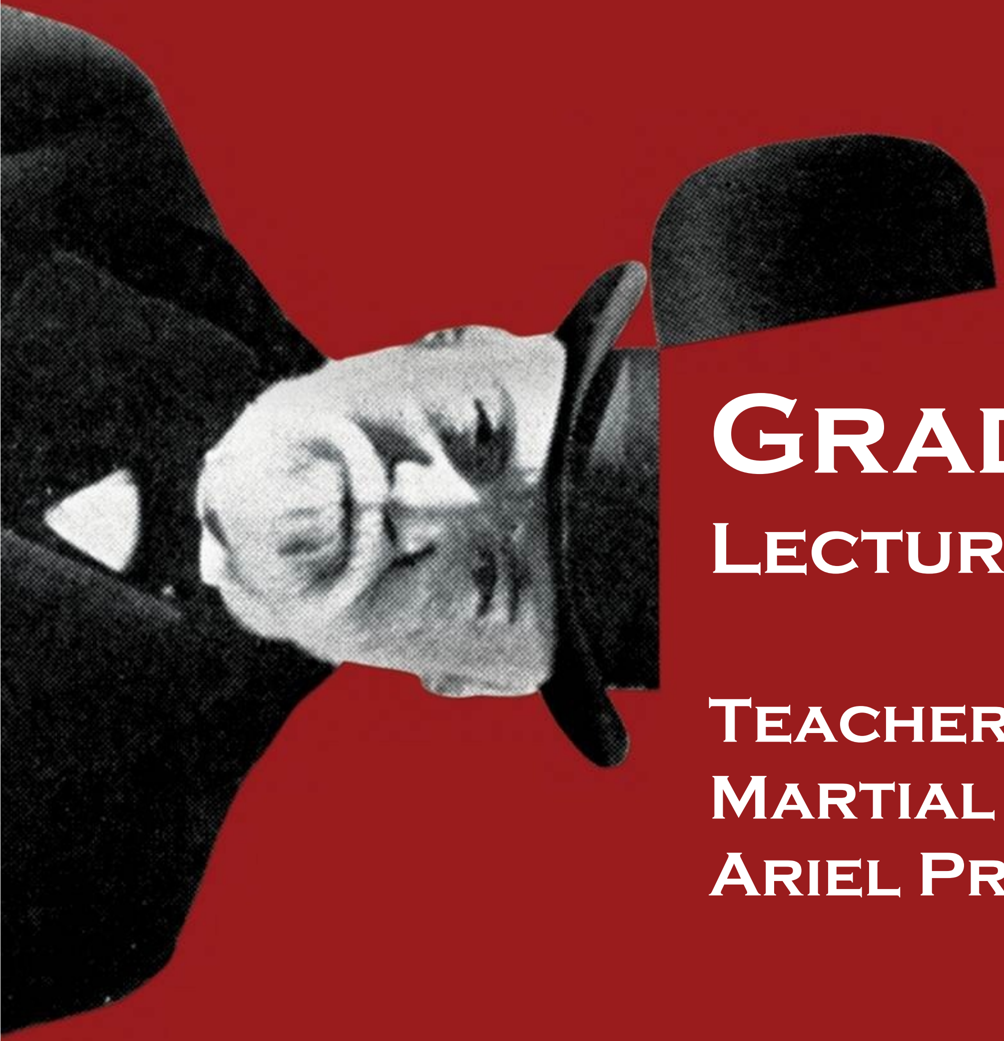


Graduate AI

Lecture 15: Planning 1

Teachers:
Martial Hebert
Ariel Procaccia (this time)

Planning: big picture

- AI studies rational action
- Devising a plan of action to achieve one's goal is a critical part of AI
- In fact planning is glorified search
- Similarly to CSPs, we will consider a factored representation of states

