CS:314 Fall 2024

Section **04**
Recitation **2**

chris.tu@rutgers.edu
Office hours: 2-3pm Thursday CoRE 335

Topics for today

- NFA to DFA conversion
- DFA minimization: Hopcroft Reduction
- DFA to RegEx

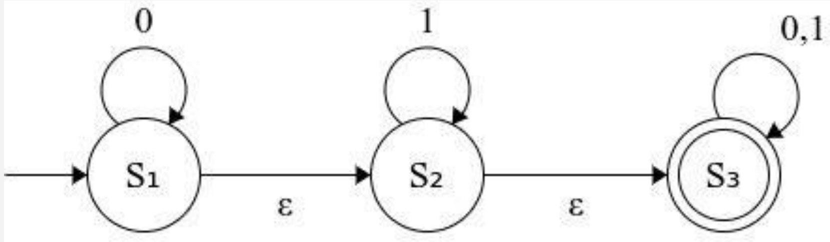
Convert NFA to DFA

- NFA can be reduced to DFA
  - Subset construction

- Epsilon closure
  - Epsilon closure for a given state X is a set of states which can be reached from the states X with only “null” or “ε transitions” **including the state X itself.**
  - We use epsilon closure to construct subsets of states. **<u>DFA states are sets of NFA states.</u>**

Convert NFA to DFA: Epsilon Closure

- Epsilon closure
  - Epsilon closure for a given state X is a set of states which can be reached from the states X with only “null” or “ε transitions” **including the state X itself.**



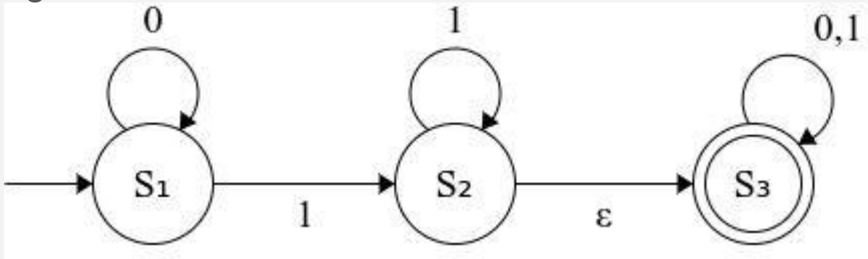
ε-closure$(S_1) = \{S_1, S_2, S_3\}$
ε-closure$(S_2) = \{S_2, S_3\}$

ε-closure$(S_3) = \{S_3\}$

ε-closure$(\{S_1, S_2\}) = \{S_1, S_2, S_3\}$

Convert NFA to DFA: Epsilon Closure

- Epsilon closure
  - Epsilon closure for a given state X is a set of states which can be reached from the states X with only “null” or “ε transitions” **including the state X itself.**
-



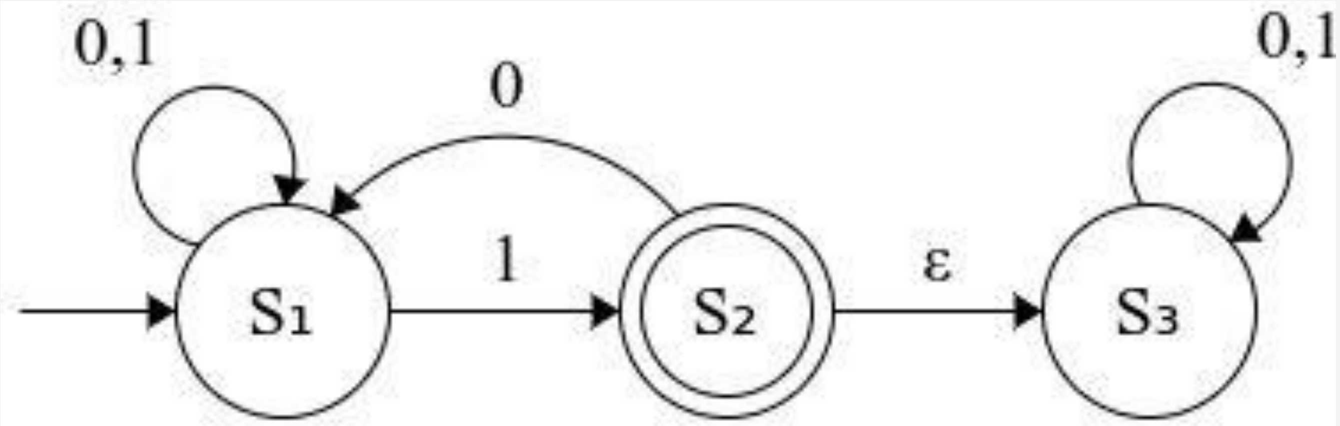
ε-closure(S_1) = {S_1}
ε-closure(S_2) = {S_2, S_3}

