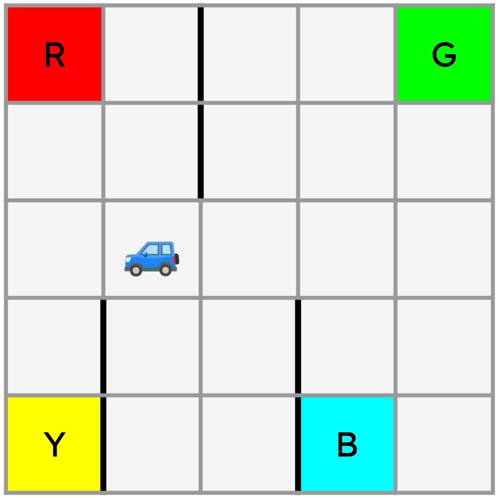
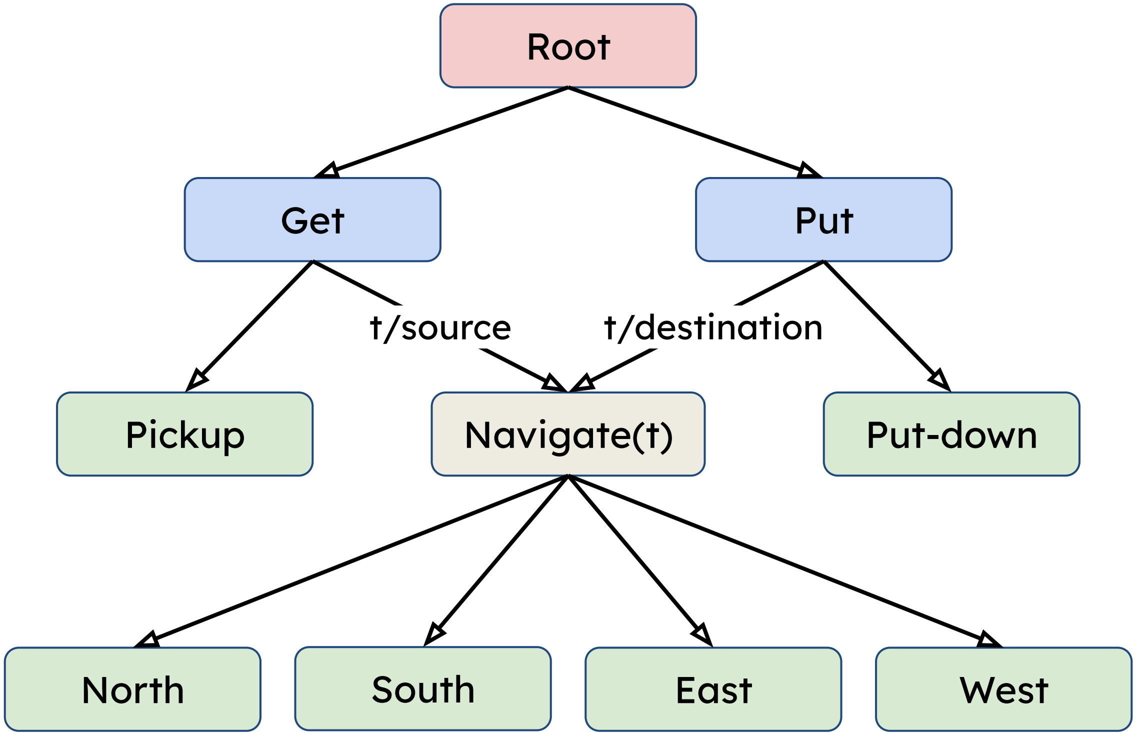
Lecture 11: Hierarchical Reinforcement Learning

