Lecture 6a: Rule Based System

CSCI 360

Introduction to Artificial Intelligence

USC

Here is where we are…

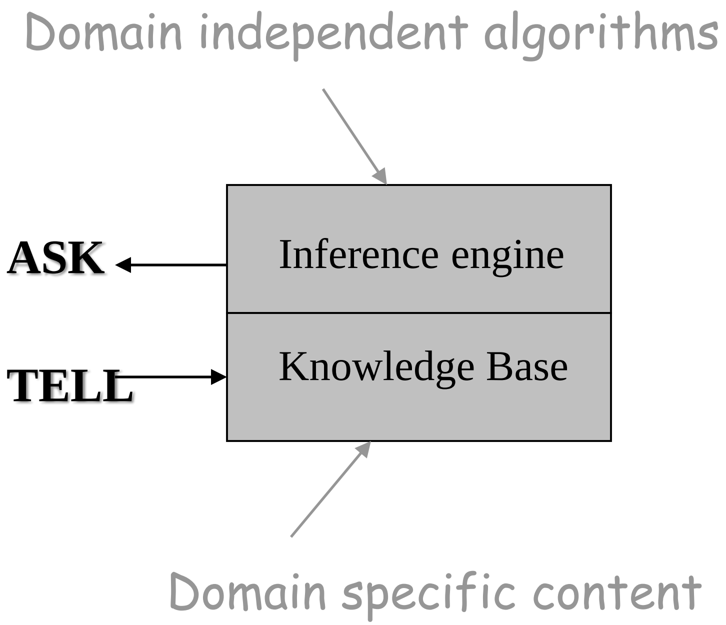
Week	30000D	30282R	Topics	Chapters
1	1/7	1/8	Intelligent Agents	[Ch 1.1-1.4 and 2.1-2.4]
	1/9	1/10	Problem Solving and Search	[Ch 3.1-3.3]
2	1/14	1/15	Uninformed Search	[Ch 3.3-3.4]
	1/16	1/17	Heuristic Search (A*)	[Ch 3.5]
3	1/21	1/22	Heuristic Functions	[Ch 3.6]
	1/23	1/24	Local Search	[Ch 4.1-4.2]
	1/25		Project 1 Out	
4	1/28	1/29	Adversarial Search	[Ch 5.1-5.3]
	1/30	1/31	Knowledge Based Agents	[Ch 7.1-7.3]
5	2/4	2/5	Propositional Logic Inference	[Ch 7.4-7.5]
	2/6	2/7	First-Order Logic	[Ch 8.1-8.4]
	2/8		Project 1 Due	
	2/8		Homework 1 Out	
6	2/11	2/12	Rule-Based Systems	[Ch 9.3-9.4]
	2/13	2/14	Search-Based Planning	[Ch 10.1-10.3]
	2/15		Homework 1 Due	
7	2/18	2/19	SAT-Based Planning	[Ch 10.4]
	2/20	2/21	Knowledge Representation	[Ch 12.1-12.5]
8	2/25	2/26	Midterm Review	
	2/27	2/28	**Midterm Exam**	

Outline

- What is AI?
- Problem-solving agent
  - Uninformed (DFS), informed (A*), and local search
  - Adversarial search (minimax, alpha-beta pruning)
- **Knowledge-based agent**
  - Propositional Logic
  - First Order Logic (FOL)
  - **Automated Reasoning in FOL**

Recap: *Logic for knowledge representation*

- **Logic** as a language for knowledge representation
  - Propositional logic (Boolean)
  - First-order logic (FOL)



- Advantage
  - Can combine and recombine information to suit many purposes

Recap: *Propositional Logic: checking entailment*

- Two methods
  - Method#1: Based on enumeration (model checking)
  - Method#2: Based on inference rules (theorem proving)

- Enumerate all models and check if "*α* *is true in all models in which KB is true*"

$$\boxed{M(KB) \subseteq M(\alpha).}$$

Recap: *PL: Checking entailment (example)*

$\boxed{KB \models \alpha_1}$

$\boxed{KB \not\models \alpha_2}$

$\alpha_1 =$ "There is no pit in [1,2]."

$\alpha_2 =$ "There is no pit in [2,2]."

$$\left.\begin{array}{lllll} R_1: & \neg P_{1,1}\,. & & & \\ R_2: & B_{1,1} & \Leftrightarrow & (P_{1,2} \vee P_{2,1})\,. \\ R_3: & B_{2,1} & \Leftrightarrow & (P_{1,1} \vee P_{2,2} \vee P_{3,1})\,. \\ R_4: & \neg B_{1,1}\,. & & \\ R_5: & B_{2,1}\,. & & \end{array}\right\} \textit{KB}$$

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	*false*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*false*	*true*	*false*	*false*
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
false	*true*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*false*	*true*	*true*	*false*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	<u>*true*</u>
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

Recap: *PL: Checking entailment (inference)*

- Two methods
  - Method#1: Based on enumeration (model checking)
  - Method#2: Based on inference rules (theorem proving)

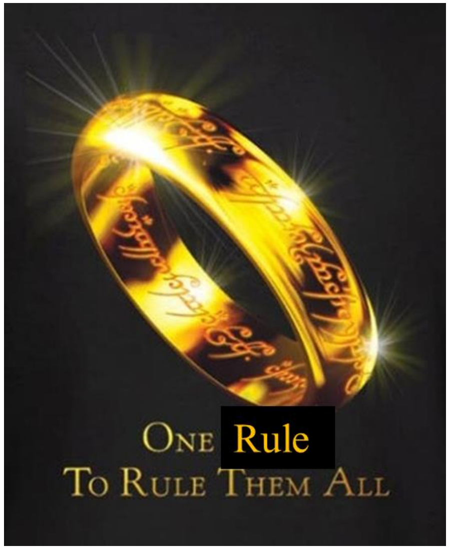
- Logical equivalences

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

Recap: *PL: Resolution rule*

- Resolvent
- Pivot variable (x)

$$\boxed{(C \vee x) \wedge (D \vee \neg x) \rightarrow (C \vee D)}$$



- Example: general case (= *transitivity of implication*)

$$\frac{P_{1,1} \vee P_{3,1}, \qquad \neg P_{1,1} \vee \neg P_{2,2}}{P_{3,1} \vee \neg P_{2,2}}$$

$$\begin{aligned} & (\neg C \rightarrow x) \wedge (x \rightarrow D) \\ = & (\neg C \rightarrow D) \\ = & (C \vee D) \end{aligned}$$

