CS 5260: Intro to Artificial Intelligence

Exam Packet

Search Strategies

Breadth-First Search: FIFO Frontier, No Evaluation Cost (Uninformed)

Depth-First Search: LIFO Stack Frontier, No Evaluation Cost (Uninformed)

Heuristic Depth-First Search: LIFO Stack Frontier, Evaluation Cost: $f = h$

A* Search: Priority Queue Frontier, Evaluation Cost: $f = g + h$

Iterative Deepening Search: Depth-First, Restart with increasing max depths

Greedy Best-First Search: Priority Queue Frontier, Evaluation Cost: $f = h$

Uniform Cost Search: Priority Queue Frontier, Evaluation Cost: $f = g$

Logic Equivalence Rules

$$
\begin{aligned}
(\alpha \wedge \beta) &\equiv (\beta \wedge \alpha) \quad \text{commutativity of } \wedge \\
(\alpha \vee \beta) &\equiv (\beta \vee \alpha) \quad \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma)) \quad \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) &\equiv (\alpha \vee (\beta \vee \gamma)) \quad \text{associativity of } \vee \\
\neg(\neg\alpha) &\equiv \alpha \quad \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \vee \beta) \quad \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) &\equiv (\neg\alpha \vee \neg\beta) \quad \text{De Morgan} \\
\neg(\alpha \vee \beta) &\equiv (\neg\alpha \wedge \neg\beta) \quad \text{De Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) &\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) \quad \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) &\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) \quad \text{distributivity of } \vee \text{ over } \wedge
\end{aligned}
$$

Figure 7.11 Standard logical equivalences. The symbols α, β, and γ stand for arbitrary sentences of propositional logic.

Logic Inference Rules

AND-Introduction:	a, b	$\vdash$	$a \cap b$
AND-Elimination:	$a \cap b$	$\vdash$	a, b
OR-Introduction:	a	$\vdash$	$a \cup b$
Modus Ponens:	$a, a \rightarrow b$	$\vdash$	b
Modus Tollens:	$\neg b, a \rightarrow b$	$\vdash$	$\neg a$
Resolution:	$a_1 \cup \cdots \cup z, b_1 \cup \cdots \cup \neg z$	$\vdash$	$a_1 \cup \cdots \cup b_1 \cup \cdots$

Rules of Probability

- **Product Rule:** $P(A \cap B) = P(A|B)P(B)$

  $P(A|B) = \frac{P(A \cap B)}{P(B)}$

- **Bayes' Rule:** $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

  $P(A|B)P(B) = P(B|A)P(A)$

- **Chain Rule:** $P(A_n \cap \cdots \cap A_1) = P(A_n|A_{n-1} \cap \cdots \cap A_1)P(A_{n-1} \cap \cdots \cap A_1)$

  Example: $P(A \cap B \cap C \cap D) = P(A|B \cap C \cap D)P(B \cap C \cap D)$

  $P(A \cap B \cap C \cap D) = P(A|B \cap C \cap D)P(B|C \cap D)P(C|D)P(D)$

- **Law of Total Probability: (i.e., "Summing Out")** $P(B) = \sum_A P(B, A)$

Hidden Markov Models

Generic Model

$$P(S_t|o_0, \ldots, o_t) \propto P(o_t|S_t) \sum_{s_{t-1}} P(S_t|s_{t-1}) P(s_{t-1}|o_0, \ldots, o_{t-1})$$

Sensor Model	Transition Model	Prior Belief

Model with Actions

$$P(S_t|o_0, \ldots, o_t, a_0, \ldots, a_t) \propto P(a_{t-1}) P(o_t|S_t) \sum_{s_{t-1}} P(S_t|s_{t-1}, a_{t-1}) P(s_{t-1}|o_0, \ldots, o_{t-1})$$

First-Order Logic Rules

- **For all**: $\forall x,y$ Predicate(x,y) === $\forall x \forall y$ Predicate(x,y) === $\forall x[\forall y$ Predicate(x,y)]
- **There exists**: $\exists x,y$ Predicate(x,y) === $\exists x \exists y$ Predicate(x,y) === $\exists x[\exists y$ Predicate(x,y)]

- $\forall x\ \exists y$ Predicate(x,y) !== $\exists x\ \forall y$ Predicate(x,y)

- DeMorgan's Law for Quantification: $\forall x$ ~Predicate(x) === ~$\exists x$ Predicate(x)