COS 470/570: Artificial Intelligence

Planning

Spring 2019

What is Planning?

- Find a way to achieve the goal using *operators*
- Consider efficiency of planning and effectiveness and efficiency of execution
- Different from state space search
  - Do not need complete representations of states
  - Can look at more about goal than distance to goal: e.g., subgoals
  - Can handle *nearly-decomposable subproblems*

Example

- How to use state space search to get a robot to return a book to the library and buy milk?

Example

- How to use state space search to get a robot to return a book to the library and buy milk?
- How would *you* decide to return a book to the library and buy milk?

Theorem proving

Planning Using Theorem Proving

- Idea:
  - Create set of axioms describing plan creation
  - Ask theorem prover to prove there is a plan
  - Proof should be the plan
- Problem: Predicate calculus is *atemporal*

Situation Calculus

- Need to add temporal information to predicate calculus
- Problem: very difficult to do!
- Early solution [McCarthy & Hayes]: add a situation argument to the end of all predicates $\Rightarrow$ *situation calculus*
- E.g.:

  $inClass(Joe, S1)$ means "Joe is in class in situation 1"

Situation Calculus

- Situations: not linked to particular time
- A situation *labels* a state of the world occurring at some time
- Situation calculus: idea that actions transform one situation into another:

$$inClass(Joe, S1) \xRightarrow{go(Joe, Home, S1)} at(Joe, Home, S2)$$

- Actions are *functions* that yield new situations

Example: MAB

- Simple “monkey and bananas” (MAB) problem
- Initial state: $\neg OnBox(S_0) \land \neg HaveBananas(S_0)$
- Goal state: $HaveBananas(s)$
- Operators: (s' means state after s; $\rightsquigarrow$ indicates effects)

$$PushBox(x,s) \rightsquigarrow At(Box,x,s')$$
$$ClimbBox(s) \rightsquigarrow OnBox(s')$$
$$Grasp(s) \rightsquigarrow HaveBananas(s')$$

Example: MAB

- Axioms:
  1. If the monkey is not on the box, then in the state after which it pushes the box, the box will be where the monkey pushed it.

$$\forall x, s\ \neg OnBox(s) \rightarrow At(Box, x, PushBox(x, s))$$

$$\mathrm{CNF} : OnBox(s_1) \vee At(Box, x_1, PushBox(x_1, s_1))$$

  2. The monkey will be on the box in any state resulting from climbing on the box.

$$\forall s\ OnBox(ClimbBox(s))$$

$$\mathrm{CNF} : OnBox(ClimbBox(s_2))$$

Example: MAB

- Axioms (cont'd):
  3. If the monkey is on the box and the box is at the goal in a state, then in the state resulting from applying "grasp", the monkey will have the bananas.

$$\forall s\ OnBox(s) \wedge At(Box, Goal, s) \rightarrow HaveBananas(Grasp(s))$$

$$\mathrm{CNF}: \neg OnBox(s_3) \vee \neg At(Box, Goal, s_3) \vee HaveBananas(Grasp(s_3))$$

  4. (Frame axiom:) The box doesn't move when the monkey climbs on it.

$$\forall x, s\ At(Box, x, s) \rightarrow At(Box, x, ClimbBox(s))$$

$$\mathrm{CNF}: \neg At(Box, x_2, s_4) \vee At(Box, x_2, ClimbBox(s_4))$$

Example: MAB

- Axioms (cont'd):

