Artificial Intelligence & Soft Computing

CSC 703

Subject In-charge

Safa Hamdare

Assistant Professor

Room No. 407

email: safahamdare@sfit.ac.in

Chapter 3

knowledge ,Reasoning & Planning

Based on CO3:

Analyze the strength and weakness of AI approaches to knowledge representation, reasoning and planning.

Outline of Knowledge & Reasoning

- **Knowledge-based agents**
  - WUMPUS World Environment
- **Propositional logic**
- **First Order Predicate Logic-** Syntax & Semantics
  - Knowledge Engineering in FOL
- **Converting to CNF**
- **Inference in FOL**
  - Resolution
  - Unification
  - Forward Chaining
  - Backward Chaining

Outline of Planning

- **Planning Agent**
- **Types of Planning**
  - Partial Order
  - Hierarchical Order
  - Conditional Order

Knowledge Based Agent

knowledge-based agent

- A knowledge-based agent includes a knowledge base and an inference system.
- A knowledge base is a set of representations of facts of the world.
- Each individual representation is called a **sentence**.
- The sentences are expressed in a **knowledge representation language**.
- **The agent operates as follows:**

  1. It TELLs the knowledge base what it perceives.
  2. It ASKs the knowledge base what action it should perform.
  3. It TELLs the knowledge base what action its taking.
  4. It performs the chosen action.

Knowledge-based agent

- Split knowledge from algorithm

  Inference engine (IE) ← Domain-independent algorithm

  $\updownarrow$

  Knowledge base (*KB*) ← Domain-specific knowledge

- Declarative approach to building an agent
  - *Provide* it with initial *KB* in *formal* language
  - Then it can *ask itself* what to do (inference algorithm)

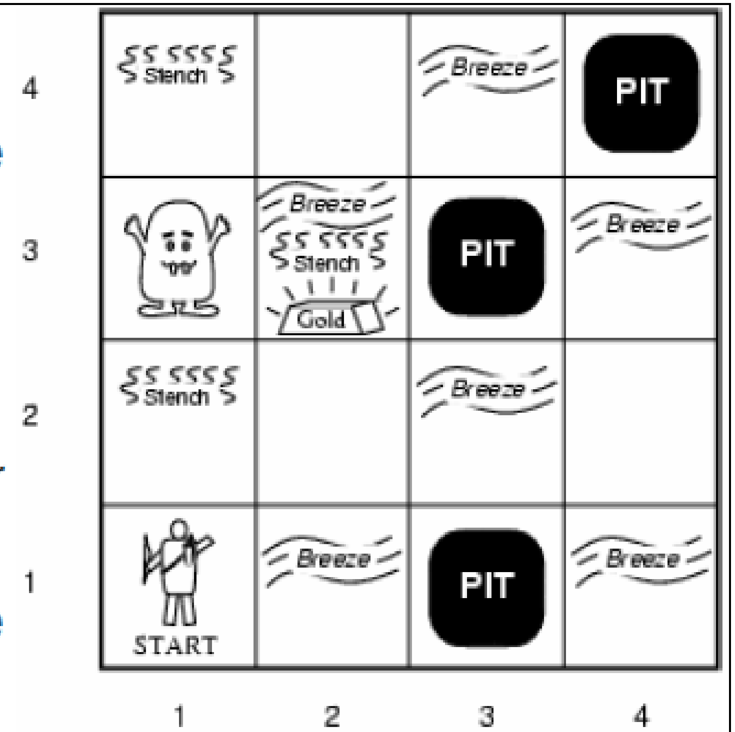
```
function action ← KB-AGENT (percept)
   static KB % Knowledge base
          t ← 0 % time/counter
   KB ← TELL (KB, MAKE-PERCEPT-SENTENCE (percept, t))
   action ← ASK (KB, MAKE-ACTION-QUERY (t))
   KB ← TELL (KB, MAKE-PERCEPT-SENTENCE (action, t))
   t ← t + 1
   return action
```

Architecture of knowledge-based agent

- **Knowledge Level.**
  - The most abstract level: describe agent by saying what it knows.
  - **Example:** A taxi agent might know that the Western express highway connects Borivali with the Bandra.
- **Logical Level.**
  - The level at which the knowledge is encoded into sentences.
  - **Example:** Links(Western express highway, Borivali, Bandra).
- **Implementation Level.**
  - The physical representation of the sentences in the logical level.
  - **Example:** ‘(links Western express highway Borivali Bandra)

Wumpus world example

- Environment:
  - Squares adjacent to wumpus are smelly
  - Squares adjacent to pit are breezy
  - Glitter iff gold is in the same square
  - Shooting kills the wumpus if player faces it
  - Grabbing picks up the gold if in the same square
  - Releasing drops the gold in the same square



- Actuators: left turn, right turn, forward, grab, release, shoot
- Sensors: breeze, glitter, smell
- Performance measure:

  gold: $+1000$, death: -1000, step: -1 , shoot: -10

Give PEAS description for – WUMPUS WORLD (10) June 2013

Performance measure:

- +1000 points for picking up the *gold* — this is the goal of the agent
- −1000 points for dying = entering a square containing a *pit* or a live *Wumpus* monster
- −1 point for each action taken, and
- −10 points for using the *arrow* trying to kill the Wumpus — so that the agent should avoid performing unnecessary actions.

Environment: A 4×4 grid of squares with...

- the agent starting from square $[1, 1]$ facing right
- the gold in one square
- the initially live Wumpus in one square, from which it never moves
- maybe pits in some squares.

The starting square $[1, 1]$ has no Wumpus, no pit, and no gold — so the agent neither dies nor succeeds straight away.

Actuators: The agent can...

turn 90° left or right

walk one square forward in the current direction

grab an object in this square

shoot the single arrow in the current direction, which flies in a straight line until it hits a wall or the Wumpus.

Sensors: The agent has 5 TRUE/FALSE sensors which report a...

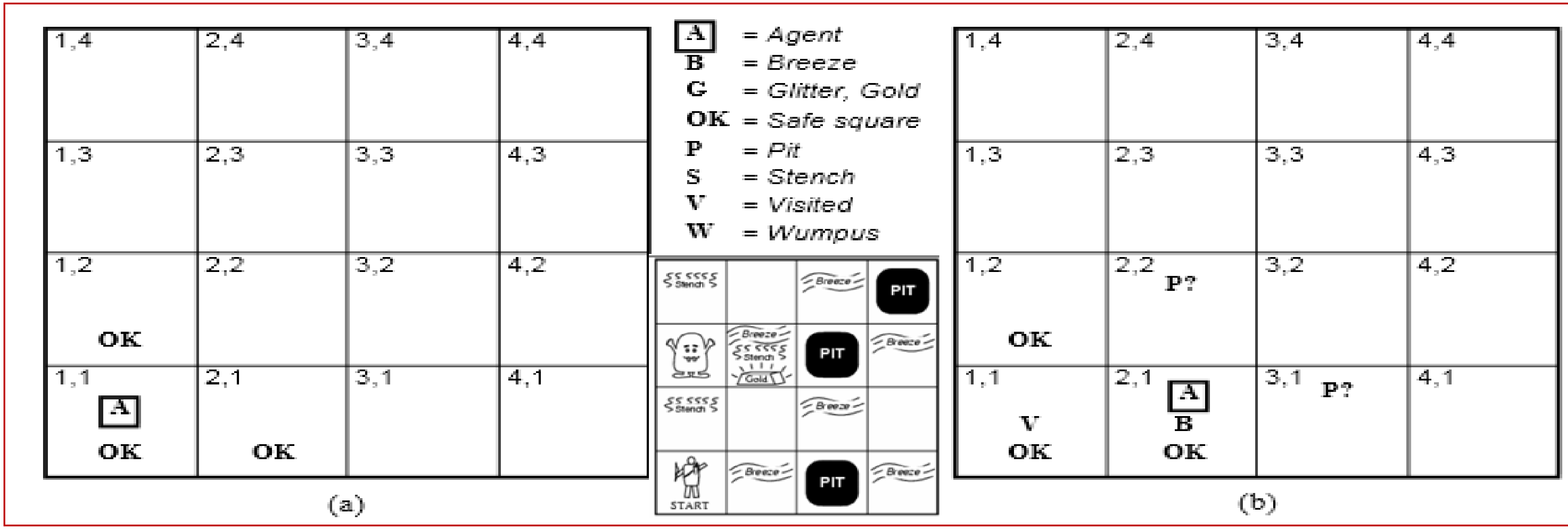
stench when the Wumpus is in an adjacent square — directly, not diagonally

breeze when an adjacent square has a pit

glitter when the agent perceives the glitter of the gold in the current square

bump when the agent walks into an enclosing wall (and then the action had no effect)

scream when the arrow hits the Wumpus, killing it.



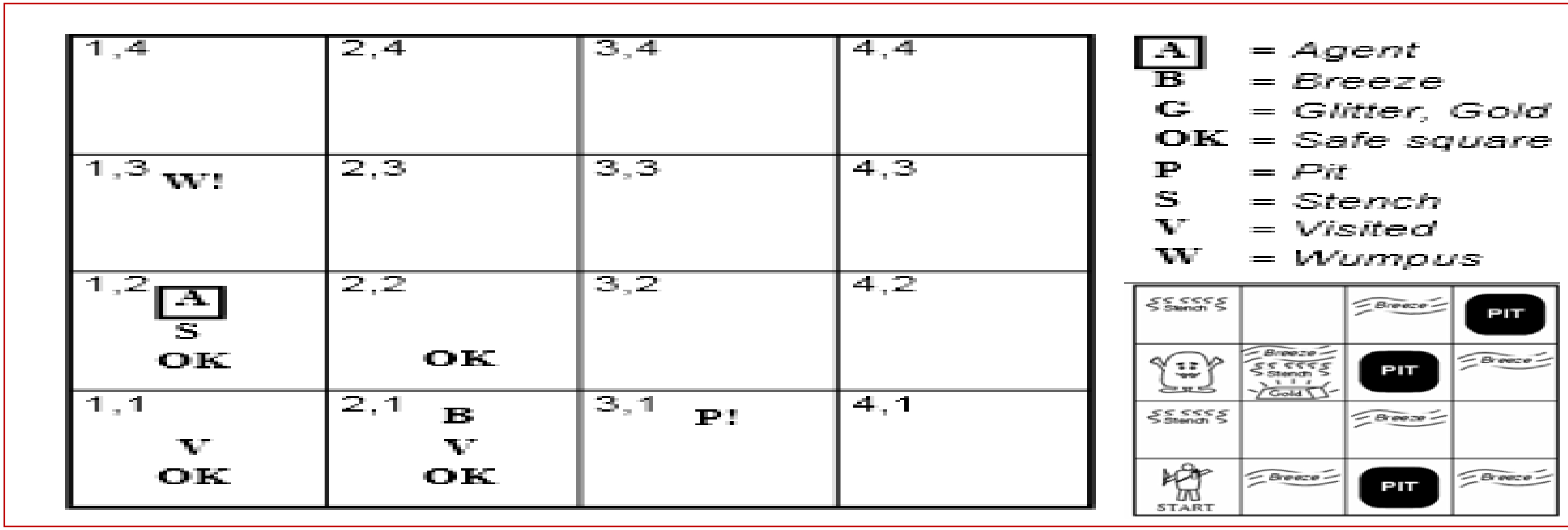
Top right: Agent **A** is cautious, and will only move to **OK** squares.

– Agent **A** walks into [2, 1], because it is **OK**, and in the direction where agent **A** is facing, so it is cheaper than the other choice [1, 2].
Agent **A** also marks [1, 1] **V**isited.

– Agent **A** perceives a **B**reeze but nothing else.

– Agent **A** infers: "At least one of the adjacent squares [1, 1], [2, 2] and [3, 1] must contain a **P**it. There is no **P**it in [1, 1] by my background knowledge β. Hence [2, 2] or [3, 1] or both must contain a **P**it."

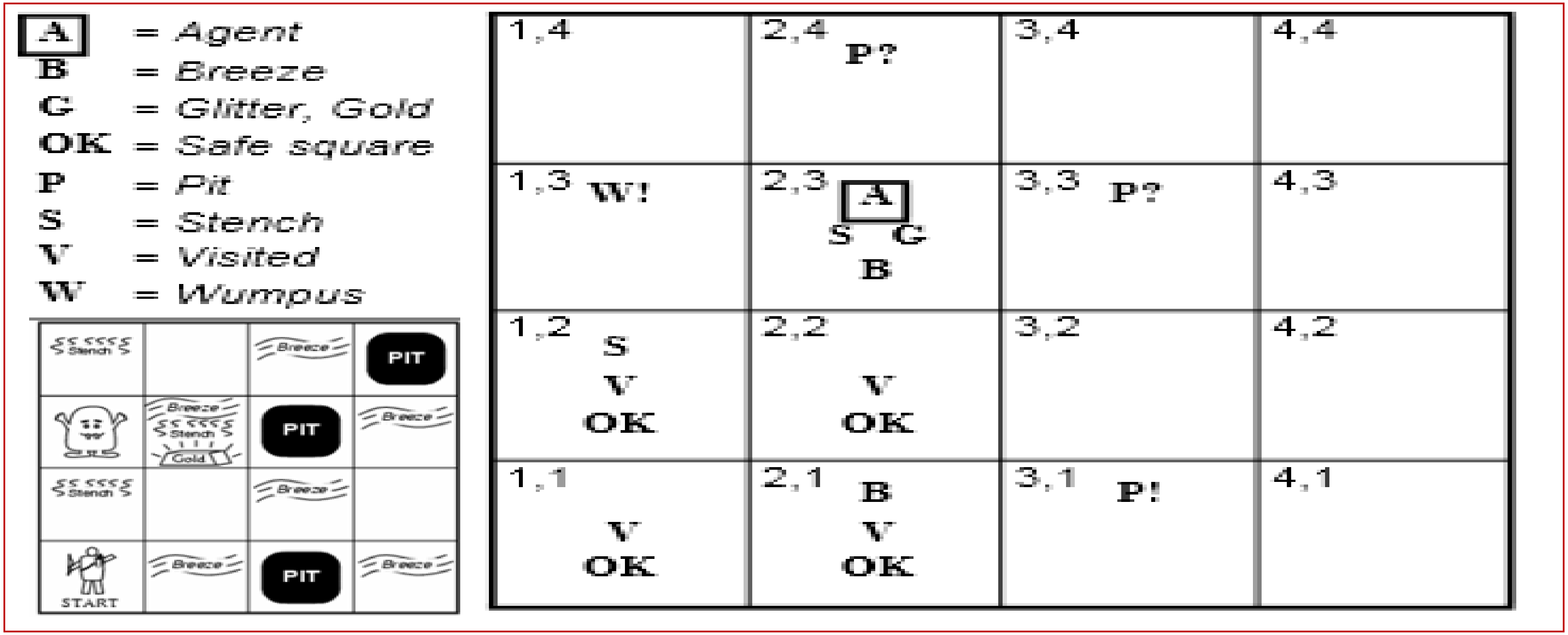
– Hence agent **A** cannot be certain of either [2, 2] or [3, 1], so [2, 1] is a dead end for a cautious agent like **A**.



Bottom left: Agent **A** has turned back from the dead end $[2, 1]$ and walked to examine the other **OK** choice $[1, 2]$ instead.

- Agent **A** preceives a **Stench** but nothing else.
- Agent **A** infers using also *earlier percepts:* "The **Wumpus** is in an adjacent square. It is not in $[1, 1]$. It is not in $[2, 2]$ either, because then I would have sensed a **Stench** in $[2, 1]$. Hence it is in $[1, 3]$."
- Agent **A** infers using also *earlier inferences:* "There is no **Breeze** here, so there is no **Pit** in any adjacent square. In particular, there is no **Pit** in $[2, 2]$ after all. Hence there is a **Pit** in $[3, 1]$."
- Agent **A** finally infers: "$[2, 2]$ is **OK** after all — now it is certain that it has neither a **Pit** nor the **Wumpus**."

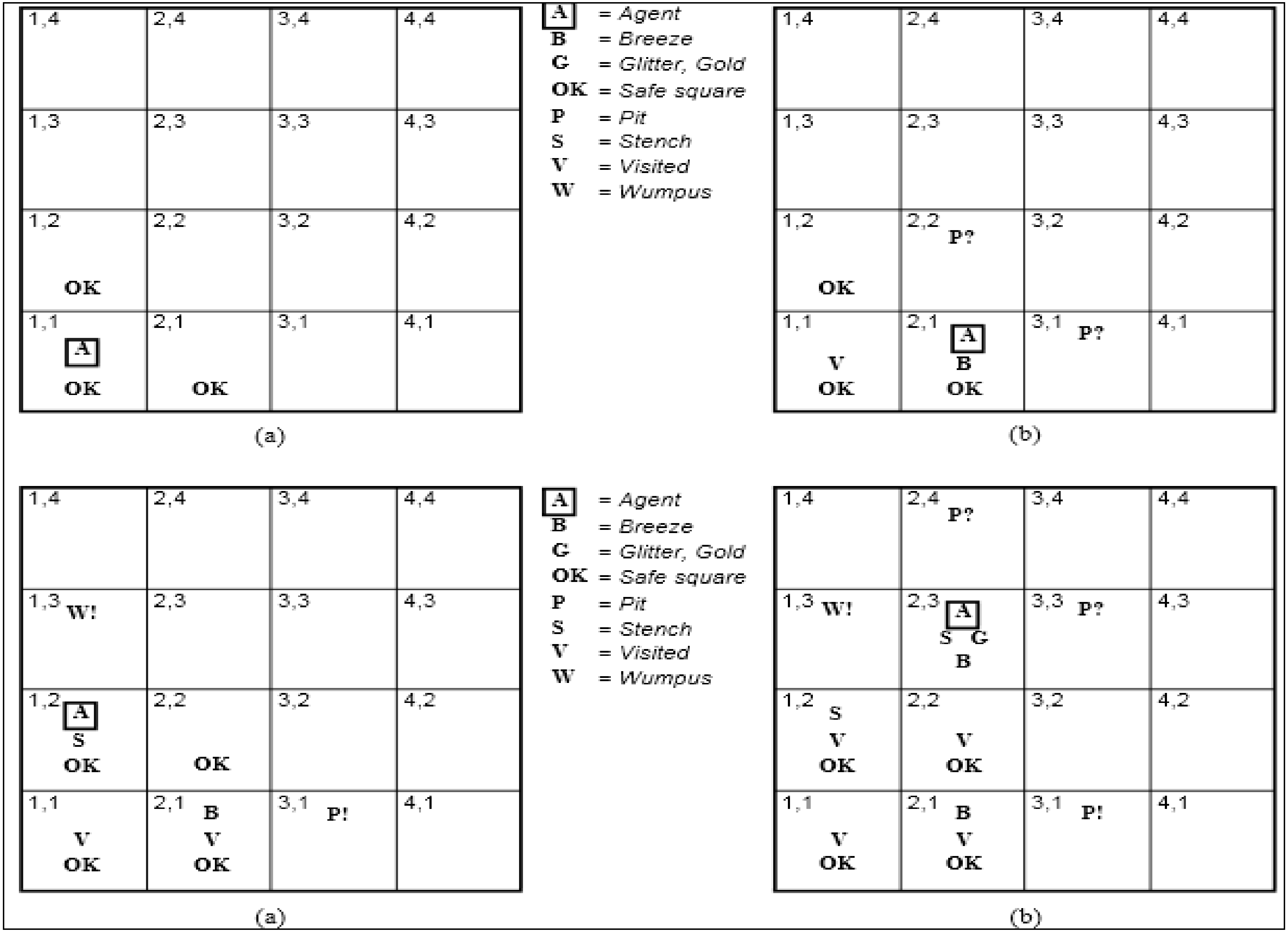
This reasoning is too complicated for many animals — but not for the logical agent **A**.



