Artificial Intelligence

Lecturer 10 – First Order Logic

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

Formal Languages and their ontological and epistemological commitment

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief $\in [0, 1]$
Fuzzy logic	facts with degree of truth $\in [0, 1]$	known interval value

First Order Logic

- Syntax
- Semantic
- Inference
  - Resolution

First Order Logic (FOL)

- First Order Logic is about
  - Objects
  - Relations
  - Facts
- The world is made of objects
  - ***Objects*** are things with individual identities and properties to distinguish them
  - Various ***relations*** hold among objects. Some of these relations are functional
  - Every ***fact*** involving objects and their relations are either *true* or *false*

FOL Syntax

- Symbols
  - Variables: x, y, z,…
  - Constants: a, b, c, …
  - Function symbols (with arities): f, g, h, …
  - Relation symbols (with arities): p, r, r
  - Logical connectives: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
  - Quantifiers: $\exists, \forall$

FOL Syntax

- Variables, constants and function symbols are used to build **terms**
  - X, Bill, FatherOf(X), …
- Relations and terms are used to build **predicates**
  - Tall(FatherOf(Bill)), Odd(X), Married(Tom,Marry), Loves(Y,MotherOf(Y)), …
- Predicates and logical connective are used to build **sentences**
  - Even(4), $\forall$ X. Even(X) $\Rightarrow$ Odd(X+1), $\exists$ X. X > 0

FOL Formal grammar

$$
\begin{array}{rcl}
\textit{Sentence} & \rightarrow & \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
\textit{AtomicSentence} & \rightarrow & \textit{Predicate} \mid \textit{Predicate}(\textit{Term}, \ldots) \mid \textit{Term} = \textit{Term} \\
\textit{ComplexSentence} & \rightarrow & (\ \textit{Sentence}\ ) \mid [\ \textit{Sentence}\ ] \\
& \mid & \neg\ \textit{Sentence} \\
& \mid & \textit{Sentence} \land \textit{Sentence} \\
& \mid & \textit{Sentence} \lor \textit{Sentence} \\
& \mid & \textit{Sentence} \ \Rightarrow\ \textit{Sentence} \\
& \mid & \textit{Sentence} \ \Leftrightarrow\ \textit{Sentence} \\
& \mid & \textit{Quantifier}\ \textit{Variable}, \ldots\ \textit{Sentence} \\
\\
\textit{Term} & \rightarrow & \textit{Function}(\textit{Term}, \ldots) \\
& \mid & \textit{Constant} \\
& \mid & \textit{Variable} \\
\\
\textit{Quantifier} & \rightarrow & \forall \mid \exists \\
\textit{Constant} & \rightarrow & A \mid X_1 \mid \textit{John} \mid \cdots \\
\textit{Variable} & \rightarrow & a \mid x \mid s \mid \cdots \\
\textit{Predicate} & \rightarrow & \textit{True} \mid \textit{False} \mid \textit{After} \mid \textit{Loves} \mid \textit{Raining} \mid \cdots \\
\textit{Function} & \rightarrow & \textit{Mother} \mid \textit{LeftLeg} \mid \cdots \\
\textsc{Operator Precedence} & : & \neg, =, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

FOL Syntax: Terms

- A **term** is a logical expression that refers to an object.
- Variables are terms
- Constants are terms
- If $t_1, \ldots, t_n$ are terms and f is a function symbol with arity n then $f(t_1, \ldots, t_n)$ is a **term**
- **Example**
  - *LeftLeg(John)*

FOL Syntax: Atomic Sentence

- If $t_1, \ldots, t_n$ are terms and p is a relation symbol with arity n then $p(t_1, \ldots, t_n)$ is a predicate
- **Examples**
  - *Brother(Richard,John)*
  - *Married(Father(Richard), Mother(John))*
- *An **atomic sentence** is **true** in a given model if the relation referred to by the **predicate** symbol holds among the objects referred to by the arguments.*

FOL Syntax: Complex Sentences

- True, False are sentences
- Predicates are sentences
- Examples

$$\neg Brother(LeftLeg(Richard), John)$$
$$Brother(Richard, John) \wedge Brother(John, Richard)$$
$$King(Richard) \vee King(John)$$
$$\neg King(Richard) \;\Rightarrow\; King(John)\;.$$

