Integrated Planning and Acting: The Actor's View

Dana Nau
Dept. of Computer Science, and Institute for Systems Research
University of Maryland

D. Nau. Integrated planning and acting: The actor's view. In *Second Annual Conference on Advances in Cognitive Systems*, Dec. 2013. Invited talk.

Invitation from Paul Bello

… We were especially impressed with your recent work on the Godel planning system, and would welcome a talk focused on those particular issues. …

What's a Godel? What's a planning system?

Shivashankar, Alford, Kuter, & Nau. The Godel planning system: A more perfect union of domain-independent and hierarchical planning. *IJCAI*, 2013.

Invitation from Paul Bello

… We were especially impressed with your recent work on the Godel planning system, and would welcome a talk focused on those particular issues. …

Shivashankar, Alford, Kuter, & Nau. The Godel planning system: A more perfect union of domain-independent and hierarchical planning. *IJCAI*, 2013.

What's a Godel? What's a planning system?

The audience won't understand Godel unless I explain the "big picture"

What is Planning?

plan *n.*

1. A scheme, program, or method worked out beforehand for the accomplishment of an objective: *a plan of attack.*
2. A proposed or tentative project or course of action: *had no plans for the evening.*

Focus of AI planning research

3. A systematic arrangement of elements or important parts; a configuration or outline: *a seating plan; the plan of a story.*
4. A drawing or diagram made to scale showing the structure or arrangement of something.
5. A program or policy stipulating a service or benefit: *a pension plan.*

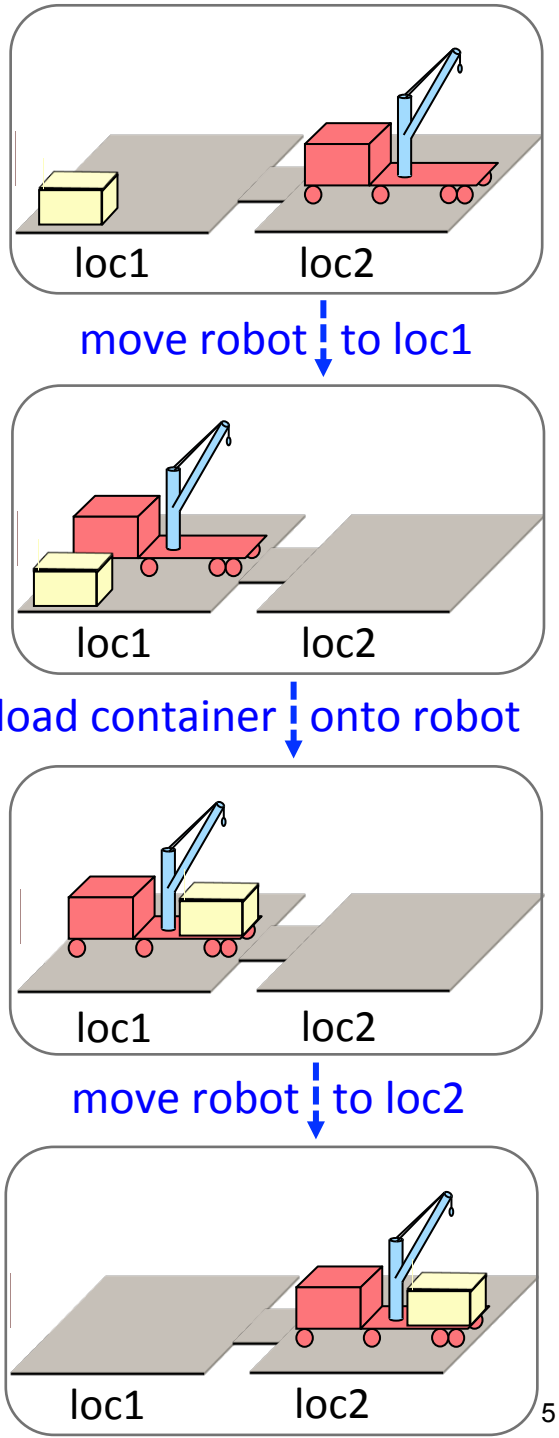
planning *n.* The process of making plans for something.

the reasoning side of acting.

