First-Order Logic Part3

Dr. Igor Ivkovic

iivkovic@uwaterloo.ca

[with material from “Mathematical Logic for Computer Science”, by Zhongwan, published by World Scientific]

Objectives

- Conversions to Conjunctive Normal Form, Prenex Normal Form, and Clausal Form
- Propositional Logic Formula Resolution
- First-Order Logic Formula Resolution

Normal Forms /1

- **Recall these definitions from earlier in the course…**
- **Definition 2.10 Literals and Clauses:** (Definition 2.7.1)
  - Atoms and their negations are called literals
  - Disjunctions (conjunctions) with literals as disjuncts (conjuncts) are called disjunctive (conjunctive) clauses
- **Definition 2.11 Normal Forms:** (Definition 2.7.2)
  - A disjunction with conjunctive clauses as its disjuncts is called a disjunctive normal form
  - A conjunction with disjunctive clauses as its conjuncts is called a conjunctive normal form
- **Theorem 2.4:** (Theorem 2.7.3)
  - Any A $\in$ Form($\mathcal{L}^p$) is logically equivalent to some disjunctive normal form
- **Theorem 2.5:** (Theorem 2.7.4)
  - Any A $\in$ Form($\mathcal{L}^p$) is logically equivalent to some conjunctive normal form

Normal Forms /2

- **Two literals are said to be complements/clashing if one is the negation of the other (e.g., p and $\neg$p)**
- Also recall these logical equivalences:
  - $A \rightarrow B \equiv \neg A \vee B$
  - $A \vee B \equiv \neg A \rightarrow B$
  - $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$
  - $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
  - $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
  - $\neg(A \vee B) \equiv \neg A \wedge \neg B$
  - $\neg(A \wedge B) \equiv \neg A \vee \neg B$

Normal Forms /3

- Let us convert the following formulas into CNF:
  - $(p \wedge q) \vee r$ in CNF is $(p \vee r) \wedge (q \vee r)$
  - $\neg(p \vee q)$ in CNF is $\neg p \wedge \neg q$
  - $p \rightarrow q$ in CNF is $\neg p \vee q$
- **Conversion to CNF:**
  - **Step 1. Eliminate all connectors but negation $\neg$, conjunction $\wedge$, and disjunction $\vee$**
  - **Step 2. Push negation inwards using De Morgan's Laws, $\neg(A \vee B) \equiv \neg A \wedge \neg B$ and $\neg(A \wedge B) \equiv \neg A \vee \neg B$**
  - **Step 3. Eliminate sequences of negations by deleting double negations with $\neg\neg A \equiv A$**
  - **Step 4. Distribute $\wedge$ over $\vee$ with distributive laws, such as $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$**

Normal Forms /4

- **Conversion to CNF Example:**
  - $(\neg p \rightarrow \neg q) \rightarrow (p \rightarrow q)$
  - Step 1. Eliminate all connectors but $\neg$, $\wedge$, and $\vee$

    $\equiv \neg(\neg\neg p \vee \neg q) \vee (\neg p \vee q)$
  - Step 2. Push negation inwards

    $\equiv (\neg\neg\neg p \wedge \neg\neg q) \vee (\neg p \vee q)$
  - Step 3. Eliminate sequences of negations

    $\equiv (\neg p \wedge q) \vee (\neg p \vee q)$
  - Step 4. Distribute $\wedge$ over $\vee$ with distributive laws

    $\equiv (\neg p \vee (\neg p \vee q)) \wedge (q \vee (\neg p \vee q))$

    $\equiv (\neg p \vee \neg p \vee q) \wedge (q \vee \neg p \vee q)$ *(by associative laws)*

Clausal Form /1

- **Recall:**
  - Atoms and their negations are called literals
  - Disjunctions with literals as disjuncts are called disjunctive clauses
  - A conjunction with disjunctive clauses as its conjuncts is called a conjunctive normal form
- Clausal form is based on the CNF:
  - A unit clause is a clause consisting of exactly one literal
  - Each clause in clausal form is implicitly a disjunction of its literals (i.e., each clause is a disjunctive clause)
  - The empty set of literals is the empty clause, denoted $\square$
  - **A formula in the clausal form is a set of disjunctive clauses; a formula is considered to be an implicit conjunction of its clauses**