Quantifiers

- Universal Quantification ($\forall$)
  - "All kings are persons" translates into
  - $\forall x \quad King(x) \Rightarrow Person(x)$
  - Not to be confused with $\forall x \ King(x) \land Person(x)$
- Existential Quantification ($\exists$)
  - "King John has a crown on his head" translates into
  - $\exists x \quad Crown(x) \land On\ Head(x, John)$
  - Not to be confused with $\exists x \quad Crown(x) \Rightarrow On\ Head(x, John)$

Nested Quantifiers

- Brothers are sibling
  - $\forall x\ \forall y$ *Brother(x, y)* $\Rightarrow$ *Sibling (x, y)*
- Consecutive quantifiers
  - $\forall x, y$ *Sibling(x, y)* $\Rightarrow$ *Sibling (y, x)*
- Everybody loves somebody: for every person, there is someone that person loves
  - $\forall x\ \exists y$ *Loves(x, y)*
- There is someone who is loved by everyone
  - $\exists y\ \forall x$ *Loves(x, y)*
- Confusion
  - $\forall x\ (Crown(x) \lor (\exists x$ *Brother (Richard,x)))*
  - $\forall x\ (Crown(x) \lor (\exists z$ *Brother (Richard.z)))*

Connections between ∃ and ∀

$$
\begin{aligned}
\forall x \ \ \neg P &\equiv \neg\exists x \ \ P \\
\neg\forall x \ \ P &\equiv \exists x \ \ \neg P \\
\forall x \ \ P &\equiv \neg\exists x \ \ \neg P \\
\exists x \ \ P &\equiv \neg\forall x \ \ \neg P
\end{aligned}
$$

$$
\begin{aligned}
\neg(P \vee Q) &\equiv \neg P \wedge \neg Q \\
\neg(P \wedge Q) &\equiv \neg P \vee \neg Q \\
P \wedge Q &\equiv \neg(\neg P \vee \neg Q) \\
P \vee Q &\equiv \neg(\neg P \wedge \neg Q)
\end{aligned}
$$

Example

How do you translate "There are exactly two apples" into First Order Logic? Consider the domain of the variables to be the entire universe (everything).

(a) $[\exists x\, \exists y\, (\textsc{Apple}(x) \wedge \textsc{Apple}(y)] \wedge [\forall z \Big(\textsc{Apple}(z) \rightarrow \big((z = x) \vee (z = y)\big)\Big)]$

(b) $\exists x\, \exists y\, \Big(\textsc{Apple}(x) \wedge \textsc{Apple}(y) \wedge [x \neq y] \wedge [\forall z\big(\textsc{Apple}(z) \rightarrow (z = x \vee z = y)\big)]\Big)$

(c) $\exists x\, \exists y \Big([x \neq y] \wedge [\forall z \big(\textsc{Apple}(z) \leftrightarrow (z = x \vee z = y)\big)]\Big)$

(d) $\exists x \exists y \Big(\textsc{Apple}(x) \wedge \textsc{Apple}(y) \wedge x \neq y \Big) \wedge \Big[\forall x \forall y \forall z \Big(\textsc{Apple}(x) \wedge \textsc{Apple}(y) \wedge \textsc{Apple}(z) \Big) \rightarrow \Big(x = y \vee x = z \vee y = z \Big) \Big]$

(e) (b) and (d)

(f) All of the above

Answer

(a) says that there is an apple x and an apple y such that every apple is identical to either x or y. But it does not guarantee that x and y are two distinct apples. Since (a) allows that $x = y$, (a) comes out true even if there is only one apple. So (a) is incorrect.
But (b) and (d) are all adequate translations. (b) is like (a) except that it adds the non-identity clause that (a) lacks.
(c) says that there are distinct objects such that anything is an apple if and only if it is identical to one or the other of them. However, there is no guarantee that we have both $\textsc{Apple}(x)$ and $\textsc{Apple}(y)$
(d) is a conjunction of "There are at least two apples" and "There are at most two apples". Some simple math shows that (d) means that there are exactly two apples. Therefore (e) is the correct answer.

