## Formal Languages, Automata and

 Codes
## Oleg Gutik



## Lecture 11

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general.
Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures. Closure under Simple Set Operations We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general.
Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures. Closure under Simple Set Operations We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general.
Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures. Closure under Simple Set Operations We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general
Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures. Closure under Simple Set Operations We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general.
$L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the
family of regular languages is closed under union. We can ask similar questions
about other types of operations on languages; this leads us to the study of the closure properties of languages in general.

Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures. Closure under Simple Set Operations We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general.
Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures. Closure under Simple Set Operations We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, ,
family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general.
Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures. Closure under Simple Set Operations We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union.
closure properties of languages in general.
Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures.
Closure under Simple Set Operations
We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union. We can ask similar questions about other types of operations on languages;

Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures.

Closure under Simple Set Operations
We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general.

> Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures.

> Closure under Simple Set Operations
> We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general.
Closure properties of various language families under different operations are of considerable theoretical interest.
practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures.

Closure under Simple Set Operations
We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general.
Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have.
very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures.

Closure under Simple Set Operations
We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general.
Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful.
nature of language families, closure properties help us answer other, more
practical questions. We shall see instances of this later in this couse of lectures.
Closure under Simple Set Operations
We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general.
Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families,
practical questions. We shall see instances of this later in this couse of lectures.
Closure under Simple Set Operations
We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general.
Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions.
Closure under Simple Set Operations
We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general.
Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures.
Closure under Simple Set Operations
We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general.
Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures.
Closure under Simple Set Operations
We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

Consider the following question: Given two regular languages $L_{1}$ and $L_{2}$, is their union also regular? In specific instances, the answer may be obvious, but here we want to address the problem in general. Is it true for all regular $L_{1}$ and $L_{2}$ ? It turns out that the answer is yes, a fact we express by saying that the family of regular languages is closed under union. We can ask similar questions about other types of operations on languages; this leads us to the study of the closure properties of languages in general.
Closure properties of various language families under different operations are of considerable theoretical interest. At first sight, it may not be clear what practical significance these properties have. Admittedly, some of them have very little, but many results are useful. By giving us insight into the general nature of language families, closure properties help us answer other, more practical questions. We shall see instances of this later in this couse of lectures.

## Closure under Simple Set Operations

We begin by looking at the closure of regular languages under the common set operations, such as union and intersection.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.1

Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that accepts $L_{1}$. Then the dfa

$$
\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, Q-F\right)
$$

accepts $\overline{L_{1}}$. This is rather straightforward. Note that in the definition of a dfa, we assumed $\delta^{*}$ to be a total function, so that $\delta^{*}\left(q_{0}, w\right)$ is defined for all $w \in \Sigma^{*}$. Consequently either $\delta^{*}\left(q_{0}, w\right)$ is a final state, in which case $w \in L$, or $\delta^{*}\left(q_{0}, w\right) \in Q-F$ and $w \in L$.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that accepts $L_{1}$. Then the dfa

$$
\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, Q-F\right)
$$

accepts $\overline{L_{1}}$. This is rather straightforward. Note that in the definition of a dfa, we assumed $\delta^{*}$ to be a total function, so that $\delta^{*}\left(q_{0}, w\right)$ is defined for all $w \in \Sigma^{*}$. Consequently either $\delta^{*}\left(q_{0}, w\right)$ is a final state, in which case $w \in L$, or $\delta^{*}\left(q_{0}, w\right) \in Q-F$ and $w \in L$.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that accepts $L_{1}$. Then the dfa

$$
\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, Q-F\right)
$$

accepts $\overline{L_{1}}$. This is rather straightforward. Note that in the definition of a dfa, we assumed $\delta^{*}$ to be a total function, so that $\delta^{*}\left(q_{0}, w\right)$ is defined for all $w \in \Sigma^{*}$. Consequently either $\delta^{*}\left(q_{0}, w\right)$ is a final state, in which case $w \in L$, or $\delta^{*}\left(q_{0}, w\right) \in Q-F$ and $w \in L$.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

> Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.

> To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that accepts $L_{1}$. Then the dfa
> accepts $\overline{L_{1}}$. This is rather straightforward. Note that in the definition of a dfa, we assumed $\delta^{*}$ to be a total function, so that $\delta^{*}\left(q_{0}, w\right)$ is defined for all $w \in \Sigma^{*}$. Consequently either $\delta^{*}\left(q_{0}, w\right)$ is a final state, in which case $w \in L$, or $\delta^{*}\left(q_{0}, w\right) \in Q-F$ and $w \in L$.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

## Proof.

```
such that L}\mp@subsup{L}{1}{}=L(\mp@subsup{r}{1}{})\mathrm{ and L}\mp@subsup{L}{2}{}=L(\mp@subsup{r}{2}{})\mathrm{ . By definition, r
```

regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively.
Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that
accepts $L_{1}$. Then the dfa
accepts $\overline{L_{1}}$. This is rather straightforward. Note that in the definition of a dfa,
we assumed $\delta^{*}$ to be a total function, so that $\delta^{*}\left(q_{0}, w\right)$ is defined for all
$w \in \Sigma^{*}$. Consequently either $\delta^{*}\left(q_{0}, w\right)$ is a final state, in which case $w \in L$, or
$\delta^{*}\left(q_{0}, w\right) \in Q-F$ and $w \in L$.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof. If $L_{1}$ and $L_{2}$ are regular, such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that accepts $L_{1}$. Then the dfa
accepts $\overline{L_{1}}$. This is rather straightforward. Note that in the definition of a dfa, we assumed $\delta^{*}$ to be a total function, so that $\delta^{*}\left(q_{0}, w\right)$ is defined for all $w \in \Sigma^{*}$. Consequently either $\delta^{*}\left(q_{0}, w\right)$ is a final state, in which case $w \in L$, or $\delta^{*}\left(q_{0}, w\right) \in Q-F$ and $w \in L$.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$.

```
regular expressions denoting the languages }\mp@subsup{L}{1}{}\cup\mp@subsup{L}{2}{},\mp@subsup{L}{1}{}\mp@subsup{L}{2}{}\mathrm{ , and }\mp@subsup{L}{1}{*}\mathrm{ , respectively.
Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let M = (Q, \Sigma, \delta, q}, F) be a dfa that
accepts }\mp@subsup{L}{1}{}\mathrm{ . Then the dfa
accepts }\overline{\mp@subsup{L}{1}{}}\mathrm{ . This is rather straightforward. Note that in the definition of a dfa,
we assumed }\mp@subsup{\delta}{}{*}\mathrm{ to be a total function, so that }\mp@subsup{\delta}{}{*}(\mp@subsup{q}{0}{},w)\mathrm{ is defined for all
w\in\mp@subsup{\Sigma}{}{*}\mathrm{ . Consequently either }\mp@subsup{\delta}{}{*}(\mp@subsup{q}{0}{},w)\mathrm{ is a final state, in which case }w\inL\mathrm{ , or}
\delta*}(\mp@subsup{q}{0}{},w)\inQ-F and w\inL
```


### 4.1 Closure Properties of Regular Languages

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that accepts $L_{1}$. Then the dfa
accepts $\overline{L_{1}}$. This is rather straightforward. Note that in the definition of a dfa, we assumed $\delta^{*}$ to be a total function, so that $\delta^{*}\left(q_{0}, w\right)$ is defined for all $w \in \Sigma^{*}$. Consequently either $\delta^{*}\left(q_{0}, w\right)$ is a final state, in which case $w \in L$, or $\delta^{*}\left(q_{0}, w\right) \in Q-F$ and $w \in L$.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that accepts $L_{1}$.
accepts $\overline{L_{1}}$. This is rather straightforward. Note that in the definition of a dfa, we assumed $\delta^{*}$ to be a total function, so that $\delta^{*}\left(q_{0}, w\right)$ is defined for all $w \in \Sigma^{*}$. Consequently either $\delta^{*}\left(q_{0}, w\right)$ is a final state, in which case $w \in L$, or $\delta^{*}\left(q_{0}, w\right) \in Q-F$ and $w \in L$.

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that accepts $L_{1}$. Then the dfa

accepts $\overline{L_{1}}$. This is rather straightforward. Note that in the definition of a dfa, we assumed $\delta^{*}$ to be a total function, so that $\delta^{*}\left(q_{0}, w\right)$ is defined for all $w \in \Sigma^{*}$. Consequently either $\delta^{*}\left(q_{0}, w\right)$ is a final state, in which case $w \in L$, or $\delta^{*}\left(q_{0}, w\right) \in Q-F$ and $w \in L$.

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that accepts $L_{1}$. Then the dfa

$$
\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, Q-F\right)
$$

accepts $\overline{L_{1}}$. This is rather straightforward. Note that in the definition of a dfa, we assumed $\delta^{*}$ to be a total function, so that $\delta^{*}\left(q_{0}, w\right)$ is defined for all $w \in \Sigma^{*}$. Consequently either $\delta^{*}\left(q_{0}, w\right)$ is a final state, in which case $w \in L$, or $\delta^{*}\left(q_{0}, w\right) \in Q-F$ and $w \in L$.

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that accepts $L_{1}$. Then the dfa

$$
\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, Q-F\right)
$$

accepts $\overline{L_{1}}$. This is rather straightforward. Note that in the definition of a dfa, we assumed $\delta^{*}$ to be a total function, so that $\delta^{*}\left(q_{0}, w\right)$ is defined for all $w \in \Sigma^{*}$. Consequently either $\delta^{*}\left(q_{0}, w\right)$ is a final state, in which case $w \in L$, or $\delta^{*}\left(q_{0}, w\right) \in Q-F$ and $w \in L$.

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that accepts $L_{1}$. Then the dfa

$$
\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, Q-F\right)
$$

accepts $\overline{L_{1}}$. This is rather straightforward. we assumed $\delta^{*}$ to be a total function, so that $\delta^{*}\left(q_{0}, w\right)$ is defined for all $w \in \Sigma^{*}$. Consequently either $\delta^{*}\left(q_{0}, w\right)$ is a final state, in which case $w \in L$, or $\delta^{*}\left(q_{0}, w\right) \in Q-F$ and $w \in L$.

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that accepts $L_{1}$. Then the dfa

$$
\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, Q-F\right)
$$

accepts $\overline{L_{1}}$. This is rather straightforward. Note that in the definition of a dfa, we assumed $\delta^{*}$ to be a total function,

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that accepts $L_{1}$. Then the dfa

$$
\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, Q-F\right)
$$

accepts $\overline{L_{1}}$. This is rather straightforward. Note that in the definition of a dfa, we assumed $\delta^{*}$ to be a total function, so that $\delta^{*}\left(q_{0}, w\right)$ is defined for all $w \in \Sigma^{*}$.

## Theorem 4.1

If $L_{1}$ and $L_{2}$ are regular languages, then so are $L_{1} \cup L_{2}, L_{1} \cap L_{2}, L_{1} L_{2}, \overline{L_{1}}$, and $L_{1}^{*}$. We say that the family of regular languages is closed under union, intersection, concatenation, complementation, and star-closure.

Proof. If $L_{1}$ and $L_{2}$ are regular, then there exist regular expressions $r_{1}$ and $r_{2}$ such that $L_{1}=L\left(r_{1}\right)$ and $L_{2}=L\left(r_{2}\right)$. By definition, $r_{1}+r_{2}, r_{1} r_{2}$, and $r_{1}^{*}$ are regular expressions denoting the languages $L_{1} \cup L_{2}, L_{1} L_{2}$, and $L_{1}^{*}$, respectively. Thus, closure under union, concatenation, and star-closure is immediate.
To show closure under complementation, let $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ be a dfa that accepts $L_{1}$. Then the dfa

$$
\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, Q-F\right)
$$

accepts $\overline{L_{1}}$. This is rather straightforward. Note that in the definition of a dfa, we assumed $\delta^{*}$ to be a total function, so that $\delta^{*}\left(q_{0}, w\right)$ is defined for all $w \in \Sigma^{*}$. Consequently either $\delta^{*}\left(q_{0}, w\right)$ is a final state, in which case $w \in L$, or $\delta^{*}\left(q_{0}, w\right) \in Q-F$ and $w \in L$.

