Artificial Intelligence

Hai Thi Tuyet Nguyen

Outline

CHAPTER 6: FIRST-ORDER LOGIC

6.1 Representation Revisited

6.1 Propositional logic

- Propositional logic is declarative: pieces of syntax correspond to facts
- Propositional logic allows partial/disjunctive/negated information
  (unlike most data structures and databases)
- Propositional logic is compositional:
  meaning of $B_{1,1} \wedge P_{1,2}$ is derived from meaning of $B_{1,1}$ and of $P_{1,2}$
- Meaning in propositional logic is context-independent
  (unlike natural language, where meaning depends on context)
- Propositional logic has very limited expressive power
  E.g., cannot say "pits cause breezes in adjacent squares"
  except by writing one sentence for each square

6.1 First-order logic

Whereas propositional logic assumes world contains facts, first-order logic (like natural language) assumes the world contains

- Objects: people, houses, numbers, theories, colors, baseball games
- 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

6.2 Syntax And Semantics Of First-Order Logic

6.2 Syntax And Semantics Of First-Order Logic

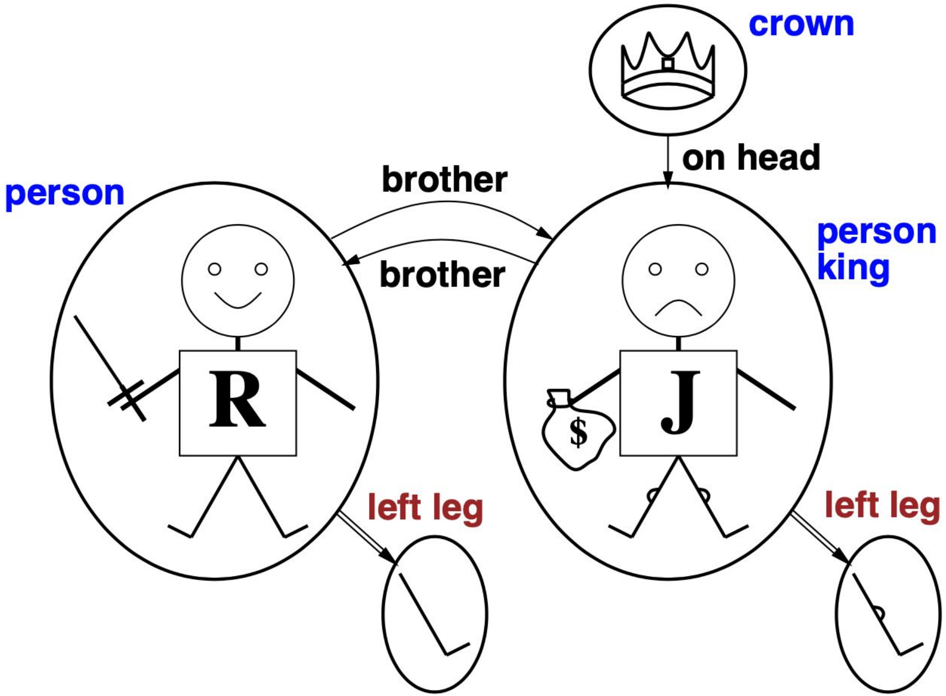
- Model contains at least objects and relations among them

- Basic symbols:
  - constant symbols → objects
  - predicate symbols → relations
  - function symbols → functional relations

- Interpretation specifies exactly which objects, relations and functions are referred to by the constant, predicate, and function symbols
  Example:
  Richard refers to *Richard the Lionheart, John* refers to *the evil King John*
  Brother refers to *the brotherhood relation*
  LeftLeg refers to *the "left leg" function*

6.2 Models for FOL: Example

- 5 objects:
  - Richard the Lionheart;
  - the evil King John;
  - the left legs of Richard and John;
  - a crown
- Relations:
  - binary relations:
    - “brother” and “on head” (they relate pairs of objects)
  - unary relations, or properties
    - “person”, “king”, “crown”
  - functions: a given object must be related to **exactly one** object
    - “left leg”



6.2 Syntax of FOL: Basic elements

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

6.2 Atomic sentences

- Atomic sentence: predicate($term_1$, ..., $term_n$) or $term_1$ = $term_2$
  E.g., Brother(KingJohn, RichardTheLionheart)

- Term: function(term1, ..., termn) or constant or variable
  E.g., > (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))

6.2 Complex sentences

Complex sentences are made from atomic sentences using connectives

$\neg S$, $S_1 \land S_2$, $S_1 \lor S_2$, $S_1 \Rightarrow S_2$, $S_1 \Leftrightarrow S_2$

E.g.

