LOGICAL AGENTS

UI VI

Mária Markošová

Summary from the last lecture

1. MINIMAX algorithm and zero sum games.
2. Alpha beta pruning.
3. Cut off search and evaluation functions.
4. Monte Carlo method and games.
5. Logical agents, intro.

Outline

1. Propositional KB
2. Inference methods in propositional knowledge base
3. KB written in first order logic
4. Inference methods in predicate KB

Soundness and completness

Algorithm ***of inference*** (vyvodzovania) is ***sound*** (korektný) if it infers only such sentences which are entailed in the KB.

Algorithm ***of inference is complet*** if it is able to infer all sentences entailed in the KB.

If our KB has a value True in a real world , then each α ***sentence infered by the sound method of inference is also True in a real world.***

Propositional logic

Real world is described with a help of facts.

Example: Agent is at [1,1]

Proposition: Sentence about which one can decide whether it is true or false.

atomic sentence, atomic formula:

has no logical connection: Agent is at [1,1]

sentence, formula: in general contains at least one

logical connection: Agent is at [1,1] **and** wumpus is at [2,3]

Agent is at [1,1] **or** wumpus is at [2,3]

If Agent is at [1,1] **then** wumpus is at [2,3]

Agent is at [1,1] **if and only if** wumpus is at [2,3]

Syntax of the propositional logic

Syntax

symbols:

-logical constants *True* , *False*

-propositional symbols *P*, *Q*, *R*

-connections $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, $\neg$

-brackets ()

unary connection

binary connections

Semantics of the propositional logic

Semantics defines the rules how to find a truth values of the sentences with respect to the given interpretation. Interpretation is a truth values addition to the atomic formulas.

Formula (sentence)-definition

1. Each propositional symbol is a formula (atomic formula)
2. If an expression *A is a* formula then also $\neg A$ is a formula.
3. If expressions *A*, *B* are formulas , then $(A)\vee(B)$, $(A)\wedge(B)$, $(A)\Rightarrow(B)$,$(A)\Leftrightarrow(B)$ are formulas.
4. Nothing else in a formula

Rules for truth values derivation

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
$false$	$false$	$true$	$false$	$false$	$true$	$true$
$false$	$true$	$true$	$false$	$true$	$true$	$false$
$true$	$false$	$false$	$false$	$true$	$false$	$false$
$true$	$true$	$false$	$true$	$true$	$true$	$true$

Modus ponens

Logical equivalence

- Two sentences are logically equivalent iff true in same models: α ≡ ß iff α ╞ β and β ╞ α

$$
\begin{aligned}
(\alpha \wedge \beta) &\equiv (\beta \wedge \alpha) && \text{commutativity of } \wedge \\
(\alpha \vee \beta) &\equiv (\beta \vee \alpha) && \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma)) && \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) &\equiv (\alpha \vee (\beta \vee \gamma)) && \text{associativity of } \vee \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \vee \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) &\equiv (\neg\alpha \vee \neg\beta) && \text{de Morgan} \\
\neg(\alpha \vee \beta) &\equiv (\neg\alpha \wedge \neg\beta) && \text{de Morgan} \\
(\alpha \wedge (\beta \vee \gamma)) &\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) && \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) &\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) && \text{distributivity of } \vee \text{ over } \wedge
\end{aligned}
$$

Inference rules of the propositional logic

Modus ponens : If formulas A and $A \Rightarrow B$ are true, then B is true p

$$\frac{A, A \Rightarrow B}{B} \qquad \frac{(A) \wedge (A \Rightarrow B)}{B}$$

Modus tollens :

$$\frac{\neg B, A \Rightarrow B}{\neg A}$$

Simplification: If conjuction is true, then all conjucts are true

$$\frac{A_1 \wedge A_2 \wedge \ldots \wedge A_n}{A_i}$$

Conjunction introduction : If $A_1, A_2, \ldots., A_n$ are true, then their conjunction is true .

$$\frac{A_1, A_2, \ldots., A_n}{A_1 \wedge A_2 \wedge \ldots \wedge A_n}$$

Adition: If formula is true, then its disjunction with all formulas is true.

$$\frac{A_i}{A_1 \vee A_2 \vee A_3 .. \vee A_i \ldots \vee A_n}$$

Double negation elimination: If some formula is negated twice one gets the same formula.

$$\frac{\neg\neg A}{A}$$

Unit resolution:

$$\frac{A \vee B, \neg B}{A}, \quad \frac{(A \vee B) \wedge (\neg B)}{A}$$

Jednotková rezolvencia (unit resolution)

General unit resolution:

$$\frac{l_1 \vee l_2 \vee \ldots . l_i \ldots . \vee l_k , \ \neg l_i}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k}$$

m_j is a negation of l_i

Full resolution

$$\frac{l_1 \vee l_2 \vee \ldots . \vee l_k , \ m_1 \vee \ldots \vee m_n}{l_1 \vee \ldots \vee l_{i-1} \vee l_{i+1} \vee \ldots \vee l_k \vee m_1 \vee \ldots \vee m_{j-1} \vee m_{j+1} \vee \ldots \vee m_n}$$

Inference in the propositional logic – exam example

1	$\neg P \wedge Q$	*assumption* 1
2	$R \Rightarrow P$	*assumption* 2
3	$\neg R \Rightarrow S$	*assumtion* 3
4	$S \Rightarrow T$	*assumption* 4
5	$\neg P$	assumption 1 simplification
6	Q	assumption 1 simplification
7	$\neg R$	result 5 and modus tollens on the assumption 2
8	S	result 7 and modus ponens on the asumption 3
9	T	result 8 and modus ponens on assumption 4

Let us suppose this is our KB at the moment and we want to infer what is possible using the rules. All sentences in KB are true.

Normal forms

1. Resolution rules are applicable on the conjunction of disjunctions (normal forms).
2. If KB has not such structure, what to do?

↓

Each sentence in the propositional logic is logically equivalent to the conjunction of disjunctions : CNF (conjunctive normal form)

CNF algorithm

- input : formula a , output CNF of the formula a
- algorithm:

1. Subformulas $X \Leftrightarrow Y$ are replaced by $(X \Rightarrow Y) \wedge (Y \Rightarrow X)$
2. $X \Rightarrow Y$ are replaced by $\neg X \vee Y$
3. Rewrite

$$\neg\neg X \longrightarrow X$$
$$\neg(X \vee Y) \longrightarrow \neg X \wedge \neg Y$$
$$\neg(X \wedge Y) \longrightarrow \neg X \vee \neg Y$$

4. Distribution rules for $\vee$ and $\wedge$:

$$X \vee (Y \wedge Z) \longrightarrow (X \vee Y) \wedge (X \vee Z)$$
$$(X \wedge Y) \vee Z \longrightarrow (X \vee Z) \wedge (Y \vee Z)$$

5. Replace redundancies such as $X \vee X$ is replaced by X.

Clause

Literal: formula or its negation, here are three literals

$$\neg P_{2,1},\ B_{1,1}, P_{1,2}, \ldots$$

Clause; disjunction of literals, here are three clauses:

$$\left(\neg P_{2,1} \vee B_{1,1}\right), \left(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}\right), \left(\neg P_{1,2} \vee B_{1,1}\right)$$

