

COMP2022: Formal Languages and Logic

2018 Semester 2, Week 6

# Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat `edu_assist_pro`



THE UNIVERSITY OF  
SYDNEY

COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

# Assignment Project Exam Help

WARNING

This  
on be  
Cop

<https://eduassistpro.github.io>

The material in this communication may be subject  
under the Act. Any further copying or communicat  
material by you may be subject of copyright protec

Add WeChat [edu\\_assist\\_pro](https://edu_assist_pro)

**Do not remove this notice.**

## OUTLINE

# Assignment Project Exam Help

## ► Regular Expressions

- ## ► Regular Expressions

<https://eduassistpro.github.io>

# Add WeChat edu\_assist\_pr

## OUTLINE

# Assignment Project Exam Help

- ▶ Regular Expressions

▶ <https://eduassistpro.github.io>

▶ Add WeChat edu\_assist\_pro

- ▶ Proving if a language is, *or is not*

## EQUIVALENCE OF REGEX AND FA

Theorem:

A language is regular if and only if a regular expression describes it.

Proo

Sho

<https://eduassistpro.github.io>

1.  $\Rightarrow$

Show that for each RegEx, there exists an NF  
recognises the same language

2. FA  $\Rightarrow$  RegEx:

Show that for each NFA, there exists a RegEx which  
recognises the same language

Add WeChat edu\_assist\_pro

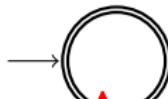
FROM REGEX TO FA: ATOMIC CASES

## Automaton for $\emptyset$

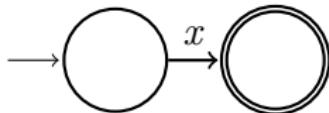


Auto

<https://eduassistpro.github.io>



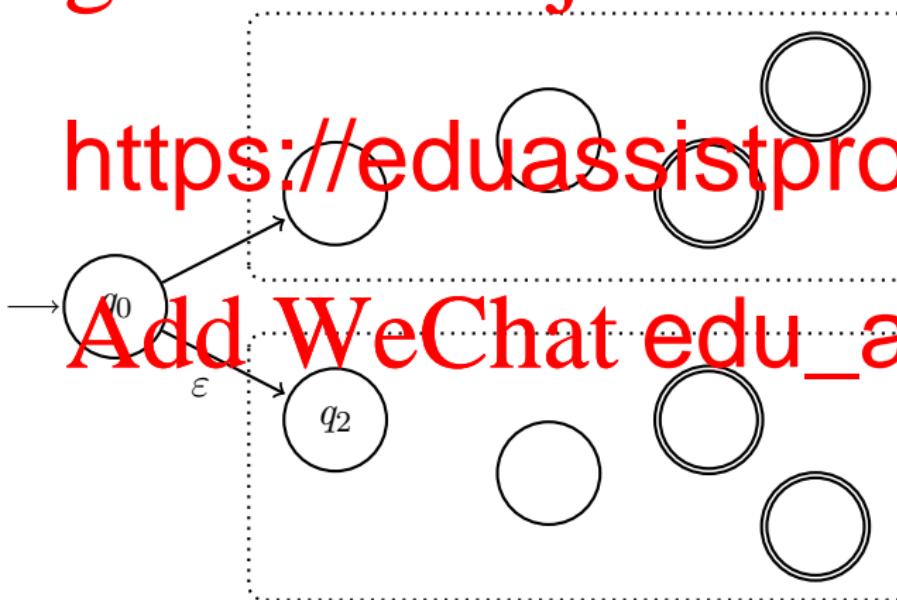
Automaton for  $x \in \Sigma$



## NFA FOR REGULAR OPERATIONS: UNION

Let  $M_1$  and  $M_2$  be automata recognising  $L(R_1)$  and  $L(R_2)$

Then an automaton  $M$  for  $R_1 \cap R_2$  is



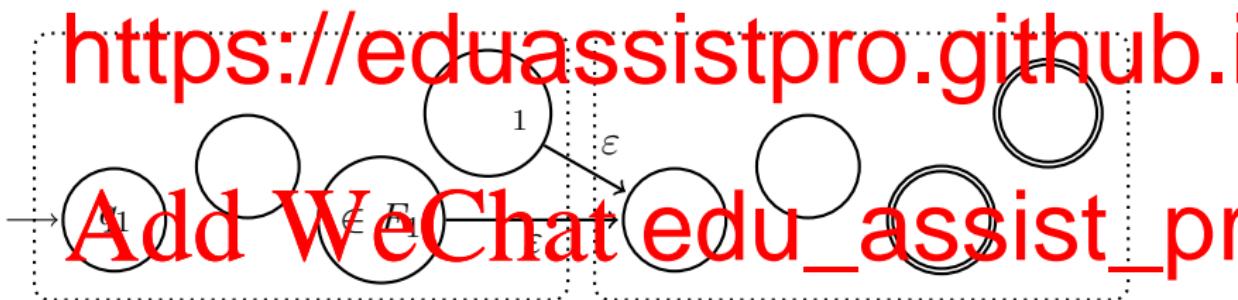
<https://eduassistpro.github.io/>

## NFA FOR REGULAR OPERATIONS: CONCATENATION

Let  $M_1$  and  $M_2$  be automata recognising  $L(R_1)$  and  $L(R_2)$

# Assignment Project Exam Help

Then an automaton  $M$  for  $R \cap R'$  is:



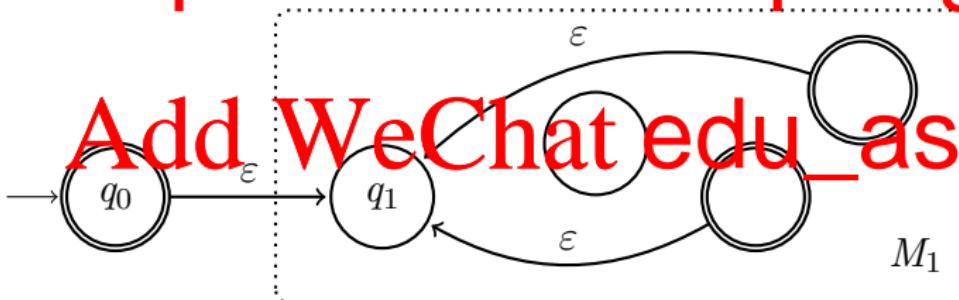
Reminder: the accept states of  $M_1$  are *not* accept states in  $M$ .

## NFA FOR REGULAR OPERATIONS: STAR CLOSURE

Assignment Project Exam Help

The

<https://eduassistpro.github.io>

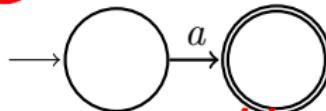


Add WeChat edu\_assist\_pr

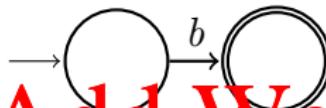
## EXAMPLE: $a \mid bc^*$ : FROM REGEX TO NFA

1. Construct automata for the atomic regular expressions  $a, b, c$

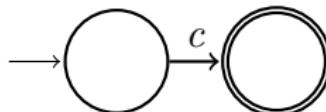
# Assignment Project Exam Help



<https://eduassistpro.github.io>



Add WeChat edu\_assist\_pro



## EXAMPLE: $a \mid bc^*$ : FROM REGEX TO NFA

1. Construct automata for the atomic regular expressions  $a, b, c$
2. Use the Star Closure operation to find an automaton for  $c^*$

# Assignment Project Exam Help

<https://eduassistpro.github.io>



Add WeChat edu\_assist\_pro

## EXAMPLE: $a \mid bc^*$ : FROM REGEX TO NFA

1. Construct automata for the atomic regular expressions  $a, b, c$
2. Use the Star Closure operation to find an automaton for  $c^*$
3. Use the Concatenation operation to find an automaton for  $bc^*$

# Assignment Project Exam Help

<https://eduassistpro.github.io>

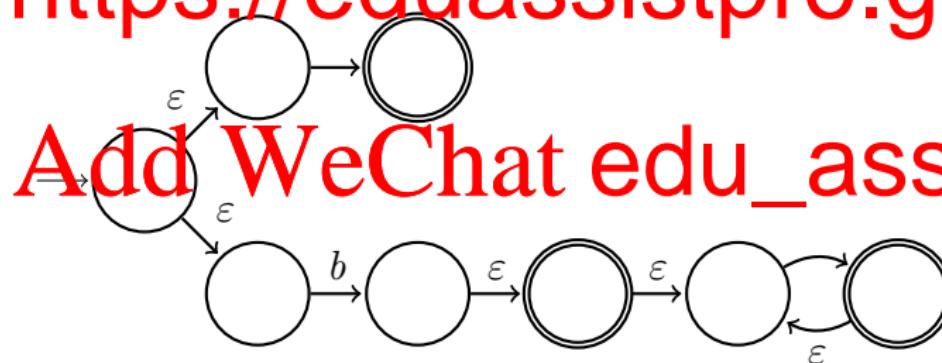


EXAMPLE:  $a \mid bc^*$ : FROM REGEx TO NFA

- Assignment Project Exam Help**

  1. Construct automata for the atomic regular expressions  $a, b, c$
  2. Use the Star Closure operation to find an automaton for  $c^*$
  3. Use the Concatenation operation to find an automaton for  $bc^*$
  - 4.

