

程序代写代做 CS编程辅导

FIT2014 Theory of Computation



Lecture 8

Kleene's Theorem. I.

WeChat: cstutorcs

Regexp \rightarrow NFA \rightarrow FA

Assignment Project Exam Help

Email: tutorcs@163.com

slides by Graham Farr

QQ: 749389476

<https://tutorcs.com>

COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

Warning

This material has been reproduced and communicated to you by or on behalf of Monash University in accordance with s113P of the Copyright Act 1968 (the Act).

The material in this communication may be subject to copyright under the Act.

Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice.

Overview

程序代写代做 CS编程辅导



- ▶ Questions
- ▶ Kleene's Theorem
- ▶ Convert Regular Expressions to NFA
- ▶ Convert NFA to FA
- ▶ *Next lecture:*
Convert FA to Regular Expression

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Stephen Cole Kleene (1909–1994)
<https://mathshistory.st-andrews.ac.uk/Biographies/Kleene/>

Questions

- ▶ Can every language which is represented by a **regular expression** be described by a **finite automaton**? 程序代写代做 CS编程辅导
- ▶ Can every language which is described by a **finite automaton** be represented by a **regular expression**? 
- ▶ Can every language be represented by a **regular expression** or a **finite automaton**?

{ all languages }

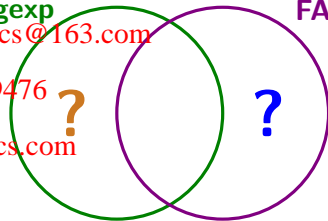
WeChat: cstutores

representable by **regexp** recognised by **FA**

Email: tutores@163.com

QQ: 749389476

<https://tutores.com>



Kleene's Theorem

程序代写代做 CS编程辅导



Theorem.

Any language which can be defined by a regular expression

- ▶ Regular Expressions
- ▶ Finite Automata
- ▶ Nondeterministic Finite Automata (NFA)
- ▶ Generalized Nondeterministic Finite Automata (GNFA)

can be defined by any of the other methods.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Kleene's Theorem



Converting Regular Expression to NFA

程序代写代做 CS编程辅导

Start with:



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

Apply the following rules until all edges are labelled with a letter or ε :

QQ: 749389476

<https://tutorcs.com>

Converting Regular Expression to NFA



Converting Regular Expression to NFA



Converting Regular Expression to NFA



Converting Regular Expression to NFA



Here, you can't match PR^*Q or SR^*T , in general.

The two ϵ transitions are necessary, in general.

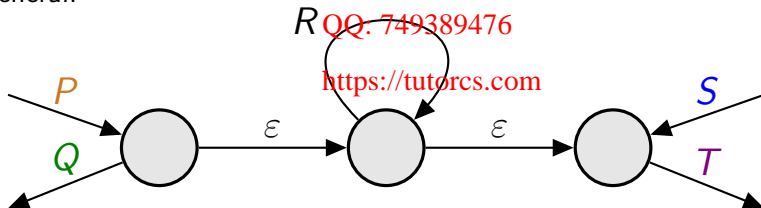
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

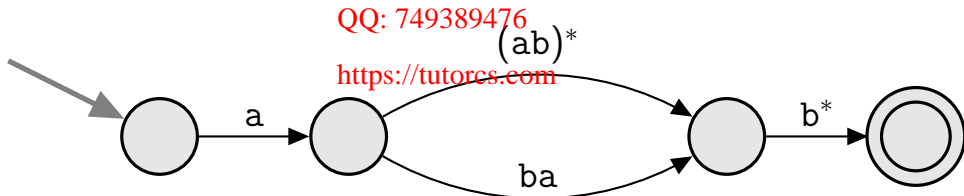
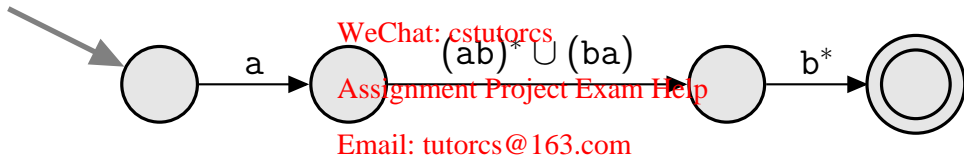
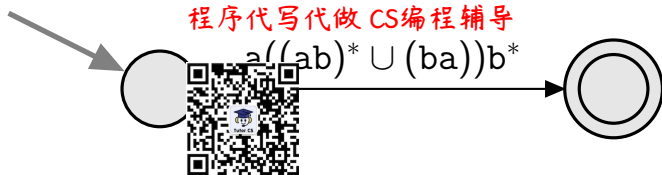
QQ: 749389476