Bottom right:

1. Agent A walks to the only unvisited OK choice [2, 2]. There is no Breeze here, and since the the square of the Wumpus is now known too, [2, 3] and [3, 2] are OK too.
2. Agent A walks into [2, 3] and senses the Glitter there, so he grabs the gold and succeeds.

Complete Solution For WUMPUS World



(a) (b)

(a) (b)

Quiz

1. **Knowledge and reasoning also play a crucial role in dealing with ______________________ environment.**

   a) Completely Observable
   b) Partially Observable
   c) Neither Completely nor Partially Observable
   d) Only Completely and Partially Observable

2. **Using how many levels can a knowledge-based agent be defined?**

   a) 3 levels
   b) 2 levels
   c) 4 levels
   d) None of the above

3. **A knowledge-based agent can combine general knowledge with current percepts to infer hidden aspects of the current state prior to selecting actions.**

   a) True or
   b) False

Quiz

4. **A) Knowledge base (KB) is consists of set of statements.**
   B) Inference is deriving a new sentence from the KB.
   Choose the correct option.
   a) A is true, B is true
   b) A is false, B is false
   c) A is true, B is false
   d) A is false, B is true
5. **Wumpus World is a classic problem, best example of ________**
   a) Single player Game
   b) Two player Game
   c) Reasoning with Knowledge
   d) Knowledge based Game

Representation, reasoning, and logic

- The object of knowledge representation is to express knowledge in a **computer-tractable** form, so that agents can perform well.
- A knowledge representation language is defined by:
  - its **syntax,** which defines all possible sequences of symbols that constitute sentences of the language.
    - Examples: Sentences in a book, bit patterns in computer memory.
  - its **semantics**, which determines the facts in the world to which the sentences refer.
    - Each sentence makes a claim about the world.
    - An agent is said to believe a sentence about the world.

Propositional logic

Syntax and Semantics

- **Syntax**
  - Rules for constructing legal sentences in the logic
  - Which symbols we can use (English: letters, punctuation)
  - How we are allowed to combine symbols
  - **Example:** "Cat Sat on the Mat" (**Valid**)

    Sat the Cat on Mat (**Invalid**)"
- **Semantics (Compute the truth)**
  - How we interpret (read) sentences in the logic
  - Assigns a meaning to each sentence
  - And we can understand the meaning (semantics)

Propositional Logic

- **Syntax**
  - Propositions, e.g. "it is wet"
  - Connectives: and, or, not, implies, iff (equivalent)

    $\wedge \quad \vee \quad \neg \quad \longrightarrow \quad \longleftrightarrow$
  - Brackets, T (true) and F (false)
- **Semantics** (Classical AKA Boolean)
  - Define how connectives affect truth
    - "P and Q" is true if and only if P is true and Q is true
  - Use **truth tables** to work out the truth of statements

Standard Translations

1. ~A: not A, A is false, A is not true, it's not the case that A
2. (A & B): A and B, A but B, although A, B
3. (~A & ~B): neither A nor B
4. (A v B): A or B, A unless B
5. (A → B): if A then B, A only if B, B if A
6. (A ↔ B): A if and only if B

The most difficult part of translations is "if" and "only if" and "if and only if"

- "If A then B" "A only if B" "B if A": (A → B)
- "If B then A" "B only if A" "A if B": (B → A)
- "A if and only if B" "B if and only if A": (A ↔ B)

Exercise to convert sentence into Proposition logic

Consider the following three sentences.

1. Sam will be sad unless we come to his party.

2. We will come to Sam's party if and only if there is food.

3. Sam won't have food at the party.

- **Translate:**

S = Sam is sad

C = We come to Sam's party

F = There is food at the party

- **Translate into Proposition Logic:**

1. $(S \lor C)$
2. $(C \leftrightarrow F)$
3. $\sim F$

Exercise

1. John went to school and Marry went to school

   P= John went to school

   Q=Marry went to school

   P $\wedge$ Q

2. Sally will go to work or sally will take off.

   P= Sally will go to work

   Q= Sally will take off

   P v Q

Exercise

3. It is not the case that I like peanut butter and jelly

   P= I like peanut butter

   Q= I like Jelly

   $\sim(P \wedge Q)$

4. I don't like peanut butter and I don't like jelly

   P= I like peanut butter

   Q= I like Jelly

   $\sim P \wedge \sim Q$

Exercise

5. It is not the case that tom and John will work late or that Henry will call in sick.

   P=Tom will work late

   Q=John will work late

   R=Henry will call in sick

   $\sim((P \wedge Q) \vee R)$

6. Getting an A on the final exam is a necessary condition for getting an A in the class

   P= I Get an A on the final exam

   Q= I get an A in the class

   $Q \Rightarrow P$

Exercise

7. I will pass the muffins if you will ask me nicely

   M= I will pass the muffins

   N= You ask me nicely

   $N \Rightarrow M$

8. Jill needs a parachute if and only if she is planning to jump from the plane

   P = Jill needs a parachute

   Q= Jill is planning to jump from the plane

   $P \leftrightarrow Q$

First Order Predicate Logic

Predicate Logic

- **Propositional logic combines atoms**
  - An atom contains no propositional connectives
  - Have no structure

  Examples: (today_is_wet, john_likes_apples)
- **Predicates allow us to talk about objects**
  - Properties: is_wet(today)
  - Relations: likes(john, apples)
- In predicate logic each atom is a predicate
  - E.g. First order logic

First Order Logic

- Also Known as First-Order Predicate Logic
- Also Known as First-Order Predicate Calculus
- More expressive logic than propositional
    - We no longer need a separate rule for each square to say which other squares are breezy/pits
- **Syntax for First Order Logic:**
    1. **Constants** are objects: john, apples
    2. **Predicates** are properties and relations:
        - fruit of(apple_tree), likes(john, apples)
    3. **Functions** transform objects:
        - likes(john, fruit of(apple_tree))
    4. **Variables** represent any object: likes(X, apples)
    5. **Quantifiers** qualify values of variables
        - True for all objects (Universal): $\forall$X. likes(X, apples)
        - Exists at least one object (Existential): $\exists$X. likes(X, apples)

Syntax of First-Order Logic

- Constants King John, 2, …
- Predicates Brother, >, …
- Functions Sqrt, LeftArmOf, …
- Variables x, y, a, b, …
- Connectives $\wedge$ $\vee$ $\neg$ $\Rightarrow$ $\Leftrightarrow$
- Equality =
- Quantifiers $\exists$ $\forall$

Syntax of First-Order Logic: Atomic Sentence

$$\underbrace{\underbrace{Brother}_{\text{predicate}}\,(\,\underbrace{\underbrace{KingJohn}_{\text{constant}}}_{\text{term}},\ \underbrace{\underbrace{RichardTheLionheart}_{\text{constant}}}_{\text{term}}\,)}_{\text{atomic sentence}}$$

$$\underbrace{\underbrace{>}_{\text{predicate}}\ (\underbrace{\underbrace{Length}_{\text{function}}(\underbrace{LeftLegOf}_{\text{function}}(\underbrace{Richard}_{\text{constant}}))}_{\text{term}},\ \underbrace{\underbrace{Length}_{\text{function}}(\underbrace{LeftLegOf}_{\text{function}}(\underbrace{KingJohn}_{\text{constant}}))}_{\text{term}}\ )}_{\text{atomic sentence}}$$

$$\underbrace{\underbrace{\underbrace{Sibling}_{\text{predicate}}(\,\underbrace{KingJohn}_{\text{term}}, \underbrace{Richard}_{\text{term}}\,)}_{\text{atomic sentence}} \ \Rightarrow\ \underbrace{\underbrace{Sibling}_{\text{predicate}}(\,\underbrace{Richard}_{\text{term}}, \underbrace{KingJohn}_{\text{term}}\,)}_{\text{atomic sentence}}}_{\text{complex sentence}}$$

Logical connectives are used to construct complex sentence

Comparison between PL & FOPL

Proposition Logic	First order Predicate logic
It uses propositions in which complete statement is represented using symbols	FOPL uses predicates which involve constants, variables, relation, functions.
PL cannot represent individual property. Eg. Meera is tall	FOPL can represent individual property. E.g. tall(Meera)

Quiz:

1. **In AI systems, Knowledge can be represented in two ways. What are these two ways?**
   a) Machine Logic
   b) Predicate Logic
   c) Propositional Logic
   d) Compound Logic
2. **How many proposition symbols are there in artificial intelligence?**
   a) 1
   b) 2
   c) 3
   d) 4
3. **"In the propositional logic system of knowledge representation, it is assumed that the word contains object, relations, and functions. The Predicate logic is a symbolized reasoning in which we can divide the sentence into a well-defined subject and predicate."**
   – By reading the above statement, State whether it is true or false?

Quiz:

4. Which is created by using single propositional symbol?

a) Complex sentences
b) Atomic sentences
c) Composition sentences
d) None of the mentioned

5. Which is used to construct the complex sentences?

a) Symbols
b) Connectives
c) Logical connectives
d) All of the mentioned

6. **How many logical connectives are there in artificial intelligence?**

a) 2
b) 3
c) 4
d) 5

Note to remember about FOPL:

- ($\Rightarrow$ Is the main connective with $\forall$ not $\wedge$)

Example

Correct: $\forall x\,(StudiesAt(x, Koblenz) \Rightarrow Smart(x))$

"Everyone who studies at Koblenz is smart"

Wrong: $\forall x\,(StudiesAt(x, Koblenz) \wedge Smart(x))$

"Everyone studies at Koblenz and is smart", i.e.,

"Everyone studies at Koblenz and everyone is smart"

- ($\wedge$ Is the main connective with $\exists$ not $\Rightarrow$)

Correct: $\exists x\,(StudiesAt(x, Landau) \wedge Smart(x))$

"There is someone who studies at Landau and is smart"

Wrong: $\exists x\,(StudiesAt(x, Landau) \Rightarrow Smart(x))$

"There is someone who, if he/she studies at Landau, is smart"

This is true if there is anyone not studying at Landau

Examples

1. **Everyone likes McDonalds**
   – $\forall x,\ likes(x, McDonalds)$
2. **Someone likes McDonalds**
   – $\exists x,\ likes(x, McDonalds)$
3. **All children like McDonalds**
   – $\forall x,\ child(x) \Rightarrow likes(x, McDonalds)$
4. **Everyone likes McDonalds unless they are allergic to it**
   – $\forall x, likes(x, McDonalds) \vee allergic(x, McDonalds)$
   – $\forall x, \neg allergic(x, McDonalds) \Rightarrow likes(x, McDonalds)$

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”

Nesting Quantifiers

- Everyone likes some kind of food

  $\forall y\ \exists x,\ \text{food}(x) \wedge \text{likes}(y, x)$
- There is a kind of food that everyone likes

  $\exists x\ \forall y,\ \text{food}(x) \wedge \text{likes}(y, x)$
- Someone likes all kinds of food

  $\exists y\ \forall x,\ \text{food}(x) \Rightarrow \text{likes}(y, x)$
- Every food has someone who likes it

  $\forall x\ \exists y,\ \text{food}(x) \Rightarrow \text{likes}(y, x)$

Exercise: Represent statements into FOPL:

- Every house is a physical Object
  - $\forall x$, House(x) $\Rightarrow$ physical_Object(x)
- Some physical objects are houses
  - $\exists x$, physical_Object(x) $\wedge$ House(x)
- Every house has an owner /Every house is owned by somebody
  - $\forall x$ (house(x) $\Rightarrow$ $\exists y$ owns(y, x))
- Everybody owns a house
  - $\forall x$ (house(y) $\wedge$ $\exists y$ owns(x,y))
- Sue owns a house
  - $\exists y$ house(y) $\wedge$ owns(Sue,y)
- Peter does not owns a house
  - $\exists y$ house(y) $\wedge$ $\neg$owns(Peter,y)
- Somebody does not own a house
  - $\exists$ x,$\exists y$ ($\neg$ owns(x,y) $\wedge$ house(y))

- **May 2016 (10 Marks)**

(b) Write first order logic statements for following statements:

(i) If a perfect square is divisible by a prime p then it is also divisible by square of p.

(ii) Every perfect square is divisible by some prime.

(iii) Alice does not like Chemistry and History.

(iv) If it is Saturday and warm, then Sam is in the park.

(v) Anything anyone eats and is not killed by is food.

i. **Divisible (Perfect_square, Prime) $\Rightarrow$ Divisible(Perfect_Square, square(Prime)**

ii. **$\forall$S $\exists$ P(Perfect_square(S) $\wedge$ Prime(P) $\Rightarrow$ Divisible (S,P))**

iii. **$\neg$ Likes(Alice, Chemistry) $\wedge$ $\neg$ Likes(Alice, History)**

iv. **Is_Saturday(day) $\wedge$ Is_warm(day) $\Rightarrow$ in_park(Sam)**

v. **$\forall$x $\forall$y (Eats(X,Y) $\wedge$ $\neg$ Killed(X, Y) $\Rightarrow$ food (Y)**

- **Dec 2015 (5 marks)**

(e) Represent the following statement into FOPL.

(i) Anyone who kills an animal is loved by no one

(ii) A square is breezy if and only if there is a pit in a neighboring square

i. $\forall x\ (\exists y\ Animal(y) \wedge Kills(x, y)) \Rightarrow \forall z\ (\neg\ Loves(z, x))$

ii. $\exists x\ \exists y\ Square(x) \wedge Breezy(x) \wedge Neighbor\_Square(x, y) \Leftrightarrow has\_Pit\ (y)$

- **Dec 2016 (5 Marks)**

[E] Represent the following statement into FOPL.

(i) Every tree in which any aquatic bird sleeps is beside some lake.

(ii) People try to assassinate rulers they are not loyal to.

i. $\forall x\ (\forall y\ (Tree(x) \wedge Aquatic\_bird(y) \wedge sleeps(y,x)) \Rightarrow (\exists z\ \ lake(z) \wedge besides(x, z))\ )$

ii. $\forall x\ \forall y\ (People(x) \wedge Rulers(y) \wedge try\_assassinate(x,y) \Rightarrow$

$\neg\ Loyal(x, y)\ )$

CONVERTING TO CNF

(a) Explain the steps involved in converting the propositional logic statement into CNF with a suitable example

1. Eliminate all $\leftrightarrow$ connectives

   $(P \leftrightarrow Q) \Rightarrow ((P \rightarrow Q) \wedge (Q \rightarrow P))$

2. Eliminate all $\rightarrow$ connectives

   $(P \rightarrow Q) \Rightarrow (\neg P \vee Q)$

3. Reduce the scope of each negation symbol to a single predicate (DeMorgan's Law)

   $\neg\neg P \Rightarrow P$

   $\neg(P \vee Q) \Rightarrow \neg P \wedge \neg Q$

   $\neg(P \wedge Q) \Rightarrow \neg P \vee \neg Q$

   $\neg(\forall x)P \Rightarrow (\exists x)\neg P$

   $\neg(\exists x)P \Rightarrow (\forall x)\neg P$

4. Standardize variables: rename all variables so that each quantifier has its own unique variable name

   $((\forall x)(\neg P(x))) \vee ((\forall y)(\neg P(y))) \wedge ((\exists y)(\neg P(y)))$

   $((\forall x)(\neg P(x))) \vee ((\forall y)(\neg P(y))) \wedge ((\exists z)(\neg P(z)))$

Converting sentences to clausal form Skolem constants and functions

5. Eliminate existential quantification by introducing Skolem constants/functions

$(\exists x)P(x) \Rightarrow P(c)$

E.g.: $(\exists x)$ lives(x) $\Rightarrow$ Lives(c)

c is a Skolem constant (a brand-new constant symbol that is not used in any other sentence)

$(\forall x)(\exists y)P(x,y) \Rightarrow (\forall x)P(x, f(x))$

since $\exists$ is within the scope of a universally quantified variable, use a **Skolem function f** to construct a new value that **depends on** the universally quantified variable

f must be a brand-new function name not occurring in any other sentence in the KB.

E.g., $(\forall x)(\exists y)$loves(x,y) $\Rightarrow$ $(\forall x)$loves(x,f(x))

In this case, f(x) specifies the person that x loves

$\exists z$ loves(x, z) $\wedge \neg$loves(z) $\Rightarrow$ **loves(x, g(x))** $\wedge \neg$loves (g(x))

Converting sentences to clausal form

6. **Remove universal quantifiers by**

(1) moving them all to the left end;

(2) making the scope of each the entire sentence;

(3) and dropping the "prefix" part

Ex: $(\forall x)P(x) \Rightarrow P(x)$

7. Put into conjunctive normal form (conjunction of disjunctions) using distributive and associative laws

$(P \wedge Q) \vee R \Rightarrow (P \vee R) \wedge (Q \vee R)$

$(P \vee Q) \vee R \Rightarrow (P \vee Q \vee R)$

8. Split conjuncts into separate clauses

$\neg P(x) \wedge (\neg P(x) \vee Q(x, g(x))) \wedge (\neg P(g(x)))$

9. Standardize variables so each clause contains only variable names that do not occur in any other clause

$\neg P(x), (\neg P(y) \vee Q(y, g(y))), (\neg P(g(z)))$

example -1

$$(\forall x)(P(x) \rightarrow ((\forall y)(P(y) \rightarrow P(f(x,y))) \wedge \neg(\forall y)(Q(x,y) \rightarrow P(y))))$$

2. Eliminate $\rightarrow$

$$(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge \neg(\forall y)(\neg Q(x,y) \vee P(y))))$$

3. Reduce scope of negation

$$(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge (\exists y)(Q(x,y) \wedge \neg P(y))))$$

4. Standardize variables

$$(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge (\exists z)(Q(x,z) \wedge \neg P(z))))$$

5. Eliminate existential quantification

$$(\forall x)(\neg P(x) \vee ((\forall y)(\neg P(y) \vee P(f(x,y))) \wedge (Q(x,g(x)) \wedge \neg P(g(x)))))$$

6. Drop universal quantification symbols

$$(\neg P(x) \vee ((\neg P(y) \vee P(f(x,y))) \wedge (Q(x,g(x)) \wedge \neg P(g(x)))))$$

Example-1 contd..

