Lecture 19 of 42

Knowledge Representation and Midterm Review
Discussion: Search, Inference, Planning

William H. Hsu
Department of Computing and Information Sciences, KSU

KSOL course page: http://snipurl.com/v9v3
Course web site: http://www.kddresearch.org/Courses/CIS730
Instructor home page: http://www.cis.ksu.edu/~bhsu

Reading for Next Class:
Review Chapters 1 – 10, Russell & Norvig 2nd edition

Lecture Outline

- Reading for Next Class: Review Chapters 1 - 10, R&N 2nd edition
- Last Class: Event and Fluent Calculi, CIKM
  - Representing time, events: from situation calculus to event, fluent calculi
  - Knowledge acquisition (KA) and capture
  - Computational information and knowledge management (CIKM)
- Today: Midterm Review
  - Section I: Intelligent Agents
  - Section II: Search
  - Section III: Knowledge and Reasoning
- Coming Week: Intro to Classical Planning
**Problem-Solving Agents: Review**

Restricted form of general agent:

```java
function SIMPLE-PROBLEM-SOLVING-AGENT( percept) returns an action
    static: seq, an action sequence, initially empty
    state, some description of the current world state
    goal, a goal, initially null
    problem, a problem formulation
    state ← UPDATE-STATE(state, percept)
    if seq is empty then
        goal ← FORMULATE-GOAL(state)
        problem ← FORMULATE-PROBLEM(state, goal)
        seq ← SEARCH( problem)
        action ← RECOMMENDATION(seq, state)
    seq ← REMAINDER(seq, state)
    return action
```

Note: this is offline problem solving; solution executed “eyes closed.”
Online problem solving involves acting without complete knowledge.

---

**State Space Graph: Review**

- **States**: integer dirt and robot locations (ignore dirt amounts etc.)
- **Actions**: Left, Right, Suck, NoOp
- **Goal Test**: no dirt
- **Path Cost**: 1 per action (0 for NoOp)

Based on slide © 2003 S. Russell & P. Norvig. Reused with permission.
**Greedy Search:**

**Review**

- **CLOSED List**
  - $\emptyset = \{\}$
- **OPEN List**
  - Adad
  - Arad
  - Sibiu
  - Fagaras
  - Bucharest

Path found: (Arad $\rightarrow$ Sibiu $\rightarrow$ Fagaras $\rightarrow$ Bucharest)$^{460}$

**Algorithm A/A**:  

**Review**

- Adad
- Bucharest
- Dnoeas
- Ooma
- Rimnicu Vilcea
- Pilestii
- Sibiu
- Timisoara
- Zerind

Nodes found/scheduled (opened): {A, S, T, Z, F, O, RV, S, B, C, P}

Nodes visited (closed): {A, S, F, RV, P, B}

Path found: (Arad $\rightarrow$ Sibiu $\rightarrow$ Rimnicu Vilcea $\rightarrow$ Pilestii $\rightarrow$ Bucharest)$^{416}$

Adapted from slides © 2003 S. Russell & P. Norvig. Reused with permission.

**Constraint Satisfaction Problems: Review**

Binary CSP: each constraint relates at most two variables
Constraint graph: nodes are variables, arcs show constraints

General-purpose CSP algorithms use the graph structure to speed up search. E.g., Tasmania is an independent subproblem!

---

**Minimax & Alpha-Beta (α - β) Pruning: Review**

*What are α, β values here?*

![Diagram](image)

Figure 6.5 p. 168 R&N 2e

Adapted from slides © 2004 S. Russell & P. Norvig. Reused with permission.
**Inference:**

**Review**

\[ KB \vdash_i \alpha \] means sentence \( \alpha \) can be derived from \( KB \) by procedure \( i \).

Consequences of \( KB \) are a haystack; \( \alpha \) is a needle.

Entailment = needle in haystack; inference = finding it.

**Soundness:** \( i \) is sound if whenver \( KB \vdash_i \alpha \), it is also true that \( KB \models \alpha \).

**Completeness:** \( i \) is complete if whenever \( KB \models \alpha \), it is also true that \( KB \vdash_i \alpha \).