$Sibling(KingJohn, Richard) \Rightarrow Sibling(Richard, KingJohn)$

$>(1, 2) \lor \leq(1, 2)$

$>(1, 2) \land \neg>(1, 2)$

6.2 Universal quantification

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

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

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

Roughly speaking, equivalent to the conjunction of instantiations of P

$(At(KingJohn,Berkeley) \Rightarrow Smart(KingJohn))$

$\wedge\ (At(Richard,Berkeley) \Rightarrow Smart(Richard))$

$\wedge\ (At(Berkeley,Berkeley) \Rightarrow Smart(Berkeley))$

$\wedge\ \ldots$

6.2 Existential quantification

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

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

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

Roughly speaking, equivalent to the disjunction of instantiations of P

$(At(KingJohn, Stanford) \land Smart(KingJohn))$

$\lor\ (At(Richard, Stanford) \land Smart(Richard))$

$\lor\ (At(Stanford, Stanford) \land Smart(Stanford))$

$\lor\ \ldots$

6.2 Properties of quantifiers

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

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

Quantifier duality: each can be expressed using the other

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

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

6.2 Equality

$term_1 = term_2$ is true if and only if $term_1$, $term_2$ refer to the same object

E.g., Father(John)=Henry

6.3 Using First-Order Logic

6.3 Using First-Order Logic

Assertions and queries in first-order logic

Sentences are added to a knowledge base using TELL, (called as assertions)

For example, we can assert that John is a king, Richard is a person, and all kings are persons:

TELL(KB, King(John)) .

TELL(KB, Person(Richard)) .

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

We can ask questions of the knowledge base using ASK (queries or goals).

ASK(KB, King(John))

6.3 Inference rules for quantifiers

- **Universal Instantiation rule:**
  SUBST(θ, α) denote the result of applying the substitution θ to the sentence α
  Rule: substituting a ground term (a term without variables) for the variable

$$\frac{\forall v \;\; \alpha}{\textsc{Subst}(\{v/g\}, \alpha)}$$

  for any variable v, ground term g

  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))

6.3 Inference rules for quantifiers

- **Existential instantiation (EI)**

  For any sentence α, variable v, and constant symbol k that does not appear elsewhere in the knowledge base:

$$\frac{\exists v \;\; \alpha}{\textsc{Subst}(\{v/k\}, \alpha)}$$

E.g., ∃x Crown(x)∧OnHead(x, John) yields

Crown(C1) ∧ OnHead(C1, John)

provided C1 is a new constant symbol, called a Skolem constant

6.3 Inference rules for quantifiers

- UI can be applied several times to add new sentences;
  the new KB is logically equivalent to the old
- EI can be applied once to replace the existential sentence;
  the new KB is not equivalent to the old,
  but is satisfiable iff the old KB was satisfiable

6.3 Reduction to propositional inference

Suppose the KB contains just the following:

$\forall x\ King(x) \land Greedy(x) \Rightarrow Evil(x)$
King(John)
Greedy(John)
Brother(Richard, John)

Instantiating the universal sentence in all possible ways, we have

$King(John) \land Greedy(John) \Rightarrow Evil(John)$
$King(Richard) \land Greedy(Richard) \Rightarrow Evil(Richard)$
King(John)
Greedy(John)
Brother(Richard, John)

The new KB is propositionalized: proposition symbols are
King(John), Greedy(John), Evil(John), King(Richard), etc.