### 4.1 Closure Properties of Regular Languages

Demonstrating closure under intersection takes a little more work. Let
$L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and
$M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined
automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\widehat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$ whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking $\widehat{\delta}\left(\left(q_{i}, p_{j}\right), a\right)=\left(q_{k}, p_{l}\right)$,
whenever
and

$$
\delta_{1}\left(q_{i}, a\right)=q_{k}
$$

$$
\delta_{2}\left(p_{j}, a\right)=p_{l} .
$$

$\widehat{F}$ is defined as the set of all $\left(q_{i}, p j\right)$, such that $q_{i} \in F_{1}$ and $p_{j} \in F_{2}$. Then it is
a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$. Consequently, $L_{1} \cap L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

Demonstrating closure under intersection takes a little more work.
$L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and
$M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined


whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking
$\widehat{\delta}\left(\left(q_{i}, p_{j}\right), a\right)=\left(q_{k}, p_{l}\right)$,
whenever
and

$$
\delta_{1}\left(q_{i}, a\right)=q_{k}
$$

$\widehat{F}$ is defined as the set of all $\left(q_{i}, p j\right)$, such that $q_{i} \in F_{1}$ and $p_{j} \in F_{2}$. Then it is
a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$. Consequently, $L_{1} \cap L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$,


pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$
whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking
$\hat{\delta}\left(\left(q_{i}, p_{j}\right), a\right)=\left(q_{k}, p_{l}\right)$
whenever
and

$$
\delta_{1}\left(q_{i}, a\right)=q_{k}
$$

$\widehat{F}$ is defined as the set of all $\left(q_{i}, p j\right)$, such that $q_{i} \in F_{1}$ and $p_{j} \in F_{2}$. Then it is
a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$ Consequently, $L_{1} \cap L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

> Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\bar{M}=\left(\hat{Q}, \Sigma, \delta,\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\hat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\hat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$ whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking $$
\hat{\delta}\left(\left(q_{i}, p_{j}\right), a\right)=\left(q_{k}, p_{l}\right),
$$

## whenever

and
$\widehat{F}$ is defined as the set of all $\left(q_{i}, p j\right)$, such that $q_{i} \in F_{1}$ and $p_{j} \in F_{2}$. Then it is a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$. Consequently, $L_{1} \cap L_{2}$ is regular

### 4.1 Closure Properties of Regular Languages

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $Q=Q \times P$ consists of whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking whenever and
a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$. Consequently, $L_{1} \cap L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\widehat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$,
whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking
whenever
and
a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$. Consequently, $L_{1} \cap L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\widehat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$
whenever
and
a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$. Consequently, $L_{1} \cap L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\widehat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$ whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$.
whenever
and
a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$ Consequently, $L_{1} \cap L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\widehat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$ whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking
whenever
and
a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$ Consequently, $L_{1} \cap L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\widehat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$ whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking

$$
\widehat{\delta}\left(\left(q_{i}, p_{j}\right), a\right)=\left(q_{k}, p_{l}\right)
$$

whenever
and
a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$ Consequently, $L_{1} \cap L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\widehat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$ whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking

$$
\widehat{\delta}\left(\left(q_{i}, p_{j}\right), a\right)=\left(q_{k}, p_{l}\right)
$$

whenever
and
a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$ Consequently, $L_{1} \cap L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\widehat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$ whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking

$$
\hat{\delta}\left(\left(q_{i}, p_{j}\right), a\right)=\left(q_{k}, p_{l}\right)
$$

whenever

$$
\delta_{1}\left(q_{i}, a\right)=q_{k}
$$

a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$. Consequently, $L_{1} \cap L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\widehat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$ whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking

$$
\hat{\delta}\left(\left(q_{i}, p_{j}\right), a\right)=\left(q_{k}, p_{l}\right)
$$

whenever

$$
\delta_{1}\left(q_{i}, a\right)=q_{k}
$$

and
and

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\widehat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$ whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking

$$
\widehat{\delta}\left(\left(q_{i}, p_{j}\right), a\right)=\left(q_{k}, p_{l}\right)
$$

whenever
and

$$
\delta_{1}\left(q_{i}, a\right)=q_{k}
$$

$$
\delta_{2}\left(p_{j}, a\right)=p_{l}
$$

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\widehat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$ whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking

$$
\widehat{\delta}\left(\left(q_{i}, p_{j}\right), a\right)=\left(q_{k}, p_{l}\right)
$$

whenever
and

$$
\delta_{1}\left(q_{i}, a\right)=q_{k}
$$

$$
\delta_{2}\left(p_{j}, a\right)=p_{l}
$$

$\widehat{F}$ is defined as the set of all $\left(q_{i}, p j\right)$, such that $q_{i} \in F_{1}$ and $p_{j} \in F_{2}$.

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\widehat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$ whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking

$$
\widehat{\delta}\left(\left(q_{i}, p_{j}\right), a\right)=\left(q_{k}, p_{l}\right)
$$

whenever
and

$$
\delta_{1}\left(q_{i}, a\right)=q_{k}
$$

$$
\delta_{2}\left(p_{j}, a\right)=p_{l}
$$

$\widehat{F}$ is defined as the set of all $\left(q_{i}, p j\right)$, such that $q_{i} \in F_{1}$ and $p_{j} \in F_{2}$. Then it is a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$.

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\widehat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$ whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking

$$
\widehat{\delta}\left(\left(q_{i}, p_{j}\right), a\right)=\left(q_{k}, p_{l}\right)
$$

whenever
and

$$
\delta_{1}\left(q_{i}, a\right)=q_{k}
$$

$$
\delta_{2}\left(p_{j}, a\right)=p_{l}
$$

$\widehat{F}$ is defined as the set of all $\left(q_{i}, p j\right)$, such that $q_{i} \in F_{1}$ and $p_{j} \in F_{2}$. Then it is a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$. Consequently, $L_{1} \cap L_{2}$ is regular.

Demonstrating closure under intersection takes a little more work. Let $L_{1}=L\left(M_{1}\right)$ and $L_{2}=L\left(M_{2}\right)$, where $M_{1}=\left(Q, \Sigma, \delta_{1}, q_{0}, F_{1}\right)$ and $M_{2}=\left(P, \Sigma, \delta_{2}, p_{0}, F_{2}\right)$ are dfa's. We construct from $M_{1}$ and $M_{2}$ a combined automaton $\widehat{M}=\left(\widehat{Q}, \Sigma, \widehat{\delta},\left(q_{0}, p_{0}\right), \widehat{F}\right)$, whose state set $\widehat{Q}=Q \times P$ consists of pairs $\left(q_{i}, p_{j}\right)$, and whose transition function $\widehat{\delta}$ is such that $\widehat{M}$ is in state $\left(q_{i}, p_{j}\right)$ whenever $M_{1}$ is in state $q_{i}$ and $M_{2}$ is in state $p_{j}$. This is achieved by taking

$$
\widehat{\delta}\left(\left(q_{i}, p_{j}\right), a\right)=\left(q_{k}, p_{l}\right)
$$

whenever
and

$$
\delta_{1}\left(q_{i}, a\right)=q_{k}
$$

$$
\delta_{2}\left(p_{j}, a\right)=p_{l}
$$

$\widehat{F}$ is defined as the set of all $\left(q_{i}, p j\right)$, such that $q_{i} \in F_{1}$ and $p_{j} \in F_{2}$. Then it is a simple matter to show that $w \in L_{1} \cap L_{2}$ if and only if it is accepted by $\widehat{M}$. Consequently, $L_{1} \cap L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages. Constructive proofs occur throughout this course of lectures; they are important because they give us insight into the results and often serve as the starting point for practical algorithms. Here, as in many cases, there are shorter but nonconstructive (or at least not so obviously constructive) arguments. For closure under intersection, we start with DeMorgan's law, taking the complement of both sides. Then

$$
L_{1} \cap L_{2}=\overline{\overline{L_{1}} \cup \overline{L_{2}}}
$$

for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$. Using closure under union, we next get that $\overline{L_{1}} \cup \overline{L_{2}}$ is regular. Using closure under complementation once more, we see that

$$
\overline{\overline{L_{1}} \cup \overline{L_{2}}}=L_{1} \cap L_{2}
$$

is regular.

### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages Constructive proofs occur throughout this course of lectures; they are important because they give us insight into the results and often serve as the starting point for practical algorithms. Here, as in many cases, there are shorter but nonconstructive (or at least not so obviously constructive) arguments. For closure under intersection, we start with DeMorgan's law, taking the complement of both sides. Then

for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$. Using closure under union, we next get that $\overline{L_{1}} \cup \overline{L_{2}}$ is regular. Using closure under complementation once more, we see that

is regular.

### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, how to construct a finite accepter for the intersection of two regular languages Constructive proofs occur throughout this course of lectures; they are important because they give us insight into the results and often serve as the starting point for practical algorithms. Here, as in many cases, there are shorter but nonconstructive (or at least not so obviously constructive) arguments. For closure under intersection, we start with DeMorgan's law, taking the complement of both sides. Then
for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$. Using closure under union, we next get that $\overline{L_{1}} \cup \overline{L_{2}}$ is regular. Using closure under complementation once more, we see that

is regular.

### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages.


```
important because they give us insight into the results and often serve as the
starting point for practical algorithms. Here, as in many cases, there are shorter
but nonconstructive (or at least not so obviously constructive) arguments. For
closure under intersection, we start with DeMorgan's law, taking the
complement of both sides. Then
```

for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure
under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$. Using closure under union, we next
get that $\overline{L_{1}} \cup \overline{L_{2}}$ is regular. Using closure under complementation once more,
we see that

is regular.

### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages. Constructive proofs occur throughout this course of lectures;

```
important because they give us insight into the results and often serve as the
starting point for practical algorithms. Here, as in many cases, there are shorter
but nonconstructive (or at least not so obviously constructive) arguments. For
closure under intersection, we start with DeMorgan's law, taking the
complement of both sides. Then
```

for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure
under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$. Using closure under union, we next
get that $\overline{L_{1}} \cup \overline{L_{2}}$ is regular. Using closure under complementation once more,
we see that

is regular

### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages. Constructive proofs occur throughout this course of lectures; they are important because they give us insight into the results and often serve as the starting point for practical algorithms.
closure under intersection, we start with DeMorgan's law, taking the complement of both sides. Then
for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$. Using closure under union, we next get that $\overline{L_{1}} \cup \overline{L_{2}}$ is regular. Using closure under complementation once more, we see that

is regular

### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages. Constructive proofs occur throughout this course of lectures; they are important because they give us insight into the results and often serve as the starting point for practical algorithms. Here, as in many cases, there are shorter but nonconstructive (or at least not so obviously constructive) arguments.
complement of both sides. Then
for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$. Using closure under union, we next get that $\overline{L_{1}} \cup \overline{L_{2}}$ is regular. Using closure under complementation once more, we see that

is regular.

### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages. Constructive proofs occur throughout this course of lectures; they are important because they give us insight into the results and often serve as the starting point for practical algorithms. Here, as in many cases, there are shorter but nonconstructive (or at least not so obviously constructive) arguments. For closure under intersection, we start with DeMorgan's law, taking the complement of both sides.
for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$. Using closure under union, we next get that $\overline{L_{1}} \cup \overline{L_{2}}$ is regular. Using closure under complementation once more, we see that

is regular

### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages. Constructive proofs occur throughout this course of lectures; they are important because they give us insight into the results and often serve as the starting point for practical algorithms. Here, as in many cases, there are shorter but nonconstructive (or at least not so obviously constructive) arguments. For closure under intersection, we start with DeMorgan's law, taking the complement of both sides. Then
for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$. Using closure under union, we next get that $\overline{L_{1}} \cup \overline{L_{2}}$ is regular. Using closure under complementation once more, we see that

is regular

### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages. Constructive proofs occur throughout this course of lectures; they are important because they give us insight into the results and often serve as the starting point for practical algorithms. Here, as in many cases, there are shorter but nonconstructive (or at least not so obviously constructive) arguments. For closure under intersection, we start with DeMorgan's law, taking the complement of both sides. Then

$$
L_{1} \cap L_{2}=\overline{\overline{L_{1}} \cup \overline{L_{2}}}
$$

for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$. Using closure under union, we next get that $\overline{L_{1}} \cup \overline{L_{2}}$ is regular. Using closure under complementation once more, we see that


[^0]
### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages. Constructive proofs occur throughout this course of lectures; they are important because they give us insight into the results and often serve as the starting point for practical algorithms. Here, as in many cases, there are shorter but nonconstructive (or at least not so obviously constructive) arguments. For closure under intersection, we start with DeMorgan's law, taking the complement of both sides. Then

$$
L_{1} \cap L_{2}=\overline{\overline{L_{1}} \cup \overline{L_{2}}}
$$

for any languages $L_{1}$ and $L_{2}$.



[^1]
### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages. Constructive proofs occur throughout this course of lectures; they are important because they give us insight into the results and often serve as the starting point for practical algorithms. Here, as in many cases, there are shorter but nonconstructive (or at least not so obviously constructive) arguments. For closure under intersection, we start with DeMorgan's law, taking the complement of both sides. Then

$$
L_{1} \cap L_{2}=\overline{\overline{L_{1}} \cup \overline{L_{2}}}
$$

for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$.
we see that
is regular.

### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages. Constructive proofs occur throughout this course of lectures; they are important because they give us insight into the results and often serve as the starting point for practical algorithms. Here, as in many cases, there are shorter but nonconstructive (or at least not so obviously constructive) arguments. For closure under intersection, we start with DeMorgan's law, taking the complement of both sides. Then

$$
L_{1} \cap L_{2}=\overline{\overline{L_{1}} \cup \overline{L_{2}}}
$$

for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$. Using closure under union, we next get that $\overline{L_{1}} \cup \overline{L_{2}}$ is regular.

[^2]
### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages. Constructive proofs occur throughout this course of lectures; they are important because they give us insight into the results and often serve as the starting point for practical algorithms. Here, as in many cases, there are shorter but nonconstructive (or at least not so obviously constructive) arguments. For closure under intersection, we start with DeMorgan's law, taking the complement of both sides. Then

$$
L_{1} \cap L_{2}=\overline{\overline{L_{1}} \cup \overline{L_{2}}}
$$

for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$. Using closure under union, we next get that $\overline{L_{1}} \cup \overline{L_{2}}$ is regular. Using closure under complementation once more, we see that

### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages. Constructive proofs occur throughout this course of lectures; they are important because they give us insight into the results and often serve as the starting point for practical algorithms. Here, as in many cases, there are shorter but nonconstructive (or at least not so obviously constructive) arguments. For closure under intersection, we start with DeMorgan's law, taking the complement of both sides. Then

$$
L_{1} \cap L_{2}=\overline{\overline{L_{1}} \cup \overline{L_{2}}}
$$

for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$. Using closure under union, we next get that $\overline{L_{1}} \cup \overline{L_{2}}$ is regular. Using closure under complementation once more, we see that

$$
\overline{\overline{L_{1}} \cup \overline{L_{2}}}=L_{1} \cap L_{2}
$$

is regular.

### 4.1 Closure Properties of Regular Languages

The proof of closure under intersection is a good example of a constructive proof. Not only does it establish the desired result, but it also shows explicitly how to construct a finite accepter for the intersection of two regular languages. Constructive proofs occur throughout this course of lectures; they are important because they give us insight into the results and often serve as the starting point for practical algorithms. Here, as in many cases, there are shorter but nonconstructive (or at least not so obviously constructive) arguments. For closure under intersection, we start with DeMorgan's law, taking the complement of both sides. Then

$$
L_{1} \cap L_{2}=\overline{\overline{L_{1}} \cup \overline{L_{2}}}
$$

for any languages $L_{1}$ and $L_{2}$. Now, if $L_{1}$ and $L_{2}$ are regular, then by closure under complementation, so are $\overline{L_{1}}$ and $\overline{L_{2}}$. Using closure under union, we next get that $\overline{L_{1}} \cup \overline{L_{2}}$ is regular. Using closure under complementation once more, we see that

$$
\overline{\overline{L_{1}} \cup \overline{L_{2}}}=L_{1} \cap L_{2}
$$

is regular.

### 4.1 Closure Properties of Regular Languages

The following example is a variation on the same idea.

## Example 4 -1

A variety of other closure properties can be derived directly by elementary arguments.

### 4.1 Closure Properties of Regular Languages

The following example is a variation on the same idea.

```
Example 4.1
```

A variety of other closure properties can be derived directly by elementary arguments.

### 4.1 Closure Properties of Regular Languages

The following example is a variation on the same idea.

## Example 4.1

Show that the family of regular languages is closed under difference. In other words, we want to show that if $L_{1}$ and $L_{2}$ are regular, then $L_{1}-L_{2}$ is necessarily regular also.
The needed set identity is immediately obvious from the definition of a set difference, namely

The fact that $L_{2}$ is regular implies that $\overline{L_{2}}$ is also regular. Then, because of the closure of regular languages under intersection, we know that $L_{1} \cap \overline{L_{2}}$ is regular, and the argument is complete.

A variety of other closure properties can be derived directly by elementary arguments.

### 4.1 Closure Properties of Regular Languages

The following example is a variation on the same idea.

## Example 4.1

Show that the family of regular languages is closed under difference. words, we want to show that if $L_{1}$ and $L_{2}$ are regular, then $L_{1}-L_{2}$ is necessarily regular also.
The needed set identity is immediately obvious from the definition of a set difference, namely

The fact that $L_{2}$ is regular implies that $\overline{L_{2}}$ is also regular. Then, because of the closure of regular languages under intersection, we know that $L_{1} \cap \overline{L_{2}}$ is regular, and the argument is complete.

A variety of other closure properties can be derived directly by elementary arguments.

### 4.1 Closure Properties of Regular Languages

The following example is a variation on the same idea.

## Example 4.1

Show that the family of regular languages is closed under difference. In other words, we want to show that if $L_{1}$ and $L_{2}$ are regular, necessarily regular also.
The needed set identity is immediately obvious from the definition of a set difference, namely

The fact that $L_{2}$ is regular implies that $\overline{L_{2}}$ is also regular. Then, because of the closure of regular languages under intersection, we know that $L_{1} \cap \overline{L_{2}}$ is regular, and the argument is complete.

A variety of other closure properties can be derived directly by elementary arguments.

### 4.1 Closure Properties of Regular Languages

The following example is a variation on the same idea.

## Example 4.1

Show that the family of regular languages is closed under difference. In other words, we want to show that if $L_{1}$ and $L_{2}$ are regular, then $L_{1}-L_{2}$ is necessarily regular also.

> The needed set identity is immediately obvious from the definition of a set difference, namely

> The fact that $L_{2}$ is regular implies that $\overline{L_{2}}$ is also regular. Then, because of the closure of regular languages under intersection, we know that $L_{1} \cap \overline{L_{2}}$ is regular, and the argument is complete.

A variety of other closure properties can be derived directly by elementary arguments.

### 4.1 Closure Properties of Regular Languages

The following example is a variation on the same idea.

## Example 4.1

Show that the family of regular languages is closed under difference. In other words, we want to show that if $L_{1}$ and $L_{2}$ are regular, then $L_{1}-L_{2}$ is necessarily regular also.
The needed set identity is immediately obvious from the definition of a set difference, namely

The fact that $L_{2}$ is regular implies that $\overline{L_{2}}$ is also regular. Then, because of the closure of regular languages under intersection, we know that $L_{1} \cap \overline{L_{2}}$ is regular, and the argument is complete.

A variety of other closure properties can be derived directly by elementary arguments.

### 4.1 Closure Properties of Regular Languages

The following example is a variation on the same idea.

## Example 4.1

Show that the family of regular languages is closed under difference. In other words, we want to show that if $L_{1}$ and $L_{2}$ are regular, then $L_{1}-L_{2}$ is necessarily regular also.
The needed set identity is immediately obvious from the definition of a set difference, namely

$$
L_{1}-L_{2}=L_{1} \cap \overline{L_{2}}
$$

The fact that $L_{2}$ is regular implies that $L_{2}$ is also regular. Then, because of the closure of regular languages under intersection, we know that $L_{1} \cap \overline{L_{2}}$ is regular, and the argument is complete.

A variety of other closure properties can be derived directly by elementary arguments.

### 4.1 Closure Properties of Regular Languages

The following example is a variation on the same idea.

## Example 4.1

Show that the family of regular languages is closed under difference. In other words, we want to show that if $L_{1}$ and $L_{2}$ are regular, then $L_{1}-L_{2}$ is necessarily regular also.
The needed set identity is immediately obvious from the definition of a set difference, namely

$$
L_{1}-L_{2}=L_{1} \cap \overline{L_{2}}
$$

The fact that $L_{2}$ is regular implies that $\overline{L_{2}}$ is also regular. regular, and the argument is complete.

A variety of other closure properties can be derived directly by elementary arguments.

### 4.1 Closure Properties of Regular Languages

The following example is a variation on the same idea.

## Example 4.1

Show that the family of regular languages is closed under difference. In other words, we want to show that if $L_{1}$ and $L_{2}$ are regular, then $L_{1}-L_{2}$ is necessarily regular also.
The needed set identity is immediately obvious from the definition of a set difference, namely

$$
L_{1}-L_{2}=L_{1} \cap \overline{L_{2}}
$$

The fact that $L_{2}$ is regular implies that $\overline{L_{2}}$ is also regular. Then, because of the closure of regular languages under intersection, regular, and the argument is complete.

A variety of other closure properties can be derived directly by elementary arguments.

### 4.1 Closure Properties of Regular Languages

The following example is a variation on the same idea.

## Example 4.1

Show that the family of regular languages is closed under difference. In other words, we want to show that if $L_{1}$ and $L_{2}$ are regular, then $L_{1}-L_{2}$ is necessarily regular also.
The needed set identity is immediately obvious from the definition of a set difference, namely

$$
L_{1}-L_{2}=L_{1} \cap \overline{L_{2}}
$$

The fact that $L_{2}$ is regular implies that $\overline{L_{2}}$ is also regular. Then, because of the closure of regular languages under intersection, we know that $L_{1} \cap \overline{L_{2}}$ is regular, and the argument is complete.

A variety of other closure properties can be derived directly by elementary arguments.

### 4.1 Closure Properties of Regular Languages

The following example is a variation on the same idea.

## Example 4.1

Show that the family of regular languages is closed under difference. In other words, we want to show that if $L_{1}$ and $L_{2}$ are regular, then $L_{1}-L_{2}$ is necessarily regular also.
The needed set identity is immediately obvious from the definition of a set difference, namely

$$
L_{1}-L_{2}=L_{1} \cap \overline{L_{2}}
$$

The fact that $L_{2}$ is regular implies that $\overline{L_{2}}$ is also regular. Then, because of the closure of regular languages under intersection, we know that $L_{1} \cap \overline{L_{2}}$ is regular, and the argument is complete.

A variety of other closure properties can be derived directly by elementary arguments.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.2

Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.

## Closure under Other Operations

In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex,
edges. It is a fairly straightforward matter to show that the modified nfa
accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.

Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges.
accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$.

Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other
operations and investigate closure properties for them. There are many such results; we select only two typical ones.

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other
operations and investigate closure properties for them. There are many such results; we select only two typical ones.

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them.

[^3]
## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

## Theorem 4.2

The family of regular languages is closed under reversal.
Proof. Suppose that $L$ is a regular language. We then construct an nfa with a single final state for it. In the previous lectures we show that this is always possible. In the transition graph for this nfa we make the initial vertex a final vertex, the final vertex the initial vertex, and reverse the direction on all the edges. It is a fairly straightforward matter to show that the modified nfa accepts $w^{R}$ if and only if the original nfa accepts $w$. Therefore, the modified nfa accepts $L^{R}$, proving closure under reversal.
Closure under Other Operations
In addition to the standard operations on languages, one can define other operations and investigate closure properties for them. There are many such results; we select only two typical ones.

