Planning

CS395 GAI

Spring 2005

Overview

- Limitations of Finite State Machines
- STRIPS-style planning
- Hierarchical Task Networks

Some Limitations of Finite State Machines for Planning

- Difficult to perform sequences of actions
- Plans become hard-coded in the execution environment
  - How to extend to achieve new goals in an expansion pack?
- Pursuing multiple goals simultaneously can lead to an explosion in the complexity of an FSN
- Coordination between agents requires explicit encodings

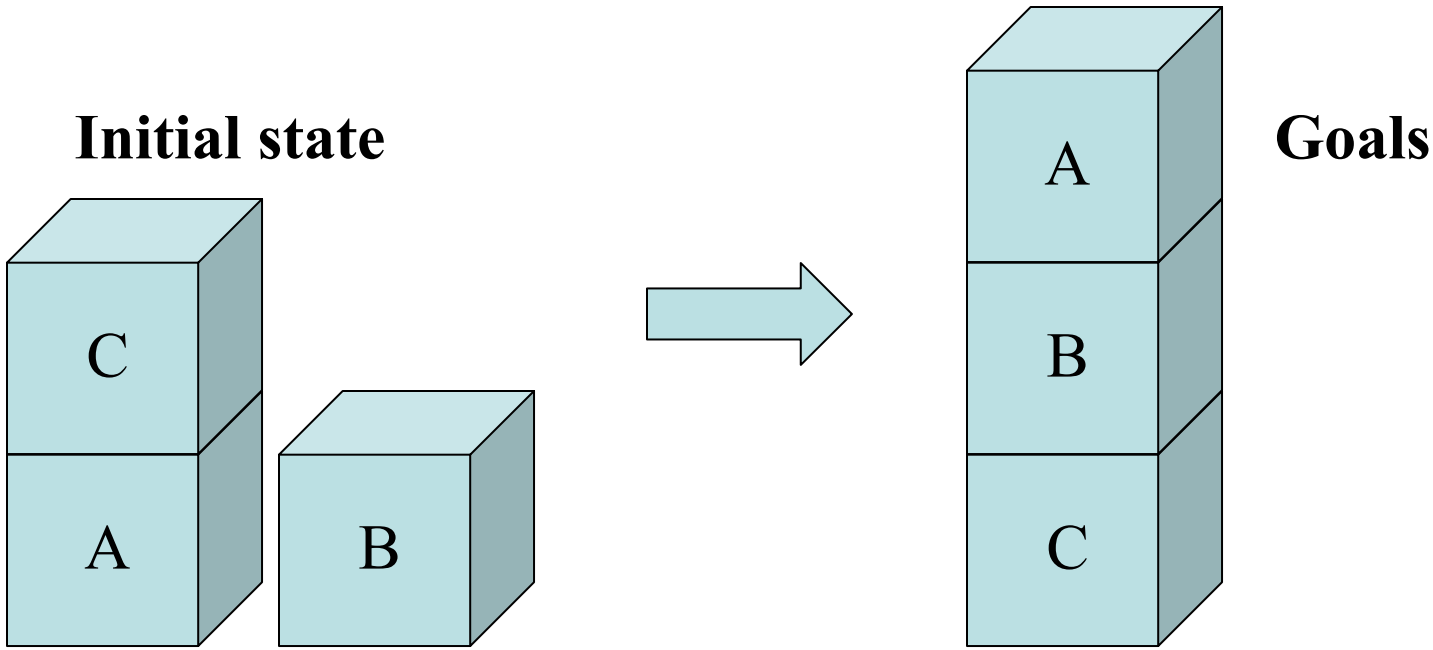
STRIPS-style planning

- STanford Research Institute Problem Solver
- Turns planning into a traditional search problem
- Core assumptions (closed-world)
  - Actions always succeed
  - Only changes that takes place are those indicated by the operators

STRIPS Components

- States
- Goals
- Operators
  - Preconditions
  - Effects
    - Add
    - Delete