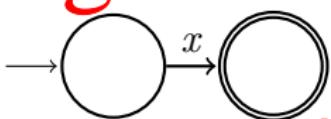
<https://eduassistpro.github.io>



## FROM NFA TO REGEx: SIMPLE EXAMPLES

This automaton gives the RegEx  $x$

# Assignment Project Exam Help

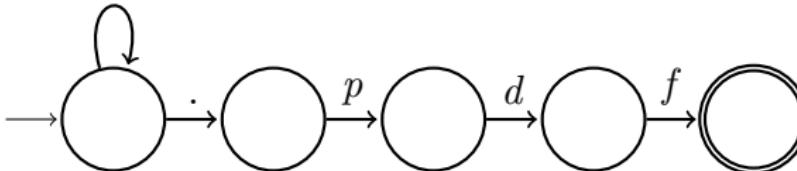


<https://eduassistpro.github.io>

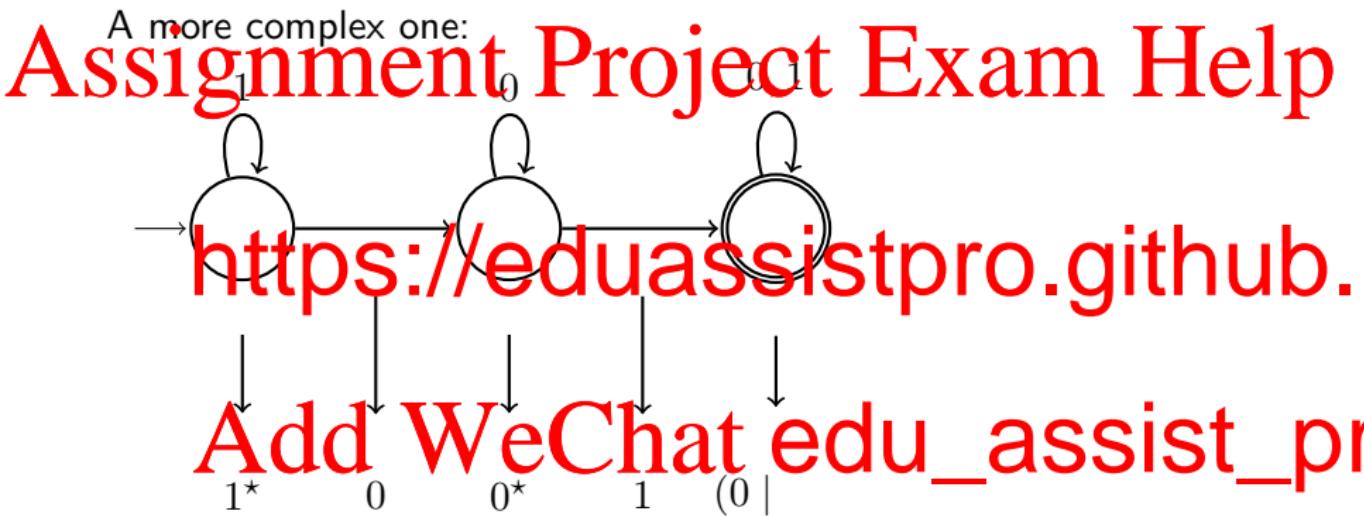
This o

||| |

a...s, A...Z  
Add WeChat edu\_assist\_pro



## FROM NFA TO REGEx: SIMPLE EXAMPLES



$1^*00^*1(0 \mid 1)^*$

## CONCEPT

Algorithm to convert an NFA to a RegEx:

# Assignment Project Exam Help

1. Convert the NFA to a Generalised NFA (GNFA)

<https://eduassistpro.github.io>

2. Progressively eliminate all states but the start state

Add WeChat [edu\\_assist\\_pro](https://edu_assist_pro)

3. The transition between the start and the accept state is now a regular expression describing  $L$

## GENERALISED NFA (GNFA)

The start state  $q_s$  is non-accepting and has no incoming transitions.

# Assignment Project Exam Help

There is exactly one transition between each ordered pair of states, label

<https://eduassistpro.github.io/>



$$\{abc, abca, abcbb, aaaaaabc\} \subseteq L$$

## GENERALISED NFA (GNFA)

The start state  $q_s$  is non-accepting and has no incoming transitions.

# Assignment Project Exam Help

There is exactly one transition between each ordered pair of states, label

<https://eduassistpro.github.io/>



We do not show the  $\emptyset$  transitions (why not?)

$$\{abc, abca, abcbb, aaaaaabc\} \subseteq L$$

## CONVERTING AN NFA TO A GNFA

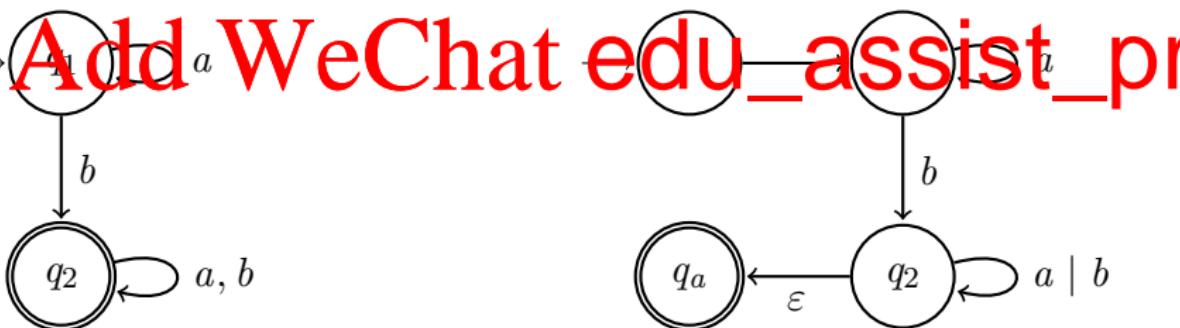
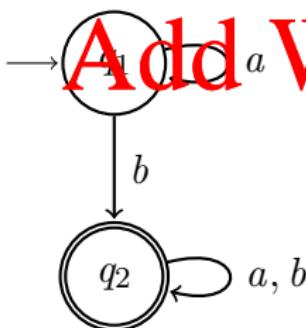
Create a new non-accepting start state  $q_s$ , with a  $\epsilon$ -transition to the original start state.

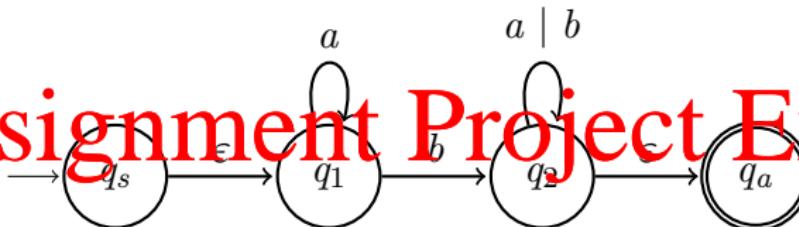
# Assignment Project Exam Help

Create a new accept state  $q_a$  with  $\epsilon$ -transitions from the original accept states.

Labels from transitions

$q_i$        $q_j$



SIMPLE EXAMPLE: ELIMINATE STATE  $q_1$ 

There

<https://eduassistpro.github.io/>

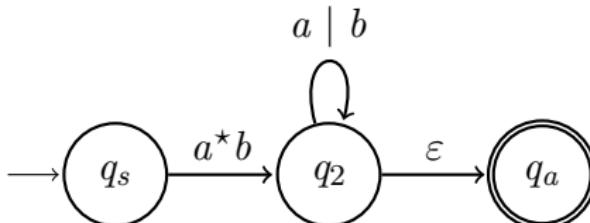
$$q_1 \rightarrow q_1$$

$$a^*$$

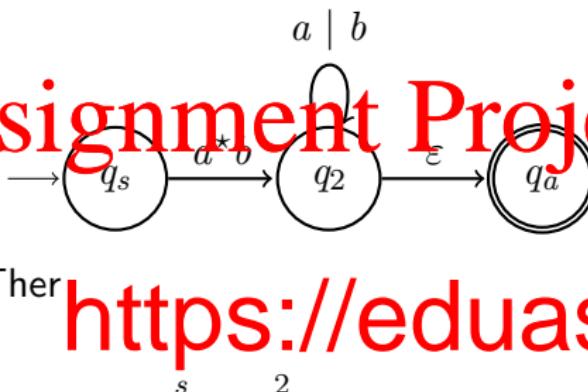
$$q_1 \rightarrow q_2$$

We set the transition  $q_s \rightarrow q_2$  to the union of its old value ( $\emptyset$ ). This is  $\epsilon a^* b \mid \emptyset$ , whic

$$a \ b$$



SIMPLE EXAMPLE: ELIMINATE STATE  $q_2$

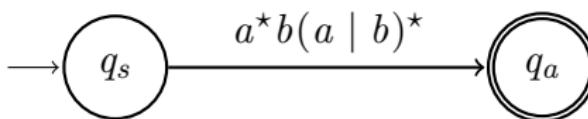


Then

<https://eduassistpro.github.io>

$q_2 \rightarrow q_2$  any number of times

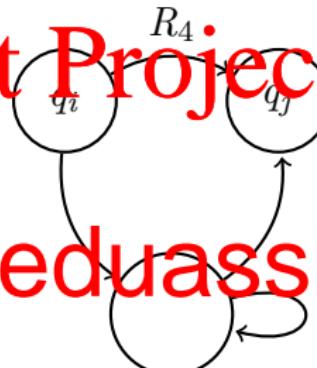
Add WeChat edu\_assist\_pro



The language of the original automaton is  $a^*b(a|b)^*$

GENERAL METHOD TO ELIMINATE STATE  $q_{elim}$ 

Consider each pair  $(q_i, q_j)$  where  $q_i, q_j \in Q \setminus \{q_{elim}\}$



<https://eduassistpro.github.io>

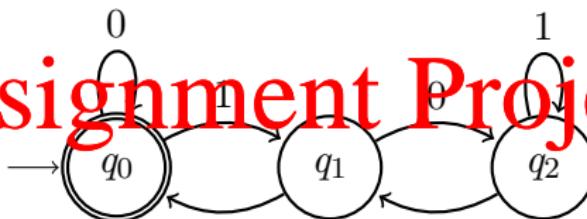
Replace the transition from  $q_i$  to  $q_j$  with

Add WeChat [edu\\_assist\\_pro](https://edu_assist_pro)

Note:

- ▶ Possibly  $q_i = q_j$
- ▶ Recall that pairs are ordered, so we also consider  $(q_j, q_i)$
- ▶ If there is no transition  $R_x$ , then  $R_x = \emptyset$

