Regular languages are those languages which are accepted by some deterministic finite automata (or a number of equivalent models for deterministic finite automata). They provide a limited, but relatively simple model of computation to analyze. In this exercise we will investigate some questions about regular languages. It will be useful to review regular languages before moving onto more complicated models of computation such as Turing machines as it is known that Turing machines with a constant amount of memory are equivalent to finite automata.

1. Let \( k \geq 1 \) be an integer. Define \( C_k \subset \{0, 1\}^\ast \) as the language consisting of all strings containing a 1 exactly \( k \) places from the end. Assume that \( C_k \) only contains strings of length at least \( k \).

(a) Design a deterministic finite automata that accepts \( C_2 \) and \( C_3 \). Can you generalize your construction for all \( C_k \)?

- Define a state \( q_s \) for any \( k \)-bit string \( s \). The transition function is \( \delta(q_s, a) = q_s' \) where \( s' \) is \( s \) with its first character removed and \( a \in \{0, 1\} \) appended to the end. The accepting states are those \( q_s \) with the first character of \( s \) = 1 and the initial state is \( q_0^k \). This construction describes a DFA for all \( C_k \). Intuitively, the automata keeps track of the last \( k \) characters read from the input string.

(b) Design a nondeterministic finite automata that accepts \( C_2 \) and \( C_3 \). Can you generalize your construction for all \( C_k \)?

- Define states \( q_0, \ldots, q_k \) where \( \delta(q_0, 0) = q_0, \delta(q_0, 1) = \{q_0, q_1\} \) and \( \delta(q_i, a) = q_{i+1} \) for \( 0 < i < k, a \in \{0, 1\} \) and \( \delta(q_k, a) = q_k \) for \( a \in \{0, 1\} \). This NFA will accept \( C_k \) provided that \( q_{k-1} \) is the unique accepting state, since the NFA guesses whether or not each 1 in the input string is the \( k^{th} \) last one from the construction.

(c) Prove that any deterministic finite automata accepting \( C_k \) must have at least \( 2^k \) states. (Hint: apply the “pigeonhole principle”.)

- Suppose \( Q \) is a DFA accepting \( C_k \) with fewer than \( 2^k \) states. Then there must be two different \( k \)-bit strings \( x, y \) leading to the same state \( q \) when \( Q \) reads them. In particular, there is a bit position \( 1 \leq d \leq k \), where \( x_d = 0, y_d = 1 \) and furthermore the \( Q \) will take \( x' = x1^{d-1}, y' = y1^{d-1} \) to the same state \( q' \). So \( Q \) accepts both strings \( x', y' \) or rejects both of them. But by construction \( y' \) should be accepted and \( x' \) should be rejected, so this is a contradiction and \( Q \) must have at least \( 2^k \) states.

2. Recall that pumping lemma is a tool that is used to show that a language \( L \subseteq \{0, 1\}^\ast \) is not a regular language. The lemma states that for any regular language \( L \), there is an non-negative
integer \( p \) such that for any string \( s \in L \) with \( |s| \geq p \), \( s \) can be decomposed into substrings \( s = xyz \) such that \( |y| > 0, |xy| \leq p \) and \( xy^iz \in L \) for any non-negative integer \( i \).

(a) Prove that the language \( L = \{0^k1^k|k \geq 0\} = \{01, 0011, 000111, \ldots \} \) is not regular using the pumping lemma.

- Assume \( L \) was regular so there is some pumping length \( p \) as in the lemma and choose \( s = 0^p1^p \). For any non-trivial decomposition \( s = xyz \) with \( |y| = i > 0 \) and \( |xy| < p \), we have \( xy^2z = 0^{p+i}1^p \notin L \), which contradicts the conditions of the pumping lemma.

(b) Prove that the language \( L = \{y^3|y \in \{0,1\}^*\} = \{000, 010101, 110110110, \ldots \} \) is not regular using the pumping lemma.

- Assume \( L \) was regular so there is some pumping length \( p \) as in the lemma and choose \( s = 0^p1^p0^p1^p0^p1^p \). For any non-trivial decomposition \( s = xyz \) with \( |y| = i > 0 \) and \( |xy| < p \), we have \( xy^2z = 0^{p+i}1^p0^p1^p0^p1^p \notin L \), which contradicts the conditions of the pumping lemma.

(c) Consider the language \( F = \{a^ib^jc^k|i,j,k \geq 0 \text{ and if } i = 1 \text{ then } j = k\} \)

i. Show that \( F \) satisfies the post-condition of the pumping lemma (i.e. there is some integer \( p \) for which a decomposition satisfying the conditions is possible).

- Let \( p = 2 \) and consider any string \( s \in L \). If \( s \) has less than or equal to one \( a \), let \( x = \epsilon, y = \) the first character of \( s \), \( z = \) the rest of \( s \). Otherwise, if \( s \) has at least two \( a \)s, let \( x = \epsilon, y = aa, z = \) the rest of \( s \) so that the pumping condition \( xy^iz \in L \) for all \( i \geq 0 \) is satisfied.

ii. Show that \( F \) is not regular.

- Let \( G = \{ab^jc^k|j,k \geq 0\} \). Note that \( G \) is regular. Then if \( F \) was regular, then \( F \cap G = \{ab^jc^j|j \geq 0\} \) would be regular. However, \( F \cap G \) is not regular, which can be proven using a pumping lemma type argument. So \( F \) is not regular.

\(^1\)\( y^i \) is string concatenation of \( y \) for \( i \) times and \( |x| \) is the length of \( x \)