Assignment 9

Karan Sikka
ksikka@andrew.cmu.edu
E
April 12, 2012

Love

The proof is by contradiction.

AFSOC that

$$|L| > |\mathcal{P}(L)|$$

This implies that there exists some surjective function $f : L \to \mathcal{P}(L)$. In other words, f maps elements from L to elements in $\mathcal{P}(L)$

Let S be a subset of L and define $S = \{b : b \notin f(b)\}$. Recall that f(b) is an element in the power set of L, which means it's actually a subset of L. Again, b is in S iff b is not an element of the subset of L which f maps to.

For all elements b in L, there are two cases:$b \in S$, $b \notin S$. Say that s is some element in L such that $f(s) = S$, which must be true for some s since f is surjective.

Case $s \in S$
If $s \in S$ then, by definition of S and f, $s \notin f(s)$. This is a contradiction, because recall that $f(s) = S$, so we are stating the equivalent of $s \notin S$. $s \notin f(s) \implies s \notin S$. This is a contradiction with the premise of the case.

Case $s \notin S$
If $s \notin S$ then, by definition of S, $s \in f(s)$. Recall, that $f(s) = S$, so we are asserting that $s \in S$. This is a contradiction with the premise of the case.

we have disproved the original AFSOC statement and we can conclude that $|\mathcal{P}(L)| > |L|$.

Hope and Comfort

(a) Call the set of the roots of rational polynomials S

Recall that all polynomials have a finite degree, which is a natural number. Also recall that a polynomial with degree k has at most k roots.

We can partition the set of all rational polynomials by their degree. Let P_k be the set of polynomials of degree k. Note that a polynomial of degree k can be uniquely expressed by it's $k+1$ coefficients. You can put the coefficients in an ordered tuple, so the cardinality of P_k is at most $|\mathbb{Q}^{k+1}|$, which is countable since the finite tuple of a countable set is countable (proven in concepts hw last year).

By the def'n of partitioning, $R_1 \cup R_2 \cup ... = S$ where R_i is the set of roots of all polynomials in P_i. Recall that P_i is countable for all $i \in \mathbb{N}$, so the cardinality of R_i is at most i times the cardinality

of P_i, so R_i is countable. Therefore its elements can be bijected to the naturals. Let C_k be a subset of the elements in S, but only the ones in R_i where $i < k$, and only the first k elements of each R_i. The C_k is a finite set: it has cardinality $k * k$ or k^2. Now, we can enumerate the elements of the list formed by the concatenation of $C_1C_2...$ except exclude repetitions. Note that this enumerates the elements in S, and therefore S is countable. □

(b) Let $f : \{X, O\}^\infty \to \{X, O\}^\infty \backslash \{(X, X, X, \dots)\}$.

Let s be a string in $\{X, O\}^\infty$.

Case s has at least one X and no O's: Call n the number of X's s has. f maps s to a string of $2n - 1$ O's.

Case s has at least one O and no X's: Call n the number of O's s has. f maps s to $2n$ O's.

Case s is not one of the above cases,
ie. s must be a combination of X and Os, or it is the empty string: $f(s) = s$

Injectivity: Say you have two strings a, b in the domain of f such that $f(a) = f(b)$. Claim: $f(a) = f(b) \implies a = b$ Proof:

Case $f(a) = f(b)$ has $2n - 1$ O's and 0 X's for some n in the naturals, ie $f(a)$ has an odd number of O's and no X's. Then $f(a) = f(b)$ must have been formed from the first case described above, since the second case only produces strings with an even number of O's, and the third method only produces strings with X's. $f(a)$ and $f(b)$ both have $2n - 1$ O's so a and b must both have exactly n X's. Then $a = b$.

Case $f(a) = f(b)$ has $2n$ O's and 0 X's for some n in the naturals, ie $f(a)$ has an even number of 0's and no X's. Then $f(a) = f(b)$ must have been formed from the second case described above, by similar logic above. $f(a)$ and $f(b)$ both have $2n$ O's so a and b must both have exactly n O's. Then $a = b$.

