Propositional Resolution
Introduction

Nilsson Book Handout

Professor Anita Wasilewska
CSE 537 Artificial Intelligence
Propositional Resolution
Part 1
SYNTAX “dictionary”

Literal – any propositional \textsc{variable} \(a\) or negation of a variable \(\neg a\), for \(a \in \text{VAR}\)

Example: variables: \(a, b, c\) .... negation of variables: \(\neg a, \neg b, \neg d\) ...

Positive Literal: any \textsc{variable} \(a \in \text{VAR}\)

Clause – any \textbf{finite set} of \textsc{literals}

Example: \(C_1, C_2, C_3\) are clauses where

\[
C_1 = \{a, b\} , \ C_2 = \{a, \neg c\} , \ C_3 = \{ a, \neg a, \ldots, a_k \}
\]
Syntax “Dictionary”

Empty Clause: \{\} is an empty set i.e. a clause without elements

Finite set of clauses

\[ \text{CL} = \{ C_1, \ldots, C_n \} \]

Example

\[ \text{CL} = \{\{a\}, \{\}, \{b, \neg a\}, \{c, \neg d\}\} \]
Semantics – Interpretation of Clauses

• Think *semantically* of a clause

• $C = \{ a_1, \ldots, a_n \}$ as *disjunction*, i.e.

  $C$ is logically equivalent to

  $a_1 \lor a_2 \lor \ldots \lor a_n \quad \text{where} \quad a_i \in \text{Literal}$

• **Formally** – given a truth assignment $v : \text{VAR} \rightarrow \{0, 1\}$

  we extended it to set of all CLAUSES $\text{CL}$ as follows:

  $v^* : \text{CL} \rightarrow \{0, 1\}$

  $v^*(C) = v^*(a_1) \lor \ldots \lor v^*(a_n)$

  for any clause $C$ in $\text{CL}$, where

  $0 – \text{False,} \quad 1 – \text{True}$

Shorthand : $v^* = v$
Example: let $v : \text{VAR} \rightarrow \{0, 1\}$ be such that
- $v(a) = 1$, $v(b) = 1$, $v(c) = 0$

and let

$C = \{a, \neg b, c, \neg a\}$

We evaluate:

$v(C) = v(a) \lor \neg v(b) \lor v(c) \lor \neg v(a) =$

$1 \lor 0 \lor 0 \lor 1 = 1$

**Observe** that $v(C) = 1$ for all $v$, i.e. the clause

$C = \{a, \neg b, c, \neg a\}$

is a **Tautology**
Satisfiability, Model, Tautology

Definitions

1. For any clause $C$, and any truth assignment $v$, we write $v \models C$ and say that $v$ satisfies $C \iff v(C) = 1$

2. Any $v$ such that $v \models C$ is called a model for $C$

3. A clause $C$ is satisfiable iff it has a model, i.e. $C$ is satisfiable iff there is a $v$ such that $v \models C$

4. A clause $C$ is a tautology iff $v \models C$ for all $v$, i.e. all truth assignments $v$ are models for $C$
Notations

• a, a, a is a finite sequence of 3 elements
• \{a, a, a\} = \{a\} is a finite set
• a, b, c \neq b, a, c are different sequences
• \{a, b, c\} = \{b, a, c\} are the same sets
• \{a, a, b, c\} is a multi–set (if needed)
Sets of Clauses CL

DEFINITIONS

1. A clause \( C \) is **unsatisfiable** iff it has no **MODEL**
i.e. \( v(C) = 0 \) for all truth assignments \( v \)

**Remark:** the empty clause \( \{\} \) is the only unsatisfiable clause

Let \( \text{CL} = \{ C_1, \ldots, C_n \} \) be a **finite set of clauses**.

2. We extended \( v : \text{VAR} \to \{0, 1\} \) to any set of clauses \( \text{CL} \)

\[
v(\text{CL}) = v(C_1) \land \ldots \land v(C_n)
\]

A finite set of clauses \( \text{CL} \) is semantically equivalent to a conjunction of all clauses in the set \( \text{CL} \)
Unsatisfiability

Definitions

1. A set of clauses $\mathbf{CL}$ is **satisfiable** iff it has a model, i.e. iff $\exists \upsilon \; \upsilon(\mathbf{CL}) = 1$

2. A set of clauses $\mathbf{CL}$ is **unsatisfiable** iff it does not have a model, i.e. iff $\forall \upsilon \; \upsilon(\mathbf{CL}) = 0$.