B. Ravindran

Hierarchical Problem Solving

The Taxi-domain



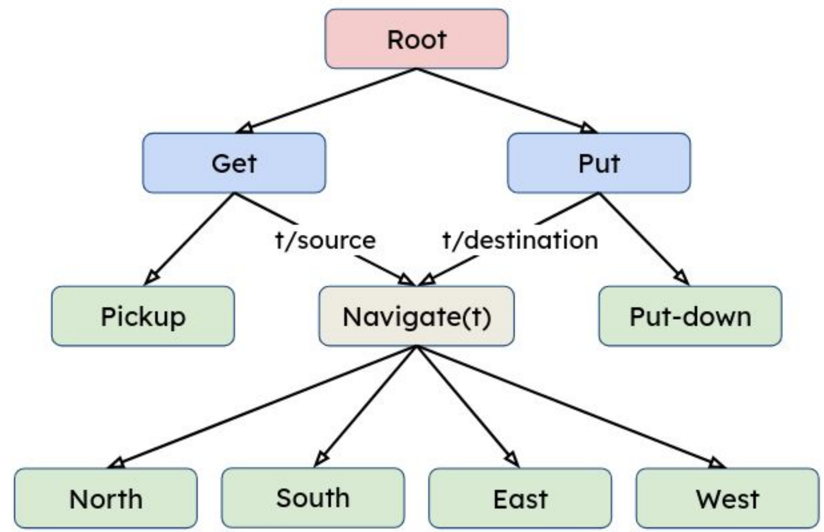


Hierarchies

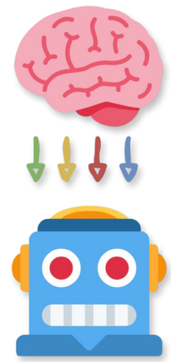
- Natural problem abstraction for humans
  - Divide and conquer
- Scaling-up
- Ease of re-use
  - Skill/Knowledge Transfer
  - Continual Learning
- Aggressive abstraction possible
  - Each sub-task requires only a small subset of the features
- More *explainable*

Hierarchical Reinforcement Learning

- Many frameworks
  - Options
  - MaxQ
  - HAM
  - Airports



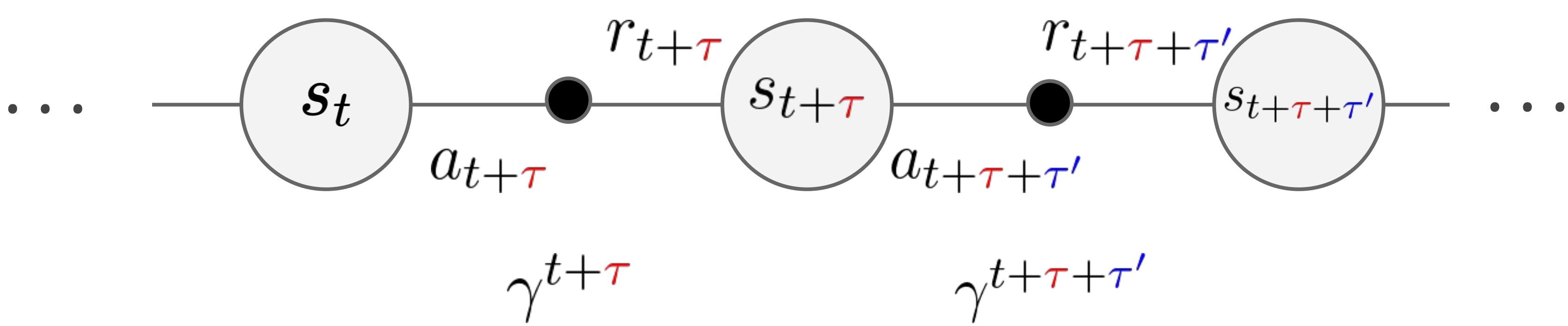
- Essentially let the agent learn skills and reuse them



Semi-Markov Decision Process

Semi-Markov Decision Process

- SMDP is a generalization of MDP
- The time between decisions is a random variable
- Consider the system remaining in each state for a random waiting time before transitioning to next state - Holding time (τ)
- Traditionally modelled as product of marginals



