Artificial Intelligence

Lecturer 6 – First Order Logic Inference – Some Examples

Brigitte Jaumard
Dept of Computer Science and Software Engineering
Concordia University
Montreal (Quebec) Canada

Recall

- **Refutation-based procedure**
  - S |= A if and only if $S \cup \{\neg A\}$ is unsatisfiable
- **Resolution procedure**
  - Transform $S \cup \{\neg A\}$ into a set of clauses
  - Apply Resolution rule to find the empty clause (contradiction)
    - If the empty clause is found
      - Conclude S |= A
    - Otherwise
      - No conclusion

Criminal Problem

- **Problem**
  - 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**
  - West is a criminal

Represent problem as first-order definite clauses

- "... it is a crime for an American to sell weapons to hostile nations"
  - $\forall x$ American(x)∧Weapon(y)∧Sells(x,y,z)∧Hostile(z) ⇒ Criminal(x)
- "Nono has some missiles"
  - ∃ x Owns(Nono, x)∧Missile(x)
  - transformed into
  - Owns(Nono, M1) and Missile(M1)
- …

Conjunctive Normal Form for FOL

- Every sentence of first-order logic can be converted into an inferentially equivalent CNF sentence

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

becomes, in CNF,

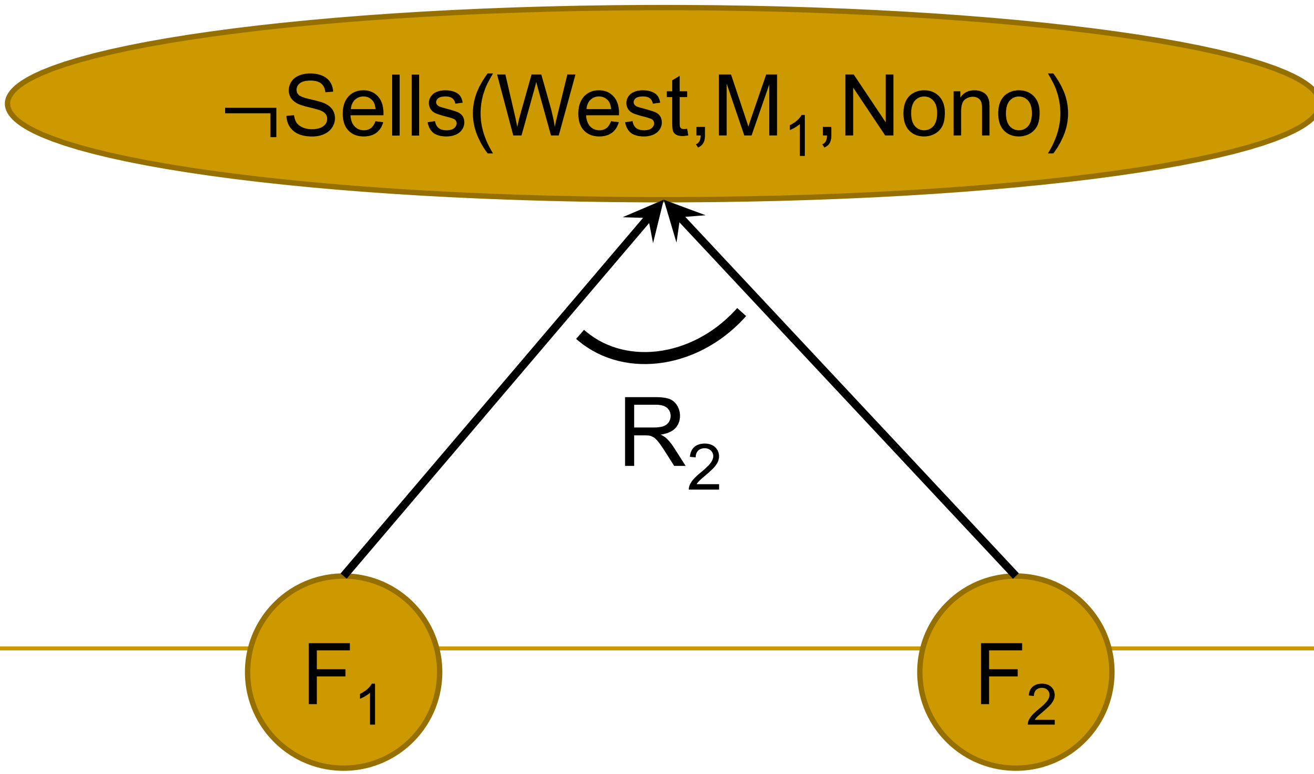
$$\neg American(x) \lor \neg Weapon(y) \lor \neg Sells(x,y,z) \lor \neg Hostile(z) \lor Criminal(x) \;.$$

