Introduction to Language Theory and Compilation
Solutions
Session 1: Regular languages

Ex. 1.

1. - $1 \in \Sigma$ and $0 \in \Sigma$, thus $\{1\}$ and $\{0\}$ are both regular languages (RL).
   - The Kleene closure of a RL is also a RL, thus $\{1\}^*$ and $\{0\}^*$ are RL.
   - The concatenation of RL is a RL, thus $\{1\}^* \cdot \{0\} \cdot \{1\} \cdot \{0\}^*$ is a RL.
2. An odd binary number always ends with a 1.

   $\{1\}$ and $\{0\}$ are RL; $\{1\} \cup \{0\}$ is regular; $(\{1\} \cup \{0\})^*$ is regular; $(\{1\} \cup \{0\})^* \cdot \{1\}$ is regular.

Ex. 2.

1. **Show that any finite language is regular** (by *induction*:) *Idea of the proof*: L is finite, so there exists $n \in \mathbb{N}$ (the *size* of the language L) and n words $w_1, \ldots, w_n \in \Sigma^*$ such that $L = \{w_1, \ldots, w_n\}$. Thus, $L = \bigcup_{i=1}^{n} L_i$, where for each $i \in \{1, \ldots, n\}$, L_i is the singleton language containing only the word w_i: $L_i = \{w_i\}$. Moreover, for each i, there exists $n_i \in \mathbb{N}$ (the *length* of the word w_i) and n_i letters $c_1, \ldots, c_{n_i} \in \Sigma$ such that $w_i = c_1 \ldots c_{n_i}$, so for each i, $L_i = \{c_1 \ldots c_{n_i}\} = \{c_1\} \cdot \ldots \cdot \{c_{n_i}\} = \bullet_{j=1}^{n} c_j$. Since each $\{c_j\}$ is regular, L_i is also regular because it is a concatenation of regular languages. Thus, $L = \bigcup_{i=1}^{n} L_i$ is regular, as a finite union of regular languages.

   Note however that the use of "$\cdots$" is not very formal here, so the really formal way of writing such proof is by induction.

   First, let us show by induction on the length of the word l that for all $l \in \mathbb{N}$, any word $w \in \Sigma^l$ (Σ^l denotes the set of words of length l), $\{w\}$ is a regular language.

   - For $l = 0$, we have that $w = \varepsilon$, and by definition $\{\varepsilon\}$ is regular.
   - Although this is not needed for the induction, we also treat the case $l = 1$ because the case $l = 0$ might seem "pathological" for some of you: for $l = 1$, $w = a$ for some $a \in \Sigma$, so $L = \{a\}$ is regular by definition.
   - Now, assume that the property holds for some $l \in \mathbb{N}$, and let w be some word of length $l+1$. w can be decomposed into $w = w'a$ for some w' of length l and some $a \in \Sigma$. By the induction hypothesis, $\{w'\}$ is regular since w' is of length n, so $\{w\} = \{w'\} \cdot \{a\}$ is regular as a concatenation of two regular languages.

   Remark. Note that we could have shown using the same technique a more general result, namely that the l-th power of a regular language is regular, where we define $L^0 = \{\varepsilon\}$ and for all $l \in \mathbb{N}$, $L^{l+1} = L^l \cdot L$. Then, we could have deduced that Σ^l, the set of all words of length l, is regular, because Σ is regular (as a finite union of regular languages $\{a\}$).

   Now, let us show by induction on n the size of L that for all $n \in \mathbb{N}$, for all finite language L of size n, L is regular:

   - Again, we can start at $n = 0$: then $L = \emptyset$, which is regular by definition.
   - Again, although this is not needed for the mathematical correctness of the proof, we treat the case $n = 1$: then, L contains a single word $w \in \Sigma^*$: $L = \{w\}$, so L is regular as we showed earlier.
   - Now, assume the property holds for some $n \in \mathbb{N}$, and let L be a language of size $n+1$. $L = L' \cup \{w\}$ for some L of size n and some $w \in \Sigma^*$. By the induction hypothesis, we known that L' is regular since it is a language of size n. Now, $\{w\}$ is regular, as shown earlier, so L is regular, as a union of two regular languages.

Remark. An entirely different way of showing that any finite language L is regular would have been to build a finite automaton recognising L, and then use Kleene's theorem. This is left as an exercise[1].

2. **Show that the language $L = \{0^n 1^n \mid n \in \mathbb{N}\}$ is not regular** (by *contradiction*:)
Assume, towards contradiction that L is a regular language. By Kleene's theorem, we know that there exists a finite *(possibly non-deterministic)* automaton $A = \langle Q, \Sigma, \delta, q_0, F \rangle$ that accepts L. By definition, the state set Q is finite. Let m be its size. Now, observe that the language L is *infinite*, as for each natural number $n \in \mathbb{N}$, there is a corresponding word $0^n 1^n$ in L. For instance, consider the word $0^{2m} 1^{2m}$. Clearly ($2m \in \mathbb{N}$ and $2m = 2m$!), this word is in L. Thus, there exists an accepting run of the automaton A on the word $0^{2m} 1^{2m}$. This run is of the following form:

$$q_0 \xrightarrow{0} q_1 \xrightarrow{0} q_2 \xrightarrow{0} \dots \xrightarrow{0} q_{2m} \xrightarrow{1} q_{2m+1} \xrightarrow{1} \dots \xrightarrow{1} q_{4m}$$