Reading and Suggested Exercises

- Chapter 8
- Exercises: 8.9, 8.11, 8.19

Inference in FOL

- Difficulties
  - Quantifiers
  - Infinite sets of terms
  - Infinite sets of sentences
- Examples: $\forall x. King(x) \wedge Greedy(x) \Rightarrow Evil(x)$
  - Infinite set of instances

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

$$King(FatherOf(Bill)) \wedge Greedy(FatherOf(Bill)) \Rightarrow Evil(FatherOf(Bill))$$

$\ldots$

Robinson's Resolution

- Herbrand's Theorem (~1930)
  - A set of sentences S is unsatisfiable if and only there exists a finite subset S_g of the set of all ground instances Gr(S), which is unsatisfiabe
- Herbrand showed that there is a procedure to demonstrate the unsatisfiability of a unsatisfiable set of sentences
- Robinson propose the Resolution procedure (~1950)

Idea 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 the empty clause (contradiction)
    - If the empty clause is found
      - Conclude $S \models A$
    - Otherwise
      - No conclusion

Clause

- A clause is a disjunction of literals, i.e., has the form

$$P_1 \vee P_2 \vee \ldots \vee P_n \qquad\qquad P_i \equiv [\neg]R_i$$

  - Example

$$P(x) \vee Q(x,a) \vee R(b)$$

$$P(y) \vee \neg Q(b,y) \vee R(y)$$

- The empty clause corresponds to a contradiction
- Any sentence can be transformed to an equi-satisfiable set of clauses

Elements of Resolution

- Resolution rule
- Unification
- Transform a sentence to a set of clauses

Resolution rule

- Resolution rule

$$\frac{A \vee B \qquad \neg C \vee D}{\theta(A \vee D)} \qquad \theta = mgu(B, C)$$

  - mgu: most general unifier
    - The most general assignment of variables to terms in such a way that two terms are equal
    - Syntactical unification algorithm
  - θ: substitution

Example of Resolution rule

- x, y are variables
- a, b are constants

$$\frac{P(x) \vee Q(x,a) \qquad \neg Q(b,y) \vee R(y)}{P(b) \vee R(a)} \qquad \theta = \{x = b, y = a\}$$

$$A \equiv P(x)$$

$$B \equiv Q(x,a)$$

$$C \equiv Q(b,y)$$

$$D \equiv R(y)$$

Example of Resolution rule

$$\frac{\neg Pet(Joe) \vee Cat(Joe) \vee \overset{\downarrow}{Bird}(Joe) \qquad Parrot(x) \vee \neg \overset{\downarrow}{Bird}(x)}{\neg Pet(Joe) \vee Cat(Joe) \vee Parrot(Joe)} \quad (1)$$

(1) $mgu(Bird(x), Bird(Joe)) = \{x/Joe\}$

$$\frac{\neg \overset{\downarrow}{On}(x,y) \vee Above(x,y) \qquad On(B,A) \vee \overset{\downarrow}{On}(A,B)}{Above(A,B) \vee On(B,A)} \quad (2)$$

(2) $mgu(On(x,y), On(A,B)) = \{x/A,\ y/B\}$

$$\frac{\neg Bird(x) \vee \overset{\downarrow}{Feathers}(x) \qquad \neg \overset{\downarrow}{Feathers}(y) \vee Flies(y)}{\neg Bird(x) \vee Flies(x)} \quad (3)$$

(3) $mgu(Feathers(x), Feathers(y)) = \{y/x\}$

Elements of Resolution

- Resolution rule
- Unification
- Transform a sentence to a set of clauses

Unification

- Input
  - Set of equalities between two terms
- Output
  - Most general assignment of variables that satisfies all equalities
  - Fail if no such assignment exists

Unification algorithm

Decompose

$$U \cup \{f(t_1, \ldots, t_n) =^? f(s_1, \ldots, s_n)\} \longrightarrow U \cup \{t_1 =^? s_1, \ldots, t_n =^? s_n\}$$

Orient.

$$U \cup \{t =^? v\} \longrightarrow U \cup \{v =^? t\}$$

Delete.

$$U \cup \{v =^? v\} \longrightarrow U$$

Eliminate.

$$U \cup \{v =^? t\},\ v \in \mathcal{V}ars(U) \setminus \mathcal{V}ars(t) \longrightarrow U[v/t] \cup \{v =^? t\}$$

Mismatch.

