Finite Automata and Regular Languages (part V)

Prof. Dan A. Simovici

UMB

Theorem

The language $L(\mathcal{M})$ accepted by a dfa $\mathcal{M}$ can be constructed starting from finite languages by using union, product, and $*$ (also known as Kleene closure).

Proof

Let $\mathcal{M} = (A, \{q_0, \ldots, q_{n-1}\}, \delta, q_0, F)$ be a dfa that has n states.
Define the language R_{ij}^k to consist of those words $x \in A^*$ that take $\mathcal{M}$ from state q_i to state q_j without passing through any state q_ℓ with $\ell \geq k$.
R_{ij}^k comprises those words $x \in A^*$ such that $\delta^*(q_i, x) = q_j$, and for every proper prefix u of x, the automaton is in an intermediate state $\delta^*(q_i, u) \in \{q_0, \ldots, q_{k-1}\}$.

Proof cont'd

R_{ij}^0 is the finite language

$$R_{ij}^0 = \begin{cases} \{a \in A \mid \delta(q_i, a) = q_j\} & \text{if } i \neq j, \\ \{\lambda\} \cup \{a \in A \mid \delta(q_i, a) = q_i\} & \text{if } i = j. \end{cases}$$

Note that no intermediate states are allowed ($k = 0$).

Proof cont'd

To reach q_j starting from q_i, allowing only $q_0, \ldots, q_{k-1}$ as intermediate states presents two choices:

- either the automaton avoids q_{k-1} entirely (which corresponds to a word in R_{ij}^{k-1}), or
- the automaton goes from q_i to the first use of q_{k-1} (along a string in R_{ik-1}^{k-1}); it then may revisit q_{k-1} zero or more times (each time along a string in R_{k-1k-1}^{k-1}), and finally, it goes from the last use of q_{k-1} to q_j (along a string in R_{k-1j}^{k-1}).

This implies

$$R_{ij}^{k} = R_{ij}^{k-1} \cup R_{ik-1}^{k-1}(R_{k-1k-1}^{k-1})^{*} R_{k-1j}^{k-1}. \tag{1}$$

This can be regarded as a recursive definition of the set R_{ij}^{k}.

In formula

$$R_{ij}^{k} = R_{ij}^{k-1} \cup R_{ik-1}^{k-1}(R_{k-1k-1}^{k-1})^{*} R_{k-1j}^{k-1},$$

the following intermediate states are permitted:

- R_{ij}^{k} : $q_0, \ldots, q_{k-1}$ between q_i and q_j;
- R_{ij}^{k-1} : $q_0, \ldots, q_{k-2}$ between q_i and q_j;
- R_{ik-1}^{k-1} : $q_0, \ldots, q_{k-2}$ between q_i and q_{k-1}
- $R_{k-1\,k-1}^{k-1}$: $q_0, \ldots, q_{k-2}$ from q_{k-1} back to q_{k-1};
- R_{k-1j}^{k-1} : $q_0, \ldots, q_{k-2}$ from q_{k-1} to q_j.

Proof cont'd

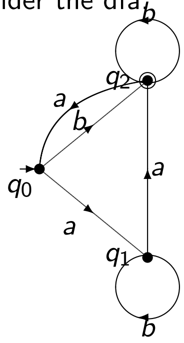
If the set of final states is $F = \{q_{i_0}, \ldots, q_{i_{m-1}}\}$, then the language accepted by $\mathcal{M}$ is

$$L(\mathcal{M}) = R^n_{0i_0} \cup \cdots \cup R^n_{0i_{m-1}},$$

since every state is allowed as an intermediate state. Thus, $L(\mathcal{M})$ can be indeed constructed starting from the finite languages R^0_{ij}, using union, product and Kleene closure.

An Example

Consider the dfa:



We have:

$$\begin{aligned} R^1_{ij} &= R^0_{ij} \cup R^0_{i0}(R^0_{00})^* R^0_{0j}, \\ R^2_{ij} &= R^1_{ij} \cup R^1_{i1}(R^1_{11})^* R^1_{1j}, \\ R^3_{ij} &= R^2_{ij} \cup R^2_{i2}(R^2_{22})^* R^2_{2j}. \end{aligned}$$

We have:

$$\begin{aligned}
R_{ij}^{1} &= R_{ij}^{0} \cup R_{i0}^{0}(R_{00}^{0})^{*}R_{0j}^{0},\\
R_{ij}^{2} &= R_{ij}^{1} \cup R_{i1}^{1}(R_{11}^{1})^{*}R_{1j}^{1},\\
R_{ij}^{3} &= R_{ij}^{2} \cup R_{i2}^{2}(R_{22}^{2})^{*}R_{2j}^{2}.
\end{aligned}$$