5. Initial state:

$$\neg OnBox(S_0)$$

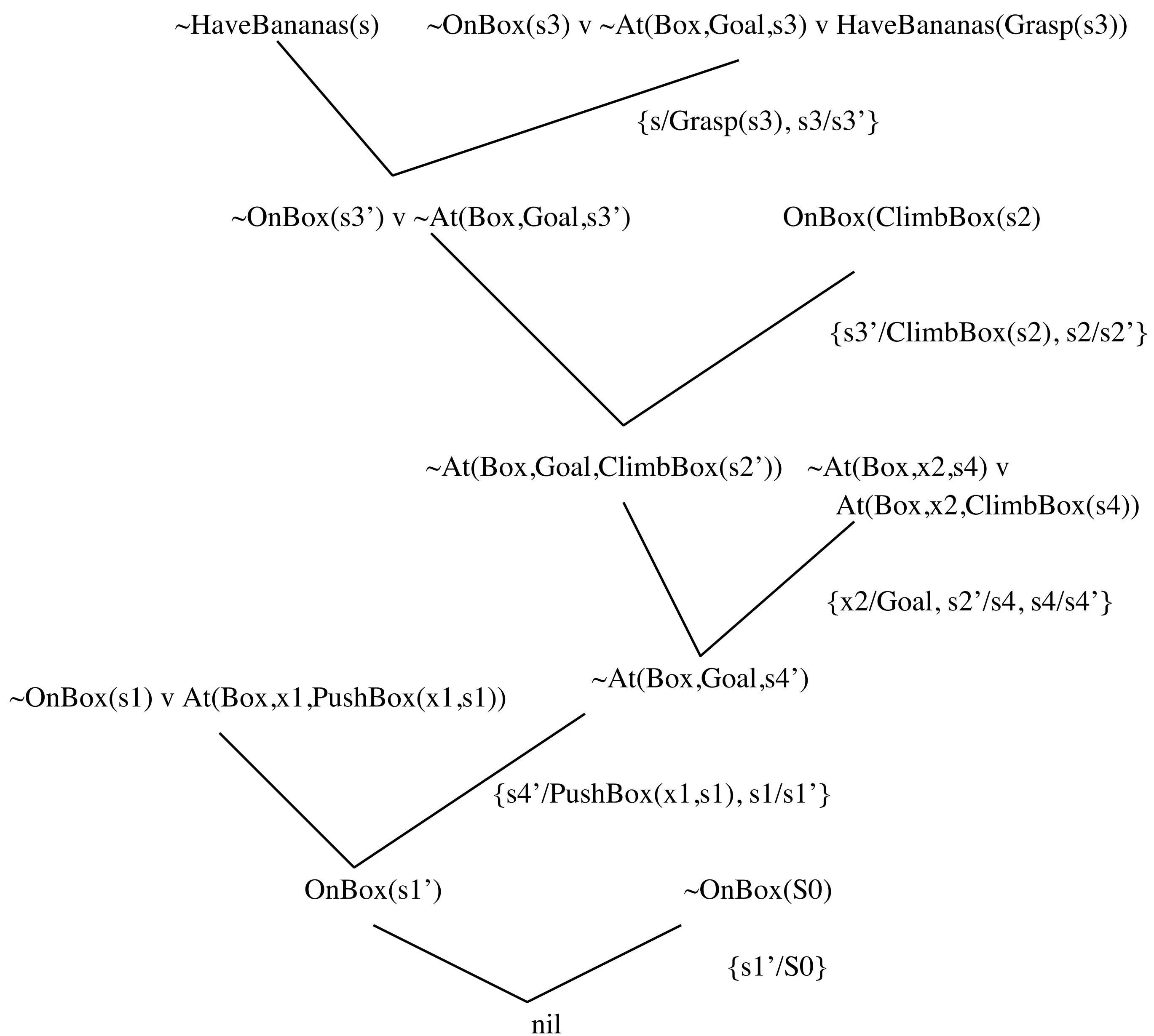
$$\mathrm{CNF} : \neg OnBox(S_0)$$

6. Thing to be proved:

$$\exists s\ HaveBananas(s)$$

$$\mathrm{CNF} : \neg HaveBananas(s)$$

MAB Example: Proof Tree



Finding the Plan

- Trace back up through the variables, find that the plan is:
  PushBox, ClimbBox, Grasp
- Use *Green's Trick*: Add extra literal to thing to be proven to automatically build plan from bindings:
  - E.g., $\neg HaveBananas(s) \vee Answer(s)$
  - Stop when the extra literal is all that's left, not when nil is reached
  - Answer will be:

$$Answer(Grasp(ClimbBox(PushBox(Goal, S_0))))$$

Artificial Intelligence

Means-Ends Analysis

Means-Ends Analysis

- How to choose operators?
- One of the best ways: *means-ends analysis* (MEA)
- MEA came from psychological protocol studies (Simon, Newell)
- Basic idea:
  - Identify differences between current state and goal state
  - Select an operator that can *reduce* the largest difference
  - If there are still differences, repeat – probably differences between:
    - state after operator applied and goal
    - initial state and state necessary for operator to be applied