Clausal Form /2

- Each $\varphi \in$ Form(L^{p}) can be transformed into a logically equivalent formula in clausal form
  - **For the formula $(\neg p \vee \neg p \vee q) \wedge (q \vee \neg p \vee q)$, its equivalent clausal form is $\{\{\neg p, \neg p, q\}, \{q, \neg p, q\}\}$**
  - Duplicate literals can be removed with the idempotent laws, $A \vee A \equiv A$ and $A \wedge A \equiv A$; Revised clausal form for above: $\{\{\neg p, q\}\}$
- **Trivial Clause:**
  - Contains a pair of clashing literals, such as p and $\neg p$
  - Removing a trivial clause from the clausal form of a formula does not change its truth value (trivial clauses are always true)
  - **If after removing all of the trivial clauses the clausal form is an empty set ∅ then the given formula is valid**
  - **Note that this is different from the empty clause □, which instead indicates that the clausal form is unsatisfiable**

Formula Resolution /1

- **Resolution Rule:**

$$\frac{\{p_1, \ldots\ p_i, \ldots\ p_n\}, \{q_1, \ldots\ q_j, \ldots\ q_m\}}{\{p_1, \ldots\ p_{i-1}, p_{i+1} \ldots\ p_n, q_1, \ldots\ q_{j-1}, q_{j+1} \ldots\ q_m\}}$$

  - $p_1, \ldots\ p_n, q_1, \ldots\ q_m$ are literals
  - **p_i and q_j are the clashing literals**
  - **The clause computed by the resolution rule is called the resolvent of the clashing (input) clauses**

- **Example:**
  - $$\frac{\{\neg p, q\}, \{p\}}{\{q\}}$$

Does this look familiar? What rule is it? ☺

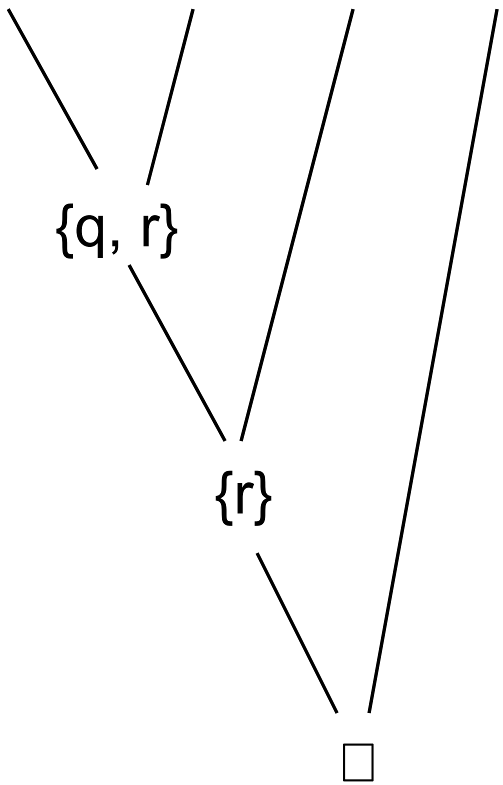
- **Resolvent Theorem:**
  - The resolvent is satisfiable iff its parent clauses are also satisfiable (for both propositional and first-order logic)

Formula Resolution /2

- **Formula resolution for a set of clauses S:**
  - **Goal: To determine if S is satisfiable or unsatisfiable**
    - **Also called resolution refutation of S**
  - Step 1. Select two clashing clauses C_1 and C_2 which are in S but that have not been selected before
  - Step 2. Compute the resolvent C for C_1 and C_2 according to the resolution rule defined previously
  - Step 3. If C is not a trivial clause, add C to S; otherwise, ignore C and continue
  - Step 4. Terminate if C is □ (S is unsatisfiable), or if all pairs of clashing clauses have been resolved but C is not an empty clause (S is satisfiable); otherwise, go to Step1
- **The formula resolution is both sound and complete**

Formula Resolution /3

- **Formula resolution example (shown as a graph):**
  - $\{\neg p, q\}, \{p, r\}, \{\neg q\}, \{\neg r\}$



