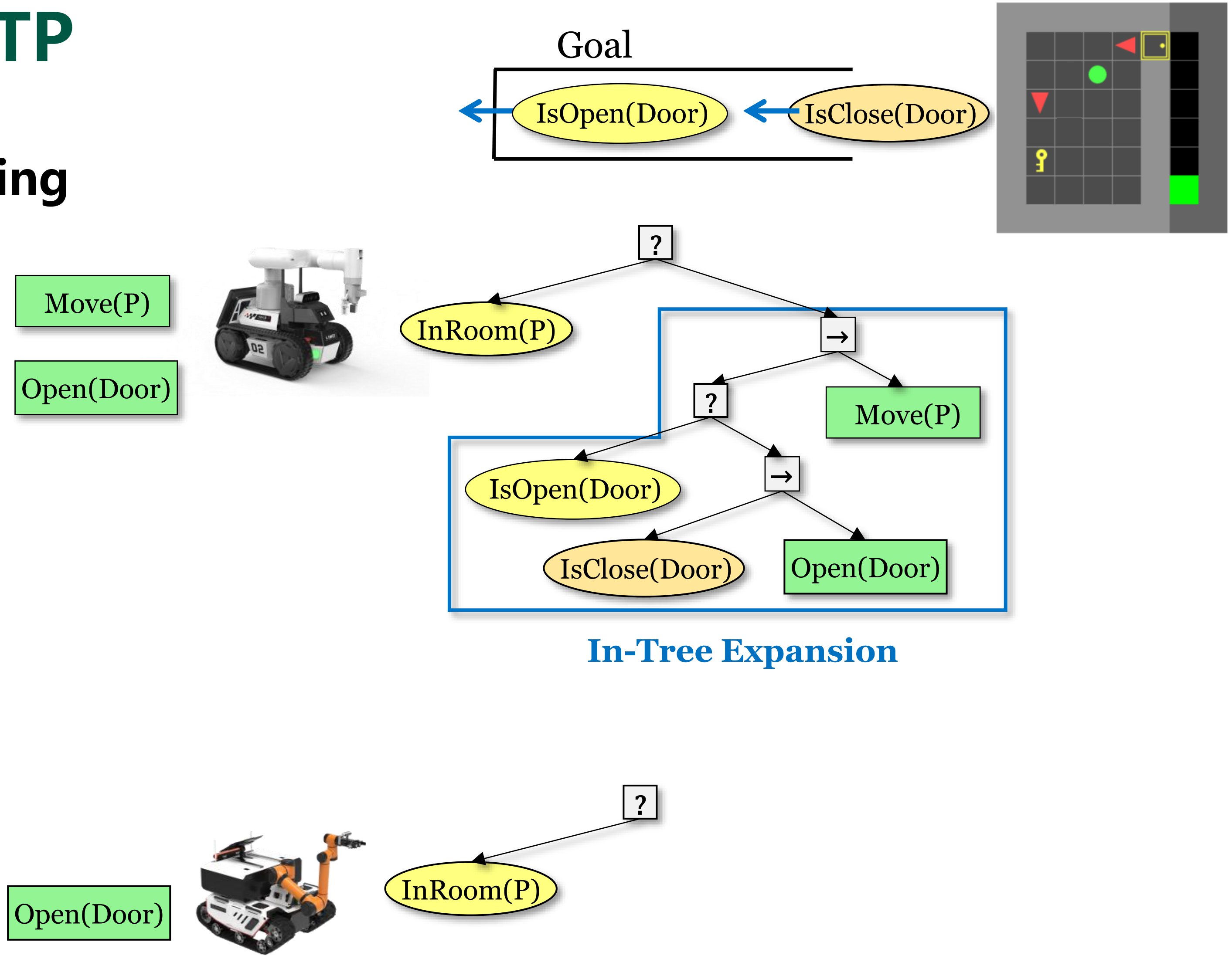
Proposed Algorithm: MRBTP

① Multi-Robot Behavior Tree Planning

1. Preserve condition g
2. One-step expansion of each robot's BT
   - In-Tree Expansion
   - Cross-Tree Expansion

Algorithm 1: One-step cross-tree expansion

function `ExpandOneRobot` $(\mathcal{T}, \mathcal{A}, c)$
    $\mathcal{T}_{new} \leftarrow c$ ▷ newly expanded subtree
    $\mathcal{C}_{new} \leftarrow \emptyset$ ▷ newly expanded conditions
    for each action $a \in \mathcal{A}$ **do**
        if $c \cap (pre(a) \cup add(a) \setminus del(a)) \neq \emptyset$ and $c \setminus del(a) = c$ **then**
            $c_a \leftarrow pre(a) \cup c \setminus add(a)$
            $\mathcal{T}_a \leftarrow Sequence(c_a, a)$
            $\mathcal{T}_{new} \leftarrow Fallback(\mathcal{T}_{new}, \mathcal{T}_a)$
            $\mathcal{C}_{new} \leftarrow \mathcal{C}_{new} \cup \{c_a\}$
    if $\mathcal{C}_{new} \neq \emptyset$ **then**
        if `ConditionInTree` $(c, \mathcal{T})$ **then**
            Replace c with $\mathcal{T}_{new}$ in $\mathcal{T}$ ▷ in-tree expand
        else if $\mathcal{T}_{new} \neq c$ **then**
            $\mathcal{T} \leftarrow Fallback(\mathcal{T}, \mathcal{T}_{new})$ ▷ cross-tree expand
return $\mathcal{T}, \mathcal{C}_{new}$