ε-closure(S_3) = {S_3}

ε-closure({S_1, S_2}) = {S_1, S_2, S_3}

Convert NFA to DFA: Subset Construction

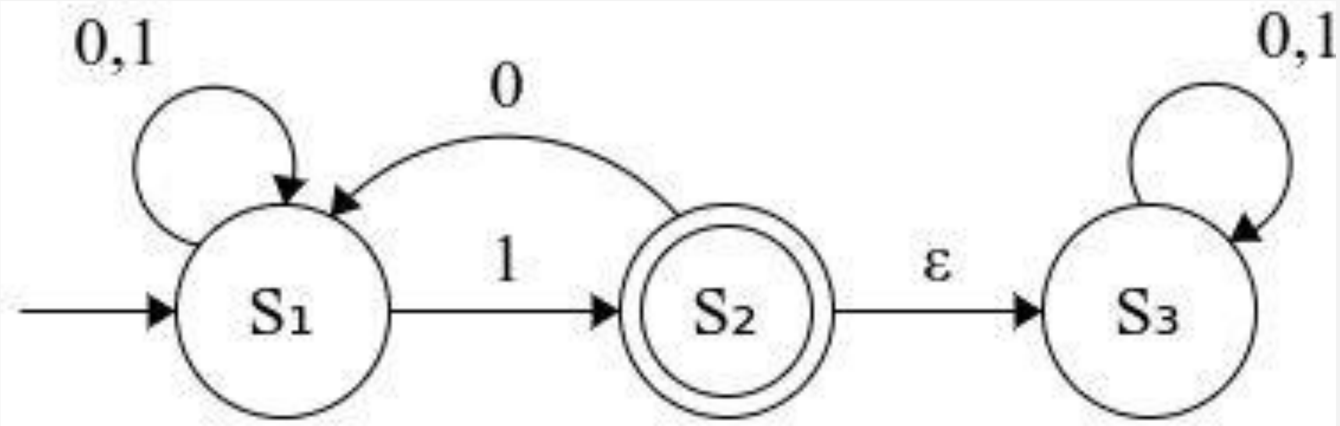
- Suppose we are given the following NFA.



Convert NFA to DFA: Subset Construction

- Suppose we are given the following NFA.

What is the start state? What are the final states?

