Resolution

The Basis of Resolution

- Simple Iterative process

In proposition logic

- At each step , two clauses , called parent clauses are compared (resolved) yielding new clause that has been inferred from them.
- The new clause represents ways that the parent clauses interact with each other
- Eg: There are two clauses in the system

Winter V summer

¬ Winter V cold } Summer V Cold

- Resolution operate by taking two clauses that each contain the same literal.
- Literal must occur in positive form in one clause and negative in other.
- Resolvent is obtained by combining all of the literals of the two parent clauses except the one that cancel.
- If the clause that is produced is empty clause, then a contradiction has been found.
- Eg two clauses

  Winter

¬ Winter

Will produce empty clause

- If contradiction exists then it will eventually be found and if no contradiction then chances are that procedure will never terminate.

- Resolution more complicated in predicate logic
- The theoretical basis of resolution principle in predicate logic is Herbrand's theorem
- → To show that a set of clauses S is unsatisfiable , it is necessary to consider only interpretations over a particular set , called Herbrand Universe of S.
- →A set of clauses S is unsatisfiable if and only if a finite subset of ground instances of S is unsatisfiable.

Resolution in Propositional Logic

Normal Forms

A sentence or well-formed formula (wff) is in **conjunctive normal form** if it is of the following form:

$$A_1 \wedge A_2 \wedge A_3 \wedge \ldots \wedge A_n$$

where each **clause**, A_i, is of the form

$$B_1 \vee B_2 \vee B_3 \vee \ldots \vee B_n$$

Each B_i is a **literal**, where a literal is a basic symbol of propositional logic.

Hence, in the following expression:

$A \wedge B \vee (\neg C \wedge D)$

A is an atom, as are *B*, *C*, and *D*. The literals are *A*, *B*, ¬*C*, and *D*.

in conjunctive normal form (often written CNF)

set of *or* phrases *and*ed together, such as:

$A \wedge (B \vee C) \wedge (\neg A \vee \neg B \vee \neg C \vee D)$

disjunctive normal form (DNF)

a set of *and* phrases *ored* together, as in

$A \vee (B \wedge C) \vee (\neg A \wedge \neg B \wedge \neg C \wedge D)$

Any wff can be converted to CNF by using the following equivalences

1. $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$
2. $A \rightarrow B \equiv \neg A \vee B$
3. $\neg(A \wedge B) \equiv \neg A \vee \neg B$
4. $\neg(A \vee B) \equiv \neg A \wedge \neg B$
5. $\neg\neg A \equiv A$
6. $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

$(A \rightarrow B) \rightarrow C$ to CNF

$(A \rightarrow B) \rightarrow C$

$\neg(A \rightarrow B) \vee C$ (2)

$\neg(\neg A \vee B) \vee C$ (3)

$(A \wedge \neg B) \vee C$ (4)

$(A \vee C) \wedge (\neg B \vee C)$ (6)

A further example

$A \leftrightarrow (B \wedge C)$

$(A \rightarrow (B \wedge C)) \wedge ((B \wedge C) \rightarrow A)$ (1)

$(\neg A \vee (B \wedge C)) \wedge (\neg(B \wedge C) \vee A)$ (2)

$(\neg A \vee (B \wedge C)) \wedge (\neg B \vee \neg C \vee A)$ (3)

$(\neg A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee \neg C \vee A)$ (6)

Having converted a wff into CNF, we can now express it as a set of clauses. So our expression above

$$(\neg A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee \neg C \vee A)$$

would be represented in clause form as

$$\{(\neg A, B), (\neg A, C), (\neg B, \neg C, A)\}$$

The Resolution Rule

$$\frac{A \vee B \qquad \neg B \vee C}{A \vee C}$$

can also be written as follows:

$$\frac{\neg A \rightarrow B \qquad B \rightarrow C}{\neg A \rightarrow C}$$

In this form, the rule can be seen to be saying that implication is transitive, or in other words, if *A* implies *B* and *B* implies *C*, then *A* implies *C*.

If a wff contains a clause that contains literal *L* and another clause that contains literal ¬*L*, then these two clauses can be combined together, and *L* and ¬*L* can be removed from those clauses. For example,

{(A, B), (¬B, C)}

can be resolved to give

{(A, C)}

Similarly,

{(A, B, C), D, (¬A, D, E), (¬D, F)}

can be resolved to give

{(B, C, D, E), D, (¬D, F)}

which can be further resolved to give either

{(B, C, D, E), F}

or

{(B, C, E, F), D}

Note that at the first step, we also had a choice and could have resolved to

{(A, B, C), D, (¬A, E, F)}

which can be further resolved to give

{(B, C, E, F), D}

Now, if wff P resolves to give wff Q, we write

$$P \models Q$$

For example, we can resolve $(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee C)$ as follows:

$$\{(A, B), (\neg A, C), (\neg B, C)\}$$

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

$$\{C\}$$

We can express this as

$$(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee C) \models C$$

If we resolve two clauses, we produce the **resolvent** of those clauses. The resolvent is a logical consequence of the two clauses.