– Ghallab, Nau, & Traverso. *Automated Planning: Theory and Practice.* Morgan Kaufmann, 2004.

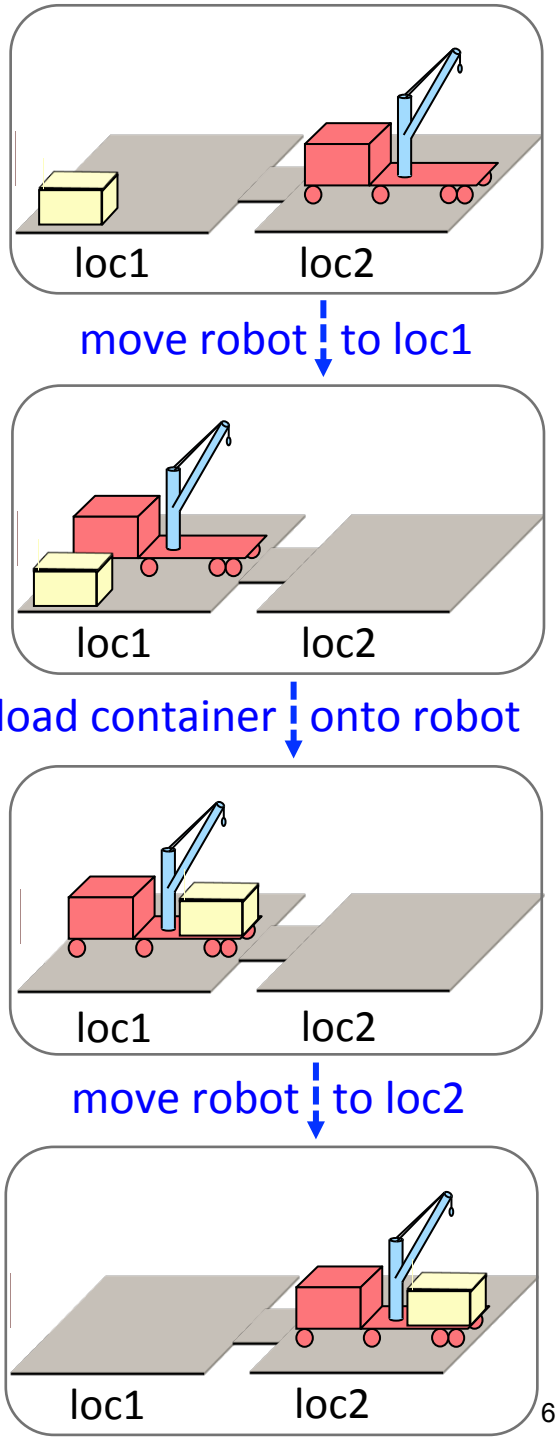
AI Planning Research

- Focus of most (not all) AI planning research:
  - » Finite world
    - finitely many states, finitely many actions
  - » Static world
    - the plan executor is the only source of change
  - » Implicit time:
    - Sequence of actions → sequence of instants
  - » Planning problem
    - Initial state, set of goal conditions
  - » Solution
    - sequence of actions or set of state-action pairs
    - takes world from initial state to a goal state
  - » Offline planning
    - generate entire solution before performing it



Capabilities and Limitations

- **Huge** advances in planning in state-transition systems
  - » Standard language (PDDL) for specifying states, actions and state-transition systems
  - » Mature technology for finding plans in huge state spaces
- But less practical impact than one might hope
  - » Why?
- Consider some of AI planning's successes
  - » What the environment is like
  - » How the planning is done



Example Planning Applications

- Robotics
  - » Mars Rovers
  - » Specialized manufacturing applications
- Management of services and operations
  - » “City in your pocket” app, Trento, Italy
- Computer games
  - » Bridge Baron
  - » Killzone 2
- Very different environments, but several common characteristics

Environment

- World is dynamic
  - » Exogenous events that aren't under the actor's control
  - » Multiple agents, some may be human
- World is not fully known
  - » Don't always know in advance when exogenous events will happen
  - » Don't always know in advance what all the possible events are
  - » Models of actions and states may not be fully accurate
- Usually not feasible to produce the final complete plan beforehand
  - » Restrictions on how much time is available for planning
  - » Planning for all possible contingencies may take too much time
  - » We might not even know what the possible contingencies are

