Announcements

- Reminder
  - P1 part 1 due Thursday
    part 2 due Tuesday
  - Office hours tomorrow
    1:00 - 2:15
  - Doc Cam Poll

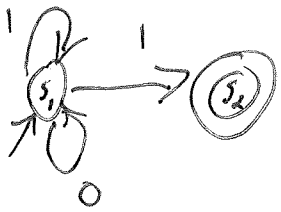
Last Time

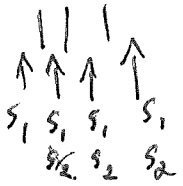
- Abstraction
- Introduced FSMs (Finite State Machines)
  - ↳ Deterministic: 1 path per string (DFAs)
  - ↳ Nondeterministic:
    - Allow duplicate edges, ε-edges
    - If a string yields <u>any</u> path through the NFA to a final state accept

This time

- Explore NFAs
  - ↳ Claim: NFAs add no power over DFAs
  - ↳ ε-transitions
  - ↳ Claim: ε-transitions add no power over NFA
- Regular Expressions

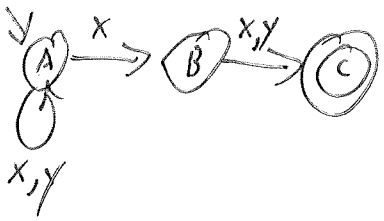
NFAs





To check if a string is in $L(NFA)$, simulate <u>set</u> of choices it <u>could</u> make

Claim: $L(NFA) \equiv L(DFA)$



Idea: Only finitely many subsets of states

$2^{|Q|}$ ← $|X|$ is the cardinality (size) of a set

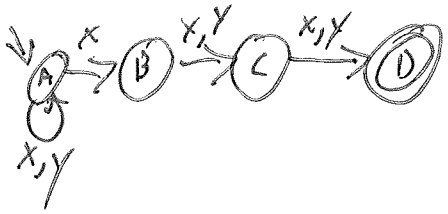
Why $2^{|Q|}$?

A	B	C	
0	0	0	$= \{\}$
0	0	1	$= \{C\}$
0	1	0	$= \{B\}$
0	1	1	$= \{B, C\}$
1	0	0	$= \{A\}$
1	0	1	$= \{A, C\}$
1	1	0	$= \{A, B\}$
1	1	1	$= \{A, B, C\}$

Build DFA that tracks set of states the NFA could be in

3rd to last char is x

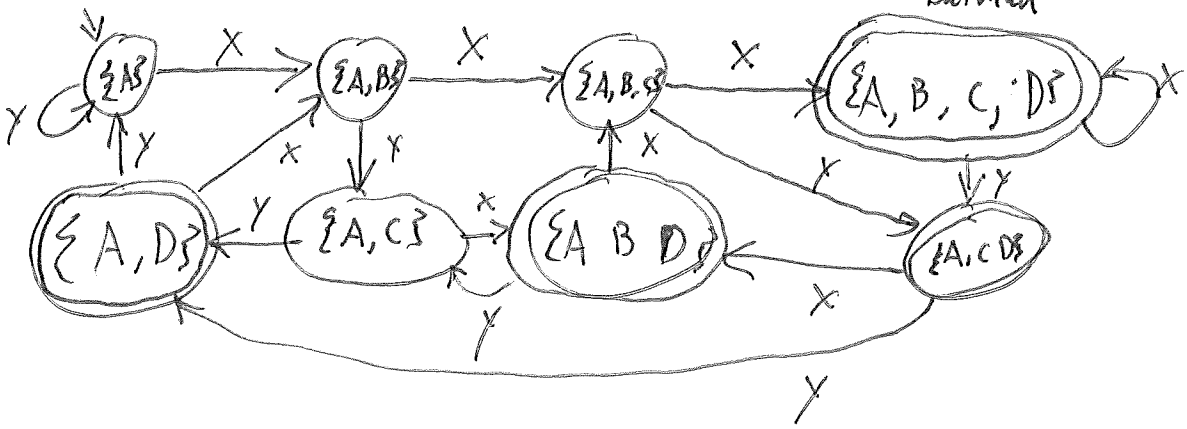
Batman



Def: Let $Successor_{Batman}(s, c)$ be the set of choices the NFA could make in state s with char c

$A, x = \{A, B\}$ $B, y = \{C\}$
$A, y = \{A\}$ $C, x = \{D\}$
$B, x = \{C\}$ $C, y = \{D\}$

