

程序代写代做 CS编程辅导

Introduction to Theory of Computation
lecture 12



cs2001

WeChat: cstutorcs

Ruth Urner
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476
February 15, 2023

<https://tutorcs.com>

程序代写代做 CS 编程辅导



Intro

eecs2001

The Theory of Computation

Lecture 12

WeChat: cstutorcs

Ruth Umer

Reminder: Assignment Project Exam Help !

Email: tutors@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



Intro

eecs2001

The Theory of Computation

Lecture 12

Q3 on A2: Ruth Ormer
Regular languages are closed under $OPD(\cdot, \cdot)$
Show regular languages are closed under
Assignment Project Exam Help

You can use Email: tutors@163.com

Regular languages are closed under
- union intersection - complement
- concatenation - complements
- short input

<https://tutorcs.com>

程序代写代做 CS编程辅导



Regular Expressions

WeChat: cstutorcs
Reading: ITC Section 1.3
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Regular expression

程序代写代做 CS编程辅导



A **regular expression** is a compact way of defining a set of words. It is a sequence of symbols that represent a language over some alphabet Σ .

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Inductive definition of sets – general pattern

程序代写代做 CS编程辅导

An inductive definition consists of

1. A universe set 
 2. A core set $C \subseteq U$
 3. A finite set $O = \{o_1, o_2, \dots, o_n\}$ of operations from
 $o_i : U^{r_i} \rightarrow U$ for some arities $r_i \in \mathbb{N}$
- WeChat: cstutorcs
Assignment Project Exam Help

Email: tutorcs@163.com

We define $\mathcal{I}(U, C, O)$ as the set of elements that we obtain by
starting with the core set and putting all those elements of U into
 $\mathcal{I}(U, C, O)$ that one can reach by successively applying the
operations in O .

QQ: 749389476

<https://tutorcs.com>

Regular expression–inductive definition

程序代写代做 CS编程辅导

Let Σ be some alphabet. We define the set \mathcal{R} of regular expressions over Σ inductively by setting $\mathcal{R}_\Sigma = \mathcal{I}(U, C, O)$, where



1. The **universe** U is the set of all strings over $\Sigma \cup \{(,), \cup, \circ, ^*\}$.
WeChat: cstutorcs
2. The **core set** C is the set of all symbols in Σ and ϵ, \emptyset and two additional symbols: $C = \Sigma \cup \{\epsilon, \emptyset\}$.
Email: tutorcs@163.com
3. Three operations:
 - $\circ_\cup(R_1, R_2) = (R_1 \cup R_2)$,
QQ: 749389476
 - $\circ_\circ(R_1, R_2) = (R_1 \circ R_2)$
<https://tutorcs.com>
 - $\circ_*(R) = (R^*)$.

程序代写代做 CS 编程辅导

Regular expression - Inductive definition



Let Σ be so

We define the set \mathcal{R} of regular expressions

1. The universe U is the set of all strings over

$$\Sigma \cup \{(\cdot) \cup \cdot^*, \epsilon, \emptyset\}$$

WeChat: cstutorcs

2. The core set C is the set of all symbols in Σ and ϵ, \emptyset and two additional symbols: $C = \Sigma \cup \{\epsilon, \emptyset\}$

Assignment Project Exam Help

3. Three operations:

$$o_1(R_1, R_2) = (R_1 \cup R_2)$$

$$o_2(R_1, R_2) = (R_1 \circ R_2),$$

$$o_3(R) = (R^*)$$

Exercise:

Email: tutorcs@163.com

induction)

that the number of "("

QQ: 749389476 always equal to the

number of ")" in a regular

expression

014(Σ

number

of universe

)co

these two

are not

regular

expressions

Example:
 $\Sigma = \{0, 1\}$

0 ✓

(1 ∪ 0) ✓

(1 ∘ 1) X

ε ✓

(ε ∘ 1) ✓

The language of a regular expression

程序代写代做 CS编程辅导

Each regular expression $R \in \mathcal{R}_{\Sigma}$ over some alphabet Σ represents a language over Σ . We define the inductive definition $L(R)$ of a regular expression R according to the inductive definition.



