CMSC 330: Organization of Programming Languages

Examples of REs & Finite Automata

Describing Regular Expressions

a) 0(0|1)*0
   - All strings beginning and ending in 0

b) ((ε|0)1*)*
   - All strings

c) (0|1)*0(0|1)(0|1)
   - All strings with 0 as third digit from right

Creating Regular Expressions

For all strings of 0's and 1's that…

a) Begin in 1
   - 1(0|1)*

b) End in 1
   - (0|1)*1

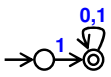
c) Contains 00
   - (0|1)*00(0|1)*

d) Do not contain 00
   - (01|1)*(ε|0)

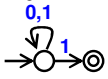
Creating NFA

For all strings of 0's and 1's that…

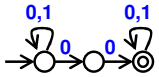
a) Begin in 1



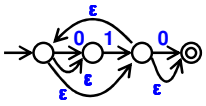
b) End in 1



c) Contains 00



d) Do not contain 00

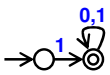


Based on regular expression

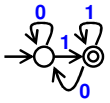
Creating DFA

For all strings of 0's and 1's that…

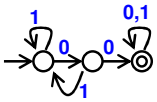
a) Begin in 1



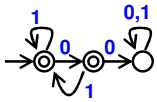
b) End in 1



c) Contains 00



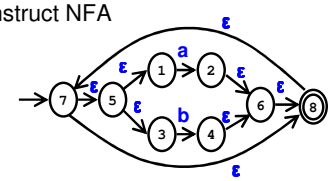
d) Do not contain 00



Swap final / non-final states!

For RE (a | b)*

a) Construct NFA



b) Accept ababbab

7,5,1,2,6,8,7,5,3,4,6,8,7,5,1,2,6,8,7,5,3,4,6,8,7,5,3,4,6, 8,7,5,1,2,6,8,7,5,3,4,6,8 accept

For RE (a | b)*

c) Reduce NFA to DFA

- Start = ε-closure(7) = {7,5,1,3,8}
- R = {{7,5,1,3,8}}
- r ∈ R = {7,5,1,3,8} // mark DFA state
- move ({7,5,1,3,8}, a) = {2}
  - e = ε-closure({2}) = {2,6,8,7,5,1,3} // new DFA state
  - R = R ∪ {2,6,8,7,5,1,3} // add to R
  - δ = δ ∪ ({7,5,1,3,8}, a, {2,6,8,7,5,1,3})
- move ({7,5,1,3,8}, b) = {4}
  - e = ε-closure({4}) = {4,6,8,7,5,1,3} // new DFA state
  - R = R ∪ {4,6,8,7,5,1,3} // add to R
  - δ = δ ∪ ({7,5,1,3,8}, b, {4,6,8,7,5,1,3})

For RE (a | b)*

- R = {{7,5,1,3,8}, {2,6,8,7,5,1,3}, {4,6,8,7,5,1,3}}
- r ∈ R = {2,6,8,7,5,1,3} // mark DFA state
- move ({2,6,8,7,5,1,3}, a) = {2}
  - e = ε-closure({2}) = {2,6,8,7,5,1,3} // existing DFA state
  - δ = δ ∪ ({2,6,8,7,5,1,3}, a, {2,6,8,7,5,1,3})
- move ({2,6,8,7,5,1,3}, b) = {4}
  - e = ε-closure({4}) = {4,6,8,7,5,1,3} // existing DFA state
  - δ = δ ∪ ({2,6,8,7,5,1,3}, b, {4,6,8,7,5,1,3})

For RE (a | b)*

- R = {{7,5,1,3,8}, {2,6,8,7,5,1,3}, {4,6,8,7,5,1,3}}
- r ∈ R = {4,6,8,7,5,1,3} // mark DFA state
- move ({4,6,8,7,5,1,3}, a) = {2}
  - e = ε-closure({2}) = {2,6,8,7,5,1,3} // existing DFA state
  - δ = δ ∪ ({4,6,8,7,5,1,3}, a, {2,6,8,7,5,1,3})
- move ({4,6,8,7,5,1,3}, b) = {4}
  - e = ε-closure({4}) = {4,6,8,7,5,1,3} // existing DFA state
  - δ = δ ∪ ({4,6,8,7,5,1,3}, b, {4,6,8,7,5,1,3})
- R = {{7,5,1,3,8}, {2,6,8,7,5,1,3}, {4,6,8,7,5,1,3}}
  - No more unmarked states to process
- F_d = {{7,5,1,3,8}, {2,6,8,7,5,1,3}, {4,6,8,7,5,1,3}}
  - Since 8 ∈ F_n