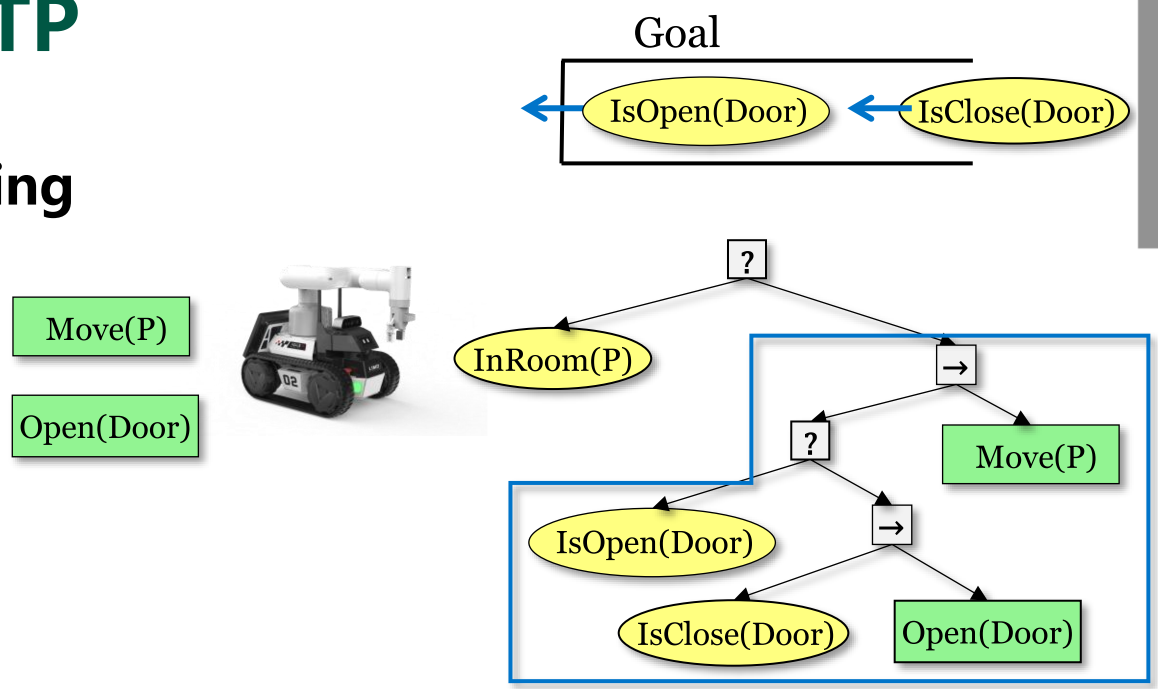
Proposed Algorithm: MRBTP

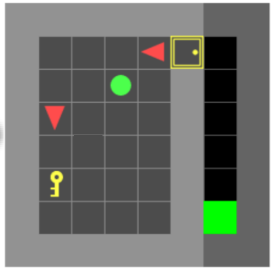
① Multi-Robot Behavior Tree Planning

1. Preserve condition g
2. One-step expansion of each robot's BT
   - In-Tree Expansion
   - Cross-Tree Expansion

Algorithm 1: One-step cross-tree expansion

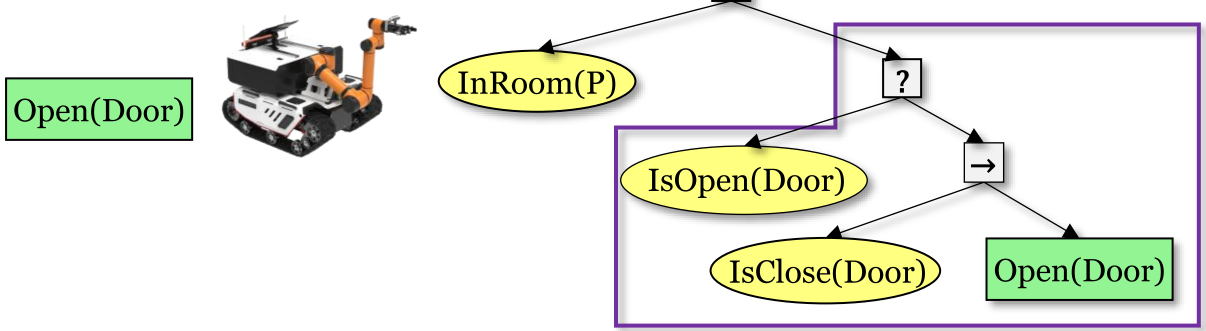
function ExpandOneRobot $(\mathcal{T}, \mathcal{A}, c)$
  $\mathcal{T}_{new} \leftarrow c$ ▷ newly expanded subtree
  $\mathcal{C}_{new} \leftarrow \emptyset$ ▷ newly expanded conditions
  for each action $a \in \mathcal{A}$ **do**
    if $c \cap (pre(a) \cup add(a) \setminus del(a)) \neq \emptyset$ and $c \setminus del(a) = c$ **then**
      $c_a \leftarrow pre(a) \cup c \setminus add(a)$
      $\mathcal{T}_a \leftarrow Sequence(c_a, a)$
      $\mathcal{T}_{new} \leftarrow Fallback(\mathcal{T}_{new}, \mathcal{T}_a)$
      $\mathcal{C}_{new} \leftarrow \mathcal{C}_{new} \cup \{c_a\}$
  if $\mathcal{C}_{new} \neq \emptyset$ **then**
    if ConditionInTree$(c, \mathcal{T})$ **then**
      Replace c with $\mathcal{T}_{new}$ in $\mathcal{T}$ ▷ in-tree expand
    else if $\mathcal{T}_{new} \neq c$ **then**
      $\mathcal{T} \leftarrow Fallback(\mathcal{T}, \mathcal{T}_{new})$ ▷ cross-tree expand
return $\mathcal{T}, \mathcal{C}_{new}$





