Mathematical Logic

Lecture 6: Propositional Logic – Resolution - Part 2

Dr. CHOUDER R

Department of Computer Science

Level: First Year LMD — Licence Computer Science

Academic Year 2025-2026

Logical Deduction

To determine whether an argument is logically valid, we can use two complementary approaches:

To determine whether an argument is logically valid, we can use two complementary approaches:

- **Model Theory (Semantics):** is based on truth tables and interpretations.

To determine whether an argument is logically valid, we can use two complementary approaches:

- **Model Theory (Semantics):** is based on truth tables and interpretations.
- **Proof Theory (Syntax):** is based on formal systems and inference rules.

To determine whether an argument is logically valid, we can use two complementary approaches:

- **Model Theory (Semantics):** is based on truth tables and interpretations.
- **Proof Theory (Syntax):** is based on formal systems and inference rules.

Model Theory (Semantics)	**Proof Theory (Syntax)**
$Q \models P$	$Q \vdash P$
P is a consequence of Q	P is deducible from Q
Semantic consequence	Syntactic deduction
$\models P$: P is a tautology	$\vdash P$: P is a theorem

Is semantic consequence the same as syntactic deduction?

Is semantic consequence the same as syntactic deduction?

This leads to two fundamental properties

Soundness and Completeness

Proof theory is based on the definition of a formal system.

Proof theory is based on the definition of a formal system.
In this part, we use a formal system that allows us to simulate reasoning through formal derivations.

Proof theory is based on the definition of a formal system.
In this part, we use a formal system that allows us to simulate reasoning through formal derivations.
A formal system is defined by:

- A language (e.g., propositional logic, predicate logic)

Proof theory is based on the definition of a formal system.
In this part, we use a formal system that allows us to simulate reasoning through formal derivations.
A formal system is defined by:

- A language (e.g., propositional logic, predicate logic)
- Well-formed formulas (WFF)

Proof theory is based on the definition of a formal system.
In this part, we use a formal system that allows us to simulate reasoning through formal derivations.
A formal system is defined by:

- A language (e.g., propositional logic, predicate logic)
- Well-formed formulas (WFF)
- Axioms

Proof theory is based on the definition of a formal system.
In this part, we use a formal system that allows us to simulate reasoning through formal derivations.
A formal system is defined by:

- A language (e.g., propositional logic, predicate logic)
- Well-formed formulas (WFF)
- Axioms
- Inference rules

Definition

A formal system is **sound** if it does not allow us to prove formulas that are not logically valid.
A formal system is **sound** if:

$$Q \vdash P \;\Rightarrow\; Q \models P$$

Definition

A formal system is **sound** if it does not allow us to prove formulas that are not logically valid.
A formal system is **sound** if:

$$Q \vdash P \;\Rightarrow\; Q \models P$$

- Everything we can prove is logically valid.

Definition

A formal system is **sound** if it does not allow us to prove formulas that are not logically valid.
A formal system is **sound** if:

$$Q \vdash P \;\Rightarrow\; Q \models P$$

- Everything we can prove is logically valid.
- The formal system never proves false statements.

Definition

A formal system is **sound** if it does not allow us to prove formulas that are not logically valid.
A formal system is **sound** if:

$$Q \vdash P \;\Rightarrow\; Q \models P$$

- Everything we can prove is logically valid.
- The formal system never proves false statements.

No incorrect derivations.

Definition

A formal system is **complete** if it allows us to prove every logically valid formula.

A formal system is **complete** if:

$$Q \models P \;\Rightarrow\; Q \vdash P$$

Definition

A formal system is **complete** if it allows us to prove every logically valid formula.
A formal system is **complete** if:

$$Q \models P \;\Rightarrow\; Q \vdash P$$

- Every logically valid statement can be proved.

Definition

A formal system is **complete** if it allows us to prove every logically valid formula.
A formal system is **complete** if:

$$Q \models P \;\Rightarrow\; Q \vdash P$$

- Every logically valid statement can be proved.
- The formal system is powerful enough to capture all semantic truths.

Definition

A formal system is **complete** if it allows us to prove every logically valid formula.
A formal system is **complete** if:

$$Q \models P \;\Rightarrow\; Q \vdash P$$

- Every logically valid statement can be proved.
- The formal system is powerful enough to capture all semantic truths.

No missing valid statements.

