Theory of Automata

Lecture No. 12

Reading Material

Introduction to Computer Theory Chapter 7

Summary

Examples of writing REs to the corresponding TGs, RE corresponding to TG accepting EVEN-EVEN language, Kleene’s theorem part III (method 1: union of FAs), examples of FAs corresponding to simple REs, example of Kleene’s theorem part III (method 1) continued

Example

Consider the following TG

To have single initial and single final state the above TG can be reduced to the following

To obtain single transition edge between 1 and 3; 2 and 4, the above can be reduced to the following

To eliminate states 1, 2, 3 and 4, the above TG can be reduced to the following TG

OR
To connect the initial state with the final state by single transition edge, the above TG can be reduced to the following

\[ -(b+aa)b^*+(a+bb)a^*+ \]

Hence the required RE is \((b+aa)b^*+(a+bb)a^*\)

**Example**

Consider the following TG, accepting EVEN-EVEN language

It is to be noted that since the initial state of this TG is final as well and there is no other final state, so to obtain a TG with single initial and single final state, an additional initial and a final state are introduced as shown in the following TG

To eliminate state 2, the above TG may be reduced to the following

\[ 3- \rightarrow (ab+ba)(aa+bb)^*(ab+ba) \]

To have single loop at state 1, the above TG may be reduced to the following

\[ (aa+bb)^+(ab+ba)(aa+bb)^*(ab+ba) \]

To eliminate state 1, the above TG may be reduced to the following

\[ \Lambda(aa+bb+(ab+ba)(aa+bb)^*(ab+ba))^*\Lambda \]

Hence the required RE is \((aa+bb+(ab+ba)(aa+bb)^*(ab+ba))^*\)

**Kleene’s Theorem Part III**

**Statement:**

If the language can be expressed by a RE then there exists an FA accepting the language.

A) As the regular expression is obtained applying addition, concatenation and closure on the letters of an
alphabet and the Null string, so while building the RE, sometimes, the corresponding FA may be built easily, as shown in the following examples

**Example**
Consider the language, defined over Σ = {a,b}, **consisting of only** b, then this language may be accepted by the following FA

![FA Example](image)

which shows that this FA helps in building an FA accepting only one letter

**Example**
Consider the language, defined over Σ = {a,b}, **consisting of only** a, then this language may be accepted by the following FA

![FA Example](image)

B) As, if r₁ and r₂ are regular expressions then their sum, concatenation and closure are also regular expressions, so an FA can be built for any regular expression if the methods can be developed for building the FAs corresponding to the sum, concatenation and closure of the regular expressions along with their FAs. These three methods are explained in the following discussion

**Method 1 (Union of two FAs):** Using the FAs corresponding to r₁ and r₂ an FA can be built, corresponding to r₁ + r₂. This method can be developed considering the following examples

**Example**
Let \( r₁ = (a+b) \) define L₁ and the FA₁ be

![FA Example](image)

and \( r₂ = (a+b) \) define L₂ and FA₂ be

![FA Example](image)

Let FA₃ be an FA corresponding to \( r₁ + r₂ \), then the initial state of FA₃ must correspond to the initial state of FA₁ or the initial state of FA₂. Since the language corresponding to \( r₁ + r₂ \) is the union of corresponding languages L₁ and L₂, consists of the strings belonging to L₁ or L₂ or both, therefore a final state of FA₃ must correspond to a final state of FA₁ or FA₂ or both. Since, in general, FA₃ will be different from both FA₁ and FA₂, so the labels of the states of FA₃ may be supposed to be \( z₁, z₂, z₃, \ldots \), where \( z₁ \) is supposed to be the initial state. Since \( z₁ \) corresponds to the states \( x₁ \) or \( y₁ \), so there will be two transitions separately for each letter read at \( z₁ \). It will give two possibilities of states either \( z₁ \) or different from \( z₁ \). This process may be expressed in the following transition table for all possible states of FA₃.
RE corresponding to the above FA may be \( r_1 + r_2 = (a+b)^*b + (a+b)^*aa(a+b)^* \).

**Note:** Further examples are discussed in the next lecture.