PURDUE CS47100

INTRODUCTION TO AI

ANNOUNCEMENTS

- No class on Thursday (October 20)!
- Midterm on Thursday!
  - 8-10pm, PHYS 112
  - Closed-book, closed-notes!
  - More on midterm later today

RECAP: FIRST-ORDER LOGIC

- Express sentences in first-order logic
- Propositionalize first-order logic sentences
- Unification: find substitution of variables to make two sentences the same!

UNIFICATION

- Two sentences α,β can be **unified** with substitution θ if SUBST(θ, α)=SUBST(θ, β)

α	β	θ
Knows(John,x)	Knows(John,Jane)	**{x/Jane}**
Knows(John,x)	Knows(y,Steve)	**{x/Steve,y/John}**
Knows(John,x)	Knows(y,Mother(y))	**{y/John,x/Mother(John)}**
Knows(John,x)	Knows(x,Steve)	**{fail}**

- Standardizing apart eliminates overlap of variables, e.g., Knows(z,Steve)

RESOLUTION

- First convert to CNF (as with propositional logic)
- Everyone who loves all animals is loved by someone:
  $\forall x\ [\forall y\ \mathit{Animal}(y) \Rightarrow \mathit{Loves}(x,y)] \Rightarrow [\exists y\ \mathit{Loves}(y,x)]$
- 1. Eliminate biconditionals and implications
  $\forall x\ [\neg\forall y\ \neg\mathit{Animal}(y) \lor \mathit{Loves}(x,y)] \lor [\exists y\ \mathit{Loves}(y,x)]$
- 2. Move ¬ inwards with following rules: $\neg\forall x\ p \equiv \exists x\ \neg p$, $\neg\exists x\ p \equiv \forall x\ \neg p$
  $\forall x\ [\exists y\ \neg(\neg\mathit{Animal}(y) \lor \mathit{Loves}(x,y))] \lor [\exists y\ \mathit{Loves}(y,x)]$
  $\forall x\ [\exists y\ \neg\neg\mathit{Animal}(y) \land \neg\mathit{Loves}(x,y)] \lor [\exists y\ \mathit{Loves}(y,x)]$
  $\forall x\ [\exists y\ \mathit{Animal}(y) \land \neg\mathit{Loves}(x,y)] \lor [\exists y\ \mathit{Loves}(y,x)]$

RESOLUTION: CONVERT TO CNF

- 3. Standardize variables: each quantifier should use a different one
  $\forall x\ [\exists y\ Animal(y) \land \neg Loves(x,y)] \lor [\exists z\ Loves(z,x)]$
- 4. Skolemize: a more general form of existential instantiation.
  Each existential variable is replaced by a **Skolem function** of the enclosing universally quantified variables:
  $\forall x\ [Animal(F(x)) \land \neg Loves(x,F(x))] \lor Loves(G(x),x)$
- 5. Drop universal quantifiers
  $[Animal(F(x)) \land \neg Loves(x,F(x))] \lor Loves(G(x),x)$
- 6. Distribute $\lor$ over $\land$:
  $[Animal(F(x)) \lor Loves(G(x),x)] \land [\neg Loves(x,F(x)) \lor Loves(G(x),x)]$

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 Colonel West is a criminal

EXAMPLE KNOWLEDGE BASE

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

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

Nono … has some missiles, i.e., ∃x Owns(Nono,x) ∧ Missile(x)

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

Missiles are weapons: **∀x Missile(x) ⇒ Weapon(x)**

An enemy of America counts as "hostile": **∀x Enemy(x,America) ⇒ Hostile(x)**

West, who is American: **American(West)**

The country Nono, an enemy of America: **Enemy(Nono,America)**

(To Prove) Colonel West is a criminal: Criminal(West)

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

$\neg$American(x) $\vee$ $\neg$Weapon(y) $\vee$
$\neg$Sells(x,y,z) $\vee$ $\neg$Hostile(z) $\vee$
Criminal(x)
Owns(Nono,M_1)
Missile(M_1)
$\neg$Missile(x) $\vee$ $\neg$Owns(Nono,x) $\vee$
Sells(West,x,Nono)
$\neg$Missile(x) $\vee$ Weapon(x)
$\neg$Enemy(x,America) $\vee$ Hostile(x)
American(West)
Enemy(Nono,America)

GENERALIZED MODUS PONENS

$$\frac{p'_1, p'_2, \ldots, p'_n \quad (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{\mathrm{SUBST}(\theta, q)}$$

where $\mathrm{SUBST}\ (\theta, p'_i) = \mathrm{SUBST}\ (\theta, p_i)$ for all i

p'_1 is King(John) p_1 is King(x)

p'_2 is Greedy(y) p_2 is Greedy(x)

q is Evil(x) θ is {x/John,y/John}

SUBST(θ, q) is Evil(John)

Example:

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

King(John)

$\forall$y Greedy(y)