  - Hence, the given set of clauses is unsatisfiable

Formula Resolution /4

- **Formula Resolution for First-Order Logic:**
  - Before we can apply the formula resolutions to first-order logic formulas, let us convert them into a special normal form called **Prenex Normal Form**
  - $\varphi \in$ WFF of first-order logic is in Prenex Normal Form if all its quantifiers are to the left of the expression
  - **That is, a formula φ is in Prenex Normal Form (PNF) if it is of the following form: $Q_1\, x_1 \,.\, \ldots\, Q_n\, x_n \,.\, M$, where Q_i's are the quantifiers, x_i's are the variables, and M is a WFF (called the matrix) that is free of quantifiers**
  - **$\varphi \in$ WFF is in Prenex Conjunctive Normal Form (PCNF) iff φ is in PNF and M is in CNF**

Formula Resolution /5

- **Conversion to PCNF:**
  - **Step 1. Rename bound variables so that no variable appears in two quantifiers**
  - **Step 2. Eliminate all connectors but $\neg$, $\wedge$, and $\vee$**
  - **Step 3. Push negation inwards**
  - **Step 4. Eliminate sequences of negations**
  - **Step 5. Extract quantifiers from the matrix using quantifier extraction rules**
  - **Step 6. Distribute $\wedge$ over $\vee$ with distributive laws**
  - **Step 7. Replace every existential quantifier $\exists x$ with a function $f(y_1, \ldots y_n)$ where y1, … yn are universally bound variables (e.g., $\forall y_1$) appearing before $\exists x$**
    - The functions f are called **Skolem functions**, and the process of replacing existential quantifiers with the Skolem functions is called **Skolemization**

Formula Resolution /6

- **Quantifier Extraction Rules:**
  - **Negation Rules:**

    $\neg\forall x\ A(x) \equiv \exists x\ \neg A(x)$

    $\neg\exists x\ A(x) \equiv \forall x\ \neg A(x)$
  - **Disjunction Rules:**
    (x does not appear in C)

    $C \vee \forall x\ A(x) \equiv \forall x(C \vee A(x))$

    $C \vee \exists x\ A(x) \equiv \exists x(C \vee A(x))$
  - **Conjunction Rules:**
    (x does not appear in C)

    $C \wedge \forall x\ A(x) \equiv \forall x(C \wedge A(x))$

    $C \wedge \exists x\ A(x) \equiv \exists x(C \wedge A(x))$
  - **Implication Rules:**
    (x does not appear in C)

    $C \rightarrow \forall x\ A(x) \equiv \forall x(C \rightarrow A(x))$

    $C \rightarrow \exists x\ A(x) \equiv \exists x(C \rightarrow A(x))$

    $\forall x\ A(x) \rightarrow C \equiv \exists x(A(x) \rightarrow C)$

    $\exists x\ A(x) \rightarrow C \equiv \forall x(A(x) \rightarrow C)$

Formula Resolution /7

- **Conversion to PCNF Example:**
  - $\exists x\ \forall y\ A(x, y) \rightarrow \forall y\ \exists x\ A(x, y)$
  - Step 1. Rename bound variables

    $\equiv \exists x\ \forall y\ A(x, y) \rightarrow \forall w\ \exists z\ A(z, w)$
  - Step 2. Eliminate all connectors but $\neg$, $\wedge$, and $\vee$

    $\equiv \neg\exists x\ \forall y\ A(x, y) \vee \forall w\ \exists z\ A(z, w)$
  - Step 3. Push negation inwards

    $\equiv \forall x\ \exists y\ \neg A(x, y) \vee \forall w\ \exists z\ A(z, w)$
    - Step 4 skipped since nothing to do
  - Step 5. Extract quantifiers from the matrix

    $\equiv \forall x\ \exists y\ \forall w\ \exists z\ (\neg A(x, y) \vee A(z, w))$
    - Step 6 skipped since nothing to do
  - Step 7. Apply Skolemization