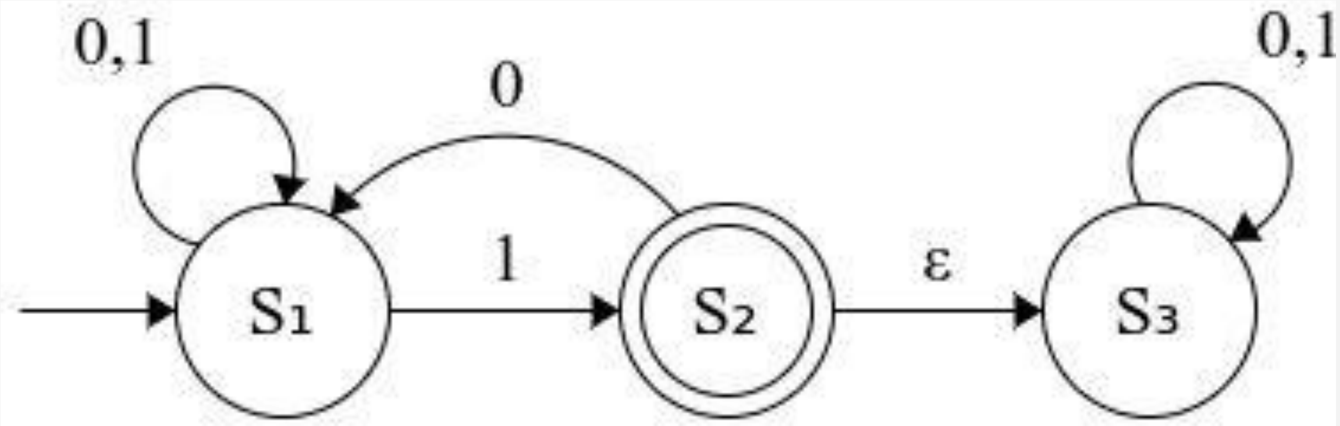


Convert NFA to DFA: Subset Construction

- Suppose we are given the following NFA.

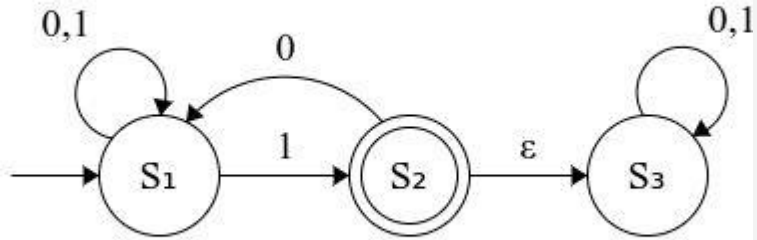
  Start state: S_1

  Final states: $\{S_2\}$



Convert NFA to DFA: Subset Construction

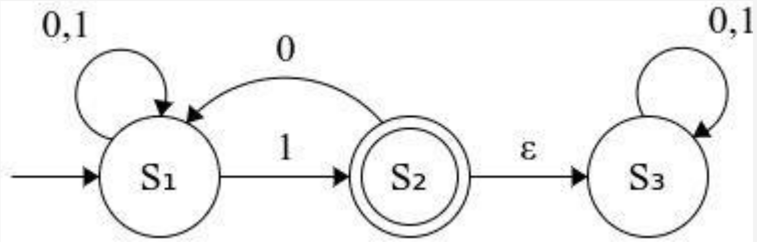
- Suppose we are given the following NFA.



What are the ε-closures?

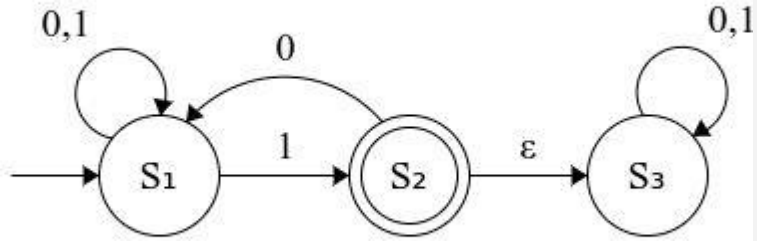
Convert NFA to DFA: Subset Construction

- Suppose we are given the following NFA.



ε-closure(S1) = {S1}
ε-closure(S2) = {S2, S3}
ε-closure(S3) = {S3}
ε-closure({S1, S2}) = {S1, S2, S3}
ε-closure({S2, S3}) = {S2, S3}
ε-closure({S1, S3}) = {S1, S3}
ε-closure({S1, S2, S3}) = {S1, S2, S3}

