Manara University
Faculty of Engineering
Department of IT

Introduction to Artificial Intelligence

Knowledge representation

Predicate Logic or First-order logic (FOL)

تمثيل المعرفة – منطق المسندات(منطق الدرجة الأولى)

Lecture 6

Prof Dr. Eng. Mariam M. Saii

outline

- Why FOL?
- Syntax and semantics of FOL
- Fun with sentences

First Order Logic

- Whereas propositional logic assumes world contains <u>facts</u>.
- **First-order Logic** (like natural language) assumes the world contains
  - ➢ **Objects**: people, houses, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries …
  - ➢ **Relations**: red, round, bogus, prime, multistoried …, brother of, bigger than, inside, part of, has color, occurred after, owns, comes between.
  - ➢ **Functions**: father of, best friend, third inning of, one more than, end of.

- بينما يفترض المنطق التقريري أن العالم يحتوي على حقائق.
- يفترض المنطق من الدرجة الأولى (مثل اللغة الطبيعية) أن العالم يحتوي على
  - أشياء: أشخاص، منازل، أرقام، نظريات، رونالد ماكدونالد، ألوان، مباريات بيسبول، حروب، قرون...
  - علاقات: أحمر، مستدير، زائف، أولي، متعدد الطوابق... أخ، أكبر من، داخل، جزء من، له لون، حدث بعد، يملك، يأتي بين.
  - وظائف: والد، أفضل صديق، الشوط الثالث من، واحد أكثر من، نهاية