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

## Example 4.2

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

```
Suppose \Sigma and \Gamma are alphabets. Then a function
    h:\Sigma-> \Gamma*
is called a homomorphism. In vords, a homomorphism is a substitution in
which a single letter is replaced with a string. The domain of the function }h\mathrm{ is
extended to strings in an obvious fashion; if
then
w}=\mp@subsup{a}{1}{}\mp@subsup{a}{2}{}\cdots\mp@subsup{a}{n}{
h(w)=h(\mp@subsup{a}{1}{})h(\mp@subsup{a}{2}{})\cdotsh(\mp@subsup{a}{n}{}).
If L}\mathrm{ is a language on }\Sigma\mathrm{ , then its homomorphic image is defined as
    h(L)}={h(w):w\inL}
```

Example 4.2

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function
is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if

```
then
\[
h(u)=1\left(a_{1}\right) n\left(a_{2}\right) \cdots \cdots\left(a_{n}\right)
\]

If \(L\) is a language on \(\Sigma\), then its homomorphic image is defined as
\[
h(L)=\{h(w): w \in L\} .
\]

\section*{Example 4.2}

\subsection*{4.1 Closure Properties of Regular Languages}

\section*{Definition 4.1}

Suppose \(\Sigma\) and \(\Gamma\) are alphabets. Then a function
is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function \(h\) is extended to strings in an obvious fashion; if
```

then

$$
h\left(u^{\prime}\right)=\pi\left(a_{1}\right) n\left(a_{2}\right) \cdots h\left(a_{n}\right)
$$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as

$$
h(L)=\{h(w): w \in L\} .
$$

## Example 4.2

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if

```
then then
```

$w=a_{1} a_{2} \cdots a_{n}$,

$$
h(w)=h\left(a_{1}\right) h\left(a_{2}\right) \cdots h\left(a_{n}\right) .
$$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as

$$
h(L)=\{h(w): w \in L\} .
$$

## Example 4.2

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if

```
then
```

$$
h(w)=h\left(a_{1}\right) h\left(a_{2}\right) \cdots h\left(a_{n}\right) .
$$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as

$$
h(L)=\{h(w): w \in L\} .
$$

## Example 4.2

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is
extended to strings in an obvious fashion; if

$h(L)=\{h(w): w \in L\}$

## Example 4.2

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion;
$\qquad$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as
$\square$

## Example 4.2

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if

```
then
```

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as
$\square$

## Example 4.2

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if

$$
w=a_{1} a_{2} \cdots a_{n},
$$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as

$$
h(L)=\{h(w): w \in L\}
$$

## Example 4.2

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if then

$$
w=a_{1} a_{2} \cdots a_{n}
$$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as

$$
h(L)=\{h(w): w \in L\} .
$$

## Example 4.2

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if
then

$$
w=a_{1} a_{2} \cdots a_{n},
$$

$$
h(w)=h\left(a_{1}\right) h\left(a_{2}\right) \cdots h\left(a_{n}\right)
$$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as

$$
h(L)=\{h(w): w \in L\}
$$

## Example 4.2

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if
then

$$
w=a_{1} a_{2} \cdots a_{n},
$$

$$
h(w)=h\left(a_{1}\right) h\left(a_{2}\right) \cdots h\left(a_{n}\right)
$$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as

## Example 4.2

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if
then

$$
w=a_{1} a_{2} \cdots a_{n},
$$

$$
h(w)=h\left(a_{1}\right) h\left(a_{2}\right) \cdots h\left(a_{n}\right)
$$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as

## Example 4.2

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if
then

$$
w=a_{1} a_{2} \cdots a_{n},
$$

$$
h(w)=h\left(a_{1}\right) h\left(a_{2}\right) \cdots h\left(a_{n}\right)
$$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as

$$
h(L)=\{h(w): w \in L\}
$$

## Example 4.2

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if
then

$$
w=a_{1} a_{2} \cdots a_{n}
$$

$$
h(w)=h\left(a_{1}\right) h\left(a_{2}\right) \cdots h\left(a_{n}\right)
$$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as

$$
h(L)=\{h(w): w \in L\} .
$$

## Example 4.2



### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if
then

$$
w=a_{1} a_{2} \cdots a_{n},
$$

$$
h(w)=h\left(a_{1}\right) h\left(a_{2}\right) \cdots h\left(a_{n}\right)
$$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as

$$
h(L)=\{h(w): w \in L\} .
$$

## Example 4.2

Let $\Sigma=\{a, b\}$ and $\Gamma=\{a, b, c\}$ and define $h$ by


Then $h(a b a)=a b b b c a b$. The homomorphic image of $L=\{a a, a b a\}$ is the language $h(L)=\{a b a b, a b b b c a b\}$.

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if
then

$$
w=a_{1} a_{2} \cdots a_{n},
$$

$$
h(w)=h\left(a_{1}\right) h\left(a_{2}\right) \cdots h\left(a_{n}\right)
$$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as

$$
h(L)=\{h(w): w \in L\} .
$$

## Example 4.2

Let $\Sigma=\{a, b\}$ and $\Gamma=\{a, b, c\}$ and define $h$ by

$$
\begin{aligned}
& h(a)=a b, \\
& h(b)=b b c .
\end{aligned}
$$

Then $h(a b a)=a b b b c a b$. The homomorphic image of $L=\{a a, a b a\}$ is the language $h(L)=\{a b a b, a b b b c a b\}$

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if
then

$$
w=a_{1} a_{2} \cdots a_{n},
$$

$$
h(w)=h\left(a_{1}\right) h\left(a_{2}\right) \cdots h\left(a_{n}\right)
$$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as

$$
h(L)=\{h(w): w \in L\} .
$$

## Example 4.2

Let $\Sigma=\{a, b\}$ and $\Gamma=\{a, b, c\}$ and define $h$ by

$$
\begin{aligned}
h(a) & =a b \\
h(b) & =b b c .
\end{aligned}
$$

Then $h(a b a)=a b b b c a b$. The homomorphic image of $L=\{a a, a b a\}$ is the language $h(L)=\{a b a b, a b b b c a b\}$.

### 4.1 Closure Properties of Regular Languages

## Definition 4.1

Suppose $\Sigma$ and $\Gamma$ are alphabets. Then a function

$$
h: \Sigma \rightarrow \Gamma^{*}
$$

is called a homomorphism. In words, a homomorphism is a substitution in which a single letter is replaced with a string. The domain of the function $h$ is extended to strings in an obvious fashion; if
then

$$
w=a_{1} a_{2} \cdots a_{n},
$$

$$
h(w)=h\left(a_{1}\right) h\left(a_{2}\right) \cdots h\left(a_{n}\right)
$$

If $L$ is a language on $\Sigma$, then its homomorphic image is defined as

$$
h(L)=\{h(w): w \in L\} .
$$

## Example 4.2

Let $\Sigma=\{a, b\}$ and $\Gamma=\{a, b, c\}$ and define $h$ by

$$
\begin{aligned}
h(a) & =a b \\
h(b) & =b b c .
\end{aligned}
$$

Then $h(a b a)=a b b b c a b$. The homomorphic image of $L=\{a a, a b a\}$ is the language $h(L)=\{a b a b, a b b b c a b\}$.

### 4.1 Closure Properties of Regular Languages

```
If we have a regular expression r for a language L, then a regular expression for
h(L) can be obtained by simply applying the homomorphism to each }\Sigma\mathrm{ symbol
of }
```


## Example 4.3

The general result on the closure of regular languages under any
homomorphism follows from this example in an obvious manner.

### 4.1 Closure Properties of Regular Languages

If we have a regular expression $r$ for a language $L$, then a regular expression for $h(L)$ can be obtained by simply applying the homomorphism to each $\Sigma$ symbol of $r$.

Example 4.3

The general result on the closure of regular languages under any
homomorphism follows from this example in an obvious manner.

### 4.1 Closure Properties of Regular Languages

If we have a regular expression $r$ for a language $L$, then a regular expression for $h(L)$ can be obtained by simply applying the homomorphism to each $\Sigma$ symbol of $r$.

## Example 4.3

The general result on the closure of regular languages under any
homomorphism follows from this example in an obvious manner.

### 4.1 Closure Properties of Regular Languages

If we have a regular expression $r$ for a language $L$, then a regular expression for $h(L)$ can be obtained by simply applying the homomorphism to each $\Sigma$ symbol of $r$.

## Example 4.3

Take $\Sigma=\{a, b\}$ and $\Gamma=\{b, c, d\}$. Define $h$ by

$$
h(a)=d b c c,
$$

$h(b)=b d c$. $r=\left(a+b^{*}\right)(a a)^{*}$,
then $r_{1}=\left(d b c c+(b d c)^{*}\right)(d b c c d b c c)^{*}$ denotes the regular language $h(L)$.

The general result on the closure of regular languages under any
homomorphism follows from this example in an obvious manner.

### 4.1 Closure Properties of Regular Languages

If we have a regular expression $r$ for a language $L$, then a regular expression for $h(L)$ can be obtained by simply applying the homomorphism to each $\Sigma$ symbol of $r$.

## Example 4.3

Take $\Sigma=\{a, b\}$ and $\Gamma=\{b, c, d\}$.

$$
h(a)=d h c c_{0}
$$

$$
h(b)=b d c
$$

If $L$ is the regular language denoted by

then

$$
r_{1}=\left(d b c c+(b d c)^{*}\right)(d b c c d b c c)^{*}
$$

denotes the regular language $h(L)$.
The general result on the closure of regular languages under any
homomorphism follows from this example in an obvious manner.

### 4.1 Closure Properties of Regular Languages

If we have a regular expression $r$ for a language $L$, then a regular expression for $h(L)$ can be obtained by simply applying the homomorphism to each $\Sigma$ symbol of $r$.

## Example 4.3

Take $\Sigma=\{a, b\}$ and $\Gamma=\{b, c, d\}$. Define $h$ by

If $L$ is the regular language denoted by

then $r_{1}=\left(d b c c+(b d c)^{*}\right)(d b c c d b c c)^{*}$
denotes the regular language $h(L)$.
The general result on the closure of regular languages under any
homomorphism follows from this example in an obvious manner.

### 4.1 Closure Properties of Regular Languages

If we have a regular expression $r$ for a language $L$, then a regular expression for $h(L)$ can be obtained by simply applying the homomorphism to each $\Sigma$ symbol of $r$.

## Example 4.3

Take $\Sigma=\{a, b\}$ and $\Gamma=\{b, c, d\}$. Define $h$ by

$$
\begin{aligned}
h(a) & =d b c c \\
h(b) & =b d c
\end{aligned}
$$

If $L$ is the regular language denoted by
then

$r_{1}=\left(d b c c+(b d c)^{*}\right)(d b c c d b c c)^{*}$
denotes the regular language $h(L)$.
The general result on the closure of regular languages under any
homomorphism follows from this example in an obvious manner.

### 4.1 Closure Properties of Regular Languages

If we have a regular expression $r$ for a language $L$, then a regular expression for $h(L)$ can be obtained by simply applying the homomorphism to each $\Sigma$ symbol of $r$.

## Example 4.3

Take $\Sigma=\{a, b\}$ and $\Gamma=\{b, c, d\}$. Define $h$ by

$$
\begin{aligned}
h(a) & =d b c c \\
h(b) & =b d c
\end{aligned}
$$

If $L$ is the regular language denoted by
$\square$
then
denotes the regular language $h(L)$.
The general result on the closure of regular languages under any
homomorphism follows from this example in an obvious manner.

### 4.1 Closure Properties of Regular Languages

If we have a regular expression $r$ for a language $L$, then a regular expression for $h(L)$ can be obtained by simply applying the homomorphism to each $\Sigma$ symbol of $r$.

## Example 4.3

Take $\Sigma=\{a, b\}$ and $\Gamma=\{b, c, d\}$. Define $h$ by

$$
\begin{aligned}
h(a) & =d b c c \\
h(b) & =b d c
\end{aligned}
$$

If $L$ is the regular language denoted by

$$
r=\left(a+b^{*}\right)(a a)^{*}
$$

then

$$
r_{1}=\left(d b c c+(b d c)^{*}\right)(d b c c d b c c)^{*}
$$

denotes the regular language $h(L)$.
The general result on the closure of regular languages under any
homomorphism follows from this example in an obvious manner.

### 4.1 Closure Properties of Regular Languages

If we have a regular expression $r$ for a language $L$, then a regular expression for $h(L)$ can be obtained by simply applying the homomorphism to each $\Sigma$ symbol of $r$.

## Example 4.3

Take $\Sigma=\{a, b\}$ and $\Gamma=\{b, c, d\}$. Define $h$ by

$$
\begin{aligned}
h(a) & =d b c c \\
h(b) & =b d c
\end{aligned}
$$

If $L$ is the regular language denoted by

$$
r=\left(a+b^{*}\right)(a a)^{*}
$$

then

$$
r_{1}=\left(d b c c+(b d c)^{*}\right)(d b c c d b c c)^{*}
$$

denotes the regular language $h(L)$.
The general result on the closure of regular languages under any
homomorphism follows from this example in an obvious manner.

### 4.1 Closure Properties of Regular Languages

If we have a regular expression $r$ for a language $L$, then a regular expression for $h(L)$ can be obtained by simply applying the homomorphism to each $\Sigma$ symbol of $r$.

## Example 4.3

Take $\Sigma=\{a, b\}$ and $\Gamma=\{b, c, d\}$. Define $h$ by

$$
\begin{aligned}
h(a) & =d b c c \\
h(b) & =b d c
\end{aligned}
$$

If $L$ is the regular language denoted by

$$
r=\left(a+b^{*}\right)(a a)^{*}
$$

then

$$
r_{1}=\left(d b c c+(b d c)^{*}\right)(d b c c d b c c)^{*}
$$

denotes the regular language $h(L)$.
The general result on the closure of regular languages under any
homomorphism follows from this example in an obvious manner.

### 4.1 Closure Properties of Regular Languages

If we have a regular expression $r$ for a language $L$, then a regular expression for $h(L)$ can be obtained by simply applying the homomorphism to each $\Sigma$ symbol of $r$.

## Example 4.3

Take $\Sigma=\{a, b\}$ and $\Gamma=\{b, c, d\}$. Define $h$ by

$$
\begin{aligned}
h(a) & =d b c c \\
h(b) & =b d c
\end{aligned}
$$

If $L$ is the regular language denoted by

$$
r=\left(a+b^{*}\right)(a a)^{*}
$$

then

$$
r_{1}=\left(d b c c+(b d c)^{*}\right)(d b c c d b c c)^{*}
$$

denotes the regular language $h(L)$.
The general result on the closure of regular languages under any
homomorphism follows from this example in an obvious manner.

