Mansoura University
Faculty of Computers and Information
Department of Computer Science

First-Order Logic

DR. Muhammad Haggag Zayyan

CS Department

OBJECTIVES

- Why FOL?
- Syntax and semantics of FOL
- Using FOL

PROS AND CONS OF PROPOSITIONAL LOGIC

☺ Propositional logic is declarative

☺ Propositional logic allows partial/disjunctive/negated information

- (unlike most data structures and databases)

☺ Propositional logic is compositional:

- meaning of $B_{1,1} \wedge P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$

☺ Meaning in propositional logic is context-independent

- (unlike natural language, where meaning depends on context)

☹ 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

FIRST-ORDER LOGIC

- Whereas propositional logic assumes the world contains facts,
- first-order logic (like natural language) assumes the world contains
  - Objects: people, houses, numbers, colors, baseball games, wars, …
  - Relations (Predicates): red, round, prime, brother of, bigger than, part of, comes between, …
  - Functions: father of, best friend, one more than, plus, …

SYNTAX OF FOL

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

A **function** takes one or more arguments and returns a **value**.
A **predicate** takes one or more arguments, and is either **true** or **false**

UNIVERSAL QUANTIFICATION

- $\forall$<*variables*> <*sentence*>

Everyone at NUS is smart:
$\forall x\ At(x,NUS) \Rightarrow Smart(x)$

- $\forall x\ P$ is true in a model m iff P is true with x being each possible object in the model

Roughly speaking, equivalent to the conjunction of instantiations of P

$$
\begin{aligned}
& At(KingJohn,NUS) \Rightarrow Smart(KingJohn) \\
\wedge\ & At(Richard,NUS) \Rightarrow Smart(Richard) \\
\wedge\ & At(NUS,NUS) \Rightarrow Smart(NUS) \\
\wedge\ & \ldots
\end{aligned}
$$

Sentences are true with respect to a model and an interpretation.

Model contains objects (domain elements) and relations among them.

A COMMON MISTAKE TO AVOID

Typically, $\Rightarrow$ is the main connective with $\forall$

- Common mistake: using $\wedge$ as the main connective with $\forall$:

  $\forall x\ At(x,NUS) \wedge Smart(x)$

  means "Everyone is at NUS and everyone is smart"

EXISTENTIAL QUANTIFICATION

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

- Someone at NUS is smart:
- $\exists x$ At(x,NUS) $\wedge$ Smart(x)$

- $\exists x$ *P* is true in a model *m* iff *P* is true with *x* being some possible object in the model

- Roughly speaking, equivalent to the disjunction of instantiations of *P*

  At(KingJohn,NUS) $\wedge$ Smart(KingJohn)
  $\vee$ At(Richard,NUS) $\wedge$ Smart(Richard)
  $\vee$ At(NUS,NUS) $\wedge$ Smart(NUS)
  $\vee$ ...

ANOTHER COMMON MISTAKE TO AVOID

- Typically, $\wedge$ is the main connective with $\exists$

- Common mistake: using $\Rightarrow$ as the main connective with $\exists$:

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

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

PROPERTIES OF QUANTIFIERS

- $\forall x\ \forall y$ is the same as $\forall y\ \forall x$
- $\exists x\ \exists y$ is the same as $\exists y\ \exists x$

- $\exists x\ \forall y$ is not the same as $\forall y\ \exists x$

- $\exists x\ \forall y$ Loves(x,y)
  - “There is a person who loves everyone in the world”

- $\forall y\ \exists x$ Loves(x,y)
  - “Everyone in the world is loved by at least one person”

- Quantifier duality: each can be expressed using the other
- $\forall x$ Likes(x,IceCream) $\neg\exists x\ \neg$Likes(x,IceCream)
- $\exists x$ Likes(x,Broccoli) $\neg\forall x\ \neg$Likes(x,Broccoli)

WHAT IS UNIFICATION?

- Unification is a process of making two different logical atomic expressions identical by finding a substitution. Unification depends on the substitution process.
- It takes two literals as input and makes them identical using substitution.
- **Example 1:** Unify(King(x), King(John))
  - Substitution θ = (John/x) is a unifier for these atoms and applying this substitution, and both expressions will be identical.
- **Example 2:** Unify(P(x, y), P(a, f(z)))
  - Substitution θ = { a/x, f(z)/y }
- **Example 3:** Unify(knows(Richard, x), knows(Richard, John))
  - Substitution θ = { John/x }
- **Example 4:** Unify(Q(a, g(x, a), f(y)), Q(a, g(f(b), a), x))
  - Substitution θ = { a/a, f(b)/x, b/y }

CONDITIONS FOR UNIFICATION

- **Following are some basic conditions for unification:**

  - Predicate symbol must be same, atoms or expression with different predicate symbol can never be unified.
  - Number of Arguments in both expressions must be identical.
  - Unification will fail if there are two similar variables present in the same expression.

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:

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

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

$\boxed{Owns(Nono, M_1)}$ and $\boxed{Missile(M_1)}$

... 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":

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

West, who is American ...

$\boxed{American(West)}$

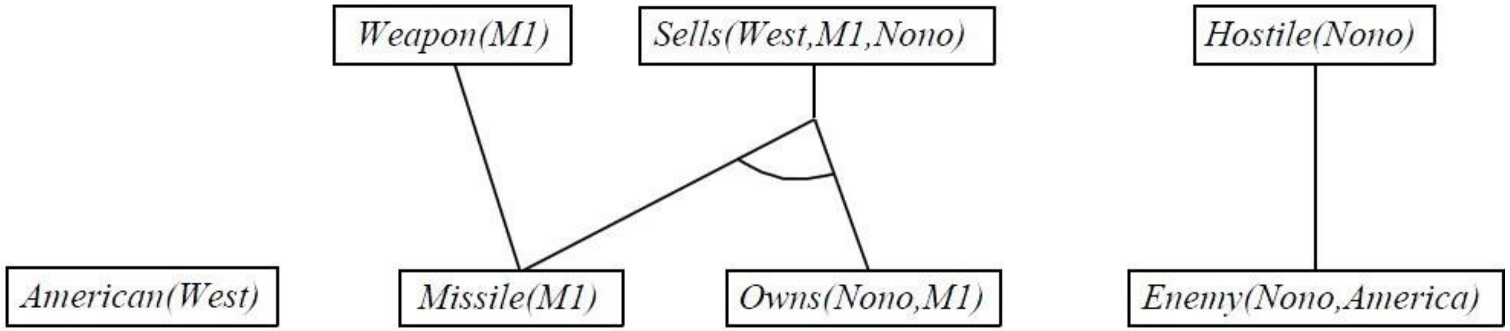
The country Nono, an enemy of America ...