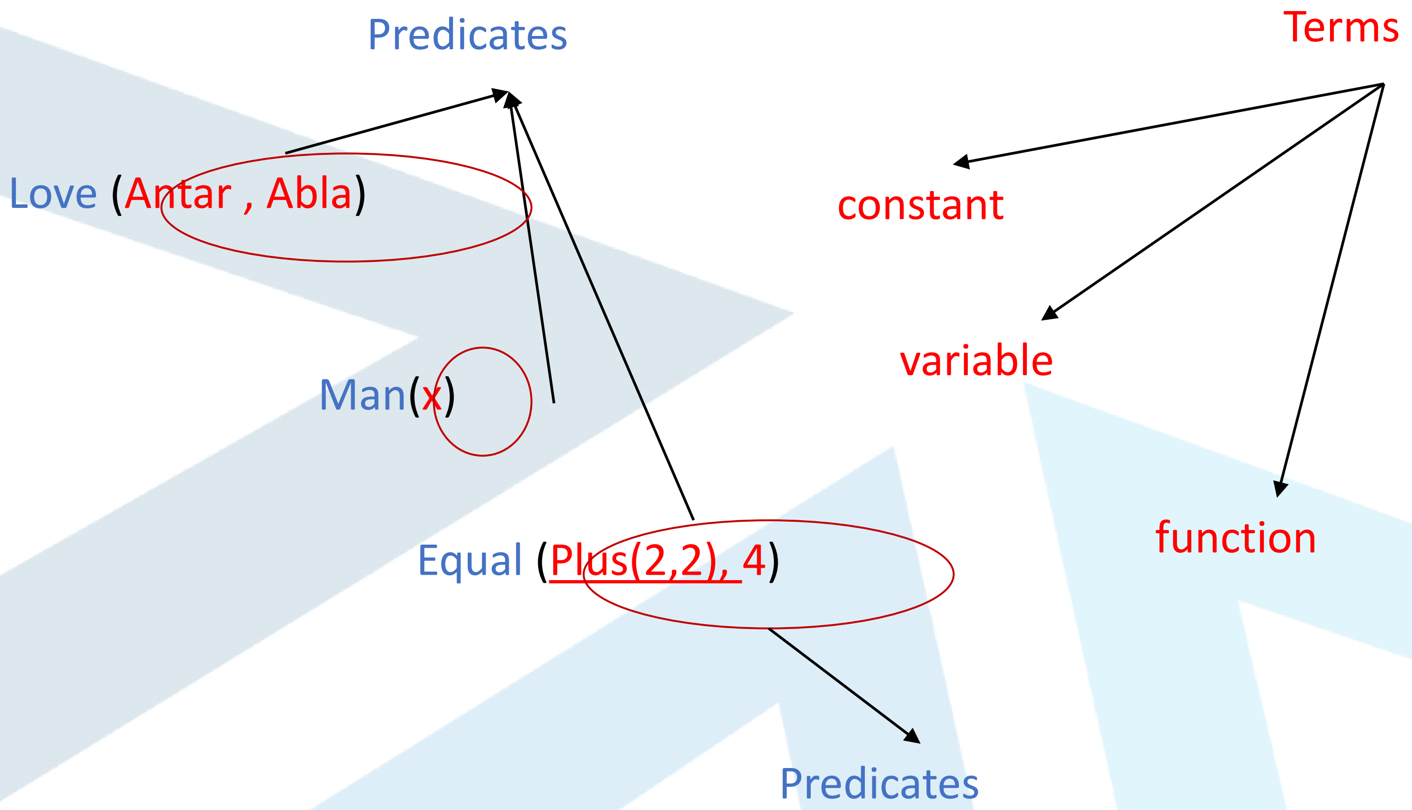
Examples

- “One plus two equals three”
  - Objects: one, two, three
  - Relation equals
  - Function: plus

Syntax of FOL

Constants	$KingJohn,\ 2,\ UCB, \ldots$
Predicates	$Brother,\ >, \ldots$
Functions	$Sqrt,\ LeftLegOf, \ldots$
Variables	$x,\ y,\ a,\ b, \ldots$
Connectives	$\land \quad \lor \quad \neg \quad \Rightarrow \quad \Leftrightarrow$
Equality	$=$
Quantifiers	$\forall\ \exists$

First Order Logic



Syntax of FOL

التركيب اللغوي للتعابير

Sentence	Atomic sentence Sentence Connective Sentence Quantifier Variable……Sentence ¬ Sentence (Sentence)
Atomic Sentence	Predicate(term,…….) Term=term
Term	Function(term,…..) Constant Variable
Connective	⇒ ∧ ∨ ⇔
Quantifier	∀ ∃
Constant	a b c yaser
Variable	A B X
Predicate	Man, part_of
Function	Mother

Syntax of FOL

- All of propositional = quantifiers, predicates, functions, and constants.
- Predicate symbol: indicate to spatial relationship between object
- Sentences present fact this fact formed from

  (predicate symbol, quantifier, terms)
- Variables can take on values of constants or terms.
  - Term = reference to object
- Variables not allowed to be predicates.
  - that's 2nd order logic
- Notation: variables are lower case, etc
- A term with no variables is a ground term.

Atomic sentences

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

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

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

Complex sentences

Complex sentences are made from atomic sentences using connectives

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

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

Goldbach's Conjecture

- **For all** n, if integer(n), even(n), greater(n,2) then there exists p1, p2, integer(p1), integer(p2), prime(p1),prime(p2), and equals(n,sum(p1,p2)).
- **Quantifiers**: for all, there exists
- **Predicates**: integer, greater, prime, even, equals.
- **Constants**: 2
- **Functions**: sum.

Universal quantification

$\forall \langle variables \rangle \ \langle sentence \rangle$

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

$\forall x \ \ P$ 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

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

A common mistake to avoid

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

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

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

means "Everyone is at Berkeley and everyone is smart"

Existential quantification

Existential quantification

$\exists \langle variables \rangle \ \langle sentence \rangle$

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

$\exists x \ \ P$ 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

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

Another common mistake to avoid

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

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

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

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

Properties of quantifiers

$\forall x\ \forall y$ is the same as $\forall y\ \forall x$ (why??)

$\exists x\ \exists y$ is the same as $\exists y\ \exists x$ (why??)

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

"Brothers are siblings"

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

$\exists x\ \forall y\ Loves(x,y)$

"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

$\forall x\ Likes(x, IceCream) \qquad \neg\exists x\ \neg Likes(x, IceCream)$

$\exists x\ Likes(x, Broccoli) \qquad \neg\forall x\ \neg Likes(x, Broccoli)$

example

- All kings are persons.
  - x, King(x) $\forall$ · Person(x). OK.
  - x, King(x) $\forall$ Person(x). Not OK.
  - this says every object is a king and a person.
- Everyone Likes icecream.
  - x, Likes(x, $\forall$ ecream).

Negating Quantifiers

$\forall X, \sim P(X)$

$\sim \exists X, P(X)$

$\forall x \ \neg Likes(x, Parsnips)$ is equivalent to $\neg \exists x \ Likes(x, Parsnips)$.

$\forall X, P(X)$

$\forall x \ Likes(x, IceCream)$ is equivalent to $\neg \exists x \ \neg Likes(x, IceCream)$.

$\sim\exists X, \sim P(X)$

Everyone Likes icecream

$\forall X, Likes(X, Icecream)$

No one don't like icecream

$\sim\exists X, \sim Likes(X, Icecream)$

No one likes liver

$\forall X, \sim Like(X, Liver)$

$\sim\exists X, Like(X, Liver$

More Translations

- Everyone loves someone.
- There is someone that everyone loves.
- Everyone loves their father.