Convert NFA to DFA: Subset Construction



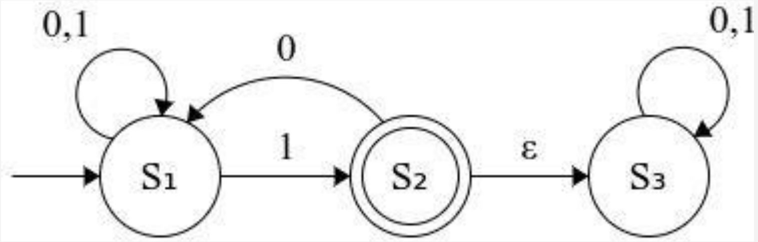
ε-closure(S1) = {S1}
ε-closure(S2) = {S2, S3}
ε-closure(S3) = {S3}
ε-closure({S1, S2}) = {S1, S2, S3}
ε-closure({S2, S3}) = {S2, S3}
ε-closure({S1, S3}) = {S1, S3}
ε-closure({S1, S2, S3}) = {S1, S2, S3}

The epsilon closure of the start state be the new initial state of the DFA

Since the start state of the NFA is S1, the epsilon closure is {S1}.

The start state of the DFA will be {S1}.

Convert NFA to DFA: Subset Construction

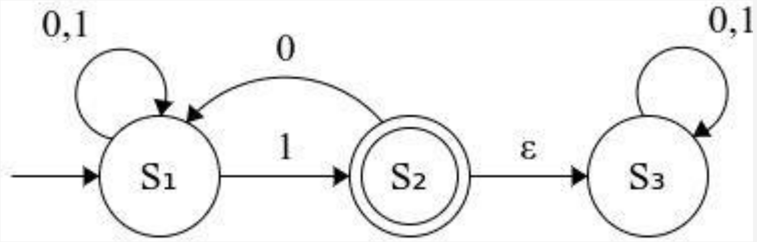


States	0	1
{S1}	{S1}	{S1, S2, S3}
{S1, S2, S3}		

Build your transition table.

- An input of 0 on S1 can only transition to S1. ε-closure(S1) = {S1}
  - Therefore input 0 on {S1} = {S1}.
- An input of 1 on S1 can go to S1, S2. ε-closure({S1, S2}) = {S1, S2, S3}
  - Therefore input 1 on {S1} = {S1,S2,S3}
  - {S1,S2,S3} is a new NFA set we found. List it as the next DFA state.

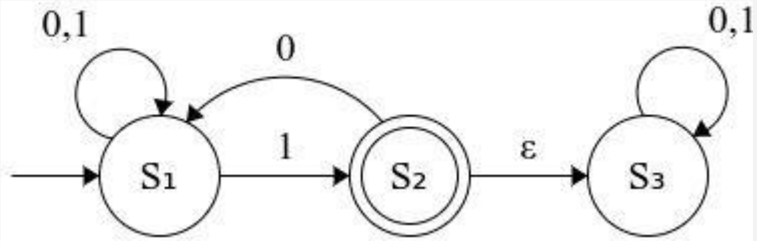
Convert NFA to DFA: Subset Construction



States	0	1
{S1}	{S1}	{S1, S2, S3}
{S1, S2, S3}	{S1, S3}	{S1, S2, S3}
{S1, S3}		

- Input 0 on {S1,S2,S3} can go to {S1,S3}.
  ε-closure({S1, S3}) = {S1, S3}
  - Therefore, input 0 on {S1,S2,S3} = {S1,S3}
- Input 1 on {S1,S2,S3} can go to {S1,S2,S3}.
  ε-closure({S1, S2, S3}) = {S1, S2, S3}
  - Therefore, input 1 on {S1,S2,S3} = {S1,S2,S3}
  - {S1,S3} is a new set we found, list it as the next DFA state