Semi-Markov Decision Process

- SMDP is a generalization of MDP
- The time between decisions is a random variable
- Consider the system remaining in each state for a random waiting time before transitioning to next state - Holding time (τ)
- Traditionally modelled as product of marginals
- Bellman equations:

$$V^*(s) = \max_{a \in A_s} \left[R(s,a) + \sum_{s',\tau} \gamma^\tau P(s', \tau \mid s, a) V^*(s') \right]$$

$$Q^*(s,a) = R(s,a) + \sum_{s',\tau} \gamma^\tau P(s', \tau \mid s, a) \max_{a' \in A_{s'}} Q^*(s', a')$$

SMDP Q-Learning

One-step Q-Learning

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a' \in A_{s_{t+1}}} Q(s_{t+1}, a') - Q(s_t, a_t) \right]$$

SMDP Q-Learning

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+\tau} + \gamma^{\tau} \max_{a' \in A_{s_{t+\tau}}} Q(s_{t+\tau}, a') - Q(s_t, a_t) \right]$$

Options Framework

Options (Sutton, Precup, & Singh, 1999): A generalization of actions to include temporally-extended courses of action

An option is a triple $o = \langle I, \pi_o, \beta \rangle$

- $I \subseteq S$ is the set of states in which o may be started
- $\pi_o : \Psi \rightarrow [0,1]$ is the (stochastic) policy followed during o
- $\beta : S \rightarrow [0,1]$ is the probability of terminating in each state

