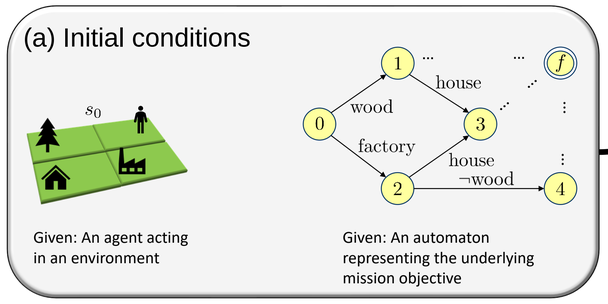
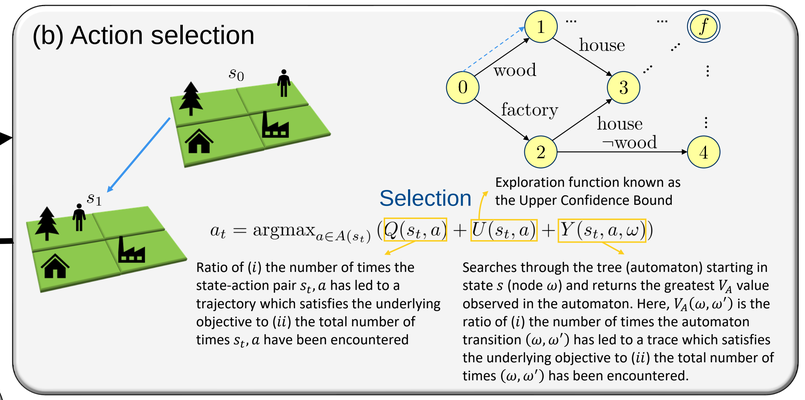
Multi-Agent Tree Search with Dynamic Reward Shaping

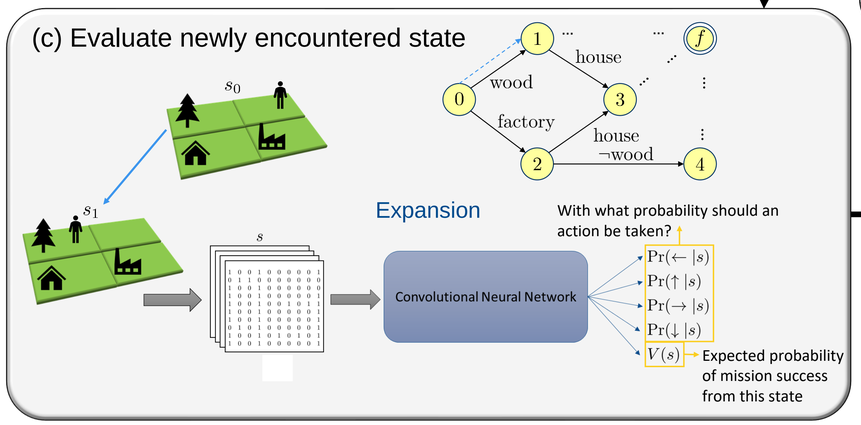
Alvaro Velasquez, AFRL
alvaro.velasquez.1@us.af.mil

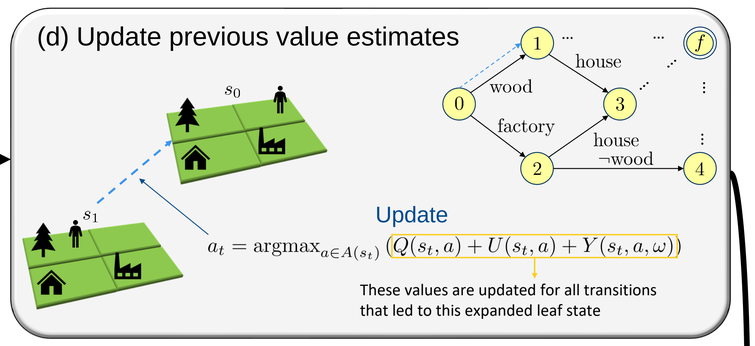
Introduction

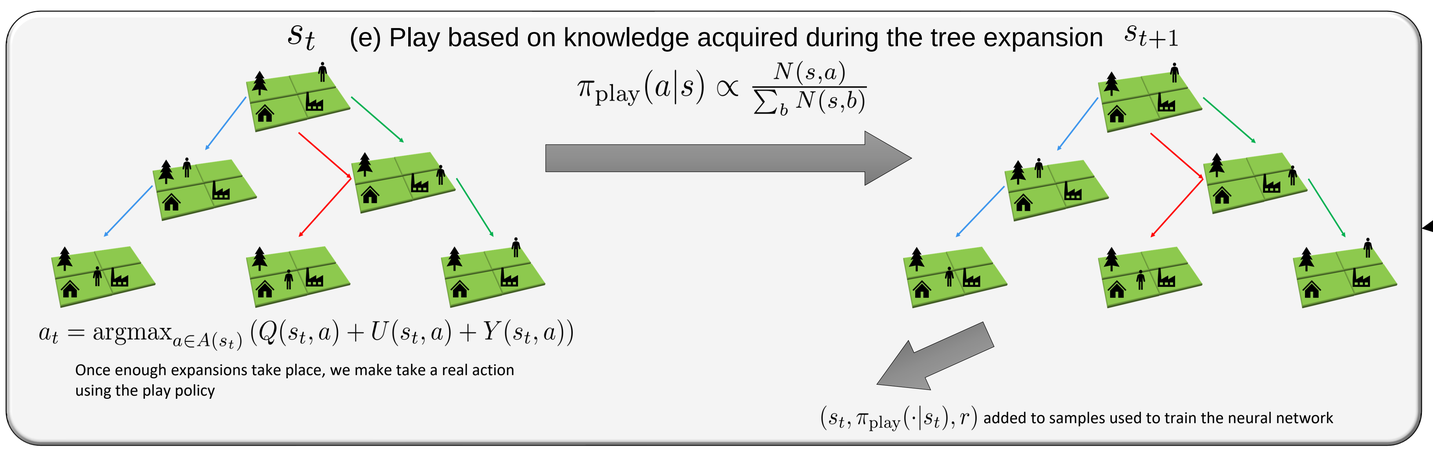
In recent years, powerful Monte Carlo Tree Search (MCTS) variants using deep Convolutional Neural Networks (CNNs) have been proposed for the game of Go and other board games. However, these modern MCTS implementations are not yet capable of handling arbitrary objectives with sparse reward signals. We seek to mitigate this by collecting statistics over the objective representation (i.e. the automaton which encodes the objective) as opposed to the implicit statistics collected in traditional reinforcement learning approaches over a state-action representation. These statistics capture the utility of individual transitions in the automaton that were particularly conducive to accomplishing the underlying objective. This is useful as a complement to existing MCTS approaches by reasoning over both the representation of agent-environment dynamics through deep convolutional neural networks as well as the representation of the underlying objective via automata. We call this Automaton-Guided Tree Search (AGTS) and argue that this is useful due to the low dimensionality of the automaton that represents the objective. This means that individual transitions within the automaton correspond to many transitions within the Monte Carlo tree. Below is the high-level idea for a single player. For the multi-agent setting, agents in each team take turns in the tree.