- Generalized Modus Ponens is a **lifted** version of Modus Ponens
  - What is the difference between MP and GMP? GMP is used with KB of **definite clauses**

FIRST-ORDER DEFINITE CLAUSES

- Disjunctions of literals where **exactly one** is positive (i.e., unegated)
  - Atomic clauses $(\alpha \wedge \delta) \Rightarrow \rho \Leftrightarrow \neg\alpha \vee \neg\delta \vee \rho$
  - Implications whose antecedent is a conjunction of positive literals and whose consequent is a single positive literal
- All variables are assume to be universally quantified and are typically omitted
- Datalog KB = first-order definite clauses + **no functions**
  - Enables efficient inference with forward/backward chaining

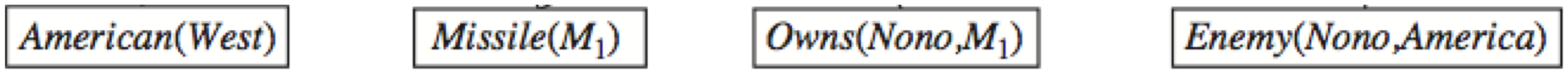
FORWARD CHAINING

- Start from known facts
- Trigger all rules whose premises are known
- Add conclusions to set of known facts
- Repeat until query is answered or no new fact is added
  - A fact is not new if it is a renaming of a known fact

Proof that Colonel West is a criminal

FORWARD CHAINING PROOF

American(x) ∧ Weapon(y) ∧
Sells(x,y,z) ∧ Hostile(z) ⇒
Criminal(x)
Owns(Nono,M_1)
Missile(M_1)
Missile(x) ∧ Owns(Nono,x) ⇒
Sells(West,x,Nono)
Missile(x) ⇒ Weapon(x)
Enemy(x,America) ⇒ Hostile(x)
American(West)
Enemy(Nono,America)

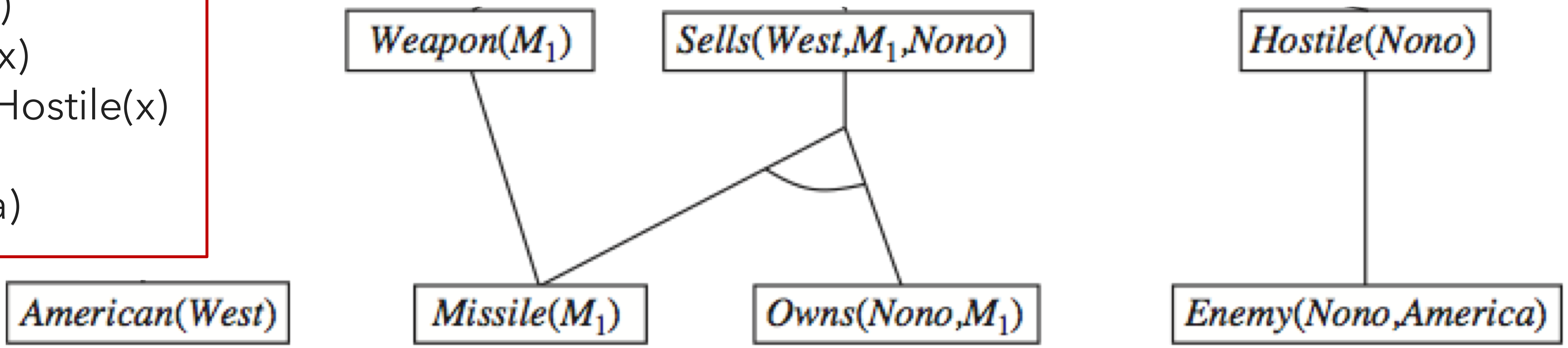


Owns(Nono,M_1) and Missile(M_1)
American(West)
Enemy(Nono,America)

Proof that Colonel West is a criminal

FORWARD CHAINING PROOF

American(x) ∧ Weapon(y) ∧ Sells(x,y,z) ∧ Hostile(z) ⇒ Criminal(x)
Owns(Nono,M_1)
Missile(M_1)
Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)
Missile(x) ⇒ Weapon(x)
Enemy(x,America) ⇒ Hostile(x)
American(West)
Enemy(Nono,America)

