

程序代写代做 CS编程辅导

FIT2014 Theory of Computation



Lecture 9

Kleene's Theorem. II.

WeChat: cstutorcs

FA \rightarrow Regexp

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

Previous lecture:

- ▶ Convert Regular Expression
- ▶ Convert NFA to FA



WeChat: cstutorcs

Today:

- ▶ Generalised Nondeterministic Finite Automata
- ▶ Convert FA to GNFA
- ▶ Convert GNFA to Regular Expression

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Kleene's Theorem



Generalised Nondeterministic Finite Automaton (GNFA)

程序代写代做 CS编程辅导

Definitions

A **Generalised Nondeterministic Finite Automaton (GNFA)** is defined as for an NFA except that transitions may be labelled by regular expressions, not just by single letters.



WeChat: cstutorcs

A string w is **accepted** by a given GNFA if it can be divided into substrings, $w = w_1 \cdots w_k$, such that there is some sequence of transitions, starting at the Start State, finishing at the Final State, and labelled by regular expressions R_1, \dots, R_k , such that, for all i , w_i matches R_i .

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

If a string w is not accepted by the GNFA, then it is **rejected**.

<https://tutorcs.com>

Standard GNFA

A **standard GNFA** is a GNFA in which

- ▶ there is just one Final State, and it is not the Start State;
- ▶ the Start State has no incoming transitions and the Final State has no outgoing transitions.



Notes:

WeChat: cstutorcs

- ▶ Sometimes (e.g., in Sipser), standard GNFA's are required to have arcs going between every pair of states, subject to the prohibition on incoming arcs into the Start State and outgoing arcs from the Final State. If a transition actually cannot occur between two states, then the arc between them is labelled by the regular expression \emptyset , which prevents the transition from being used.
- ▶ This is not really needed, for the algorithm we will describe. But it may make proving properties of that algorithm easier.
- ▶ Standard GNFA's are sometimes called GNFA's of "special form" in Sipser.

Assignment Project Exam Help

Email: tutores@163.com

QQ: 749389476

https://tutorcs.com

From FA to Standard GNFA

程序代写代做 CS编程辅导

Given a FA (or an NFA):

1. Ensure there is a single Final State, with incoming arcs only.
 - ▶ If necessary: add new Final State, add new transitions labelled ε from the previous Final States to this new Final State, and make those states no longer Final.
2. Ensure there is a single Start State, with outgoing arcs only.
 - ▶ If necessary: add a new Start State, add new transitions labelled ε from this new Start State to the previous Start States, and make those states no longer Start states.

Email: tutorcs@163.com

The letters on the arcs of the FA/NFA are already regular expressions in their own right.

QQ: 749389476

<https://tutorcs.com>

The GNFA constructed by this process accepts the same language as the original FA.



WeChat: [cstutorcs](#)

Assignment Project Exam Help

Kleene's Theorem



From Standard GNFA to Regular Expression

程序代写代做 CS编程辅导

Starting with a Standard GNFA, convert it to an equivalent GNFA with one fewer state.



We keep doing this until we have a GNFA with just a Start State, a Final State, and one transition:

WeChat: cstutorcs



Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

The regular expression R on this transition defines the same language as the original GNFA.

<https://tutorcs.com>

From GNFA to Regular Expression

程序代写代做 CS编程辅导



Notation:

- q some non-Start, non-Final state.
- q_{in} any non-Final state.
- R_{in} the regular expression on the transition from q_{in} to q .
- q_{out} any non-Start state.
- R_{out} the regular expression on the transition from q to q_{out} .
- R_{loop} the regular expression on the transition from q to itself.
- R_{direct} the regular expression on the transition from q_{in} to q_{out} .

