Automata and Formal Languages - CM0081
Algebraic Laws for Regular Expressions

Andrés Sicard-Ramírez

Universidad EAFIT

Semester 2018-2
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.
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 $L$, $M$ and $N$ 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 $L$, $M$ and $N$ be regular expression variables.

Sugar syntax

$$L^+ \overset{\text{def}}{=} LL^*,$$

$$L? \overset{\text{def}}{=} \varepsilon + L.$$
Algebraic Laws for Regular Expressions

Some laws for union

\[(L + M) + N = L + (M + N)\]  \hspace{1cm} \text{(associativity)}

\[L + \emptyset = \emptyset + L = L\]  \hspace{1cm} \text{(identity)}

\[L + M = M + L\]  \hspace{1cm} \text{(commutativity)}

\[L + L = L\]  \hspace{1cm} \text{(idempotence)}

Remark

There is no inverse for union.
Algebraic Laws for Regular Expressions

Some laws for concatenation

\[(LM)N = L(MN)\] (associativity)

\[L\varepsilon = \varepsilon L = L\] (identity)

\[LM \neq ML\] (non-commutativity)

\[L\emptyset = \emptyset L = \emptyset\] (\(\emptyset\) is the annihilator for concatenation)

Remark

There is no inverse for concatenation.
Algebraic Laws for Regular Expressions

Some laws for union and concatenation

\[ L(M + N) = LM + LN \]  \hspace{1cm} \text{(distributive)}

\[ (L + M)N = LN + LM \]  \hspace{1cm} \text{(distributive)}
Algebraic Laws for Regular Expressions

Some laws for closure

\[(L^*)^* = L^*\]  \hspace{2cm} (idempotence)
\[\emptyset^* = \varepsilon\]
\[\varepsilon^* = \varepsilon\]
\[(\varepsilon + L)^* = L^*\]
\[L^* = L^+ + \varepsilon\]
Example

\[0 + (\varepsilon + 1)(\varepsilon + 1)^*0 = 0 + (\varepsilon + 1)1^*0\]
\[= 0 + (\varepsilon1^* + 11^*)0\]
\[= 0 + (1^* + 11^*)0\]
\[= 0 + (1^* + 1^+)0\]
\[= 0 + 1^*0\]
\[= 1^*0\]
Method
Let \( E \) and \( F \) be two regular expressions with the same set of variables \( \{L_1, \ldots, L_n\} \).

To test if \( E = F \):

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

Let $E$ and $F$ be two regular expressions with the same set of variables $\{L_1, \ldots, L_n\}$.

To test if $E = F$:

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

Observation

We are proving by example!
Example
Prove or disprove that $L^* = L^*L^*$. 
Example

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

1. We replace the variable $L$ by the concrete regular expression $a$. 
Example

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

1. We replace the variable $L$ by the concrete regular expression $a$.
2. $a^* \neq a^*a^*$. 
Discovering Laws for Regular Expressions

Example

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

1. We replace the variable $L$ by the concrete regular expression $a$.
2. $a^* \equiv a^*a^*$.
3. Because $L(a^*) = L(a^*a^*)$, we conclude that $L^* = L^*L^*$. 
Example

Prove or disprove that $L + ML = (L + M)L$. 

1. We replace the variables $L$ and $M$ by the concrete regular expressions $a$ and $b$ respectively.

2. $a + ba \not\equiv (a + b)a$.

3. $aa \not\in L(a + ba)$ and $aa \in L((a + b)a)$, which implies $L(a + ba) \neq L((a + b)a)$, therefore $L + ML \neq (L + M)L$. 
Example

Prove or disprove that $L + ML = (L + M)L$.

1. We replace the variables $L$ and $M$ by the concrete regular expressions $a$ and $b$ respectively.
Discovering Laws for Regular Expressions

Example
Prove or disprove that $L + ML = (L + M)L$.

1. We replace the variables $L$ and $M$ by the concrete regular expressions $a$ and $b$ respectively.
2. $a + ba \neq (a + b)a$. 
Discovering Laws for Regular Expressions

Example

Prove or disprove that $L + ML = (L + M)L$.

1. We replace the variables $L$ and $M$ by the concrete regular expressions $a$ and $b$ respectively.
2. $a + ba \not\equiv (a + b)a$.
3. $aa \not\in L(a + ba)$ and $aa \in L((a + b)a)$
   \[ \Rightarrow L(a + ba) \neq L((a + b)a) \]
   \[ \Rightarrow L + ML \neq (L + M)L \]
Example (Exercise 3.4.2.d)

Prove or disprove that \((L + M)^* M = (L^* M)^*\).
Example (Exercise 3.4.2.d)

Prove or disprove that \((L + M)^* M = (L^* M)^*\).

