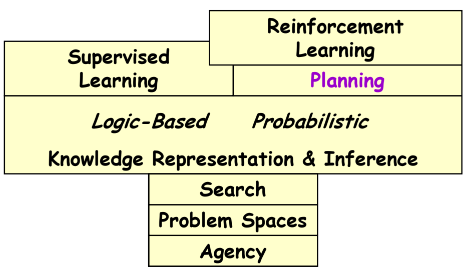
Knowledge Representation II

CSE 573

Logistics

- Reading for Monday
  ???
- Office Hours
  No Office Hour Next Monday (10/25)
  Bonus Office Hour: Today 3-4
  - Or email me

573 Topics



Ways to make "plans"

Generative Planning
  Reason from first principles (knowledge of actions)
  Requires formal model of actions
Case-Based Planning
  Retrieve old plan which worked on similar problem
  Revise retrieved plan for this problem
Reinforcement Learning
  Act "randomly" - noticing effects
  Learn reward, action models, policy

Generative Planning

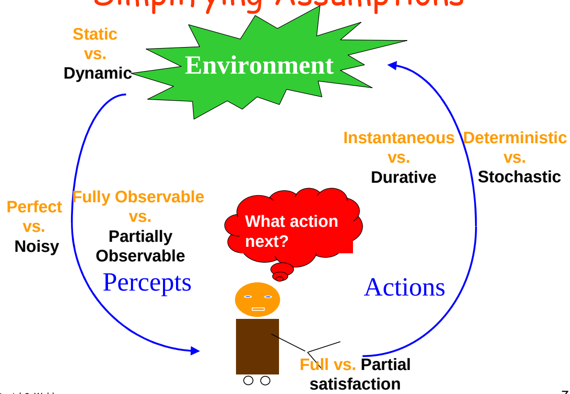
Input
  Description of (initial state of) world *(in some KR)*
  Description of goal *(in some KR)*
  Description of available actions *(in some KR)*

Output
  Controller
    E.g. Sequence of actions
    E.g. Plan with loops and conditionals
    E.g. Policy = f: states -> actions

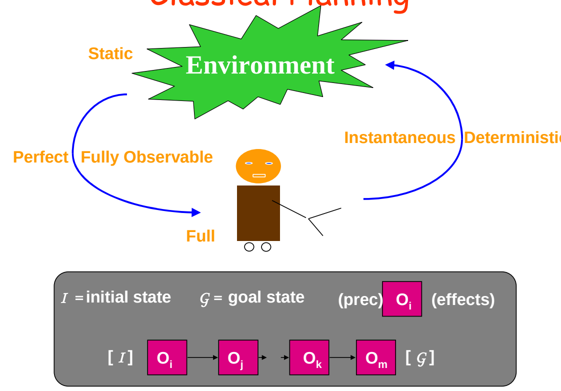
Input Representation

- *Description of initial state of world*
  E.g., Set of propositions:
  ((block a) (block b) (block c) (on-table a) (on-table b) (clear a) (clear b) (clear c) (arm-empty))
- *Description of goal: i.e. set of worlds or ??*
  E.g., Logical conjunction
  Any world satisfying conjunction is a goal
  (and (on a b) (on b c)))
- *Description of available actions*

Simplifying Assumptions



Classical Planning

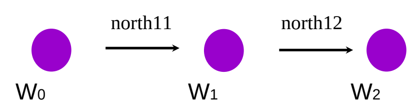


Planning Outline

- The planning problem
- Representation
- Compilation to SAT
- Searching world states
  - Regression
  - Heuristics
- Graphplan
- Reachability analysis & heuristics
- Planning under uncertainty

How Represent Actions?

- ***Simplifying assumptions***
  - Atomic time
  - Agent is omniscient (no sensing necessary).
  - Agent is sole cause of change
  - Actions have deterministic effects
- ***STRIPS representation***
  - World = set of true propositions
  - Actions:
    - Precondition: (conjunction of literals)
    - Effects (conjunction of literals)



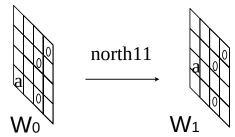
How Represent Actions?

