CSCE 222-200 Discrete Structures for Computing

Fall 2024

Instructor: Dr. Jianer Chen
Office: PETR 428
Phone: (979) 845-4259
Email: `chen@cse.tamu.edu`
Office Hours: T+R 2:00pm–3:30pm

Teaching Assistant: Evan Kostov
Office: EABC Cubicle 6
Phone: (469) 996-5494
Email: `evankostov@tamu.edu`
Office Hours: MW 4:00pm-5:00pm

Assignment #6 Solutions

1. Let X be the random variable that equals the sum of the numbers that appear when n fair dice are rolled. What is the expected value of X?

Solution. Fix an i, $1 \leq i \leq n$, let X_i be the random variable that equals the number that appears on the i-th die in the rolling. That is, we have $X_i = b$ if the number b is shown on the i-th die in the rolling, where $1 \leq b \leq 6$.

It is easy to see that for a fixed b, $1 \leq b \leq 6$, $\mathbf{Pr}[X_i = b] = 1/6$. Thus, we have $\mathbf{E}[X_i] = \sum_{b=1}^{6} b \cdot \mathbf{Pr}[X_i = b] = 21/6 = 7/2$. This holds true for all $1 \leq i \leq n$.

Now define a new random variable $X = \sum_{i=1}^{n} X_i$. Then X is equal to the sum of the numbers that appear when the n fair dice are rolled. By the Linearity of Expectation, we have

$$\mathbf{E}[X] = \mathbf{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbf{E}[X_i] = 7n/2.$$

2. Find the conjunctive normal form of the Boolean function $F(x, y, z)$ that is equal to 1 if and only if (a) $x = y = 1$, $z = 0$; (b) $x = y = z = 0$, (c) $x = z = 0$, $y = 1$, and (d) $x = 0$, $y = z = 1$.

Solution. We use the method discussed in the class. We first find the disjunctive normal form for the complement $\overline{F(x, y, z)}$ of the function $F(x, y, z)$. We have

$\overline{F(x, y, z)} = 0$ when

(a) $x = y = 1$, $z = 0$; (b) $x = y = z = 0$,
(c) $x = z = 0$, $y = 1$, (d) $x = 0$, $y = z = 1$,

and

$\overline{F(x, y, z)} = 1$ when

(e) $x = 1$, $y = 0$, $z = 0$; (f) $x = 1$, $y = 1$, $z = 1$;

(g) $x = 0,\ y = 0,\ z = 1$; (h) $x = 1,\ y = 0,\ z = 1$.

From (e)-(h), we can directly write out the disjunctive normal form for the function $\overline{F(x, y, z)}$, which is

$$\overline{F(x, y, z)} = (x\,\overline{y}\,\overline{z}) + (x\,y\,z) + (\overline{x}\,\overline{y}\,z) + (x\,\overline{y}z).$$

Using De Morgan's law, we can get the conjunctive normal form for $F(x, y, z) = \overline{\overline{F(x, y, z)}}$:

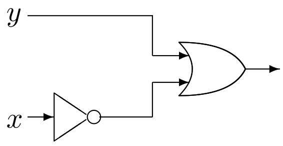
$$\begin{aligned} F(x, y, z) = \overline{\overline{F(x, y, z)}} &= \overline{(x\,\overline{y}\,\overline{z}) + (x\,y\,z) + (\overline{x}\,\overline{y}\,z) + (x\,\overline{y}z)} \\ &= (\overline{x} + y + z)(\overline{x} + \overline{y} + \overline{z})(x + y + \overline{z})(\overline{x} + y + \overline{z}). \end{aligned}$$

3. Construct circuits using inverters, AND gates, and OR gates to produce the following outputs:

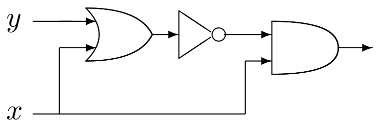
(a) $\overline{x} + y$; (b) $\overline{(x + y)}x$; (c) $xyz + \overline{x}\,\overline{y}\,\overline{z}$; (d) $\overline{(\overline{x} + z)(y + \overline{z})}$.

Solution. The circuits are given as follows.

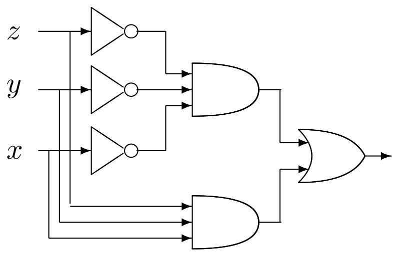
(a)



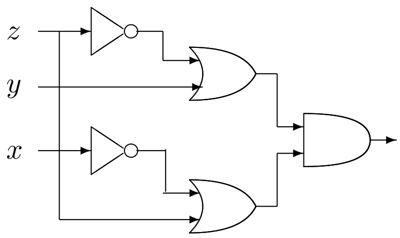
(b)



(c)



(d)



4. Determine whether each of the following Boolean functions is satisfiable.

(a) $(x_1 + x_2 + \overline{x}_3)(x_1 + \overline{x}_2 + \overline{x}_4)(x_1 + \overline{x}_3 + \overline{x}_4)(\overline{x}_1 + \overline{x}_2 + \overline{x}_4)(x_1 + x_2 + \overline{x}_4)$