If a formal system is sound and complete, then:

$$Q \vdash P \iff Q \models P$$

If a formal system is sound and complete, then:

$$Q \vdash P \iff Q \models P$$

$$\text{Syntax} = \text{Semantics}$$

If a formal system is sound and complete, then:

$$Q \vdash P \iff Q \models P$$

$$\text{Syntax} = \text{Semantics}$$

Fundamental Result

In propositional logic, most standard formal systems are both sound and complete.

From Theory to Automation

Soundness and completeness guarantee that logical reasoning can be automated.

From Theory to Automation

Soundness and completeness guarantee that logical reasoning can be automated.
This motivates the Resolution Method as a mechanical procedure.

Clausal Form

Clausal Form

Reminder

- **Literal:** an atomic formula or its negation (p or $\neg p$).

Clausal Form

Reminder

- **Literal:** an atomic formula or its negation (p or $\neg p$).
- **Clause:** a disjunction (logical OR) of literals $l_1 \vee \cdots \vee l_n$.

Clausal Form

Reminder

- **Literal:** an atomic formula or its negation (p or $\neg p$).
- **Clause:** a disjunction (logical OR) of literals $l_1 \vee \cdots \vee l_n$.
- In clausal form, only two logical connectors are used: negation ($\neg$) and disjunction ($\vee$).

Clausal Form

Reminder

- **Literal:** an atomic formula or its negation (p or $\neg p$).
- **Clause:** a disjunction (logical OR) of literals $l_1 \vee \cdots \vee l_n$.
- In clausal form, only two logical connectors are used: negation ($\neg$) and disjunction ($\vee$).
- A formula is in Conjunctive Normal Form (CNF) if it is a conjunction of clauses, where each clause is a disjunction of literals:

$$F = C_1 \wedge C_2 \wedge \cdots \wedge C_n$$

where each C_i is a clause.

Clausal Form

Reminder

- **Literal:** an atomic formula or its negation (p or $\neg p$).
- **Clause:** a disjunction (logical OR) of literals $l_1 \vee \cdots \vee l_n$.
- In clausal form, only two logical connectors are used: negation ($\neg$) and disjunction ($\vee$).
- A formula is in Conjunctive Normal Form (CNF) if it is a conjunction of clauses, where each clause is a disjunction of literals:

$$F = C_1 \wedge C_2 \wedge \cdots \wedge C_n$$

  where each C_i is a clause.
- A clause with no literals is the empty clause ($\square$), representing contradiction.

To obtain the set of clauses from a formula:

To obtain the set of clauses from a formula:

1. Put the formula in CNF form,

To obtain the set of clauses from a formula:

1. Put the formula in CNF form,
2. We eliminate conjunction connectors,

To obtain the set of clauses from a formula:

1. Put the formula in CNF form,
2. We eliminate conjunction connectors,
3. Then, we obtain the set of clauses of the initial formula.

Example

Let the formula:

$$F : (p \vee q) \Rightarrow (p \wedge q)$$

find the set of equivalent clauses.

Example

Let the formula:

$$F : (p \vee q) \Rightarrow (p \wedge q)$$

find the set of equivalent clauses.

1. We write the formula F in CNF form:

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

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

Example

Let the formula:

$$F : (p \vee q) \Rightarrow (p \wedge q)$$

find the set of equivalent clauses.

1. We write the formula F in CNF form:

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

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

2. We eliminate conjunction connectors, the set of clauses of formula F is:

Example

Let the formula:

$$F : (p \vee q) \Rightarrow (p \wedge q)$$

find the set of equivalent clauses.

1. We write the formula F in CNF form:

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

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

2. We eliminate conjunction connectors, the set of clauses of formula F is:

$$\{\neg p \vee q,\ \neg q \vee p\}$$

Resolution Principle

- The resolution rule (introduced by **J. Alan Robinson - resolution principle inventor**) is a rule of logical inference.

- The resolution rule (introduced by **J. Alan Robinson - resolution principle inventor**) is a rule of logical inference.
- A rule of logical inference derives new formulas from existing ones.

- The resolution rule (introduced by **J. Alan Robinson - resolution principle inventor**) is a rule of logical inference.
- A rule of logical inference derives new formulas from existing ones.
- If a literal appears in one clause and its negation appears in another, we remove both and combine the remaining literals.

