Problem Set for Tutorial 4 — TDDD14/TDDD85

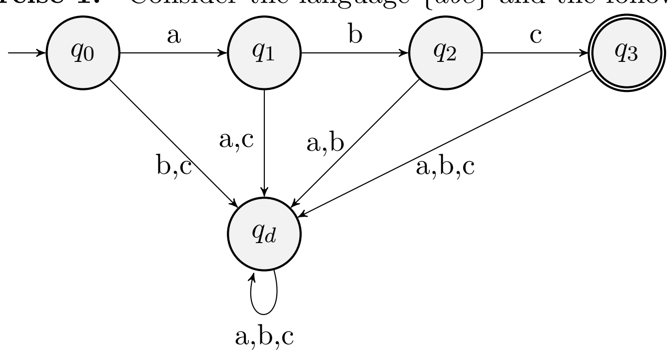
1 Pumping Lemma and Myhill-Nerode

We first recall the pumping lemma from the lecture notes.

Lemma 1. *(Pumping lemma) If L is a regular language, then there exists a positive integer p (the pumping length) such that every string $s \in L$, where $|s| \geq p$, can be partitioned into three pieces, $s = xyz$, such that the following conditions hold:*

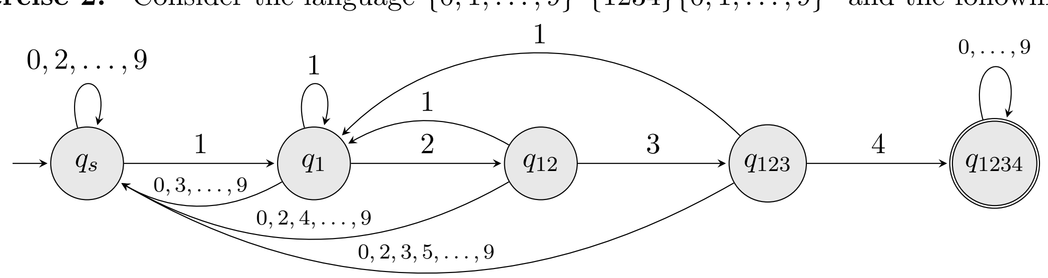
1. $|y| > 0$,
2. $|xy| \leq p$, *and*
3. *for each* $i \geq 0$, $xy^iz \in L$.

Exercise 1. Consider the language $\{abc\}$ and the following DFA.



First, give a possible value for the pumping length $p \geq 1$. Second, can the pumping lemma be used to construct a string $xy^iz \in L$ (for arbitrary $i \geq 0$)? Why, or why not?

Exercise 2. Consider the language $\{0, 1, \ldots, 9\}^*\{1234\}\{0, 1, \ldots, 9\}^*$ and the following DFA.



First, give a possible value for the pumping length $p \geq 1$. Second, can the pumping lemma be used to construct a string $xy^iz \in L$ (for arbitrary $i \geq 0$)? Why, or why not?

Exercise 3.

- Assume a person puts forth the following **incorrect** argument for proving that $L = \{0^n1^n \mid n \geq 1\}$ is not regular.

  1. I will prove that L is not regular by finding a string in the language which can be pumped so that the resulting string is not in L.
  2. Hence, define the string $s = 000111$ where the pumping length $p \leq 3$.
  3. Then, for any partitioning $s = xyz$ where $|xy| \leq p$ and every $i \geq 0$ the string xy^iz is not in L.
  4. Hence, L is not regular.

  What is/are the error(s) in the above claims?

- Assume that the person in the previous question is very stubborn and now claims to have yet another proof (sadly, also **incorrect**) of non-regularity of L.

  1. I will prove that L is not regular by finding a string in the language which can be pumped so that the resulting string is not in L.
  2. Hence, let $p > 0$ and pick a string s from $\{0^n1^n \mid n \geq 1\}$ of length at least p.
  3. Then, partition s into xyz where $|xy| \leq p$ and $xy = 0^n$. Then, for every $i \geq 2$ the string xy^iz is not in L.
  4. Hence, L is not regular.

  What is/are the error(s) in the above claims?