In-Tree Expansion

Cross-Tree Expansion



Proposed Algorithm: MRBTP

① Multi-Robot Behavior Tree Planning

1. Preserve condition g
2. One-step expansion of each robot's BT
   - In-Tree Expansion
   - Cross-Tree Expansion

Algorithm 2: MABTP

Input: problem $\langle \mathcal{S}, \mathcal{L}, \{\mathcal{A}_i\}_{i=1}^{n}, \mathcal{M}, s_0, c \rangle$
Output: solution $\Phi = \{\mathcal{T}_i\}_{i=1}^{n}$

```
C_U ← {g}                                         ▷ conditions to be explored
C_E ← ∅                                           ▷ expanded conditions
for i = 1 to n do
    T_i ← Fallback(g)                             ▷ init the BTs
while C_U ≠ ∅ do
    c ← Pop(C_U)                                  ▷ explore c
    if HasSubSet(c, C_E) then continue            ▷ prune
    for i = 1 to n do
        T_i, C_new ← ExpandOneRobot(T_i, A_i, c)
        if HasSubSet(s_0, C_new) then
            return Φ = {T_i}_{i=1}^n              ▷ return a solution
        C_E ← C_E ∪ C_new
        C_U ← C_U ∪ C_new                         ▷ add new conditions
return Unsolvable
```

Algorithm 1: One-step cross-tree expansion

```
function ExpandOneRobot(T, A, c)
    T_new ← c                                     ▷ newly expanded subtree
    C_new ← ∅                                     ▷ newly expanded conditions
    for each action a ∈ A do
        if c ∩ (pre(a) ∪ add(a) \ del(a)) ≠ ∅ and c \ del(a) = c then
            c_a ← pre(a) ∪ c \ add(a)
            T_a ← Sequence(c_a, a)
            T_new ← Fallback(T_new, T_a)
            C_new ← C_new ∪ {c_a}
    if C_new ≠ ∅ then
        if ConditionInTree(c, T) then
            Replace c with T_new in T             ▷ in-tree expand
        else if T_new ≠ c then
            T ← Fallback(T, T_new)                ▷ cross-tree expand
    return T, C_new
```

□ Proof of Soundness and Completeness

Definition 1 (Multi-BT System). *A n-robot BT system is a four-tuple $\langle \Phi, f_\Phi, r_\Phi, \Delta t_\Phi \rangle$, where $\Phi = \{\mathcal{T}_i\}_{i=1}^{n}$ is the set of BTs, $f_\Phi : \mathcal{S} \mapsto \mathcal{S}$ is the team state transition function, Δt_Φ is the team time step, $r_\Phi : \mathcal{S} \mapsto \{\, \mathcal{S}, \mathcal{R}, \mathcal{F} \,\}$ is the team region partition.*

Definition 2 (Finite Time Successful). *Φ is finite time successful (FTS) from region of attraction (ROA) R to condition c, if $\forall s_0 \in R$ there is a finite time τ such that for any $t < \tau$, $r_\Phi(s_t) = \mathcal{R}$, and for any $t \geq \tau$, $r_\Phi(s_t) = \mathcal{S}$, $c \in s_t$.*

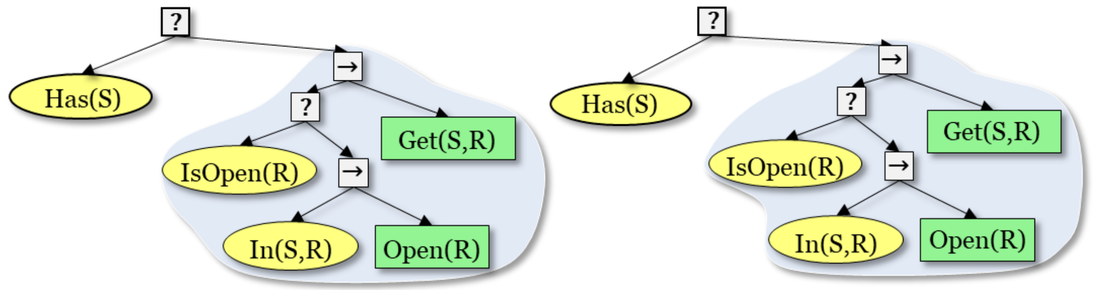
Proposition 1. *Given $\mathcal{T}$ is FTS from R to g, if $\mathcal{T}$ is expanded by Algorithm 1 to $\mathcal{T}'$ given c, c is in $\mathcal{T}$ and $\mathcal{C}_{new} \neq \emptyset$, then $\mathcal{T}'$ is FTS from $R' = R \cup \{s \in \mathcal{S} | c_a \subseteq s, c_a \in \mathcal{C}_{new}\}$ to g.*