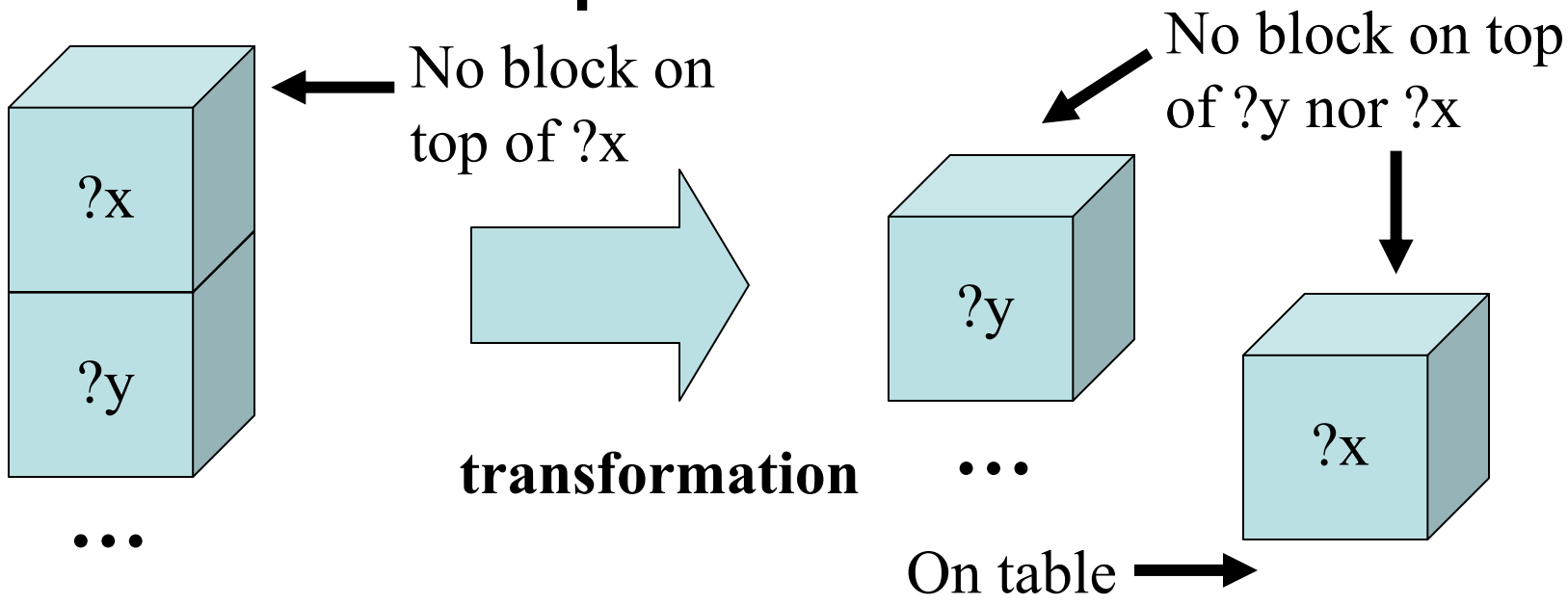
General-Purpose Planning: State & Goals



- **Initial state**: (on A Table) (on C A) (on B Table) (clear B) (clear C)
- **Goals**: (on C Table) (on B C) (on A B) (clear A)

(Ke Xu)

