Teaching Computers To Make Plans

Chad Hogg

November 14, 2018
Millersville University
UNIV 103
What Is True Now?

- There are blocks A and C on a table and a block B on top of block A
What Is True Now?
- There are blocks A and C on a table and a block B on top of block A

What Do I Want To Be True?
- The block C is on the table, with block B on top of it and block A on top of B
## Making Block-Moving Plans

### What Is True Now?
- There are blocks A and C on a table and a block B on top of block A

### What Do I Want To Be True?
- The block C is on the table, with block B on top of it and block A on top of B

### What Actions Can I Take?
- A robotic arm can move a block \(x\) from the table to on top of a block \(y\), as long as neither \(x\) nor \(y\) is covered.
- Or it can move a block \(x\) from on top of a block \(y\) to the table, as long as \(x\) is not covered.
- Or it can move a block \(x\) from on top of a block \(y\) to on top of a block \(z\), as long as neither \(x\) or \(z\) is covered.
Part Of The Block-Moving State Space

Chad Hogg
Teaching Computers To Make Plans
Representations

**Purpose**

- Computers are not good at interpreting natural language.
- They are very good at manipulating symbols in a formal language.

Chad Hogg | Teaching Computers To Make Plans
Representations

Purpose

- Computers are not good at interpreting natural language
- They are very good at manipulating symbols in a formal language

Approach

- Each of the things that is (or could be) true of the world will be a sentence in first-order logic
- The current state is a set of logical atoms
- An action has preconditions, negative effects, and positive effects, all sets of logical atoms
- Our goal is a set of logical atoms
\{ \text{OnTable}(A), \text{OnTable}(C), \text{OnBlock}(B,A), \text{Clear}(B), \text{Clear}(C) \} \quad \{ \text{OnTable}(A), \text{OnTable}(B), \text{OnTable}(C), \text{Clear}(A), \text{Clear}(B), \text{Clear}(C) \} \quad \{ \text{OnTable}(C), \text{OnBlock}(A,B), \text{OnBlock}(B,C), \text{Clear}(A) \}
( :action MoveBlockToBlock( B, A, C )
  :preconditions { OnBlock( B, A ), Clear( B ), Clear( C ) } 
  :negative-effects { OnBlock( B, A ), Clear( C ) } 
  :positive-effects { OnBlock( B, C ), Clear( A ) } 
)
Linear Planning

Finding A Plan

- If the goal is true in the current state, no actions are necessary.
- Otherwise, think about what the new state would be if we took one of the applicable actions.
- If the goal is true in that successor state, then that action solves the problem.
- If not, maybe some new action will be applicable in the successor state.
Linear Planning

Finding A Plan

- If the goal is true in the current state, no actions are necessary.
- Otherwise, think about what the new state would be if we took one of the applicable actions.
- If the goal is true in that successor state, then that action solves the problem.
- If not, maybe some new action will be applicable in the successor state.

Search Strategies

- Breadth-first search looks at all plans of 1 action, then all plans of 2 actions, etc.
- Depth-first search chooses an action, immediately looks for actions from the successor state, chooses one, etc.
Linear Search For A Plan (Depth-First)
Linear Search For A Plan (Depth-First)
Linear Search For A Plan (Depth-First)
Linear Search For A Plan (Depth-First)

Chad Hogg  
Teaching Computers To Make Plans
Linear Search For A Plan (Depth-First)

Chad Hogg
Teaching Computers To Make Plans
Linear Search For A Plan (Depth-First)

Chad Hogg | Teaching Computers To Make Plans
Linear Search For A Plan (Depth-First)
Linear Search For A Plan (Depth-First)
Linear Search For A Plan (Depth-First)
What Is True Now?