Proposition 2. *If $\mathcal{T}$ is expanded by Algorithm 1 to $\mathcal{T}'$ given c, c is not in $\mathcal{T}$ and $\mathcal{C}_{new} \neq \emptyset$, then $\mathcal{T}'$ is FTS from $\mathcal{S}_{new} = \{s \in \mathcal{S} | c_a \subseteq s, c_a \in \mathcal{C}_{new}\}$ to c.*

Proposition 4. *Algorithm 2 is sound, i.e. if it returns a result Φ rather than* `Unsolvable`*, then Φ is a solution of Problem 1.*

Proposition 5. *Algorithm 2 is complete, i.e., if Problem 1 is solvable, the algorithm returns a Φ which is a solution.*

Proposed Algorithm: MRBTP



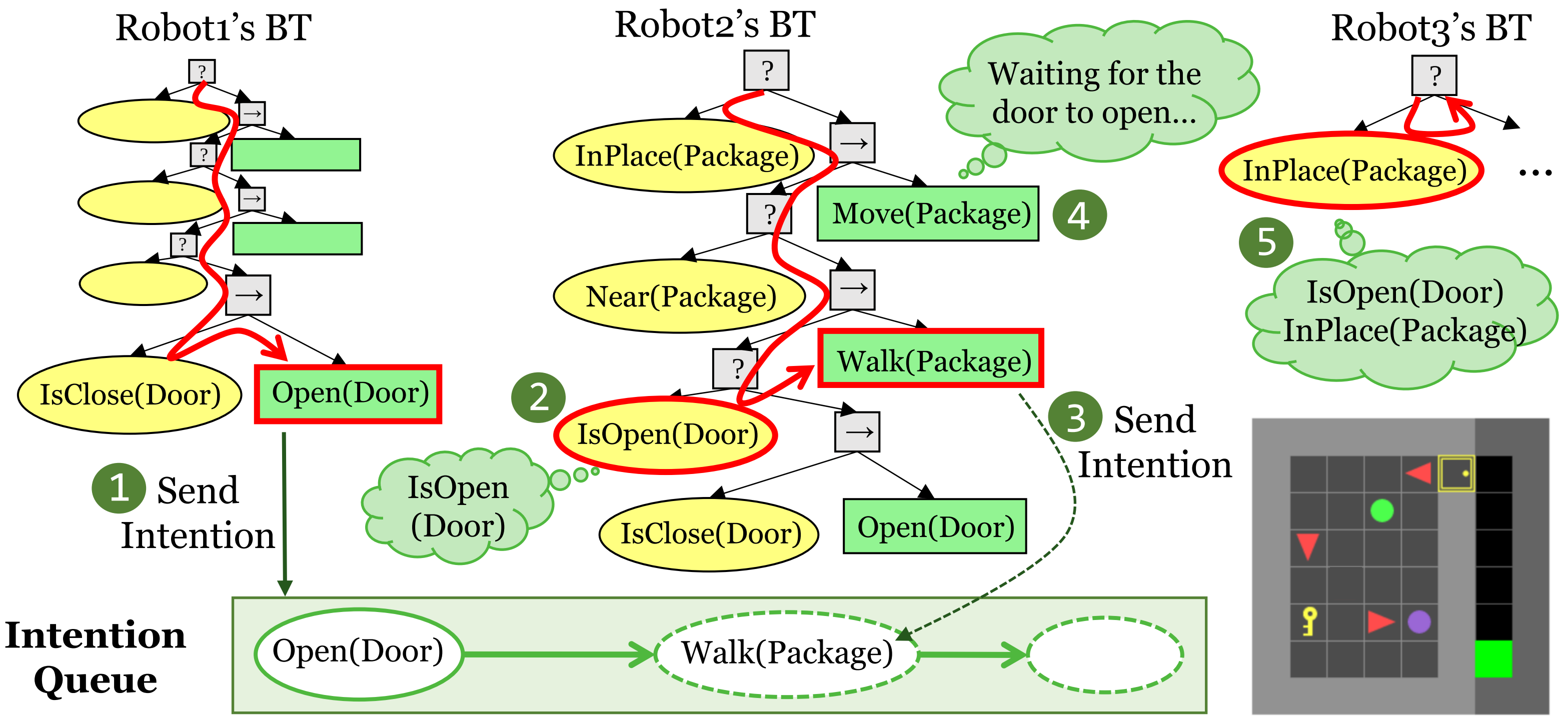
② Intention Sharing for Multi-BT Execution

- Enhancing fault tolerance for homogeneous actions without redundant execution
- Intention Queue
  - ➢ Belief Success Space: The set of states a robot anticipates will be satisfied
  - ➢ Belief Failure Space: The set of states a robot anticipates will not be satisfied
- Parallelism and Blocking