$\boxed{Enemy(Nono, America)}$

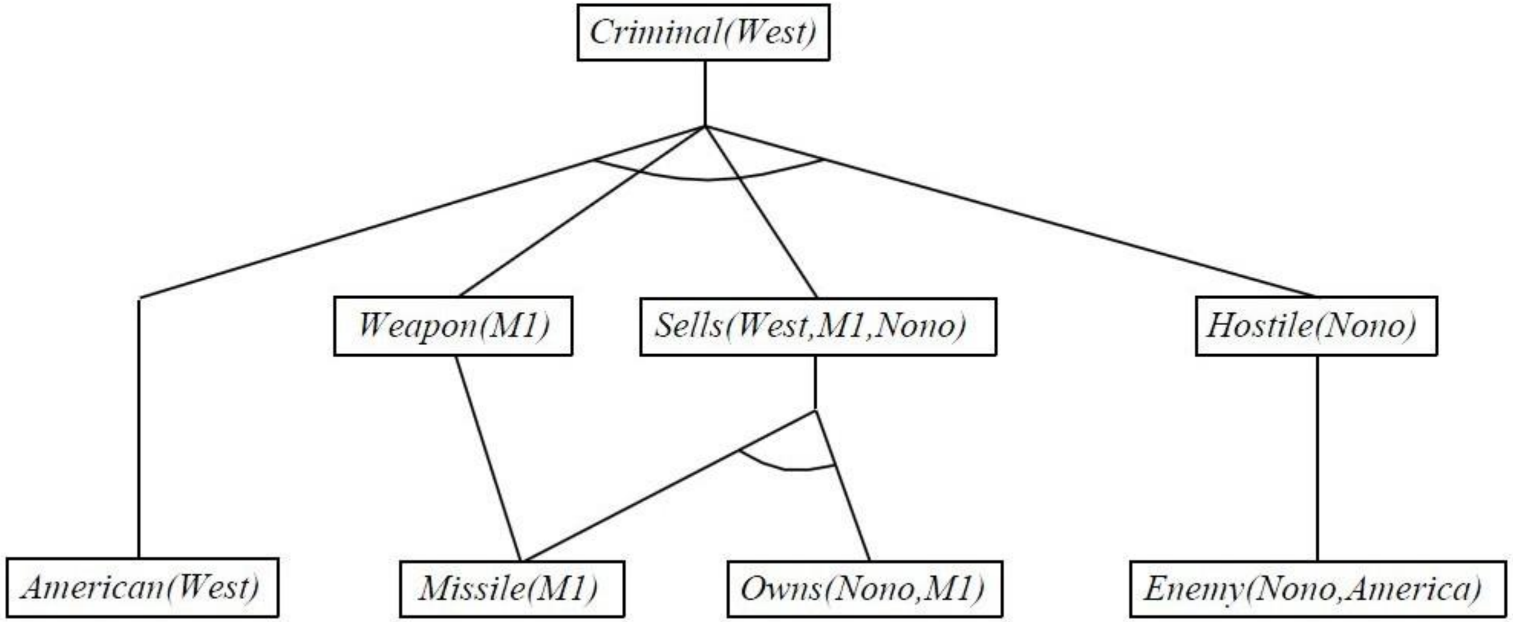
Forward chaining proof

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

Forward chaining proof



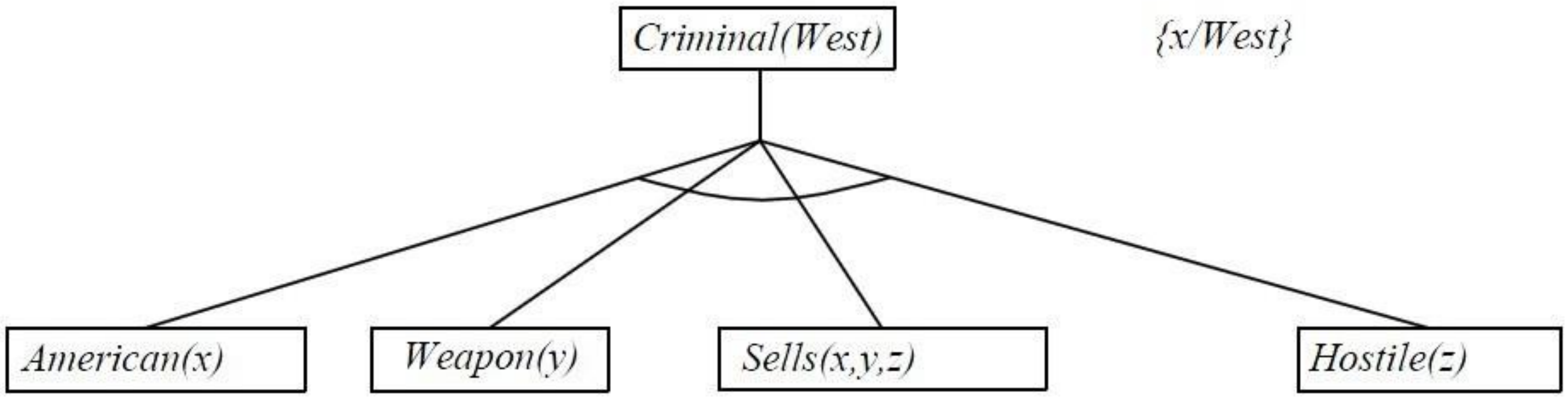
Forward chaining proof



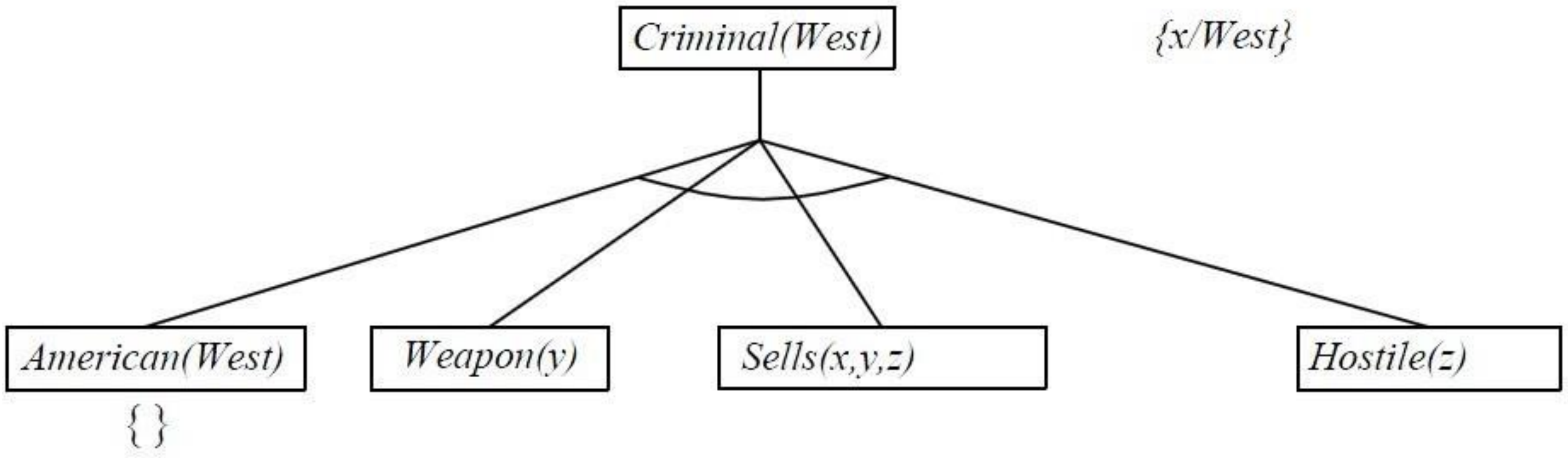
Backward chaining example

$\boxed{Criminal(West)}$

