First-Order Logic

Compiled from multiple sources

Why First-Order Logic?

- So far, we studied the simplest logic: propositional logic
- But for some applications, propositional logic is not expressive enough
- First-order logic is more expressive: allows representing more complex facts and making more sophisticated inferences

A Motivating Example

- For instance, consider the statement “Anyone who drives fast gets a speeding ticket”
- From this, we should be able to conclude ”If Joe drives fast, he will get a speeding ticket”
- Similarly, we should be able to conclude ”If Rachel drives fast, she will get a speeding ticket” and so on.
- But Propositional Logic does not allow inferences like that because we cannot talk about concepts like ”everyone”, ”someone” etc.
- First-order logic (predicate logic) allows making such kinds of inferences

Building Blocks of FOL

- The building blocks of propositional logic were propositions
- In first-order logic, there are three kinds of basic building blocks: constants, variables, predicates
- Constants: refer to specific objects (in a universe of discourse)
- Examples: George, 6, Austin, CS311, ...
- Variables: range over objects (in a universe of discourse)
- Examples: x,y,z, ...
- If universe of discourse is cities in Texas, x can represent Houston, Austin, Dallas, San Antonio, ...

Building Blocks of FOL

- Predicates describe properties of objects or relationships between objects
- Examples: ishappy, betterthan, loves, $>$ ...
- Predicates can be applied to both constants and variables
- Examples: ishappy(George), betterthan(x,y), loves(George, Rachel), $x > 3$, ...
- A predicate $P(x)$ is true or false depending on whether property P holds for x
- Example: ishappy(George) is true if George is happy, but false otherwise

Predicate Examples

- Consider predicate $Q(x, y)$ which indicates that $x = y + 3$
- What is the truth value of $Q(3, 0)$?
- What is the truth value of $Q(1, 2)$?

Formulas in FOL

- Formulas in first-order logic are formed using predicates and logical connectives.
- Example: $\text{even}(x) \lor \text{odd}(x)$ is a formula
- Example: $(\text{odd}(x) \rightarrow \neg\, \text{even}(x)) \land \text{even}(x)$

Semantics of FOL

- In propositonal logic, the truth value of formula depends on a truth assignment to variables.
- In FOL, truth value of a formula depends interpretation of predicate symbols and variables over some domain D
- Consider a FOL formula $\neg P(x)$
- A possible interpretation:

$$D = \{\star, \circ\}, P(\star) = \text{true}, P(\circ) = \text{false}, x = \star$$

- Under this interpretation, what's truth value of $\neg P(x)$?
- What about if $x = \circ$?

More Examples

- Consider interpretation I over domain $D = \{1, 2\}$
  - $P(1,1) = P(1,2) = \text{true}, \; P(2,1) = P(2,2) = \text{false}$
  - $Q(1) = \text{false}, \; Q(2) = \text{true}$
  - $x = 1, \; y = 2$
- What is truth value of $P(x, y) \wedge Q(y)$ under I?
- What is truth value of $P(y, x) \rightarrow Q(y)$ under I?
- What is truth value of $P(x, y) \rightarrow Q(x)$ under I?

Quantifiers

- Real power of first-order logic over propositional logic: quantifiers
- Quantifiers allow us to talk about all objects or the existence of some object
- There are two quantifiers in first-order logic:
  1. Universal quantifier ($\forall$): refers to all objects
  2. Existential quantifier ($\exists$): refers to some object

Universal Quantifiers

- Universal quantification of $P(x)$, $\forall x.P(x)$, is the statement "P(x) holds for all objects x in the universe of discourse."
- $\forall x.P(x)$ is true if predicate P is true for every object in the universe of discourse, and false otherwise
- Consider domain $D = \{\circ, \star\}$, $P(\circ) = \text{true}, P(\star) = \text{false}$
- What is truth value of $\forall x.P(x)$?
- Object o for which $P(o)$ is false is counterexample of $\forall x.P(x)$
- What is a counterexample for $\forall x.P(x)$ in previous example?

More Universal Quantifier Examples

- Consider the domain D of real numbers and predicate $P(x)$ with interpretation $x^2 \geq x$
- What is the truth value of $\forall x.P(x)$?
- What is a counterexample?
- What if the domain is integers?
- Observe: Truth value of a formula depends on a universe of discourse!

Existential Quantifiers

- Existential quantification of $P(x)$, written $\exists x.P(x)$, is "There exists an element x in the domain such that $P(x)$".
- $\exists x.P(x)$ is true if there is at least one element in the domain such that $P(x)$ is true
- In first-order logic, domain is required to be non-empty.
- Consider domain $D = \{\circ, \star\}$, $P(\circ) = \text{true}, P(\star) = \text{false}$
- What is truth value of $\exists x.P(x)$?

Existential Quantifier Examples

- Consider the domain of reals and predicate $P(x)$ with interpretation $x < 0$.
- What is the truth value of $\exists x. P(x)$?
- What if domain is positive integers?
- Let $Q(y)$ be the statement $y > y^2$
- What's truth value of $\exists y. Q(y)$ if domain is reals?
- What about if domain is integers?

Quantified Formulas

- So far, only discussed how to quantify individual predicates.
- But we can also quantify entire formulas containing multiple predicates and logical connectives.
- $\exists x.(\text{even}(x) \land \text{gt}(x, 100))$ is a valid formula in FOL
- What's truth value of this formula if domain is all integers?
  - assuming $\text{even}(x)$ means "x is even" and $\text{gt}(x, y)$ means $x > y$
- What about $\forall x.(\text{even}(x) \lor \text{gt}(x, 100))$?

More Examples of Quantified Formulas

- Consider the domain of integers and the predicates even(x) and div4(x) which represents if x is divisible by 4
- What is the truth value of the following quantified formulas?
  - $\forall x.\ (div4(x) \rightarrow even(x))$
  - $\forall x.\ (even(x) \rightarrow div4(x))$
  - $\exists x.\ (\neg div4(x) \land even(x))$
  - $\exists x.\ (\neg div4(x) \rightarrow even(x))$
  - $\forall x.\ (\neg div4(x) \rightarrow even(x))$

Translating English into Quantified Formulas

Assuming $\text{freshman}(x)$ means "x is a freshman" and $\text{inCS311}(x)$ "x is taking CS311", express the following in FOL

- Someone in CS311 is a freshman
- No one in CS311 is a freshman
- Everyone taking CS311 are freshmen
- Every freshman is taking CS311

DeMorgan's Laws for Quantifiers

- Learned about DeMorgan's laws for propositional logic:

$$\begin{array}{rcl} \neg(p \wedge q) & \equiv & \neg p \vee \neg q \\ \neg(p \vee q) & \equiv & \neg p \wedge \neg q \end{array}$$

- DeMorgan's laws extend to first-order logic, e.g.,
  $\neg(even(x) \vee div4(x)) \equiv (\neg even(x) \wedge \neg div4(x))$

- Two new DeMorgan's laws for quantifiers:

$$\begin{array}{rcl} \neg\forall x.P(x) & \equiv & \exists x.\neg P(x) \\ \neg\exists x.P(x) & \equiv & \forall x.\neg P(x) \end{array}$$

- When you push negation in, $\forall$ flips to $\exists$ and vice versa

Using DeMorgan's Laws

- Expressed "Noone in CS311 is a freshman" as $\neg\exists x.(\text{inCS311}(x) \land \text{freshman}(x))$
- Let's apply DeMorgan's law to this formula:
- Using the fact that $p \rightarrow q$ is equivalent to $\neg p \lor q$, we can write this formula as:
- Therefore, these two formulas are equivalent!