Convert NFA to DFA: Subset Construction

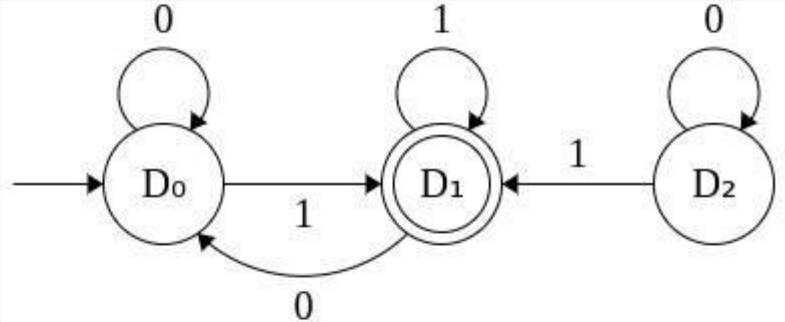


States	0	1
{S1}	{S1}	{S1, S2, S3}
{S1, S2, S3}	{S1, S3}	{S1, S2, S3}
{S1, S3}	{S1, S3}	{S1, S2, S3}

Convert NFA to DFA: Subset Construction

Given the new sets, we can draw the new DFA states.
Since S2 was the accepting state in the NFA, **every new state that contains S2 is now an accepting state.** Recall that D0 = {S1} is the new start state.

States	0	1
D0 = {S1}	D0 = {S1}	D1 = {S1, S2, S3}
D1 = {S1, S2, S3}	D2 = {S1, S3}	D1 = {S1, S2, S3}
D2 = {S1, S3}	D2 = {S1, S3}	D1 = {S1, S2, S3}



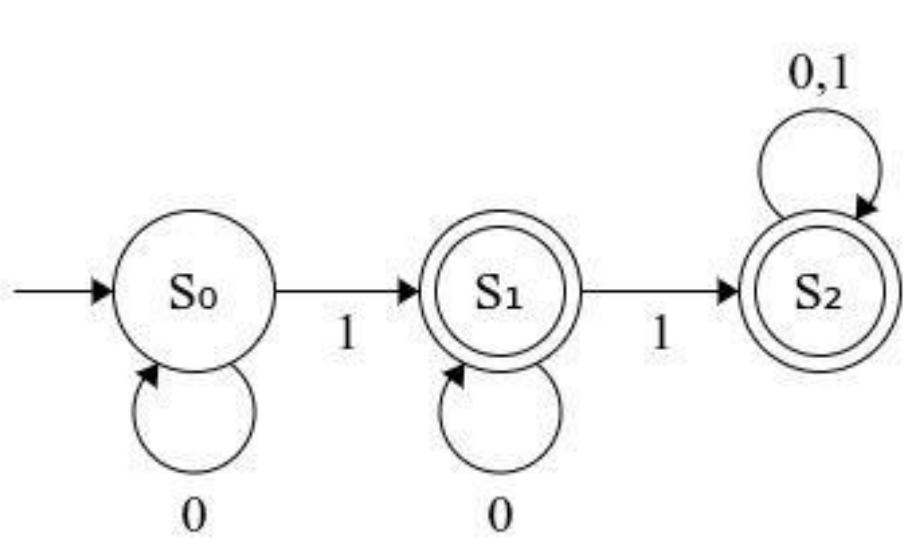
To Summarize NFA to DFA using Epsilon Closure/Subset:

1. Make start NFA state’s ε-closure the first DFA set.
2. For each input, make sets of NFA states reachable from the that DFA state.
3. In the transition table, use the ε-closure of these sets as the next DFA states.
4. Repeat Steps 2-3 until you don’t find any new distinct DFA states.
5. Label Each State set in your transition table, and use that to construct your DFA.

DFA Minimization: Hopcroft Reduction

- Construct TWO initial partitions
    - Divide states into accepting (final) & non-accepting (non-final) states.
- Iteratively split partitions (until partitions remain fixed)
    - Split a partition if the members in partition have transitions to different partitions for the same input.
    - Two states x, y belong in same partition if and only if for all symbols in Σ (the inputs) they transition to the same partition.
- Update transitions.

DFA Minimization: Hopcroft Reduction



- Initial partitions:
  - Accepting states: P1 = {S1, S2}
  - Non-Accepting states: P2 = {S0}