Preview: we will define a logic (first-order logic) which is expressive enough to say almost anything of interest, and for which there exists a sound and complete inference procedure.

That is, the procedure will answer any question whose answer follows from what is known by the \( KB \).

---

**Logics in General:**

**Review**

<table>
<thead>
<tr>
<th>Language</th>
<th>Ontological Commitment</th>
<th>Epistemological Commitment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propositional logic</td>
<td>facts</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>First-order logic</td>
<td>facts, objects, relations</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>Temporal logic</td>
<td>facts, objects, relations, times</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>Probability theory</td>
<td>facts</td>
<td>true/false/unknown</td>
</tr>
<tr>
<td>Fuzzy logic</td>
<td>facts + degree of truth</td>
<td>degree of belief known interval value</td>
</tr>
</tbody>
</table>

**Ontological commitment** – what entities, relationships, and facts exist in world and can be reasoned about.

**Epistemic commitment** – what agents can know about the world.

Adapted from slide © 2004 S. Russell & P. Norvig. Reused with permission.
Clausal Form (CNF) Conversion: Review

- Implications Out (Replace with Disjunctive Clauses)
- Negations Inward (DeMorgan’s Theorem)
- Standardize Variables Apart (Eliminate Duplicate Names)
- Existentials Out (Skolemize)
- Universals Made Implicit
- Distribute And Over Or (i.e., Disjunctions Inward)
- Operators Made Implicit (Convert to List of Lists of Literals)
- Rename Variables (Independent Clauses)
- A Memonic for Star Trek: The Next Generation Fans

Captain Picard:

I’ll Notify Spock’s Eminent Underground Dissidents On Romulus

I’ll Notify Sarek’s Eminent Underground Descendant On Romulus


Resolution Strategies [1]: Review

- Unit Preference
  - Idea: Prefer inferences that produce shorter sentences
  - Compare: Occam’s Razor
  - How? Prefer unit clause (single-literal) resolvents \((a \lor \beta \text{ with } \neg \beta \lor a)\)
  - Reason: trying to produce a short sentence (\(\bot\equiv\text{True} \Rightarrow \text{False}\))

- Input Resolution
  - Idea: “diagonal” proof (proof “list” instead of proof tree)
  - Every resolution combines some input sentence with some other sentence
  - Input sentence: in original KB or query
Resolutions Strategies [2]: Review

- **Linear Resolution**
  - Generalization of input resolution
  - Include any ancestor in proof tree to be used

- **Set of Support (SoS)**
  - Idea: try to eliminate some potential resolutions
  - Prevention as opposed to cure
  - How?
    - Maintain set SoS of resolution results
    - Always take one resolvent from it
  - Caveat: need right choice for SoS to ensure completeness

Logic and Decision Problems: Review

- Undecidable duals:
  - $\alpha \in L_{\text{VALID}} \iff \neg \alpha \in L_{\text{SAT}}$

- Universe of Decision Problems

- Recursive Languages (REC)
- Recursive Enumerable Languages (RE)
- Co-RE (RE$^c$)

- Recursive Languages (REC)
- Recursive Enumerable Languages (RE)
- Co-RE (RE$^c$)

- Universe of Decision Problems

- Undecidable duals:
  - $\alpha \in L_{\text{VALID}} \iff \neg \alpha \in L_{\text{SAT}}$

- Semi-decidable duals:
  - $\alpha \in L_{\text{VALID}} \iff \neg \alpha \in L_{\text{SAT}}$

- $\alpha \vdash \bot \rightarrow ?$

- $\alpha \models \bot \rightarrow ?$

- Y
- N
“Concept” and “Class” are used synonymously.

Class: concept in the domain
- wines
- wineries
- red wines

Collection of elements with similar properties

Instances of classes
- Particular glass of California wine

Slots in class definition C
- Describe attributes of instances of C
- Describe relationships to other instances
- e.g., each wine will have color, sugar content, producer, etc.

Property constraints (facets): describe/limit possible values for slot

Slots & facets for Concept/Class Wine
A knowledge engineer attempts to understand how a subject matter expert reasons and solves problems and then encodes the acquired expertise into the agent's knowledge base.

The expert analyzes the solutions generated by the agent (and often the knowledge base itself) to identify errors, and the knowledge engineer corrects the knowledge base.