$(\neg P(x) \vee ((\neg P(y) \vee P(f(x,y))) \wedge (Q(x,g(x)) \wedge \neg P(g(x)))))$

7. Convert to conjunction of disjunctions

$$(\neg P(x) \vee \neg P(y) \vee P(f(x,y))) \wedge (\neg P(x) \vee Q(x,g(x))) \wedge (\neg P(x) \vee \neg P(g(x)))$$

8. Create separate clauses

$\neg P(x) \vee \neg P(y) \vee P(f(x,y))$

$\neg P(x) \vee Q(x,g(x))$

$\neg P(x) \vee \neg P(g(x))$

9. Standardize variables

$\neg P(x) \vee \neg P(y) \vee P(f(x,y))$

$\neg P(z) \vee Q(z,g(z))$

$\neg P(w) \vee \neg P(g(w))$

Example on CNF

Practice

Convert the following sentences to CNF

- $\neg Q \Rightarrow (P \wedge R)$

$\neg(\neg Q) \vee (P \wedge R)$ using rule 2

$Q \vee (P \wedge R)$ using rule 3

$(Q \vee P) \wedge (Q \vee R)$ rule 7

- $(P \vee \neg Q) \Rightarrow R$

$\neg(P \vee \neg Q) \vee R$ using rule 2

$(\neg P \wedge \neg \neg Q) \vee R$ using rule 3

$(\neg P \wedge Q) \vee R$ using rule 3

$(\neg P \vee R) \wedge (Q \vee R)$ using rule 7

University Questions

- **Dec 2015 (4 marks)**

Convert the following propositional logic statement into CNF

$$A \rightarrow (B \leftrightarrow C)$$

- Convert the following propositional logic statement into CNF

$(A \leftrightarrow B) \rightarrow C$

- (a) Explain the steps involved in converting the propositional logic statement into CNF with a suitable example

University Questions

- **Dec 2015 (4 marks)**

Convert the following propositional logic statement into CNF

$A \rightarrow (B \leftrightarrow C)$

  - $A \rightarrow (B \leftrightarrow C)$
  - $A \rightarrow ((B \rightarrow C) \wedge (C \rightarrow B))$---step 1
  - $\neg A \vee ((B \rightarrow C) \wedge (C \rightarrow B))$---step 2
  - $\neg A \vee ((\neg B \vee C) \wedge (\neg C \vee B))$—step 2
  - $(\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg C \vee B)$—step 7

University Questions

- **Dec 2016 (4 marks)**

  Convert the following propositional logic statement into CNF
  $(A \leftrightarrow B) \rightarrow C$

  - $((A \rightarrow B) \wedge (B \rightarrow A)) \rightarrow C$
  - $\neg((A \rightarrow B) \wedge (B \rightarrow A)) \vee C$
  - $\neg((\neg A \vee B) \wedge (\neg B \vee A)) \vee C$
  - $((A \wedge \neg B) \vee (B \wedge \neg A)) \vee C$
  - $(((A \wedge \neg B) \vee B) \wedge((A \wedge \neg B) \vee \neg A)) \vee C$
  - $((A \vee B) \wedge (\neg B \vee B) \wedge(A \vee \neg A) \wedge (\neg B \vee \neg A)) \vee C$
  - $((A \vee B) \wedge (T) \wedge(T) \wedge (\neg B \vee \neg A)) \vee C$
  - $((A \vee B) \wedge (\neg B \vee \neg A)) \vee C)$
  - $(A \vee B \vee C) \wedge (\neg B \vee \neg A \vee C)$

Examples: Convert into CNF

1. $\neg(((p \vee \neg Q) \rightarrow R) \rightarrow (P \wedge R))$
2. $p \rightarrow \neg(R \vee \neg Q)$
3. $(Food \rightarrow Party) \vee (Drinks \rightarrow Party)$
4. $(Food \wedge Drinks) \rightarrow Party$
5. $(P \vee Q \vee R) \wedge (P \vee Q) \wedge (P \vee R) \wedge P$

Examples: Convert into CNF

1. $A \Rightarrow B$
2. $A \Rightarrow (B \vee C)$
3. $(A \vee B) \Rightarrow C$
4. $(A \wedge B) \Rightarrow C$
5. $A \Leftrightarrow B$
6. $A \Leftrightarrow (B \vee C)$
7. $(A \vee B) \Leftrightarrow (C \vee D)$
8. $(A \wedge B) \Leftrightarrow (C \wedge D)$
9. $(\neg A \wedge \neg B) \Leftrightarrow (C \wedge D)$
10. $(A \wedge B) \vee (C \wedge D) \Leftrightarrow E$

Inference in FOPL

Resolution

Resolution

- **Resolution** is a theorem proving technique that proceeds by building refutation proofs, i.e., proofs by contradictions.
- **Resolution** is a single inference rule which can efficiently operate on the conjunctive normal form or clausal form.

Resolution reasoning of propositional logic

Resolution rule

Resolution rule

- sound inference rule that works for the KB in the CNF form

$$\frac{A \vee B, \quad \neg B \vee C}{A \vee C}$$

A	B	C	$A \vee B$	$\neg B \vee C$	$A \vee C$
False	*False*	*False*	*False*	*True*	*False*
False	*False*	*True*	*False*	*True*	*True*
False	*True*	*False*	*True*	*False*	*False*
False	*True*	*True*	*True*	*True*	*True*
True	*False*	*False*	*True*	*True*	*True*
True	*False*	*True*	*True*	*True*	*True*
True	*True*	*False*	*True*	*False*	*True*
True	*True*	*True*	*True*	*True*	*True*

RESOLUTION REASONING OF PROPOSITIONAL LOGIC

Proof and Entailment

- Proof is a syntactic notion. In constructing proofs, we don't consider the meaning of sentences.

  KB |- p means we can prove/derive p from KB

- Entailment is a semantic notion. It depends on the meaning we give to logical connectives by associating them with truth tables.

  KB |= p means that p is entailed by KB, that is, whenever KB is true the sentence p is true

- Although we can show entailment by truth tables, we can usually construct proofs more efficiently.

Resolution reasoning of propositional logic

Properties of a Formal System

- Sound
  - if KB |- p, then KB |= p
- Complete
  - if KB |= p, then KB |- p
- Decidable
  - there is an algorithm that can decide in finite time whether any proposition is a theorem or not

Resolution reasoning of propositional logic

Resolution Refutation

- By itself, resolution is a sound rule of inference but not complete.
- However, a technique called resolution refutation is sound and complete.
- Resolution refutation is a form of proof by contradiction. It shows that KB $\models$ p by showing that the set of clauses KB $\cup$ $\{\neg p\}$ is unsatisfiable.
- To use resolution refutation, we first convert sentences to conjunctive normal form

Resolution reasoning of propositional logic

Resolution algorithm

Algorithm:

- **Convert KB to the CNF form;**
- **Apply iteratively the resolution rule** starting from $(KB \wedge \neg \alpha)$ (in CNF form)
- **Stop when**:
  - Contradiction (empty clause) is reached:
    - $A, \neg A \rightarrow \emptyset$
    - proves entailment.
  - No more new sentences can be derived
    - disproves it.

Resolution reasoning of propositional logic

Example. Resolution.

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S

Step 1. convert KB to CNF:

- $P \wedge Q \longrightarrow P \wedge Q$
- $P \Rightarrow R \longrightarrow (\neg P \vee R)$
- $(Q \wedge R) \Rightarrow S \longrightarrow (\neg Q \vee \neg R \vee S)$

KB: $P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S)$

Step 2. Negate the theorem to prove it via refutation

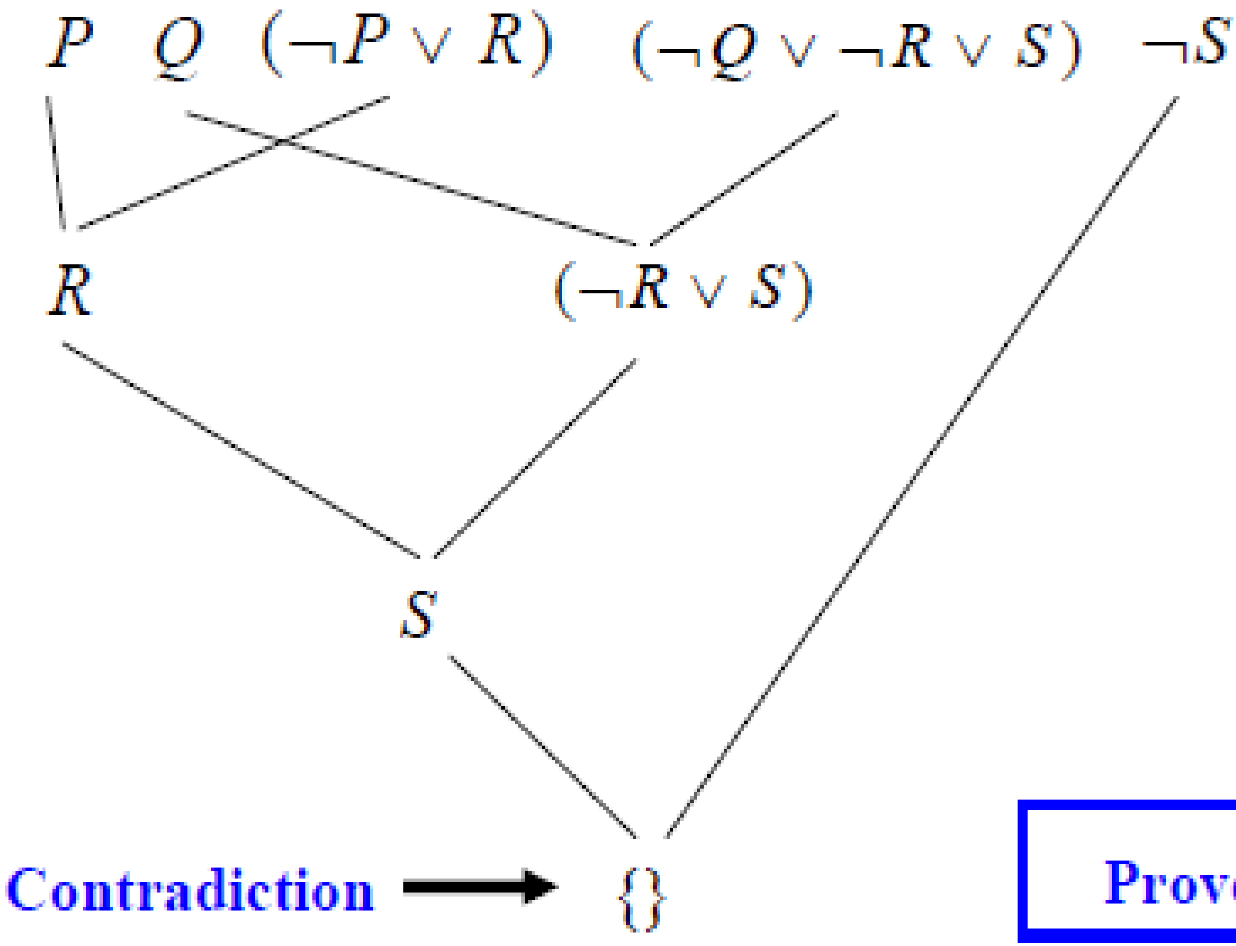
$S \longrightarrow \neg S$

Step 3. Run resolution on the set of clauses

$P \quad Q \quad (\neg P \vee R) \quad (\neg Q \vee \neg R \vee S) \quad \neg S$

Example. Resolution.

KB: $(P \wedge Q) \wedge (P \Rightarrow R) \wedge [(Q \wedge R) \Rightarrow S]$ **Theorem:** S



Proved: $\boxed{S}$

Example 1 on Resolution

Practice

- Express the following statements in propositional logic
  - If it is hot and humid, then it is raining.
  - If it is humid, then it is hot.
  - It is humid.
- Use resolution refutation to prove the following statement
  - It is raining.

- It is Hot=H
- It is Humid=M
- It is Raining=R

Into Proposition logic:

$(H \wedge M) \rightarrow R$

$M \rightarrow H$

M

$\neg R$

Into CNF:

$(\neg H \vee \neg M \vee R)$

$(\neg M \vee H)$

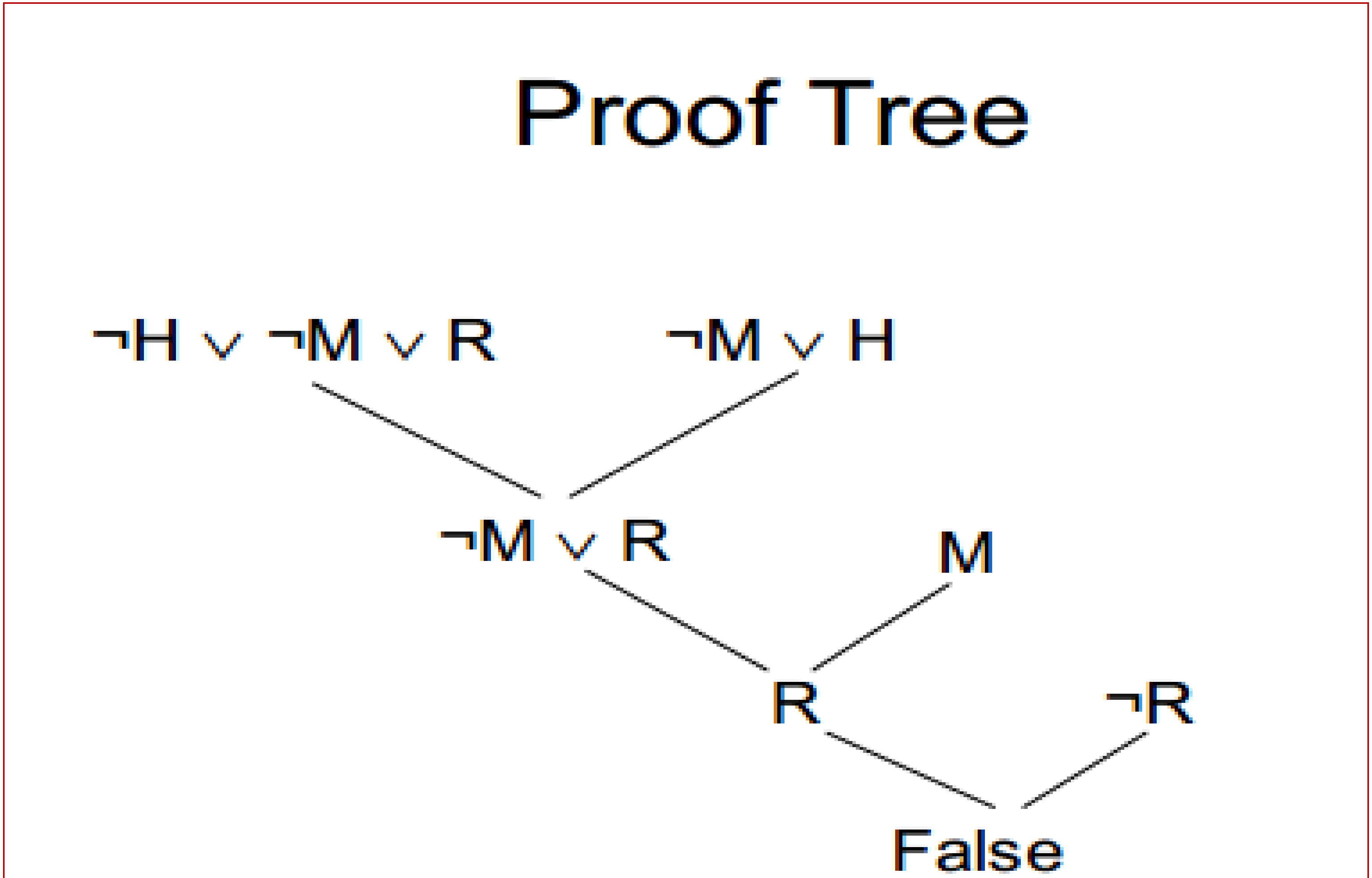
M

$\neg R$

4 Clauses in KB:

$(\neg H \vee \neg M \vee R)$ $(\neg M \vee H)$ M $\neg R$

Example on Resolution



Example 2 on Resolution

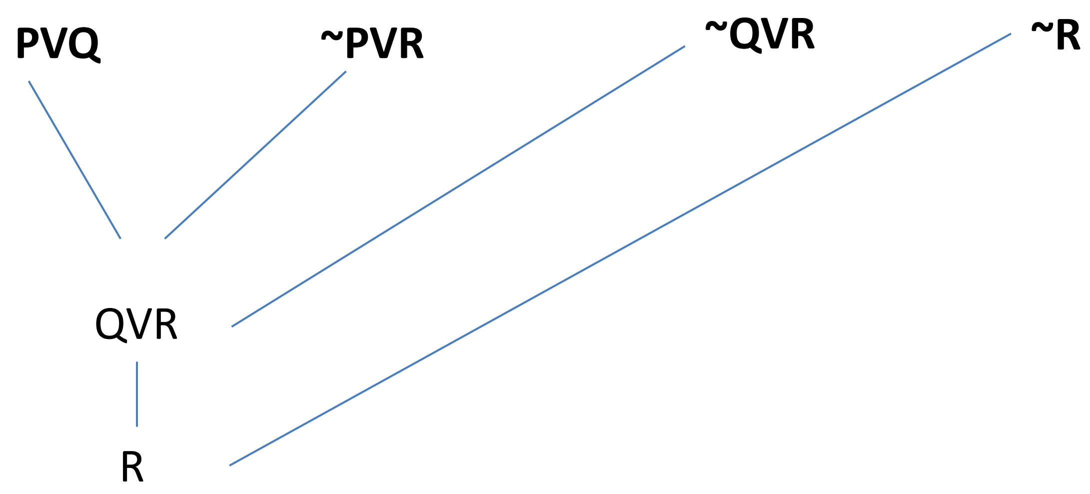
Prove R

1	P v Q
2	P → R
3	Q → R

Step 1: Convert into CNF

1. **PVQ**
2. **~PVR**
3. **~QVR**

To prove R so we will take ~R



Null set

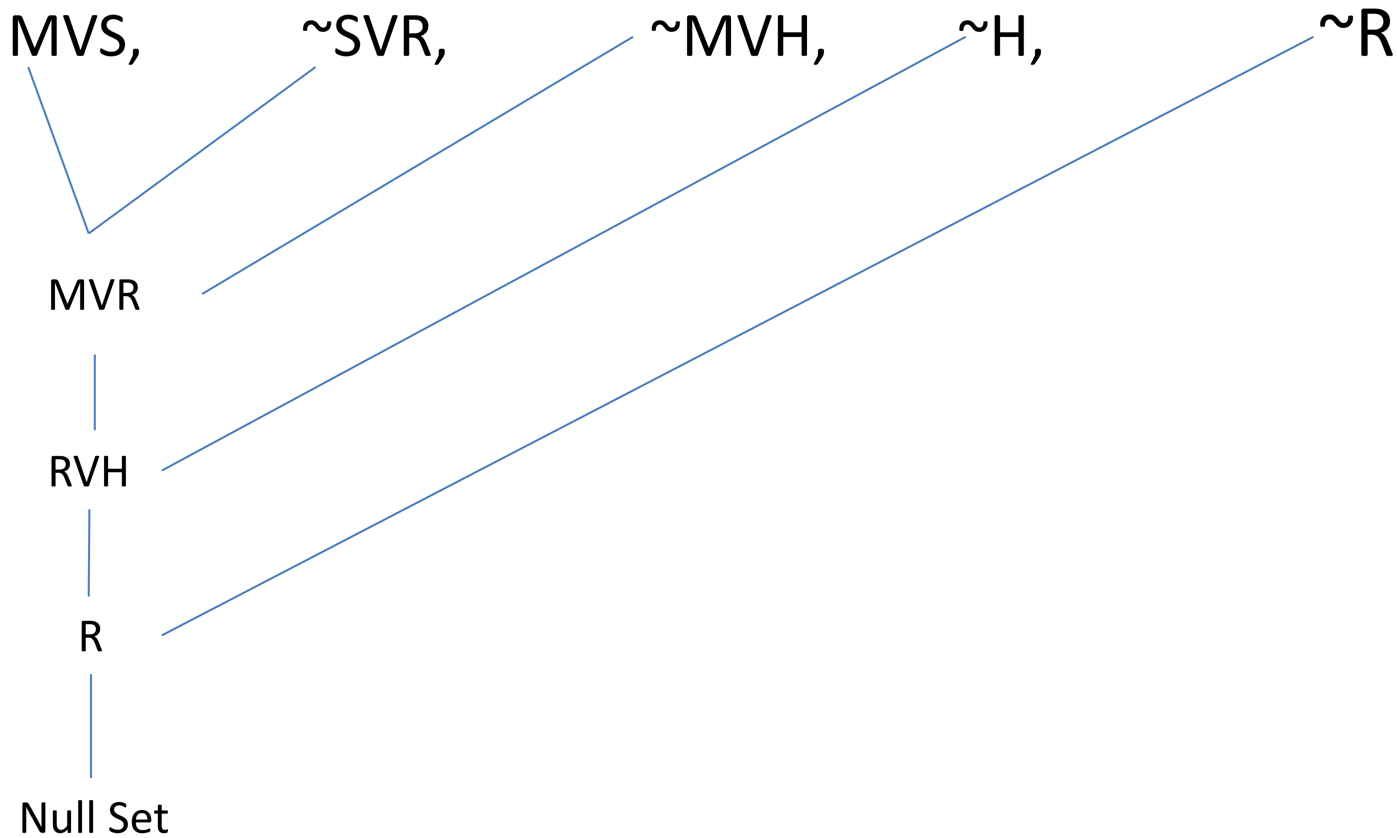
Example 3 on Resolution

- Consider the following Knowledge Base:
- The humidity is high(M) or the sky is cloudy(S).
- If the sky is cloudy(S), then it will rain(R).
- If the humidity is high(M), then it is hot(H).
- It is not hot.
- **Goal:** It will rain(R).