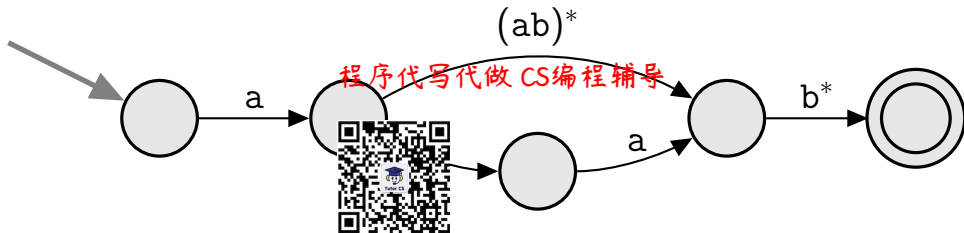
<https://tutorcs.com>



So the R loop cannot be at left node or right node.

Converting Regular Expression to NFA. Example: $a((ab)^* \cup (ba))b^*$





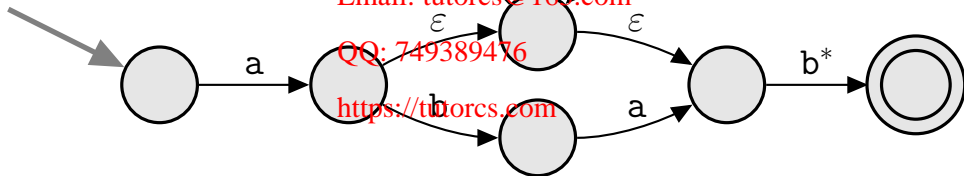
WeChat: cstutorcs

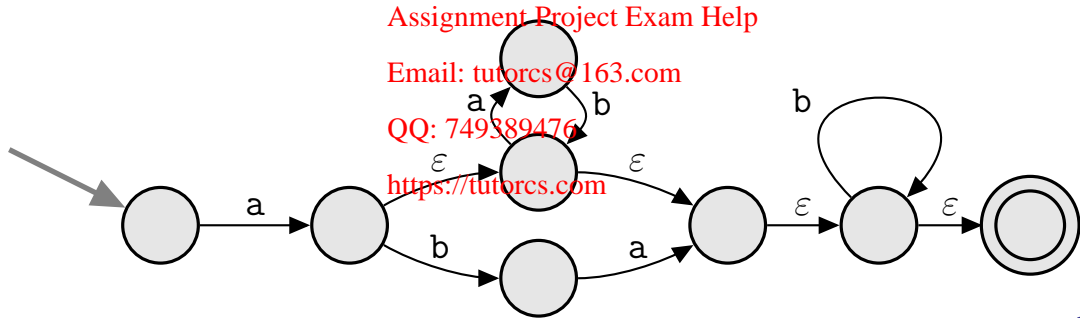
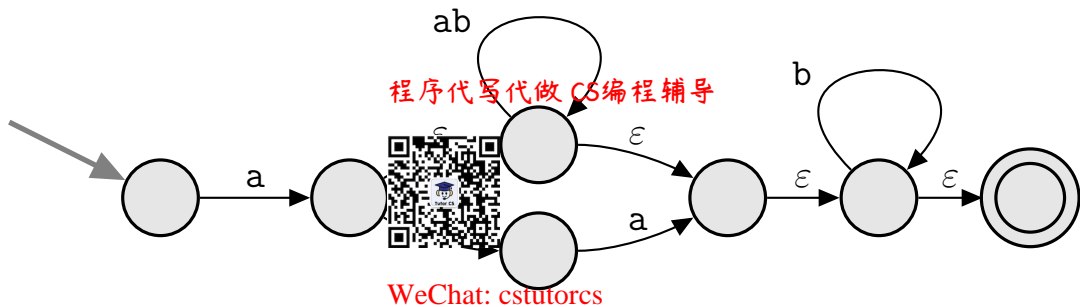
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>





Converting a NFA to a FA

程序代写代做 CS编程辅导



Complexity?

WeChat: cstutorcs

How reversible is this construction?

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Kleene's Theorem



Converting a NFA to a FA

In a FA:

- ▶ Any string w traces a unique path starting from the Start State and ending at some unique state, which we call $\text{endState}(w)$.
- ▶ The string w is accepted if $\text{endState}(w)$ is a Final State, otherwise it is rejected.
- ▶ $\text{endState}(\varepsilon) = \text{Start State}$.