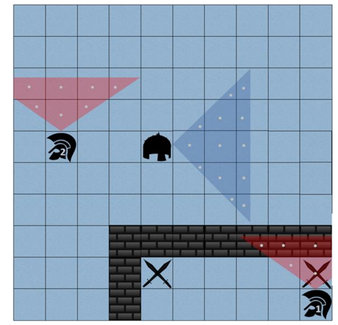




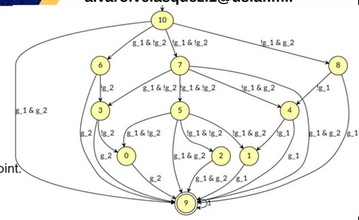




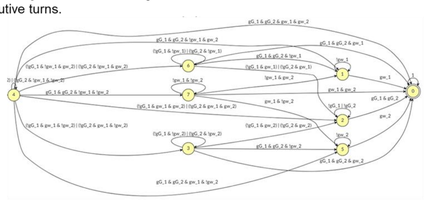
Gladiators and Goliath Domain



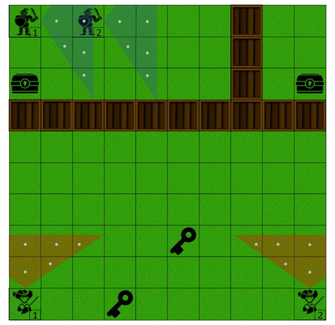
Gladiators' objective: Each obtain a weapon and both gladiators must have the goliath in their sights simultaneously at some point.



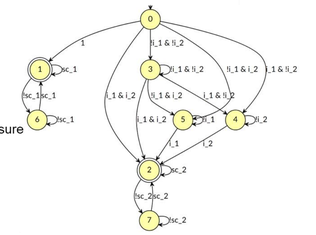
Goliath's objective: have each gladiator in its cone of vision for three consecutive turns.



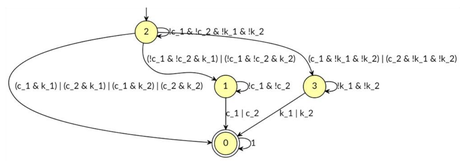
Infiltrators and Guards Domain



Infiltrators' objective: Obtain a key in and open the treasure chest.



Guards' objective: Find both infiltrators and monitor the treasure to ensure it has not been stolen.

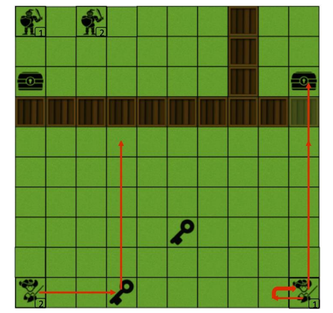


Results

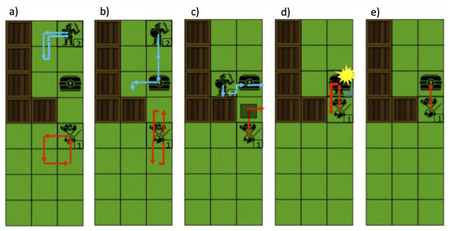
- **Deterministic:** DFA infiltrators (guards) achieved 69.49% (61.94%) higher rewards than their non-DFA counterparts.
- **Deterministic:** DFA gladiators (goliath) scoring 63.73% (13.34%) higher than their non-DFA counterparts.
- **Stochastic:** DFA infiltrators (guards) achieve 71.80% (154.07%) higher rewards than their non-DFA counterparts.
- **Stochastic:** DFA gladiators (goliath) score 41.11% (23.98%) higher than non-DFA counterparts.
- **Efficiency:** in the infiltrators and guards environment, the deterministic (stochastic) DFA games have an average episode length that is 80.33% (42.64%) less than that of non-DFA counterparts.
- **Efficiency:** In the gladiators and goliath environment, we similarly observe deterministic (stochastic) DFA games have an average episode length that is 16.28% (34.53%) less than that of their non-DFA counterparts.

Emergent Behavior

Cooperative



Adversarial



Data-Efficient!

Episode length (Automaton): 53
Episode length (No Automaton): 332