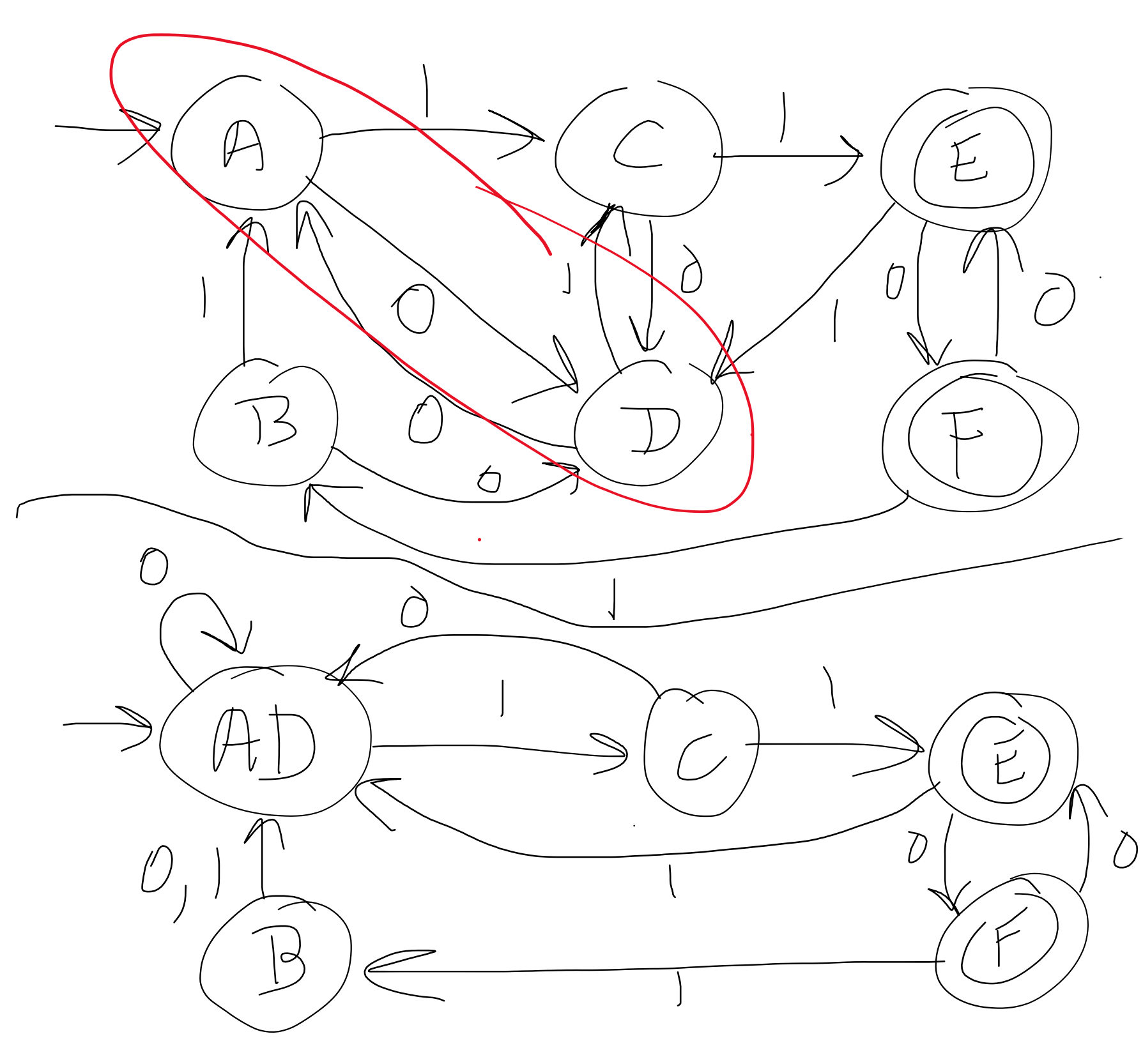
Thursday, September 8, 2022 11:40 AM

CSCE 355
9/8/22

DFA min
Closure properties of reg langs
Regular expressions

DFA min example



B	X				
C	X	X			
D		X	X		
E	X	X	X	X	
F	X	X	X	X	X
	A	B	C	D	E

Let REG_Σ be the class of all regular languages (over a fixed alphabet Σ)

Closure properties of REG_Σ:

1. closed under complements (if A is regular, then so is $\overline{A}$)
2. closed under intersection (A, B regular $\Rightarrow A \cap B$ regular) ← product construction for DFAs
3. closed under union ($A \cup B = \overline{\overline{A} \cap \overline{B}}$ by De Morgan's Laws)
4. closed under any Boolean combo: A, B regular
$\Rightarrow \begin{cases} A - B = A \cap \overline{B} \\ A \Delta B = (A-B) \cup (B-A) \end{cases}$ both regular

Define 2 more operations on languages A, B:

1) $AB = \{xy : x \in A \text{ and } y \in B\}$ (associative)

2) $A^* = \underbrace{\{\varepsilon\}}_{A^0 \text{ by convention}} \cup A \cup AA \cup AAA \cup AAAA \cup \dots \cup A^n \cup \dots$ (A^n = $\underbrace{AA \cdots A}_{n \text{ times}}$)

$= \bigcup_{i=0}^{\infty} A^i$

Ex $\{a\}^* = \{\varepsilon, a, aa, aaa, \dots\}$

$\{a,b\}\{b,c\} = \{ab, ac, bb, bc\}$

$\{a,b\}^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \dots\}$ (all strings over $\{a,b\}$)

Σ^* same

$\{aa, bc\}^* = \{\varepsilon, aa, bc, aaaa, aabc, bcaa, bcbc, aaaaaa, \dots\}$

$*$ — Kleene $*$-operator or Kleene closure operator

Will "show" next week that REG_Σ is closed under concatenation and the $*$-operator.

Regular Expressions (Regexes) over an alphabet

Def: A regex r over an alphabet Σ is an expression built from the following:

$\emptyset$
a (for alphabet symbol $a \in \Sigma$)
] atomic regexes (no proper subexprs)

OR built from regexes s and t as either

$s \cup t$
st
s^*
] built from subexpressions using $\cup$, concat, or $*$

Parentheses can be used for grouping. Can drop parens by abiding by precedence and associativity rules:

$r \cup s$ lowest prec, associative
st next higher prec, assoc.
s^* highest precedence

Regexes denote regular langs as follows:

Def: A regex r denotes a a language $L(r) \subseteq \Sigma^*$ according to the following recursive rules:

$L(\emptyset) = \emptyset = \{\}$ (no strings)
$a \in \Sigma$: $L(a) = \{a\}$ a single string of length 1
] atomic regexes

nonatomic regexes, let s and t be regexes and assume that $L(s)$ and $L(t)$ are already defined. Then

$L(s \cup t) = L(r) \cup L(t)$

$L(st) = L(s)L(t)$

$L(s^*) = L(s)^*$

Ex

$L(a \cup b) = \{a, b\} = L(a) \cup L(b)$ (where $L(a) = \{a\}$, $L(b) = \{b\}$)

$L(ab) = \{ab\} = L(a)L(b)$

$L(a^*) = L(a)^* = \{a\}^* = \dots$

Lots of practical use in pattern matching & text processing

describing token syntax in programming langs:
- identifiers
- int constants
- fp constants
- operators, etc.