CNF: conjunction of clauses

$$\left(\neg P_{2,1} \vee B_{1,1}\right) \wedge \left(\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}\right) \wedge \left(\neg P_{1,2} \vee B_{1,1}\right) \wedge \neg B_{1,1} \wedge P_{1,2}$$

Resolution algorithm

We want to prove that $KB \models \alpha$. We shal prove it with a help of contradiction by showing that $KB \wedge (\neg \alpha)$ cannot be satisfied.

a) Convert $KB \wedge \neg \alpha$ to the CNF form .

b) Do resolution and add new sentences into KB.

c) Process finishes if: - there are no new expressions we can add, then

$KB \not\models \alpha$ sentence is not entailed in KB

- we get an empty clause, $KB \models \alpha$ then sentence is entailed in KB

Example

We want to prove, that there is no pit at [1,2] position $\alpha = \neg P_{1,2}$

- $KB = (B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})) \land \neg B_{1,1}$ $\alpha = \neg P_{1,2}$

1. Ad a negation : $(B_{1,1} \Leftrightarrow (P_{1,2} \lor P_{2,1})) \land \neg B_{1,1} \land P_{1,2}$ ← $\neg\alpha$
2. Rewrite 1. to the CNF form:

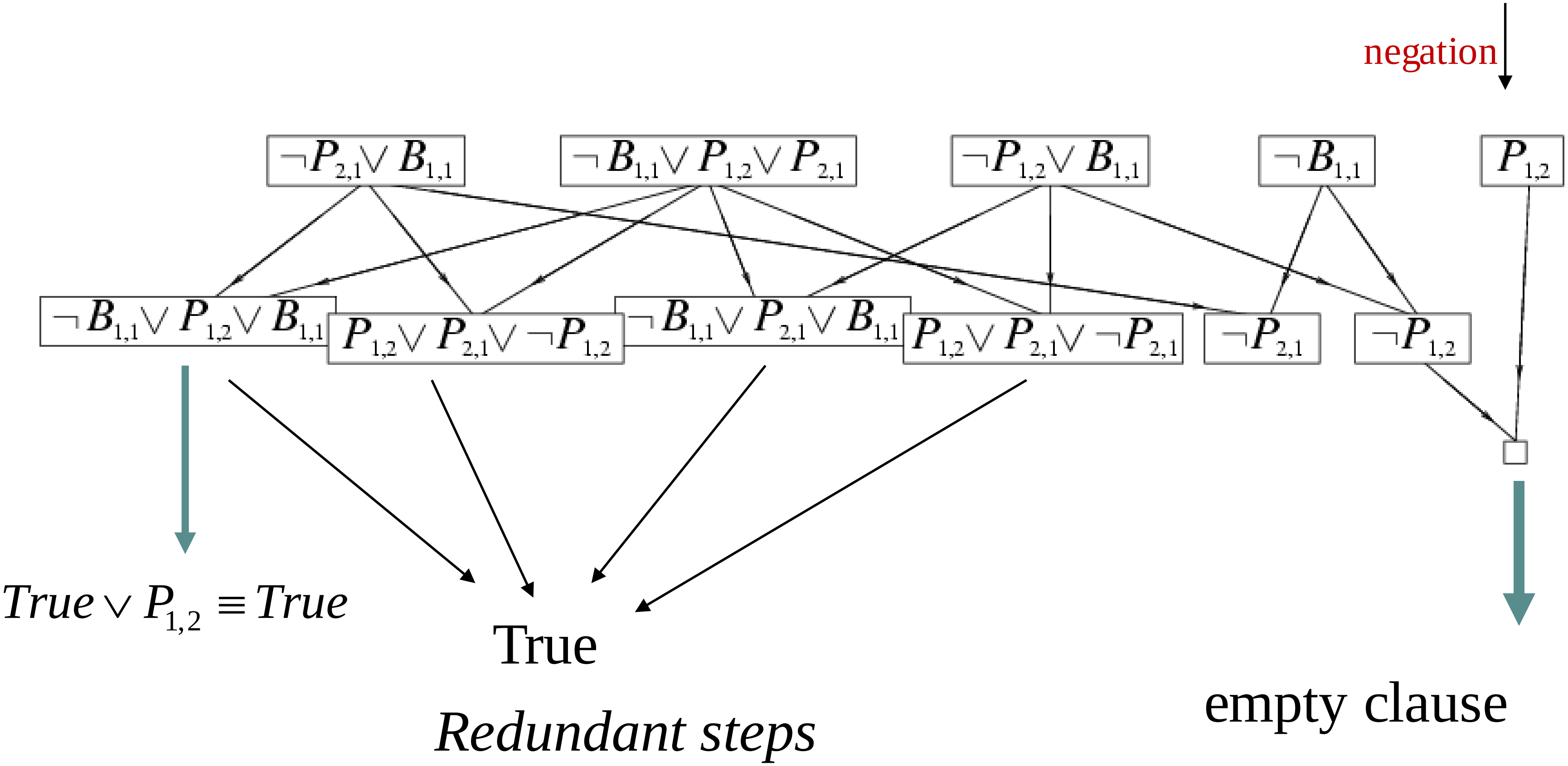
$$\left(\neg P_{2,1} \lor B_{1,1}\right) \land \left(\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}\right) \land \left(\neg P_{1,2} \lor B_{1,1}\right) \land \neg B_{1,1} \land P_{1,2}$$

3. Use the resolution process.

Resolution process example

- $KB = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$ $\alpha = \neg P_{1,2}$

$$\text{CNFKB} = (\neg P_{2,1} \vee B_{1,1}) \wedge (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge \neg B_{1,1} \wedge P_{1,2}$$



Horn clauses

Horn clause: disjunction of literals where at most one of them is positive.

Example: $\left(\neg L_{1,1} \vee \neg Breeze \vee B_{1,1}\right)$

Agent does not feel breeze

Agent is not localized at the position [1,1].

Agent in the wumpus world is not at the position [1,1] or agent does not feel breeze or breeze is at the position [1,1].

Why Horn clauses?

1. Each Horn clause can be written as an implication. At the assumption side is a conjunction of initially negative literals and at the conclusion side is a positive literal.

Príklad:

$$\left(\neg L_{1,1} \vee \neg Breeze \vee B_{1,1}\right) \equiv \left(\left(L_{1,1} \wedge Breeze\right) \Rightarrow B_{1,1}\right)$$

$$because \quad \left(\neg L_{1,1} \vee \neg Breeze\right) \equiv \neg\left(L_{1,1} \wedge Breeze\right)$$

$$and \;\; so \quad \neg\left(L_{1,1} \wedge Breeze\right) \vee B_{1,1}$$

$$and \;\; from\, there$$

2. Clause is more readable as an implication.

$$\left(\neg L_{1,1} \vee \neg Breeze \vee B_{1,1}\right)$$

Agent is not at the position [1,1], or agent does not feel breeze, or breeze is at [1,1].

$$\left(\left(L_{1,1} \wedge Breeze\right) \Rightarrow B_{1,1}\right)$$

If agent is at [1,1] and feels breeze then there is breeze at [1,1].