1. We replace the variables \(L\) and \(M\) by the concrete regular expressions \(a\) and \(b\) respectively.
Example (Exercise 3.4.2.d)

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

1. We replace the variables $L$ and $M$ by the concrete regular expressions $a$ and $b$ respectively.

2. $(a + b)^* b = (a^* b)^*$. 
Example (Exercise 3.4.2.d)

Prove or disprove that \((L + M)^*M = (L^*M)^*\).

1. We replace the variables \(L\) and \(M\) by the concrete regular expressions \(a\) and \(b\) respectively.

2. \((a + b)^*b \neq (a^*b)^*\).

3. Since \(\varepsilon \notin (a + b)^*b\) and \(\varepsilon \in (a^*b)^*\) 

\[ \Rightarrow (L + M)^*M \neq (L^*M)^* \]
Example (counter-example)

Extensions of the previous test beyond regular expressions may fail.

1. Add $\cap$ to the regular expression operators.
2. Test if $L \cap M \cap N = L \cap M$.
3. From $L = a$, $M = b$, and $N = c$ 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 $L = M = a$ and $N = \emptyset$ then $L \cap M \cap N \neq L \cap M$.
5. Therefore, the test is not valid!
Example (counter-example)

Extensions of the previous test beyond regular expressions may fail.

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

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

we should conclude that the “property” is true.

But, the “property” is false. For example, if $L = M = a$ and $N = \emptyset$ then $L \cap M \cap N \neq L \cap M$.

Therefore, the test is not valid!
Example (counter-example)

Extensions of the previous test beyond regular expressions may fail.

1. Add $\cap$ to the regular expression operators.
2. Test if $L \cap M \cap N = L \cap M$.
3. From $L = a$, $M = b$ and $N = c$ 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 $L = M = a$ and $N = \emptyset$ then

$$L \cap M \cap N \neq L \cap M.$$

5. Therefore, the test is not valid!
Derivative of a Regular Expression

From [Brzozowski 1964].

Definition

Let $L \subseteq \Sigma^*$ be a language and $a \in \Sigma$ a symbol. We define the derivative of $L$ by $a$, denoted by $a\backslash L$, by

$$a\backslash L = \{x \in \Sigma^* \mid ax \in L\}.$$
Derivative of a Regular Expression

From [Brzozowski 1964].

Definition
Let \( L \subseteq \Sigma^* \) be a language and \( a \in \Sigma \) a symbol. We define the derivative of \( L \) by \( a \), denoted by \( a \setminus L \), by

\[
a \setminus L = \{ x \in \Sigma^* \mid ax \in L \}.
\]

Examples

\[
a \setminus \{abab, abba\} = \{bab, bba\},
\]
\[
a \setminus L(ab^*) = L(b^*),
\]
\[
b \setminus L(ab^*) = \emptyset.
\]
Derivative of a Regular Expression

Definition
Let $E$ be a regular expression and $a \in \Sigma$ a symbol. We define recursively the **derivative** of $E$ by $a$, denoted $a\backslash E$, by

- $a\backslash \emptyset = \emptyset$,
- $a\backslash \varepsilon = \emptyset$,
- $a\backslash a = \varepsilon$,
- $a\backslash b = \emptyset$ for $a \neq b$, 

Algebraic Laws for Regular Expressions
Derivative of a Regular Expression

Definition

Let $E$ be a regular expression and $a \in \Sigma$ a symbol. We define recursively the derivative of $E$ by $a$, denoted $a \backslash E$, by

\[
\begin{align*}
    a \backslash \emptyset &= \emptyset, \\
    a \backslash \varepsilon &= \emptyset, \\
    a \backslash a &= \varepsilon, \\
    a \backslash b &= \emptyset \text{ for } a \neq b, \\
    a \backslash (E + F) &= a \backslash E + a \backslash F, \\
    a \backslash (EF) &= \begin{cases} 
        (a \backslash E)F + a \backslash F, & \text{if } \varepsilon \in L(E), \\
        (a \backslash E)F, & \text{otherwise},
    \end{cases} \\
    a \backslash (E^*) &= (a \backslash E)E^*.
\end{align*}
\]
Derivative of a Regular Expression

Definition

Let $E$ be a regular expression and $w \in \Sigma^*$ a string. We define recursively the derivative of $E$ by $w$, denoted $w\backslash E$, by

$$
\begin{align*}
\varepsilon\backslash E &= E, \\
a x\backslash E &= a\backslash(x\backslash E).
\end{align*}
$$
Derivative of a Regular Expression

Definition

Let $E$ be a regular expression and $w \in \Sigma^*$ a string. We define recursively the derivative of $E$ by $w$, denoted $w \backslash E$, by

\[
\varepsilon \backslash E = E, \\
a \cdot x \backslash E = a \cdot (x \backslash E).
\]

Theorem (Brzozowski [1964], Theorem 4.2)

\[w \in L(E) \iff \varepsilon \in L(w \backslash E).\]