COL765 - Lec 19

Last Time

- Propositional Logic
  - Syntax [as a BNF]
    - Associativity, Precedence, AST
    - Syntactic identity
  - Semantics
    - truth value, models, satisfiability
    - validity
    - satisfaction relation ($\models$)
    - Equisatisfiability and equivalence

Today

- Identities,
- Literals, Normal forms
- Tautology Checking
- Propositional resolution, Complexity results

Identities

Negation $\neg\neg\phi \iff \phi$

Condition $\phi \rightarrow \psi \iff \neg\phi \vee \psi$

Bicondition $\phi \leftrightarrow \psi \iff (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$

Identity : $\phi \wedge \top \iff \phi \iff \phi \vee \bot$

Idempotence : $\phi \wedge \phi \iff \phi \vee \phi \iff \phi$

Commutativity : $\phi \wedge \psi \iff \psi \wedge \phi$

Associativity : $(\phi \wedge \psi) \wedge \chi \iff \phi \wedge (\psi \wedge \chi)$

DeMorgan : $\neg(\phi \wedge \psi) \iff \neg\phi \vee \neg\psi$

Inversion : $\neg\bot \iff \top$ (and vice-versa)

Absorption : $\phi \wedge (\phi \vee \psi) \iff \phi \vee (\phi \wedge \psi)$

$\iff \phi$

Literals

Adequacy of $\{\neg, \wedge, \vee\}$ is used to build Normal forms

- A literal is an atom $p \in A$ or its negation $\neg p$ for $p \in A$
- Atoms are called positive literals or negative literals

NNF : Built with literals & $\wedge$ and $\vee$ only.

$$\mu, \nu \; := \; \lambda \in \underbrace{\mathcal{L}_0} \;|\; \mu \wedge \nu \;|\; (\mu \vee \nu)$$

Set of literals

Lemma: Every possible in P is logically equivalent to one in NNF.

<u>CNF</u>

$$\bigwedge_{i=1}^{n} \delta_i$$

where

$$\delta \equiv \bigvee_{j=1}^{m} \lambda_j$$

<u>DNF</u> : analogously define a conjunction of literals & a DNF

<u>Lemma</u>: Every formula in P_ϕ has a logically Equivalent CNF & DNF

Tautology Checking

- Using truth tables can be quite tedious

- We have to show that $\bigwedge \text{hypothesis}_i \Rightarrow \text{Concl}$ is a tautology or can be falsified

For Tautology

if (Hypothesis == NULL) then tautology (Concl)

Else tautology (COND (Hypotesis, Concl.))

for falsification :

```
if (Hypothesis == NULL)
    then falsify (CNF (Concl.))

Else
    falsify ( CNF (COND (Hypothesis,
                         Conclusion)))
```

Checking Tautology

```
tautology (P) =
    let  Q = CNF (P)
         λ = falsify (Q)
         if (λ == NULL) then true
                   Else false
```

Computing CNF

- Rewrite cond^n & bicond^n using equivalences
- Convert to NNF by pushing negation inward
- Convert to CNF by distributing $\vee$ over $\wedge$

Falsifying a CNF

Assume CNF $Q \equiv \bigwedge_{i=1}^{m} D_i$

where $D_i \equiv \bigvee_{j=1}^{n_i} \ell_{ij}$

Observation

D_i can be falsified

iff $\underbrace{P_i}_{\text{Positive literal set}} \cap \underbrace{N_i}_{\text{Negative literal set}} = \emptyset$

Eg: $(\underbrace{a \vee b}_{})$

$a=0$ AND $b=0$ can falsify the above formula

$(a \vee \neg a \vee b)$

$\Leftrightarrow$ can never be falsified!

Which means

- Validity checking of CNF formula is in P-Time.

- Alternatively Satisfiability test for DNF is in P

Exercise :

Prove that $\phi \in P$, then ϕ is SAT iff $\neg\phi$ is NOT valid!

Propositional Resolution

To show $\Gamma \models \varphi$

or equivalently $(\bigwedge \Gamma \rightarrow \varphi)$ is valid

$a \rightarrow b$
$\equiv \neg a \vee b$

$\neg (a \wedge \neg b)$

or equivalently

$(\bigwedge \Gamma \wedge \neg \varphi)$ is a contradiction

- Represent $\bigwedge \Gamma \wedge \neg \varphi$ as CNF
- The UNSAT of the CNF can be shown by deriving an empty clause $\{\}$

Resolution Method

Let Δ be the set of clauses

- Δ be a clean set

  i.e $\Delta_1 \cap \overline{\Delta}_1 = \phi$

  where $\Delta_1 = \{ C \in \Delta \mid p \in C \}$

  $\overline{\Delta}_1 = \{ \overline{C} \in \Delta \mid \neg p \in \overline{C} \}$

- Δ_1 & $\overline{\Delta}_1$ may not be disjoint

- New set of clauses obtained after resolution is

$$\text{resolve}(\Delta, p) \triangleq (\Delta - (\Delta_1 \cup \bar{\Delta}_1)) \cup \{D \mid D = (C - \{p\}) \cup (\bar{C} - \{\neg p\}), C \in \Delta_1, \bar{C} \in \bar{\Delta}_1\}$$

Algo.

while $\{\} \not\in \Delta \wedge \exists$ Complimentary pair $(P, \neg P) \in \Delta$

do

$$\Delta' = \text{resolve}(\Delta, P)$$

$$\Delta := \text{Cleanup}(\Delta')$$

- If $c \subseteq c'$ in Δ then c' can be deleted from Δ
- If c has $p, \neg p$ then c can be deleted from Δ.

Eg: $p \wedge q \models p$ which is equivalent to $(p \wedge q) \wedge \neg p$

$$\Delta = \{ \underbrace{\{p\}}, \{q\}, \underbrace{\{\neg p\}} \}$$

Resolve

$$\Delta = \{ \{\}, \{q\} \}$$

$$\equiv \perp \text{. QED.}$$

Converting into CNF

- Rewrite conditionals & biconditional
- Convert to NNF by pushing the negation inward
- Convert to CNF by distributing OR over AND