Intelligent Control and Cognitive Systems

brings you...

Planning and Action Selection

Joanna J. Bryson

University of Bath

Intelligence
Action Selection

- What is intelligence?
- Judged by expressed behaviour.
  - Judgement by people.
  - “Judgement” by Natural Selection.
- What matters: doing the right thing at the right time.

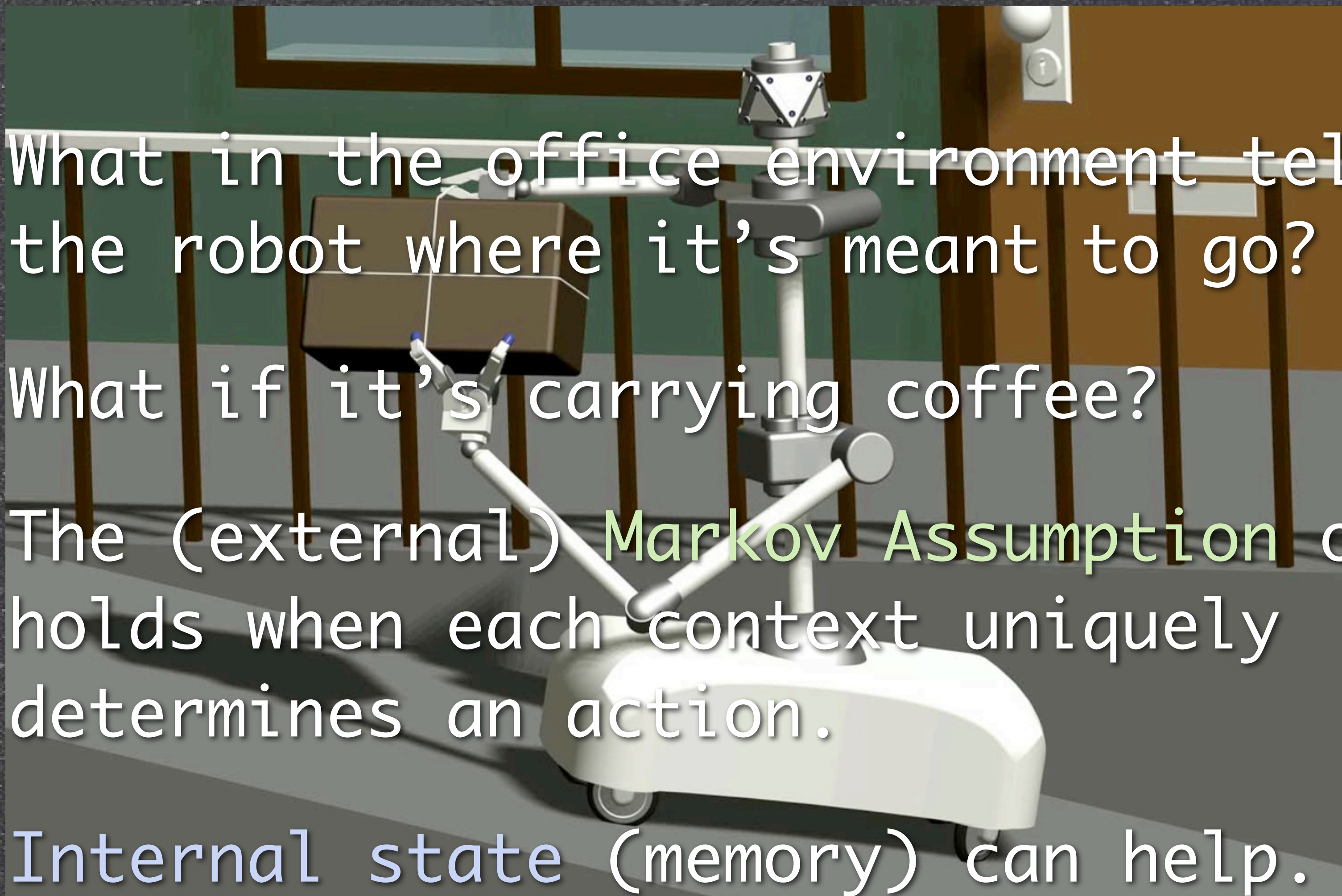
Strategies of Action Selection / Outline

- Productions
- Formal / Optimal Planning
- Reactive / Dynamic Plans
- Learning Plans

Productions & The Markov Assumption

- A production is a tuple: <sensory precondition, action>
- A production system (or expert system) is a set of productions used to solve a particular problem.
- Problem: much human behaviour cannot be determined only from the environment.

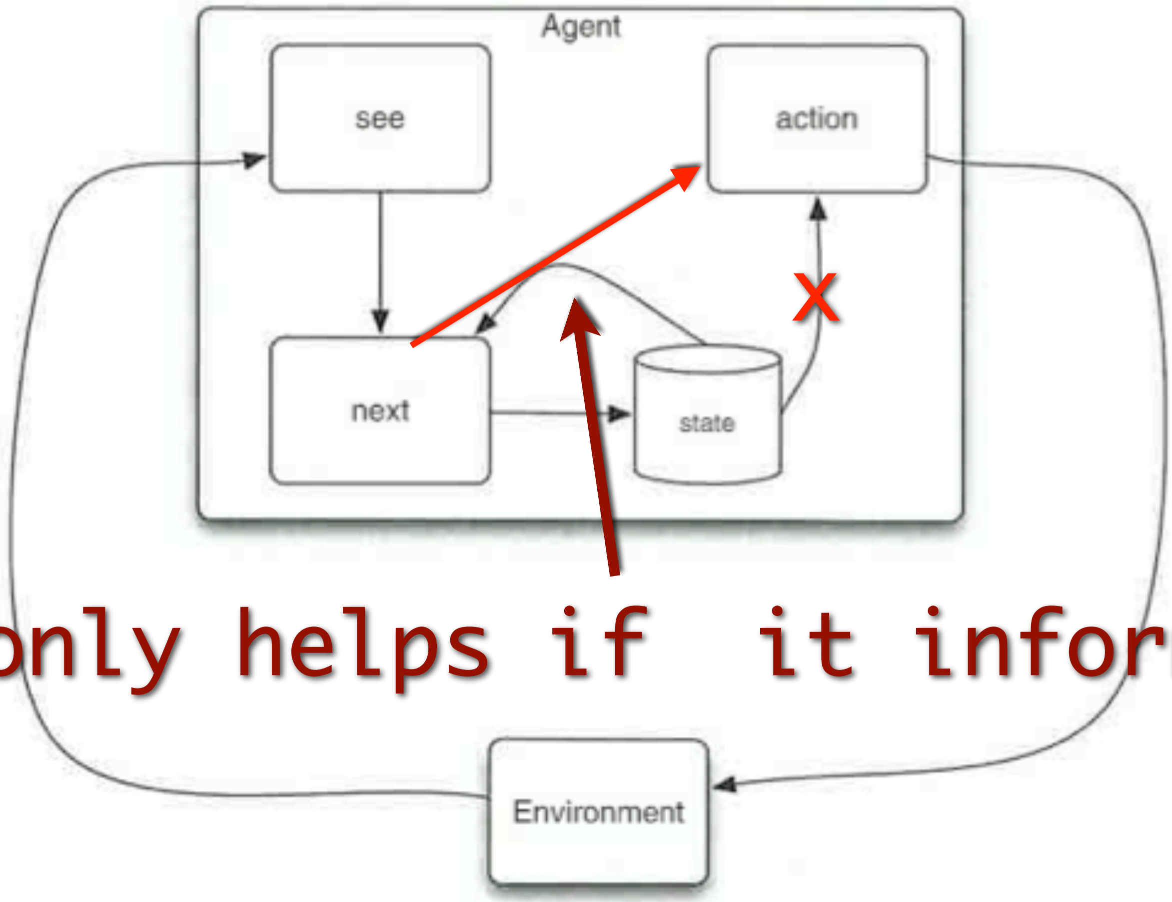
Delivery Robot



- What in the office environment tells the robot where it's meant to go?
- What if it's carrying coffee?
- The (external) Markov Assumption only holds when each context uniquely determines an action.
- Internal state (memory) can help.

Moravec (1998), ROBOT, page 108
Oxford University Press.

From last semester / Agents



State only helps if it informs AS!

Figure 2.2: An agent that maintains state.

AS–not state–chooses the A!

Agents with state

Strategies of Action Selection / Outline

- Productions
- Formal / Optimal Planning
- Reactive / Dynamic Plans
- Learning Plans

What Do We Want from Action Selection?

Optimality

Formal Planning for *Optimality*

- Provably correct: know we can get from here to the goal.
- Prove we can do it in the least amount of steps.
- Totally impossible. (Agre 1987, Simon 1956).

Satisficing

Heuristic

Yet people keep trying...