Resolution Refutation

Now let us resolve the following clauses:

$$\{(\neg A, B), (\neg A, \neg B, C), A, \neg C\}$$

We begin by resolving the first clause with the second clause, thus eliminating *B* and ¬*B*:

$$\{(\neg A, C), A, \neg C\}$$

$$\{C, \neg C\}$$

$$\perp$$

The fact that this resolution has resulted in falsum means that the original clauses were inconsistent. We have **refuted** the original clauses, using **resolution refutation**. We can write

$$\{(\neg A, B), (\neg A, \neg B, C), A, \neg C\} \models \perp$$

Proof by Refutation (also known as **proof by contradiction**).

used in resolution refutation, is a powerful method for solving problems.

example, let us imagine that we want to determine whether the following logical argument is valid:

If it rains and I don't have an umbrella, then I will get wet.

It is raining, and I don't have an umbrella.

Therefore, I will get wet.

We can rewrite this in propositional calculus as follows:

$$(A \wedge \neg B) \rightarrow C$$

$$A \wedge \neg B$$

$$\therefore C$$

To prove this by refutation, we first negate the conclusion and convert the expressions into clause form.

The first expression is the only one that is not already in CNF, so first we convert this to CNF as follows:

$$(A \wedge \neg B) \rightarrow C$$
$$\equiv \neg(A \wedge \neg B) \vee C$$
$$\equiv \neg A \vee B \vee C$$

Now, to prove that our conclusion is valid, we need to show that

$$\{(\neg A, B, C), A, \neg B, \neg C\} \models \bot$$

We resolve these clauses as follows:

$$\{(B, C), \neg B, \neg C\}$$
$$\{C, \neg C\}$$
$$\bot$$

Hence, in showing that by negating our conclusion we lead to a contradiction, we have shown that our original conclusion must have been true.

If this process leads to a situation where some clauses are unresolved, and falsum cannot be reached, we have shown that the clauses with the negated conclusion are *not* contradictory and that therefore the original conclusion was not valid.

resolution proof in the form of a tree,

$A \to B$

$B \to C$

$C \to D$

$D \to E \vee F$

$\therefore A \to F$

First we negate the conclusion, to give: $\neg(A \to F)$.

Next we convert to clause form:

$$D \to E \vee F$$
$$\equiv \neg D \vee (E \vee F)$$

and

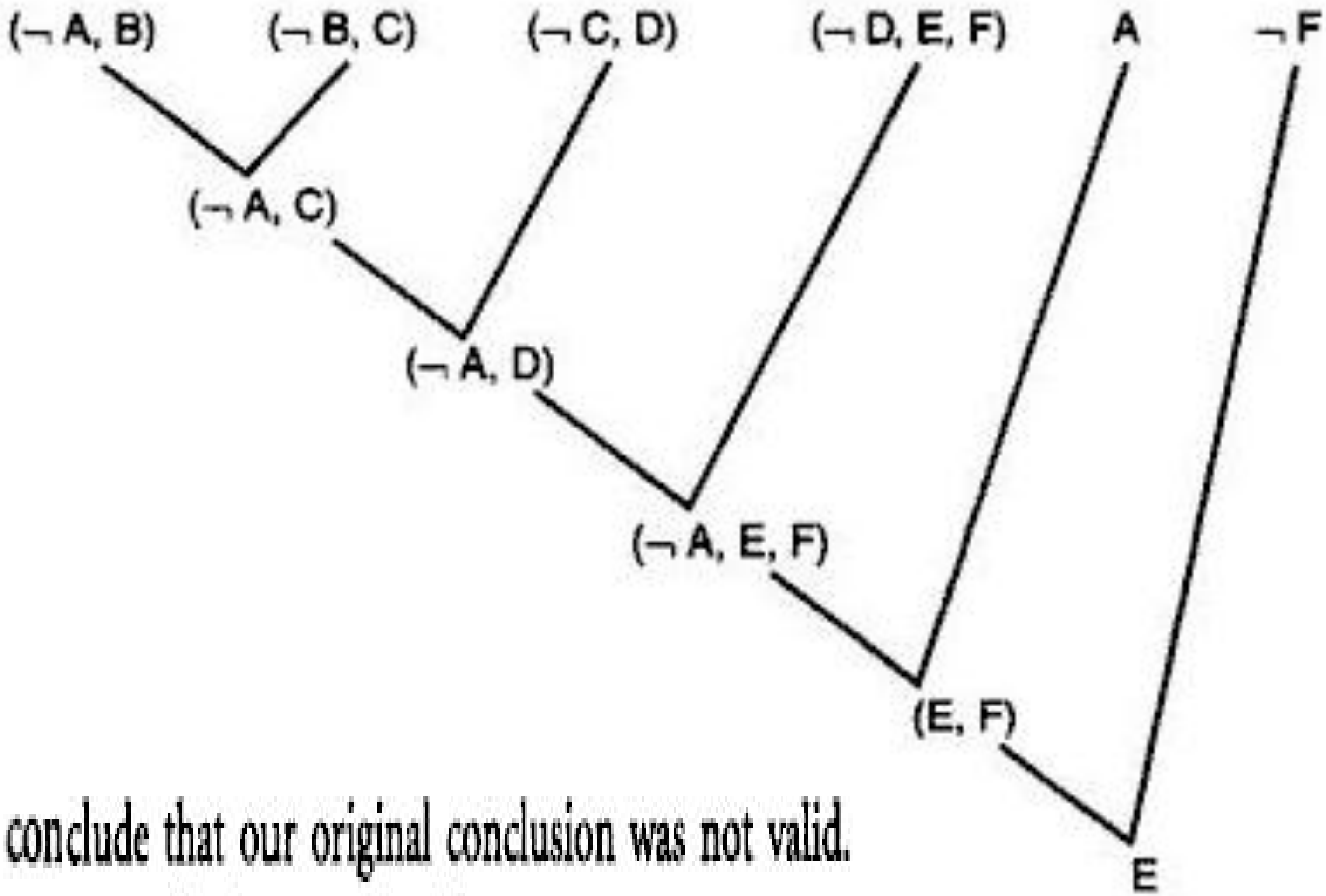
$$\neg(A \to F)$$
$$\equiv \neg(\neg A \vee F)$$
$$\equiv A \wedge \neg F$$