Planning in a Dynamic and Open World

- Planning is continual and online
  - » Monitor, refine, extend, update, change, and repair plans **throughout the acting process**
  - » Generate activities dynamically at run-time
    - to carry out other higher-level activities that one is currently performing
    - to respond to unexpected developments
  - » Plans remain *partial* as long as the cost of possible mistakes is lower than the cost of modeling, information gathering, thorough planning

Planning *in* Acting

- Planner is part of a larger system: **the actor**
- **Deliberation is organized hierarchically**
  - » View, prepare, and carry out actions hierarchically
  - » An action may be a task that needs further refinement and planning

- This goes beyond HTN planning
  - » Refinement usually needs to be done online
  - » High-level state and action models are abstract approximations
  - » Refining may require different representations, tools, and techniques at other levels of the hierarchy

Comparison

	typical AI planning	typical applications
Planner's role	stand-alone system	part of larger system
Objective	complete solution	partial solution
Planning	in advance, offline	continual, online
World	static	dynamic
Agents	one (plan executor)	possibly many
Computing time	unlimited	constrained
Replanning	call planner again	often needed quickly
Plan	flat	hierarchical
States	sets of propositions	heterogeneous
Actions	descriptive	heterogeneous
Domain model	states & actions	application-specific
Goals	given in advance	depends on situation

Huh?

- Most planning research has ignored many of those requirements
  - » How have the existing successes been accomplished?

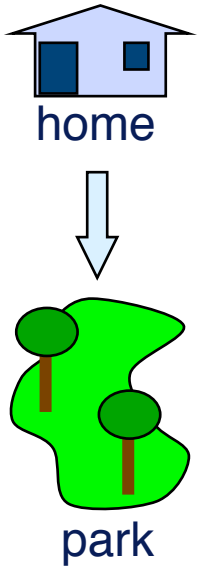
- **Approximate** some part of the overall problem as a planning problem
  - » Develop a **special-purpose** planner for that problem
  - » Use the planner are part of a larger system

- I'll discuss some examples that involve Hierarchical Task Network (HTN) planning
  - » First, I'll explain what HTN planning is

HTN Planning

- For some planning problems
  - » The available knowledge may be about tasks rather than states
  - » The natural problem statement may be not a goal, but a task to perform
  - » We may already have some ideas for how to perform the task
- Example: travel to a destination *D* that's far away:
  - » Brute-force search for a state in which one is at *D*:
    - Many ways to combine vehicles and routes
  - » Experienced human: small number of "recipes" for carrying out various tasks
    - e.g., flying:
      1. buy ticket from local airport to remote airport
      2. travel to local airport
      3. fly to remote airport
      4. travel to final destination
  - » HTN planners use such recipes to generate the search space
- Ingredients
  - » states, tasks, operators, methods, planning algorithm

States and Tasks



- **State**: description of the current situation
  - » I'm at home, I have $20, there's a park 8 miles away
- **Task**: description of an activity to perform
  - » Travel to the park
- Two kinds of tasks
  - » ***Primitive*** task: a task that corresponds to a basic action
  - » ***Compound*** task: a task that is composed of other simpler tasks

Operators

- **Operators**: parameterized descriptions of what the basic actions do
  - ***walk*** from location x to location y
    - Precond: agent is at x
    - Effects: agent is at y
  - ***call taxi*** to location x
    - Precond: (none)
    - Effects: taxi is at x
  - ***ride taxi*** from location x to location y
    - Precond: agent and taxi are at x
    - Effects: agent and taxi at y, agent owes $1.50 + \frac{1}{2}$ distance(x,y)
  - ***pay driver***
    - Precond: agent owes amount of money r, agent has money $m \geq r$
    - Effects: agent owes nothing, agent has money $m - r$
- **Actions**: operators with arguments