Intro to CS 541 (AI Planning)

http://www.isi.edu/~blythe/cs541

Jim Blythe

Jose Luis Ambite

Yolanda Gil

With Annotations by – JJB

Generating plans

- Given:
  - A way to describe the world
  - An initial state of the world
  - A goal description
  - A set of possible actions to change the world
- Find:
  - A prescription for actions to change the initial state into one that satisfies the goal

The situation calculus (McCarthy 63)

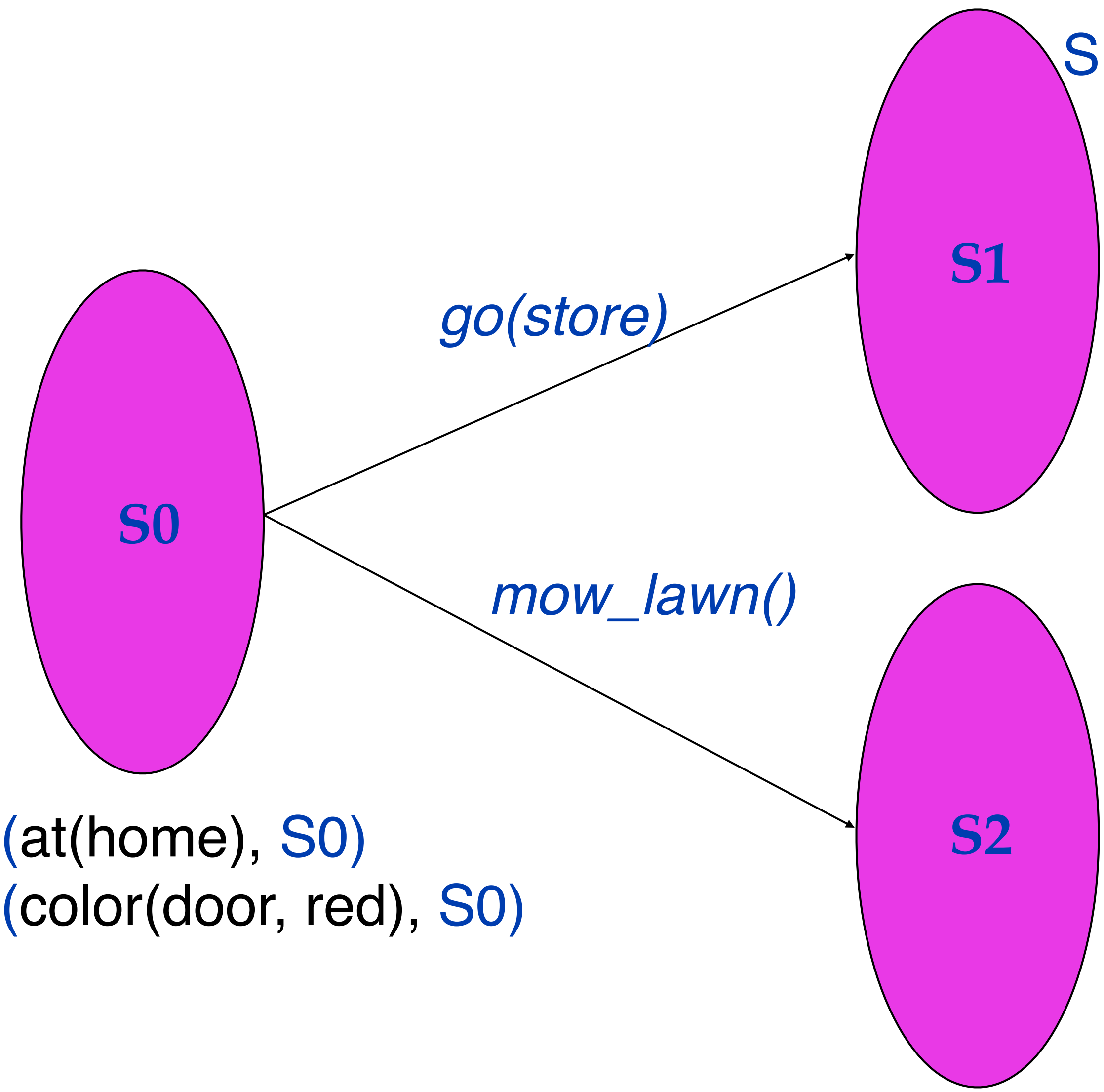
- Key idea: represent a snapshot of the world, called a ‘situation’ explicitly.

- ‘Fluents’ are statements that are true or false in any given situation, e.g. ‘I am at home’

- **Actions map situations to situations.**

Actions in formal planning are essentially functions used by agents to transition the world from one state to the next – JJB

S1 = **result**(*go(store)*, S0)



¬ **holds**(at(home), S1)
holds(at(store), S1)

holds(at(home), S0)
holds(color(door, red), S0)

My boldface – JJB

Frame problem

- I go from home to the store, creating a new situation S’. In S’:
  - My friend is still at home
  - The store still sells chips
  - My age is still the same
  - Los Angeles is still the largest city in California…
- **How can we efficiently represent everything that hasn’t changed?**

Successor state axioms

- Normally, things stay true from one state to the next -- unless an action changes them:

holds(at(X),result(A,S)) iff A = go(X)
or [holds(at(X),S) and A != go(Y)]

- We need one or more of these for every fluent.

- Now we can use theorem proving to deduce a plan.
- Class dismissed!

Well, not quite..

- Theorem proving can be really inefficient for planning
- How do we handle concurrent events? uncertainty? metric time? preferences about plans? …

Strips (Fikes and Nilsson 71)

- For efficiency, separates theorem-proving within a world state from searching the space of possible states

- Highly influential **representation for actions**:
  - **Preconditions** (list of propositions to be true)
  - **Delete list** (list of propositions that will *become* false)
  - **Add list** (list of propositions that will *become* true)

} These two together are the action!

My boldface – important terms. Others you might want: **Production** (precondition⇒action pairs), **guarding** (what preconditions do for actions) JJB

Example problem:

Initial state: at(home), ¬ have(beer), ¬ have(chips)

Goal: have(beer), have(chips), at(home)

Actions:

Buy (X):
Pre: at(store)
Add: have(X)

Go (X, Y):
Pre: at(X)
Del: at(X)
Add: at(Y)

Frame problem (again)

- I go from home to the store, creating a new situation S'. In S':
  - The store still sells chips
  - My age is still the same
  - Los Angeles is still the largest city in California…
- How can we efficiently represent everything that hasn't changed?
  - Strips provides a good solution for simple actions

Ramification problem

- I go from home to the store, creating a new situation S'. In S':
  - I am now in Marina del Rey
  - The number of people in the store went up by 1
  - The contents of my pockets are now in the store..
- Do we want to say all that in the action definition?

Satisficing

Formal systems often assumed to be completely, logically, provably correct, but all AI requires design & abstraction decisions. – JJB

Solutions to the ramification problem

- In Strips, some facts are inferred within a world state,
  - e.g. the number of people in the store
- ‘primitive’ facts, e.g. at(home) persist between states unless changed. ‘inferred’ facts are not carried over and must be re-inferred.
  - Avoids making mistakes, perhaps inefficient.

This teeny tiny line about “inefficiency” is the entire difference between formal planning and reactive / dynamic systems AI. Efficiency should also be optimised, sensing may beat planning. – JJB

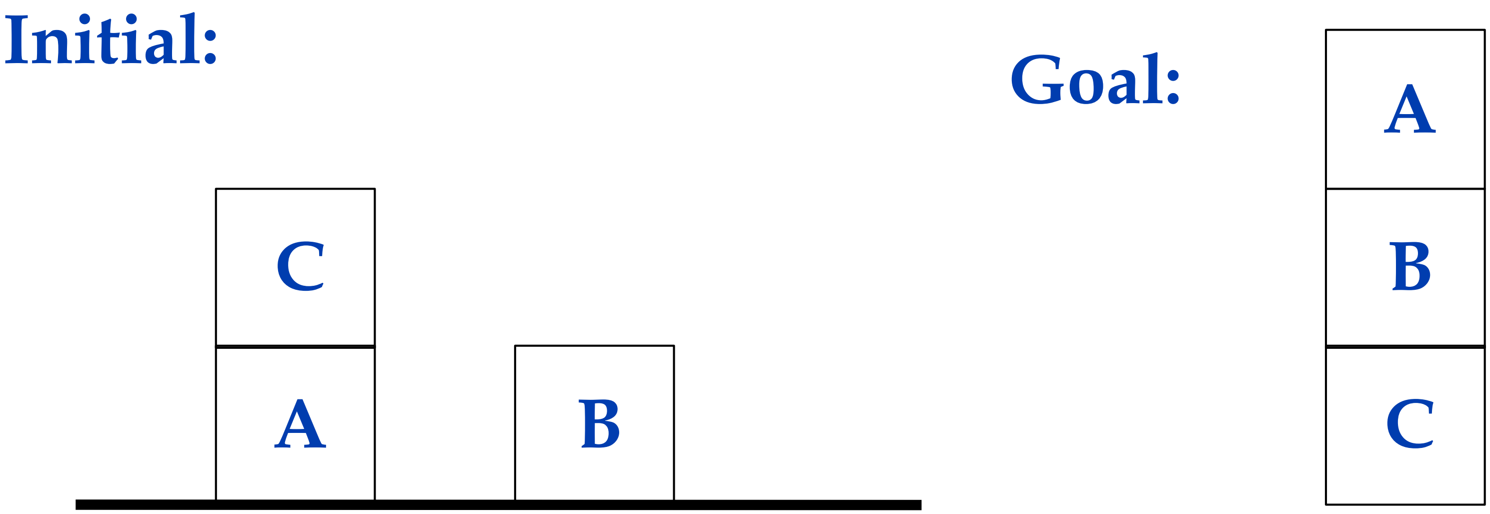
Questions about Strips

- What would happen if the order of goals was
  at(home), have(beer), have(chips) ?

- When Strips returns a plan, is it always correct? efficient?

