

程序代写代做 CS编程辅导

FIT2014 Theory of Computation



Lecture 13

- (A) Regular Grammars,
 - (B) Pushdown Automata
- Assignment Project Exam Help

Email: tutorcs@163.com

slides by Graham Farr

based in part on previous slides by David Albrecht

<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



- ▶ $\{\text{Regular Languages}\} \subseteq \{\text{CFLs}\}$
- ▶ Pushdown Automaton (PDA)
- ▶ Constructing PDA to accept a Regular Grammar

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

all languages

程序代写代做 CS编程辅导



Is every Regular Language
a Context-Free Language?

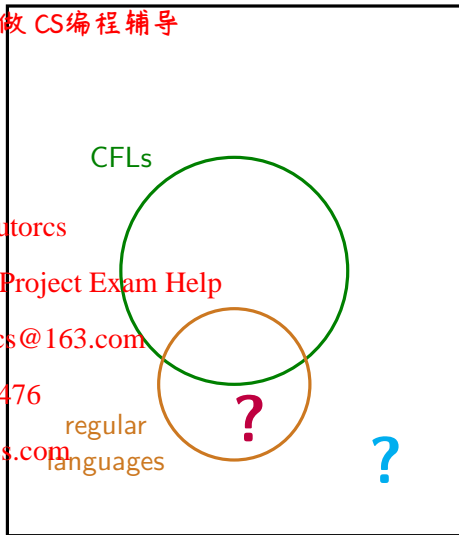
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



NFA to CFG

Input: an NFA

程序代写代做 CS编程辅导

1. Name all the states in the symbol.
► Call the Start State S .
2. For each ordered pair of states X, Y linked by an arc labelled z , create production rule $X \xrightarrow{z} Y$.
3. For each Final State X , create production rule $X \rightarrow \epsilon$.



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

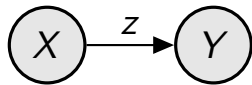
Output: the CFG consisting of

- terminals: the alphabet of the NFA;
- non-terminals: the symbols representing the states of the NFA;
- all the production rules we have created.

QQ: 749389476

http://tutorcs.com

NFA to CFG



$X \longrightarrow zY$



WeChat: cstutorcs

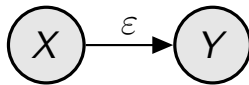
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

$X \longrightarrow zX$

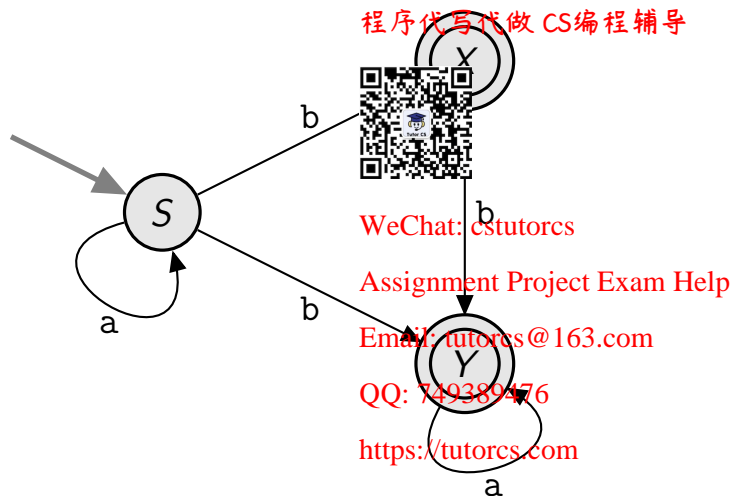


$X \longrightarrow Y$



$X \longrightarrow \varepsilon$

NFA to CFG: Example



$S \rightarrow aS$

$S \rightarrow bX$

$S \rightarrow bY$

$X \rightarrow bY$

$Y \rightarrow aY$

$X \rightarrow \varepsilon$

$Y \rightarrow \varepsilon$

Regular Grammar

程序代写代做 CS编程辅导



Definitions

Semiwords are of the form:

terminal terminal ... terminal Nonterminal

WeChat: cstutorcs

A CFG is called a **Regular Grammar** if all its production rules are in one of the following forms:

Assignment Project Exam Help

Nonterminal \rightarrow semiword

Email: tutorcs@163.com

or

Nonterminal \rightarrow string of terminals

QQ: 749389476

<https://tutorcs.com>

Theorem.