So, our clauses are

$$\{(\neg A, B), (\neg B, C), (\neg C, D), (\neg D, E, F), A, \neg F)\}$$

$\{(\neg A, B), (\neg B, C), (\neg C, D), (\neg D, E, F), A, \neg F\}$

Our proof in tree form is as follows:



conclude that our original conclusion was not valid.

Assignment: Applications of Resolution

Refer : **Artificial intelligence illuminated**

By Ben Coppin

Resolution in Predicate Logic

Normal Forms for Predicate Logic

To apply resolution to FOPL expressions, we first need to deal with the presence of the quantifiers ∀ and ∃. The method that is used is to move these quantifiers to the beginning of the expression, resulting in an expression that is in **prenex normal form.**

1. $A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$
2. $A \rightarrow B \equiv \neg A \vee B$
3. $\neg(A \wedge B) \equiv \neg A \vee \neg B$
4. $\neg(A \vee B) \equiv \neg A \wedge \neg B$
5. $\neg\neg A \equiv A$
6. $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
7. $\neg(\forall x)A(x) \equiv (\exists x)\neg A(x)$
8. $\neg(\exists x)A(x) \equiv (\forall x)\neg A(x)$
9. $(\forall x)A(x) \wedge B \equiv (\forall x)(A(x) \wedge B)$
10. $(\forall x)A(x) \vee B \equiv (\forall x)(A(x) \vee B)$
11. $(\exists x)A(x) \wedge B \equiv (\exists x)(A(x) \wedge B)$
12. $(\exists x)A(x) \vee B \equiv (\exists x)(A(x) \vee B)$
13. $(\forall x)A(x) \wedge (\forall y)B(y) \equiv (\forall x)(\forall y)(A(x) \wedge B(y))$
14. $(\forall x)A(x) \wedge (\exists y)B(y) \equiv (\forall x)(\exists y)(A(x) \wedge B(y))$
15. $(\exists x)A(x) \wedge (\forall y)B(y) \equiv (\exists x)(\forall y)(A(x) \wedge B(y))$
16. $(\exists x)A(x) \wedge (\exists y)B(y) \equiv (\exists x)(\exists y)(A(x) \wedge B(y))$

Skolemization

Before resolution can be carried out on a wff, we need to eliminate all the existential quantifiers, ∃, from the wff.

This is done by replacing a variable that is existentially quantified by a constant, as in the following case:

$$\exists(x)\ P(x)$$

would be converted to

$$P(c)$$

where *c* is a constant *that has not been used elsewhere in the wff.* .

This process is called **skolemization,** and the variable *c* is called a **skolem constant**

$\exists x(x \vee b)$

This would leave us with

$b \vee b$

which reduces to

b

This clearly is not the same expression, and, in fact, it should be skolemized using a different constant, such as:

$c \vee b$

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

In this case, rather than replacing y with a skolem constant, we must replace it with a **skolem function**, such as in the following:

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

Having removed the existential quantifiers in this way, the wff is said to be in **skolem normal form**, and has been **skolemized**.