Remark:

If $\emptyset \in \mathbf{CL}$ then $\mathbf{CL}$ is unsatisfiable
Unsatisfiability

Consider a set of clauses
\[ CL = \{ \{a\}, \{a,b\}, \{\neg b\} \} \]

\( CL \) is satisfiable because any \( v \), such that \( v(a) = 1, v(b) = 0 \) is a model for \( CL \)

Check: \( v(CL) = 1 \land (1 \lor 0) \land 1 = 1 \)

**FACT:** When \( \{a\} \) and \( \{\neg a\} \) are in \( CL \), then the set \( CL \) is unsatisfiable

Remember: \((a \land \neg a)\) is a contradiction
Syntax and Semantics

• Example:
  - \( C_1 = \{ a, b, \neg c \}, \quad C_2 = \{ c, a \} \) - syntax
  - \( C_1 = a \cup b \cup \neg c \) - semantics
  - \( C_2 = c \cup a \) - semantics

• \( CL = \{ C_1, C_2 \} = \{ \{ a, b, \neg c \}, \{ c, a \} \} \) – syntax

\[ CL = (a \cup b \cup \neg c) \land (c \cup a) \] - semantics
Syntax and Semantics

Definitions:

CL is **satisfiable** iff there is \( v \), such that \( v(CL) = 1 \)

CL is **unsatisfiable** iff for all \( v \), \( v(CL) = 0 \)

- **CL = \{ C1,C2,\ldots,Cn\}** - syntax
- **CL = C1 \land \ldots \land Cn** - semantics
Semantical Decidability

• A statement:

  • “A finite set \( CL \) of clauses is/ is not satisfiable” is a decidable statement.

• \( CL \) has \( n \) propositional variables, hence we have \( 2^n \) possible truth assignments \( v \) to examine and evaluate whether \( v(CL) = 1 \) or \( v(CL) = 0 \)

• This is called **Semantical Decidability**

• **Problem:** Exponential complexity
Syntactical Decidability Method: Resolution Deduction

- **Goal**: We want to show that a finite set $CL$ of clauses is **unsatisfiable**
- **Method**: Resolution deduction:
  - **Start** with $CL$; apply a transformation rule called Resolution as long as it is possible.
  - **If** you get $\{\}$, then answer is **Yes**, i.e. $CL$ is unsatisfiable
  - **If** you **never** get $\{\}$, then answer is **NO**, i.e. $CL$ is satisfiable
Resolution Completeness Theorem 1

Completeness of the Resolution:

\( \text{CL is unsatisfiable} \iff \text{we obtain the empty clause} \ \{\} \ \text{by a multiple use of the Resolution Rule} \)

- Symbolically: \( \text{CL} \vdash \{\} \)
- It means we deduce the empty clause \( \{\} \) from \( \text{CL} \) by use of the resolution rule;
- We prove \( \{\} \) from \( \text{CL} \) by resolution
Resolution Complementeness Theorem 1

\(|=\) CL denotes CL is a tautology

\(|=\) CL denotes CL is unsatisfiable (contradiction)

• We write symbolically:

Resolution Completeness Theorem 1

\(|=\) CL iff CL \(\vdash\) \{\}
Refutation

- **Refutation:** proving the contradiction

In classical logic we have that:

A formula $A$ is a tautology iff $\neg A$ is a contradiction

Symbolically: $|\models A$ iff $|=|\neg A$

Observe:

$|\models (A_1 \land \ldots \land A_n \Rightarrow B)$ iff $|=|(A_1 \land \ldots \land A_n \land \neg B)$

Because $\neg (A \Rightarrow B) \equiv (A \land \neg B)$
Refutation

By Resolution Completeness Theorem this is almost equivalent to

\[ \vdash (A_1 \land \ldots \land A_n \rightarrow B) \iff (A_1 \land \ldots \land A_n \land \neg B) \vdash \{ \} \]

Almost- means not YET Resolution works for clauses not formulas!

The IDEA is the following:

to prove \( B \) from \( A_1, \ldots, A_n \) we keep \( A_1, \ldots, A_n \), ADD \( \neg B \) to it and use the Resolution Rule

If we get \( \{ \} \), we have proved \( (A_1 \land \ldots \land A_n \Rightarrow B) \)

It is called a proof by REFUTATION; to prove \( C \) we start with \( \neg C \) and if we get a contradiction \( \{ \} \), we have proved \( C \)
Formulas – Clauses

Resolution works only for clauses

To use Resolution Deduction we need to transform our formulas into clauses i.e. we need to prove the following

Theorem

For any formula $A \in F$, there is a set of clauses $CL_A$ such that $A$ is logically equivalent to the set of clauses $CL_A$

$CL_A$ is called a clausal form of the formula $A$

We have good set of rules for automatic transformation of $A$ into its clausal form and we will study it as next step
Completeness