Our knowledges about reality are often in the form of implications

3. Inference in KB containing Horn clauses can be done very effectively, in a linear time with respect to the input:

- **forward chaining algorithm**
- **backward chaining algorithm**

Forward chaining

Can we infer Q from our KB? Is Q true in our KB?

- Idea: fire any rule whose premises are satisfied in the *KB*,
  - add its conclusion to the *KB*, until query is found

KB

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

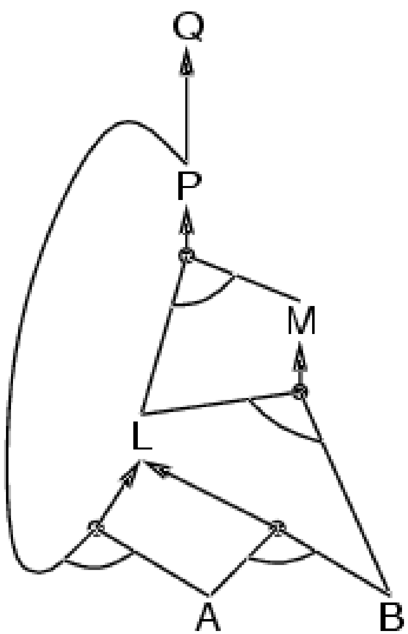
$B \wedge L \Rightarrow M$

$A \wedge P \Rightarrow L$

$A \wedge B \Rightarrow L$

A

B L M P Q



Modus ponens

Modus ponens

Modus ponens

Linearity of forward chaining (intuitively):

KB: is true

$$A$$

$$A \Rightarrow B$$

$$A \wedge B \wedge Z \Rightarrow G$$

$$A \wedge B \Rightarrow Z$$

$$A \wedge G \Rightarrow Q$$

1. ***Start with fact A (atomic formula), it is true.***
2. ***Infer B using modus ponens. Use and elimination. Remove A from all formulas. There is no A in KB any more.***
3. ***Ad inferred B into KB.***
4. ***Repeat from 2. with new facts, atomic formulas (B), remove these facts from all formulas using and elimination rule.***
5. ***From above steps is clear, that algorithm runs KB as many times only, as many facts is in KB.***

Backward chaining: Is hypothesis, that Q is true valid?

1. We have a literal Q. We do not know, whether it is true.
2. We seek such implications in KB, which implies Q.
3. If all premises of this implications are true, then Q is true.

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$

Implication is true, in its assumption is P

Implication is true, in its assumption are L, M and in result is P

Implications are true, in the assumption parts are L, B, A, P and in the result M, L

Implication is true, in the assumption are A, B and in the result is L. A., B are also facts which are true, from this point we start forward chaining and we prove Q .

Forward vs. backward chaining

- FC is data-driven.
- Can infer a lot of facts not connected with Q.
- BC is goal-driven, making only those steps which support goal.
- Complexity of BC can be thus better then that of FC.

Logical agent in the wumpus world

KB of the agent: $\neg P_{1,1}$

$\neg W_{1,1}$

Should be written for all possible concrete values of *x* and *y*.

$$B_{x,y} \Leftrightarrow \left(P_{x,y+1} \vee P_{x,y-1} \vee P_{x+1,y} \vee P_{x-1,y}\right)$$

$$S_{x,y} \Leftrightarrow \left(W_{x,y+1} \vee W_{x,y-1} \vee W_{x+1,y} \vee W_{x-1,y}\right)$$

There is only one wumpus W in the environment

There at least one wumpus W in the environment.

$$W_{1,1} \vee W_{1,2} \vee W_{1,3} \ldots\ldots \vee W_{4,3} \vee W_{4,4}$$

For each two squares at least one is without W

$$W_{k,l} \vee W_{l,m}$$

155 such sentences

First-order logic, FOL

- In the propositional logic the world is described by the facts
- In the FOL (as in the natural language) the world is supposed to consist of
  - Objects: people, houses, numbers, colours, …
  - Relations, properties: red, round, brother of, greater then, part of …
  - Functions: father of somebody, the best friend, one unit greater then, plus, …

  - FOL has better expressivity then propositional logic, is closer to the natural language

Syntax

1. ***Individuum variables***: x, y, z..... denotes unspecified objects, they can have value from the certain domain
2. ***Predicate symbols***: P, Q, R,.... denotes properties of the objects and relations between objects
3. ***Constants*** a,b,c....they play the role of the name of the individuals, specify them : Peter, Jan, 2,3,5
4. ***Functional symbols*** f,g,h, also can name, specify the individuum: $\sin(\pi)$, head(x)
5. ***Logical symbols*** $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, $\forall$, $\exists$

Predicate:

n – ary relation expressing the property of the object or relation among the objects. Adds the truth value to the object.

- ***unary predicate*** - describes the property
- ***binary, ternary, ... n-ary predicate*** – expresses relations among objects.

Arity of predicates

Arity is changing by the concretization, giving valuo to the variables.

$V(x,y)$	$D^2 \to \{0,1\}$	**x is taller then y,** arity 2
$V(Jozo,y)$	$D^1 \to \{0,1\}$	**Jozo is taller then y**, arity 1
$V(Jozo,Fero)$	$D^0 \to \{0,1\}$	**Jozo is taller then Fero**, arity 0

Arity of the functions

$f(t_1, t_2, \ldots, t_n)$ $\quad D^n \rightarrow D$

$f(a, t_2, \ldots, t_n)$ $\quad D^{n-1} \rightarrow D$

$\ldots$

$f(a, b, c, \ldots, k)$ $\quad$ ***constant***

Example:

Number 2 is even. If the positive integer is even, then its successor is odd.

Successor of the number two is odd.

Predicate: ***P(x) ; x is even number ,*** ***Q(x) ; x is odd number***

Function: ***f(x)=x+1 ; returns the successor of x***

Domain: ***positive integers***

Difference between the predicate and the function

Predicate: ***n*-ary predicate *P* is a mapping of**

$$P : D^n \Rightarrow \{0,1\}$$

which each of the *n*- tuple from the domain *D* relates the truth value.

Function: **Function *f* is a mapping of**

$$f : D^n \Rightarrow D$$

which each of the *n*-tuple from the domain *D* relates another object from *D*.

Terms:

More completely defines the name of the individuum

1. Variables and constants are terms.
2. If f is a functional symbol and $t_1, t_2, \dots, t_n$ are terms, then also $f(t_1, t_2, \dots, t_n)$ is a term.
3. Nothing else is a term.

Atomic formula:

1. If P is n-ary predicate symbol and $t_1, t_2, \dots, t_n$ are terms, then the expression $P(t_1, \dots, t_n)$ is an atomic formula.
2. Nothing else is an atomic formula.