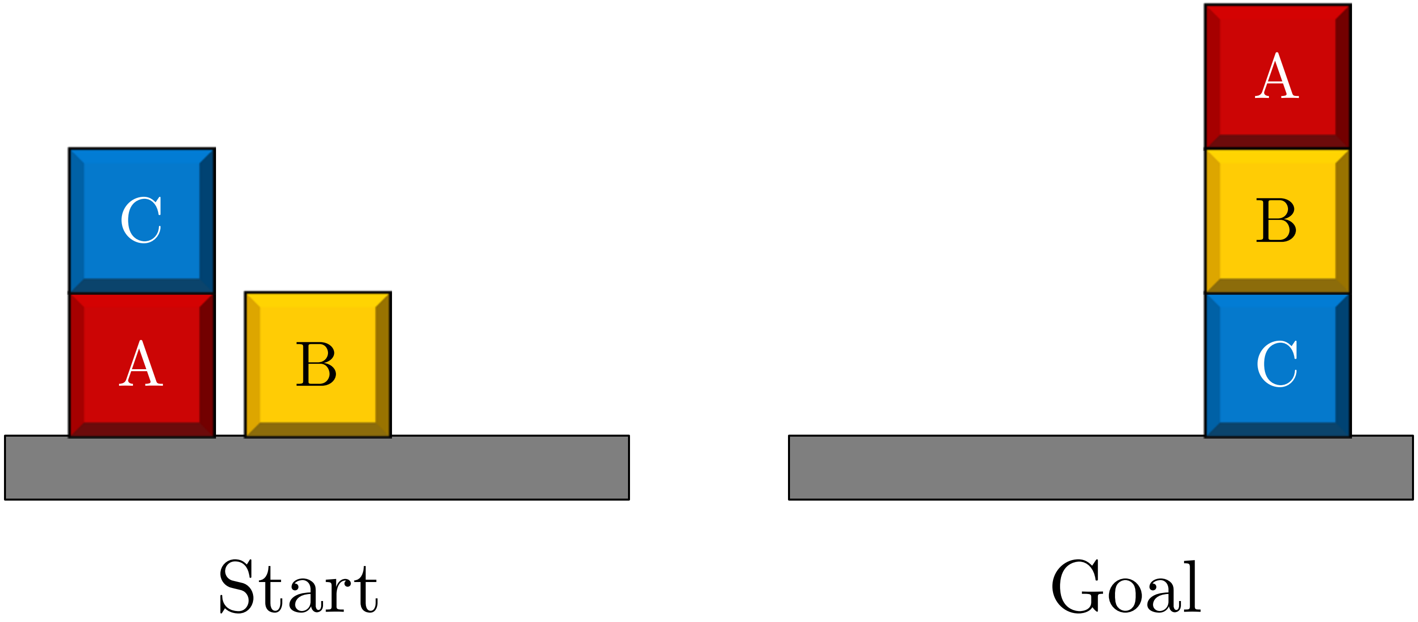
Propositional STRIPS planning

- STRIPS = Stanford Research Institute Problem Solver (1971)
- State is a conjunction of **conditions**, e.g., $at(Truck_1, Shadyside) \wedge at(Truck_2, Oakland)$
- States are transformed via **operators** that have the form Preconditions $\Rightarrow$ Postconditions)

Propositional STRIPS planning

- Pre is a conjunction of positive and negative conditions that must be satisfied to apply the operation
- Post is a conjunction of positive and negative conditions that become true when the operation is applied
- We are given the initial state
- We are also given the **goals**, a conjunction of positive and negative conditions
- We think of a state as a set of positive conditions, hence an operation has an "add list" and a "delete list"