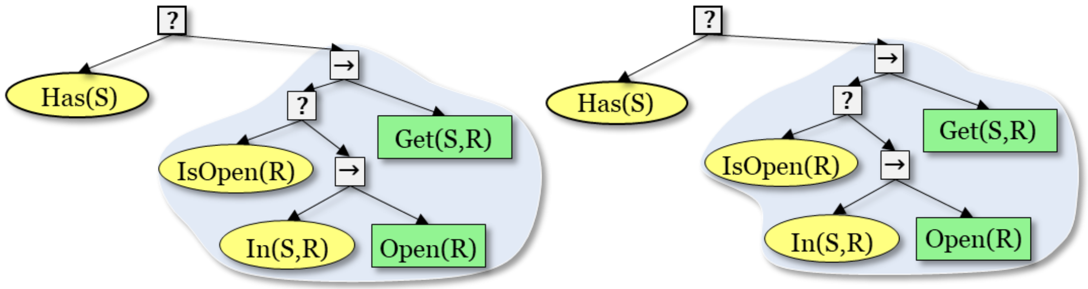
$$\mathcal{B}_i^S = \bigcup_{k=1}^{j-1} (add(a_k) \setminus del(a_k))$$

$$\mathcal{B}_i^F = \bigcup_{k=1}^{j-1} (del(a_k) \setminus add(a_k))$$

Robot i's Belief Space
(j: Index in the Queue)

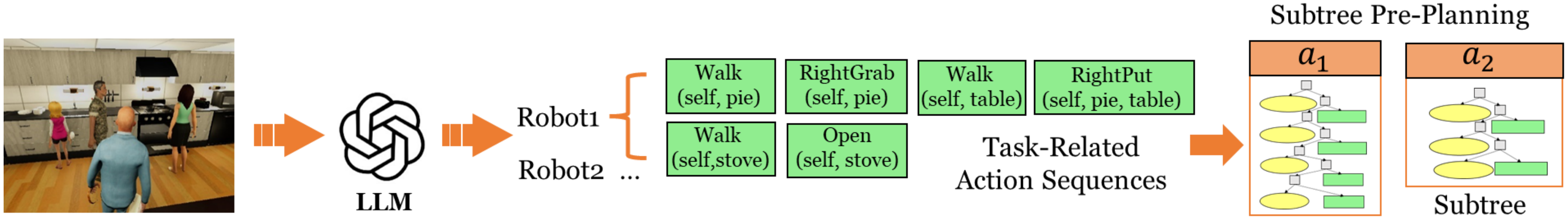


Proposed Algorithm: MRBTP



③ Optional Plugin: Subtree Pre-Planning

- During planning: Repeated tree generation due to overlapping action spaces
- During execution: High communication overhead from short-horizon atomic actions



- Acquiring task-related action sequences via LLM
  - Preconditions, Add Effects, and Delete Effects of the Action Sequence $A = (a_1, a_1, \dots, a_1)$

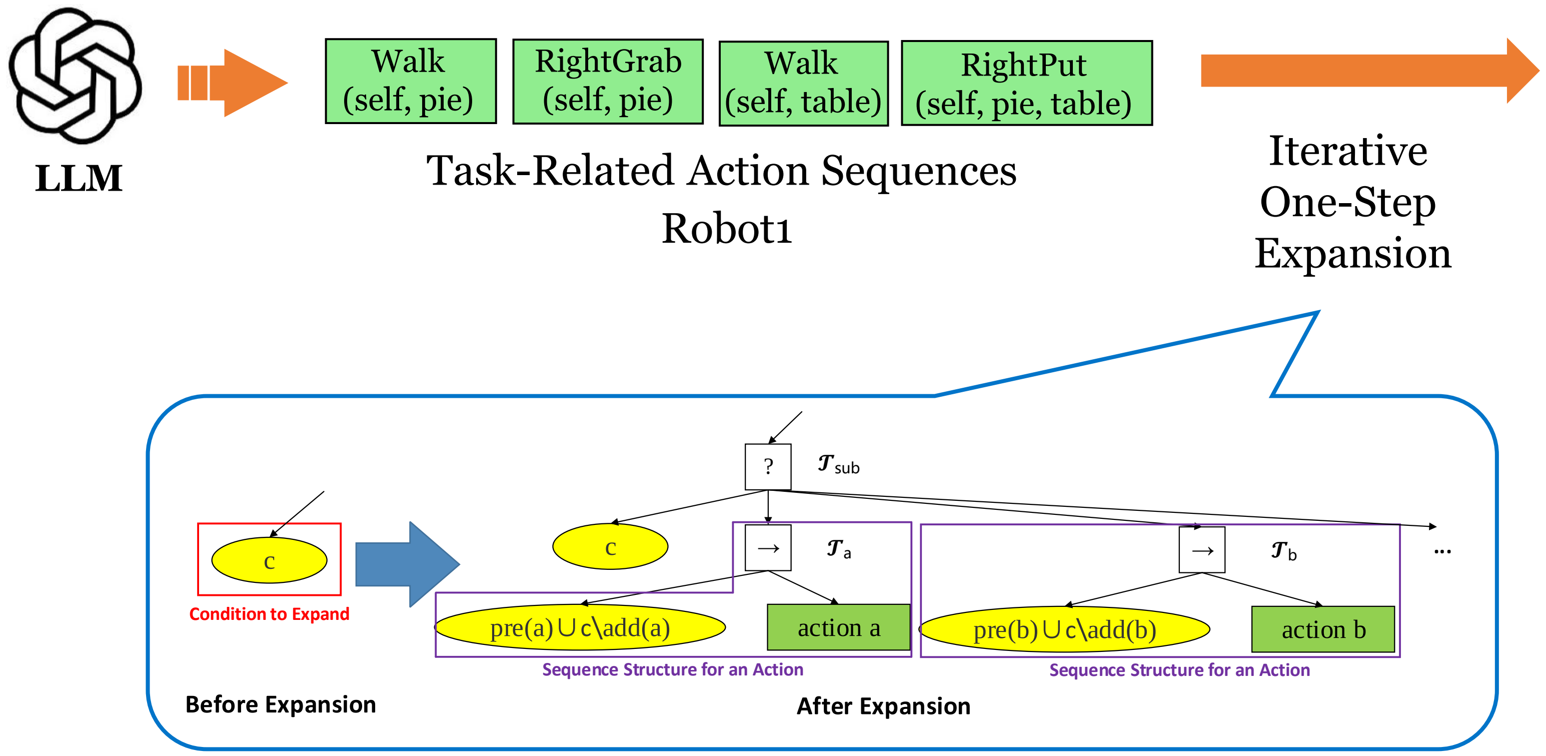
$$pre(A) = \bigcup_{j=1}^{m} \left(pre(a_j) \setminus \bigcup_{k=1}^{j} add(a_j) \right) \qquad add(A) = \bigcup_{j=1}^{m} (add(a_j) \setminus del(a_j)) - pre(A) \qquad del(A) = \bigcup_{j=1}^{m} (del(a_j) \setminus add(a_j))$$