Example 3 on Resolution

- humidity is high= M Sky is cloudy= S
- It will rain=R   It is hot=H
- Into Propositional Logic:
  - MVS
  - S→R
  - M→H
  - ~H
- Into CNF:
  - MVS, ~SVR, ~MVH, ~H
- To Prove: R
  - ~R Contradiction

Example 3 on Resolution



Example 4 on Resolution University Questions

- Dec 2015 (12 Marks)

(a) Consider the following axioms: [4+4+4]
All people who are graduating are happy.
All happy people smile.
Someone is graduating.
(i) Represent these axioms in first order predicate logic.
(ii) Convert each formula to clause form
(iii) Prove that "Is someone smiling?" using resolution technique. Draw the resolution tree.

1. **Represent in FOPL**

   $People(x) \wedge Graduating(x) \Rightarrow Happy(x)$

   $People(x) \wedge Happy(x) \Rightarrow Smile(x)$

   $People(x) \wedge Graduating(x)$

2. **Convert each in Clause Form**

   $\neg People(x) \vee \neg Graduating(x) \vee Happy(x)$

   $\neg People(x) \vee \neg Happy(x) \vee Smile(x)$

   $People(x) \wedge Graduating(x)$

3. **T**o Prove " Is Someone Smiling" i.e. Smile(x) so Refutation says $\neg Smile(x)$ will be in Knowledge base

$\neg$ People(x) $\vee$ $\neg$ Graduating(x) $\vee$ Happy(x)

$\neg$ People(x) $\vee$ $\neg$ Happy(x) $\vee$ Smile(x)

$\neg$ People(x) $\vee$ $\neg$ Graduating(x) $\vee$ Smile(x)

People(x) $\wedge$ Graduating(x)

Smile(x)

$\neg$Smile(x)

$\{\ \}$

Example 5 on Resolution University Questions

- Dec 2016 (12 Marks)

[A] 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 an American.

(i) Represent the above sentences in first order predicate logic (FOPL). [04]

(ii) Convert them to clause form [04]

(iii) Prove that **"West is Criminal"** using resolution technique [04]

Knowledge Base in FOL: *Example*

- 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.

1. it is a crime for an American to sell weapons to hostile nations:
   $American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \Rightarrow Criminal(x)$
2. Nono … has some missiles, i.e., $\exists x\ Owns(Nono,x) \wedge Missile(x)$:
   $Owns(Nono,M_1) \wedge Missile(M_1)$
3. … all of its missiles were sold to it by Colonel West
   $Missile(x) \wedge Owns(Nono,x) \Rightarrow Sells(West,x,Nono)$
4. Missiles are weapons:
   $Missile(x) \Rightarrow Weapon(x)$
5. An enemy of America counts as "hostile“:
   $Enemy(x,America) \Rightarrow Hostile(x)$
6. West, who is American …
   $American(West)$
7. The country Nono, an enemy of America …
   $Enemy(Nono,America)$

FOL into Clause: *Example*

1. it is a crime for an American to sell weapons to hostile nations:
   $American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \Rightarrow Criminal(x)$
   $\neg American(x) \vee \neg Weapon(y) \vee \neg Sells(x,y,z) \vee \neg Hostile(z) \vee Criminal(x)$

2. Nono … has some missiles, i.e., $\exists x\ Owns(Nono,x) \wedge Missile(x)$:
   $Owns(Nono,M_1) \wedge Missile(M_1)$
   $Owns(Nono,M_1) \wedge Missile(M_1)$

3. … all of its missiles were sold to it by Colonel West
   $Missile(x) \wedge Owns(Nono,x) \Rightarrow Sells(West,x,Nono)$
   $\neg Missile(x) \vee \neg Owns(Nono,x) \vee Sells(West,x,Nono)$

4. Missiles are weapons:
   $Missile(x) \Rightarrow Weapon(x)$
   $\neg \mathrm{Missile}(x) \vee \mathrm{Weapon}(x)$

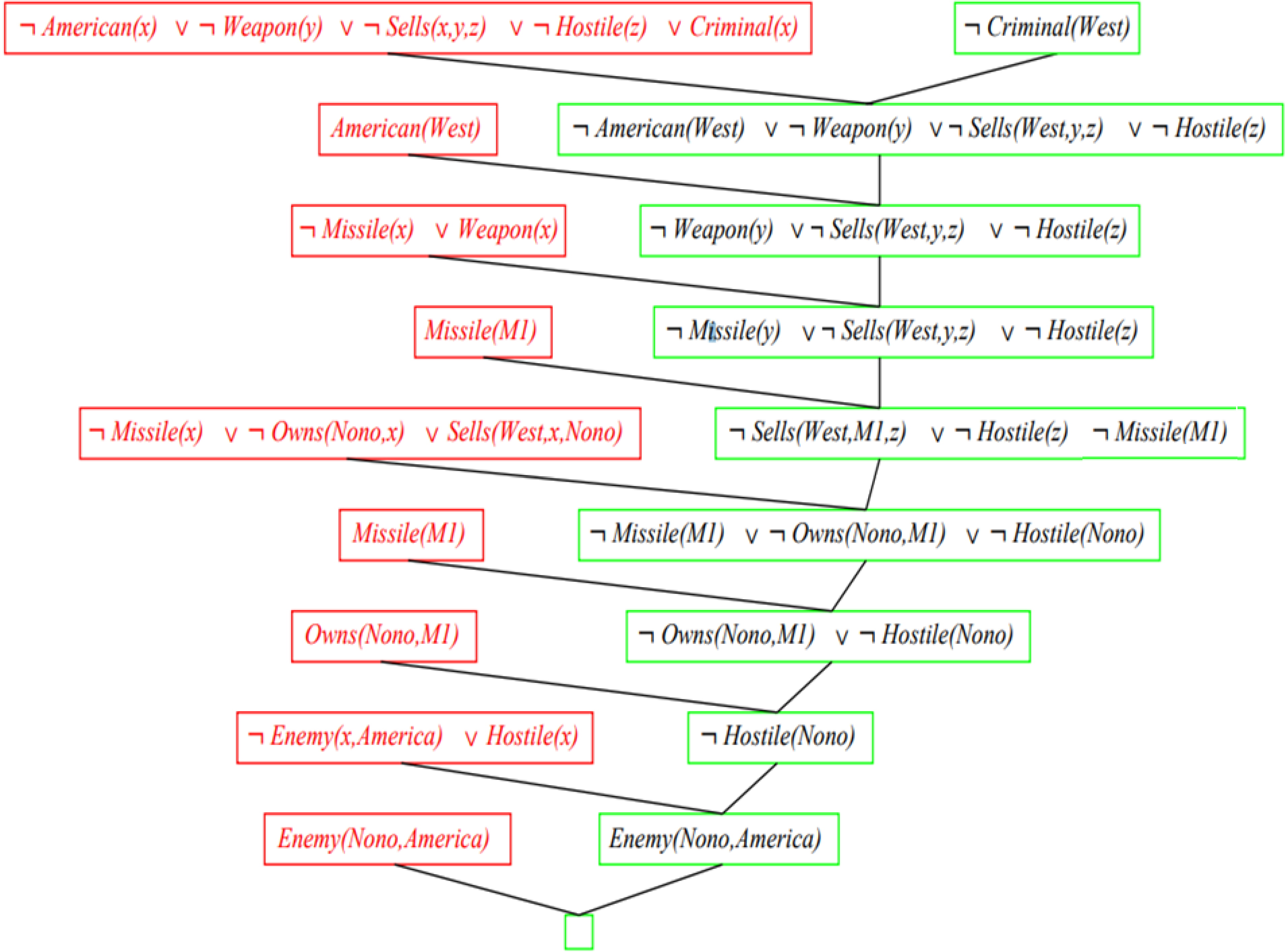
5. An enemy of America counts as "hostile":
   $Enemy(x,America) \Rightarrow Hostile(x)$
   $\neg \mathrm{Enemy}(x,\mathrm{America}) \vee \mathrm{Hostile}(x)$

6. West, who is American …
   $American(West)$
   $American(West)$

7. The country Nono, an enemy of America …
   $Enemy(Nono,America)$
   $Enemy(Nono,America)$

<u>KB To Prove "West is Criminal"</u> ¬ Criminal(West)

1. $\neg$ *American(x)* $\vee$ $\neg$ *Weapon(y)* $\vee$ $\neg$ *Sells(x,y,z)* $\vee$ $\neg$ *Hostile(z)* $\vee$ *Criminal(x)*
2. *Owns(Nono,* M_1*)* $\wedge$ *Missile(*M_1*)*
3. $\neg$ *Missile(x)* $\vee$ $\neg$ *Owns(Nono,x)* $\vee$ *Sells(West,x,Nono)*
4. $\neg$ Missile(x) $\vee$ Weapon(x)
5. $\neg$ Enemy(x,America) $\vee$ Hostile(x)
6. *American(West)*
7. *Enemy(Nono,America)*
8. $\neg$ *Criminal(West)*



¬ American(x) ∨ ¬ Weapon(y) ∨ ¬ Sells(x,y,z) ∨ ¬ Hostile(z) ∨ Criminal(x)
¬ Criminal(West)
American(West)
¬ American(West) ∨ ¬ Weapon(y) ∨ ¬ Sells(West,y,z) ∨ ¬ Hostile(z)
¬ Missile(x) ∨ Weapon(x)
¬ Weapon(y) ∨ ¬ Sells(West,y,z) ∨ ¬ Hostile(z)
Missile(M1)
¬ Missile(y) ∨ ¬ Sells(West,y,z) ∨ ¬ Hostile(z)
¬ Missile(x) ∨ ¬ Owns(Nono,x) ∨ Sells(West,x,Nono)
¬ Sells(West,M1,z) ∨ ¬ Hostile(z) ¬ Missile(M1)
Missile(M1)
¬ Missile(M1) ∨ ¬ Owns(Nono,M1) ∨ ¬ Hostile(Nono)
Owns(Nono,M1)
¬ Owns(Nono,M1) ∨ ¬ Hostile(Nono)
¬ Enemy(x,America) ∨ Hostile(x)
¬ Hostile(Nono)
Enemy(Nono,America)
Enemy(Nono,America)

Example 6 on Resolution with Refutation Numerical:

Consider the following facts about dolphins:

Whoever can read is literate. Dolphins are not literate. Some dolphins are intelligent.

(i) Represent the above sentences in first order predicate logic (FOPL).
(ii) Convert them to clause form
(iii)Prove that **"Some who are Intelligent cannot read"** using resolution technique

Convert into FOPL:

1. Whoever can read is literate:

   $\forall$x, read(x) -> literate (x)

2. Dolphins are not literate.

   $\forall$x, dolphin(x) -> ~literate(x)

3. Some dolphins are intelligent.

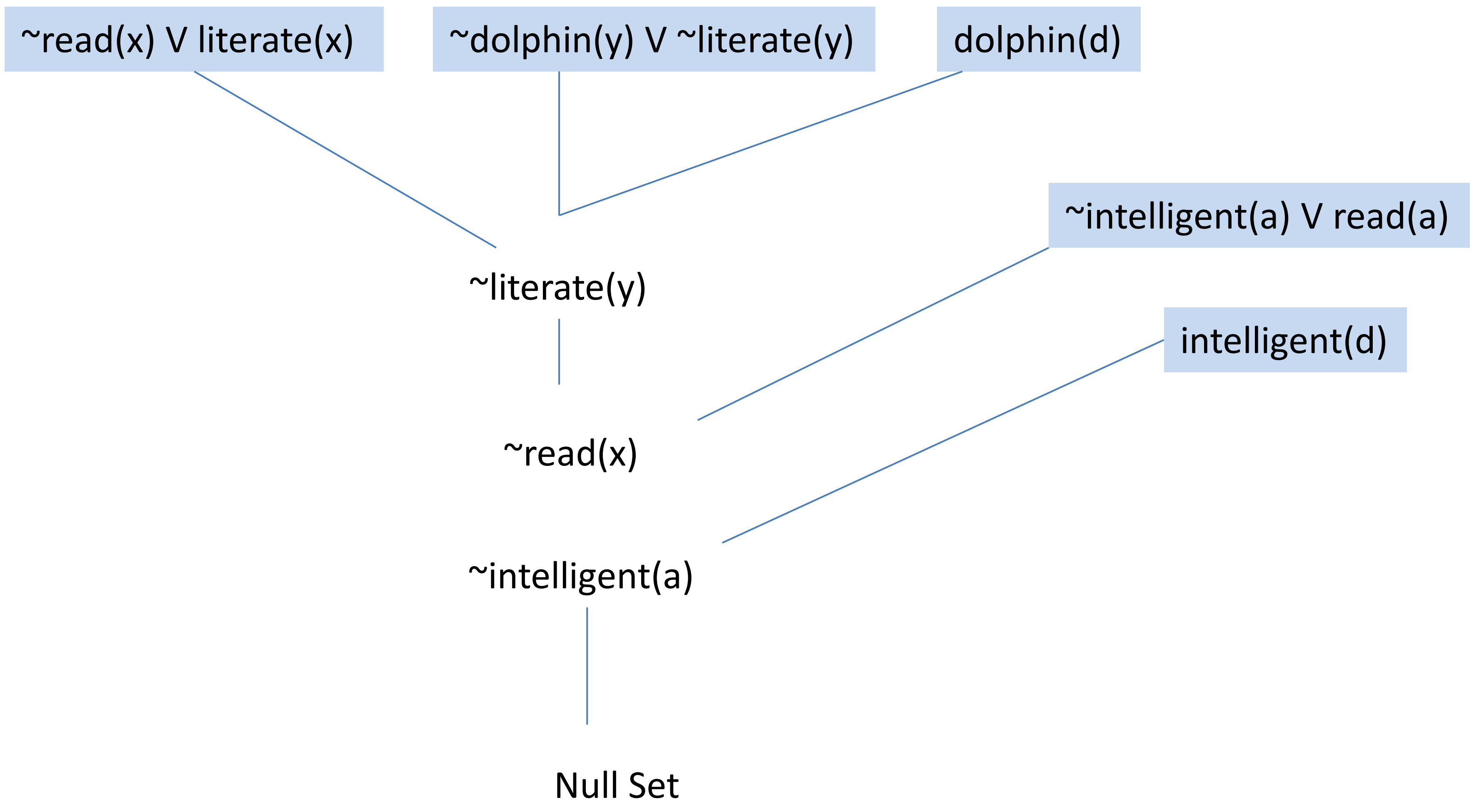
   $\exists$x, dolphin(x) ^ intelligent(x)

To Prove: Some who are intelligent cannot read

4. **$\exists$x, intellgent(x) ^ ~read(x)**

Convert into CNF:

1. ~read(x) V literate(x) ,
2. ~dolphin(y) V ~literate(y),
3. dolphin(d) , intelligent(d),
4. ~intelligent(a) V read(a)


~read(x) V literate(x)
~dolphin(y) V ~literate(y)
dolphin(d)
~intelligent(a) V read(a)
~literate(y)
intelligent(d)
~read(x)
~intelligent(a)
Null Set

Example 7 on Resolution

Did Curiosity kill the cat

- Jack owns a dog
- Every dog owner is an animal lover
- No animal lover kills an animal
- Either Jack or Curiosity killed the cat, who is named Tuna.
- Did Curiosity kill the cat?

Practice example
Did Curiosity kill the cat

- Jack owns a dog. Every dog owner is an animal lover. No animal lover kills an animal. Either Jack or Curiosity killed the cat, who is named Tuna. Did Curiosity kill the cat?
- These can be represented as follows:

  A. $(\exists x)\ Dog(x) \wedge Owns(Jack,x)$

  B. $(\forall x)\ ((\exists y)\ Dog(y) \wedge Owns(x, y)) \rightarrow AnimalLover(x)$

  C. $(\forall x)\ AnimalLover(x) \rightarrow ((\forall y)\ Animal(y) \rightarrow \neg Kills(x,y))$

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

  E. $Cat(Tuna)$

  F. $(\forall x)\ Cat(x) \rightarrow Animal(x)$

  G. $Kills(Curiosity, Tuna)$ GOAL

$\exists x$ Dog(x) $\wedge$ Owns(Jack,x)
$\forall x$ ($\exists y$) Dog(y) $\wedge$ Owns(x, y) $\rightarrow$ AnimalLover(x)
$\forall x$ AnimalLover(x) $\rightarrow$ ($\forall y$ Animal(y) $\rightarrow$ $\neg$Kills(x,y))
Kills(Jack,Tuna) $\vee$ Kills(Curiosity,Tuna)
Cat(Tuna)
$\forall x$ Cat(x) $\rightarrow$ Animal(x)
Kills(Curiosity, Tuna)

- **Convert to clause form**

  A1. (Dog(D))

  A2. (Owns(Jack,D))

  B. ($\neg$Dog(y), $\neg$Owns(x, y), AnimalLover(x))

  C. ($\neg$AnimalLover(a), $\neg$Animal(b), $\neg$Kills(a,b))

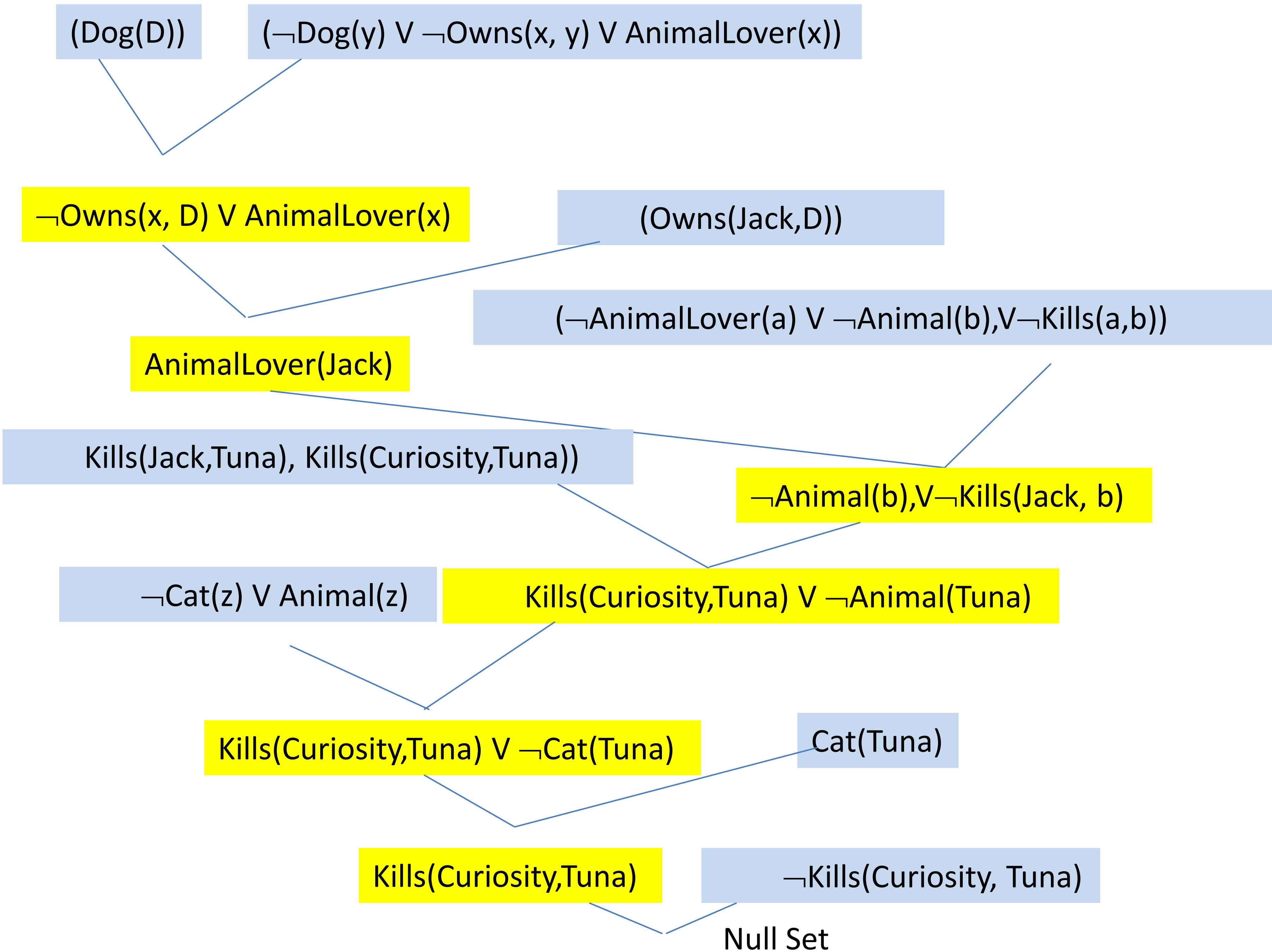
  D. (Kills(Jack,Tuna), Kills(Curiosity,Tuna))

  E. Cat(Tuna)

  F. ($\neg$Cat(z), Animal(z))

- **Add the negation of query:**

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


(Dog(D))
(¬Dog(y) V ¬Owns(x, y) V AnimalLover(x))
¬Owns(x, D) V AnimalLover(x)
(Owns(Jack,D))
(¬AnimalLover(a) V ¬Animal(b),V¬Kills(a,b))
AnimalLover(Jack)
Kills(Jack,Tuna), Kills(Curiosity,Tuna))
¬Animal(b),V¬Kills(Jack, b)
¬Cat(z) V Animal(z)
Kills(Curiosity,Tuna) V ¬Animal(Tuna)
Kills(Curiosity,Tuna) V ¬Cat(Tuna)
Cat(Tuna)
Kills(Curiosity,Tuna)
¬Kills(Curiosity, Tuna)
Null Set

Unification

Inference rules

- "It is illegal for all students to copy music."
- "Joe is a student."
- "Every student copies some music."
- Is Joe a criminal?

