CM0081 Automata and Formal Languages
§ 3.4 Algebraic Laws for Regular Expressions

Andrés Sicard-Ramírez

Universidad EAFIT

Semester 2024-1

Preliminaries

Conventions

- The number and page numbers assigned to chapters, examples, exercises, figures, quotes, sections and theorems on these slides correspond to the numbers assigned in the textbook [Hopcroft, Motwani and Ullman 2007].
- The natural numbers include the zero, that is, $\mathbb{N} = \{0, 1, 2, \dots\}$.
- The power set of a set A, that is, the set of its subsets, is denoted by $\mathcal{P}\,A$.

Algebraic Laws for Regular Expressions

Definition

Two regular expressions with variables are **equivalent** if whatever languages we substitute for the variables, the results of the two expressions are the same language.

Algebraic Laws for Regular Expressions

Definition

Two regular expressions with variables are **equivalent** if whatever languages we substitute for the variables, the results of the two expressions are the same language.

Notation

Let M, N and P be regular expression variables.

Algebraic Laws for Regular Expressions

Definition

Two regular expressions with variables are **equivalent** if whatever languages we substitute for the variables, the results of the two expressions are the same language.

Notation

Let M, N and P be regular expression variables.

Sugar syntax

$$M^+ := MM^*,$$
$$M? := \varepsilon + M.$$

Algebraic Laws for Regular Expressions

Some laws for union

$$(M + N) + P = M + (N + P) \quad \text{(associativity)}$$

$$M + \emptyset = \emptyset + M = M \quad \text{(identity)}$$

$$M + N = N + M \quad \text{(commutativity)}$$

$$M + M = M \quad \text{(idempotence)}$$

Observation

There is no inverse for union.

Algebraic Laws for Regular Expressions

Some laws for concatenation

$(MN)P = M(NP)$	(associativity)
$M\varepsilon = \varepsilon M = M$	(identity)
$MN \neq NM$	(non-commutativity)
$M\emptyset = \emptyset M = \emptyset$	($\emptyset$ is the annihilator for concatenation)

Observation

There is no inverse for concatenation.

Algebraic Laws for Regular Expressions

Some laws for union and concatenation

$$M(N+P) = MN + MP \qquad \text{(distributive)}$$

$$(M+N)P = MP + NP \qquad \text{(distributive)}$$

Algebraic Laws for Regular Expressions

Some laws for closure

$$(M^*)^* = M^* \qquad \text{(idempotence)}$$

$$\emptyset^* = \varepsilon$$

$$\varepsilon^* = \varepsilon$$

$$(\varepsilon + M)^* = M^*$$

$$M^* = M^+ + \varepsilon$$

Algebraic Laws for Regular Expressions

Observation

A complete set of axioms for the regular expressions is indicated in [Kozen 2012, Lecture 9].

Simplification of Regular Expressions

Example

$$
\begin{aligned}
\mathbf{0} + (\varepsilon + \mathbf{1})\underline{(\varepsilon + \mathbf{1})^*}\mathbf{0} &= \mathbf{0} + \underline{(\varepsilon + \mathbf{1})\mathbf{1}^*}\mathbf{0} && ((\varepsilon + M)^* = M^*) \\
&= \mathbf{0} + (\underline{\varepsilon\mathbf{1}^*} + \mathbf{1}\mathbf{1}^*)\mathbf{0} && \text{(distributive)} \\
&= \mathbf{0} + (\mathbf{1}^* + \underline{\mathbf{1}\mathbf{1}^*})\mathbf{0} && \text{(identity)} \\
&= \mathbf{0} + (\underline{\mathbf{1}^* + \mathbf{1}^+})\mathbf{0} && \text{(def. } L^+\text{)} \\
&= \underline{\mathbf{0} + \mathbf{1}^*\mathbf{0}} && \text{(equivalence)} \\
&= \mathbf{1}^*\mathbf{0} && \text{(equivalence)}
\end{aligned}
$$

Discovering Laws for Regular Expressions

Method

Let E and F be two regular expressions with the same set of variables $\{M_1, \ldots, M_n\}$.

To test if $E = F$:

1. Convert E and F to concrete regular expressions C and D, replacing each M_i by a different symbol $\boldsymbol{a_i}$, for $i = 1, 2, \ldots, n$.
2. Test whether $\mathrm{L}(C) = \mathrm{L}(D)$. If so, then $E = F$, and if not $E \neq F$.

Discovering Laws for Regular Expressions

Method

Let E and F be two regular expressions with the same set of variables $\{M_1, \dots, M_n\}$.

To test if $E = F$:

1. Convert E and F to concrete regular expressions C and D, replacing each M_i by a different symbol $\boldsymbol{a_i}$, for $i = 1, 2, \dots, n$.
2. Test whether $\mathrm{L}(C) = \mathrm{L}(D)$. If so, then $E = F$, and if not $E \neq F$.

Observation

We are proving by example!

Discovering Laws for Regular Expressions

Example

Prove or disprove that $M^* = M^*M^*$.

Discovering Laws for Regular Expressions

Example

Prove or disprove that $M^* = M^*M^*$.

