# COMP30026 Models of Computation

## Regular Expressions and Non-Regular Languages

Assignment Project Exam Help

https://eduassistpro.github.i

Lecture Week 8 Part

Add WeChat edu_assist_pr

Semester 2, 2020

# Regular Expressions

Regular expressions is a notation for languages.

Assignment Project Exam Help

You are probably familiar with similar notation in Unix, Python or JavaScr ifferent things to d

https://eduassistpro.github.i

**Example:**
$(0 \cup 1)(0 \cup 1)(0 \cup 1)((0 \cup 1)(0 \cup 1)(0$
non-empty strings with the lengths that are multip

Add WeChat edu_assist_pr

The star binds tighter than concatenation, which in turn binds tighter than union.

**Syntax:**

The regular expressions over an alphabet $\Sigma = \{a_1, \ldots, a_n\}$ are given by the grammar

Assignment Project Exam Help

*regex*

*regexp*$^*$

https://eduassistpro.github.i

**Semantics:**

$$
\begin{array}{lcl}
L(a) & = & \{a\} \\
L(\epsilon) & = & \{\epsilon\} \\
L(\emptyset) & = & \emptyset \\
L(R_1 \cup R_2) & = & L(R_1) \cup L(R_2) \\
L(R_1 \; R_2) & = & L(R_1) \circ L(R_2) \\
L(R^*) & = & L(R)^*
\end{array}
$$

Add WeChat edu_assist_pr

Assignment Project Exam Help

$$\epsilon \quad : \quad \{\epsilon\}$$

https://eduassistpro.github.i

$$(0 \cup \epsilon)(\epsilon \cup 1) \quad : \quad \{\epsilon, 0, 1$$

$1^*$ : all finite seque

Add WeChat edu_assist_pr

$\epsilon \cup 1 \cup (\epsilon \cup 1)^*(\epsilon \cup 1)$ : all finite sequence

$$(1^*0^*)^* \quad : \quad ?$$

# Regular Expressions vs Automata

**Theorem:** $L$ is regular iff $L$ can be described by a regular expression.

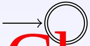
Assignment Project Exam Help

Let us first show the 'if' direction, by showing how to convert a regular expression $R$ into an NFA that recognises $L(R)$.

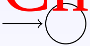
The proo i

https://eduassistpro.github.i

Case $R$ 

Case $R = \epsilon$: Construct 

Add WeChat edu_assist_pr

Case $R = \emptyset$: Construct 

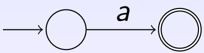
Case $R = R_1 \cup R_2$, $R = R_1\ R_2$, or $R = R_1^*$:
We already gave the constructions when we showed that regular languages were closed under the regular operations.

# NFAs from Regular Expressions

Let us construct, in the proposed systematic way, an NFA for $(\mathtt{a} \cup \mathtt{b})^*\mathtt{bc}$.

Assignment Project Exam Help

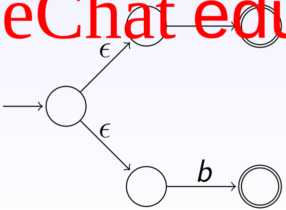
Start from innermost expressions and work out:




https://eduassistpro.github.i

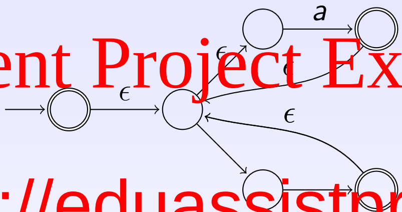
So $a \cup b$ yields:

Add WeChat edu_assist_pr



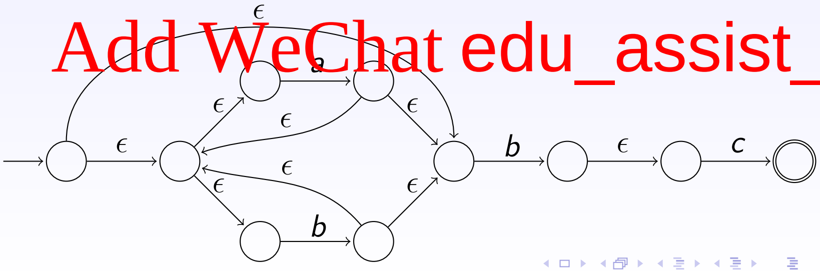
# NFAs from Regular Expressions

Then $(\mathtt{a} \cup \mathtt{b})^*$ yields:




Assignment Project Exam Help
https://eduassistpro.github.i
Add WeChat edu_assist_pr

Finally (

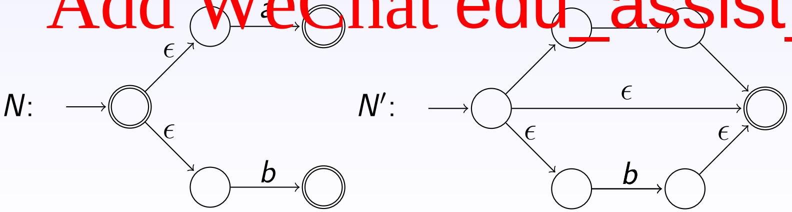


# Regular Expressions from NFAs

We now show the 'only if' direction of the theorem.