• **Resolution Completeness 2**
  For any propositional formula $A$
  
  \[ \models A \text{ iff } \text{CL}_{\neg A} \vdash \{ \} \]
  
  where $\text{CL}_{\neg A}$ is the clausal form of $\neg A$

• **Resolution Proof** of $A$ definition:
  
  \[ \vdash_R A \text{ iff } \text{CL}_{\neg A} \vdash \{ \} \]

**Resolution Completeness 2:**

\[ \models A \text{ iff } \vdash_R A \]}
Resolution Rule  \( R \)

- \( C_1(a) \) means: clause \( C_1 \) contains a positive literal \( a \)
- \( C_2(\neg a) \) means: clause \( C_2 \) contains a negative literal \( \neg a \)

Resolution Rule  \( R \) (two Premises)

\[
C_1(a) : C_2(\neg a) \quad \text{Resolve on } a
\]

\[
(C_1-\{a\} \cup C_2-\{\neg a\}) \quad \text{Resolvent}
\]

Clauses \( C_1(a) \) and \( C_2(\neg a) \) are called a complementary pair
Resolution Rule

- **Resolution Rule** takes 2 clauses and returns one. We usually write it in a form of a graph:

- **Definition:** $C_1(a)$, $C_1(\neg a)$ is called a complementary pair.
- $C_1(a)$ $\quad$ $C_1(\neg a)$
  
  Resolve on $a$

  $$(C_1\setminus\{a\}) \cup (C_2\setminus\{\neg a\}) \leftarrow \text{Resolvent on } a$$
Resolution Rule R

• Clauses are SETS!
• \{C_1, C_2\} Complementary Pair

\[C_1 = \{a, b, c, \neg d\}\]
\[C_2 = \{\neg a, \neg b, d\}\]

Resolve on a

\[\{b, c, \neg d, \neg b, d\}\] Resolvent on a
Example

\[ C_1 = \{a, b, c, \neg d\} \quad C_2 = \{\neg a, \neg b, d\} \]

- Resolution Rule: R (Two Premises)
  - \( C_1(a) : C_2(\neg a) \)
  - Resolve on \( a \)
  - \( (C_1\{-a\} U C_2\{-\neg a\}) \leftarrow \text{Resolvent} \)
Exercise

• **CL** - set of clauses

**Find all resolvents of CL**

It means locate all clauses in **CL** that are Complementary Pairs and Resolve them

\[ C_1 = \{a, b, c, \neg d\} \quad \text{and} \quad C_2 = \{\neg a, \neg b, d\} \]

**CL** = \{C_1, C_2\} has 3 Complementary Pairs

\[ C_1(a), C_2(\neg a) \rightarrow P1 \]
\[ C_1(b), C_2(\neg b) \rightarrow P2 \]
\[ C_2(d), C_1(\neg d) \rightarrow P3 \]
Example

• \( CL = \{C_1, C_2\} = \{C_2, C_1\} \)

\( C_1 = \{a, b, c, \neg d\} \quad C_2 = \{\neg a, \neg b, d\} \)

Remember:

Resolution Rule uses **one literal** at the time!

\( C_1(a); C_2(\neg a) \text{ Resolve on } a : \text{ we get } \{b, c, \neg d, \neg b, d\} \)

\( C_1(b); C_2(\neg b) \text{ Resolve on } b : \text{ we get } \{a, c, \neg d, \neg a, d\} \)

\( C_1(d); C_2(\neg d) \text{ Resolve on } d : \text{ we get } \{a, b, c, \neg a, \neg b\} \)
Example

$C_1(b) : C_2(\neg b)$  
Pair $\{C_1, C_2\}$

$(C_1-\{b\}) U (C_2-\{\neg b\})$

\{a, b, c, \neg d\}  \{\neg a, \neg b, d\}$

Resolve on $b$

\{a, c, \neg d, \neg a, d\}  \leftarrow \text{Resolvent on } b
Example

\[ C_1(d) : C_2(\neg d) \quad \text{on} \quad \{C_1 \ C_2\} \]

\[(C_1-\{d\}) \cup (C_2-\{\neg d\})\]

\{a, b, c, \neg d\} ;\{\neg a, \neg b, d\}

Resolve on d

\{a, b, c, \neg a, \neg b\}
Example

\( C_1 = \{a, b, c, \neg d\}; C_2 = \{\neg a, \neg b, c, d\} \)

Resolve on \( b \)

\( \{a, c, \neg d, \neg a, d\} \)

Two clauses (one complementary pair) can have more than one resolvent – you can also resolve the complementary pair \( C_1 C_2 \) on \( a \)
Example