Recap: *Why first-order logic (FOL)?*

- Propositional logic (PL) is limited because it only makes the **ontological commitment** that a world consists of **facts**
  - Facts: propositions that are either *true* or *false*
  - "*Don't go forward if the Wumpus is in front of you*" takes 64 rules

Language	Ontological Commitment (What exists in the world)
Propositional logic	facts
First-order logic	facts, objects, relations
Temporal logic	facts, objects, relations, times
Probability theory	facts
Fuzzy logic	facts with degree of truth $\in [0, 1]$

Recap: *First-order logic (FOL)*

- Ontological commitments:
  - **Objects**: Wheel, door, body, engine, seat, car, passenger, driver
  - **Functions**: ColorOf(car)
  - **Relations**: Inside(car, passenger), Beside(driver, passenger)
  - **Properties**: IsOpen(door), IsOn(engine)

CarA

CarB

	CarA	CarB
Function:	ColorOf(CarA) = BLACK	ColorOf(CarB) = BLUE
Relation:	ColorOfCar(CarA,BLACK) = True ColorOfCar(CarA,BLUE) = False	ColorOfCar(CarB,BLACK) = False ColorOfCar(CarB,BLUE) = True
Property:	IsBlackCar(CarA) = True IsBlueCar(CarA) = False	IsBlackCar(CarB) = False IsBlueCar(CarB) = True

Recap: *Universal quantification (for all):* ∀

∀ *<variables>* <***sentence***>

Example

- *"Everyone in the cs360 class is smart"*:

  ∀ *x* In (cs360, *x*) ⇒ Smart (*x*)

- **∀ x means the conjunction of all instantiations of x**

  (In (cs360, Markus) ⇒ Smart (Markus)) ∧
  (In (cs360, Dora) ⇒ Smart (Dora)) ∧
  …
  (In (cs360, Hao) ⇒ Smart (Hao))

Recap: *Existential quantification (there exists): ∃*

$\exists$ *<variables>* *<sentence>*

Example

- *"Someone in the cs360 class is smart"*:

$$\exists x \quad In(cs360, x) \land Smart(x)$$

- **∃ x represents the disjunction of all instantiations of x**

$In(cs360, Markus) \land Smart(Markus) \lor$
$In(cs360, Dora) \land Smart(Dora) \lor$
…
$In(cs360, Hao) \land Smart(Hao)$

Using FOL Knowledge Base *(Wumpus)*

$$\forall\, x, y, a, b \;\; Adjacent([x, y], [a, b]) \;\Leftrightarrow$$

Using FOL Knowledge Base *(Wumpus)*

$$\forall\, x, y, a, b \;\; Adjacent([x, y], [a, b]) \;\Leftrightarrow$$
$$(x = a \land (y = b - 1 \lor y = b + 1)) \lor (y = b \land (x = a - 1 \lor x = a + 1))$$

Squares are breezy near a pit:

<u>Diagnostic</u> rule—infer cause from effect

$$\forall y \;\; Breezy(y) \;\Rightarrow\; \exists x \;\; Pit(x) \land Adjacent(x, y)$$

Using FOL Knowledge Base *(Wumpus)*

$$\forall\, x, y, a, b \;\; Adjacent([x, y], [a, b]) \;\Leftrightarrow$$
$$(x = a \land (y = b - 1 \lor y = b + 1)) \lor (y = b \land (x = a - 1 \lor x = a + 1))$$

Squares are breezy near a pit:

Diagnostic rule—infer cause from effect

$$\forall\, y \;\; Breezy(y) \;\Rightarrow\; \exists\, x \;\; Pit(x) \land Adjacent(x, y)$$

Causal rule—infer effect from cause

$$\forall\, x, y \;\; Pit(x) \land Adjacent(x, y) \;\Rightarrow\; Breezy(y)$$

Using FOL Knowledge Base *(Wumpus)*

$$\forall\, x, y, a, b \;\; Adjacent([x, y], [a, b]) \;\Leftrightarrow$$
$$(x = a \wedge (y = b - 1 \vee y = b + 1)) \vee (y = b \wedge (x = a - 1 \vee x = a + 1))$$

Squares are breezy near a pit:

Diagnostic rule—infer cause from effect

$$\forall\, y \;\; Breezy(y) \;\Rightarrow\; \exists\, x \;\; Pit(x) \wedge Adjacent(x, y)$$

Causal rule—infer effect from cause

$$\forall\, x, y \;\; Pit(x) \wedge Adjacent(x, y) \;\Rightarrow\; Breezy(y)$$

Neither of these is complete—e.g., the causal rule doesn't say whether squares far away from pits can be breezy

Using FOL Knowledge Base *(Wumpus)*

$$\forall\, x, y, a, b \;\; Adjacent([x, y], [a, b]) \;\Leftrightarrow$$
$$(x = a \wedge (y = b - 1 \vee y = b + 1)) \vee (y = b \wedge (x = a - 1 \vee x = a + 1))$$

Squares are breezy near a pit:

Diagnostic rule—infer cause from effect

$$\forall y \;\; Breezy(y) \;\Rightarrow\; \exists x \;\; Pit(x) \wedge Adjacent(x, y)$$

Causal rule—infer effect from cause

$$\forall x, y \;\; Pit(x) \wedge Adjacent(x, y) \;\Rightarrow\; Breezy(y)$$

Neither of these is complete—e.g., the causal rule doesn't say whether squares far away from pits can be breezy

Definition for the $Breezy$ predicate:

$$\forall y \;\; Breezy(y) \;\Leftrightarrow\; [\exists x \;\; Pit(x) \wedge Adjacent(x, y)]$$

Equality

$term_1 = term_2$ is true under a given interpretation
if and only if $term_1$ and $term_2$ refer to the same object

Equality

$term_1 = term_2$ is true under a given interpretation
if and only if $term_1$ and $term_2$ refer to the same object

E.g., definition of (full) $Sibling$ in terms of $Parent$:

$$\forall\, x,y \;\; Sibling(x,y) \;\Leftrightarrow\; [\;\; \exists\, m,f$$
$$Parent(m,x) \land Parent(f,x) \land Parent(m,y) \land Parent(f,y)]$$

Equality

$term_1 = term_2$ is true under a given interpretation
if and only if $term_1$ and $term_2$ refer to the same object

E.g., definition of (full) $Sibling$ in terms of $Parent$:

$$\forall\, x, y \;\; Sibling(x,y) \;\Leftrightarrow\; [\neg(x = y) \land \exists\, m, f \;\; \neg(m = f) \land$$
$$Parent(m,x) \land Parent(f,x) \land Parent(m,y) \land Parent(f,y)]$$

Higher-order logic?

