

Tut 6

CNF & DNF + Resolution

Conjunction:

$$A_1 \wedge A_2 \wedge \dots \wedge A_n$$

association to the left

means

use paranthesis

$n \geq 3$

$$(((A_1 \wedge A_2) \wedge A_3) \wedge \dots) \wedge A_n$$

Disjunction:

$$A_1 \vee A_2 \vee \dots \vee A_n$$

Also association to the left

Defn.

literal

atom P or negated atom $\neg P$

clause

disjunction of literals

$(\vee \vee) \wedge$

(no var. occurs twice)

Conjunctive Normal Form (CNF): A formula is a CNF if it is a conjunction of one or more clauses.

Eg.

Q
 $\neg R$
 $(P \vee \neg Q \vee R)$

$\wedge \emptyset$
 $\neg R \wedge (R \vee S) \wedge (\neg R \vee \neg S)$

Disjunctive Normal Form (DNF): dual notion to CNF

$\wedge$ -clause $\rightarrow$ conjunction of literals
(no repeat var.)

DNF $\rightarrow$ disjunction
of $\wedge$ -clauses

(*) CNF or DNF is not unique.

$$\underbrace{(P \vee Q)}_{\substack{Q=F \\ P \wedge 1}} \wedge \underbrace{(P \vee \neg Q)}_{\substack{Q=T \\ 1 \wedge (P)}} \equiv P$$

$$Q=F \quad P \wedge 1$$

$$Q=T \quad 1 \wedge (P)$$

Resolution

⇒ Proof system for establishing unsatisfiability of CNF formulas.

⇒ for any prop. form. A validity of A
unsat of A

SAT
Theorem

✓ $\Phi \models A$

poly-time reducible
to unsat of CNF

⇒ $\bar{l}$ is complement of l

⇒ If 2 clause have same literals, they are same, order doesn't matter

$$C_1 \equiv C_2$$

$$x_1 \vee x_2 \quad x_2 \vee x_1$$

$$\frac{A \vee \underline{l} \quad B \vee \underline{\bar{l}}}{A \vee B}$$

⇒ Resolution Rule:

$$\underline{C_1} = (\underline{A \vee \underline{l}})$$

$$\underline{C_2} = (\underline{B \vee \underline{\bar{l}}})$$

where A, B do not contain l or l̄.
Then

$$\underline{C_3} = (\underline{A \vee B}) \quad \boxed{\text{Resolvent}}$$

Eg. Resolvent

$$(\underline{P \vee Q})$$

$$(\underline{\bar{Q} \vee P})$$

P

P

Q̄

⊥

$$(\underline{P \vee Q})$$

$$(\underline{\bar{P} \vee \bar{Q}})$$

P ∨ P̄

$$(\underline{P \vee \bar{Q} \vee R})$$

$$(\underline{Q \vee S})$$

P ∨ R ∨ S

⇒ Resolvent Soundness Principle: If C_3 is
resolvent of C_1 & C_2

$$C_1, C_2 \models C_3$$

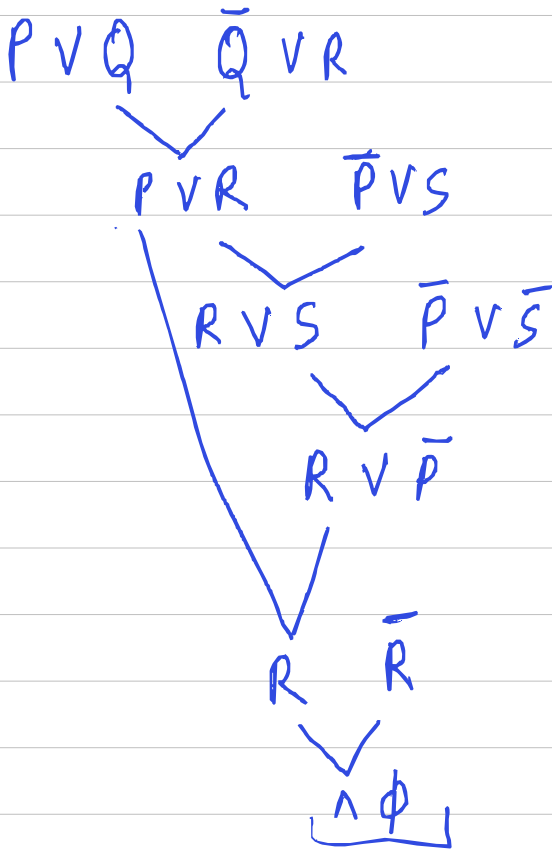
If $C_3 = \perp \Rightarrow C_1 \wedge C_2$ is UNSAT

UNSAT
↓

RES: resolution refutation of S (set of clauses)

is a seq. $C_1, C_2, \dots, C_q$ of clauses
 s.t. $C_q = \bigwedge \emptyset$ & each C_i either in S
 or is resolvent of
earlier clauses.

Eg. $S = \{(P \vee Q), (\bar{Q} \vee R), (\bar{P} \vee S), (\bar{P} \vee \bar{S}), R\}$



RES soundness: If a set S of clauses has a resolution refutation, then S is UNSAT.

Proof: $C_1, \dots, C_q$ has res ref.

$$\underbrace{C_1, C_2} \models \underbrace{C_3, C_4}_{\text{4}} \models \underbrace{C_5}_{\text{5}} \models \underbrace{\perp}$$

soundness prin
transitivity

apply inductively on
proof length

RES Completeness: Every UNSAT set of clauses has a resolution refutation.

Proof: Give a procedure that gives S

→ If S is UNSAT, a resolution refutation of S .

→ If S is SAT, a satisfying assignment of S .

⇒ It maintains

→ a set $S' \supseteq S$ of clauses, which is a partial refutation of S .

→ a stack of literals $l_1, \dots, l_k$
represent a partial truth assignment (τ)
to atoms of S

τ makes all $l_1, \dots, l_k$ true

no clause in S' is falsified by τ .

1) If S has $V \emptyset$, the refutation = $V \emptyset$, done.
 else $S' = S$ & stack of literals empty

2) If τ satisfies every clause in S ,
 output τ & halt.
 else

- select C in S not sat. by τ
- Pick l in C not falsified by τ
- Push l to stack & $\tau' = \tau \cup \{l = T\}$
- If τ' doesn't falsify any clause in S' ,
 goto 2. with $\tau \leftarrow \tau'$.

3) O/w τ' falsifies C' in S' .

- Replace l with $\bar{l}$ on stack
- $\tau'' = \tau \cup \{l = T\}$
- If τ'' doesn't falsify any clause in S' ,
 goto 2, with $\tau \leftarrow \tau''$

4) O/w τ'' falsifies C'' in S'

C' $\tau' \quad l = T$

$(\bar{l} \vee \underline{A})$
 False with τ

C'' $\tau'' \quad \bar{l} = T$

$(\underline{l} \vee \underline{B})$
 False with τ

resolve

R

$(A \vee B)$
 False with τ

If $R = \emptyset$

Output $S' \cup \{R\}$
 & halt

else

- pop stack until R is not falsified
- goto 2 with $S' \leftarrow S' \cup \{R\}$.

Halts?

⇒ Each step either adds a literal to stack

or

adds a clause to S' .

⇒ Finitely many literals

⇓

at some point new clause has to be added

⇓

Finitely many clauses

⇓

at some point $V \cap \emptyset$ has to be added ⇒ halts.

⇒ Note: $\exists$ formula with min. res. refutations are of exponential size.