### 4.1 Closure Properties of Regular Languages

If we have a regular expression $r$ for a language $L$, then a regular expression for $h(L)$ can be obtained by simply applying the homomorphism to each $\Sigma$ symbol of $r$.

## Example 4.3

Take $\Sigma=\{a, b\}$ and $\Gamma=\{b, c, d\}$. Define $h$ by

$$
\begin{aligned}
h(a) & =d b c c \\
h(b) & =b d c
\end{aligned}
$$

If $L$ is the regular language denoted by

$$
r=\left(a+b^{*}\right)(a a)^{*}
$$

then

$$
r_{1}=\left(d b c c+(b d c)^{*}\right)(d b c c d b c c)^{*}
$$

denotes the regular language $h(L)$.
The general result on the closure of regular languages under any homomorphism follows from this example in an obvious manner.

### 4.1 Closure Properties of Regular Languages

Theorem 4.3

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix, belongs to $L_{1} / L_{2}$.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

[^4]
### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

[^5]
### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular.
under arbitrary homomorphisms.
Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular

## Definition 4.2

[^6]
### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

```
Proof. Let L be a regular language denoted by some regular expression r. We
find h(r) by substituting h(a) for each symbol }a\in\Sigma\Sigma\mathrm{ of r. It can be shown
directly by an appeal to the definition of a regular expression that the result is a
regular expression. It is equally easy to see that the resulting expression denotes
h(L). All we need to do is to show that for every }w\inL(r)\mathrm{ , the corresponding
h(w) is in L(h(r)) and conversely that for every v}\mathrm{ in L(h(r)) there is a word w
in L, such that v}=h(w). Leaving the details as an exercise, we claim that
h(L) is regular
```


## Definition 4.2

To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that
have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix,
belongs to $L_{1} / L_{2}$.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression r. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that
have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix,
belongs to $L_{1} / L_{2}$.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$.
directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that
have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix,
belongs to $L_{1} / L_{2}$.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

$\square$

### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression.
$h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

$\square$

### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

$\square$
To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix, belongs to $L_{1} / L_{2}$

### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$
in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that
$h(L)$ is regular

## Definition 4.2

[^7]
### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$.
_eaving the details as an exercise, we claim that

## Definition 4.2

[^8]
### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.
$\square$

To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix, belongs to $L_{1} / L_{2}$

### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.
$\square$

To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix, belongs to $L_{1} / L_{2}$

### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

$L_{1}$ with $L_{2}$ is defined as

$$
L_{1} / L_{2}=\left\{x: x y \in L_{1} \text { for some } y \in L_{2}\right\}
$$

[^9]
### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

Let $L_{1}$ and $L_{2}$ be languages on the same alphabet.
is defined as

[^10]
### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

Let $L_{1}$ and $L_{2}$ be languages on the same alphabet. Then the right quotient of $L_{1}$ with $L_{2}$ is defined as

[^11]
### 4.1 Closure Properties of Regular Languages

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

Let $L_{1}$ and $L_{2}$ be languages on the same alphabet. Then the right quotient of $L_{1}$ with $L_{2}$ is defined as

$$
\begin{equation*}
L_{1} / L_{2}=\left\{x: x y \in L_{1} \text { for some } y \in L_{2}\right\} \tag{1}
\end{equation*}
$$

To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix, belongs to $L_{1} / L_{2}$.

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

Let $L_{1}$ and $L_{2}$ be languages on the same alphabet. Then the right quotient of $L_{1}$ with $L_{2}$ is defined as

$$
\begin{equation*}
L_{1} / L_{2}=\left\{x: x y \in L_{1} \text { for some } y \in L_{2}\right\} . \tag{1}
\end{equation*}
$$

To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix, belongs to $L_{1} / L_{2}$.

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

Let $L_{1}$ and $L_{2}$ be languages on the same alphabet. Then the right quotient of $L_{1}$ with $L_{2}$ is defined as

$$
\begin{equation*}
L_{1} / L_{2}=\left\{x: x y \in L_{1} \text { for some } y \in L_{2}\right\} . \tag{1}
\end{equation*}
$$

To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix,

## Theorem 4.3

Let $h$ be a homomorphism. If $L$ is a regular language, then its homomorphic image $h(L)$ is also regular. The family of regular languages is therefore closed under arbitrary homomorphisms.

Proof. Let $L$ be a regular language denoted by some regular expression $r$. We find $h(r)$ by substituting $h(a)$ for each symbol $a \in \Sigma$ of $r$. It can be shown directly by an appeal to the definition of a regular expression that the result is a regular expression. It is equally easy to see that the resulting expression denotes $h(L)$. All we need to do is to show that for every $w \in L(r)$, the corresponding $h(w)$ is in $L(h(r))$ and conversely that for every $v$ in $L(h(r))$ there is a word $w$ in $L$, such that $v=h(w)$. Leaving the details as an exercise, we claim that $h(L)$ is regular.

## Definition 4.2

Let $L_{1}$ and $L_{2}$ be languages on the same alphabet. Then the right quotient of $L_{1}$ with $L_{2}$ is defined as

$$
\begin{equation*}
L_{1} / L_{2}=\left\{x: x y \in L_{1} \text { for some } y \in L_{2}\right\} . \tag{1}
\end{equation*}
$$

To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix, belongs to $L_{1} / L_{2}$.

### 4.1 Closure Properties of Regular Languages

### 4.1 Closure Properties of Regular Languages

## Example 4.4

If
and

$$
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\}
$$

then

$$
L_{2}=\left\{b^{m}: m \geqslant 1\right\}
$$

$$
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\}
$$

The strings in $L_{2}$ consist of one or more $b$ 's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.

Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, let us see how it applies to this example. We start with a dfa for $L_{1}$; say the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

### 4.1 Closure Properties of Regular Languages

## Example 4.4 <br> If <br> and <br> $$
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\}
$$ <br> then <br> $$
I_{2}=\left\{b^{m}: m \geqslant 1\right\}
$$ <br> $$
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\}
$$ <br> The strings in $L_{2}$ consist of one or more $b$ 's. Therefore, we arrive at the answer by removing one or more b's from those strings in $L_{1}$ that terminate with at least one $b$. <br> Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, let us see how it applies to this example. We start with a dfa for $L_{1}$; say the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

### 4.1 Closure Properties of Regular Languages

## Example 4.4

If

$$
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\}
$$

then

$$
L_{2}=\left\{b^{m}: m \geqslant 1\right\},
$$

$$
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} .
$$

The strings in $L_{2}$ consist of one or more b's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.

Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, let us see how it applies to this example. We start with a dfa for $L_{1}$; say the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

### 4.1 Closure Properties of Regular Languages

$$
\begin{aligned}
& \text { Example } 4.4 \\
& \text { If } \\
& \text { and } \\
& \text { then } L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\} \\
& \qquad L_{2}=\left\{b^{m}: m \geqslant 1\right\}, \\
& \qquad L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\}
\end{aligned}
$$

The strings in $L_{2}$ consist of one or more $b$ 's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.

Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, let us see how it applies to this example. We start with a dfa for $L_{1}$; say the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

### 4.1 Closure Properties of Regular Languages

$$
\begin{aligned}
& \text { Example } 4.4 \\
& \text { If } \\
& \text { and } \\
& \text { then } L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\} \\
& \qquad L_{2}=\left\{b^{m}: m \geqslant 1\right\}, \\
& \qquad L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\}
\end{aligned}
$$

The strings in $L_{2}$ consist of one or more b's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.
Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, let us see how it applies to this example. We start with a dfa for $L_{1}$; say the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

### 4.1 Closure Properties of Regular Languages

## Example 4.4

If
and
then

$$
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\}
$$

$$
L_{2}=\left\{b^{m}: m \geqslant 1\right\}
$$

$$
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\}
$$

The strings in $L_{2}$ consist of one or more b's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.

Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, let us see how it applies to this example. We start with a dfa for $L_{1}$; say the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

## Example 4.4

If
and
then

$$
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\}
$$

$$
L_{2}=\left\{b^{m}: m \geqslant 1\right\}
$$

$$
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\}
$$

The strings in $L_{2}$ consist of one or more b's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.
Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, let us see how it applies to this example. We start with a dfa for $L_{1}$; say the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

### 4.1 Closure Properties of Regular Languages

## Example 4.4

If
and
then

$$
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\}
$$

$$
L_{2}=\left\{b^{m}: m \geqslant 1\right\},
$$

$$
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\}
$$

The strings in $L_{2}$ consist of one or more $b$ 's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.

Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, let us see how it applies to this example. We start with a dfa for $L_{1}$; say the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

## Example 4.4

If
and
then

$$
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\}
$$

$$
L_{2}=\left\{b^{m}: m \geqslant 1\right\}
$$

$$
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} .
$$

The strings in $L_{2}$ consist of one or more $b$ 's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at
least one $b$.
Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, let us see how it applies to this example. We start with a dfa for $L_{1}$; say the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

## Example 4.4

If
and
then

$$
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\}
$$

$$
L_{2}=\left\{b^{m}: m \geqslant 1\right\}
$$

$$
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\}
$$

The strings in $L_{2}$ consist of one or more $b$ 's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.

> Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, let us see how it applies to this example. We start with a dfa for $L_{1}$; say the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

## Example 4.4

If
and
then

$$
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\}
$$

$$
L_{2}=\left\{b^{m}: m \geqslant 1\right\}
$$

$$
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\}
$$

The strings in $L_{2}$ consist of one or more $b$ 's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.

Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular.
quotient of any two regular languages is also regular. We shall prove this in the
next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and
constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in
full, let us see how it applies to this example. We start with a dfa for $L_{1}$; say
the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

## Example 4.4

If
and
then

$$
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\}
$$

$$
L_{2}=\left\{b^{m}: m \geqslant 1\right\}
$$

$$
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\}
$$

The strings in $L_{2}$ consist of one or more $b$ 's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.

Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular.
> next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, let us see how it applies to this example. We start with a dfa for $L_{1}$; say the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

### 4.1 Closure Properties of Regular Languages

## Example 4.4

If
and
then

$$
\begin{gathered}
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\} \\
L_{2}=\left\{b^{m}: m \geqslant 1\right\}, \\
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} .
\end{gathered}
$$

The strings in $L_{2}$ consist of one or more $b$ 's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.

Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and
full, let us see how it applies to this example. We start with a dfa for $L_{1}$; say the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

## Example 4.4

If
and
then

$$
\begin{gathered}
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\} \\
L_{2}=\left\{b^{m}: m \geqslant 1\right\}, \\
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} .
\end{gathered}
$$

The strings in $L_{2}$ consist of one or more $b$ 's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.

Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$.
the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

## Example 4.4

If
and
then

$$
\begin{gathered}
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\} \\
L_{2}=\left\{b^{m}: m \geqslant 1\right\}, \\
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} .
\end{gathered}
$$

The strings in $L_{2}$ consist of one or more $b$ 's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.

Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

## Example 4.4

If
and
then

$$
\begin{gathered}
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\} \\
L_{2}=\left\{b^{m}: m \geqslant 1\right\}, \\
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} .
\end{gathered}
$$

The strings in $L_{2}$ consist of one or more $b$ 's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.

Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, let us see how it applies to this example.

## Example 4.4

If
and
then

$$
\begin{gathered}
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\} \\
L_{2}=\left\{b^{m}: m \geqslant 1\right\}, \\
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} .
\end{gathered}
$$

The strings in $L_{2}$ consist of one or more $b$ 's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.

Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, let us see how it applies to this example. We start with a dfa for $L_{1}$;

## Example 4.4

If
and

$$
L_{1}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\} \cup\{b a\}
$$

then

$$
L_{2}=\left\{b^{m}: m \geqslant 1\right\}
$$

$$
L_{1} / L_{2}=\left\{a^{n} b^{m}: n \geqslant 1, m \geqslant 0\right\}
$$

The strings in $L_{2}$ consist of one or more $b$ 's. Therefore, we arrive at the answer by removing one or more $b$ 's from those strings in $L_{1}$ that terminate with at least one $b$.

Note that here $L_{1}, L_{2}$, and $L_{1} / L_{2}$ are all regular. This suggests that the right quotient of any two regular languages is also regular. We shall prove this in the next theorem by a construction that takes the dfa's for $L_{1}$ and $L_{2}$ and constructs from them a dfa for $L_{1} / L_{2}$. Before we describe the construction in full, let us see how it applies to this example. We start with a dfa for $L_{1}$; say the automaton $M_{1}=\left(Q, \Sigma, \delta, q_{0}, F\right)$ in the Figure.