- The resolution rule (introduced by **J. Alan Robinson - resolution principle inventor**) is a rule of logical inference.
- A rule of logical inference derives new formulas from existing ones.
- If a literal appears in one clause and its negation appears in another, we remove both and combine the remaining literals.
- The resolution rule is defined as:

$$\frac{p \qquad (\neg p \vee q)}{q} \quad \text{or} \quad \{p,\ \neg p \vee q\} \vdash q$$

Reads: from p and $(\neg p \vee q)$, we deduce q.

- The resolution rule is generalized as:

$$\frac{p \vee L \qquad \neg p \vee L'}{L \vee L'}$$

where L and L' are disjunctions of literals:

$$L = l_1 \vee l_2 \vee \cdots \vee l_n$$

$$L' = l'_1 \vee l'_2 \vee \cdots \vee l'_m$$

- The resolution rule is generalized as:

$$\frac{p \vee L \qquad \neg p \vee L'}{L \vee L'}$$

where L and L' are disjunctions of literals:

$$L = l_1 \vee l_2 \vee \cdots \vee l_n$$

$$L' = l'_1 \vee l'_2 \vee \cdots \vee l'_m$$

- From the two clauses C_1 and C_2, we derive a new clause:

$$C_1 = p \vee L, \quad C_2 = \neg p \vee L', \quad P = L \vee L'$$

- The resolution rule is generalized as:

$$\frac{p \vee L \qquad \neg p \vee L'}{L \vee L'}$$

where L and L' are disjunctions of literals:

$$L = l_1 \vee l_2 \vee \cdots \vee l_n$$
$$L' = l'_1 \vee l'_2 \vee \cdots \vee l'_m$$

- From the two clauses C_1 and C_2, we derive a new clause:

$$C_1 = p \vee L, \quad C_2 = \neg p \vee L', \quad P = L \vee L'$$

- By resolution, P is deduced from C_1 and C_2.

- The resolution rule is generalized as:

$$\frac{p \vee L \qquad \neg p \vee L'}{L \vee L'}$$

where L and L' are disjunctions of literals:

$$L = l_1 \vee l_2 \vee \cdots \vee l_n$$

$$L' = l'_1 \vee l'_2 \vee \cdots \vee l'_m$$

- From the two clauses C_1 and C_2, we derive a new clause:

$$C_1 = p \vee L, \quad C_2 = \neg p \vee L', \quad P = L \vee L'$$

- By resolution, P is deduced from C_1 and C_2.
- If L and L' become empty, the result is the empty clause $\square$ (contradiction).

Example

Consider the clauses: $C_1 = p \vee q, \qquad C_2 = \neg p \vee r$.
The literal p in C_1 and its negation $\neg p$ in C_2 are complementary.
By general resolution, we eliminate them and obtain:

$$Res(C_1, C_2) = q \vee r$$

Example

Consider the clauses: $C_1 = p \vee q, \qquad C_2 = \neg p \vee r$.
The literal p in C_1 and its negation $\neg p$ in C_2 are complementary.
By general resolution, we eliminate them and obtain:

$$Res(C_1, C_2) = q \vee r$$

Explanation:

- From C_1: either p or q is true.
- From C_2: either $\neg p$ or r is true.
- If p is true, C_2 forces r.
- If $\neg p$ is true, C_1 forces q.
- Therefore $q \vee r$ must hold.

Resolution Principle

Empty Clause

- Resolving two singleton clauses produces the empty clause, which contains no literals.

- Resolving two singleton clauses produces the empty clause, which contains no literals.
- The empty clause represents unsatisfiability; its derivation indicates that the database contains a contradiction.

Resolution Principle

Empty Clause

- Resolving two singleton clauses produces the empty clause, which contains no literals.
- The empty clause represents unsatisfiability; its derivation indicates that the database contains a contradiction.

Example

Consider the singleton clauses: $C_1 = p, \quad C_2 = \neg p$

These clauses contain complementary literals. By resolution, we eliminate them:

$$Res(C_1, C_2) = \square \quad \text{(empty clause)}$$

Resolution Principle

Empty Clause

- Resolving two singleton clauses produces the empty clause, which contains no literals.
- The empty clause represents unsatisfiability; its derivation indicates that the database contains a contradiction.

Example

Consider the singleton clauses: $C_1 = p, \quad C_2 = \neg p$

These clauses contain complementary literals. By resolution, we eliminate them:

$$Res(C_1, C_2) = \Box \quad \text{(empty clause)}$$

Explanation:

- C_1 states that p is true.
- C_2 states that p is false.
- Both cannot be true simultaneously.
- Therefore a contradiction arises.

