Propositional Resolution
Part 1

Short Review
Professor Anita Wasilewska
CSE 352 Artificial Intelligence
SYNTAX “dictionary”

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

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

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

Clause – any finite set of literals

Example- C1, C2, C3 are clauses where

\[
\text{C1} = \{a, b\}, \text{C2} = \{a, \neg c\}, \text{C3} = \{a, \neg a, \ldots, a_k\}
\]
Syntax "Dictionary"

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

Finite set of clauses

\[ \text{CL} = \{ \text{C}1, \ldots, \text{C}n \} \]

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
  
  \( a_1 \lor a_2 \lor \ldots \lor a_n \quad a_i \in \text{Literal} \)

• Formally – a truth assignment \( v : \text{VAR} \rightarrow \{0, 1\} \) we extended it to set of all CLAUSES \( \text{CL} \) as follows:
  
  \( v^* : \text{CL} \rightarrow \{0, 1\} \) we extend
  
  \( v^*(C) = v^*(a_1) \lor \ldots \lor v^*(a_n) \)

  for any clause \( C \) in \( \text{CL} \), where
  
  0 – False, \quad 1 – True

  Shorthand : \( v^* = v \)
Satisfiability, Model, Tautology

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

\( C = \{a, \neg b, c, \neg a\} \)

We evaluate:
\[
v(C) = v(a) \cup \neg v(b) \cup v(c) \cup \neg v(a) = 1 \cup 0 \cup 0 \cup 1 = 1
\]

**OBSERVE** that \( v(C) = 1 \) for all \( v \), i.e.
\[
C = \{a, \neg b, c, \neg a\} \quad \text{is a Tautology}
\]
Satisfiability, Model, Tautology

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

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

A clause $C$ is satisfiable iff it has a model, i.e.

$C$ is satisfiable iff there is a $v$ such that $v \models C$.

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

• $a, a, a$ (finite sequence of 3 elements)
• $\{a, a, a\} = \{a\}$ finite set
• $a, b, c \neq b, a, c$ (different sequences)
• $\{a, b, c\} = \{b, a, c\}$ (same sets)
• $\{a, a, b, c\}$ (multi - sets)
Sets of Clauses CL

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 $CL = \{ C_1, ..., C_n \}$ be a finite set of clauses.

We extended $v : VAR \rightarrow \{0, 1\}$ to any set of clauses $CL$

$v(CL) = v(C_1) \land ... \land v(C_n)$

A finite set of clauses $CL$ is semantically equivalent to a conjunction of all clauses in the set $CL$. 
Unsatisfiability

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

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

Remark:

If $\{\} \in \text{CL}$ then $\text{CL}$ is unsatisfiable.
Unsatisfiability

Consider a set of clauses

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

\text{CL} is \textbf{satisfiable} because \( v \), such that \( v(a) = 1 \), 
\( v(b) = 0 \) is a \textbf{model} for \text{CL}

Check: \( v(\text{CL}) = 1 \land (1 \cup 0) \land 1 = 1 \)

Remark: When \( \{a\} \) and \( \{\neg a\} \) are in it, then the set \text{CL} is \textbf{unstisfiable}

Remember: \( (a \land \neg a) \) is a contradiction
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 \( \nu \), such that \( \nu(\text{CL}) = 1 \)

CL is **unsatisfiable** iff for all \( \nu \), \( \nu(\text{CL}) = 0 \)

- **CL** = \{C1, C2, ...... , Cn\} - syntax
- **CL** = C1 \( \wedge \) ...... \( \wedge \) Cn - semantics
Semantical Decidability

• A statement:
  • “A finite set $CL$ of clauses is/not satisfiable” is a **decidable statement**.

• $CL$ has a $n$ propositional variables, hence we have $2^n$ possible truth assignments $v$ to examine and we can check them all by Truth Tables.

• This is called **Semantical Decidability**

• Problem: Exponential complexity
Syntactical Decidability Method: Resolution Deduction

• **Goal**: We want to show that a finite set $\mathbf{CL}$ of clauses is **unsatisfiable**.

• **Method**: Resolution deduction:

  • **Start** with $\mathbf{CL}$; apply a transformation rule called **Resolution** as long as it is possible.

  • **If** you get $\{\}$, **then** answer is **Yes**, i.e. $\mathbf{CL}$ is **unsatisfiable**

  • **If** you never get $\{\}$, **then** answer is **NO**, i.e. $\mathbf{CL}$ is **satisfiable**.
Resolution Completeness Theorem 1

• Completeness of the Resolution:
• **CL** is **unsatisfiable** iff we obtain the empty clause **\{\}** by a multiple use of the **Resolution Rule**

• Symbolically: **CL \rhd \{\}** means:
• deduce **\{\}** from **CL** by resolution rule;
• prove **\{\}** from **CL** by resolution
Resolution Completeness Theorem 1

$|=$ CL denotes CL is a tautology

$|=|$ CL denotes CL is unsatisfiable

• Completeness 1 of the Resolution:

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

Completeness for a certain proof system S

$|=$ A iff $\vdash$ A
Refutation

**Refutation:** proving the contradiction

In classical logic we have that:

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

Symbolically:

$$|= A \iff =| \neg A$$

Observe:

$$|= (A_1 \land \cdots \land A_n \Rightarrow B) \iff =|(A_1 \land \cdots \land A_n \land \neg B)$$
Refutation

By Resolution Completeness Theorem this is (almost, i.e. we need clauses not formulas!) equivalent to

$$|= (A_1 \land \ldots \land A_n \Rightarrow B) \ \text{iff} \ (A_1 \land \ldots \land A_n \land \neg B) \vdash \{\}$$

It means that
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 \( B \).