6.3 Reduction to propositional inference

- Claim: every FOL KB can be propositionalized so as to preserve entailment
- Idea: propositionalize KB and query, apply propositional resolution, obtain result
- Problem: with function symbols, there are infinitely many ground terms,
  e.g., Father(Father(Father(John))
- Theorem: Herbrand (1930).
  If a sentence α is entailed by an FOL KB, it is entailed by a finite subset of the propositional KB
  Idea: For n=0 to ∞ do
  create a propositional KB by instantiating with depth-n terms
  see if α is entailed by this KB
  Problem: works if α is entailed, loops if α is not entailed
- Theorem: Turing (1936), Church (1936), entailment in FOL is semidecidable

6.3 Generalized Modus Ponens (GMP)

- For atomic sentences p_i, p'_i, and q, where there is a substitution θ, such that $\text{SUBST}(\theta, p_i) = \text{SUBST}(\theta, p'_i)$, for all i

$$\frac{p_1{}', \;\; p_2{}', \;\; \ldots, \;\; p_n{}', \;\; (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{\text{SUBST}(\theta, q)}$$

p_1' is King(John)	p_1 is King(x)
p_2' is Greedy(y)	p_2 is Greedy(x)
θ is {x/John, y/John}	q is Evil(x)
SUBST(θ, q) is Evil(John)	

6.3 Unification

- We can get the inference if we can find a substitution θ such that King(x) and Greedy(x) match King(John) and Greedy(y)

UNIFY(p, q) = θ where SUBST(θ, p) = SUBST(θ, q) .

p	q	θ
$Knows(John, x)$	$Knows(John, Jane)$	$\{x/Jane\}$
$Knows(John, x)$	$Knows(y, OJ)$	$\{x/OJ, y/John\}$
$Knows(John, x)$	$Knows(y, Mother(y))$	$\{y/John, x/Mother(John)\}$
$Knows(John, x)$	$Knows(x, OJ)$	$fail$

Standardizing apart (renaming) eliminates overlap of variables, e.g., Knows(x_{17},OJ)

6.3 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 it by Colonel West, who is American.
Prove that Col. West is a criminal.

6.3 Example knowledge base contd.

It is a crime for an American to sell weapons to hostile nations:

$$American(x) \land Weapon(y) \land Sells(x, y, z) \land Hostile(z) \Rightarrow Criminal(x) \qquad (9.3)$$

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

$$Owns(Nono, M1) \qquad (9.4)$$

$$Missile(M1) \qquad (9.5)$$

with M1 as a new constant

. . . all of its missiles were sold to it by Colonel West

$$\forall x\ Missile(x) \land Owns(Nono,x) \Rightarrow Sells(West,x,Nono) \qquad (9.6)$$

Missiles are weapons: $Missile(x) \Rightarrow Weapon(x)$ (9.7)

An enemy of America counts as “hostile”: $Enemy(x,America) \Rightarrow Hostile(x)$ (9.8)

West, who is American . . . : $American(West)$ (9.9)

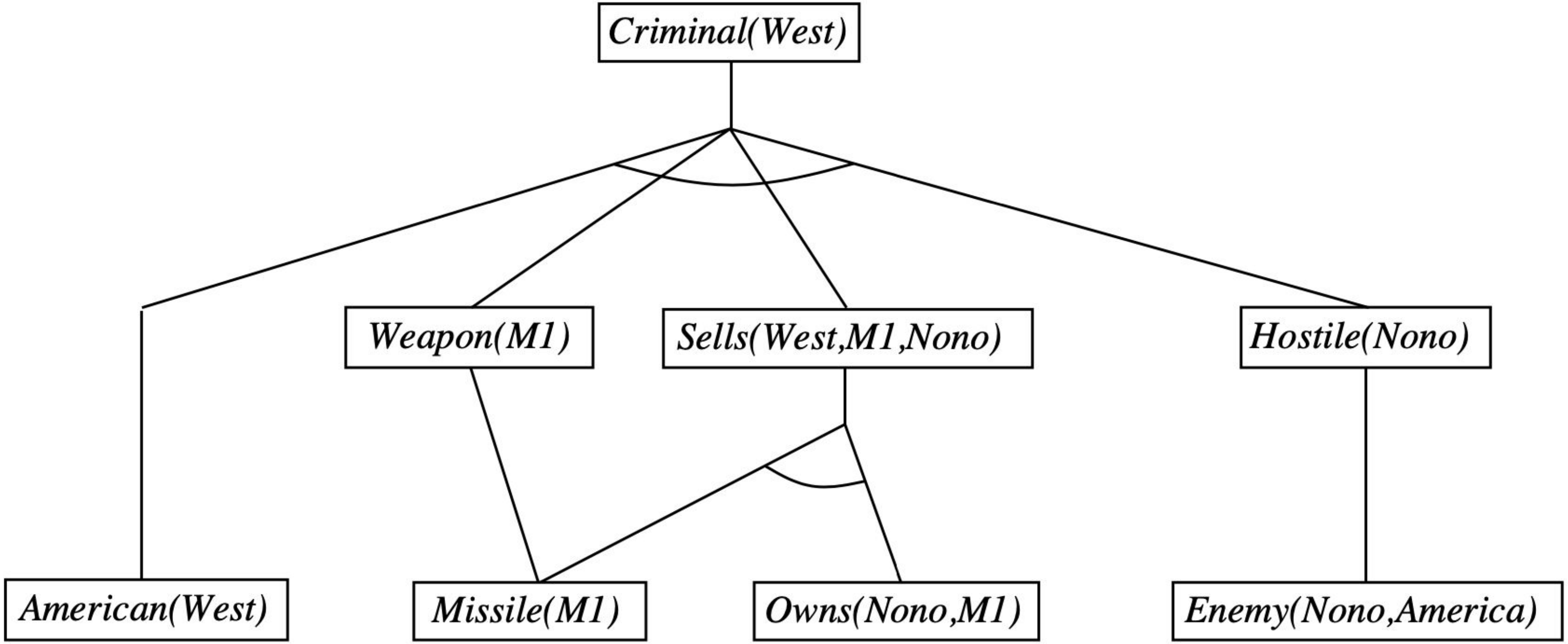
The country Nono, an enemy of America . . . $Enemy(Nono, America)$ (9.10)