We now continue with proving non-regularity. First, recall that to prove that a language is *not* regular, one performs a proof by contradiction by assuming that the language is regular, and show that this assumption violates the pumping lemma by producing a string outside the language. Alternatively, one can use the "inverted pumping lemma" from the lecture where the conditions are already spelled out.

Lemma 2. *(Inverted pumping lemma) If there for each positive integer p (the pumping length) exists a string $s \in L$, $|s| \geq p$, such that for each partitioning $s = xyz$ where*

1. $|y| > 0$,
2. $|xy| \leq p$, *and*
3. *there exists* $i \geq 0$ *such that* $xy^iz \notin L$,

then L is not regular .

Exercise 4. Show, by using the (inverted) pumping lemma or the Myhill-Nerode theorem, that the following languages are *not* regular.

1. $L_1 = \{0^n1^n \mid n \geq 1\}$,
2. $L_2 = \{x \in \{0,1\}^* \mid x = x^R\}$ where x^R for a string x is x reversed.

Solutions

Solution to Exercise 1. A possible value for p is 5. But the pumping lemma is not useful for this language since there does not exist any string $s \in \{abc\}$ where $|s| \geq 5$. Hence, the pumping lemma is still technically true since the statement "for every string $s \in L$ where $|s| > p$..." becomes trivially true.

The example can be scaled further: the pumping lemma does not say anything interesting about finite languages (which are always regular). The "regular" structure of strings thus only becomes apparent when considering regular languages that are infinite.

Solution to Exercise 2. A possible value for p is 5 One could then e.g. choose the string $s = 9812340$. Via the pumping lemma we then know that there *exists* a partitioning $s = xyz$ where $|xy| \leq 5$, $|y| > 0$, and where $xy^i z$ is in the language for any $i \geq 0$. One such partitioning is $x = 9$, $y = 8$, $z = 12340$, another one is $x = \varepsilon, y = 98, z = 12340$. By pumping these strings we produce strings which, intuitively, are created by looping in the initial state q_s in the automaton.

Solution to Exercise 3.

- The most severe mistake is that it is never stated that $p \geq 1$ is arbitrary. If this had been made explicit in the first line of the proof then it would have been clear that we could not guarantee that the length of the string (000111) has length at least p.
- Here we make everything much more difficult than necessary by not making the choice of s explicit: it is only assumed that it is some string in the language of length at least p. The partitioning step is also wrong: in order to apply the inverted pumping lemma we need to prove that we can construct a string $xy^i z$ outside the language for *every* partitioning of s into xyz where $|xy| \leq p$, and $|y| > 0$. This is never established: it is incorrectly assumed that $xy = 0^n$ for some (undefined) n.

Solution to Exercise 4.

1. See the lecture notes for a complete solution.
2. Let $p \geq 1$. Then consider $s = 0^p 1 0^p$. Clearly, $s \in L_2$ and $|s| = 2p + 1 \geq p$. Let $s = xyz$ be an arbitrary partitioning where $|xy| \leq p$ and $|y| > 0$. This implies that $x = 0^k$, $y = 0^l$ $(k + l \leq p)$ and $z = 0^{p-k-l} 1 0^p$. Now consider $xy^0 z = xz = 0^{p-l} 1 0^p$. Since $|y| \geq 1 \Rightarrow l \geq 1$, we have $p - l < p$, i.e. $xy^0 z = xz \notin L_2$. We conclude that the language is not regular.

   Second, we use the Myhill-Nerode theorem. For each $k \geq 1$ consider the string $s_k = 0^k$. We claim that $[s_k] \neq [s_l]$ for any distinct $k \neq l$ which is sufficient to prove the claim. Indeed: if we let $z = 10^k$ then we see that $s_k z \in L_2$ but $s_l z \notin L_2$, and, hence, $[s_k] \neq [s_l]$.

2 Homomorphisms

Exercise 5. Let $\Sigma = \{0, 1\}$ and $\Gamma = \{a, b\}$ be two alphabets. Suppose $h\colon \Sigma \to \Gamma^*$ is a homomorphism such that $h(0) = aaab$ and $h(1) = bbba$.

1. What is $h(01)$?
2. What is $h(101)$?
3. Suppose $L \subseteq \Sigma^*$ is the language 001^*. Give a regular expression for $h(L)$.

