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

- Agencies
- Problem Spaces
- Search
- Knowledge Representation & Inference
- Supervised Learning
- Reinforcement Learning
- Planning
- Case-Based Planning
  - Retrieve old plan which worked on similar problem
  - Revise retrieved plan for this problem
- Logic-Based Probabilistic
- Reinforcement Learning
  - Act "randomly" - noticing effects
  - Learn reward, action models, policy

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 \(\rightarrow\) 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
### 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 Encode STRIPS → Logic?

- The 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)
Time in STRIPS Representation

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

```
north11
precond: (and (agent-at 1 1) (agent-facing north))
effect: (and (agent-at 1 2) (not (agent-at 1 1)))
```

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

Action Schemata

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

```
(:operator pickup
  :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)
```

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 t₀
  - Goal true at T₀
  - 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 coded 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

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Description / Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init</td>
<td>The initial state holds at t=0</td>
</tr>
<tr>
<td>Goal</td>
<td>The goal holds at t=2n</td>
</tr>
<tr>
<td>A ⇒ P, E</td>
<td>Paint(A, Red, t) ⇒ Block(A, t-1)</td>
</tr>
<tr>
<td></td>
<td>Paint(A, Red, t) ⇒ Color(A, Red, t+1)</td>
</tr>
<tr>
<td>Frame</td>
<td>At-least-one</td>
</tr>
<tr>
<td>Exclude</td>
<td></td>
</tr>
</tbody>
</table>

Space of Encodings

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

- Classical
  \[ \forall P, A, t \text{ if } P(t-1) \land A(t) \land A \text{ doesn't affect } P \text{ then } P(t+1) \]

- Explanatory

Explanatory Frame Axioms

- Classical
  \[ \forall P, A, t \text{ if } P(t-1) \land A(t) \land A \text{ doesn't affect } P \text{ then } P(t+1) \]

- Explanatory

Action Representation

<table>
<thead>
<tr>
<th>Representation</th>
<th>One Propositional Variable per</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simply-split</td>
<td>fully-instantiated action's argument</td>
<td>Paint-Arg1-A, Paint-Arg2-Red</td>
</tr>
<tr>
<td>Overloaded-split</td>
<td>fully-instantiated argument</td>
<td>Act-Paint \land Arg1-A \land Arg2-Red</td>
</tr>
<tr>
<td>Bitwise</td>
<td>Binary encodings of actions</td>
<td>Bit1 \land \neg Bit2 \land Bit3</td>
</tr>
</tbody>
</table>

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

  \[ \text{HasColor-A-Blue-(t-1)} \land \text{Paint-Arg1-B-t} \land \text{Paint-Arg2-Red-t} \Rightarrow \text{HasColor-A-Blue-(t+1)} \]

Optimization 2: Types

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

<table>
<thead>
<tr>
<th>Type opts</th>
<th>No type opts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Literals</td>
<td></td>
</tr>
<tr>
<td>Classical</td>
<td>.27</td>
</tr>
<tr>
<td>Explanatory</td>
<td>.10</td>
</tr>
<tr>
<td>Overloaded</td>
<td>.34</td>
</tr>
<tr>
<td>Bitwise</td>
<td>.30</td>
</tr>
<tr>
<td>Bitwise</td>
<td>.97</td>
</tr>
<tr>
<td>Bitwise</td>
<td>.67</td>
</tr>
<tr>
<td>Bitwise</td>
<td>.74</td>
</tr>
</tbody>
</table>
Domain-Specific Axioms

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

<table>
<thead>
<tr>
<th>domain info</th>
<th>Vars</th>
<th>Clauses</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>no domain info</td>
<td>a</td>
<td>.86</td>
<td>1.53</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>.88</td>
<td>1.84</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>.86</td>
<td>2.24</td>
</tr>
</tbody>
</table>

Future Work

- Negation, disjunctive preconds, ∀
- Domain axioms
  \[ \forall t \text{ clear}(x, t) \equiv \neg \exists y \text{ on}(y, x, t) \]

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

Domain Axioms

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