- The clause P derived from two clauses C_1 and C_2 (i.e., $\{C_1, C_2\} \vdash P$) is a logical consequence:

$$\{C_1, C_2\} \models P$$

- The clause P derived from two clauses C_1 and C_2 (i.e., $\{C_1, C_2\} \vdash P$) is a logical consequence:

$$\{C_1, C_2\} \models P$$

- The resolution method reveals semantic consequence through syntactic derivation.

Example

Consider the clauses: $C_1 = p, \qquad C_2 = \neg p \lor q$
By resolution, we eliminate the complementary literals p and $\neg p$:

$$Res(C_1, C_2) = q$$

Example

Consider the clauses: $C_1 = p, \qquad C_2 = \neg p \vee q$
By resolution, we eliminate the complementary literals p and $\neg p$:

$$Res(C_1, C_2) = q$$

Explanation:

- C_1 states that p is true.
- C_2 states that either $\neg p$ or q is true.
- Since p is true, $\neg p$ cannot hold.
- Therefore q must be true.

Resolution Principle

Note

Example

Consider the clauses: $C_1 = p, \qquad C_2 = \neg p \vee q$

By resolution, we eliminate the complementary literals p and $\neg p$:

$$Res(C_1, C_2) = q$$

Explanation:

- C_1 states that p is true.
- C_2 states that either $\neg p$ or q is true.
- Since p is true, $\neg p$ cannot hold.
- Therefore q must be true.

Thus:

$$\{C_1, C_2\} \vdash q \quad \Rightarrow \quad \{C_1, C_2\} \models q$$

The resolution derivation highlights the semantic consequence.

Refutation

Proof by Refutation

The idea of proof by refutation is to demonstrate that the set $\Sigma \cup \{\neg P\}$ is unsatisfiable.

Theorem

Let Σ be a set of clauses.
$\Sigma \vdash P$ is equivalent to $\Sigma \cup \{\neg P\} \vdash \Box$.

Proof by Refutation

Example

Consider the set of clauses:

$$\Sigma = \{p,\ \neg p \vee q\}.$$

Prove that $\Sigma \vdash q$.

Proof by Refutation

Example

Consider the set of clauses:

$$\Sigma = \{p,\ \neg p \vee q\}.$$

Prove that $\Sigma \vdash q$.
To prove q by refutation, we show that $\Sigma \cup \{\neg q\}$ is unsatisfiable.

Example

Consider the set of clauses:

$$\Sigma = \{p,\ \neg p \vee q\}.$$

Prove that $\Sigma \vdash q$.
To prove q by refutation, we show that $\Sigma \cup \{\neg q\}$ is unsatisfiable.
Resolution steps:

- Resolve p with $(\neg p \vee q)$:

$$\{p,\ \neg p \vee q\} \vdash q.$$

Proof by Refutation

Example

Consider the set of clauses:

$$\Sigma = \{p,\ \neg p \vee q\}.$$

Prove that $\Sigma \vdash q$.
To prove q by refutation, we show that $\Sigma \cup \{\neg q\}$ is unsatisfiable.
Resolution steps:

- Resolve p with $(\neg p \vee q)$:

$$\{p,\ \neg p \vee q\} \vdash q.$$

- Now we have both q and $\neg q$:

$$\{q,\ \neg q\} \vdash \square.$$

Proof by Refutation

Example

Consider the set of clauses:

$$\Sigma = \{p,\ \neg p \vee q\}.$$

Prove that $\Sigma \vdash q$.
To prove q by refutation, we show that $\Sigma \cup \{\neg q\}$ is unsatisfiable.
Resolution steps:

- Resolve p with $(\neg p \vee q)$:

$$\{p,\ \neg p \vee q\} \vdash q.$$

- Now we have both q and $\neg q$:

$$\{q,\ \neg q\} \vdash \square.$$

The empty clause $\square$ indicates a contradiction, so $\Sigma \vdash q$.

Resolution Methods

- To demonstrate that P is deducible from a set of clauses C we use resolution proof.

- To demonstrate that P is deducible from a set of clauses C we use resolution proof.
- From the initial clauses C we apply the resolution rule to calculate the resolvent of two clauses.

- To demonstrate that P is deducible from a set of clauses C we use resolution proof.
- From the initial clauses C we apply the resolution rule to calculate the resolvent of two clauses.
- We apply the same principle for the resolving clauses until we obtain P.