- Can Strips always find a plan if there is one?

Example: blocks world (Sussman anomaly)



State I: (on-table A) (on C A) (on-table B) (clear B) (clear C)

Goal: (on A B) (on B C)

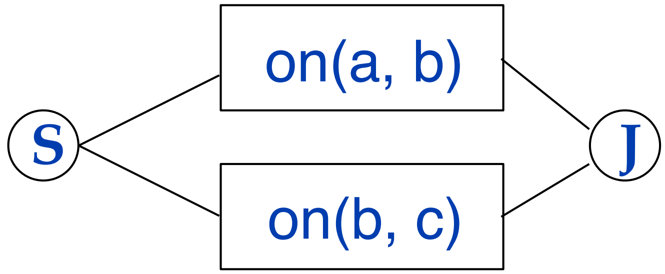
Pursuing either subgoal gets you stuck!

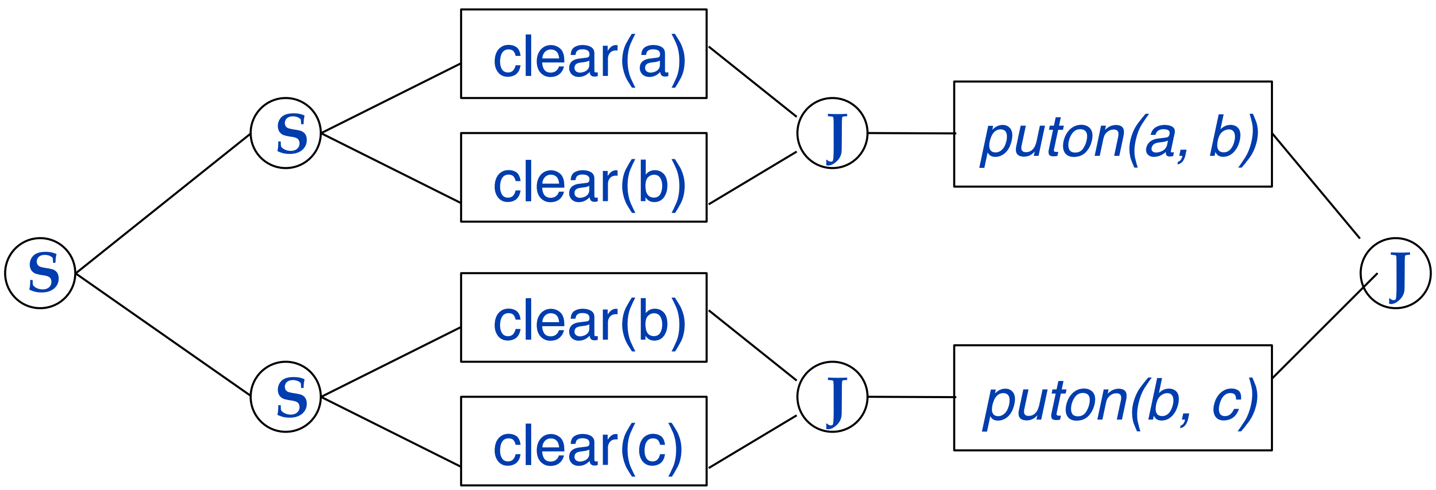
Noah (Sacerdoti 75)

- Explicitly views plans as a partial order of steps. Add ordering into the plan as needed to guarantee it will succeed.
- Avoids the problem in Strips, that focussing on one subgoal forces the actions that resolve that goal to be contiguous.

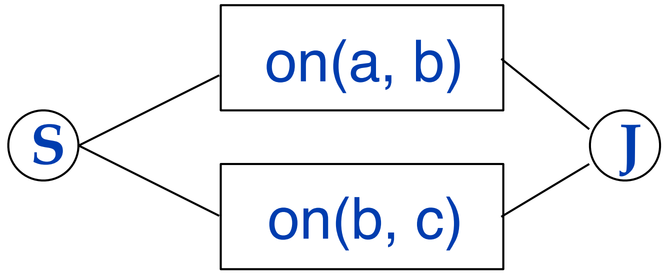
Translation: You can hack around this...

