CIS511
Notes on the "Closure Definition" Of the Regular Languages

Jean Gallier

Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA
jean@saul.cis.upenn.edu

May 8, 2010

Abstract. The definition of the regular languages (over some alphabet Σ) as the smallest family of languages that contains some basic languages and is closed under union, concatenation and Kleene $*$, is often confusing to novices. In these notes, we attempt to explain clearly how this definition goes and what it achieves.

1 The Closure Definition of the Regular Languages

Let $\Sigma = \{a_1, \ldots, a_m\}$ be some alphabet. We would like to define a family of languages, $R(\Sigma)$, by singling out some very basic (atomic) languages, namely the languages $\{a_1\}, \ldots, \{a_m\}$, the empty language, and the trivial language, $\{\epsilon\}$, and then forming more complicated languages by repeatedly forming union, concatenation and Kleene $*$ of previously constructed languages. By doing so, we hope to get a family of languages ($R(\Sigma)$) that is closed under union, concatenation, and Kleene $*$. This means that for any two languages, $L_1, L_2 \in R(\Sigma)$, we also have $L_1 \cup L_2 \in R(\Sigma)$ and $L_1L_2 \in R(\Sigma)$, and for any language $L \in R(\Sigma)$, we have $L^* \in R(\Sigma)$. Furthermore, we would like $R(\Sigma)$ to be the smallest family with these properties. How do we achieve this rigorously?

First, let us look more closely at what we mean by a family of languages. Recall that a language (over Σ) is *any* subset, L, of Σ^*. Thus, the set of all languages is 2^{Σ^*}, the power set of Σ^*. If Σ is nonempty, this is an uncountable set. Next, we define a *family*, $\mathcal{L}$, of languages to be any subset of 2^{Σ^*}. This time, the set of families of languages is $2^{2^{\Sigma^*}}$. This is a huge set. We can use the inclusion relation on $2^{2^{\Sigma^*}}$ to define a partial order on families of languages. So, $\mathcal{L}_1 \subseteq \mathcal{L}_2$ iff for every language, L, if $L \in \mathcal{L}_1$ then $L \in \mathcal{L}_2$.

We can now state more precisely what we are trying to do. Consider the following properties for a family of languages, $\mathcal{L}$:

(1) We have $\{a_1\}, \ldots, \{a_m\}, \emptyset, \{\epsilon\} \in \mathcal{L}$, i.e., $\mathcal{L}$ contains the "atomic" languages.

(2a) For all $L_1, L_2 \in \mathcal{L}$, we also have $L_1 \cup L_2 \in \mathcal{L}$.

(2b) For all $L_1, L_2 \in \mathcal{L}$, we also have $L_1L_2 \in \mathcal{L}$.

(2c) For all $L \in \mathcal{L}$, we also have $L^* \in \mathcal{L}$.

In other words, $\mathcal{L}$ is closed under union, concatenation and Kleene $*$.

Now, what we want is the smallest (w.r.t. inclusion) family of languages that satisfies properties (1) and (2)(a)(b)(c). We can construct such a family using an *inductive definition*. This inductive definition constructs a sequence of families of languages, $(R(\Sigma)_n)_{n\geq 0}$, called the *stages of the inductive definition*, as follows:

$$\begin{aligned} R(\Sigma)_0 &= \{\{a_1\}, \ldots, \{a_m\}, \emptyset, \{\epsilon\}\}, \\ R(\Sigma)_{n+1} &= R(\Sigma)_n \cup \{L_1 \cup L_2,\ L_1L_2,\ L^* \ \mid L_1, L_2, L \in R(\Sigma)_n\}. \end{aligned}$$

Then, we define $R(\Sigma)$ by

$$R(\Sigma) = \bigcup_{n\geq 0} R(\Sigma)_n.$$

Thus, a language L belongs to $R(\Sigma)$ iff it belongs L_n, for some $n \geq 0$. Observe that

$$R(\Sigma)_0 \subseteq R(\Sigma)_1 \subseteq R(\Sigma)_2 \subseteq \cdots R(\Sigma)_n \subseteq R(\Sigma)_{n+1} \subseteq \cdots \subseteq R(\Sigma),$$

so that if $L \in R(\Sigma)_n$, then $L \in R(\Sigma)_p$, for all $p \geq n$. Also, there is some smallest n for which $L \in R(\Sigma)_n$ (the *birthdate* of L!). In fact, all these inclusions are strict. Note that each $R(\Sigma)_n$ only contains a finite number of languages (but some of the languages in $R(\Sigma)_n$ are infinite, because of Kleene $*$). Then we define the *Regular languages, Version 2*, as the family $R(\Sigma)$.