Nested Quantifiers

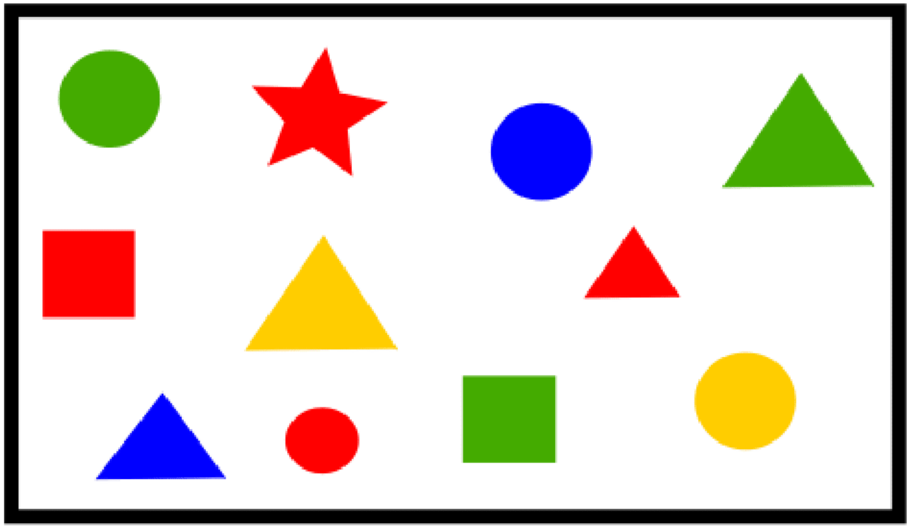
- Sometimes may be necessary to use multiple quantifiers
- For example, can't express "Everybody loves someone" using a single quantifier
- Suppose predicate $\text{loves}(x, y)$ means "Person x loves person y"
- What does $\forall x.\exists y.\text{loves}(x, y)$ mean?
- What does $\exists y.\forall x.\text{loves}(x, y)$ mean?
- Observe: Order of quantifiers is very important!

More Nested Quantifier Examples

Using the loves(x,y) predicate, how can we say the following?

- "Someone loves everyone"
- "There is somone who doesn't love anyone"
- "There is someone who is not loved by anyone"
- "Everyone loves everyone"
- "There is someone who doesn't love herself/himself."

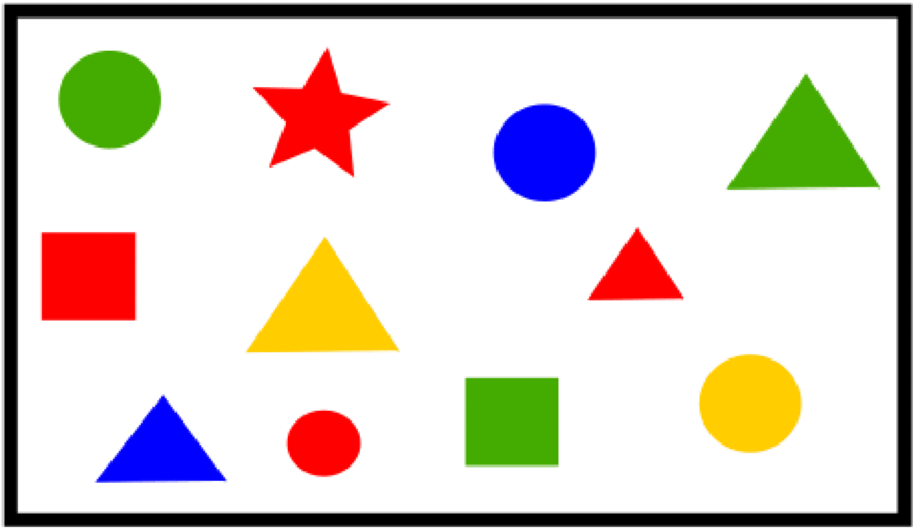
Understanding Quantifiers



Which formulas are true/false? If false, give a counterexample

- $\forall x. \exists y.\ (\text{sameShape}(x, y) \wedge \text{differentColor}(x, y))$
- $\forall x. \exists y.\ (\text{sameColor}(x, y) \wedge \text{differentShape}(x, y))$
- $\forall x.\ (\text{triangle}(x) \rightarrow (\exists y.\ (\text{circle}(y) \wedge \text{sameColor}(x, y))))$

Understanding Quantifiers contd.



Which formulas are true/false? If false, give a counterexample

- $\forall x.\forall y.\ ((\text{triangle}(x) \land \text{square}(y)) \rightarrow \text{sameColor}(x, y))$
- $\exists x.\forall y.\neg\text{sameShape}(x, y)$
- $\forall x.\ (\text{circle}(x) \rightarrow (\exists y.(\neg\text{circle}(y) \land \text{sameColor}(x, y))))$

Well-Formed Formula (wff)

- A **free variable** is a variable that isn't bound by a quantifier
  - $\exists$**`y Likes(x,y)`**

    `x` is free, **`y`** is bound

- A **well-formed formula** is a sentence where all variables are quantified

Summary so far

- **Term**: Constant, variable, or Function($term_1$, …, $term_n$)

  denotes an object in the world

  Ground Term has no variables
- **Atom**: Predicate($term_1$, …, $term_n$), $term_1$ = $term_2$

  is smallest expression assigned a truth value
- **Sentence**: atom, quantified sentence with variables or complex sentence using connectives is assigned a truth value
- **Well-Formed Formula** (wff):

  sentence where all variables are quantified

Equivalence

- Two formulas F_1 and F_2 are equivalent if $F_1 \leftrightarrow F_2$ is valid
- In PL, we could prove equivalence using truth tables, but not possible in FOL
- However, we can still use known equivalences to rewrite one formula as the other
- Example: Prove that $\neg(\forall x.\ (P(x) \rightarrow Q(x)))$ and $\exists x.\ (P(x) \land \neg Q(x))$ are equivalent.
- Example: Prove that $\neg\exists x.\forall y.P(x, y)$ and $\forall x.\exists y.\neg P(x, y)$ are equivalent.

