Resolution Refutement

$F_1 \cap F_2 \cap F_3 \cap \ldots\ldots F_n \rightarrow G$

$\hookrightarrow$ CNF

$A \rightarrow B = (\neg A \cup B)$

$\downarrow$

$(C_1 \cap C_2 \cap C_3 \ldots\ldots \cap \neg G) \rightarrow \alpha$

CNF

$\alpha \in \{\ \}$

$\boxed{G \text{ is true}}$

$\neg G \quad \{F\}$

$\boxed{\text{FOL:}}$ $\{\underline{\forall, \exists}\}$

$\forall \rightarrow$ Universal Quantifiers

$\exists \rightarrow$ Existential Quantifier.

$\forall x \{p(x)\} = p(K)$

where $K \in C$.

All Man are mortal $= \forall x(m(x) \rightarrow mor(x))$

Socrates is a man $= m(\text{Socrates})$

Socrates is mortal.

$S_1 \quad \underline{\forall x} \; (m(x) \rightarrow mortal(x))$

$S_2 \quad m(\text{Socrates})$

$\}\ P$

G : mortal(socrates)

$\boxed{\alpha \to \beta \to \neg\alpha \lor \beta}$

$$\begin{cases} S_1 : \forall x \, (\neg(m(x)) \lor mortal(x)) \\ S_2 : m(socrates) \end{cases}$$

$(\neg(m(?x)) \lor mortal(?x)$

S_1' : $\cancel{\forall x}$ $(\neg\, m(socrates)) \lor mortal(socrates))$

S_2 : $m(socrates)$ Modus Pons

---

mortal (socrates) G

Universal Instantiation

$$\forall x \, (P(\ldots\ldots x \ldots\ldots))$$

$$\Downarrow$$

Skolemization : $P(K)$ where $K \in C$

$$\forall x \, \exists y \, (P(x, y)) \Rightarrow \forall x \, (P(x, f(x)))$$

$$\to (P(x, f(x)))$$

$$\boxed{y = f(x)}$$

$$\exists x \forall y \left(p(x, y) \right) \to \cancel{\forall y} \left(p(k, y) \right)$$

$$\boxed{p(k, y)}$$

Predicate Logic

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.

New Additions in Proposition (First Order Logic)

Variables, Constants, Predicate Symbols and

New Connectors: ∃ (there exists), ∀(for all)

Wherever Mary goes, so does the Lamb. Mary goes to School. So the Lamb goes to School.

Predicate: goes(x,y) to represent x goes to y

New Connectors: ∃ (there exists), ∀(for all)

F1: ∀x(goes(Mary, x) → goes(Lamb, x))

F2: goes(Mary, School)

G: goes(Lamb, School)

To prove: (F1 ∧ F2) → G) is always true

Resolution Refutation for Propositional Logic

To prove validity of

F = ((F1 Λ F2 Λ … Λ Fn) → G)

we shall attempt to prove that

~F = (F1 Λ F2 Λ … Λ Fn Λ ~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.**

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 → (b Λ c)) = (~a V b) Λ (~a V c)

F2: ~b, G: ~a, ~G: a

Clauses of Clause Form: ~F = (C1 Λ C2 Λ C3 Λ C4)

where: C1: (~a V b)

C2: (~a V c)

C3: ~b

C4: a

To prove that ~F is False

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

$F_1 : a \rightarrow (b \wedge c) = \underbrace{(\neg a \vee b)}_{C_1} \wedge \underbrace{(\sim a \vee c)}_{C_2}$

$F_2 : \underline{\neg b}\ C_3$

$G : \neg a$

$\neg G : \underline{a}\ C_4$

$\frac{a \wedge \neg a}{b \wedge \neg b}$

$C_1 \wedge C_2 \wedge C_3 \wedge C_4$

$\underset{C_1}{\boxed{(\neg a \vee b)}} \wedge (\neg a \vee c) \wedge \underset{C_3}{\boxed{(\neg b)}} \wedge a$

$\boxed{\neg a}$

$\boxed{F}$

$b,\ \neg b$

$$\frac{\begin{matrix}\neg a \vee b \\ \neg b\end{matrix}}{\neg a}$$

G is valid

$\neg G$ is true.
$(\sim G)$

Applying Resolution Refutation

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 ∧ **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

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 → (b ∧ c)) = (~a V b) **∧** (~a V c)

F2: ~b

G: ~a

~G: a

Clauses of Clause Form: ~F = (C1 ∧ C2 ∧ C3 ∧ C4)
where: C1: (~a V b)
C2: (~a V c)
C3: ~b
C4: a
To prove that ~F is False

New Clauses Derived
C5: ~a (Using C1 and C3)
C6: False (using C4 and C5)

SEMI-DECIDABLE

Example

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 ∧ 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

Rajesh either took the bus or came by cycle to class. If he came by cycle or walked to class he arrived late. Rajesh did not arrive late. Therefore he took the bus to class.