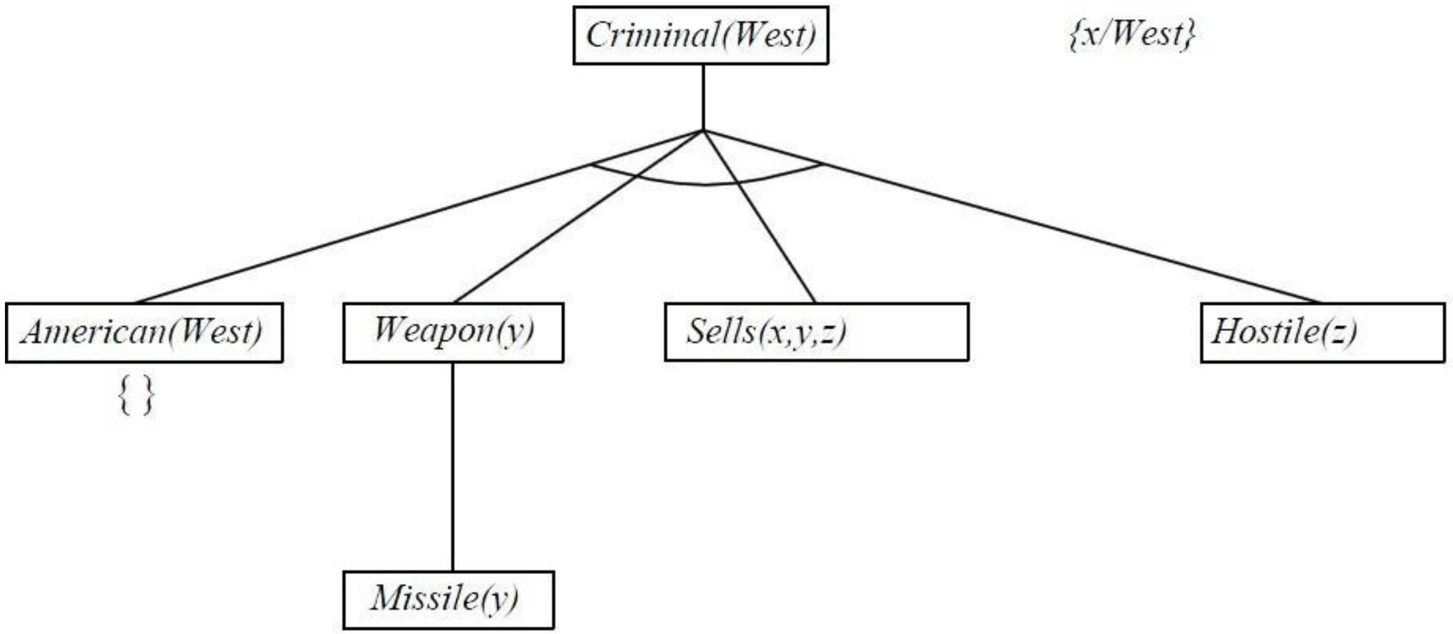
Backward chaining example



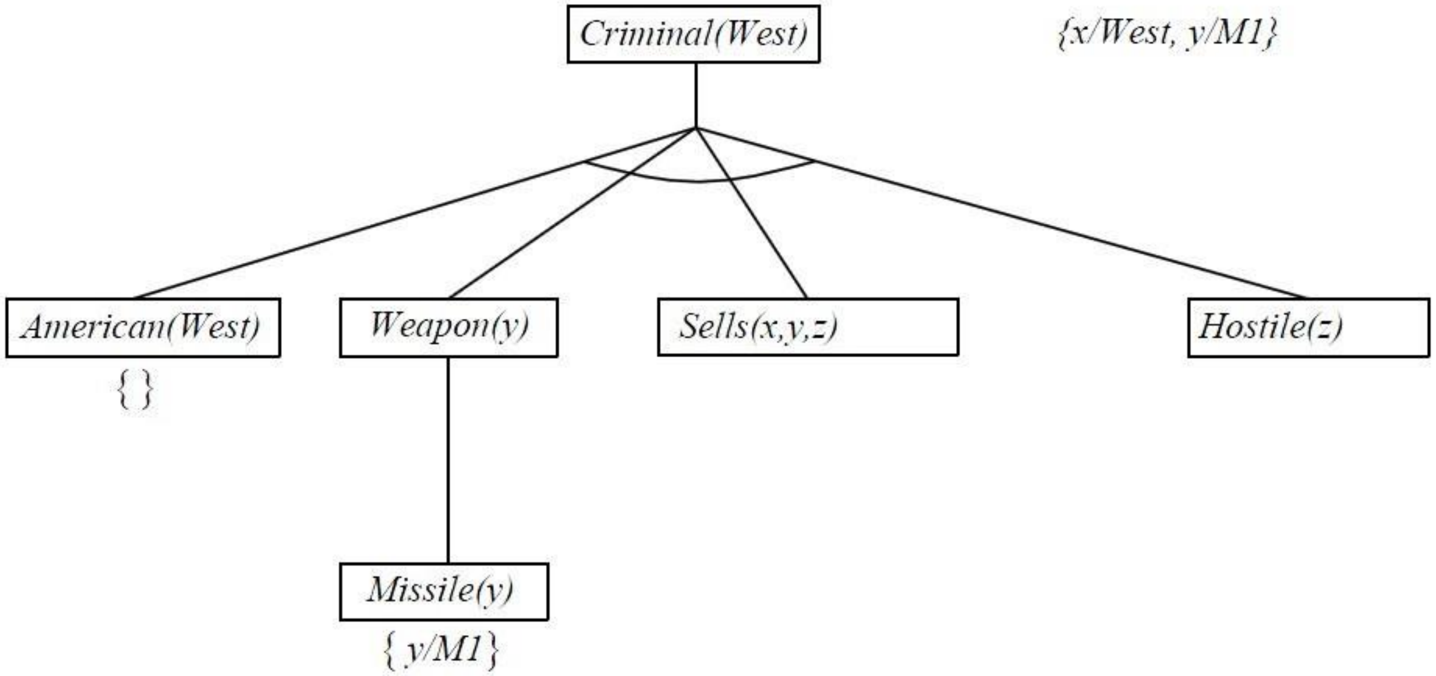
Backward chaining example



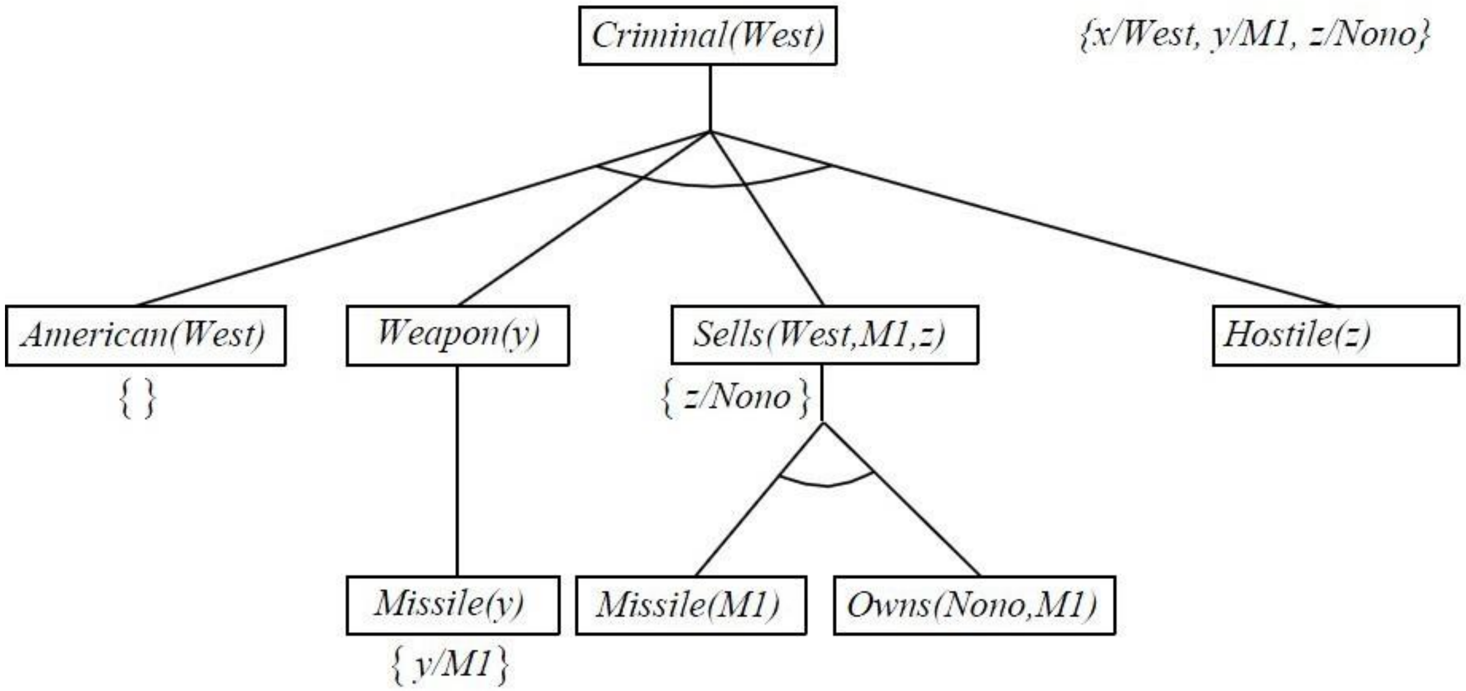
Backward chaining example



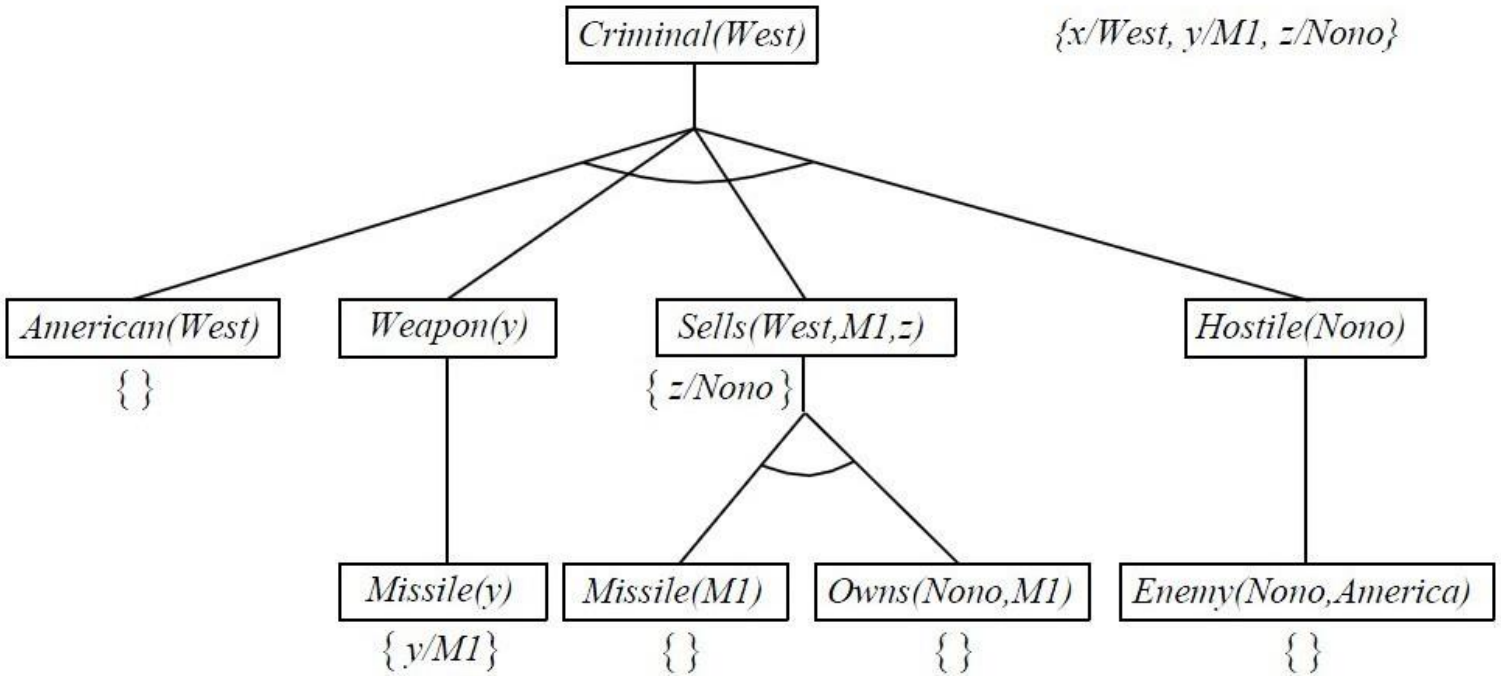
Backward chaining example



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