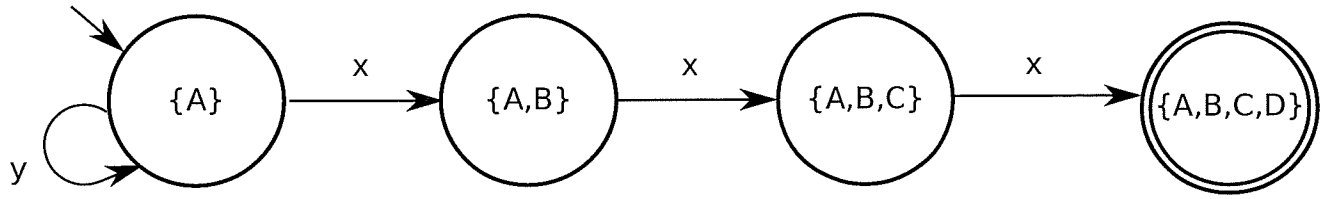
1) Build new DFA Batman' where $Q_{Batman'} = 2^{Q_{Batman}}$



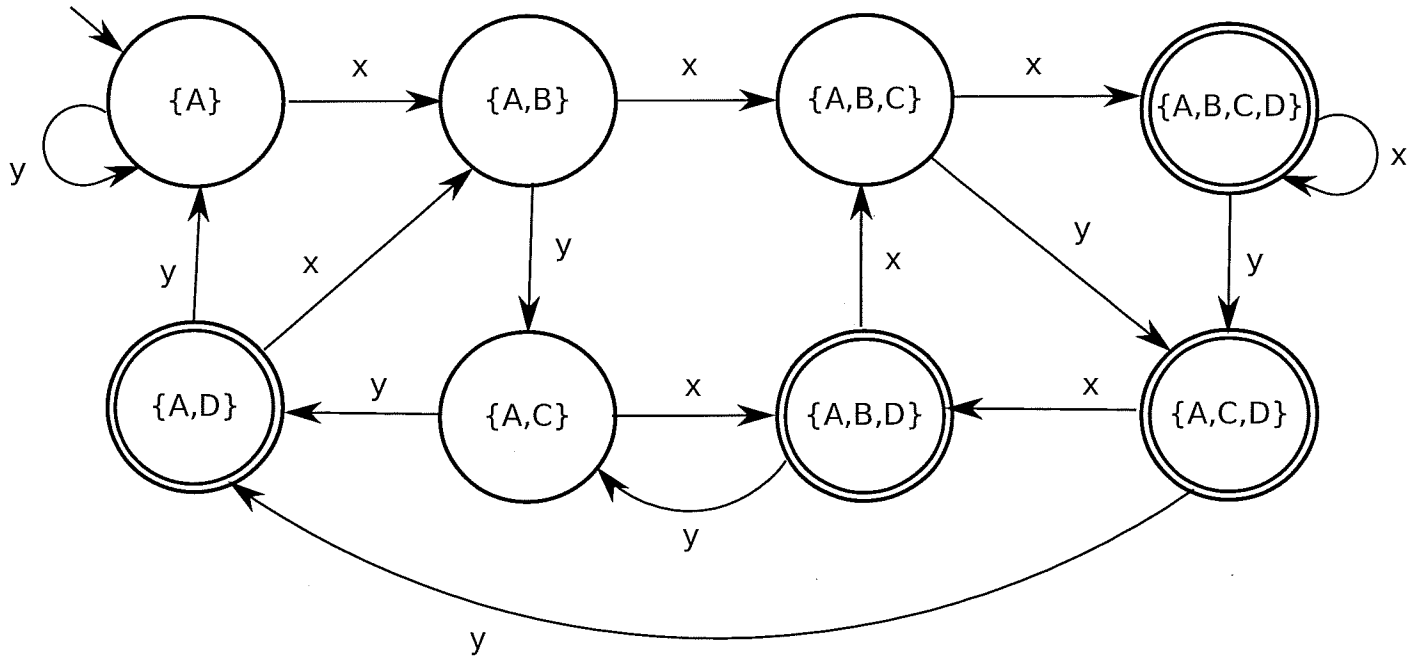
2) $F_{Batman'}$ = States α where some element of α is in F_{Batman}

3) $\delta(\alpha, c) = \beta$ where β is the $\cup$ of possible choices Batman can make from states in α

Initial NFA



Corresponding DFA



It's interesting in this particular case that each state intuitively represents a configuration of the last 3 characters seen:

state {A} = none of the last 3 are x (i.e 3-character *suffix* of the string seen so far is yyy)

state {A,B} = suffix xyy

state {A,B,C} = suffix xxy

state {A,B,C,D} = suffix xxx

state {A,D} = suffix xyy

state {A,C} = suffix yxy)