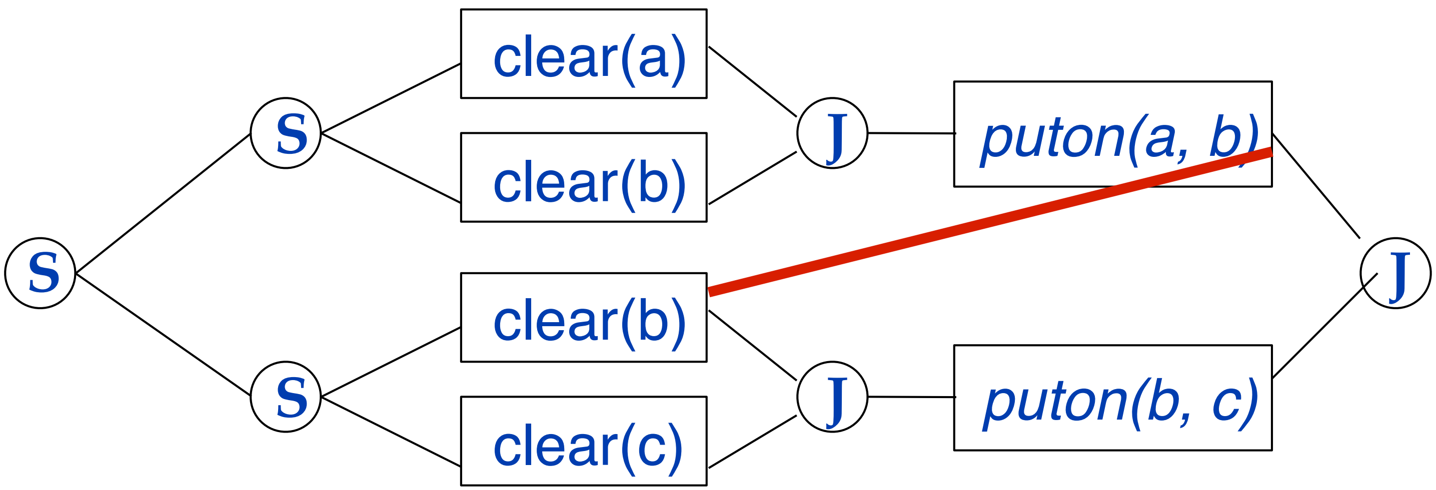
Nets Of Action Hierarchies



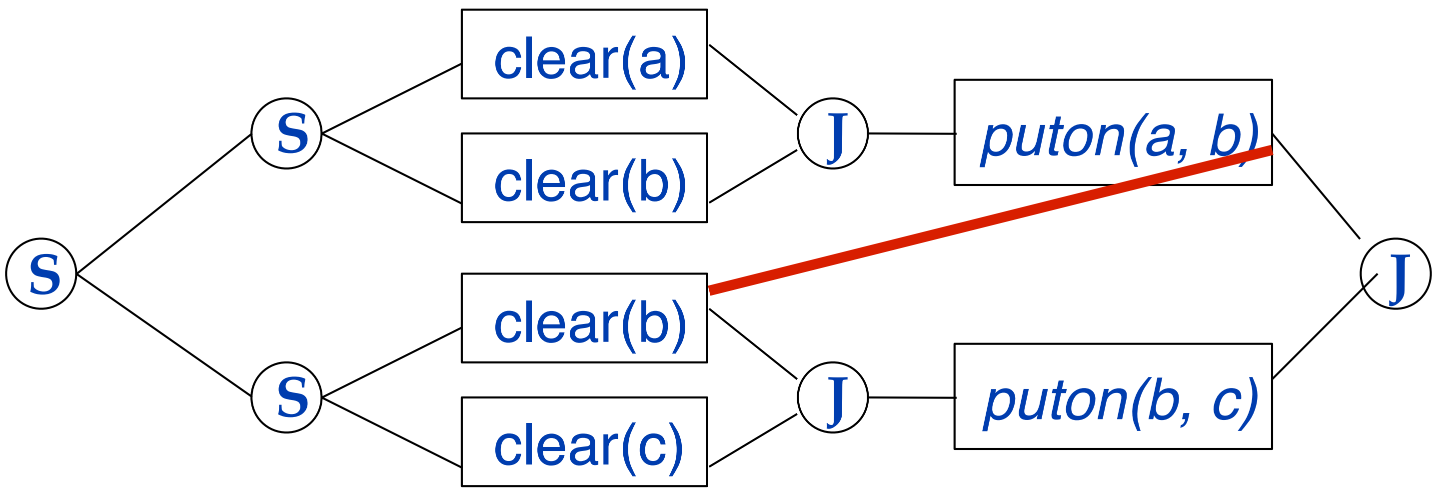


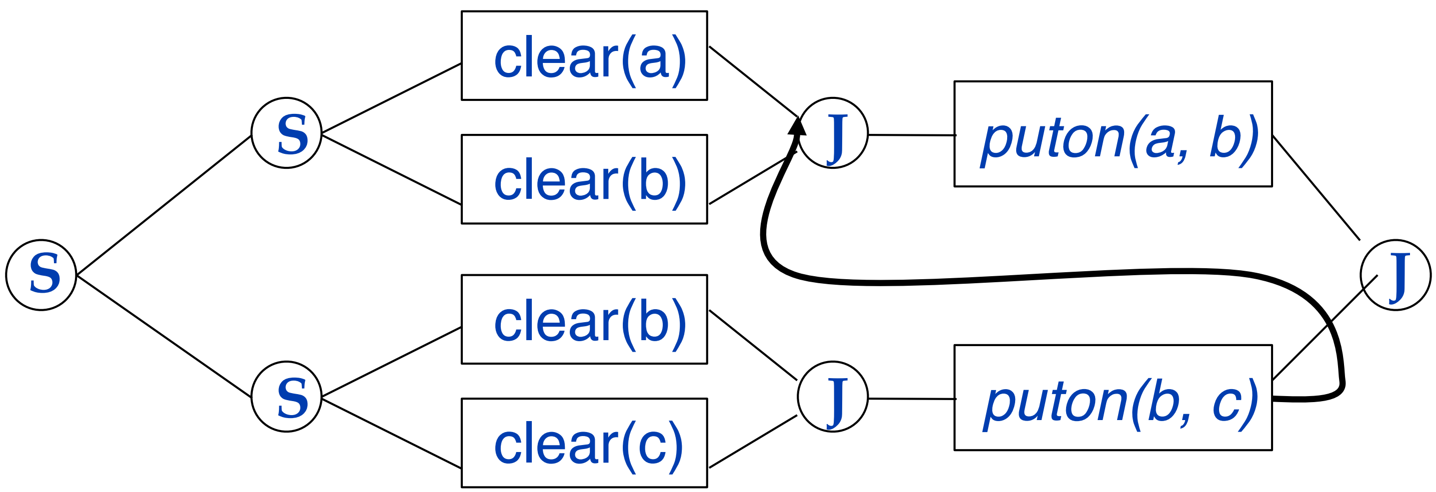
Nets Of Action Hierarchies

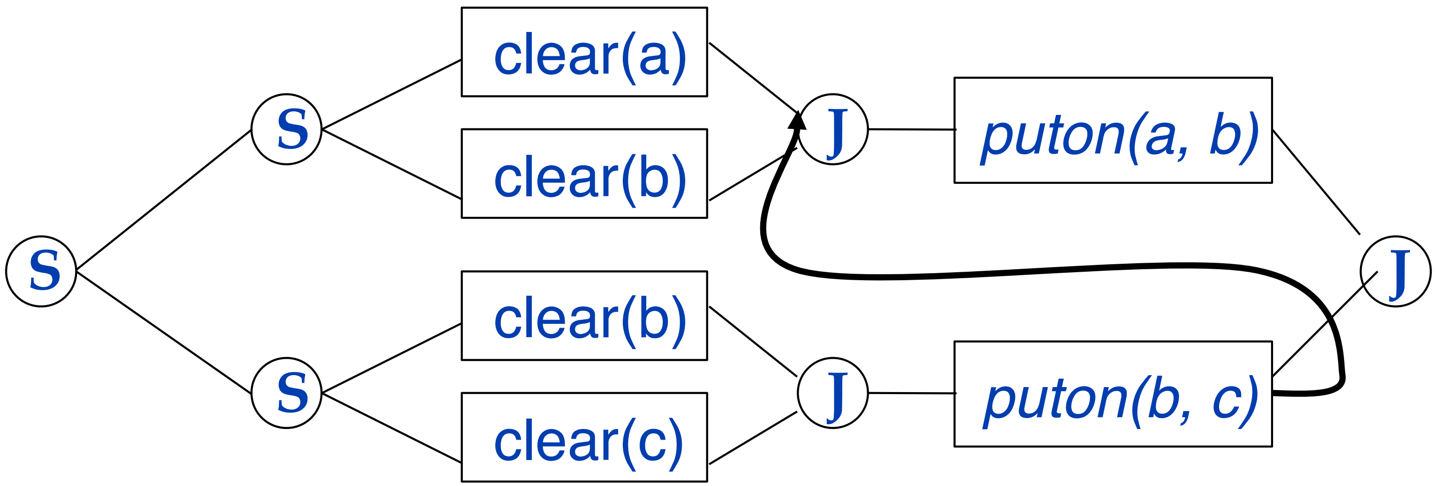


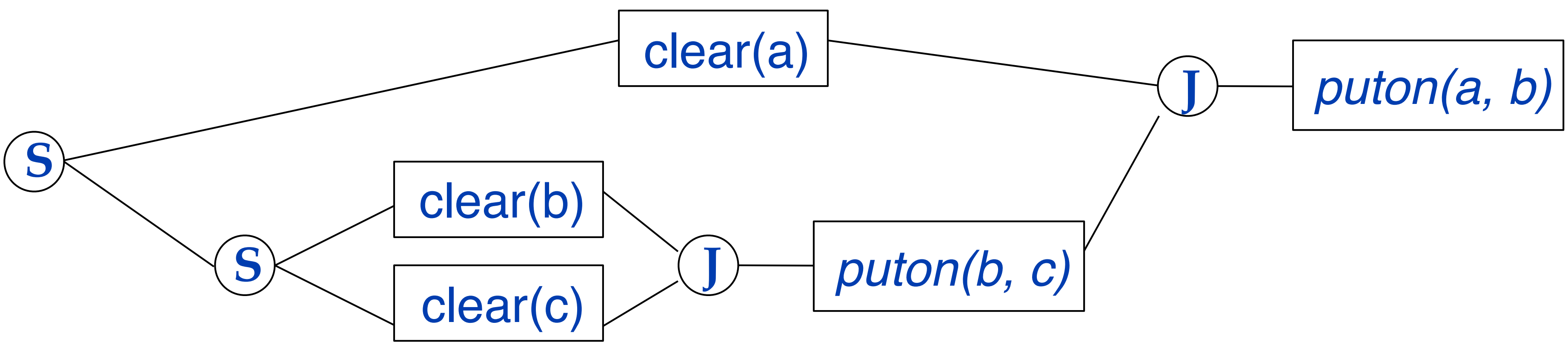


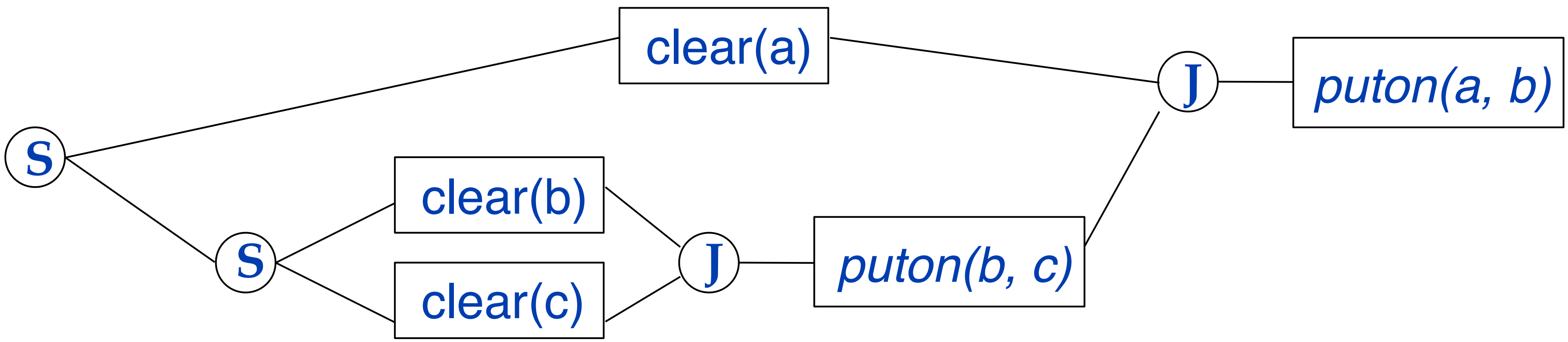
Resolve conflicts 'critic':

