ARTIFICIAL INTELLIGENCE

Inference in First Order Logic

MORE SENTENCES

King(Richard) $\lor$ King(John)

- Richard is a king or John is a king

$\neg$Brother(LeftLeg(Rihcard), John)

- Richard’s left leg is not the brother of John

In(Paris,France) $\land$ In(Monaco,France)

- Paris is in france and Monaco is in France

MORE SENTENCES

$\forall c\ Country(c) \land Border(c, Ecuador) \Rightarrow In(c, SouthAmerica)$

- For every object c, if c is a country and c shares its border with Ecuador, then c must be in southamerica.

$\exists c\ Country(c) \land Border(c, Spain) \land Border(c, Italy)$

- There exists an object c which is a country and which shares its border with Spain and Italy

THE REVERSAL

Richard has only two brothers, John and Geoff.

$\boxed{\text{Brother(John, Richard)} \land \text{Brother(Geoff, Richard)}}$

This establishes that John
And Geoff are brothers of Richard,
not that they are the Only two

$$\text{Brother(John, Richard)} \land \text{Brother(Geoff, Richard)} \land (\text{John} \neq \text{Geoffrey}) \land$$

$$\forall x\ \text{Brother}(x,\text{Richard}) \Rightarrow (x=\text{John} \lor x=\text{Geoff})$$

THE REVERSAL

No region in south America borders any region in Europe.

- $\forall c,d\ In(c, SouthAmerica) \land In(d, Europe) \Rightarrow \neg Border(c, d)$

No two adjacent countries have same map color

- $\forall x,y\ Country(x) \land Country(y) \land Border(x, y) \Rightarrow (Color(x) \neq Color(y)) \land (x \neq y)$

DATABASE SEMANTICS: ASSUMPTIONS

Database semantics assume that,

- Every constant symbol refer to a distinct object (unique names assumption)
- Atomic sentences not known to be true are in fact false (closed-world assumption)
- Each model contains no more domain elements than those named by the constant symbols (domain closure assumption)

Under these assumptions the two statements are equal,

- $Brother(John, Richard) \land Brother(Geoff, Richard)$
- $Brother(John, Richard) \land Brother(Geoff, Richard) \land (John \neq Geoffrey) \land$

$$\forall x\ Brother(x,Richard) \Rightarrow (x=John \lor x=Geoff)$$

KB: ASSERTIONS AND QUERIES

TELL(KB, King(John)) , TELL(KB, Person(Richard)) , TELL(KB, $\forall$ x King(x) $\Rightarrow$ Person(x))

Above three statements mentioned above are assertions

ASK(KB, King(John)), ASK(KB, Person(John)), ASK(KB, $\exists$ x Person(x))

Above three statements mentioned above are queries.

ASKVARS(KB, Person(x)) returns all those values of x for which Person(x) is True

KINSHIP

The son of my father is my brother; One's grandmother is the mother of one's parent etc….

Unary Predicate:

- Male / Female

Relations:

- Parent, Sibling, Sister, Child, Daughter, Son, Spouse, etc ……

Functions:

- Mother, Father, etc ……

KINSHIP

One's mother is one's female parent

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

One's husband is one's male spouse

- $\forall$ w,h Husband(h,w) $\Leftrightarrow$ Male(h) $\wedge$ Spouse(h,w)

Male and female are disjoint categories

- $\forall$ x Male(x) $\Leftrightarrow$ $\neg$Female(x)

Parent and child are inverse relations

- $\forall$ p, c Parent(p, c) $\Leftrightarrow$ Child (c, p)

KINSHIP

A grandparent is a parent of one's parent

- $\forall$ g, c Grandparent (g, c) $\Leftrightarrow$ $\exists$p Parent(g, p) $\wedge$ Parent(p, c)

A sibling is another child of one's parents

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

THE WUMPUS WORLD

At 5^{th} time step, Percept ([Stench, Breeze, Glitter , None, None], 5)

Here, Percept is a binary predicate, and Stench and so on are constants placed in a list.

The actions in the wumpus world can be represented by logical terms:

- Turn(Right), Turn(Left), Forward , Shoot , Grab, Climb

To determine which is best, the agent program executes the query

- ASKVARS(∃ a BestAction(a, 5)) ,
- which returns a binding list such as {a/Grab}

ENCODING THE WUMPUS WORLD

$\forall$ t, s, b, m, c Percept ([s, b, Glitter,m, c], t) $\Rightarrow$ Glitter (t) $\longrightarrow$ (raw percept)

$\forall$ t Glitter (t) $\Rightarrow$ BestAction(Grab, t) $\longrightarrow$ (reflex action)

Instead of encoding stuff like

- Adjacent($Square_{1,2}$, $Square_{1,1}$)
- Adjacent($Square_{3,4}$, $Square_{3,3}$)

Encode,

- $\forall$ x, y, a, b Adjacent ([x, y], [a, b]) $\Leftrightarrow$

$$(x = a \wedge (y = b - 1 \vee y = b + 1)) \vee (y = b \wedge (x = a - 1 \vee x = a + 1))$$

INFERENCE IN FOL

Convert an FOL sentence into propositional logic and then all methods of inference that we studied for propositional logic are valid for FOL.

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

$King(John) \wedge Greedy(John) \Rightarrow Evil(John)$

$King(Richard) \wedge Greedy(Richard) \Rightarrow Evil(Richard)$

$King(Father(John)) \wedge Greedy(Father(John)) \Rightarrow Evil(Father(John))$

All of the above statements must be valid for $\forall$ while any single one can be valid for $\exists$.

