Lecture 4

More on Regular Sets

Here is another example of a regular set that is a little harder than the example given last time. Consider the set

$$\{x \in \{0,1\}^* \mid x \text{ represents a multiple of three in binary}\} \qquad (4.1)$$

(leading zeros permitted, ϵ represents the number 0). For example, the following binary strings represent multiples of three and should be accepted:

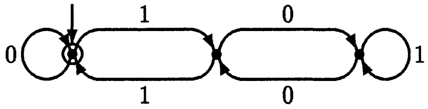
Binary	*Decimal equivalent*
0	0
11	3
110	6
1001	9
1100	12
1111	15
10010	18
⋮	⋮

Strings not representing multiples of three should be rejected. Here is an automaton accepting the set (4.1):

		0	1
→	$0F$	**0**	**1**
	1	**2**	**0**
	2	**1**	**2**

D. C. Kozen, *Automata and Computability*
© Springer Science+Business Media, Inc. 1997

The states **0**, **1**, **2** are written in boldface to distinguish them from the input symbols 0, 1.



In the diagram, the states are **0**, **1**, **2** from left to right. We prove that this automaton accepts exactly the set (4.1) by induction on the length of the input string. First we associate a meaning to each state:

if the number represented by the string scanned so far is[1]	*then the machine will be in state*
0 mod 3	**0**
1 mod 3	**1**
2 mod 3	**2**

Let $\#x$ denote the number represented by string x in binary. For example,

$$\begin{aligned} \#\epsilon &= 0, \\ \#0 &= 0, \\ \#11 &= 3, \\ \#100 &= 4, \end{aligned}$$

and so on. Formally, we want to show that for any string x in $\{0,1\}^*$,

$$\begin{aligned} \widehat{\delta}(\mathbf{0},x) &= \mathbf{0} \text{ iff } \#x \equiv 0 \bmod 3, \\ \widehat{\delta}(\mathbf{0},x) &= \mathbf{1} \text{ iff } \#x \equiv 1 \bmod 3, \\ \widehat{\delta}(\mathbf{0},x) &= \mathbf{2} \text{ iff } \#x \equiv 2 \bmod 3, \end{aligned} \tag{4.2}$$

or in short,

$$\widehat{\delta}(\mathbf{0},x) = \#x \bmod 3. \tag{4.3}$$

This will be our induction hypothesis. The final result we want, namely (4.2), is a weaker consequence of (4.3), but we need the more general statement (4.3) for the induction hypothesis.

We have by elementary number theory that

$$\#(x0) = 2(\#x) + 0,$$

[1] Here $a \bmod n$ denotes the remainder when dividing a by n using ordinary integer division. We also write $a \equiv b \bmod n$ (read: a is congruent to b modulo n) to mean that a and b have the same remainder when divided by n; in other words, that n divides $b - a$. Note that $a \equiv b \bmod n$ should be parsed $(a \equiv b) \bmod n$, and that in general $a \equiv b \bmod n$ and $a = b \bmod n$ mean different things. For example, $7 \equiv 2 \bmod 5$ but not $7 = 2 \bmod 5$.

$$\#(x1) = 2(\#x) + 1,$$

or in short,

$$\#(xc) = 2(\#x) + c \tag{4.4}$$

for $c \in \{0,1\}$. From the machine above, we see that for any state $q \in \{\mathbf{0},\mathbf{1},\mathbf{2}\}$ and input symbol $c \in \{0,1\}$,

$$\delta(q,c) = (2q + c) \bmod 3. \tag{4.5}$$

This can be verified by checking all six cases corresponding to possible choices of q and c. (In fact, (4.5) would have been a great way to *define* the transition function formally—then we wouldn't have had to prove it!) Now we use the inductive definition of $\widehat{\delta}$ to show (4.3) by induction on $|x|$.