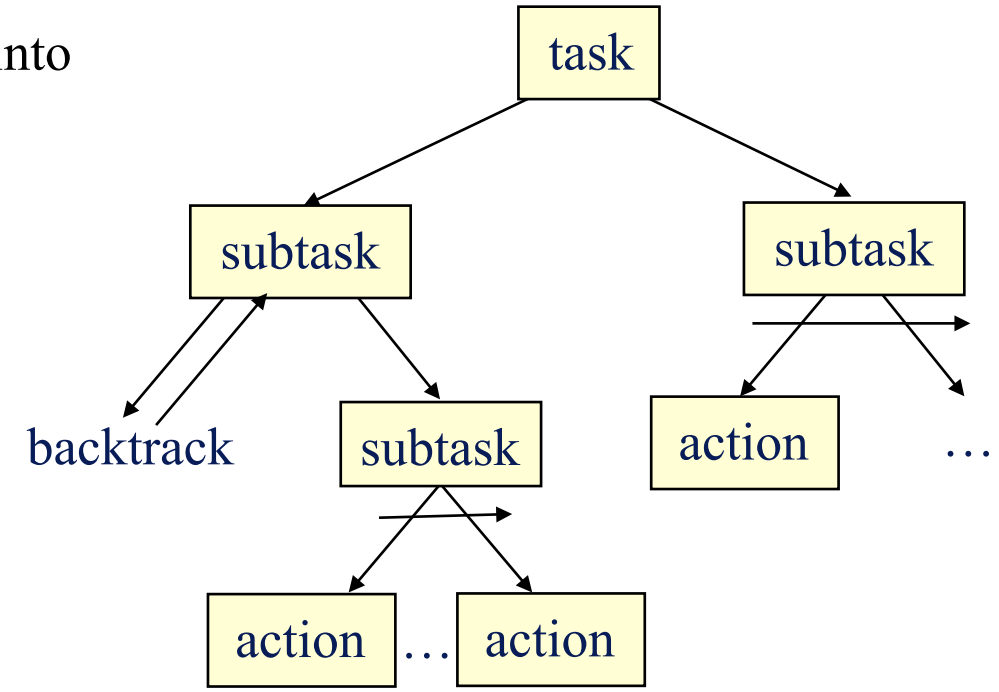
Methods

- Method: parameterized description of a possible way to perform a compound task by performing a collection of subtasks
- There may be more than one method for the same task

  - ***travel by foot*** from *x* to *y*
    - Task: travel from *x* to *y*
    - Precond: agent is at *x*, distance to *y* is ≤ 2 miles
    - Subtasks: walk from *x* to *y*
  - ***travel by taxi*** from *x* to *y*
    - Task: travel from *x* to *y*
    - Precond: agent is at *x*, agent has money ≥ 1.5 + ½ distance(*x*,*y*)
    - Subtasks: call taxi to *x*,
      ride taxi from *x* to *y*,
      pay driver

