COMP30026 Models of Computation

Regular Languages

Assignment Project Exam Help

https://powcoder.com

Anna Kalenkova

Lecture 15

Add WeChat powcoder

Semester 2, 2020

This Lecture is Being Recorded

DFAs vs NFAs

Assignment Project Exam Help

The class of languages recognised by NFAs is exactly the class of regular languages.

Theorem: Every NFA has an equivalent DFA.

https://powcoder.com

The proof rests on the so-called subset construction.

Given NFA N, we construct DFA M, each of whose states corresponds to a set of N-states.

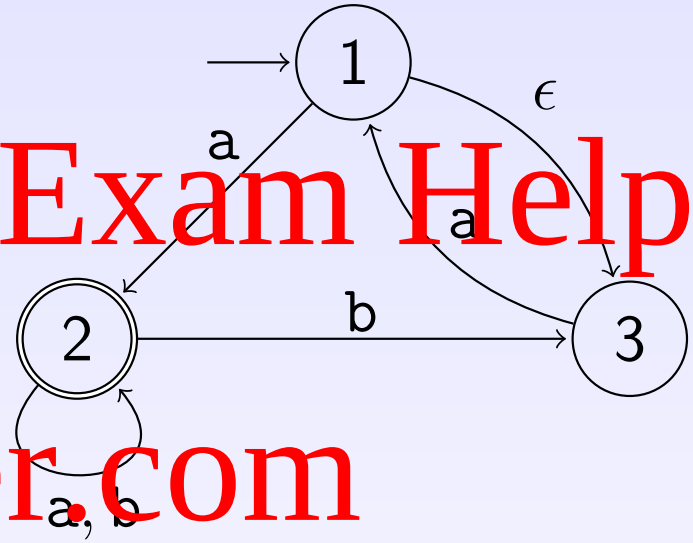
Add WeChat powcoder

If N has k states then M may have up to 2^k states (but it will often have far fewer than that).

DFAs vs NFAs

Assignment Project Exam Help

Consider the NFA on the right. We can systematically construct an equivalent DFA from the NFA.



https://powcoder.com

The DFA's start state is $\{1, 3\}$.
From $\{1, 3\}$ a takes us to $\{1, 2, 3\}$.
From $\{1, 2, 3\}$, a takes us back to $\{1, 2, 3\}$, b takes us to $\{2, 3\}$.

Add WeChat powcoder

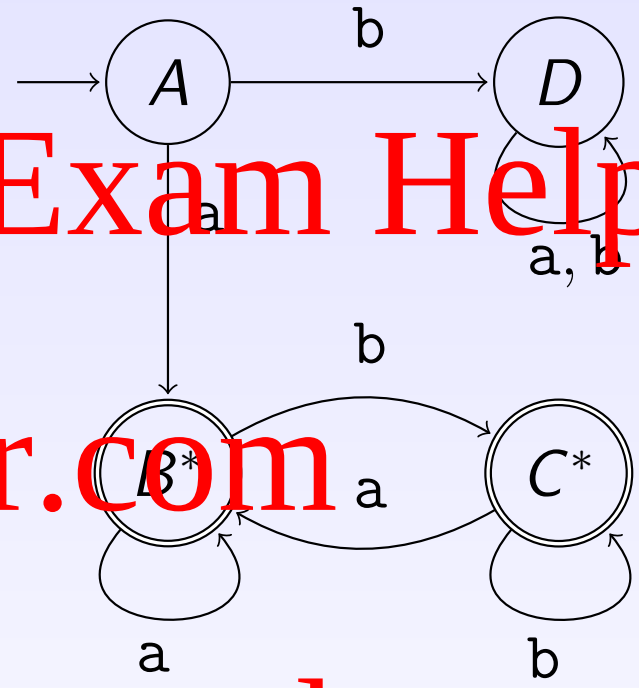
Any state S which contains an accept state from the NFA will be an accept state for the DFA. Here we mark accept states with a star.

	a	b
A=$\{1,3\}$	B*	–
B*=$\{1,2,3\}$	B*	C*
C*=$\{2,3\}$	B*	C*

DFAs vs NFAs

Assignment Project Exam Help

Here is the equivalent DFA that we derive.



https://powcoder.com

Any state S which contains an accept state from the NFA (in this case the NFA has just one, namely state 2) becomes an accept state for the DFA.