Basis

For $x = \epsilon$,

$$\begin{aligned} \widehat{\delta}(\mathbf{0},\epsilon) &= \mathbf{0} && \text{by definition of } \widehat{\delta} \\ &= \#\epsilon && \text{since } \#\epsilon = 0 \\ &= \#\epsilon \bmod 3. \end{aligned}$$

Induction step

Assuming that (4.3) is true for $x \in \{0,1\}^*$, we show that it is true for xc, where $c \in \{0,1\}$.

$$\begin{aligned} \widehat{\delta}(\mathbf{0},xc) &= \delta(\widehat{\delta}(\mathbf{0},x),c) && \text{definition of } \widehat{\delta} \\ &= \delta(\#x \bmod 3, c) && \text{induction hypothesis} \\ &= (2(\#x \bmod 3) + c) \bmod 3 && \text{by (4.5)} \\ &= (2(\#x) + c) \bmod 3 && \text{elementary number theory} \\ &= \#xc \bmod 3 && \text{by (4.4).} \end{aligned}$$

Note that each step has its reason. We used the definition of δ, which is specific to this automaton; the definition of $\widehat{\delta}$ from δ, which is the same for all automata; and elementary properties of numbers and strings.

Some Closure Properties of Regular Sets

For $A, B \subseteq \Sigma^*$, recall the following definitions:

$$\begin{aligned} A \cup B &= \{x \mid x \in A \text{ or } x \in B\} && \text{union} \\ A \cap B &= \{x \mid x \in A \text{ and } x \in B\} && \text{intersection} \\ {\sim}A &= \{x \in \Sigma^* \mid x \notin A\} && \text{complement} \end{aligned}$$

$$\begin{aligned} AB &= \{xy \mid x \in A \text{ and } y \in B\} && \text{concatenation} \\ A^* &= \{x_1x_2\cdots x_n \mid n \geq 0 \text{ and } x_i \in A,\ 1 \leq i \leq n\} \\ &= A^0 \cup A^1 \cup A^2 \cup A^3 \cup \cdots && \text{asterate.} \end{aligned}$$

Do not confuse set concatenation with string concatenation. Sometimes $\sim A$ is written $\Sigma^* - A$.

We show below that if A and B are regular, then so are $A \cup B$, $A \cap B$, and $\sim A$. We'll show later that AB and A^* are also regular.

The Product Construction

Assume that A and B are regular. Then there are automata

$$\begin{aligned} M_1 &= (Q_1,\ \Sigma,\ \delta_1,\ s_1,\ F_1), \\ M_2 &= (Q_2,\ \Sigma,\ \delta_2,\ s_2,\ F_2) \end{aligned}$$

with $L(M_1) = A$ and $L(M_2) = B$. To show that $A \cap B$ is regular, we will build an automaton M_3 such that $L(M_3) = A \cap B$.

Intuitively, M_3 will have the states of M_1 and M_2 encoded somehow in its states. On input $x \in \Sigma^*$, it will simulate M_1 and M_2 simultaneously on x, accepting iff both M_1 and M_2 would accept. Think about putting a pebble down on the start state of M_1 and another on the start state of M_2. As the input symbols come in, move both pebbles according to the rules of each machine. Accept if both pebbles occupy accept states in their respective machines when the end of the input string is reached.

Formally, let

$$M_3 = (Q_3,\ \Sigma,\ \delta_3,\ s_3,\ F_3),$$

where

$$\begin{aligned} Q_3 &= Q_1 \times Q_2 = \{(p,q) \mid p \in Q_1 \text{ and } q \in Q_2\}, \\ F_3 &= F_1 \times F_2 = \{(p,q) \mid p \in F_1 \text{ and } q \in F_2\}, \\ s_3 &= (s_1, s_2), \end{aligned}$$

and let

$$\delta_3 : Q_3 \times \Sigma \rightarrow Q_3$$

be the transition function defined by

$$\delta_3((p,q),a) = (\delta_1(p,a), \delta_2(q,a)).$$

The automaton M_3 is called the *product* of M_1 and M_2. A state (p,q) of M_3 encodes a configuration of pebbles on M_1 and M_2.