- First-order logic allows us to quantify over **objects**.

- Higher-order logic allows us to quantify over **objects**, **relations**, and **functions**.

  e.g., "two objects are equal if and only if all **properties** applied to them are equivalent":

  $$\forall x,y \quad (x=y) \Leftrightarrow (\forall \boldsymbol{p} \;\; \boldsymbol{p}(x) \Leftrightarrow \boldsymbol{p}(y))$$

- Higher-order logics are ***more expressive***; however, it is not clear (yet) ***how to effectively reason about*** sentences in higher-order logic.

Outline for Today

- What is AI?
- Problem-solving agent
  - Uninformed (DFS), informed (A*), and local search
  - Adversarial search (minimax, alpha-beta pruning)
- **Knowledge-based agent**
  - Propositional Logic
  - First Order Logic (FOL)
  - **Reasoning in FOL**
    - Substitution
    - Unification
    - Chaining (forward and backward)
    - Resolution

Substitution

- Assume that (KB) has the following axiom:

$$\forall x \;\; King(x) \land Greedy(x) \;\Rightarrow\; Evil(x)\ .$$

- How to infer any of the following sentences?

$$\begin{array}{l}
King(John) \land Greedy(John) \;\Rightarrow\; Evil(John) \\
King(Richard) \land Greedy(Richard) \;\Rightarrow\; Evil(Richard) \\
King(Father(John)) \land Greedy(Father(John)) \;\Rightarrow\; Evil(Father(John))\ . \\
\quad \vdots
\end{array}$$

Substitution

- Assume that (KB) has the following axiom:

$$\forall x \;\; King(x) \wedge Greedy(x) \;\Rightarrow\; Evil(x)\;.$$

- How to infer any of the following sentences?

$$\begin{array}{l} King(John) \wedge Greedy(John) \;\Rightarrow\; Evil(John) \\ King(Richard) \wedge Greedy(Richard) \;\Rightarrow\; Evil(Richard) \\ King(Father(John)) \wedge Greedy(Father(John)) \;\Rightarrow\; Evil(Father(John))\;. \\ \quad\vdots \end{array}$$

- Replacing a variable by a ground term
  - i.e., a term without variables (constant, or a function applied to a constant, or a function applied to another ground term)

$$\boxed{\{x/John\}, \{x/Richard\}, \text{ and } \{x/Father(John)\}}$$

Universal Elimination

- Substitute a variable with a ground term (one without variables)

$$\frac{\forall v \quad \alpha}{\textsc{Subst}(\{v/g\}, \alpha)}$$

- Example:

$$\forall x \quad King(x) \land Greedy(x) \Rightarrow Evil(x) .$$

$$\begin{aligned}
&King(John) \land Greedy(John) \Rightarrow Evil(John) \\
&King(Richard) \land Greedy(Richard) \Rightarrow Evil(Richard) \\
&King(Father(John)) \land Greedy(Father(John)) \Rightarrow Evil(Father(John)) . \\
&\quad \vdots
\end{aligned}$$

Existential Elimination

- Substitute a variable with a single, new, constant symbol

$$\boxed{\frac{\exists v \;\; \alpha}{\textsc{Subst}(\{v/k\}, \alpha)}}$$

- Example:

$$\exists x \;\; Crown(x) \land OnHead(x, John)$$

$$Crown(C_1) \land OnHead(C_1, John)$$

as long as C_1 does not appear elsewhere in the knowledge base.

Reducing FOL to PL for inference

- Apply both "existential" and "universal" eliminations and then discard the quantified sentences
  - Universal elimination using all **ground-term substitutions** from the vocabulary of the knowledge base

$$\forall x \;\; King(x) \land Greedy(x) \;\Rightarrow\; Evil(x)$$
$$King(John)$$
$$Greedy(John)$$
$$Brother(Richard, John)\,.$$

$$\boxed{\{x/John\} \text{ and } \{x/Richard\}}$$

$$King(John) \land Greedy(John) \;\Rightarrow\; Evil(John)$$
$$King(Richard) \land Greedy(Richard) \;\Rightarrow\; Evil(Richard)$$

Reducing FOL to PL for inference

- Apply both “existential” and “universal” eliminations and then discard the quantified sentences
  - Universal elimination using all **ground-term substitutions** from the vocabulary of the knowledge base

KB (in FOL)

$$\forall x \;\; King(x) \land Greedy(x) \;\Rightarrow\; Evil(x)$$
$$King(John)$$
$$Greedy(John)$$
$$Brother(Richard, John)\,.$$

KB (in PL)

$$\cancel{\forall x \;\; King(x) \land Greedy(x) \;\Rightarrow\; Evil(x)}$$
$$King(John)$$
$$Greedy(John)$$
$$Brother(Richard, John)\,.$$
$$King(John) \land Greedy(John) \;\Rightarrow\; Evil(John)$$
$$King(Richard) \land Greedy(Richard) \;\Rightarrow\; Evil(Richard)$$

Sound and Complete?

- Every FOL knowledge base ***(KB)*** and query ***(α)*** can be reduced to propositional (PL) logic to preserve entailment
  - Since PL inference is both “sound” and “complete”, it seems that we will have a similar procedure for FOL…
- **Not so fast!**
- When KB includes a function symbol, the set of possible ground-term substitution is **infinite**!
  - Father(John), Father(Father(John)), Father(Father(Father(John))),…

Herbrand's Theorem (1930)

- If a sentence is entailed by the KB in FOL, there is a proof involving just a **finite subset** of the propositionalized KB.