- We can also resolve $\{C_1 C_2\}$ on $a$
- $\{a, b, c, \neg d\}$ $\{\neg a, \neg b, d\}$

Resolve on $a$

$\{b, c, \neg d, \neg b, d\}$

These are all resolvent of pair $\{C_1 C_2\}$:

$\{b, c, \neg d, \neg b, d\}$, $\{a, c, \neg d, \neg a, d\}$

$\{a, b, c, \neg a, \neg b\}$
Resolution Deduction

- **CL** - set of clauses

Procedure: Deduce a clause \( C \) from **CL**: \( \text{CL} \vdash_R \{C\} \)

**Start** with **CL**, apply the **resolution** rule \( R \) to **CL**
**Add** resolvent to **CL** and
**Repeat** adding resolvents to already obtained set of resolvents
**until** you get \( C \)

**Example**

\( \text{CL} = \{\{a, b\}, \{\neg a, c\}, \{\neg b, c\}\} \)

\( R \) on \( a \) \( \{b, c\} \)

\( R \) on \( b \) \( \{c\} \)

\( \text{CL} \vdash_R \{c\} \)
Example

- \( CL = \{ \{ a, b \}, \{ \neg a, c \}, \{ \neg b, c \} \} \)

We have 2 possible deduction of \( \{ c \} \) from \( CL \)

\[ CL \vdash_R \{ c \} \]
Example

- $\text{CL} = \{\{a, b\}, \{\neg a, c\}, \{\neg b, c\}, \{\neg c\}\}$

  $\{b, c\} \{c\}$

  $\{\}$

  $\text{CL} \vdash R \{\}$

  $\text{CL is unsatisfiable}$ by Completeness Theorem

  $|=|\text{CL}$ iff $\text{CL} \vdash R \{\}$

Resolution deduction is not unique!

Next: Strategies for Resolution
Example

- CL = \{ \{ a, b \}, \{ \neg a, c \}, \{ \neg b, c \}, \{ \neg c \} \}

Another deduction of {} from CL
Exercise

• Let $\text{CL} = \{\{a, b\}, \{\neg a, c\}, \{\neg b, c\}\}$
Find all possible deduction from $\text{CL}$

Remember:
1. If you get $\{\}$, it means $\text{CL}$ is unsatisfiable.
2. If you never get $\{\}$, it means $\text{CL}$ is satisfiable.

1 and 2 is true by Completeness Theorem:

$$=| \text{CL} \iff \text{CL} \vdash \{\}$$

$\text{CL}$ is unsatisfiable iff there is a deduction of $\{\}$ from it

$\text{CL}$ is satisfiable iff there is NO deduction of $\{\}$ from it
Exercise

\[ \text{CL} = \{\{a, b\}, \{\neg a, c\}, \{\neg b, c\}\} \]

Derivation 1: \{\{a, b\}, \{\neg a, c\}, \{\neg b, c\}\}

R on a \{b, c\}

R on a \{c\}

R on b \{c\} \quad \text{STOP}

Derivation 2: \{\{a, b\}, \{\neg a, c\}, \{\neg b, c\}\}

R on b \{a, c\}

R on a \{c\}

R on a \{c\} \quad \text{STOP}

No more (possible) Derivations, i.e. by Completeness Theorem we have that \text{CL} is satisfiable
Exercise

- **CL is unsatisfiable** iff there is deduction of {} from it, i.e.
  \[ \text{CL} \vdash R \{ \} \]

- **CL is satisfiable** iff never \( \text{CL} \vdash R \{ \} \) (must cover all possibilities of deduction)

\[
\text{CL} = \{\{a, b\}, \{\neg b\}, \{a, c\}, \{\neg a, d\}\}
\]

This is just **one** derivation.
You must consider **ALL possible** derivations and show that none ends with {} to prove that **CL is satisfiable**
Exercise

• Given: \( CL = \{C_1, C_2, C_3, C_4\} \)
  \( CL = \{\{a, b, \neg b\}, \{\neg a, \neg b, d\}, \{a, b, \neg c\}, \{\neg a, c, b, e\}\} \)

1. Find all complementary pairs. Here they are:
   \( \{C_1, C_2\} \), \( \{C_1, C_4\} \),
   \( \{C_3, C_2\} \), \( \{C_2, C_3\} \),
   \( \{C_3, C_4\} \), \( \{C_2, C_4\} \)

2. Find all resolvents for your complementary pairs

For example: \( C_1 = \{a, b, \neg b\} \), \( C_2 = \{\neg a, \neg b, d\} \) has 2 resolvents.