QQ: 749389476

<https://tutorcs.com>

From GNFA to Regular Expression

程序代写代做 CS编程辅导



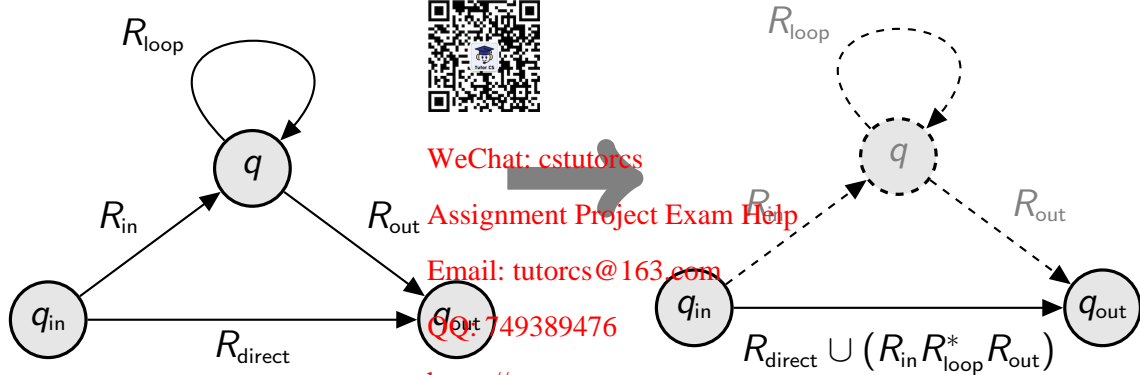
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



From GNFA to Regular Expression

程序代写代做 CS编程辅导

Ensure this replacement is done for all q_{in}, q_{out} .



Then q is removed, and we have an equivalent GNFA to the one we started with.

Keep doing this whole procedure, removing one state at a time, until you are left with just the Start State and the Final State, with a single transition between them.

WeChat: cstutorcs

Assignment Project Exam Help

The regular expression on this transition is the one you want.

Email: tutorcs@163.com

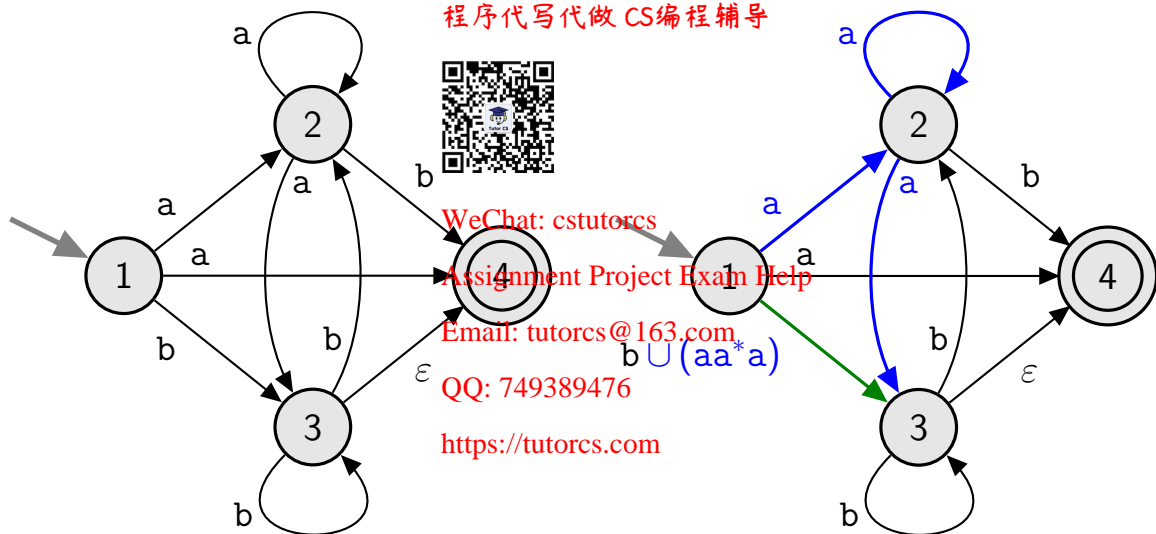
It matches precisely those strings accepted by the original GNFA.