Symbols: Pl:- $bus(R)$; $cycle(R)$, $late(R)$; $walk(R)$

S_1:- $(bus(R) \wedge \neg cycle(R)) \vee (\neg bus(R) \wedge cycle(R))$

$bus(R) \vee cycle(R)$

C_1 C_2

S_1:- $\boxed{(bus(R) \vee cycle(R))} \wedge \boxed{(\neg cycle(R) \vee \neg bus(R))}$

S_2:- $(cycle(R) \vee walk(R) \rightarrow late(R))$.

S_2 :- $(\mathrm{Cycle}(R) \vee \mathrm{walk}(R) \rightarrow \mathrm{late}(R))$.

$\neg((\mathrm{Cycle}(R) \vee \mathrm{walk}(R))) \vee \mathrm{late}(R)$

$(\neg(\mathrm{Cycle}(R)) \wedge \neg(\mathrm{walk}(R))) \vee \mathrm{late}(R)$.

S_2 :- $\underbrace{\boxed{(\neg(\mathrm{Cycle}(R)) \vee \mathrm{late}(R))}}_{C_3} \wedge \underbrace{\boxed{(\neg(\mathrm{walk}(R)) \vee \mathrm{late}(R))}}_{C_4}$

S_3 :- $\boxed{\neg \mathrm{late}(R)}$ C_5

G :- $\mathrm{bus}(R)$

$\sim G$:- $\boxed{\neg \mathrm{bus}(R)}$ C_6.

$C_1 \wedge C_2 \wedge C_3 \wedge C_4 \; C_5 \wedge C_6$. $\rightarrow$ F.

G & C_6 :- $(\neg \cancel{\mathrm{bus}}(R))$; $\boxed{(\cancel{\mathrm{bus}}(R) \vee \mathrm{Cycle}(R))}$

$\rightarrow$ $\boxed{(\mathrm{Cycle}(R))}$ C_7

$\boxed{(\mathrm{late}(R))}$

$\boxed{\square}$

$\sim G$ is

$\rightarrow$ G is valid.

Resolution Refutation for Predicate Logic

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 CLAUSAL FORM IN PREDICATE LOGIC

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**

Resolution Refutation for Predicate Logic

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**

F1: ~~∀~~x(goes(Mary, x) → goes(Lamb, x))

F2: goes(Mary, School)

G: goes(Lamb, School)

To prove: (F1 ∧ F2) → G) is valid

CONVERSION TO CLAUSAL FORM IN PREDICATE LOGIC

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**

n~G

F1:- ¬CNF.

G is valid

School School

S1:- goes(Mary, A) → goes(Lamb, A

School School

S1:- (¬(goes(Mary, A)) → goes(Lamb, A

S2:- (goes Mary, School)

S3:- goes(Lamb, Sch

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

$\boxed{\Box}$ goes(lamb, school)

~G is false

G is valid

Examples

F1: ∀x(**contractor(x) → ~dependable(x))**

F2: ∃x(engineer(x) ∧ contractor(x))

G: ∃x(engineer(x) ∧ ~dependable(x))

F1: ∀x(dancer(x) → graceful (x))

F2: student(Ayesha), F3: dancer(Ayesha)

G: ∃x(student (x) ∧ graceful(x))

A

S_1 :- (¬(Contractor(A)) ∨ ¬ dependable(A)).

S_3 :- Engineer (A) ∧ Contractor (A)

C_2 C_3

~G :- ¬∃x(Engineer(x) ∧ ¬dependable(x))

G ∃x(Engineer(A) ∧ ¬ dependable(A))

~G :- (¬Engineer(A) ∨ dependable(A))

C_4

C_5 dependable (A)

¬Contractor (A)

x is a student

$\forall x$ (Exam(x, pass) $\longrightarrow$ Course(x, happy))

Cause

Effect

faculty (x, happy) ✓

SAC (x, Co