Case $f(a) = f(b)$ has a positive number of X's. Then it must have been formed by the third case above, because that is the only way to produce strings with X's in them. This is the identity function, so $f(a) = f(b) \implies a = b$ by the definition of the identity function.

Case $f(a) = f(b)$ is the empty string. a and b must both be the empty string and $f(a)$ must have been formed by case 3. Then $a = b$ since the empty string equals itself.

Surjectivity: Say you have two strings a, b in the domain of f such that $f(a) = f(b)$. Claim: f is onto. Proof: Case 1 of f produces all strings with an odd number of Os and no Xs, for all odd natural numbers. Case 2 of f produces all strings with an even number of Os and no Xs, for all even natural numbers. Case 3 of f is the identity function for all inputs not taken care of by Case 1 and 2. Recall that the input string must be a combination of X and Os, or it the empty string to be in this case. Therefore, this case produces combinations of X and Os, or the empty string, by the definition of the identity function.

Therefore, f produces strings with a positive number of O's and no X's, and it produces all other combinations of X and O except strings consisting of X's without O's. Therefore every element in the target of f can be mapped to from the domain of f. f is onto.

Therefore we have shown a direct bijectiion f between the two sets.

Patience

Grace

Lemma: For $i, j \in \mathbb{N}, i < j$, there exists an $x \in \mathbb{N}, x \geq 1$ such that $i + x$ is prime and $j + x$ is composite, or vice versa.

The proof is by induction on j

Base Case: If $j = 0$, then it is not possible to find an i less than j in the naturals. So we start with $j = 1$.

If $j = 1$, then i must be 0. Let x be a prime number greater than 2. $i + x = 0 + x = x$, which we already established was prime. $j + x = 1 + x$, which must be an even number since all primes greater than two are odd, and one plus an odd number is an even number. Therefore, $j + x$ is composite since it is an even number greater than 2. The claim holds for the base case.

Inductive Hypothesis: Assume that for some $j \in \mathbb{N}$, the lemma holds all values from 1 upto and including j.

Inductive Step: WTS that the lemma holds for $j + 1$.

Case $i = 0$: Let x be a prime factor of $j + 1$. Therefore,

$$x | j + 1$$

$$\implies x | (j + 1 + x)$$

Which means $j + 1 + x$ is composite since x is greater than 1. Since $x + 0$ is prime and $j + 1 + x$ is composite, the inductive step holds for this case.

Case $0 < i < j$: By the IH, there exists an x that satisfies $i - 1$ and j, by our IH. This implies that we have some x_{IH} such that one element in $\{(i - 1 + x_{IH}), (j + x_{IH})\}$ is prime and one is composite. We can rewrite this as follows:

$$\{(i - 1 + x_{IH}), (j + x_{IH})\}$$

$$\implies \{(i + (x_{IH} - 1)), (j + 1 + (x_{IH} - 1))\}$$

If we set our new x to be $x_{IH} - 1$, then we get the set $\{i + x, j + x\}$. Since we haven't modified the original set in doing so, the claim still holds for this case.

The inductive step holds in all cases of i. We have proved the lemma by induction. □

Proof: AFSOC that the language $L = \{a^c | c \text{ is composite}\}$ can be modeled by a DFA M. Say WLOG that M has k states. Given $k + 1$ distinct elements in L, at least two elements must terminate on the same state, by the pidgeonhole principle. Say those strings are a^i and a^j. Note that $i \neq j$. Since they terminate on the same state, they are either both composite or both prime. Append a^x to a^i and a^j, and you should get a^{i+x} and a^{j+x} respectively. When you put them in the DFA, they will end on the same state (we appended the same number of a's to both strings) which

means that they are both prime or both composite. This is a contradiction because the lemma says we can choose x such that either $i + x$ or $j + x$ is composite and the other is prime.