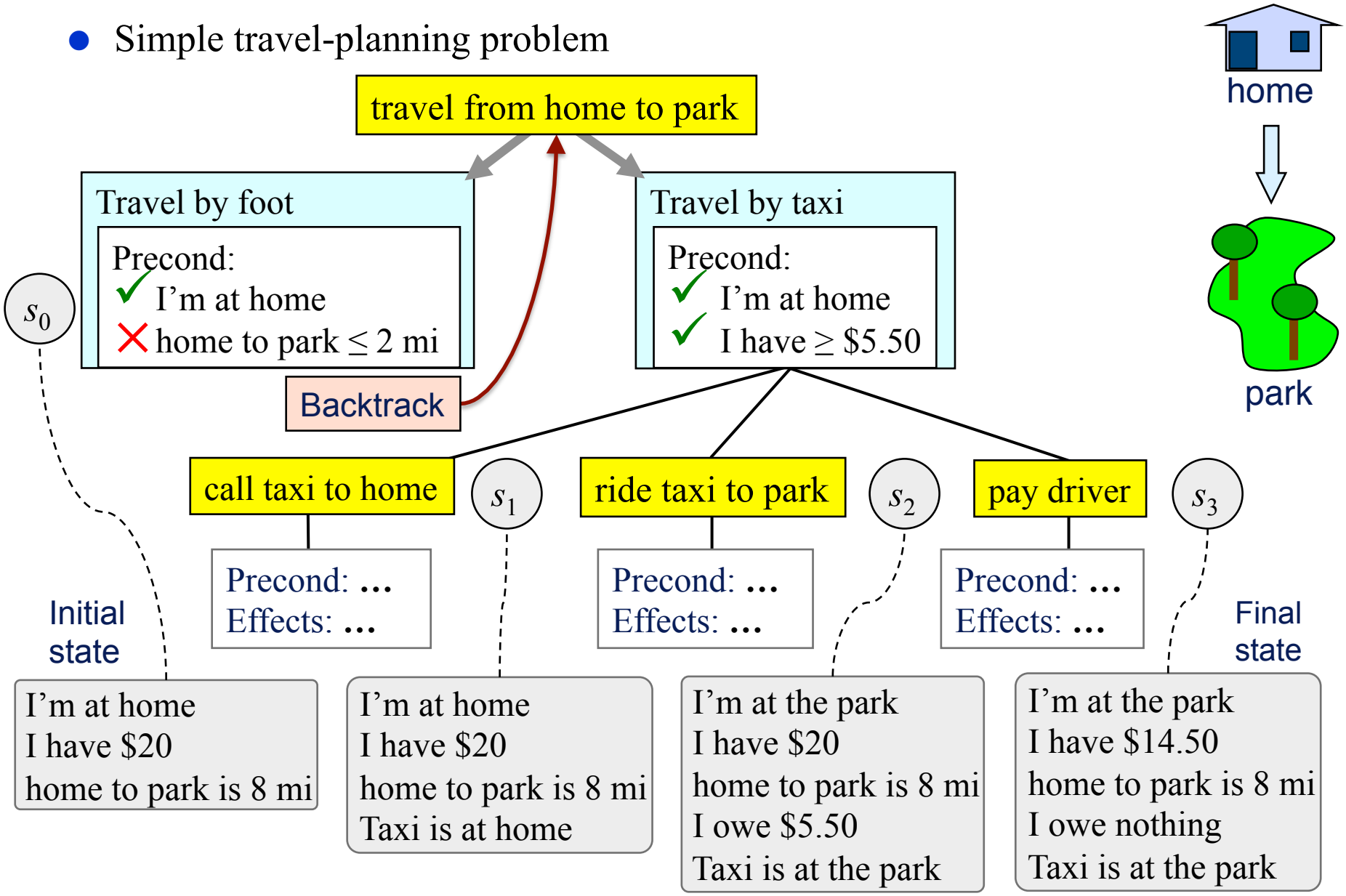
Planning Algorithm

- Planning problem:
  - » Initial state
  - » Initial task or sequence of tasks
- Solution:
  - » Recursively decompose tasks into subtasks until every leaf node contains an action
  - » If the sequence of actions is executable then it is a solution
- Planning algorithm
  - » Left-to-right backtracking search



Example

- Simple travel-planning problem



Comparison

	typical AI planning	HTN planning	typical applications
Planner's role	stand-alone system	depends on application	part of larger system
Objective	complete solution	complete solution	partial solution
Planning	in advance, offline	usu. offline but not always	continual, online
World	static	usu. static but not always	dynamic
Agents	one (plan executor)	usu. one but not always	possibly many
Computing time	unlimited	unlimited but faster	constrained
Replanning	call planner again	call planner again	often needed quickly
Plan	flat	hierarchical	hierarchical
States	sets of propositions	propositions or data struct.	heterogeneous
Actions	descriptive	descriptive	heterogeneous
Domain model	states & actions	states, actions, methods	application-specific
Goals	given in advance	tasks	depends on situation

Implementations

- SHOP and SHOP2
  - » Written around 2000; Lisp and Java versions available
  - » SHOP2 received an award in the AIPS-2002 Planning Competition
  - » Open source: http://www.cs.umd.edu/projects/shop
  - » Used in hundreds of projects worldwide

- Pyhop
  - » Written in Python, released summer 2013
  - » Objective: something easy to integrate into ordinary computer programs
  - » Easy to understand – less than 150 lines of code
  - » Representation of states, actions, methods are closer to what you'd use in ordinary computer programming
    - Manipulate computer variables rather than logical predicates
  - » Open-source: http://bitbucket.org/dananau/pyhop