$$U \cup \{f(t_1, \ldots, t_m) =^? g(s_1, \ldots, s_n)\},\ f, g \text{ distinct or } m \neq n \longrightarrow FAIL$$

Occurs.

$$U \cup \{v =^? t\},\ v \neq t \text{ but } v \in \mathcal{V}ars(t) \longrightarrow FAIL$$

- Vars(U), Vars(t) are sets of variables in U and t
- v is a variable
- s and t are terms
- f and g are function symbols

Example of Unification

$$\{\underline{F(G(H(y)),H(A)) =^? F(G(x),x)}\} \xrightarrow{\text{Decompose}}$$

$$\{\underline{G(H(y)) =^? G(x)},\ H(A) =^? x\} \xrightarrow{\text{Decompose}}$$

$$\{\underline{H(y) =^? x},\ H(A) =^? x\} \xrightarrow{\text{Orient}}$$

$$\{\underline{x =^? H(y)},\ H(A) =^? x\} \xrightarrow{\text{Eliminate } x}$$

$$\{x =^? H(y),\ \underline{H(A) =^? H(y)}\} \xrightarrow{\text{Decompose}}$$

$$\{x =^? H(y),\ \underline{A =^? y}\} \xrightarrow{\text{Orient}}$$

$$\{x =^? H(y),\ \underline{y =^? A}\} \xrightarrow{\text{Eliminate } y}$$

$$\{x =^? H(A),\ y =^? A\}$$

Elements of Resolution

- Resolution rule
- Unification
- Transform a sentence to a set of clauses

Transform a sentence to a set of clauses

1. Eliminate implication
2. Move negation inward
3. Standardize variable scope
4. Move quantifiers outward
5. Skolemize existential quantifiers
6. Eliminate universal quantifiers
7. Distribute and, or
8. Flatten and, or
9. Eliminate and

Eliminate implication

$$\{\forall x\ (\forall y\ P(x,y)) \to \neg(\forall y\ Q(x,y) \to R(x,y))\}$$

$\alpha \to \beta$	$\longrightarrow$	$\neg\alpha \vee \beta$
$\alpha \leftrightarrow \beta$	$\longrightarrow$	$(\neg\alpha \vee \beta) \wedge (\neg\beta \vee \alpha)$

$$\{\forall x\ \neg(\forall y\ P(x,y)) \vee \neg(\forall y\ \neg Q(x,y) \vee R(x,y)))\}$$

Move negation inward

$$\{\forall x \, \neg(\forall y \, P(x,y)) \vee \neg(\forall y \, \neg Q(x,y) \vee R(x,y))\}$$

$$\begin{array}{lcl}
\neg\neg\alpha & \longrightarrow & \alpha \\
\neg(\alpha \vee \beta) & \longrightarrow & \neg\alpha \wedge \neg\beta \\
\neg(\alpha \wedge \beta) & \longrightarrow & \neg\alpha \vee \neg\beta
\end{array}
\qquad
\begin{array}{lcl}
\neg\forall v \, \alpha & \longrightarrow & \exists v \, \neg\alpha \\
\neg\exists v \, \alpha & \longrightarrow & \forall v \, \neg\alpha
\end{array}$$

$$\{\forall x \, (\exists y \, \neg P(x,y)) \vee (\exists y \, Q(x,y) \wedge \neg R(x,y))\}$$

Standardize variable scope

$$\{\forall x\ (\exists y\ \neg P(x,y)) \vee (\exists y\ Q(x,y) \wedge \neg R(x,y))\}$$

Each variable for each quantifier

$$\{\forall x\ (\exists y\ \neg P(x,y)) \vee (\exists z\ Q(x,z) \wedge \neg R(x,z))\}$$

Move quantifiers outward

$$\{\forall x\ (\exists y\ \neg P(x,y)) \vee (\exists z\ Q(x,z) \wedge \neg R(x,z))\}$$

$$\begin{array}{lclclcl} (Qx\ \alpha) \wedge \beta & \longrightarrow & Qx\ (\alpha \wedge \beta) & \quad & \alpha \wedge (Qx\ \beta) & \longrightarrow & Qx\ (\alpha \wedge \beta) \\ (Qx\ \alpha) \vee \beta & \longrightarrow & Qx\ (\alpha \vee \beta) & \quad & \alpha \vee (Qx\ \beta) & \longrightarrow & Qx\ (\alpha \vee \beta) \end{array}$$

$$\{\forall x\ \exists y\ \exists z\ \neg P(x,y) \vee (Q(x,z) \wedge \neg R(x,z))\}$$