Thus, there is no DFA for which decides the language $L = \{a^c | c \text{ is composite}\}$. The language is irregular. □

Joy

(a)
Claim:
$L_1 \cup L_2$ is regular and L_1 is finite $\implies$ L_2 is regular.

Proof:
We know by the Principle of Inclusion/Exclusion that
$|L_1 \cup L_2| = |L_1 + L_2| - |(L_1 \cap L_2)|$
$\implies |L_2| = |(L_1 \cup L_2)| - |L_1| + |(L_1 \cap L_2)|$
$\implies |L_2| = |(L_1 \cup L_2)| - (|L_1| - |(L_1 \cap L_2)|)$
Assume $L_1 \cup L_2$ is regular and L_1 is finite. Since L_1 is finite, $L_1 \cap L_2$ is finite. Therefore, $L_1 - (L_1 \cap L_2)$ is finite because the difference of two finite languages is finite, and therefore a regular language. $(L_1 \cup L_2) - (L_1 - (L_1 \cap L_2))$ is then the difference of two regular languages, which, by problem 5 in this assignment, is a regular language. □

(b)
Claim:
$L_1 \cup L_2$ is regular and L_1 is regular $\implies$ L_2 is regular.

Counterexample:
Let $L_1 = \Sigma^*$ with $\{0, 1\} = \Sigma$. L_1 is regular, the DFA is that which has a single accept state with an arrow to itself, accepting all strings. Then let $L_2 = \{0^n 1^n : n \in \mathbb{N}\}$. L_2 is not regular as shown in lecture. However, $L_1 \cup L_2$ is regular because $L_2 \subseteq L_1$ so $L_1 \cup L_2 = L_1$, which is regular.

(c)
Claim:
$L_1 \cdot L_2$ is regular and L_1 is finite $\implies$ L_2 is regular.

Counterexample:
Let $L_2 = \{a^c | c \text{ is composite}\}$ which is irregular as proved in problem 3. Let a $L_1 = \{a\}$. Observe that 2 is the only prime even number and any even number greater than 2 is composite. Hence, let L_3 be a subset of L_2 such that $L_3 = \{a^x | x > 2 \wedge \text{ x is even}\}$, and the concatenation of the elements in L_3 with $a \in L_1$ is $L_4 = \{a^x | x > 3 \wedge \text{ x is odd}\}$. $L_1 \cdot L_2 = \{a^n | n \geq 4\}$ is regular since $L_3 \subseteq L_2$ and $L_1 \cdot L_3 = L_3 \cup L_4 = \{a^n | n \geq 4\}$, and we can make a DFA that reaches an accepting state that self-loops after four occurrences of a. Therefore, it is possible for L_2 to not be regular when $L_1 \cdot L_2$ is regular and L_1 is finite.

(d)
Claim:

$L_1 \cdot L_2$ is regular and L_1 is regular $\implies$ L_2 is regular.

Counterexample:
Refer to the proof of (c), and note that if L_1 is finite then it is regular. This was proven in lecture.

(e)
Claim:
L_1^* is regular $\implies$ L_1 is regular.

Counterexample:
Let $L_1 = \{a^c | c \text{ is composite}\} \cup \{a\}$. Proof: That L_1 is not regular.
AFSOC L_1 is regular. $\{a\}$ is regular because it is finite.

If $\{a^c | c \text{ is composite}\} \cup \{a\}$ and $\{a\}$ are regular then $\{a^c | c \text{ is composite}\}$ is regular because the difference of two regular languages is regular (given in the problem statement for 5). However, as shown in problem 3, $\{a^c | c \text{ is composite}\}$ is not regular, so this is a contradiction and what was assumed must be false. Therefore the claim is not true.

Then $L_1^* = \{a^n | n \in \mathbb{N}\}$ because L_1 contains $\{a\}$. This is a regular language because a DFA with a single accept state which loops back to itself and accepts any string of a's will accept L_1^*. Therefore, we have shown an irregular L_1 and a regular L_1^*.