Add WeChat powcoder

We add (dead) state D that corresponds to the empty set.

	a	b
A=$\{1,3\}$	B*	D
B*=$\{1,2,3\}$	B*	C*
C*=$\{2,3\}$	B*	C*
D=$\emptyset$	D	D

More Formally . . .

Let $N = (Q, \Sigma, \delta, q_0, F)$. Let $\xrightarrow{*}_\epsilon$ be the reflexive transitive closure of $\rightarrow_\epsilon$, which in turn is defined by $s \rightarrow_\epsilon s'$ iff $s' \in \delta(s, \epsilon)$.

Assignment Project Exam Help

Let $E(S)$ be the 'ϵ-closure' of $S \subseteq Q$, that is, S together with all states reachable from states in S, using only ϵ steps:

https://powcoder.com

$$E(S) = \bigcup_{s \in S} \{s' \in Q \mid s \xrightarrow{*}_\epsilon s'\}$$

We construct $M = (\mathcal{P}(Q), \Sigma, \delta', q_0', F')$ as follows:

Add WeChat powcoder

- $q_0' = E(\{q_0\})$.
- $\delta'(S, v) = \bigcup_{s \in S} E(\delta(s, v))$.
- $F' = \{S \subseteq Q \mid S \cap F \neq \emptyset\}$.

Note: This construction may include unreachable states.

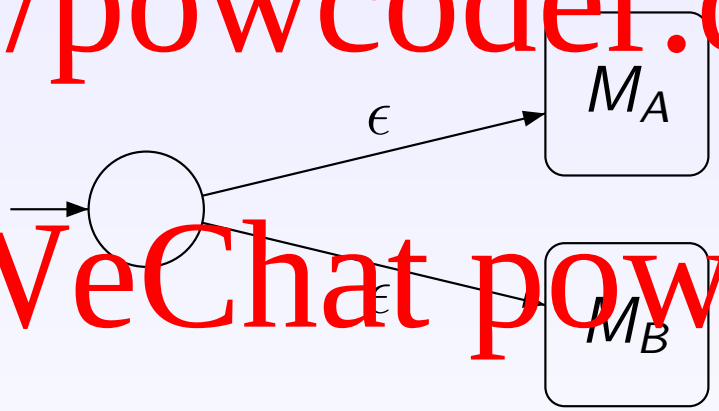
Closure Results for Regular Languages

Theorem: The class of regular languages is closed under union.

Assignment Project Exam Help

Proof: Let A and B be regular languages, with recognisers M_A and M_B. An NFA that recognises $A \cup B$ is easily constructed:

https://powcoder.com



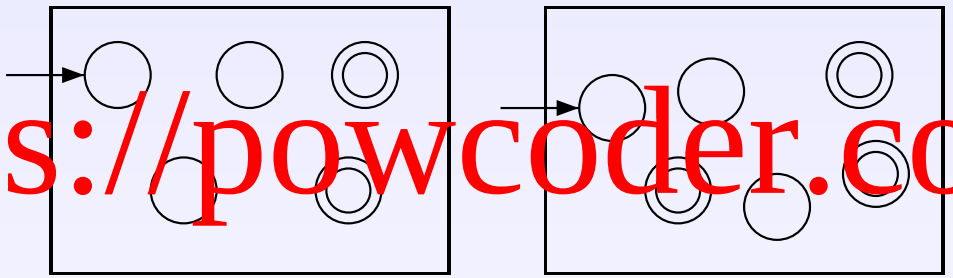
Add WeChat powcoder

The ϵ-transitions go to the start states of M_A and M_B.