Members of the core-set

- The expression a for $a \in \Sigma$ represents the language $\{a\}$, that is $L(a) = \{a\}$.
WeChat: cstutorcs
- The expression ϵ represents the language $\{\epsilon\}$, that is $L(\epsilon) = \{\epsilon\}$.
Assignment Project Exam Help
- The expression \emptyset represents the language \emptyset , that is $L(\emptyset) = \emptyset$.

Email: tutorcs@163.com

Result of operation: For regular expressions R_1, R_2 and R , we define:

- $L((R_1 \cup R_2)) = L(R_1 \cup R_2)$
QQ:749389476
- $L((R_1 \circ R_2)) = L(R_1) \circ L(R_2)$
https://tutorcs.com
- $L((R^*)) = L(R)^*$

We call $L(R)$ the language of R .

$\Sigma = \{0, 1\}$

$0 \in \Sigma$

$L(0) = \{0\}$

$0, \epsilon \in \Sigma$

$(0 \cup \epsilon) \in \Sigma$

$L((0 \cup \epsilon))$

$= \{0, \epsilon\}$

$(0 \cup \emptyset) \in \Sigma$

$L((0 \cup \emptyset))$

$= \{0\}$

The language of a regular expression



Each regular expression over some alphabet Σ represents a language $L(R)$ of a regular expression R according to the induction principle.

Members of

- The expression a for $a \in \Sigma$ represents the language $\{a\}$, that is $L(a) = \{a\}$.
- The expression ϵ represents the language $\{\epsilon\}$, that is $L(\epsilon) = \{\epsilon\}$.
- The expression \emptyset represents the language \emptyset , that is $L(\emptyset) = \emptyset$.

WeChat: cstutorcs

Assignment Project Exam Help

Result of operation: For regular expressions R_1 , R_2 and R , we define:

- $L((R_1 \cup R_2)) = L(R_1) \cup L(R_2)$

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Regular expression—additional notation

程序代写代做 CS编程辅导

For convenience and readability we use the following notational conventions:



1. For an alphabet Σ , \emptyset is a regular expression representing all words of length 1 over Σ . And then Σ^* is a regular expression for the set of all words over Σ .
2. We often omit brackets. The order of precedence then is: *, o, U.
3. The o-symbol is typically omitted: we use $R_1 R_2$ as shorthand for $R_1 \circ R_2$.
4. We let R^+ be shorthand for RR^* .
5. We let R^k be the k times repeated concatenation of R with itself:
$$R^k = R \circ R \circ R \circ \dots \circ R$$

QQ: 749389476

<https://tutorcs.com>

Regular expressions—examples

程序代写代做 CS编程辅导

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ :



- 0^*10^*
- $\Sigma^*1\Sigma^*$
- $\Sigma^*001\Sigma^*$
- $1^*(01^+)^*$
- $(\Sigma\Sigma)^*$
- $(\Sigma\Sigma\Sigma)^*$

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

$L((\sigma^*))$
 $= \{\text{set of all words that have only letters}\}$

↑
It includes
 ϵ !

$L((\sigma^*))$
 $= \{\epsilon, 0, 00, \dots\}$

程序代写代做 CS 编程辅导

precedence



We let $\Sigma =$

Now we can

following regular expressions over Σ :

- $0^* 1 0^*$ ($\underbrace{0^* 1}_{\text{is included}} \underbrace{0^*})$)

$= \{\omega \text{ such that } \omega \text{ has exactly one } 1\}$

$1 \in L(\sigma^*)$ Assignment Project Exam Help

$0010011(\sigma^* 1 \sigma^*)$ Email: tutorcs@163.com

$100000 \in L(\sigma^* 1 \sigma^*)$ QQ: 749389476

<https://tutorcs.com>

程序代写代做CS编程辅导

Regular expressions-examples

We let $\Sigma =$



Now we can

following regular expressions over Σ :

- 0^*10^*

- $\Sigma^*1\Sigma^* \cup (\Sigma^*)1 \cup (\Sigma^*)1 = (\Sigma^*)0 \cup 1 \cup (\Sigma^*)$