- **Corollary**: we can always find that proof, by iteratively deepening the function's nesting depth
  - Iteration 1: John, …
  - Iteration 2: Father(John), …
  - Iteration 3: Father(Father(John)), …
  - Iteration 4: Father(Father(Father(John))), …

- Complete: Any entailed sentence can be proved
- Unsound: Can't prove that a sentence is not entailed

Outline

- What is AI?
- Problem-solving agent
  - Uninformed (DFS), informed (A*), and local search
  - Adversarial search (minimax, alpha-beta pruning)
- **Knowledge-based agent**
  - Propositional Logic
  - First Order Logic (FOL)
  - **Reasoning in FOL**
    - Substitution
    - Unification
    - Chaining (forward and backward)
    - Resolution

FOL inference rule

- Reducing FOL to PL is rather inefficient; sometimes, direct inference seems more intuitive

$$\forall x \;\; King(x) \land Greedy(x) \;\Rightarrow\; Evil(x)$$
$$King(John)$$
$$Greedy(John)$$

- Step 1: Find some (x) such that (x) is a king and (x) is greedy

$$\text{substitution } \theta = \{x/John\}$$

- Step 2: Infer that this particular (x) is evil.

$$Evil(John)$$

FOL inference rule

- Reducing FOL to PL is rather inefficient; sometimes, direct inference seems more intuitive

$$\forall x \;\; King(x) \wedge Greedy(x) \;\Rightarrow\; Evil(x)$$

$$King(John)$$

$$\cancel{Greedy(John)} \qquad\qquad \forall y \;\; Greedy(y)$$

- Step 1: Find some (x) such that (x) is a king and (x) is greedy

$$\cancel{\text{substitution } \theta = \{x/John\}} \qquad \text{substitution } \{x/John, y/John\}$$

- Step 2: Infer that this particular (x) is evil.

$$Evil(John)$$

Generalized Modus Ponens

- There is a **substitution** such that

$$\text{SUBST}(\theta, p_i{}') = \text{SUBST}(\theta, p_i), \text{ for all } i,$$

$$\boxed{\frac{p_1{}', \quad p_2{}', \quad \ldots, \quad p_n{}', \quad (p_1 \land p_2 \land \ldots \land p_n \Rightarrow q)}{\text{SUBST}(\theta, q)}}$$

- Example

$\boxed{King(John)}$ $\boxed{\forall y \;\; Greedy(y)}$ $\boxed{\forall x \;\; King(x) \land Greedy(x) \;\Rightarrow\; Evil(x)}$

$p_1{}'$ is $King(John)$ $\quad$ p_1 is $King(x)$

$p_2{}'$ is $Greedy(y)$ $\quad$ p_2 is $Greedy(x)$

Generalized Modus Ponens

- There is a **substitution** such that

$$\text{SUBST}(\theta, p_i{}') = \text{SUBST}(\theta, p_i), \text{ for all } i,$$

$$\boxed{\frac{p_1{}',\ \ p_2{}',\ \ \ldots,\ \ p_n{}',\ \ (p_1 \land p_2 \land \ldots \land p_n \Rightarrow q)}{\text{SUBST}(\theta, q)}}$$

- Example

$\boxed{King(John)}$ $\boxed{\forall y\ \ Greedy(y)}$ $\boxed{\forall x\ \ King(x) \land Greedy(x)\ \Rightarrow\ Evil(x)}$

$p_1{}'$ is $King(John)$ p_1 is $King(x)$
$p_2{}'$ is $Greedy(y)$ p_2 is $Greedy(x)$
$\boxed{\theta \text{ is } \{x/John, y/John\}}$ q is $Evil(x)$

Generalized Modus Ponens

- There is a **substitution** such that

$$\text{SUBST}(\theta, p_i{}') = \text{SUBST}(\theta, p_i), \text{ for all } i,$$

$$\boxed{\frac{p_1{}', \quad p_2{}', \quad \ldots, \quad p_n{}', \quad (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{\text{SUBST}(\theta, q)}}$$

- Example

$\boxed{King(John)}$ $\boxed{\forall y \;\; Greedy(y)}$ $\boxed{\forall x \;\; King(x) \wedge Greedy(x) \;\Rightarrow\; Evil(x)}$

$p_1{}'$ is $King(John)$ $\quad$ p_1 is $King(x)$

$p_2{}'$ is $Greedy(y)$ $\quad$ p_2 is $Greedy(x)$

$\boxed{\theta \text{ is } \{x/John, y/John\}}$ $\quad$ q is $Evil(x)$

$\text{SUBST}(\theta, q)$ is $\boxed{Evil(John)}$.

GMP example

Bob is a buffalo	1. $Buffalo(Bob)$
Pat is a pig	2. $Pig(Pat)$
Buffaloes outrun pigs	3. $\forall\, x, y \;\; Buffalo(x) \land Pig(y) \;\Rightarrow\; Faster(x, y)$
Bob outruns Pat	

GMP example

Bob is a buffalo	1. $Buffalo(Bob)$
Pat is a pig	2. $Pig(Pat)$
Buffaloes outrun pigs	3. $\forall x, y \;\; Buffalo(x) \wedge Pig(y) \;\Rightarrow\; Faster(x, y)$
Bob outruns Pat	
AI 1 & 2	4. $Buffalo(Bob) \wedge Pig(Pat)$

GMP example

Bob is a buffalo	1. $Buffalo(Bob)$
Pat is a pig	2. $Pig(Pat)$
Buffaloes outrun pigs	3. $\forall x, y \;\; Buffalo(x) \land Pig(y) \;\Rightarrow\; Faster(x, y)$
Bob outruns Pat	
AI 1 & 2	4. $Buffalo(Bob) \land Pig(Pat)$
UE 3, $\{x/Bob, y/Pat\}$	5. $Buffalo(Bob) \land Pig(Pat) \;\Rightarrow\; Faster(Bob, Pat)$

GMP example

Bob is a buffalo	1. $Buffalo(Bob)$
Pat is a pig	2. $Pig(Pat)$
Buffaloes outrun pigs	3. $\forall x, y \;\; Buffalo(x) \wedge Pig(y) \;\Rightarrow\; Faster(x, y)$
Bob outruns Pat	
AI 1 & 2	4. $Buffalo(Bob) \wedge Pig(Pat)$
UE 3, $\{x/Bob, y/Pat\}$	5. $Buffalo(Bob) \wedge Pig(Pat) \;\Rightarrow\; Faster(Bob, Pat)$
MP 6 & 7	6. $Faster(Bob, Pat)$