Of course, it is far from obvious that $R(\Sigma)$ coincides with the family of languages accepted by DFA's (or NFA's), what we call the regular languages, version 1. However, this is the case, and this can be demonstrated by giving two algorithms. Actually, it will be slightly more convenient to define a notation system, the *regular expressions*, to denote the languages in $R(\Sigma)$. Then, we will give an algorithm that converts a regular expression, R, into an NFA, N_R, so that $L_R = L(N_R)$, where L_R is the language (in $R(\Sigma)$) denoted by R. We will also give an algorithm that converts an NFA, N, into a regular expression, R_N, so that $L(R_N) = L(N)$.

But before doing all this, we should make sure that $R(\Sigma)$ is indeed the family that we are seeking. This is the content of

Lemma 1.1 *The family, $R(\Sigma)$, is the smallest family of languages which contains the atomic languages $\{a_1\}, \ldots, \{a_m\}$, $\emptyset$, $\{\epsilon\}$, and is closed under union, concatenation, and Kleene $*$.*

Proof. There are two things to prove.

(i) We need to prove that $R(\Sigma)$ has properties (1) and (2)(a)(b)(c).

(ii) We need to prove that $R(\Sigma)$ is the smallest family having properties (1) and (2)(a)(b)(c).

(i) Since

$$R(\Sigma)_0 = \{\{a_1\}, \ldots, \{a_m\}, \emptyset, \{\epsilon\}\},$$

it is obvious that (1) holds. Next, assume that $L_1, L_2 \in R(\Sigma)$. This means that there are some integers $n_1, n_2 \geq 0$, so that $L_1 \in R(\Sigma)_{n_1}$ and $L_2 \in R(\Sigma)_{n_2}$. Now, it is possible that $n_1 \neq n_2$, but if we let $n = \max\{n_1, n_2\}$, as we observed that $R(\Sigma)_p \subseteq R(\Sigma)_q$ whenever $p \leq q$, we are guaranteed that both $L_1, L_2 \in R(\Sigma)_n$. However, by the definition of $R(\Sigma)_{n+1}$ (that's why we defined it this way!), we have $L_1 \cup L_2 \in R(\Sigma)_{n+1} \subseteq R(\Sigma)$. The same argument proves that $L_1L_2 \in R(\Sigma)_{n+1} \subseteq R(\Sigma)$. Also, if $L \in R(\Sigma)_n$, we immediately have $L^* \in R(\Sigma)_{n+1} \subseteq R(\Sigma)$. Therefore, $R(\Sigma)$ has properties (1) and (2)(a)(b)(c).

(ii) Let $\mathcal{L}$ be any family of languages having properties (1) and (2)(a)(b)(c). We need to prove that $R(\Sigma) \subseteq \mathcal{L}$. If we can prove that $R(\Sigma)_n \subseteq \mathcal{L}$, for all $n \geq 0$, we are done (since then, $R(\Sigma) = \bigcup_{n \geq 0} R(\Sigma)_n \subseteq \mathcal{L}$). We prove by induction on n that $R(\Sigma)_n \subseteq \mathcal{L}$, for all $n \geq 0$.

The base case $n = 0$ is trivial, since $\mathcal{L}$ has (1), which says that $R(\Sigma)_0 \subseteq \mathcal{L}$. Assume inductively that $R(\Sigma)_n \subseteq \mathcal{L}$. We need to prove that $R(\Sigma)_{n+1} \subseteq \mathcal{L}$. Pick any $L \in R(\Sigma)_{n+1}$. Recall that

$$R(\Sigma)_{n+1} = R(\Sigma)_n \cup \{L_1 \cup L_2,\ L_1L_2,\ L^* \ \mid L_1, L_2, L \in R(\Sigma)_n\}.$$

If $L \in R(\Sigma)_n$, then $L \in \mathcal{L}$, since $R(\Sigma)_n \subseteq \mathcal{L}$, by the induction hypothesis. Otherwise, there are three cases:

(a) $L = L_1 \cup L_2$, where $L_1, L_2 \in R(\Sigma)_n$. By the induction hypothesis, $R(\Sigma)_n \subseteq \mathcal{L}$, so, we get $L_1, L_2 \in \mathcal{L}$; since $\mathcal{L}$ has 2(a), we have $L_1 \cup L_2 \in \mathcal{L}$.

(b) $L = L_1 L_2$, where $L_1, L_2 \in R(\Sigma)_n$. By the induction hypothesis, $R(\Sigma)_n \subseteq \mathcal{L}$, so, we get $L_1, L_2 \in \mathcal{L}$; since $\mathcal{L}$ has 2(b), we have $L_1 L_2 \in \mathcal{L}$.

(c) $L = L_1^*$, where $L_1 \in R(\Sigma)_n$. By the induction hypothesis, $R(\Sigma)_n \subseteq \mathcal{L}$, so, we get $L_1 \in \mathcal{L}$; since $\mathcal{L}$ has 2(c), we have $L_1^* \in \mathcal{L}$.

Thus, in all cases, we showed that $L \in \mathcal{L}$, and so, $R(\Sigma)_{n+1} \subseteq \mathcal{L}$, which proves the induction step. □

Students should study carefully the above proof. Although simple, it is the prototype of many proofs appearing in the theory of computation.