- For all x, there is a y such that Loves(x,y).
- There is an M such that for all y, Loves(x,M).
- M is skolem constant
- For all x, Loves(x,Father(x)).
- Father(x) is skolem function.

More Translations

- All cats are mammals
- There are a main cat
- All cats like fish
- Cats eat everything they like
- Momo is a cat

$\forall$ x, cat (x) ⟹ mammal(x)

$\exists$ x, cat (x) ^ main(x).

$\forall$ x, cat (x) ⟹ like(x,Fish)

$\forall$ x $\forall$ y, cat (x)^like(x, y) ⟹ eat(x,y)

$\forall$ Momo(x) ⟹ cat(x).

FOL Inference

مقدمات :

- if X is a parent of Y, then X older then Y
- If X is mother of Y, then X is a parent of Y
- Fatima is mother of Samira

نتائج :

- Fatima is older then Samira

الترجمة إلى FOL:

- $\forall X\ \forall Y$ parent(X,Y) $\Rightarrow$ older(X,Y)
- $\forall X\ \forall Y$ mother(X,Y) $\Rightarrow$ parent(X,Y)
- Mother(fatima, samira)

→ older(fatima, samira)

نحصل على الاستدلال انطلاقا من الحقائق و الوقائع و بعمليات التحويل اللغوي

example

Suppose we have the following facts

- All cats like fish
- Cats eat everything they like
- And Momo is a cat

Prove that Momo eats fish, using FOL, and backward and forward chining

- $\forall$ x, cat (x) ⟹ like(x, Fish)
- $\forall$ x $\forall$ y, cat (x)^like(x, y) ⟹ eat(x, y)
- Momo(x) ⟹cat(x).
- Prove eat(Mamo, Fish)

solution

1. ∀ x, cat (x) ⇒ Like(x, Fish)
2. ∀ x ∀ y, cat (x)^Like(x, y) ⇒ Eat(x, y)
3. Momo(x) ⇒ Cat(x).
4. Prove eat(Mamo, Fish)

Combine 1,3 with resolution using substitution X=Momo

4. like(Momo, Fish)

- Combine 2,3 with resolution using substitution X=Momo

5. Eat(Momo, y)

- Combine 4,5 with resolution using substitution Fish Eat(Momo, Fish)

Example I

Suppose we have the following facts.

- It is crime for an American to sell weapons to hostile nations.
  تعد بيع الأسلحة من الامريكي لأمم معادية جريمة
- Nono has some missiles. يمتلك نونو بعض الصواريخ
- All its missiles were sold to it by colonel West.
  جميع الصواريخ بيعت له من قبل مستعمر غربي
- Missiles are weapons. الصواريخ هي أسلحة
- An enemy of America count as hostile. اي عدو لامريكا هو معادي
- West is American الغربي امريكي
- The Nono's country is an enemy of America. بلد نونو عدو لامريكا

Prove that **west is a criminal**, using FOL, and backward and forward chining

Example I

- It is crime for an American to sell weapons to hostile nations.

  x,y,z, An∀ican (x) ^ Weapons (y) ^ Hostile (z) ^ Sell(x,y,z) => Criminal(x).
- Nono has some missiles. Owns (NONO, M). ^ Missile(M)
- All its missiles were sold to it by colonel West.

  Missile(x) ^ Owns (NONO, x) => Sell(West, x,NONO)
- Missiles are weapons. Missile(x) => Weapons (x)
- An enemy of America count as hostile Enemy (x, America) => Hostile(x)
- West is American American (West)
- The Nono's country is an enemy of America Enemy (NONO, America)

prove west is a criminal Criminal (West)

Example I

- ∀ x,y,z, American (x) ^ Weapons (y) ^ Hostile (z) ^ Sell(x,y,z) => Criminal(x).
- Owns (NONO, M). ^ Missile(M)
- Missile(x) ^ Owns (NONO, x) => Sell(West, x,NONO)
- Missile(x) => Weapons (x)
- Enemy (x, America) => Hostile(x)
- American (West)
- Enemy (NONO, America)

Example I

Prove: West is a criminal
Forward Chaining

- Start with facts and apply rules until no new facts appear. Apply means use substitutions.
- Iteration 1: using facts.
- Missile(M1),American(West), Owns(Nono,M1), Enemy(Nono,America)
- Derive: Hostile(Nono), Weapon(M1), Sells(West,M1,Nono).
- Next Iteration: Criminal(West).
- Forward chaining ok if few facts and rules, but it is undirected.