Overview

Theorem proving

Means-Ends Analysis

- Overview
- Example of MEA
- GPS

STRIPS

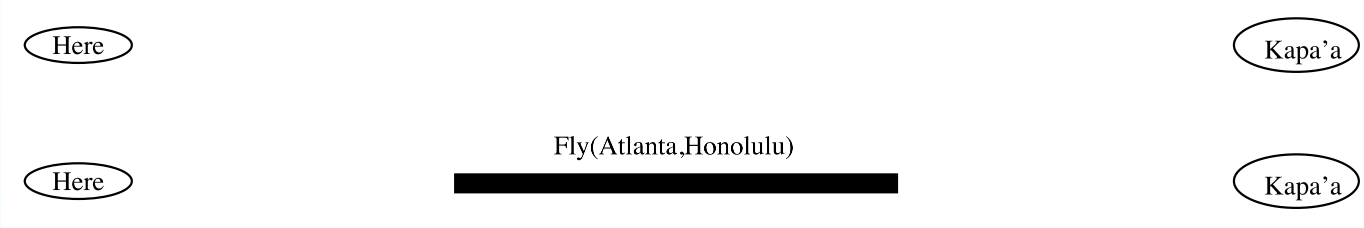
Example of MEA

- Problem: Go from here to Kapa'a, Kahuai, Hawaii
- Major difference: I'm here, I want to be there
- Operators: Many, only some of which can reduce the difference

Artificial Intelligence

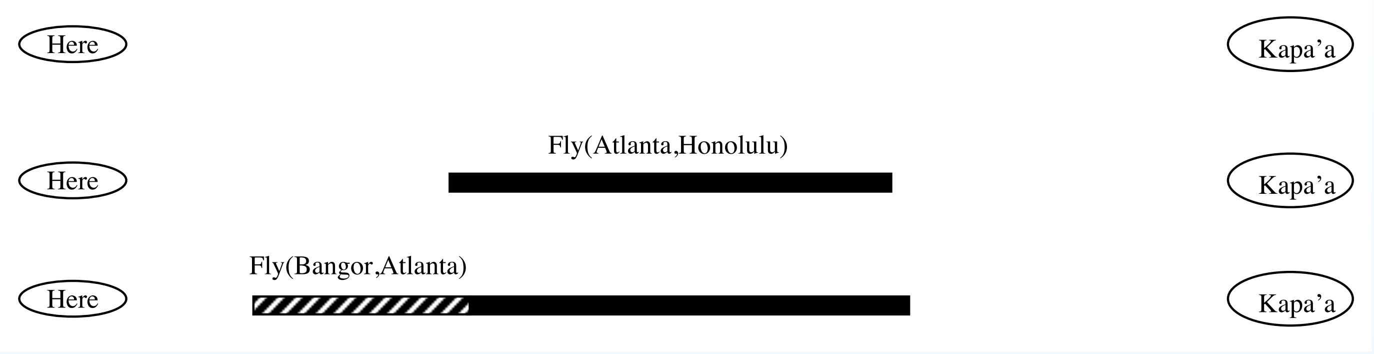
Example of MEA

- Problem: Go from here to Kapa’a, Kahuai, Hawaii
- Major difference: I’m here, I want to be there
- Operators: Many, only some of which can reduce the difference