Unification

- **Finding substitutions** that make different logical formulas look identical

$$\text{UNIFY}(p, q) = \theta \text{ where } \text{SUBST}(\theta, p) = \text{SUBST}(\theta, q)$$

- Example

$$\begin{aligned}
&\text{UNIFY}(Knows(John, x),\ Knows(John, Jane)) = \{x/Jane\}\\
&\text{UNIFY}(Knows(John, x),\ Knows(y, Bill)) = \{x/Bill, y/John\}\\
&\text{UNIFY}(Knows(John, x),\ Knows(y, Mother(y))) = \{y/John, x/Mother(John)\}\\
&\text{UNIFY}(Knows(John, x),\ Knows(x, Elizabeth)) = fail\ .
\end{aligned}$$

Unification example

VARIABLE term

1 – unify(P(a,X), P(a,b)) σ = { X / b }

2 – unify(P(a,X), P(Y,b)) σ = { Y/a, X/b }

3 – unify(P(a,X), P(Y,f(a))) σ = { Y/a, X/f(a) }

4 – unify(P(a,X), P(X,b)) σ = **failure**

Outline

- What is AI?
- Problem-solving agent
  - Uninformed (DFS), informed (A*), and local search
  - Adversarial search (minimax, alpha-beta pruning)
- **Knowledge-based agent**
  - Propositional Logic
  - First Order Logic (FOL)
  - **Automated Reasoning in FOL**
    - Substitution
    - Unification (GMP)
    - Chaining (forward and backward)
    - Resolution

FOL definite clause

- **Universally quantified** clause in which exactly one literal is a positive literal

- Example:

$$King(x) \land Greedy(x) \Rightarrow Evil(x)\ .$$
$$King(John)\ .$$
$$Greedy(y)\ .$$

- We omit the “universal quantifier” in FOL definite clauses, but you should assume that, implicitly, all variables are universally quantified

FOL definite clause *(what about ∃ ?)*

- Apply "**Existential Instantiation**" to make sure the KB has only universally quantified clauses

$$\exists\, x\ Owns(Nono, x) \wedge Missile(x)$$

$$Owns(Nono, M_1)$$
$$Missile(M_1)$$

FOL forward chaining

- When a new fact (p) is added to the (KB)
  - For each rule such that (p) unifies with a premise
    - If the other premises are known, then
      - add the conclusion to the (KB) and continue chaining
- Forward chaining is “data-driven”
  - E.g., inferring facts from percepts

FOL forward chaining (example)

Add facts 1, 2, 3, 4, 5, 7 in turn.

Number in [] = unification literal; $\surd$ indicates rule firing

FOL forward chaining (example)

Add facts 1, 2, 3, 4, 5, 7 in turn.

Number in [] = unification literal; $\surd$ indicates rule firing

1. $Buffalo(x) \land Pig(y) \Rightarrow Faster(x, y)$
2. $Pig(y) \land Slug(z) \Rightarrow Faster(y, z)$
3. $Faster(x, y) \land Faster(y, z) \Rightarrow Faster(x, z)$
4. $Buffalo(Bob)$ [1a, $\times$]
5. $Pig(Pat)$ [1b, $\surd$] $\rightarrow$ 6. $Faster(Bob, Pat)$ [3a, $\times$], [3b, $\times$]

   [2a, $\times$]

7. $Slug(Steve)$ [2b, $\surd$]

   $\rightarrow$ 8. $Faster(Pat, Steve)$ [3a, $\times$], [3b, $\surd$]

   $\rightarrow$ 9. $Faster(Bob, Steve)$ [3a, $\times$], [3b, $\times$]

Fixed-point reached!

Properties of FOL forward chaining

- Sound: **Yes**, because every inference is an application of Generalized Modus Pones

- Complete: **Yes**, it answers every query whose answer is entailed by the KB of FOL definite clauses

Question: *What if the query is not entailed by the KB?*

The algorithm may fail to terminate…

$$NatNum(0)$$
$$\forall n \;\; NatNum(n) \;\Rightarrow\; NatNum(S(n))$$

FOL backward chaining

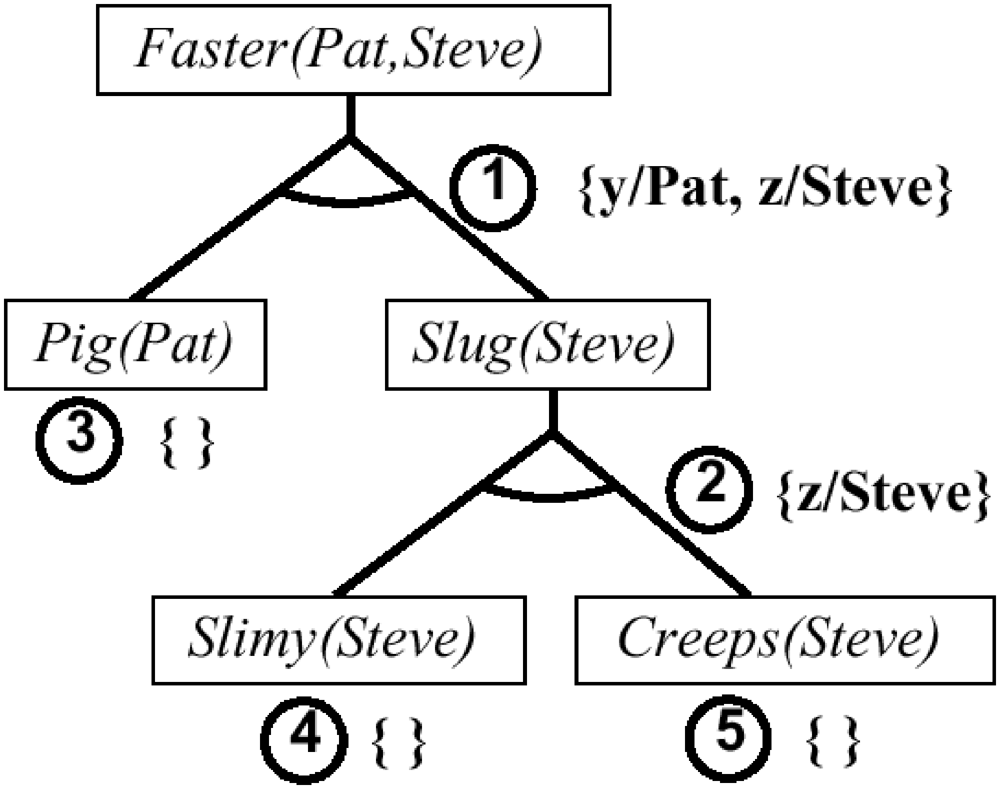
- When a query (q) is asked
  - If a matching fact (q’) is known in the (KB), return the unifier
  - For each rule whose consequence (q’) matches (q)
    - Attempt to prove each of its premises, by backward chaining