6.3 Forward chaining algorithm

```
function FOL-FC-ASK(KB, α) returns a substitution or false
  inputs: KB, the knowledge base, a set of first-order definite clauses
          α, the query, an atomic sentence
  local variables: new, the new sentences inferred on each iteration

  repeat until new is empty
     new ← { }
     for each rule in KB do
        (p1 ∧ ... ∧ pn ⇒ q) ← STANDARDIZE-VARIABLES(rule)
        for each θ such that SUBST(θ, p1 ∧ ... ∧ pn) = SUBST(θ, p'1 ∧ ... ∧ p'n)
                    for some p'1, ..., p'n in KB
           q' ← SUBST(θ, q)
           if q' does not unify with some sentence already in KB or new then
              add q' to new
              φ ← UNIFY(q', α)
              if φ is not fail then return φ
     add new to KB
  return false
```

6.3 Forward chaining proof

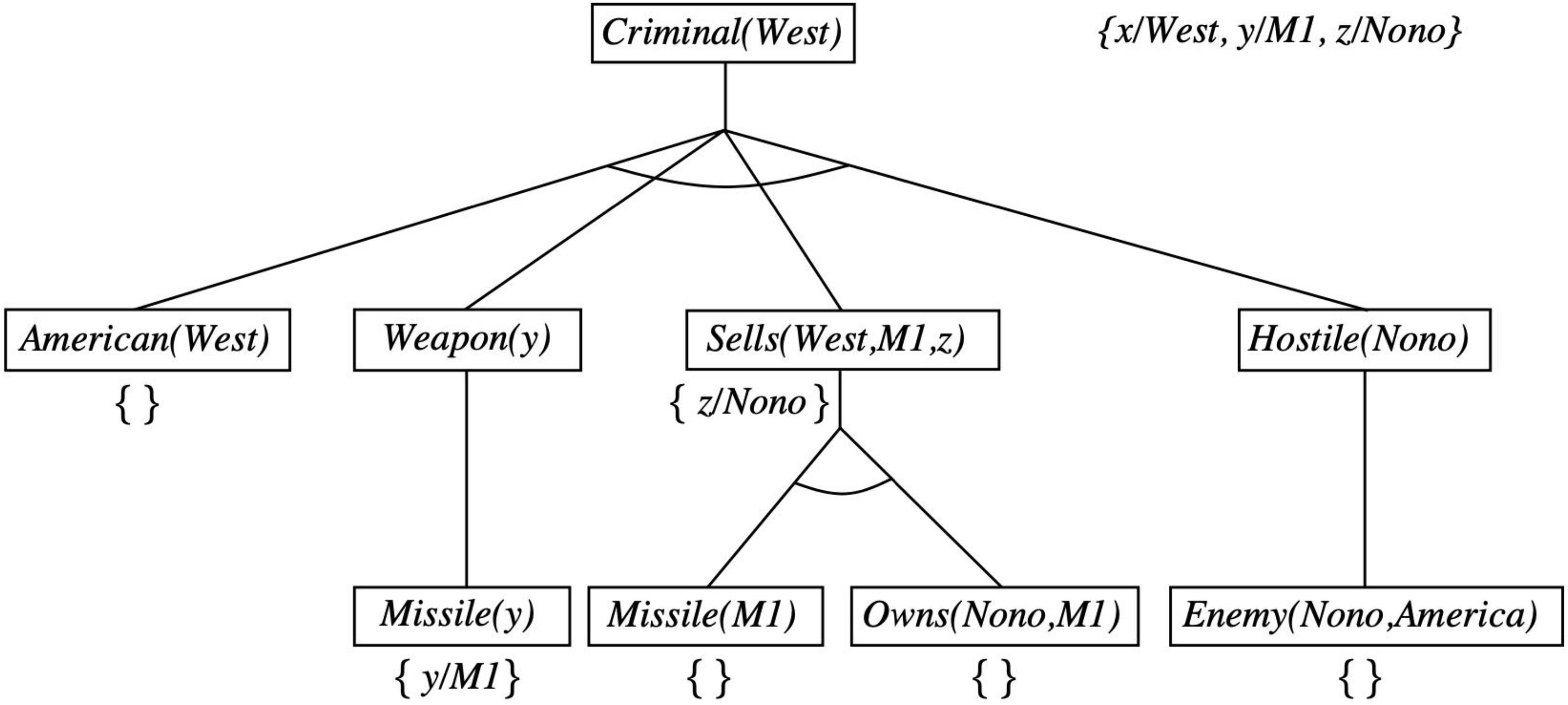


6.3 Backward chaining algorithm

function FOL-BC-ASK(KB, $goals$, θ) **returns** a set of substitutions
  inputs: KB, a knowledge base
    $goals$, a list of conjuncts forming a query (θ already applied)
    θ, the current substitution, initially the empty substitution $\{\,\}$
  local variables: $answers$, a set of substitutions, initially empty

  if $goals$ is empty **then return** $\{\theta\}$