Formula:

- Each atomic formula is a formula.
- If φ , ψ are formulas and x is a variable, then the expressions as

  $$\neg\varphi,\ \varphi \wedge \psi,\ \varphi \wedge \psi,\ \varphi \Rightarrow \psi,\ \varphi \Leftrightarrow \psi,\ (\forall x)\varphi,\ (\exists x)\psi$$

  are formulas.
- There are no other formulas, then those created by the above mentioned rules.

Inference in the FOL KB

1. Reduce FOL KB to propositional KB and use the previous methods.
2. Generalized Modus ponens.
3. Generalized forward and backward chaining.
4. Generalized resolution.

Methods:

1. Substitution.
2. Unification.

I. Reduction to the propositional KB

FOL KB is transformed to the propositional KB with a help of substitution.

Substitution:

$Subst(\theta, \alpha)$ in the sentence *alpha* substitution θ is made.

$$\left.\begin{array}{l} \alpha: \ (King(x) \wedge Greedy(x)) \Rightarrow Evil(x) \\ \beta: \ King(John) \end{array}\right\} \text{KB}$$

$$Substitucia \ (\theta, \alpha): x / John$$

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

We need to unify the formulas.

That means to find a substitution, which makes two atomic formulas identical.

$$\alpha: \; (King(x) \wedge Greedy(x)) \Rightarrow Evil(x)$$

$$\beta: \; King(John)$$

$$Substitucia \; (\theta, \alpha): x / John$$

Substitution $\theta : x / John$ unifies King(x) and King(John).

1.universal instantiation

Example

Our KB $\forall x \; King(x) \wedge Greedy(x) \Rightarrow Evil(x)$

$$King(Bob), Greedy(John), \\ Evil(Father(John))$$

facts

$Father(John)$ function

Because the sentence holds for all x, the sentence has **universal quantifier**, we have to make the concretization with all possible facts from the current KB.

$$\forall x \ \ King(x) \wedge Greedy(x) \Rightarrow Evil(x)$$

we infere

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

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

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

$$Evil(Father(John))$$

2. Existential instantiation :

The sentence with existence quantifier tells, that there exists an object fulfilling the condition. Instantiation gives a name to the object. This name cannot be used again.

Aplied only once! After the application, the sentence with the existence quantifier is removed from the KB.

Example

$$\frac{\exists x \;\; \alpha}{Subst(\{x / C_1\}, \alpha)}$$

$$\left.\begin{array}{l} \exists x \;\; Crown(x) \wedge OnHead(x, John), \\ Crown(C_1), Crown(C_2) \end{array}\right\} \text{Our KB}$$

$we \;\; find$

$Crown(C_1) \wedge OnHead(C_1, John)$

C_1 **is a Scolem constant.**

II. Generalized Modus Ponens

Exact formulation- generalized Modus ponens

Atomic sentence

$$\frac{p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q, \left(p_1', p_2', \ldots, p_n'\right)}{Subst\left(\theta, q\right)}$$

Substitution θ ***is such, that***

$$Subst\left(\theta, p_i'\right) = Subst\left(\theta, p_i\right)$$

Example

$$\frac{King(x) \wedge Greedy(x) \Rightarrow Evil(y), (King(John), Greedy(y))}{Subst(\theta, q)}$$

Let: $Substitution \quad \theta \quad is \quad \{x / John, \quad y / John\}$

Then :

$$\frac{King(x) \wedge Greedy(x) \Rightarrow Evil(y), \ (King(John), \ Greedy(y))}{Evil(John)}$$

Infer with a help of substitution θ

$Evil(John)$

More complex example (Russell, Norwig)

Sentences:

"It is illegal for a UK student to sell Coke."
"Bob is an UK student."
"Everyone sells some Coke."

We want to know:

„Is Bob criminal?“

KB:

1. $\forall (x, y)\ Student(x) \wedge Coke(y) \wedge Sells(x,y) => Criminal(x)$

2. $Student(Bob),\ Coke(COKE1),$
3. $\forall x\ \exists y\ Student(x) \wedge Coke(y) \wedge Sells(x,y)$

Inference:

From 3. by general instantanation

4. $\exists y\ Student(Bob) \wedge Coke(y) \wedge Sells(Bob,y)$

From 4. by existence instantanation

5. $Student(Bob) \wedge Coke(COKE1) \wedge Sells(Bob,COKE1)$

From 1. and 5. by Modus ponens

6. $Criminal\ (Bob)$

Exam example

Slytherin is a school.

Griffindor is a school.

Hufflepuf is a school.

If the student studies on the school and has a wand, then he is a whizard.

Students Harry, Ron and Hermiona study on Griffindore.

Ron has a wand named Wanda.

Prove that Ron is a whizard.

Predicates?

School(y), HasWand(x,z), Studies(x,y), Student(x),

Wand(z), Whizard(x)

Facts?

School(Sl), School(G), School (Hp), Student(Ron), Wand(Wanda), HasWand(Ron,Wanda), Studies(Ron,G), Student(Hermiona), Studies(Hermiona, G), Student(Harry), Studies(Harry,G)

Coding the problem?

School(Sl), School(G), School(Hp)

$$\forall x, \forall y, \forall z \; Student(x) \wedge Studies(x, y) \wedge HasWand(z) \Rightarrow Whizard(x)$$

$$Student(Ron) \wedge Student(Hermiona) \wedge Student(Harry) \wedge$$

$$Studies(Ron, G) \wedge Studies(Harry, G) \wedge Studies(Hermiona, G)$$

$$HasWand(Ron, Wanda) \wedge Wand(Wanda)$$

Forward and backward chaining : example KB according Russell, Norwig

- The law says that it is a crime for an American to sell weapons to a 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

IV. Forward chaining

1. *American(x) ∧ Weapon(y) ∧ Sells(x,y,z) ∧ Hostile(z) ⇒ Criminal(x)*
2. *Owns(Nono,M_1) , Missile(M_1)*
3. *Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)*
4. *Missile(x) ⇒ Weapon(x)*
5. *Enemy(x,America) ⇒ Hostile(x)*
6. *American(West)*
7. *Enemy(Nono,America)*

Inference: Is *Criminal(West)* entailed in the KB

a) Sentences 1, 3, 4, 5 are implications. Find a substitution which makes the premises satisfable.

b) **Sentence 3 is satisfable due to the** $\{x/M_1\}$**. To the KB is added** $Sells(West, M_1, Nono)$**.**