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: $\boxed{\neg \forall x, p \ \ \equiv \exists x \ \ \neg p, \quad \neg \exists x, p \ \ \equiv \forall x \ \ \neg p:}$

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

$X \Rightarrow Y \equiv \neg X \lor Y$
$X \Leftrightarrow Y \equiv X \Rightarrow Y \land Y \Rightarrow X$

a formula is in **conjunctive normal form (CNF)** or **clausal normal form** if it is a conjunction of one or more **clauses**, where a **clause** is a disjunction of literals

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:

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

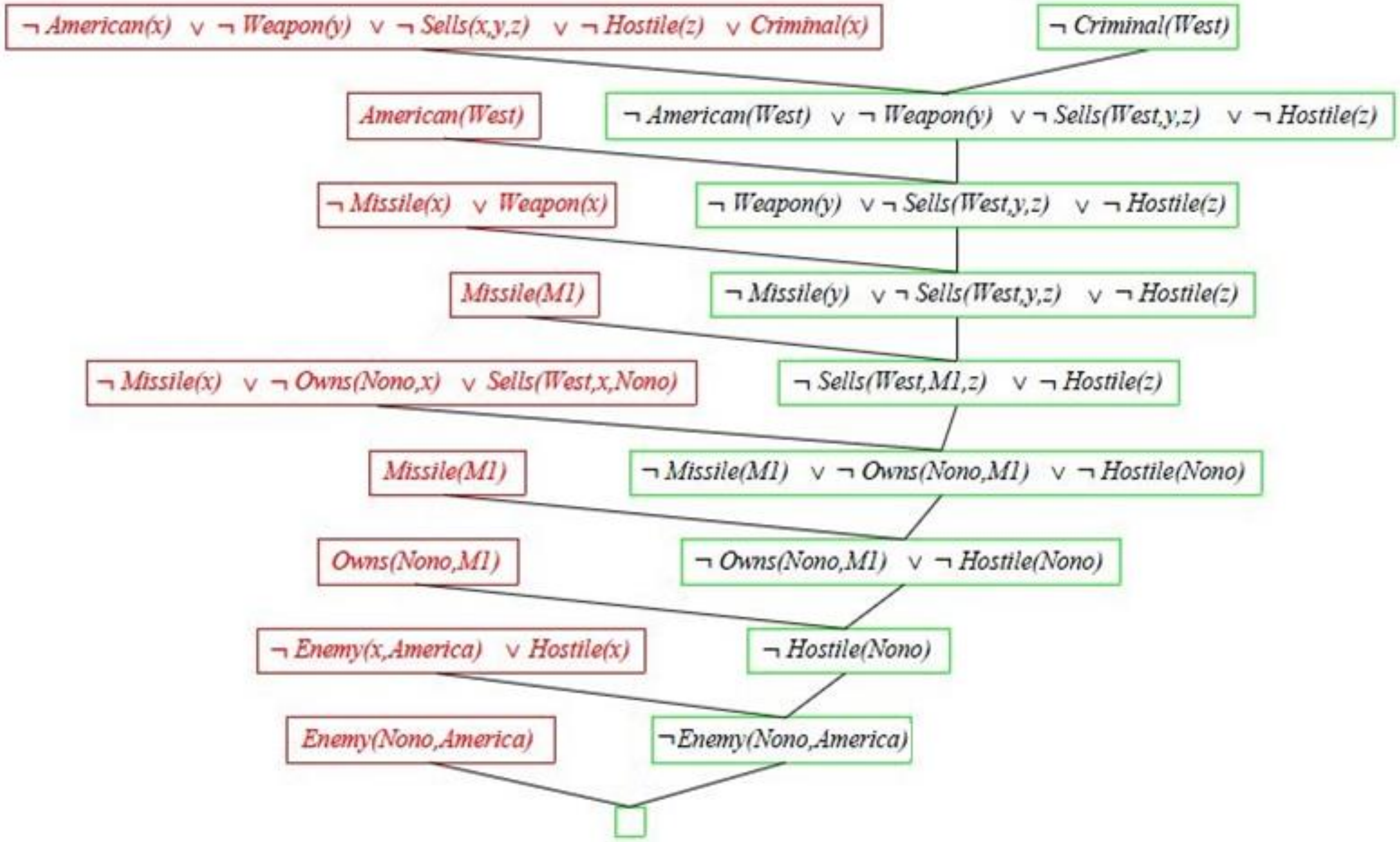
5. Drop universal quantifiers:

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

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



RESOLUTION REFUTATION PROOFS INVOLVES THE FOLLOWING STEPS:

1. Put the premises or axioms into *clause form.*
2. Add the negation of what is to be proved, in clause form, to the set of axioms.
3. *Resolve* these clauses together, producing new clauses that logically follow from them*.*
4. Produce a contradiction by generating the empty clause.
5. The substitutions used to produce the empty clause are those under which the opposite of the negated goal is true*.*

A FACTS IN PROPOSITIONAL LOGIC

<u>Given Axioms</u>	<u>Clause Form</u>	
P	P	(1)
$(P \wedge Q) \rightarrow R$	$\neg P \vee \neg Q \vee R$	(2)
$(S \vee T) \rightarrow Q$	$\neg S \vee Q$	(3)
	$\neg T \vee Q$	(4)
T	T	(5)

Prove R

RESOLUTION IN PROPOSITIONAL LOGIC

$\neg P \vee \neg Q \vee R$ (2) $\qquad \neg R$

$\neg P \vee \neg Q \qquad P$ (1)

$\neg T \vee Q$ (4) $\qquad \neg Q$

$\neg T \qquad T$ (5)

$\square$

EXAMPLE

Example: We wish to prove that "**Fido will die**" from the statements that "**Fido is a dog**" and "**all dogs are animals**" and "**all animals will die**." Changing these three premises to predicates gives:

Fido is a dog: dog (fido).

All dogs are animals: $\forall$(X) (dog (X) $\rightarrow$ animal (X)).

All animals will die: $\forall$(Y) (animal (Y) $\rightarrow$ die (Y)).

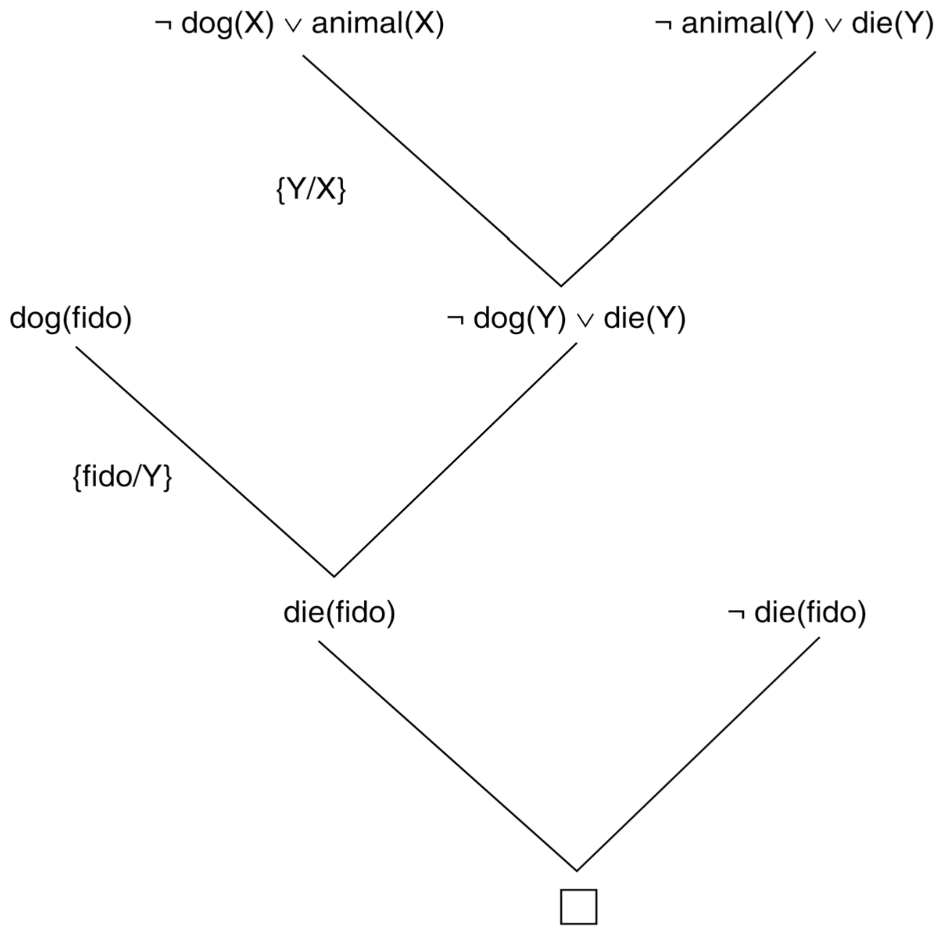
converts these predicates to clause form:

PREDICATE FORM	CLAUSE FORM
dog (fido)	dog (fido)
∀(X) (dog) (X) → animal (X))	¬ dog (X) ∨ animal (X)
∀(Y) (animal (Y) → die (Y))	¬ animal (Y) ∨ die (Y)

Negate the conclusion that Fido will die:

¬ die (fido)

RESOLUTION PROOF FOR THE “DEAD DOG” PROBLEM.



LUCKY STUDENT

1. Anyone passing his history exams and winning the lottery is happy.
2. Anyone who studies or is lucky can pass all his exams.
3. John did not study but he is lucky.
4. Anyone who is lucky wins the lottery.

Prove that John is happy!

Anyone passing his history exams and winning the lottery is happy.

$\forall$ **X (pass (X,history) $\wedge$ win (X,lottery) $\rightarrow$ happy (X))**

Anyone who studies or is lucky can pass all his exams.

$\forall$ **X** $\forall$ **Y (study (X) $\vee$ lucky (X) $\rightarrow$ pass (X,Y))**

John did not study but he is lucky.

$\neg$ **study (john) $\wedge$ lucky (john)**

Anyone who is lucky wins the lottery.

$\forall$ **X (lucky (X) $\rightarrow$ win (X,lottery))**

1. $\neg$ pass (X, history) $\vee$ $\neg$ win (X, lottery) $\vee$ happy (X)
2. $\neg$ study (Y) $\vee$ pass (Y, Z)
3. $\neg$ lucky (W) $\vee$ pass (W, V)
4. $\neg$ study (john)
5. lucky (john)
6. $\neg$ lucky (U) $\vee$ win (U, lottery)

Into these clauses is entered, in clause form, the negation of the conclusion:

7. $\neg$ happy (john)

Anyone passing his history exams and winning the lottery is happy.

