2017-09-27

- Lemma: Let $\phi_1, \phi_2, \ldots, \phi_n, \psi$ be propositional logic formulae. $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$ iff $\vDash \phi_1 \implies (\phi_2 \implies \ldots (\phi_n \implies \psi))$.
- **Validity**: Let ϕ be a propositional logic formula. ϕ is **valid** if $\vDash \phi$.
- Lemma: A clause $D = L_1 \vee L_2 \vee \ldots \vee L_m$ is **valid** iff there is a propositional atom p such that L_i is p and L_j is $\neg p$ for some $1 \leq i, j \leq m$.
- **Satisfiability**: Let ϕ be a propositional logic formula. ϕ is **satisfiable** if it evaluates to T under some valuation.
- Proposition: Let ϕ be a propositional logic formula. ϕ is satisfiable iff $\neg\phi$ is not valid.
- **Conjunctive normal form (CNF)**: a formula C is in *conjunctive normal form (CNF)* has the following form:
  - **Literal**: $L ::== p | \neg p$
  - **Clause**: $D ::== L | L \vee D$
  - **Formula**: $C ::== D | D \wedge C$
- From truth table to CNF:
  - For each line l where ϕ evaluates to F, construct a clause ψ_l as $\psi_l = L_{l,1} \vee L_{l,2} \vee \ldots \vee L_{l,n}$, where $L_{l,i} = \neg p_i$ if p_j is T; otherwise, $L_{l,i} = p_i$.
  - Then $\phi \equiv \psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_m$ where ψ are constructed for every line evaluating ϕ to F.
- Given a propositional logic formula in CNF, we can check the validity of the formula in *linear* time.
- Any propositional logic formula can be transformed to CNF by three steps:
  - Remove every implications via $\phi \implies \psi \equiv \neg\phi \vee \psi$.
  - Push every negation to literals via De Morgan's laws.
  - Apply law of distribution.
- Algorithm for checking the satisfiability of a propositional logic formula:
  - Compute a CNF formula ψ such that $\psi = \neg\phi$.
  - Check the validity of ψ.
  - Return ϕ is satisfiable if ψ is not valid; otherwise, ϕ is unsatisfiable.
- Given a propositional logic formula in CNF, it is easy to check its validity, but hard to check its satisfiability.
- **Horn formula**: a propositional logic formula H of the following form:
  - $P ::== \bot | \top | p$
  - $A ::== P | P \wedge A$
  - **Horn clause**: $C ::== A \implies P$

- $H ::== C | C \wedge H$

- Algorithm for checking the satisfiability of a Horn formula ϕ:
  - Mark $\top$ if it occurs in ϕ.
  - If there is a Horn clause $P_1 \wedge P_2 \wedge \ldots \wedge P_n \implies Q$ in ϕ s.t. all P_i for $1 \leq i \leq n$ are marked, then mark Q.
  - Return ϕ is unsatisfiable if $\perp$ is marked; otherwise, ϕ is satisfiable.
- **Equisatisfiability**: ϕ and ψ are **equisatisfiable** if ϕ is satisfiable iff ψ is satisfiable.
- **Tseitin transformation**: For every propositional logic formula ϕ, there is a propositional logic formula ψ in CNF s.t. ϕ and ψ are *equisatisfiable*.