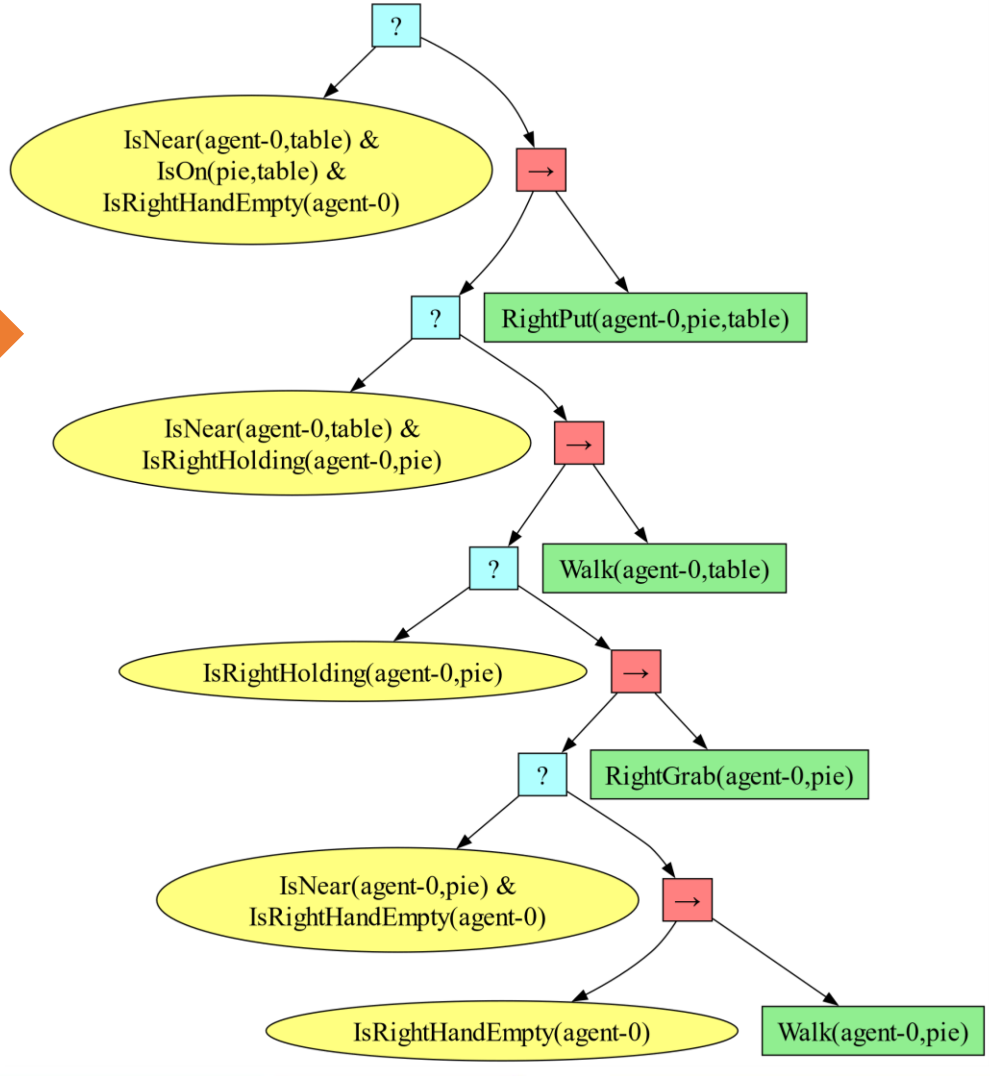
- Subtree Pre-Planning
  - Iterative one-step expansion to generate the subtree of A