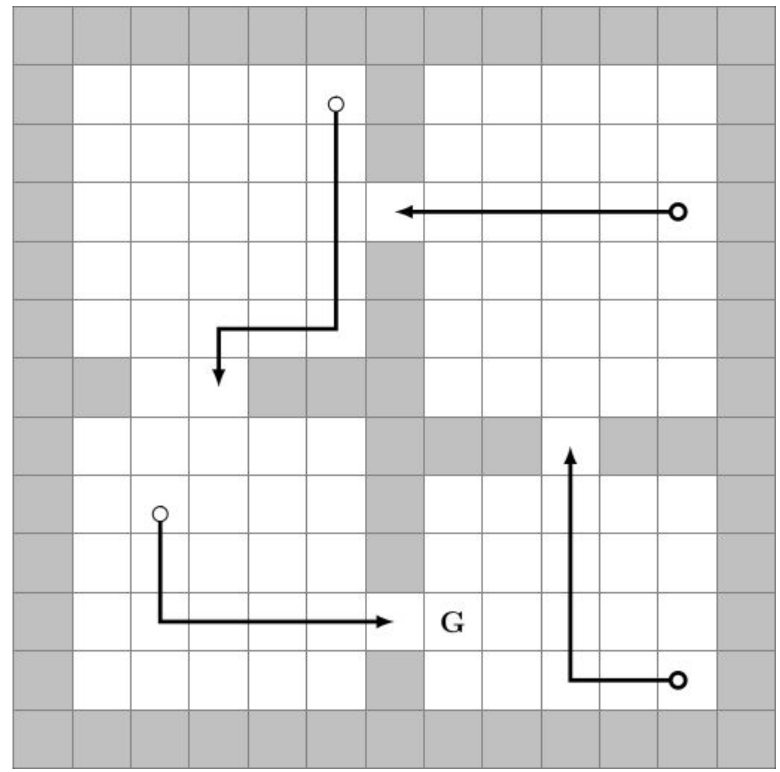
Generalising over Tasks

- Each task has a different reward structure in the state space

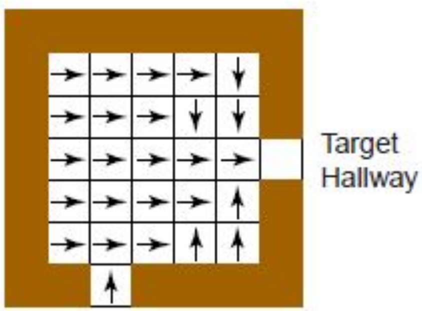
- Options provide a model for subtasks

- Semi-Markov Processes

- Can use generalization of TD, Q-learning, SARSA, etc. with options



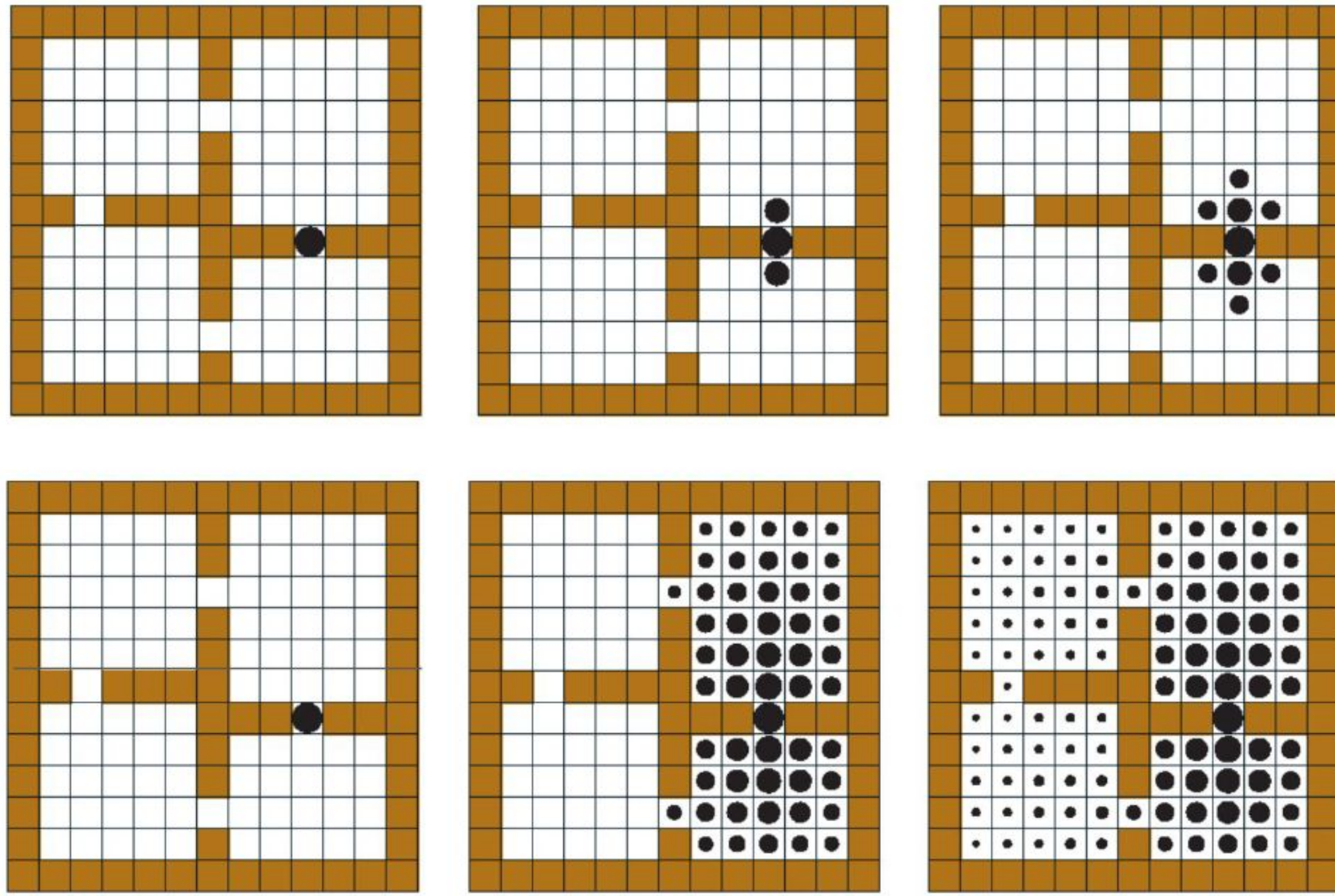