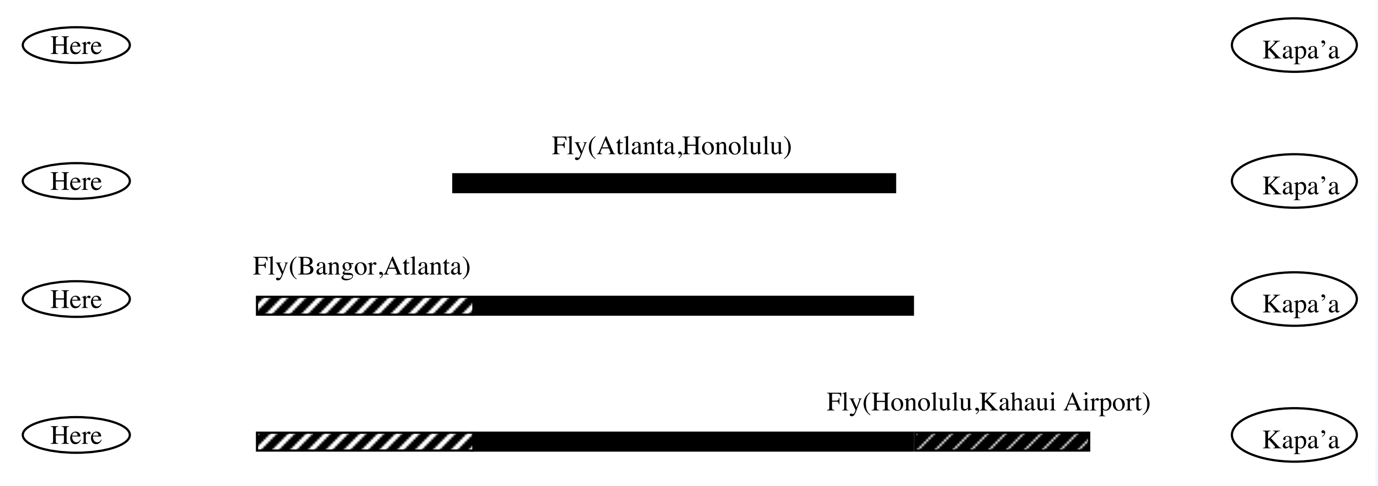
Example of MEA

- Problem: Go from here to Kapa'a, Kahuai, Hawaii
- Major difference: I'm here, I want to be there
- Operators: Many, only some of which can reduce the difference



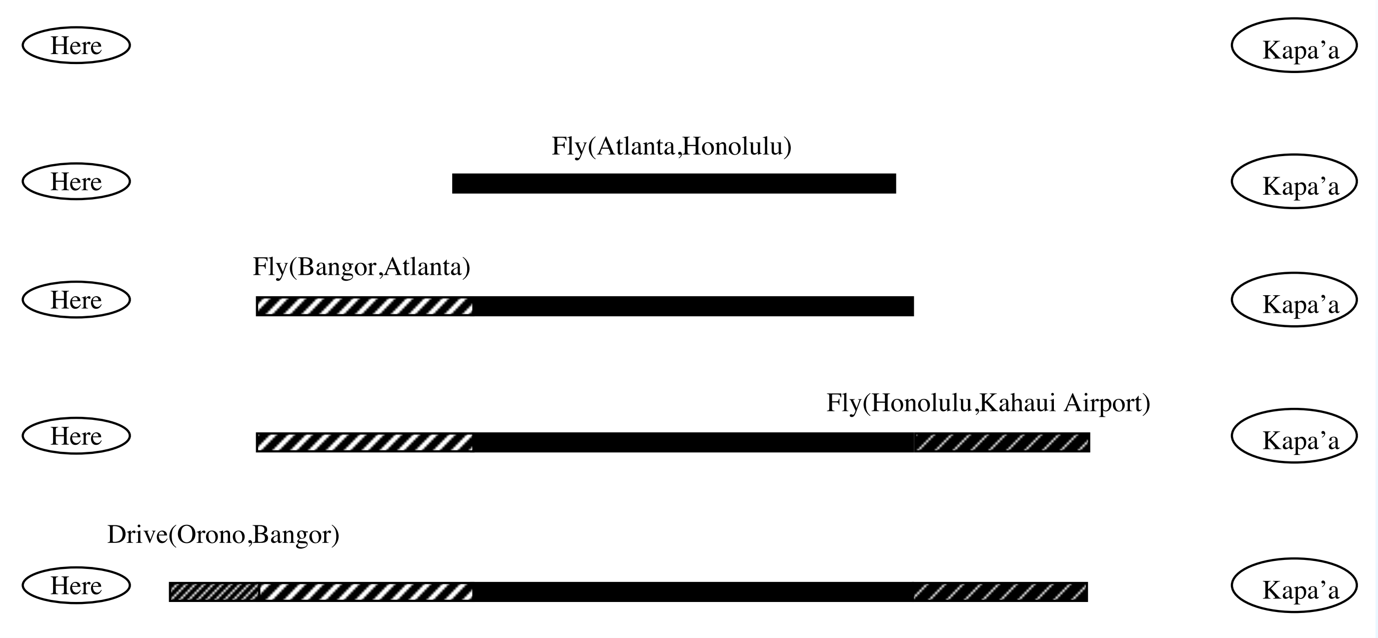
Example of MEA

- Problem: Go from here to Kapa'a, Kahuai, Hawaii
- Major difference: I'm here, I want to be there
- Operators: Many, only some of which can reduce the difference



