Resolution

Normal forms

Negation Normal Form (NNF)

- Every negation is used only on propositions
- E.g., $\neg p \vee \neg q$
- $\neg(\alpha_1 \wedge \alpha_2) \Rightarrow \neg\alpha_1 \vee \neg\alpha_2$
- Atomic p or $\neg p$ is called a literal

Conjunctive Normal Form (CNF)

- $\bigwedge_{i=1}^{m}(\bigvee_{j=1}^{n} l_{i,j})$
- E.g., $(p_1 \vee p_2 \vee \neg p_3) \wedge (\neg p_1 \vee p_2 \vee p_3)$
- $\alpha_1 \vee (\alpha_2 \wedge \alpha_3) \Rightarrow (\alpha_1 \vee \alpha_2) \wedge (\alpha_1 \vee \alpha_3)$
- Every $\bigvee_{j=1}^{n} l_{i,j}$ is called a clause/conjunct

Disjunctive Normal Form (DNF)

- $\bigvee_{i=1}^{m}(\bigwedge_{j=1}^{n} l_{i,j})$
- The SAT problem becomes trivial!

Why normal form?

CNF-SAT: Given a propositional formula α in CNF, check if α is satisfiable

Theorem: there is no polynomial blow-up translation from wff to CNF/DNF.

Theorem: SAT $\leq_P$CNF-SAT

- Proof: Coming soon
- Is α satisfiable (Is $\neg\alpha$ a tautology?)

  ➔ Is the equisatisfiable $f(\alpha) = D_1 \wedge D_2 \wedge \cdots \wedge D_k$ satisfiable?
- This is a preprocessing step for the resolution algorithm

Resolution algorithm

Resolution: $$\frac{D \vee p \qquad D' \vee \neg p}{D \vee D'}$$

Apply resolution:

- If $D \vee p$ and $D' \vee \neg p$ are clauses, add $D \vee D'$ as a new clause
- Repeat until no more resolution can be done
- Resolution is *closed* if the empty clause is contained
- Return Unsatisfiable iff. Closed

Example

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

$\Gamma = \{\{p, q\}, \{\neg p, r\}, \{\neg q, r\}, \{\neg r\}\}$

$\{p, q\}$	(1)
$\{\neg p, r\}$	(2)
$\{\neg q, r\}$	(3)
$\{\neg r\}$	(4)
$\{\neg p\}$	(5) (resolvent of 2 and 4)
$\{q\}$	(6) (resolvent of 1 and 5)
$\{r\}$	(7) (resolvent of 3 and 6)
$\{\}$	(8) (resolvent of 4 and 7)

Madhusudan and Viswanathan. Logic in Computer Science. 2021.

Soundness

Theorem: the resolution algorithm is sound.

- If the resolution is closed, Γ is unsat.
- Easy to prove.

Completeness

Theorem: the resolution algorithm is complete.

- If Γ is unsat, show the resolution will be closed.
- Proof by induction on the number of propositions: if Γ involves $n+1$ propositions, resolution of Γ will produce Γ' involving n propositions, which is already unsat; hence the resolution will be closed.
- Pick a proposition p and have $\Gamma = \Gamma_1 \wedge \Gamma_2 \wedge \Gamma_3$
- Γ_1 are conjuncts containing p; Γ_2 are conjuncts containing $\neg p$; Γ_3 are conjuncts containing neither.
- $\Gamma_1 = \bigwedge_{i=1}^{m}(D_i \vee p) \qquad \Gamma_2 = \bigwedge_{j=1}^{n}\left(E_j \vee \neg p\right)$
- $\Gamma_1 \times \Gamma_2 = \bigwedge_{i,j=1,1}^{m,n}\left(D_i \vee E_j\right)$
- Claim: $\Gamma' = \Gamma_1 \times \Gamma_2 \wedge \Gamma_3$ is already unsat!
- If Γ' is satisfiable, let $v \vDash \Gamma'$, then $v[T/p]$ or $v[F/p]$ will satisfy Γ
  - If $v[T/p]$ does not, all D_i are satisfiable. Then $v' = v[F/p]$ and $v' \vDash \Gamma$
  - Similarly if $v[F/p]$ does not

Beyond Resolution: DPLL algorithm

Backtracking based search

- Assign a value to a proposition to simplify the CNF
- Stop if all propositions are assigned
- Backtrack if unsatisfiable
- Propositions are chosen heuristically

Most efficient SAT solving algorithm since 1960s

- Implementations: zChaff, Minisat, etc.