Speedup using Options



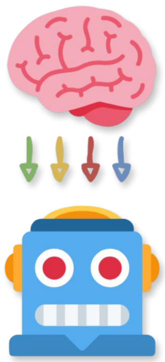
Underlying policy of one hallway option

Primitive Actions

Hallway Options

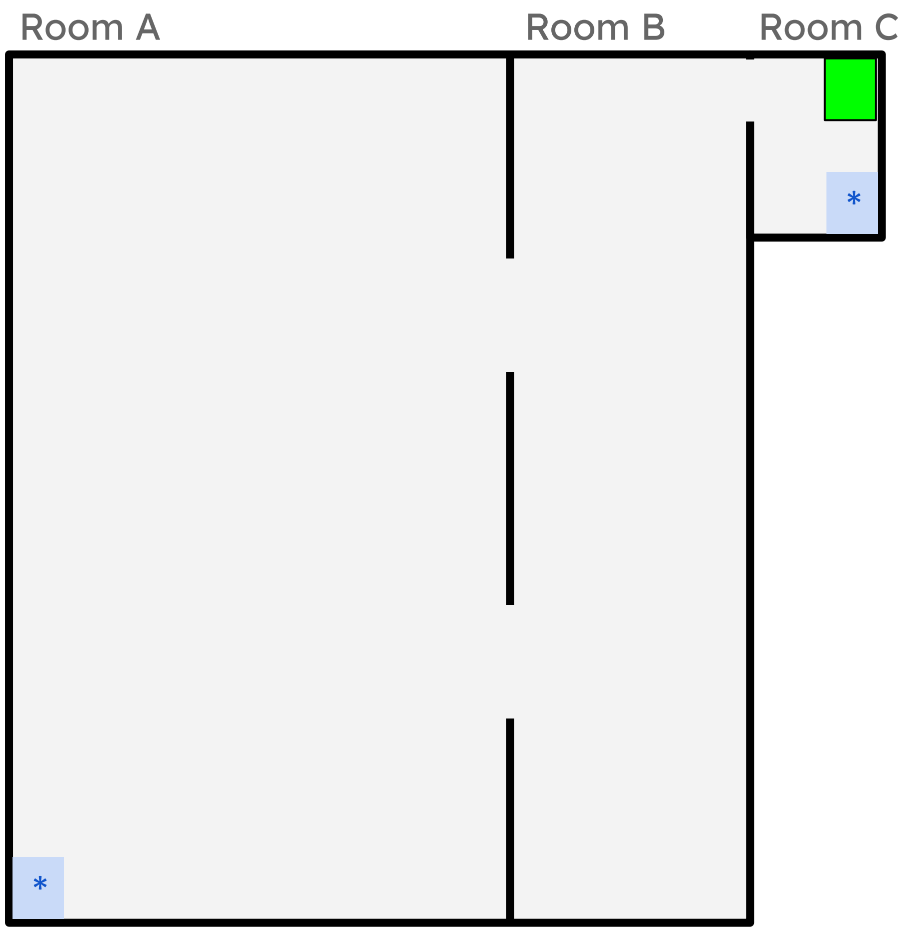


Initial Values | Iteration # 1 | Iteration # 2



Notions of Optimality

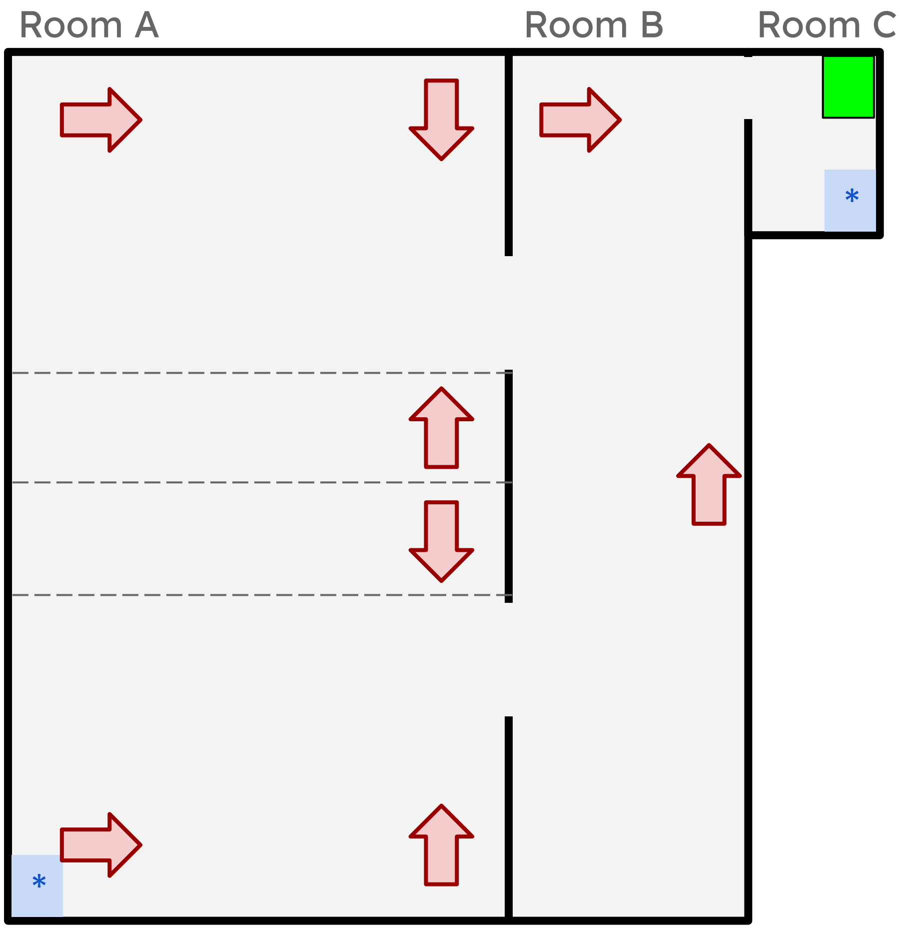
Notions of Optimality



