

Given a formula F , can we determine whether it is satisfiable

Let F is over X variables, where $X = \{x_1, \dots, x_n\}$
i.e. $|Variables(F)| = n$

checkSAT(F)

for c in 2^n do

if $F(c) = 1$, then

return SAT, c

return UNSAT

checkSAT(F)

```
for  $\mathcal{C}$  in  $2^n$  do
    if  $F(\mathcal{C}) = 1$ , then
        return SAT,  $\mathcal{C}$ 
return UNSAT
```

find an satisfying
assignment of F,
can take
exponential
time

Can we do better?

↳ we don't know!

Resolution

$$(p \vee \alpha) \wedge (\neg p \vee \beta) \iff (\alpha \vee \beta)$$

$$\left. \begin{array}{l} F = (p \vee \alpha) \wedge (\neg p \vee \beta) \\ G = (\alpha \vee \beta) \end{array} \right\} F \text{ \& } G \text{ are} \\ \text{equisatisfiable formulas}$$

Resolution Refutation

list of clauses C_1, C_2, \dots, C_t is a resolution refutation of formula F_{CNF} if:

i) C_t is empty \square

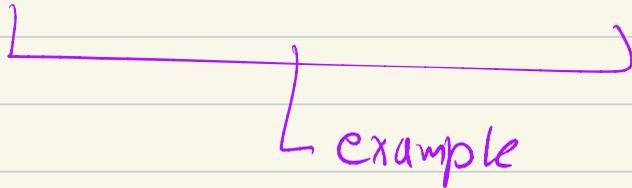
ii) $C_k \in F_{CNF}$ or C_k is derived using resolution from C_i, C_j , where $i, j < k$.

Resolution Refutation

list of clauses C_1, C_2, \dots, C_t is a resolution refutation of formula F_{CNF} if:

i) C_t is empty \square

ii) $C_k \in F_{CNF}$ or C_k is derived using resolution from C_i, C_j , where $i, j < k$.



Models $(F) = 0$
 F is UNSAT

$$C_i = P \vee \alpha$$

$$C_j = \neg P \vee B$$

$$C_k = \alpha \vee B$$

C_k is derived from C_i & C_j

Example:

$$F = \underbrace{(\neg P \vee \neg q \vee r)}_{C_1} \wedge \underbrace{(\neg P \vee q)}_{C_2} \wedge \underbrace{P}_{C_3} \wedge \underbrace{\neg r}_{C_4}$$

$$\text{Resolution on } C_1 \text{ \& } C_3 : \frac{(\neg P \vee \neg q \vee r) \wedge (P)}{C_5 : \neg q \vee r}$$

$$\text{Resolution on } C_2 \text{ \& } C_3 : \frac{(\neg P \vee q) \wedge P}{C_6 : q}$$

$$\text{Resolution on } C_5 \text{ \& } C_4 : \frac{(\neg q \vee r) \wedge (\neg r)}{C_7 : \neg q}$$

$$\text{Resolution on } C_6 \text{ \& } C_7 : \frac{q \wedge \neg q}{C_8 : \square} \quad \left. \vphantom{\frac{q \wedge \neg q}{C_8 : \square}} \right\} \text{Refutation}$$

list of clauses C_1, C_2, \dots, C_8 is a resolution refutation of F

Thm: A formula F_{CNF} is refutable if and only if F_{CNF} is unsatisfiable.

→ direction is easy to see: if a formula F_{CNF} is refutable then F_{CNF} is unsatisfiable.

← direction: if F_{CNF} is unsatisfiable then F_{CNF} is refutable
↳ Homework: Induction on # of propositional variables

① Use resolution Refutation to prove the validity of a formula:

To prove: F is valid

1. convert F to $(\neg F)$
2. convert $(\neg F)$ to CNF, say F'_{CNF}
3. Find a resolution refutation of F'_{CNF}
4. if F'_{CNF} is refutable, then F is valid

Bottleneck of Resolution:

→ Space required to perform resolution

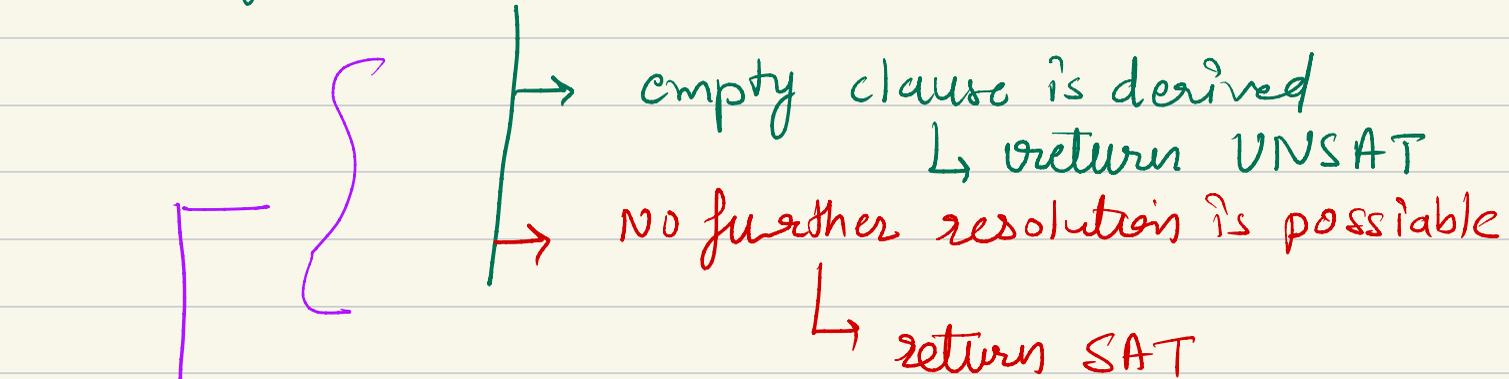
- ↳ at every resolution step: nC_2 $n = |\text{clauses}|$
- ↳ new clauses is added to the set of clauses.
- ↳ this is done linear # of times, hence overgrowth can be exponential.

