MACM-300: Intro to Formal Languages and Automata

Anoop Sarkar – anoop@cs.sfu.ca

Substitutions and homomorphisms.
Let L be a regular language over an alphabet Σ. Consider a new regular language R_a (unrelated to L) for each symbol $a \in \Sigma$. Let $a_1 a_2 \ldots a_n$ be a string in L where $a_i \in \Sigma$. Replace each a_i with some arbitrary string w_i in R_{a_i} giving us a new string $w_1 w_2 \ldots w_n$. A *substitution* is the mapping produced by replacing each symbol a_i for each string in L with all possible strings from R_{a_i}. We shall show that each such string $w_1 w_2 \ldots w_n$ is generated by a regular language.
Formally, a *substitution* f is a mapping of alphabet Σ onto subsets of Δ^* for some alphabet Δ. Thus f associates a language with each symbol in Σ. The mapping f is extended to strings as follows: $f(\varepsilon) = \varepsilon$ and $f(xa) = f(x)f(a)$. And for a language L, we have

$$f(L) = \bigcup_{x \in L} f(x)$$

Example. Let $L = 0^*(0 \cup 1)1^*$ and let $f(0) = a$ and $f(1) = b^*$. Then a substitution for language L is $f(L) = a^*(a \cup b^*)(b^*)^* = a^*b^*$.

Theorem. The class of regular languages is closed under substitution.

Proof. Let $R \subseteq \Sigma^*$ be a regular language over alphabet Σ and for each $a \in \Sigma$ let $R_a \subseteq \Delta^*$ be a regular language. Let $f : \Sigma \rightarrow \Delta^*$ be the substitution defined by $f(a) = R_a$. Pick a regular expression that is equivalent to R and regular expressions for each R_a. Replace each occurrence of symbol a in the regular expression for R by the regular expression for R_a. The new regular expression derived using this method is equivalent to the language $f(R)$. This can be proved using induction on the regular expression operators:

1. Base case: for regular expression with a single symbol a, $f(a) = \varepsilon$ or $f(a) = b$ where $b \in \Delta$. In both cases the regular expression provided by the replacement operation above is equivalent to the language $f(R)$.

2. Recursive case: if R_1 and R_2 are regular expressions such that the replacement operation provided above provide new regular expressions equivalent to $f(R_1)$ and $f(R_2)$ then,

   - $f(R_1 \cup R_2) = f(R_1) \cup f(R_2)$
   - $f(R_1 R_2) = f(R_1)f(R_2)$
   - $f(R_1 *) = f(R_1)*$

A type of substitution that is often used is called a *homomorphism*. A homomorphism h is a substitution such that $h(a)$ contains a single string for each symbol a from Σ.

Example. Let $h(0) = aa$ and $h(1) = aba$. Then if 010 is a string in some regular language, then $h(010) = aabaaa$. For a regular language L equivalent to regular expression $(01)^*$ then $h(L)$ is language equivalent to $(aaaba)^*$.

An *inverse homomorphism* of a language L is defined as:

$$h^{-1}(L) = \{x \mid h(x) \in L\}$$

Example. Let $h(0) = aa$ and $h(1) = aba$. Let language $L = (ab \cup ba)^*a$. Then $h^{-1}(L)$ consists of only the string 1.

Note that homomorphisms is just a special case of substitution and so regular languages are closed under homomorphisms as well.

Theorem. The class of context-free languages is closed under substitution.

Proof. Let L be a CFL, $L \subseteq \Sigma^*$ and for each $a \in \Sigma$ let L_a be a CFL. Let L be $L(G)$ and for each $a \in \Sigma$ let L_a be $L(G_a)$. Without loss of generality assume that the variables of G and all the G_a's are disjoint. Construct a new grammar G' as follows. The variables of G' is all the variables from G and all G_a's. The start variable of G' is the start symbol of G. The rules of G' are all the productions of the G_a's together with all the rules formed by taking a rule $A \to \alpha$ of G and substituting S_a the start symbol of G_a for each instance of an $a \in \Sigma$ appearing in α.

Example. Let L be the language with equal number of a's and b's. Let G be the grammar for L:

$$S \to aSbS \mid bSaS \mid \varepsilon$$

Let $L_a = \{0^n1^n \mid n \geq 1\}$ and let $L_b = \{ww^R \mid \ w \text{ is in } (0 \cup 2)^* \ \}$. Let G_a be:

$$S_a \to 0S_a1 \mid 01$$

And let G_b be:

$$S_b \to 0S_b0 \mid 2S_b2 \mid \varepsilon$$

If f is the substitution $f(a) = L_a$ and $f(b) = L_b$ then $f(L)$ is generated by the grammar:

$$\begin{array}{rcl} S & \to & S_aSS_bS \mid S_bSS_aS \mid \varepsilon \\ S_a & \to & 0S_a1 \mid 01 \\ S_b & \to & 0S_b0 \mid 2S_b2 \mid \varepsilon \end{array}$$

This proof also shows that CFLs are closed under homomorphisms.

Note that the languages $\{a, b\}$, $\{ab\}$ and a^* are CFLs, and so we can substitute any two CFLs L_a and L_b into $\{a, b\}$ and this shows that CFLs are *closed under union*, similarly we can substitute CFLs L_a and L_a into $\{ab\}$ to show CFLs are *closed under concatenation*, and substituting any CFL L_a into the CFL for a^* shows CFLs are *closed under* $*$.