General-Purpose Planning: Operators

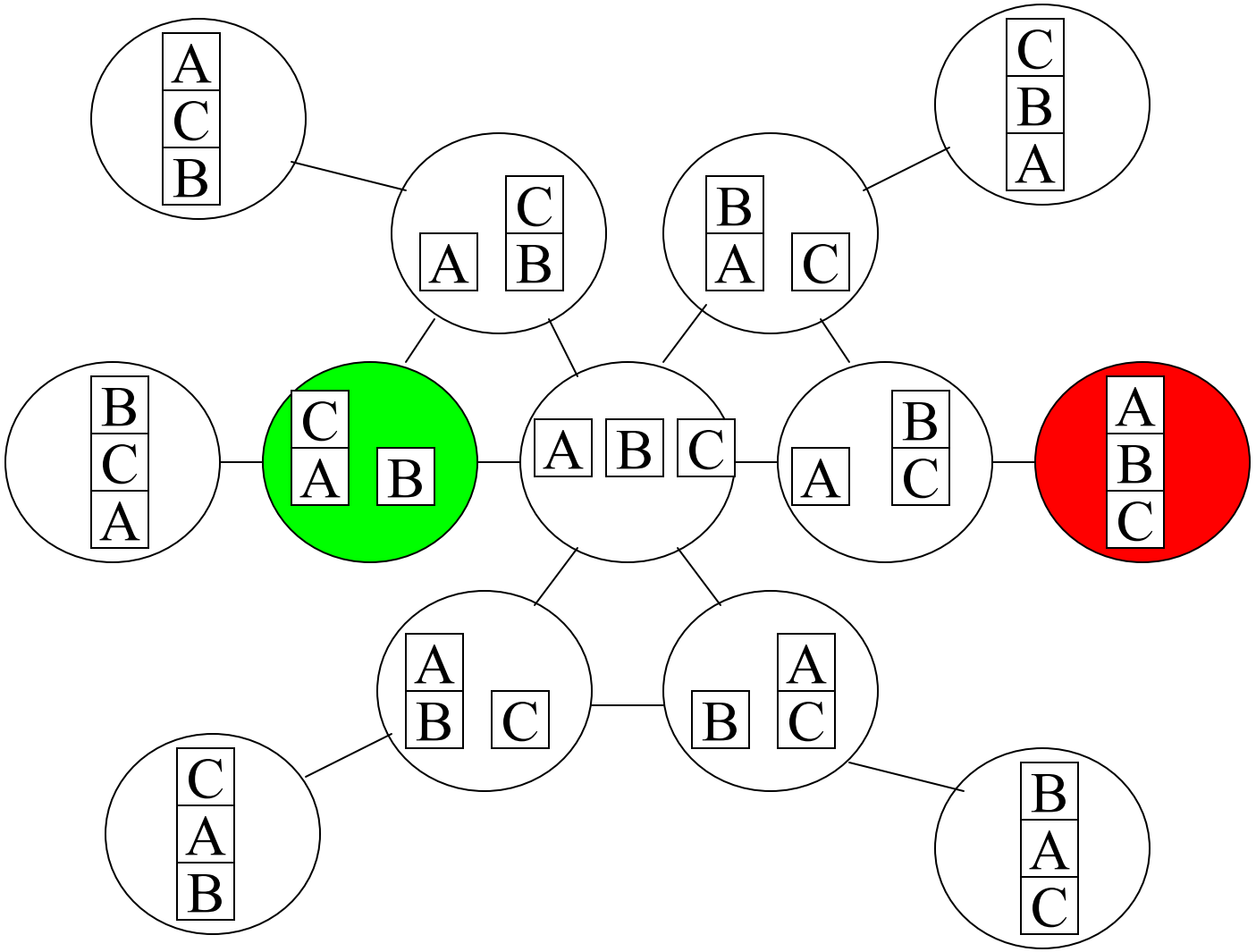


Operator: (Unstack ?x)

- **Preconditions**: (on ?x ?y) (clear ?x)
- **Effects**:
  - **Add**: (on ?x table) (clear ?y)
  - **Delete**: (on ?x ?y)

(Ke Xu)

Planning: Search Space



(Michael Moll)