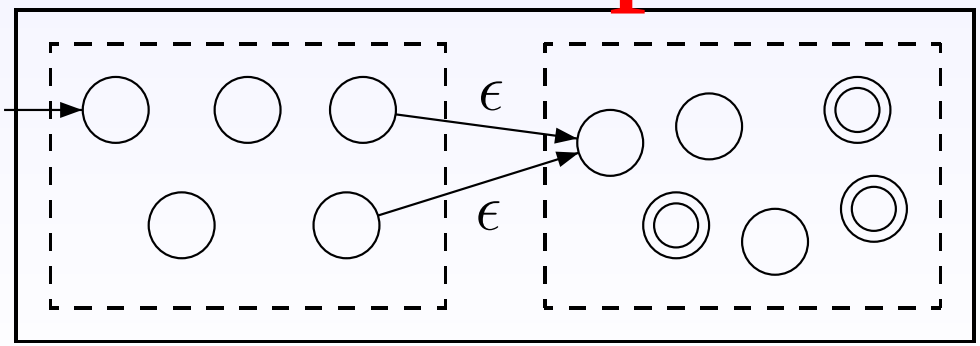
Closure Results for Regular Languages

Theorem: The class of regular languages is closed under $\circ$.

Proof: Let A and B be regular languages with these recognisers:



From these we can easily construct an NFA that recognises $A \circ B$:



Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder

The Last Construction, Formally

Assignment Project Exam Help

Let recognisers for A and B be these DFAs, respectively:

- $M_A = (Q, \Sigma, \delta, q_0, F)$
- $M_B = (Q', \Sigma, \delta', q_0', F')$

https://powcoder.com

Then a recogniser for $A \circ B$ is the NFA $(Q \cup Q', \Sigma, \delta'', q_0, F')$, where

Add WeChat powcoder

$$\delta''(q, v) = \begin{cases} \{\delta'(q, v)\} & \text{if } q \in Q' \text{ and } v \in \Sigma \\ \{\delta(q, v)\} & \text{if } q \in Q \text{ and } v \in \Sigma \\ \{q_0'\} & \text{if } q \in F \text{ and } v = \epsilon \end{cases}$$

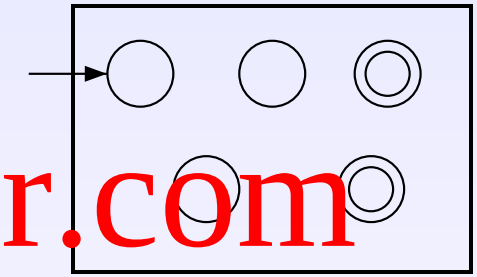
Closure Results for Regular Languages

Assignment Project Exam Help

Theorem: The class of regular languages is closed under Kleene star.

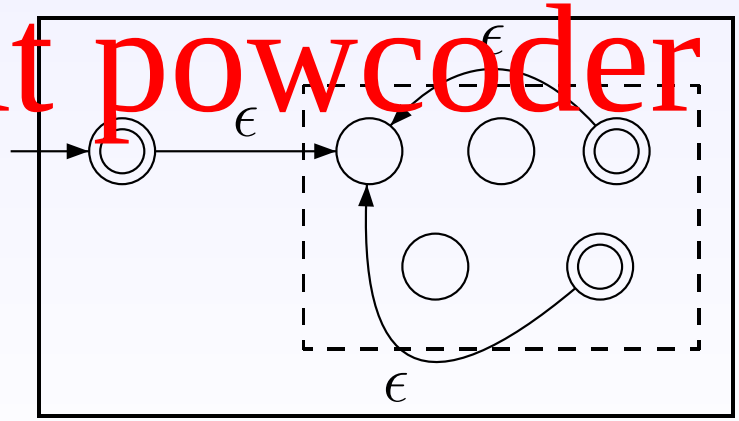
https://powcoder.com

Proof: Let A be a regular language with the recogniser shown on the right.



Add WeChat powcoder

Here is how we construct an NFA to recognise A^*:



Closure Results for Regular Languages

Assignment Project Exam Help

Regular languages have several other closure properties.

They are closed under

https://powcoder.com

- intersection,
- complement, A^c
- difference (this follows, as $A \setminus B = A \cap B^c$)
- reversal.