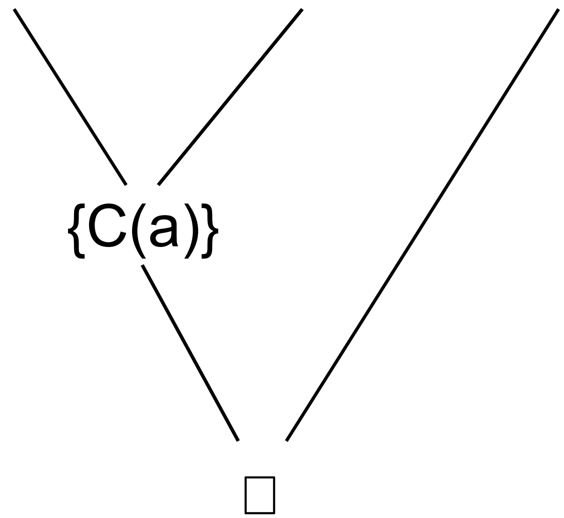
    $\equiv \forall x\ \forall w\ (\neg A(x, f(x)) \vee A(g(x, w), w))$

Formula Resolution /8

- **What is left to do before resolution?**
  - Drop the universal quantifiers and write the formula in the clausal form
  - $\exists x\ \forall y\ A(x, y) \rightarrow \forall y\ \exists x\ A(x, y)$ *(original formula)*
  - $\forall x\ \forall w\ (\neg A(x, f(x)) \vee A(g(x, w), w))$ *(PCNF formula)*
  - $\{\{\neg A(x, f(x)), A(g(x, w), w)\}\}$ *(clausal form)*
- **Ground Clause:**
  - A clause of a formula with no quantifiers and no variables
  - For instance, p(a, b) would be a ground clause for p(x, y) obtained by substituting a for x and b for y
- **Ground Formula Resolution:**
  - Converts all clauses into ground clauses and then applies the resolution rule

Formula Resolution /9

- **Ground resolution example (shown as a graph):**
  - {B(f(b)), C(a)}, {¬B(f(b)}, {¬C(a)}



  - Hence, the given set of clauses is unsatisfiable

Formula Resolution /10

- **What else can we prove with this method?**
  - We can prove that a formula is valid (or not valid)
  - We can prove that Σ satisfies φ (also that $\Sigma \vdash \varphi$ holds)
- **To prove that $\varphi \in$ WFF is valid:**
  - Step1. Convert $\neg\varphi$ into the clausal form
  - Step2. Apply ground resolution and check if the derived set of clauses is unsatisfiable
  - Step3. If $\neg\varphi$ is unsatisfiable then φ is valid
- **To prove that Σ satisfies φ:**
  - Step1. Convert Σ and $\neg\varphi$ into the clausal form
  - Step2. Apply ground resolution and check if the clauses of $\Sigma \cup \{\neg\varphi\}$ are unsatisfiable
  - Step3. If $\Sigma \cup \{\neg\varphi\}$ is unsatisfiable then Σ satisfies φ

Formula Resolution /11

- **Ground resolution example (shown as a graph):**
  - $\Sigma = \{\{A(x), B(x), C(x)\}, \{\neg A(a)\}, \{\neg B(x), \neg D(x)\}, \{D(a)\}\}$
  - $\varphi = C(a)$
  - $\{A(a), B(a), C(a)\}, \{\neg A(a)\}, \{\neg B(a), \neg D(a)\}, \{D(a)\}, \{\neg C(a)\}$

$\{B(a), C(a)\}$

$\{C(a), \neg D(a)\}$

$\{C(a)\}$

$\square$ (hence, $\Sigma \vdash \varphi$ holds)

Formula Resolution /12

- **Additional Notes:**
  - **Unification –** Apply the substitution to the input clauses to make the otherwise diverse clauses match
  - **Unifier** – A substitution θ is a unifier for two terms t_1 and t_2 that do not share any variables if $\theta(t_1) = \theta(t_2)$
  - For instance, A(x, f(y)) and A(f(x), f(z)) can be unified using θ(x/f(x), y/z) into A(f(x), f(z))

  - **Programs Defined as Formulas:**

    $\forall x$ PLUS(0, x, x)

    $\forall x \forall y \forall z$ (PLUS(x, y, z) $\rightarrow$PLUS(s(x), y, s(z)))

Food for Thought

- **Read:**
  - Chapter 3, Section 3.6 from Zhongwan
  - Chapter 6, Section 6.3 from Zhongwan
    - Read the material discussed in class in more detail
    - Cursory reading of the material not emphasized in class
- **Answer Assignment #5 questions**
  - Assignment #5 includes several practice exercises related to Formula Resolutions