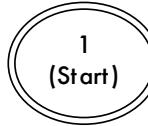
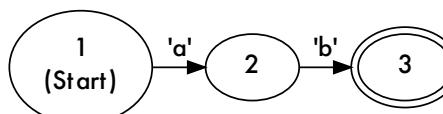
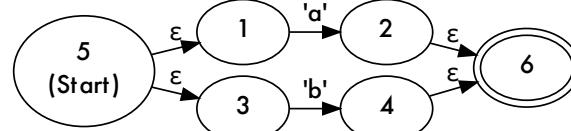
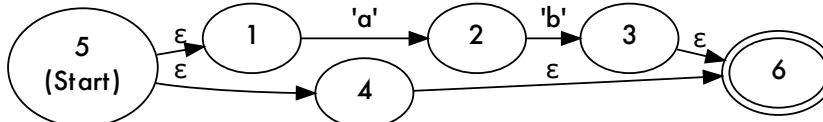
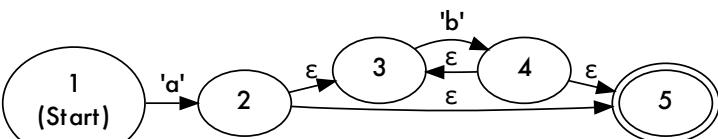


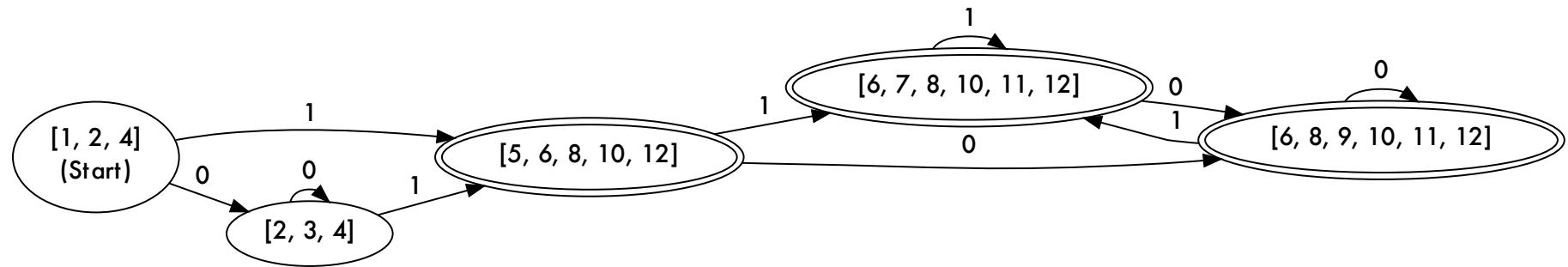
## Part 1 (NFA Construction)

Command	NFA
java Show NFA ''	 A single oval labeled '1 (Start)' with a self-loop arrow.
java Show NFA 'ab'	 A sequence of three ovals labeled '1 (Start)', '2', and '3'. An arrow labeled 'a' points from state 1 to state 2. An arrow labeled 'b' points from state 2 to state 3. State 3 is highlighted with a double circle.
java Show NFA 'a b'	 A set of six ovals labeled '5 (Start)', '1', '2', '3', '4', and '6'. State 5 is the start state. There are two arrows from state 5 to states 1 and 3, both labeled with the symbol ε. From state 1, an arrow labeled 'a' leads to state 2. From state 3, an arrow labeled 'b' leads to state 4. From state 2, an arrow labeled ε leads to state 6. From state 4, an arrow labeled ε leads to state 6. State 6 is highlighted with a double circle.
java Show NFA 'ab '	 A set of six ovals labeled '5 (Start)', '1', '2', '3', '4', and '6'. State 5 is the start state. There are two arrows from state 5 to states 1 and 4, both labeled with the symbol ε. From state 1, an arrow labeled 'a' leads to state 2. From state 4, an arrow labeled 'a' leads to state 2. From state 2, an arrow labeled 'b' leads to state 3. From state 3, an arrow labeled ε leads to state 6. State 6 is highlighted with a double circle.
java Show NFA 'ab*'	 A set of five ovals labeled '1 (Start)', '2', '3', '4', and '5'. State 1 is the start state. An arrow labeled 'a' points from state 1 to state 2. From state 2, an arrow labeled ε points to state 3. From state 3, an arrow labeled 'b' points to state 4. From state 4, two arrows labeled ε point to state 5 and state 3 respectively. State 5 is highlighted with a double circle.