Blocks world

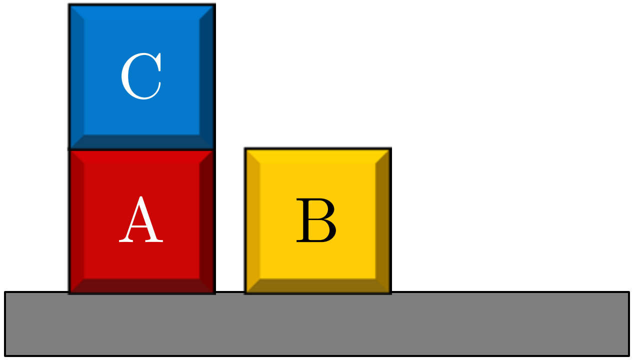


Blocks world

- Conditions: on(A,B), on(A,C), on(B,A), on(B,C), on(C,A), on(C,B), clear(A), clear(B), clear(C), on(A,Table), on(B,Table), on(C,Table)
- Operators for moving blocks
  - Move C from A to the table:
    clear(C) $\wedge$ on(C,A)
    $\Rightarrow$ on(C,Table) $\wedge$ clear(A) $\wedge$ $\neg$on(C,A)
  - Move A from the table to B
    clear(A) $\wedge$ on(A,Table) $\wedge$ clear(B)
    $\Rightarrow$ on(A,B) $\wedge$ $\neg$clear(B) and $\neg$on(A,Table)

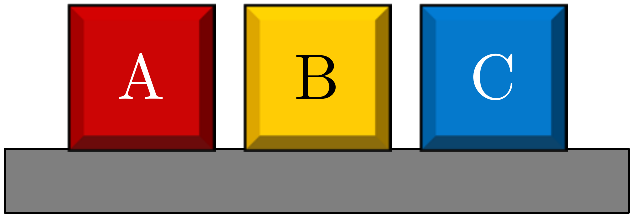
The plan

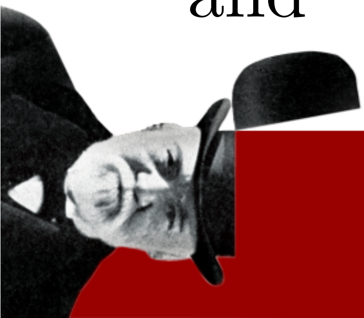
- State: on(C,A), on(A,Table), on(B,Table), clear(B), clear(C)
- Action: clear(C) $\wedge$ on(C,A) $\Rightarrow$ on(C,Table) $\wedge$ clear(A) $\wedge$ $\neg$on(C,A)



The plan

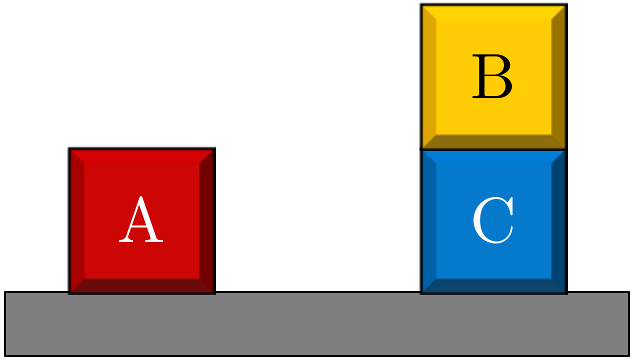
- State: on(A,Table), on(B,Table), clear(B), clear(C), on(C,Table), clear(A)
- Action:
  clear(C) $\wedge$ on(B,Table) $\wedge$ clear(B)
  $\Rightarrow$ on(B,C) $\wedge$ $\neg$clear(C) and $\neg$on(B,Table)





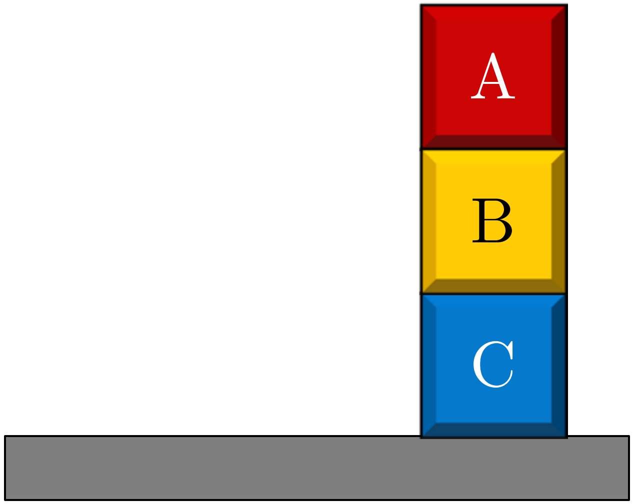
The plan

- State: on(A,Table), clear(B), on(C,Table), clear(A), on(B,C)
- Action:
  clear(B) $\wedge$ on(A,Table) $\wedge$ clear(A)
  $\Rightarrow$ on(A,B) $\wedge$ $\neg$clear(B) and $\neg$on(A,Table)