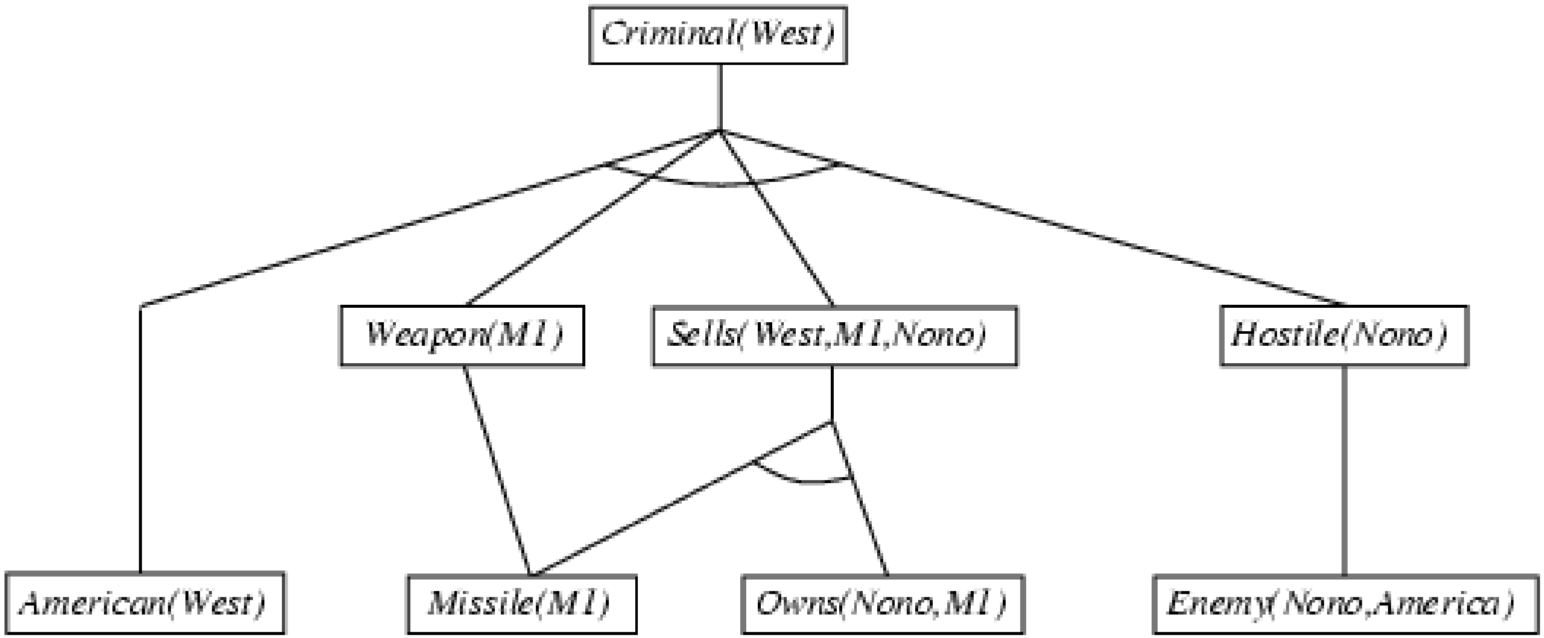
c) **Sentence 4 is satisfable due to** $\{x/M_1\}$ **and to the KB is added** $Weapon(M_1)$

d) **Sentence 5 is satisfable due to** $\{x/Nono\}$ **and *Hostile(Nono)* is added to the KB.**

e) **Sentence 1 is satisfable due to** $\{x/West, y/M_1, z/Nono\}$ **, derived also with a help of added formulas.**

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

Forward chaining scheme



V. Backward chaining

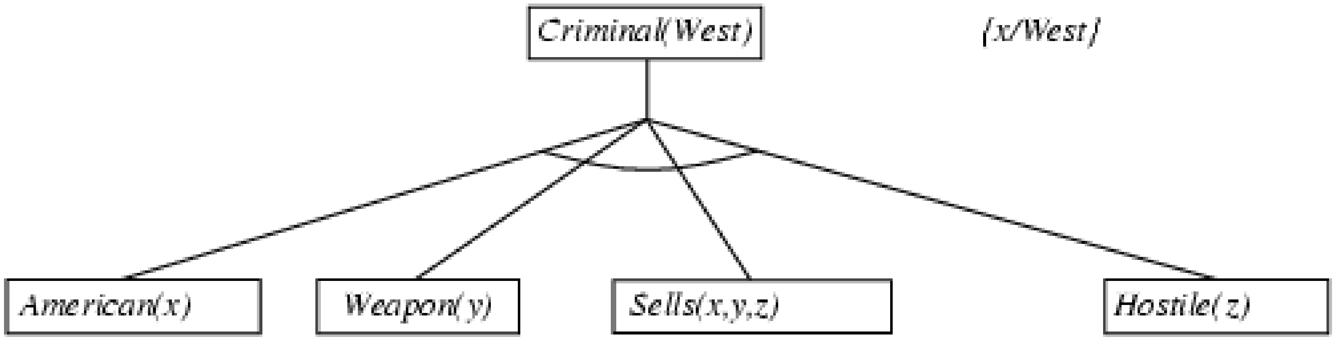
Starts with a list of goal propositions and returns a set of substitutions fulfilling the goals.

Backward chaining example

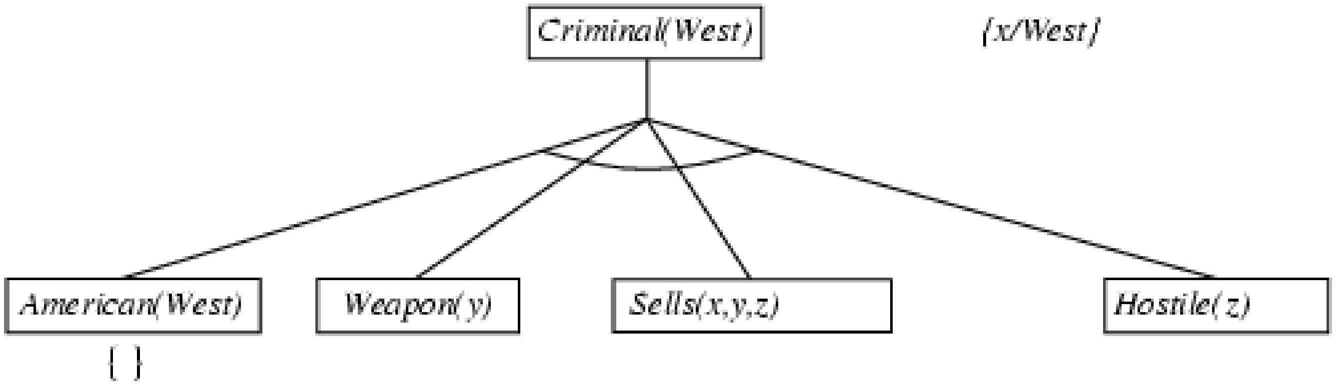
Criminal(West)

What is the truth value of Criminal West? Is it true?