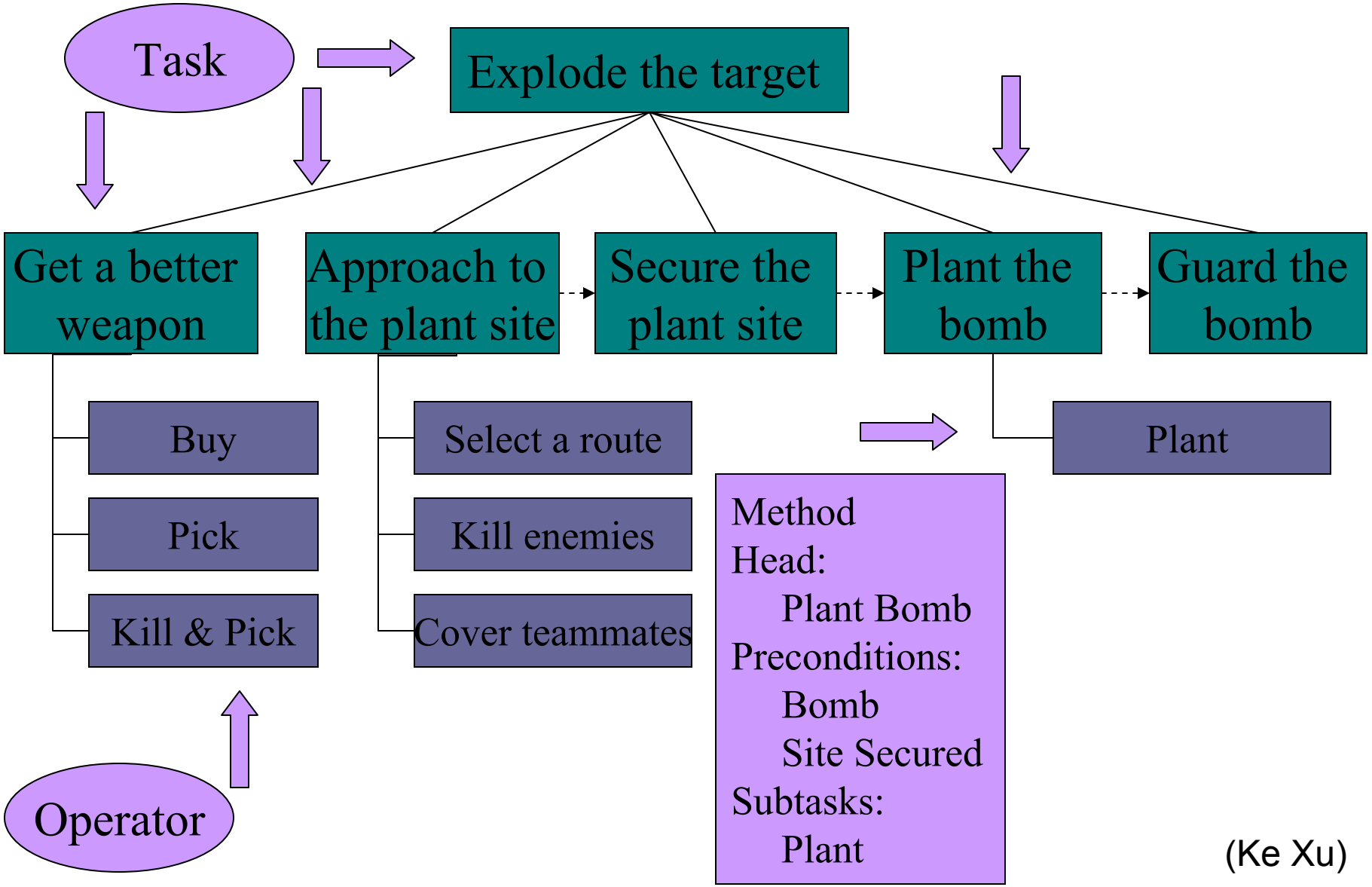
Hierarchical Task Networks

- Decomposition of higher-level tasks or strategies into lower-level components
- Components of a HTN
  - Tasks
  - Methods
  - Operators
  - Critics
- More robust with imperfect information
- Support for partial re-planning

