# CSE 473: First Order Logic

Luke Zettlemoyer

(With many slides from Dan Weld, Raj Rao, Mausam, Stuart Russell, Dieter Fox, Henry Kautz, Min-Yen Kan…)

# Outline

- First-Order Logic
  - Definitions
  - Universal and Existential Quantifiers
  - Skolemization
  - Unification
  - Chaining and Resolution

# Pros and Cons of Propositional Logic

- Propositional logic is *declarative*: pieces of syntax correspond to facts
- Propositional logic allows *partial/disjunctive/negated* information (unlike most data structures and databases)
- Propositional logic is *compositional*:
  - meaning of $B_{1,1} \wedge P_{1,2}$ derived from meanings of $B_{1,1}$ and $P_{1,2}$
- Propositional logic has very limited expressive power (unlike natural language)
  - E.g., cannot say "pits cause breezes in adjacent squares" except by writing one sentence for each square

# Why First Order Logic

Propositional logic: Deals with facts and propositions (can be true or false):

- $P_{1,1}$ -- “there is a pit in (1,1)”
- George_Monkey -- “George is a monkey”
- George_Curious -- “George is curious”
- Luke_Monkey – “Luke is a monkey’’
- 473student1_curious – “student 1 is a curious’’
- (George_Monkey ∧ ¬473student1_Monkey) ∨ …

# FOL Definitions

*Constants*: Name a specific object.

George,  Monkey2, Larry,Luke  …

*Variables*: Refer to an object without naming it.

X, Y, …

*Relations (predicates)*: Properties of or relationships between objects.

Curious(.), PokesInTheEyes(.,.), SmarterThan(.,.)…

*Functions*: Mapping from objects to objects.

banana-of(.), grade-of(.), child-of(.,.)

# Syntax of First Order Logic

| | |
|---|---|
| Constants | $KingJohn,\ 2,\ UCB, ..$ |
| Predicates | $Brother,\ >, \ldots$ |
| Functions | $Sqrt,\ LeftLegOf, \ldots$ |
| Variables | $x,\ y,\ a,\ b, \ldots$ |
| Connectives | $\wedge\ \vee\ \neg\ \Rightarrow\ \Leftrightarrow$ |
| Equality | $=$ |
| Quantifiers | $\forall\ \exists$ |

Atomic sentence $=$ $predicate(term_1, \ldots, term_n)$
or $term_1 = term_2$

Term $=$ $function(term_1, \ldots, term_n)$
or $constant$ or $variable$

Atomic Sentences:

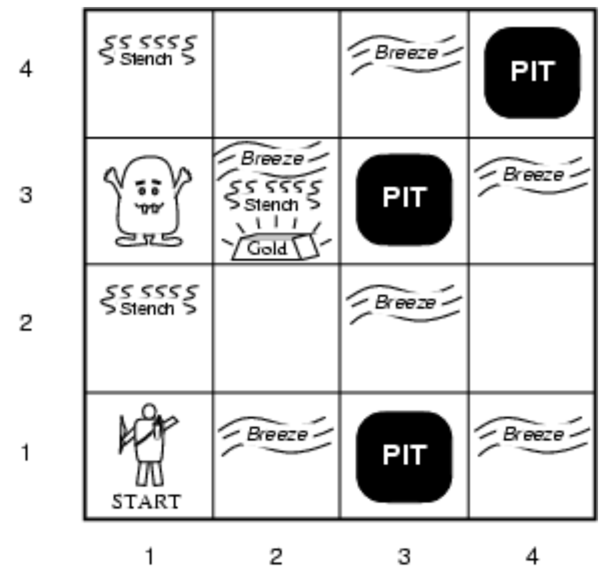
E.g., $Brother(KingJohn, RichardTheLionheart)$
$> (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))$

Complex Sentences:

E.g. $Sibling(KingJohn, Richard) \ \Rightarrow \ Sibling(Richard, KingJohn)$
$>(1,2) \vee \leq(1,2)$
$>(1,2) \wedge \neg{>}(1,2)$