Example (cont'd)

$$\begin{array}{lll} R_{00}^0 = \{\lambda\}, & R_{01}^0 = a & R_{02}^0 = b \\ R_{10}^0 = \emptyset & R_{11}^0 = \{\lambda, b\} & R_{12}^0 = a \\ R_{20}^0 = a & R_{21}^0 = \emptyset & R_{22}^0 = \{\lambda, b\}. \end{array}$$

$$\begin{aligned}
R^1_{00} &= R^0_{00} \cup R^0_{00}(R^0_{00})^* R^0_{00}, \\
&= \lambda, \\
R^1_{01} &= R^0_{01} \cup R^0_{00}(R^0_{00})^* R^0_{01}, \\
&= a, \\
R^1_{02} &= R^0_{02} \cup R^0_{00}(R^0_{00})^* R^0_{02}, \\
&= b,
\end{aligned}$$

$$\begin{aligned}
R_{10}^{1} &= R_{10}^{0} \cup R_{10}^{0}(R_{00}^{0})^{*}R_{00}^{0}, \\
&= \emptyset, \\
R_{11}^{1} &= R_{11}^{0} \cup R_{10}^{0}(R_{00}^{0})^{*}R_{01}^{0}, \\
&= \{\lambda, b\} \\
R_{12}^{1} &= R_{12}^{0} \cup R_{10}^{0}(R_{00}^{0})^{*}R_{02}^{0}, \\
&= a,
\end{aligned}$$

$$\begin{aligned}
R_{20}^{1} &= R_{20}^{0} \cup R_{20}^{0}(R_{00}^{0})^{*}R_{00}^{0}, \\
&= a, \\
R_{21}^{1} &= R_{21}^{0} \cup R_{20}^{0}(R_{00}^{0})^{*}R_{01}^{0}, \\
&= aa, \\
R_{22}^{1} &= R_{22}^{0} \cup R_{20}^{0}(R_{00}^{0})^{*}R_{02}^{0}, \\
&= \{\lambda, b, ab\}.
\end{aligned}$$

Note that we used the fact that $\emptyset^{*} = \{\lambda\}$.

$$\begin{aligned}
R_{00}^2 &= R_{00}^1 \cup R_{01}^1(R_{11}^1)^* R_{10}^1, \\
&= \emptyset, \\
R_{01}^2 &= R_{01}^1 \cup R_{01}^1(R_{11}^1)^* R_{11}^1, \\
&= a \cup a\{\lambda, b\}^+ = ab^*, \\
R_{02}^2 &= R_{02}^1 \cup R_{01}^1(R_{11}^1)^* R_{12}^1, \\
&= b \cup a\{\lambda, b\}^* a = b \cup ab^* a,
\end{aligned}$$

$$\begin{aligned}
R^2_{10} &= R^1_{10} \cup R^1_{11}(R^1_{11})^* R^1_{10}, \\
&= \emptyset, \\
R^2_{11} &= R^1_{11} \cup R^1_{11}(R^1_{11})^* R^1_{11}, \\
&= b^*, \\
R^2_{12} &= R^1_{12} \cup R^1_{11}(R^1_{11})^* R^1_{12}, \\
&= b^* a,
\end{aligned}$$

$$\begin{aligned}
R_{20}^{2} &= R_{20}^{1} \cup R_{21}^{1}(R_{11}^{1})^{*}R_{10}^{1}, \\
&= a, \\
R_{21}^{2} &= R_{21}^{1} \cup R_{21}^{1}(R_{11}^{1})^{*}R_{11}^{1}, \\
&= aa \cup aab^{*} = aab^{*}, \\
R_{22}^{2} &= R_{22}^{1} \cup R_{21}^{1}(R_{11}^{1})^{*}R_{12}^{1}, \\
&= \{\lambda, b, ab\} \cup aab^{*}a,
\end{aligned}$$

Finally,

$$\begin{aligned} R_{02}^3 &= R_{02}^2 \cup R_{02}^2 (R_{22}^2)^* R_{22}^2. \\ &= (b \cup ab^* a)\{\lambda, b, ab\} \cup aab^* a)^+. \end{aligned}$$

Theorem

Kleene's Theorem *The class $\mathcal{R}$ of regular languages is the least class of languages that contains the class of finite languages and is closed with respect to union, product and Kleene closure.*

Proof.

Closure properties previously discussed imply that the class $\mathcal{R}$ contains the class of finite languages and is closed with respect to union, product, and Kleene closure, respectively. □

Proof cont'd

Let $\mathcal{R}'$ be an arbitrary class of languages that contains the class of finite languages and is closed with respect to union, product and Kleene closure. If $L \in \mathcal{R}$, the previous theorem implies that $L \in \mathcal{R}'$, so $\mathcal{R} \subseteq \mathcal{R}'$.