Sentences in CNF
Resolution Proof that West is Criminal

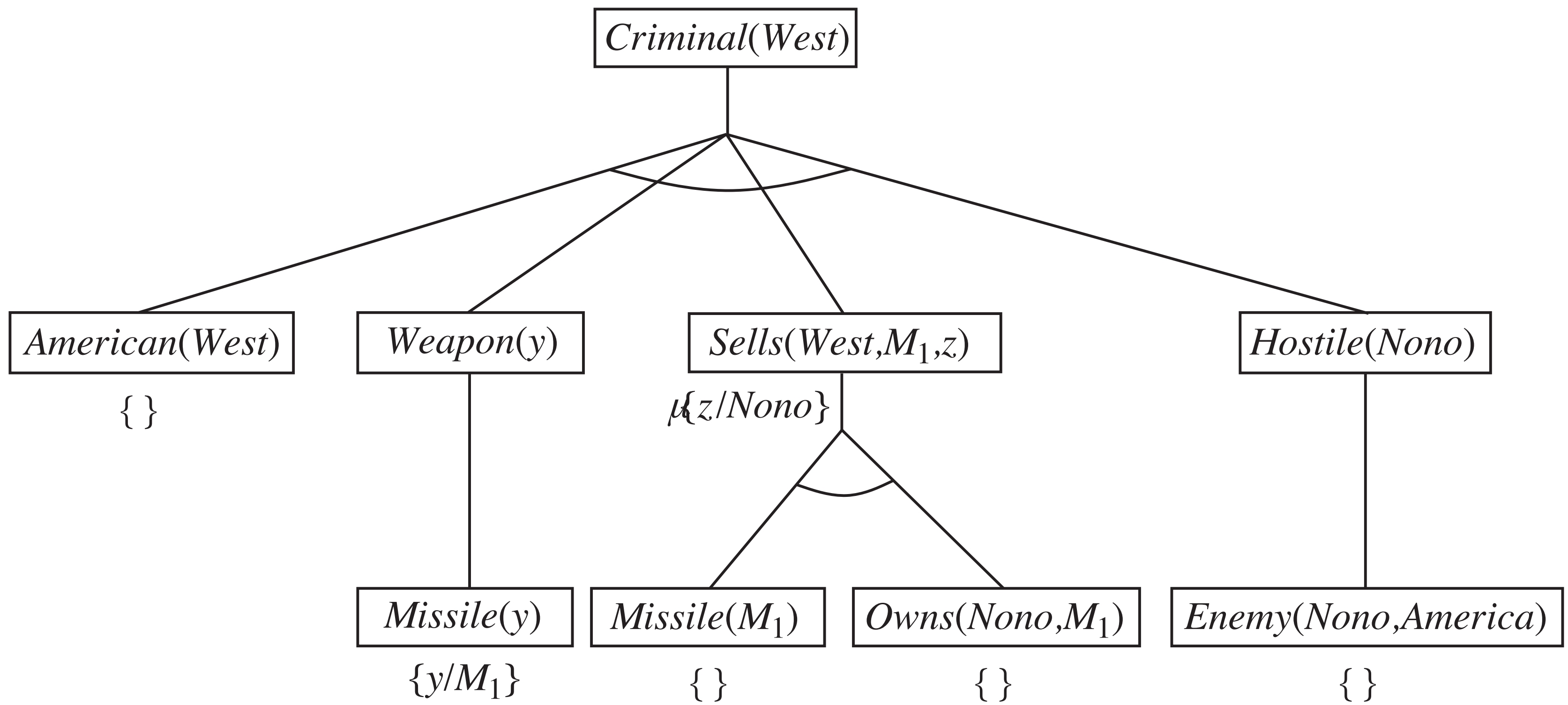
- **C_1:** $\neg$American(x) ∨ $\neg$Weapon(y) ∨ $\neg$Sells(x,y,z) ∨ $\neg$Hostile(z) ∨ Criminal(x)
- **C_2:** $\neg$Missile(x) ∨ $\neg$Owns(Nono,x) ∨ $\neg$Sells(West,x,Nono)
- **C_3:** $\neg$Enemy(x, America) ∨ $\neg$Hostile(x)
- **C_4:** $\neg$Missile(x) ∨ Weapon(x)
- **F_1:** Owns (Nono, M_1)
- **F_2:** American(West)
- **F_3:** Missile(M_1)
- **F_4:** Enemy(Nono, America)
- **Negated Goal**: $\neg$ Criminal(West)