Components of a HTN

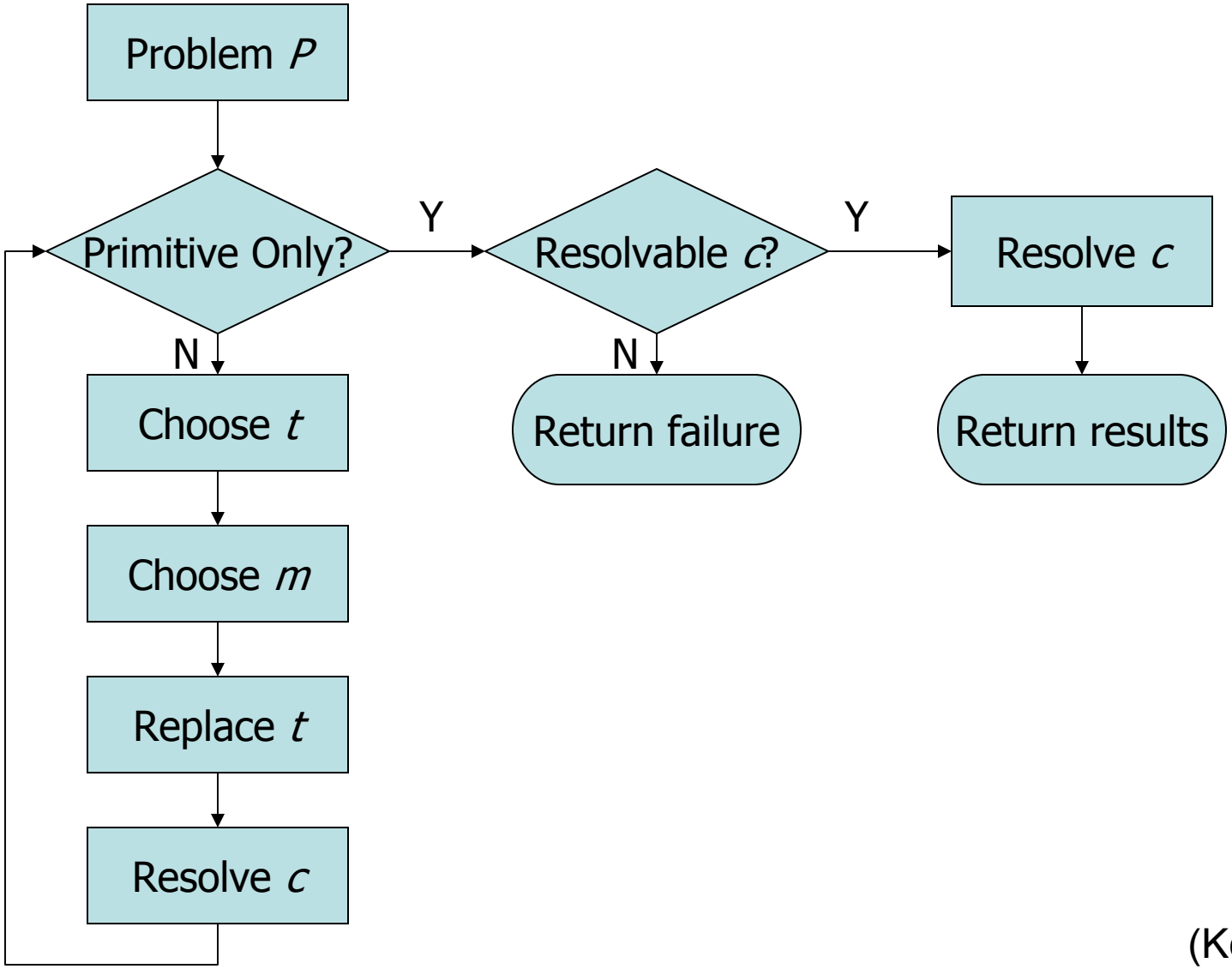
- Tasks
  - Non-primitive (compound tasks)
  - Primitive (actions)
- Methods
  - Expand or reduce non-primitive tasks
  - Defines preconditions that must be met for expansion to occur
- Operators
  - Effects only
  - Unlike STRIPS-style operators, have no preconditions
- Critics
  - De-conflict choices
    - Which of these should I try first?
  - Small bits of heuristic knowledge

HTN Example



(Ke Xu)