state {A,B,D} = suffix xyx

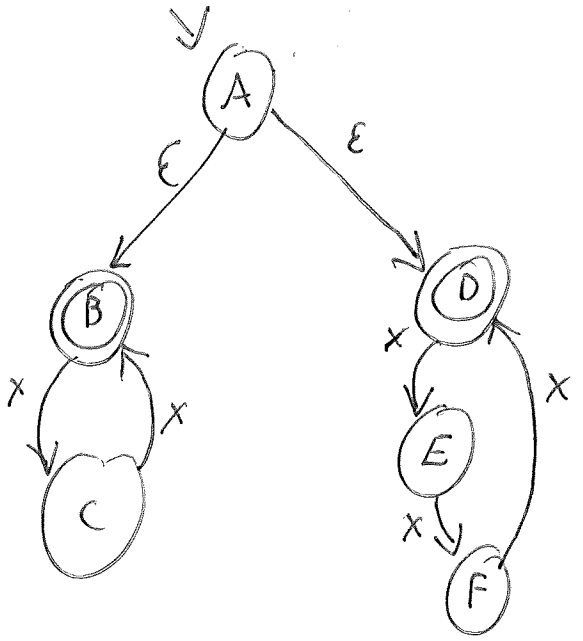
state {A,C,D} = suffix xyx

To build the DFA:

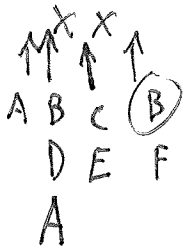
Add an edge from state α on character c to state β if β represents the union of states that all states in α could possibly transition to on character c

ε-transitions

$L(\{x^n \mid n \text{ is even or divisible by } 3\})$

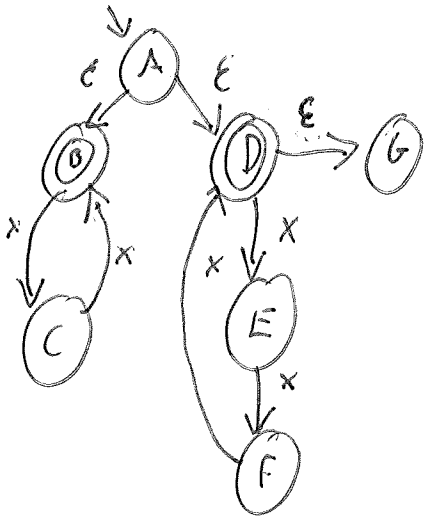


Note: ε-transitions
Good for DFA union



NFA eliminating ε-transitions

Let ε-close(s) be defined as the set of states reachable from 0 or more ε-transitions



ε-close(A) = {A, B, D, G}

ε-close(D) = {D, G}

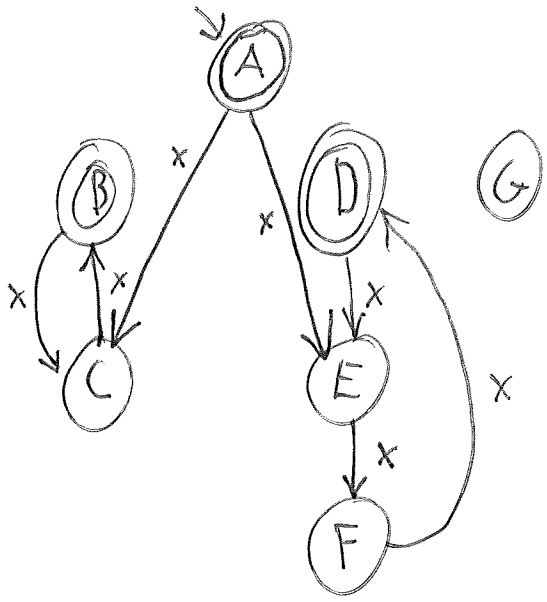
for all others

ε-close(x) = {x}

ε-close(G) = {G}

Construct no-ε NFA N' from N

- Put s in F' if ε-close(s) contains a state in F
- Put s, c → t in δ' if there is a c-edge in ε-close(s)
- Add all non-ε edges from N to N'



Regular Expressions

Pattern describing a language

Operands : single chars, ε

Operators :

lowest precedence →	alternation	"or"	$a \mid b$
	catenation		$a.b$ ab $a^3 = aaa$
highest precedence →	iteration	"Kleene star"	a^* (0 or more as)

Conventions

$a+$	is	$a.a^*$
letter		$a \mid b \mid c \mid \ldots \mid y \mid z \mid A \mid B \mid \ldots \mid Z$
digit		$0 \mid 1 \mid 2 \mid \ldots 9$

quotes for literals

parens for grouping $(ab)^*$

ε, ab, abab, ababab

Example RegEx

Hex strings

- start with $0x$ or $0X$ followed by ≥ 1 hexadecimal digit
- Optionally end with l or L

$$0(x|X)\,hexdigit+(L|l|\epsilon)$$

$$hexdigit = digit\,|\,a\,|\,A\,|\,b\,|\,B\,|\ldots|\,f\,|\,F$$

$$\Big(0(x|X)\,\underbrace{hexdigit+}_{\text{lowercase}}(L|l|\epsilon)\Big)\;\Big|\;\Big(\ldots\;\underbrace{hexdigit+}_{\text{uppercase}}\Big)$$