$\forall$ **X (pass (X,history) $\wedge$ win (X,lottery) $\rightarrow$ happy (X))**

Anyone who studies or is lucky can pass all his exams.

$\forall$ **X** $\forall$ **Y (study (X) $\vee$ lucky (X) $\rightarrow$ pass (X,Y))**

John did not study but he is lucky.

$\neg$ **study (john) $\wedge$ lucky (john)**

Anyone who is lucky wins the lottery.

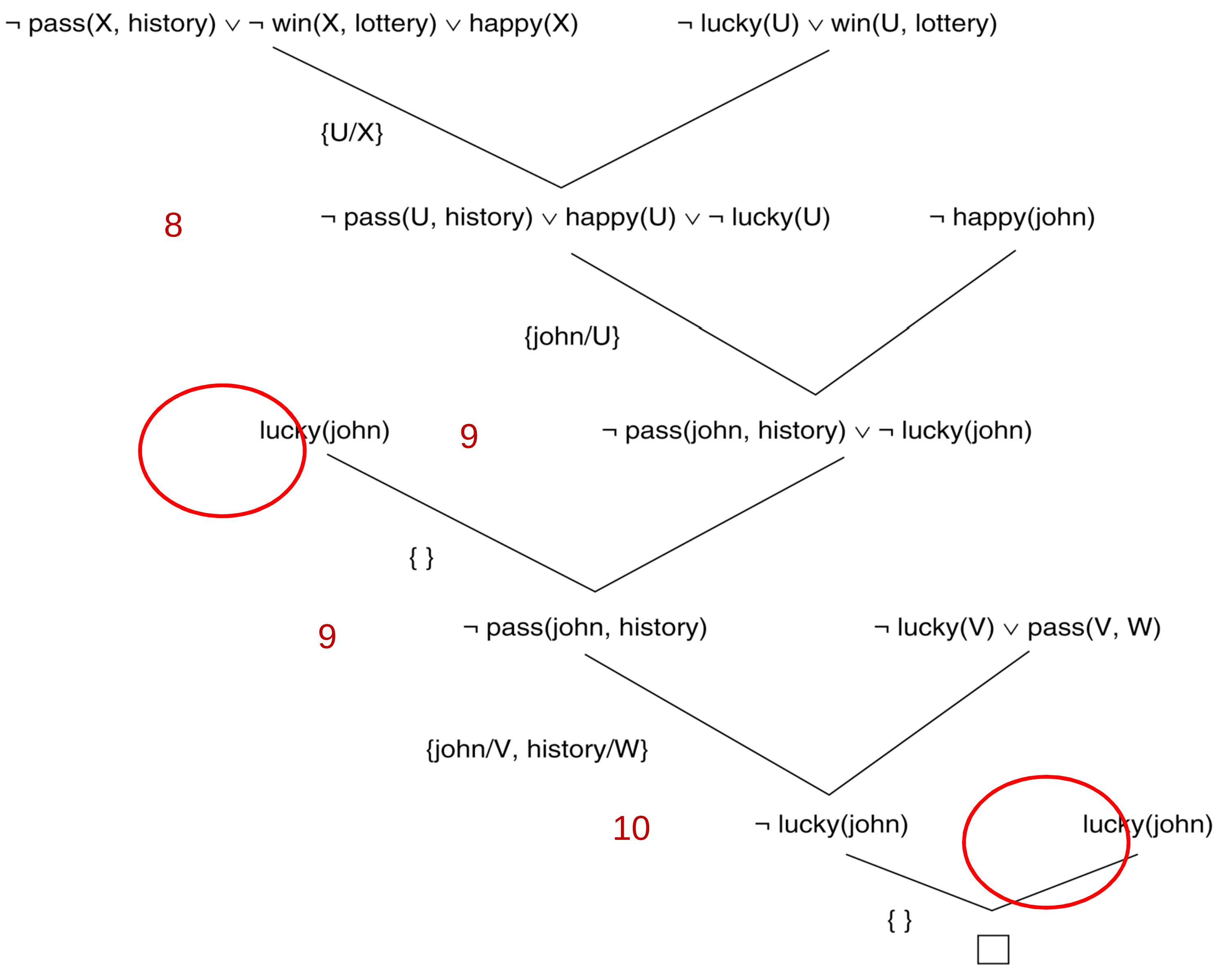
$\forall$ **X (lucky (X) $\rightarrow$ win (X,lottery))**

1. $\neg$ **pass (X, history) $\vee$ $\neg$ win (X, lottery) $\vee$ happy (X)**
2. $\neg$ **study (Y) $\vee$ pass (Y, Z)**
3. $\neg$ **lucky (W) $\vee$ pass (W, V)**
4. $\neg$ **study (john)**
5. $\boxed{\textbf{lucky (john)}}$
6. $\neg$ **lucky (U) $\vee$ win (U, lottery)**

Into these clauses is entered, in clause form, the negation of the conclusion:

7. $\boxed{\neg \textbf{ happy (john)}}$

ONE RESOLUTION REFUTATION FOR THE “HAPPY STUDENT” PROBLEM.



EXCITING LIFE

1. All people that are not poor and are smart are happy.
2. Those people that read are not stupid.
3. John can read and is wealthy.
4. Happy people have exiting lives.

Can anyone be found with an exciting life?

All people who are not poor and are smart are happy. Those people who read are not stupid. John can read and is wealthy. Happy people have exciting lives. Can anyone be found with an exciting life?

We assume $\boxed{\forall X\ (\text{smart}(X) \equiv \neg\ \text{stupid}(X)) \text{ and } \forall Y\ (\text{wealthy}(Y) \equiv \neg\ \text{poor}(Y))}$, **and get**:

$\forall X\ (\neg\ \text{poor}(X) \wedge \text{smart}(X) \rightarrow \text{happy}(X))$

$\forall Y\ (\text{read}(Y) \rightarrow \text{smart}(Y))$

$\text{read}(\text{john}) \wedge \neg\ \text{poor}(\text{john})$

$\forall Z\ (\text{happy}(Z) \rightarrow \text{exciting}(Z))$

The negation of the conclusion is:

$\boxed{\neg\ \exists\ W\ (\text{exciting}(W))}$

These predicate calculus expressions for the "exciting life" problem are transformed into the following clauses:

$\text{poor}(X) \vee \neg\ \text{smart}(X) \vee \text{happy}(X)$ 1

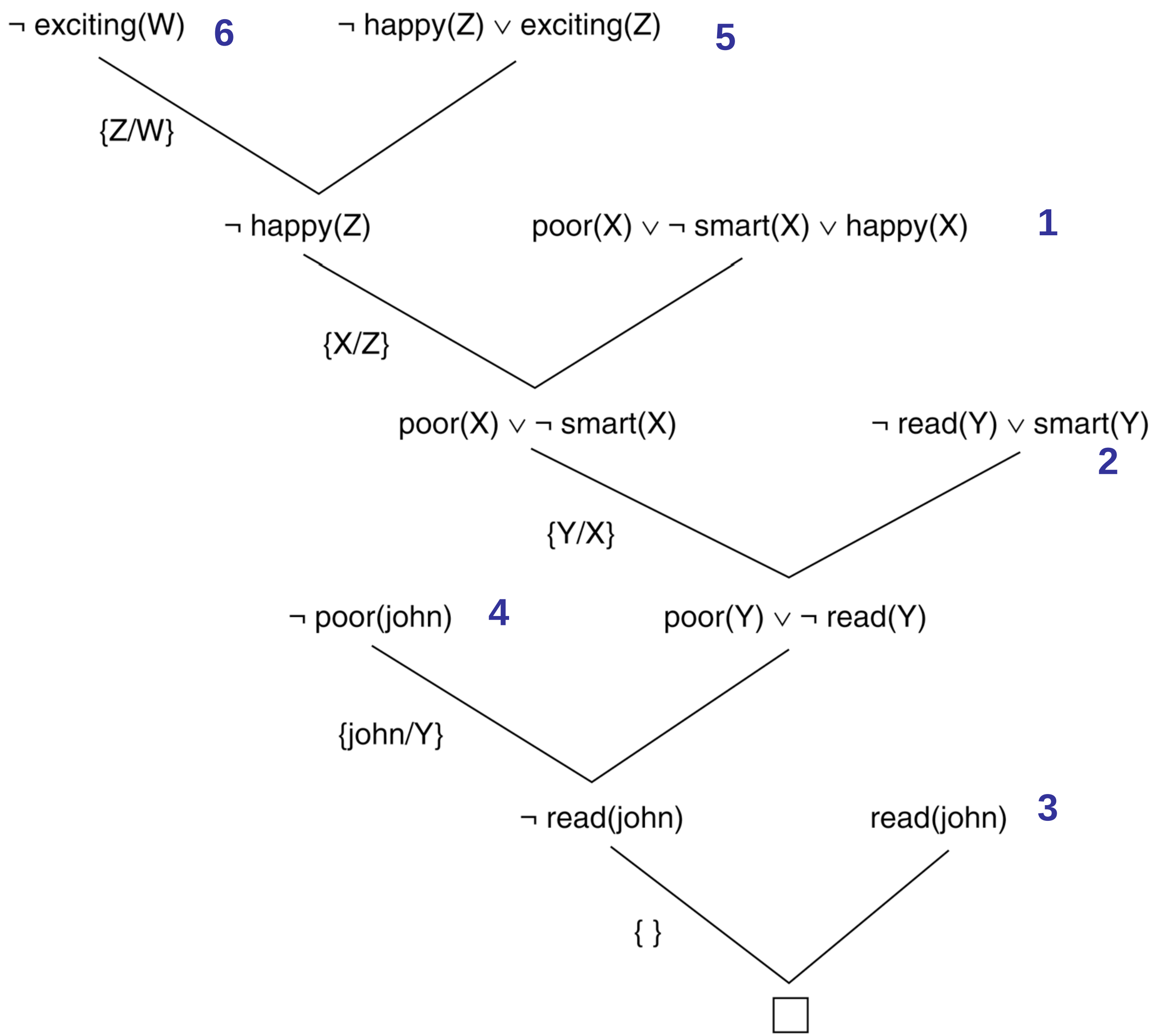
$\neg\ \text{read}(Y) \vee \text{smart}(Y)$ 2

$\text{read}(\text{john})$ 3

$\neg\ \text{poor}(\text{john})$ 4

$\neg\ \text{happy}(Z) \vee \text{exciting}(Z)$ 5

$\neg\ \text{exciting}(W)$ 6



RESOLUTION PROOF FOR THE “EXCITING LIFE” PROBLEM.

RESOLUTION EXAMPLE (ENGLISH → FOL)

1. **John likes all kinds of food.**

   $\forall x: food(x) \rightarrow likes(john, x)$

2. **Apples are food.**

   $food(apple)$

3. **Chicken is food.**

   $food(chicken)$

4. **Anything anyone eats and isn't killed by is food.**

   $\forall x: (\exists y: eats(y, x) \wedge \neg killedby(y, x)) \rightarrow food(x)$

5. **Bill eats peanuts and is still alive.**

   A. $eats(Bill, peanuts)$ B. $alive(Bill)$

6. **Sue eats everything Bill eats.**

   $\forall x: eats(Bill, x) \rightarrow eats(Sue, x)$

7. $\forall x: \forall y: alive(x) \rightarrow \neg killedby(x,y)$

CLAUSE FORM:

1. $\neg$ *food(x1)* $\vee$ *likes(John, x1)*
2. *food(apples)*
3. *food(chicken)*
4. $\neg$ *eats(y4,x4)* $\vee$ *killedby(y4 , x4)* $\vee$ food(x4)
5. *Eats (Bill, peanuts)*
6. *Alive (Bill)*
7. $\neg$ *eats(Bill,x7)* $\vee$ *eats(Sue,x7)*
8. $\neg$ *alive(x8)* $\vee$ $\neg$ *killedby(x8, y8)*

¬ likes (John, peanuts)　　¬ *food(x1)* ∨ *likes (John, x1)*

peanuts/x1

¬ *food(peanuts)*　　¬ *eats(y4,x4)* ∨ *killedby(y4* . x4) ∨ food(x4)

peanuts/x4

¬ *eats(y4,peanuts)* ∨ *killedby(y4*. peanuts)　　*eats(Bill.peanuts)*

Bill/y4

killedby(Bill. peanuts)　　¬ *alive(x8)* ∨ ¬ *killedby(x8 y8)*

Bill/x8, peanuts/y8

¬ *alive(Bill)*　　*alive(Bill)*

□

RESOLUTION PROOF THAT JOHN LIKES PEANUTS

T

h

a

n

k

Y

o

u