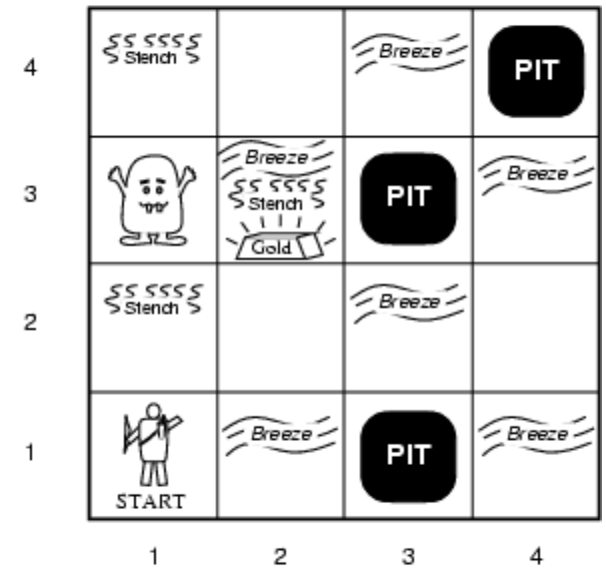
# Wumpus World

- Performance measure
  - Gold: +1000, death: -1000
  - -1 per step, -10 for using the arrow
- Environment
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter iff gold is in the same square
  - Shooting kills wumpus if you are facing it
  - Shooting uses up the only arrow
  - Grabbing picks up gold if in same square
  - Releasing drops the gold in same square




- Sensors: Stench, Breeze, Glitter, Bump, Scream
- Actuators: Left turn, Right turn, Forward, Grab, Release, Shoot

# Wumpus World




Properties of locations:

$\forall x, t \;\; At(Agent, x, t) \land Smelt(t) \;\Rightarrow\; Smelly(x)$

$\forall x, t \;\; At(Agent, x, t) \land Breeze(t) \;\Rightarrow\; Breezy(x)$

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

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

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:

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

# First Order Models

Sentences are true with respect to a model and an interpretation

Model contains $\geq 1$ objects (domain elements) and relations among them

Interpretation specifies referents for

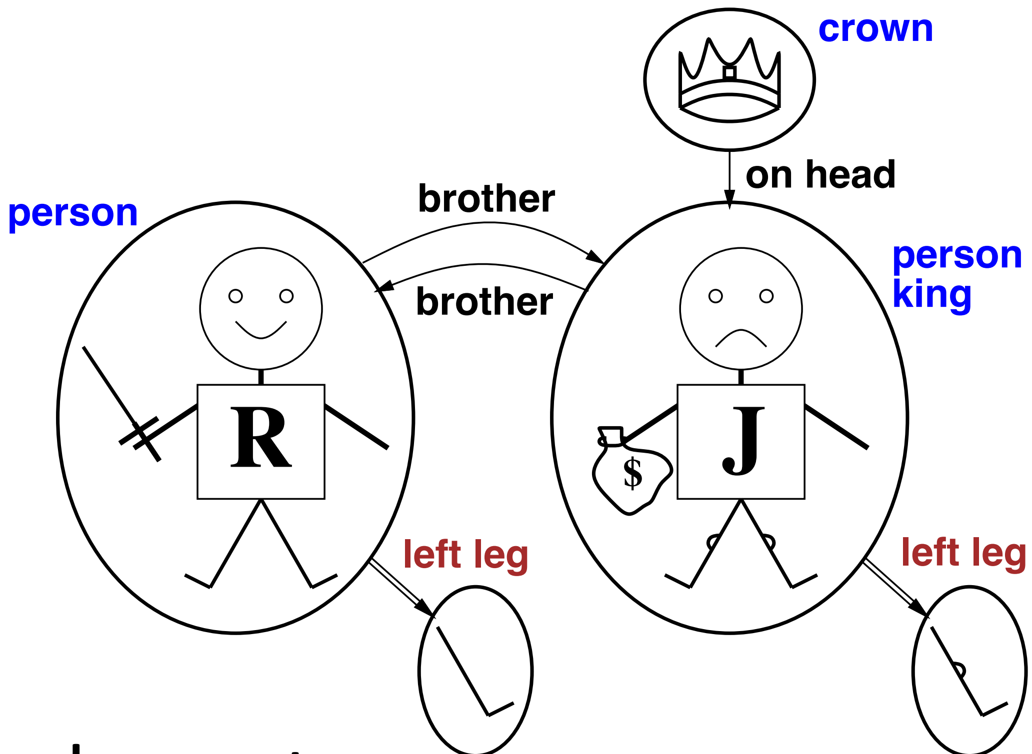
constant symbols $\rightarrow$ objects

predicate symbols $\rightarrow$ relations

function symbols $\rightarrow$ functional relations

An atomic sentence $predicate(term_1, \ldots, term_n)$ is true
iff the objects referred to by $term_1, \ldots, term_n$
are in the relation referred to by $predicate$

# Example: A World of Kings and Legs




- Syntactic elements:

**Constants:**

**Richard, John, RsLeftLeg, …**

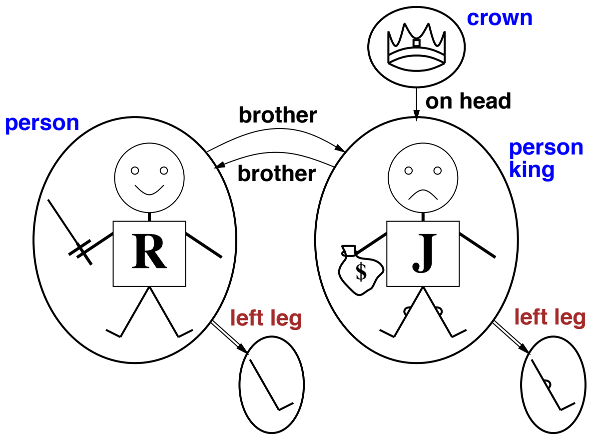
**Functions:**

**leftleg(.), onheadof(.), …**

**Relations:**

**On(.,.) IsKing(.), IsPerson(.), …**

# All Possible Models




We **can** enumerate the FOL models for a given KB vocabulary:

For each number of domain elements $n$ from $1$ to $\infty$
    For each $k$-ary predicate $P_k$ in the vocabulary
        For each possible $k$-ary relation on $n$ objects
            For each constant symbol $C$ in the vocabulary
                For each choice of referent for $C$ from $n$ objects . . .

Lesson: Computing entailment by enumerating models will be challenging!

# More Definitions

- Logical connectives: and, or, not, $\Rightarrow$, $\Leftrightarrow$
- Quantifiers:
  - $\forall$ For all (Universal quantifier)
  - $\exists$ There exists (Existential quantifier)
- Examples
  - George is a monkey and he is curious

    $Monkey(George) \wedge Curious(George)$
  - All monkeys are curious

    $\forall m\text{:}\ Monkey(m) \Rightarrow Curious(m)$
  - There is a curious monkey

    $\exists m\text{:}\ Monkey(m) \wedge Curious(m)$

# Quantifier / Connective Interaction

$M(x)$ == "x is a monkey"
$C(x)$ == "x is curious"

$\forall x: \; M(x) \wedge C(x)$

**"Everything is a curious monkey"**

$\forall x: \; M(x) \Rightarrow C(x)$

**"All monkeys are curious"**

$\exists x: \; M(x) \wedge C(x)$

**"There exists a curious monkey"**

$\exists x: \; M(x) \Rightarrow C(x)$

**"There exists an object that is *either* a curious monkey, *or* not a monkey at all"**

# Nested Quantifiers: Order matters!

$$\forall x\, \exists y\;\; P(x,y) \;\neq\; \exists y\, \forall x\, P(x,y)$$

- Example

  Every monkey has a tail

$$\forall m\, \exists t\;\; \text{has}(m,t)$$

Every monkey ***shares*** a tail!

$$\exists t\, \forall m\;\; \text{has}(m,t)$$

**Try:**

Everybody loves somebody *vs.* Someone is loved by everyone

$$\forall x\, \exists y\;\; \text{loves}(x, y) \qquad\qquad \exists y\, \forall x\;\; \text{loves}(x, y)$$

# Fun With Sentences

- Brothers are siblings.

