Computability and Complexity
COSC 4200

Regular Expressions

Regular Expressions

A regular expression is a way of defining a language. For example:

- 10^*
  Strings that begin with a 1 and end in any number of 0's.
- $(0 \cup 1)^*01$
  Binary strings that end with 01.

Examples of regular expressions

ϵ

0

011

$(01 \cup 1)$

$(0 \cup 01)^*$

$(01^* \cup 0^*1)^*$

$0(0 \cup 1)^*1$

Note: $\cup$ is the same as $|$

Regular Expressions

A regular expression is built from alphabet symbols, ϵ, $\emptyset$, $\cup$, $\cdot$, $*$, and parentheses. Defined formally using induction:

Regular Expressions

A regular expression is built from alphabet symbols, ϵ, $\emptyset$, $\cup$, $\cdot$, $*$, and parentheses. Defined formally using induction:

Definition

Let Σ be an alphabet. R is a *regular expression* if R is

1. a for some $a \in \Sigma$
2. ϵ
3. $\emptyset$

Regular Expressions

A regular expression is built from alphabet symbols, ϵ, $\emptyset$, $\cup$, $\cdot$, $*$, and parentheses. Defined formally using induction:

Definition

Let Σ be an alphabet. R is a *regular expression* if R is

1. a for some $a \in \Sigma$
2. ϵ
3. $\emptyset$
4. $(R_1 \cup R_2)$ where R_1 and R_2 are regular expressions

Regular Expressions

A regular expression is built from alphabet symbols, ϵ, $\emptyset$, $\cup$, $\cdot$, $*$, and parentheses. Defined formally using induction:

Definition

Let Σ be an alphabet. R is a *regular expression* if R is

1. a for some $a \in \Sigma$
2. ϵ
3. $\emptyset$
4. $(R_1 \cup R_2)$ where R_1 and R_2 are regular expressions
5. $(R_1 \cdot R_2)$ where R_1 and R_2 are regular expressions

Regular Expressions

A regular expression is built from alphabet symbols, ϵ, $\emptyset$, $\cup$, $\cdot$, $*$, and parentheses. Defined formally using induction:

Definition

Let Σ be an alphabet. R is a *regular expression* if R is

1. a for some $a \in \Sigma$
2. ϵ
3. $\emptyset$
4. $(R_1 \cup R_2)$ where R_1 and R_2 are regular expressions
5. $(R_1 \cdot R_2)$ where R_1 and R_2 are regular expressions
6. (R_1^*) where R_1 is a regular expressions

Regular Expressions

A regular expression is built from alphabet symbols, ϵ, $\emptyset$, $\cup$, $\cdot$, $*$, and parentheses. Defined formally using induction:

Definition

Let Σ be an alphabet. R is a *regular expression* if R is

1. a for some $a \in \Sigma$
2. ϵ
3. $\emptyset$
4. $(R_1 \cup R_2)$ where R_1 and R_2 are regular expressions
5. $(R_1 \cdot R_2)$ where R_1 and R_2 are regular expressions
6. (R_1^*) where R_1 is a regular expressions

Often the parentheses and concatenation are omitted, if the meaning is clear.

Language of a Regular Expression

Definition

Let R be a regular expression. The *language of R* (or the *language described by R*), $L(R)$, is defined as follows.

1. If $R = a$ for some $a \in \Sigma$, then $L(R) = \{a\}$.
2. If $R = \epsilon$, then $L(R) = \{\epsilon\}$.
3. If $R = \emptyset$, then $L(R) = \emptyset$.

Language of a Regular Expression

Definition

Let R be a regular expression. The *language of* R (or the *language described by* R), $L(R)$, is defined as follows.

1. If $R = a$ for some $a \in \Sigma$, then $L(R) = \{a\}$.
2. If $R = \epsilon$, then $L(R) = \{\epsilon\}$.
3. If $R = \emptyset$, then $L(R) = \emptyset$.
4. If $R = (R_1 \cup R_2)$ for two regular expressions R_1 and R_2, then $L(R) = L(R_1) \cup L(R_2)$.

Language of a Regular Expression

Definition

Let R be a regular expression. The *language of* R (or the *language described by* R), $L(R)$, is defined as follows.