where $q_{4m} \in F$ (and where $q_i \xrightarrow{0} q_{i+1}$ stands for "A moves from state q_i to state q_{i+1} by reading 0"). Let us look at the run prefix $q_0 \xrightarrow{0} q_1 \xrightarrow{0} q_2 \xrightarrow{0} \dots \xrightarrow{0} q_{2m}$ that goes through the first half of the word $0^{2m} 1^{2m}$. This run prefix is composed of $2m+1$ states. Recall that A has only m different states. Thus, by the pigeonhole principle, there exist $q \in Q$, $i, j \in \mathbb{N}$ such that $0 \le i < j \le 2m$ and $q_i = q_j = q$. That is, the run prefix is of the following form:

$$q_0 \xrightarrow{0} q_1 \xrightarrow{0} q_2 \xrightarrow{0} \dots \xrightarrow{0} q_i = q \xrightarrow{0} \dots \xrightarrow{0} q_j = q \xrightarrow{0} \dots \xrightarrow{0} q_{2m} \xrightarrow{1} q_{2m+1} \xrightarrow{1} \dots \xrightarrow{1} q_{2m}$$

This means that the path $q_i = q \xrightarrow{0} \dots \xrightarrow{0} q_j = q$ is actually a *loop*, and, furthermore, that we can repeat it (or delete it) and still obtain an accepting run of A.

Indeed, if this:

$$q_0 \xrightarrow{0} \dots \xrightarrow{0} q_i = q \xrightarrow{0} \dots \xrightarrow{0} q_j = q \xrightarrow{0} \dots \xrightarrow{0} q_{2m} \xrightarrow{1} q_{2m+1} \xrightarrow{1} \dots \xrightarrow{1} q_{2m} \xrightarrow{1} q_{2m+1} \xrightarrow{1} \dots \xrightarrow{1} q_{4m}$$

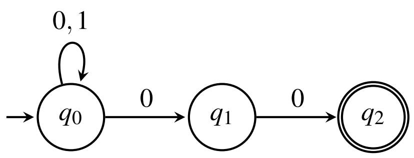
is an accepting run, then the following one, where the loop is repeated twice,

$$q_0 \xrightarrow{0} \dots \xrightarrow{0} q_i = q \xrightarrow{0} \dots \xrightarrow{0} q_j = q = q_i \xrightarrow{0} \dots \xrightarrow{0} q_j = q \xrightarrow{0} \dots \xrightarrow{0} q_{2m} \xrightarrow{1} q_{2m+1} \xrightarrow{1} \dots \xrightarrow{1} q_{2m} \xrightarrow{1} q_{2m+1} \xrightarrow{1} \dots \xrightarrow{1} q_{4m}$$

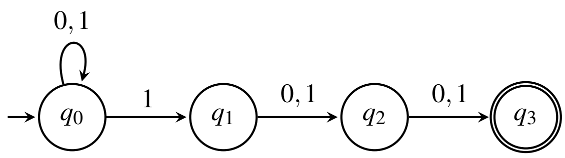
is also an accepting run, as it fully respects the transition function of A, and ends in q_{4m} which is an accepting state. Let $k = j - i$ and $\ell = 2m - j$. Observe now which word is accepted by this run: $0^i 0^k 0^k 0^\ell 1^{2m}$. Recall that $i + k + \ell = 2m$, thus $i + 2k + \ell > 2m$ as $k > 0$. Thus, the word $0^i 0^k 0^k 0^\ell 1^{2m}$ is *not* in the language L. This is a contradiction with the assumption that L was accepted by A, which means, in particular, that any word not in L has to be rejected by A. This shows that there cannot exist a finite automaton accepting L. Hence, L is not regular.

Remark. Note that by repeating the loop j times, or by deleting it, we can also show that for all $j \in \mathbb{N}$ (even $j = 0$), $0^i 0^{j \times k} 0^l 1^{2m}$ is accepted by A, so A *wrongly* accepts an infinite number of words.

Ex. 3.

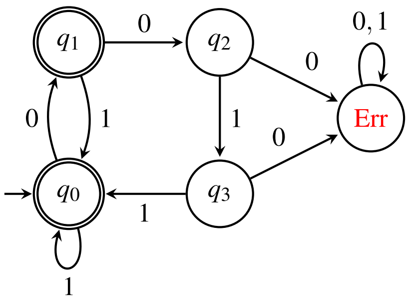


1 The set of strings ending with 00

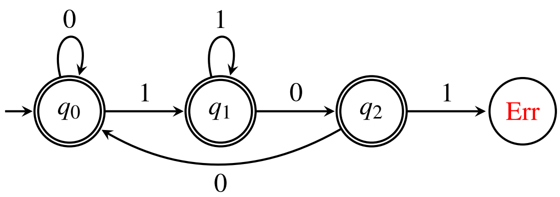


2 The set of strings whose 3rd symbol, counted from the end of the string, is a 1

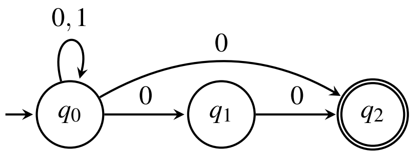
[1]Do not hesitate to ask us if you want advice on this.



3 The set of strings where each pair of zeroes is directely followed by a pair of ones



4 The set of strings that do not contain the sequence 101



5 The set of binary numbers divisible by 4

Ex. 4.