Resolution is EXP SPACE

Davis - Putnam (DP '1960) Algorithm

Martin Davis & Hil

1. Start with FCNF
2. perform Resolution until



One of those two case will occur becoz resolution refutation is sound & complete

Davis - Putnam (DP '1960) Algorithm

Pick a literal l that occurs with both polarities in F .

for every clause c in F containing l and every clause c' in F containing its negation $\neg l$ do

Resolve c & c'

$\gamma \leftarrow (c \setminus \{l\}) \cup (c' \setminus \{\neg l\})$

$F \leftarrow \text{add_to_formula}(\gamma, F)$

for every clauses c that contain l or $\neg l$ do

$F \leftarrow \text{remove_from_formula}(c, F)$

Davis-Putnam (DP '1960) Algorithm

if F has empty clause then
return UNSAT

Pick a literal l that occurs with both polarities
in F .

for every clause c in F containing l and every
clause c' in F containing its negation $\neg l$ do

Resolve c & c'

$r \leftarrow (c \setminus \{l\}) \cup (c' \setminus \{\neg l\})$

$F \leftarrow \text{add_to_formula}(r, F)$

for every clauses c that contain l or $\neg l$ do

$F \leftarrow \text{remove_from_formula}(c, F)$.

Davis-Putnam (DP '1960) Algorithm

if F has empty clause then
return UNSAT

if $\nexists l$ that occur with both polarities in F ,
return SAT

Pick a literal l that occurs with both polarities
in F .

for every clause c in F containing l and every
clause c' in F containing its negation $\neg l$ do

Resolve c & c'

$\gamma \leftarrow (c \setminus \{l\}) \cup (c' \setminus \{\neg l\})$

$F \leftarrow \text{add_to_formula}(\gamma, F)$

for every clauses c that contain l or $\neg l$ do

$F \leftarrow \text{remove_from_formula}(c, F)$.

Davis-Putnam (DP '1960) Algorithm

if F has empty clause then
return UNSAT

if $\exists l$ that occur with both polarities in F ,
return SAT

} is this
correct?
↳ does it
terminate

Pick a literal l that occurs with both polarities
in F .

for every clause c in F containing l and every
clause c' in F containing its negation $\neg l$ do

Resolve c & c'

$\gamma \leftarrow (c \setminus \{l\}) \cup (c' \setminus \{\neg l\})$

$F \leftarrow \text{add-to-formula}(\gamma, F)$

for every clauses c that contain l or $\neg l$ do

$F \leftarrow \text{remove-from-formula}(c, F)$.

when
 $F = P \vee \neg P$

Davis-Putnam (DP'1960) Algorithm

DPCF1

}

For every clause c in F that contains both l and $\neg l$ do
 $F \leftarrow \text{remove-from-formula}(c, F)$

stopping conditions

if F is empty then

return SAT

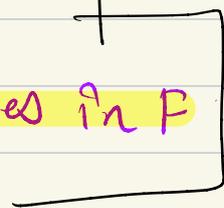
if F has empty clause then

return UNSAT

if $\exists l$ that occur with both polarities in F

return SAT

Can we make it better?



Pick a literal l that occurs with both polarities in F .

for every clause c in F containing l and every clause c' in F containing its negation $\neg l$ do

Resolve c & c'

$r \leftarrow (c \setminus \{l\}) \cup (c' \setminus \{\neg l\})$

$F \leftarrow \text{add-to-formula}(r, F)$

for every clauses c that contain l or $\neg l$ do

$F \leftarrow \text{remove-from-formula}(c, F)$.

DPCF2

}

Pure literal elimination :-

Pure literal :- a literal l all of which occurrences in F have the same polarity

Example:

$$F = (p \vee q \vee r) \wedge (\neg q \vee r) \wedge (p \vee \neg r) \wedge (p \vee \neg q)$$

→ literal p has positive polarity in all occurrence in F
→ p is pure literal

$$F = (p \vee \neg q \vee r) \wedge (\neg q \vee r) \wedge (\neg p \vee \neg r) \wedge (p \vee \neg q)$$

→ literal q has negative polarity in all occurrence in F
→ q is pure literal

Pure literal elimination :-

Pure literal :- a literal l all of which occurrences in F have the same polarity

for every clause that contains a pure literal:
 $F \leftarrow \text{remove-from-formula}(F)$

→ pure literal can be set as per polarity to satisfy the clause.

Davis-Putnam (DP '1960) Algorithm

DP(F)

{

For every clause c in F that contains both l and $\neg l$ do
 $F \leftarrow \text{remove-from-formula}(c, F)$

while there is a pure literal l do
for every clause c that contains l do
 $F \leftarrow \text{remove-from-formula}(c, F)$

Stopping conditions

if F is empty then
return SAT

if F has empty clause then
return UNSAT

Pick a literal l that occurs with both polarities in F .

for every clause c in F containing l and every clause c' in F containing its negation $\neg l$ do
Resolve c & c'
 $\gamma \leftarrow (c \setminus \{l\}) \cup (c' \setminus \{\neg l\})$
 $F \leftarrow \text{add-to-formula}(\gamma, F)$

for every clauses c that contain l or $\neg l$ do
 $F \leftarrow \text{remove-from-formula}(c, F)$.

DP(F)

}