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 $\neg a$, $a \in VAR$,

Example - variables: a, b, c, negation of variables: ¬a, ¬b, -d …

Positive Literal – any variable $a \in VAR$

Clause – any **finite set** of **literals**

Example- C1, C2, C3 are clauses where

$C1 = \{a, b\}$, $C2 = \{a, \neg c\}$, $C3 = \{ a, \neg a, \ldots\ldots, a_k \}$

Syntax "Dictionary"

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

Finite set of clauses

CL = { C1, …., Cn}

Example

CL = {{a}, { }, { b, ¬a} , {c, ¬ d}}

Semantics – Interpretation of Clauses

- Think semantically of a clause
- $C = \{ a_1, \ldots\ldots, a_n\}$ as disjunction, i.e.

  C is logically equivalent to

  $a_1 \cup a_2 \cup \ldots\ldots \cup a_n$ $a_i \in$ Literal
- Formally – a truth assignment $v : VAR \to \{0, 1\}$ we extended it to set of all CLAUSES **CL** as follows:

  $v^* :$ **CL** $\to \{0, 1\}$ we extend

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

for any clause C in **CL**, where

0 – False, 1 – True

Shorthand : $v^* = v$

Satisfability, Model, Tautology

Example: let v : VAR -> {0, 1} be such that

- v(a) = 1, v(b) = 1 , v(c)= 0

  C = { a, ¬ b, c, ¬a}

  We evaluate :

v(C) = v(a) U ¬v(b) U v(c) U ¬v(a) =

1 U 0 U 0 U 1 = 1

OBSERVE that v(C) =1 for all v, i.e.

C = { a, ¬ b, c, ¬a} is a **Tautology**

Satisfability, Model, Tautology

For any clause **C**, and any truth assignment **v** we write **v I= C** and say that **v satisfies C** iff **v(C) =1**

Any v such that **v I= C is called a MODE L 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 I= C

A clause C is a **tautology** iff **v I= 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 ≠ 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** = { $C_1, \ldots, C_n$) be a **finite set of clauses**.

We extended v : VAR -> {0, 1} to any set of clauses **CL**

$v(\mathbf{CL}) = v(c_1) \wedge \ldots\ldots \wedge V(c_n)$

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

Unsatisfiability

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

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

Remark:

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

Unsatisfability

Consider a set of clauses

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

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

Check: $v(CL) = 1 \wedge (1 \cup 0) \wedge 1 = 1$

Remark: When $\{a\}$ and $\{\neg a\}$ are in it, then the set **CL** is **unstisfiable**

Remember: $(a \wedge \neg a)$ is a contradiction

Syntax and Semantics

- Example:
- C1 = { a, b, ¬c}, C2 ={c , a} - syntax
- C1 = a U b U ¬c - semantics
- C2 = c U a - semantics

- **CL** = {C1, C2} = {{a , b, ¬c} , {c , a}} – syntax

CL = (a U b U ¬c) ∧ (c U 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,…… ,Cn} - synatx
- **CL** = C1 ∧ …… ∧ 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 **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:**
- **CL** is **unsatisfiable** iff we obtain the empty clause **{}** by a multiple use of the **Resolution Rule**
- Symbolically: **CL ⊢ {}** means:
- deduce **{}** from **CL** by resolution rule;
- prove **{}** from **CL** by resolution

Resolution Completeness Theorem 1

$\models$ CL **denotes** **CL is a tautology**

$=\mid$ CL **denotes** **CL is unsatisfiable**

- **Completeness 1 of the Resolution:**

$=\mid$ CL iff **CL $\vdash$ {}**

Completeness for a certain proof system S

$\models$ A iff $\vdash$ A

Refutation

- **Refutation: proving the contradiction**

In classical logic we have that:

a formula A is a tautology iff ¬A is a contradiction

Symbolically:

$$\models A \quad \text{iff} \quad \dashv \neg A$$

Observe:

$$\models (A1 \wedge \ldots\ldots \wedge An \Rightarrow B) \quad \text{iff} \quad \dashv (A1 \wedge \ldots\ldots \wedge An \wedge \neg B)$$

Refutation

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

$\models (A1 \wedge \ldots\ldots \wedge An \Rightarrow B)$ iff $(A1 \wedge \ldots\ldots \wedge An \wedge \neg B) \vdash \{\}$