FOL backward chaining (example)

1. $Pig(y) \wedge Slug(z) \;\Rightarrow\; Faster(y, z)$
2. $Slimy(z) \wedge Creeps(z) \;\Rightarrow\; Slug(z)$
3. $Pig(Pat)$ 4. $Slimy(Steve)$ 5. $Creeps(Steve)$

FOL backward chaining (example)

1. $Pig(y) \land Slug(z) \Rightarrow Faster(y, z)$
2. $Slimy(z) \land Creeps(z) \Rightarrow Slug(z)$
3. $Pig(Pat)$ 4. $Slimy(Steve)$ 5. $Creeps(Steve)$



Backward Chaining

- Question: Has ***Reality Man*** done anything ***criminal***?
  - Criminal(Reality Man)

- Possible answers:
  - $Steal(x, y) \Rightarrow Criminal(x)$
  - $Kill(x, y) \Rightarrow Criminal(x)$
  - $Grow(x, y) \wedge Illegal(y) \Rightarrow Criminal(x)$
  - $HaveSillyName(x) \Rightarrow Criminal(x)$
  - $Programmer(x) \wedge Emulator(y) \wedge People(z) \wedge Provide(x,z,y) \Rightarrow Criminal(x)$

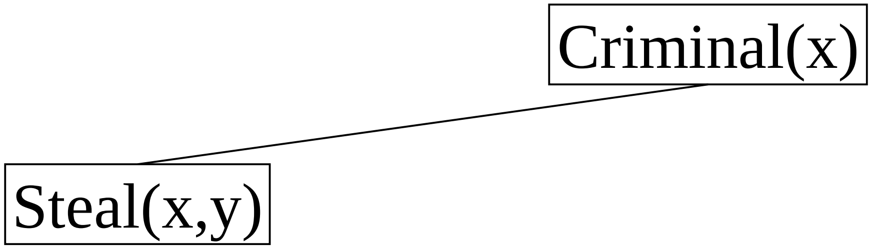
Backward Chaining

- Question: Has Reality Man done anything criminal?

$\boxed{\text{Criminal}(x)}$

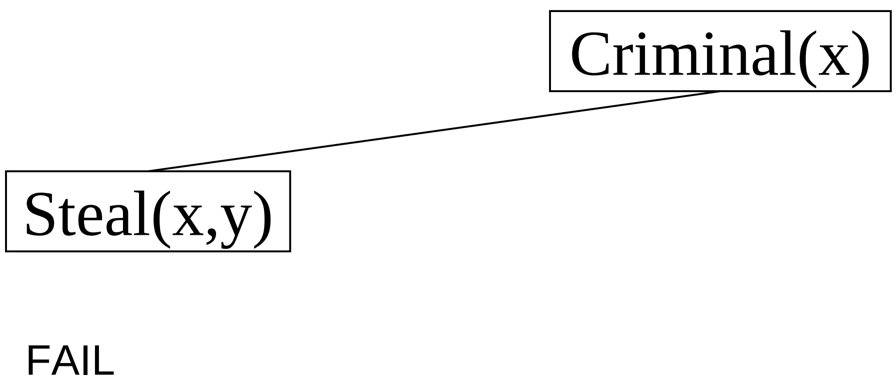
Backward Chaining

- Question: Has Reality Man done anything criminal?



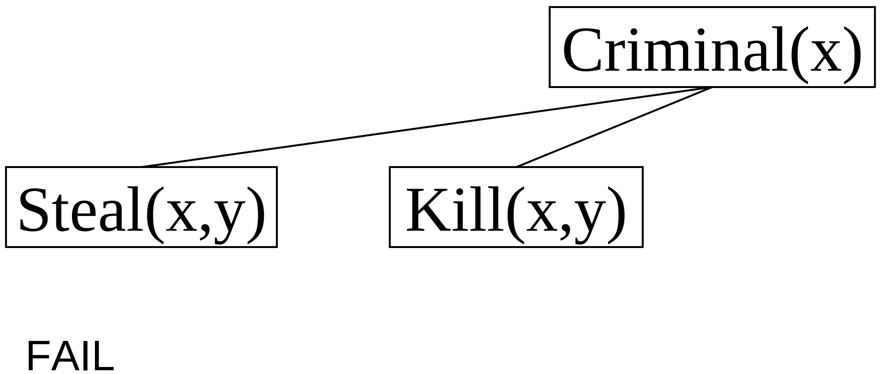
Backward Chaining

- Question: Has Reality Man done anything criminal?



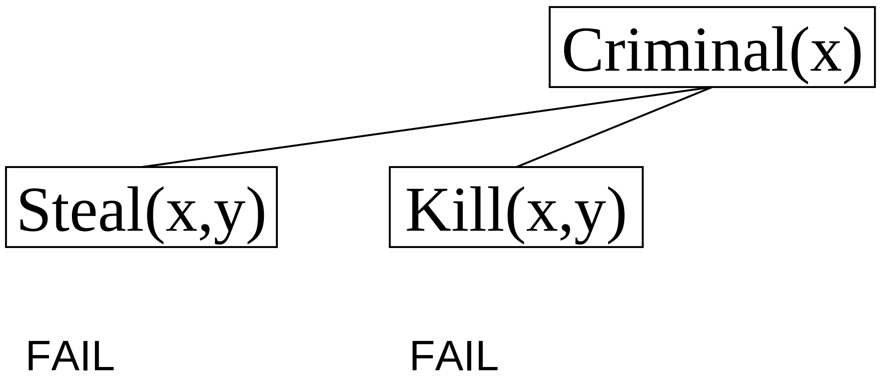
Backward Chaining

- Question: Has Reality Man done anything criminal?