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Solution:

1. Let the set of clauses be:

$$C = \{C_1, C_2, C_3\}, \quad C_1 = q \vee \neg r,\ C_2 = \neg q \vee t,\ C_3 = \neg t$$

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Solution:

1. Let the set of clauses be:

$$C = \{C_1, C_2, C_3\}, \quad C_1 = q \vee \neg r,\ C_2 = \neg q \vee t,\ C_3 = \neg t$$

2. Apply the resolution rule to C_2 and C_3: $Res(C_2, C_3) = \neg q$

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Solution:

1. Let the set of clauses be:

$$C = \{C_1, C_2, C_3\}, \quad C_1 = q \vee \neg r,\ C_2 = \neg q \vee t,\ C_3 = \neg t$$

2. Apply the resolution rule to C_2 and C_3: $Res(C_2, C_3) = \neg q$
3. Update the set of clauses: $C' = \{C_1 = q \vee \neg r,\ \neg q\}$

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Solution:

1. Let the set of clauses be:

$$C = \{C_1, C_2, C_3\}, \quad C_1 = q \vee \neg r,\ C_2 = \neg q \vee t,\ C_3 = \neg t$$

2. Apply the resolution rule to C_2 and C_3: $Res(C_2, C_3) = \neg q$
3. Update the set of clauses: $C' = \{C_1 = q \vee \neg r,\ \neg q\}$
4. Apply the resolution rule to C_1 and $\neg q$: $Res(C_1, \neg q) = \neg r$

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Solution:

1. Let the set of clauses be:

$$C = \{C_1, C_2, C_3\}, \quad C_1 = q \vee \neg r,\ C_2 = \neg q \vee t,\ C_3 = \neg t$$

2. Apply the resolution rule to C_2 and C_3: $Res(C_2, C_3) = \neg q$
3. Update the set of clauses: $C' = \{C_1 = q \vee \neg r,\ \neg q\}$
4. Apply the resolution rule to C_1 and $\neg q$: $Res(C_1, \neg q) = \neg r$
5. Therefore, we have deduced:

$$\{q \vee \neg r,\ \neg q \vee t,\ \neg t\} \vdash \neg r$$

- From $C \cup \{\neg P\}$, we apply a sequence of resolution rules to compute resolving clauses in order to obtain an empty clause $\square$, if possible.

- From $C \cup \{\neg P\}$, we apply a sequence of resolution rules to compute resolving clauses in order to obtain an empty clause $\square$, if possible.
- A set of clauses admitting refutation $C \cup \{\neg P\}$ is unsatisfiable.

- From $C \cup \{\neg P\}$, we apply a sequence of resolution rules to compute resolving clauses in order to obtain an empty clause $\square$, if possible.
- A set of clauses admitting refutation $C \cup \{\neg P\}$ is unsatisfiable.

Algorithm

- Let $C' = C \cup \{\neg P\}$.

- From $C \cup \{\neg P\}$, we apply a sequence of resolution rules to compute resolving clauses in order to obtain an empty clause $\square$, if possible.
- A set of clauses admitting refutation $C \cup \{\neg P\}$ is unsatisfiable.

Algorithm

- Let $C' = C \cup \{\neg P\}$.
- While $\square \notin C'$ and (there exist new possible choices)

- From $C \cup \{\neg P\}$, we apply a sequence of resolution rules to compute resolving clauses in order to obtain an empty clause $\square$, if possible.
- A set of clauses admitting refutation $C \cup \{\neg P\}$ is unsatisfiable.

Algorithm

- Let $C' = C \cup \{\neg P\}$.
- While $\square \notin C'$ and (there exist new possible choices)
  - Find two clauses to apply the resolution rule.

- From $C \cup \{\neg P\}$, we apply a sequence of resolution rules to compute resolving clauses in order to obtain an empty clause $\square$, if possible.
- A set of clauses admitting refutation $C \cup \{\neg P\}$ is unsatisfiable.

Algorithm

- Let $C' = C \cup \{\neg P\}$.
- While $\square \notin C'$ and (there exist new possible choices)
  - Find two clauses to apply the resolution rule.
  - Add the resolving clause to the clause list.

- From $C \cup \{\neg P\}$, we apply a sequence of resolution rules to compute resolving clauses in order to obtain an empty clause $\square$, if possible.
- A set of clauses admitting refutation $C \cup \{\neg P\}$ is unsatisfiable.