Proposed Algorithm: MRBTP

③ Optional Plugin: Subtree Pre-Planning

- Subtree Pre-Planning
  - ➢ Iterative one-step expansion to generate the subtree of A





Evaluation

- Scenarios
  - Warehouse Management
    - Minigrid[1] , 4-8 robots in 4 rooms with random packages
  - Home Service
    - VirtualHome[2], 2-4 robots handle 10+ objects and 100+ actions
- Settings
  - Homogeneity α
    - The proportion of redundant actions assigned to robots
    - $\alpha = 1$: complete homogeneity
    - $\alpha = 0$: complete heterogeneity
  - Intention Sharing
    - Atomic IS: intention sharing among atomic action
    - Subtree IS: intention sharing among subtree

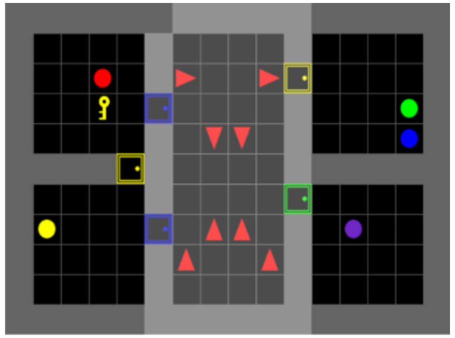


Table 1: Action predicates for different scenarios

Scenarios	Included Action Predicates
Warehouse Mangement	`GoToInRoom`, `GoBtwRoom`, `PickUp`, `PutInRoom`, `PutNearInRoom`, `Toggle`
Home Service	`Walk`, `LeftPut`, `LeftPutIn`, `LeftGrab`, `RightGrab` , `Open`, `Close` `RightPut`, `RightPutIn`, `SwitchOn`,`SwitchOff`

Evaluation

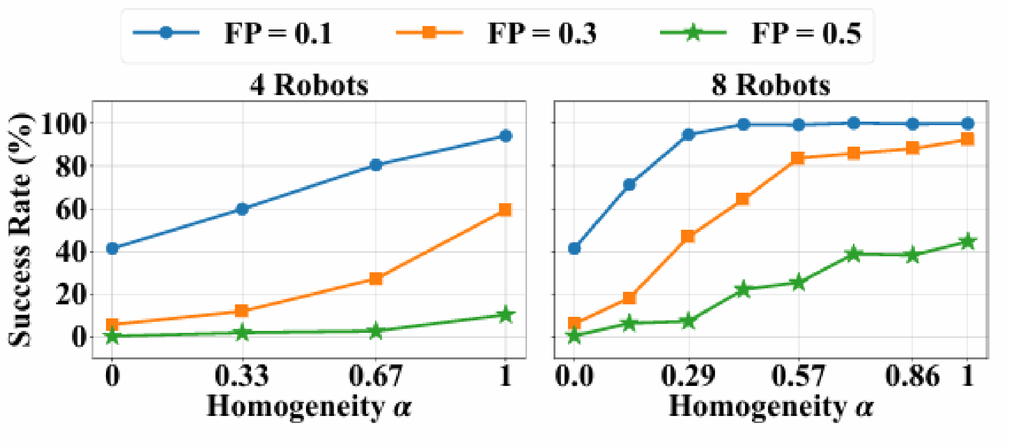
- Evaluation Metrics
  - Success Rate (SR): Task success rate across trials
  - Team Steps (TS): Total steps for all robots to complete tasks in parallel
  - Total Robot Steps (RS): Sum of steps taken by each robot independently
- Performance Comparison
  - MRBTP achieves a **100% success rate** across all settings due to its **cross-tree expansion**
  - MRBTP outperforms BT-Expansion in execution efficiency due to intention sharing

Table 1: Performance comparison with baseline in warehouse management (4 robots, averaged over 500 trials).

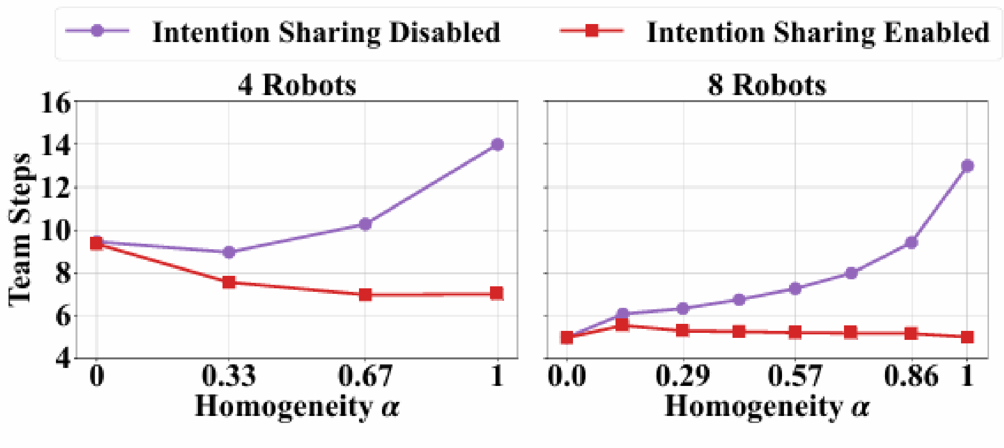
Method	$\alpha = 1$			$\alpha \approx 0.5$	$\alpha = 0$
	SR(%)	**TS**	**RS**	**SR(%)**	**SR(%)**
BT-Expansion	100	8.8	33.8	12.4	4.6
MRBTP	100	5.8	15.3	100	100