## EXAMPLE 2: DIVISIBLE BY 3

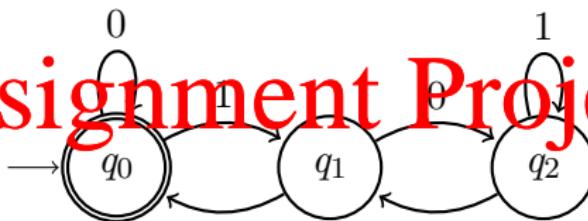


Assignment Project Exam Help  
<https://eduassistpro.github.io>

Conv

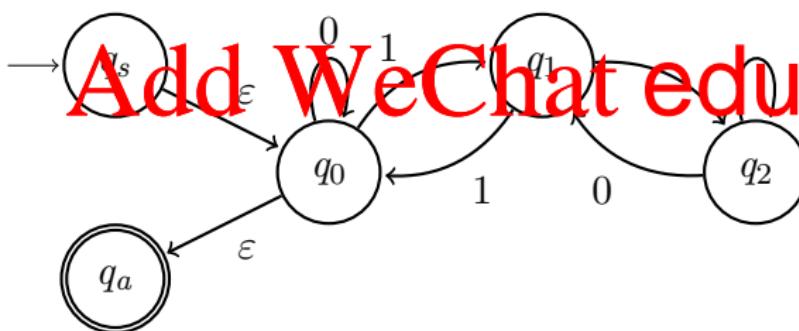
Add WeChat edu\_assist\_pro

### EXAMPLE 2: DIVISIBLE BY 3



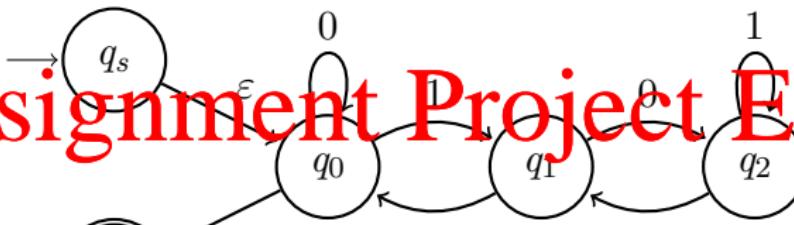
<https://eduassistpro.github.io/>

Copy



Add WeChat `edu_assist_pr`

## EXAMPLE 2: DIVISIBLE BY 3



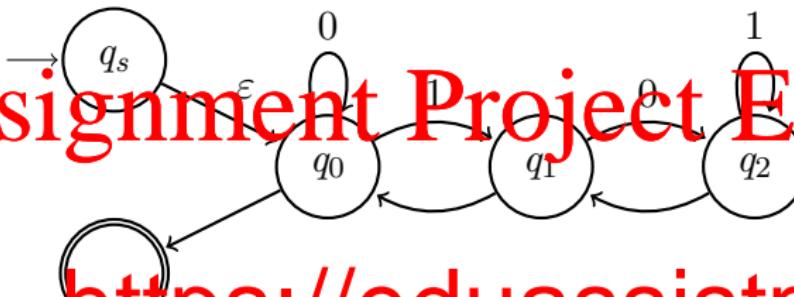
Assignment Project Exam Help  
<https://eduassistpro.github.io>

We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- ▶
- ▶
- ▶
- ▶

## EXAMPLE 2: DIVISIBLE BY 3



<https://eduassistpro.github.io>

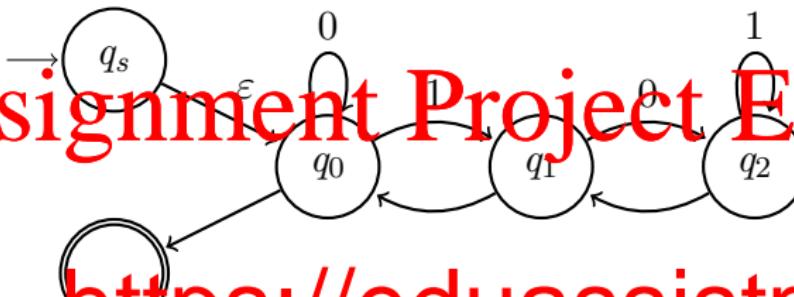
We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- $(q_s, q_1)$  :
- $(q_s, q_a)$  :
- $(q_1, q_1)$  :
- $(q_1, q_a)$  :

Add WeChat edu\_assist\_pro

## EXAMPLE 2: DIVISIBLE BY 3



<https://eduassistpro.github.io>

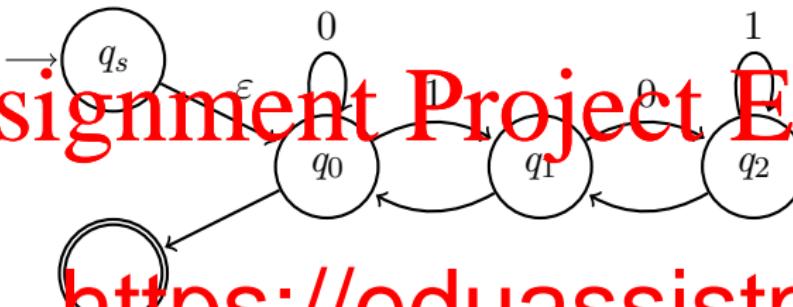
We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- $(q_s, q_1) : R_1 = \epsilon,$
- $(q_s, q_a) :$
- $(q_1, q_1) :$
- $(q_1, q_a) :$

Add WeChat edu\_assist\_pro

## EXAMPLE 2: DIVISIBLE BY 3



<https://eduassistpro.github.io>

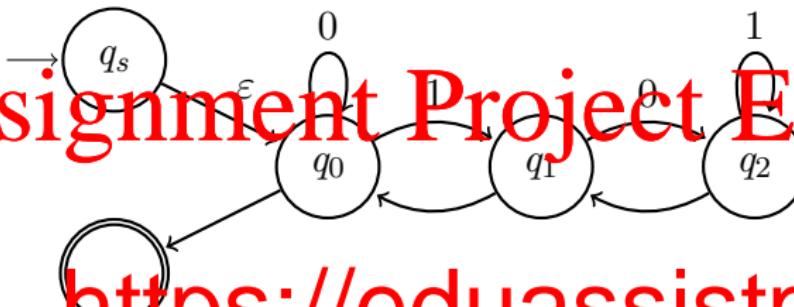
We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- $(q_s, q_1) : R_1 = \varepsilon, R_2 = 0,$
- $(q_s, q_a) :$
- $(q_1, q_1) :$
- $(q_1, q_a) :$

Add WeChat edu\_assist\_pro

## EXAMPLE 2: DIVISIBLE BY 3



<https://eduassistpro.github.io>

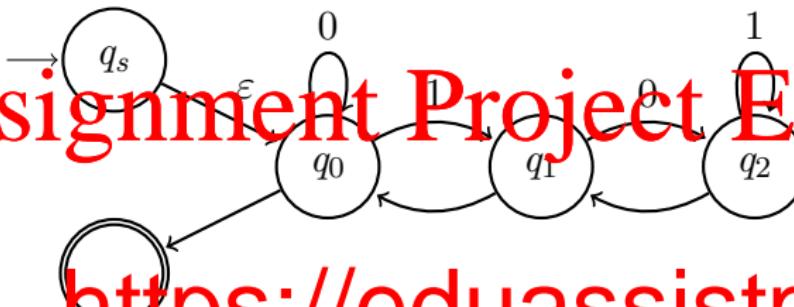
We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- $(q_s, q_1) : R_1 = \epsilon, R_2 = 0, R_3 = 1,$
- $(q_s, q_a) :$
- $(q_1, q_1) :$
- $(q_1, q_a) :$

Add WeChat edu\_assist\_pro

## EXAMPLE 2: DIVISIBLE BY 3



<https://eduassistpro.github.io>

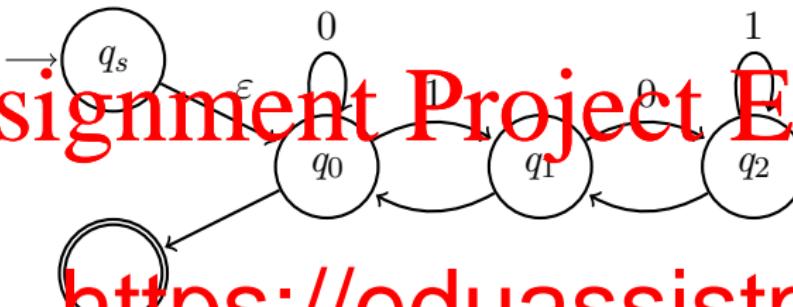
We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- $(q_s, q_1) : R_1 = \epsilon, R_2 = 0, R_3 = 1,$
- $(q_s, q_a) :$
- $(q_1, q_1) :$
- $(q_1, q_a) :$

Add WeChat edu\_assist\_pro

## EXAMPLE 2: DIVISIBLE BY 3



<https://eduassistpro.github.io>

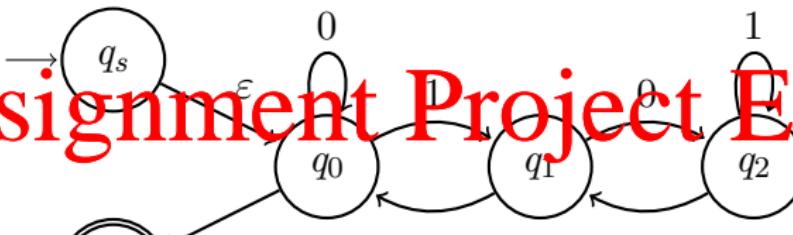
We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- $(q_s, q_1) : R_1 = \epsilon, R_2 = 0, R_3 = 1,$
- $(q_s, q_a) :$
- $(q_1, q_1) :$
- $(q_1, q_a) :$

Add WeChat edu\_assist\_pro

## EXAMPLE 2: DIVISIBLE BY 3



<https://eduassistpro.github.io>

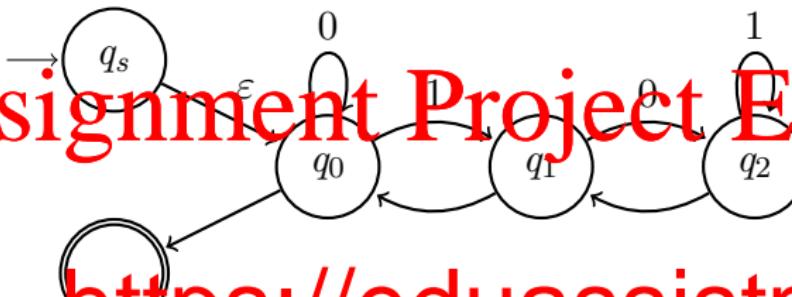
We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- $(q_s, q_1) : R_1 = \varepsilon, R_2 = 0, R_3 = 1,$
- $(q_s, q_a) :$
- $(q_1, q_1) :$
- $(q_1, q_a) :$