DFA Minimization: Hopcroft Reduction

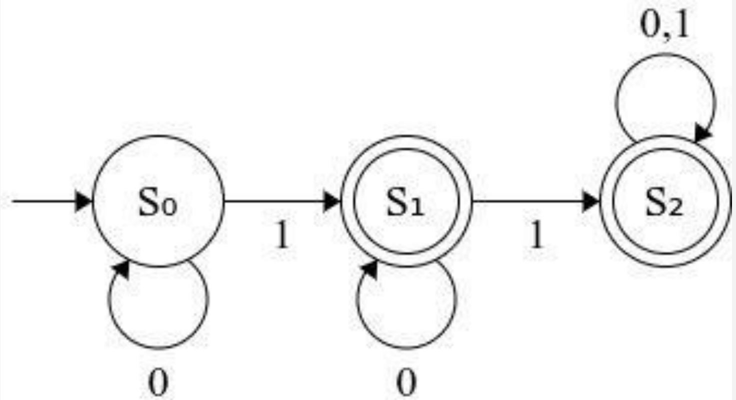
- Do we split partitions? P1 = {S1, S2}.

  (S1, 0) -> S1 in P1

  (S1, 1) -> S2 in P1

  (S2, 0) -> S2 in P1

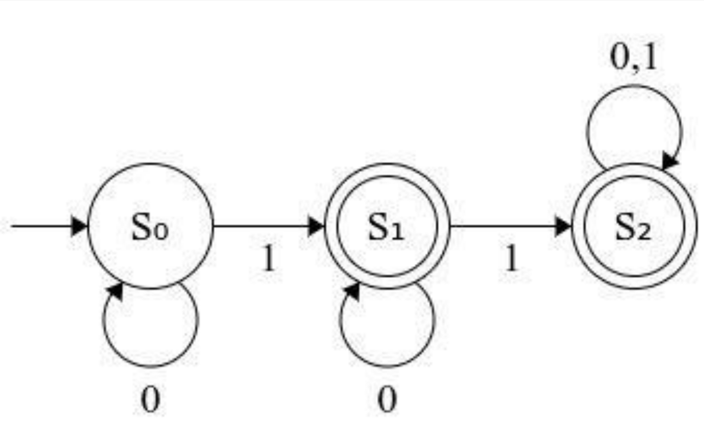
  (S2, 1) -> S2 in P1



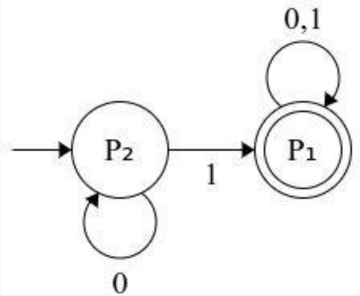
- Since every member of P1 has transitions in P1, we do NOT split the partition P1.