Recall the inductive definition (3.1) and (3.2) of the extended transition function $\widehat{\delta}$ from Lecture 2. Applied to δ_3, this gives

$$\begin{aligned}
&\widehat{\delta}_3((p,q),\epsilon) = (p,q),\\
&\widehat{\delta}_3((p,q),xa) = \delta_3(\widehat{\delta}_3((p,q),x),a).
\end{aligned}$$

Lemma 4.1 *For all $x \in \Sigma^*$,*

$$\widehat{\delta}_3((p,q),x) = (\widehat{\delta}_1(p,x),\widehat{\delta}_2(q,x)).$$

Proof. By induction on $|x|$.

Basis

For $x = \epsilon$,

$$\widehat{\delta}_3((p,q),\epsilon) = (p,q) = (\widehat{\delta}_1(p,\epsilon),\widehat{\delta}_2(q,\epsilon)).$$

Induction step

Assuming the lemma holds for $x \in \Sigma^*$, we show that it holds for xa, where $a \in \Sigma$.

$$\begin{aligned}
&\widehat{\delta}_3((p,q),xa) &&\\
&= \delta_3(\widehat{\delta}_3((p,q),x),a) && \text{definition of } \widehat{\delta}_3\\
&= \delta_3((\widehat{\delta}_1(p,x),\widehat{\delta}_2(q,x)),a) && \text{induction hypothesis}\\
&= (\delta_1(\widehat{\delta}_1(p,x),a),\delta_2(\widehat{\delta}_2(q,x),a)) && \text{definition of } \delta_3\\
&= (\widehat{\delta}_1(p,xa),\widehat{\delta}_2(q,xa)) && \text{definition of } \widehat{\delta}_1 \text{ and } \widehat{\delta}_2. \qquad \square
\end{aligned}$$

Theorem 4.2 $L(M_3) = L(M_1) \cap L(M_2)$.

Proof. For all $x \in \Sigma^*$,

$$\begin{aligned}
&x \in L(M_3) &&\\
&\Longleftrightarrow \widehat{\delta}_3(s_3,x) \in F_3 && \text{definition of acceptance}\\
&\Longleftrightarrow \widehat{\delta}_3((s_1,s_2),x) \in F_1 \times F_2 && \text{definition of } s_3 \text{ and } F_3\\
&\Longleftrightarrow (\widehat{\delta}_1(s_1,x),\widehat{\delta}_2(s_2,x)) \in F_1 \times F_2 && \text{Lemma 4.1}\\
&\Longleftrightarrow \widehat{\delta}_1(s_1,x) \in F_1 \text{ and } \widehat{\delta}_2(s_2,x) \in F_2 && \text{definition of set product}\\
&\Longleftrightarrow x \in L(M_1) \text{ and } x \in L(M_2) && \text{definition of acceptance}\\
&\Longleftrightarrow x \in L(M_1) \cap L(M_2) && \text{definition of intersection.} \qquad \square
\end{aligned}$$

To show that regular sets are closed under complement, take a deterministic automaton accepting A and interchange the set of accept and nonaccept states. The resulting automaton accepts exactly when the original automaton would reject, so the set accepted is $\sim A$.

Once we know regular sets are closed under $\cap$ and $\sim$, it follows that they are closed under $\cup$ by one of the De Morgan laws:

$$A \cup B = \sim(\sim A \cap \sim B).$$

If you use the constructions for $\cap$ and $\sim$ given above, this gives an automaton for $A \cup B$ that looks exactly like the product automaton for $A \cap B$, except that the accept states are

$$F_3 = \{(p,q) \mid p \in F_1 \text{ or } q \in F_2\} = (F_1 \times Q_2) \cup (Q_1 \times F_2)$$

instead of $F_1 \times F_2$.

Historical Notes

Finite-state transition systems were introduced by McCulloch and Pitts in 1943 [84]. Deterministic finite automata in the form presented here were studied by Kleene [70]. Our notation is borrowed from Hopcroft and Ullman [60].