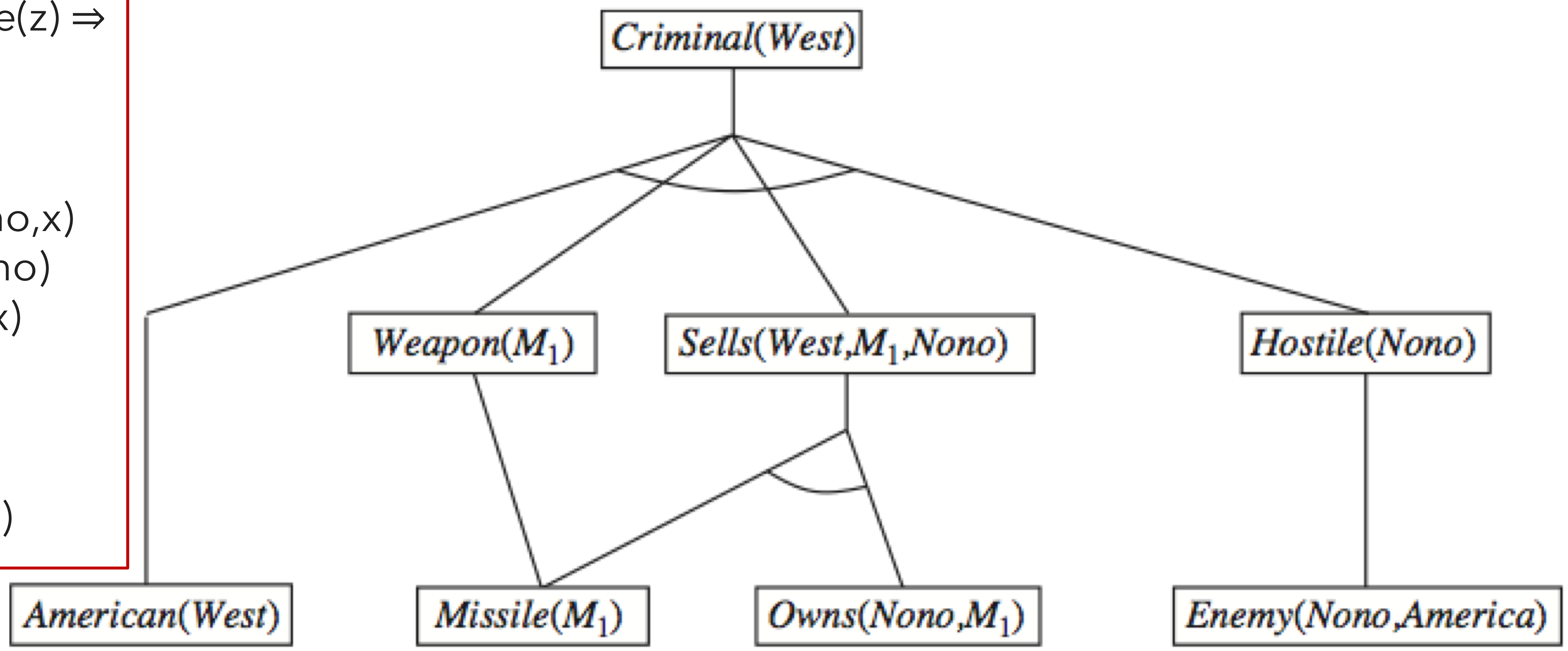


Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)
Missile(x) ⇒ Weapon(x)
Enemy(x,America) ⇒ Hostile(x)

Proof that Colonel West is a criminal

FORWARD CHAINING PROOF

American(x) ∧ Weapon(y) ∧ Sells(x,y,z) ∧ Hostile(z) ⇒ Criminal(x)
Owns(Nono,M_1)
Missile(M_1)
Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)
Missile(x) ⇒ Weapon(x)
Enemy(x,America) ⇒ Hostile(x)
American(West)
Enemy(Nono,America)



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

PROPERTIES OF FORWARD CHAINING

- Forward chaining is widely used in **deductive databases**
- Forward chaining (FC) is sound and complete for first-order definite clauses
- FC terminates for Datalog in finite number of iterations
- May not terminate in general if α is not entailed
- This is unavoidable – entailment with definite clauses is semidecidable

BACKWARD CHAINING

- Start with query in the “stack” of goals
- Take each goal in the stack
  - Find every clause whose head unifies with the goal
  - Add body of the clause to the stack
  - Facts are clauses with only a head (no body), so when a clause unifies with a fact no new goal is added to the stack
- When stack is empty, query is proved
  - Return the set of satisfying substitutions
  - Uses composition of substitutions, which is identical to applying each substitution in turn