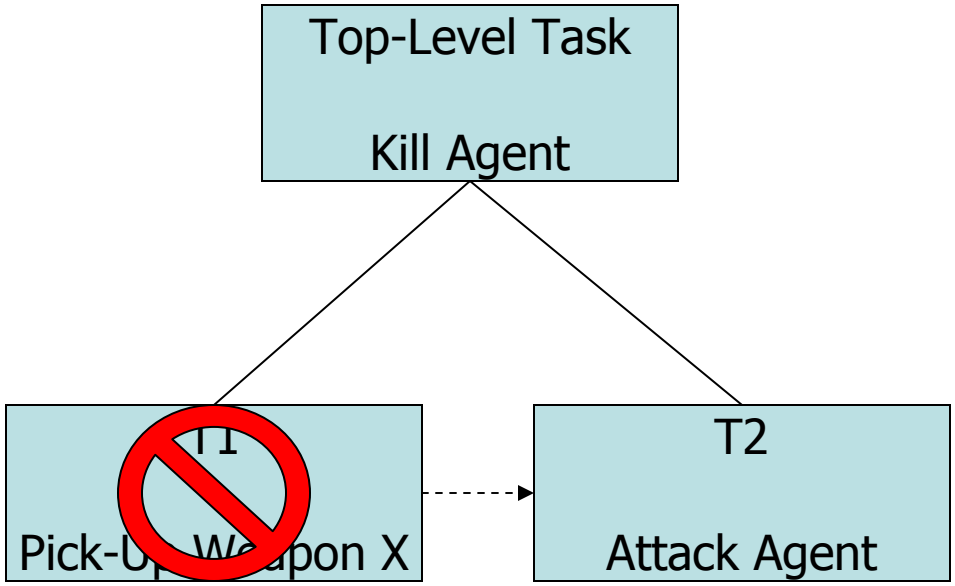
The HTN Planning Procedure



(Ke Xu)

Re-planning in HTN

- Partial Re-planning
  - Anytime
  - Anywhere
  - Repair rather than re-plan from sketch
  - Propagate the effect of re-planning
- Example



(Ke Xu)

Case Study: Computer Bridge

- Closed-world domain
  - But much more complex than chess

Case Study: Computer Bridge

- Chess: better than all but the best humans
- Bridge: worse than many good players

- Why bridge is difficult for computers
  - It is an imperfect information game
  - Don't know what cards the others have (except the dummy)
  - Many possible card distributions, so many possible moves
- If we encode the additional moves as additional branches in the game tree, this increases the number of nodes exponentially
  - worst case: about 6×10^{44} leaf nodes
  - average case: about 10^{24} leaf nodes

Not enough time to search the game tree

(Dana S. Nau)

Case Study: Computer Bridge

- Bridge is a game of planning
  - Declarer plans how to play the hand
    by combining various strategies (ruffing, finessing, etc.)
  - If a move doesn't fit into a sensible strategy,
    then it probably doesn't need to be considered
- HTN approach for declarer play
  - Use HTN planning to generate a game tree in which each move corresponds to a different *strategy*, not a different *card*
    - Reduces average game-tree size to about 26,000 leaf nodes
- Bridge Baron: implements HTN planning
  - Won the 1997 World Bridge Computer Challenge
  - All commercial versions of Bridge Baron since 1997 have include an HTN planner (has sold many thousands of copies)

(Dana S. Nau)

Case Study: Unreal Bots

- Effectively an open-world domain
  - Virtually impossible to represent all possible states of a multiplayer Unreal game