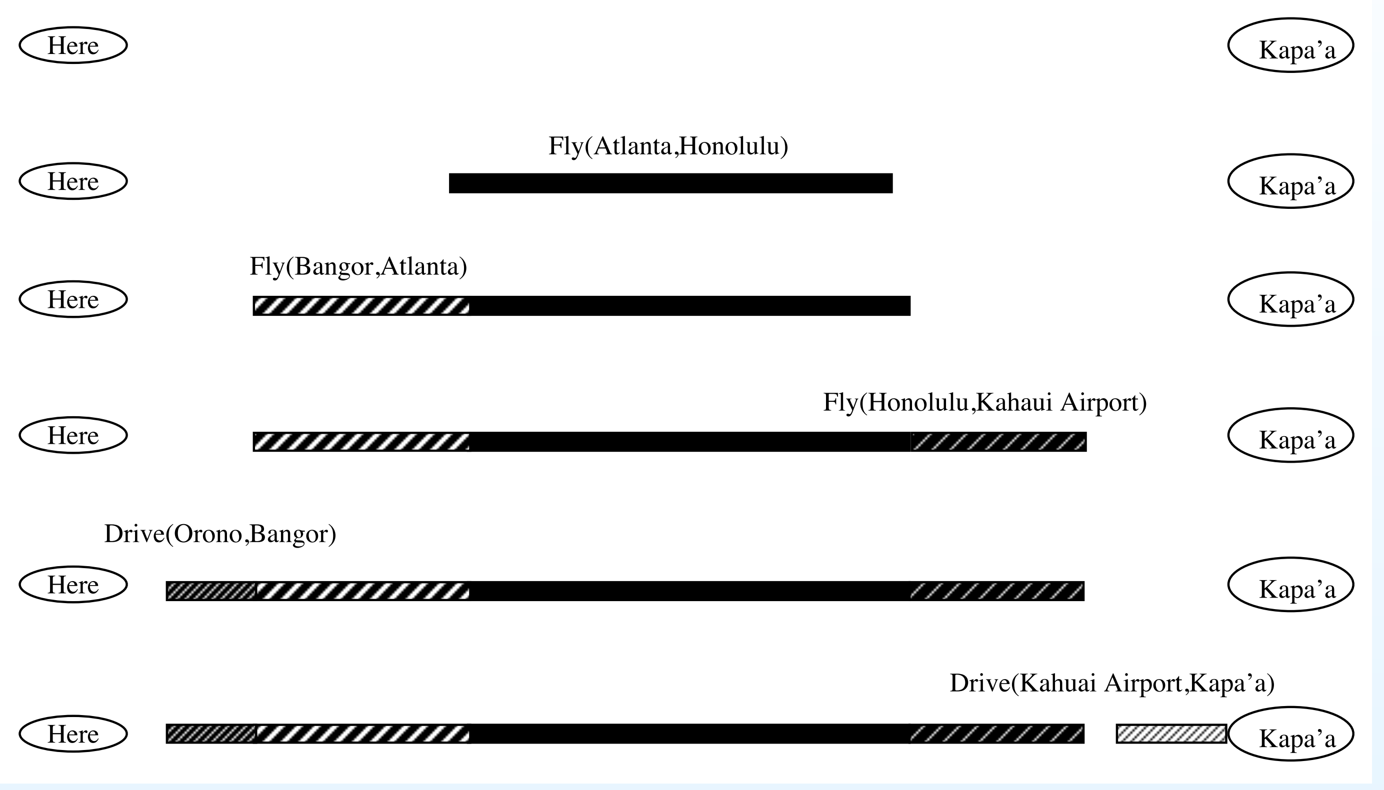
Example of MEA

- Problem: Go from here to Kapa'a, Kahuai, Hawaii
- Major difference: I'm here, I want to be there
- Operators: Many, only some of which can reduce the difference