The main focus of this section is a powerful tool for proving that certain languages are not regular. Although traditionally named the *Pumping Lemma*, we state it as a theorem to reflect its importance.

Theorem

If L is a regular language, then there exists a number $n_0 \in \mathbb{P}$ such that if $x \in L$ and $|x| \geq n_0$, then x can be written as $x = uvw$ such that $v \neq \lambda$, $|uv| \leq n_0$, and $uv^n w \in L$ for every $n \in \mathbb{N}$.

Proof

Since L is a regular language there exists a deterministic finite automaton $\mathcal{M} = (A, Q, \delta, q_0, F)$ such that $L = L(\mathcal{M})$. Choose $n_0 = |Q|$, and let $x = a_0 \dots a_{\ell-1}$ be a word of length ℓ, where $\ell \geq n_0$.
Consider the sequence of states $(q_0, q_1, \dots, q_\ell)$, where $q_i = \delta(q_{i-1}, a_{i-1})$ for $1 \leq i \leq \ell$. This sequence is of length $\ell + 1 > n_0$, so at least two of these states must coincide. Pick k to be the position of the first repeated state and let q_j be its first occurrence.

Proof cont'd

Let $u = a_0 a_1 \cdots a_{j-1}$, $v = a_j \cdots a_{k-1}$, and $w = a_k \cdots a_{\ell-1}$. Then, $q_j = \delta^*(q_0, u) = \delta^*(q_0, uv) = q_k$. Consequently, $\delta^*(q_j, v) = q_k = q_j$. Since $\delta^*(q_k, w) = q_\ell \in F$, we have $\delta^*(q_j, w) = q_\ell \in F$, so $\delta^*(q_0, uw) = q_\ell \in F$, and $uw \in L$.
By a simple induction, we can prove that $\delta^*(q_j, v^n) = q_j$ for all $n \in \mathbb{N}$, so $\delta^*(q_0, uv^n w) = \delta^*(q_j, v^n w) = \delta^*(q_j, w) = q_\ell \in F$. Hence, $uv^n w \in L$ for all $n \in \mathbb{N}$.
By the selection of k, $0 \leq j < k \leq n_0$, which implies $|v| = k - j > 0$, that is $v \neq \lambda$, and also that $|uv| \leq n_0$.

We refer to a number n_0 whose existence is established by the Pumping Lemma as a *pumping threshold* for the language L.

Decidable Problems

One of most interesting areas of study of twentieth century mathematics is the area of “decidability”. In this section, we approach decidability informally. Specifically, we say that a question is decidable when there is an effective procedure that always accurately answers the question, either yes or no. The notion of “effective procedure” may be made explicit using computer programs with unlimited resources or other computing models.

Example

If A is an alphabet, $L \subseteq A^*$ is a regular language, and $x \in A^*$ is a word, then it is decidable whether $x \in L$. Indeed, in order to decide this problem it suffices to write the word x on the input tape of a dfa $\mathcal{M}$ that accepts L and determine if the state q of $\mathcal{M}$ is a final state. If this is the case, $x \in L$; otherwise $x \notin L$.

Theorem

It is decidable whether a regular language is empty.

Proof.

Let L be a regular language, and let n_0 be a pumping threshold for L. We claim that if $L \neq \emptyset$, then there is a word z in L such that $|z| < n_0$. Clearly, if such a word exists in L, then $L \neq \emptyset$. Suppose that $L \neq \emptyset$. Then, let x be a word in L having minimal length. If $|x| \geq n_0$, then, by the Pumping Lemma, we can write $x = uvw$ with $|v| > 0$ and $uv^n w \in L$ for every $n \in \mathbb{N}$. Taking $n = 0$, we obtain that $uw \in L$. The inequality $|uw| < |x|$, contradicts the minimality of x. Therefore, L must contain a word shorter than n_0.
The previous argument shows that in order to decide whether a regular language L is empty it is sufficient to test whether any word $x \in A^*$ with $|x| < n_0$ belongs to L. This implies the decidability result. $\square$

Theorem

It is decidable whether a regular language is infinite.

Proof.

