CS 208 : Automata Theory and Logic	17/02/2024

Lecture - 18

Topic: Regular Expressions

Scribed by: Sanskar Shaurya (22B0985)
Checked and compiled by:

Disclaimer. Please note this document has not received the usual scrutiny that formal publications enjoy. This may be distributed outside this class only with the permission of the instructor.

1 Regular Expressions

We know that NFAs with ϵ are equivalent to NFAs without ϵ which are equivalent to DFAs. Now we will introduce another way of defining a language known as **Regular Expressions** which are much easier to write.

1.1 Syntax of Regular Expressions

Let us say our alphabet Σ is $\{a, b\}$.
Our syntax will include two special symbols a,b. These are our basic building blocks alongside the symbol ϵ.
If e_1 and e_2 are two regular expressions, then the following will also be regular expression:

- $e_1 + e_2$
- $e_1 \cdot e_2$
- e_1*
- (e_1)

1.2 Semantics of Regular Expressions

Semantics of a Regular Expression will be a language.

- $[\![a]\!] = \{b\}$ $\quad [\![b]\!] = \{b\}$ $\quad [\![\epsilon]\!] = \{\epsilon\}$
- $[\![e_1 + e_2]\!] = [\![e_1]\!] \cup [\![e_1]\!]$
- $[\![e_1 \cdot e_2]\!] = [\![e_1]\!] \cdot [\![e_1]\!] = \{u \cdot v \mid u \in [\![e_1]\!],\ v \in [\![e_1]\!]\}$

Example: $[\![(a \cdot b) + a]\!] = \{ab, a\}$
The order of precedence among these operators is $*$ then $\cdot$ then $+$.
So the above regular expression is equivalent to $a \cdot b + a$

- $e_1^n = \underbrace{e_1 \cdot e_1 \cdots e_1}_{n \text{ times}}$

Suppose $e_1 = a + b$
Then $[\![e_1]\!] = \{\text{a,b}\}$ and $[\![e_1^3]\!] = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$
$[\![e_1^0]\!]$ is used to denote $\{\epsilon\}$, the set containing just the empty string.

- $[\![e_1*]\!] = \bigcup_{i \geq 0} [\![e_1^i]\!]$

Examples:

- $[\![(a+b)*]\!] = \{\epsilon, a, b, aa, ab, ba, bb, \ldots\}$
  $[\![(a+b)*]\!]$ is equivalent to Σ^* as it represents all finite length strings formed from $\Sigma = \{a, b\}$
- $[\![a * + b*]\!] = \{u \in \Sigma^* \mid u = a^n \text{ or } b^n,\ n \geq 0\}$
- $[\![(a*) \cdot (b*)]\!] = \{u \in \Sigma^* \mid u = a^n \cdot b^m,\ n, m \geq 0\}$

Q) Is $[\![((a*) \cdot (b*))*]\!] = L((a+b)*)$[1]?
Easy to see that $[\![((a*) \cdot (b*))*]\!] \subseteq L((a+b)*)$ as the language defined by the second regular expression contains every finite length string formed by {a,b}.
Claim: $L((a+b)*) \subseteq [\![((a*) \cdot (b*))*]\!]$
Proof: Consider any arbitrary string of length n, and then consider $((a*) \cdot (b*))^n$. As we are concatenating n strings from $((a*) \cdot (b*))$, we can pick a single letter a or b from each of the n strings corresponding to the letter at that position. As we can do this for any string with a positive length n, this means that every finite length string is in the language defined by the regular expression. ϵ is by definition in $[\![((a*) \cdot (b*))*]\!]$ and hence our claim is correct.
This means that $[\![((a*) \cdot (b*))*]\!] = L((a+b)*)$. Hence just looking at syntax, we cannot decide whether the language defined by them is different or not, but we can do this easily after converting the regular expression to a DFA.

[1] $L(e_1)$ denotes the Language defined by the regular expression e_1