Add WeChat edu\_assist\_pro

## EXAMPLE 2: DIVISIBLE BY 3



<https://eduassistpro.github.io>

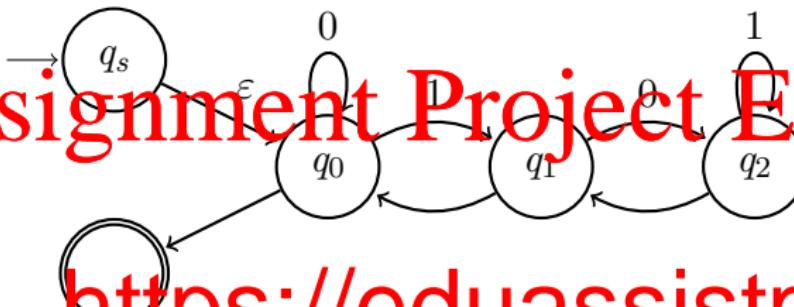
We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- $(q_s, q_1) : R_1 = \epsilon, R_2 = 0, R_3 = 1,$
- $(q_s, q_a) : R_1 = \epsilon,$
- $(q_1, q_1) :$
- $(q_1, q_a) :$

Add WeChat edu\_assist\_pro

## EXAMPLE 2: DIVISIBLE BY 3



<https://eduassistpro.github.io>

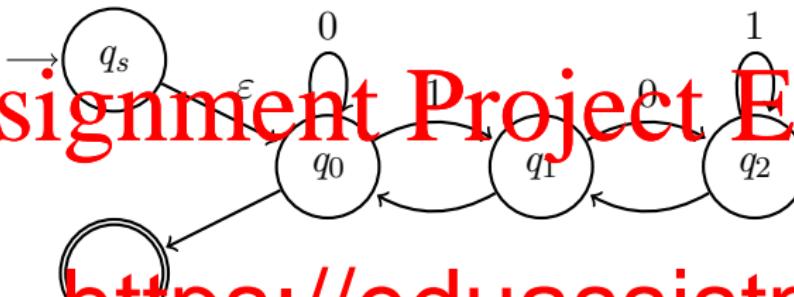
We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- $(q_s, q_1) : R_1 = \epsilon, R_2 = 0, R_3 = 1,$
- $(q_s, q_a) : R_1 = \epsilon, R_2 = 0,$
- $(q_1, q_1) :$
- $(q_1, q_a) :$

Add WeChat edu\_assist\_pro

## EXAMPLE 2: DIVISIBLE BY 3



<https://eduassistpro.github.io>

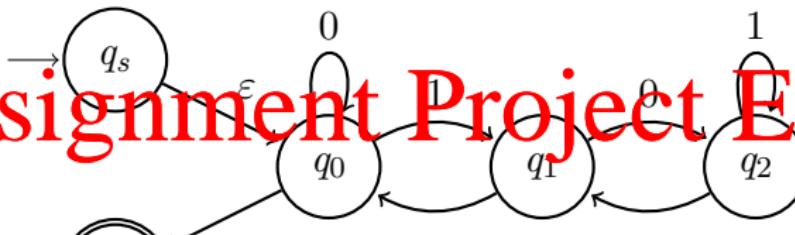
We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- $(q_s, q_1) : R_1 = \varepsilon, R_2 = 0, R_3 = 1,$
- $(q_s, q_a) : R_1 = \varepsilon, R_2 = 0, R_3 = \varepsilon,$
- $(q_1, q_1) :$
- $(q_1, q_a) :$

Add WeChat edu\_assist\_pro

## EXAMPLE 2: DIVISIBLE BY 3



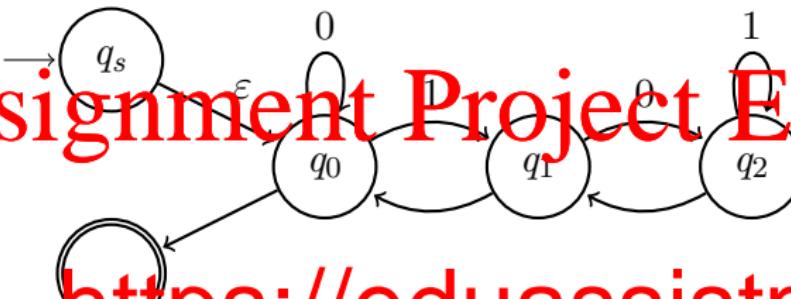
<https://eduassistpro.github.io>

We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- $(q_s, q_1) : R_1 = \epsilon, R_2 = 0, R_3 = 1,$
- $(q_s, q_a) : R_1 = \epsilon, R_2 = 0, R_3 = \epsilon, R_4 = \emptyset$
- $(q_1, q_1) :$
- $(q_1, q_a) :$

## EXAMPLE 2: DIVISIBLE BY 3



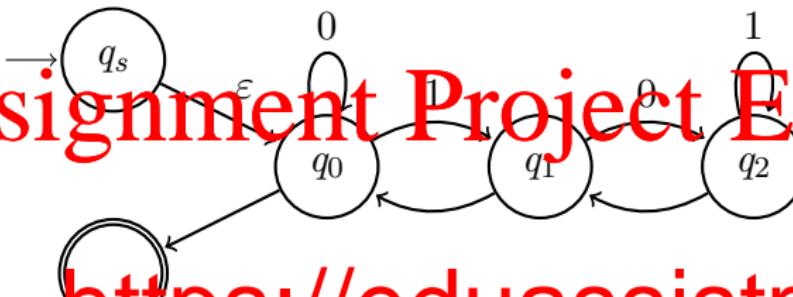
<https://eduassistpro.github.io>

We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- Add WeChat [edu\\_assist\\_pro](https://edu_assist_pro)
- $(q_s, q_1) : R_1 = \epsilon, R_2 = 0, R_3 = 1,$
  - $(q_s, q_a) : R_1 = \epsilon, R_2 = 0, R_3 = \epsilon, R_4 = \emptyset \Rightarrow \epsilon 0^* \epsilon \mid \emptyset$
  - $(q_1, q_1) :$
  - $(q_1, q_a) :$

### EXAMPLE 2: DIVISIBLE BY 3



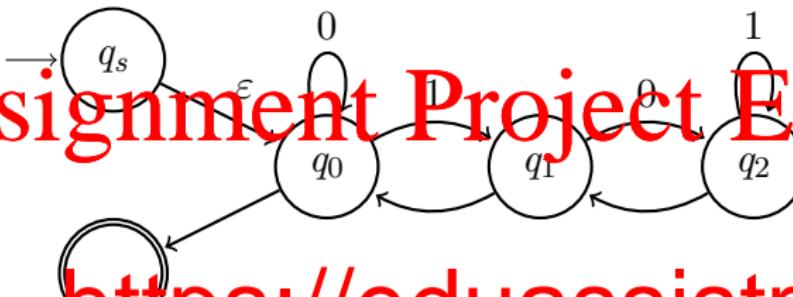
<https://eduassistpro.github.io>

We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not **Weighted**

- $(q_s, q_1) : R_1 = \varepsilon, R_2 = 0, R_3 = 1,$
  - $(q_s, q_a) : R_1 = \varepsilon, R_2 = 0, R_3 = \varepsilon, R_4 = \emptyset \Rightarrow \varepsilon 0^* \varepsilon \mid \emptyset = 0^*$
  - $(q_1, q_1) :$
  - $(q_1, q_a) :$

## EXAMPLE 2: DIVISIBLE BY 3



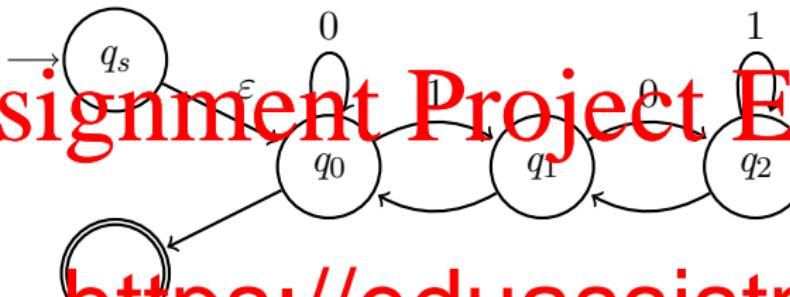
<https://eduassistpro.github.io>

We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- Add WeChat [edu\\_assist\\_pro](https://edu_assist_pro)
- $(q_s, q_1) : R_1 = \varepsilon, R_2 = 0, R_3 = 1,$
  - $(q_s, q_a) : R_1 = \varepsilon, R_2 = 0, R_3 = \varepsilon, R_4 = \emptyset \Rightarrow \varepsilon 0^* \varepsilon | \emptyset = 0^*$
  - $(q_1, q_1) : R_1 = 1, R_2 = 0, R_3 = 1, R_4 = \emptyset \Rightarrow 10^* 1 | \emptyset = 10^* 1$
  - $(q_1, q_a) :$

## EXAMPLE 2: DIVISIBLE BY 3



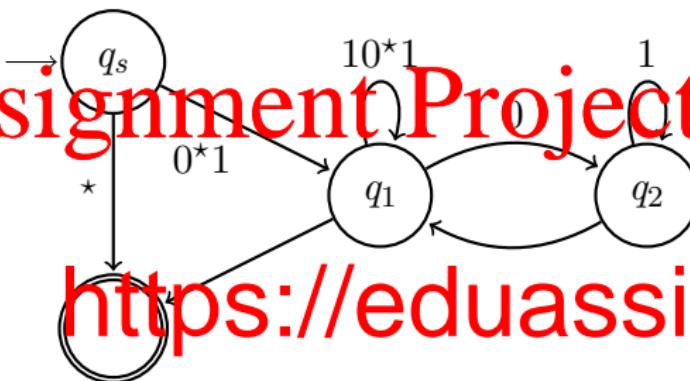
<https://eduassistpro.github.io>

We can eliminate the states in any order. Eliminate

All pairs of states with transitions which are not

- Add WeChat [edu\\_assist\\_pro](https://edu_assist_pro)
- $(q_s, q_1) : R_1 = \varepsilon, R_2 = 0, R_3 = 1,$
  - $(q_s, q_a) : R_1 = \varepsilon, R_2 = 0, R_3 = \varepsilon, R_4 = \emptyset \Rightarrow \varepsilon 0^* \varepsilon | \emptyset = 0^*$
  - $(q_1, q_1) : R_1 = 1, R_2 = 0, R_3 = 1, R_4 = \emptyset \Rightarrow 10^* 1 | \emptyset = 10^* 1$
  - $(q_1, q_a) : R_1 = 1, R_2 = 0, R_3 = \varepsilon, R_4 = \emptyset \Rightarrow 10^* \varepsilon | \emptyset = 10^*$