Exercise 6. Recall that if L_1 and L_2 are languages where L_2 is not regular, and L_1 admits a homomorphism to L_2, then L_1 is not regular, either. Prove that $\{a^n b^n c^n \mid n \geq 1\}$ is not regular by giving a homomorphism to a suitable language already established not be non-regular. Hint: a very simple homomorphism is sufficient.

Solutions

Solution to Exercise 5.

1. $h(01) = h(0)h(1) = aaabbbba$.

2. $h(101) = h(1)h(01) = bbbaaaabbbba$

3. The point of this part of the exercise is to verify that we understand $h(L)$, and the strings included in this language can be concisely described by a regular expression. We first recall that $h(L) = \{h(s) \mid s \in L\}$. A possible regular expression for $h(L)$ is then $aaabaaab(bbba)^*$.

Solution to Exercise 6. We use the language $L_2 = \{0^n 1^n \mid n \geq 0\}$. It is then easy to verify that $h(a) = 0$, $h(b) = 1$, and $h(c) = \varepsilon$, satisfies the required properties. Naturally, one could easily prove that our language was not regular by using the pumping lemma or Myhill-Nerode theorem, but the homomorphism argument is substantially simpler and makes it possible to reuse what we have already established in a nice way.

3 Advanced and Exam Like Exercises

Exercise 7. Consider the sequence of all square integers from 2 upwards, i.e., the sequence $2, 4, 16, 256, \ldots$. Note that $2 = 2^1$ and $(2^i)^2 = 2^{2i}$; so the sequence can alternatively be written as $2^2, 2^4, 2^8, \ldots$. Now consider the language L that contains all numbers of this sequence in binary, i.e. L contains the sequence $10, 100, 10000, 100000000, \ldots$. Note that 2^i is represented by the string 10^i in binary.

Either prove that L is a regular language or use the (inverted) pumping lemma for regular languages (or the Myhill-Nerode theorem) to prove that L is not regular.

Exercise 8. Given an expression of the form $b_1 + \ldots + b_n$ where $b_i \in \{0, 1\}$, assume that we are interested in evaluating this (modulo 2) and checking whether the result is 0 or 1. For example, $0 + 1 + 1$ is 0 (modulo 2), and $1 + 1 + 1$ is 1 (modulo 2). For conciseness, we could represent each such expression as a string over $\{0, 1\}$, e.g., 011 and 111. Let L be the set of such strings over the alphabet $\{0, 1\}$ evaluating to 1. Is this language regular?

Solutions

Solution to Exercise 7. We suspect that the language is not regular since, in order to recognize a string 10^{2i} for $i \geq 0$ one has to count the number of zeroes and check that the total count is exactly $2i$, which we cannot accomplish with a fixed number of states. First, let us prove that the language is not regular by using the (inverted) pumping lemma. Hence, let $p \geq 1$ be arbitrary, and consider the string $s = 10^{2p}$, which is in L and represents the number 2^{2p}. Let $s = xyz$ be an arbitary partitioning where the length of xy is at most p and y is not empty. We have the following case analysis depending on whether x is empty or not.

1. $|x| = 0$ and $y = 10^k$ for some $0 \leq k < p$. Hence, $z = 0^{2p-l}$.
2. $x = 10^k$ for some $k \geq 0$ and $y = 0^l$ for some $1 + k + l \leq p$ (i.e., $k + l < p$).

Let us begin with the first case. Here, we easily see that $xy^0z = xz = z = 0^{2p-l} \notin L$, and we are done. For the second case, note that the string $xy^2z = 10^k0^{2l}0^{2p-k-l} = 10^{2p-k-l+k+2l} = 10^{2p+l}$. Since $k + l < p$ we conclude that $l < p$. But then $2p + l < 3p$, and $10^{2p+l} \notin L$.

The proof using the Myhill-Nerode theorem is similar. For each $i \geq 1$ define the string $s_i = 10^i$. Let $i \neq j$ and assume without loss of generality that $i < j$. Then $s_i0^i = 10^{2i} \in L$ but $s_j0^i = 10^{i+j} \notin L$ since $j < i + j < 2j$.

Solution to Exercise 8. Hint: this is easier than what one might think. How much memory does one need in order to compute addition modulo 2? How can this be modeled as a DFA?