1. If $R = a$ for some $a \in \Sigma$, then $L(R) = \{a\}$.
2. If $R = \epsilon$, then $L(R) = \{\epsilon\}$.
3. If $R = \emptyset$, then $L(R) = \emptyset$.
4. If $R = (R_1 \cup R_2)$ for two regular expressions R_1 and R_2, then $L(R) = L(R_1) \cup L(R_2)$.
5. If $R = (R_1 \cdot R_2)$ for two regular expressions R_1 and R_2, then $L(R) = L(R_1) \cdot L(R_2)$.

Language of a Regular Expression

Definition

Let R be a regular expression. The *language of* R (or the *language described by* R), $L(R)$, is defined as follows.

1. If $R = a$ for some $a \in \Sigma$, then $L(R) = \{a\}$.
2. If $R = \epsilon$, then $L(R) = \{\epsilon\}$.
3. If $R = \emptyset$, then $L(R) = \emptyset$.
4. If $R = (R_1 \cup R_2)$ for two regular expressions R_1 and R_2, then $L(R) = L(R_1) \cup L(R_2)$.
5. If $R = (R_1 \cdot R_2)$ for two regular expressions R_1 and R_2, then $L(R) = L(R_1) \cdot L(R_2)$.
6. If $R = (R_1^*)$ for some regular expression R_1, then $L(R) = [L(R_1)]^*$.

Examples

- $L(0) =$

Examples

- $L(0) = \{0\}$

Examples

- $L(0) = \{0\}$
- $L(00 \cup 01) =$

Examples

- $L(0) = \{0\}$
- $L(00 \cup 01) = \{00, 01\}$

Examples

- $L(0) = \{0\}$
- $L(00 \cup 01) = \{00, 01\}$
- $L(0^*) =$

Examples

- $L(0) = \{0\}$
- $L(00 \cup 01) = \{00, 01\}$
- $L(0^*) = \{0^n \mid n \geq 0\}$

Examples

- $L(0) = \{0\}$
- $L(00 \cup 01) = \{00, 01\}$
- $L(0^*) = \{0^n \mid n \geq 0\}$
- $L((0 \cup 1)^*) =$

Examples

- $L(0) = \{0\}$
- $L(00 \cup 01) = \{00, 01\}$
- $L(0^*) = \{0^n \mid n \geq 0\}$
- $L((0 \cup 1)^*) = \{0, 1\}^*$

Examples

- $L(0) = \{0\}$
- $L(00 \cup 01) = \{00, 01\}$
- $L(0^*) = \{0^n \mid n \geq 0\}$
- $L((0 \cup 1)^*) = \{0, 1\}^*$
- $L(00(0 \cup 1)^*11) =$

Examples

- $L(0) = \{0\}$
- $L(00 \cup 01) = \{00, 01\}$
- $L(0^*) = \{0^n \mid n \geq 0\}$
- $L((0 \cup 1)^*) = \{0, 1\}^*$
- $L(00(0 \cup 1)^*11) = \left\{ x \in \{0,1\}^* \;\middle|\; \begin{array}{l} x \text{ begins with } 00 \\ \text{and ends with } 11 \end{array} \right\}$

Examples

- $L(0) = \{0\}$
- $L(00 \cup 01) = \{00, 01\}$
- $L(0^*) = \{0^n \mid n \geq 0\}$
- $L((0 \cup 1)^*) = \{0, 1\}^*$
- $L(00(0 \cup 1)^*11) = \left\{ x \in \{0,1\}^* \;\middle|\; \begin{array}{l} x \text{ begins with } 00 \\ \text{and ends with } 11 \end{array} \right\}$
- $L((0 \cup 1)^*010(0 \cup 1)^*) =$

Examples

- $L(0) = \{0\}$
- $L(00 \cup 01) = \{00, 01\}$
- $L(0^*) = \{0^n \mid n \geq 0\}$
- $L((0 \cup 1)^*) = \{0, 1\}^*$
- $L(00(0 \cup 1)^*11) = \left\{ x \in \{0,1\}^* \;\middle|\; \begin{array}{l} x \text{ begins with } 00 \\ \text{and ends with } 11 \end{array} \right\}$
- $L((0 \cup 1)^*010(0 \cup 1)^*) = \{x \in \{0,1\}^* \mid x \text{ contains } 010\}$

Conventions. In a regular expression, Σ is used as an abbreviation for the $\cup$ over all symbols in Σ. For example, if $\Sigma = \{0, 1\}$, then Σ means $(0 \cup 1)$ in a regular expression.