$$\forall x, y \;\; Brother(x, y) \;\Rightarrow\; Sibling(x, y).$$

- "Sibling" is symmetric.

$$\forall x, y \;\; Sibling(x, y) \;\Leftrightarrow\; Sibling(y, x).$$

- One's mother is one's female parent.

$$\forall x, y \;\; Mother(x, y) \;\Leftrightarrow\; (Female(x) \wedge Parent(x, y)).$$

- A first cousin is a child of a parent's sibling.

$$\forall x, y \;\; FirstCousin(x, y) \;\Leftrightarrow\; \exists p, ps \;\; Parent(p, x) \wedge Sibling(ps, p) \wedge Parent(ps, y)$$

# Propositional. Logic *vs.* First Order

| | | |
|---|---|---|
| ***Ontology*** | Facts (P, Q,…) | Objects,<br>Properties,<br>Relations |
| ***Syntax*** | Atomic sentences<br>Connectives | Variables & quantification<br>Sentences have structure: terms<br>father-of(mother-of(X))) |
| ***Semantics*** | Truth Tables | Interpretations & Models<br>(Much more complicated) |
| ***Inference Algorithm*** | DPLL, WalkSAT<br>Fast in practice | Unification<br>Forward, Backward chaining<br>Prolog, theorem proving |
| ***Complexity*** | NP-Complete | Semi-decidable<br>May run forever if KB $\not\models \alpha$ |

# FOL Reasoning: Outline

- Basics of FOL reasoning
- Classes of FOL reasoning methods
  - Compilation to propositional logic
  - Forward & Backward Chaining
  - Resolution

# FOL Reasoning: Brief History

| | | |
|---|---|---|
| 450B.C. | Stoics | propositional logic, inference (maybe) |
| 322B.C. | Aristotle | "syllogisms" (inference rules), quantifiers |
| 1565 | Cardano | probability theory (propositional logic + uncertainty) |
| 1847 | Boole | propositional logic (again) |
| 1879 | Frege | first-order logic |
| 1922 | Wittgenstein | proof by truth tables |
| 1930 | Gödel | $\exists$ complete algorithm for FOL |
| 1930 | Herbrand | complete algorithm for FOL (reduce to propositional) |
| 1931 | Gödel | $\neg\exists$ complete algorithm for arithmetic |
| 1960 | Davis/Putnam | "practical" algorithm for propositional logic |
| 1965 | Robinson | "practical" algorithm for FOL—resolution |

# Basics: Universal Instantiation

- Universally quantified sentence:
  - $\forall x$: Monkey(x) $\Rightarrow$ Curious(x)
- Intuitively, x can be anything:
  - Monkey(George) $\Rightarrow$ Curious(George)
  - Monkey(473Student1) $\Rightarrow$ Curious(473Student1)
  - Monkey(DadOf(George)) $\Rightarrow$ Curious(DadOf(George))
- Formally:

$$\frac{\forall x \ \ S}{\text{Subst}(\{x/p\}, S)}$$

Example:

$$\frac{\forall x \ \ \text{Monkey}(x) \rightarrow \text{Curious}(x)}{\text{Monkey}(\text{George}) \rightarrow \text{Curious}(\text{George})}$$

**x is replaced with p in S, and the quantifier removed**

**x is replaced with George in S, and the quantifier removed**

# Basics: Existential Instantiation

- Existentially quantified sentence:

  $\exists x$: Monkey(x) $\wedge$ ¬Curious(x)

- Can we conclude:

  Monkey(George) $\wedge$ ¬Curious(George) ???

  No! S might not be true for George!

- Use a *Skolem Constant* and draw the conclusion:

  Monkey(K) $\wedge$ ¬Curious(K)

- Formally:

$$\frac{\exists x\ S}{\text{Subst}(\{x/K\}, S)}$$

**K is called a Skolem constant**

- Existential instantiation changes the KB, but still entails the same set of formulas!