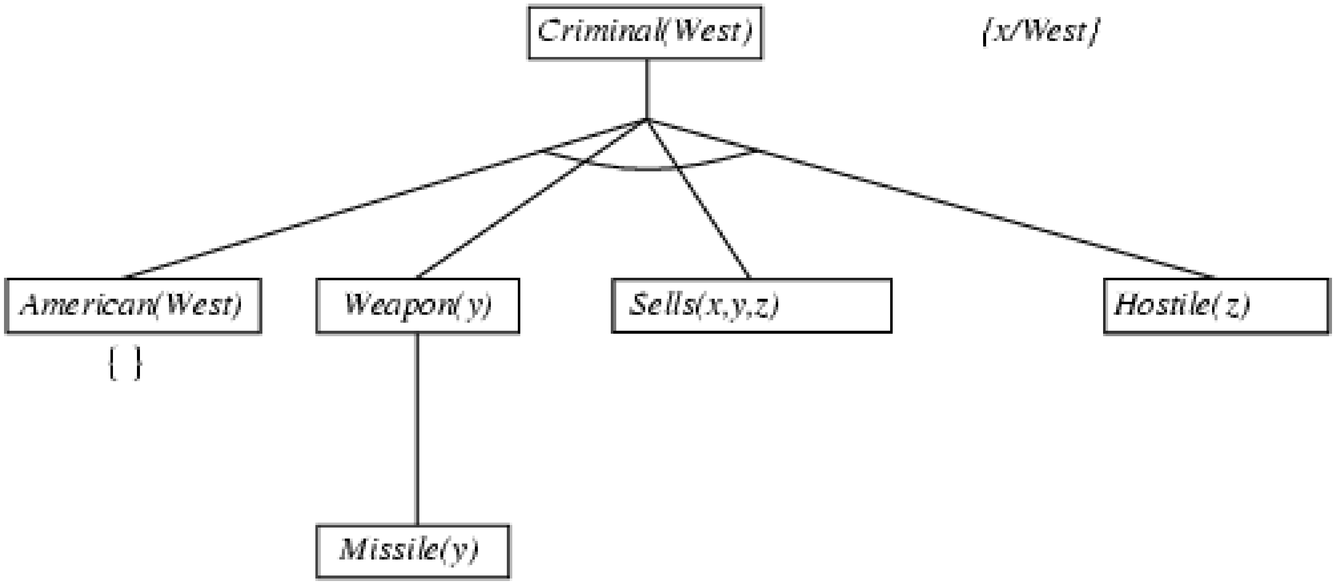
Backward chaining example



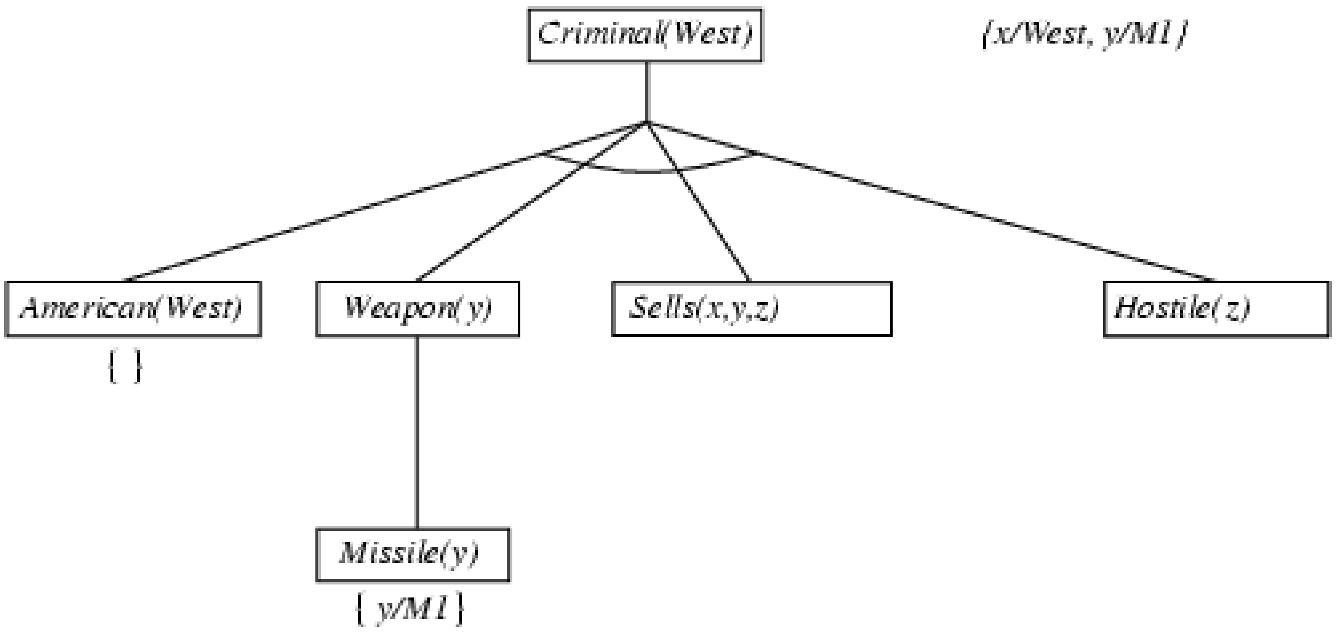
Backward chaining example



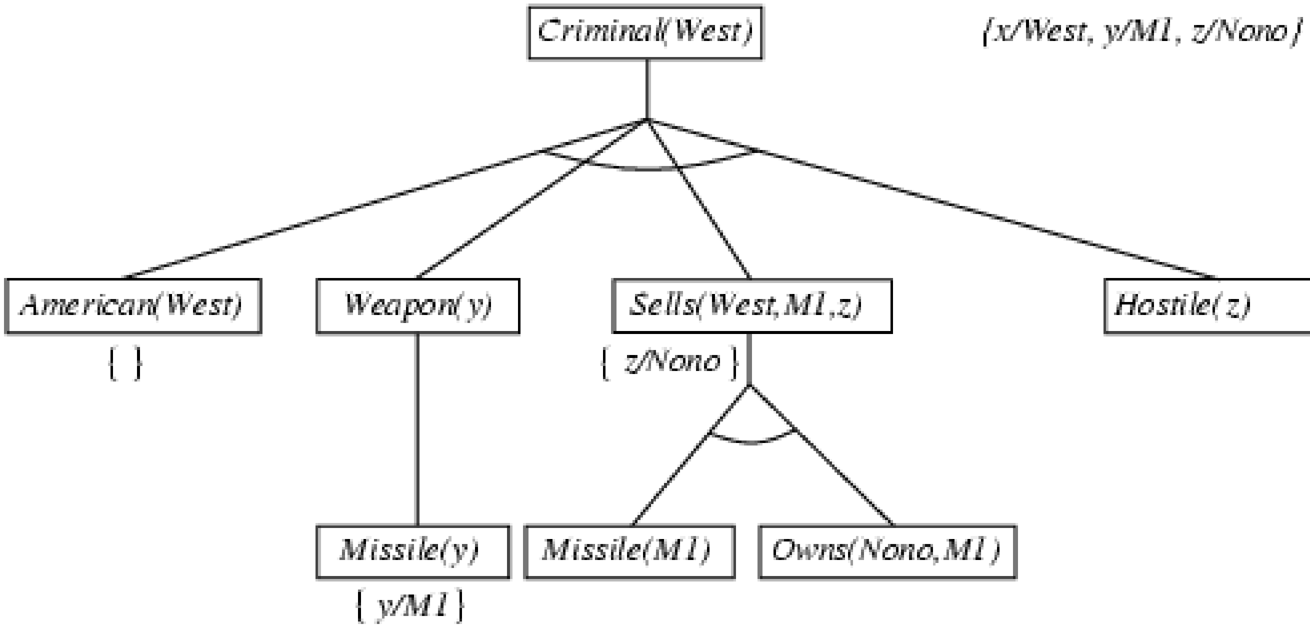
Backward chaining example



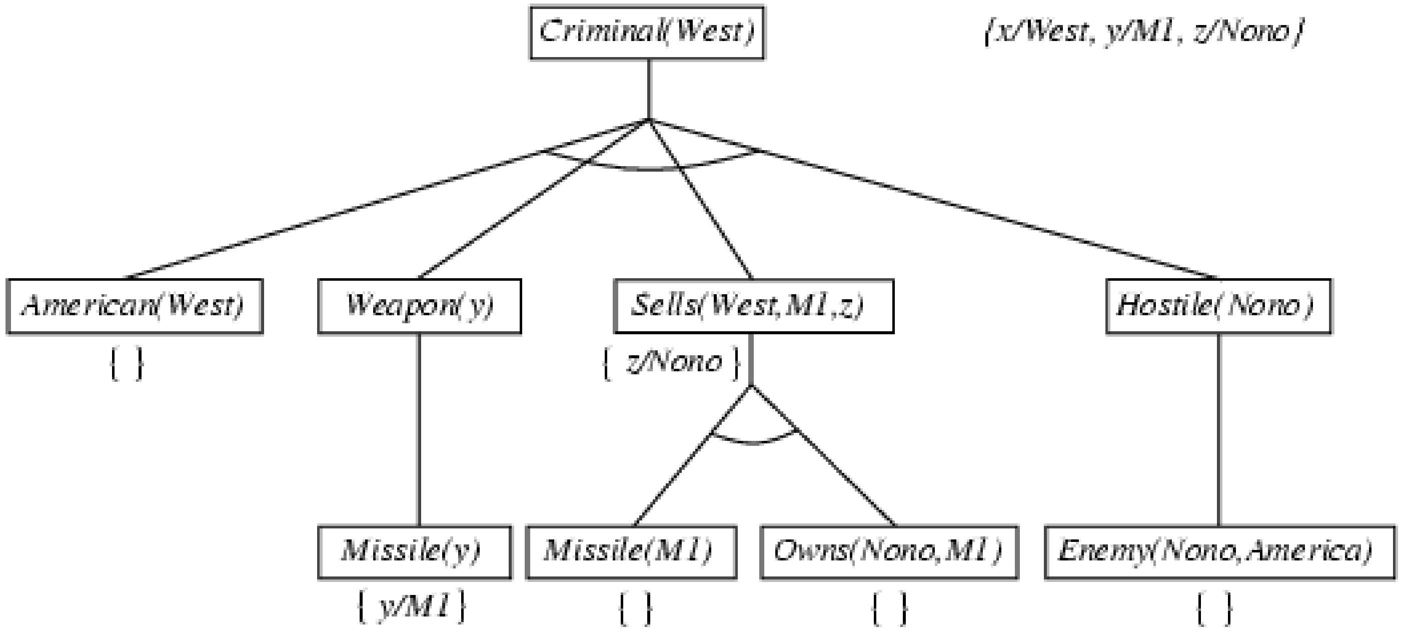
Backward chaining example