- ***Simplifying assumptions***
  - Atomic time
  - Agent is omniscient (no sensing necessary).
  - Agent is sole cause of change
  - Actions have deterministic effects
- ***STRIPS representation***
  - World = set of true propositions
  - Actions:
    - Precondition: (conjunction of literals)
    - Effects (conjunction of literals)

How Encode STRIPS → Logic ?

Time in STRIPS Representation

- Action = function: worldState→ worldState
- Precondition
  says where function defined
- Effects
  say how to change set of propositions



Note: strips doesn't allow derived effects; you must be complete!

north11
precond: (and (agent-at 1 1)
(agent-facing north))

effect: (and (agent-at 1 2)
(not (agent-at 1 1)))

Action Schemata

- Instead of defining:
  pickup-A and **pickup-B** and …
- Define a schema:

```
(:operator pick-up
    :parameters ((block ?ob1))
    :precondition (and (clear ?ob1)
                       (on-table ?ob1)
                       (arm-empty))
    :effect (and (not (clear ?ob1))
                 (not (on-table ?ob1))
                 (not (arm-empty))
                 (holding ?ob1)))
```

Note: strips doesn't allow derived effects; you must be complete!

Time Arguments in Logic

Initial Conditions

On(a, b, 0)
Have(bluePaint, 0)
Red(a, 0)

Goal

On(b,a, ?)
Blue(a, ?)

Closed World Assumption

Preconditions & Effects

- If action is executed at time t

Paint(a, blue, t) => Have(bluePaint, t-1)

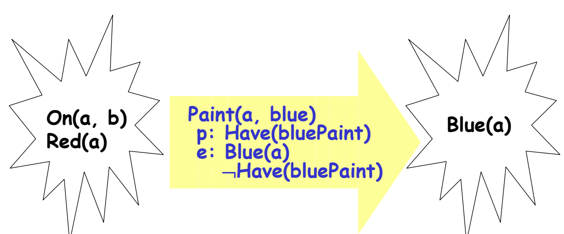
Paint(a, blue, t) => Blue(a, t+1) ∧ ¬Have(bluePaint, t+1)

Propositions: even
Actions: odd

Paint(a, blue, t)
p: Have(bluePaint, t-1)
e: Blue(a, t+1)
¬Have(bluePaint, t+1)

Issues

- Frame problem
- Ramification problem
- Qualification problem



Compilation to SAT

- Init state
- Actions
- Goal

→ ?

The Idea

- Suppose a plan of length n exists
- Encode this hypothesis in SAT
  - Init state true at t_0
  - Goal true at T_n
  - Actions imply effects, etc
- Look for **satisfying** assignment
- Decode into plan

RISC: The Revolutionary Excitement

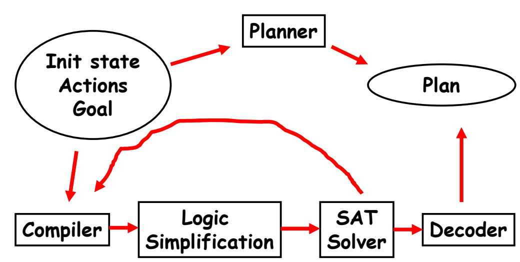
History

- Green IJCAI-69
- STRIPS AIJ-71
- Decades of work on "specialized theorem provers"
- Kautz+Selman ECAI-92
- Rapid progress on SAT solving
- Kautz+Selman AAAI-96
  - Electrifying results (on hand coded formulae)
- Kautz, McAllester & Selman KR-96
  - Variety of encodings (but no compiler)
- CSE 573 => Ernst *et al.* IJCAI-97

Blackbox

- Blackbox solves planning problems by converting them into SAT.
  - Very fast
  - Initially hand copiled SAT; later…
  - Tried different solvers
    - Local search (GSAT)
    - Systematic search with EBL (RelSAT)
- In 2000, GP-CSP could beat Blackbox
  - But in 2001, a newer "SUPER-DUPER" SAT solver called CHAFF was developed,
  - CSP people are trying to copy over the ideas from CHAFF to CSP.
- In 2004, Blackbox…