e.g. can write $\Sigma^*010\Sigma^*$ instead of $(0 \cup 1)^*010(0 \cup 1)^*$

Conventions. In a regular expression, Σ is used as an abbreviation for the $\cup$ over all symbols in Σ. For example, if $\Sigma = \{0, 1\}$, then Σ means $(0 \cup 1)$ in a regular expression.

e.g. can write $\Sigma^* 010 \Sigma^*$ instead of $(0 \cup 1)^* 010 (0 \cup 1)^*$

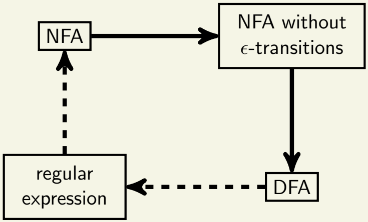
Often, we just write R when we formally mean $L(R)$. For example:

$$0^* = \{0^n \mid n \geq 0\}$$

$$0^* 1 0^* = \{w \in \{0,1\}^* \mid w \text{ has exactly one } 1\}$$

Theorem

A language is regular if and only if it can be described by a regular expression.



We've already shown conversion procedures for the solid arrows.
We now need to show the conversions for the dashed arrows.

Now: regular expression $\rightarrow$ NFA.

Lemma

If a language is described by a regular expression, then it is regular (there is an NFA that accepts it).

Proof. For any regular expression R, we will show that $L(R)$ is regular by explaining how to construct an NFA N with $L(N) = L(R)$. The proof is by induction on the structure of regular expressions.

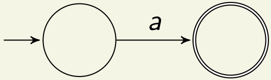
Lemma

If a language is described by a regular expression, then it is regular (there is an NFA that accepts it).

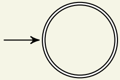
Proof. For any regular expression R, we will show that $L(R)$ is regular by explaining how to construct an NFA N with $L(N) = L(R)$. The proof is by induction on the structure of regular expressions.

We have three base cases:

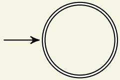
1. $R = a$ for some $a \in \Sigma$. Then $L(R) = \{a\}$, and the following NFA accepts $L(R)$.



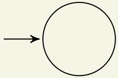
2. $R = \epsilon$. Then $L(R) = \{\epsilon\}$, and the following NFA accepts $L(R)$.



2. $R = \epsilon$. Then $L(R) = \{\epsilon\}$, and the following NFA accepts $L(R)$.



3. $R = \emptyset$. Then $L(R) = \emptyset$, and the following NFA accepts $L(R)$.



Assume that R_1 and R_2 are regular expressions that have been shown regular, and N_1 and N_2 are NFAs with $L(N_1) = L(R_1)$ and $L(N_2) = L(R_2)$. (This is the inductive hypothesis.) We have three inductive steps.

Assume that R_1 and R_2 are regular expressions that have been shown regular, and N_1 and N_2 are NFAs with $L(N_1) = L(R_1)$ and $L(N_2) = L(R_2)$. (This is the inductive hypothesis.) We have three inductive steps.

4. $R = R_1 \cup R_2$. Then $L(R) = L(R_1) \cup L(R_2)$ by definition, so $L(R)$ is also regular because the regular languages are closed under union. Specifically, we may use the NFA union construction to obtain an NFA N with $L(N) = L(N_1) \cup L(N_2) = L(R_1) \cup L(R_2) = L(R)$.

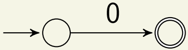
Assume that R_1 and R_2 are regular expressions that have been shown regular, and N_1 and N_2 are NFAs with $L(N_1) = L(R_1)$ and $L(N_2) = L(R_2)$. (This is the inductive hypothesis.) We have three inductive steps.

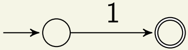
4. $R = R_1 \cup R_2$. Then $L(R) = L(R_1) \cup L(R_2)$ by definition, so $L(R)$ is also regular because the regular languages are closed under union. Specifically, we may use the NFA union construction to obtain an NFA N with $L(N) = L(N_1) \cup L(N_2) = L(R_1) \cup L(R_2) = L(R)$.
5. $R = R_1 \cdot R_2$. Then $L(R) = L(R_1) \cdot L(R_2)$ by definition, so $L(R)$ is regular by closure under concatenation. We may use the concatenation construction to obtain an NFA N with $L(N) = L(N_1) \cdot L(N_2) = L(R_1) \cdot L(R_2) = L(R)$.

