

Java application Features

Convert Regular grammars into Finite Automat as such:

1. RG to NFA (as a formal definition with transition table)
2. ϵ -NFA into an NFA without ϵ -transitions (as a formal definition with transition table)
3. NFA into a DFA (transition table)
4. Minimize the DFA (transition table), can Draw the DFA (optional)
5. Testing strings (up to 5 strings at once) a statement to inform user whether each string is accepted or rejected.

App Design:

The application interface consists of three main tabs: Home, RG \rightarrow FA, and Help. The RG \rightarrow FA tab is active in all screenshots.

Regular Grammars (input):

User is allowed to import grammars as text file or type directly into the window

Must accept any number of variables, Min 3 alphabet and ϵ

Regular Grammars (input):

$A \rightarrow 1B \mid 0C$
 $B \rightarrow \epsilon$
 $C \rightarrow B \mid 1A$

Import Clear

Conversion Options:

NFA
NFA w/o ϵ
DFA
Min DFA
Test

Final Output (NFA and DFA):

NFA:

$M = (Q, \Sigma, \delta, p_0, F)$
 $Q = \{A, B, C\}$
 $\Sigma = \{a, b\}$
 $\delta: Q \times \Sigma \rightarrow \text{Pow}(Q)$
 $p_0 = A$
 $F = \{B\}$

δ_{NFA}	0	1	ϵ
A	$\{C\}$	$\{B\}$	\emptyset
B	\emptyset	\emptyset	\emptyset
C	\emptyset	$\{A\}$	$\{B\}$

DFA:

$M = (Q, \Sigma, \delta, p_0, F)$
 $Q = \{A, B, C\}$
 $\Sigma = \{a, b\}$
 $\delta: Q \times \Sigma \rightarrow \text{Pow}(Q)$
 $p_0 = A$
 $F = \{B\}$

δ_{NFA}	0	1
$\rightarrow A$	$\{B, C\}$	$\{B\}$
$* B$	\emptyset	\emptyset
C	\emptyset	$\{A\}$

HomeRG ⇌ FAFHelp

Regular Grammars (input)

A → 1B | 0C

B → ε

C → B | 1A

Import

Clear

NFA

NFA w/o ε

DFA

Min DFA

Test

δ _{DFA}	0	1
Z	Z	Z
→ A	F	B
• B	Z	Z
C	Z	A
• D	F	B
E	F	D
• F	Z	A
• G	F	D

HomeRG ⇌ FAFHelp

Regular Grammars (input)

A → 1B | 0C

B → ε

C → B | 1A

Import

Clear

NFA

NFA w/o ε

DFA

Min DFA

Test

Must

δ _{DFA}	0	1
Z	Z	Z
→ A	F	B
• B	Z	Z
• F	Z	A

Optional

HomeRG ⇌ FAFHelp

Regular Grammars (input)

A → 1B | 0C

B → ε

C → B | 1A

Import

Clear

NFA

NFA w/o ε

DFA

Min DFA

Test

Check strings (input)

ε

1111

10101010

01010101

1

NO

NO

NO

OK

OK

Can check Min 5 strings at once

Check