Formal languages and automatas

Ruben Triwari

Seminar: Algebra & Computer Science, LMU Munich

July 6, 2024

Outline

Alphabets & Languages

An Alphabet Σ is a set of characters.
Examples:

1. $\Sigma = \{a, b, c\}$
2. $\Sigma = \{x\}$
3. $\Sigma = \emptyset$

A Language L is a set of words with characters out of an Alphabet Σ.
Examples:

1. $\Sigma = \{a, b, c\} \rightsquigarrow L = \{aaa, bbb, ccc, abc\}$
2. $\Sigma = \{x\} \rightsquigarrow L = \{x, xx, xxx, xxxx\}$
3. $\Sigma = \emptyset \rightsquigarrow L = \{\epsilon\}$

$\rightsquigarrow L \subset \Sigma^*$

We can also write a Language with a formal series, with a fixed Σ:

$$L = \sum (L, w) w$$

$$\leftrightsquigarrow L = \bigcup_{w \in \Sigma^*} \underbrace{(L, w)}_{\text{0 or 1}} \{w\}$$

Examples:

1. $\Sigma = \{a, b, c\} \rightsquigarrow L = \{aaa, abc\}$
   $\rightsquigarrow L = 1aaa + 1abc + 0a + 0b + 0c + \dots$
2. $\Sigma = \{x\} \rightsquigarrow L = \{x, xx\}$
   $\rightsquigarrow L = 1x + 1xx + 0xxx + 0xxxx + 0xxxxx + \dots$
3. $\Sigma = \emptyset \rightsquigarrow L = \{\epsilon\}$
   $\rightsquigarrow L = 1\epsilon$

Addition

Now we can define Addition on formal series:

$$U + V = \sum (\underbrace{(U, w) + (V, w)}_{\text{Boolean addition}})w$$

$$\leftrightsquigarrow U + V = U \cup V$$

Exmaple:
Let $U = \{x, xx\}$ and $V = \{aaa, abc\}$ languages:

$$\begin{aligned} U + V &= (1+0)x + (1+0)xx + (0+1)aaa + (0+1)abc + (0+0)a + \dots \\ &= 1x + 1xx + 1aaa + 1abc + 0a + \dots \end{aligned}$$

Next we can define multiplication on formal series:

$$U \cdot V = \sum \Big(\underbrace{(U, s) \cdot (V, t)}_{\text{Boolean multiplication}} \Big) w, \text{ such that } st = w$$

$$\leftrightsquigarrow U \cdot V = \{\ st \mid s \in U \wedge t \in V\}$$

Exmaple:
Let $U = \{x, xx\}$ and $V = \{aaa, abc\}$ languages:

$$\begin{aligned} U \cdot V = (1 \cdot 1)xaaa + (1 \cdot 1)xxaaa + (1 \cdot 1)xabc + (1 \cdot 1)xxabc \\ + (0 \cdot 0)axxx + \dots \\ = 1xaaa + 1xxaaa + 1xabc + 1xxabc + 0axxx + \dots \end{aligned}$$

With these definitions, all languages with a fixed alphabet Σ form an algebra $\mathbb{B}\langle\Sigma\rangle$ over $\mathbb{B}$.

Kleene star:
Let U be a Language.

$$U^* = \epsilon + U + U^2 + U^3 + \dots$$

Exmaple:
Let $U = x$.

$$U^* = \epsilon + x + x^2 + x^3 + x^4 + \dots$$

Outline

All languages generated by a finite number of additions, multiplications, and kleene star is a rational (regular) language.

Examples:

$\Sigma = \{x, y, z\}$

$$L_1 = x + y \tag{1}$$

$$L_2 = (x + y + z)^* \tag{2}$$

$$L_3 = (x + y^*)^* z^* \tag{3}$$

$$L_4 = (xyz)^*(y + x^* zxyx^*)^* \tag{4}$$

We show that not all languages generated by a single letter are rational.

Let $L_x = x$.

A gap in a language generated by L_x is the amount of consecutive missing powers.