Bridge Baron

- Won the 1997 world championship of computer bridge
- Used a special-purpose HTN planner that generated game trees
  - » Only generated branches corresponding to known bridge strategies
    - finesse, ruff, cash out, …
    - About 10^5 leaf nodes
    - A conventional game tree would have had about 10^{24} leaf nodes
  - » Less time needed to search the tree and compute expected utility values

- **Why it worked**:
  - » Special-purpose HTN planner
    - multiple agents
    - generate game trees, not linear plans
  - » Lots of human effort to make the HTN methods as complete as possible
  - » Can run it repeatedly during the game

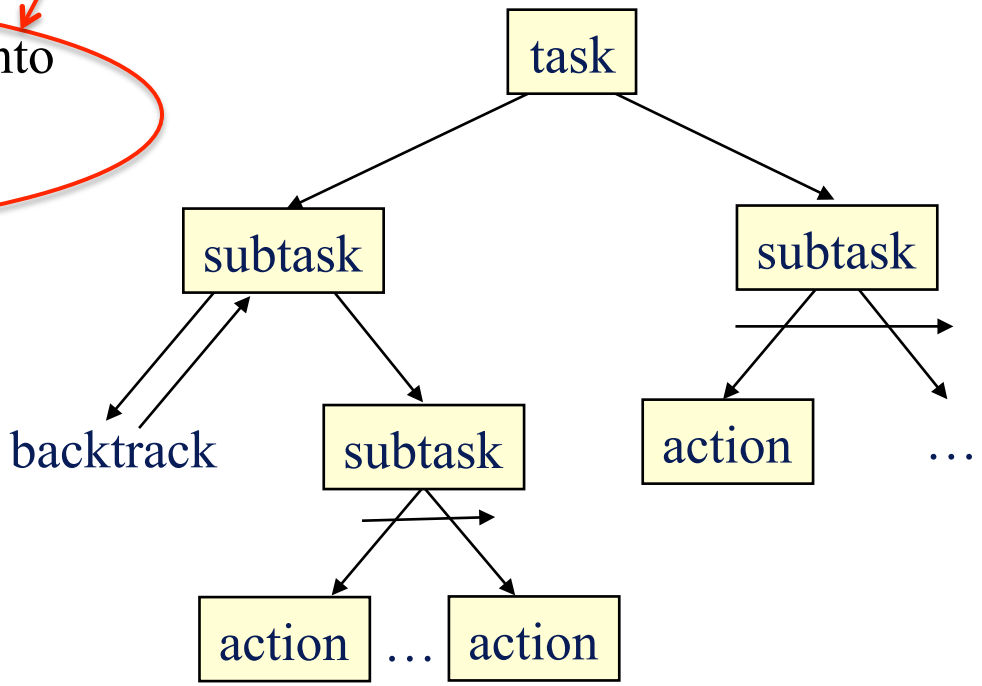
KILLZONE 2

- Developed by Guerrilla Games, released in early 2009
- Incorporates a special-purpose HTN planner for planning at the squad level
  - » Method and operator syntax similar to SHOP and SHOP2
  - » Quickly generates a linear plan that would work if nothing interferes
  - » Replan several times per second as the world changes
- **Why it worked:**
  - » Very different objective from a bridge tournament
  - » Don't ***want*** to look for the best possible play
  - » Need actions that appear believable and consistent to human users
  - » Need them very quickly