Rules of Inference

- Proof rules are written as rules of inference:

$$\frac{\begin{array}{c}\text{Hypothesis1}\\ \text{Hypothesis2}\\ \ldots\end{array}}{\text{Conclusion}}$$

- An example inference rule:

$$\frac{\begin{array}{c}\text{All men are mortal}\\ \text{Socrates is a man}\end{array}}{\therefore \text{Socrates is mortal}}$$

- We'll learn about more general inference rules that will allow constructing formal proofs

Modus Ponens

- Most basic inference rule is modus ponens:

$$\frac{\begin{array}{l}\phi_1 \\ \phi_1 \rightarrow \phi_2\end{array}}{\phi_2}$$

- Modus ponens applicable to both propositional logic and first-order logic

Example Uses of Modus Ponens

- Application of modus ponens in propositional logic:

$$\begin{array}{l} p \wedge q \\ (p \wedge q) \rightarrow r \\ \hline \end{array}$$

- Application of modus ponens in first-order logic:

$$\begin{array}{l} P(a) \\ P(a) \rightarrow Q(b) \\ \hline \end{array}$$

Modus Tollens

- Second imporant inference rule is modus tollens:

$$\frac{\begin{array}{l}\phi_1 \rightarrow \phi_2 \\ \neg\phi_2\end{array}}{\neg\phi_1}$$

Example Uses of Modus Tollens

- Application of modus tollens in propositional logic:

$$\begin{array}{l} p \rightarrow (q \vee r) \\ \neg(q \vee r) \\ \hline \end{array}$$

- Application of modus tollens in first-order logic:

$$\begin{array}{l} Q(a) \\ \neg P(a) \rightarrow \neg Q(a) \\ \hline \end{array}$$

Hypothetical Syllogism (HS)

$$\frac{\begin{array}{c}\phi_1 \rightarrow \phi_2 \\ \phi_2 \rightarrow \phi_3\end{array}}{\phi_1 \rightarrow \phi_3}$$

- Basically says "implication is transitive"
- Example:

$$\frac{\begin{array}{c}P(a) \rightarrow Q(b) \\ Q(b) \rightarrow R(c)\end{array}}{}$$

Or Introduction and Elimination

- Or introduction:

$$\frac{\phi_1}{\phi_1 \vee \phi_2}$$

- Example application: "Socrates is a man. Therefore, either Socrates is a man or there are red elephants on the moon."

- Or elimination:

$$\frac{\begin{array}{l}\phi_1 \vee \phi_2 \\ \neg\phi_2\end{array}}{\phi_1}$$

- Example application: "It is either a dog or a cat. It is not a dog. Therefore, it must be a cat."

And Introduction and Elimination

- And introduction:

$$\frac{\begin{array}{c}\phi_1 \\ \phi_2\end{array}}{\phi_1 \wedge \phi_2}$$

- And elimination:

$$\frac{\phi_1 \wedge \phi_2}{\phi_1}$$

Resolution

- Final inference rule: resolution

$$\frac{\begin{array}{c}\phi_1 \lor \phi_2 \\ \neg\phi_1 \lor \phi_3\end{array}}{\phi_2 \lor \phi_3}$$

- To see why this is correct, observe ϕ_1 is either true or false.
- Suppose ϕ_1 is true. Then, $\neg\phi_1$ is false. Therefore, by second hypothesis, ϕ_3 must be true.
- Suppose ϕ_1 is false. Then, by 1st hypothesis, ϕ_2 must be true.
- In any case, either ϕ_2 or ϕ_3 must be true; $\therefore$ $\phi_2 \lor \phi_3$

Resolution Example

- Example 1:

$$\begin{array}{l} P(a) \vee \neg Q(b) \\ Q(b) \vee R(c) \\ \hline \end{array}$$

- Example 2:

$$\begin{array}{c} p \vee q \\ q \vee \neg p \\ \hline \end{array}$$

Summary

Name	Rule of Inference
Modus ponens	$\dfrac{\phi_1 \quad \phi_1 \rightarrow \phi_2}{\phi_2}$
Modus tollens	$\dfrac{\phi_1 \rightarrow \phi_2 \quad \neg\phi_2}{\neg\phi_1}$
Hypothetical syllogism	$\dfrac{\phi_1 \rightarrow \phi_2 \quad \phi_2 \rightarrow \phi_3}{\phi_1 \rightarrow \phi_3}$
Or introduction	$\dfrac{\phi_1}{\phi_1 \vee \phi_2}$
Or elimination	$\dfrac{\phi_1 \vee \phi_2 \quad \neg\phi_2}{\phi_1}$
And introduction	$\dfrac{\phi_1 \quad \phi_2}{\phi_1 \wedge \phi_2}$
And elimination	$\dfrac{\phi_1 \wedge \phi_2}{\phi_1}$
Resolution	$\dfrac{\phi_1 \vee \phi_2 \quad \neg\phi_1 \vee \phi_3}{\phi_2 \vee \phi_3}$

Using the Rules of Inference

Assume the following hypotheses:

1. It is not sunny today and it is colder than yesterday.
2. We will go to the lake only if it is sunny.
3. If we do not go to the lake, then we will go hiking.
4. If we go hiking, then we will be back by sunset.

Show these lead to the conclusion: "We will be back by sunset."

Encoding in Logic

- First, encode hypotheses and conclusion as logical formulas.
- To do this, identify propositions used in the argument:
  - s = "It is sunny today"
  - c= "It is colder than yesterday"
  - l = "We'll go to the lake"
  - h = "We'll go hiking"
  - b= "We'll be back by sunset"

Encoding in Logic contd.

- "It's not sunny today and colder than yesterday."
- "We will go to the lake only if it is sunny"
- "If we do not go to the lake, then we will go hiking."
- "If we go hiking, then we will be back by sunset."
- Conclusion: "We'll be back by sunset"

Formal Proof Using Inference Rules

1.	$\neg s \wedge c$	Hypothesis
2.	$l \rightarrow s$	Hypothesis
3.	$\neg l \rightarrow h$	Hypothesis
4.	$h \rightarrow b$	Hypothesis

Additional Inference Rules for Quantified Formulas

- Inference rules we learned so far are sufficient for reasoning about quantifier-free statements
- Four more inference rules for making deductions from quantified formulas
- These come in pairs for each quantifier (universal/existential)
- One is called generalization, the other one called instantiation

Universal Instantiation

- If we know something is true for all members of a group, we can conclude it is also true for a specific member of this group
- This idea is formally called universal instantiation:

$$\frac{\forall x.P(x)}{P(c)} \quad \text{(for any c)}$$

- If we know "All CS classes at UT are hard", universal instantiation allows us to conclude "CS311 is hard"!