Example I

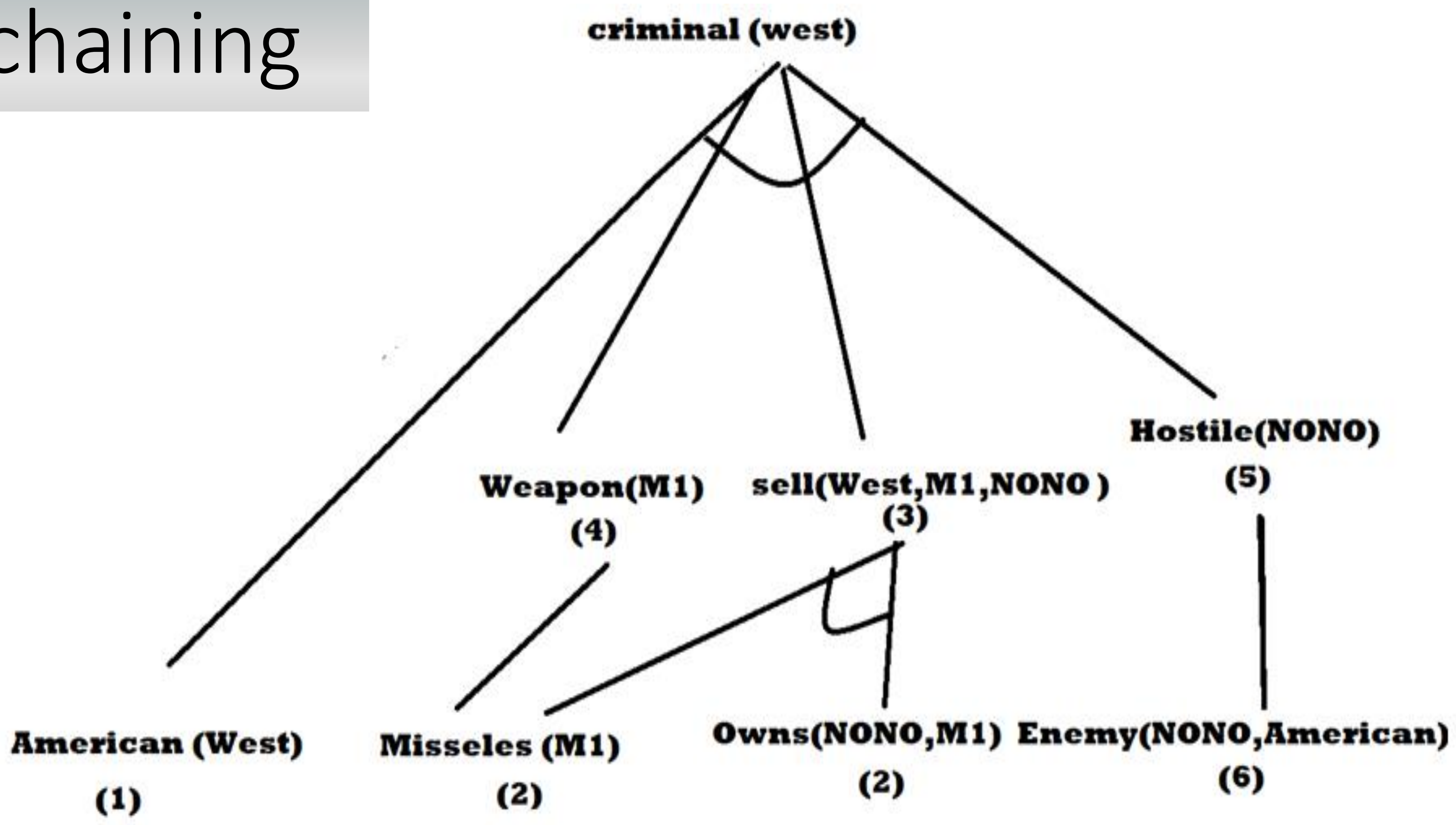
Resolution gives forward chaining

- Enemy(x,America) =>Hostile(x)
- Enemy(Nono,America)
- |- Hostile(Nono)

- Not Enemy(x,America) or Hostile(x)
- Enemy(Nono,America)
- Resolve by {x/Nono}
- To Hostile(Nono)