- **Knowledge Base:**

$$\forall x, \exists y \;\; Student(x) \wedge Music(y) \wedge Copies(x, y) \Rightarrow Criminal(x) \quad (1)$$

$$Student(Joe) \quad (2)$$

$$\forall x \;\; \exists y \;\; Student(x) \wedge Music(y) \wedge Copies(x, y) \quad (3)$$

$$\text{From}: \forall x \;\; \exists y \;\; Student(x) \wedge Music(y) \wedge Copies(x, y)$$

$$\exists y \;\; Student(Joe) \wedge Music(y) \wedge Copies(Joe, y)$$

Universal Elimination

$$Student(Joe) \wedge Music(SomeSong) \wedge Copies(Joe, SomeSong)$$

Modus Ponens

$$Criminal(Joe)$$

Unification

- Lifted inference rules require finding substitutions that make different logical expression looks identical. This process is called unification.
- Two terms UNIFY if there is a common substitution for all variables which makes them identical.
  - **Recall:** Subst(θ, p) = result of substituting θ into sentence p
- **Unify algorithm:** Takes 2 sentences p and q and returns a unifier if one exists

$$\boxed{\textbf{Unify}(p, q) = \theta \quad \textbf{where Subst}(\theta, p) = \textbf{Subst}(\theta, q)}$$

- **Example:**
  p = Knows(John, x)
  q = Knows(John, Jane)

Unify(p, q) = {x/Jane}

Unification

- It's a matching procedure that compares two literals and discovers whether there exists a set of substitutions that can make them identical.

- E.g.

  Hate(marcus , X)   Hate (marcus , caesar)

  caesar/ X

  e.g. 2.

  Hate(X,Y)   Hate(john, Z)   could be unified as:

  John/X   and   y/z

Can we unify the following clauses:

Loves(x,x)
¬Loves(Bill,Paula)

No cant!!

Loves(Mother-of(x),x)
¬Loves(Mother-of(Bill),Bill)

Yes!! X=bill

Loves(Mother-of(x),x)
¬Loves(Father-of(y),y)

No cant!!

Loves(y,Mother-of(y)) | {x/y}
¬Loves(x,x)

No cant!!

Chaining algorithm:

Forward & Backward

Chaining Algorithm

- Simple methods used by most inference engines to produce a line of reasoning

1. **Forward chaining:** the engine begins with the initial content of the workspace and proceeds toward a final conclusion
2. **Backward chaining:** the engine starts with a goal and finds knowledge to support that goal

Forward chaining example

- **Start with facts and apply rules until no new facts appear. Use substitutions.**
- This defines a **forward-chaining** inference procedure because it moves "forward" from the KB to the goal [eventually]
- **Example**
  - KB:
    - allergies(X) $\rightarrow$ sneeze(X)
    - cat(Y) $\wedge$ allergic-to-cats(X) $\rightarrow$ allergies(X)
    - cat(Felix)
    - allergic-to-cats(Lise)
  - Goal:
    - sneeze(Lise)

Forward chaining algorithm

- **Read the initials facts**
- **Begin**
  - Filter Phase => Find the fired rules
  - **While** Fired rules not empty **AND** not end **DO**
    - Choice Phase => Solve the conflicts
    - Apply the chosen rule
    - Modify (if any) the set of rule
  - **End do**
- **End**

Forward Chaining Example 1

- Knowledge Base:
  - If [X croaks and eats flies] Then [X is a frog]
  - If [X is a frog] Then [X is colored green]
  - [Fritz croaks and eats flies]
- Goal:
  - [Fritz is colored Green]?

Forward Chaining Example

Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

[Fritz croaks and eats flies]

Goal

[Fritz is colored Green]?

Forward Chaining Example

Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

[Fritz croaks and eats flies]

Goal

[Fritz is colored Green]?

Forward Chaining Example

If [X croaks and eats flies]
Then [X is a frog]

[Fritz croaks and eats flies]

[Fritz is a frog]

Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

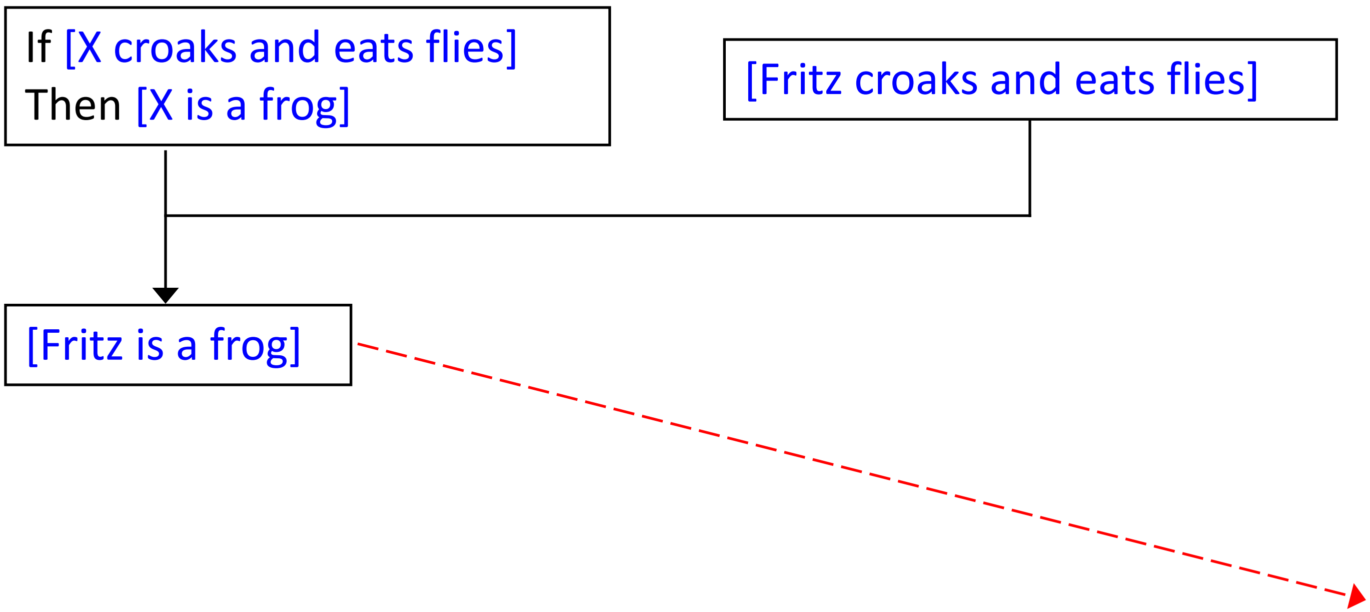
If [X is a frog]
Then [X is colored green]

[Fritz croaks and eats flies]

Goal

[Fritz is colored Green]?

Forward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

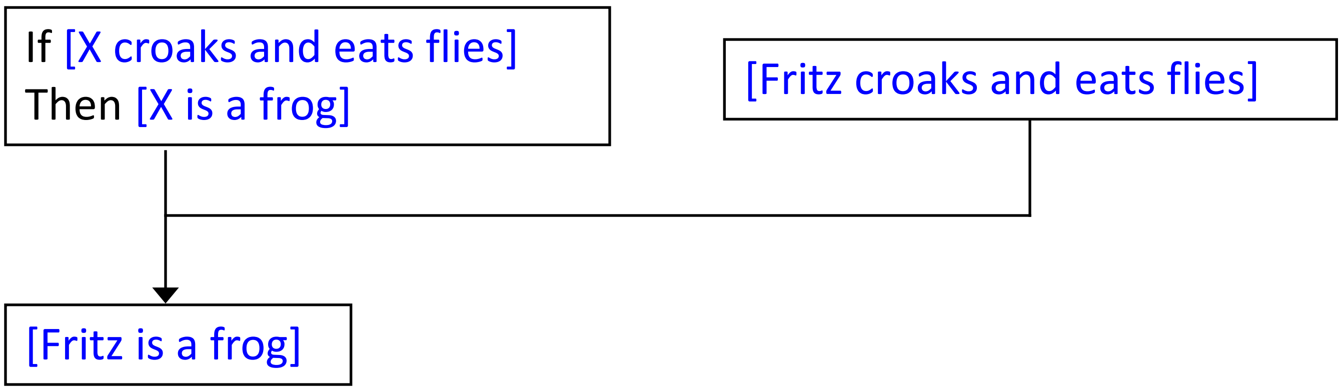
[Fritz croaks and eats flies]

[Fritz is a frog]

Goal

[Fritz is colored Green]?

Forward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

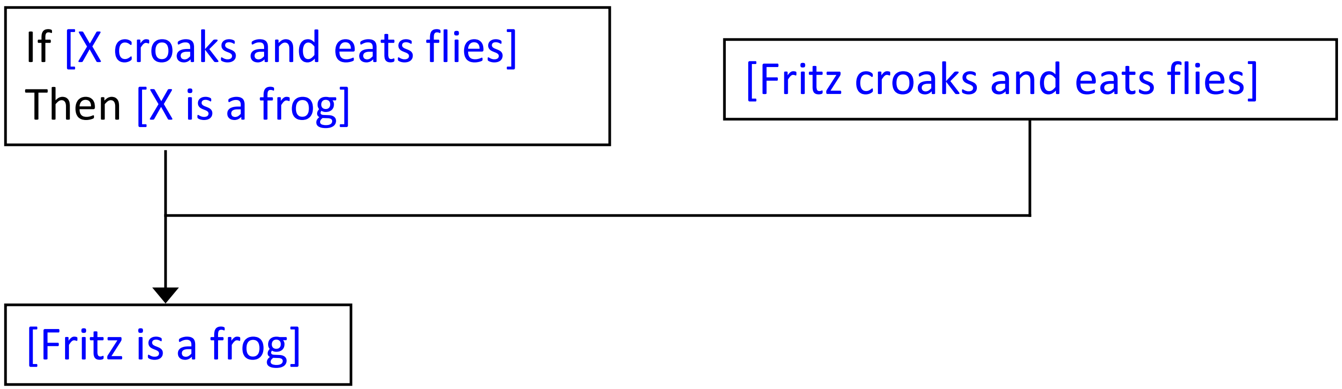
[Fritz croaks and eats flies]

[Fritz is a frog]

Goal

? [Fritz is colored Green]?

Forward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

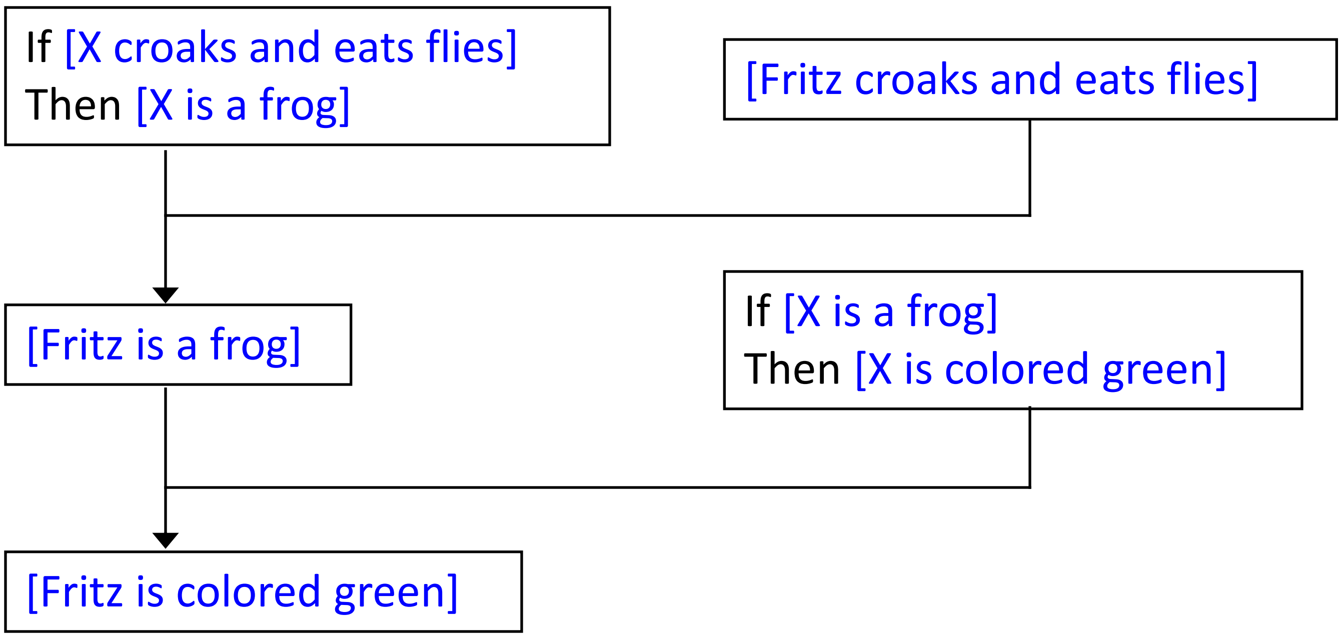
[Fritz croaks and eats flies]

[Fritz is a frog]

Goal

[Fritz is colored Green]?

Forward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

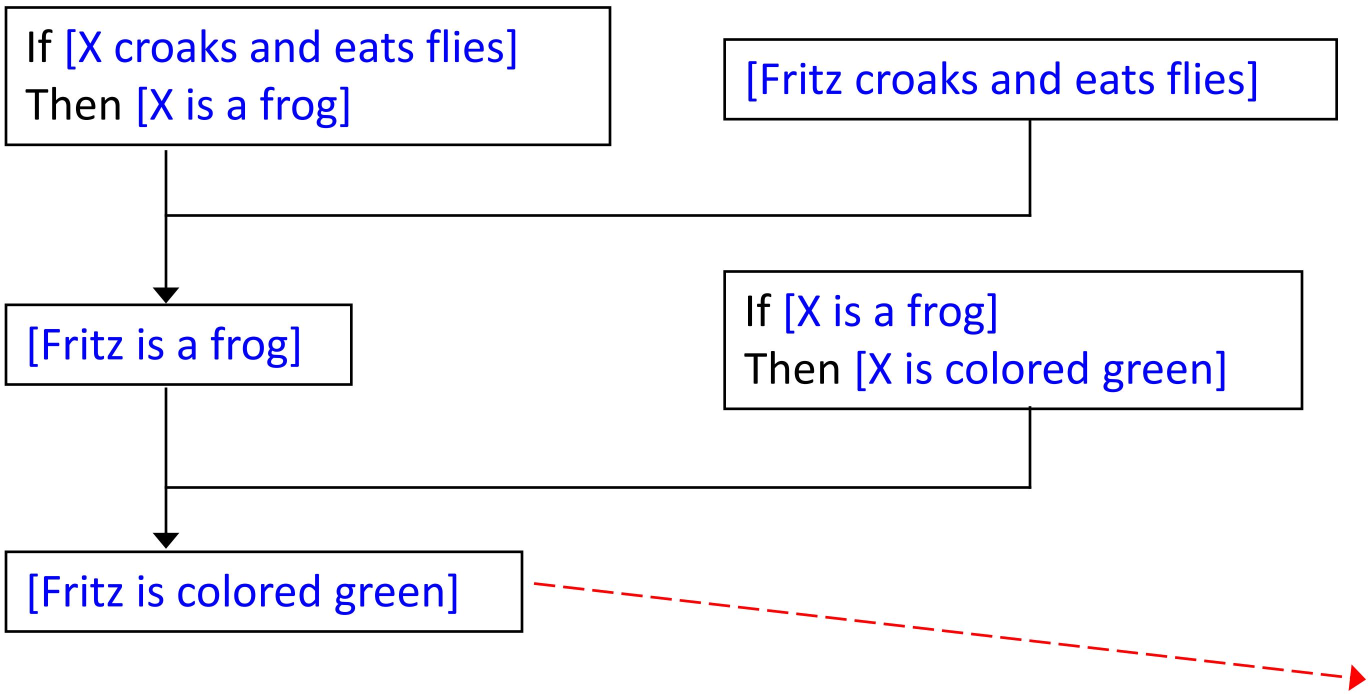
[Fritz croaks and eats flies]

[Fritz is a frog]

Goal

[Fritz is colored Green]?

Forward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

[Fritz croaks and eats flies]

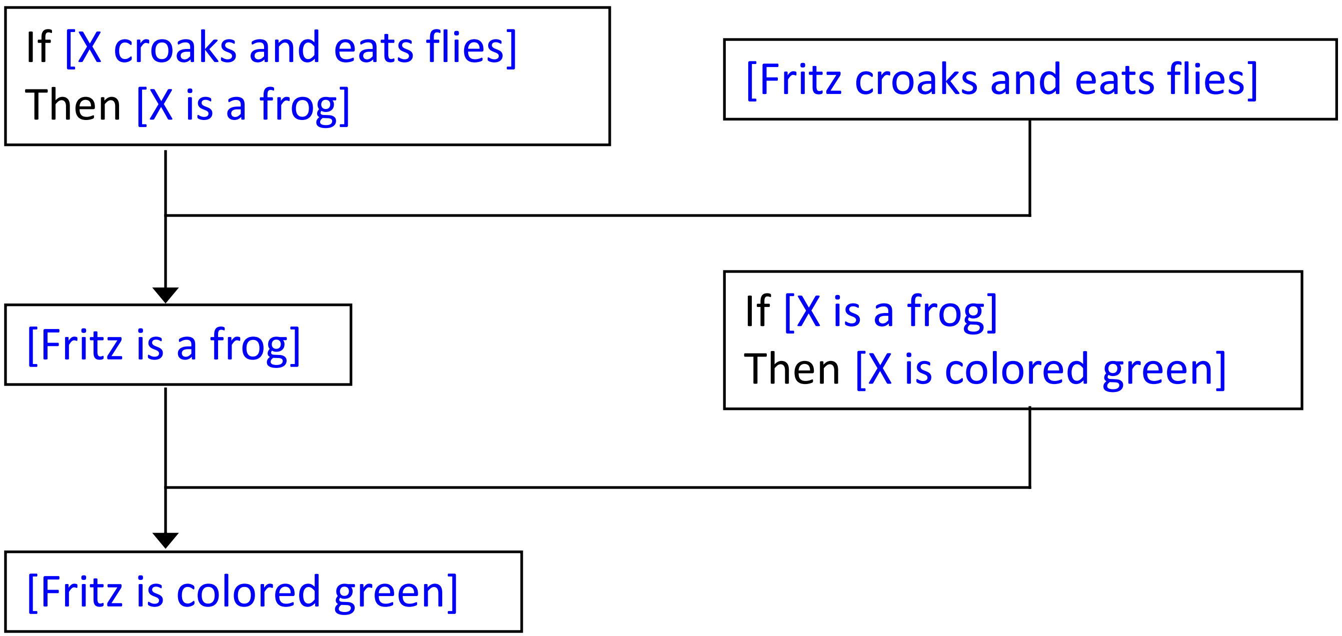
[Fritz is a frog]

[Fritz is colored green]

Goal

[Fritz is colored Green]?

