Artificial Intelligence (CS303)

Lecture 6: First-Order Logic

Hints for this lecture

- **We prefer representations that are similar to natural language (at least sometimes).**

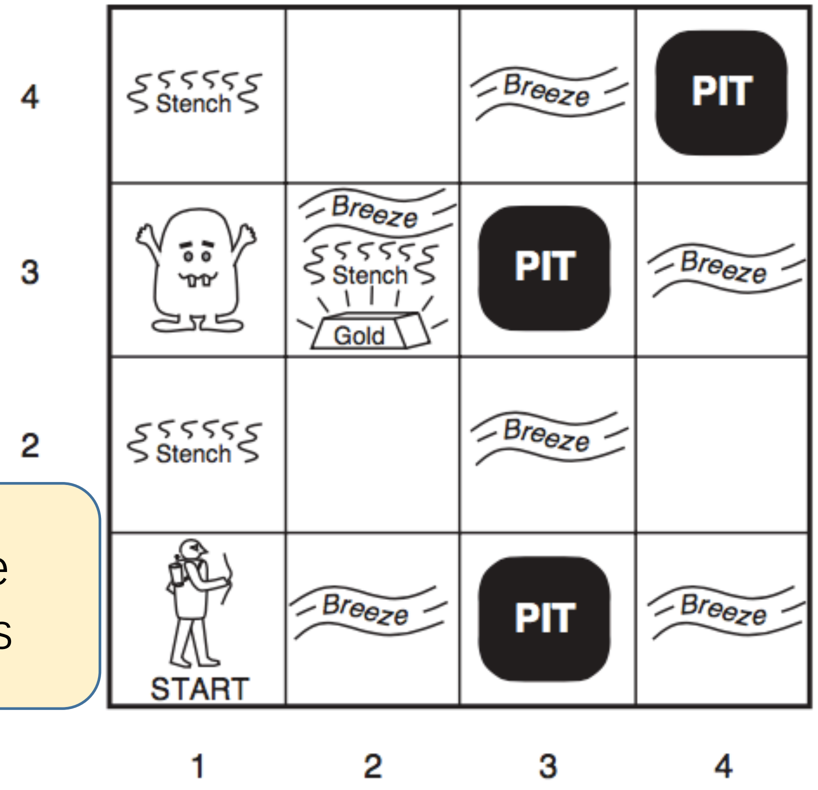
Outline of this lecture

- **Definitions**
- **Knowledge Base Engineering**
- **Inference by Propositionalization**
- **Proof by Resolution**

I. Definitions

Why need FOL

- A different knowledge representation that
  - might be easier to construct KB
  - or to inference
  - more natural to human thoughts

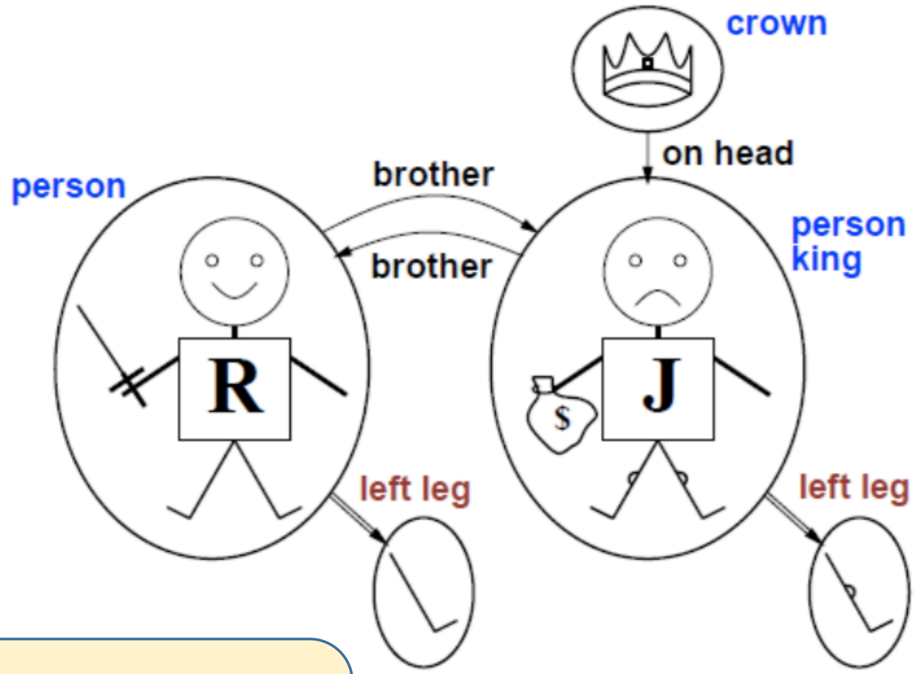


Boring to enumerate events for all squares

“Viewpoint” of FOL

- **The world is a “graph”**

Objects	people, houses, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries . . .
Relations	red, round, bogus, prime, multistoried, brother of, bigger than, inside, part of, has color, occurred after, owns, comes between,
Function	father of, best friend, third inning of, one more than, end of . . .