程序代写代做 CS编程辅导



WeChat: cstutorcs

In a NFA:

- ▶ Any string w traces a set of paths, starting from the Start State and ending at some set of states, which we'll call $\text{endStates}(w)$.
- ▶ The set might have zero, one or more members.
- ▶ The string w is accepted if $\text{endStates}(w)$ contains a Final State, otherwise it is rejected.
- ▶ $\text{endStates}(\varepsilon) = \{\text{Start State}\}$ if there are no ε transitions.

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Converting a NFA to a FA



WeChat: cstutorcs

endStates(ab) = {1, 2}

endStates(aba) = {1, 3}

Email: tutorcs@163.com

In general, if w is a string and x is a single letter, then

$\text{endStates}(wx) = \{q : \text{for some state } p \in \text{endStates}(w), \text{ there is a transition } p \xrightarrow{x} q\}$
... provided there are no *empty* transitions.

This suggests part of a method for constructing $\text{endStates}(w)$ for all strings w .

Converting a NFA to a FA

Idea:

sets of states in the NFA \longrightarrow states in the FA.

Informally (and *assuming no epsilon transitions*, for the time being):



Start with the one-element set $\{\text{start state}\}$.

- ▶ This is $\text{endStates}(\epsilon)$.
- ▶ It's the set of NFA-states we can possibly be in at the very start.

Assignment Project Exam Help

Construct $\text{endStates}(a)$, the set of all states we could then get to by reading a single a .

Construct $\text{endStates}(b)$, the set of all states we could then get to by reading a single b .

QQ: 749389476

For each set of states, X , that we construct:

- ▶ find the set of states we can get to from X , by reading a single a .
- ▶ find the set of states we can get to from X , by reading a single b .

Keep doing this, until we no longer get any new sets of states.

Converting a NFA to a FA



WeChat: cstutorcs
Assignment Project Exam Help
Email: tutores@163.com
QQ: 749389476
<https://tutores.com>

state	a	b
Start {1}		

→

state	a	b
Start {1}	{1}	{1,2}
{1,2}	{1,3}	{1,2,3}
{1,3}	{1}	{1,2}
{1,2,3}		

→

state	a	b
Start {1}	{1}	{1,2}
{1,2}	{1,3}	{1,2,3}
{1,3}	{1}	{1,2}
{1,2,3}		

→

state	a	b
Start {1}	{1}	{1,2}
{1,2}	{1,3}	{1,2,3}
{1,3}	{1}	{1,2}
{1,2,3}		

.....

Converting a NFA to a FA



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

... →

	state	a	b		state	a	b
...	Start {1}	{1}	{1,2}	→	Start {1}	{1}	{1,2}
	{1,2}	{1,3}	{1,2,3}		{1,2}	{1,3}	{1,2,3}
	{1,3}	{1}	{1,2}		Final {1,3}	{1}	{1,2}
	{1,2,3}	{1,3}	{1,2,3}		Final {1,2,3}	{1,3}	{1,2,3}

Converting a NFA to a FA



WeChat: cstutorcs

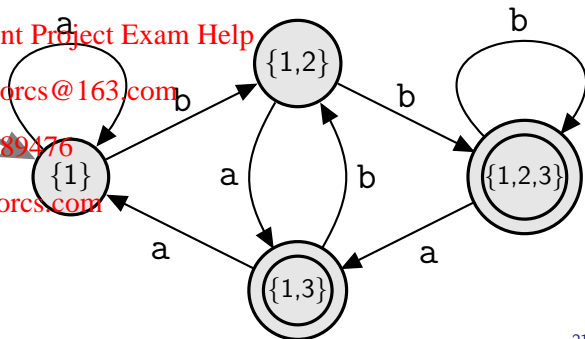
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

state		a	b
Start	{1}	{1}	{1,2}
	{1,2}	{1,3}	{1,2,3}
Final	{1,3}	{1}	{1,2}
Final	{1,2,3}	{1,3}	{1,2,3}



Algorithm: Conversion of NFA *without empty transitions* to FA

Input: a NFA

NextSetOfStatesOfNFA := { *程序代写代做CS编程辅导* Start State of NFA }.