Proof that Colonel West is a criminal

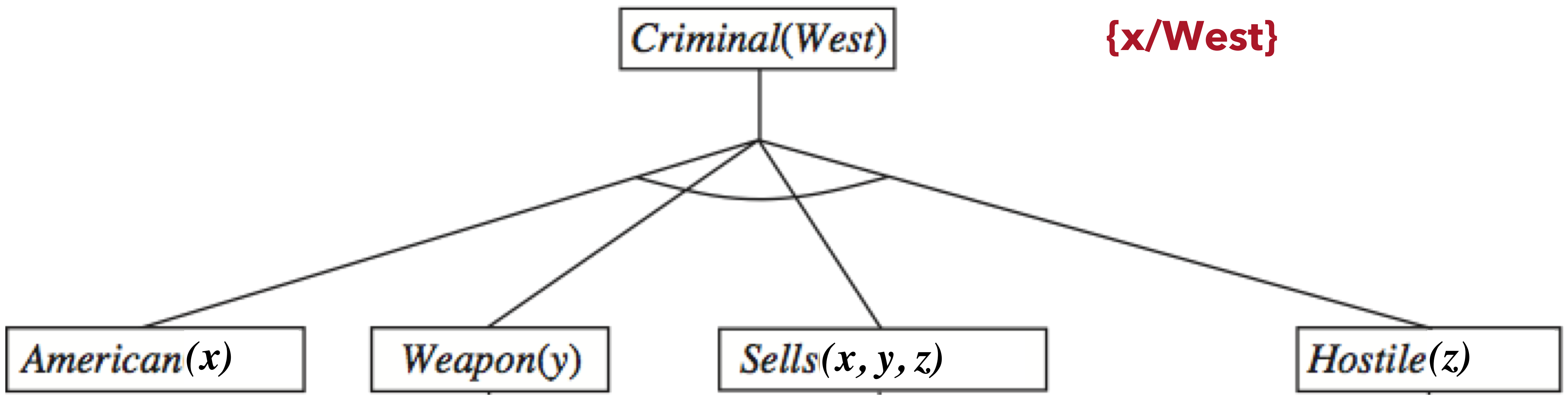
BACKWARD CHAINING EXAMPLE

American(x) ∧ Weapon(y) ∧
  Sells(x,y,z) ∧ Hostile(z) ⇒
  Criminal(x)
Owns(Nono,M_1)
Missile(M_1)
Missile(x) ∧ Owns(Nono,x)
  ⇒ Sells(West,x,Nono)
Missile(x) ⇒ Weapon(x)
Enemy(x,America) ⇒
  Hostile(x)
American(West)
Enemy(Nono,America)

Criminal(West)

Proof that Colonel West is a criminal

BACKWARD CHAINING EXAMPLE



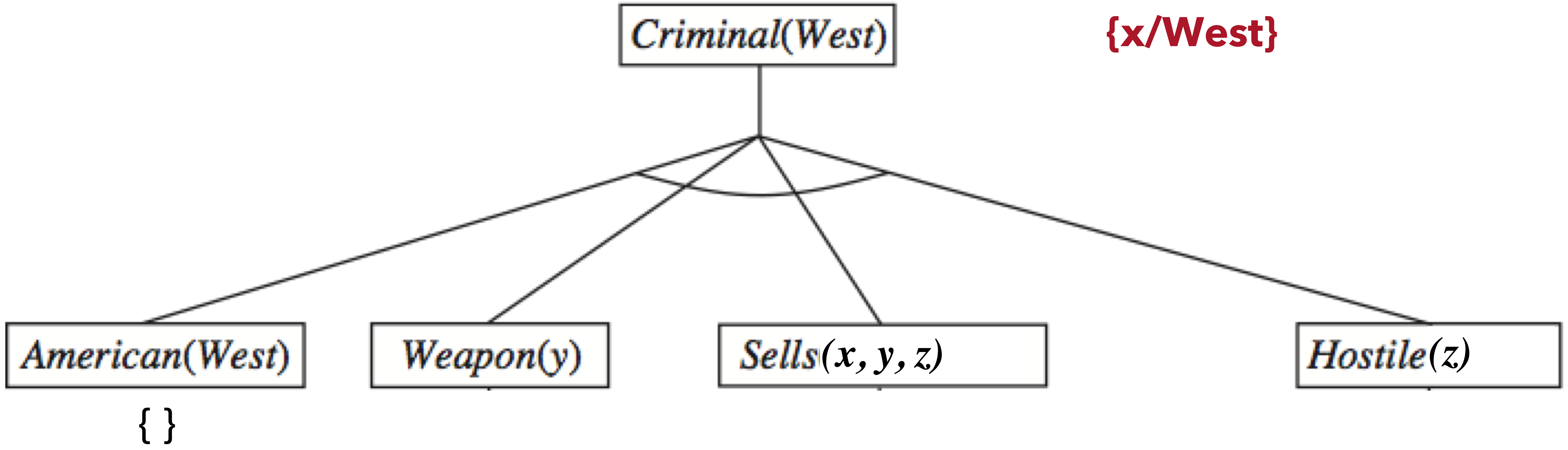
{x/West}

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

American(x) ∧ Weapon(y) ∧ Sells(x,y,z) ∧
Hostile(z) ⇒ Criminal(x)
Owns(Nono,M_1)
Missile(M_1)
Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)
Missile(x) ⇒ Weapon(x)
Enemy(x,America) ⇒ Hostile(x)
American(West)
Enemy(Nono,America)