Forward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

[Fritz croaks and eats flies]

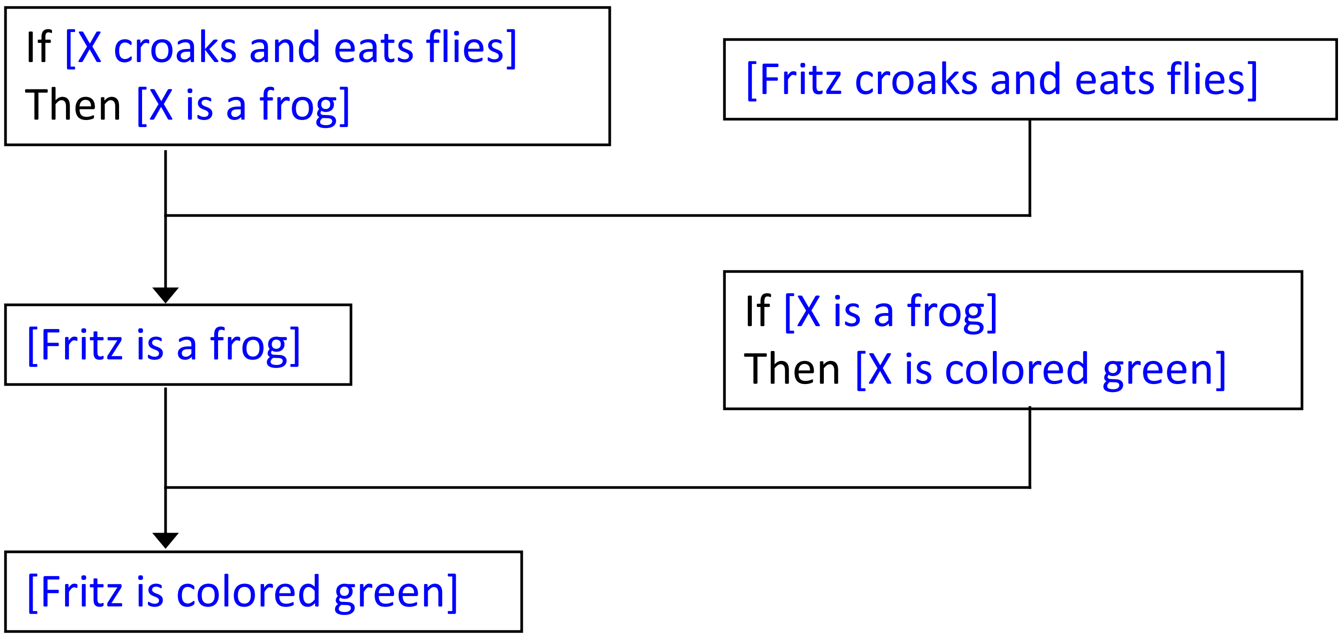
[Fritz is a frog]

[Fritz is colored green]

Goal

? [Fritz is colored Green]?

Forward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X chirps and sings]
Then [X is a canary]

If [X is a frog]
Then [X is colored green]

If [X is a canary]
Then [X is colored yellow]

[Fritz croaks and eats flies]

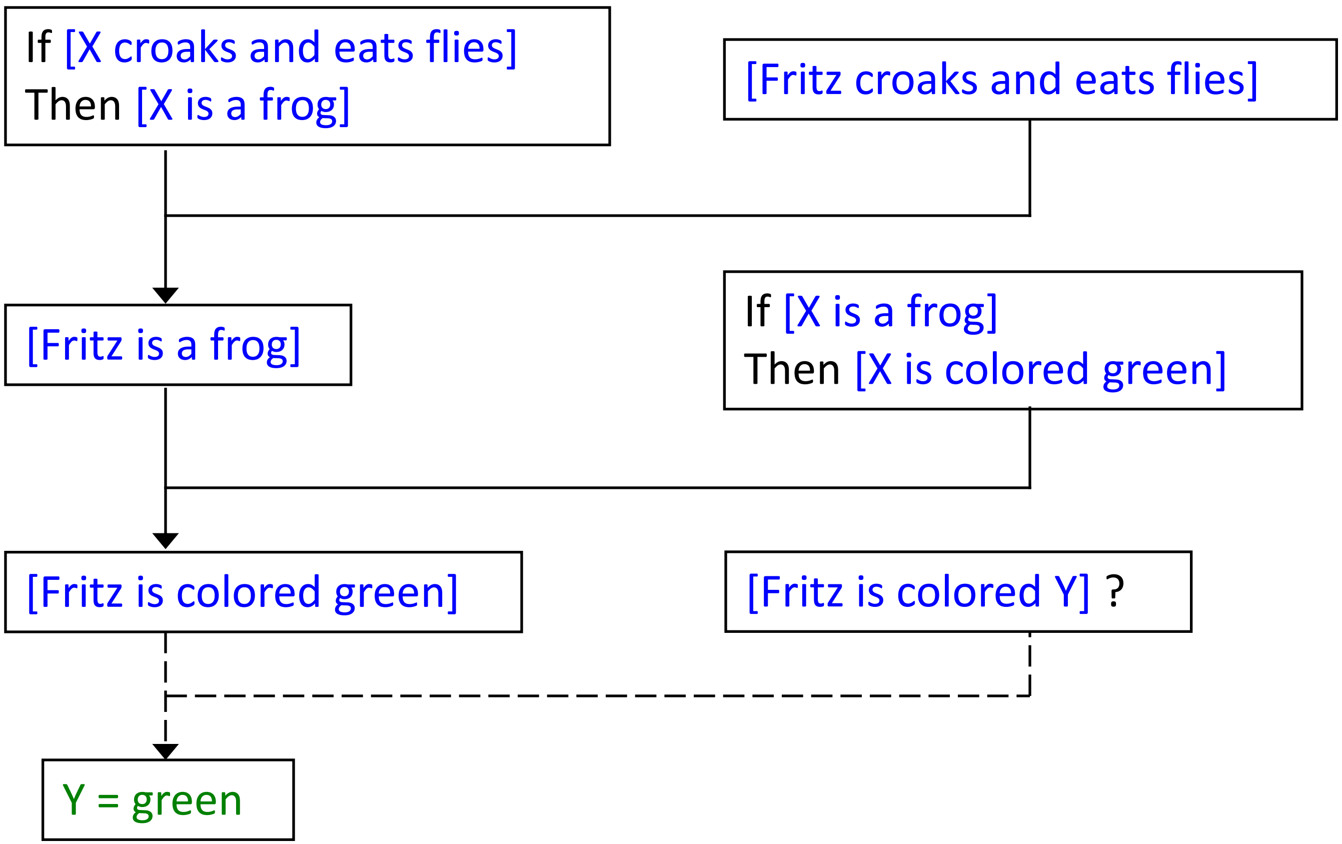
[Fritz is a frog]

[Fritz is colored green]

Goal

[Fritz is colored Green]?

Forward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X chirps and sings]
Then [X is a canary]

If [X is a frog]
Then [X is colored green]

If [X is a canary]
Then [X is colored yellow]

[Fritz croaks and eats flies]

[Fritz is a frog]

[Fritz is colored green]

Goal

[Fritz is colored Green]?

Knowledge Base in FOL: *Example 2*

- 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.

1. it is a crime for an American to sell weapons to hostile nations:
   American(x) ∧ Weapon(y) ∧ Sells(x,y,z) ∧ Hostile(z) ⇒ Criminal(x)

2. Nono … has some missiles, i.e., ∃x Owns(Nono,x) ∧ Missile(x):
   $Owns(Nono,M_1) \wedge Missile(M_1)$

3. … all of its missiles were sold to it by Colonel West
   Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)

4. Missiles are weapons:
   Missile(x) ⇒ Weapon(x)

5. An enemy of America counts as "hostile“:
   Enemy(x,America) ⇒ Hostile(x)

6. West, who is American …
   American(West)

7. The country Nono, an enemy of America …
   Enemy(Nono,America)

Forward Chaining

- Start with facts and apply rules until no new facts appear. Use substitutions.
- **Iteration 1: using facts:**
  - *Missile(M1), American(West), Owns(Nono,M1), Enemy (Nono ,America)*
- **Derive:**
  - $Enemy(x, America) \Rightarrow Hostile(x)$ **[Hostile(Nono)]**
  - $Missile(x) \Rightarrow Weapon(x)$ **[Weapon(M1)]**
  - $Missile(x) \wedge Owns(Nono, x) \Rightarrow Sells(West, x, Nono)$

    [Sells(West, M1, Nono)].
- **Next Iteration:**
  - $American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \Rightarrow Criminal(x)$

    [Criminal(West)]
- Forward chaining ok if few facts and rules, but it is undirected.

Forward chaining proof

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

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

Owns(Nono,M1) and Missile(M1)

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

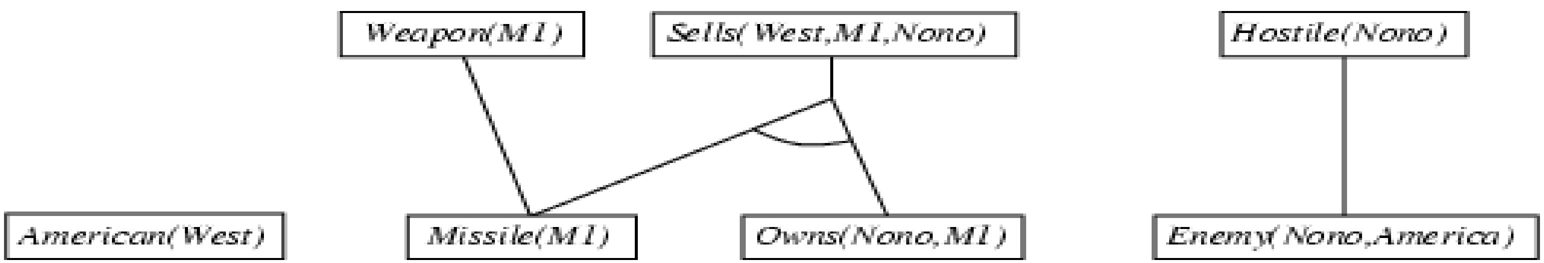
$Missile(x) \Rightarrow Weapon(x)$

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

American(West)

Enemy(Nono,America)

Forward chaining proof



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

Owns(Nono,M1) and Missile(M1)

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

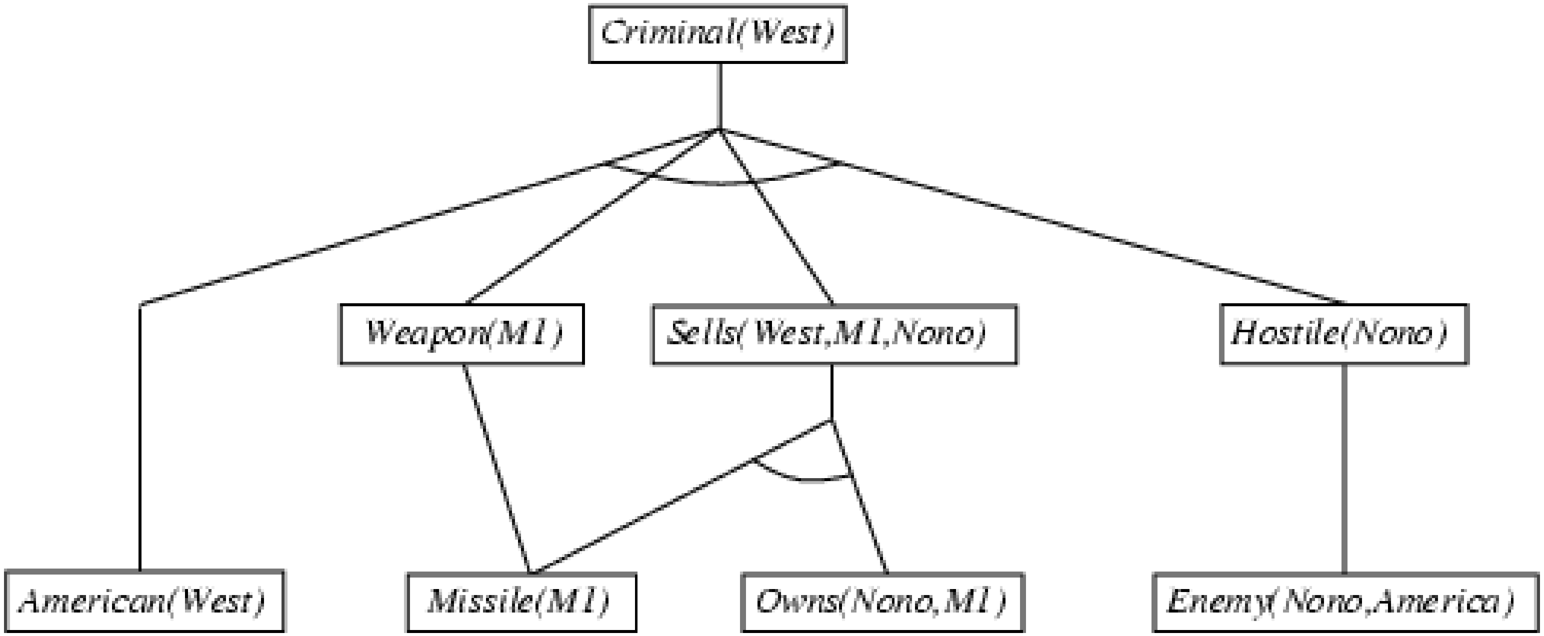
$Missile(x) \Rightarrow Weapon(x)$

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

American(West)

Enemy(Nono,America)

Forward chaining proof



American(x) ∧ Weapon(y) ∧ Sells(x,y,z) ∧ Hostile(z) ⇒ Criminal(x)

Owns(Nono,M1) and Missile(M1)

Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)

Missile(x) ⇒ Weapon(x)

Enemy(x,America) ⇒ Hostile(x)

American(West)

Enemy(Nono,America)

Solve using Forward Chaining Algorithm

- **Example 3**
  - KB:
    - allergies(X) $\rightarrow$ sneeze(X)
    - cat(Y) $\wedge$ allergic-to-cats(X) $\rightarrow$ allergies(X)
    - cat(Felix)
    - allergic-to-cats(Lise)
  - Goal:
    - sneeze(Lise)

Solve using Forward Chaining Algorithm

– KB:
- allergies(X) → sneeze(X) ------1
- cat(Y) ∧ allergic-to-cats(X) → allergies(X) ------2
- cat(Felix) ------3
- allergic-to-cats(Lise)-------4

– Goal:
- sneeze(Lise)

sneeze(Lise) X|Lise

allergies(Lise) X|Lise, Y|Felix

cat(Felix) allergic-to-cats(Lise)

Example 4 on Forward chaining

Did Curiosity kill the cat

- Jack owns a dog
- Every dog owner is an animal lover
- No animal lover kills an animal
- Either Jack or Curiosity killed the cat, who is named Tuna.
- Did Curiosity kill the cat?

Practice example
Did Curiosity kill the cat

- Jack owns a dog. Every dog owner is an animal lover. No animal lover kills an animal. Either Jack or Curiosity killed the cat, who is named Tuna. Did Curiosity kill the cat?
- These can be represented as follows:

A. $(\exists x)\ Dog(x) \wedge Owns(Jack,x)$

B. $(\forall x)\ ((\exists y)\ Dog(y) \wedge Owns(x, y)) \rightarrow AnimalLover(x)$

C. $(\forall x)\ AnimalLover(x) \rightarrow ((\forall y)\ Animal(y) \rightarrow \neg Kills(x,y))$

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

E. $Cat(Tuna)$

F. $(\forall x)\ Cat(x) \rightarrow Animal(x)$

G. $Kills(Curiosity, Tuna)$ ← GOAL

Did Curiosity kill the cat

A. $Dog(x) \wedge Owns(Jack,x)$

B. $(Dog(y) \wedge Owns(x, y)) \rightarrow AnimalLover(x)$

C. $AnimalLover(x) \rightarrow (Animal(y) \rightarrow \neg Kills(x,y))$

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

E. $Cat(Tuna)$

F. $Cat(x) \rightarrow Animal(x)$

$Kills(Curiosity,Tuna)$

$Kills(Jack,Tuna)$ V $Kills(Curiosity,Tuna)$

$AnimalLover(Jack) \rightarrow (Animal(Tuna) \rightarrow \neg Kills(Jack,Tuna))$

$Animal(Tuna)$

$AnimalLover(Jack)$

$Cat(Tuna)$

$Dog(x) \wedge Owns(Jack,x)$

Backward Chaining

- Proofs start with the goal query, find rules with that conclusion, and then prove each of the antecedents in the implication
- Keep going until you reach premises
- **Start with goal, Criminal(West) and set up sub goals.**
- This ends when all sub goals are validated.
- **Iteration 1:** sub goals American(x), Weapons(y) and Hostile(z).
  - Etc. Eventually all sub goals unify with facts.

Backward chaining algorithm

- Filter Phase
- **IF** set of selected rules is empty **THEN** Ask the user
- **ELSE**
  - **WHILE** not end **AND** we have a selected rules **DO**
    - Choice Phase
    - Add the conditions of the rules
    - **IF** the condition not solved **THEN** put the condition as a goal to solve
  - **END WHILE**

Backward Chaining Example 1

- Knowledge Base:
  - If [X croaks and eats flies] Then [X is a frog]
  - If [X is a frog] Then [X is colored green]
  - [Fritz croaks and eats flies]
- Goal:
  - [Fritz is colored Green]?

Backward Chaining Example

Knowledge Base
If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

[Fritz croaks and eats flies]

Goals
[Fritz is colored Green]?

Backward Chaining Example

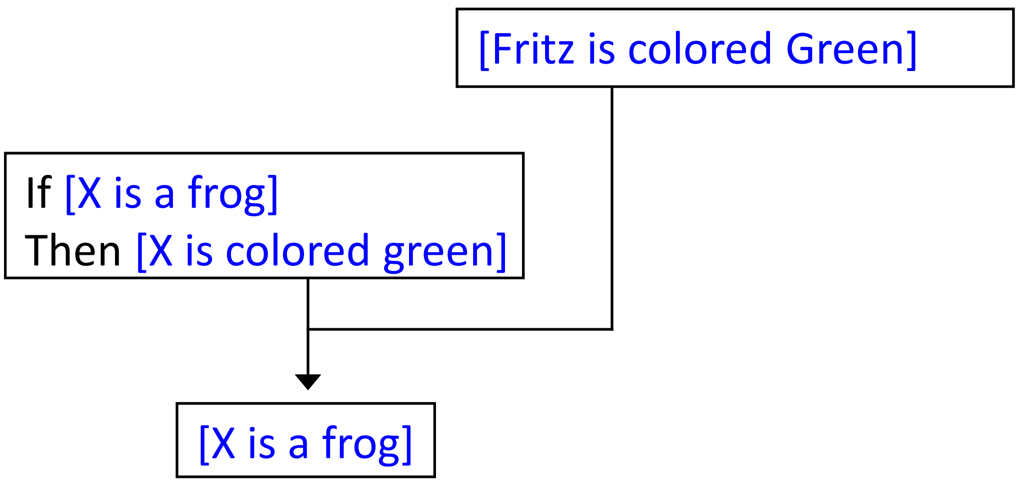
Knowledge Base
If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

[Fritz croaks and eats flies]

Goals
[Fritz is colored Green]?

