ICCS310: Assignment 1

Possawat Sanorkam
possawat2017@hotmail.com
January 13, 2021

1: Review: Something About Sets

---

(1) Let A_1, A_2, A_3 be any sets from a universe $\mathcal{U}$. *Prove that* $\overline{A_1 \cup A_2 \cup A_3} = \overline{A_1} \cap \overline{A_2} \cap \overline{A_3}$.

Proof: We want to show that $\overline{A_1 \cup A_2 \cup A_3} \subseteq \overline{A_1} \cap \overline{A_2} \cap \overline{A_3}$ and $\overline{A_1} \cap \overline{A_2} \cap \overline{A_3} \subseteq \overline{A_1 \cup A_2 \cup A_3}$

Let A_1, A_2, A_3 be any three given sets. We'll first prove that $\overline{A_1 \cup A_2 \cup A_3} \subseteq \overline{A_1} \cap \overline{A_2} \cap \overline{A_3}$. Let $x \in \overline{A_1 \cup A_2 \cup A_3}$. Then, $x \notin A_1 \cup A_2 \cup A_3$ by the definition of complement, so then $x \notin A_1$, $x \notin A_2$ and $x \notin A_3$, by the definition of union. This means that $x \in \overline{A_1}$, $x \in \overline{A_2}$, and $x \in \overline{A_3}$, by the definition of complement. Hence, $x \in \overline{A_1} \cap \overline{A_2} \cap \overline{A_3}$ since x is in $\overline{A_1}$, $\overline{A_2}$, and $\overline{A_3}$.

Also, we will show that $\overline{A_1} \cap \overline{A_2} \cap \overline{A_3} \subseteq \overline{A_1 \cup A_2 \cup A_3}$. Let $y \in \overline{A_1} \cap \overline{A_2} \cap \overline{A_3}$, so y is in $\overline{A_1}$, $\overline{A_2}$, and $\overline{A_3}$, by the definition of intersection. This means $y \notin A_1$, $y \notin A_2$, and $y \notin A_3$, by the definition of complement. It follows that $y \notin A_1 \cup A_2 \cup A_3$, and so $y \in \overline{A_1 \cup A_2 \cup A_3}$.

In conclusion, $\overline{A_1 \cup A_2 \cup A_3} = \overline{A_1} \cap \overline{A_2} \cap \overline{A_3}$.

(2) Let A and B be any sets from a universe $\mathcal{U}$. *Prove that* $\overline{A \cup B} = \overline{A} \cap \overline{B}$.

Proof: We want to show that $\overline{A \cup B} \subseteq \overline{A} \cap \overline{B}$ and $\overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$

Let A and B be any two given sets. We'll first prove that $\overline{A \cup B} \subseteq \overline{A} \cap \overline{B}$. Let $x \in \overline{A \cup B}$. Then, $x \notin A \cup B$ by the definition of complement, so then $x \notin A$, and $x \notin B$, by the definition of union. This means that $x \in \overline{A}$, and $x \in \overline{B}$, by the definition of complement. Hence, $x \in \overline{A} \cap \overline{B}$ since x is in $\overline{A}$ and $\overline{B}$.

Also, we will show that $\overline{A} \cap \overline{B} \subseteq \overline{A \cup B}$. Let $y \in \overline{A} \cap \overline{B}$, so y is in $\overline{A}$ and $\overline{B}$, by the definition of intersection. This means $y \notin A$ and $y \notin B$, by the definition of complement. It follows that $y \notin A \cup B$, and so $y \in \overline{A \cup B}$.

In conclusion, $\overline{A \cup B} = \overline{A} \cap \overline{B}$.

2: Prime and Irrational

---

(1) Let $p \geq 2$ be a prime and a be a positive integer. Prove that if p divides a^2, then p divides a.

Proof:

(2) Prove that if p is any positive prime number, then $\sqrt{p}$ is irrational.

Proof:

3: Spacing

Possawat Sanorkam
possawat2017@hotmail.com

(1) Prove *Theorem*: Any 2^n by 2^n grid with one painted cell can be tiled using L-shaped triominoes such that the entire grid is covered by triominoes but no triominoes overlap with each other nor the painted cell

Predicate : P(x) $\equiv \forall x \geq 1$, 2^x by 2^x grid with one painted cell can be tiled using L-shaped triominoes such that the entire grid is covered by triominoes but no triominoes overlap with each other nor the painted cell

Base case : P(1) $\equiv$ one of the basic 4 missing tiles perfectly fits 2 by 2 grid with one painted cell since there is one empty space left which is true