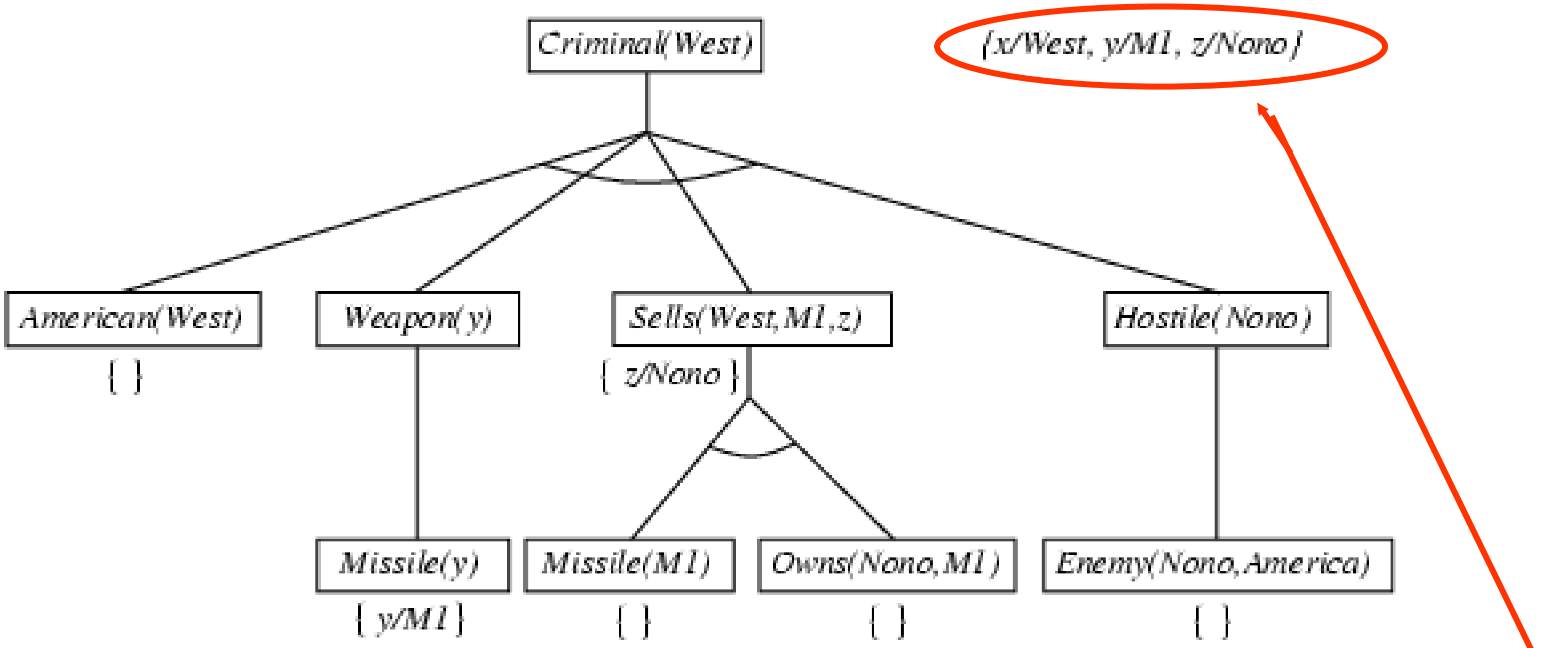
Backward chaining example



Backward chaining example



Backward chaining example



This substitution makes the premises satisfable.

VI. Resolution

Conjunctive normal form for FOL:

CNF: Conjunction of clauses of which each is a disjunction of literals.