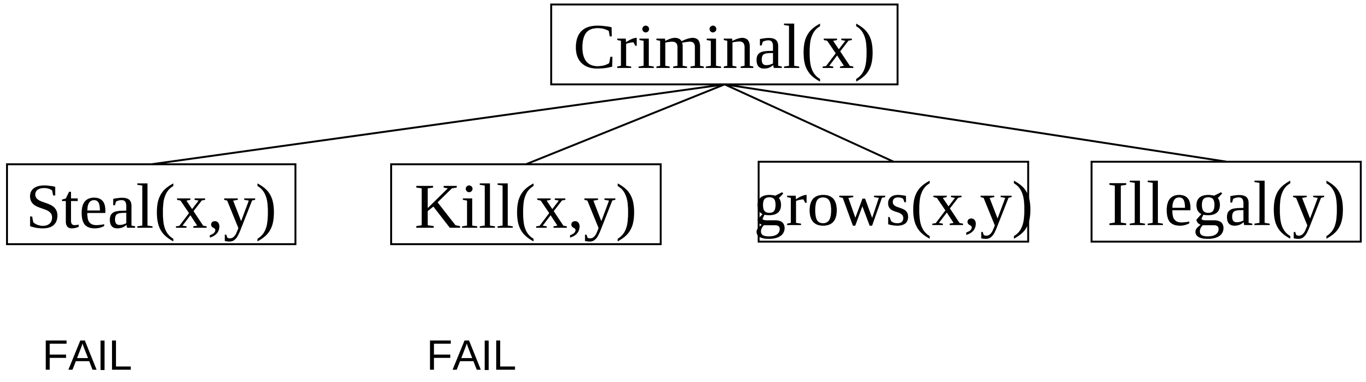
Backward Chaining

- Question: Has Reality Man done anything criminal?



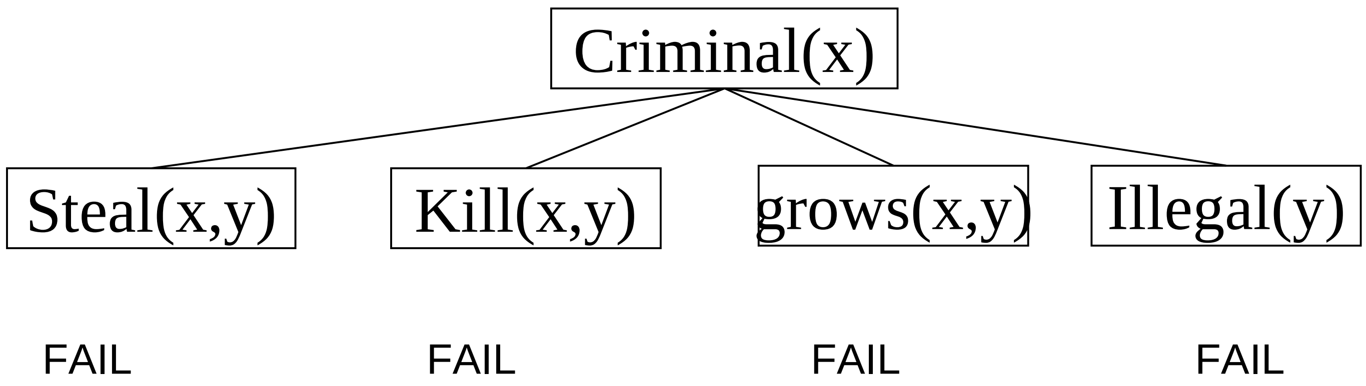
Backward Chaining

- Question: Has Reality Man done anything criminal?



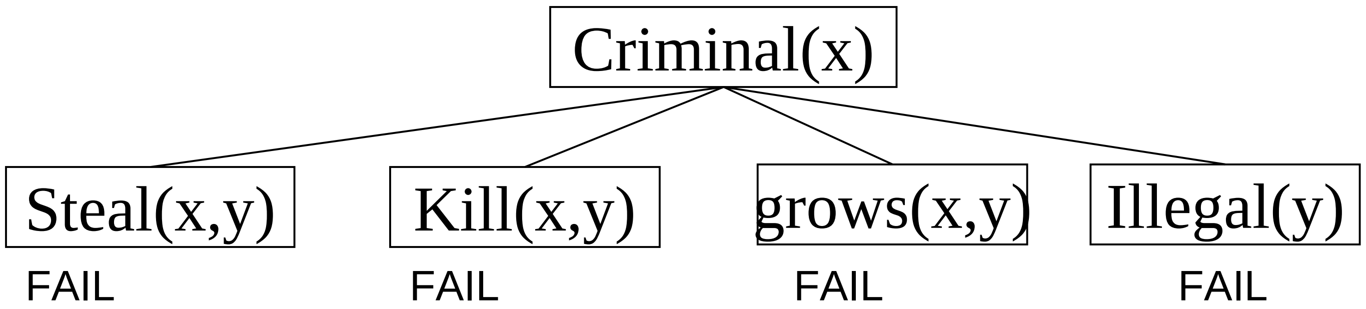
Backward Chaining

- Question: Has Reality Man done anything criminal?



Backward Chaining

- Question: Has Reality Man done anything criminal?



- Backward Chaining is a **depth-first search**: in any knowledge base of realistic size, many search paths will result in failure.

Outline

- What is AI?
- Problem-solving agent
  - Uninformed (DFS), informed (A*), and local search
  - Adversarial search (minimax, alpha-beta pruning)
- **Knowledge-based agent**
  - Propositional Logic
  - First Order Logic (FOL)
  - **Automated Reasoning in FOL**
    - Substitution
    - Unification (GMP)
    - Chaining (forward and backward)
    - Resolution

Resolution (a simple example)

KB:

(1) father (art, jon)

(2) father (bob, kim)

(3) father (X, Y) $\Rightarrow$ parent (X, Y)

Goal: parent (art, jon)?

(KB) $\wedge$ (¬ Goal) is "Unsatisfiable"

Resolution (a simple example)

KB:

(1) father (art, jon)

(2) father (bob, kim)

(3) father (X, Y) $\Rightarrow$ parent (X, Y)

Goal: parent (art, jon)?