Assume that R_1 and R_2 are regular expressions that have been shown regular, and N_1 and N_2 are NFAs with $L(N_1) = L(R_1)$ and $L(N_2) = L(R_2)$. (This is the inductive hypothesis.) We have three inductive steps.

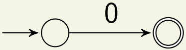
4. $R = R_1 \cup R_2$. Then $L(R) = L(R_1) \cup L(R_2)$ by definition, so $L(R)$ is also regular because the regular languages are closed under union. Specifically, we may use the NFA union construction to obtain an NFA N with $L(N) = L(N_1) \cup L(N_2) = L(R_1) \cup L(R_2) = L(R)$.
5. $R = R_1 \cdot R_2$. Then $L(R) = L(R_1) \cdot L(R_2)$ by definition, so $L(R)$ is regular by closure under concatenation. We may use the concatenation construction to obtain an NFA N with $L(N) = L(N_1) \cdot L(N_2) = L(R_1) \cdot L(R_2) = L(R)$.
6. $R = R_1^*$. Then $L(R) = L(R_1)^*$ by definition, so $L(R)$ is regular by closure under the star operation. We may use the star operation to obtain an NFA N with $L(N) = L(N_1)^* = L(R_1)^* = L(R)$. $\square$

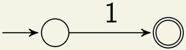
Example: Convert $(01 \cup 10^*)^*$ to an NFA

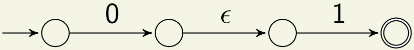
0: 

1: 

Example: Convert $(01 \cup 10^*)^*$ to an NFA

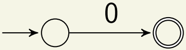
0: 

1: 

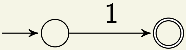
01: 

Example: Convert $(01 \cup 10^*)^*$ to an NFA

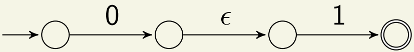
0:



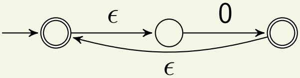
1:



01:

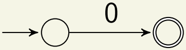


0^*:

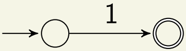


Example: Convert $(01 \cup 10^*)^*$ to an NFA

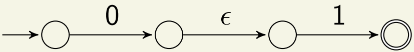
0:



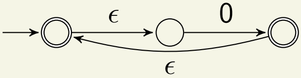
1:



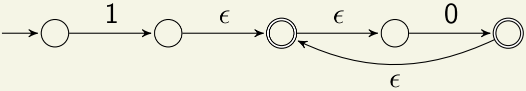
01:



0^*:

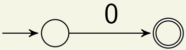


10^*:

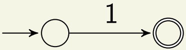


Example: Convert $(01 \cup 10^*)^*$ to an NFA

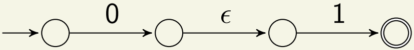
0:



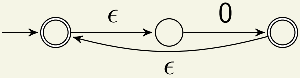
1:



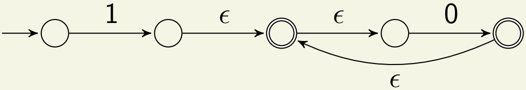
01:



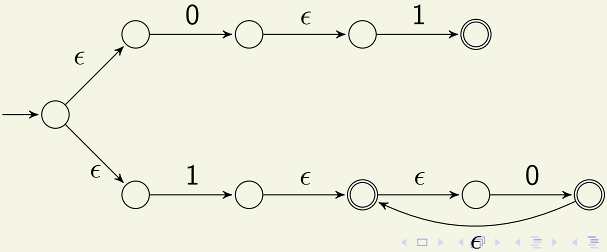
0^*:



10^*:

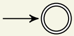


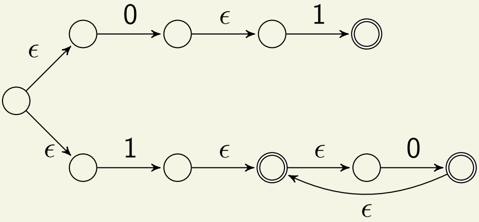
$01 \cup 10^*$:



Example: Convert $(01 \cup 10^*)^*$ to an NFA

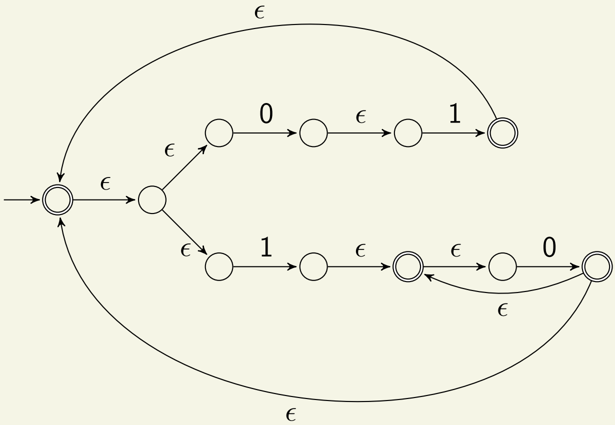
$(01 \cup 10^*)^*$:



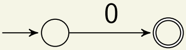


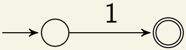
Example: Convert $(01 \cup 10^*)^*$ to an NFA

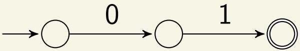
$(01 \cup 10^*)^*$:



$(01 \cup 10^*)^*$ - Simpler NFA

0: 

1: 

01: 

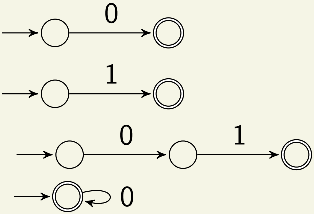
$(01 \cup 10^*)^*$ - Simpler NFA

0:

1:

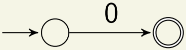
01:

0^*:

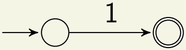


$(01 \cup 10^*)^*$ - Simpler NFA

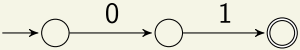
0:



1:

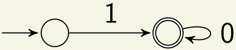


01:



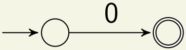
0^*:

10^*:

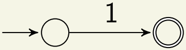


$(01 \cup 10^*)^*$ - Simpler NFA

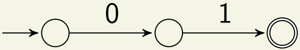
0:



1:

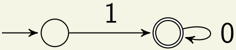


01:

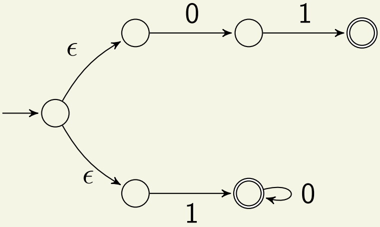


0^*:

10^*:

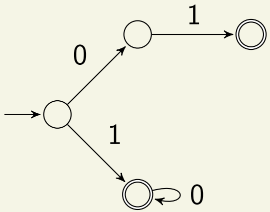


$01 \cup 10^*$:



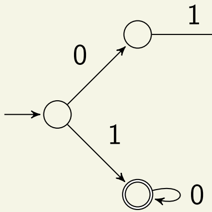
$(01 \cup 10^*)^*$ - Simpler NFA

$01 \cup 10^*$:

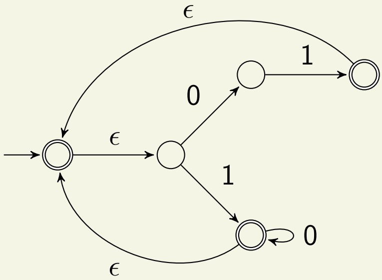


$(01 \cup 10^*)^*$ - Simpler NFA

$01 \cup 10^*$:

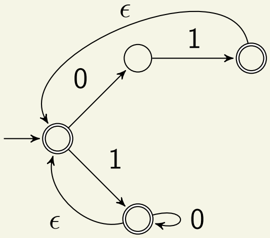


$(01 \cup 10^*)^*$:



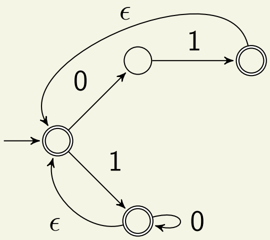
$(01 \cup 10^*)^*$ - Simpler NFA

$(01 \cup 10^*)^*$:

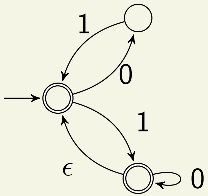


$(01 \cup 10^*)^*$ - Simpler NFA

$(01 \cup 10^*)^*$:

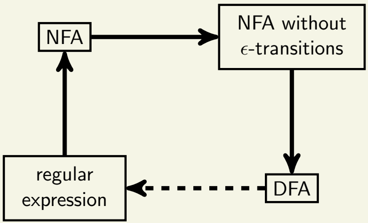


$(01 \cup 10^*)^*$:



Theorem

A language is regular if and only if it can be described by a regular expression.



Next: DFA $\rightarrow$ regular expression.