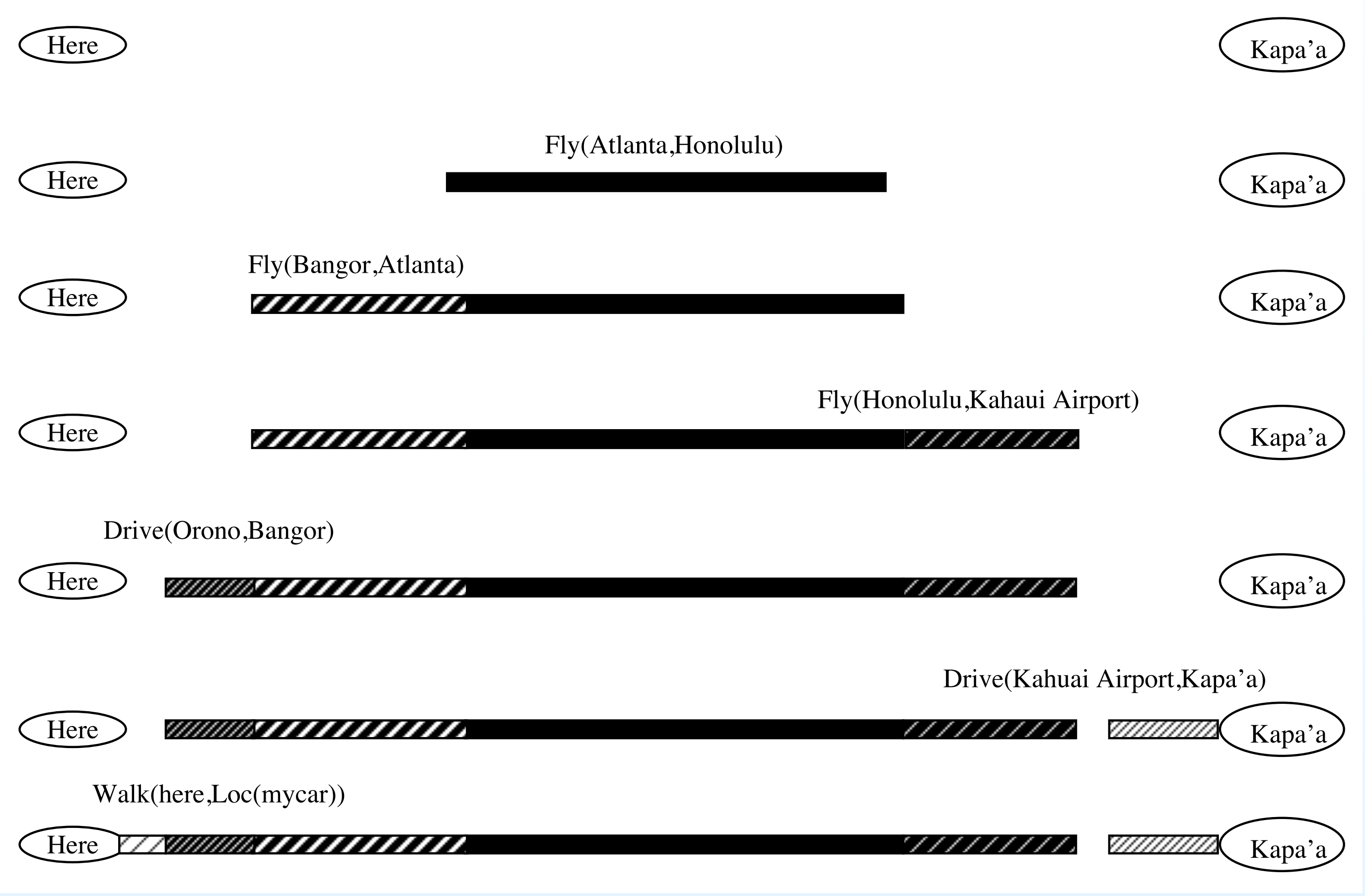
Example of MEA

- Problem: Go from here to Kapa'a, Kahuai, Hawaii
- Major difference: I'm here, I want to be there
- Operators: Many, only some of which can reduce the difference