Medic



Axioms

Axiom	Description / Example
Init	The initial state holds at t=0
Goal	The goal holds at t=2n
$A \Rightarrow P, E$	Paint(A,Red,t) $\Rightarrow$ Block(A, t-1) Paint(A,Red,t) $\Rightarrow$ Color(A, Red, t+1)
Frame	
At-least-one	
Exclude	

Space of Encodings

- Action Representations
  - Regular
  - Simplyu-Split
  - Overloaded-Split
  - Bitwise
- Frame Axioms
  - Classical
  - Explanitory

Frame Axioms

- Classical
  ∀P, A, t if P@t-1 ∧
  A@t ∧
  A doesn't affect P
  then P@t+1
- Explanatory

Action Representation

Representation	One Propositional Variable per	Example
Regular	fully-instantiated action	Paint-A-Red, Paint-A-Blue, Move-A-Table
Simply-split	fully-instantiated action's argument	Paint-Arg1-A ∧ Paint-Arg2-Red
Overloaded-split	fully-instantiated argument	Act-Paint ∧ Arg1-A ∧ Arg2-Red
Bitwise	Binary encodings of actions	Bit1 ∧ ~Bit2 ∧ Bit3

more vars ↕ more clses

Paint-A-Red = 5

Main Ideas

- Clear taxonomy
- Utility of
  Explanatory frame axioms *(most things don't change)*
  Parallelism & conflict exclusion
  Type inference
  Domain axioms
- Surprising
  Effectiveness of regular action encodings

Comparison Among Encodings

- **Explanatory Frames beat classical**
  - few actions affect each fluent
  - explanatory frames aid simplifications
- **Parallelism is a major factor**
  - fewer mutual exclusion clauses
  - fewer time steps
- ***Regular actions representation is smallest!***
  - exploits full parallelism
  - aids simplification
- **Overloaded, bitwise reps. are infeasible**
  - prohibitively many clauses
  - sharing hinders simplification

Optimization 1: Factored Splitting

- use partially-instantiated actions

HasColor-A-Blue-(t-1) ^ Paint-Arg1-B-t ^
Paint-Arg2-Red-t ⇒ HasColor-A-Blue-(t+1)

Explanatory Frames

factored / unfactored	Simple	Overloaded
Variables	.46	.69
Clauses	.30	.50
Literals	.20	.38

Optimization 2: Types

A *type* is a fluent which no actions affects.
- type interference
- prune impossible operator instantiations
- type elimination

Type opts / No type opts				
Literals	Regular	Simple	Overloaded	Bitwise
Classical	.27	.39	.34	.30
Explanatory	.10	.97	.67	.74

Domain-Specific Axioms

Adding domain-specific axioms
increases clauses
decreases variables
decreases solve time dramatically.

domain info
no domain info

bw-large	Vars	Clauses	Time
a	.86	1.53	.26
b	.88	1.84	.38
c	.86	2.24	<.05

Future Work

- Negation, disjunctive preconds, ∀
- Domain axioms

  ∀t clear(x, t) ≡ ¬ ∃y on(y, x, t)

Future Work

- Automatically choose best encoding
  - Might do this for frame axioms
- Analyze SAT formulae structure
  - Generate WalkSAT params
  - Which SAT solver works best (DPLL *vs* ?
- Handle continuous vars (resource planning)
  - Steve Wolfman's quals project, IJCAI99

Future Work

- Reachability analysis
- Domain axioms
- Compilation to …?
  - CSP
  - LP (Linear programming)
  - Integer LP
  - SAT + LP

Domain Axioms

- Domain knowledge
  - Synchronic *vs.* Diachronic constraints
- Speedup knowledge
  - Action conflicts (=> by action schemata alone)
  - Domain invariants (=> by initial state+schemata)
  - Optimality heuristics
  - Simplifying assumptions