  $q' \leftarrow \text{SUBST}(\theta, \text{FIRST}(goals))$
  for each sentence r **in** KB
    where STANDARDIZE-APART(r) $= (\, p_1 \;\wedge\; \ldots \;\wedge\; p_n \;\Rightarrow\; q)$
    and $\theta' \leftarrow \text{UNIFY}(q, q')$ succeeds
   $new\_goals \leftarrow [\, p_1, \ldots, p_n | \text{REST}(goals)]$
   $answers \leftarrow \text{FOL-BC-ASK}(KB, new\_goals, \text{COMPOSE}(\theta', \theta)) \cup answers$
  return $answers$

6.3 Backward chaining example



Backward chaining example

- Proof tree constructed by backward chaining to prove that West is a criminal.
- The tree should be read depth first, left to right.
- To prove Criminal (West), we have to prove the four conjuncts below it.
- Some of these are in the knowledge base, and others require further backward chaining.
- Bindings for each successful unification are shown next to the corresponding subgoal. Note that once one subgoal in a conjunction succeeds, its substitution is applied to subsequent subgoals.
- Thus, by the time FOL-BC-ASK gets to the last conjunct, originally Hostile(z), z is already bound to Nono.

6.3 Resolution

Full first-order version:

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{(\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n)\theta}$$

where Unify(l_i, $\neg m_j$) = θ.