Every Regular Language can be generated by a Regular Grammar.

Proof idea:

A regular language is recognised by some NFA

Observe that our construction $NFA \rightarrow CFG$ produces a regular grammar.

Theorem.

Every Regular Grammar generates a Regular Language.

Proof: Exercise.

程序代写代做 CS编程辅导



WeChat: cstutorcs

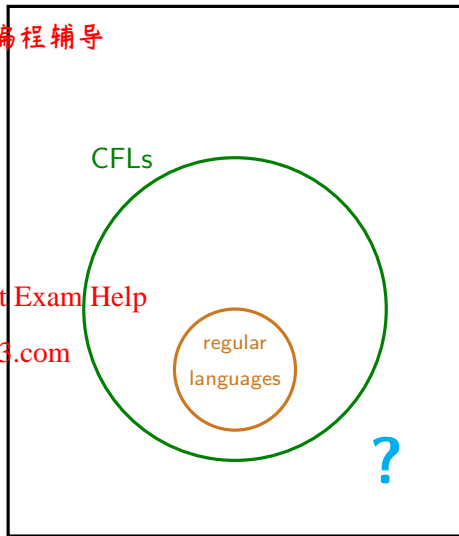
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

all languages



Overview

程序代写代做 CS编程辅导



Is there a state machine for Context-Free Languages?

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Pushdown Automaton (PDA)

程序代写代做 CS编程辅导



- ▶ A Nondeterministic Finite Automaton (NFA) with a **Stack**.

- ▶ Can be used to represent Context Free Languages.

Assignment Project Exam Help

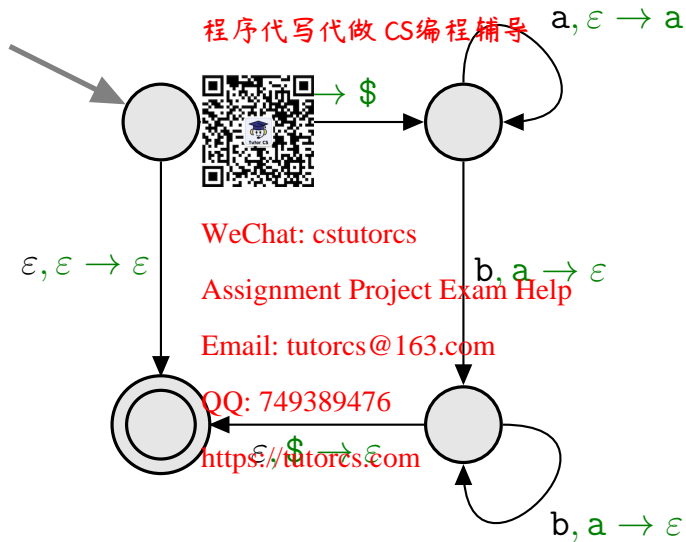
- ▶ Many parsers use a Pushdown Automaton
 - ▶ ...including the parsers generated by some compiler-compilers.

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Pushdown Automaton (PDA)



Pushdown Automaton (PDA)

程序代写代做 CS编程辅导



The Stack

- ▶ storage for letters
 - ▶ serves as a memory
- ▶ two operations:
 - ▶ **Push**: puts a letter on the top of the stack
 - ▶ **Pop**: takes a letter off the top of the stack.

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Pushdown Automaton (PDA)

程序代写代做 CS编程辅导

Transitions



$y \rightarrow z$

which means when the machine is reading x , if there is a y on top of the stack, it is replaced by z .

WeChat: cstutorcs

- ▶ If x is ε then no letter is read from the tape.

Assignment Project Exam Help

Email: tutorcs@163.com

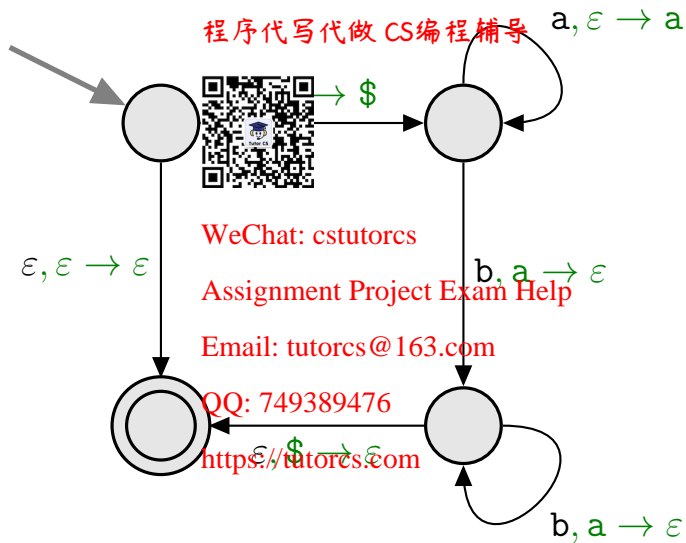
- ▶ If y is ε then no letter is popped from the stack.