Note that, given an NFA $N$, we can easily build an equivalent NFA with at most one accept state. We transform $N = (Q, \Sigma, \delta, q_0, F)$ to $N' = (Q \cup \{q_f\}, \Sigma, \delta', q_0, \{q_f\})$ by adding a new $q_f$, with $\epsilon$ transitions to $q_f$ fr

Assignment Project Exam Help
https://eduassistpro.github.i
Add WeChat edu_assist_pr



# Regular Expressions from NFAs

We sketch how an NFA can be turned into a regular expression in a systematic process of "state elimination".

Assignment Project Exam Help

In the process, arcs get labelled with regular expressions.

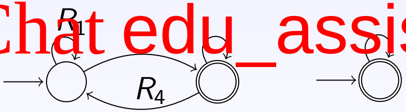
Start by m

https://eduassistpro.github.i

Repeate pt states.

Add WeChat edu_assist_pr

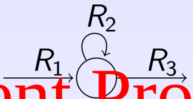
The process produces either




We get $(R_1 \cup R_2 R_3^* R_4)^* R_2 R_3^*$ in the first case; $R^*$ in the second.

Note that some $R$s may be $\epsilon$ or $\emptyset$.

# The State Elimination Process

Consider a node 

Assignment Project Exam Help

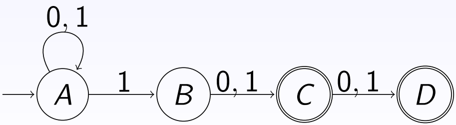
Any such pair of incoming/outgoing arcs get replaced by a single arc that by 3.

https://eduassistpro.github.i

If there are ed by $m \times n$ bypassing arcs when the node is remov

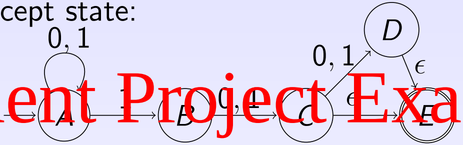
Add WeChat edu_assist_pr

Let us illustrate the process on this example:



# State Elimination Example

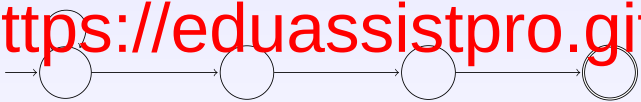
Create a single accept state:




Assignment Project Exam Help

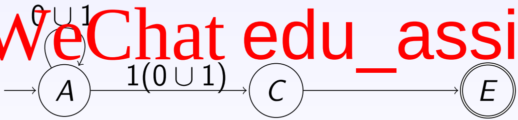
Elimina

https://eduassistpro.github.i

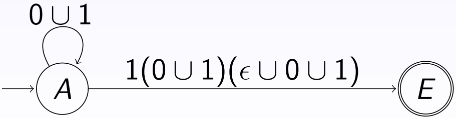


Add WeChat edu_assist_pr

Now eliminate $B$:



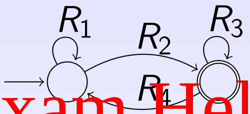

and then $C$:



# State Elimination Example

Note that


is




Assignment Project Exam Help

with

- $R_1$
- $R_2$
- $R_3 = R_4 = \emptyset$

https://eduassistpro.github.i

Add WeChat edu_assist_pr

Hence the instance of the general "recipe" ( $_3^*$ is

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

Sipser (see "Readings Online" on Canvas) provides more details of this kind of translation.

# Some Useful Laws for Regular Expressions

$A \cup A = A$

Assignment Project Exam Help

$A \cup B = B \cup A$

$(A \cup B)$

https://eduassistpro.github.i

$(A\, B)$

$\emptyset \cup A = A \cup \emptyset = A$

Add WeChat edu_assist_pr

$\epsilon\, A = A\, \epsilon = A$

$\emptyset\, A = A\, \emptyset = \emptyset$

# More Useful Laws for Regular Expressions

$(A \cup B)\,C = A\,C \cup B\,C$

Assignment Project Exam Help

$A\,(B \cup$

$(A^*)^* =$

https://eduassistpro.github.i

$\emptyset^* = \epsilon^* = \epsilon$

$(\epsilon \cup A)^* = A^*$

Add WeChat edu_assist_pr

$(A \cup B)^* = (A^* B^*)^*$

# Limitations of Finite-State Automata

Assignment Project Exam Help

Consider the language

$$\{0^n1^n \mid n \geq 0\} = \{\epsilon, 01, 0011, 000111, \ldots\}$$

https://eduassistpro.github.i

Intuitive , because
a DFA has n

**Exercise:** Is the language $L_1 = \{0^n1^n \mid$ egular?

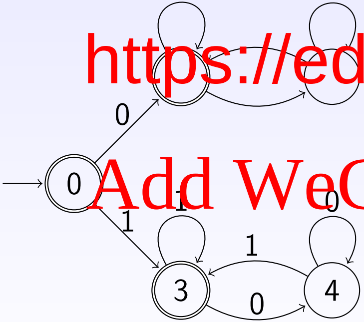
Add WeChat edu_assist_pr

What about $L_2 = \left\{ w \;\middle|\; \begin{array}{l} w \text{ has an equal nu} \\ \text{of the substrings 01 and 10} \end{array} \right\}$?