Algorithm

- Let $C' = C \cup \{\neg P\}$.
- While $\square \notin C'$ and (there exist new possible choices)
  - Find two clauses to apply the resolution rule.
  - Add the resolving clause to the clause list.
- If $\square \in C'$ then C' is unsatisfiable.
  Otherwise C' is satisfiable.

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Solution:

1. Form the set for refutation: $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r\}$

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Solution:

1. Form the set for refutation: $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r\}$
2. Apply resolution steps:

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Solution:

1. Form the set for refutation: $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r\}$
2. Apply resolution steps:
   - Resolve $q \vee \neg r$ and $\neg q \vee t$ on q: $Res(q \vee \neg r, \neg q \vee t) = \neg r \vee t$

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Solution:

1. Form the set for refutation: $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r\}$
2. Apply resolution steps:
   - Resolve $q \vee \neg r$ and $\neg q \vee t$ on q: $Res(q \vee \neg r, \neg q \vee t) = \neg r \vee t$
   - Add the new clause to C': $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r,\ \neg r \vee t\}$

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Solution:

1. Form the set for refutation: $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r\}$
2. Apply resolution steps:
   - Resolve $q \vee \neg r$ and $\neg q \vee t$ on q: $Res(q \vee \neg r, \neg q \vee t) = \neg r \vee t$
   - Add the new clause to C': $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r,\ \neg r \vee t\}$
   - Resolve $\neg r \vee t$ and $\neg t$ on t: $Res(\neg r \vee t, \neg t) = \neg r$

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Solution:

1. Form the set for refutation: $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r\}$
2. Apply resolution steps:
   - Resolve $q \vee \neg r$ and $\neg q \vee t$ on q: $Res(q \vee \neg r, \neg q \vee t) = \neg r \vee t$
   - Add the new clause to C': $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r,\ \neg r \vee t\}$
   - Resolve $\neg r \vee t$ and $\neg t$ on t: $Res(\neg r \vee t, \neg t) = \neg r$
   - Add the new clause to C': $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r,\ \neg r \vee t,\ \neg r\}$

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Solution:

1. Form the set for refutation: $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r\}$
2. Apply resolution steps:
   - Resolve $q \vee \neg r$ and $\neg q \vee t$ on q: $Res(q \vee \neg r, \neg q \vee t) = \neg r \vee t$
   - Add the new clause to C': $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r,\ \neg r \vee t\}$
   - Resolve $\neg r \vee t$ and $\neg t$ on t: $Res(\neg r \vee t, \neg t) = \neg r$
   - Add the new clause to C': $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r,\ \neg r \vee t,\ \neg r\}$
   - Resolve r and $\neg r$: $Res(r, \neg r) = \square$

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$.

Solution:

1. Form the set for refutation: $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r\}$
2. Apply resolution steps:
   - Resolve $q \vee \neg r$ and $\neg q \vee t$ on q: $Res(q \vee \neg r, \neg q \vee t) = \neg r \vee t$
   - Add the new clause to C': $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r,\ \neg r \vee t\}$
   - Resolve $\neg r \vee t$ and $\neg t$ on t: $Res(\neg r \vee t, \neg t) = \neg r$
   - Add the new clause to C': $C' = \{q \vee \neg r,\ \neg q \vee t,\ \neg t,\ r,\ \neg r \vee t,\ \neg r\}$
   - Resolve r and $\neg r$: $Res(r, \neg r) = \square$
3. The empty clause $\square$ indicates a contradiction.

Conclusion:

$$\{q \vee \neg r,\ \neg q \vee t,\ \neg t\} \vdash \neg r$$

This completes the refutation resolution proof.

- We can use a clause multiple times.

- We can use a clause multiple times.
- We don't require using all clauses to deduce the empty clause.

Selection strategies

- To apply the resolution rule, one of the chosen clauses must be a positive clause (with no negative laterals).

- To apply the resolution rule, one of the chosen clauses must be a positive clause (with no negative laterals).
- This strategy is complete for refutation.

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$ using the positive resolution strategy.

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$ using the positive resolution strategy.

1. **Step 1: Select a positive clause.** The positive literal clause is $q \vee \neg r$. We choose this as our starting point.

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$ using the positive resolution strategy.

1. **Step 1: Select a positive clause.** The positive literal clause is $q \vee \neg r$. We choose this as our starting point.
2. **Step 2: Find a clause containing the negation of a literal in the positive clause.** The literal q in $q \vee \neg r$ has a negation $\neg q$ in the clause $\neg q \vee t$. Apply resolution: $Res(q \vee \neg r, \neg q \vee t) = \neg r \vee t$

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$ using the positive resolution strategy.