Resolve on \( a \): \( \{\neg b, d, b\} \)

Resolve on \( b \):
   \( \{a, \neg a, d, \neg b\} \)
Exercise

• \( CL = \{C_1, C_2\} \), for \( C_1 = \{a, b, c, \neg d\} \), \( C_2 = \{\neg a, \neg b, d\} \)

\( CL \) has 3 resolvents:

1. \( \{\neg a, \neg b, a, b, c\} \) – resolve on \( d \)
2. \( \{\neg a, c, \neg d, d, a\} \) – resolve on \( b \)
3. \( \{b, c, \neg d, d\} \) – resolve on \( a \)

Let now \( CL = \{C_1, C_2, C_3\} \), for \( C_1 = \{a\} \), \( C_2 = \{b, \neg a\} \), \( C_3 = \{\neg b, \neg a\} \)

Exercise:

Find all Complementary Pairs + find all their resolvents
Propositional Resolution
Part 2
GOAL: Use Resolution to prove/ disapprove  \(|= A\)

PROCEDURE

Step 1: Write \(\neg A\) and transform \(\neg A\) info set of clauses \(CL_{\neg A}\) using Transformation rules

Step 2: Consider \(CL_{\neg A}\) and look at if you can get a deduction of \(\{\}\) from \(CL_{\neg A}\)

ANSWER

1. \(CL_{\neg A} \vdash R \{\}\) – Yes, \(|= A\)

2. \(CL_{\neg A} \vdash \{\}\) (i.e. you never get \(\{\}\)) – No, not \(|= A\)
Rules of transformation

• **Rules of transformation** of a formula $A$ into a logically equivalent set of clauses $\text{CL}_A$

• **Rule (U):** $(A \cup B) + \text{Information}$

What “Information” mean?

**Example:** $a, b, (a \cup \neg(\neg a \Rightarrow b)), \neg c$

$s, b, a, \neg(\neg a \Rightarrow b), \neg c$

$a, b, \neg c$ is Information

**Rule (U):** $I, (A \cup B), J$

$I, A, B, J$

$I, J$ --- Information around
Implication Rule (=>)

- I, (A=>B), J
  (A=>B)

I, ¬A, B, J
  ¬A, B

Example: a, (a U b), (a => ¬a), (a ∧ b), c
  (=>)

a, (a U b), ¬a, ¬a, (a ∧ b), c
  (U)

a, a, b, ¬a, ¬a, (a ∧ b), c

next step? we need (∧) Rule!
Conjunction Rule ($\land$)

Example:

\begin{align*}
\text{a, a, b, } \neg a, \neg a, \ (a \land b), \ c \\
\text{a, a, b, } \neg a, \neg a, \ a, \ c \\
a, \ a, \ b, \neg a, \neg a, \ b, \ c
\end{align*}

STOP when get only literals

Form clauses out of the leaves
Set of Clauses

Procedure: Leaves – to – Clauses

1. make **SETS** out of each leaf; each leaf becomes a **clause C**
2. make a set of clauses **CL** as a **set of all clauses C** obtained in 1.

   Leaf 1: \{a, a, b, \neg a, \neg a, a, c\} = \{a, b, \neg a, c\}
   Leaf 2: \{a, a, b, \neg a, \neg a, b, c\} = \{a, b, \neg a, c\}

• Observe that we end-up with only **one set of clauses**

• \[ CL = \{ \text{Leaf 1, Leaf 2} \} = \{ \{a, b, \neg a, c\} \} \]
Negation of Implication Rule ($\neg =>$)

$I, \neg (A => B), I^\prime
\neg (=>)
I, A, I^\prime, I, \neg B, I^\prime

Example:

$a, b, a, \neg (a => b), \neg c
\neg (=>)
a, b, a, \neg c
a, b, a, \neg b, \neg c

Stop – when only literals:
Form clauses out of $a, b, a, \neg c$ and $a, b, a, \neg b, \neg c$
Clauses

• Leaf1: $a, b, a, a, \neg c$ makes clause $\{a, b, \neg c\}$
• Leaf 2: $a, b, a, \neg b, \neg c$ makes clause $\{a, b, \neg b, c\}$

• $CL = \{\{a, b, \neg c\}, \{a, b, \neg b, c\}\}$

• $CL$ is set of clauses corresponding to $a, b, a, \neg (a \Rightarrow b), \neg c$
Negation of Disjunction Rule ($\neg U$)

I, $\neg(A \cup B)$, J

($\neg U$)

I, $\neg A$, J

I, $\neg B$, J

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

$\neg A$$

$\neg B$

• Coresponds to DeMorgan Law:

$$\neg(A \cup B) \equiv (\neg A \land \neg B)$$
Negation of Conjunction Rule ($\neg \land$)