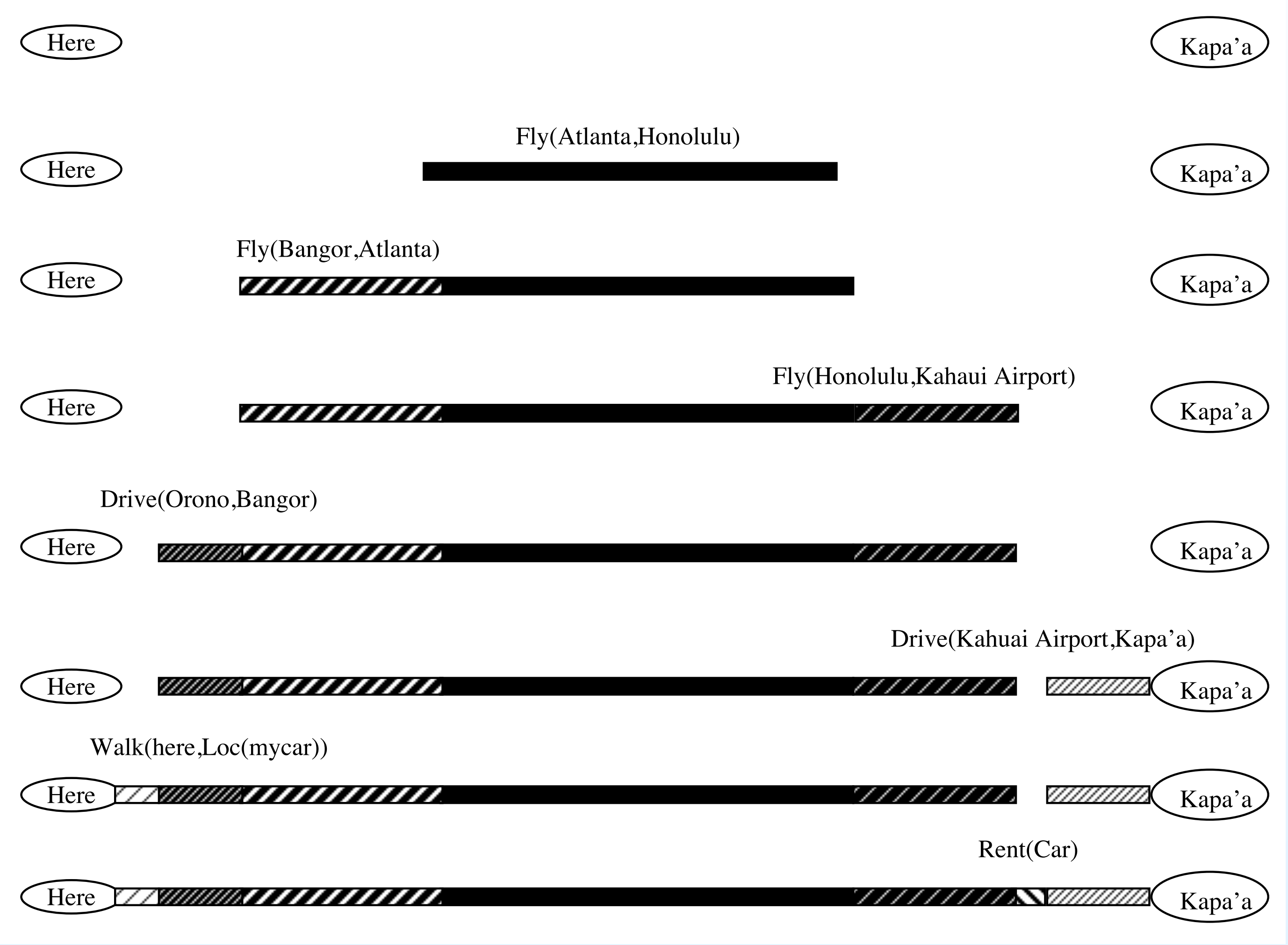
Example of MEA

- Problem: Go from here to Kapa'a, Kahuai, Hawaii
- Major difference: I'm here, I want to be there
- Operators: Many, only some of which can reduce the difference



Example of MEA

- Problem: Go from here to Kapa'a, Kahuai, Hawaii
- Major difference: I'm here, I want to be there
- Operators: Many, only some of which can reduce the difference



General Problem Solver

- MEA first embodied in the *General Problem Solver* (GPS) [Newell & Simon, 1963]
- *Difference table*
- Three (meta) operators: Transform(state,state), Reduce(difference), and Apply(operator)
- Heuristics: e.g., pick hardest difference first, don't generate same goal twice, ensure each goal easier than previous

General Problem Solver

- Solved:
  - Toy problems: water jug, missionaries and cannibals, towers of Hanoi, monkey and bananas
  - Theorem proving using predicate calculus
  - Symbolic integration
  - Parsed simple sentences
  - Letter series completion
- So, was it intelligent?
- Benefits , shortcomings?

STRIPS

STRIPS

- Stanford Research Institute Planning System [Fikes, 1971]
- Based on MEA
- *Regression planner* – work from goal to initial state deciding what must be true to apply an operator to achieve the state that considering
- See if goal is achieved
- If cannot prove, apply operator that will achieve goal

STRIPS

- Select operator
  - Identify *differences* between current state and goal
  - Find operator that can reduce the most important one
  - No difference table!
- To apply operator, make new goals of the operator's preconditions
- Plan is the list of operators that must be applied to complete the proof

Formalism for STRIPS

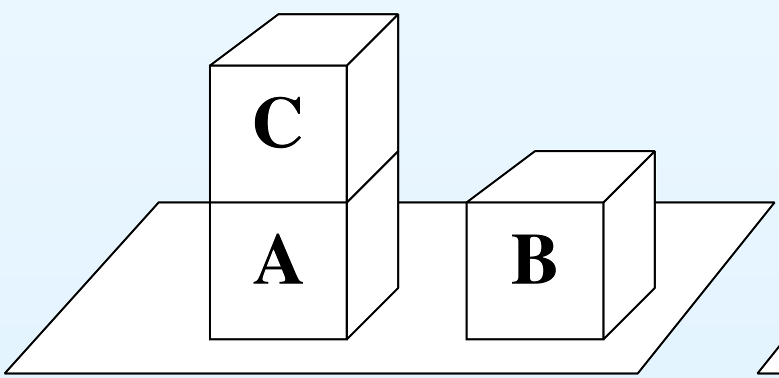
- Predicate calculus used to represent the world
  - conjunction of positive literals
  - ground clauses for states and goals
  - variables allowed in operators (variables are assumed to be universally quantified)
- Check to see if goal is achieved using RTP

