

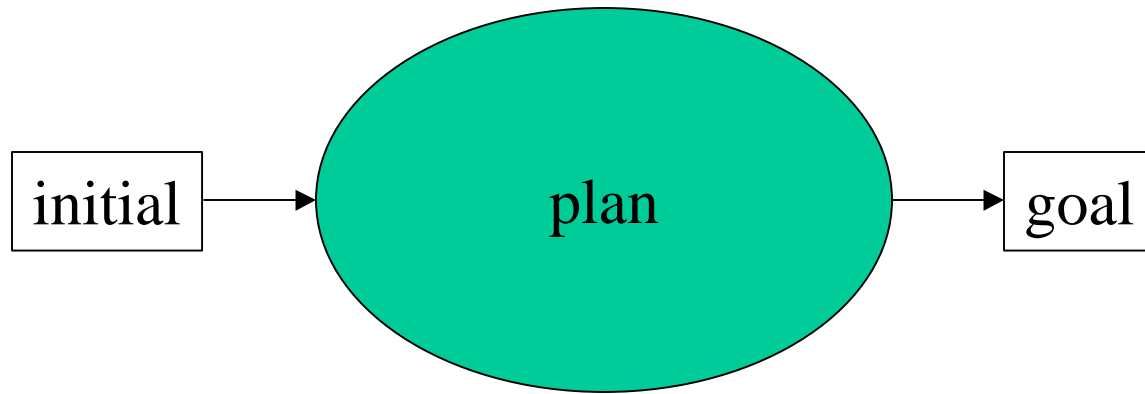
Directions In Planning

Understanding the Flow of
Time In Planning

Bart Massey

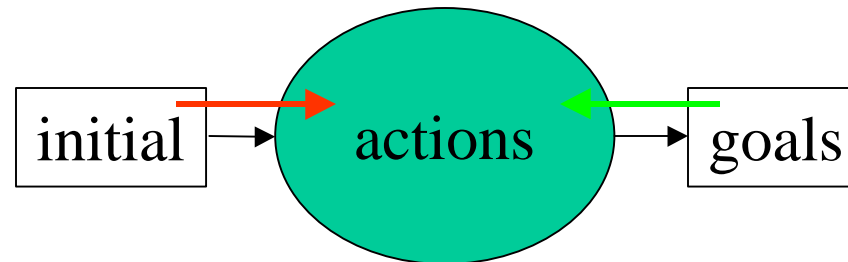
May 13, 1999

Planning



- **Given**
 - an **initial** situation
 - a **goal** situation
- **Find** a way to achieve goal from initial

Forward or Backward?



- **Forward**: consider actions which can **come first**, actions which can follow...
- **Backward**: consider actions which can **come last**, actions which can precede...
- Why are these different? Does it matter?

Message

- **Planning** is fundamentally time symmetric
- Many **problems** are not
- Many **planners** are not
- What we can do
 - Can **reverse** problems
 - Can **detect** planner direction