## EXAMPLE 2: DIVISIBLE BY 3



<https://eduassistpro.github.io>

We can eliminate the states in any order. Eliminate

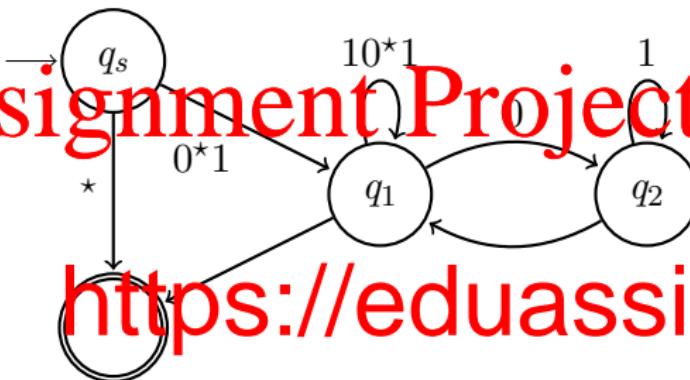
Add WeChat [edu\\_assist\\_pro](https://edu_assist_pro)

All pairs of states with transitions which are not

2



## EXAMPLE 2: DIVISIBLE BY 3



Assignment Project Exam Help  
<https://eduassistpro.github.io>

We can eliminate the states in any order. Eliminate

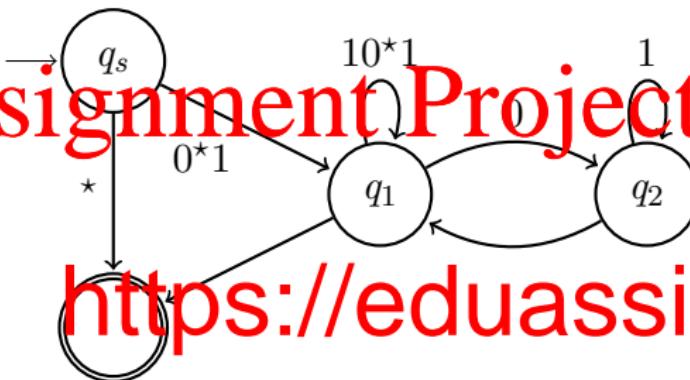
Add WeChat edu\_assist\_pro

All pairs of states with transitions which are not

2

- $(q_1, q_1)$ :

## EXAMPLE 2: DIVISIBLE BY 3



Assignment Project Exam Help  
<https://eduassistpro.github.io>

We can eliminate the states in any order. Eliminate

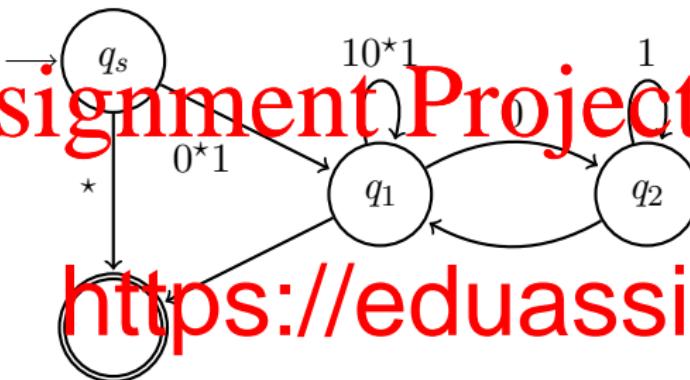
Add WeChat edu\_assist\_pro

All pairs of states with transitions which are not

2

- $(q_1, q_1) : R_1 = 0,$

## EXAMPLE 2: DIVISIBLE BY 3



Assignment Project Exam Help  
<https://eduassistpro.github.io>

We can eliminate the states in any order. Eliminate

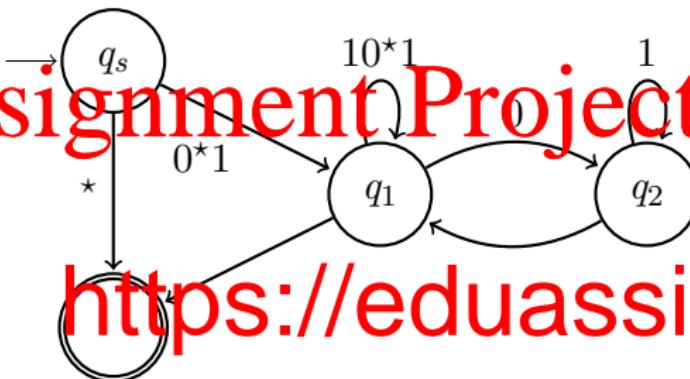
Add WeChat edu\_assist\_pro

All pairs of states with transitions which are not

2

- $(q_1, q_1) : R_1 = 0, R_2 = 1,$

## EXAMPLE 2: DIVISIBLE BY 3



Assignment Project Exam Help  
<https://eduassistpro.github.io>

We can eliminate the states in any order. Eliminate

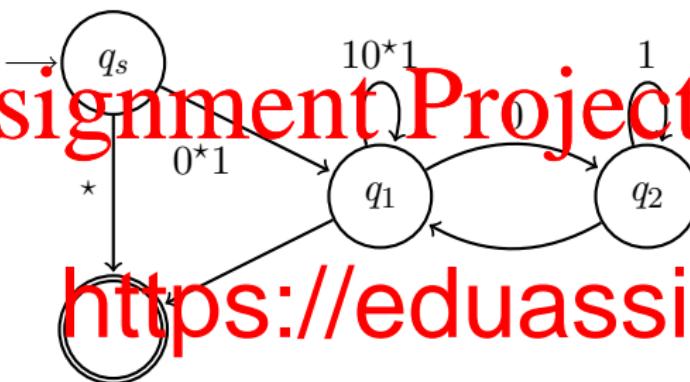
Add WeChat edu\_assist\_pro

All pairs of states with transitions which are not

2

- $(q_1, q_1) : R_1 = 0, R_2 = 1, R_3 = 0,$

## EXAMPLE 2: DIVISIBLE BY 3



Assignment Project Exam Help  
<https://eduassistpro.github.io>

We can eliminate the states in any order. Eliminate

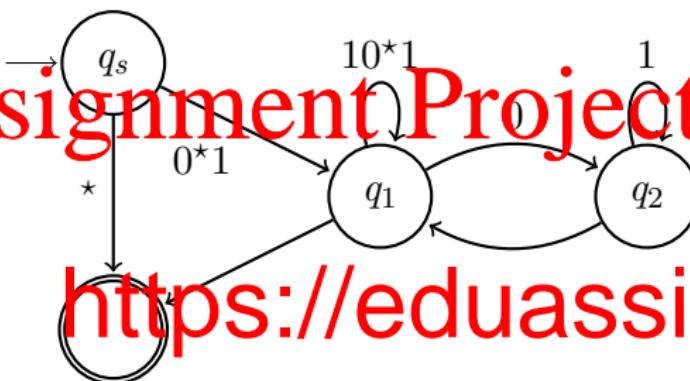
Add WeChat edu\_assist\_pro

All pairs of states with transitions which are not

2

- $(q_1, q_1) : R_1 = 0, R_2 = 1, R_3 = 0, R_4 = 10^*1$

## EXAMPLE 2: DIVISIBLE BY 3



Assignment Project Exam Help  
<https://eduassistpro.github.io>

We can eliminate the states in any order. Eliminate

Add WeChat [edu\\_assist\\_pro](https://edu_assist_pro)

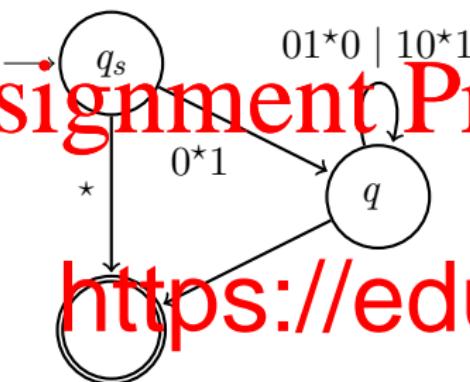
All pairs of states with transitions which are not

2

- $(q_1, q_1) : R_1 = 0, R_2 = 1, R_3 = 0, R_4 = 10^*1 \Rightarrow 01^*0 \mid 10^*1$

Much easier!

### EXAMPLE 2: DIVISIBLE BY 3



There is only one left to remove.

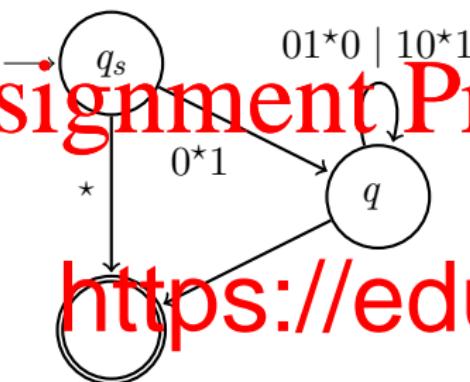
# Add WeChat edu\_assist\_pr

All pairs of states with transitions which are not Ø

$q_1$



## EXAMPLE 2: DIVISIBLE BY 3



Assignment Project Exam Help  
<https://eduassistpro.github.io>

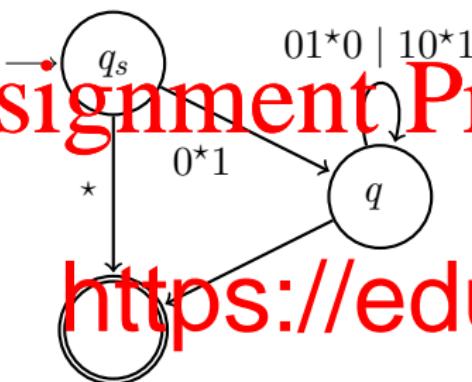
There is only  $q_1$  left to remove.  
Add WeChat edu\_assist\_pro

All pairs of states with transitions which are not  $\emptyset$

$q_1$

- $(q_s, q_a) :$

## EXAMPLE 2: DIVISIBLE BY 3



Assignment Project Exam Help  
<https://eduassistpro.github.io>

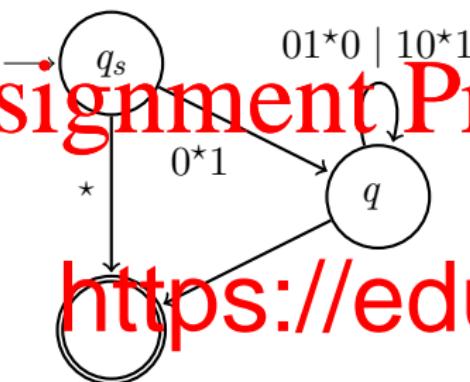
There is only  $q_1$  left to remove.  
Add WeChat edu\_assist\_pro

All pairs of states with transitions which are not  $\emptyset$

- $(q_s, q_a) : R_1 = 0^*1,$

$q_1$

### EXAMPLE 2: DIVISIBLE BY 3



There is only one left to remove.