Create new incomplete row in FA table for Start State called **NextSetOfStatesOfNFA**.

while the FA table still has at least one incomplete row **do**

CurrentStateInFA := the state for the first incomplete row of the FA.

for each letter x in the alphabet **do**

NextSetOfStatesOfNFA *Assignment Project Exam Help*

$\{q : \text{for some NFA-state } p \text{ in } \mathbf{CurrentStateInFA}, \exists \text{ transition } p \xrightarrow{x} q\}$
 Email: tutorscs@163.com

 Write **NextSetOfStatesOfNFA** *QQ: 749389476* in table entry for row **CurrentStateInFA**, column x .

if **NextSetOfStatesOfNFA** *https://tutorscs.com* is new **then**

 Create new incomplete row in table, using set **NextSetOfStatesOfNFA** as state.

Any FA state which (as a set) contains an NFA Final State is labelled Final.

Output: the FA

Converting a NFA to a FA

Now suppose that the NFA might have empty transitions, $q_1 \xrightarrow{\epsilon} q_2$.

These allow change of state without reading any letter of the input string.

Every time we include a new state q in **NextSetOfStatesOfNFA**, we also need to include any state we can reach from it along empty transitions.

Look at all paths from q that just use empty transitions

$$q \xrightarrow{\epsilon} q_1 \xrightarrow{\epsilon} q_2 \xrightarrow{\epsilon} \dots \xrightarrow{\epsilon} q_i$$

...and include all states on such paths.

Modify earlier algorithm, for constructing the sets of NFA states, to take account of empty transitions.



程序代写代做 CS编程辅导

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Algorithm: Conversion of NFA to FA

Input: a NFA

NextSetOfStatesOfNFA := { Start State of NFA }.

for *each* $q \in \text{NextSetOfStatesOfNFA}$ **do**
 Add, to **NextSetOfStatesOfNFA**, all states reachable from q along ϵ -transitions.
Create new incomplete row in FA table for Start State called **NextSetOfStatesOfNFA**.

while the FA table still has at least one incomplete row **do**

CurrentStateInFA := the state for the first incomplete row of the FA.

for *each* letter x in the alphabet **do**

NextSetOfStatesOfNFA := { q : for some NFA-state p in **CurrentStateInFA**, \exists transition $p \xrightarrow{x} q$ }

for *each* $q \in \text{NextSetOfStatesOfNFA}$ **do**
 Add, to **NextSetOfStatesOfNFA**, all states reachable from q along ϵ -transitions.

 Write **NextSetOfStatesOfNFA** in table entry for row **CurrentStateInFA**, column x .

if **NextSetOfStatesOfNFA** is new **then**

 Create new incomplete row in table, using set **NextSetOfStatesOfNFA** as state.

Any FA state which (as a set) contains an NFA Final State is labelled Final.

Output: the FA

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorscs@163.com

QQ: 749389476

https://tutorcs.com

Converting a NFA to a FA



WeChat: cstutorcs

state	a	b
Start {1,2,3}		



state	a	b
Start {1,2,3}	{1,2,3}	{2,3}
{2,3}		

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

state	a	b
Start {1,2,3}	{1,2,3}	{2,3}
{2,3}	∅	{2,3}
∅		



state	a	b
Start {1,2,3}	{1,2,3}	{2,3}
{2,3}	∅	{2,3}
∅	∅	∅

Converting a NFA to a FA



WeChat: cstutorcs

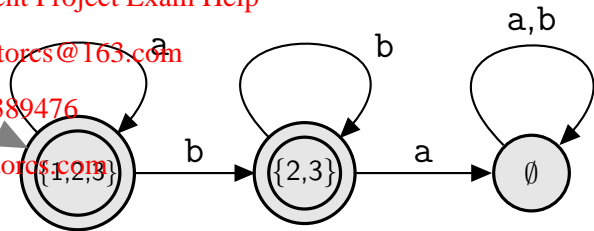
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

state	a	b
Start {1,2,3}	{1,2,3}	{2,3}
{2,3}	\emptyset	{2,3}
\emptyset	\emptyset	\emptyset



Converting a NFA to a FA

程序代写代做 CS编程辅导



Complexity?

- Think about how many states the FA may have, as a function of the number of states in the NFA.

WeChat: [cstutorcs](#)

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Revision

程序代写代做 CS编程辅导

Today:

- ▶ Understand Kleene's Theorem
- ▶ Be able to convert Regular Expression \rightarrow NFA
- ▶ Be able to convert NFA \rightarrow Finite Automaton

WeChat: cstutorcs

Next lecture:

Assignment Project Exam Help

- ▶ Be able to convert FA \rightarrow Regular Expression

Email: tutors@163.com

Reading:

QQ: 749389476

Sipser, Ch 1, especially pp. 54–58, 66–69.

http://tutorcs.com