1. **Step 1: Select a positive clause.** The positive literal clause is $q \vee \neg r$. We choose this as our starting point.
2. **Step 2: Find a clause containing the negation of a literal in the positive clause.** The literal q in $q \vee \neg r$ has a negation $\neg q$ in the clause $\neg q \vee t$. Apply resolution: $Res(q \vee \neg r, \neg q \vee t) = \neg r \vee t$
3. **Step 3: Repeat the process.** The new clause is $\neg r \vee t$. Choose t (positive literal) and find a clause with $\neg t$, which is $\neg t$. Apply resolution: $Res(\neg r \vee t, \neg t) = \neg r$

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$ using the positive resolution strategy.

1. **Step 1: Select a positive clause.** The positive literal clause is $q \vee \neg r$. We choose this as our starting point.
2. **Step 2: Find a clause containing the negation of a literal in the positive clause.** The literal q in $q \vee \neg r$ has a negation $\neg q$ in the clause $\neg q \vee t$. Apply resolution: $Res(q \vee \neg r, \neg q \vee t) = \neg r \vee t$
3. **Step 3: Repeat the process.** The new clause is $\neg r \vee t$. Choose t (positive literal) and find a clause with $\neg t$, which is $\neg t$. Apply resolution: $Res(\neg r \vee t, \neg t) = \neg r$
4. **Conclusion:** We have derived $\neg r$. Therefore: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$

- To apply the resolution rule, one of the chosen clauses must be a negative clause, (with no positive laterals).

Selection strategies

Negative resolution

- To apply the resolution rule, one of the chosen clauses must be a negative clause, (with no positive laterals).
- This strategy is complete for refutation.

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$ using the negative resolution strategy.

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$ using the negative resolution strategy.

1. **Step 1: Identify negative clauses** $\neg t$ is a negative clause (contains only negated literals)

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$ using the negative resolution strategy.

1. **Step 1: Identify negative clauses** $\neg t$ is a negative clause (contains only negated literals)
2. **Step 2: Apply resolution with negative clause** $Res(\neg t, \neg q \vee t) = \neg q$

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$ using the negative resolution strategy.

1. **Step 1: Identify negative clauses** $\neg t$ is a negative clause (contains only negated literals)
2. **Step 2: Apply resolution with negative clause** $Res(\neg t, \neg q \vee t) = \neg q$
3. **Step 3: Continue resolution** $Res(\neg q, q \vee \neg r) = \neg r$

Example

Demonstrate that: $\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$ using the negative resolution strategy.

1. **Step 1: Identify negative clauses** $\neg t$ is a negative clause (contains only negated literals)
2. **Step 2: Apply resolution with negative clause** $Res(\neg t, \neg q \vee t) = \neg q$
3. **Step 3: Continue resolution** $Res(\neg q, q \vee \neg r) = \neg r$
4. **Conclusion:**

$$\{q \vee \neg r; \neg q \vee t; \neg t\} \vdash \neg r$$

- The principle of linear resolution: reduce the clauses at each step by eliminating one of the two clauses which is used to calculate the resolvent.

- The principle of linear resolution: reduce the clauses at each step by eliminating one of the two clauses which is used to calculate the resolvent.
- Horn Clause:

- The principle of linear resolution: reduce the clauses at each step by eliminating one of the two clauses which is used to calculate the resolvent.
- Horn Clause:
  - Any clause that has at most one positive literal.
  - Examples: $p \vee \neg q$, $p \vee \neg q \vee \neg r$, $\neg p$.

- The principle of linear resolution: reduce the clauses at each step by eliminating one of the two clauses which is used to calculate the resolvent.
- Horn Clause:
  - Any clause that has at most one positive literal.
  - Examples: $p \vee \neg q$, $p \vee \neg q \vee \neg r$, $\neg p$.
- Positive unit clause: it contains one positive literal and no negative literal.

- The principle of linear resolution: reduce the clauses at each step by eliminating one of the two clauses which is used to calculate the resolvent.
- Horn Clause:
  - Any clause that has at most one positive literal.
  - Examples: $p \vee \neg q$, $p \vee \neg q \vee \neg r$, $\neg p$.
- Positive unit clause: it contains one positive literal and no negative literal.
- The goal of the linear resolution algorithm for Horn clauses is to demonstrate that a set of clauses is unsatisfiable.