Hierarchy: Room A -> Room B -> Room C

- ❑ Recursive
- ❑ Hierarchical
- ❑ Flat

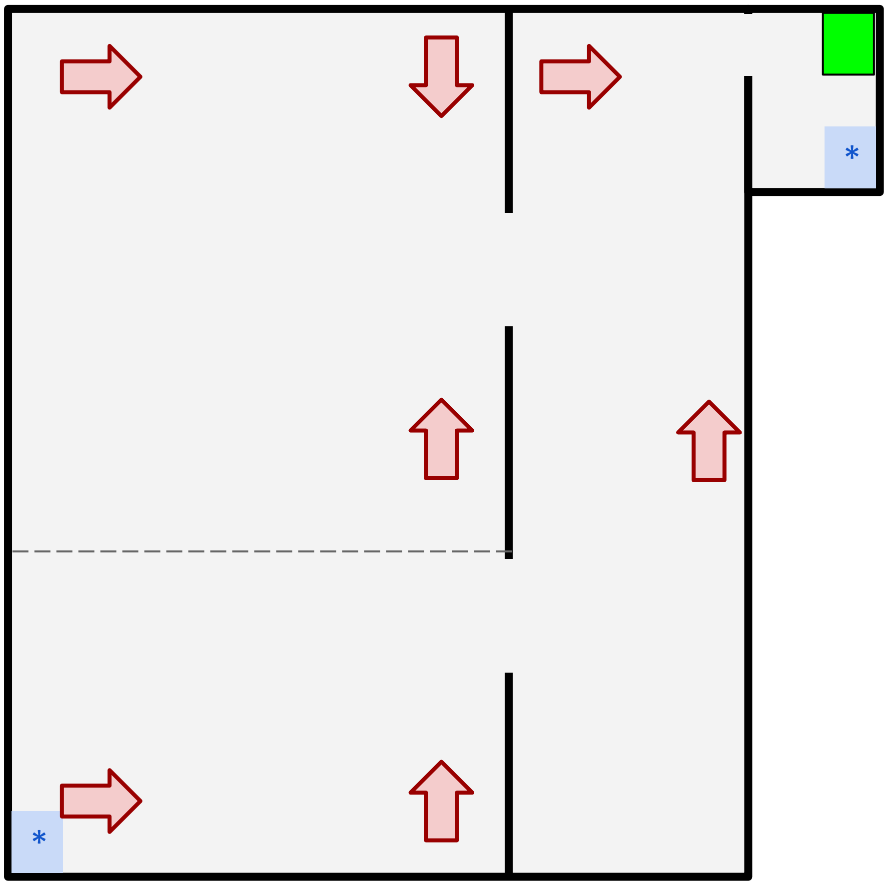
Recursive Optimality



Hierarchy: Room A -> Room B -> Room C

- Recursive
  - Component-wise optimal policies
- Hierarchical
- Flat

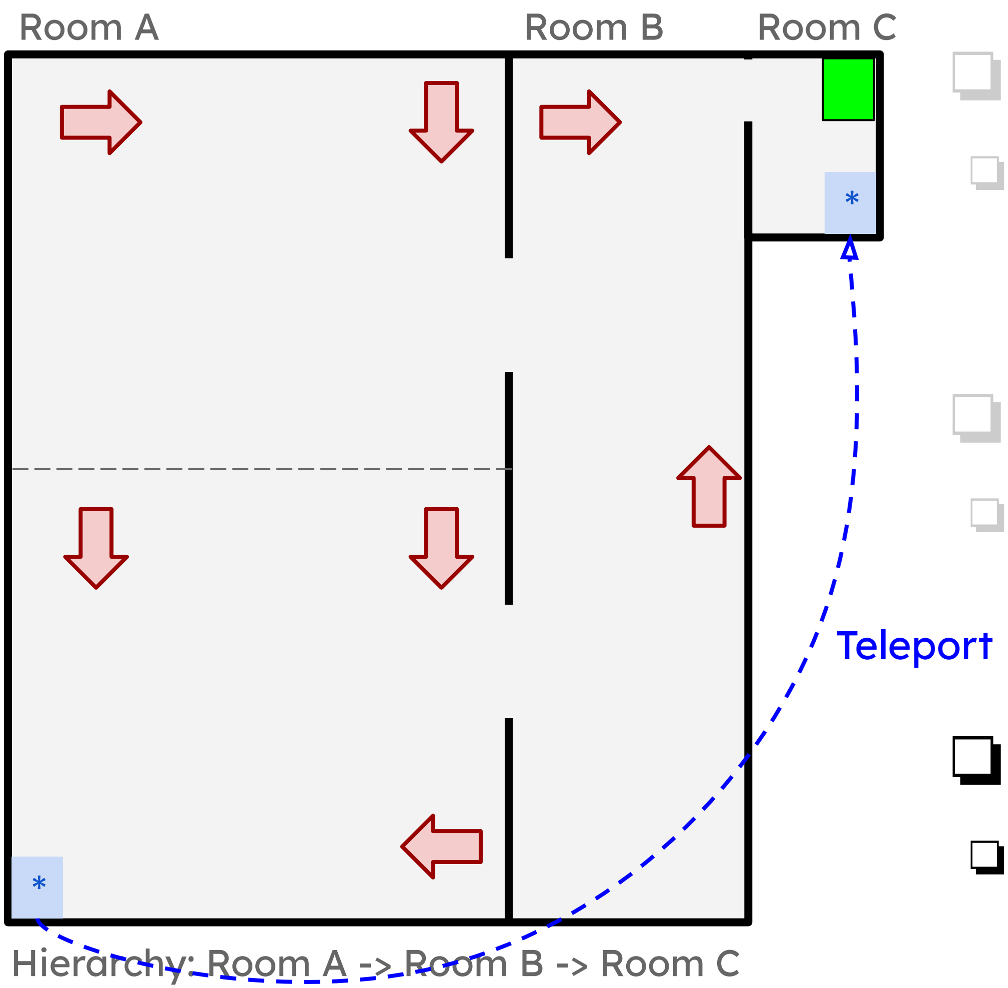
Hierarchical Optimality



Hierarchy: Room A -> Room B -> Room C

- Recursive
  - Component-wise optimal policies
- Hierarchical
  - Best policy consistent with given hierarchy
- Flat

Flat Optimality



- Recursive
  - Component-wise optimal policies
- Hierarchical
  - Best policy consistent with given hierarchy
- Flat
  - The optimal policy (does not consider any hierarchy)