Example

- Consider predicates man(x) and mortal(x) and the hypotheses:

  1. All men are mortal:
  2. Socrates is a man:

- Using rules of inference, prove mortal(Socrates)

Universal Generalization

- Suppose we can prove a claim for an arbitrary element in the domain.
- Since we've made no assumptions about this element, proof should apply to all elements in the domain.
- This correct reasoning is captured by universal generalization

$$\frac{P(c) \text{ for arbitrary c}}{\forall x. P(x)}$$

Example

Prove $\forall x.Q(x)$ from the hypotheses:

1. $\forall x.\ (P(x) \to Q(x))$ Hypothesis
2. $\forall x.\ P(x)$ Hypothesis
3. $P(c) \to Q(c)$ $\forall$-inst (1)
4. $P(c)$ $\forall$-inst (2)
5. $Q(c)$ Modus ponens (3), (4)
6. $\forall x.Q(x)$ $\forall$-gen (5)

Caveat About Universal Generalization

- When using universal generalization, need to ensure that c is truly arbitrary!
- If you prove something about a specific person Mary, you cannot make generalizations about all people

$$\cancel{\frac{\text{even}(2)}{\forall x.\text{even}(x)}}$$

Existential Instantiation

- Consider formula $\exists x.P(x)$.
- We know there is some element, say c, in the domain for which $P(c)$ is true.
- This is called existential instantiation:

$$\frac{\exists x.P(x)}{P(c)} \text{ (for unused c)}$$

- Here, c is a fresh name (i.e., not used before in proof).
  - Otherwise, can prove non-sensical things such as: "There exists some animal that can fly. Thus, rabbits can fly"!

Existential Instantiation Example

Consider the hypotheses $\exists x.P(x)$ and $\forall x.\neg P(x)$. Prove that we can derive a contradiction (i.e., false) from these hypotheses.

1.	$\exists x.P(x)$	Hypothesis
2.	$\forall x.\neg P(x)$	Hypothesis
3.		
4.		
5.		
6.		

Existential Generalization

- Suppose we know $P(c)$ is true for some constant c
- Then, there exists an element for which P is true
- Thus, we can conlude $\exists x. P(x)$
- This inference rule called existential generalization:

$$\frac{P(c)}{\exists x. P(x)}$$

Existential Generalization Example

Consider the hypotheses $atUT(George)$ and $smart(George)$.
Prove $\exists x.\ (atUT(x) \wedge smart(x))$

1.	$atUT(George)$	Hypothesis
2.	$smart(George)$	Hypothesis
3.		
4.		

Summary of Inference Rules for Quantifiers

Name	Rule of Inference
Universal Instantiation	$\dfrac{\forall x. P(x)}{P(c)} \quad (\text{any}\, c)$
Universal Generalization	$\dfrac{P(c) \quad (\text{for arbitrary}\, c)}{\forall x. P(x)}$
Existential Instantiation	$\dfrac{\exists x. P(x)}{P(c) \text{ for fresh } c}$
Existential Generalization	$\dfrac{P(c)}{\exists x. P(x)}$

Example 1

- Prove that these hypotheses imply $\exists x.(P(x) \land \neg B(x))$:

  1. $\exists x.\ (C(x) \land \neg B(x))$ (Hypothesis)
  2. $\forall x.\ (C(x) \to P(x))$ (Hypothesis)

Example 2

- Prove the below hypotheses are contradictory by deriving false

1. $\forall x.(P(x) \rightarrow (Q(x) \land S(x)))$ (Hypothesis)
2. $\forall x.(P(x) \land R(x))$ (Hypothesis)
3. $\exists x.(\neg R(x) \lor \neg S(x))$ (Hypothesis)

Example 3

Prove $\exists x.\ father(x, Evan)$ from the following premises:

1. $\forall x. \forall y.\ ((\mathrm{parent}(x, y) \land male(x)) \rightarrow father(x, y))$
2. $\mathrm{parent}(Tom, Evan)$
3. $\mathrm{male}(Tom)$

Inference in FOL

Universal instantiation (UI)

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

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

Existential instantiation (EI)

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

Existential instantiation contd.

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 iff the old KB was satisfiable

Reduction to propositional inference

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.

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!

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

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

p	q	θ
$Knows(John, x)$	$Knows(John, Jane)$	
$Knows(John, x)$	$Knows(y, OJ)$	
$Knows(John, x)$	$Knows(y, Mother(y))$	
$Knows(John, x)$	$Knows(x, OJ)$	

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

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

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

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

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

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

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

$\text{UNIFY}(\alpha, \beta) = \theta$ if $\alpha\theta = \beta\theta$

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

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

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

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., $Knows(z_{17}, OJ)$

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 $Greedy(x)$
θ 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

Example knowledge base

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.

Prove that Col. West is a criminal

Example knowledge base contd.

. . . it is a crime for an American to sell weapons to hostile nations:

Example knowledge base contd.

. . . it is a crime for an American to sell weapons to hostile nations:

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

Nono . . . has some missiles

Example knowledge base contd.

. . . it is a crime for an American to sell weapons to hostile nations:

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

Nono . . . has some missiles, i.e., $\exists\, x \; Owns(Nono, x) \land Missile(x)$:

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

. . . all of its missiles were sold to it by Colonel West

Example knowledge base contd.

. . . it is a crime for an American to sell weapons to hostile nations:

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

Nono . . . has some missiles, i.e., $\exists\, x\ Owns(Nono, x) \wedge Missile(x)$:

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

. . . all of its missiles were sold to it by Colonel West

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

Missiles are weapons:

Example knowledge base contd.

. . . it is a crime for an American to sell weapons to hostile nations:

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

Nono . . . has some missiles, i.e., $\exists\, x\ Owns(Nono, x) \land Missile(x)$:

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

. . . all of its missiles were sold to it by Colonel West

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

Missiles are weapons:

$Missile(x) \Rightarrow Weapon(x)$

An enemy of America counts as "hostile":

Example knowledge base contd.