Forward Chaining

- **F_1:** Owns (Nono, M_1)
- **F_3:** Missile(M_1)
- **C_2:** $\neg$Missile(x) $\vee$ $\neg$Owns(Nono,x) $\vee$ $\neg$Sells(West,x,Nono)
- **R_2:** Missile(x) $\wedge$ Owns(Nono,x) $\Rightarrow$ $\neg$Sells(West,x,Nono)

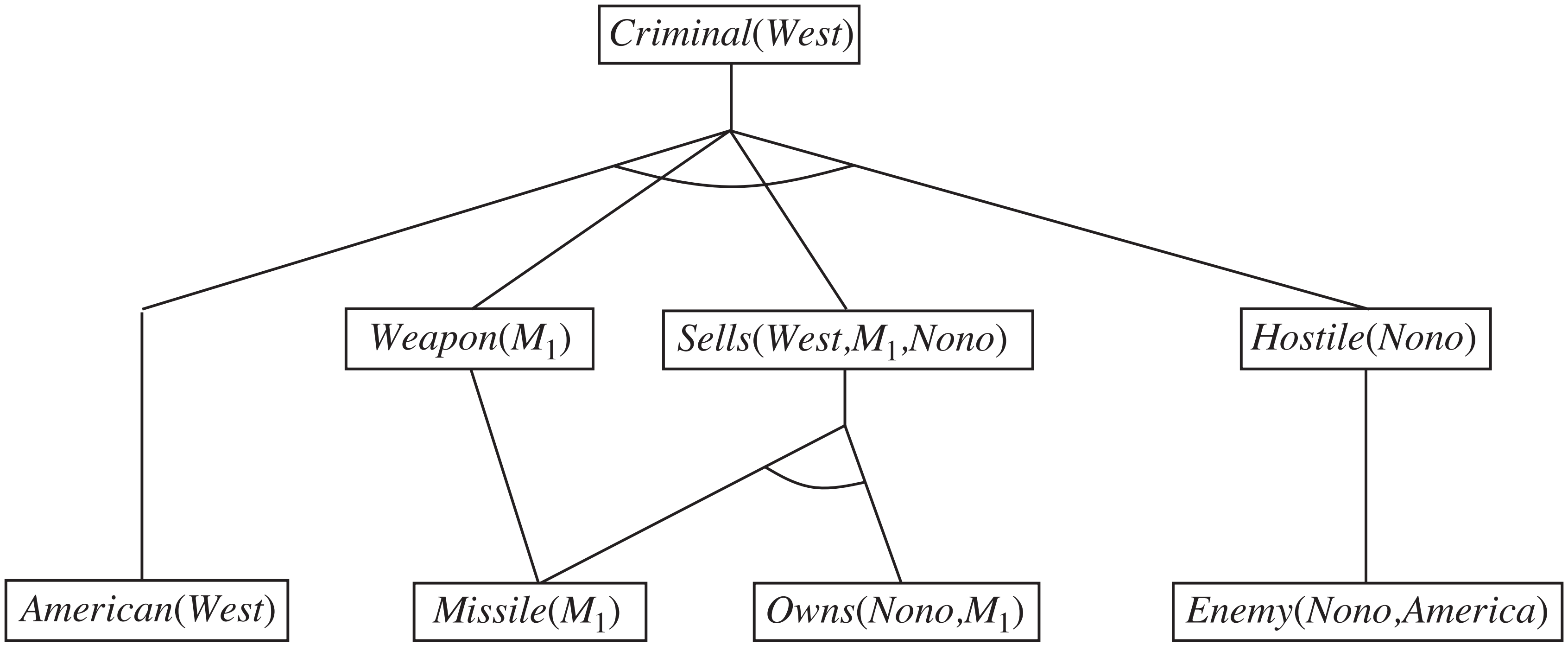


Backward Chaining



- Proof tree constructed by backward chaining to prove that West is a criminal
- 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
  - 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.
- By the time FOL-BC gets to the last conjunct, originally Hostile(z), z is already bound to Nono.