Existential Instantiation

$$\{\forall x\, \exists y\, \exists z\, \neg P(x,y) \vee (Q(x,z) \wedge \neg R(x,z))\}$$

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

$$\{\ \forall x\ \neg P(x,a) \vee (Q(x,b) \wedge \neg R(x,b)\ \}$$

Skolemize existential quantifiers

$$\{\forall x\, \exists y\, \exists z\, \neg P(x, y) \vee (Q(x, z) \wedge \neg R(x, z))\}$$

$$\exists v\, \alpha \quad \longrightarrow \quad \alpha[v/\pi(v_1, \ldots, v_n)]$$

with π *new* and $v_1, \ldots, v_n$ universally quantified outside $\exists v\, \alpha$

$$\{\forall x\, \neg P(x, F_1(x)) \vee (Q(x, F_2(x)) \wedge \neg R(x, F_2(x)))\}$$

Eliminate universal quantifiers

$$\{\forall x\, \neg P(x, F_1(x)) \vee (Q(x, F_2(x)) \wedge \neg R(x, F_2(x)))\}$$

$$\boxed{\forall v\, \alpha \quad \longrightarrow \quad \alpha}$$

$$\{\neg P(x, F_1(x)) \vee (Q(x, F_2(x)) \wedge \neg R(x, F_2(x)))\}$$

Distribute and, or

$$\{\neg P(x, F_1(x)) \vee (Q(x, F_2(x)) \wedge \neg R(x, F_2(x)))\}$$

$$\begin{array}{lcl} \alpha \vee (\beta \wedge \gamma) & \longrightarrow & (\alpha \vee \beta) \wedge (\alpha \vee \gamma) \\ (\beta \wedge \gamma) \vee \alpha & \longrightarrow & (\beta \vee \alpha) \wedge (\gamma \vee \alpha) \end{array}$$

$$\{(\neg P(x, F_1(x)) \vee Q(x, F_2(x))) \wedge (\neg P(x, F_1(x)) \vee \neg R(x, F_2(x)))\}$$

Flatten and, or

$$\{(\neg P(x, F_1(x)) \vee Q(x, F_2(x))) \wedge (\neg P(x, F_1(x)) \vee \neg R(x, F_2(x)))\}$$

$$\begin{array}{lcl} (\alpha \wedge (\beta \wedge \gamma)) & \longrightarrow & (\alpha \wedge \beta \wedge \gamma) \\ (\alpha \vee (\beta \vee \gamma)) & \longrightarrow & (\alpha \vee \beta \vee \gamma) \\ ((\alpha \wedge \beta) \wedge \gamma) & \longrightarrow & (\alpha \wedge \beta \wedge \gamma) \\ ((\alpha \vee \beta) \vee \gamma) & \longrightarrow & (\alpha \vee \beta \vee \gamma) \end{array}$$

$$\{(\neg P(x, F_1(x)) \vee Q(x, F_2(x))) \wedge (\neg P(x, F_1(x)) \vee \neg R(x, F_2(x)))\}$$

Eliminate and

$$\{(\neg P(x, F_1(x)) \vee Q(x, F_2(x))) \wedge (\neg P(x, F_1(x)) \vee \neg R(x, F_2(x)))\}$$

$$\boxed{\{\alpha \wedge \beta\} \quad \longrightarrow \quad \{\alpha,\ \beta\}}$$

$$\{\neg P(x, F_1(x)) \vee Q(x, F_2(x)),\ \neg P(x, F_1(x)) \vee \neg R(x, F_2(x))\}$$