. . . it is a crime for an American to sell weapons to hostile nations:

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

Nono . . . has some missiles, i.e., $\exists\, x\ Owns(Nono, x) \wedge Missile(x)$:

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

. . . all of its missiles were sold to it by Colonel West

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

Missiles are weapons:

$$Missile(x) \Rightarrow Weapon(x)$$

An enemy of America counts as “hostile”:

$$Enemy(x, America) \ \Rightarrow\ Hostile(x)$$

West, who is American . . .

$$American(West)$$

The country Nono, an enemy of America . . .

$$Enemy(Nono, America)$$

Forward chaining algorithm

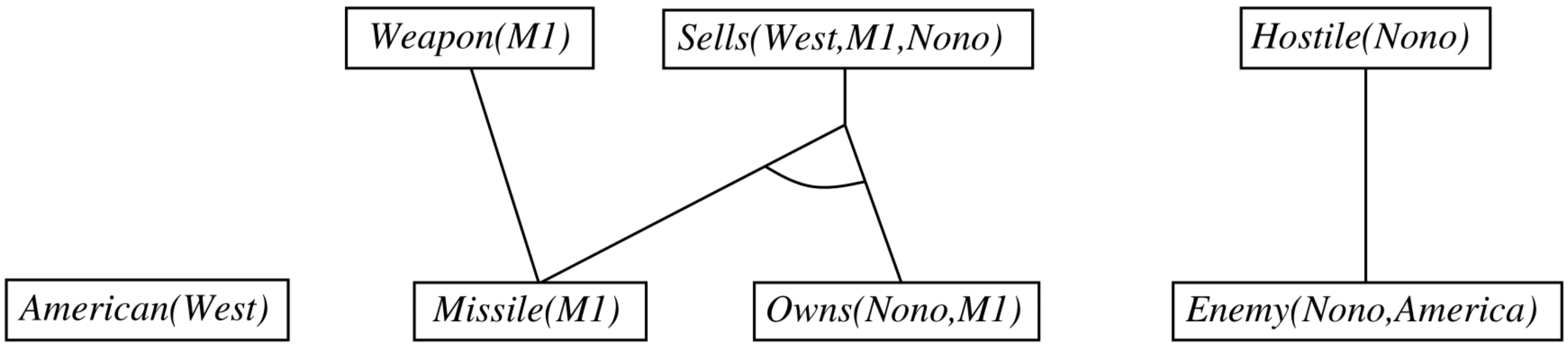
function FOL-FC-ASK(KB, α) **returns** a substitution or $false$

    repeat until new is empty
        $new \leftarrow \{\,\}$
        for each sentence r **in** KB **do**
            $(\,p_1 \land \ldots \land\ p_n\ \Rightarrow\ \ q) \leftarrow$ STANDARDIZE-APART(r)
            for each θ such that $(p_1\ \land\ \ldots\ \land\ p_n)\theta\ =\ (p'_1\ \land\ \ldots\ \land\ p'_n)\theta$
                        for some $p'_1, \ldots, p'_n$ in KB
                $q' \leftarrow$ SUBST(θ, q)
                if q' is not a renaming of a sentence already in KB or new **then do**
                    add q' to new
                    $\phi \leftarrow$ UNIFY(q', α)
                    if ϕ is not $fail$ **then return** ϕ
        add new to KB
    return $false$

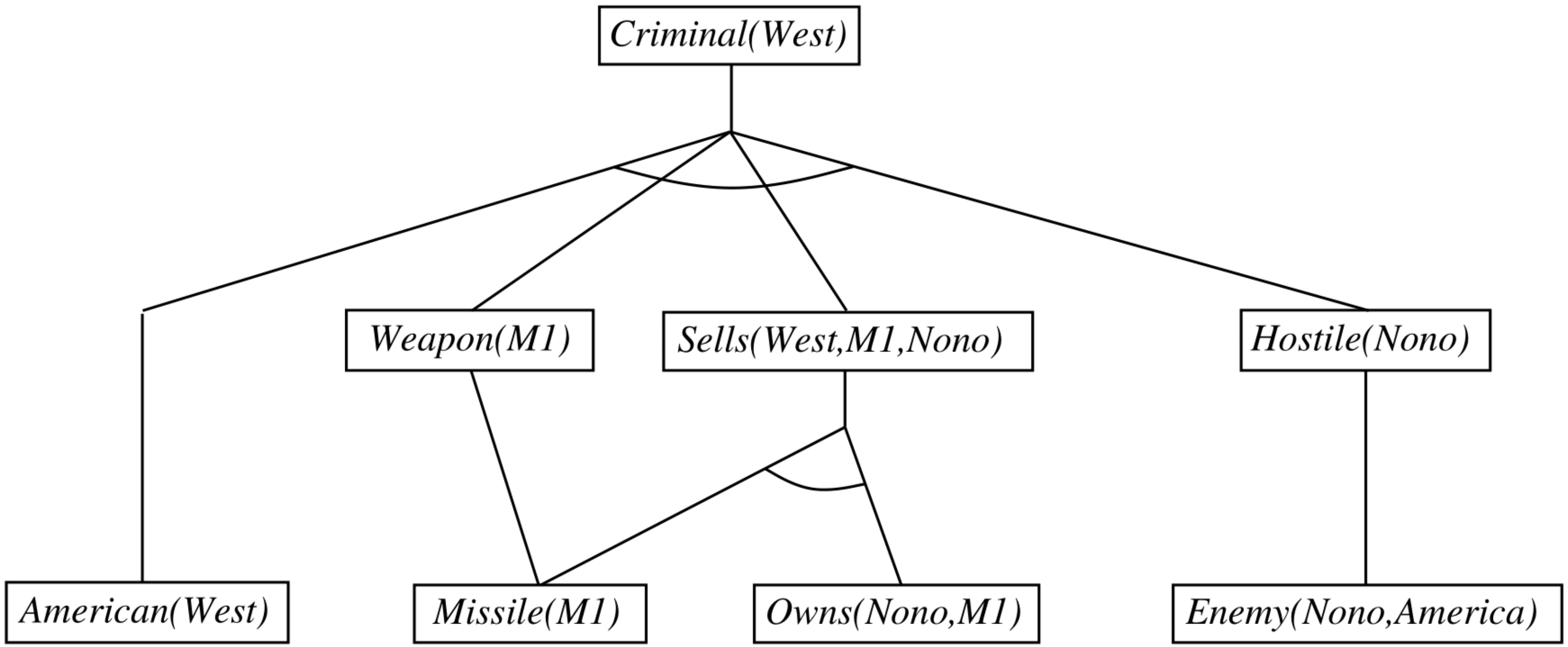
Forward chaining proof

American(West) *Missile(M1)* *Owns(Nono,M1)* *Enemy(Nono,America)*

Forward chaining proof



Forward chaining proof



Properties of forward chaining

Sound and complete for first-order definite clauses
(proof similar to propositional proof)

Datalog = first-order definite clauses + **no functions** (e.g., crime KB)
FC terminates for Datalog in poly iterations: at most $p \cdot n^k$ literals