(b) $(\overline{x}_1+\overline{x}_2+x_3)(\overline{x}_1+x_2+\overline{x}_4)(x_1+\overline{x}_2+\overline{x}_4)(\overline{x}_1+\overline{x}_3+\overline{x}_4)(x_1+x_2+\overline{x}_3)(x_1+\overline{x}_3+\overline{x}_4)$

(c) $(x_1 + x_2 + x_3)(x_1 + \overline{x}_2 + \overline{x}_4)(x_1 + \overline{x}_3 + x_4)(\overline{x}_1 + x_3 + x_4)(\overline{x}_1 + x_2 + \overline{x}_4)$
$(x_1 + \overline{x}_2 + \overline{x}_3)(\overline{x}_1 + \overline{x}_2 + x_4)(\overline{x}_1 + \overline{x}_3 + \overline{x}_4)$

Solution. We first remark that there is **NO** known efficient way to determine whether an arbitrarily given Boolean function is satisfiable. This is one of the biggest open problems in computer science that researchers have spent significant time and efforts trying to solve but with very little progress. Therefore, whenever we need to determine if a Boolean function is satisfiable, we, in general, try to find some special structures or properties of the given function. If we are lucky and discover those helpful structures, we are able to solve it. On the other hand, there are Boolean functions that do not seem to give us any helpful hints. In this case, we, based on our current understanding of the problem, have to enumerate all assignments to see if any of them satisfies the function.

All given functions in the question are in conjunctive normal form (CNF). To satisfy a function in CNF, we need to find an assignment that satisfies *all* clauses. To satisfy a clause, we need to satisfy at least one literal in the clause. Therefore, in the construction of a satisfying assignment for a CNF function f, we are looking for an assignment that makes each clause in f have at least one satisfied literal.

(a) This one is easy. Obeserve that x_1 appears in all but one clauses. Thus, if we assign $x_1 = 1$, then all clauses except $(\overline{x}_1 + \overline{x}_2 + \overline{x}_4)$ are satisfied. For the remaining clause, we can simply satisfy it by assigning, say, $x_2 = 0$. Therefore, if we assigne $x_1 = 1$, $x_2 = 0$, and arbitrarily assign values to x_3 and x_4, we make the given function have value 1. That is, the given function is satisfiable.

(b) This is a little bit more complicated, compared to (a), but still quite easy. Note that x_1 appears in three clauses: $(x_1+\overline{x}_2+\overline{x}_4)$, $(x_1+x_2+\overline{x}_3)$, and $(x_1+\overline{x}_3+\overline{x}_4)$. Thus, assigning $x_1 = 1$ makes these three clauses satisfied. For the remaining three clauses $(\overline{x}_1+\overline{x}_2+x_3)$, $(\overline{x}_1+x_2+\overline{x}_4)$, and $(\overline{x}_1+\overline{x}_3+\overline{x}_4)$ (note that $\overline{x}_1 = 0$ after assigning $x_1 = 1$), assigning $x_4 = 0$ satisfies two (i.e., $(\overline{x}_1 + x_2 + \overline{x}_4)$ and $(\overline{x}_1 + \overline{x}_3 + \overline{x}_4)$), and assigning $x_3 = 1$ satisfies the last one $(\overline{x}_1 + \overline{x}_2 + x_3)$. Thus, the assignment $x_1 = 1$, $x_3 = 1$, $x_4 = 0$ (with x_2 assigned arbitrarily) will make the given function equal to 1. In conclusion, the given function is satisfiable.

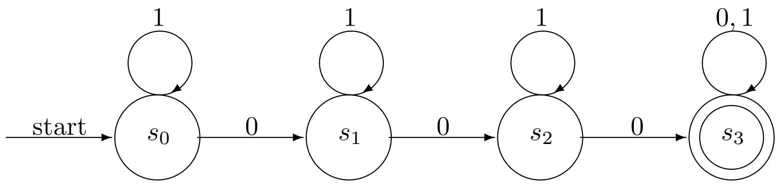
(c) This is quite long, but we can still make it satisfied, by trying several possible assignments. For example, by assigning $x_1 = 1$, we satisfy four of the eight clauses: $(x_1 + x_2 + x_3)$, $(x_1 + \overline{x}_2 + \overline{x}_4)$, $(x_1 + \overline{x}_3 + x_4)$, and $(x_1 + \overline{x}_2 + \overline{x}_3)$. For the remaining four clauses $(\overline{x}_1 + x_3 + x_4)$, $(\overline{x}_1 + x_2 + \overline{x}_4)$, $(\overline{x}_1 + \overline{x}_2 + x_4)$, and $(\overline{x}_1 + \overline{x}_3 + \overline{x}_4)$ (again noticing that $\overline{x}_1 = 0$), assigning $x_4 = 0$ satisfies $(\overline{x}_1 + x_2 + \overline{x}_4)$ and $(\overline{x}_1 + \overline{x}_3 + \overline{x}_4)$, assigning $x_2 = 0$ satisfies $(\overline{x}_1 + \overline{x}_2 + x_4)$, and assigning $x_3 = 1$ satisfies $(\overline{x}_1 + x_3 + x_4)$. Thus, the assignment $x_1 = 1$, $x_2 = 0$, $x_3 = 1$, and $x_4 = 0$ makes the given function to have value 1. In conclusion, the given function is satisfiable.

5. (a) Construct a deterministic finite-state automaton that recognizes the set of all bit strings that contain at least three 0s.

(b) Construct a deterministic finite-state automaton that recognizes the set of all bit strings that contain an even number of 1s.

Solution. The deterministic finite-state automata are given in the following.

(a)



(b)