Conjunctive Normal Form for FOL

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

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

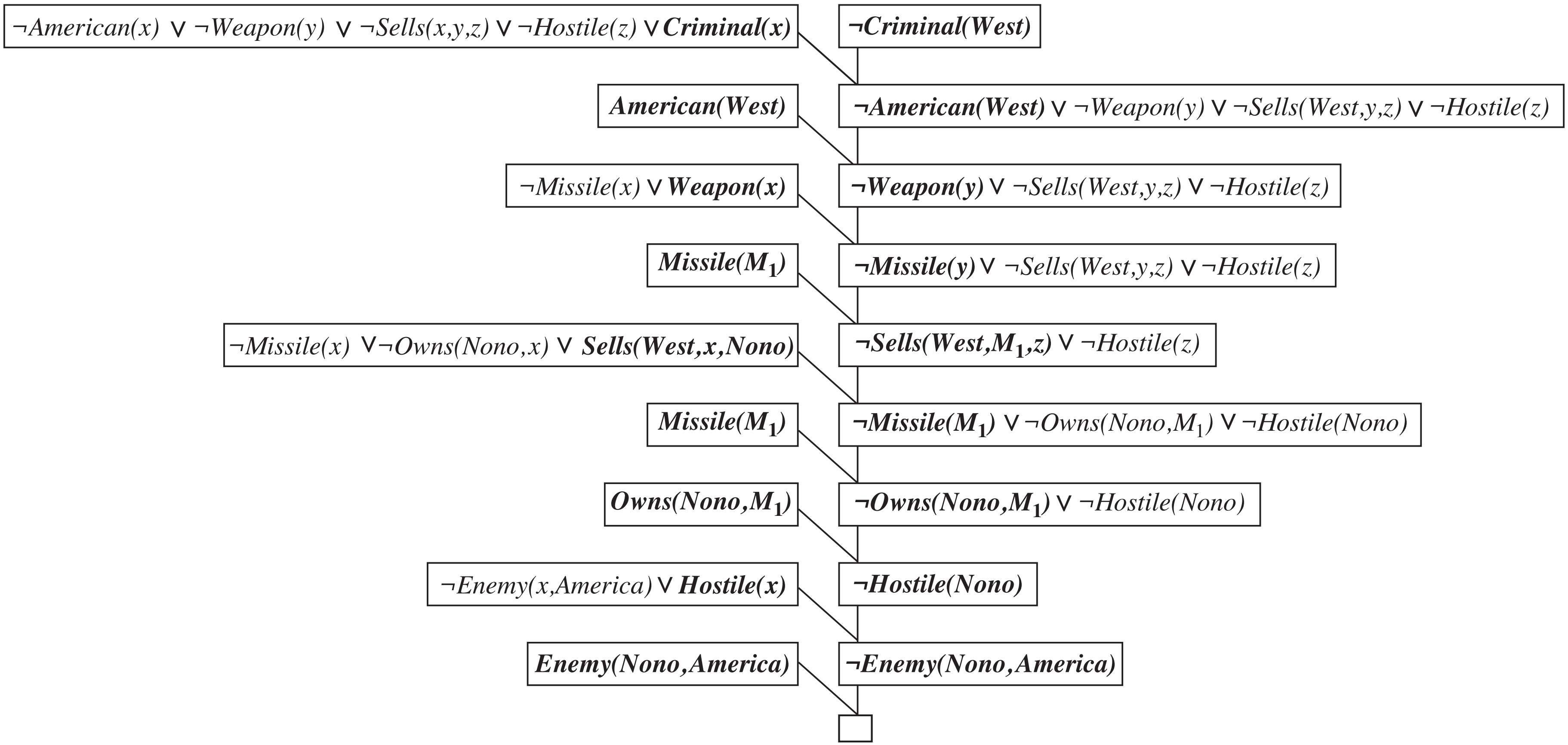
becomes, in CNF,

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

Example of proof by Resolution

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

Crime-Resolution



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

¬G. $\neg Kills(Curiosity, Tuna)$

Curiosity killed the cat? Original sentences: their conversion

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. $\neg Kills(Curiosity, Tuna)$

Explanations

- **Eliminate implications**:

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

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

$$\neg \forall x \;\; p \qquad \text{becomes} \qquad \exists x \;\; \neg p$$
$$\neg \exists x \;\; p \qquad \text{becomes} \qquad \forall x \;\; \neg p \,.$$

  Our sentence goes through the following transformations:

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

  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 $(\exists x \, P(x)) \lor (\exists x \, Q(x))$ 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

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

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

Transform the problem to set of clauses

$$Dog(D)$$

$$Owns(Jack, D)$$

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

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

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

$$Cat(Tuna)$$

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

¬Kills(Curiosity,Tuna)

Reading and Suggested Exercises

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