Proof that Colonel West is a criminal

BACKWARD CHAINING EXAMPLE



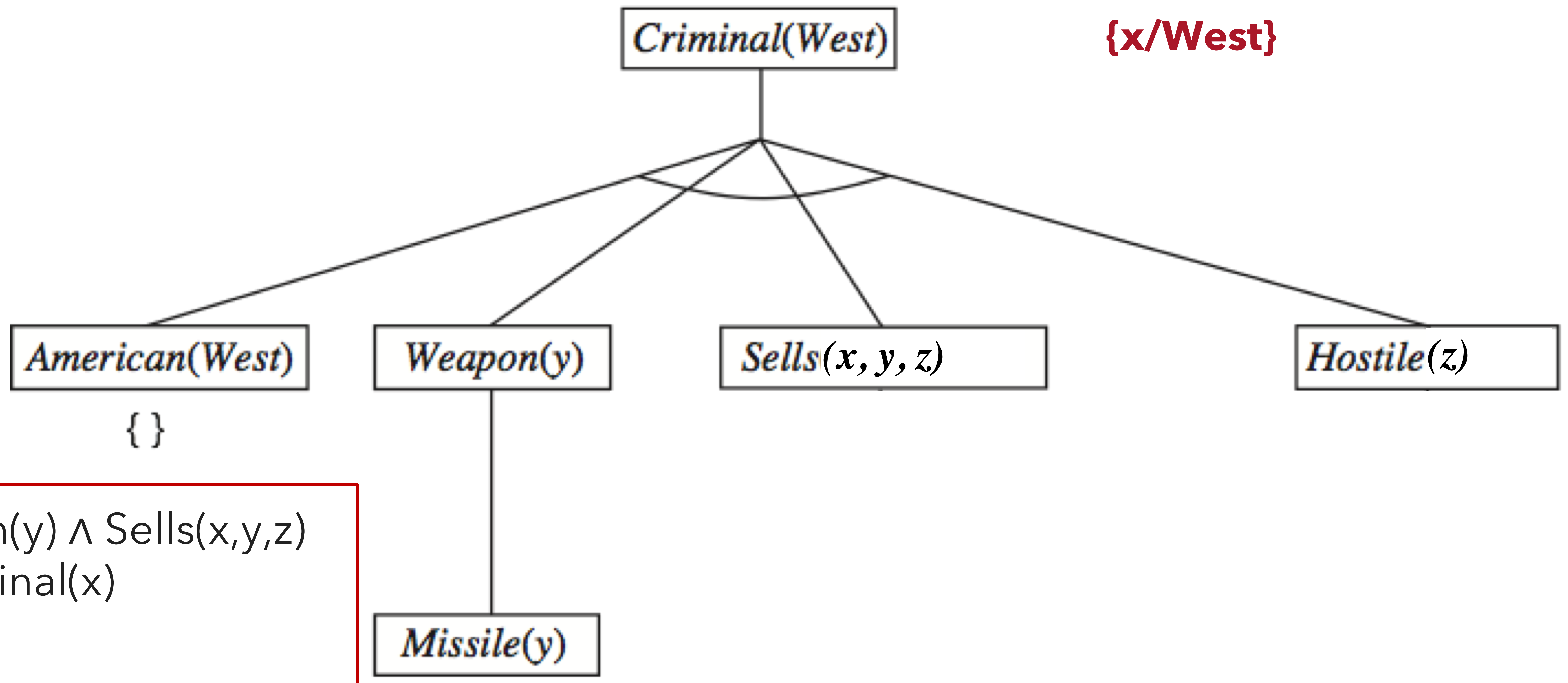
{x/West}

American(x) ∧ Weapon(y) ∧ Sells(x,y,z) ∧ Hostile(z) ⇒ Criminal(x)
Owns(Nono,M_1)
Missile(M_1)
Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)
Missile(x) ⇒ Weapon(x)
Enemy(x,America) ⇒ Hostile(x)
American(West)
Enemy(Nono,America)

American(West)