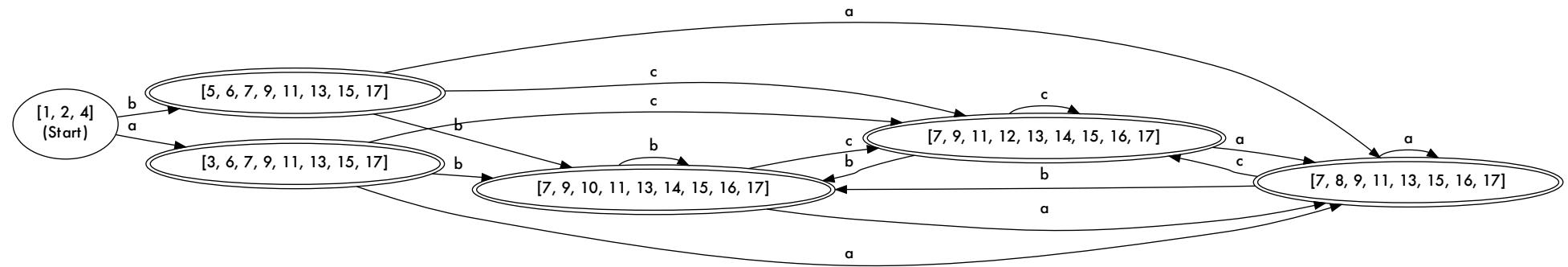
Command	NFA
java Show NFA '(ab)*'	<pre> graph LR     S1((1 (Start))) -- "ε" --&gt; S2((2))     S2 -- "'a'" --&gt; S3((3))     S2 -- "'b'" --&gt; S4((4))     S3 -- "ε" --&gt; S2     S4 -- "ε" --&gt; S2     S4 -- "ε" --&gt; S5(((5)))     style S5 fill:none,stroke:none     </pre>
java Show NFA '(ab c)*'	<pre> graph LR     S1((1 (Start))) -- "ε" --&gt; S10((10))     S10 -- "ε" --&gt; S2((2))     S10 -- "ε" --&gt; S8((8))     S2 -- "'a'" --&gt; S3((3))     S3 -- "'b'" --&gt; S4((4))     S5((5)) -- "'c'" --&gt; S6((6))     S8 -- "ε" --&gt; S5     S8 -- "ε" --&gt; S7((7))     S7 -- "ε" --&gt; S6     S6 -- "ε" --&gt; S4     S6 -- "ε" --&gt; S9((9))     S7 -- "ε" --&gt; S9     S9 -- "ε" --&gt; S11((11))     S11 -- "ε" --&gt; S12(((12)))     style S12 fill:none,stroke:none     </pre>

## Part 2 (DFA Construction)

```
java Show DFA-DETAIL '0*1(1|0)*'
```



```
java Show DFA-DETAIL '(a|b)(a|b|c)*'
```



### Part 3 (DFA Optimization)

Command	DFA
java Show DFA-OPT '0*1(1 0)*'	<pre> graph LR     A((A ---&gt; (Start))) -- 0 --&gt; A     A -- 1 --&gt; B(((B)))     B -- "0, 1" --&gt; B     </pre>
java Show DFA-OPT (a b)(a b c)*'	<pre> graph LR     A((A ---&gt; (Start))) -- "a, b" --&gt; B(((B)))     B -- "a, b, c" --&gt; B     </pre>
java Show DFA-OPT (a b  )(a b c)*'	<pre> graph LR     A(((A ---&gt; (Start)) -- "a, b, c" --&gt; A))     </pre>
java Show DFA-OPT '((0 ((1 2 3 4 5 6 7 8 9)((0 (1 2 3 4 5 6 7 8 9))(0 (1 2 3 4 5 6 7 8 9))*) ))((1 L) ))'	<pre> graph LR     A((A ---&gt; (Start))) -- 0 --&gt; B(((B ---&gt; (Final))) -- "0, 1, 2, 3, 4, 5, 6, 7, 8, 9" --&gt; B)     A -- "1, 2, 3, 4, 5, 6, 7, 8, 9" --&gt; B     C(((C))) -- 0 --&gt; B     C -- "L, I" --&gt; D(((D ---&gt; (Final))) -- "L, I" --&gt; D)     </pre>

## Part 4 (Tokenization)

```
java Tokenize simple-tok.txt ab.txt
```

```
001:01 A2(aaa)
001:04 B1(bbaa)
001:08 A2(aa)
003:03 A2(aa)
003:06 B2(bb)
003:09 A2(aa)
005:01 A2(a)
005:02 B2(b)
005:03 A2(a)
005:04 B1(bbaa)
005:08 A1(aabb)
005:12 A2(a)
EOF.
```

```
cat num.txt | java Tokenize num-tok.txt
```

```
001:01 SpecialNumber(123)
001:06 DecimalIntegerLiteral(1231)
001:12 DecimalIntegerLiteral(92399239L)
001:23 HexIntegerLiteral(0x1231)
001:31 DecimalIntegerLiteral(0)
001:32 DecimalIntegerLiteral(987L)
003:01 HexIntegerLiteral(0x123)
003:06 DecimalFloatingPointLiteral(.23)
004:01 DecimalFloatingPointLiteral(23e-123)
EOF.
```