$I, \neg(A \land B), J$  $\neg(A \land B)$

$I, \neg A, \neg B, J$  $\neg A, \neg B$

Corresponds to DeMorgan Law

$\neg(A \land B) \equiv (\neg A \cup \neg B)$
Negation of Negation Rule ($\neg\neg$)

$I, \neg\neg (A), J \quad \neg\neg(A)$

$I, A, J \quad (\neg\neg) \quad A$

Corresponds to
$\neg\neg (A) \equiv A$

Transformation Rules:
$(\land), (\lor), (\Rightarrow), (\neg\land), (\neg\lor), (\neg\Rightarrow)$
Transformation Rules Shorthand Form

(AUB) (U)
A, B

¬A

¬B

¬(A ∧ B) (∨)
A, B

¬(A ∧ B) (¬∧)
¬A, ¬B

¬(A => B) (¬=>)
A, B

¬¬A (¬¬)
A

+ Keep all information

End when all leaves are literals
Example

• Let A be a Formula \(((a\rightarrow \neg b)\lor c) \land (\neg a \lor \neg b)\)

• Find \(C_{LA}\)

• \(((a\rightarrow \neg b)\lor c) \land (\neg a \lor \neg b)\)

\((a\rightarrow \neg b)\lor c\) \quad (\neg a \lor \neg b)

\(a\rightarrow \neg b\), c \quad \{\neg a , b \} \text{ STOP}

\{\neg a , \neg b , c\} \text{ STOP}

\[\mathcal{C}_{LA} = \{\{\neg a , \neg b , c\} , \{\neg a , b \}\}\]

\(A \equiv \mathcal{C}_{LA}\)
ARGUMENTS (rules of inference)

• From (premises) $A_1, \ldots, A_n$ we conclude $B$

\[
\frac{A_1, \ldots, A_n}{B}
\]

Definition:

Argument $A_1, \ldots, A_n$ is VALID iff

\[| = ((A_1 \land \ldots \land A_n) \Rightarrow B)\]
ARGUMENTS

• Otherwise
  Argument is **NOT VALID**

Valid Arguments $\equiv$ Tautologically Valid

$A_1, \ldots, A_n, C$

are formulas of **Propositional or Predicate Language**
Validity of Arguments

Remember: \( |= \text{A} \iff |= \neg \text{A} \)

Tautology (always true), Contradiction (always false)

This means that if we want to decide \( |= \text{A} \) we decide \( |= \neg \text{A} \)
and use Resolution for that

**STEPS**

**Step 1:** Negate \( \text{A} \); i.e. take \( \neg \text{A} \) and find the set of clauses corresponding to \( \neg \text{A} \) i.e. find \( \text{CL}_{\neg \text{A}} \)

**Step 2:** Use Completeness of Resolution

\[ |= \text{A} \iff \text{CL}_{\neg \text{A}} \vdash_{R} {} \]

i.e.

1. Look for a deduction of \( {} \)
2. If YES – we have \( |= \text{A} \)
3. If there is no deduction of \( {} \) we have: \( |= \text{A} \)
Basic Theorems

T1.  \( \models \text{CL} \iff \text{CL} \vdash_R \{\} \)

\( \text{CL} \) is inconsistent iff there is a resolution deduction of \( \{\} \) from \( \text{CL} \)

T2. For any formula \( A \), there is a set of clauses \( \text{CL}_A \) such that \( A \equiv \text{CL}_A \)

T3.  \( \models A \iff \models \neg A \)

By T2 we get that

\( \models A \iff \models \text{CL}_{\neg A} \)

And by T1 and T3 we get

T4.  \( \models A \iff \text{CL}_{\neg A} \vdash_R \{\} \)
Exercise

• **Prove** By Propositional Resolution

  • $|= (- (a \Rightarrow b) \Rightarrow (a \land \neg b))$

**Remember:** $|= A$ iff $|= \neg A$ + use Resolution

**Steps**

**Step 1:** Find set of clauses corresponding to $\neg A$

  i.e. $CL_{\neg A}$

**Step 2:** Find deduction of $\{\}$ from $CL_{\neg A}$

  i.e. show that $CL_{\neg A} \vdash_R \{\}$

DO IT!
Exercise Solution

• **Step 1:** Negate $A$ and find the set of clauses for $\neg A$
  i.e. $\text{CL}_{\neg A}$

• $\neg (\neg (a \Rightarrow b) \Rightarrow (a \land \neg b))$

\[
\begin{align*}
\neg(a \Rightarrow b) & \quad \neg(a \land \neg b) \\
\neg \neg (a \Rightarrow b) & \quad \neg a, \neg \neg b \\
a & \quad \neg b & \quad \neg a, b
\end{align*}
\]