Proof that Colonel West is a criminal

BACKWARD CHAINING EXAMPLE

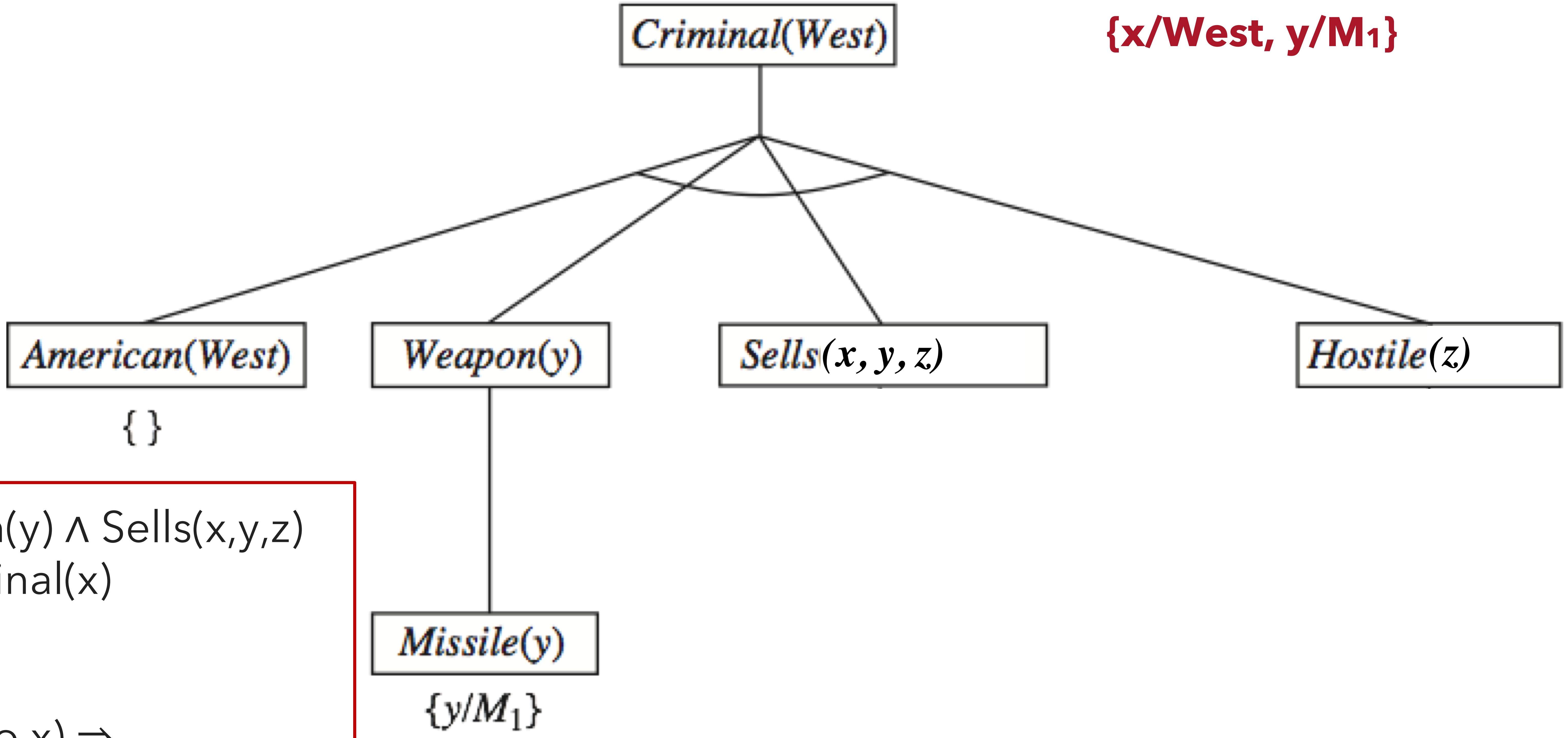


American(x) ∧ Weapon(y) ∧ Sells(x,y,z) ∧ Hostile(z) ⇒ Criminal(x)
Owns(Nono,M_1)
Missile(M_1)
Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)
Missile(x) ⇒ Weapon(x)
Enemy(x,America) ⇒ Hostile(x)
American(West)
Enemy(Nono,America)

Missile(x) ⇒ Weapon(x)