WeChat: cstutorcs
 $L(\Sigma^*1\Sigma^*) = \{\omega \in \Sigma^* \mid \omega \text{ contains at least one } 1\}$
Assignment Project Exam Help

$1 \in L(\Sigma^*1\Sigma^*)$
Email: tutorcs@163.com

$\underbrace{01011}_{\in \Sigma^*} - \underbrace{01}_{\in \Sigma^*} \cup (\Sigma^*1\Sigma^*)$
QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS 编程辅导

Regular expressions—examples

We let $\Sigma =$



Now we can

following regular expressions over Σ :

- 0^*10^*
- $\Sigma^*1\Sigma^*$ WeChat: cstutorcs
- $\Sigma^*001\Sigma^*$

*↳ any string containing 001
as a substring*

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做CS编程辅导

Regular expressions-examples

* o U

We let $\Sigma =$



Now we can

following regular expressions over Σ :

- 0^*10^*
- $\Sigma^*1\Sigma^*$ WeChat: cstutorcs
- $\Sigma^*001\Sigma^*$
- $1^*(01^+)^*$

Assignment Project Exam Help

$L(1^*(01^+)^*) = \{w \in \Sigma^* \mid \text{every } 0$
 $\text{is followed by a } 1\}$

Email: tutorcs@163.com

QQ: 749389476
 $w \in 1^*$

<https://tutorcs.com>

程序代写代做 CS 编程辅导

Regular exp



Who can see what you share here? Recording On

precedence

We let $\Sigma =$



Now we can

following regular expressions over Σ :

- 0^*10^*
- $\Sigma^*1\Sigma^*$
- $\Sigma^*001\Sigma^*$
- $1^*(01^+)^*$

Assignment Project Exam Help

$L(1^*(01^+)^*) = \{ \omega \in \Sigma^* \mid \text{every } 0 \text{ is followed by a } 1 \}$

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS 编程辅导

Regular expression



We let $\Sigma =$

Now we can

following regular expressions over Σ :

- 0^*10^* $L(\Sigma) = \{w \in \Sigma^* \mid w \text{ has only letters}\}$
- $\Sigma^*1\Sigma^*$ WeChat: cstutorcs $= \{0, 1\}$
- $\Sigma^*001\Sigma^*$ Assignment Project Exam Help $= \{\text{Assignment, Project, Exam, Help}\}$
- $1^*(01^+)^*$ Email: tutorcs@163.com
- $L((\Sigma\Sigma)^*) = \{w \in \Sigma^* \mid w \text{ has an even length}\}$
QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS 编程辅导

Regular exp

Mouse Select T Draw Stamp Eraser Format Undo Redo Clear Save
Who can see what you share here! Recording On

Saved as PNG Show In Folder

We let $\Sigma =$



Now we can follow regular expressions over Σ :

- 0^*10^*

- $\Sigma^*1\Sigma^*$ WeChat: cstutorcs

- $\Sigma^*001\Sigma^*$

Assignment Project Exam Help

- $1^*(01^+)^*$

- $(\Sigma\Sigma)^*$ Email: tutorcs@163.com

- $(\Sigma\Sigma\Sigma)^*$ QQ: 749389476 | the length of σ

$L((\Sigma\Sigma\Sigma)^*)$ is divisible by 3

<https://tutorcs.com>

Regular expressions—examples

程序代写代做 CS编程辅导

We let $\Sigma = \{0, 1\}$.

Now we can interpret the following regular expressions over Σ :



- $01 \cup 10$

- $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$

- $(0 \cup \epsilon)1^*$

- $(0 \cup \epsilon)(1 \cup \epsilon)$

- $1^*\emptyset$

- \emptyset^*

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

程序代写代做CS编程辅导

Regular exp

$\Sigma = \{0, 1\}$

We let $\Sigma =$



Now we can define the following regular expressions over Σ :