Example:

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

CNF:

$$(\neg American(x) \vee \neg Weapon(y) \vee \neg Sells(x, y, z) \vee \neg Hostile(z))$$
$$\vee\; Criminal\;(x)$$

Conversion to the CNF

Example: **Everyone, who loves all animals, is loved by someone.**

$$\forall x[\forall y\ Animal(y) \Rightarrow Loves(x, y)]$$
$$\Rightarrow [\exists y\ Loves(y, x)]$$

Steps: **1. Implication elimination**

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

2. **Use rules for the quantifier negation.**

$$\neg \forall x \ p \qquad \exists x \ \neg p$$

$$\neg \exists x \ p \qquad \forall x \ \neg p$$

So:

$$\forall x [\exists y \neg (\neg Animal(y) \vee Loves(x, y))] \vee [\exists y \ Loves(y, x)]$$
$$\forall x [\exists y \neg \neg Animal(y) \wedge \neg Loves(x, y)] \vee [\exists y \ Loves(y, x)]$$
$$\forall x [\exists y \ Animal(y) \wedge \neg Loves(x, y)] \vee [\exists y \ Loves(y, x)]$$

3. Variable standarization

If there is one name for different objects, variables, change it.

$$\forall x[\exists y\ Animal(y) \wedge \neg Loves(x, y)] \vee [\exists z\ Loves(z, x)]$$

4. Scolemization (elimination of the existence quantifier)

$$\forall x[Animal(A) \wedge \neg Loves(x, A)] \vee [Loves(B, x)]$$

Attention, bad scolemization

Correct scolmization:

$$\forall x\left[Animal(F(x)) \wedge \neg Loves(x, F(x))\right] \vee Loves(G(x), x)$$

Scolem functions

5. Skip general quantifiers

$$\left[Animal(F(x)) \wedge \neg Loves(x, F(x))\right] \vee Loves(G(x), x)$$

6. Rewrite $\vee$ on $\wedge$ and final adjustments.

$$[Animal(F(x)) \vee Loves(G(x), x)] \wedge$$
$$[\neg Loves(x, F(x)) \vee Loves(G(x), x)]$$

$$\Downarrow$$

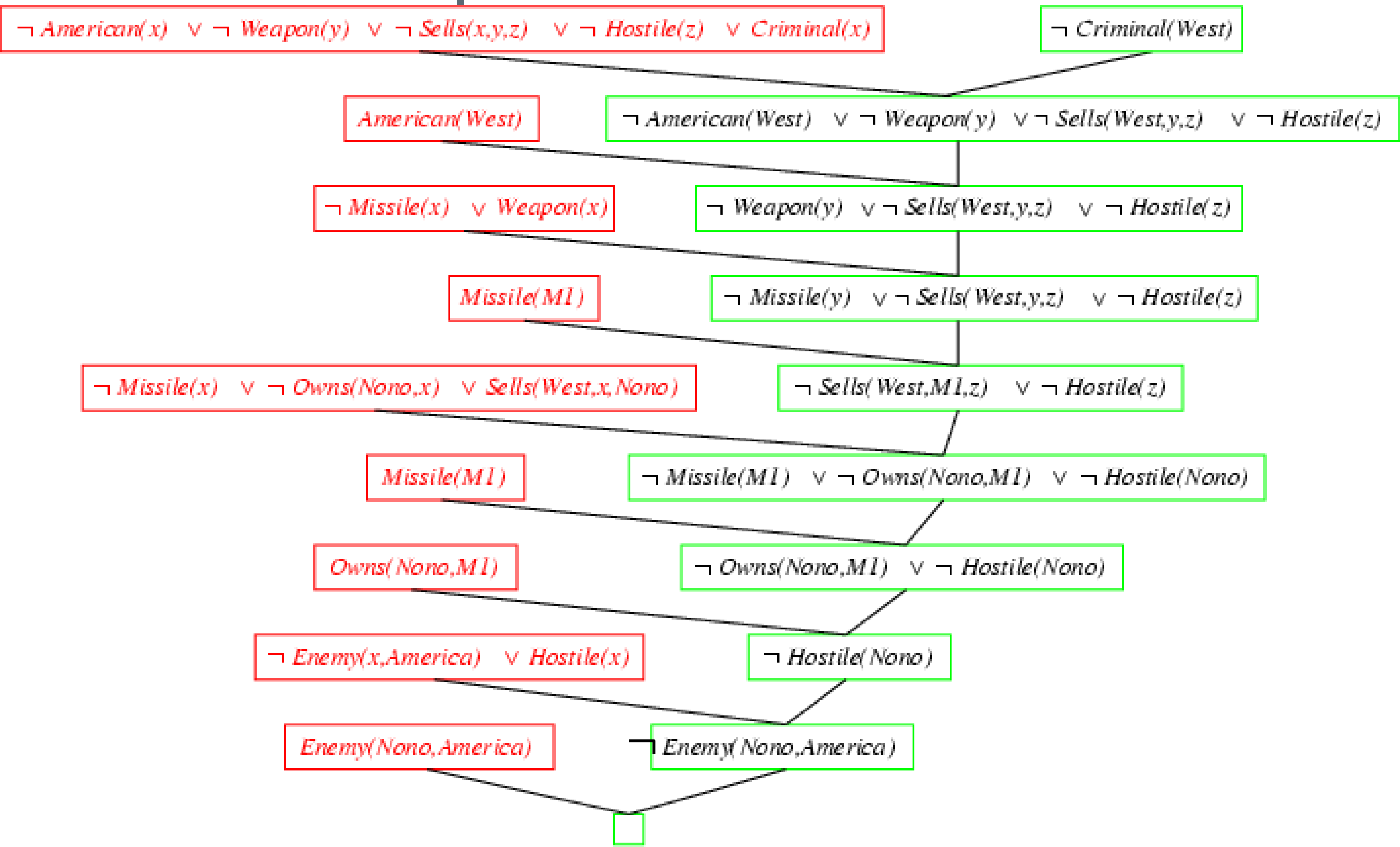
The sentence is in the CNF form, it is a conjunction of two disjunctions.

Rezolution:

Resolution proves that $KB \models \alpha$ **by contradiction,**

by proving unsatisfability of $KB \land \neg \alpha$

Resolution proof: definite clauses



One more example from an exam

Situation: If a person is a climber then he is fit.
If a person is fit, then he endures low temperature.
Jan is a climber and person. Peter is fit. -10 C is a low temperature.

Prove that Jan endures low temperature -10 C.

Predicates: *Person(x), Climber(x), EnduresLowTemp(x,y), Fit(x), LowTemp(y)*

KB: $\forall x\ Person(x) \wedge Climber(x) \Rightarrow Fit(x)$

$\forall x \exists y\ Fit(x) \wedge Person(x) \wedge LowTemp(y) \Rightarrow EnduresLowTemp(x, y)$

Climber(Jan), Person(Jan)
Fit(Peter)
LowTemp(-10)

Summary

- Propositional KB, semantics, syntax
- CNF forma
- Inference methods in propositional KB, resolution, forward and backward chaining
- FOL KB
- Unification, substitution
- Generalized inference methods in FOL KB