# Language $L_2$

$$L_2 = \left\{ w \;\middle|\; \begin{array}{l} w \text{ has an equal number of occurrences} \\ \text{of the substrings 01 and 10} \end{array} \right\} ?$$

Assignment Project Exam Help




https://eduassistpro.github.i
Add WeChat edu_assist_pr

# The Pumping Lemma for Regular Languages

This is the standard tool for proving languages non-regular.

Loosely, it says that if we have a regular language $A$ and consider a sufficiently long string $s \in A$, then a recogniser for $A$ must traverse some loop to accept $s$. So $A$ must contain infinitely many strings exhibiti

Assignment Project Exam Help

**Pumpi** $p$ such that for any string $s \in A$ with $|s| \geq p$, $xyz$, satisfying

https://eduassistpro.github.i

Add WeChat edu_assist_pr

1. $xy^iz \in A$ for all $i \geq 0$
2. $y \neq \epsilon$
3. $|xy| \leq p$

We call $p$ the pumping length.

# Proving the Pumping Lemma

Let DFA $M = (Q, \Sigma, \delta, q_0, F)$ recognise $A$.

Let $p = |Q|$ and consider $s$ with $|s| \geq p$.

In an accepting run for $s$,
some state must be re-visited.

Let $q_i$ b

At the first v

consum

(strictly l

suggests a way of splitting $s$
into $x$, $y$ and $z$ such that
$xz, xyz, xyyz, \ldots$ are all in $A$.


Assignment Project Exam Help
https://eduassistpro.github.i
Add WeChat edu_assist_pr

Notice that $y \neq \epsilon$. Also, if input consumed has length $k$ then the number of state visits is $k + 1$. Let $m + 1$ be the number of state visits when reading $xy$, then $|xy| = m \leq p$. Notice that $m \leq p$, because $m + 1$ is the number of state visits with only one repetition.

# Using the Pumping Lemma

The pumping lemma says:

Assignment Project Exam Help

$s$ can be written

https://eduassistpro.github.i

We can us ... egular:

$$\forall p \exists s \in A : \begin{cases} s \text{ can't be written} \\ xyz \text{ such that} \ldots \end{cases}$$

Add WeChat edu_assist_pr

Coming up with such an $s$ is sometimes easy, sometimes difficult.

# Pumping Example 1

We show that $B = \{0^n 1^n \mid n \geq 0\}$ is not regular.

Assignment Project Exam Help

Assume it is, and let $p$ be the pumping length.

Conside

https://eduassistpro.github.i

By the pu $i \geq 0$.

But $y$ cannot consist of all 0s, since $xy$

Add WeChat edu_assist_pr

Similarly $y$ cannot consist of all 1s. And if
one 1, then some 1 comes before some 0 in

So we inevitably arrive at a contradiction if we assume that $B$ is regular.

# Pumping Example 2

$C = \{w \mid w$ has an equal number of 0s and 1s$\}$ is not regular.

Assignment Project Exam Help

Assume it is, and let $p$ be the pumping length.

Conside $^{p}$ $^{p}$

https://eduassistpro.github.i

By the pu $i \geq 0$,
$y \neq \epsilon$, an

But then $xyyz \notin C$, a contradiction.

Add WeChat edu_assist_pr

# Pumping Example 2

$C = \{w \mid w$ has an equal number of 0s and 1s$\}$ is not regular.

Assignment Project Exam Help

Assume it is, and let $p$ be the pumping length.

Conside $\quad ^p \; ^p$

https://eduassistpro.github.i

By the pu $\quad i \geq 0$,
$y \neq \epsilon$, an

But then $xyyz \notin C$, a contradiction.

Add WeChat edu_assist_pr

A simpler alternative proof: If $C$ were regular then also $B$ from before would be regular, since $B = C \cap 0^*1^*$ and regular languages are closed under intersection.

# Pumping Example 3

We show that $D = \{ww \mid w \in \{0,1\}^*\}$ is not regular.

Assignment Project Exam Help

Assume it is, and let $p$ be the pumping length.

Conside

https://eduassistpro.github.i

By the pu ll $i \geq 0$, $y \neq \epsilon$, and $|xy| \leq p$.

Add WeChat edu_assist_pr

Since $|xy| \leq p$, $y$ consists entirely of 0s.

But then $xyyz \notin D$, a contradiction.

# Example 4 – Pumping Down

We show that $E = \{0^i 1^j \mid i > j\}$ is not regular.

Assignment Project Exam Help

Assume it is, and let $p$ be the pumping length.

Conside

https://eduassistpro.github.i

By the pu ll $i \geq 0$, $y \neq \epsilon$, and $|xy| \leq p$.

Add WeChat edu_assist_pr

Since $|xy| \leq p$, $y$ consists entirely of 0s.

But then $xz \notin E$, a contradiction.