Inductive Steps : Assume that if P(x) is true then P(x+1) is true
P(x) $\equiv$ missing tiles perfectly fits 2^x by 2^x grid with one painted cell
P(x+1) $\equiv$ missing tiles perfectly fits 2^{x+1} by 2^{x+1} grid with one painted cell

Assume that each grid was filled up before having its tiles taken out by one for painting that cell. We can break it down to many $2 * 2$ grid which can fit at least one trimino according to P(1) Moreover, since P(1) is true, the size of the grid will always left an even number of empty space which is equal to 4^{x-1} which can be divided by 3 and left a remainder = 1

The total missing tiles for 2^x by 2^x grid is 4^{x-1} tiles
The total missing tiles for 2^{x+1} by 2^{x+1} grid is 4^x tiles

To show that this is true in mathematically way, starting at 2^x by 2^x grid

$$4^{x-1} = 3a + 1 \quad (1)$$
$$4^x = 12a + 4 \quad (2)$$

Now, for 2^{x+1} by 2^{x+1} grid given that b is the number of triminoes used to filled the bigger grid

$$4^x = 3b + 1 \quad (3)$$
$$12a + 4 = 3b + 1 \quad (4)$$
$$3a + 1 = \frac{3b+1}{4} \quad (5)$$

So, it is possible for all grids since any 3n+1 = 4*m so $\frac{3b+1}{4}$ will be an integer. Also, since it is true for P(x), the 2^{x+1} by 2^{x+1} grid which made from 4 grids of 2^x by 2^x grid will be able to fit one painted cell into the grid.

From the induction hypothesis, P(x-1) $\implies$ P(x) and P(x) holds for all $x \geq 1$, 2^x by 2^x grid with one painted cell can be tiled using L-shaped triominoes such that the entire grid is covered by triominoes but no triominoes overlap with each other nor the painted cell Q.E.D.

4: Curious Fact about Graphs

Done. Check the text file for ID.

5: Basic DFAs

(1) Prove *Theorem*: for any $n \geq 0$, `Solve_Hanoi` (n, From_Peg, To_Peg, Aux_Peg) generates exactly $2^n - 1$ lines of instruction

Predicate : P(x) $\equiv$ for any $x \geq 0$, `Solve_Hanoi` (x, ...) generates exactly $2^x - 1$ lines of instruction

Base case : P(0) $\equiv$ `Solve_Hanoi` (n, ...) generates exactly 0 lines of instruction which is true

Inductive Steps : Assume that if P(x) is true then P(x+1) is true
P(x) $\equiv$ `Solve_Hanoi` (x, ...) generates exactly $2^x - 1$ lines of instruction
P(x+1) $\equiv$ `Solve_Hanoi` (x, ...) generates exactly $2^{x+1} - 1$ lines of instruction

To show that this is true in mathematically way, T(x) is the number of line generated from the function using recurrence.

$$T(x) = 2T(x-1) + 1; T(1) = 1; T(0) = 0 \tag{6}$$
$$T(x) = 2^{x-1} + ... + 2 + 1 \tag{7}$$
$$T(x) = 2^x - 1 \tag{8}$$
$$LHS = 2^x - 1; T(x) \tag{9}$$
$$RHS = 2^x - 1 \tag{10}$$
$$LHS \equiv RHS \tag{11}$$

So, P(x) is true, this time we will prove the P(x+1) by using the equations above.

$$T(x+1) = 2T(x) + 1; T(1) = 1; T(0) = 0 \tag{12}$$
$$T(x+1) = 2^x + ... + 2 + 1 \tag{13}$$
$$T(x+1) = 2^x + 2^x - 1; \tag{14}$$

Using T(x) to solve the equation below

$$T(x+1) = 2^{x+1} - 1 \tag{15}$$
$$LHS = 2^{x+1} - 1; T(x+1) \tag{16}$$
$$RHS = 2^{x+1} - 1; \tag{17}$$
$$LHS \equiv RHS \tag{18}$$

From the induction hypothesis, P(x-1) $\implies$ P(x) and P(x) holds for any $x \geq 0$, `Solve_Hanoi` (x, ...) generates exactly $2^x - 1$ lines of instruction. Q.E.D.