Evaluation

- Robustness
  - Action Failure Probability (FP): The probability that a robot fails to execute an action
  - Robustness improves with higher homogeneity and more robots
- Impact of **Intention Sharing** on Execution Efficiency
  - Intention sharing **reduces team steps** and maintains superior execution efficiency
  - In **fully heterogeneous** scenarios, intention sharing achieves the **greatest efficiency gains**



(a) The impact of homogeneity on robustness



(b) The impact of intention sharing on execution efficiency

Evaluation

☐ Impact of **Task-Specific Subtree Pre-Planning**

- Subtree pre-planning and reuse **significantly reduces BTs planning time**
- Intention sharing within subtrees **improves execution efficiency while minimizing communication overhead**

☐ LLMs and Feedback

- Feedback (F) and No Feedback (NF)
- Feedback: grammar errors, Insufficient/incoherent action sequences,
- Feedback effectively **enhances both planning and execution efficiency**
- It becomes more effective as the reasoning ability of LLMs improves

Table 4: Execution efficiency across different LLMs under $\alpha = 1$ with subtree and subtree intention sharing.

Models	No Feedback			Feedback		
	TS	RS	Comm.	TS	RS	Comm.
GPT-3.5-turbo	81.6	223.6	5.1	80.0	219.0	5.1
GPT-4o-mini	78.9	217.4	6.7	77.1	205.2	6.6
GPT-4o	77.4	200.9	6.3	74.9	190.7	6.3

Table 3: Planning efficiency with pre-planned subtrees. The average response time per LLM invocation is 4.2 seconds.

Homogeneity	Subtree	Feedback	EC	PT (s)
$\alpha = 1$	-	-	8033.3	Timeout
	✓	-	998.1	12.4
	✓	✓	**384.3**	**3.7**
$\alpha \approx 0.5$	-	-	7882.5	Timeout
	✓	-	623.8	7.2
	✓	✓	**267.9**	**2.6**
$\alpha = 0$	-	-	2695.5	20.2
	✓	-	576.6	5.6
	✓	✓	**146.8**	**1.4**

Table 2: Execution efficiency with subtree pre-planning and intention sharing.

Homogeneity		$\alpha = 1$						$\alpha \approx 0.5$						$\alpha = 0$					
Subtree		-	-	✓	✓	✓	✓	-	-	✓	✓	✓	✓	-	-	✓	✓	✓	✓
Subtree IS		-	-	-	-	✓	✓	-	-	-	-	✓	✓	-	-	-	-	✓	✓
Atomic IS		-	✓	-	✓	-	✓	-	✓	-	✓	-	✓	-	✓	-	✓	-	✓
TS	NF	161	159.4	114.4	109.6	78.9	79.8	139.7	137.5	126.2	119.6	86.9	102.5	73.7	68.5	96.1	94.8	75.6	78.0
	F	-	-	116.7	114.2	**77.1**	79.16	-	-	124.6	126.0	**80.8**	96.2	-	-	107.1	106.4	**70.4**	76.8
RS	NF	570.8	557.3	374	359.4	217.4	219.1	385.2	380.1	345.8	326.6	209.6	222	128.6	128.6	128.6	128.6	128.6	128.6
	F	-	-	377	370.9	**205.2**	208	-	-	380.7	348.2	**192.2**	209	-	-	128.6	128.6	128.6	128.6
Comm.	NF	0.0	63.8	0.0	7.1	6.7	14.1	0.0	43.5	0.0	8.0	7.1	20.9	0.0	15.2	0.0	2.8	6.5	9.3
	F	-	-	0.0	4.8	**6.6**	12.2	-	-	0.0	4.4	**6.4**	13.2	-	-	0.0	0.7	**5.2**	6.0

Summary

☐ Conclusion

- MRBTP, the first **sound and complete** algorithm for solving the multi-robot BT planning problem
- MRBTP uses **cross-tree expansion** to coordinate heterogeneous actions and enhances robustness and execution efficiency with backup structures and **intention sharing**
- **Optional LLMs plugin** speeds up planning and improves collaboration with pre-planned long-horizon subtrees

☐ Future Work

- Real-time deadlock detection and resolution in BT execution
- Efficient large-scale BT communication, coordination, and emergent intelligence
- Planning under uncertainty and periodic tasks

国防科技大学
NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY

AAAI-25 / IAAI-25 / EAAI-25
FEBRUARY 25 – MARCH 4, 2025 | PHILADELPHIA, USA

MRBTP: Efficient Multi-Robot Behavior Tree Planning and Collaboration

Thank you!

Q & A

Email: dids_ei@163.com

Code: https://github.com/DIDS-EI/MRBTP