QQ: 749389476

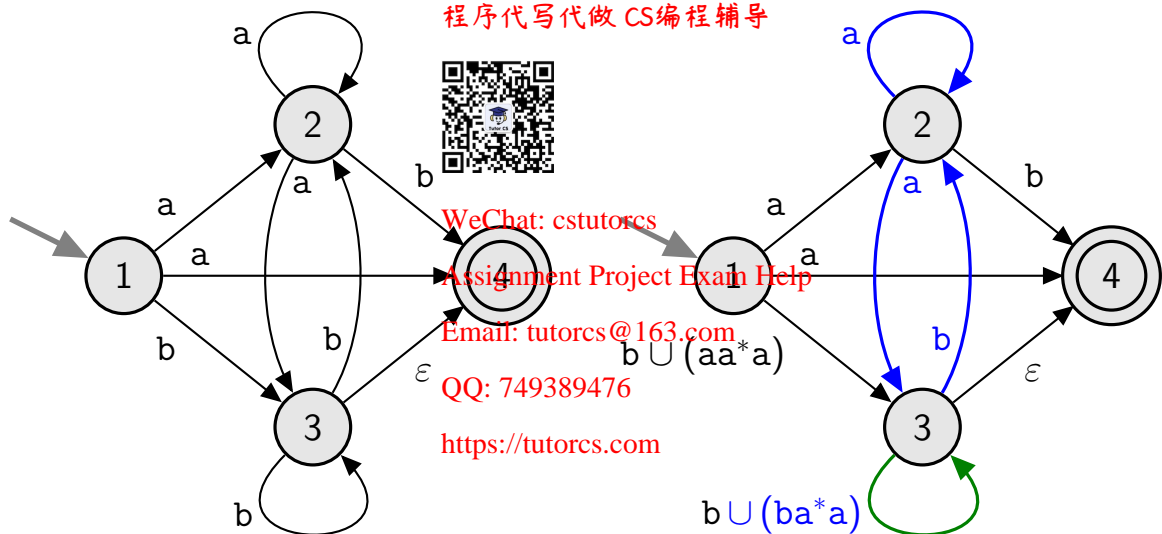
Examples: Sipser, pp. 75–76.

