Automata, Computability and Complexity with Applications

Additional Exercises

Elaine Rich

Part I: Introduction

1 Why Study Automata Theory?

2 Languages and Strings

1) Let L be the language {Austin, Houston, Dallas}. How many elements are there in L^*?

   Countably infinitely many.

2) Let Σ = {a, b, c}. How many elements are there in Σ^*?

   Countably infinitely many.

3) Let L = {w ∈ {a, b}* : every a in w is immediately followed by a b}. List the first six elements in a lexicographic enumeration of L.

   ε, b, ab, bb, abb, bab

4) Let L = {w ∈ {1, 2, 3}* : $|w|$ is even}. List the first twelve elements in a lexicographic enumeration of L.

   ε, 11, 12, 13, 21, 22, 23, 31, 32, 33, 1111, 1112

5) Are the following sets closed under the following operations? If not, what are their respective closures?
   a) The even length strings over the alphabet {a, b} under Kleene star.

      Closed.

   b) The odd length strings over the alphabet {a, b} under concatenation.

      Not closed. The closure is all strings over the alphabet {a, b} with length at least 1.

6) For each of the following statements, state whether it is *True* or *False*. Prove your answer.
   a) $\forall L\ ((L^+)^+ = L^+)$.

      True.

   b) $\forall L\ ((L^*)^+ = (L^+)^*)$.

      True.

   c) $\forall L\ (L^* = L^+ \cup \varnothing)$.

      False. $L^+ \cup \varnothing = L^+$. If L = {a}, then L^* contains ε, but L^+ does not.

   d) $\forall L_1, L_2, L_3\ ((L_1L_2L_3)^* = L_1^*L_2^*L_3^*)$.

      False.

   e) $\forall L_1, L_2\ ((L_1^* \cup L_2^*) = (L_1^* \cup L_2^*)^*)$.

      False.

f) $\forall L\ (L^*\ L = L^+)$.

True.

g) $\forall L_1, L_2, L_3\ (L_1^*\ (L_2 \cup L_3)^+ = (L_1^*\ L_2^+ \cup L_1^*\ L_3^+))$.

False.

h) $\forall L\ (\varnothing\ L^* = \varnothing)$.

True.

i) $\forall L\ (L\varnothing = L^*)$.

False. $L\varnothing = \varnothing$.

j) $\forall L_1, L_2\ ((L_1 - L_2) = (L_2 - L_1))$.

False. Counterexample:
Let $L_1 = \{\mathtt{a}\}$ and $L_2 = \varnothing$. Then $L_1 - L_2 = \{\mathtt{a}\}$, but $L_2 - L_1 = \varnothing$.

k) $\forall L_1, L_2, L_3\ (((L_1\ L_2) \cup (L_1\ L_3))^* = (L_1\ (L_2 \cup L_3))^*)$.

True.

l) $\forall L\ (((L \cup \{\varepsilon\})^*\ (L \cup \{\varepsilon\})^*)^* = L^*)$.

True.

m) $\forall L\ (\{\varepsilon\} \cup L^+ = L^+)$.

False. Any L that does not contain ε is a counterexample.

n) $\forall L_1, L_2$ where $L_1 \neq L_2$ $((L_1\ L_2) \neq (L_2\ L_1))$.

False. Counterexample:
Let $L_1 = \varnothing$ and $L_2 = \mathtt{a}^*$. Then $L_1\ L_2 = \varnothing = L_2\ L_1$.

o) If $L_1 - L_2$ is finite, then at least one of L_1 or L_2 must also be finite.

False. Counterexample:
Let $L_1 = \{\mathtt{a}\} \cup \mathtt{b}^*$. Let $L_2 = \mathtt{b}^*$. Then $L_1 - L_2 = \{\mathtt{a}\}$, which is finite. But neither L_1 or L_2 is.

p) The set of strings that correspond to binary encodings of positive integers is closed under concatenation.

True.

3 The Big Picture: A Language Hierarchy

1) [Ben Wiedermann] Using the technique we use in Example 3.8 to describe addition, describe exponentiation as a language recognition problem.