# Reduction to Propositional Inference

Suppose the KB contains just the following:

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

Instantiating the universal sentence in **all possible** ways, we have

$$King(John) \wedge Greedy(John) \;\Rightarrow\; Evil(John)$$
$$King(Richard) \wedge Greedy(Richard) \;\Rightarrow\; Evil(Richard)$$
$$King(John)$$
$$Greedy(John)$$
$$Brother(Richard, John)$$

The new KB is propositionalized: proposition symbols are

$King(John),\; Greedy(John),\; Evil(John), King(Richard)$ etc.

# Reduction to Propositional Inference

Claim: a ground sentence* is entailed by new KB iff entailed by original KB

Claim: every FOL KB can be propositionalized so as to preserve entailment

Idea: propositionalize KB and query, apply resolution, return result

Problem: with function symbols, there are infinitely many ground terms,
e.g., $Father(Father(Father(John)))$

Theorem: Herbrand (1930). If a sentence $\alpha$ is entailed by an FOL KB,
it is entailed by a **finite** subset of the propositional KB

Idea: For $n = 0$ to $\infty$ do
create a propositional KB by instantiating with depth-$n$ terms
see if $\alpha$ is entailed by this KB

Problem: works if $\alpha$ is entailed, loops if $\alpha$ is not entailed

Theorem: Turing (1936), Church (1936), entailment in FOL is semidecidable

# Problems with Propositionalization

Propositionalization seems to generate lots of irrelevant sentences.
E.g., from

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

it seems obvious that $Evil(John)$, but propositionalization produces lots of facts such as $Greedy(Richard)$ that are irrelevant

With $p$ $k$-ary predicates and $n$ constants, there are $p \cdot n^k$ instantiations

With function symbols, it gets nuch much worse!

# Motivation for Unification

- What if we want to use modus ponens?

  Propositional Logic:

$$\frac{a \wedge b, \quad a \wedge b \Rightarrow c}{c}$$

- In First-Order Logic?

$$\frac{\forall x\ Monkey(x) \Rightarrow Curious(x) \qquad Monkey(George)}{????}$$

- Must "*unify*" x with George:

  Need to substitute {x/George} in Monkey(x)
  $\Rightarrow$ Curious(x) to infer Curious(George)

# Unification Examples

We can get the inference immediately if we can find a substitution $\theta$ such that $King(x)$ and $Greedy(x)$ match $King(John)$ and $Greedy(y)$

$\theta = \{x/John, y/John\}$ works

$\textsc{Unify}(\alpha, \beta) = \theta$ if $\alpha\theta = \beta\theta$

| $p$ | $q$ | $\theta$ |
|---|---|---|
| $Knows(John, x)$ | $Knows(John, Jane)$ | $\{x/Jane\}$ |
| $Knows(John, x)$ | $Knows(y, OJ)$ | $\{x/OJ, y/John\}$ |
| $Knows(John, x)$ | $Knows(y, Mother(y))$ | $\{y/John, x/Mother(John)\}$ |
| $Knows(John, x)$ | $Knows(x, OJ)$ | $fail$ |

Standardizing apart eliminates overlap of variables, e.g., $Knows(z_{17}, OJ)$

# Generalized Modus Ponens (GMP)

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

where $p_i{}'\theta = p_i\theta$ for all $i$

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

GMP used with KB of definite clauses (**exactly** one positive literal)
All variables assumed universally quantified

# Knowledge Base Example

- Knowledge: The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.
- Goal: Prove that Col. West is a criminal – $Criminal(West)$

$$American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \Rightarrow Criminal(x)$$