What is model in such a definition?

What is the advantages?

A few more thoughts about logic

Language	Ontological Commitment	Epistemological Commitment
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Probability theory	facts	degree of belief
Fuzzy logic	facts	degree of truth known interval value

Syntax of FOL (Overview)

$$
\begin{array}{rcl}
\textit{Sentence} & \rightarrow & \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
\textit{AtomicSentence} & \rightarrow & \textit{Predicate} \mid \textit{Predicate}(\textit{Term}, \ldots) \mid \textit{Term} = \textit{Term} \\
\textit{ComplexSentence} & \rightarrow & (\ \textit{Sentence}\ ) \mid [\ \textit{Sentence}\ ] \\
 & \mid & \neg\ \textit{Sentence} \\
 & \mid & \textit{Sentence} \land \textit{Sentence} \\
 & \mid & \textit{Sentence} \lor \textit{Sentence} \\
 & \mid & \textit{Sentence} \Rightarrow \textit{Sentence} \\
 & \mid & \textit{Sentence} \Leftrightarrow \textit{Sentence} \\
 & \mid & \textit{Quantifier}\ \textit{Variable}, \ldots\ \textit{Sentence} \\
 & & \\
\textit{Term} & \rightarrow & \textit{Function}(\textit{Term}, \ldots) \\
 & \mid & \textit{Constant} \\
 & \mid & \textit{Variable} \\
 & & \\
\textit{Quantifier} & \rightarrow & \forall \mid \exists \\
\textit{Constant} & \rightarrow & A \mid X_1 \mid \textit{John} \mid \cdots \\
\textit{Variable} & \rightarrow & a \mid x \mid s \mid \cdots \\
\textit{Predicate} & \rightarrow & \textit{True} \mid \textit{False} \mid \textit{After} \mid \textit{Loves} \mid \textit{Raining} \mid \cdots \\
\textit{Function} & \rightarrow & \textit{Mother} \mid \textit{LeftLeg} \mid \cdots \\
\textsc{Operator Precedence} & : & \neg, =, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

Syntax of FOL

Constants	$KingJohn,\ 2,\ UCB, \ldots$
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*

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

Syntax of FOL

Complex sentences are made from atomic sentences using connectives

$$\neg S, \quad S_1 \wedge S_2, \quad S_1 \vee S_2, \quad S_1 \Rightarrow S_2, \quad S_1 \Leftrightarrow S_2$$

E.g. $Sibling(KingJohn, Richard) \Rightarrow Sibling(Richard, KingJohn)$

$>(1,2) \vee \leq(1,2)$

$>(1,2) \wedge \neg{>}(1,2)$

Syntax of FOL

Sentences are true with respect to a model and an interpretation

Model contains ≥ 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 $predicate$

Syntax of FOL - Universal/Existential Quantification and Equality

$\forall \langle variables \rangle \; \langle sentence \rangle$
Everyone at Berkeley is smart:
$\forall x \; At(x, Berkeley) \Rightarrow Smart(x)$
$\forall x \; P$ is equivalent to the conjunction of instantiations of P

$$\begin{array}{ll} & At(KingJohn, Berkeley) \Rightarrow Smart(KingJohn) \\ \wedge & At(Richard, Berkeley) \Rightarrow Smart(Richard) \\ \wedge & At(Berkeley, Berkeley) \Rightarrow Smart(Berkeley) \\ \wedge & \ldots \end{array}$$

Typically, $\Rightarrow$ is the main connective with $\forall$.
Common mistake: using $\wedge$ as the main connective with $\forall$:

$$\forall x \; At(x, Berkeley) \wedge Smart(x)$$

means "Everyone is at Berkeley and everyone is smart"

$\exists \langle variables \rangle \; \langle sentence \rangle$
Someone at Stanford is smart:
$\exists x \; At(x, Stanford) \wedge Smart(x)$
$\exists x \; P$ is equivalent to the disjunction of instantiations of P

$$\begin{array}{ll} & At(KingJohn, Stanford) \wedge Smart(KingJohn) \\ \vee & At(Richard, Stanford) \wedge Smart(Richard) \\ \vee & At(Stanford, Stanford) \wedge Smart(Stanford) \\ \vee & \ldots \end{array}$$

Typically, $\wedge$ is the main connective with $\exists$.
Common mistake: using $\Rightarrow$ as the main connective with $\exists$:

$$\exists x \; At(x, Stanford) \Rightarrow Smart(x)$$

is true if there is anyone who is not at Stanford!