<https://tutorcs.com>

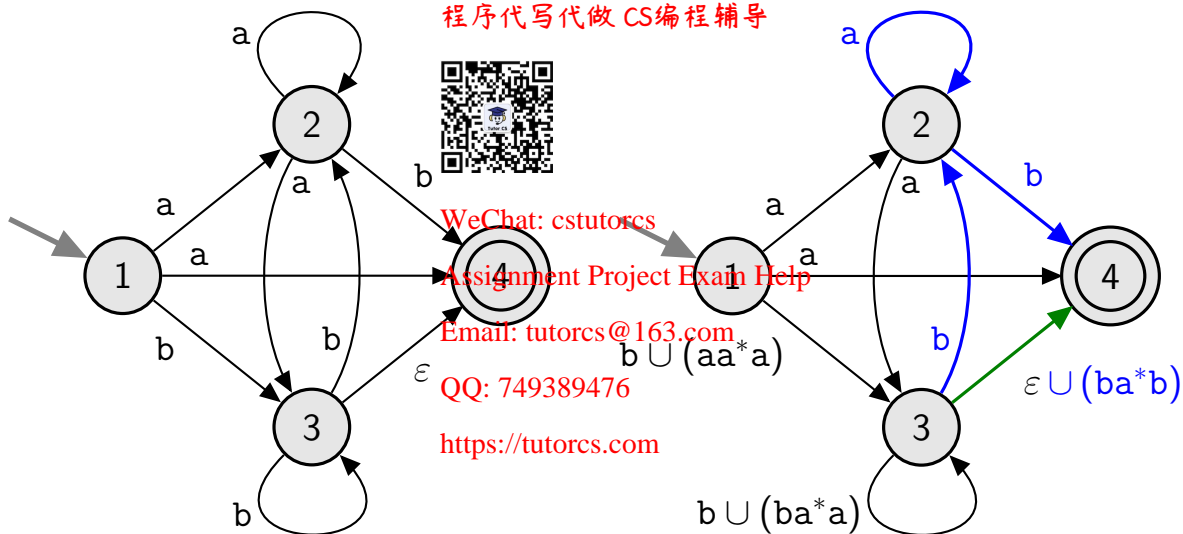
From GNFA to Regular Expression



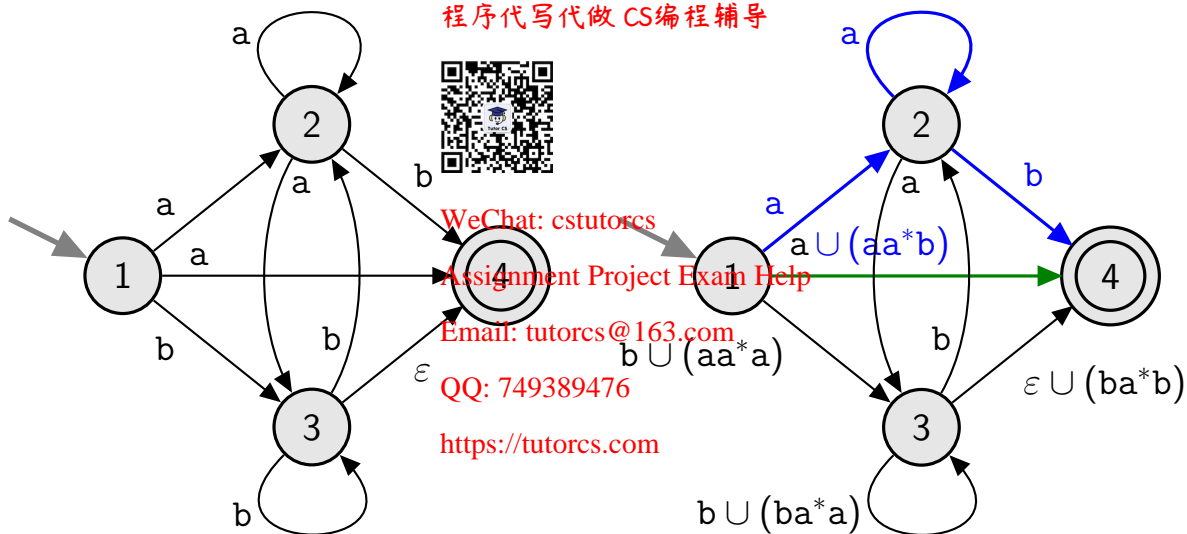
From GNFA to Regular Expression



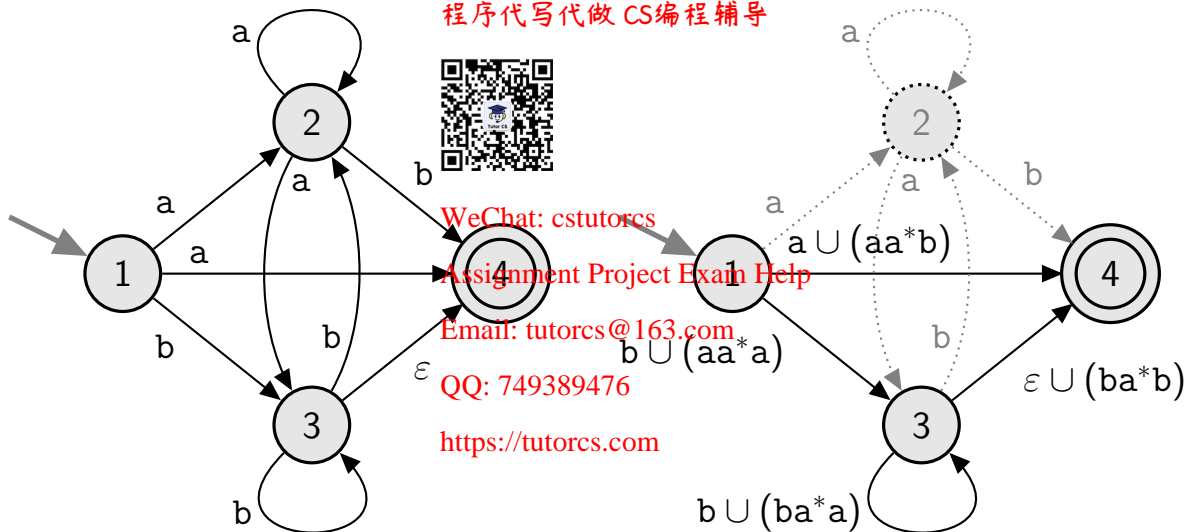
From GNFA to Regular Expression



From GNFA to Regular Expression



From GNFA to Regular Expression



From GNFA to Regular Expression

程序代写代做 CS编程辅导



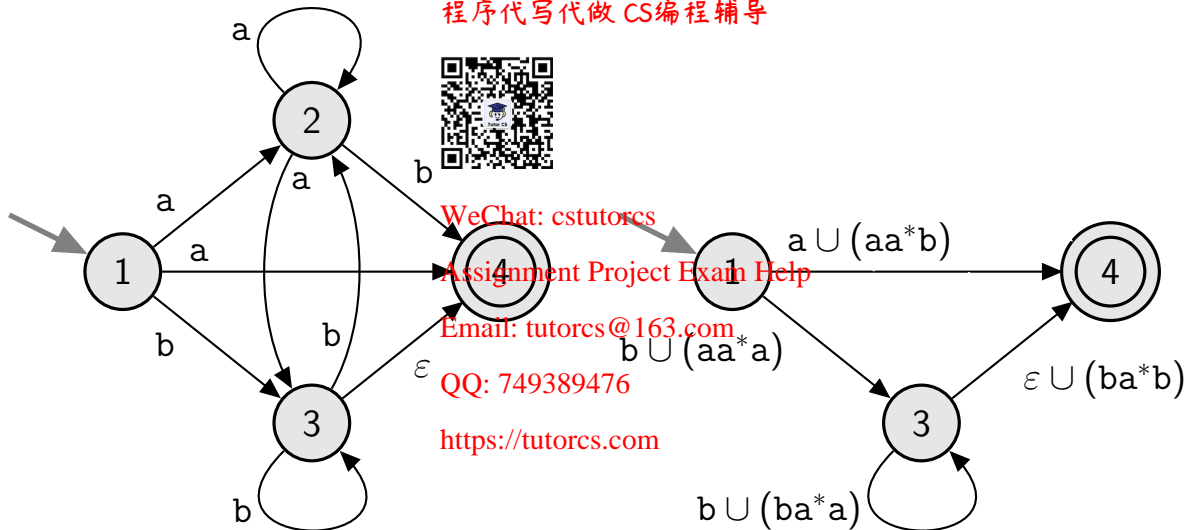
WeChat: cstutorcs

Assignment Project Exam Help

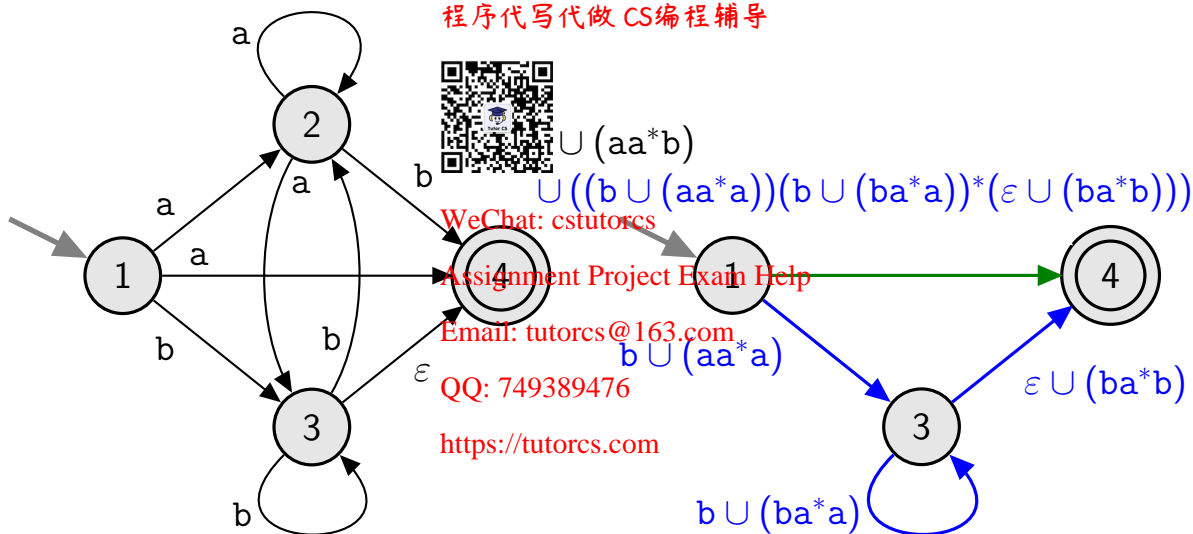
Email: tutorcs@163.com

QQ: 749389476

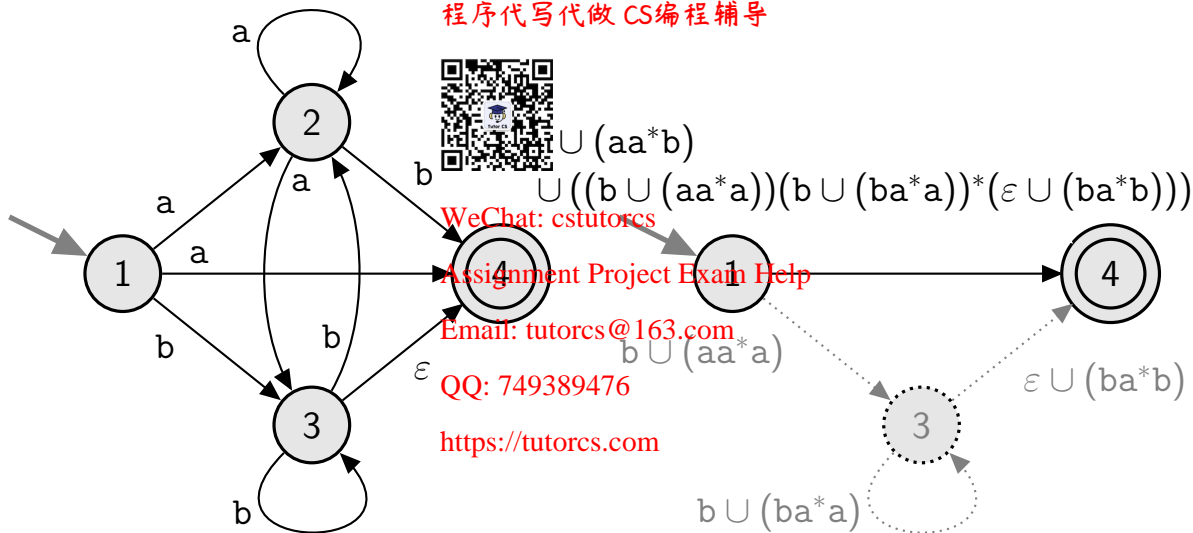
<https://tutorcs.com>