Proof

We replace the variable M by the concrete regular expression $\boldsymbol{a}$.

$$\boldsymbol{a}^* \stackrel{?}{=} \boldsymbol{a}^*\boldsymbol{a}^*.$$

Discovering Laws for Regular Expressions

Example

Prove or disprove that $M^* = M^*M^*$.

Proof

We replace the variable M by the concrete regular expression $\boldsymbol{a}$.

$$\boldsymbol{a}^* \stackrel{?}{=} \boldsymbol{a}^*\boldsymbol{a}^*.$$

Because $\mathrm{L}(\boldsymbol{a}^*) = \mathrm{L}(\boldsymbol{a}^*\boldsymbol{a}^*)$ we conclude that

$$M^* = M^*M^*.$$

∎

Discovering Laws for Regular Expressions

Example

Prove or disprove that $M + NM = (M + N)M$.

Discovering Laws for Regular Expressions

Example

Prove or disprove that $M + NM = (M + N)M$.

Refutation

We replace the variables M and N by the concrete regular expressions $\boldsymbol{a}$ and $\boldsymbol{b}$ respectively.

$$\boldsymbol{a} + \boldsymbol{ba} \stackrel{?}{=} (\boldsymbol{a} + \boldsymbol{b})\boldsymbol{a}.$$

Discovering Laws for Regular Expressions

Example

Prove or disprove that $M + NM = (M + N)M$.

Refutation

We replace the variables M and N by the concrete regular expressions $\boldsymbol{a}$ and $\boldsymbol{b}$ respectively.

$$\boldsymbol{a} + \boldsymbol{ba} \stackrel{?}{=} (\boldsymbol{a} + \boldsymbol{b})\boldsymbol{a}.$$

Because $aa \notin \mathrm{L}(\boldsymbol{a} + \boldsymbol{ba})$ and $aa \in \mathrm{L}((\boldsymbol{a} + \boldsymbol{b})\boldsymbol{a})$ then

$$\mathrm{L}(\boldsymbol{a} + \boldsymbol{ba}) \neq \mathrm{L}((\boldsymbol{a} + \boldsymbol{b})\boldsymbol{a}).$$

So, we can conclude

$$M + NM \neq (M + N)M.$$

■

Discovering Laws for Regular Expressions

Example (Exercise 3.4.2.d)

Prove or disprove that $(M + N)^*N = (M^*N)^*$.

Discovering Laws for Regular Expressions

Example (Exercise 3.4.2.d)

Prove or disprove that $(M + N)^* N = (M^* N)^*$.

Refutation

We replace the variables M and N by the concrete regular expressions $\boldsymbol{a}$ and $\boldsymbol{b}$ respectively.

$$(\boldsymbol{a} + \boldsymbol{b})^* \boldsymbol{b} \stackrel{?}{=} (\boldsymbol{a}^* \boldsymbol{b})^*.$$

Discovering Laws for Regular Expressions

Example (Exercise 3.4.2.d)

Prove or disprove that $(M + N)^*N = (M^*N)^*$.

Refutation

We replace the variables M and N by the concrete regular expressions $\boldsymbol{a}$ and $\boldsymbol{b}$ respectively.

$$(\boldsymbol{a} + \boldsymbol{b})^*\boldsymbol{b} \stackrel{?}{=} (\boldsymbol{a}^*\boldsymbol{b})^*.$$

Since $\varepsilon \notin (\boldsymbol{a} + \boldsymbol{b})^*\boldsymbol{b}$ and $\varepsilon \in (\boldsymbol{a}^*\boldsymbol{b})^*$ then

$$(M + N)^*N \neq (M^*N)^*.$$

■

Discovering Laws for Regular Expressions

Example (counter-example)

Extensions of the previous test beyond regular expressions may fail.

Discovering Laws for Regular Expressions

Example (counter-example)

Extensions of the previous test beyond regular expressions may fail.

1. Add $\cap$ to the regular expression operators.
2. Test if $M \cap N \cap P = M \cap N$.
3. From $M = \boldsymbol{a}$, $N = \boldsymbol{b}$ and $P = \boldsymbol{c}$ and since

   $$\{a\} \cap \{b\} \cap \{c\} = \emptyset = \{a\} \cap \{b\},$$

   we should conclude that the 'property' is true.

Discovering Laws for Regular Expressions

Example (counter-example)

Extensions of the previous test beyond regular expressions may fail.

1. Add $\cap$ to the regular expression operators.
2. Test if $M \cap N \cap P = M \cap N$.
3. From $M = \boldsymbol{a}$, $N = \boldsymbol{b}$ and $P = \boldsymbol{c}$ and since

   $$\{a\} \cap \{b\} \cap \{c\} = \emptyset = \{a\} \cap \{b\},$$

   we should conclude that the 'property' is true.
4. But, the 'property' is false. For example, if $M = N = \boldsymbol{a}$ and $P = \emptyset$ then

   $$M \cap N \cap P \neq M \cap N.$$

5. Therefore, the test is not valid!

References

- Hopcroft, J. E., Motwani, R. and Ullman, J. D. [1979] (2007). Introduction to Automata Theory, Languages, and Computation. 3rd ed. Pearson Education (cit. on p. 2).
- Kozen, D. C. [1997] (2012). Automata and Computability. Third printing. Undergraduate Texts in Computer Science. Springer. DOI: 10.1007/978-1-4612-1844-9 (cit. on p. 10).