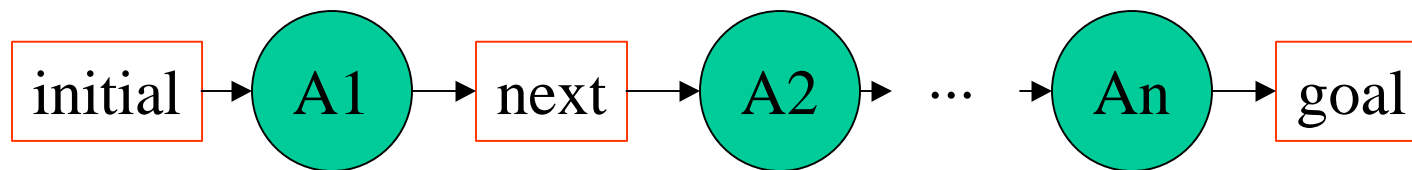
Propositional STRIPS Planning



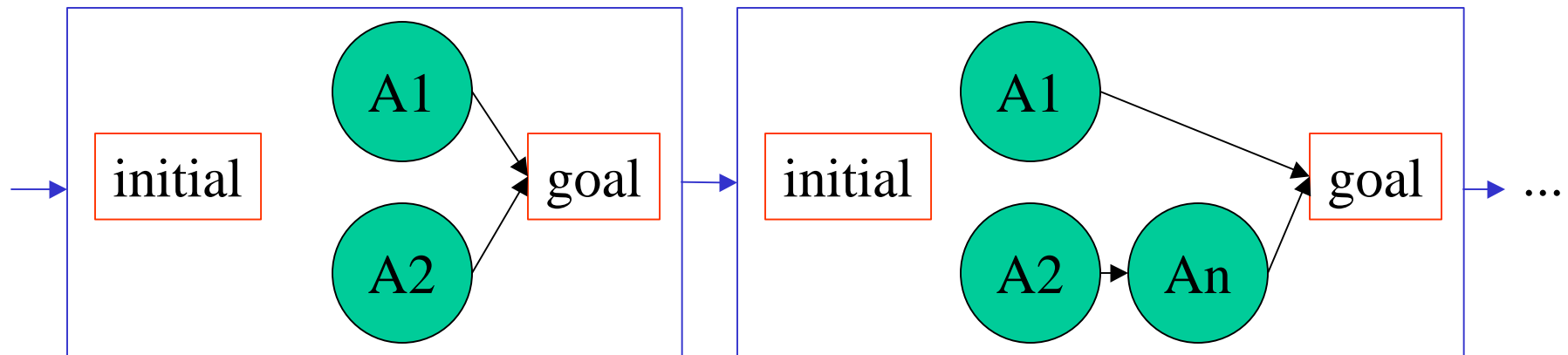
- Situations are Boolean state vectors
- Operators/actions
 - Are atomic, deterministic
 - Have preconditions, effects
- Plan: a sequence of actions

Standard Planning Algorithms

- **State-Space Search**: chain forward via states



- **Partial-Order Causal Link**: chain backward via plans



Newer Planning Algorithms

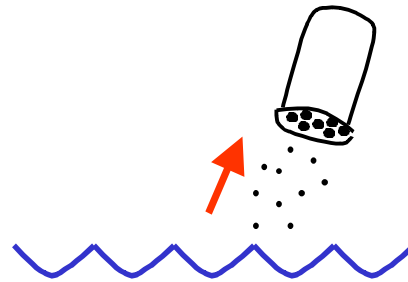
- **Graphplan**: forward chaining to prune search space, then backward search
- **SATplan**: transform planning problem to SAT problem, solve, transform soln back
- **Blackbox**: SATplan with planning problem constructed using plan graph

Sources Of Planning Directionality

- If (Prop. STRIPS) planning is directionally biased, where could the biases come from?
 - real world (**physics**)
 - biased **encodings**
 - underlying bias in Prop. STRIPS **formalism**
- Biases in **planning** interact with biases in **planners**

Time's Arrow

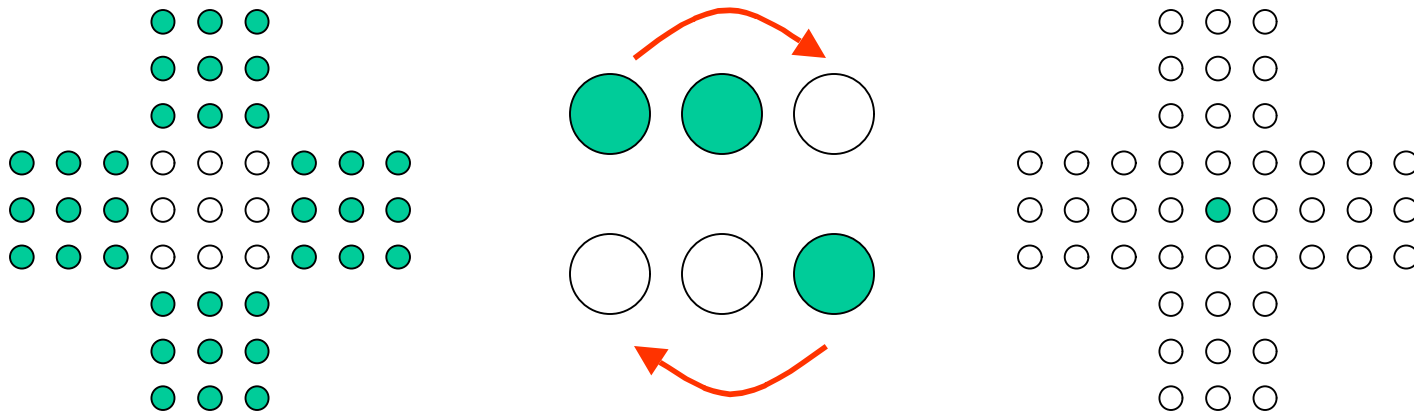
- **Physical** bias: action of “unsalting” water unlikely to succeed



- Prop. STRIPS doesn't appear to capture this
- Information theoretic “entropy”
 - not the same as physical entropy
 - doesn't occur in STRIPS

The Mind's Eye

- Perceptual bias leading to biased encodings
 - surely happens
 - poorly understood
 - either direction