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

$Father(John) = Henry$ implies $Father(John)$ and $Henry$ refers to the same object

used with negation to insist two terms are not the same object

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

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

II. Knowledge Base Engineering

Creating a Knowledge Base

- Identify the Task
- Assemble the relevant knowledge
- Decide on a vocabulary of predicates, functions and constants
- Encode general knowledge about the domain (rules)
- Encode a description of the problem
- Pose queries to the inference procedure and get answers

Examples for Encoding Knowledge/Rules

"One's mother is one's female parent"

$$\forall m, c \; Mother(c) = m \;\Leftrightarrow\; Female(m) \wedge Parent(m, c)$$

"A sibling is another child of one's parents"

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

"Wendy is female"

$$Female(Wendy)$$

These are axioms

III. Inference by Propositionalization

Inference with FOL

- **naïve idea: reduce to propositional logic**

Universal Instantiation

Every instantiation of a universally quantified sentence is entailed by it:

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

for any variable v and ground term g

E.g., $\forall x \ \ King(x) \land Greedy(x) \ \Rightarrow \ Evil(x)$ yields

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

UI can be applied several times to **add** new sentences; the new *KB* is logically equivalent to the old

EI can be applied once to **replace** the existential sentence; the new *KB* is not equivalent to the old, but is satisfiable if the old *KB* was satisfiable

Existential Instantiation

For any sentence α, variable v, and constant symbol k
that does not appear elsewhere in the knowledge base:

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

E.g., $\exists x \ \ Crown(x) \land OnHead(x, John)$ yields

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

provided C_1 is a new constant symbol, called a Skolem constant

Another example: from $\exists x \ \ d(x^y)/dy = x^y$ we obtain

$$d(e^y)/dy = e^y$$

provided e is a new constant symbol

Example: Reduction to Propositional Logic

Suppose the KB contains just the following:

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

Instantiating the universal sentence in all possible ways, we have

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

The new KB is **propositionalized**: proposition symbols are

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

Problem with Reduction

Claim: a ground sentence $\star$ 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 α is entailed by an $FOL\ KB$, it is entailed by a finite subset of the propositional KB

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

Problem: works if α is entailed, loops if α is not entailed

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

Representation unlikely changes the complexity, this is because FOL expresses a more complicated world.

Problem with Reduction

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 much much worse!

IV. Proof by Resolution

Unification

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 much much worse!

Unification

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

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

$UNIFY(\alpha, \beta) = \theta$ if $\alpha\theta = \beta\theta$

Idea: Unify rule premises with known facts, apply unifier to conclusion

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

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

Unification

$$\frac{p_1{}',\ p_2{}',\ \ldots,\ p_n{}',\ (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{q\sigma} \qquad \text{where } p_i{}'\sigma = p_i\sigma \text{ for all } i$$

E.g.
$$\begin{aligned}
p_1{}' &= \quad Faster(Bob, Pat) \\
p_2{}' &= \quad Faster(Pat, Steve) \\
p_1 \wedge p_2 \ \Rightarrow\ q &= \quad Faster(x, y) \wedge Faster(y, z) \ \Rightarrow\ Faster(x, z) \\
\sigma &= \quad \{x/Bob, y/Pat, z/Steve\} \\
q\sigma &= \quad Faster(Bob, Steve)
\end{aligned}$$

GMP used with KB of definite clauses (exactly one positive literal):
either a single atomic sentence or
(conjunction of atomic sentences) $\Rightarrow$ (atomic sentence)
All variables assumed universally quantified

Resolution Inference Rule (again)

Entailment in first-order logic is only semidecidable:

can find a proof of α if $KB \models \alpha$

cannot always prove that $KB \not\models \alpha$

Cf. Halting Problem: proof procedure may be about to terminate with success or failure, or may go on for ever

Resolution is a refutation procedure:

to prove $KB \models \alpha$, show that $KB \wedge \neg\alpha$ is unsatisfiable

Resolution uses KB, $\neg\alpha$ in CNF (conjunction of clauses)

Resolution inference rule combines two clauses to make a new one:

Inference continues until an empty clause is derived (contradiction)

Resolution Inference Rule (again)

Basic propositional version:

$$\frac{\alpha \vee \beta, \quad \neg\beta \vee \gamma}{\alpha \vee \gamma} \quad \text{or equivalently} \quad \frac{\neg\alpha \Rightarrow \beta, \quad \beta \Rightarrow \gamma}{\neg\alpha \Rightarrow \gamma}$$