DFA Minimization: Hopcroft Reduction

Original
DFA

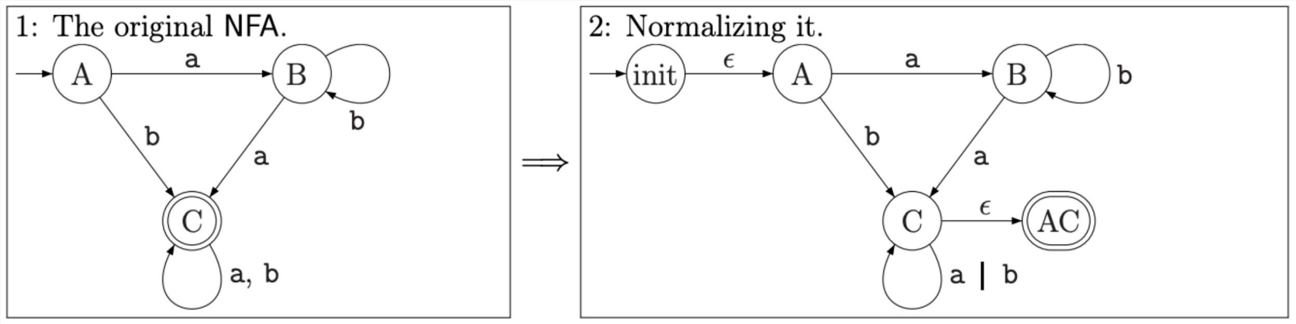


New DFA: since we do not split partitions, there’s two states: P1, P2



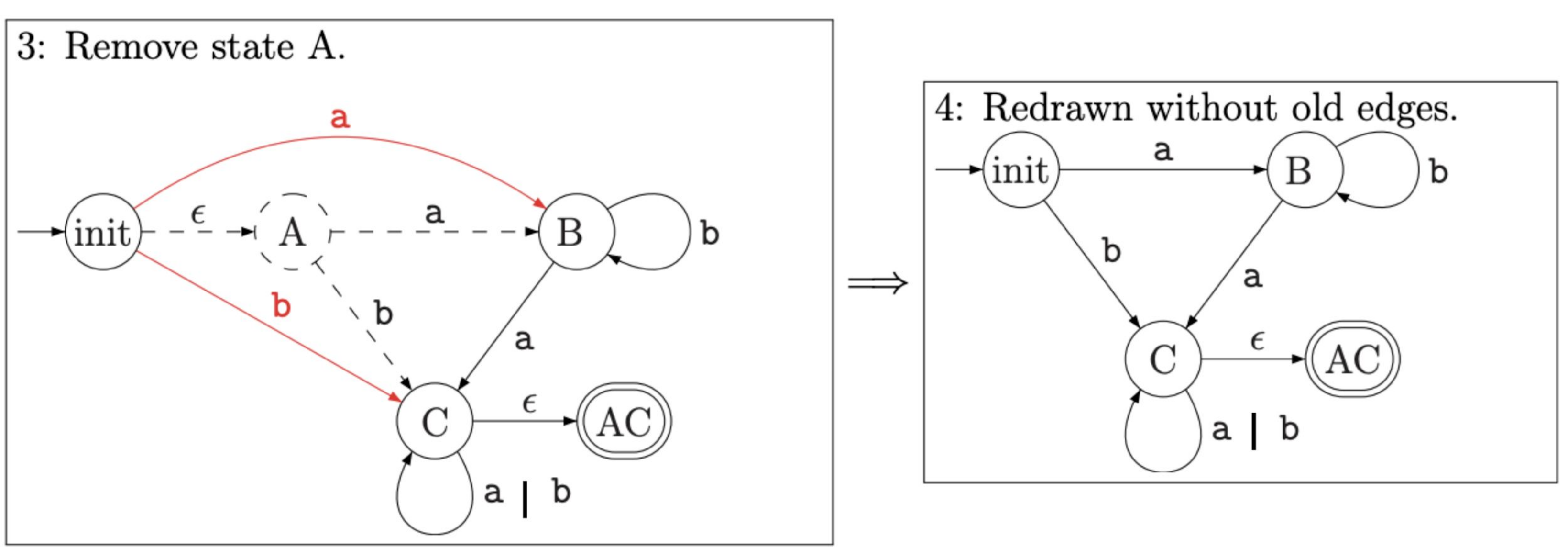
DFA to RE

1. Normalization: Add two new states that connect the original start state & final states with epsilon transition resp.



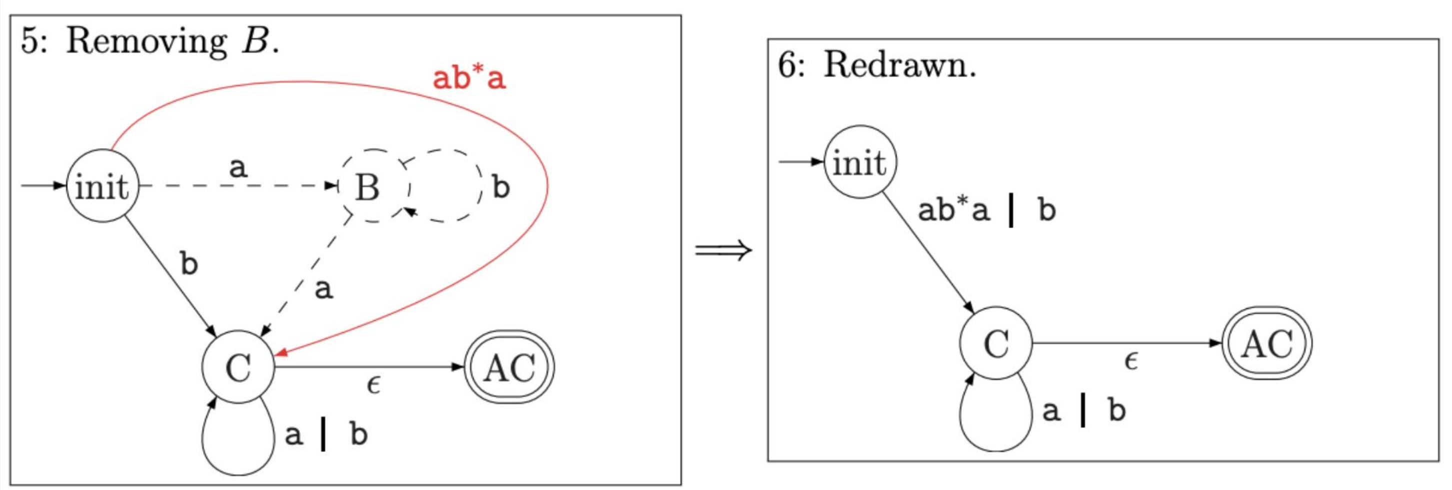
DFA to RE

1. Normalization: Add two new states that connect the original start state & final states with epsilon transition resp.