- The principle of linear resolution: reduce the clauses at each step by eliminating one of the two clauses which is used to calculate the resolvent.
- Horn Clause:
  - Any clause that has at most one positive literal.
  - Examples: $p \vee \neg q$, $p \vee \neg q \vee \neg r$, $\neg p$.
- Positive unit clause: it contains one positive literal and no negative literal.
- The goal of the linear resolution algorithm for Horn clauses is to demonstrate that a set of clauses is unsatisfiable.
- This algorithm is complete for refutation.

Algorithm

- Let there be a set of clauses C'.

Algorithm

- Let there be a set of clauses C'.
- While $\square \notin C'$ and (there exist new possible choices)
  - Select two clauses: p is a positive unit clause and h is a clause containing $\neg p$.

Algorithm

- Let there be a set of clauses C'.
- While $\square \notin C'$ and (there exist new possible choices)
  - Select two clauses: p is a positive unit clause and h is a clause containing $\neg p$.
  - Calculate the resolving clause r from p and h.

Algorithm

- Let there be a set of clauses C'.
- While $\square \notin C'$ and (there exist new possible choices)
  - Select two clauses: p is a positive unit clause and h is a clause containing $\neg p$.
  - Calculate the resolving clause r from p and h.
  - Replace C' with $(C' - \{h\}) \cup \{r\}$.

Algorithm

- Let there be a set of clauses C'.
- While $\square \notin C'$ and (there exist new possible choices)
  - Select two clauses: p is a positive unit clause and h is a clause containing $\neg p$.
  - Calculate the resolving clause r from p and h.
  - Replace C' with $(C' - \{h\}) \cup \{r\}$.
- If $\square \in C'$ then C' is unsatisfiable.
  Otherwise C' is satisfiable.

Example

Demonstrate that the set C' is unsatisfiable. $C' : \{q \vee \neg r; \neg q \vee t; \neg t; r\}$.

Example

Demonstrate that the set C' is unsatisfiable. $C' : \{q \vee \neg r; \neg q \vee t; \neg t; r\}$.

1. Step 1: Positive unit clause r, resolve with $q \vee \neg r$: $Res(r, q \vee \neg r) = q$
   Update $C' \leftarrow \{q,\ \neg q \vee t,\ \neg t,\ r\}$

Example

Demonstrate that the set C' is unsatisfiable. $C' : \{q \vee \neg r; \neg q \vee t; \neg t; r\}$.

1. Step 1: Positive unit clause r, resolve with $q \vee \neg r$: $Res(r, q \vee \neg r) = q$
   Update $C' \leftarrow \{q,\ \neg q \vee t,\ \neg t,\ r\}$
2. Step 2: Positive unit clause q, resolve with $\neg q \vee t$: $Res(q, \neg q \vee t) = t$
   Update $C' \leftarrow \{q,\ \neg t,\ t,\ r\}$

Example

Demonstrate that the set C' is unsatisfiable. $C' : \{q \vee \neg r; \neg q \vee t; \neg t; r\}$.

1. Step 1: Positive unit clause r, resolve with $q \vee \neg r$: $Res(r, q \vee \neg r) = q$
   Update $C' \leftarrow \{q,\ \neg q \vee t,\ \neg t,\ r\}$
2. Step 2: Positive unit clause q, resolve with $\neg q \vee t$: $Res(q, \neg q \vee t) = t$
   Update $C' \leftarrow \{q,\ \neg t,\ t,\ r\}$
3. Step 3: Positive unit clause t, resolve with $\neg t$: $Res(t, \neg t) = \square$

Example

Demonstrate that the set C' is unsatisfiable. $C' : \{q \vee \neg r; \neg q \vee t; \neg t; r\}$.

1. Step 1: Positive unit clause r, resolve with $q \vee \neg r$: $Res(r, q \vee \neg r) = q$
   Update $C' \leftarrow \{q,\ \neg q \vee t,\ \neg t,\ r\}$
2. Step 2: Positive unit clause q, resolve with $\neg q \vee t$: $Res(q, \neg q \vee t) = t$
   Update $C' \leftarrow \{q,\ \neg t,\ t,\ r\}$
3. Step 3: Positive unit clause t, resolve with $\neg t$: $Res(t, \neg t) = \square$
4. Conclusion: C' is unsatisfiable.