Full first-order version:

$$\frac{\begin{array}{c} p_1 \vee \ldots\ p_j\ \ldots \vee p_m, \\ q_1 \vee \ldots\ q_k\ \ldots \vee q_n \end{array}}{(p_1 \vee \ldots\ p_{j-1} \vee p_{j+1}\ \ldots p_m \vee q_1 \ldots\ q_{k-1} \vee q_{k+1}\ \ldots \vee q_n)\sigma}$$

where $p_j\sigma = \neg q_k\sigma$

For example,

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

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

Conversion to CNF

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

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

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

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

Conversion to CNF

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 $\lor$:

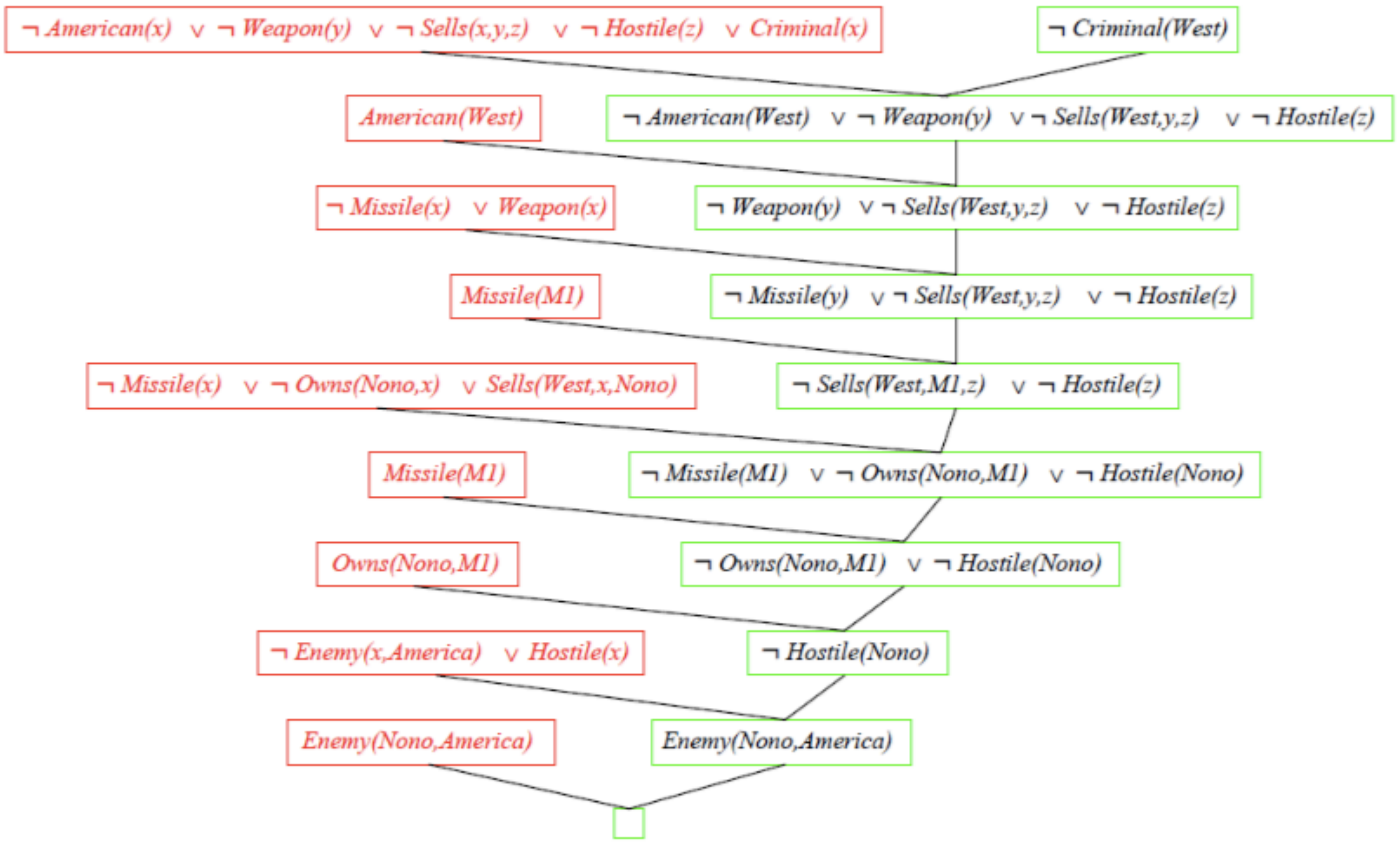
$$[Animal(F(x)) \lor Loves(G(x), x)] \land [\neg Loves(x, F(x)) \lor Loves(G(x), x)]$$

Proof by Resolution (again)

To prove α:

- negate it
- convert to CNF
- add to CNF KB
- infer contradiction

Proof by Resolution (again)



Summary

- **Three ways for inference with FOL**
  - **Propositionalization**
  - **Unification**

    => Resolution

    => Chaining Algorithms

To be continued