Forward Chaining

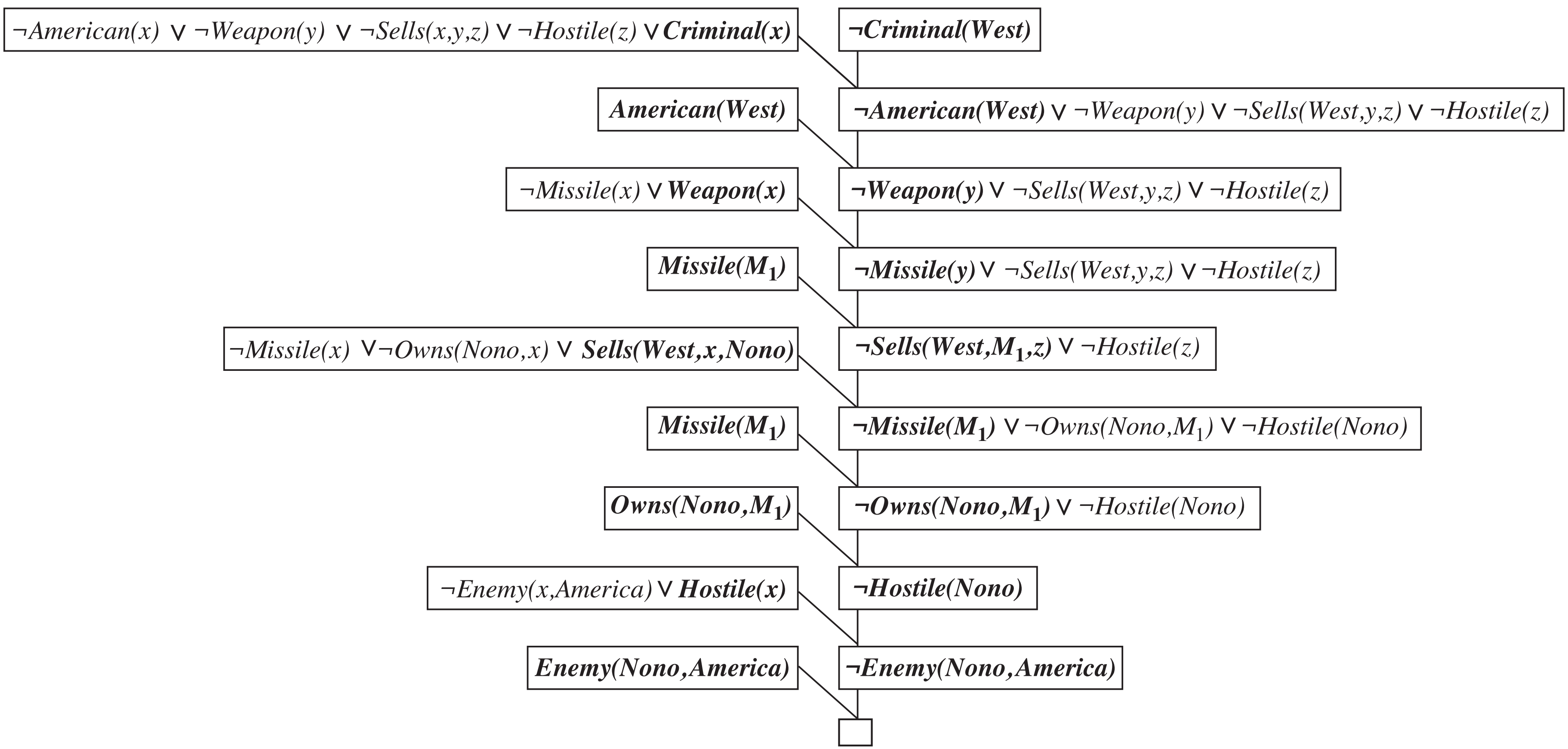


- **Proof tree** generated by forward chaining on the crime example
- **Initial facts** appear at the bottom level
- **Facts inferred on the first iteration** in the middle level
- **Facts inferred on the second iteration** at the top level.

Example of Proof by Resolution

- **Negated Goal**: ¬ Criminal(West)
- **C_1:** ¬American(x) ∨ ¬Weapon(y) ∨ ¬Sells(x,y,z) ∨ ¬Hostile(z) ∨ Criminal(x)
- **Resolution Principle**
  - mgu: most general unifier
  - x / West
- **Conclusion**
  - ¬American(West) ∨ ¬Weapon(y) ∨ ¬Sells(West,y,z) ∨ ¬Hostile(z)

Crime: Resolution Proof



Curiosity killed the cat? Original sentences

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

B. $\forall x \; [\exists z \;\; Animal(z) \wedge Kills(x,z)] \Rightarrow [\forall y \;\; \neg Loves(y,x)]$

C. $\forall x \;\; Animal(x) \Rightarrow Loves(Jack, x)$

D. $Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna)$

E. $Cat(Tuna)$

F. $\forall x \;\; Cat(x) \Rightarrow Animal(x)$

$\neg$G. $\neg Kills(Curiosity, Tuna)$

Curiosity killed the cat? Original sentences: their conversion

A1. $Animal(F(x)) \vee Loves(G(x), x)$

A2. $\neg Loves(x, F(x)) \vee Loves(G(x), x)$

B. $\neg Loves(y, x) \vee \neg Animal(z) \vee \neg Kills(x, z)$

C. $\neg Animal(x) \vee Loves(Jack, x)$

D. $Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna)$

E. $Cat(Tuna)$

F. $\neg Cat(x) \vee Animal(x)$

¬G. $\neg Kills(Curiosity, Tuna)$

Explanations

- **Eliminate implications**:

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

- **Move $\neg$ inwards**: In addition to the usual rules for negated connectives, we need rules for negated quantifiers. Thus, we have

$$\begin{array}{lll} \neg\forall\, x \;\; p & \text{becomes} & \exists\, x \;\; \neg p \\ \neg\exists\, x \;\; p & \text{becomes} & \forall\, x \;\; \neg p \;. \end{array}$$

  Our sentence goes through the following transformations:

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

  Notice how a universal quantifier ($\forall\, y$) in the premise of the implication has become an existential quantifier. The sentence now reads "Either there is some animal that x doesn't love, or (if this is not the case) someone loves x." Clearly, the meaning of the original sentence has been preserved.