© 2001 G. Tecuci, George Mason University
CS 785 Knowledge Acquisition and Problem-Solving  http://lalab.gmu.edu/cs785/

Elicitation Methodology: Review
(based primarily on Gammack, 1987)

1. Concept elicitation: methods
(elicit concepts of domain, i.e. agreed-upon vocabulary)
2. Structure elicitation: card-sort method
(elicit some structure for concepts)
3. Structure representation
(formally represent structure in semantic network)
4. Transformation of representation
(transform representation to be used for some desired purpose)
HIERARCHY AND TAXONOMY: REVIEW

- Hierarchy or taxonomy is a natural way to view the world
  - It is used in frames (IS-A relation) and in DL
- Importance of abstraction in remembering and reasoning
  - Groups of things share properties in the world
  - We do not have to repeat representations

Example:
- Saying “elephants are mammals” is sufficient to know a lot about them

Inheritance is the result of reasoning over paths in a hierarchy
- “does a inherit from b?” is the same as “is b in the transitive closure of IS-A (or subsumption) from a?”

INHERITANCE: REVIEW

- IS relations:
  - Clyde is an Elephant, Elephant is Gray
  - Grey
  - Elephant
  - Clyde
- Reasoning with paths and conclusions they represent:
  - Transitive relations
  - Transitive closure:
  - Clyde is Gray. Elephant is Gray, Clyde is Elephant
Actions, Situations, Time & Events: Review

- **Axioms**: Truth of Predicate $P$
  - Fully specify situations where $P$ true
  - $\vdash$ biconditional ($\iff$)
- **Original Predicates**
  - Describe state of world
  - Each augmented with situation argument $s$

\[
\forall a, s \quad \text{Holding(Gold, Result(a, s))} \iff \begin{cases} 
(a = \text{Grab} \land \text{AtGold(s)}) \\
(\text{Holding(Gold, s)} \land a \neq \text{Release})
\end{cases}
\]

Successor-state axioms solve the representational frame problem

Each axiom is “about” a predicate (not an action per se):

- $P$ true afterwards $\iff$ [an action made $P$ true]
- $P$ true already and no action made $P$ false

Looking Ahead: Planning & Blocks World

"Sussman anomaly" problem

- **Start State**
  - $\text{Clear}(x)$, $\text{On}(x, z)$, $\text{Clear}(y)$
  - $\text{PutOn}(x, y)$
  - $\neg\text{On}(x, z)$, $\neg\text{Clear}(y)$, $\text{Clear}(z)$, $\text{On}(x, y)$

- **Goal State**
  - $\text{Clear}(x)$, $\text{On}(x, z)$
  - $\text{PutOnTable}(x)$
  - $\neg\text{On}(x, z)$, $\text{Clear}(z)$, $\text{On}(x, \text{Table})$

+ several inequality constraints
TERMINOLOGY

- **Intelligent Agents**
  - Chapter 1: Overview
  - Chapter 2: Definition of IAs
  - Types: Reflex, Reflex with State, Goal-Based, Preference-Based

- **Search**
  - Chapter 3: blind search
  - Chapter 4: informed search, heuristics, Best-First & variants
  - Chapter 5: constraints
  - Chapter 6: game tree search

- **Section III: Knowledge Representation and Reasoning**
  - Chapter 7: propositional logic
  - Chapter 8: first-order logic
  - Chapter 9: inference in FOL (resolution)
  - Chapter 10: knowledge representation

SUMMARY POINTS

- **Section I: Intelligent Agents, Chapters 1 – 2**
  - Chapter 1: Overview
  - Chapter 2: Definition of IAs
  - Types: Reflex, Reflex with State, Goal-Based, Preference-Based

- **Section II: Search, Chapters 3 – 6**
  - Chapter 3: blind search
  - Chapter 4: informed search, heuristics, Best-First & variants
  - Chapter 5: constraints
  - Chapter 6: game tree search

- **Section III: Knowledge Representation and Reasoning**
  - Chapter 7: propositional logic
  - Chapter 8: first-order logic
  - Chapter 9: inference in FOL (resolution)
  - Chapter 10: knowledge representation