- $01 \cup 10$ (union) $(01) \cup (10)$

Construction of regular expression
WeChat: **tutorcs**

1. 1 (one-set) $L(1) = \{1\}$
2. 0 (one-set) $L(0) = \{0\}$
3. $(0 \cdot 1)$ (apply in $0_0 \cdot 1_0 \cdot 2+1$) $L((0 \cdot 1)) = \{01\}$
4. $(1 \cdot 0)$ (apply in $1_1 \cdot 0_0 \cdot 2+1$) $L((1 \cdot 0)) = \{10\}$
5. $((0 \cdot 1) \cup (1 \cdot 0))$ (combining $0_0 \cdot 1_0 \cdot 3+4$)
QQ: 749389476 $L((0 \cdot 1) \cup (1 \cdot 0)) = \{01, 10\}$

<https://tutorcs.com>

程序代写代做 CS 编程辅导

Regular exp



We let $\Sigma =$

Now we can

following regular expressions over Σ :

- $01 \cup 10$

- $(0\Sigma^*0) \cup (\Sigma^*1) \cup 0111$

WeChat: cstutorcs

$$L(0) = \{0\}$$

$$L(1) = \{\}$$

Assignment Project Exam Help

$$L(0\Sigma^*0) = \{w \in \Sigma^* \mid w \text{ starts and ends with } 0 \text{ and has length at least 2}\}$$

$$L(1\Sigma^*1) = \{w \in \Sigma^* \mid w \text{ starts and ends with } 1 \text{ and has length at least 2}\}$$

$$L(0\Sigma^*1) = \{w \in \Sigma^* \mid w \text{ starts with } 0 \text{ and ends with } 1\}$$

Email: tutorcs@163.com

QQ: 749389476

$$L(0\Sigma^*0 \cup 1\Sigma^*1 \cup 011) = \{w \in \Sigma^* \mid w \neq \epsilon \text{ and starts and ends on the same letter}\}$$

<https://tutorcs.com>

程序代写代做 CS 编程辅导

Regular exp

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

Saved as PNG Show in Folder

We let $\Sigma =$



Now we can

following regular expressions over Σ :

- $01 \cup 10$

- $0\Sigma^*0 \cup \Sigma^*1 \cup \epsilon$

- $(0 \cup \epsilon)^*$

$L((0 \cup \epsilon)^*) = \{w \in \Sigma^* \mid w \text{ has only } 1 \text{ or }$

$L((0 \cup \epsilon)^*) = \{w \in \Sigma^* \mid w \text{ contains a single } 0 \text{ at }$

$\{\text{the start}\}$

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS 编程辅导

Regular exp

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording On

We let $\Sigma =$



Now we can

following regular expressions over Σ :

- $01 \cup 10$
- $0\Sigma^*0 \cup \Sigma^*1 \cup \epsilon$
- $(0 \cup \epsilon)1^*$
- $(0 \cup \epsilon)(1 \cup \epsilon)$
- $1^*\emptyset$
- \emptyset^*

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476
(converting any language with \emptyset results in \emptyset)

<https://tutorcs.com>

Talking: Ruth Umer

程序代写代做 CS 编程辅导

Regular expr

Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

What can you see what you share here? Recording On

Saved as PNG Show in Folder

We let $\Sigma =$



Now we can

following regular expressions over Σ :

- $01 \cup 10$

- $0\Sigma^*0 \cup \Sigma^*1 \cup \epsilon$

- $(0 \cup \epsilon)1^*$

Assignment Project Exam Help

- $(0 \cup \epsilon)(1 \cup \epsilon)$

- $\underline{1^*\emptyset}$

Email: tutorcs@163.com

- \emptyset^*

QQ: 749389476

L (\emptyset^*) = $\{\epsilon\}$

https://tutorcs.com

Regular expressions—more examples

Let R be some regular expression. Then we have:

- $L(R \cup \emptyset) = L(R)$



- $L(R \circ \epsilon) = L(R)$

WeChat: cstutorcs

Assignment Project Exam Help

- $L(R \cup \epsilon)$ is not necessarily equal to $L(R)$

Email: tutorcs@163.com

QQ: 749389476

- $L(R \circ \emptyset)$ is not necessarily equal to $L(R)$

<https://tutorcs.com>

程序代写代做 CS编程辅导

Regular exp

House Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save

Who can see what you share here? Recording Off

Let R be so



expression. Then we have:

- $L(R \cup e) = L(R) \cup L(e)$

- $L(R \circ e) = L(R)$

WeChat: cstutorcs

- $L(R \cup e) = L(R) \cup L(e)$

Example: $L(\{1\}) = \{\epsilon, 1, 10\}$, $\text{且 } L(\{1\} \cup \{e\}) = \{\epsilon, 1, 10\}$

Email: tutorcs@163.com

- $L(R \circ \emptyset)$ is not necessarily equal to $L(R)$

Example: $L(\{1\}) = \{\epsilon, 1, 10\}$, $L(\{1\} \circ \emptyset) = \emptyset$

QQ: 749389476

<https://tutorcs.com>

Expressive power of regular expressions

程序代写代做 CS编程辅导

Theorem

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



WeChat: cstutorcs

Assignment Project Exam Help

Lemma 1

If a language is described by a regular expression, then it is regular.

Email: tutors@163.com

QQ: 749389476

Lemma 2

If a language is regular, there is a regular expression that describes it.

<https://tutorcs.com>

程序代写代做 CS 编程辅导

Expressive p.



Theorem

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



WeChat: cstutorcs

we will prove this part

Lemma 1

If a language is described by a regular expression, then it is regular.



Lemma 2 Email: tutorcs@163.com

If a language is regular, then there is regular expression that describes it.

QQ: 749389476
for the proof in the textbook

<https://tutorcs.com>

Expressive power of regular expressions—proof of Lemma 1

Let Σ be some alphabet. We prove by induction (according to the inductive definition of regular expressions) that for every regular expression R there exists and NFA that recognizes language $L(R)$ (and this implies that $L(R)$ is a regular language).

Base case

We prove that the claim is true for members of the core-set.



1. If $R = a$ for some member of the alphabet Σ then we have $L(R) = \{a\}$.
WeChat: cstutorcs
We can construct an NFA recognizing $L(R)$ as follows:

Assignment Project Exam Help

2. If $R = \epsilon$, then we have $L(R) = \{\epsilon\}$. We can construct an NFA recognizing $L(R)$ as follows:
Email: tutorcs@163.com
QQ: 749389476

<https://tutorcs.com>

3. If $R = \emptyset$, then we have $L(R) = \emptyset$. We can construct an NFA recognizing $L(R)$ as follows:

Expressive power

程序代写代做 CS编程辅导

Let Σ be some definition of regular language L and NFA that



prove by induction (according to the inductive hypothesis) that for every regular expression R there exists an NFA that recognizes the language $L(R)$ (and this implies that $L(R)$ is a regular language).

Base case

We prove that the claim is true for members of the core-set.

1. If $R = a$ for some member of the alphabet Σ then we have $L(R) = \{a\}$.

We can construct an NFA recognizing $L(R)$ as follows:



WeChat: cstutorcs

2. If $R = \epsilon$, then we have $L(R) = \{\epsilon\}$. We can construct an NFA recognizing $L(R)$ as follows:

Email: tutorcs@163.com

3. If $R = \emptyset$, then we have $L(R) = \emptyset$. We can construct an NFA recognizing $L(R)$ as follows:



QQ: 749389476

https://tutorcs.com

Talking: Ruth Umer

Expressive power of regular expressions—proof of Lemma 1

Induction Hypothesis

程序代写代做 CS编程辅导

We assume that for two regular expressions R_1 and R_2 there exists NFAs N_1 and N_2 such that

$$L(R_1)$$



$$\text{and}$$

$$L(R_2) = L(N_2)$$

Induction Step

We need to show that, given the induction hypothesis, there exist NFAs that recognize the languages obtained by applying the three operations to the expressions R_1 and R_2 .

1. We have $o_{\cup}(R_1, R_2) = (R_1 \cup R_2) \circ N_1 \cup (R_2 \cup R_1) \circ N_2 = L(R_1) \cup L(R_2)$. By the induction hypothesis, there exist NFAs N_1 and N_2 recognizing $L(R_1)$ and $L(R_2)$. We have seen in Lecture 10, how to construct an NFA N recognizing the language $L(R_1) \cup L(R_2)$ (regular languages are closed under unions).