Proof that Colonel West is a criminal

BACKWARD CHAINING EXAMPLE



{x/West, y/M_1}

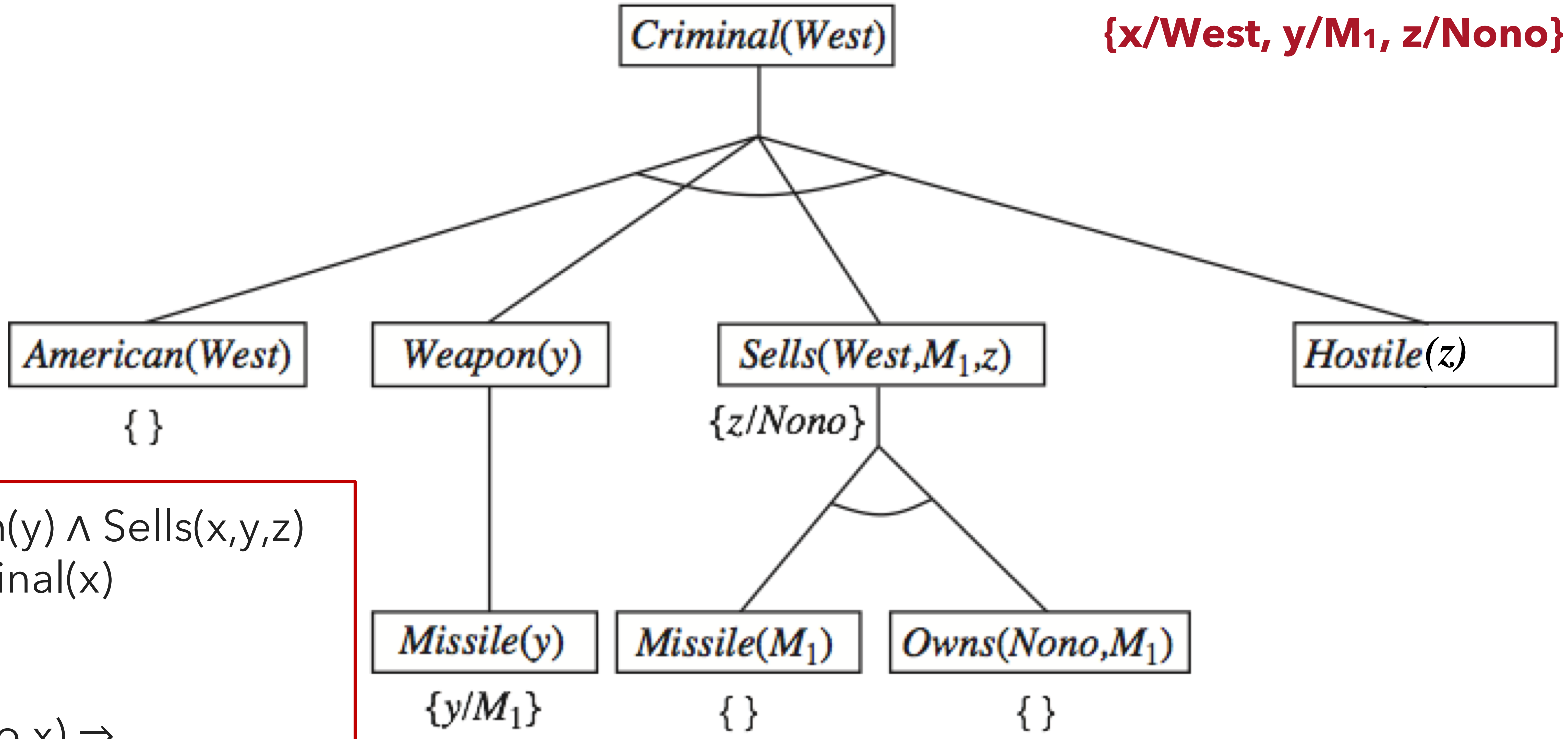
American(x) ∧ Weapon(y) ∧ Sells(x,y,z)
∧ Hostile(z) ⇒ Criminal(x)
Owns(Nono,M_1)
Missile(M_1)
Missile(x) ∧ Owns(Nono,x) ⇒
Sells(West,x,Nono)
Missile(x) ⇒ Weapon(x)
Enemy(x,America) ⇒ Hostile(x)
American(West)
Enemy(Nono,America)

Missile(M_1)