Backward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

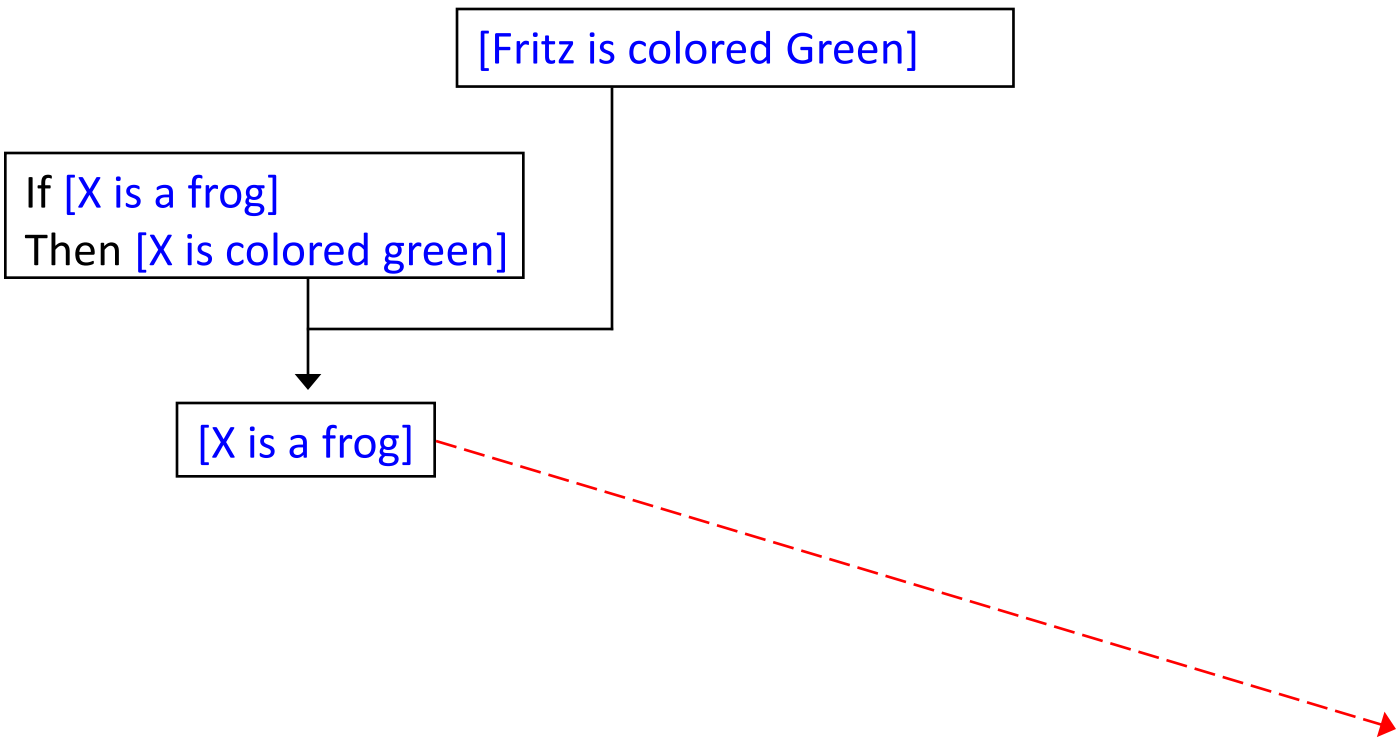
If [X is a frog]
Then [X is colored green]

[Fritz croaks and eats flies]

Goals

[Fritz is colored Green]?

Backward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

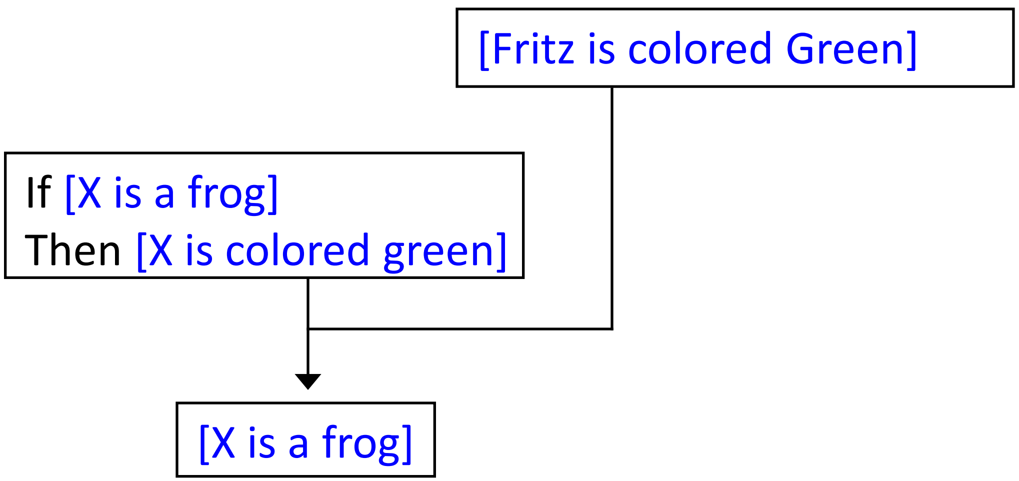
If [X is a frog]
Then [X is colored green]

[Fritz croaks and eats flies]

Goals

[Fritz is colored Green]?
[X is a frog]

Backward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

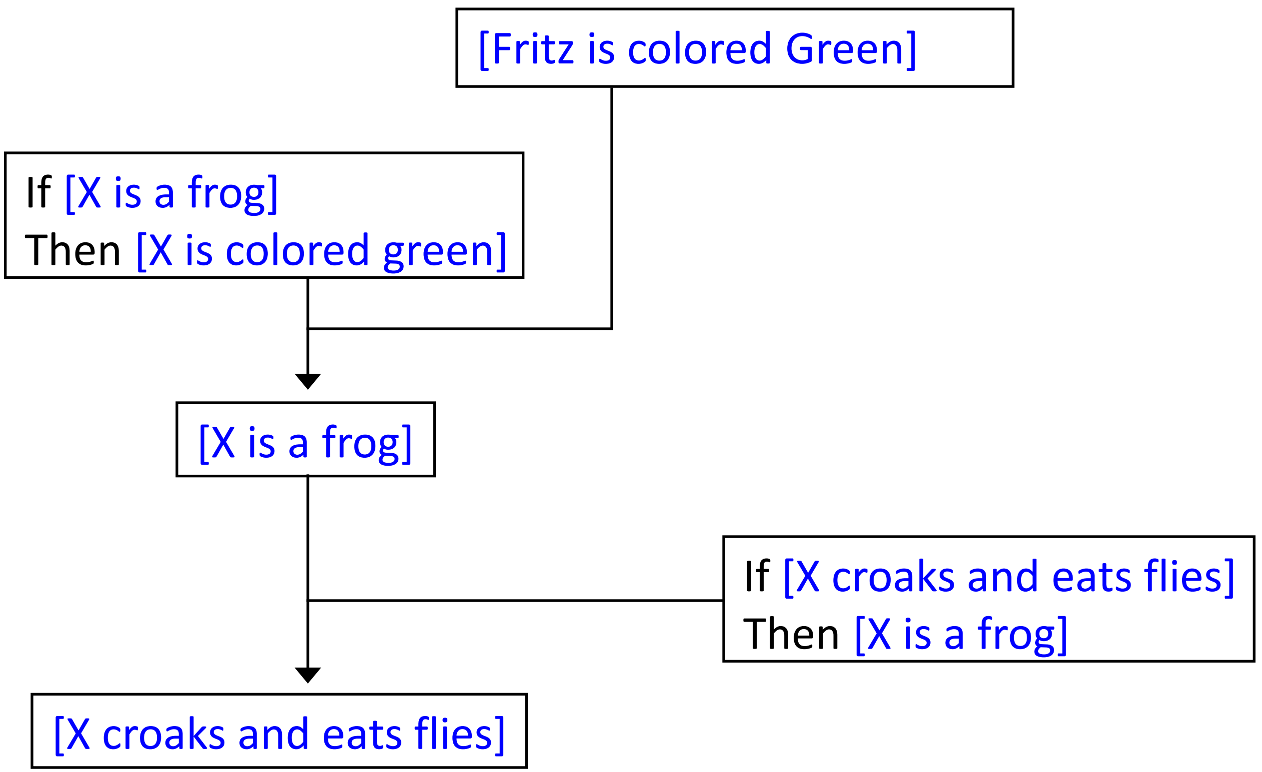
[Fritz croaks and eats flies]

Goals

[Fritz is colored Green]?

[X is a frog]

Backward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

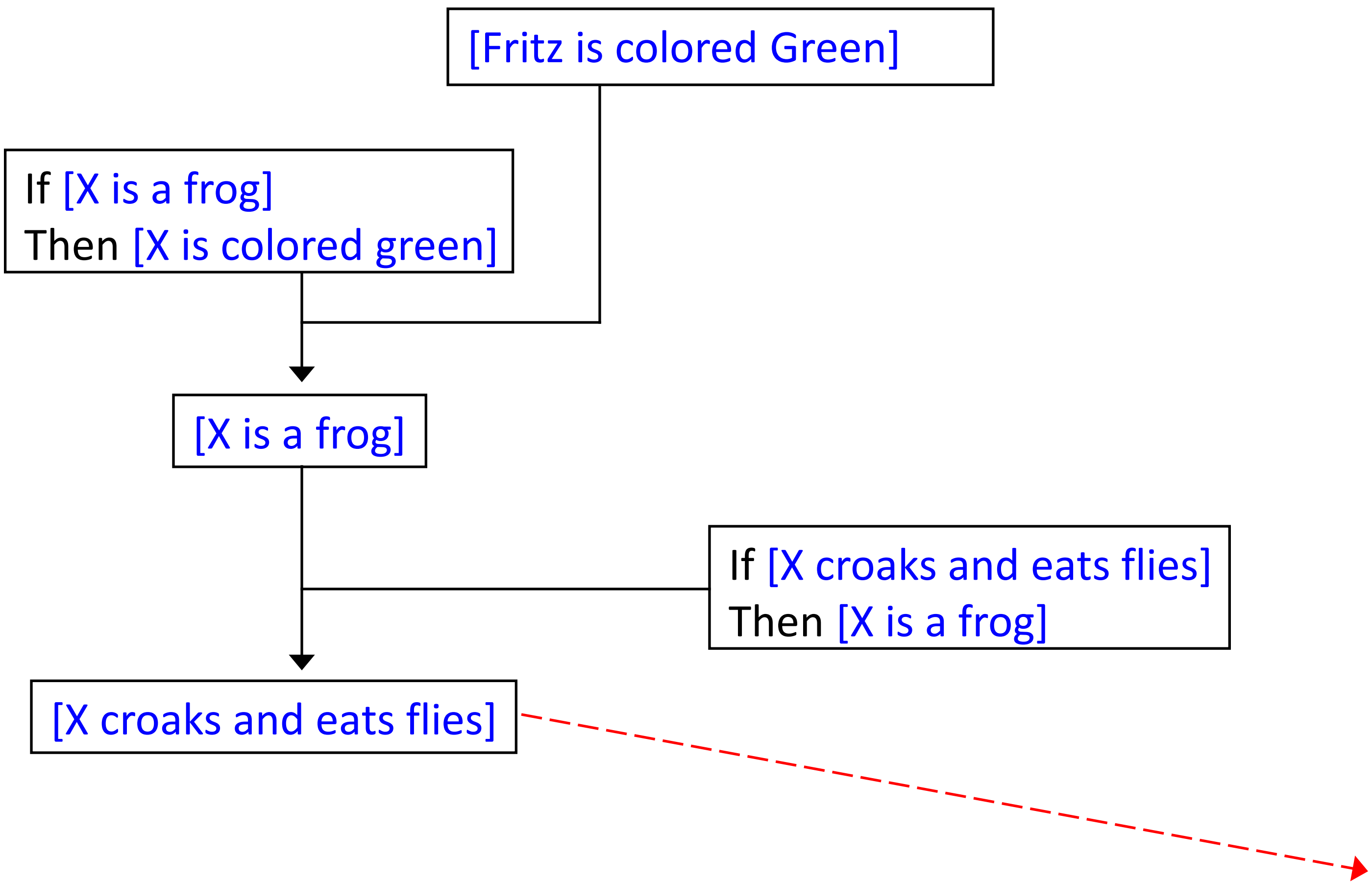
[Fritz croaks and eats flies]

Goals

[Fritz is colored Green]?

[X is a frog]

Backward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

[Fritz croaks and eats flies]

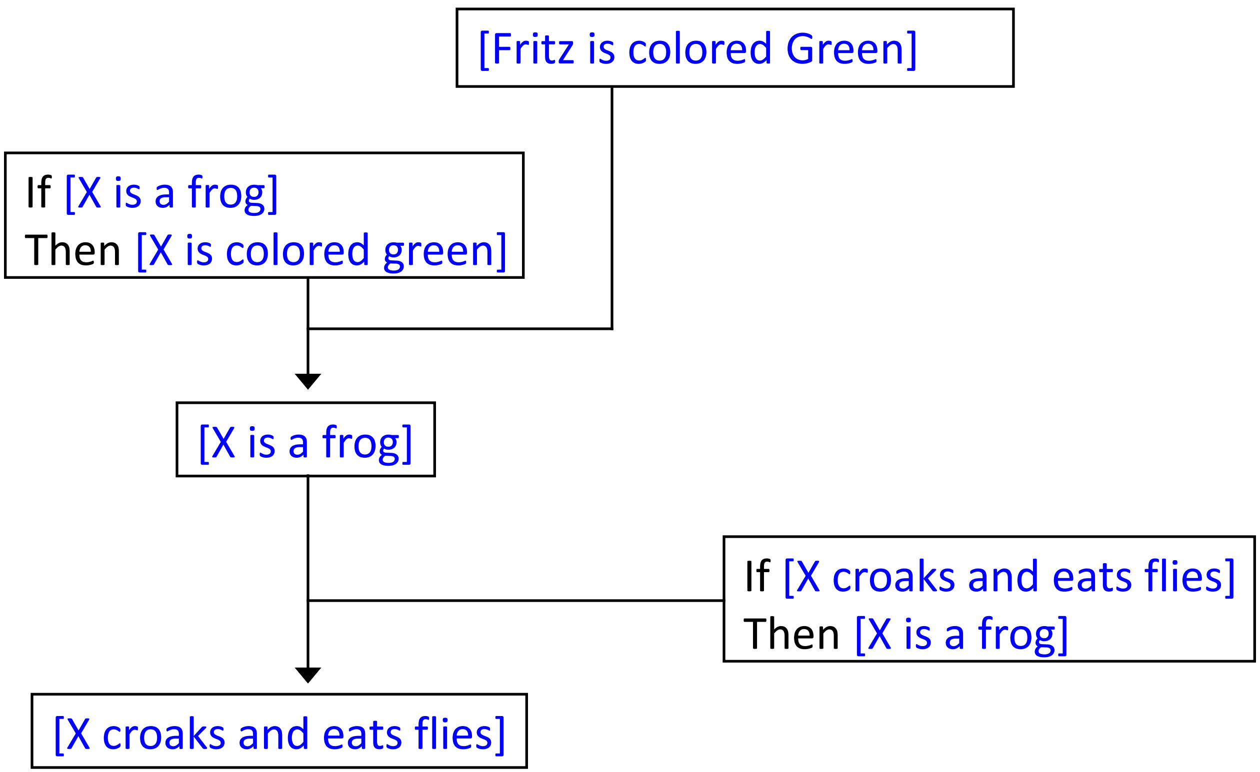
Goals

[Fritz is colored Green]?

[X is a frog]

[X croaks and eats flies]

Backward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

Fritz croaks and eats flies]

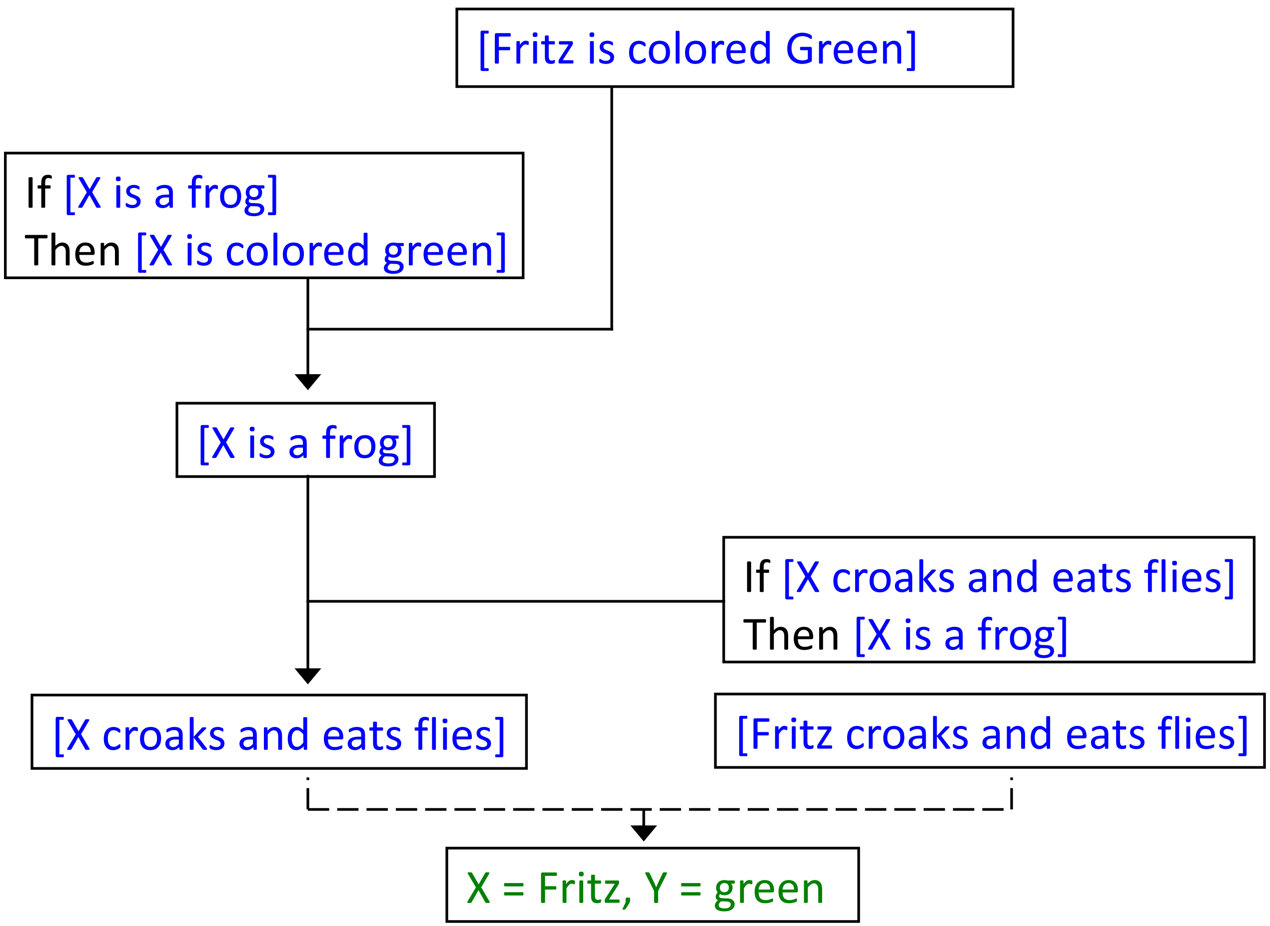
Goals

[Fritz is colored Green]?

