Arden's Theorem

Ayush Tandon (2K22/CO/133)

September 18, 2024

1 Definition

Arden's theorem is used to convert a given **finite automata** to a **regular expression**. It is used to solve linear equations in the form of $R = Q + RP$, where P and Q are regular expressions and R is the unknown regular expression to be solved.

2 Theorem

Arden's Theorem: Let P and Q be two regular expressions over a given alphabet. The equation

$$R = Q + RP$$

has a unique solution for R, which is:

$$R = QP^*$$

provided that P does not contain the empty string ϵ.

Note: The term P^* represents the Kleene star of P, which denotes zero or more occurrences of P.

3 Proof

Given: The equation $R = Q + RP$.

To prove: The solution for R is $R = QP^*$ if P does not contain the empty string ϵ.

Proof:

- Consider the equation $R = Q + RP$.
- Substituting the value of R in the equation recursively:

$$R = Q + (Q + RP)P = Q + QP + RP^2.$$

- Continuing this process, we get:

$$R = Q + QP + QP^2 + RP^3$$

- We notice that the expression for R is a series. This series can be expressed as:

$$R = Q + QP + QP^2 + QP^3 + \cdots$$

- Taking Q common we get

$$R = Q(\epsilon + P + P^2 + P^3 + \cdots)$$

- The closed form of the infinite geometric series is given by:

$$R = QP^*$$

- Hence, the solution to the equation is $R = QP^*$.

This proves the theorem, provided P does not contain ϵ.

3.1 Why P cannot be empty string?

In Arden's Theorem, the condition that P must not be a null (empty) string is essential because if $P = \epsilon$, the equation would lead to an **infinite recursion** that cannot be resolved in a meaningful way.

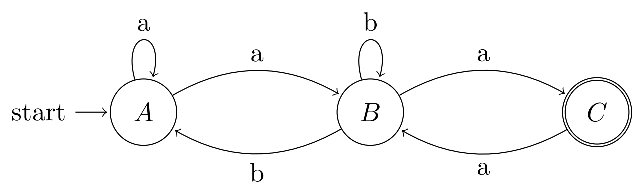
4 Steps for Solving Regular Equations using Arden's Theorem

To solve regular equations using Arden's Theorem, follow these steps:

1. Express the regular equation in the form $R = Q + RP$.
2. Ensure that P does not contain the empty string ϵ. If it does, the theorem cannot be directly applied.
3. Apply Arden's Theorem: The solution is $R = QP^*$.
4. If there are multiple equations, solve them sequentially using substitution.

5 Illustration

Consider the finite automaton for the alphabet $\Sigma = \{a, b\}$:



We will write the equation for each state:

$$A = \epsilon + A.a + B.b \tag{i}$$
$$B = A.a + B.b + C.a \tag{ii}$$
$$C = B.a \tag{iii}$$

Substituting equation (iii) in (ii), we get:

$$B = A.a + B.b + B.a.a$$
$$B = A.a + B.(b + aa)$$

This is of the form $R = Q + R.P$
The solution by Arden's Theorem is $R = Q.P^*$

$$B = A.a(b + aa)^* \tag{iv}$$

From (iv) and (i):

$$A = \epsilon + A.a + A.a(b + aa)^*b$$
$$A = \epsilon + A.(a + a(b + aa)^*b)$$

Again, by Arden's Theorem, we get:

$$A = \epsilon.(a + a(b + aa)^*b)^*$$
$$A = (a + a(b + aa)^*b)^* \tag{v}$$

Substitute A into B to get:

$$B = (a + a(b + aa)^*b)^*a(b + aa)^* \tag{vi}$$

From (vi) and (iii), we get the **final state** C:

$$C = (a + a(b + aa)^*b)^*a(b + aa)^*a \tag{vii}$$

Therefore, the **regular expression** for A, B, C are:

$$A = (a + a(b + aa)^*b)^*$$

$$B = (a + a(b + aa)^*b)^*a(b + aa)^*$$

$$C = (a + a(b + aa)^*b)^*a(b + aa)^*a$$

For a finite automata, regular expression may not always be unique because there can be multiple ways of solving equations by substitution. It is similar to language and DFA. For a language L, there may be multiple DFAs but minimal DFA is always unique.

6 Conclusion

Arden's Theorem provides a direct method for solving regular equations in formal language theory. By converting equations into a standard form, we can derive the corresponding regular expressions, which describe the languages accepted by finite automata.