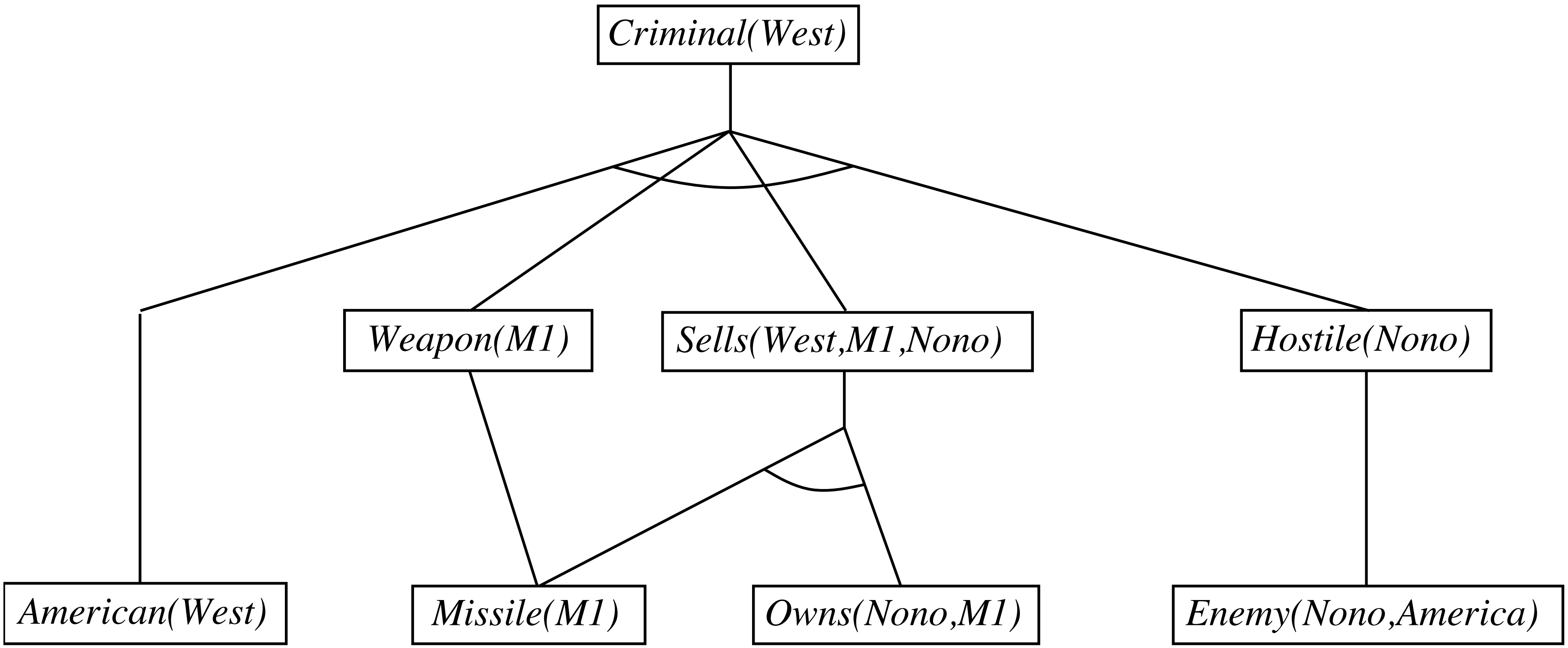
$$Owns(Nono, M_1) \qquad Missile(M_1)$$

$$Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$$

$$American(West) \qquad Enemy(Nono, America)$$

$$Missile(x) \Rightarrow Weapon(x) \qquad Enemy(x, America) \Rightarrow Hostile(x)$$

# Forward/Backward Chaining




$$American(x) \wedge Weapon(y) \wedge Sells(x, y, z) \wedge Hostile(z) \Rightarrow Criminal(x)$$

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

$$Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$$

$$American(West) \qquad Enemy(Nono, America)$$

$$Missile(x) \Rightarrow Weapon(x) \qquad Enemy(x, America) \Rightarrow Hostile(x)$$

# First-order Resolution

Full first-order version:

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

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

For example,

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

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

Apply resolution steps to $CNF(KB \wedge \neg\alpha)$; complete for FOL

# Conversion to CNF (Part 1)

Everyone who loves all animals is loved by someone:

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

1. Eliminate biconditionals and implications

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

2. Move $\neg$ inwards: $\neg\forall x, p \;\; \equiv \exists x \;\; \neg p$, $\;\; \neg\exists x, p \;\; \equiv \forall x \;\; \neg p$:

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

# Conversion to CNF (cont.)

3. Standardize variables: each quantifier should use a different one

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

4. Skolemize: a more general form of existential instantiation.
   Each existential variable is replaced by a Skolem function
   of the enclosing universally quantified variables:

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

5. Drop universal quantifiers:

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

6. Distribute $\land$ over $\lor$:

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

# A Resolution Proof