It means that

to prove B from A1, ….,An we keep A1,……, An , ADD ¬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 ¬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 $\mathbf{CL}_A$ such that A is logically equivalent to the set of clauses $\mathbf{CL}_A$

$\mathbf{CL}_A$ 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:**
- $\models A$ iff $\mathbf{CL}_{\neg A} \vdash \{\}$
- $\mathbf{CL}_{\neg A}$ = clausual form of $\neg A$.
- **Resolution Proof of A definition:**

$\vdash_R \mathbf{A}$ **iff** $\mathbf{CL}_{\neg A} \vdash \{\}$

Resolution Completeness 2:

$\models A$ iff $\vdash_R \mathbf{A}$

Resolution Rule: R

- $\mathbf{C_1(a)}$ means: clause $\mathbf{C_1}$ contains a positive literal **a**
- $\mathbf{C_2(\neg a)}$ means: clause $\mathbf{C_2}$ contains a negative literal **¬a**
- **Resolution Rule: R** (Two Premises)

$\underline{C_1(a) : C_2(\neg a)}$ **Resolve on a**

$(C_1 - \{a\} \cup C_2 - \{\neg a\})$ <- **Resolvent**

Resolution Rule: R

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

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

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
- $C_1(a)$ $\qquad$ $C_1(\neg a)$

Resolve on a

$(C_1\text{-}\{a\})\ \mathrm{U}\ (C_2\text{-}\{\neg a\})$ **<-** 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

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

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

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

(tranform)

$\{ \}$

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

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

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

- **Resolution Rule: R (Two Premises)**

<u>$C_1(a) : C_2(\neg a)$</u> **Resolve on a**

$(C_1 - \{a\} \cup C_2 - \{\neg a\})$ <- Resolvent

Example

- $C_1 = \{a, b, c, \neg d\}$ $C_2 = \{\neg a, \neg b, d\}$
- **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, ¬ d} ;{¬ a ,¬b ,d}　　　　　　$\{C_1\ C_2\}$

Resolve on a

{b , c, ¬d, ¬ b, d}

These are all resolvent of pair P2.

Example

<u>$C_1(d) : C_2(\neg d)$</u> on Pair P3

$(C_1\text{-}\{d\})\ U\ (C_2\text{-}\{\neg d\})$

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

Resolve on d

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

Example

$C_1(b) : C_2(\neg b)$ Pair P2 $\{C_1\ C_2\}$

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

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

Resolve on b

$\{a, c, \neg d, \neg a, d\}$ <- Resolvent on b

Example

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

Resolve on b

$\{a, c, \neg d, \neg a, d\}$ 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** $\vdash_R$ 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** $\vdash_R$ {c}

Example

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

Resolve on b

$\{a, c\}$

Resolve on a

$\{c\}$

We have 2 possible deduction of { c } from **CL**

$\mathbf{CL} \vdash_R \{c\}$

Example

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

{b,c}

{c}

{} **CL ⊢ {}**

CL is unsatisfiable by completeness theorem :

$\models$ **CL** iff **CL** ⊢ {}

Resolution deduction is not unique!

–see another on next slide.

Next: Strategies for Resolution

Example

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

{¬a}

{b}

{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**

$\models$ **CL** iff **CL** $\vdash \{\ \}$

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

CL is satisfiable iff there is NO deduction of {} From it.

Exercise

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

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

on a $\{b, c\}$

$\{c\}$ on b **STOP**

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

on b $\{a, c\}$

$\{c\}$ on a **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.

  CL $\vdash_R$ {}

CL is satisfifable iff **CL** $\vdash_R$ {} (must cover all possibilities of deduction)

CL = {{ a, b},{¬b},{a, c},{ ¬a, d}}

{a}

{b, d}

{c,d} STOP

This is 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 ,¬ b}, {¬ a ,¬ b, d},{a ,b , ¬c}, {¬ 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 ,¬ b} , C_2 = {¬a , ¬b , d} has 2 resolvents.

Resolve on a: {¬b, d, b}

Resolve on b;

{a, ¬a, d ,¬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}   {b, ¬a}

{b}   resolve on a

{a}   {¬b, ¬a}

{¬b}   resolve on a

{b, ¬a}   {¬b, ¬a}

{¬a}   resolve on b

Complementary Pair:

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