Add WeChat powcoder

Algorithms for Manipulating Automata

Assignment Project Exam Help

For some of these closure results, we will use the tutorials to develop useful DFA manipulation algorithms.

https://powcoder.com

You will see, for example, how to systematically build DFAs for languages $A \cap B$, out of DFAs for A and B.

Add WeChat powcoder

Equivalence of DFAs

Assignment Project Exam Help

We can always find a minimal DFA for a given regular language (by minimal we mean having the smallest possible number of states).

https://powcoder.com

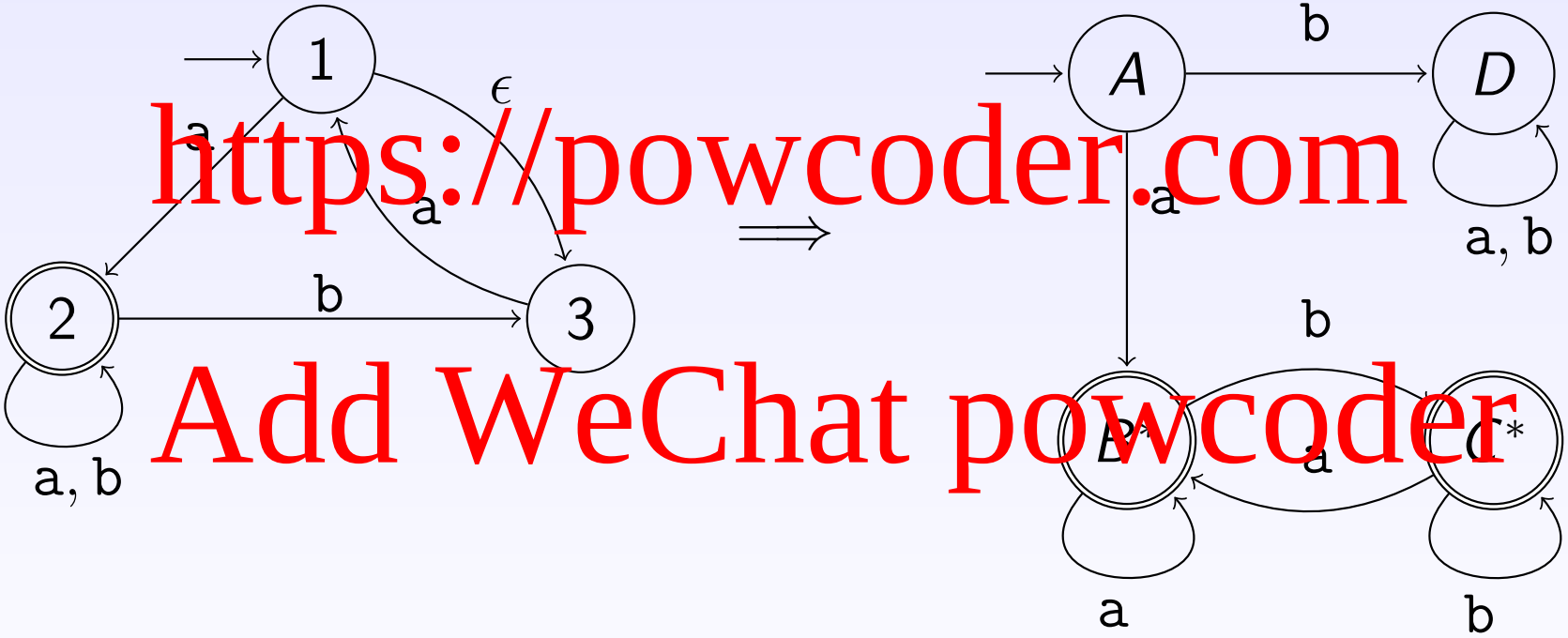
Since a DFA has a unique start state and the transition function is total and deterministic, we can test two DFAs for equivalence by minimizing them.

Add WeChat powcoder

Minimizing DFAs

There is no guarantee that DFAs that are produced by the various algorithms, such as the subset construction method, will be minimal.