### 4.1 Closure Properties of Regular Languages

### 4.1 Closure Properties of Regular Languages

Example 4.4 (continuation)


Since an automaton for $L_{1} / L_{2}$ must accept any prefix of strings in $L_{1}$, we shall try to modify $M_{1}$ so that it accepts $x$ if there is any $y$ satisfying (1).
$L_{1} / L_{2}=\left\{x: x y \in L_{1}\right.$ for some $\left.y \in L_{2}\right\}$.
The difficulty comes in finding whether there is some $y$ such that $x y \in L_{1}$ and $y \in L_{2}$. To solve it, we determine, for each $q \in Q$, whether there is a walk to a final state labeled $v$ such that $v \in L_{2}$. If this is so, any $x$ such that $\delta(q 0, x)=q$ will be in $L_{1} / L_{2}$. We modify the automaton accordingly to make $q$ a final state.

### 4.1 Closure Properties of Regular Languages

Example 4.4 (continuation)


Since an automaton for $L_{1} / L_{2}$ must accept any prefix of strings in $L_{1}$, we shall try to modify $M_{1}$ so that it accepts $x$ if there is any $y$ satisfying (1).

$$
\begin{equation*}
L_{1} / L_{2}=\left\{x: x y \in L_{1} \text { for some } y \in L_{2}\right\} . \tag{1}
\end{equation*}
$$

The difficulty comes in finding whether there is some $y$ such that $x y \in L_{1}$ and $y \in L_{2}$. To solve it, we determine, for each $q \in Q$, whether there is a walk to a final state labeled $v$ such that $v \in L_{2}$. If this is so, any $x$ such that $\delta\left(q_{0}, x\right)=q$ will be in $L_{1} / L_{2}$. We modify the automaton accordingly to make $q$ a final state.

### 4.1 Closure Properties of Regular Languages

Example 4.4 (continuation)


Since an automaton for $L_{1} / L_{2}$ must accept any prefix of strings in $L_{1}$, we shall try to modify $M_{1}$ so that it accepts $x$ if there is any $y$ satisfying (1)

$$
\begin{equation*}
L_{1} / L_{2}=\left\{x: x y \in L_{1} \text { for some } y \in L_{2}\right\} \text {. } \tag{1}
\end{equation*}
$$

The difficulty comes in finding whether there is some $y$ such that $x y \in L_{1}$ and $y \in L_{2}$. To solve it, we determine, for each $q \in Q$, whether there is a walk to a final state labeled $v$ such that $v \in L_{2}$. If this is so, any $x$ such that $\delta(q 0, x)=q$ will be in $L_{1} / L_{2}$. We modify the automaton accordingly to make $q$ a final state.

### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)



Since an automaton for $L_{1} / L_{2}$ must accept any prefix of strings in $L_{1}$, we shall try to modify $M_{1}$ so that it accepts $x$ if there is any $y$ satisfying (1).

The difficulty comes in finding whether there is some $y$ such that $x y \in L_{1}$ and $y \in L_{2}$. To solve it, we determine, for each $q \in Q$, whether there is a walk to a final state labeled $v$ such that $v \in L_{2}$. If this is so, any $x$ such that $\delta(q 0, x)=q$ will be in $L_{1} / L_{2}$. We modify the automaton accordingly to make $q$ a final state.

### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)



Since an automaton for $L_{1} / L_{2}$ must accept any prefix of strings in $L_{1}$, we shall try to modify $M_{1}$ so that it accepts $x$ if there is any $y$ satisfying (1).

$$
\begin{equation*}
L_{1} / L_{2}=\left\{x: x y \in L_{1} \text { for some } y \in L_{2}\right\} . \tag{1}
\end{equation*}
$$

The difficulty comes in finding whether there is some $y$ such that $x y \in L_{1}$ and
$y \in L_{2}$. To solve it, we determine, for each $q \in Q$, whether there is a walk to a final state labeled $v$ such that $v \in L_{2}$. If this is so, any $x$ such that $\delta(q 0, x)=q$ will be in $L_{1} / L_{2}$. We modify the automaton accordingly to make $q$ a final state.

### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)



Since an automaton for $L_{1} / L_{2}$ must accept any prefix of strings in $L_{1}$, we shall try to modify $M_{1}$ so that it accepts $x$ if there is any $y$ satisfying (1).

$$
\begin{equation*}
L_{1} / L_{2}=\left\{x: x y \in L_{1} \text { for some } y \in L_{2}\right\} . \tag{1}
\end{equation*}
$$

The difficulty comes in finding whether there is some $y$ such that $x y \in L_{1}$ and $y \in L_{2}$. To solve it, we determine, for each $q \in Q$, whether there is a walk to a final state labeled $v$ such that $v \in L_{2}$. If this is so, any $x$ such that $\delta(q 0, x)=q$ will be in $L_{1} / L_{2}$. We modify the automaton accordingly to make $q$ a final state.

### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)



Since an automaton for $L_{1} / L_{2}$ must accept any prefix of strings in $L_{1}$, we shall try to modify $M_{1}$ so that it accepts $x$ if there is any $y$ satisfying (1).

$$
\begin{equation*}
L_{1} / L_{2}=\left\{x: x y \in L_{1} \text { for some } y \in L_{2}\right\} . \tag{1}
\end{equation*}
$$

The difficulty comes in finding whether there is some $y$ such that $x y \in L_{1}$ and $y \in L_{2}$. To solve it, we determine, for each $q \in Q$, whether there is a walk to a final state labeled $v$ such that $v \in L_{2}$. If this is so, any $x$ such that $\delta(q 0, x)=q$ will be in $L_{1} / L_{2}$. We modify the automaton accordingly to make $q$ a final state.

### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)



Since an automaton for $L_{1} / L_{2}$ must accept any prefix of strings in $L_{1}$, we shall try to modify $M_{1}$ so that it accepts $x$ if there is any $y$ satisfying (1).

$$
\begin{equation*}
L_{1} / L_{2}=\left\{x: x y \in L_{1} \text { for some } y \in L_{2}\right\} . \tag{1}
\end{equation*}
$$

The difficulty comes in finding whether there is some $y$ such that $x y \in L_{1}$ and $y \in L_{2}$. To solve it, we determine, for each $q \in Q$, whether there is a walk to a final state labeled $v$ such that $v \in L_{2}$. If this is so, any $x$ such that $\delta\left(q_{0}, x\right)=q$ will be in $L_{1} / L_{2}$. We modify the automaton accordingly to make $q$ a final state.

### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)



Since an automaton for $L_{1} / L_{2}$ must accept any prefix of strings in $L_{1}$, we shall try to modify $M_{1}$ so that it accepts $x$ if there is any $y$ satisfying (1).

$$
\begin{equation*}
L_{1} / L_{2}=\left\{x: x y \in L_{1} \text { for some } y \in L_{2}\right\} . \tag{1}
\end{equation*}
$$

The difficulty comes in finding whether there is some $y$ such that $x y \in L_{1}$ and $y \in L_{2}$. To solve it, we determine, for each $q \in Q$, whether there is a walk to a final state labeled $v$ such that $v \in L_{2}$. If this is so, any $x$ such that $\delta\left(q_{0}, x\right)=q$ will be in $L_{1} / L_{2}$. We modify the automaton accordingly to make $q$ a final state.

### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)

### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)

To apply this to our present case, we check each state $q_{0}, q_{1}, q_{2}, q_{3}, q_{4}, q_{5}$ to see whether there is a walk labeled $b b^{*}$ to any of the $q_{1}, q_{2}$, or $q_{4}$. We see that only $q_{1}$ and $q_{2}$ qualify: $q_{0}, q_{3}, q_{4}$ do not. The resulting automaton for $L_{1} / L_{2}$ is shown in the following Figure.


Check it to see that the construction works. The idea is generalized in the next theorem.

### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)

To apply this to our present case, we check each state $q_{0}, q_{1}, q_{2}, q_{3}, q_{4}, q_{5}$ to see whether there is a walk labeled $b b^{*}$ to any of the $q_{1}, q_{2}$, or $q_{4}$. We see that only $q_{1}$ and $q_{2}$ qualify; $q_{0}, q_{3}, q_{4}$ do not. The resulting automaton for $L_{1} / L_{2}$ is shown in the following Figure.


Check it to see that the construction works. The idea is generalized in the next theorem.

### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)

To apply this to our present case, we check each state $q_{0}, q_{1}, q_{2}, q_{3}, q_{4}, q_{5}$ to see whether there is a walk labeled $b b^{*}$ to any of the $q_{1}, q_{2}$, or $q_{4}$. We see that only $q_{1}$ and $q_{2}$ qualify; $q_{0}, q_{3}, q_{4}$ do not. The resulting automaton for $L_{1} / L_{2}$ is shown in the following Figure.


Check it to see that the construction works. The idea is generalized in the next theorem.

### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)

To apply this to our present case, we check each state $q_{0}, q_{1}, q_{2}, q_{3}, q_{4}, q_{5}$ to see whether there is a walk labeled $b b^{*}$ to any of the $q_{1}, q_{2}$, or $q_{4}$. We see that only $q_{1}$ and $q_{2}$ qualify; $q_{0}, q_{3}, q_{4}$ do not. The resulting automaton for $L_{1} / L_{2}$ is shown in the following Figure.


Check it to see that the construction works. The idea is generalized in the next theorem.

### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)

To apply this to our present case, we check each state $q_{0}, q_{1}, q_{2}, q_{3}, q_{4}, q_{5}$ to see whether there is a walk labeled $b b^{*}$ to any of the $q_{1}, q_{2}$, or $q_{4}$. We see that only $q_{1}$ and $q_{2}$ qualify; $q_{0}, q_{3}, q_{4}$ do not. The resulting automaton for $L_{1} / L_{2}$ is shown in the following Figure.


Check it to see that the construction works. The idea is generalized in the next
theorem.

### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)

To apply this to our present case, we check each state $q_{0}, q_{1}, q_{2}, q_{3}, q_{4}, q_{5}$ to see whether there is a walk labeled $b b^{*}$ to any of the $q_{1}, q_{2}$, or $q_{4}$. We see that only $q_{1}$ and $q_{2}$ qualify; $q_{0}, q_{3}, q_{4}$ do not. The resulting automaton for $L_{1} / L_{2}$ is shown in the following Figure.


Check it to see that the construction works. The idea is generalized in the next
theorem.

### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)

To apply this to our present case, we check each state $q_{0}, q_{1}, q_{2}, q_{3}, q_{4}, q_{5}$ to see whether there is a walk labeled $b b^{*}$ to any of the $q_{1}, q_{2}$, or $q_{4}$. We see that only $q_{1}$ and $q_{2}$ qualify; $q_{0}, q_{3}, q_{4}$ do not. The resulting automaton for $L_{1} / L_{2}$ is shown in the following Figure.


Check it to see that the construction works. The idea is generalized in the next

[^12]
### 4.1 Closure Properties of Regular Languages

## Example 4.4 (continuation)

To apply this to our present case, we check each state $q_{0}, q_{1}, q_{2}, q_{3}, q_{4}, q_{5}$ to see whether there is a walk labeled $b b^{*}$ to any of the $q_{1}, q_{2}$, or $q_{4}$. We see that only $q_{1}$ and $q_{2}$ qualify; $q_{0}, q_{3}, q_{4}$ do not. The resulting automaton for $L_{1} / L_{2}$ is shown in the following Figure.


Check it to see that the construction works. The idea is generalized in the next theorem.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.4

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

$$
\delta^{*}\left(q_{i}, y\right)=q_{f} \in F
$$

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$ is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the construction for the intersection of two regular languages given in Theorem 4.1, finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$. If there is any path between its initial vertex and any final vertex, then $L_{2} \cap L\left(M_{i}\right)$ is not empty. In that case, add $q_{i}$ to $\widehat{F}$. Repeating this for every $q_{i} \in Q$, we determine $\widehat{F}$ and thereby construct $\widehat{M}$.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

$$
\delta^{*}\left(q_{i}, y\right)=q_{f} \in F .
$$

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$ is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the construction for the intersection of two regular languages given in Theorem 4.1, finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$. If there is any path between its initial vertex and any final vertex, then $L_{2} \cap L\left(M_{i}\right)$ is not empty. In that case, add $q_{i}$ to $\widehat{F}$. Repeating this for every $q_{i} \in Q$, we determine $\widehat{F}$ and thereby construct $M$.