The plan

- State: on(C,Table), clear(A), on(B,C), on(A,B)
- Goals: on(A,B), on(B,C)



Complexity of planning

- PLANSAT is the problem of determining whether a given planning problem is satisfiable
- In general PLANSAT is PSPACE-complete
- We will look at some special cases

Complexity of Planning

- **Theorem 1.** Assume that actions have only positive preconditions and a single postcondition. Then SATPLAN is in P [T. Bylander. The Computational Complexity of Propositional STRIPS Planning. AIJ 1994]
- **Theorem 2.** Blocks world problems can be encoded as above
- **Silly corollary.** Blocks world problems can be solved in polynomial time (Duh)

Proof of Theorem 2

- We will convert blocks world operators to operators that have only positive preconditions and a single postcondition
- Let the blocks be $B_1,\ldots,B_n$
- Conditions: B_i is *not* on top of B_j, off(i,j)
- If B_i is clear, off(k,i) is true for all k
- If B_i is on the table, off(i,k) is true for all k
- Move B_i from the top of B_j to the table:
  $\bigwedge_k \mathrm{off}(k,i) \wedge \bigwedge_{k \neq i} \mathrm{off}(k,j) \Rightarrow \mathrm{off}(i,j)$
- Move B_i from the table on top of B_j:
  $\bigwedge_k \mathrm{off}(k,i) \wedge \bigwedge_k \mathrm{off}(i,k) \wedge \bigwedge_k \mathrm{off}(k,j) \Rightarrow \neg\mathrm{off}(i,j)$ ∎

Proof of Theorem 1

- **Claim.** It is sufficient to consider plans that first make conditions true, then make conditions false
- **Proof:**
  - Suppose that o_i and o_{i+1} are adjacent operators s.t. the postcondition p of o_i is negative and the postcondition q of o_{i+1} is positive
  - If $p=q$ then we can delete o_i because its effect is reversed
  - Otherwise, can switch o_i and o_{i+1} ■

Proof of Theorem 1

- Thus, if there is a solution, there is an intermediate state S such that
  - S can be reached from the initial state using operations with positive postconditions
  - The positive goals are a subset of S
  - Negative goals can be achieved via operations with negative postconditions
- Search for an intermediate state S with these properties

Proof of Theorem 1

- Implement procedure TurnOn(X): given set of conditions X, find maximal state S such that $S \cap X = \phi$ that can be reached from initial state using operators with positive postconditions
  - Preconditions are positive, so:
  - Simply apply all such operators until it makes no difference

Proof of Theorem 1

- Denote S' the intermediate state S after removing negative goals
- Implement procedure TurnOff(S): find the maximal S'' such that S' is reachable from S'' using operators with negative postconditions
  - Simply search backwards from S' and reverse operators with (i) negative postconditions, (ii) preconditions in S

Proof of Theorem 1

- In the first iteration, if positive goals are not satisfied by S, there is no way to achieve them
- If $S \setminus S' \neq \phi$, it is impossible to remove these conditions; must be added to X
- X grows monotonically $\Rightarrow$ polynomial time

```
X = ϕ
loop
    S = TurnOn(X)
    If S does not contain positive
        goals then return reject
    S' = TurnOff(S)
    If S=S' then return accept
    X = X ∪ (S\S')
    If X intersects with initial
        state then return reject
```

More on complexity

- We prove the following statement on the board
- The proof also appears in:
  T. Bylander. The Computational Complexity of Propositional STRIPS Planning. AIJ 1994 (link on course website)
- **Theorem.** Assume that actions have only positive postconditions. Then SATPLAN is NP-complete [Bylander 94, Theorem 3.5 on page 15]
- **Corollary.** This is true even if actions have one precondition and one positive postcondition