For RE (a | b)*

- Resulting DFA

  Σ = {a,b}

  R = {{7,5,1,3,8}, {2,6,8,7,5,1,3}, {4,6,8,7,5,1,3}}

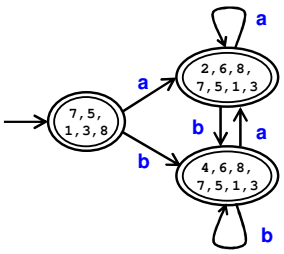
  r_0 = {7,5,1,3,8}

  F_d = {{7,5,1,3,8}, {2,6,8,7,5,1,3}, {4,6,8,7,5,1,3}}

  δ = { ({7,5,1,3,8}, a, {2,6,8,7,5,1,3}),
  ({7,5,1,3,8}, b, {4,6,8,7,5,1,3}),
  ({2,6,8,7,5,1,3}, a, {2,6,8,7,5,1,3}),
  ({2,6,8,7,5,1,3}, b, {4,6,8,7,5,1,3}),
  ({4,6,8,7,5,1,3}, a, {2,6,8,7,5,1,3}),
  ({4,6,8,7,5,1,3}, b, {4,6,8,7,5,1,3}) }

For RE (a | b)*

- NFA to DFA reduction pictorial

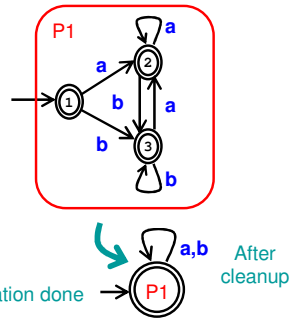


For RE (a | b)*

d) Minimize DFA

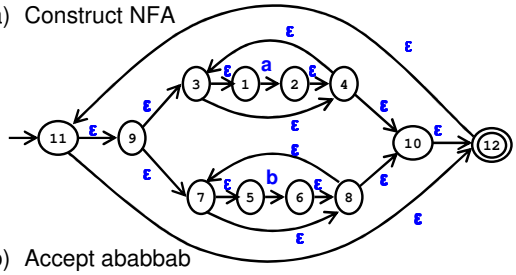
- Initial partitions
  - Accept → {1,2,3} → P1
  - Reject → Ø
- Split partition?
  - move(1,a) → P1
  - move(2,a) → P1
  - move(3,a) → P1
  - move(1,b) → P1
  - move(2,b) → P1
  - move(3,b) → P1

Not required, minimization done



For RE (a* | b*)*

a) Construct NFA



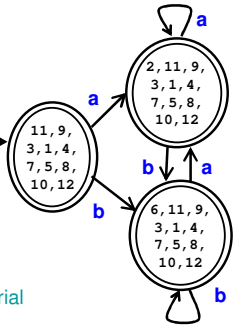
b) Accept ababbab

11,9,3,1,2,4,10,12,11,9,7,5,6,8,10,12,11,9,3,1,2,4,10,12…

For RE (a* | b*)*

c) Reduce NFA to DFA

- Start = ε-closure(1)
  = {11,9,3,1,4,7,5,8,10,12}
- R = {{11,9…12}}
- r ∈ R = {11,9…12}}}
- move ({11,9…12}, a) = 2
  - ε-closure({2}) = {2,11,9…12}
- move ({11,9,…,12}, b) = 6
  - ε-closure({6}) = {6,11,9…12}
- …
- NFA to DFA reduction pictorial

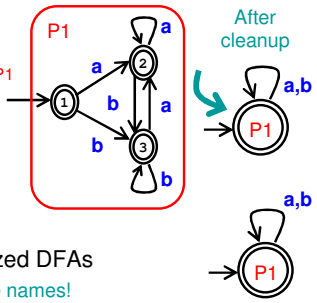


For RE (a* | b*)*

d) Minimize DFA

- Initial partitions
  - Accept → {1,2,3} → P1
  - Reject → Ø
- Split partition?
  - move(1,a) → P1
  - …

  Not required, minimization done



e) Compare 2 minimized DFAs

- Identical up to state names!