Planning

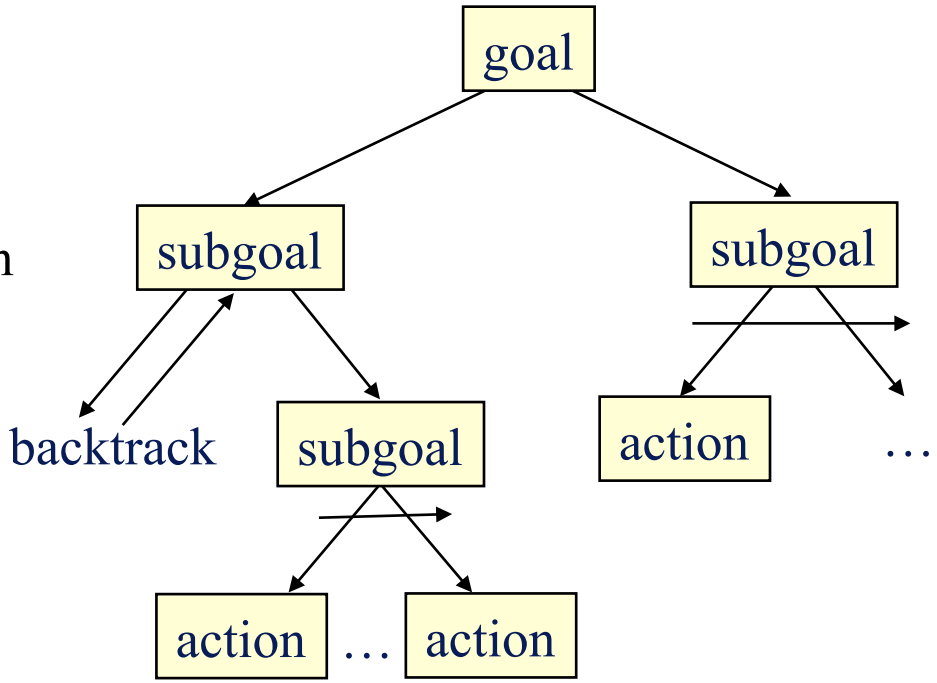
- Planning problem:
  - » Initial state
  - » Initial task or sequence of tasks
- Solution:
  - » Recursively decompose tasks into subtasks until every leaf node contains an action
  - » If the sequence of actions is executable then it is a solution
- Planning algorithm
  - » Left-to-right backtracking search

A human must write a complete set of HTN methods



Godel

- Instead of decomposing tasks into subtasks, Godel's methods decompose goals into subgoals
- Planning problem:
  - » initial state and goal (as in classical planning)
- For each subgoal
  - » If a subgoal matches an action
    - then the subgoal is solved
  - » Otherwise if there's an applicable method
    - use it to produce subgoals
  - » Otherwise
    - generate **landmarks**
      - › Classical planning algorithm
    - use the landmarks as subgoals
  - » Repeat until all subgoals are solved



Properties

- Properties of Godel
  - With a complete set of methods, Godel behaves like an HTN planner
  - With no methods, Godel behaves like a classical planner
  - With an **incomplete** set of methods, Godel uses a combination of HTN-style decomposition and classical planning
- This reduces the "method writing" bottleneck
  - Don't need to write all the methods
    - write methods for high-level strategy, let Godel fill in the detail
    - write methods for details, let Godel figure out the high-level strategy
- Potential foundation for integration with goal-reasoning algorithms
  - Quick replanning in response to unexpected events or new goals
  - Vikas Shivashankar will talk about this in the Goal Reasoning workshop this afternoon

Summary

- **Acting** is a key capability of integrated systems
  - » Neglected, needs more attention
- Some requirements
  - » Continual online planning during acting
  - » Hierarchical organized deliberation

- How AI planning algorithms have been used effectively
  - » Find a subproblem that approximates AI planning
  - » Special-purpose planner, plan multiple times as the world changes

- Hierarchical requirements go beyond HTN planning
  - » Heterogeneous representations
- Extensions of HTN planning
  - » Pyhop: state variable representation
  - » Godel: goal reasoning

References

- Planning:
  - Ghallab, Nau, & Traverso. *Automated Planning: Theory and Practice.* Morgan Kaufmann, 2004.
- Planning and acting:
  - Ghallab, Nau, and Traverso. The actor's view of automated planning and acting: a position paper. *Artificial Intelligence*, to appear.
- Pyhop:
  - D. Nau. Game applications of HTN planning with state variables. *ICAPS Workshop on Planning in Games*, 2013. Invited talk.
- Godel:
  - Shivashankar, Alford, Kuter, and Nau. The Godel planning system: A more perfect union of domain-independent and hierarchical planning. *IJCAI*, 2013.
  - Shivashankar, Alford, Kuter, and Nau. Hierarchical Goal Networks and Goal-Driven Autonomy: Going where AI Planning Meets Goal Reasoning. *ACS Workshop on Goal Reasoning*, **this afternoon**