There is only 1 left to remove.

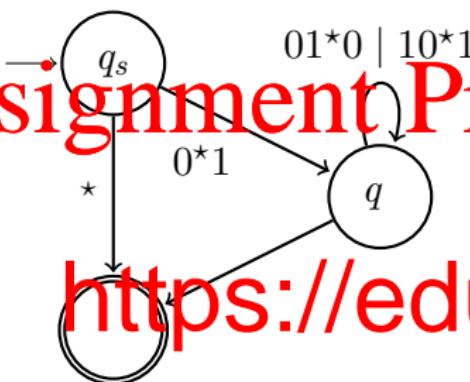
# Add WeChat edu\_assist\_pr

All pairs of states with transitions which are not Ø

$q_1$

- $(q_s, q_a) : R_1 = 0^*1, R_2 = 01^*0 \mid 10^*1,$

## EXAMPLE 2: DIVISIBLE BY 3



# Assignment Project Exam Help

<https://eduassistpro.github.io>

There is only  $q_1$  left to remove.

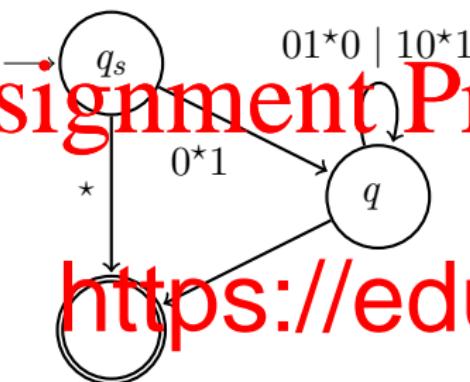
## Add WeChat edu\_assist\_pro

All pairs of states with transitions which are not  $\emptyset$

$q_1$

- $(q_s, q_a) : R_1 = 0^*1, R_2 = 01^*0 \mid 10^*1, R_3 = 10^*$ ,

### EXAMPLE 2: DIVISIBLE BY 3



There is only one left to remove.

There is only 1 left to remove.

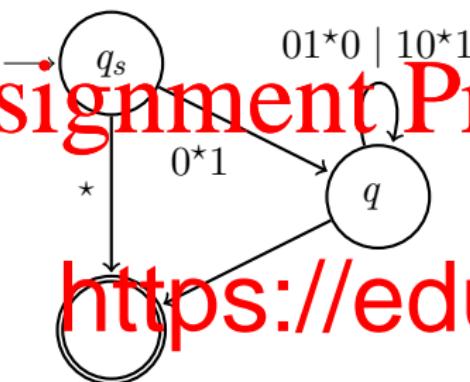
# Add WeChat edu\_assist\_pr

All pairs of states with transitions which are not Ø

$q_1$

- $(q_s, q_a) : R_1 = 0^*1, R_2 = 01^*0 \mid 10^*1, R_3 = 10^*, R_4 = 0^*$

### EXAMPLE 2: DIVISIBLE BY 3



There is only one left to remove.

# Add WeChat edu\_assist\_pr

All pairs of states with transitions which are not Ø

- $(q_s, q_a) : R_1 = 0^*1, R_2 = 01^*0 \mid 10^*1, R_3 = 10^*, R_4 = 0^*$   
 so  $(R_1)(R_2)^*(R_3) \mid (R_4) = 0^*1(01^*0 \mid 10^*1)^*10^* \mid 0^*$

## EXAMPLE 2: DIVISIBLE BY 3

Assignment Project Exam Help



So,  
strin

<https://eduassistpro.github.io>

Add WeChat edu\_assist\_pro

## EXAMPLE 2: DIVISIBLE BY 3

Assignment Project Exam Help

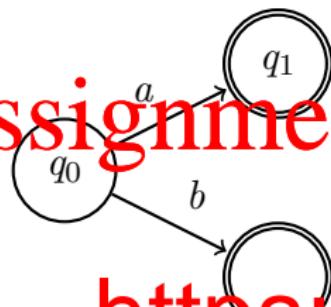


So, <https://eduassistpro.github.io>

Add WeChat edu\_assist\_pro  
Eliminating states in a different order can result in a different equivalent RegEx.

e.g. eliminating  $q_2$ ,  $q_1$ , then  $q_0$  yields:  $(1(01^*0)^*1|0)^*$

## EXAMPLE 3: MULTIPLE ACCEPT STATES

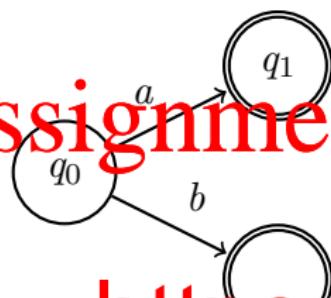


Assignment Project Exam Help  
<https://eduassistpro.github.io>

As GNFA:

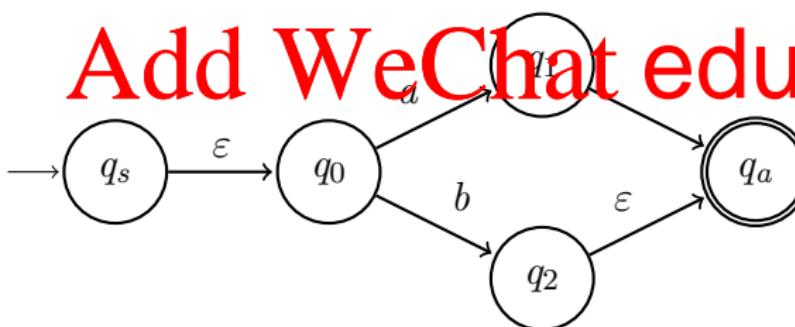
Add WeChat edu\_assist\_pro

## EXAMPLE 3: MULTIPLE ACCEPT STATES

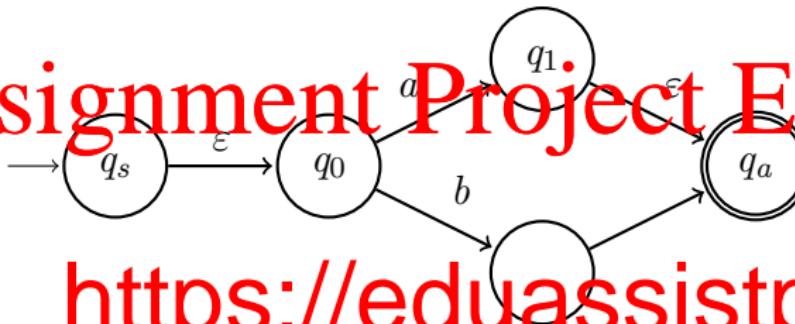


Assignment Project Exam Help  
<https://eduassistpro.github.io>

As GNFA:



## EXAMPLE 3: MULTIPLE ACCEPT STATES

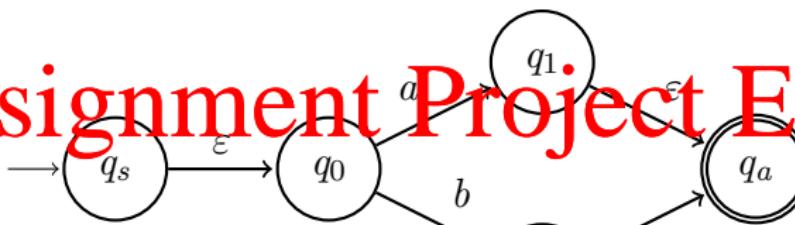


Assignment Project Exam Help  
<https://eduassistpro.github.io>

Eliminate  $q_2$ :

Add WeChat edu\_assist\_pro

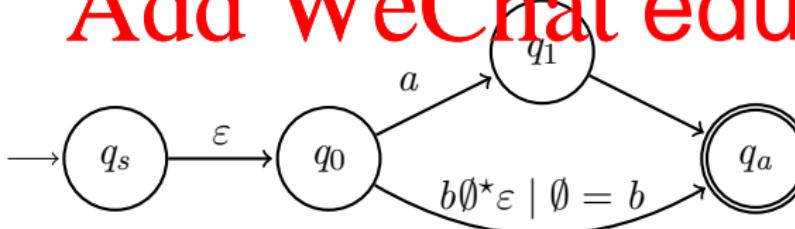
## EXAMPLE 3: MULTIPLE ACCEPT STATES



Assignment Project Exam Help  
<https://eduassistpro.github.io>

Eliminate  $q_2$ :

Add WeChat edu\_assist\_pro



## EXAMPLE 3: MULTIPLE ACCEPT STATES

Assignment Project Exam Help  
<https://eduassistpro.github.io>

Eliminate  $q_1$ :

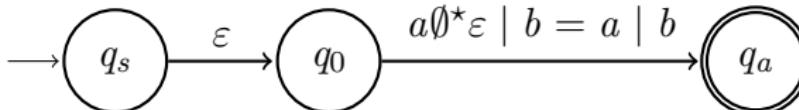
Add WeChat edu\_assist\_pro

## EXAMPLE 3: MULTIPLE ACCEPT STATES

Assignment Project Exam Help  
<https://eduassistpro.github.io>

Eliminate  $q_1$ :

Add WeChat edu\_assist\_pr



## EXAMPLE 3: MULTIPLE ACCEPT STATES



Elim

<https://eduassistpro.github.io>

Add WeChat edu\_assist\_pro

Simplify, to get the expected:

## EXAMPLE 3: MULTIPLE ACCEPT STATES



Elim

<https://eduassistpro.github.io>



Add WeChat edu\_assist\_pro

Simplify, to get the expected:

## EXAMPLE 3: MULTIPLE ACCEPT STATES

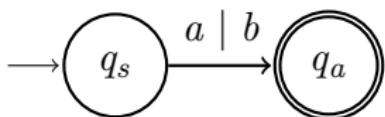
Assignment Project Exam Help

Elim



Add WeChat edu\_assist\_pro

Simplify, to get the expected:



## SUMMARY

Algorithm to convert an NFA to a RegEx:

Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu\_assist\_pro

## SUMMARY

Algorithm to convert an NFA to a RegEx:

# Assignment Project Exam Help

1. Convert the NFA to a Generalised NFA (GNFA)  
► Only one start state, no incoming transitions

## 2. <https://eduassistpro.github.io/>

- $(q_i, q_j)$        $q_i, q_j$        $Q$        $q_{elim}$
- Replace arrow  $q_i \rightarrow q_j$  with  $((R_1 \cup R_2) \cdot R_3)^*$
  - $R_1$  is the RegEx on transition  $i \xrightarrow{a} q_i$
  - $R_2$  is the RegEx on transition  $i \xrightarrow{a} q_j$
  - $R_3$  is the RegEx on transition  $q_i \xrightarrow{a} q_j$
  - $R_4$  is the RegEx on transition  $q_i \rightarrow q_j$

Add WeChat `edu_assist_pro`

## SUMMARY

Algorithm to convert an NFA to a RegEx:

# Assignment Project Exam Help

1. Convert the NFA to a Generalised NFA (GNFA)  
► Only one start state, no incoming transitions

## 2. <https://eduassistpro.github.io>

- $(q_i, q_j)$        $q_i, q_j$        $Q$        $q_{elim}$
- Replace arrow  $q_i \rightarrow q_j$  with  $((R_1 \cdot R_2) \cdot R_3)^{elim} \cdot R_4$
  - $R_1$  is the RegEx on transition  $i \rightarrow i$
  - $R_2$  is the RegEx on transition  $i \rightarrow j$
  - $R_3$  is the RegEx on transition  $j \rightarrow j$
  - $R_4$  is the RegEx on transition  $q_i \rightarrow q_j$

3. The transition between the start and the accept state is now a regular expression describing  $L$

Add WeChat `edu_assist_pro`

## OUTLINE

# Assignment Project Exam Help

- ▶ Regular Expressions

▶ <https://eduassistpro.github.io>

▶ Add WeChat edu\_assist\_pro

- ▶ Proving if a language is, or is not

## PROVING IF A LANGUAGE IS REGULAR or *not*

Recall the definition:

A language is *regular* if and only if there exists a finite automaton which recognises it.

Supp

<https://eduassistpro.github.io>

Add WeChat edu\_assist\_pro

# PROVING IF A LANGUAGE IS REGULAR or *not*

Recall the definition:

A language is *regular* if and only if there exists a finite automaton which recognises it.

Supp

- ▶ <https://eduassistpro.github.io>
  - ▶ Devise a DFA which recognises it, or
  - ▶ Devise a RegEx which describes it
- Add WeChat edu\_assist\_pro

## PROVING IF A LANGUAGE IS REGULAR or not

Recall the definition:

A language is *regular* if and only if there exists a finite automaton which recognises it.

Supp

► <https://eduassistpro.github.io>

- Devise a DFA which recognises it, or
- Devise a RegEx which describes it

Add WeChat edu\_assist\_pro

What if the language was not regular? How can we *prove* that no suitable automata exists? We can use the *pumping lemma for regular languages*

PUMPING  $s$  USING  $y$ 

# Assignment Project Exam Help



Suppose some string  $s$  exists, passing through

- from  $q_0$  to  $q_i$ , using a string  $x$ ; then
- from  $q_i$  back to  $q_i$ , using a string  $/$
- from  $q_i$  to some accept state  $f \in F$ , using a string  $z$

Add WeChat `edu_assist_pro`

PUMPING  $s$  USING  $y$ 

# Assignment Project Exam Help



Suppose some string  $s$  exists, passing through

- from  $q_0$  to  $q_i$ , using a string  $x$ ; then
- from  $q_i$  back to  $q_i$ , using a string  $/$  /
- from  $q_i$  to some accept state  $f \in F$ , using a string  $z$

Then  $s = xyz \in L$ , and  $xy^kz \in L$  for all  $k \geq 0$

e.g. if  $x = aa$ ,  $y = b$ ,  $z = c$ , then  $\{aac, aabc, aabbc, \dots\} \subseteq L$

## LONG STRINGS IN FINITE AUTOMATA

# Assignment Project Exam Help



Suppose  $M$  is a DFA with  $n$  states, and

Add WeChat  $\text{edu\_assist\_pr}$ <sup>+1</sup>

## LONG STRINGS IN FINITE AUTOMATA

## Assignment Project Exam Help



Suppose  $M$  is a DFA with  $n$  states, and

Add WeChat  $edu\_assist\_pr$ <sup>+1</sup>

Then there exists at least one state which was visited

once ( $q_i$ ), and a substring  $y \neq \varepsilon$  corresponding to the path followed between the two of those visits.

## LONG STRINGS IN FINITE AUTOMATA

## Assignment Project Exam Help



Suppose  $M$  is a DFA with  $n$  states, and

Add WeChat  $edu\_assist\_pr$ <sup>+1</sup>

Then there exists at least one state which was visited

once ( $q_i$ ), and a substring  $y \neq \varepsilon$  corresponding to the path followed between the two of those visits.

Hence we can *pump*  $s$  to find other strings in the language

## FINITE AUTOMATA, INFINITE LANGUAGES

# Assignment Project Exam Help

All finite languages are regular. (Why?)

<https://eduassistpro.github.io>

Add WeChat edu\_assist\_pro

## FINITE AUTOMATA, INFINITE LANGUAGES

# Assignment Project Exam Help

All finite languages are regular. (Why?)

Supp <https://eduassistpro.github.io>

- ▶  $L$   $M$
- ▶  $L$  is an infinite set over a finite alphabet, so it must contain arbitrarily long words.
- ▶ Therefore words exist which can be

Add WeChat edu\_assist\_pro

## PUMPING LEMMA FOR REGULAR LANGUAGES

# Assignment Project Exam Help

If  $L$  is a regular language, then there exists a number  $p$  (the *pumping length*) such that for any string  $s \in L$  of length at least  $p$ , the

t:

1. <https://eduassistpro.github.io/>
2. /
3.  $|xy| \leq p$

# Add WeChat edu\_assist\_pro

*If a language does not satisfy this lemma, then it cannot be regular*

## USING THE PUMPING LEMMA

We can (only) use the pumping lemma to prove that a language is not regular

# Assignment Project Exam Help

Proof

1. <https://eduassistpro.github.io/>
2.
  - This is often the hardest part of the proof
    - We must find one which will allow us to:
3. Apply the lemma to find a contradiction
4. Thereby deduce that the assumption is false
  - i.e. The language cannot be regular

Add WeChat edu\_assist\_pro

## EXAMPLE

Show that  $L = \{a^n b^n \mid n \geq 0\}$  is not regular

# Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu\_assist\_pro

## EXAMPLE

Show that  $L = \{a^n b^n \mid n \geq 0\}$  is not regular

1. Assumption:  $L$  is regular.

# Assignment Project Exam Help

<https://eduassistpro.github.io>

Add WeChat edu\_assist\_pro

## EXAMPLE

Show that  $L = \{a^n b^n \mid n \geq 0\}$  is not regular.

- # Assignment Project Exam Help
1. Assumption:  $L$  is regular.
  2. Then there exists some  $p \geq 0$  satisfying the pumping lemma.

<https://eduassistpro.github.io>

Add WeChat edu\_assist\_pro

## EXAMPLE

Show that  $L = \{a^n b^n \mid n \geq 0\}$  is not regular.

1. Assumption:  $L$  is regular.
2. Then there exists some  $p \geq 0$  satisfying the pumping lemma.
- 3.

<https://eduassistpro.github.io>

Add WeChat edu\_assist\_pro

## EXAMPLE

Show that  $L = \{a^n b^n \mid n \geq 0\}$  is not regular.

1. Assumption:  $L$  is regular.
2. Then there exists some  $p \geq 0$  satisfying the pumping lemma.

3.

4. <https://eduassistpro.github.io>

4.2  $|y| > 0$

4.3  $|xy| \leq p$

Add WeChat edu\_assist\_pro

## EXAMPLE

Show that  $L = \{a^n b^n \mid n \geq 0\}$  is not regular.

1. Assumption:  $L$  is regular.
2. Then there exists some  $p \geq 0$  satisfying the pumping lemma.

3.

4. <https://eduassistpro.github.io>

4.2  $|y| > 0$

4.3  $|xy| \leq p$

5. Therefore  $y = a^i$  for some  $0 < i$

## EXAMPLE

Show that  $L = \{a^n b^n \mid n \geq 0\}$  is not regular.

1. Assumption:  $L$  is regular.
2. Then there exists some  $p \geq 0$  satisfying the pumping lemma.

3.

4. <https://eduassistpro.github.io>

4.2  $|y| > 0$

4.3  $|xy| \leq p$

5. Therefore  $y = a^i$  for some  $0 < i \leq p$ .
6. Let  $k = 2$ . Then  $xy^kz$  has more  $a$  than  $b$ , so  $xy^kz \notin L$ .

Add WeChat edu\_assist\_pro

## EXAMPLE

Show that  $L = \{a^n b^n \mid n \geq 0\}$  is not regular.

1. Assumption:  $L$  is regular.
2. Then there exists some  $p \geq 0$  satisfying the pumping lemma.

3.

4. <https://eduassistpro.github.io>

4.2  $|y| > 0$

4.3  $|xy| \leq p$

5. Therefore  $y = a^i$  for some  $0 < i \leq p$ .
6. Let  $k = 2$ . Then  $xy^kz$  has more  $a$  than  $b$ , so  $xy^kz \notin L$ .
7. Lines 6 and 4.1 form a contradiction, therefore the assumption is false (i.e.  $L$  is *not* regular.)

Add WeChat edu\_assist\_pro

## INTRODUCTION

So far we have seen two different, but equivalent, methods of describing languages: finite automata and regular expressions, which describe *regular languages*

We have seen the language  $\{0^n 1\}$

<https://eduassistpro.github.io/>

Today we will introduce context-free grammars, which will help us to describe the next category of languages, the

Add WeChat [edu\\_assist\\_pro.com](https://edu_assist_pro.com)

Later, we will see grammars called *regular grammars*, which describe exactly *regular languages*

## GRAMMARS

# Assignment Project Exam Help

Grammars are another way to describe a language

A grammar defines a language

Add WeChat edu\_assist\_pro

The language generated is the set of all strings which can be derived from the grammar

INTRODUCTORY EXAMPLE ( $G_1$ )

Assignment Project Exam Help

 $G \rightarrow 01$ Base case:  $01 \in L$  $L$  $G_1$  g  
kno<https://eduassistpro.github.io>

How does it derive 000111?