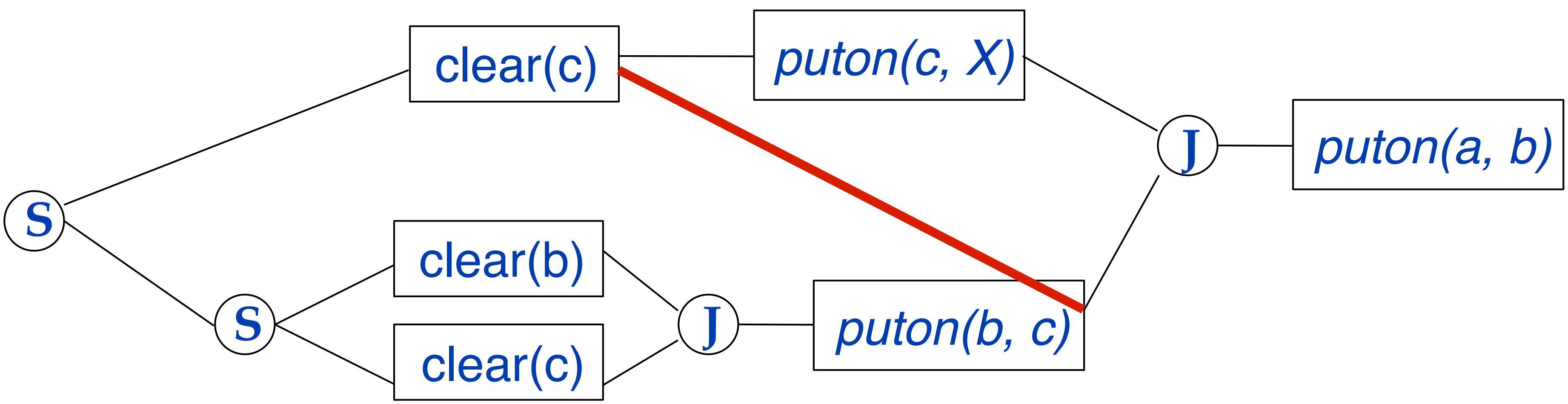




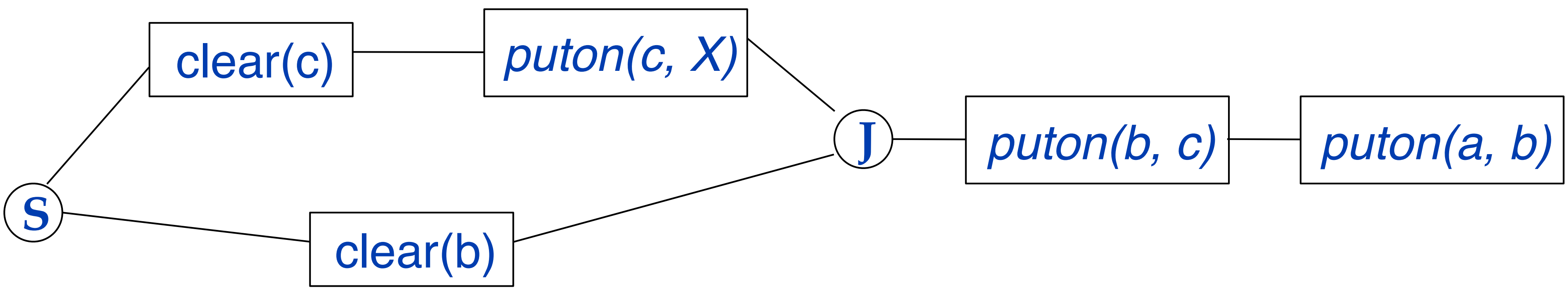

S
S
S
clear(a)
clear(b)
clear(b)
clear(c)
J
J
puton(a, b)
puton(b, c)
J


S
clear(a)
S
clear(b)
clear(c)
J
puton(b, c)
J
puton(a, b)


S
clear(a)
J
puton(a, b)
S
clear(b)
clear(c)
J
puton(b, c)


clear(c)
puton(c, X)
J
puton(a, b)
S
S
clear(b)
clear(c)
J
puton(b, c)

Final plan



(Yeah, right) But anyway...

More Planning Concepts

- Forward chaining: start from world & look for goal.
- Backward chaining: start at goal, look back for current world.
  - Often combine these to somewhat limit combinatorics.
- Affordances: Perceptual system delivers set of possible actions with object ID.
- Robust vs brittle, graceful degradation.

More recent formal planning

- Temporal logics
- Non monotonic logics
- Answer set programming

Marina De Vos

But let’s go back to one of the first slides:

Generating plans

- Given:
  - A way to describe the world
  - An initial state of the world
  - A goal description
  - A set of possible actions to change the world

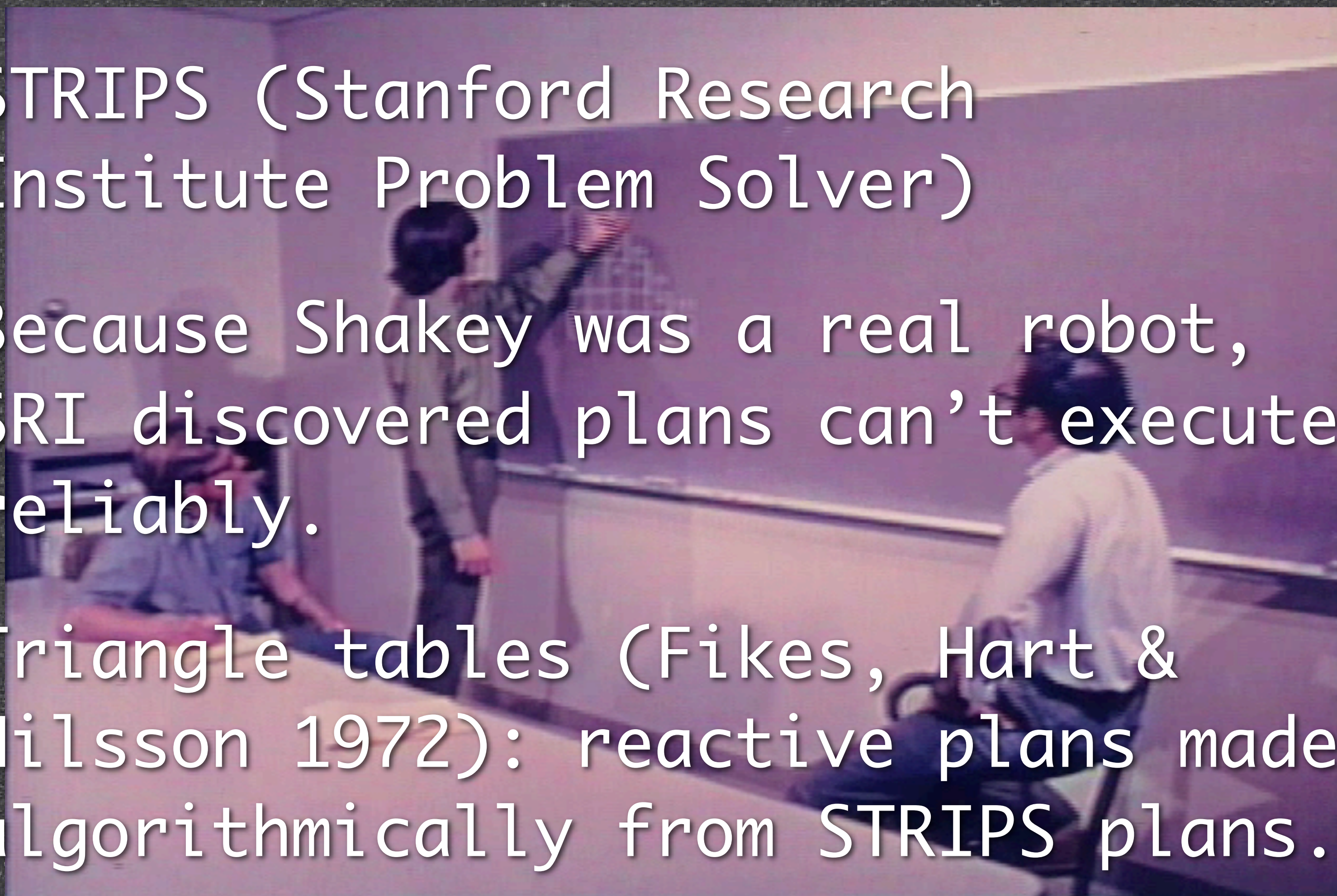
} **This part is well nigh impossible.**

- Find:
  - A prescription for actions to change the initial state into one that satisfies the goal

Describing the world in ways that can be sensed is the hard part, whatever planning approach you take.

