- Propositional Logic
  - proposition: statement TVF
  - predicate: proposition whose truth depends on vars.
    - func: \{vars\} \rightarrow \{T,F\}

- Satisfiability
  - P = set of yes/no pb that can be solved in polynomial time
    (poly time = depends on input length)
  - NP = set of yes/no pb that can be verified in polynomial time

- Theorem: P = NP \implies SAT \in P

- Predicate logic formulas:
  - occurrence of a var is quantified if it is in a subformula
    quantified (Q \in \mathcal{D}, E(x)). Otherwise it is free (unquantified)
  - Interpretation of a formula:
    - non-empty sets for each domain
    - element of relevant domain for each csx symbol
    - func: \text{for each csx symbol of a func from relevant domain to \{T,F\} for each predicate symbol}
    - if it has free vars they have to be mapped to an element in the relevant domain (called evaluation)

- A PLF is valid/tautology if:
  - true for any interpretation
  - unsatisfiable if false under all interpretations
  - satisfiable if true under some interpretation

- Proofs
  - See 16.5

- Note: by cases: \quad P(x) \implies P_1(x) \lor P_2(x),... 
  \quad P_3(x) \implies Q(x) \land P_4(x) \implies Q(x),...
  \quad (P_1(x) \lor P_2(x),...) \implies Q(x) \quad \#\text{modus ponens}
--- Complete set of connectives

- \{V, \land, \neg\} is complete  
  # because every formula can be
  rewritten in CNF/DNF.

- \{\land, \neg\} (or \{V, \neg\}) is complete  
  # De Morgan.

- \{NAND\} is complete  
  # PAQ => (PNANDQ)NAND(PNANDP)
  
- \{if-then-else\}  
  # PAQ => i-t-e (P,A,Q)
  
- PVQ => i-t-e (P,V,Q)
  
- \neg P => i-t-e (P,F,T)

A set of connectives is complete iff any prop. formula can be
rewritten using its elements.

--- Substitution

Theorem: Let R be a formula, let S be a subformula of R,
let S' be a formula logically \( \equiv \) to S, let R' be the
formula that results by replacing S by S' in R.
Then R' is logically \( \equiv \) to R.

- \( R = (\neg (A \land \neg B)) \lor (B \Rightarrow (\neg A \land \neg B)) \)
  \( S = \neg A \land \neg B \)
  \( S' = \neg (A \lor B) \)

- \( R \equiv R' \)

Theorem: Suppose P is a prop. variable and R is a prop.
formula that contains some occurrences of P.
Let R' be the formula obtained from R by
replacing every occurrence of P by \( \neg P \) (where \( \neg P \) is \( \text{true} \)).
Then R' is a tautology.

- \( R = (A \lor P) \equiv (P \lor A) \) is a tautology

Let \( Q = (V \lor P) \), then \( R' = (A \lor C \lor V \lor D) \equiv (B \lor V \lor B \lor V) \) is a taut.

0.3999... = 0.4000...

Proof: 0.3999... = \( x \)

\[ \frac{3.9999...}{10} = 0.3999... \]

\[ 3.6 = 3x \]

\[ x = 0.4 \]

Reduce

The above is no prop. A that takes
a prop. P and determines that
P is syntactically connect and
holds on all inputs.

Proof: \( H'(P,x) = (P \lor P) \) is not true,
so H' solves halting P5.