### 4.1 Closure Properties of Regular Languages

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\overparen{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$ is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the construction for the intersection of two regular languages given in Theorem 4.1, finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$. If there is any path between its initial vertex and any final vertex, then $L_{2} \cap L\left(M_{i}\right)$ is not empty. In that case, add $q_{i}$ to $\widehat{F}$. Repeating this for every $q_{i} \in Q$, we determine $\widehat{F}$ and thereby construct $\overparen{M}$

### 4.1 Closure Properties of Regular Languages

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\overparen{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$ is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the construction for the intersection of two regular languages given in Theorem 4.1, finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$. If there is any path between its initial vertex and any final vertex, then $L_{2} \cap L\left(M_{i}\right)$ is not empty. In that case, add $q_{i}$ to $\widehat{F}$. Repeating this for every $q_{i} \in Q$, we determine $\widehat{F}$ and thereby construct $\overparen{M}$

### 4.1 Closure Properties of Regular Languages

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

```
Proof. Let L L = L(M), where M= (Q,\Sigma,\delta,q0,F) is a dfa. We construct
another dfa }\widehat{M}=(Q,\Sigma,\delta,\mp@subsup{q}{0}{},\widehat{F})\mathrm{ as follows. For each }\mp@subsup{q}{i}{}\inQ\mathrm{ , determine if there
exists a word y}\in\mp@subsup{L}{2}{}\mathrm{ such that
This can be done by looking at dfa's Mi}=(Q,\Sigma,\delta,\mp@subsup{q}{i}{},F)\mathrm{ . The automaton M}\mp@subsup{M}{i}{
is M}\mathrm{ with the initial state q}\mp@subsup{q}{0}{}\mathrm{ replaced by }\mp@subsup{q}{i}{}\mathrm{ . We now determine whether there
exists a word }y\inL(\mp@subsup{M}{i}{})\mathrm{ that is also in L}\mp@subsup{L}{2}{}\mathrm{ . For this, we can use the
construction for the intersection of two regular languages given in Theorem 4.1
finding the transition graph for L2 \capL(M})\mathrm{ . If there is any path between its
initial vertex and any final vertex, then L}\mp@subsup{L}{2}{}\capL(\mp@subsup{M}{i}{})\mathrm{ is not empty. In that case,
add qi to }\widehat{F}\mathrm{ . Repeating this for every }\mp@subsup{q}{i}{}\inQ\mathrm{ , we determine }\widehat{F}\mathrm{ and thereby
construct M
```


### 4.1 Closure Properties of Regular Languages

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

## Proof.

another dfa $M=\left(Q, \Sigma, \delta, q_{0}, \bar{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$ is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the
construction for the intersection of two regular languages given in Theorem 4.1 finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$. If there is any path between its initial vertex and any final vertex, then $L_{2} \cap L\left(M_{i}\right)$ is not empty. In that case, add $q_{i}$ to $F$. Repeating this for every $q_{i} \in Q$, we determine $F$ and thereby construct $M$

### 4.1 Closure Properties of Regular Languages

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa.

```
another dfa M M (Q, \Sigma, \delta, q0, F) as follows. For each }\mp@subsup{q}{i}{}\inQ\mathrm{ , determine if there
exists a word y}\in\mp@subsup{L}{2}{}\mathrm{ such that
This can be done by looking at dfa's Mi = (Q, \Sigma, \delta, qi,F). The automaton Mi
is M}\mathrm{ with the initial state q}\mp@subsup{q}{0}{}\mathrm{ replaced by }\mp@subsup{q}{i}{}\mathrm{ . We now determine whether there
exists a word }y\inL(\mp@subsup{M}{i}{})\mathrm{ that is also in L2. For this, we can use the
construction for the intersection of two regular languages given in Theorem 4.1
finding the transition graph for L2 \capL(M})\mathrm{ . If there is any path between its
initial vertex and any final vertex, then L L2\capL(M}\mp@subsup{M}{i}{})\mathrm{ is not empty. In that case,
add qi to }\widehat{F}\mathrm{ . Repeating this for every }\mp@subsup{q}{i}{}\inQ\mathrm{ , we determine }\widehat{F}\mathrm{ and thereby
construct M
```


### 4.1 Closure Properties of Regular Languages

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows.

```
exists a word }y\in\mp@subsup{L}{2}{}\mathrm{ such that
```

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$
is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there
exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the
construction for the intersection of two regular languages given in Theorem 4.1
finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$. If there is any path between its
initial vertex and any final vertex, then $L_{2} \cap L\left(M_{i}\right)$ is not empty. In that case,
add $q_{i}$ to $\widehat{F}$. Repeating this for every $q_{i} \in Q$, we determine $\widehat{F}$ and thereby
construct $\overparen{M}$

### 4.1 Closure Properties of Regular Languages

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that


### 4.1 Closure Properties of Regular Languages

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

$$
\delta^{*}\left(q_{i}, y\right)=q_{f} \in F
$$

> This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$ is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the construction for the intersection of two regular languages given in Theorem 4.1 finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$. If there is any path between its initial vertex and any final vertex, then $L_{2} \cap L\left(M_{i}\right)$ is not empty. In that case, add $q_{i}$ to $\widehat{F}$. Repeating this for every $q_{i} \in Q$, we determine $\widehat{F}$ and thereby construct $M$

### 4.1 Closure Properties of Regular Languages

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

$$
\delta^{*}\left(q_{i}, y\right)=q_{f} \in F
$$

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. $\qquad$
is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the construction for the intersection of two regular languages given in Theorem 4.1, finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$. If there is any path between its initial vertex and any final vertex, then $L_{2} \cap L\left(M_{i}\right)$ is not empty. In that case, add $q_{i}$ to $\widehat{F}$. Repeating this for every $q_{i} \in Q$, we determine $\widehat{F}$ and thereby construct $M$

### 4.1 Closure Properties of Regular Languages

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

$$
\delta^{*}\left(q_{i}, y\right)=q_{f} \in F
$$

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$ is $M$ with the initial state $q_{0}$ replaced by $q_{i}$.
exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the
construction for the intersection of two regular languages given in Theorem 4.1, finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$. If there is any path between its initial vertex and any final vertex, then $L_{2} \cap L\left(M_{i}\right)$ is not empty. In that case, add $q_{i}$ to $\widehat{F}$. Repeating this for every $q_{i} \in Q$, we determine $\widehat{F}$ and thereby construct $\overparen{M}$

### 4.1 Closure Properties of Regular Languages

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

$$
\delta^{*}\left(q_{i}, y\right)=q_{f} \in F
$$

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$ is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$.
> construction for the intersection of two regular languages given in Theorem 4.1, finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$. If there is any path between its initial vertex and any final vertex, then $L_{2} \cap L\left(M_{i}\right)$ is not empty. In that case, add $q_{i}$ to $\widehat{F}$. Repeating this for every $q_{i} \in Q$, we determine $\widehat{F}$ and thereby construct $\widehat{M}$.

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

$$
\delta^{*}\left(q_{i}, y\right)=q_{f} \in F
$$

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$ is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the construction for the intersection of two regular languages given in Theorem 4.1, initial vertex and any final vertex, then $L_{2} \cap L\left(M_{i}\right)$ is not empty. In that case, add $q_{i}$ to $\widehat{F}$. Repeating this for every $q_{i} \in Q$, we determine $\widehat{F}$ and thereby construct $M$.

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

$$
\delta^{*}\left(q_{i}, y\right)=q_{f} \in F
$$

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$ is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the construction for the intersection of two regular languages given in Theorem 4.1, finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$.
add $q_{i}$ to $\widehat{F}$. Repeating this for every $q_{i} \in Q$, we determine $\widehat{F}$ and thereby construct $\widehat{M}$.

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

$$
\delta^{*}\left(q_{i}, y\right)=q_{f} \in F
$$

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$ is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the construction for the intersection of two regular languages given in Theorem 4.1, finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$. If there is any path between its initial vertex and any final vertex,

[^13]
## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

$$
\delta^{*}\left(q_{i}, y\right)=q_{f} \in F
$$

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$ is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the construction for the intersection of two regular languages given in Theorem 4.1, finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$. If there is any path between its initial vertex and any final vertex, then $L_{2} \cap L\left(M_{i}\right)$ is not empty.

[^14]
## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

$$
\delta^{*}\left(q_{i}, y\right)=q_{f} \in F
$$

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$ is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the construction for the intersection of two regular languages given in Theorem 4.1, finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$. If there is any path between its initial vertex and any final vertex, then $L_{2} \cap L\left(M_{i}\right)$ is not empty. In that case, add $q_{i}$ to $\widehat{F}$.

## Theorem 4.4

If $L_{1}$ and $L_{2}$ are regular languages, then $L_{1} / L_{2}$ is also regular. We say that the family of regular languages is closed under right quotient with a regular language.

Proof. Let $L_{1}=L(M)$, where $M=\left(Q, \Sigma, \delta, q_{0}, F\right)$ is a dfa. We construct another dfa $\widehat{M}=\left(Q, \Sigma, \delta, q_{0}, \widehat{F}\right)$ as follows. For each $q_{i} \in Q$, determine if there exists a word $y \in L_{2}$ such that

$$
\delta^{*}\left(q_{i}, y\right)=q_{f} \in F
$$

This can be done by looking at dfa's $M_{i}=\left(Q, \Sigma, \delta, q_{i}, F\right)$. The automaton $M_{i}$ is $M$ with the initial state $q_{0}$ replaced by $q_{i}$. We now determine whether there exists a word $y \in L\left(M_{i}\right)$ that is also in $L_{2}$. For this, we can use the construction for the intersection of two regular languages given in Theorem 4.1, finding the transition graph for $L_{2} \cap L\left(M_{i}\right)$. If there is any path between its initial vertex and any final vertex, then $L_{2} \cap L\left(M_{i}\right)$ is not empty. In that case, add $q_{i}$ to $\widehat{F}$. Repeating this for every $q_{i} \in Q$, we determine $\widehat{F}$ and thereby construct $\widehat{M}$.

### 4.1 Closure Properties of Regular Languages

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that

$$
\delta^{*}\left(q_{0}, x\right)=q
$$

and

$$
\delta^{*}(q, y) \in F .
$$

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$.
Conversely, for any $x$ accepted by $\widehat{M}$, we have

$$
\delta^{*}\left(q_{0}, x\right)=q \in \widehat{F} .
$$

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that

$$
L(\widehat{M})=L_{1} / L_{2}
$$

and from this that $L_{1} / L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that
so that there must be some $q \in Q$ such that
$\delta^{*}\left(q_{0}, x\right)=q$
and

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$
Conversely, for any $x$ accepted by $\widehat{M I}$, we have

$$
\delta^{*}\left(q_{0}, x\right)=q \in \widehat{F}
$$

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that

$$
L(\widehat{M})=L_{1} / L_{2},
$$

and from this that $L_{1} / L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that
so that there must be some $q \in Q$ such that
and

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$
Conversely, for any $x$ accepted by $\widehat{M}$, we have

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that

$$
L(\widehat{M})=L_{1} / L_{2}
$$

and from this that $L_{1} / L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that
so that there must be some $q \in Q$ such that
and

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$
Conversely, for any $x$ accepted by $\widehat{M}$, we have

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that

$$
L(\widehat{M})=L_{1} / L_{2},
$$

and from this that $L_{1} / L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that
and

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$. Conversely, for any $x$ accepted by $\widehat{M}$, we have

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that

$$
L(\widehat{M})=L_{1} / L_{2}
$$

and from this that $L_{1} / L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that
and

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$
Conversely, for any $x$ accepted by $\widehat{M}$, we have
But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that

$$
L(\widehat{M})=L_{1} / L_{2},
$$

and from this that $L_{1} / L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that

$$
\delta^{*}\left(q_{0}, x\right)=q
$$

and

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$. Conversely, for any $x$ accepted by $\widehat{M}$, we have

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that

$$
L(\widehat{M})=L_{1} / L_{2},
$$

and from this that $L_{1} / L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that

$$
\delta^{*}\left(q_{0}, x\right)=q
$$

and

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$.
Conversely, for any $x$ accepted by $\widehat{M}$, we have
But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that

$$
L(\widehat{M})=L_{1} / L_{2},
$$

and from this that $L_{1} / L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that

$$
\delta^{*}\left(q_{0}, x\right)=q
$$

and

$$
\delta^{*}(q, y) \in F .
$$

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$ Conversely, for any $x$ accepted by $\widehat{M}$, we have

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that

and from this that $L_{1} / L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that

$$
\delta^{*}\left(q_{0}, x\right)=q
$$

and

$$
\delta^{*}(q, y) \in F .
$$

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$.
Conversely, for any $x$ accepted by $\widehat{M}$, we have
But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that
and from this that $L_{1} / L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that

$$
\delta^{*}\left(q_{0}, x\right)=q
$$

and

$$
\delta^{*}(q, y) \in F .
$$

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$. Conversely, for any $x$ accepted by $\widehat{M}$, we have

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that
and from this that $L_{1} / L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that

$$
\delta^{*}\left(q_{0}, x\right)=q
$$

and

$$
\delta^{*}(q, y) \in F .
$$

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$.
Conversely, for any $x$ accepted by $\widehat{M}$, we have

$$
\delta^{*}\left(q_{0}, x\right)=q \in \widehat{F} .
$$

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that
and from this that $L_{1} / L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that

$$
\delta^{*}\left(q_{0}, x\right)=q
$$

and

$$
\delta^{*}(q, y) \in F .
$$

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$.
Conversely, for any $x$ accepted by $\widehat{M}$, we have

$$
\delta^{*}\left(q_{0}, x\right)=q \in \widehat{F} .
$$

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$.
conclude that
and from this that $L_{1} / L_{2}$ is regular.