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

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

: : : 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,

$Owns(Nono, M_1)$ 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:

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

West, who is American : : :

$$American(West)$$

The country Nono, an enemy of America : : :

$$Enemy(Nono, America)$$

FORWARD CHAINING 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. Prove that Col. West is a criminal

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

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

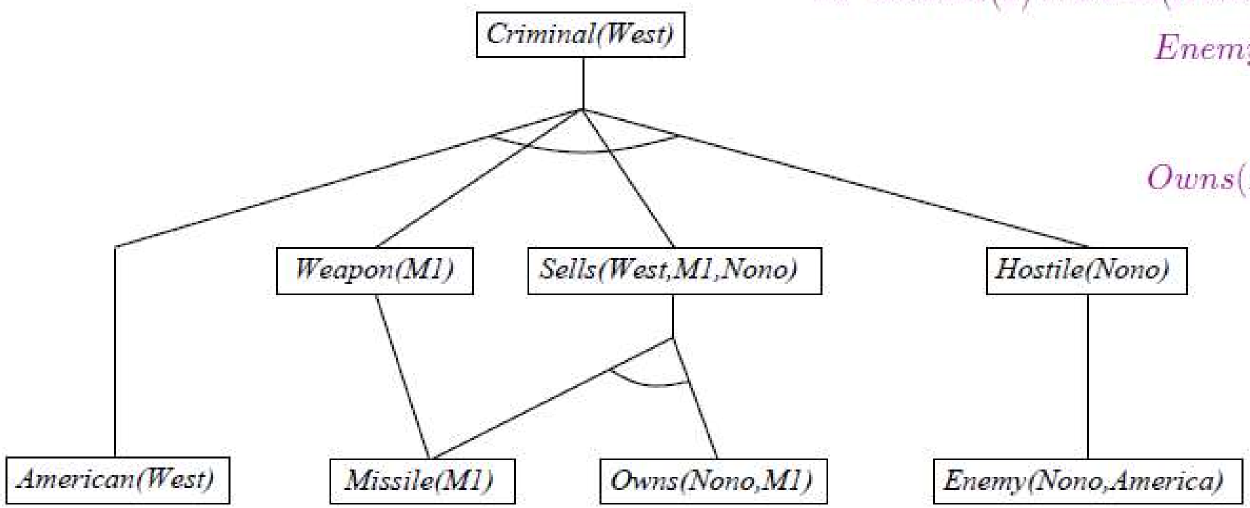
$$Enemy(x, America) \;\Rightarrow\; Hostile(x)$$

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

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

$Enemy(Nono, America)$

$American(West)$



BACKWARD CHAINING 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. Prove that Col. West is a criminal

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

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

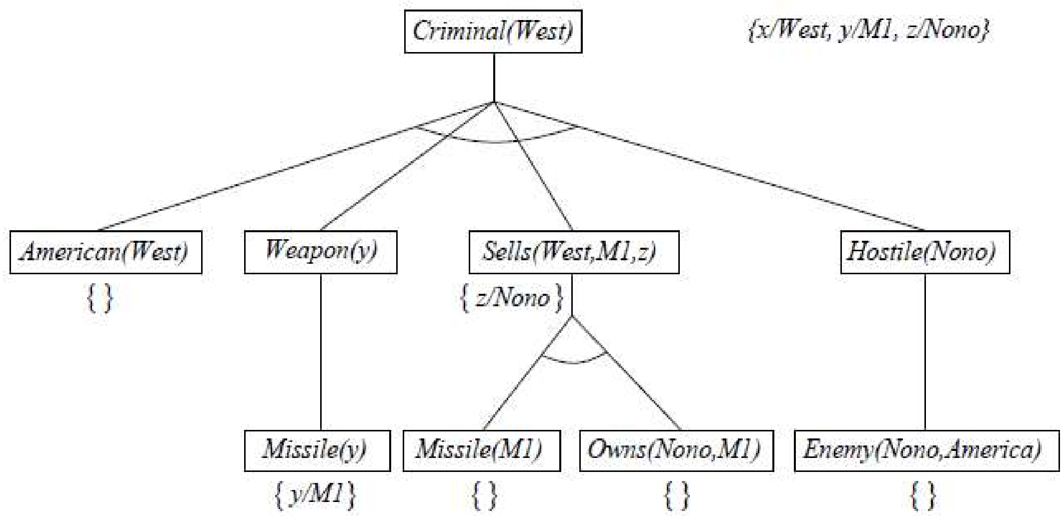
$$Enemy(x, America) \;\Rightarrow\; Hostile(x)$$

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

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

$$Enemy(Nono, America)$$

$$American(West)$$



PROPERTIES OF BACKWARD CHAINING

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

Incomplete due to infinite loops

- x by checking current goal against every goal on stack

Inefficient due to repeated subgoals (both success and failure)

- x using caching of previous results (extra space!)

Widely used (without improvements!) for logic programming

LOGIC PROGRAMMING

Sound bite: computation as inference on logical KBs

	Logic programming	Ordinary programming
1.	Identify problem	Identify problem
2.	Assemble information	Assemble information
3.	Tea break	Figure out solution
4.	Encode information in KB	Program solution
5.	Encode problem instance as facts	Encode problem instance as data
6.	Ask queries	Apply program to data
7.	Find false facts	Debug procedural errors

Should be easier to debug $Capital(NewYork, US)$ than $x := x + 2$!