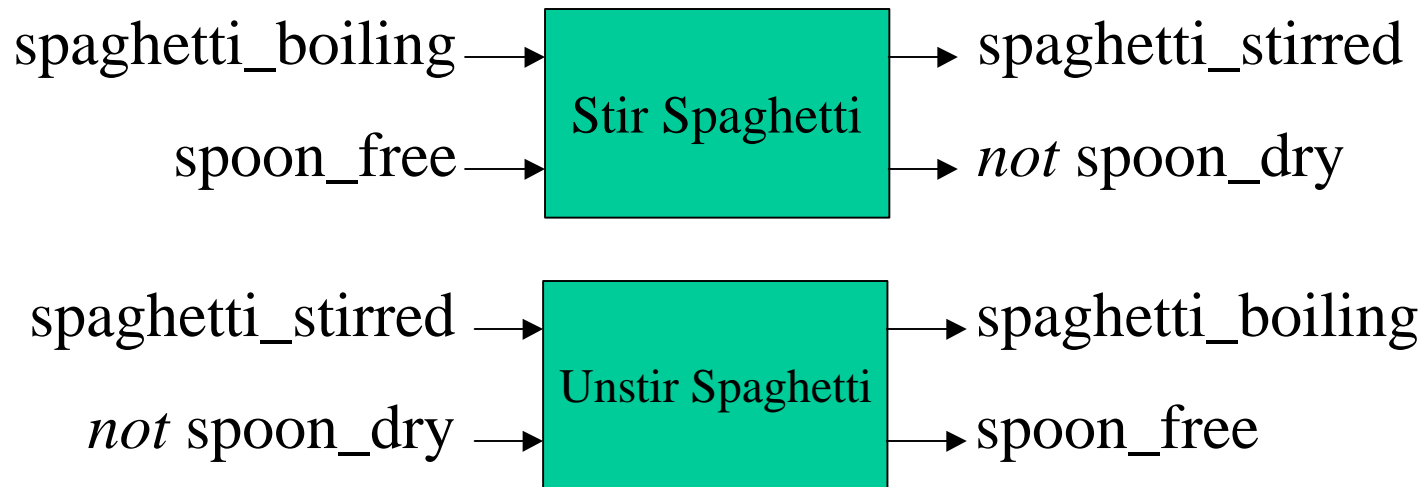


The Electronic Brain

- The underlying STRIPS **formalism** may be directionally biased
 - goal **descriptions**
 - **irreversible** actions
- Widespread belief in community
 - most argue that backward search is better
- Not true

Reversing STRIPS

- STRIPS appears time-symmetric
 - preconditions look like effects
 - turn planning algorithm around
 - turn problem around (!) [actions, init/goal]



Irreversible Actions

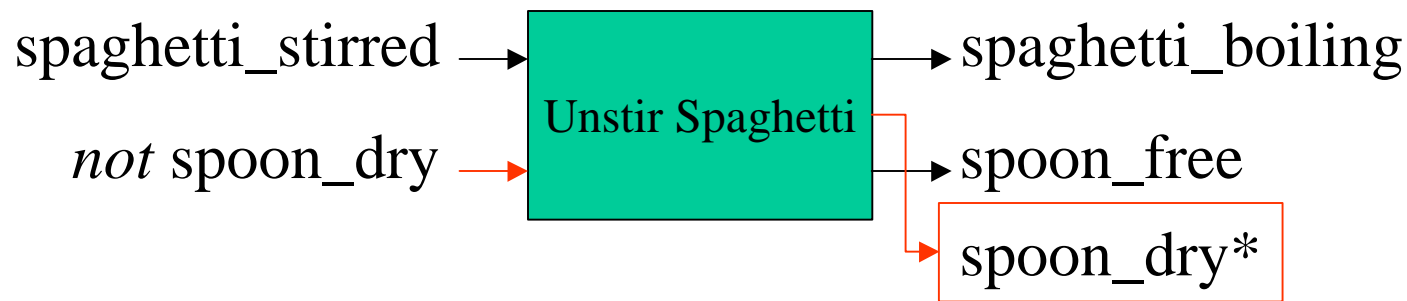
- Some STRIPS actions appear irreversible
 - Change fluent not mentioned in preconditions



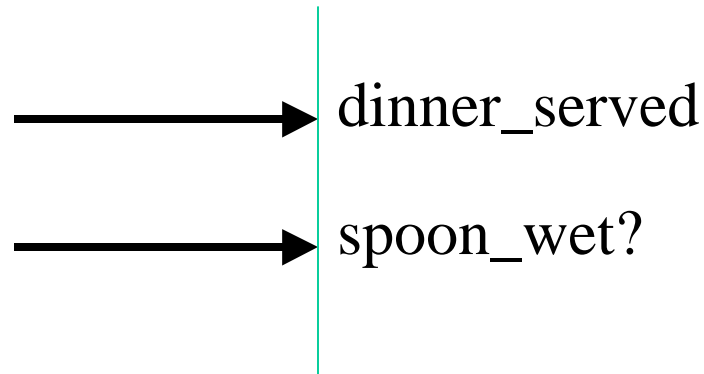
- Naïve problem reversal does not work

Compiling Out Irreversibility

- Solution: **transform** actions to allow richer effect descriptions
 - without changing solution space
 - without exploding complexity
- Introduce explicit Don't Care effects