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that

$$
\delta^{*}\left(q_{0}, x\right)=q
$$

and

$$
\delta^{*}(q, y) \in F .
$$

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$.
Conversely, for any $x$ accepted by $\widehat{M}$, we have

$$
\delta^{*}\left(q_{0}, x\right)=q \in \widehat{F} .
$$

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$.

[^15]and from this that $L_{1} / L_{2}$ is regular

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that

$$
\delta^{*}\left(q_{0}, x\right)=q
$$

and

$$
\delta^{*}(q, y) \in F .
$$

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$.
Conversely, for any $x$ accepted by $\widehat{M}$, we have

$$
\delta^{*}\left(q_{0}, x\right)=q \in \widehat{F} .
$$

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that
and from this that $L_{1} / L_{2}$ is regular.

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that

$$
\delta^{*}\left(q_{0}, x\right)=q
$$

and

$$
\delta^{*}(q, y) \in F .
$$

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$.
Conversely, for any $x$ accepted by $\widehat{M}$, we have

$$
\delta^{*}\left(q_{0}, x\right)=q \in \widehat{F} .
$$

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that

$$
L(\widehat{M})=L_{1} / L_{2},
$$

and from this that $L_{1} / L_{2}$ is regular.

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that

$$
\delta^{*}\left(q_{0}, x\right)=q
$$

and

$$
\delta^{*}(q, y) \in F .
$$

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$.
Conversely, for any $x$ accepted by $\widehat{M}$, we have

$$
\delta^{*}\left(q_{0}, x\right)=q \in \widehat{F} .
$$

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that

$$
L(\widehat{M})=L_{1} / L_{2},
$$

and from this that $L_{1} / L_{2}$ is regular.

To prove that $L(\widehat{M})=L_{1} / L_{2}$, let $x$ be any element of $L_{1} / L_{2}$. Then there must be a word $y \in L_{2}$ such that $x y \in L_{1}$. This implies that

$$
\delta^{*}\left(q_{0}, x y\right) \in F,
$$

so that there must be some $q \in Q$ such that

$$
\delta^{*}\left(q_{0}, x\right)=q
$$

and

$$
\delta^{*}(q, y) \in F .
$$

Therefore, by construction, $q \in \widehat{F}$, and $\widehat{M}$ accepts $x$ because $\delta^{*}\left(q_{0}, x\right)$ is in $\widehat{F}$.
Conversely, for any $x$ accepted by $\widehat{M}$, we have

$$
\delta^{*}\left(q_{0}, x\right)=q \in \widehat{F} .
$$

But again by construction, this implies that there exists a word $y \in L_{2}$ such that $\delta^{*}(q, y) \in F$. Therefore, $x y$ is in $L_{1}$, and $x$ is in $L_{1} / L_{2}$. We therefore conclude that

$$
L(\widehat{M})=L_{1} / L_{2},
$$

and from this that $L_{1} / L_{2}$ is regular.

### 4.1 Closure Properties of Regular Languages

## Example 4.5

### 4.1 Closure Properties of Regular Languages

## Example 4.5

Find $L_{1} / L_{2}$ for

$$
\begin{aligned}
& L_{1}=L\left(a^{*} b a a^{*}\right), \\
& L_{2}=L\left(a b^{*}\right) .
\end{aligned}
$$

We first find a dfa that accepts $L_{1}$. This is easy, and a solution is given in the following Figure.


The example is simple enough so that we can skip the formalities of the construction.

### 4.1 Closure Properties of Regular Languages

Example 4.5
Find $L_{1} / L_{2}$ for

$$
\begin{aligned}
& L_{1}=L\left(a^{*} b a a^{*}\right), \\
& L_{2}=L\left(a b^{*}\right) .
\end{aligned}
$$

We first find a dfa that accepts $L_{1}$. This is easy, and a solution is given in the following Figure.


The example is simple enough so that we can skip the formalities of the construction.

### 4.1 Closure Properties of Regular Languages

Example 4.5
Find $L_{1} / L_{2}$ for

$$
\begin{aligned}
L_{1} & =L\left(a^{*} b a a^{*}\right), \\
L_{2} & =L\left(a b^{*}\right)
\end{aligned}
$$

We first find a dfa that accepts $L_{1}$. This is easy, and a solution is given in the following Figure.


The example is simple enough so that we can skip the formalities of the construction.

### 4.1 Closure Properties of Regular Languages

Example 4.5
Find $L_{1} / L_{2}$ for

$$
\begin{aligned}
& L_{1}=L\left(a^{*} b a a^{*}\right) \\
& L_{2}=L\left(a b^{*}\right)
\end{aligned}
$$

We first find a dfa that accepts $L_{1}$. This is easy, and a solution is given in the following Figure.


The example is simple enough so that we can skip the formalities of the construction.

### 4.1 Closure Properties of Regular Languages

## Example 4.5

Find $L_{1} / L_{2}$ for

$$
\begin{aligned}
& L_{1}=L\left(a^{*} b a a^{*}\right), \\
& L_{2}=L\left(a b^{*}\right)
\end{aligned}
$$

We first find a dfa that accepts $L_{1}$. This is easy, and a solution is given in the following Figure.


The example is simple enough so that we can skip the formalities of the construction.

### 4.1 Closure Properties of Regular Languages

## Example 4.5

Find $L_{1} / L_{2}$ for

$$
\begin{aligned}
& L_{1}=L\left(a^{*} b a a^{*}\right) \\
& L_{2}=L\left(a b^{*}\right)
\end{aligned}
$$

We first find a dfa that accepts $L_{1}$. This is easy, and a solution is given in the following Figure.


The example is simple enough so that we can skip the formalities of the construction.

### 4.1 Closure Properties of Regular Languages

## Example 4.5

Find $L_{1} / L_{2}$ for

$$
\begin{aligned}
& L_{1}=L\left(a^{*} b a a^{*}\right) \\
& L_{2}=L\left(a b^{*}\right)
\end{aligned}
$$

We first find a dfa that accepts $L_{1}$. This is easy, and a solution is given in the following Figure.


The example is simple enough so that we can skip the formalities of the construction.

### 4.1 Closure Properties of Regular Languages

Example 4.5 (continuation)

### 4.1 Closure Properties of Regular Languages

Example 4.5 (continuation)
From the graph in the previous Figure it is quite evident that
$L\left(M_{0}\right) \cap L_{2}=\varnothing$,
$I\left(M_{1}\right) \cap I_{2}=\{a\}=\varnothing$,
$L\left(M_{2}\right) \cap L_{2}=\{a\} \neq \varnothing$,
$L\left(M_{3}\right) \cap L_{2}=\varnothing$.
Therefore, the automaton accepting $L_{1} / L_{2}$ is determined. The result is shown in the Figure.


### 4.1 Closure Properties of Regular Languages

## Example 4.5 (continuation)

From the graph in the previous Figure it is quite evident that
$L\left(M_{1}\right) \cap L_{2}=\{a\} \neq \varnothing$,
$L\left(M_{2}\right) \cap L_{2}=\{a\} \neq \varnothing$.
$L\left(M_{3}\right) \cap L_{2}=\varnothing$.
Therefore, the automaton accepting $L_{1} / L_{2}$ is determined. The result is shown in the Figure.


### 4.1 Closure Properties of Regular Languages

Example 4.5 (continuation)
From the graph in the previous Figure it is quite evident that

$$
\begin{aligned}
& L\left(M_{0}\right) \cap L_{2}=\varnothing \\
& L\left(M_{1}\right) \cap L_{2}=\{a\} \neq \varnothing \\
& L\left(M_{2}\right) \cap L_{2}=\{a\} \neq \varnothing \\
& L\left(M_{3}\right) \cap L_{2}=\varnothing
\end{aligned}
$$

Therefore, the automaton accepting $L_{1} / L_{2}$ is determined. The result is shown in the Figure.


### 4.1 Closure Properties of Regular Languages

Example 4.5 (continuation)
From the graph in the previous Figure it is quite evident that

$$
\begin{aligned}
& L\left(M_{0}\right) \cap L_{2}=\varnothing \\
& L\left(M_{1}\right) \cap L_{2}=\{a\} \neq \varnothing \\
& L\left(M_{2}\right) \cap L_{2}=\{a\} \neq \varnothing \\
& L\left(M_{3}\right) \cap L_{2}=\varnothing
\end{aligned}
$$

Therefore, the automaton accepting $L_{1} / L_{2}$ is determined. The result is shown in the Figure.


### 4.1 Closure Properties of Regular Languages

Example 4.5 (continuation)
From the graph in the previous Figure it is quite evident that

$$
\begin{aligned}
& L\left(M_{0}\right) \cap L_{2}=\varnothing \\
& L\left(M_{1}\right) \cap L_{2}=\{a\} \neq \varnothing \\
& L\left(M_{2}\right) \cap L_{2}=\{a\} \neq \varnothing \\
& L\left(M_{3}\right) \cap L_{2}=\varnothing
\end{aligned}
$$

Therefore, the automaton accepting $L_{1} / L_{2}$ is determined. The result is shown in the Figure.

### 4.1 Closure Properties of Regular Languages

Example 4.5 (continuation)
From the graph in the previous Figure it is quite evident that

$$
\begin{aligned}
& L\left(M_{0}\right) \cap L_{2}=\varnothing \\
& L\left(M_{1}\right) \cap L_{2}=\{a\} \neq \varnothing \\
& L\left(M_{2}\right) \cap L_{2}=\{a\} \neq \varnothing \\
& L\left(M_{3}\right) \cap L_{2}=\varnothing
\end{aligned}
$$

Therefore, the automaton accepting $L_{1} / L_{2}$ is determined. The result is shown in the Figure.


### 4.1 Closure Properties of Regular Languages

## Example 4.5 (continuation)

### 4.1 Closure Properties of Regular Languages

## Example 4.5 (continuation)

It accepts the language denoted by the regular expression of $a^{*} b+a^{*} b a a^{*}$, which can be simplified to $a^{*} b a^{*}$. Thus $L_{1} / L_{2}=L\left(a^{*} b a^{*}\right)$.

### 4.1 Closure Properties of Regular Languages

## Example 4.5 (continuation)

It accepts the language denoted by the regular expression of $a^{*} b+a^{*} b a a^{*}$, which can be simplified to $a^{*} b a^{*}$.

### 4.1 Closure Properties of Regular Languages

## Example 4.5 (continuation)

It accepts the language denoted by the regular expression of $a^{*} b+a^{*} b a a^{*}$, which can be simplified to $a^{*} b a^{*}$. Thus $L_{1} / L_{2}=L\left(a^{*} b a^{*}\right)$.

## Thank You for attention!


[^0]:    is regular.

[^1]:    is regular

[^2]:    is regular.

[^3]:    results; we select only two typical ones.

[^4]:    To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix, belongs to $L_{1} / L_{2}$.

[^5]:    To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix, belongs to $L_{1} / L_{2}$.

[^6]:    To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix, belongs to $L_{1} / L_{2}$

[^7]:    To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix belongs to $L_{1} / L_{2}$.

[^8]:    To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix belongs to $L_{1} / L_{2}$

[^9]:    To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix, belongs to $L_{1} / L_{2}$.

[^10]:    To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix, belongs to $L_{1} / L_{2}$.

[^11]:    To form the right quotient of $L_{1}$ with $L_{2}$, we take all the strings in $L_{1}$ that have a suffix belonging to $L_{2}$. Every such string, after removal of this suffix, belongs to $L_{1} / L_{2}$.

[^12]:    theorem

[^13]:    add $q_{i}$ to $F$. Repeating this for every $q_{i} \in Q$, we determine $F$ and thereby construct $M$.

[^14]:    add $q_{i}$ to $F$. Repeating this for every $q_{i} \in Q$, we determine $F$ and thereby construct $M$.

[^15]:    conclude that