Example I

forward chaining



Example I

Backward Chaining

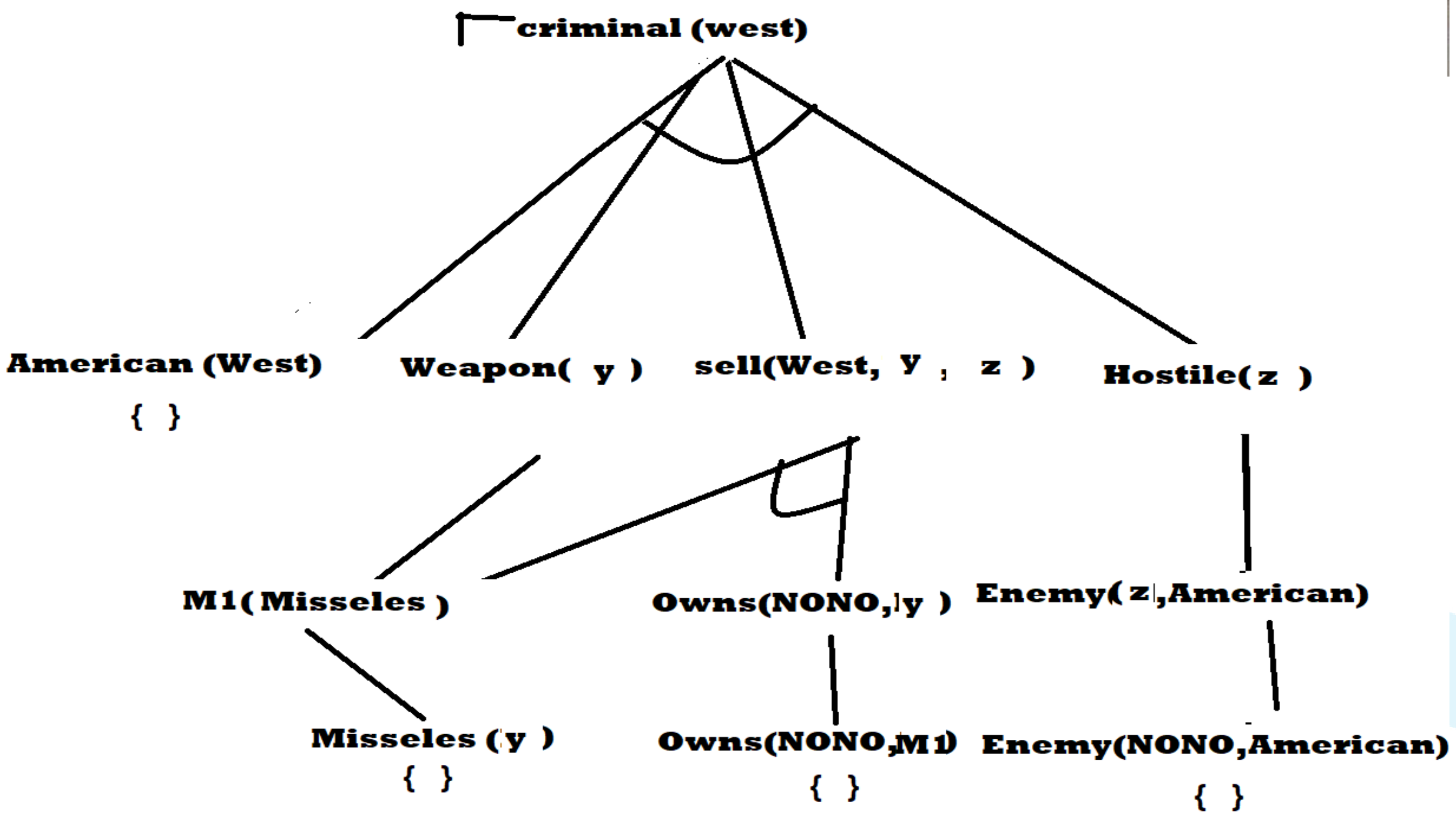
- Start with goal, Criminal(West) and set up subgoals. This ends when all subgoals are validated.
- Iteration 1: subgoals American(x), Weapons(y) and Hostile(z).
- Etc. Eventually all subgoals unify with facts.

Resolution yields Backward Chaining

- A(x) &W(y)&H(z)& S(x,y,z) =>C(x)

- -A(x) or –W(y) or
  –H(z) or –S(x,y,z) or C(x).
- Add goal –C(West).
- Yields –A(West) or
  -W(y) or –H(z) or
  -S(West,y,z).   Etc.

Example I:Backward Chaining



Example I

1. John loves pizza
2. Cheese is a key ingredient of pizza
3. Sause is a key ingredient of pizza
4. Anything with cheese as key ingredient is fatering
5. Anything with Sause as key ingredient is delicious
6. Fatering and delicious food is Junk
7. Any one who loves junk food is unhealthy