$\{a\} \quad \{\neg b\} \quad \{\neg a, b\}$

\[
\text{CL}_{\neg A} = \{\{a\}, \{\neg b\}, \{\neg a, b\}\}
\]

Step 2: Check if $\text{CL}_{\neg A} \vdash_{R} \{\} \rightarrow \text{YES!}$

\[
\{b\}
\]

Remark: $\models A$ iff there is no deduction of $\{\}$ from $\text{CL}_{\neg A}$
Back To Arguments

• Use resolution to show that from $A_1, \ldots, A_n$ we can deduce $B$

“We can” deduce $B$ from $A_1, \ldots, A_n$ means validity of argument $A_1, \ldots, A_n \Rightarrow B$

iff by definition

$|\models (A_1 \land \ldots \land A_n \Rightarrow B)$

We have to use **Resolution** to prove that this is a **Tautology**
Arguments

\[ |-- (A_1 \land \ldots \land A_n \Rightarrow B) \iff

\neg |-- (A_1 \land \ldots \land A_n \Rightarrow B) \iff

\neg |-- (A_1 \land \ldots \land A_n \land \neg B) \]

**Step 1:** we transform \((A_1 \land \ldots \land A_n \land \neg B)\) to clauses

**Step 2:** examine whether \(CL \vdash_r \{\}\)
Remember

• Argument $A_1, \ldots, A_n$ is valid iff

$$B \quad Cl_{A_1} U \ldots U Cl_{A_n} U Cl_{\neg B} \vdash \{\}$$

\[\downarrow\]

Argument is not valid iff never $Cl_{A_1} U \ldots U Cl_{A_n} U Cl_{\neg B} \vdash \{\}$

We have some Resolution Strategies that allow us to cut down number of cases to consider
Example

Check if you can deduce

\[ B = (\neg(a \cup \neg b) \Rightarrow (\neg a \land b)) \]

from \( A_1 = ((a \Rightarrow \neg b) \Rightarrow a) \) and \( A_2 = (a \Rightarrow (b \Rightarrow a)) \)

Procedure:
1. Find \( \text{CL}_{A_1} \), \( \text{CL}_{A_2} \) and \( \text{CL}_{\neg B} \)
2. Form \( \text{CL} = \text{CL}_{A_1} \cup \text{CL}_{A_2} \cup \text{CL}_{\neg B} \)
3. Check if \( \text{CL} \vdash_R \{\} \) or if never \( \text{CL} \nvdash_R \{\} \)

Yes, we can \hspace{5cm} \text{No, we can’t}
Example Solution

A1 = ((a => ¬b) => a)

We get: \( \text{CL}_{A1} = \{\{a\}, \{b, a\}\} \)

A2 = ((a => (b=>a))

We get: \( \text{CL}_{A2} = \{\neg a, \neg b, a\} \)
Example Solution

• \( \neg B = \neg((\neg(a \lor \neg b) \Rightarrow (\neg a \land b)) \)

\( \neg(a \lor \neg b) \)

\( \neg a \quad \neg \neg b \)

\( b \quad \neg a, \neg b \quad a, \neg b \)

\( \text{CL} = \{\{a\},\ \{b,\ a\},\ \{\neg a,\ \neg b,\ a\},\ \{\neg a\},\ \{b\},\ \{a,\ \neg b\}\} \)

Remove Tautology Strategy gives us the set

\( \text{CL} = \{\{a\},\ \{b,\ a\},\ \{\neg a\},\ \{b\},\ \{a,\ \neg b\}\} \)
Example Solution

- \( \text{CL} = \{\{a\}, \{b, a\}, \{\neg a, \neg b, c\}, \{\neg a\}, \{b\}, \{a, \neg b\}\} \)

\[ \{a\} \quad \text{R on} \quad \{b\} \quad \{\} \]

Yes Argument is Valid

Next: Strategies for Resolution
Propositional Resolution
Part 3
Resolution Strategies

- We present here some **Deletion Strategies** and discuss their **Completeness**.

**Deletion Strategies** are restriction techniques in which **clauses** with specified properties are **eliminated** from set of clauses **CL** before they are used.
Pure Literals

Definition
A literal is pure in CL if it has no complementary literal in any other clause in CL.

Example: CL = { {a, b}, {¬c, d}, {c,b}, {¬d} }
a, b are pure and c, d, ¬c, ¬d are not pure

c has complement literal ¬c in {¬c, d} and
¬c has complement literal c in {c,b}
d has a complement literal ¬d in the clause {¬d} and
¬d has a complement literal d in {¬c, d}
S1: Pure Literals Deletion Strategy

S1 Strategy: Remove all clauses that contain Pure Literals

Clauses that contain pure literals are useless for retention process.