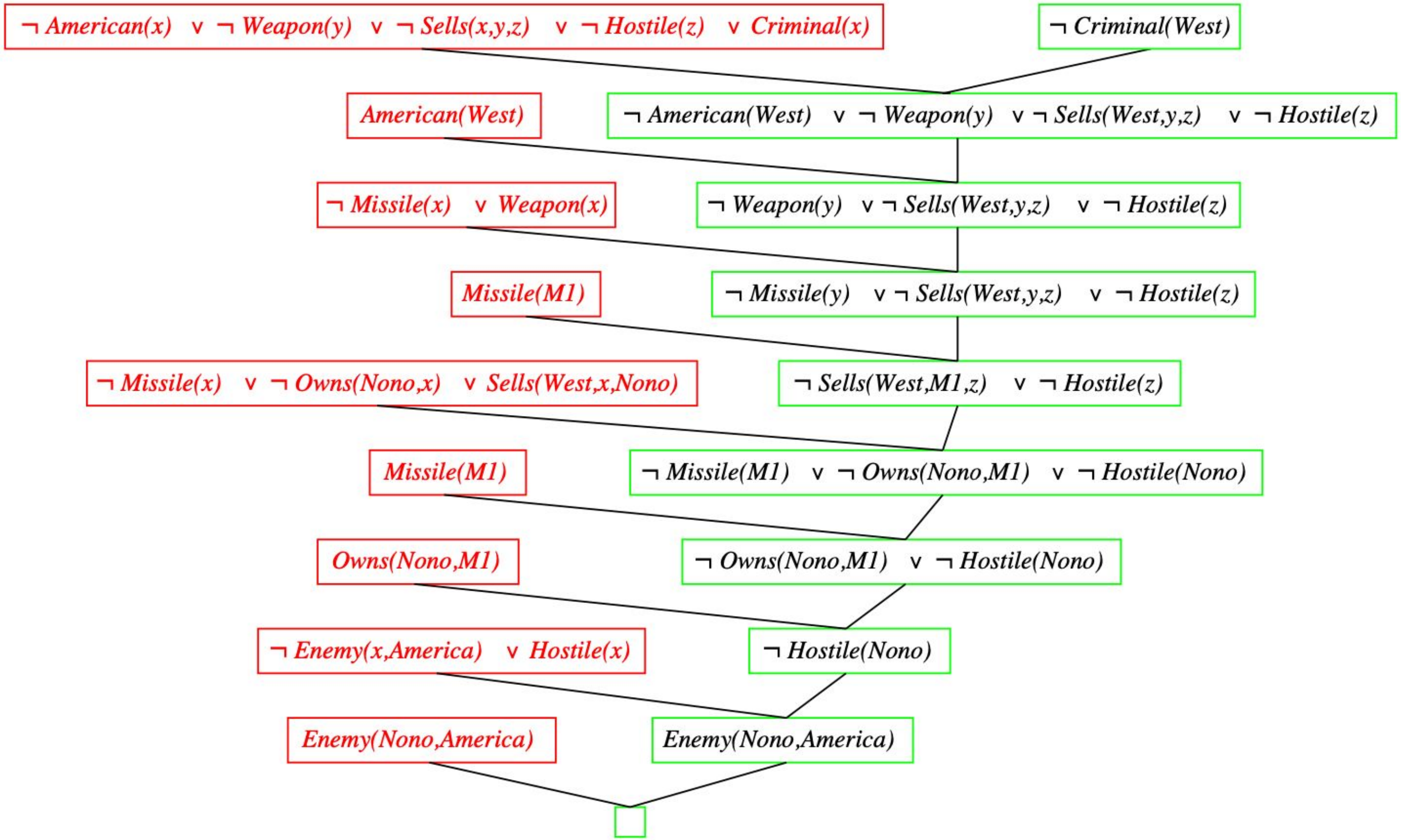
For example,

$$\frac{\begin{array}{l}\neg Rich(x) \vee Unhappy(x) \\ Rich(Ken)\end{array}}{Unhappy(Ken)}$$

with $\theta = \{x/Ken\}$

Apply resolution steps to CNF(KB $\wedge$ $\neg\alpha$); complete for FOL

6.3 Resolution proof: definite clauses



6.3 Conversion to CNF

1. Eliminate biconditionals and implications

$S_1 => S_2 \equiv \neg S_1 \lor S_2$

$S_1 <=> S_2 \equiv (S_1 => S_2) \land (S_2 => S_1)$

2. Move $\neg$ inwards:

$\neg \forall x,p \equiv \exists x \neg p, \neg \exists x,p \equiv \forall x \neg p$

3. Standardize variables: each quantifier should use a different one

4. Skolemize: each existential variable is replaced by a Skolem function of the enclosing universally quantified variables

5. Drop universal quantifiers

6. Distribute $\land$ over $\lor$