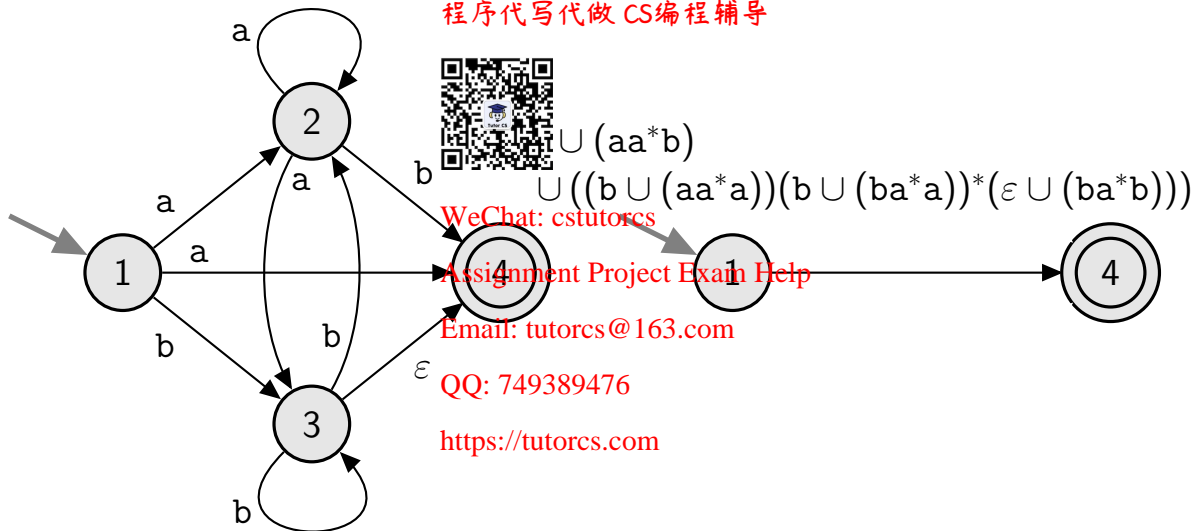
From GNFA to Regular Expression



From GNFA to Regular Expression



From GNFA to Regular Expression



From GNFA to Regular Expression

程序代写代做 CS编程辅导



Complexity?

For FIT2004 students:

Compare this algorithm with the Floyd-Warshall algorithm for the All Pairs Shortest Path problem.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

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 YES
- ▶ Can every language which is described by a **finite automaton** be represented by a **regular expression**? YES
- ▶ Can every language be represented by a **regular expression** or a **finite automaton**?



{ all languages }

WeChat: estutores

representable by recognised by

regex

FA

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

?

Revision

程序代写代做 CS编程辅导

Previous lecture:

- ▶ Understand Kleene's Theorem
- ▶ Be able to convert Regular Expressions into NFA
- ▶ Be able to convert NFA into a Finite Automaton



WeChat: cstutorcs

Today:

Assignment Project Exam Help

- ▶ Be able to convert a FA into a Regular Expression

Email: tutorcs@cs.cmu.edu

Reference

QQ: 749389476

Sipser, Ch. 1, especially pp. 66, 69-76. <https://tutorcs.com>