- **Standardize variables**: For sentences like $(\exists\, x\, P(x)) \vee (\exists\, x\, Q(x))$ which use the same variable name twice, change the name of one of the variables. This avoids confusion later when we drop the quantifiers. Thus, we have

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

Rxplanations (Cont'd)

- **Skolemize**: **Skolemization** is the process of removing existential quantifiers by elimination. In the simple case, it is just like the Existential Instantiation rule of Section 9.1: translate $\exists\, x\; P(x)$ into $P(A)$, where A is a new constant. However, we can't apply Existential Instantiation to our sentence above because it doesn't match the pattern $\exists\, v\; \alpha$; only parts of the sentence match the pattern. If we blindly apply the rule to the two matching parts we get

  $$\forall\, x \;\; [Animal(A) \land \neg Loves(x, A)] \lor Loves(B, x) \;,$$

  which has the wrong meaning entirely: it says that everyone either fails to love a particular animal A or is loved by some particular entity B. In fact, our original sentence allows each person to fail to love a different animal or to be loved by a different person. Thus, we want the Skolem entities to depend on x and z:

  $$\forall\, x \;\; [Animal(F(x)) \land \neg Loves(x, F(x))] \lor Loves(G(z), x) \;.$$

Explanations (Cont'd)

- **Drop universal quantifiers**: At this point, all remaining variables must be universally quantified. Moreover, the sentence is equivalent to one in which all the universal quantifiers have been moved to the left. We can therefore drop the universal quantifiers:

$$[Animal(F(x)) \land \neg Loves(x, F(x))] \lor Loves(G(z), x) \ .$$

- **Distribute** $\lor$ **over** $\land$:

$$[Animal(F(x)) \lor Loves(G(z), x)] \land [\neg Loves(x, F(x)) \lor Loves(G(z), x)] \ .$$

Conversion

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

B. $\forall x\ [\exists z\ \ Animal(z) \land Kills(x,z)] \Rightarrow [\forall y\ \ \neg Loves(y,x)]$

C. $\forall x\ \ Animal(x) \Rightarrow Loves(Jack,x)$

D. $Kills(Jack, Tuna) \lor Kills(Curiosity, Tuna)$

E. $Cat(Tuna)$

F. $\forall x\ \ Cat(x) \Rightarrow Animal(x)$

¬G. $\neg Kills(Curiosity, Tuna)$

A1. $Animal(F(x)) \lor Loves(G(x),x)$

A2. $\neg Loves(x,F(x)) \lor Loves(G(x),x)$

B. $\neg Loves(y,x) \lor \neg Animal(z) \lor \neg Kills(x,z)$

C. $\neg Animal(x) \lor Loves(Jack,x)$

D. $Kills(Jack, Tuna) \lor Kills(Curiosity, Tuna)$

E. $Cat(Tuna)$

F. $\neg Cat(x) \lor Animal(x)$

¬G. $\neg Kills(Curiosity, Tuna)$

Summary of Resolution

- Refutation-based procedure
  - $S \models A$ if and only if $S \cup \{\neg A\}$ is unsatisfiable
- Resolution procedure
  - Transform $S \cup \{\neg A\}$ into a set of clauses
  - Apply Resolution rule to find a the empty clause (contradiction)
    - If the empty clause is found
      - Conclude $S \models A$
    - Otherwise
      - No conclusion

Summary of Resolution

- Theorem
  - A set of clauses S is unsatisfiable if and only if upon the input S, Resolution procedure finds the empty clause (after a finite time).

Exercice

- 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
- Is West a criminal?

Exercice

- Jack owns a dog *own(Jack, dog)*
- Every dog owner is an animal lover
- No animal lover kills an animal
- Either Jack or Curiosity killed the cat, who is named Tuna
- Did Curiosity kill the cat?