May not terminate in general if α is not entailed

This is unavoidable: entailment with definite clauses is semidecidable

Backward chaining algorithm

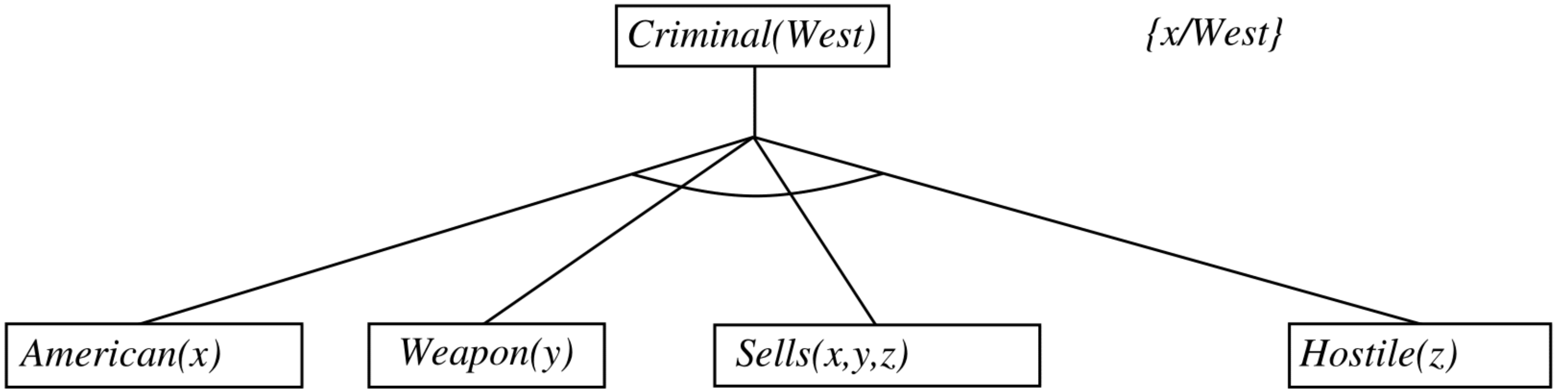
function FOL-BC-ASK($KB, goals, \theta$) **returns** a set of substitutions
  inputs: KB, a knowledge base
    $goals$, a list of conjuncts forming a query (θ already applied)
    θ, the current substitution, initially the empty substitution $\{\,\}$
  local variables: $answers$, a set of substitutions, initially empty

  if $goals$ is empty **then return** $\{\theta\}$
  $q' \leftarrow \text{SUBST}(\theta, \text{FIRST}(goals))$
  for each sentence r **in** KB
    where $\text{STANDARDIZE-APART}(r) = (\,p_1 \wedge \ldots \wedge p_n \Rightarrow q)$
    and $\theta' \leftarrow \text{UNIFY}(q, q')$ succeeds
   $new\_goals \leftarrow [\,p_1, \ldots, p_n | \text{REST}(goals)]$
   $answers \leftarrow \text{FOL-BC-ASK}(KB, new\_goals, \text{COMPOSE}(\theta', \theta)) \cup answers$
  return $answers$