Examples:

$$L_1 = x \rightsquigarrow gaps(L_1) = \{0\}$$
$$L_2 = x + x^2 + x^3 \rightsquigarrow gaps(L_2) = \{0\}$$
$$L_3 = x + x^5 + x^7 \rightsquigarrow gaps(L_3) = \{0, 3, 1\}$$
$$L_4 = x + x^5 + x^7 \rightsquigarrow maxgap(L_4) = 3$$

Raional languages generated by x have alway a finite maximum gap.
Examples of not rational languages:

$$L_1 = x + x^2 + x^4 + x^7 + x^{11} \cdots \rightsquigarrow gaps(L_1) = \{0, 1, 2, 3, \ldots\}$$
$$L_2 = x^3 + x^{31} + x^{314} + x^{3141} \cdots \rightsquigarrow gaps(L_2) = \{27, 282, 2826, \ldots\}$$

$\rightsquigarrow$ Only way to get an endless sequence is using the star operator.
$\rightsquigarrow$ The star operator just repeats the language an abritary amount of times
$\rightsquigarrow$ Informally we can say rational languages are finite or have an repeating pattern.

Proof sketch

Claim: **All rational languages generated by L_x have a maximum gap**
Let L be language generated by L_x

$$maxgap(L) < \infty.$$

Structural induction over $+, \cdot, ^*$:

Base case:

$$L = L_x = x$$

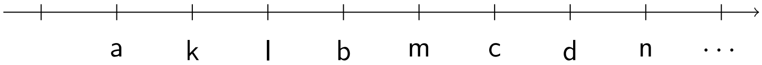
$$\implies maxgap(L) = maxgap(L_x) = maxgap(x) = 0$$

Addition case:
Let L_1, L_2 languages generated by L_x with a maximum gap.
We need to show that:

$$maxgap(L_1 + L_2) < \infty$$

$$L_1 = x^a + x^b + x^c + x^d + \dots$$
$$L_2 = x^k + x^l + x^m + x^n + \dots$$

$$\downarrow$$



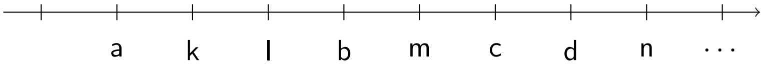
Proof sketch

Addition case:
Let L_1, L_2 languages generated by L_x with a maximum gap.
We need to show that:

$$maxgap(L_1 + L_2) < \infty$$

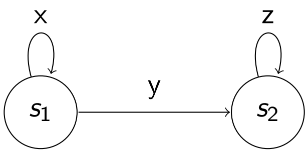
$$L_1 = x^a + x^b + x^c + x^d + \dots$$
$$L_2 = x^k + x^l + x^m + x^n + \dots$$

$$\downarrow$$



Outline

Automatas and transistion matrices



$$\downarrow$$

$$\begin{array}{cc} \begin{matrix} s_1 \\ s_2 \end{matrix} & \begin{bmatrix} x & y \\ 0 & z \end{bmatrix} \\ & \begin{matrix} s_1 & s_2 \end{matrix} \end{array}$$

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Formally, we can write:
Let $S = \{s_1, s_2, \dots, s_n\}$, $A \in (\Sigma^*)^{n \times n}$.
Then a transition from state s_1 to s_2:

$$s_i A_{ij} = s_j$$

Exmaple:

Let $S = \{s_1, s_2\}$ and $A = \begin{pmatrix} x & y \\ 0 & z \end{pmatrix}$

$$s_1 A_{11} = s_1 x = s_1$$
$$s_1 A_{12} = s_1 y = s_2$$
$$s_2 A_{22} = s_2 z = s_2$$

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

What happends if we apply matrix multiplication on $A \in (\Sigma^*)^{n \times n}$?

$$C = A^1 : \quad c_{ij} = a_{ij} \rightsquigarrow \text{from state i to state j in 1 steps}$$

$$C = A^2 : \quad c_{ij} = \sum_{k=1}^{n} a_{ik} a_{kj} \rightsquigarrow \text{from state i to state j in 2 steps}$$

$$C = A^3 : \quad c_{ij} = \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} a_{ik_1} a_{k_1 k_2} a_{k_2 j} \rightsquigarrow \text{from state i to state j in 3 steps}$$

$$\vdots$$

$$C = A^n : \quad c_{ij} = \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \cdots \sum_{k_{n-1}=1}^{n} a_{ik_1} a_{k_1 k_2} \ldots a_{k_{n-2} k_{n-1}} a_{k_{n-1} j}$$

$$\rightsquigarrow \text{from state i to state j in n steps}$$

Combining these will give us all possible inputs from one state in another:

$$A^* = E_n + A + A^2 + A^3 + \dots$$
$$A^* = \sum_{k \in \mathbb{N}_0} A^k$$

$\rightsquigarrow$ In $a_{ij} \in A$ are all possible inputs carrying state i to state j in any abitrary amount of steps.

Outline

Outline

Outline

Outline

Thanks for your attention!