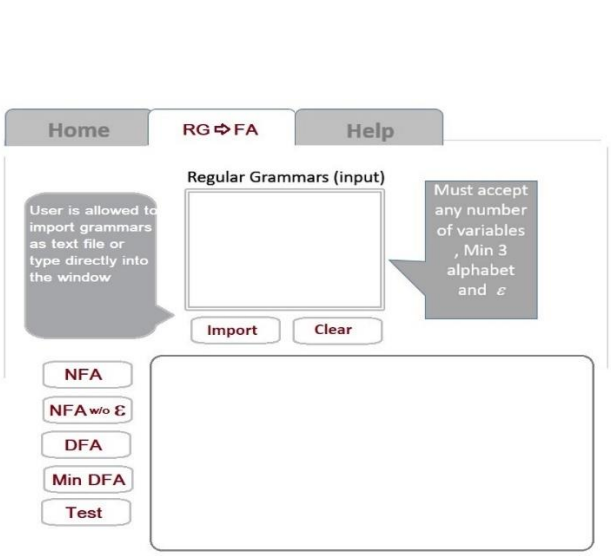
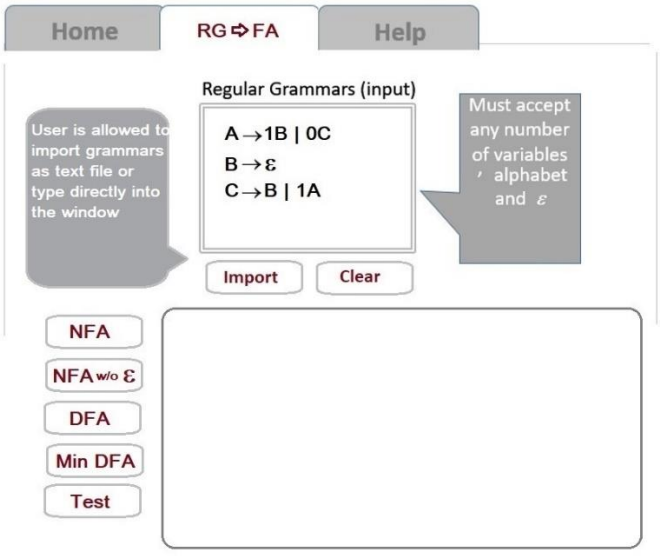
Java application Features

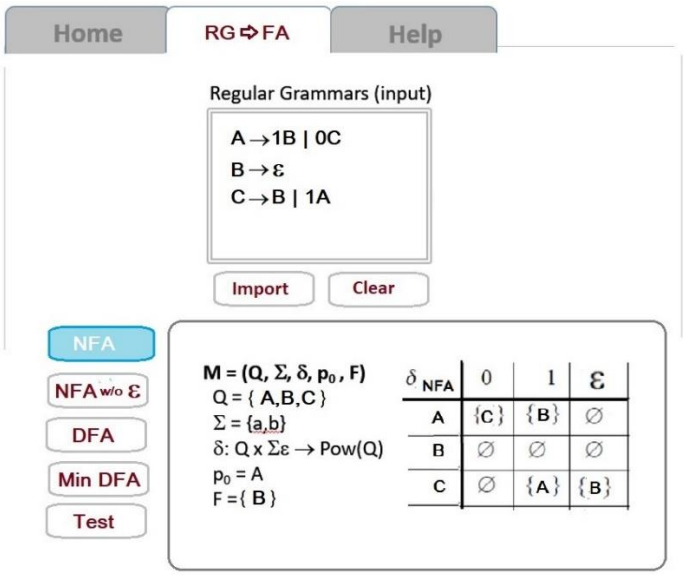
Convert Regular grammars into Finite Automat as such:

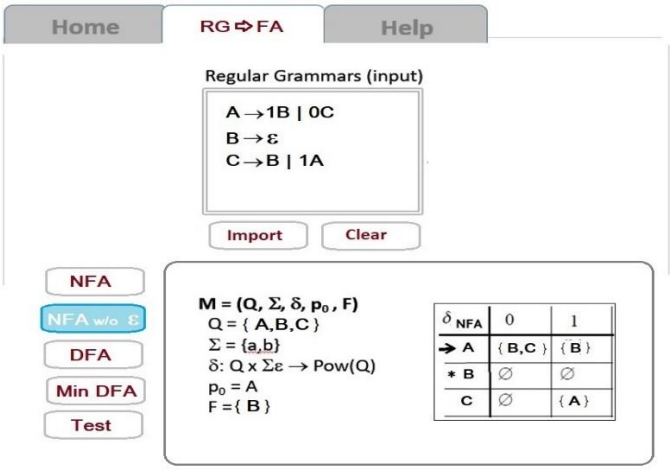
1. RG to NFA (as a formal definition with transition table)
2. ε-NFA into an NFA without ε-transitions (as a formal definition with transition table)
3. NFA into a DFA (transition table)
4. Minimize the DFA (transition table), can Draw the DFA (optional)
5. Testing strings (up to 5 strings at once) a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

App Design:









Home | RG ⇨ FA | Help

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

Home | RG ⇨ FA | Help

Regular Grammars (input)

A → 1B | 0C
B → ε
C → B | 1A

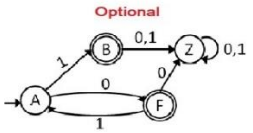
Import | Clear

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



Home | RG ⇨ FA | Help

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
0101010
1

NO
NO
NO
OK
OK

Can check Min 5 strings at once

Check