The Transition from Productive to Reactive Planning

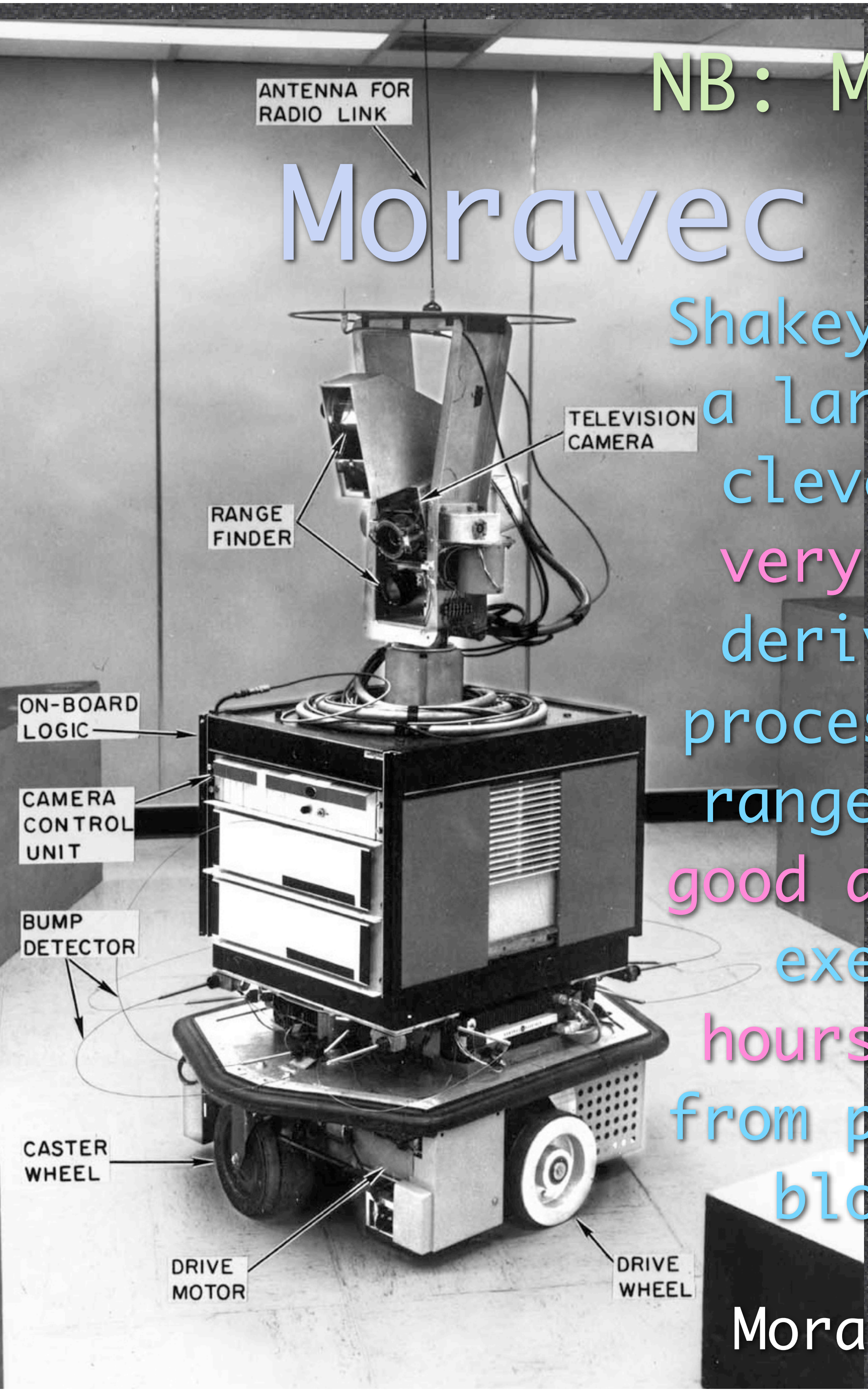
Shakey, STRIPS & Triangle Tables



- STRIPS (Stanford Research Institute Problem Solver)
- Because Shakey was a real robot, SRI discovered plans can't execute reliably.
- Triangle tables (Fikes, Hart & Nilsson 1972): reactive plans made algorithmically from STRIPS plans.

NB: Moravec worked at CMU

Moravec on Shakey



Shakey was remote controlled by a large computer. It hosted a clever reasoning program fed very selective spatial data, derived from weak edge-based processing of camera and laser range measurements. On a very good day it could formulate and execute, over a period of hours, plans involving moving from place to place and pushing blocks to achieve a goal.

Moravec (1998), ROBOT, page 27.

Perception versus On-Line Reasoning

- Brooks (1986) “The world is its own best model.”
- Shakey did update its model (SRI found they had to) but it took minutes to process a single frame.
  - cost / benefit tradeoffs of reasoning vs perceiving were different then.

cf. Richard Gregory

From Planning to Systems AI

Manuela Veloso (CMU) started in formal planning & MAS, thought she should be able to solve RoboCup football, couldn’t, added systems AI and machine learning, won every RoboCup league.

Strategies of Action Selection / Outline

- Productions
- Formal / Optimal Planning
- Reactive / Dynamic Plans
- Learning Plans

Reactive Planning

- Reactive planning is an oxymoron.
  - It means “action selection by look up”, but planning had become synonymous with action selection.
- Now conferences about proactive AI.
  - Attempted rebranding: Dynamic Planning (hasn’t caught on yet).

What are Plans For?

- Plans as communication (Agre & Chapman 1989).
  - Parsing semantic content from gamer communication (e.g. “uh”).
- “Plans are worthless, but planning is everything” – Eisenhower
  - Plans need to be adaptable to the unforeseen.

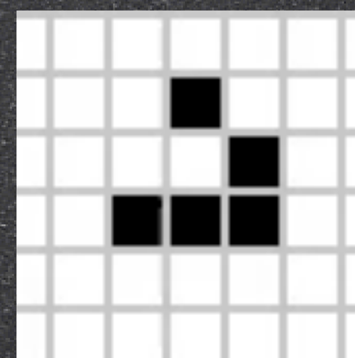
Three Methods of Dynamic Planning

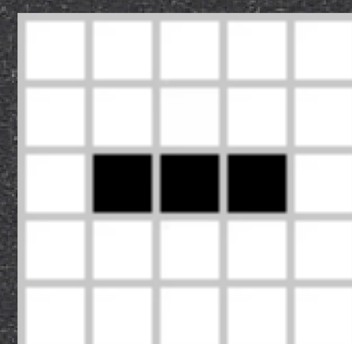
- Environmental Determinism
- Finite State Machines
- Basic Reactive Plans

(Bryson, *Agent*, 2003)

Environmental Determinism

- Figure out a way to recognise all possible / relevant states of the world.
- Say what to do in each one.

