Formal Languages and the Theory of Computation
Assignment 1

Joshua Lande

January 22, 2006

1 Problems

Problems taken from Chapter 0 of [1].

Problem 1.1 *Problem* 0.3 *from [1]. Let A be the set $\{x, y, z\}$ and let B be the set $\{x, y\}$. Is A a subset of B? Is B a subset of A? What is $A \cup B$? What is $A \cap B$? What is $A \times B$? What is the power set of B?*

- A is not a subset of B, because $z \in A$, but $z \notin B$.
- B is a subset of A, because every item in B is also in A.
- $A \cup B = \{x, y, z\}$.
- $A \cap B = \{x, y\}$.
- $A \times B = \{(x, x), (x, y), (y, x), (y, y), (z, x), (z, y)\}$.
- $\mathcal{P}(B) = \{\emptyset, \{x\}, \{y\}, \{x, y\}\}$.

Problem 1.2 *Problem* 0.7 *from [1]. For each part, give a relation that satisfies the condition:*

1. *Reflexive and symmetric but not transitive.*
2. *Reflexive and transitive but not symmetric.*
3. *And symmetric and transitive but not reflexive.*

1. We can use the relation $R = \{(a, b)|a, b \in \mathbb{Z}$ and $a + b$ is odd or $a \times b$ is a perfect square$\}$. This is a little bit kluge but it works. aRa is true for all a because $a \times a$ is a perfect square. If aRb, then bRa because either $a + b$ is odd, in which case $b + a$ is also odd, or $a \times b$ is a perfect square, in which case $b \times a$ is also a perfect square. But R is not transitive because $3R2$ and $2R1$ but not $3R1$.

2. $\leq$ works because $a \leq a$ for all a. If $a \leq b$ and $b \leq c$, then $a \leq c$. But it fails symmetry, $1 \leq 2$ but $2 \not\leq 1$.

3. We can use the relation $R = \{(a, b)|a \text{ and } b \text{ are both even}\}$. It is symmetric because if aRb, then b and a are both even, so bRa. If aRb and bRc, then aRc because a and c are both even. But it is not reflexive because $1R1$ is not true.

Problem 1.3 *Problem* 0.11 *from [1]. Find the error in the following proof that all horses are the same color.*

The proof is good up until the last statement: "Therefore all the horses in H must have the same color". It dose not account for the fact that H_1 and H_2 could be mutually exclusive piles. This happens when $h = 2$ and there are only 2 horses, each which could be different colors. Even though every horse in H_1 and H_2 have the same color, each pile could have have horses of different colors, and H need not have horses of only one color.

Problem 1.4 *Problem* 0.12 *from [1]. Show that every graph with 2 or more nodes contains two nodes that have equal degree.*

If a graph has n nodes, the minimum degree that it can have is 0. A node can touch at maximum the $n - 1$ other nodes, so the maximum degree is $n - 1$. There are n possible degrees. We now break into two separate cases:

- If no node has degree 0, then there are only $n - 1$ possible degrees. Because there are n nodes, two nodes must have the same degree.

- If, on the other hand, one of the nodes of the graph has degree 0, the maximum degree is $n-2$ instead of $n-1$. This is because a degree of $n-1$ only happens when a node touches ever other node. Because no node touches the node with degree 0, the maximum must be $n - 2$. Therefore, there are only $n - 1$ possible degrees and we can again conclude that two nodes must have the same degree.

Problem 1.5 *Show that* $1 \times 2 + 2 \times 3 + \ldots + n(n+1) = n(n+1)(n+2)/3$ *for all positive integer n.*

We can prove this with induction on n. Our base case is $n = 1$. We can easily verify that $1(1 + 1) = 1(1 + 1)(1 + 2)/3$. We then proceed to the induction step, where we assume that

$$1 \times 2 + \ldots + n(n + 1) = n(n + 1)(n + 2)/3$$

and attempt to show that

$$1 \times 2 + \ldots + n(n + 1) + (n + 1)(n + 2) = (n + 1)(n + 2)(n + 3)/3,$$

or that the series is true for $n + 1$. We can do so as follows:

$$
\begin{aligned}
1 \times 2 + \ldots + n(n+1) &= n(n+1)(n+2)/3 \\
1 \times 2 + \ldots + n(n+1) + (n+1)(n+2) &= n(n+1)(n+2)/3 + (n+1)(n+2) \\
&= (n+1)(n+2)\frac{n}{3} + (n+1)(n+2)\frac{3}{3} \\
&= (n+1)(n+2)(n+3)/3
\end{aligned}
$$

Which is precisely what we wanted to show.

References

[1] M. Sipser. *Introduction to the Theory of Computation*. Thomson Learning, Inc., Massachusetts, 2006.