Add WeChat edu\_assist\_pro

INTRODUCTORY EXAMPLE ( $G_1$ )

Assignment Project Exam Help

 $G \rightarrow 01$ Base case:  $01 \in L$  $L$  $G_1$  g  
kno<https://eduassistpro.github.io>

How does it derive 000111?

Add WeChat  $edu\_assist\_pr$

 $S$

INTRODUCTORY EXAMPLE ( $G_1$ )

Assignment Project Exam Help  
 $G \rightarrow 01$  Base case:  $01 \in L$   
 $L$

$G_1$  g kno <https://eduassistpro.github.io>

How does it derive 000111?

Add WeChat edu\_assist\_pro  
 $S \Rightarrow 0S1$  usi

INTRODUCTORY EXAMPLE ( $G_1$ )

Assignment Project Exam Help  
 $G \rightarrow 01$  Base case:  $01 \in L$   
 $L$

$G_1$  g kno <https://eduassistpro.github.io>

How does it derive 000111?

Add WeChat edu\_assist\_pro  
 $S \Rightarrow 0S1$  usi  
 $\Rightarrow 00S11$  using rule  $S \rightarrow 0S1$

INTRODUCTORY EXAMPLE ( $G_1$ )

Assignment Project Exam Help  
 $G \rightarrow 01$  Base case:  $01 \in L$   
 $L$

$G_1$  g kno <https://eduassistpro.github.io>

How does it derive 000111?

Add WeChat edu\_assist\_pr

$S \Rightarrow 0S1$

usi

$\Rightarrow 00S11$

using rule  $S \rightarrow 0S1$

$\Rightarrow 000111$

using rule  $S \rightarrow 01$

INTRODUCTORY EXAMPLE ( $G_2$ )

# Assignment Project Exam Help

$$S \rightarrow NounPhrase\ VerbPhrase$$

<https://eduassistpro.github.io>

*Verb*  $\rightarrow$  likes | see

What language does  $G_2$  generate?

Add WeChat edu\_assist\_pro

INTRODUCTORY EXAMPLE ( $G_2$ )

# Assignment Project Exam Help

$$S \rightarrow NounPhrase\ VerbPhrase$$

<https://eduassistpro.github.io>

*Verb*  $\rightarrow$  likes | see

What language does  $G_2$  generate?

{ the girl likes the girl, the girl likes the ball,  
the girl sees the girl, the girl sees the ball,  
the ball likes the girl, the ball likes the ball,  
the ball sees the girl, the ball sees the ball }

Add WeChat edu\_assist\_pro

## DEFINITIONS

*Terminals*

- The finite set of symbols which make up strings of the language

*Non-Terminal symbols*

- A finite set of symbols used to generate the strings of the language
- They never appear in the language.

Add WeChat `edu_assist_pro`

*Start symbol*

- The variable used to start every derivation

# DEFINITIONS

*Production rules*

**Assignment Project Exam Help**

- ▶ Sometimes called substitution or derivation rules
- ▶ Define strings of *variables* and *terminals* which can be

<https://eduassistpro.github.io>

Add WeChat edu\_assist\_pro

## DEFINITIONS

### *Production rules*

Assignment Project Exam Help

- ▶ Sometimes called substitution or derivation rules
- ▶ Define strings of *variables* and *terminals* which can be

<https://eduassistpro.github.io>

A variable can have many rules:

Add WeChat edu\_assist\_pr  
*Noun*  $\rightarrow$  girl

*Noun*  $\rightarrow$  ball

*Noun*  $\rightarrow$  quokka

## DEFINITIONS

## *Production rules*

- Sometimes called substitution or derivation rules
  - Define strings of *variables* and *terminals* which can be

<https://eduassistpro.github.io>

A variable can have many rules:      The

**Add WeChat *edu\_assist\_pr***

*Noun* → ball

*Noun* → quokka

## ANOTHER EXAMPLE

# Assignment Project Exam Help

$$S \rightarrow T \mid (S \cdot S) \mid (\lambda T.S)$$

<https://eduassistpro.github.io/>

This is a grammar for lambda calculus expression

- Add WeChat edu\_assist\_pro
- ▶ The variables are  $S, T$
  - ▶  $S$  is the start variable
  - ▶ The terminals are  $a, b, \dots, (, ), \lambda, ., \cdot$  (i.e. atoms and operators)

## SOME COMMON NOTATIONAL CONVENTIONS

# Assignment Project Exam Help

*If not stated otherwise:*

- ▶
- ▶ <https://eduassistpro.github.io>
- ▶ ...,  $X, Y, Z$  are either terminals or variables
- ▶ ... $w, x, y, z$  are strings of terminals
- ▶  $\alpha, \beta, \gamma, \dots$  are strings of terminals and/or variables

Add WeChat **edu\_assist\_pro**

NIL

# Assignment Project Exam Help

In your tree encoding, NIL should represent the empty tree (a tree with 0 nodes)

This is <https://eduassistpro.github.io>

If you need to use both and they are different, then you can rename them, e.g.

Add WeChat edu\_assist\_pro

- $\text{NIL} = \text{PAIR } \text{TRUE } \text{TRUE}$ , for Ch
- $\text{NILTREE}$ , for your tree encoding

## EXAMPLE ANSWER

(*HEAD* *z*) should be the head of list *z*.

*HEAD* =  $\lambda z.FIRST\ (SECOND\ z)$

The second pair in a Church list contains the data `[heat, tail]`, so we just need to get the first expression from the second pair.

Exa

<https://eduassistpro.github.io/>

$= \text{HEAD} (\text{PAIR} \text{ } \text{TRUE} \text{ } (\text{PAIR} \text{ } (1$

$\equiv (\lambda z.FIRST\ (SECOND\ z))PAT$

$= FIRST (SECOND (PAIR TRUE (PAIR 1 \{2,3\}))))$

$= FIRST (PAIR 1 \{2, 3\}))$

(reduced SECOND)

= 1

(reduced FIRST)

IMMUTABLE DATA STRUCTURES

# Assignment Project Exam Help

Data s

- ▶ <https://eduassistpro.github.io>
  - ▶ Add WeChat edu\_assist\_pro

EXAMPLE: DELETE THE SECOND ELEMENT OF A LIST

# Assignment Project Exam Help

DE

AIL z))

<https://eduassistpro.github.io>

- ▶ CONS to make a new list using:
- ▶ the head of the old list
- ▶ the tail of the tail of the old list:
  - ▶ i.e. we skipped the second element

Add WeChat edu\_assist\_pro

EXAMPLE: INSERT  $e$  TO THE SECOND POSITION OF A LIST

# Assignment Project Exam Help

*INS* <https://eduassistpro.github.io><sup>TAIL<sub>(z)</sub>)</sup>

- ▶ CONS a new list using:
- ▶ the head of the old list
  - ▶  $e$
  - ▶ the tail of the old list

Add WeChat edu\_assist\_pro

## RECURSIVE DATA STRUCTURES

# Assignment Project Exam Help

- ▶ Each position has a reference (link) to the next one

<https://eduassistpro.github.io/>

Chur

they link to the list which *starts with t*

- ▶ CONS head tail
  - ▶ head is the element stored at the start of the list
  - ▶ tail is the sublist containing all the remaining elements

Add WeChat edu\_assist\_pro

## RECURSIVE DATA STRUCTURES

# Assignment Project Exam Help

You can implement a tree data structure in a very similar way.

Excel  
and a n

<https://eduassistpro.github.io>

In your assignment `MAKETREE` needs to define a tree data structure which will take a number to store, and two trees (which will be left and right children).

## JAVA EXAMPLE

```
public class Tree {  
    public final int element;  
    public final Tree left;
```

p

<https://eduassistpro.github.io>

~~this.left = left;~~

# Add Weights

}

}

MAKETREE in the assignment is like writing: new Tree(e,a,b);

## RECURSIVE FUNCTIONS

Simple recursion, condition decides if you're at the base case

# Assignment Project Exam Help

You c

$H = \lambda fxyz. (\text{condition})(\text{base})(\text{recursive})$   
 $F = (Y H)$

$F = (Y H)$

## Add WeChat edu\_assist\_pro

- If  $x < 5$  and  $y < 5$  then  $y$  (base case)
- If  $x < 5$  and  $y \geq 5$  then  $f(1, x)$  (recursive case)
- If  $x \geq 5$  and  $x = y$  then  $f(y, x)$  (recursive case)
- If  $x \geq 5$  and  $x \neq y$  then 3 (base case)

## HELPER FUNCTIONS

# Assignment Project Exam Help

It's a good idea to define some extra expressions sometimes

- ▶
- ▶

<https://eduassistpro.github.io>

# Add WeChat edu\_assist\_pro

Example: a MAX function which returns the largest value in a list would be helpful when calculating the height of a tree

# LISP

I discourage you from trying to directly implement your lambda calculus expressions in LISP. It'll mostly work, but some things get messy (especially conditional logic) if you try to implement ever

I recd <https://eduassistpro.github.io>

- ▶ elements
- ▶ A Church list can be represented by a Lisp list

Add WeChat edu\_assist\_pro

Then you can just use standard Lisp list operations to manipulate your data, instead of needing to implement all your lambda calculus expressions directly.

LISP

Don't forget to indent your code to make it easier to see what is being applied to what. Most errors people have shown me were just scope errors.

```
;; example of s
(def https://eduass
  (if
    (...);; some condition
    (...);; true case
    (...);; false case
  )
)

```

# Add WeChat edu\_assist\_pr

LISP

# Assignment Project Exam Help

<https://eduassistpro.github.io/>

( . . . ) ; ; true 2

(...); *false*?  
dd Wech

( . . . ) ; ; false 1

)

)

Add WeChat edu\_assist\_pr

## LISP

# Assignment Project Exam Help

You don't need any notation we haven't provided in tutorials or  
lectures

<https://eduassistpro.github.io>

Look back at the earlier tutorials and solutions for some examples:

Add WeChat edu\_assist\_pro

## Regular Expressions

- ▶ What they are and how to use them
- ▶ Regular operations

# Assignment Project Exam Help

## Equivalence of FA and Regular Expressions

- ▶
- ▶ <https://eduassistpro.github.io>

Proving if a language is, *or is not*, regular

- ▶ Prove regularity by finding a DFA, NFA, or REGEX
- ▶ Prove non-regularity by finding a contradiction

Pumping Lemma

Grammars (basic concepts)

Lambda calculus revision

Add WeChat `edu_assist_pro`