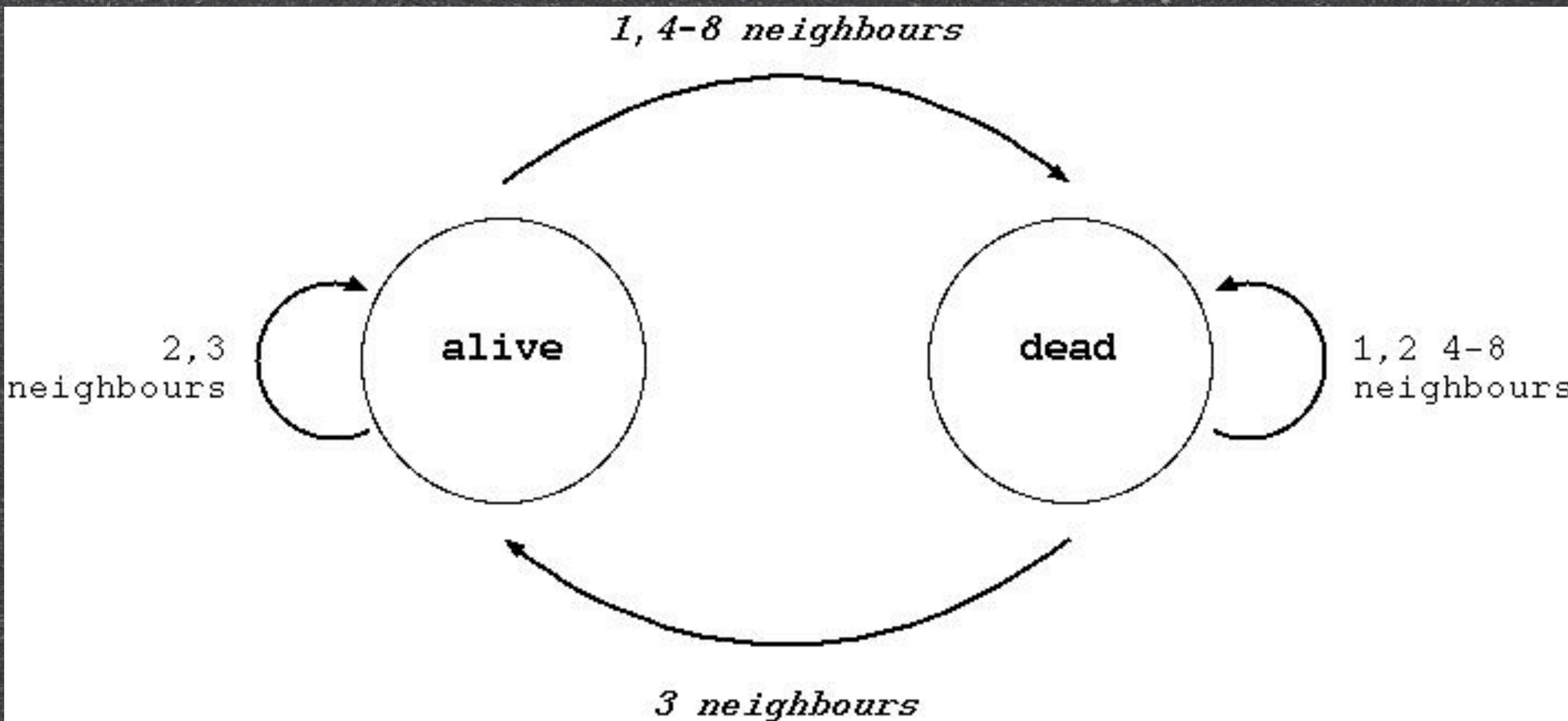




0-1	2	3	4-8
die	stay	be-born	die

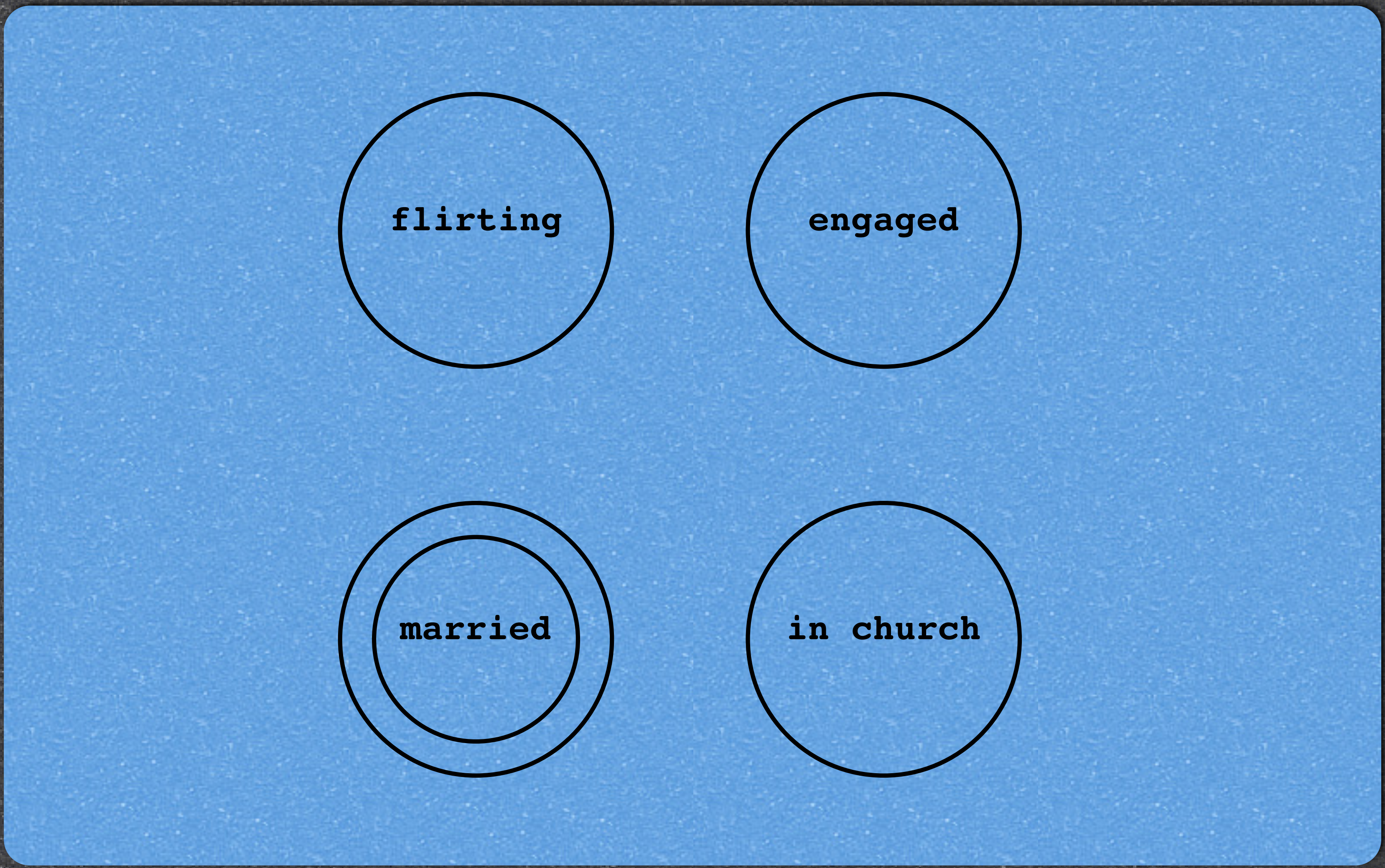
Conway's Life: # of neighbours

Finite State Machine



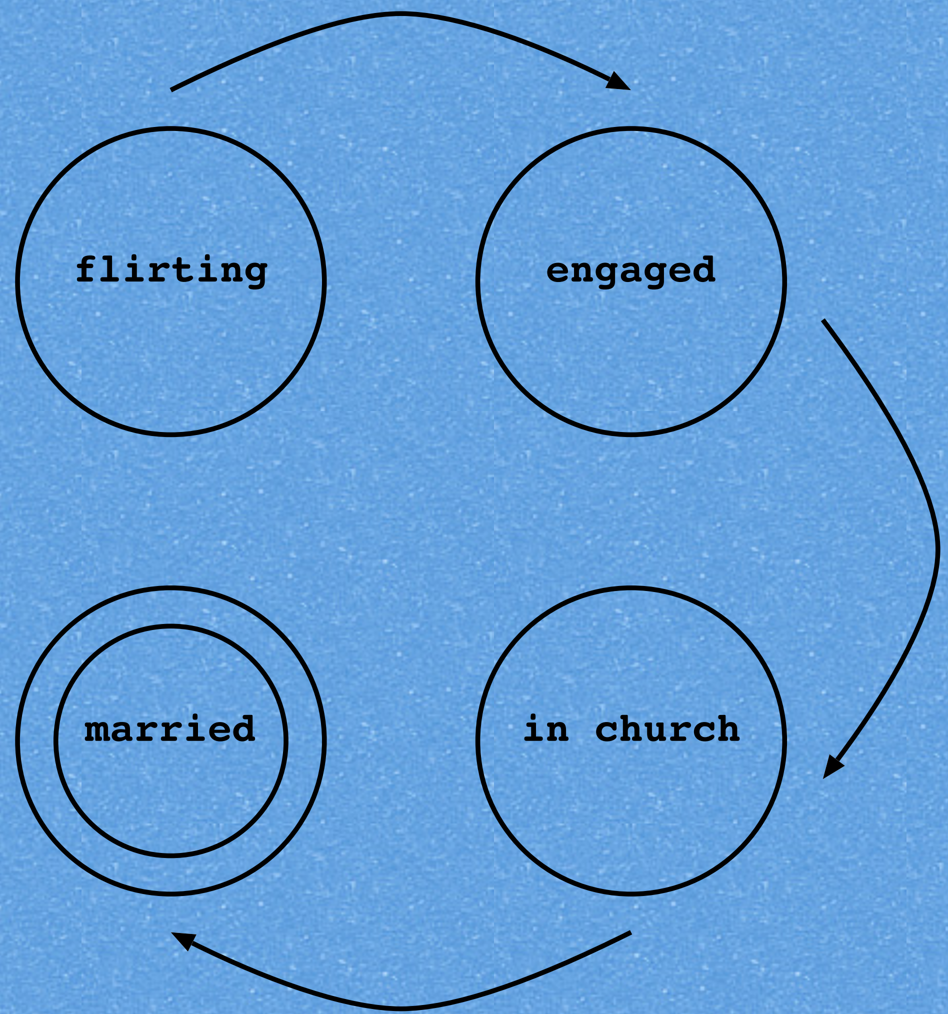
Conway's Life

Finite State Machine



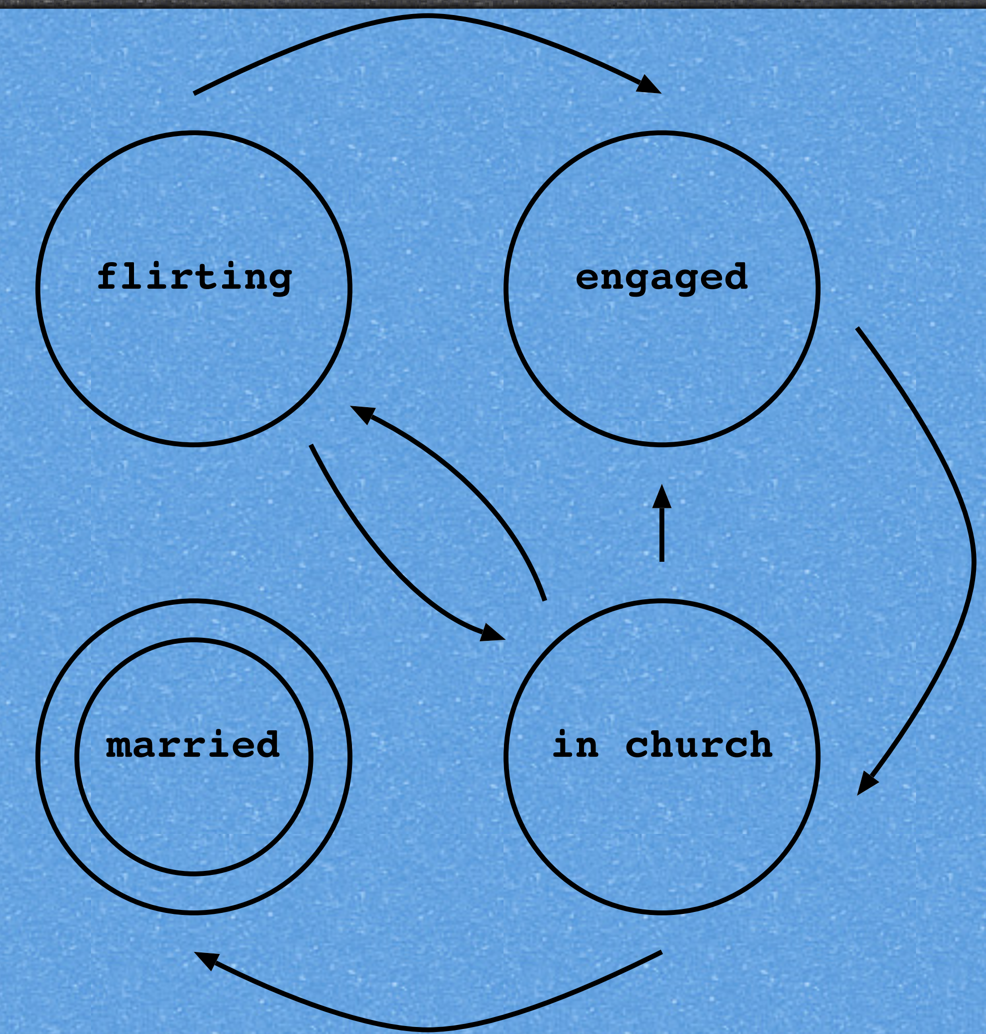
Human-like Behaviour (Austen)