Proof that Colonel West is a criminal

BACKWARD CHAINING EXAMPLE

$\{x/West, y/M_1, z/Nono\}$

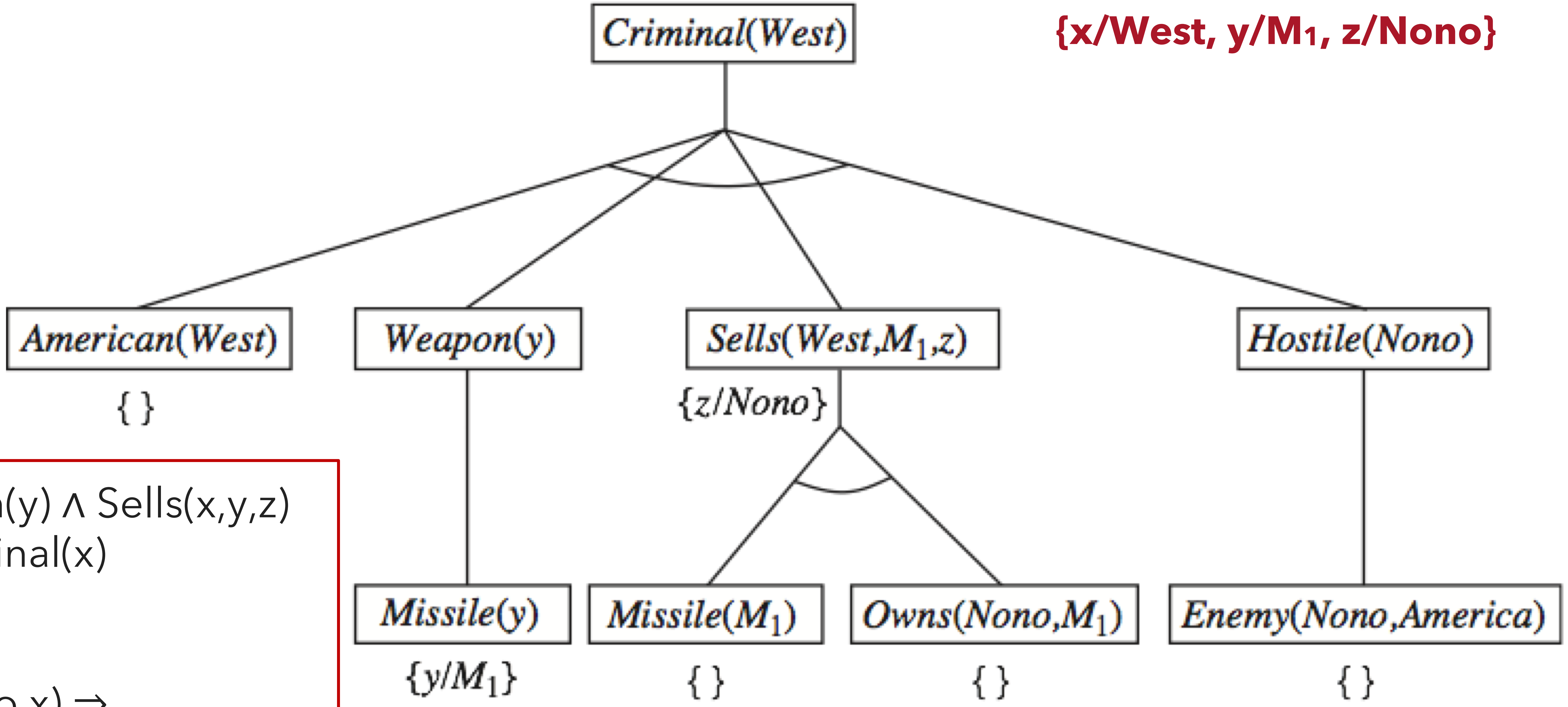


American(x) ∧ Weapon(y) ∧ Sells(x,y,z) ∧ Hostile(z) ⇒ Criminal(x)
Owns(Nono,M_1)
Missile(M_1)
Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)
Missile(x) ⇒ Weapon(x)
Enemy(x,America) ⇒ Hostile(x)
American(West)
Enemy(Nono,America)

Missile(x) ∧ Owns(Nono,x) ⇒ Sells(West,x,Nono)
Owns(Nono,M_1)

Proof that Colonel West is a criminal

BACKWARD CHAINING EXAMPLE



{x/West, y/M_1, z/Nono}

American(x) ∧ Weapon(y) ∧ Sells(x,y,z)
∧ Hostile(z) ⇒ Criminal(x)
Owns(Nono,M_1)
Missile(M_1)
Missile(x) ∧ Owns(Nono,x) ⇒
Sells(West,x,Nono)
Missile(x) ⇒ Weapon(x)
Enemy(x,America) ⇒ Hostile(x)
American(West)
Enemy(Nono,America)

Enemy(x,America) ⇒ Hostile(x)
Enemy(Nono,America)

PROPERTIES OF BACKWARD CHAINING

- Depth-first recursive proof search – space is linear in size of proof
- Incomplete due to infinite loops
  - $\Rightarrow$ fix by checking current goal against every goal on stack
- Inefficient due to repeated subgoals (both success and failure)
  - $\Rightarrow$ fix using caching of previous results (extra space)

LOGIC

- The types of systems we have discussed are commonly called expert systems.
  - Even if you don't realize it, they are being used all around you. For routing airplanes, for supply chain management, for military logistics, etc.
  - Extremely useful for well-known domains, where it is practical to enumerate all the rules in KB
  - Very brittle when there is uncertainty involved. Since the 80s, a probabilistic approach to AI has taken root, and these systems have proven to be extremely robust in the noisy real world
- … But we still don't have a good way for modern probabilistic AI to reason about the world in an abstract way, as is done with logic.

MIDTERM: QUESTION TYPES

- Multiple choice questions
- Short answers: each question can be answered using at most a few sentences/a figure; True or False
- Long answers: application of algorithms

MIDTERM: KEY TOPICS

- Search
  - How to formulate a problem into a search problem?
  - How do different types of search algorithms work? E.g., BFS, DFS, Uniform cost search, depth limited search, iterative deepening depth-first search, bidirectional search, greedy search, A* search…
  - Is a search algorithm complete/optimal? When is it complete/optimal?
  - What is the difference between tree search and graph search?
- Local search
  - How to formulate a problem into a local search problem?
  - How do different types of local search algorithms work?

MIDTERM: KEY TOPICS

- Adversarial search
  - How to formulate a problem into an adversarial search problem?
  - How does the minimax search algorithm work? How does alpha-beta pruning work?
  - How to utilize evaluation functions in adversarial search?
  - What should we do if the game tree includes chance element?
- Constraint satisfaction problem (CSP)
  - How to formulate a problem into a CSP?
  - How to solve CSP using the basic backtracking search algorithm?
  - How to improve the backtracking search algorithm through filtering, ordering, and utilizing the structure of the problem?

MIDTERM: KEY TOPICS

- Logic
  - What does entailment mean? What's the relationship between entailment and models of KB/sentences?
  - How to express a statement in propositional logic/first-order logic?
  - How to conduct inference by enumeration in propositional logic?
  - What is logical equivalence? What does validity and satisfiability mean?
  - How to conduct inference by resolution, forward chaining, or backward chaining in propositional logic/first-order logic?