Lecture-11 Main Points

- Boolean expression
  - constructed from variables and connectives of BA
    ($\vee, \wedge, \neg$ and derived connective $\rightarrow$)
  - Example: $x \vee y \rightarrow \neg z$
  - Same as propositions, with atomic propositions being called variables.
  - can be interpreted in any BA.
- Normal forms
- NNF
  - Negation ($\neg$) appears only in front of a variable.
  - Constructed from literals using $\vee, \wedge$ only (no $\rightarrow$).
  - Every Boolean expression is equal to a Boolean expression in NNF.
  - Proof yields an algorithm to convert to an equivalent NNF.
    * Eliminate all occurrences of $\rightarrow$.
    * Push negation inward using DeMorgan's laws.
  - Can be implemented in linear time.
- CNF
  - Conjunction of clauses. Each clause is a disjunction of literals.
  - Example: $(x \vee \neg y) \wedge (z \vee w \vee \neg x)$
  - Every Boolean expression is equal to a Boolean expression in CNF.
  - Proof yields an algorithm to convert to an equivalent CNF.

    * Convert into NNF
    * Use recursion, with distributive law when topmost connective is $\vee$.
  - Blows size of the formula exponentially.

- DNF
  - Disjunction of terms. Each disjunct is a conjunction of literals.
  - Example: $(x \wedge \neg y) \vee (z \wedge w \wedge \neg x)$
  - Every Boolean expression is equal to an equivalent Boolean expression in DNF.
  - Proof and algorithm are similar to CNF case.

- NF algorithms yield formulae which are equivalent to the original formulae over *all* BA. (This is because we use laws of BAs only in our conversion algorithms).