¬ parent(art, jon)   father(X, Y) => parent(X, Y)

\   /

¬ father (art, jon)   father (art, jon)

\   /

[ ]

FOL resolution rule

$\textsc{Unify}(\ell_i, \neg m_j) = \theta.$

$$\boxed{\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{\textsc{Subst}(\theta, \ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n)}}$$

- Example:

$[Animal(F(x)) \vee Loves(G(x), x)]$ and $[\neg Loves(u, v) \vee \neg Kills(u, v)]$

$$\Big\downarrow \; \theta = \{u/G(x), v/x\}$$

$[Animal(F(x)) \vee \neg Kills(G(x), x)]$

FOL resolution (example)

$$\frac{\begin{array}{l}\neg Rich(x) \vee Unhappy(x) \\ Rich(Me)\end{array}}{Unhappy(Me)}$$

with $\theta = \{x/Me\}$

FOL Conjunctive normal form (CNF)

Steps:

1. Replace $P \Rightarrow Q$ by $\neg P \vee Q$
2. Move $\neg$ inwards, e.g., $\neg \forall x\, P$ becomes $\exists x\, \neg P$
3. Standardize variables apart, e.g., $\forall x\, P \vee \exists x\, Q$ becomes $\forall x\, P \vee \exists y\, Q$
4. Move quantifiers left in order, e.g., $\forall x\, P \vee \exists x\, Q$ becomes $\forall x \exists y\, P \vee Q$
5. Eliminate $\exists$ by Skolemization (next slide)
6. Drop universal quantifiers
7. Distribute $\wedge$ over $\vee$, e.g., $(P \wedge Q) \vee R$ becomes $(P \vee Q) \wedge (P \vee R)$

Skolemization

- Why can't (y) be replaced by a constant symbol?

$$\forall x \; [\exists y \;\; Animal(y) \wedge \neg Loves(x, y)] \vee [\exists z \;\; Loves(z, x)]$$

  - Everyone loves the same animal (F), and the same (G) loves everyone

$$\boxed{\forall x \;\; [Animal(F) \wedge \neg Loves(x, F)] \vee Loves(G, x)}$$

- Each person loves a different animal *F(x)*, and a different *G(x)* loves each person

$$\boxed{\forall x \;\; [Animal(F(x)) \wedge \neg Loves(x, F(x))] \vee Loves(G(x), x)}$$

- Distribution

$$\forall x \;\; [Animal(F(x)) \vee Loves(G(x), x)\;] \wedge [\neg Loves(x, F(x)) \vee Loves(G(x), x)\;]$$

Converting to CNF

$$\boxed{\forall x \;\; [\forall y \;\; Animal(y) \;\Rightarrow\; Loves(x,y)] \;\Rightarrow\; [\exists y \;\; Loves(y,x)]}$$

- Eliminate implications

$$\forall x \;\; [\neg\forall y \;\; \neg Animal(y) \lor Loves(x,y)] \lor [\exists y \;\; Loves(y,x)]$$

- Move negation inwards

$$\forall x \;\; [\exists y \;\; \neg(\neg Animal(y) \lor Loves(x,y))] \lor [\exists y \;\; Loves(y,x)] \;.$$
$$\forall x \;\; [\exists y \;\; \neg\neg Animal(y) \land \neg Loves(x,y)] \lor [\exists y \;\; Loves(y,x)] \;.$$
$$\forall x \;\; [\exists y \;\; Animal(y) \land \neg Loves(x,y)] \lor [\exists y \;\; Loves(y,x)] \;.$$

- Standardize variables

$$\forall x \;\; [\exists y \;\; Animal(y) \land \neg Loves(x,y)] \lor [\exists z \;\; Loves(z,x)]$$

- Skolemization

$$\forall x \;\; [Animal(F(x)) \land \neg Loves(x,F(x))] \lor Loves(G(x),x)$$

FOL resolution (another example)

A. $\forall x \; [\forall y \; Animal(y) \Rightarrow Loves(x,y)] \Rightarrow [\exists y \; Loves(y,x)]$

B. $\forall x \; [\exists z \; Animal(z) \wedge Kills(x,z)] \Rightarrow [\forall y \; \neg Loves(y,x)]$

C. $\forall x \; Animal(x) \Rightarrow Loves(Jack,x)$

D. $Kills(Jack,Tuna) \vee Kills(Curiosity,Tuna)$

E. $Cat(Tuna)$

F. $\forall x \; Cat(x) \Rightarrow Animal(x)$

$\neg$G. $\neg Kills(Curiosity,Tuna)$

Transform to CNF

FOL resolution (another example)

A. $\forall x \; [\forall y \; Animal(y) \Rightarrow Loves(x,y)] \Rightarrow [\exists y \; Loves(y,x)]$

B. $\forall x \; [\exists z \; Animal(z) \land Kills(x,z)] \Rightarrow [\forall y \; \neg Loves(y,x)]$

C. $\forall x \; Animal(x) \Rightarrow Loves(Jack,x)$

D. $Kills(Jack,Tuna) \lor Kills(Curiosity,Tuna)$

E. $Cat(Tuna)$

F. $\forall x \; Cat(x) \Rightarrow Animal(x)$

¬G. $\neg Kills(Curiosity,Tuna)$

A1. $Animal(F(x)) \lor Loves(G(x),x)$

A2. $\neg Loves(x,F(x)) \lor Loves(G(x),x)$

B. $\neg Loves(y,x) \lor \neg Animal(z) \lor \neg Kills(x,z)$

C. $\neg Animal(x) \lor Loves(Jack,x)$

D. $Kills(Jack,Tuna) \lor Kills(Curiosity,Tuna)$

E. $Cat(Tuna)$

F. $\neg Cat(x) \lor Animal(x)$

¬G. $\neg Kills(Curiosity,Tuna)$

Outline

- What is AI?
- Problem-solving agent
  - Uninformed (DFS), informed (A*), and local search
  - Adversarial search (minimax, alpha-beta pruning)
- **Knowledge-based agent**
  - Propositional Logic
  - First Order Logic (FOL)
  - **Reasoning in FOL**
    - Substitution
    - Unification
    - Chaining (forward and backward)
    - Resolution