2. **Remove states one by one. You remove states that are in line with one edge, an[d] can be represented by a single regular expression transition.**
   - w/o loops



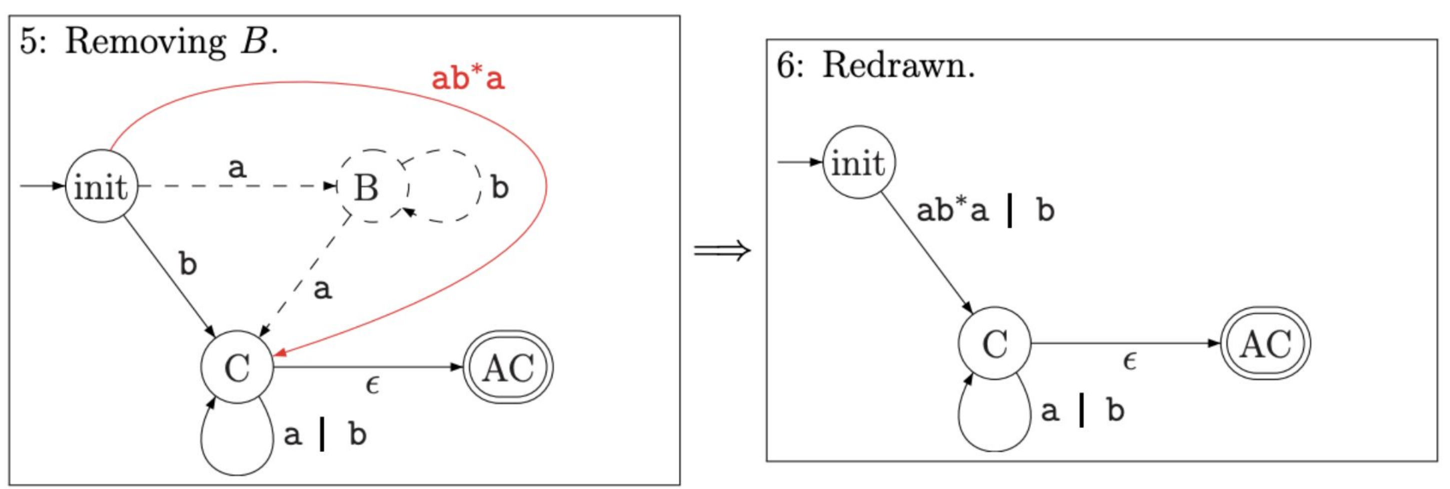
DFA to RE

1. Normalization: Add two new states that connect the original start state & final states with epsilon transition resp.
2. Remove states one by one
    - w/o loops
    - w loops



DFA to RE

1. Normalization: Add two new states that connect the original start state & final states with epsilon transition resp.
2. Remove states one by one



Final Answer: (ab*a | b) (a | b)*