• Propositional Logic
  – Know the operator precedence rules and the order of association.
  – Given a formula, know how to construct its truth table.
    Example: Assignment 1, question 3, part b).
  – Given a formula, know how to create its formation diagram.
    Example: Assignment 1, question 3, part c)
  – Given a formation diagram, know how to give its inorder and preorder traversals. Why is one more useful than the other?
    Example: Assignment 1, question 3, part d).
  – What is logical consequence ( |= ) and logical equivalence (≡)? How do they relate to implication (→) and equivalence (↔)?
    Example: Assignment 1, question 4, part a) and b).
  – Given a formula, construct its semantic tableau. Know the α and β rules for the operators ∧, ∨, →, ↔ and their variations with ¬ (see Page 32 of the textbook).
    Example: Assignment 1, question 5, part a).
  – Know the concepts of satisfiability, unsatisfiability, falsifiability, and unfalsifiability.
    Example: Assignment 1, question 5, part c).
  – Be prepared to solve decision problems using other decision problems. In particular, think about the decision problems NOT-SAT(A), SAT(A), CONSEQ(A,B), EQUIV(A,B).
    Example: Assignment 1, question 6.

• Binary decision diagrams
  – When we say BDD we implicitly mean ordered BDD. Make sure you know what ordered means.
    Example: Given a BDD, identify what the ordering of the propositions is. When the BDD is reduced, is the ordering unique?
  – Given a BDD for formula A that contains proposition p, know how to give the corresponding BDD for A|p=T or A|p=F where A|p=v is the result of restricting the proposition p to value v.
  – What is Shannon’s expansion and how can you use it to apply a binary operation to two formulas (see Page 91 of the textbook)? How did we use Shannon’s expansion on BDDs?
    Example: If A1 = p ∧ q ∨ r and A2 = p ∨ ¬q then what is Shannon’s expansion for A1 ∧ A2 using p?
  – Reduce a BDD by hand.
    Example: Assignment 2, question 2, part c).
  – Have an understanding of how to count the number of satisfying assignments in a BDD (numSAT aka countSAT) or find the maximum number of True assignments in a satisfying assignment (maxTSAT) or how to find a random satisfying assignment (randSAT). You won’t be asked to implement these routines in Prolog, but you may be asked to do an example by hand for a certain BDD.
    Example: Assignment 2, question 2, part h)

• Logic Programming (Prolog)
  – Be able to write short Prolog rules and predicates.
    Example: Write factorial(N,X) so that X = N!.
    Example: Write remove(List1,S,List2) so that List2 results from removing the first copy of S from List1.
  – Have a general understanding of how Prolog’s internal routines of resolution and unification work. There are links on the course webpage for these two algorithms.
    Example: Does hello(X,a,joe(R,c) unify with hello(z,Y,Z)? If so, give variable bindings showing the unification.
    Example: What does tail-recursive mean and why is it desirable in Prolog?