- Final (optional) project
  - 30% of final grade was from projects
  - We've had 4 projects
  - Each was worth 7.5 of final grade

  - This project can help as much as 5 points on final grade:
  - Exceptional (A+): +5
  - Good (A): +4
  - Pretty Good (A-): +3
  - B+: +2
  - B: +1

- Projects: this Friday (hopefully)

- Search
- Genetic Algorithms
- Adversarial Search
- MDPs
- Reinforcement Learning
- Probabilistic Reasoning
- Bayesian Networks
- Particle Filters
- Neural Networks: Classification
- Classical Planning
- Ethics & Philosophy

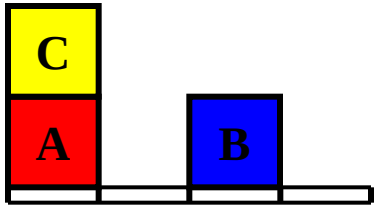
NEAT, UCT, +1 more?

- AI Game Programming Wisdom
- State of the Art AI in Games
- Compared to search
- Scheduling: baseball games, navy yard
- PDDL
- Closed-world assumption

- Why is it that fluents are not allowed to have negation, such as -At(p, JFK)? After all, aren't we implementing these using boolean variables anyway? If the answer is that negation is a (unary) function, then my question is why is it that we cannot reason with function. The book does not really give an example where functions can break our convenient treatment of fluents using set operations.

State Space

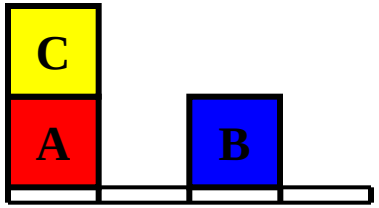
On(C, A)
On(A, Table)
On(B, Table)
Clear(C)
Clear(B)



- Representation
  - States described by propositions or ground predicates
  - Sparse encoding (database semantics): all unstated literals are false
  - Unique names: each object has its own single symbol

Actions

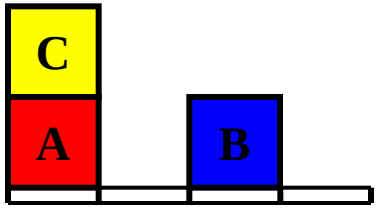
On(C, A)
On(A, Table)
On(B, Table)
Clear(C)
Clear(B)



ACTION: Move(b,x,y)
PRECONDITIONS: On(b,x), Clear(b), Clear(y)
POSTCONDITIONS: On(b,y), Clear(x)
¬On(b,x), ¬Clear(y)

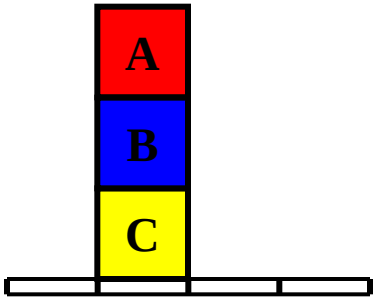
ACTION: Move(C,A,Table)
PRECONDITIONS: On(C,A), Clear(C), Clear(Table)
POSTCONDITIONS: On(C,Table), Clear(A)
¬On(C,A), ¬Clear(Table)

Start and Goal States



Start State

On(C, A)
On(A, Table)
On(B, Table)
Clear(C)
Clear(B)
Block(A)
…



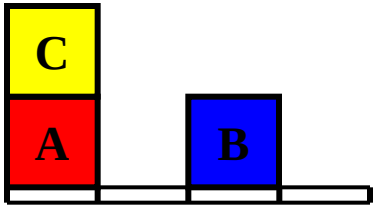
Goal State

On(B, C)
On(A, B)

Important: goal satisfied by any state which entails goal list

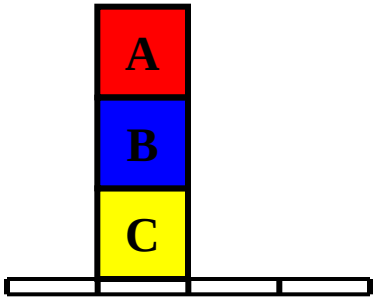
[MoveToTable(C,A), Move(B,Table,C), Move(A,Table,B)]

Planning Problems



On(C, A)
On(A, Table)
On(B, Table)
Clear(C)
Clear(B)

Sparse encoding, but complete state spec



Goal

Set of goal states, only requirements specified (think unary constraints)

On(B, C)
On(A, B)

Action schema, instantiates to give specific ground actions

Which goal first?

ACTION: MoveToTable(b,x)
PRECONDITIONS: On(b,x), Clear(b), Block(b), Block(x), (b≠x)
POSTCONDITIONS: On(b,Table), Clear(x)
¬On(b,x)

Practice

Action(Fly(p, from, to),

PRECOND: At(p, from) ^ Plane(p) ^ Airport(from) ^ Airport(to)

Effect: ~At(p, from) ^ At(p, to)

- What are all applicable concrete instances of Fly (p, from, to) for state:

At(P1,JFK) ^ At(P2,SFO) ^ Plane(P1) ^ Plane(P2) ^ Airport(JFK) ^ Airport(SFO)