After doing this tutorial you should be able to:

1. use the recursive definition to tell if something is a Boolean formula or not,
2. evaluate a given Boolean formula for a given assignment,
3. tell if a given Boolean formula is satisfiable/valid,
4. model facts and reasoning in Boolean logic.

Problem 1. Are the following expressions propositional logic formulas (according to the definition of the syntax, the conventions, as well as the extended definition of the syntax, given in the lecture)?

1. $(p \wedge q)$
2. (p)
3. $p \wedge q$
4. $\neg(p \wedge q)$
5. $(p \bigoplus q)$
6. $(p \rightarrow q)$
7. $(p \rightarrow (q \vee r))$
8. $\neg\neg\neg\neg\neg p$

Problem 2. A formula G is **valid** if every assignment satisfies G. Valid formulas capture "logical truths". Give some examples of valid formulas.

Problem 3. This is Exercise 7.10 in Russel and Norvig's *Artificial Intelligence* (2010).

Decide whether each of the following sentences is valid, unsatisfiable, or neither. Verify your answer using truth-tables (or any method you like).

1. $Smoke \rightarrow Smoke$
2. $Smoke \rightarrow Fire$
3. $(Smoke \rightarrow Fire) \rightarrow (\neg Smoke \rightarrow \neg Fire)$
4. $Smoke \vee Fire \vee \neg Fire$
5. $((Smoke \wedge Heat) \rightarrow Fire) \leftrightarrow ((Smoke \rightarrow Fire) \vee (Heat \rightarrow Fire))$
6. $(Smoke \rightarrow Fire) \rightarrow ((Smoke \wedge Heat) \rightarrow Fire)$
7. $Big \vee Dumb \vee (Big \rightarrow Dumb)$

Problem 4. Suppose you had code that solved the satisfiability problem. Show how you could call that code to solve the validity problem, i.e., to test if a given formula is valid or not.

Problem 5.[Exam 2022] Is it true that if the formula $F \vee G$ is valid then either F is valid or G is valid? Give a short explanation/justification of your answer.

Problem 6.[Assignment problem in 2021] Each of the following statements about propositional logic is false. For each statement, show why the statement is false with a counterexample using only the variables p, q (you don't need to use both variables).

1. If F is valid and G is satisfiable then $(F \rightarrow G)$ is valid.
2. If F is satisfiable and G is satisfiable then $(F \wedge G)$ is satisfiable.
3. If F is valid then $\neg F$ is satisfiable.
4. If F is satisfiable then $\neg F$ is satisfiable.

Problem 7. Write a recursive function **SubForms**(G) that returns the set of subformulas of G (i.e. in a similar style to the way that the truth value function tv is defined in the lecture). Do this for the basic syntax, and for the extended syntax.

Problem 8. Intuitively, the size of a propositional formula F is the number of symbols in F, ignoring parentheses. So, e.g., $|(p \vee \neg p)| = 4$.

1. Give a recursive procedure defining the size of F, denoted $|F|$.
2. Show by induction that the number of subformulas of F, denoted $|\textbf{SubForms}(F)|$, is at most $|F|$. Hint: use induction on the size of F.