Finite State Machine



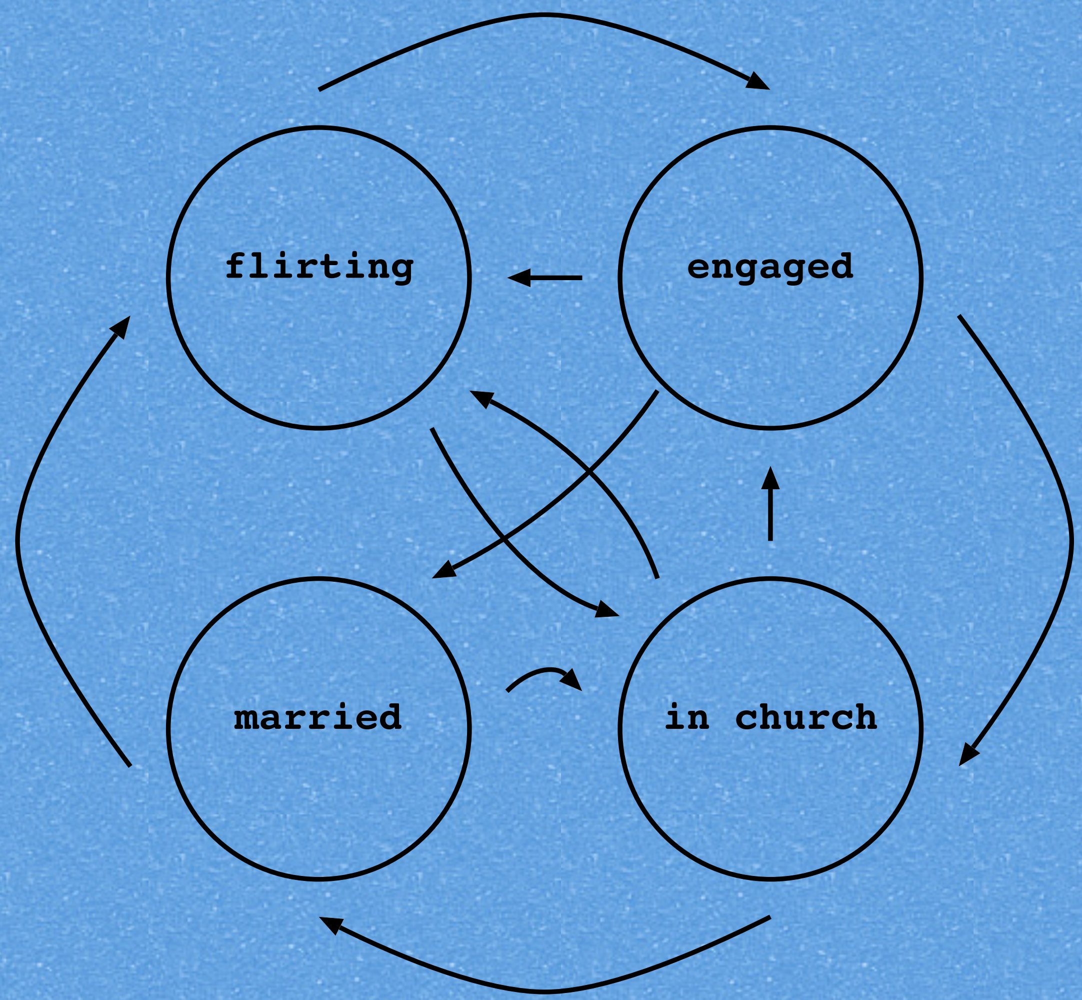
Human-like Behaviour (Austen)

Finite State Machine



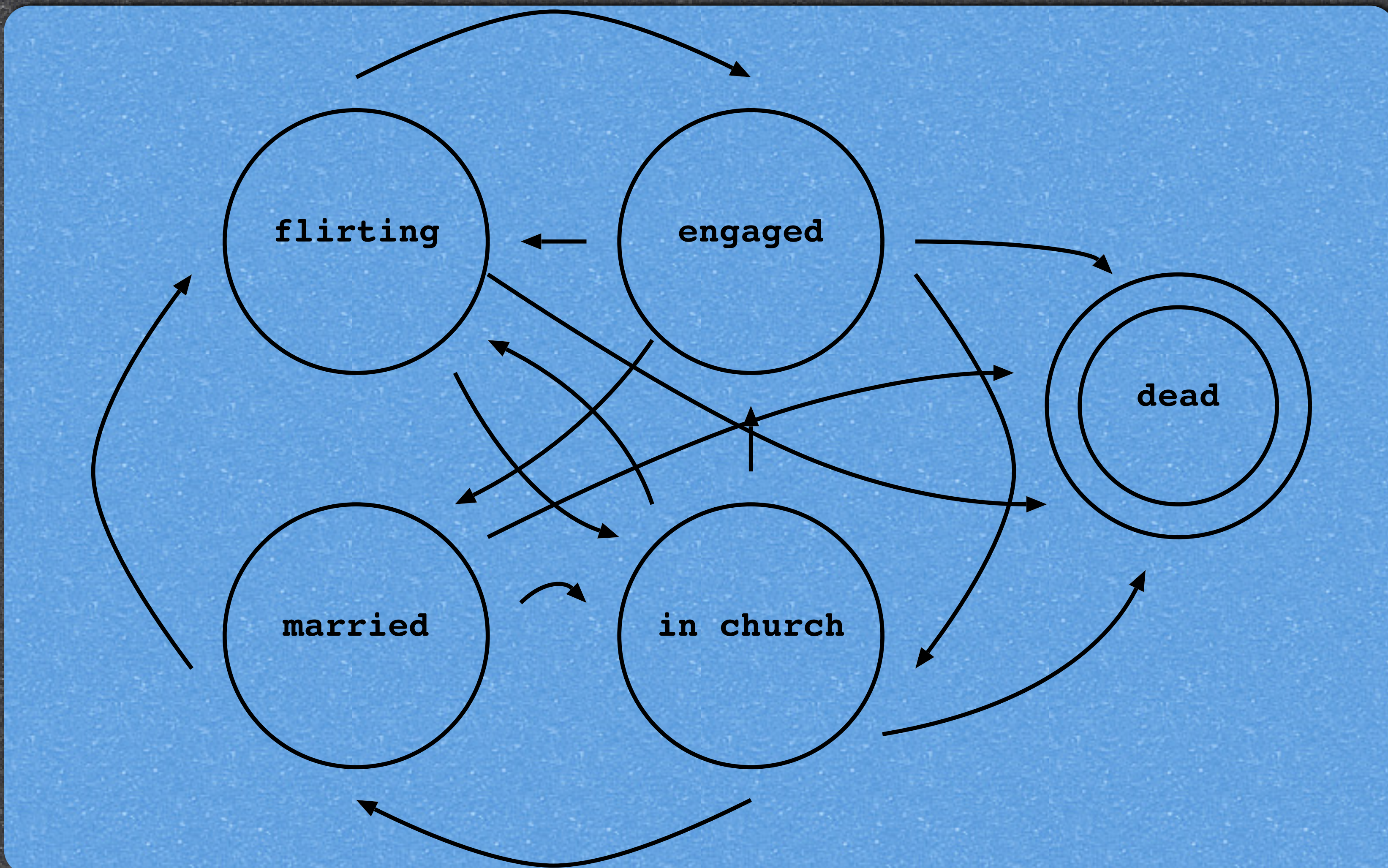
Human-like Behaviour (Austen)

Finite State Machine



Human-like Behaviour (Austen)

Finite State Machine



Human-like Behaviour (Austen)

FSM vs AI

- Prefer not to specify "actions" that the world will take for itself.
  - Not always possible in VR, but more likely in robotics.
- Want to focus on intentional goals, but to be able to handle contingencies.

Basic Reactive Plans

$$\left\langle \begin{array}{r} \text{(fiancé here \& in church)} \Rightarrow \text{marry} \\ \text{(fiancé here)} \Rightarrow \text{goto church} \\ \text{(engaged)} \Rightarrow \text{goto fiancé} \\ \text{(receiving attention)} \Rightarrow \text{become engaged} \\ \text{()} \Rightarrow \text{flirt} \end{array} \right\rangle$$

- Prioritised list of actions converging to a goal, each guarded by its environmental context requirement.
- STRIPS Triangle tables (became Nilsson's teleo-reactive plans) one example.

Basic Reactive Plans

$$\uparrow \left\langle \begin{array}{rcl} (\text{fiancé here \& in church}) & \Rightarrow & \text{marry} \\ (\text{fiancé here}) & \Rightarrow & \text{goto church} \\ (\text{engaged}) & \Rightarrow & \text{goto fiancé} \\ (\text{receiving attention}) & \Rightarrow & \text{become engaged} \\ () & \Rightarrow & \text{flirt} \end{array} \right\rangle$$

- Exploit representations & insights of earlier AI planning, e.g. preconditions
- But reactive – pre-programmed, very little real-time search.

Strategies of Action Selection / Outline

- Productions
- Formal / Optimal Planning
- Reactive / Dynamic Plans
- Learning Plans

Learning is another form of search

- Evolve plans.
- Learn by observation.
  - Create Markov model of knowledgeable agent’s actions.
  - Use Markov model as a reactive plan.

e.g. Matt Brand

Relevance for Robots

(interactive)

- What are the environmental conditions you can discriminate?
- What are the conditions you need to discriminate for action?
    - How certain are you that you are in a state?
    - Can you increase that certainty? or act robustly?

Summary

- “Real” (productive) planning is intractable.
  - But we know we do it, probably over limited search spaces.
- Reactive planning is efficient, but requires planning in advance.
  - Programming, learning, even productive planning (maybe).