Goal Descriptions



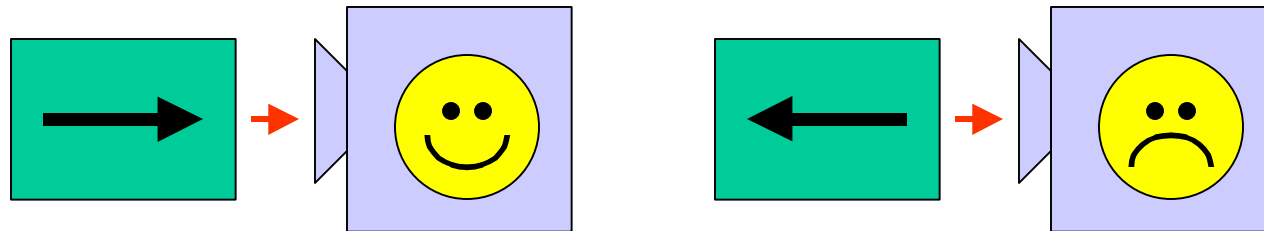
- Some elements of goal description **irrelevant**
- Instead, just require relevant elements (DC)
- Repair **asymmetry** by allowing initial descriptions (DK)

Planner Direction

- Older planners directionally biased
- But
 - **formalism** is unbiased
 - **problem** bias can vary
- Argues for **nondirectional** planner
- Direction of newer planners?

Determining Planner Direction

- Idea: construct problems that are
 - provably easy in one direction
 - provably hard in the other
- Can get direction from planner performance without examining planning algorithm

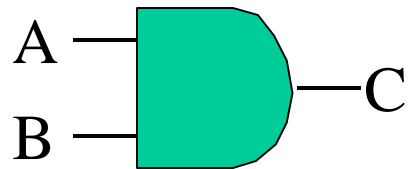


One-Way Functions

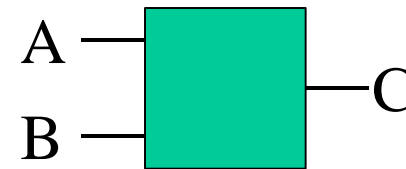
- Building block for one-way problems
- Functions that are
 - easy to compute
 - difficult to invert
- Provably exist, but no known construction
- Cryptographic functions OK approximation
- Can build crypto/one-way **circuits** as well

From Circuits To Plans

- Idea: transform each **gate** in one-way circuit into **operators** of planning problem
- Transformation via truth table (DNF)



A	B	C
0	0	0
0	1	0
1	0	0
1	1	1



Caveat: The “Easy Direction”

- (1999) planners are not very powerful
- Get lost on
 - large problems
 - deep problems (long solutions)
- Mostly do not **propagate**
- Crypto one-way functions too hard
 - develop tractable function with OK one-wayness

Using Directional Problems

- To detect planner directionality
 - Feed planner increasing size one-way problems
 - easy forward
 - easy backward
 - Measure planner performance
 - solution found?
 - time/nodes to solution
 - Answers: backward, forward, both, **bad**

Testbed Structure

Directionality of UCPOP

Directionality of ASP

Directionality of O-Plan

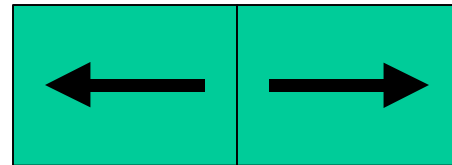
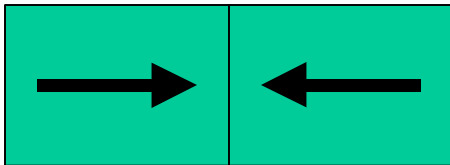
Directionality of Graphplan

Directionality of Blackbox/Relsat

New Planners Are Bidirectional

Composing One-Way Problems

- One-way problems can be composed
 - outside-in problems (meet in middle)
 - middle-out problems (islands)
- Raises tractability questions



The Flow Of Time In Planning

- **Planning** is fundamentally time symmetric
- Many **problems** are not
- Many **planners** are not
- What we can do
 - Can **reverse** problems
 - Can **detect** planner direction

DC/DK Conditions/Effects

Complexity Results

Proving Compilation Correctness

Proving Reversal Correctness