- I am at Millersville, have a car full of gasoline, and have $500
What Is True Now?
- I am at Millersville, have a car full of gasoline, and have $500

What Do I Want To Be True?
- I am at Notre Dame
Making Travel Plans

What Is True Now?
- I am at Millersville, have a car full of gasoline, and have $500

What Do I Want To Be True?
- I am at Notre Dame

What Actions Can I Take?
- Drive from location $x$ to location $y$, as long as I am in location $x$ and have an automobile with enough gasoline
- Buy an airplane ticket from airport $x$ to airport $y$, as long as I am in airport $x$ and have enough money
- Fly from airport $x$ to airport $y$, as long as I am in airport $x$ and have a ticket
- Take a taxi from location $x$ to location $y$, as long as I am in location $x$ and have enough money and a driver is willing to take me
Hierarchical Planning

<table>
<thead>
<tr>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Instead of a goal set of logical atoms to make true, a hierarchical planner has a sequence of <strong>tasks</strong> to accomplish.</td>
</tr>
<tr>
<td>• Some tasks are <strong>primitive</strong>, which means they are equivalent to actions, others are <strong>nonprimitive</strong>.</td>
</tr>
</tbody>
</table>
Hierarchical Planning

Tasks
- Instead of a goal set of logical atoms to make true, a hierarchical planner has a sequence of tasks to accomplish.
- Some tasks are primitive, which means they are equivalent to actions, others are nonprimitive.

Decomposition Methods
- Decomposition methods state that one nonprimitive task can be accomplished by accomplishing a sequence of simpler or more concrete tasks.
- Components:
  - A head, which is the nonprimitive task to decompose.
  - A sequence of subtasks, which are the way the task may be accomplished.
  - A set of preconditions that specify when it is legal / possible to accomplish the task this way.
Example Decomposition Methods

The Task Of Traveling Millersville To Notre Dame

- If I have a lot of money, I could travel Millersville to PHI, travel PHI to ORD, and then travel ORD to Notre Dame.

- If I have at least some money and a car, I could drive, stopping at various waypoints for fuel.

- I could hitchhike, no matter what.
### The Task Of Traveling Millersville To Notre Dame

- If I have a lot of money, I could travel Millersville to PHI, travel PHI to ORD, and then travel ORD to Notre Dame
- If I have at least some money and a car, I could drive, stopping at various waypoints for fuel
- I could hitchhike, no matter what

### The Task Of Traveling Millersville To PHI

- If I have a car with enough fuel, I could drive there
- If I have enough money, I could take a taxi
Example Decomposition Methods

The Task Of Traveling Millersville To Notre Dame
- If I have a lot of money, I could travel Millersville to PHI, travel PHI to ORD, and then travel ORD to Notre Dame
- If I have at least some money and a car, I could drive, stopping at various waypoints for fuel
- I could hitchhike, no matter what

The Task Of Traveling Millersville To PHI
- If I have a car with enough fuel, I could drive there
- If I have enough money, I could take a taxi

The Task Of Traveling PHI To ORD
- If I have enough money, I can buy a ticket then take a flight
- If I have a car and some money, I could drive there
Hierarchical Search For A Plan

```plaintext
Travel(
  Millersville,
  Notre Dame )
```
Hierarchical Search For A Plan

Travel( Millersville, Notre Dame )
  \rightarrow
Travel( Millersville, PHI )
  \rightarrow
Travel( PHI, ORD )
  \rightarrow
Travel( ORD, Notre Dame )
Hierarchical Search For A Plan

- Travel(Millersville, Notre Dame)
  - Travel(Millersville, PHI)
    - Drive(Millersville, PHI, MyCar)
  - Travel(PHI, ORD)
  - Travel(ORD, Notre Dame)
Hierarchical Search For A Plan

Travel(Millersville, Notre Dame)

Travel(Millersville, PHI)
  - BuyTicket(Flight102)
  - Drive(Millersville, PHI, MyCar)

Travel(PHI, ORD)
  - TakeFlight(Flight102)

Travel(ORD, Notre Dame)
Hierarchical Search For A Plan

Travel(Millersville, Notre Dame)

Travel(Millersville, PHI)
- BuyTicket(Flight102)
  - Drive(Millersville, PHI, MyCar)

Travel(PHI, ORD)
- CallTaxi(ORD, Taxi1)
  - TakeFlight(Flight102)

Travel(ORD, Notre Dame)
- RideInTaxi(ORD, Notre Dame, Taxi1)