QQ: 749389476

- ▶ If z is ε then no letter is pushed onto the stack.

<https://tutorcs.com>

Pushdown Automaton (PDA)



Pushdown Automaton (PDA)

A **Pushdown Automaton** consists of:

- ▶ an **input alphabet**: the possible input letters.
- ▶ a **stack alphabet**: the possible stack letters.
- ▶ a stack
- ▶ a finite set of states
 - ▶ One called the **Start State**
 - ▶ Some (maybe none) called **Final States**
- ▶ A set of transitions between states

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assessment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476 $\rightarrow z$

which means when the machine is reading x , if there is a y on top of the stack, it is replaced by z .

<https://tutorcs.com>

Pushdown Automaton (PDA)

程序代写代做 CS编程辅导



Definitions

A string is **accepted** by a PDA if there exists at least one path through the PDA for this string that ends in a Final State.

WeChat: cstutorcs

A string is **rejected** by a PDA if for all paths through the PDA for this string, the PDA either crashes or ends in a non-Final State.

Email: tutorcs@163.com

The set of strings accepted by the PDA is called the **language accepted** by the PDA.

QQ: 749389476

<https://tutorcs.com>

HALF-AND-HALF

程序代写代做 CS编程辅导



$$\begin{aligned} \text{HALF-AND-HALF} &= \{a^n b^n : n \geq 0\} \\ &= \{\varepsilon, ab, aabb, aaabbb, \dots\} \end{aligned}$$

WeChat: cstutorcs

Using the Pumping Lemma we showed that this language was non-regular.

Assignment Project Exam Help

Consider:

Email: tutorcs@163.com

QQ: 749389476

So it is a Context-Free Language.

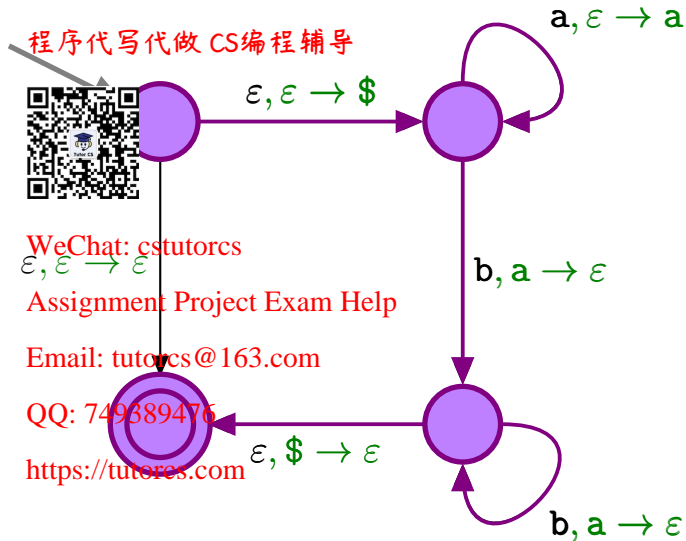
<https://tutorcs.com>

PDA for HALF-AND-HALF

Input: aaabbb

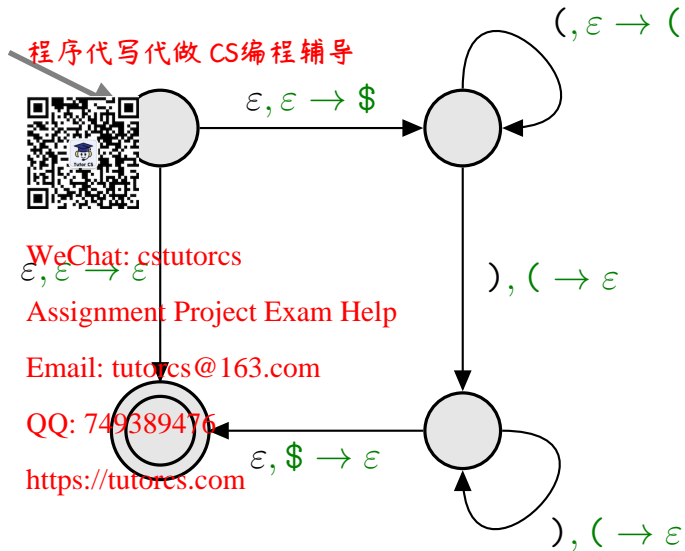
Stack:

a
a
a
\$



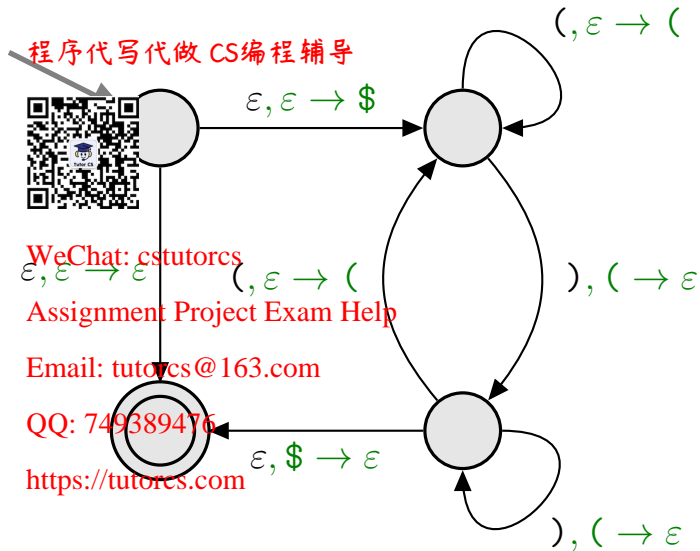
PDA for PARENTHESES

1. $S \rightarrow \varepsilon$
2. $S \rightarrow (S)$
- 3.



PDA for PARENTHESES

1. $S \rightarrow \epsilon$
2. $S \rightarrow (S)$
3. $S \rightarrow SS$



Regular Languages \subseteq languages accepted by PDAs

程序代写代做 CS编程辅导



Exercise:

- What do you have to do to restrict a PDA so that it is just an NFA?

WeChat: cstutorcs

An NFA is a special case of a PDA.

So

Assignment Project Exam Help

{regular languages} \subseteq {languages recognised by a PDA}

QQ: 749389476

<https://tutorcs.com>

Regular Languages \subseteq languages accepted by PDAs

$S \rightarrow aS \mid bY \mid bX$

$X \rightarrow bY \mid \varepsilon$

$Y \rightarrow aY \mid \varepsilon$

程序代写代做 CS编程辅导



$a, S \rightarrow S$

$b, S \rightarrow Y$

$b, S \rightarrow X$

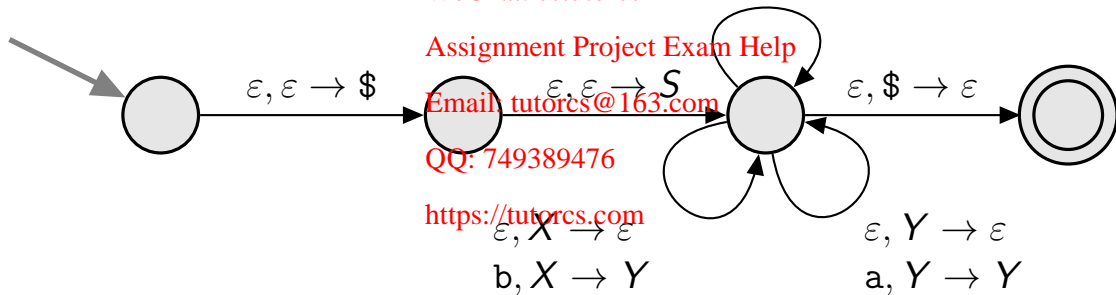
WeChat: cstutorcs

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

Things to think about:

- ▶ Which of the PDAs we've seen is *deterministic*?
- ▶ Suppose we plot **stack height** against **time** for our PDA for the Dyck language. What would it look like?
- ▶ How can we construct, from a given CFG, a PDA for the same language?
- ▶ How can we construct, from a given PDA, a CFG for the same language?

WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com

QQ: 749389476

Reading: Sipser, Section 2.2, pp. 111–116.

<https://tutorcs.com>