One pure literal in a clause is enough for the clause removal

This Strategy is complete, i.e.

\[ \text{CL} \vdash \{\} \iff \text{CL}' \vdash \{\} \]

where \( \text{CL}' \) is obtained from \( \text{CL} \) by pure literal clauses deletion
Example

- $CL = \{-a, -b, c\}, \{-p, d\}, \{-b, d\}, \{a\}, \{b\}, \{-c\}$
  - $d, \neg p$ are pure,

$CL' = \{-a, -b, c\}, \{a\}, \{b\}, \{-c\}$
**S2. Tautology Deletion Strategy**

- **Tautology** – a clause containing a pair of complementary literals (a and ¬a)
- **S2: Tautology Deletion:**
  
  \[ \text{CL'} = \text{Remove all Tautologies from } \text{CL} \]

- **Example:**
  - \[ \text{CL} = \{ \{a, b, ¬a\}, \{b, ¬b, c\}, \{a\} \} \]
  - \[ \text{CL'} = \{\{a\}\} \]

- **Tautology Deletion Strategy S2 is COMPLETE.**
  
  \[ \text{CL is satisfiable } \equiv \text{CL'} is satisfiable \]
  
  \[ \text{CL unsatisfiable } \equiv \text{CL'} unsatisfiable \]
Exercise

• Example:

• $CL = \{\{a, \neg a, b\}, \{b, \neg b, c\}\}$ - remove tautologies;
  
  $CL'$ has no elements, i.e. $CL' = \emptyset$

$CL$ is always satisfiable and so is $CL'$ as $\emptyset$ is always satisfiable!

Exercise

Prove correctness of Tautology Deletion Strategy
S3. Unit Resolution Strategy

- **A unit resolvent** – resolvent in which at least one of the parent clauses is a **unit clause** i.e. is a clause containing a single literal.

- **A unit deduction** – all derived clauses are **unit resolvents**.

- **A unit Refutation** – unit deduction of the empty clause { }.

- Example: { {a, b}, {¬a, c}, {¬b, c}, {¬c} }  
  { ¬a }  
  { ¬b }  
  { b }  
  { }  

Efficient but not Complete!
Unit Resolution not complete

Example

- \( \text{CL} = \{\{a, b\}, \{-a, b\}, \{a, -b\}, \{-a, -b\}\} \)

\( \{b\} \)  
\( \{a\} \)  
\( \{-a\} \)  
\( \{\}\)  

\( \text{CL is unsatisfiable, but does not have unit deduction.} \)

Horn Clause: a clause with at most one positive literal.

Theorem: Unit Resolution is complete on Horn Clauses.
Example of Unit Resolution Deduction

- **CL** = \{\{-a, c\}, \{-c\}, \{a, b\}, \{-b, c\}, \{-c\}\}

  - \{-a\}
  - \{b\}
  - \{c\}

  \{\}

**CL** is not Horn but **CL\{\}\} by unit deduction.

Remark: if we get \{\}\} by unit deduction we are OK but if we don’t get \{\}\} by unit deduction it does not mean that **CL** is satisfiable, because unit strategy is not a Complete Strategy on non-Horn clauses.
S4. Input Resolution

• **Input Resolution**- At least one of the two parent clauses is in the initial database.

• **Input Deduction**- all derived clauses are input resolvents

• **Input Refutation**- Input deduction of \{\}

**THM 1:** Unit and Input Resolution are equivalent.

**THM 2:** Input Resolution is complete only on Horn Clauses
Example: \[ \text{CL} = \{\{a, b\}, \{-a, c\}, \{-b, c\}, \{-c\}\} \]
5. Linear Resolution

- **Linear Resolution** also called Ancestry-Filtered resolution is a slight generalization of Input Resolution.

- **A Linear Resolution**: At least one of the parents is either in the initial DB or is in an Ancestor of the other parent.

- **A Linear Deduction**: Uses only linear resolvents: each derived clauses is a linear resolvent

- **A Linear Refutation**: Linear deduction of { }. Linear Resolution is complete
Example

\( CL = \{\{a, b\}, \{-a, b\}, \{a, -b\}, \{-a, -b\}\} \)

Here:

\{a\} is a parent of \{-b\}

\{b\} is the ancestor of \{-b\} (other parent of \{-b\})
Linear Resolution

Linear Resolution is complete

There are also more modifications of the LR that are complete

Our Strategies work also for Predicate Logic Resolution
First papers

**Kowalski** 1974, 1976 “Logic for problem solving” “Predicate Logic as a programming language”.

**Robinson** 1965 “A Machinery Oriented logic based on the resolution principle” J Assoc. for Computing Machinery 12(1)