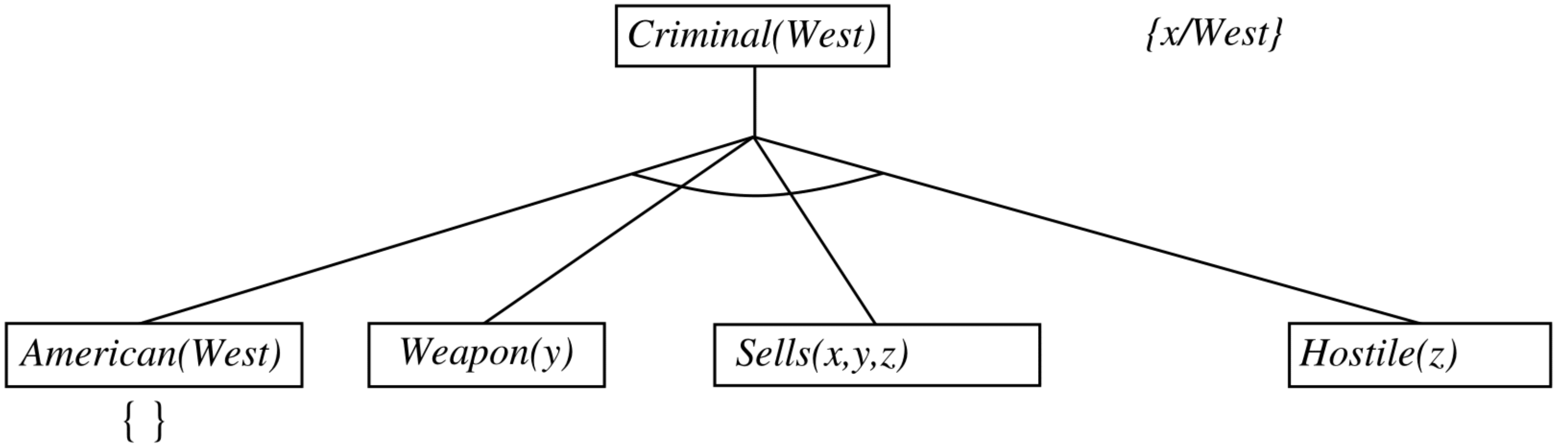
Backward chaining example

$\boxed{Criminal(West)}$

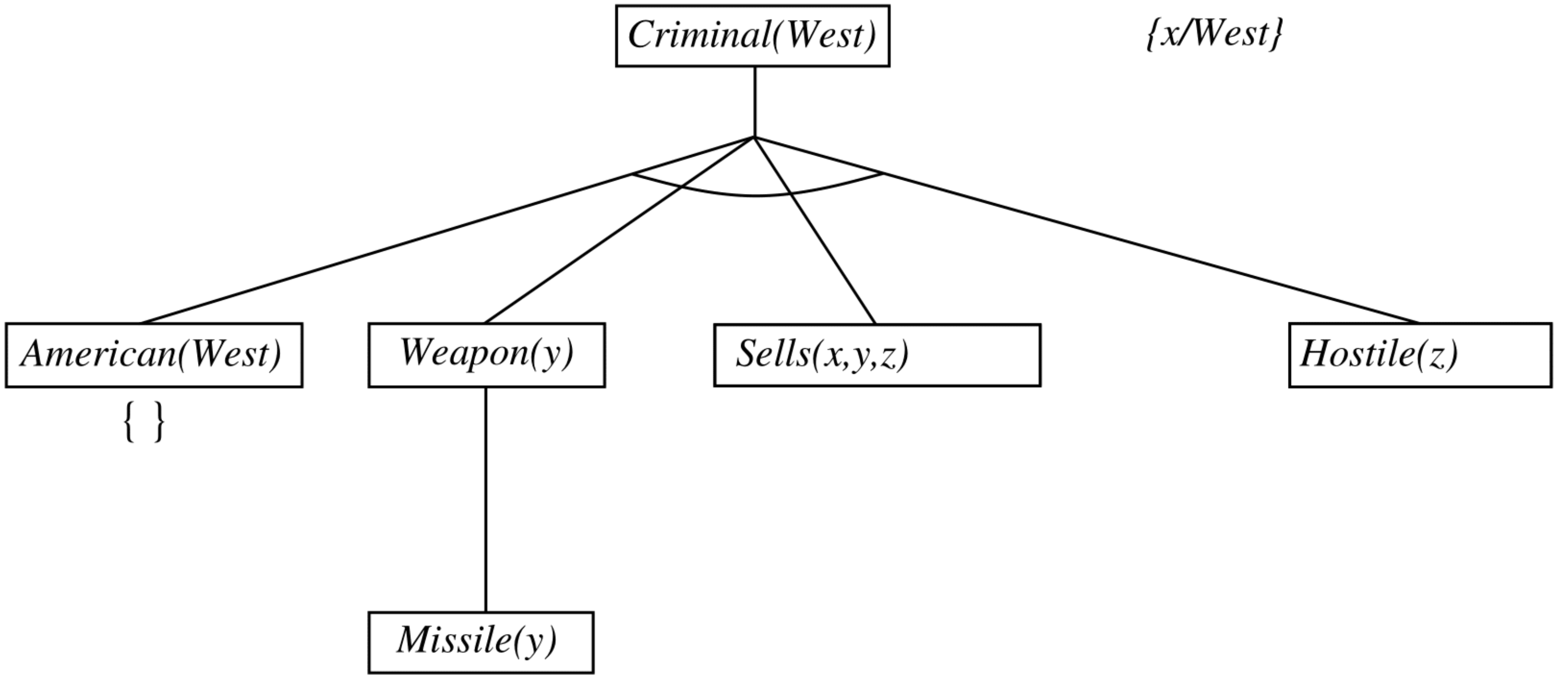
Backward chaining example



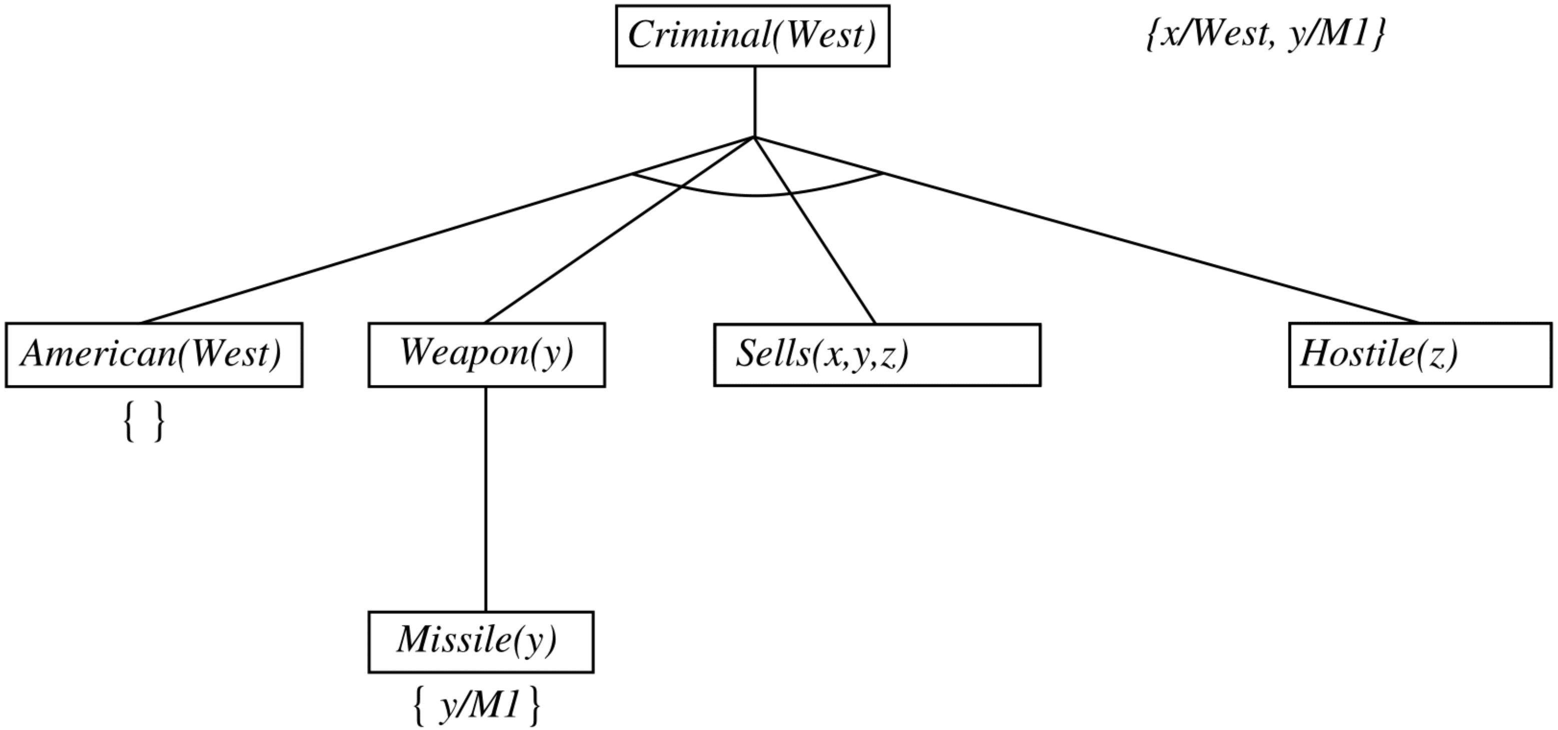
Backward chaining example



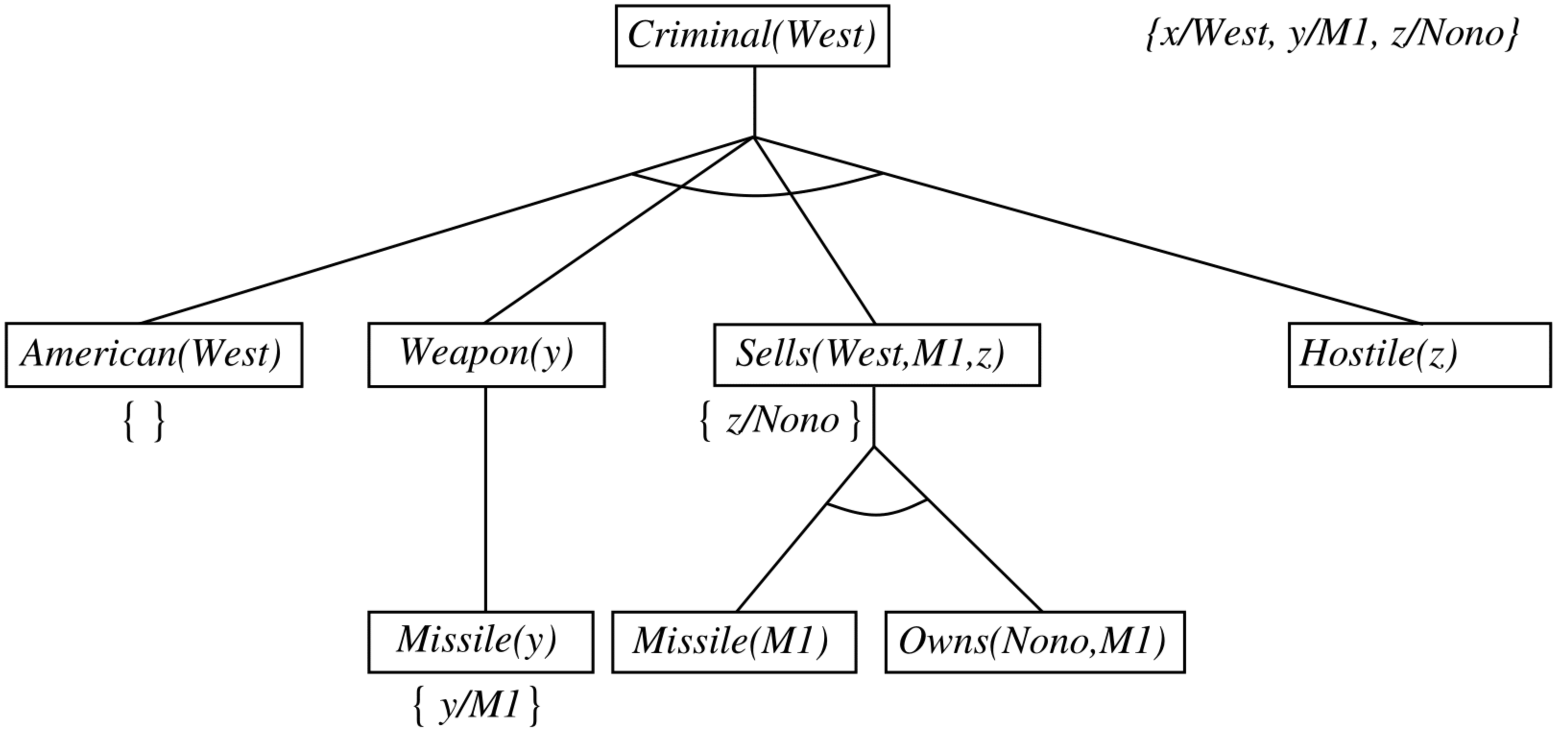
Backward chaining example



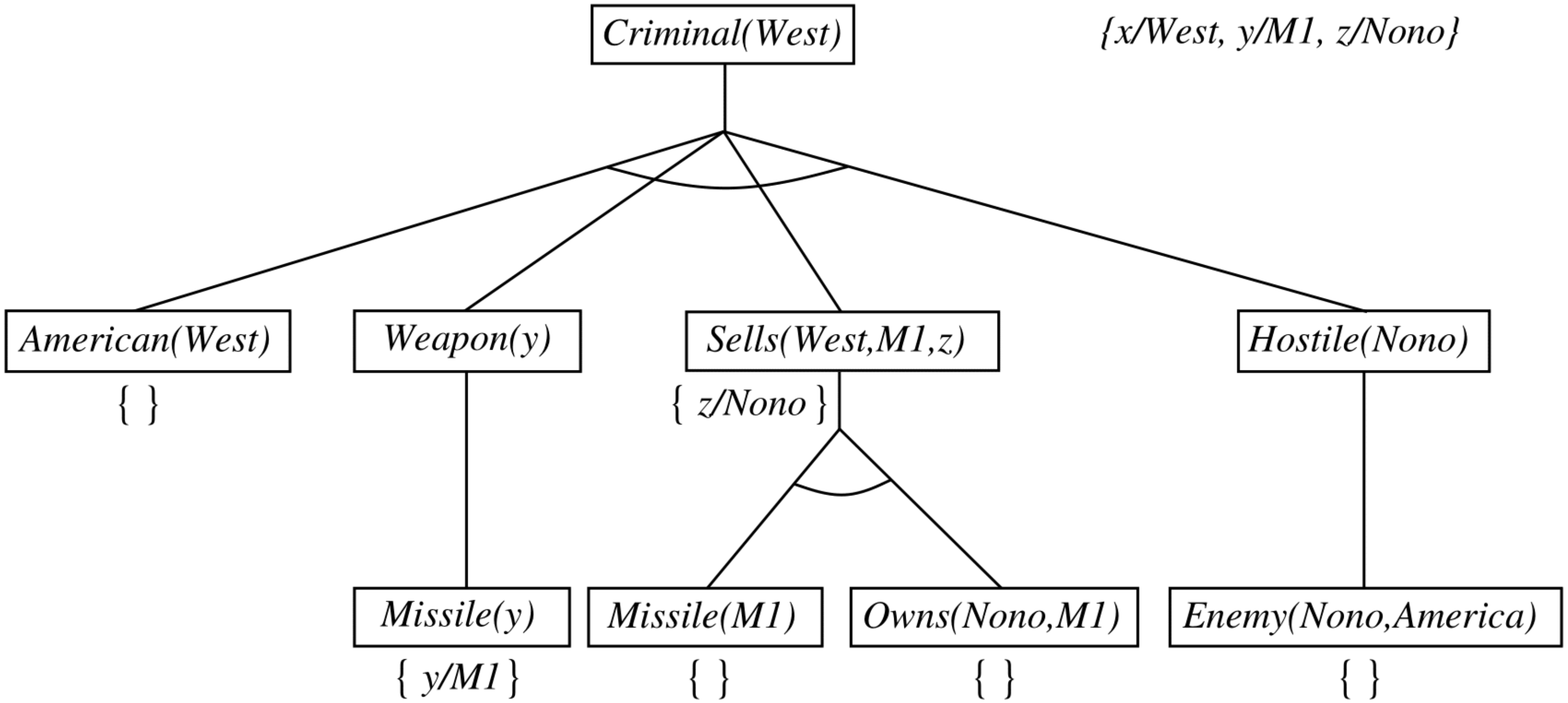
Backward chaining example



Backward chaining example



Backward chaining example



Properties of backward chaining

Depth-first recursive proof search: space is linear in size of proof

Incomplete due to infinite loops
$\Rightarrow$ fix by checking current goal against every goal on stack

Inefficient due to repeated subgoals (both success and failure)
$\Rightarrow$ fix using caching of previous results (extra space!)

Widely used (without improvements!) for logic programming

Resolution: brief summary

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

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

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 contd.

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 $\lor$:

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

Resolution proof: definite clauses