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

Resolution works only for clauses!
To use it we need to transform our formulas into clauses. i.e. we 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 $A$.

We have good set of Rules for Automatic of Transformation of $A$ into the set of clauses and we will study it as next step.
Completeness

- Resolution Completeness 2:
  - $|= A$ iff $\text{CL}_{\neg A} \vdash \{\}$
  - $\text{CL}_{\neg A} = \text{clausal form of } \neg A$.
- Resolution Proof of $A$ definition:
  - $\vdash_R A$ iff $\text{CL}_{\neg A} \vdash \{\}$

Resolution Completeness 2:

$|= A$ 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)

\[ C_1(a) : C_2(\neg a) \quad \text{Resolve on a} \quad (C_1\{-a\} \cup C_2\{-\neg a\}) \quad \text{Resolvent} \]
Resolution Rule: R

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

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

\[\text{Resolve on } a\]
\[\{b, c, \neg d, \neg b, d\}\] (Resolvent on a)
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 a **Complementary Pair**

• $C_1(a) \quad C_1(\neg a)$

  Resolve on $a$

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

- **CL** - set of clauses

Find all resolvents of **CL** means: locate all clauses in **CL** that are Complementary Pairs and Resolve them

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

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

- \( C_1(a), C_2(\neg a) \) – P1
- \( C_1(b), C_2(\neg b) \) – P2
- \( C_2(d), C_1(\neg d) \) – P3
Example

- \{C_1(a) \ , C_2(\neg a)\}

\{\}

\{a\} \land \{\neg a\} = \phi \equiv \{\}

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

\{b ,c ,d\} (never \{\} from\{C_1, C_2\})

- Resolution Rule: R (Two Premises)
  
  \underline{C_1(a) : C_2(\neg a)} \quad \text{Resolve on a}

  (C_1-\{a\} \cup C_2-\{\neg a\}) \quad \text{\textless - Resolvent}
Example

- \( C_1 = \{a, b, c, \neg d\} \quad C_2 = \{\neg a, \neg b, d\} \)
- \( CL = \{C_1, C_2\} = \{C_2, C_1\} \) we have more than 1 resolvent!

- **Resolve on a:** We get \( \{b, c, \neg d, \neg a, d\} \)
- **Resolve on b:** We get \( \{a, c, \neg d, \neg a, d\} \)
- **Resolve on d:** We get \( \{a, b, c, \neg a, \neg b\} \)

All resolvents of \( CL \)
Example

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

Remember: Resolution Rule uses one literal at the time!

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

• We can also resolve PAIR P2 on a
  \{a, b, c, \neg d\} ; \{\neg a, \neg b, d\} \quad \{C_1 C_2\}

  Resolve on a

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

  These are all resolvent of pair P2.
Example

\( C_1(d) : C_2(\neg d) \) on Pair P3

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

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

Resolve on d

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

\[ C_1(b) : C_2(\neg b) \]

Pair P2 \[ \{C_1, C_2\} \]

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

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

Resolve on \( b \)

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

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

Resolve on b

\[ \{a, c, \neg d, \neg a, d\} \]

Resolvent on b

Two clauses can have more than one resolvent (one complementary pair) – you can also resolve \( C_1 \) \( C_2 \) on \( d \)
Resolution Deduction

- **CL** - set of clauses
  Deduce **C** from **CL**
- **CL ⊢ R C**

**Procedure:**
- **Start** with **CL**, apply the resolution rule **R** to **CL**
- **Add** resolvent to **CL** (Data base) and
- **Repeat** adding resolvents to already obtained Data base
- **until** you get **C**.

**CL** = {{ a, b}, {¬ a, c}, {¬ b, c}}

**R** on **a** {b, c}

**R** on **b** { c }

**CL ⊢ R** {c}
Example

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

\[ \text{Resolve on } b \]
\[ \{ a, c \} \]

\[ \text{Resolve on } a \]
\[ \{ c \} \]

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

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

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

\( \{b, c\} \)

\( \{c\} \)

\( \{\} \)

\( \text{CL} \vdash \{\} \)

\( \text{CL is unsatisfiable} \) by completeness theorem:

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

Resolution deduction is not unique!

–see another on next slide.

Next: Strategies for Resolution
Example

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

Another deduction of \{\}\ from **CL**.
Exercise

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

Find all possible deduction from $CL$

Remember:

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

1 and 2 is true by Completeness Theorem

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

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

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

Derivation 1:

\[ \{ \{ a, b \}, \{ \neg a, c \}, \{ \neg b, c \} \} \]

on a \quad \{ b, c \}

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

Derivation 2:

\[ \{ \{ a, b \}, \{ \neg a, c \}, \{ \neg b, c \} \} \]

on b \quad \{ a, c \}

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

No more (possible) Derivations, i.e. by Completeness Theorem

CL is satisfiable
Exercise

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

**CL** is satisfiable iff **CL** \[ \vdash \text{R} \{\} \] (must cover all possibilities of deduction)

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

This is one derivation.
You must consider \textbf{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\}, C_1 = \{a\}, C_2 = \{b, \neg a\},
C_3 = \{\neg b, \neg a\}

Exercise:
  - Find all Complementary Pairs + find all their resolvents
Exercise Solution

CL contains 3 Complementary Pairs, each has one resolvent.

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

\{b\} \quad \text{resolve on } a

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

\{-b\} \quad \text{resolve on } a

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

\{-a\} \quad \text{resolve on } b

Complementary Pair:

\(C_1(x) ; C_2(\neg x)\)