(3) Prove *printRuler*:

These are the equations we know from this problem
$f(n) = 2f(n-1) + 1, f(0) = 0$ is number of lines
$g(n) = 2g(n-1) + n, g(0) = 0$ is number of dashes
$g(n) = a * f(n) + b * n + c$

1. Basically, I just followed the hint

$$\begin{aligned} g(0) &= a * f(0) + b * 0 + c && (19)\\ g(0) &= c && (20)\\ g(0) &= 0 && (21) \end{aligned}$$

So, c = 0

2. We will find a and b

$$\begin{aligned} g(n) &= 2g(n-1) + n && (22)\\ a * f(n) + b * n &= 2(a * f(n-1) + b * (n-1)) + n && (23)\\ a * f(n) + b * n &= 2a * f(n-1) + 2b * (n-1) + n && (24)\\ a * f(n) &= 2a * f(n-1) + b * n - 2b + n && (25)\\ a * f(n) - 2a * f(n-1) &= b * n - 2b + n && (26) \end{aligned}$$

Let's do it side by side

$$\begin{aligned} a(f(n) - 2f(n-1)) &= b * n + n - 2b && (27)\\ a(1) &= n * (b+1) - 2b && (28)\\ a + 2b &= n * (b+1) && (29) \end{aligned}$$

$$a + 2b - n * (b+1) = 0 \quad (30)$$

To find a and b, we know that substitute P and Q = 0 will solve this equation

$$\begin{aligned} P + Qn &= 0 && (31)\\ a + 2b &= P && (32)\\ (b+1) &= Q && (33)\\ b &= -1 && (34)\\ a &= 2 && (35) \end{aligned}$$

$$\begin{aligned} g(n) &= a * f(n) + b * n && (36)\\ g(n) &= 2 * f(n) - n && (37) \end{aligned}$$

3. Previously, we got $g(n) = 2 * f(n) - n$
Also, $f(n) = 2^n - 1$. In fact, $g(n) = 2^{n+1} - n - 2$.

4. Theorem : $g(n) = 2^{n+1} - n - 2$ works for all $n \geq 0$
Predicate : P(x) $\equiv g(x) = 2^{x+1} - x - 2$ works for all $x \geq 0$
Base case : P(0) $\equiv g(0) = 0$ is true
Inductive Steps : Assume that if P(x) is true then P(x+1) is true
P(x) $\equiv g(x) = 2^{x+1} - x - 2$
P(x+1) $\equiv g(x+1) = 2^{x+2} - x - 3$
Actually, we know that $g(n) = 2g(n-1) + n$ has a close form of $g(x) = 2^{x+1} - x - 2$ according to what we have done on part 2.

So, P(x) is true, this time we will prove the P(x+1) by using the equations above.

$$g(x+1) = 2^{x+2} - x - 3 \quad (38)$$
$$g(x+1) = 2 * 2^{x+1} - x - 3 \quad (39)$$
$$g(x+1) = 2(g(x) + x + 2) - x - 3 \quad (40)$$
$$g(x+1) = 2g(x) + (x+1) \quad (41)$$
$$LHS = 2^{x+2} - x - 3 \quad (42)$$
$$RHS = 2g(x) + (x+1) \quad (43)$$
$$LHS \equiv RHS \quad (44)$$

From the induction hypothesis, P(x) $\implies$ P(x+1) and P(x) holds for any $x \geq 0$ which will make $g(x) = 2^{x+1} - x - 2$ true. So, $g(x) = 2^{x+1} - x - 2$ works. Q.E.D.

6: Penultimate

(1) Prove *Theorem*: for any $n \geq 0$, `Solve_Hanoi` (n, From_Peg, To_Peg, Aux_Peg) generates exactly $2^n - 1$ lines of instruction

Predicate : P(x) $\equiv$ for any $x \geq 0$, `Solve_Hanoi` (x, ...) generates exactly $2^x - 1$ lines of instruction

Base case : P(0) $\equiv$ `Solve_Hanoi` (n, ...) generates exactly 0 lines of instruction which is true

Inductive Steps : Assume that if P(x) is true then P(x+1) is true
P(x) $\equiv$ `Solve_Hanoi` (x, ...) generates exactly $2^x - 1$ lines of instruction
P(x+1) $\equiv$ `Solve_Hanoi` (x, ...) generates exactly $2^{x+1} - 1$ lines of instruction

7: Digit Sum

(1) Prove *Theorem*: for any $n \geq 0$, `Solve_Hanoi` (n, From_Peg, To_Peg, Aux_Peg) generates exactly $2^n - 1$ lines of instruction

Predicate : P(x) $\equiv$ for any $x \geq 0$, `Solve_Hanoi` (x, ...) generates exactly $2^x - 1$ lines of instruction

Base case : P(0) $\equiv$ `Solve_Hanoi` (n, ...) generates exactly 0 lines of instruction which is true

Inductive Steps : Assume that if P(x) is true then P(x+1) is true
P(x) $\equiv$ `Solve_Hanoi` (x, ...) generates exactly $2^x - 1$ lines of instruction
P(x+1) $\equiv$ `Solve_Hanoi` (x, ...) generates exactly $2^{x+1} - 1$ lines of instruction