Case Study: Unreal Bots

- Method

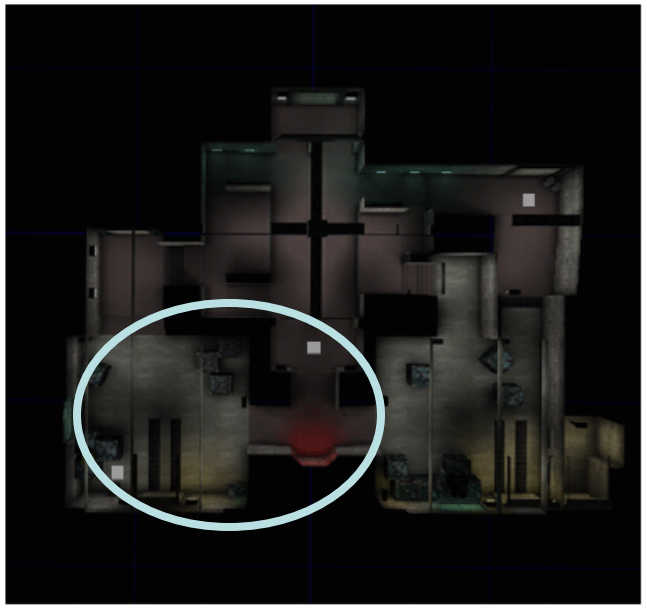
Method
Head: Domination(X)
Preconditions:
1. numberPlayersTeam(Nteam),
2. numberLocations(X,N),
3. Nteam > N/2 + 2
4. SelectLocsGeographTogether(X,P,N/2+1)
5. Divide3Groups(N/2+1,T1,T2,T3),
6. RemainingLocations(RP,X,P)

Subtasks:
1. CoverLocations(T1,P)
2. PatrolLocations (T2,P)
3. HarrassLocations(T3,RP)

Orderings:
none

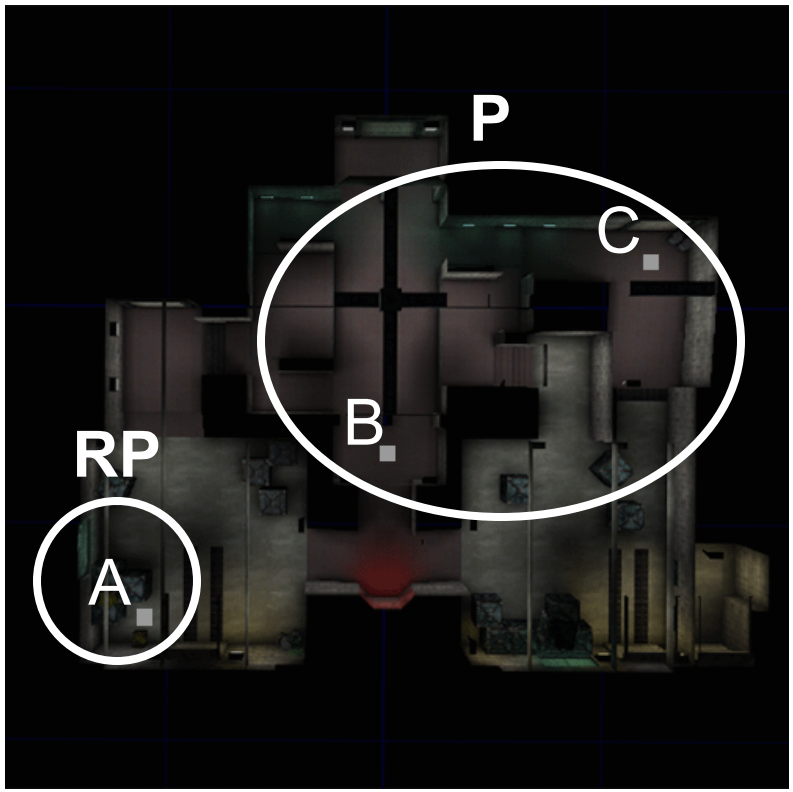
- Operator

Operator
Head: CoverLocation(B,L)
Effects:
Move(B,L)
Defend(B,L)



(Ke Xu, Héctor Muñoz-Avila)

Case Study: Unreal Bots



HTN Method

Head: Domination

Preconditions:

LocsGeographTogether(P,RP)

Divide3Groups(T1,T2,T3)

Subtasks:

CoverLocations(T1,P)

PatrolLocations (T2,P)

HarrassLocations(T3,RP)

(Ke Xu, Héctor Muñoz-Avila)

Case Study: Unreal Bots

- Coordinated Actions
  - Coordination is represented in the hierarchy, but not in the operators
- Dealing with changing conditions
  - Pre-defined thresholds for strategies used to trigger partial re-planning