Exercice

- Jack owns a dog *Dog(x) Owns(Jack, dog)*
  - $\exists x \; \mathrm{dog}(x) \wedge \mathrm{Owns}(\mathrm{Jack}, \mathrm{dog})$
- Every dog owner is an animal lover
- No animal lover kills an animal
- Either Jack or Curiosity killed the cat, who is named Tuna
- Did Curiosity kill the cat?

Exercice

- Jack owns a dog *Dog(x) Owns(Jack, dog)*
  - $\exists x \; \mathrm{dog}(x) \wedge \mathrm{Owns}(\mathrm{Jack}, \mathrm{dog})$
- Every dog owner is an animal lover
  - $\forall x \forall y \; \left(\mathrm{dog}(y) \wedge \mathrm{Owns}(x, y)\right) \Rightarrow \mathrm{AnimalLover}(x)$
- No animal lover kills an animal
- Either Jack or Curiosity killed the cat, who is named Tuna
- Did Curiosity kill the cat?

Exercice

- Jack owns a dog *Dog(x) Owns(Jack, dog)*
  - $\exists x \; \text{dog}(x) \wedge \text{Owns}(\text{Jack}, \text{dog})$
- Every dog owner is an animal lover
  - $\forall x \forall y \; \big(\text{dog}(y) \wedge \text{Owns}(x, y)\big) \Rightarrow \text{AnimalLover}(x)$
- No animal lover kills an animal
  - $\forall x \forall y \; \text{AnimalLover}(x) \wedge \text{Animal}(y) \Rightarrow \neg \, \text{Kills}(x, y)$
- Either Jack or Curiosity killed the cat, who is named Tuna
- Did Curiosity kill the cat?

Exercice

- Jack owns a dog *Dog(x) Owns(Jack, dog)*
  - $\exists x \; \text{dog}(x) \land \text{Owns}(\text{Jack}, \text{dog})$
- Every dog owner is an animal lover
  - $\forall x \forall y \; \big(\text{dog}(y) \land \text{Owns}(x, y)\big) \Rightarrow \text{AnimalLover}(x)$
- No animal lover kills an animal
  - $\forall x \forall y \; \text{AnimalLover}(x) \land \text{Animal}(y) \Rightarrow \neg \, \text{Kills}(x, y)$
- Either Jack or Curiosity killed the cat, who is named Tuna
  - Kills(Jack, Tuna) ∨ Kills(Curiosity, Tuna)

Exercice

- Jack owns a dog *Dog(x) Owns(Jack, dog)*
  - $\exists x \; \text{dog}(x) \wedge \text{Owns}(\text{Jack}, \text{dog})$
- Every dog owner is an animal lover
  - $\forall x \forall y \; \big(\text{dog}(y) \wedge \text{Owns}(x, y)\big) \; \Rightarrow \text{AnimalLover}(x)$
- No animal lover kills an animal
  - $\forall x \forall y \; \text{AnimalLover}(x) \wedge \text{Animal}(y) \Rightarrow \neg \, \text{Kills}(x, y)$
- Either Jack or Curiosity killed the cat, who is named Tuna
  - Kills(Jack, Tuna) ∨ Kills(Curiosity, Tuna)

Transform the problem to set of clauses

$$Dog(D)$$

$$Owns(Jack, D)$$

$$\neg Dog(y) \vee \neg Owns(x, y) \vee AnimalLover(x)$$

$$\neg AnimalLover(x) \wedge \neg Animal(y) \vee \neg Kills(x, y)$$

$$Kills(Jack, Tuna) \vee Kill(Curiosity, Tuna)$$

$$Cat(Tuna)$$

$$\neg Cat(x) \vee Animal(x)$$

$\neg$Kills(Curiosity,Tuna)

Reading and Suggested Exercises

- Chapter 9
- Exercises: 9.9, 9.11, 9.19, 9.24