Assignment Project Exam Help
https://powcoder.com
Add WeChat powcoder



$A = \{1, 3\}$, $B^* = \{1, 2, 3\}$, $C^* = \{2, 3\}$, and $D = \emptyset$.

Generating a Minimal DFA

The following algorithm takes an NFA and produces an equivalent minimal DFA. Of course the input can also be a DFA.

Assignment Project Exam Help

1. Reverse the NFA;
2. Determinize the result;
3. Reverse again;
4. Determinize.

https://powcoder.com

To reverse an NFA A with initial state q_0 and accept states $F \neq \emptyset$:

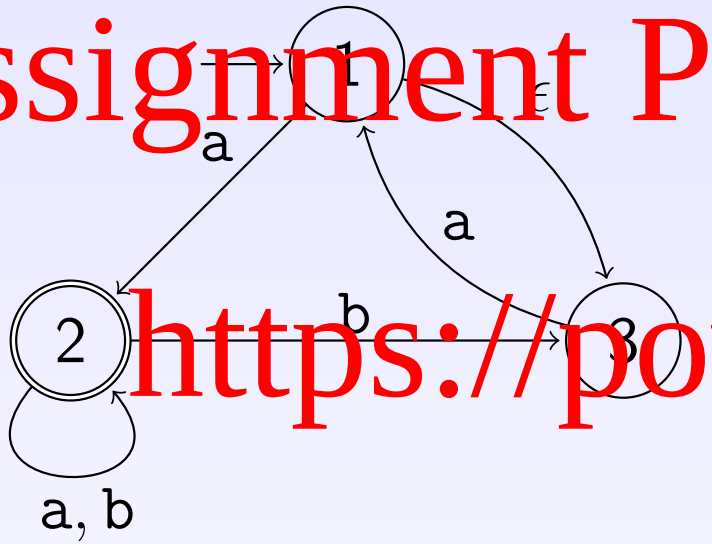
Add WeChat powcoder

1. If $F = \{q\}$, let q be the accept state.
2. Otherwise add a new state q which becomes the only accept state; then, for each state in F, add an ϵ transition to q.
3. Reverse every transition in the resulting NFA, making q the initial state and q_0 the (only) accept state.

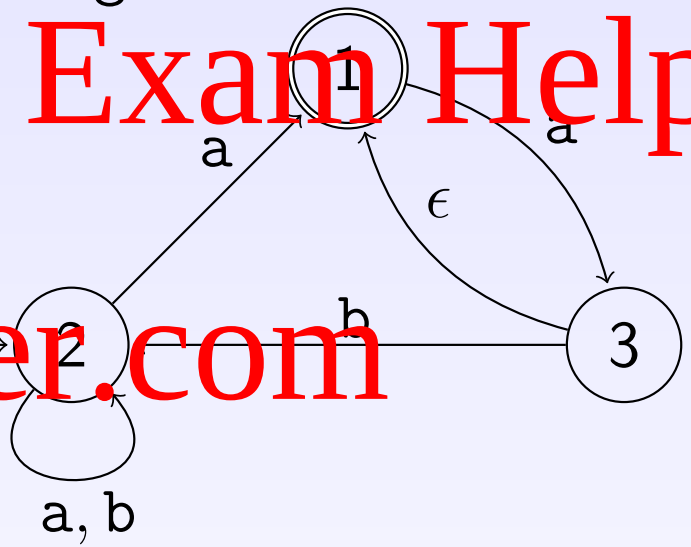
Minimization Example

Consider the NFA that we determinized two slides ago.
Here it is on the left, with its reversal on the right:

Assignment Project Exam Help

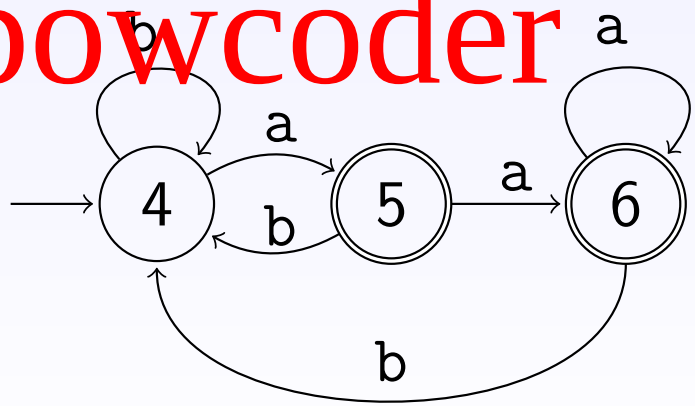


https://powcoder.com



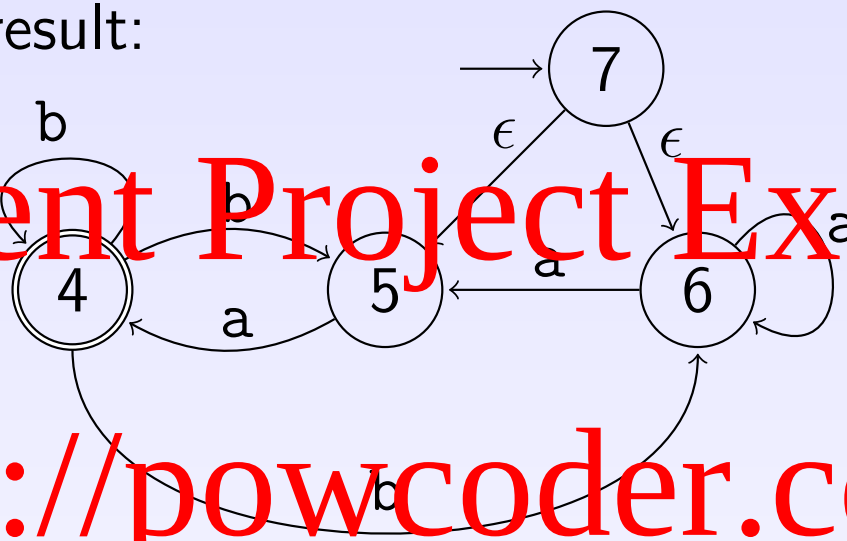
Add WeChat powcoder

Making the reversed NFA deterministic.
We renamed the states to avoid later confusion;
4 corresponds to $\{2\}$, 5 to $\{1, 2\}$, and 6 to $\{1, 2, 3\}$.



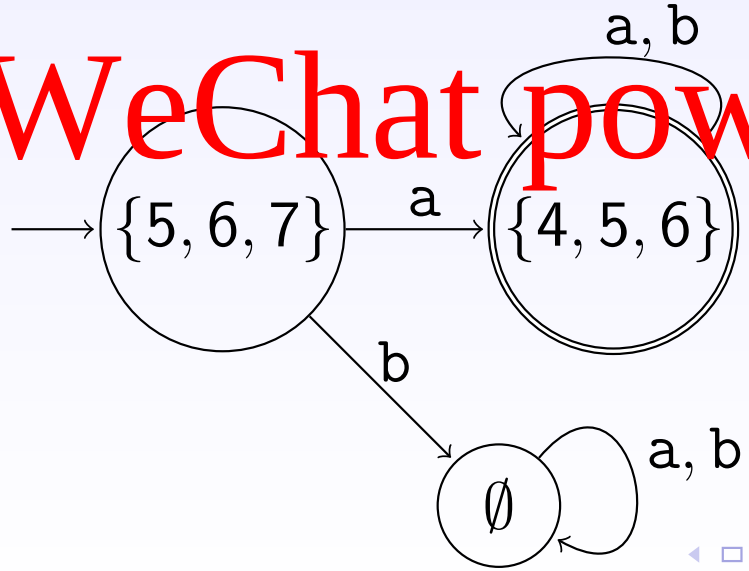
Minimization Example

Now reversing the result:



Assignment Project Exam Help
https://powcoder.com

And finally making the result deterministic:



Add WeChat powcoder

Next Week

Assignment Project Exam Help

We next look at regular expressions—another way to capture the regular languages.

https://powcoder.com

Also, we will use a technique called "pumping", for proving that a language is not regular and introduce context-free languages.

Add WeChat powcoder