Let L be a regular language. We prove that L is infinite if and only if it contains a word x such that $n_0 \leq |x| < 2n_0$, where n_0 a pumping threshold for L.
Suppose that the language L is infinite. Then, there exists a word $y \in L$ such that $|y| \geq 2n_0$. Suppose that y is one of the shortest such words. By the Pumping Lemma we can write $y = uvw$ with $v \neq \lambda$, $|v| \leq n_0$ and $uv^n w \in L$ for every $n \in \mathbb{N}$. Therefore, taking $n = 0$, we obtain $y' = uw \in L$. Note that $n_0 \leq |y'| = |y| - |v| < |y|$. Since y was supposed to be a shortest word in L such that $|y| \geq 2n_0$ we obtain $n_0 \leq |y'| < 2n_0$.
Conversely, if L contains a word y with $n_0 \leq |y| < 2n_0$ it is obvious, by the Pumping Lemma, that L is infinite. Therefore, in order to determine if L is infinite it is sufficient to examine the finite set $L_f = \{x \in L \mid n_0 \leq |x| < 2n_0\}$. L is infinite if and only if $L_f \neq \emptyset$. □

Example

The language $L = \{a^m b^m \mid m \in \mathbb{N}\}$ is not regular.

To justify this claim, suppose that L is regular, and let n_0 be a pumping threshold for L. Choose $m \in \mathbb{N}$ such that $m \geq n_0$. Then, $x = a^m b^m$ can be factored as $x = uvw$ such that $1 \leq |uv| \leq n_0$ and $uv^n w \in L$ for every $n \in \mathbb{N}$. This implies that every symbol of v is equal to a, since uv is a prefix of a^m. So, $u = a^k$, $v = a^\ell$, and $w = a^h b^m$, where $k + \ell + h = m$, and $\ell \geq 1$. By "pumping" v we obtain $uv^2 w = a^k a^{2\ell} a^h b^m = a^{m+\ell} b^m \in L$, which contradicts the definition of L.

Example

The language $L = \{a^p \mid p \text{ is prime }\}$ is not regular.

Suppose that L were regular and that n_0 is a pumping threshold for L. If $x \in L$ and $|x| \geq n_0$, then x can be written as $x = uvw$ such that $v \neq \lambda$, $|uv| \leq n_0$, and $uv^n w \in L$ for every $n \in \mathbb{N}$. Let $k = |v| \geq 1$. We have $p = h + k$ such that $h + nk$ is prime for every value of $n \in \mathbb{N}$. If $h = 0$ this is a clear contradiction. If $h = 1$ choose, for example, $n = 6k + 5$. This would imply $h + nk = 1 + (6k + 5)k = (3k + 1)(2k + 1)$, which is not prime. Otherwise, if $h > 1$, choose $n = h$. This implies $a^{h(1+k)} \in L$, which contradicts the definition of L since $h(1 + k)$ is never a prime.

Example

The language $L = \{a^p b^q \mid p, q \in \mathbb{N} \text{ and } p \leq q\}$ is not regular.
Suppose that L were regular, and let n_0 be a pumping threshold for L. Choose $p > n_0$. Then, if $x = a^p b^q \in L$ with $p \leq q$, x can be factored as $x = uvw$ with $|uv| \leq n_0$ such that $uv^n w \in L$ for every $n \in \mathbb{N}$. By the choice for p, the words u and v consist only of as, so we can write $u = a^k$, $v = a^\ell$ and $w = a^h b^q$, where $k + \ell + h = p$, and $\ell \geq 1$. Thus, $uv^n w = a^{k+n\ell+h} b^q = a^{p+(n-1)\ell} b^q \in L$ for every $n \in \mathbb{N}$. By choosing n such that $p + (n-1)\ell > q$, that is $n > \frac{q-p}{\ell} + 1$ we obtain a word that violates the definition of L, which requires every word to contain more bs than as.

Example

Consider the language $L = \{a^p b^{2q} a^p \mid p, q \in \mathbb{N}, q \geq 1\}$.
We claim that L is not regular. Suppose that L were regular and let $x = a^p b^{2q} a^p \in L$ such that $p \geq n_0$, where n_0 is a pumping threshold for L. Consider the factorization $x = uvw$ whose existence follows from the Pumping Lemma, where $|uv| \leq n_0$, such that $uv^n w \in L$ for every $n \in \mathbb{N}$. Both u and v consist of a symbols, so any kind of pumping of v alters the balance between the p a symbols located at the beginning of the word and the last p a symbols. Therefore, L is not regular.

The nonregularity of

$$L = \{a^p b^{2q} a^p \mid p, q \in \mathbb{N}, q \geq 1\}.$$

implies that the language $K = \{xx^R \mid x \in \{a, b\}^*\}$ is not regular. Indeed, note that $K \cap \{a\}^*\{b\}^+\{a\}^* = L$, so the regularity of K would imply the regularity of L.

Example

The language $L = \{a^p b^q a^p b^q \mid p, q \in \mathbb{P}\}$ is not regular. This can be easily shown using an argument similar to the one used previously. Hence, since

$$\{xx \mid x \in \{a,b\}^*\} \cap \{a\}^*\{b\}^*\{a\}^*\{b\}^* = L$$

we can conclude that the language $\{xx \mid x \in \{a,b\}^*\}$ is not regular.