WeChat: estutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

2. We have $o_{\circ}(R_1, R_2) = (R_1 \circ R_2) \circ N_1 = L(R_1) \circ L(R_2)$. By the induction hypothesis, there exist NFAs N_1 and N_2 recognizing $L(R_1)$ and $L(R_2)$. We have seen in Lecture 10, how to construct an NFA N recognizing the language $L(R_1) \circ L(R_2)$ (regular languages are closed under concatenation).

QQ: 749389476

<https://tutorcs.com>

3. We have $o_*(R_1) = (R_1^*) \circ N_1 = L(R_1)^*$. By the induction hypothesis, there exists an NFA N_1 recognizing $L(R_1)$. We have seen in Lecture 10, how to construct an NFA N recognizing the language $L(R_1)^*$ (regular languages are closed under the star-operation).

Example: transforming a regular expression R into an NFA recognizing $L(R)$

程序代写代做 CS编程辅导

Consider regular expression $R = (ab \cup a)^*$. We develop an automaton N recognizing $L(R)$ following construction sequence of R :



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Construction sequence:

- 1 a
- 2 b
- $\rightarrow (a \cdot b)$
- $\rightarrow ((a \cdot b) \cup a)$
- $\rightarrow ((a \cdot b) \cup a)^*$

Example: transform $(ab \cup a)^*$ into an NFA recognizing $L(R)$

Consider regular expression $(ab \cup a)^*$. We develop an automaton N recognizing $L(R)$:

- ①
- ②
- ③
- ④
- ⑤

QQ: 749389476

<https://tutorcs.com>

程序代写代做 CS编程辅导



Non-regular Languages

WeChat: cstutorcs

Reading: ITC Section 1.4
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Proving a language is not regular

程序代写代做 CS编程辅导

We have seen **multiple techniques** to prove that a **language L** over some alphabet Σ **is not regular** now:



- It **suffices** to prove that there exists a **DFA** that recognizes (accepts) L .

WeChat: cstutorcs

- It **suffices** to prove that there exists an **NFA** that recognizes (accepts) L .

Email: tutorcs@163.com

- It **suffices** to prove that there exists a **regular expression** that is interpreted as L .

QQ: 749389476

<https://tutorcs.com>

To prove that some language L is not regular, we need to prove that none of these is possible.

Proving a language is not regular

程序代写代做 CS编程辅导

Here is a general technique showing that some object t is not a member of some set



showing that some object t is not a

- Prove that all members of S have some property P .
WeChat: cstutorcs
- Prove that t doesn't have property P
Assignment Project Exam Help
- Conclude that $t \notin S$
Email: tutorcs@163.com

QQ: 749389476

We will next state (and prove) a property that all regular languages have.
<https://tutorcs.com>

The pumping lemma

程序代写代做 CS编程辅导

Pumping lemma

Let A be a regular language. Then there exists a number p (the pumping length) with the following property:



For every word $w \in A$ of length at least p , there exists three words x, y, z with:

1. $w = xyz$

Assignment Project Exam Help

2. for every $i \geq 0$ we have $xy^iz \in A$

Email: tutorcs@163.com

3. $|y| > 0$, and

QQ: 749389476

4. $|xy| \leq p$.

<https://tutorcs.com>

$\Sigma = \{a, b\}$

$\{(aabb)^*\}$

$= \{w \in \Sigma^* |$

$|w| \text{ divisible}$

$\text{by } 3\}$

ababab
 x
 y
 z

abababab

ababababab

The pumping

程序代写代做 CS 编程辅导

House Select Text Draw Stamp Spotlight Eraser Paint Undo Redo Clear Save



Pumping le

Let A be a language. Then there exists a number p (the pumping length) such that the following property:

For every word $w \in A$ of length at least p , there exists three words x, y, z with:

1. $w = xyz$

2. for every $i \geq 0$ we have $xy^i z \in A$

3. $|y| > 0$ and

Email: tutorcs@163.com

4. $|xy| \leq p$.

QQ: 749389476

<https://tutorcs.com>