Assignment 2

Rachel Hwang

October 17, 2013

1 Exercise 0.1

Prove that the following languages are not regular.

1. $L = \{0^n : \text{n is a perfect square}\}$
   Since the set of perfect cubes is infinite, L is infinite. Suppose L is regular. If so, by the Pumping Lemma, $\exists$ integer N, st. for every string $w \in L$, $|w| \geq N$, we can find strings $x, y, z \in L$ st. $w = xyz$ and $|y| > 0$, $|xy| \leq N$ and $xy^kz \in L$ for all $k \geq 0$.

   Let $w = 000...00 = xyz$ where $|w| = N^2, |xy| \leq N, |y| = k$

   By the Pumping Lemma, $xy^2z \in L$.

   However, $|xy^2z| = |xyz| + |y| = N^2 + k$

   Since $k > 0$,
   $N^2 + k < N^2 + N < N^2 + 2N + 1$
   $N^2 + k < (N + 1)^2$
   Since $k > 0$, $N^2 < N^2 + k$
   Because $N^2 < N^2 + k < (N + 1)^2$, $xy^2z \notin L$

   This contradicts the Pumping Lemma, so L is not regular.

2. $L = \{0^n : \text{n is a perfect cube}\}$
   Since the set of perfect cubes is infinite, L is infinite. Suppose L is regular. If so, by the Pumping Lemma, $\exists$ integer N, st. for every string $w \in L$, $|w| \geq N$, we can find strings $x, y, z \in L$ st. $w = xyz$ and $|y| > 0$, $|xy| \leq N$ and $xy^kz \in L$ for all $k \geq 0$.

   Let $w = 000...00 = xyz$ where $|w| = N^3, |xy| \leq N, |y| = k$

   By the Pumping Lemma, $xy^2z \in L$.

   However, $|xy^2z| = |xyz| + |y| = N^3 + k$

   Since $k > 0$,
   $N^3 + k < N^3 + N < N^3 + 3N^2 + 3N + 1$
   $N^3 + k < (N + 1)^3$
   Since $k > 0$, $N^3 < N^3 + k$
   Because $N^3 < N^3 + k < (N + 1)^3$, $xy^2z \notin L$

   This contradicts the Pumping Lemma, so L is not regular.

3. $L = \{0^n : \text{n is a power of 2}\}$
   Since the set of powers of 2 is infinite, L is infinite. Suppose L is regular. If so, by the Pumping Lemma, $\exists$ integer N, st. for every string $w \in L$, $|w| \geq N$, we can find strings $x, y, z \in L$ st. $w = xyz$ and $|y| > 0$, $|xy| \leq N$ and $xy^kz \in L$ for all $k \geq 0$.

   Let $w = 000...00 = xyz$ where $|w| = 2^N, |xy| \leq N, |y| = k$

By the Pumping Lemma, $xy^2z \in L$.

However, $|xy^2z| = |xyz| + |y| = 2^N + k$

Since $k > 0, 2^N > N$
$2^N + k < 2^N + N < 2^N + 2^N$
$2^N + k < 2^{N+1}$
Since $k > 0$, $2^N < 2^N + k$
Because $2^N < 2^N + k < 2^{N+1}$, $xy^2z \notin L$

This contradicts the Pumping Lemma, so L is not regular.

4. The Pumping Lemma proof for this set would look identical to that of problem 1.

5. $L =$ The set of strings of 0's and 1's that are of the form ww, that is, some string repeated.
L is infinite. Suppose L is regular. If so, by the Pumping Lemma, $\exists$ integer N, st. for every string $w \in L$, $|w| \geq N$, we can find strings $x, y, z \in L$ st. $w = xyz$ and $|y| > 0$, $|xy| \leq N$ and $xy^kz \in L$ for all $k \geq 0$.

Let $w = 1^N0^N1^N0^N = xyz$, where $|xy| \leq N, |y| = k$

By the Pumping Lemma, $xy^2z \in L$.

However, since $|xy| < N$, $y = 1^k$
This means that $xy^2z = 1^{N+k}0^N1^N0^N$, therefore, $xy^2z \notin L$

This contradicts the Pumping Lemma, so L is not regular.

6. $L =$ The set of strings of 0's and 1's that are of the form ww^R , that is, some string followed by its reverse.
L is infinite. Suppose L is regular. If so, by the Pumping Lemma, $\exists$ integer N, st. for every string $w \in L$, $|w| \geq N$, we can find strings $x, y, z \in L$ st. $w = xyz$ and $|y| > 0$, $|xy| \leq N$ and $xy^kz \in L$ for all $k \geq 0$.

Let $w = 1^N0^N0^N1^N = xyz$, where $|xy| \leq N, |y| = k$

By the Pumping Lemma, $xy^2z \in L$.

However, since $|xy| < N$, $y = 1^k$
This means that $xy^2z = 1^{N+k}0^N0^N1^N$, therefore, $xy^2z \notin L$

This contradicts the Pumping Lemma, so L is not regular.

7. $L =$ The set of strings of 0's and 1's of the form ww', where w' is formed from w by replacing all 0's by 1's, and vice-versa; e.g. $011 = 100$, and 011100 is an example of a string in the language.

Let $w = 0^N1^N1^N0^N = xyz$, where $|xy| \leq N, |y| = k$

By the Pumping Lemma, $xy^2z \in L$.

However, since $|xy| < N$, $y = 0^k$
This means that $xy^2z = 0^{N+k}1^N1^N0^N$, therefore, $xy^2z \notin L$

This contradicts the Pumping Lemma, so L is not regular.

