Ryan Farruggia

CSC 28

Professor Mayer

17 March 2021

Midterm Exam #1

1. The power set P(S) can be defined as the set with all possible subsets of S as each of its individual elements. P(S) will contain 2n elements. If S is empty, P(S) still contains one element which is the empty set S.

2. There exist **450** even numbers *n* that satisfy the given logical statement. Set **S** contains subset **N** where *n* is even and lies between 99 and 999 (including 99 and 999).

3. { {}, {{m},{n},{o}}, {{m,n}, {{m,o}}, {n,o}}, {{m,n,o}}}

4. S1 contains 4 elements: { 12, 1943, 2000, 2019 }

S2 contains 6 elements: { 2019, 2020, 6, 1944, 2000, 12 }

S1 U S2 contains 7 elements: { 6, 12, 1943, 1944, 2000, 2019, 2020 }

5. A recursive function relies on solutions to smaller pieces of a similar or identical function to have a solution for itself.

6.

|  |  |  |
| --- | --- | --- |
| **a** | **b** | **(a → b)** |
| F | F | T |
| F | T | T |
| T | F | F |
| T | T | T |

7. A tautology is a Boolean expression that evaluates to true regardless of accompanying expressions or arguments. A contradiction is a proof by which something is assumed to be false, but is then further proven to be true.

8. Start with the base case. This should hold a property (P) to be true for an initial element *n* with an indicated index (often 0). This can be written as P(0). Move to induction which requires that a hypothesis be made including the assertions made in the base case/starting claim. Finally, prove that because property P(n) holds true for the initial case, it is also true for P(n+1) and every iteration following it: (n+2), (n+3), …

9. I think there can certainly be perfect encryption methods, but in contrast I do not believe perfect security will ever exist. If we ever reach a point where quantum computers become commercially available, our current standard of 256-bit security will become immediately exploitable. Given infinite time, almost any type of encryption can eventually be cracked.

10. There are 4 levels of Chomsky’s hierarchy. Standard programming languages fall into the Type-3 category for Regular Grammar.

11. ¬ (A V B) = ¬ (A) ∧ ¬ (B) 🡪 not (A or B) equals not A and not B

¬ (A ∧ B) = ¬ (A) V ¬ (B) 🡪 not (A and B) equals not A or not B

12. Cardinality of set X: 6

13. for (int i = 0; i < a[MAX]; i++) {

if (a[val]) {

cout << a[val];

} else {

cout << “not found.”;

}  
 }

14. |A ⋃B ⋃ C| = |A| + |B| + |C|

-|A ⋂ B| - |A ⋂ C| - |B ⋂ C|

+|A ⋂ B ⋂ C|

15. Prove P(*n*) for *n* > 1:

Let the base case = P(2) where k = n(n+1)

P(2) = 2(2+1)/2 = 3 = True, since P(1) + P(2) = 3  
Hypothesis: S = k/2

k/2 = n(n+1)/2, where n is a positive integer.

This statement proves to be true for all numbers *n* such that n > 0.