> Problem: Given two nonnegative numbers n and m, compute n^m.
> Encoding of the problem: We transform the problem of computing n^m into the problem of checking whether a third number p is the result of raising n to the m power. We can use the same encoding we used in Example 3.5.
> The language to be decided: INTEGEREXP =
> {*w* of the form: <*integer*$_1$>**<*integer*$_2$>=<*integer*3> : each of the substrings <*integer*1>, <*integer*2> and <*integer*3> is an element of $[0 - 9]^+$ and *integer*3 = $integer_1^{integer2}$.

2) [Ben Wiedermann] Consider the problem of computing the length of a route in a weighted graph. Formulate this problem as a language recognition problem. Assume that a route is given as a sequence of edges to be traversed.

> Problem: Given a sequence of edges in an undirected graph, compute the length of the route that is defined by that sequence of edges.
> Encoding of the problem: We transform the problem of computing the length of a route into the problem of checking a proposed length to see if it is correct.
> The language to be decided: ROUTELEN =
> {*w* of the form: <$edge_1$>/<$edge_2$>/ … /<$edge_n$>/<k> : each of the substrings $edge_i$ describes an edge (and its weight), the endpoint of $edge_i$ = the starting point of $edge_{i+1}$, and k is equal to the sum of the edge weights}.

3) Consider the following problem: Given a database D and a query Q, what result is returned when Q is executed against D?

> L = {<D, Q, R> : R is the result of executing Q against D}.

4 Some Important Ideas Before We Start

1) [Ben Wiedermann] Describe in clear English or pseudocode a decision procedure to answer the question: "Given a list of integers N and an individual integer n, does N correspond to a prime factorization of n?

> *decidefactor*(n: integer, N: list of integers) =
>   *result* = 1
>   For each element i of N do:
>   If i is not prime, then halt and return *False*.
>   *result* = *result* * i.
>   If *result* = n then halt and return *True*. Else halt and return *False*.

2) Recall the function *chop*(L) defined in Example 4.10. *Chop*(L) = $\{w : \exists x \in L\ (x = x_1cx_2,\ x_1 \in \Sigma_L^*,\ x_2 \in \Sigma_L^*,\ c \in \Sigma_L,\ |x_1| = |x_2|$, and $w = x_1x_2)\}$. What is *chop*($\{\mathtt{a}^n\mathtt{b}^{2n} : n \geq 0\}$)?

> $\{\mathtt{a}^n\mathtt{b}^{2n\text{-}1} : n \geq 1$ and odd$\}$.

3) Are the following sets closed under the following operations? Prove your answer. If a set is not closed under the operation, what is its closure under the operation? Assume an alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.

a) FIN under complement.

> No, which we prove by counterexample. Let $L = \{\mathtt{a}\}$. L is finite. But $\neg L$ is infinite.

b) INF under complement.

> No, which we prove by counterexample. Let $L = \{\mathtt{a}, \mathtt{b}\}^*$. L is infinite. But $\neg L = \varnothing$ is infinite.

c) FIN under reverse.

> Yes. For any language L, there is a one-to-one correspondence between the elements of L and L^R, so the cardinalities of the two sets are the same.

d) INF under reverse.

> No, which we prove by counterexample. Let $L = \{\mathtt{a}\}$. L is finite. But $\neg L$ is infinite.

e) FIN under Kleene star.

> No, which we prove by counterexample. Let $L = \{\mathtt{a}\}$. L is finite. But L^* is infinite.

f) INF under Kleene star.

> Yes. For any language L, every string in L must also be in L^*. So L^* has at least as many elements as L does.

4) [Ben Wiedermann] Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. Let S be the set of all languages over Σ. Let f be a binary function defined as follows:

$$f: S \times S \rightarrow S$$
$$f(x, y) = x\,y$$

Answer each of the following questions and defend your answers:

a) Is f one-to-one?

> No, f is not one-to-one. For any language L, $L\,\varnothing = \varnothing$.

b) Is *f* onto?

Yes, *f* is onto. For any language L, $L\{\varepsilon\} = L$.

c) Is *f* commutative?

No. *f* is not commutative because concatenation is not commutative. Counterexample: {`a`}{`b`} = {`ab`}, but {`b`}{`a`} = { `ba`}.