8. $L =$ The set of strings of the form $w1^n$, where w is a string of 0's and 1's of length n.
L is infinite. Suppose L is regular. If so, by the Pumping Lemma, $\exists$ integer N, st. for every string $w \in L$,

$|w| \geq N$, we can find strings $x, y, z \in L$ st. $w = xyz$ and $|y| > 0$, $|xy| \leq N$ and $xy^kz \in L$ for all $k \geq 0$.

Let $w = 0^N1^N = xyz$ where $|xy| \leq N, |y| = k$

By the Pumping Lemma, $xy^2z \in L$.

However, since $|xy| < N$, $y = 0^k$
This means that $xy^2z0^{N+k}1^N$, therefore, $xy^2z \notin L$

This contradicts the Pumping Lemma, so L is not regular.

2 Exercise 0.2

Show that the regular languages are closed under the following operations:

1. $min(L) = \{$w : w is in L, but no proper prefix of w is in $L\}$

   Conceptually, given the DFA A_L for L, $min(L)$ is simply the set of strings accepted by the machine which do not pass through any accepting states in the intermediate. This is the same as modifying A_L by deleting all outgoing edges from the final states. This ensures that no element of $min(L)$ is ever contains another element of $min(L)$ as a prefix. Formally,

   Let DFA of L be $A_L = (Q, \sigma, \delta, q, F)$. The DFA of $min(L)$, $A_{min(L)}$ is simply a modification. Let p be a state in L and a be an input symbol. If $p \in F$, then $\delta_{min(L)}(p, a) = \emptyset$. If $p \notin F$, then $\delta_{min(L)}(p, a)$ is unchanged. The rest of $A_{min(L)}$ is the same as A_L.

   $A_{min(L)}$ is a valid DFA for $min(L)$, so $min(L)$ is regular.

2. $max(L) = \{$w : w is in L and for no x other than ε is wx in $L\}$.

   Conceptually, given the DFA A_L for L, $max(L)$ is simply the set of strings accepted by the machine which do not serve as prefixes for any other element in L. This is the same as modifying A_L by making all accepting states which have outgoing edges into non-accepting states. This ensures that no element of $max(L)$ is ever a prefix in another element of $max(L)$. Formally,

   Let DFA of L be $A_L = (Q, \sigma, \delta, q, F)$. The DFA of $max(L)$, $A_{max(L)}$ is simply a modification. Let f be a state in F and a be an input symbol. If $\exists\delta(f, a)$, then $f \notin F_{max(L)}$. $A_{max(L)}$ is otherwise the same as A_L.

   $A_{max(L)}$ is a valid DFA for $max(L)$, so $max(L)$ is regular.

3. $init(x) = \{$w : for some x, wx is in $L\}$.

   Conceptually, given the DFA A_L for L, $init(L)$ is simply the set of strings accepted by the machine which serve as prefixes for any other element in L. This is the same as modifying A_L by making all accepting states which have no outgoing edges into non-accepting states. This ensures that we only have elements of L that prefix another element of L. Formally,

   Let DFA of L be $A_L = (Q, \sigma, \delta, q, F)$. The DFA of $init(L)$, $A_{init(L)}$ is simply a modification. Let f be a state in F and a be an input symbol. If $\exists\delta(f, a)$, then $f \in F_{init(L)}$, otherwise $f \notin F_{init(L)}$. $A_{init(L)}$ is otherwise the same as A_L.

   $A_{init(L)}$ is a valid DFA for $init(L)$, so $init(L)$ is regular.

3 Exercise 0.3

Give an algorithm to tell whether two regular languages L1 and L2 have at least one string in common.

If L1 and L2 have at least one string in common, this is the same as saying L1 ∩ L2 is non-empty. We can therefore build a DFA which represents the intersection of L1 and L2 and see if it has any accepting states accessible from its state state.

1. Construct the DFA A for $L1 \cap L2$.
   (This is the intersection DFA from the textbook.)
   Let $A_{L1} = (Q_{L1}, \sigma, \delta_{L1}, q_{L1}, F_{L1})$ and $A_{L2} = (Q_{L2}, \sigma, \delta_{L2}, q_{L2}, F_{L2})$ be the automata representing $L1$ and $L2$. Note that if L1 and L2 do not use the same alphabet, then σ is the union of their alphabets.

   The states of A are pairs of states, the first from A_{L1} and the second from A_{L2} To design the transtions of A, suppose A is in the state (p, q), where p is the state of A_{L1} and q is the state of A_{L2}. If a is the input symbol, we see what A_{L1} does on that input; say it goes to state s. We also see what A_{L2} does on input a; say it makes a transition to state t. Then the next state of A will be (s, t). In that manner, A has simulated the effect of both A_{L1} and A_{L2}. The start state of A is the pair of start states of A_{L1} and A_{L2}. Since we want to accept if and only if both automata accept, we select as the accepting states of A all those pairs (p, q) such that p is an accepting state of A_{L1} and q is an accepting state of A_{L2}.

   Formally, $A = (Q_{L1} \text{ x } Q_{L2}, \sigma, \delta, (q_{L1}, q_{L2}), F_{L1} \text{ x } F_{L2})$ where $\delta((p, q), a) = (\delta_{L1}(p, a), \delta_{L2}(p, a))$.

2. Run a depth first search on A for accepting states beginning at the start state. If this search returns at least one accepting state that is not the start state, then L1 and L2 have at least one string in common.