[X is a frog]

[X croaks and eats flies]

Backward Chaining Example



Knowledge Base

If [X croaks and eats flies]
Then [X is a frog]

If [X is a frog]
Then [X is colored green]

[Fritz croaks and eats flies]

Goals

[Fritz is colored Green]?

[X is a frog]

[X croaks and eats flies]

Knowledge Base in FOL: *Example 2*

- 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.

1. it is a crime for an American to sell weapons to hostile nations:
   $American(x) \wedge Weapon(y) \wedge Sells(x,y,z) \wedge Hostile(z) \Rightarrow Criminal(x)$

2. Nono … has some missiles, i.e., $\exists x\ Owns(Nono,x) \wedge Missile(x)$:
   $Owns(Nono,M_1) \wedge Missile(M_1)$

3. … all of its missiles were sold to it by Colonel West
   $Missile(x) \wedge Owns(Nono,x) \Rightarrow Sells(West,x,Nono)$

4. Missiles are weapons:
   $Missile(x) \Rightarrow Weapon(x)$

5. An enemy of America counts as "hostile“:
   $Enemy(x,America) \Rightarrow Hostile(x)$

6. West, who is American …
   $American(West)$

7. The country Nono, an enemy of America …
   $Enemy(Nono,America)$

Backward chaining example 2

$\boxed{Criminal(West)}$

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

Owns(Nono,M1) and Missile(M1)

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

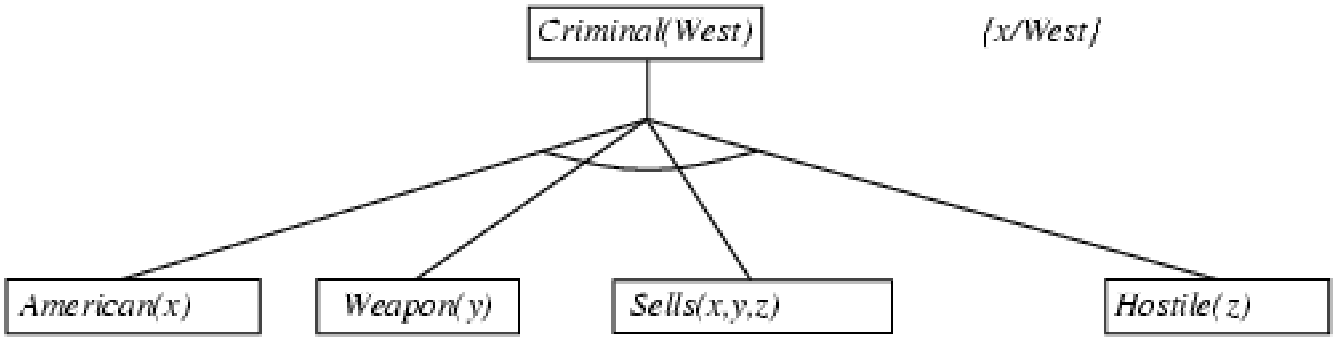
Missile(x) $\Rightarrow$ *Weapon(x)*

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

American(West)

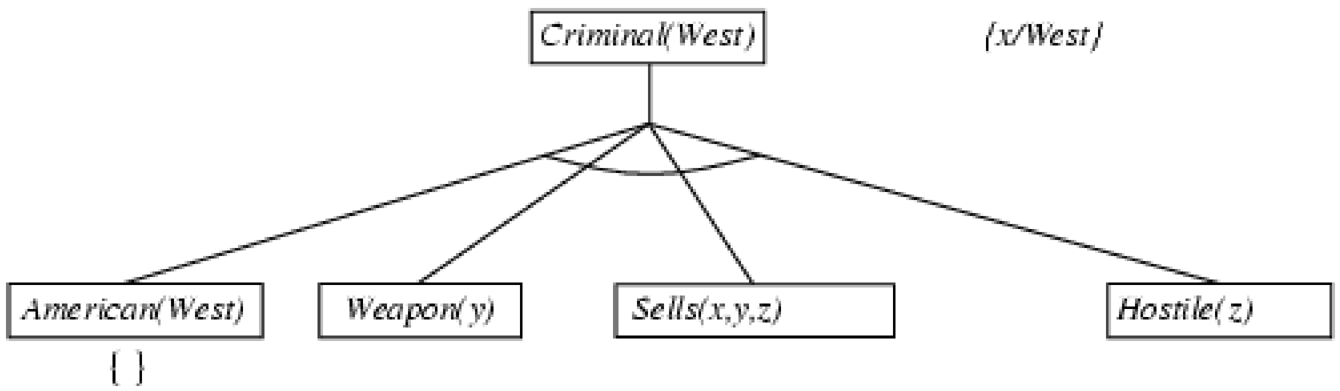
Enemy(Nono,America)

Backward chaining example



American(x) ∧ Weapon(y) ∧ Sells(x,y,z) ∧ Hostile(z) ⇒ Criminal(x)
Owns(Nono,M1) and Missile(M1)
Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)
Missile(x) ⇒ Weapon(x)
Enemy(x,America) ⇒ Hostile(x)
American(West)
Enemy(Nono,America)

Backward chaining example



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

Owns(Nono,M1) and Missile(M1)

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

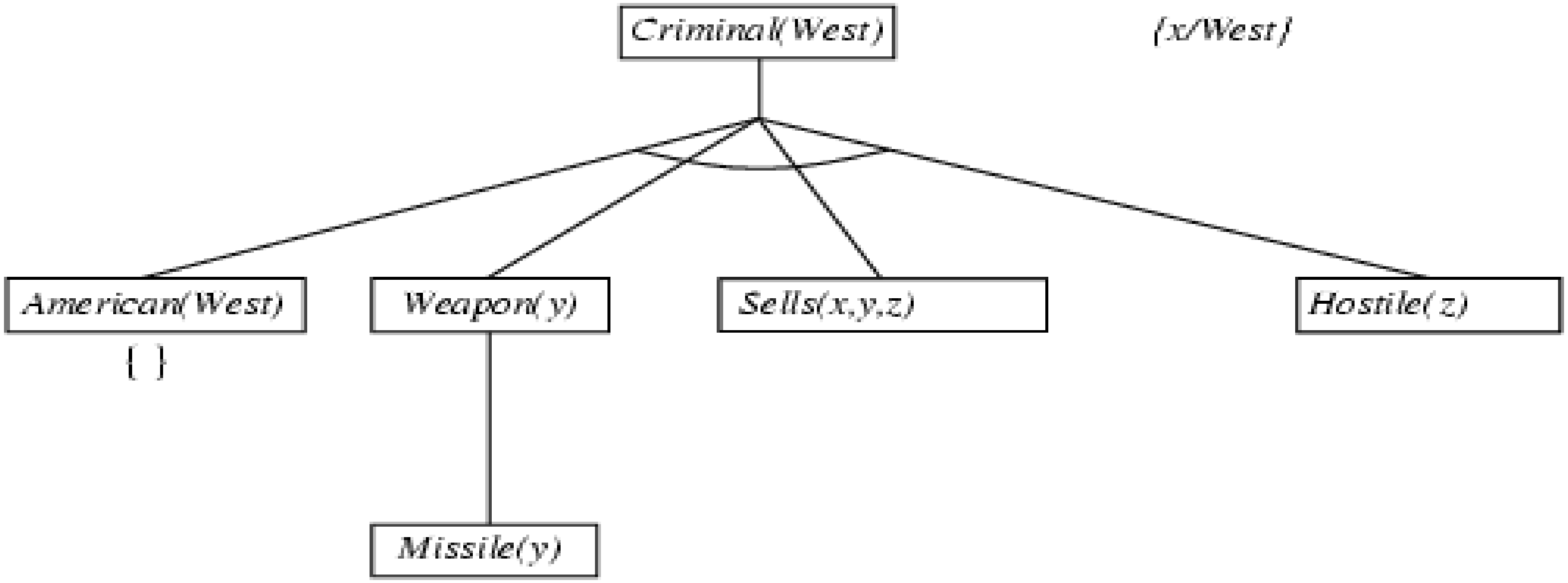
Missile(x) $\Rightarrow$ *Weapon(x)*

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

American(West)

Enemy(Nono,America)

Backward chaining example



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

Owns(Nono,M1) and Missile(M1)

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

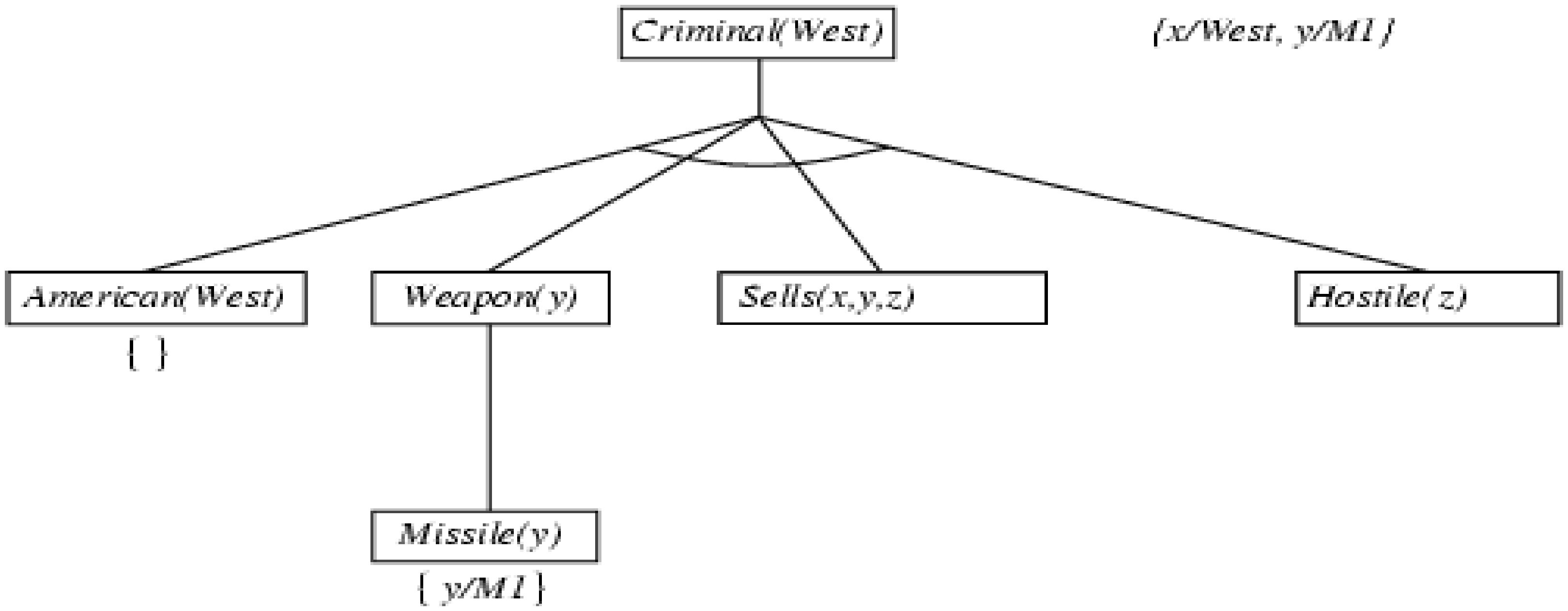
Missile(x) $\Rightarrow$ ***Weapon(x)***

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

American(West)

Enemy(Nono,America)

Backward chaining example



American(x) ∧ Weapon(y) ∧ Sells(x,y,z) ∧ Hostile(z) ⇒ Criminal(x)

Owns(Nono,M1) and ***Missile(M1)***

Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)

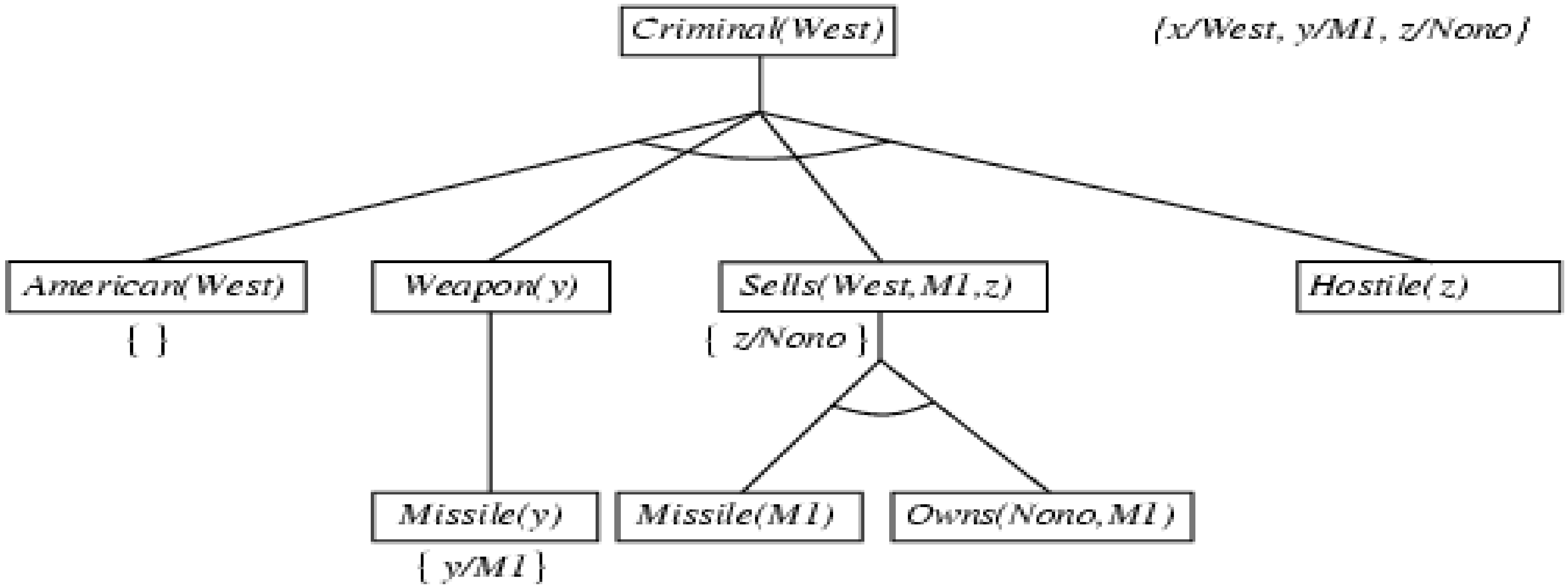
Missile(x) ⇒ Weapon(x)

Enemy(x,America) ⇒ Hostile(x)

American(West)

Enemy(Nono,America)

Backward chaining example



American(x) ∧ Weapon(y) ∧ Sells(x,y,z) ∧ Hostile(z) ⇒ Criminal(x)

Owns(Nono,M1) and ***Missile(M1)***

Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)

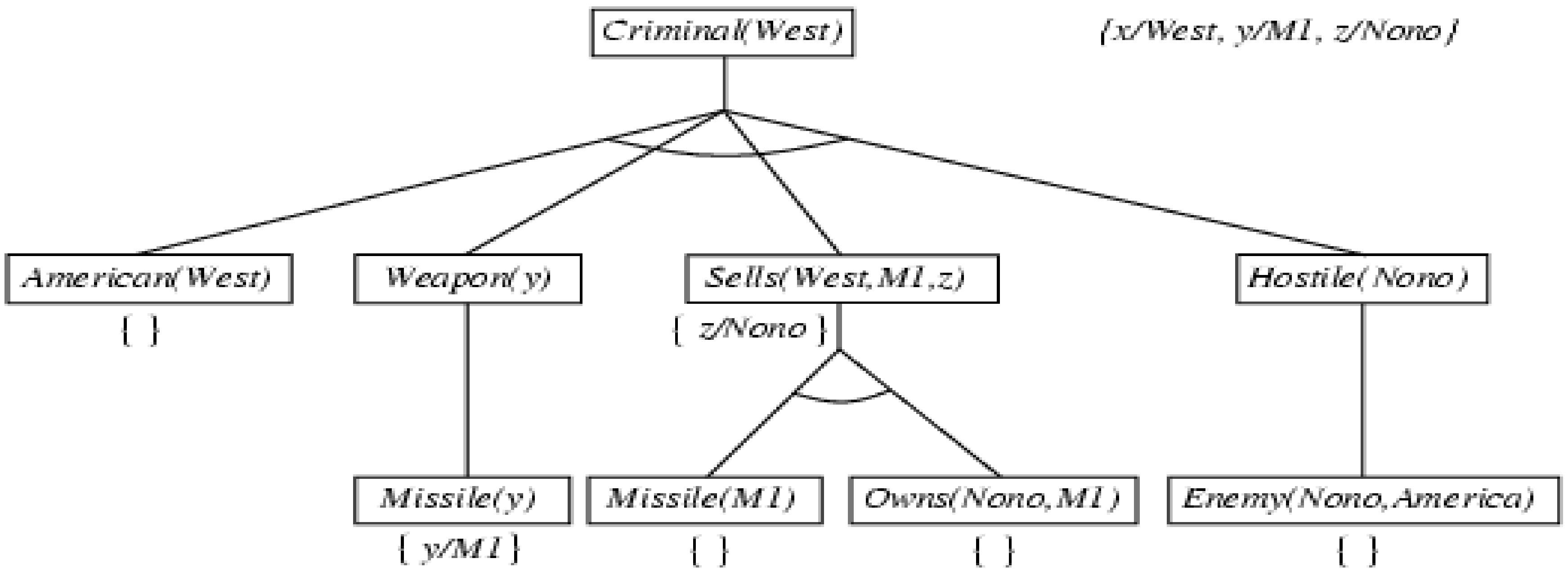
Missile(x) ⇒ Weapon(x)

Enemy(x,America) ⇒ Hostile(x)

American(West)

Enemy(Nono,America)

Backward chaining example



American(x)* ∧ *Weapon(y)* ∧ *Sells(x,y,z)* ∧ *Hostile(z)* ⇒ *Criminal(x)

Owns(Nono,M1) and ***Missile(M1)***

Missile(x)* ∧ *Owns(Nono,x)* ⇒ *Sells(West,x,Nono)

Missile(x)* ⇒ *Weapon(x)

Enemy(x,America)* ⇒ *Hostile(x)

American(West)

Enemy(Nono,America)

Solve using backward Chaining Algorithm

- **Example 3**
  - KB:
    - allergies(X) → sneeze(X)
    - cat(Y) ∧ allergic-to-cats(X) → allergies(X)
    - cat(Felix)
    - allergic-to-cats(Lise)
  - Goal:
    - sneeze(Lise)

Solve using Backward Chaining Algorithm

– **KB:**
  - allergies(X) → sneeze(X) ------1
  - cat(Y) ∧ allergic-to-cats(X) → allergies(X) ------2
  - cat(Felix) ------3
  - allergic-to-cats(Lise)-------4
– **Goal:**
  - sneeze(Lise)

sneeze(Lise) X|Lise

allergies(Lise) X|Lise

cat(Y) ∧ allergic-to-cats(Lise) X|Lise

cat(Felix) allergic-to-cats(Lise)

{}

Application

- Wide use in expert systems
  - Backward chaining: **Diagnosis systems**
    - start with set of hypotheses and try to prove each one, asking additional questions of user when fact is unknown.
  - Forward chaining: **design/configuration systems**
    - see what can be done with available components.

Thank you