Logical Deduction: III
Method of Resolution Refutation

Partha P Chakrabarti

Indian Institute of Technology
Kharagpur

August 28, 2019

Method of Resolution Refutation (Initially for Propositional Logic)

To prove validity of

$$F = ((F1 \land F2 \land \ldots \land Fn) \rightarrow G)$$

we shall attempt to prove that

$$\sim F = (F1 \land F2 \land \ldots \land Fn \land \sim G)$$

is unsatisfiable

Steps for Proof by Resolution Refutation:

1. Convert of Clausal Form / Conjunctive Normal Form (CNF, Product of Sums).
2. Generate new clauses using the resolution rule.
3. At the end, either False will be derived if the formula ~F is unsatisfiable implying F is valid.

Example 1

If Asha is elected VP then Rajat is chosen as G-Sec and Bharati is chosen as Treasurer. Rajat is not chosen as G-Sec. Therefore Asha is not elected VP.

F1: $(a \rightarrow (b \land c)) = (\sim a \lor b) \land (\sim a \lor c)$

F2: $\sim b$

G: $\sim a$

~G: a

Clauses of Clause Form: $\sim F = (C1 \land C2 \land C3 \land C4)$ where

- **C1: $(\sim a \lor b)$**
- **C2: $(\sim a \lor c)$**
- **C3: $\sim b$**
- **C4: a**

To prove that ~F is False

Resolution Rule for propositional Logic

Let C1 = a V b and C2 = ~a V c

then a new clause C3 = b V c can be derived.

(Proof by showing that ((C1 Λ C2) → C3) is a valid formula).

To prove unsatisfiability use the Resolution Rule repeatedly to reach a situation where we have two contradictory clauses of the form C1 = a and C2 = ~a from which False can be derived.

If the propositional formula is satisfiable then we will not reach a contradiction and eventually no new clauses will be derivable.

For propositional logic the procedure terminates.

Resolution Rule is Sound and Complete

Proving Example 1

If Asha is elected VP then Rajat is chosen as G-Sec and Bharati is chosen as Treasurer. Rajat is not chosen as G-Sec. Therefore Asha is not elected VP.

F1: $(a \rightarrow (b \wedge c)) = (\sim a \vee b) \wedge (\sim a \vee c)$, F2: $\sim b$

G: $\sim a$, $\sim$G: a

Clauses of Clause Form: $\sim F = (C1 \wedge C2 \wedge C3 \wedge C4)$ where

- **C1: $(\sim a \vee b)$, C2: $(\sim a \vee c)$, C3: $\sim b$, C4: a**

New Clauses Derived

C5: $\sim a$ (Using C1 and C3)

C6: False (using C4 and C5)

Classwork Examples 2 and 3

- If Asha is elected VP then <u>either</u> Rajat is chosen as G-Sec or Bharati is chosen as Treasurer. Rajat is not chosen as G-Sec. Therefore if Asha is elected as VP then Bharati is chosen as Treasurer
- If Asha is elected VP then Rajat is chosen as G-Sec and Bharati is selected as Treasurer. Rajat is chosen as G-Sec. Therefore if Asha is elected as VP then Bharati is not chosen as Treasurer

Classwork Examples 4 and 5

- Rajesh either took the bus or came by cycle to Nalanda. If he came by cycle or walked down to Nalanda he arrived Nalanda late and missed the first class. Rakesh did not arrive late. Therefore he took the bus to Nalanda.
- Either the tariff is lowered or imports continue to decrease and thereby our own industries prosper. If the tariff is lowered then our own industries prosper. Therefore our own industries prosper.

Resolution Method: Predicate Calculus

Given a formula F which we wish to check for validity, we first check if there are any free variables. We then quantify all free variables universally.

Create F' = ~F and check for unsatisfiability of F'

STEPS:

- **Conversion to Clausal (CNF) Form:**
  - **Handling of Variables and Quantifiers, Ground Instances**
- **Applying the Resolution Rule:**
  - **Concept of Unification**
  - **Principle of Most General Unifier (mgu)**
  - **Repeated application of Resolution Rule using mgu**

Conversion to Clause Form: Steps

1. Remove implications and other Boolean symbols converting to equivalent forms using ~, V, Λ
2. Move negates (~) inwards as close as possible
3. Standardize (Rename) variables to make them unambiguous
4. Remove Existential Quantifiers by an appropriate new function /constant symbol taking into account the variables dependent on the quantifier (Skolemization)
5. Drop Universal Quantifiers
6. Distribute V over Λ and convert to CNF

Work out the Example: All those who like every student are liked by someone.

$\forall x(\forall y(student(y) \rightarrow likes(x, y)) \rightarrow (\exists z(likes(z,x)))$

Substitution, Unification, Resolution

Consider clauses:

- C1: ~studies(x,y) V passes(x,y)
- C2: studies(Madan,z)
- C3: ~passes(Chetan, Physics)
- C4: ~passes(w, Mechanics)

What new clauses can we derive by the resolution principle?

Ground Clause and a more general clause

Concept of substitution / unification and the Most General Unifier (mgu)

Resolution Rule for Predicate Calculus: Repeated Application of Resolution using mgu

Classwork Examples

- **Wherever Mary goes, so does the lamb. Mary goes to school. So the lamb goes to school.**
- **No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.**
- **All dancers are graceful. Ayesha is a student. Ayesha is a dancer. Therefore some student is graceful.**
- **Every passenger is either in first class or second class. Each passenger is in second class if and only if he or she is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore some passengers are in second class.**

Alpine Club Problem

Tony, Mike, and John belong to the Alpine Club. Every member of the Alpine Club is either a skier or a mountain climber or both. No mountain climber likes rain, and all skiers like snow. Mike dislikes whatever Tony likes and likes whatever Tony dislikes. Tony dislikes rain and snow. Represent this information by predicate-calculus sentences in such a way that you can represent the question: Is there a member of the alpine club who is a skier but not a mountain climber? Prove the validity of the deduction problem using resolution refutation.

Algorithms for Resolution Refutation

- Soundness and Completeness of the Resolution Refutation Method.
- Efficient Algorithm for Resolution using the Most General Unifier
- Smart Search Schemes and their Completeness
  - Breadth-first Search
  - Depth-first Search
  - Unit Resolution, Unit Preference
  - Set of Support
  - Ancestry Filtered Form

Some Techniques for Deduction

- Rule Based Deduction
  - Forward Chaining
  - Backward Chaining
- Horn Clause form and the Programming Language Prolog

Thank you