STRIPS Operators

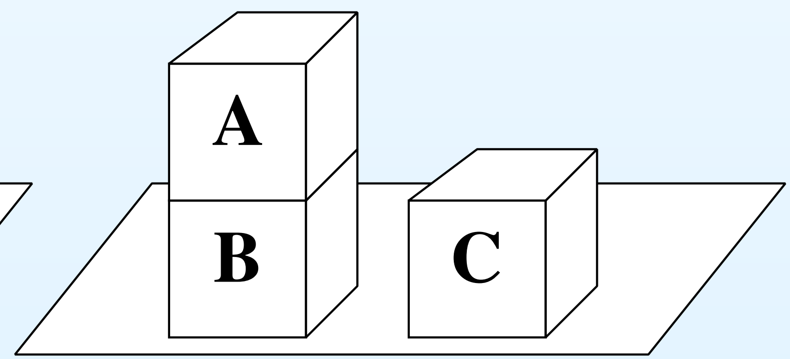
- Action - name of action with variable parameters
- Preconditions - what must be true for the operator to be successful
- Add-list - clauses that become true when the operator is applied
- Delete-list - clauses that are no longer true once the operator is applied
  - sometime combined and called “effects”
  - remove delete-list before adding add-list
  - STRIPS Assumption - these are the only things that change in the world

A Simple Example from the Blocks World

- Operators
  - Unstack(a,b)
    - preconditions: Armempty() $\wedge$ On(a,b) $\wedge$ Clear(a)
    - add list: Holding(a) $\wedge$ Clear(b)
    - delete list: Armempty() $\wedge$ On(a,b)
  - Stack(a,b)
  - PickUp(a)
  - PutDown(a)



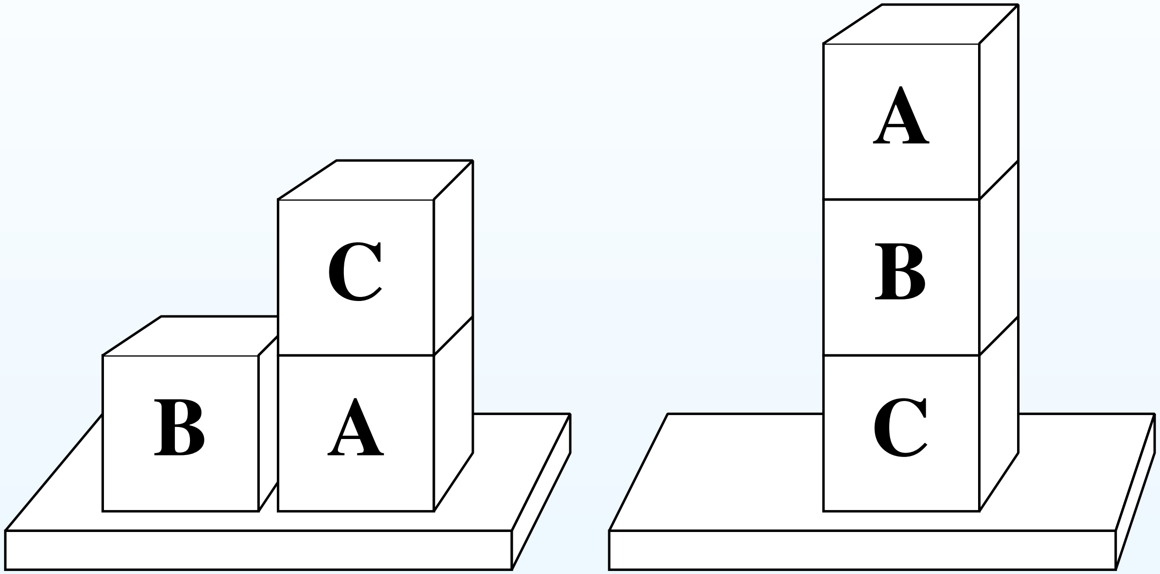
On(C,A) & OnTable(A) & OnTable(B)



On(A,B) & OnTable(B)

The Sussman Anomaly

- The Problem



- The Goal Stack?