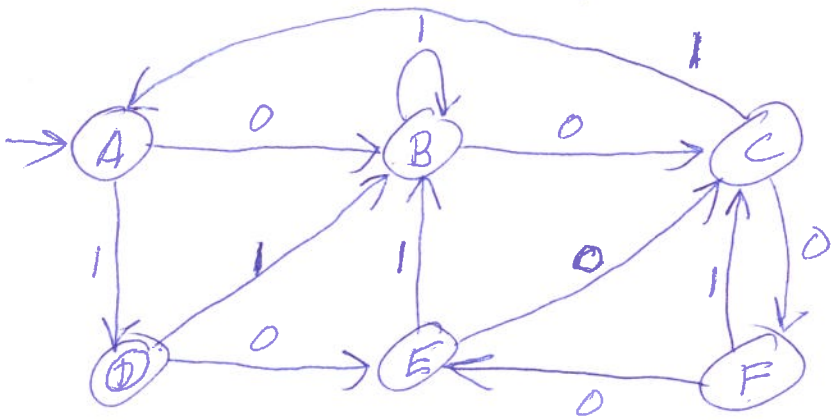
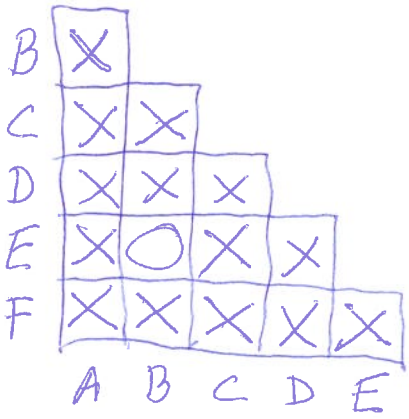
CSCE 355
1/31/2024

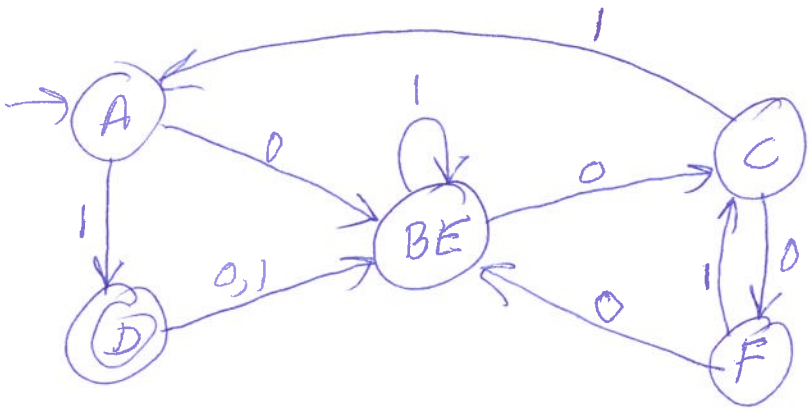
One more DFA min example

Regexes





Merge B & E to get min DFA:



Regexes — used to denote certain languages

Def: Fix alphabet Σ. A regular expression (regex) over Σ is an expression built from $\emptyset$, symbols in Σ, λ, using operators $+, \cdot, *$, as follows:

Atomic regexes

$\emptyset$ and a (for any $a \in \Sigma$)

Nonatomic regexes: If r and s are regexes, then so are

$r + s$ (union + — binary infix)

rs (concatenation — binary infix)

r^* (Kleene closure — unary postfix)

Nothing else

Ex: $ab + c = (ab) + c$ $\quad \Sigma = \{a, b, c\}$

$a(b + c^*) = a(b + (c^*))$

$ab^* = a(b^*)$

Def: (Regex semantics) Every regex r over Σ denote a language $L(r)$ over Σ according to the rules:

$L(\emptyset) = \emptyset = \{\}$] atomic regexes

$a \in \Sigma \quad L(a) = \{a\}$]

Let r and s be any regexes over Σ!

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

$$L(rs) = L(r)L(s)$$

Side note: $L, M \subseteq \Sigma^*$ languages,

$$LM = \{xy : x \in L \text{ and } y \in M\}$$

$L(r^*)$: Letting $L := L(r)$,

$$L(r^*) = \{\varepsilon\} \cup L \cup LL \cup LLL \cup \cdots$$

$$= \{w_1 \cdots w_n : n \geq 0 \ \& \ w_1, \ldots, w_n \in L\}$$

Ex: $L(ab+c) = L(ab) \cup L(c) = L(a)L(b) \cup L(c)$

$= \{a\}\{b\} \cup \{c\} = \{ab\} \cup \{c\} = \{ab, c\}$

$L(ab^*) = L(a)L(b^*) = L(a)(\{\varepsilon\} \cup L(b) \cup L(b)L(b) \cup \cdots)$

~~$= a(\varepsilon \cup b \cup$~~ $\{a\}(\{\varepsilon\} \cup \{b\} \cup \{b\}\{b\} \cup \cdots)$

$= \{\underset{a}{a\varepsilon}, ab, abb, abbb, \ldots\}$

"one a followed by 0 or more b's"

Side note: $L \subseteq \Sigma^*$ language

$L^* = \{\varepsilon\} \cup L \cup LL \cup LLL \cup \cdots$

so $L(r^*) = (L(r))^*$

Def: Let r be a regex over Σ and $w \in \Sigma^*$.

Say that r matches w (or w matches r) if $w \in L(r)$.

Matching token types:

Regex that matches all (unsigned) integer constants (in a typical prog. lang.):

$(0+1+2+3+4+5+6+7+8+9)(0+\cdots+9)^*$

$(0+1+2+3+4+5+6+7+8+9)$: matches single digits

$(0+\cdots+9)^*$: matches zero or more digits concatenated

Shorthands ("syntactic sugar")

$[0123456789] \equiv 0+1+2+\cdots+9$

$[0123456789]$: character class

Generally, [...] matches any single symbol between the brackets

Integer constants: [012...9][012...9]*

Char class allow subranges:

[0-9] means [012...9] (contiguous in ASCII sequence)

Int const. [0-9][0-9]*

Named subexprs:

{digit} [0-9]

{int_const} {digit}{digit}*

For regex r, r+ means rr*

(one or more occurrences of r)

{int_const} {digit}+

or [0-9]+

r? means "optional r" either ε or something matching r

Regex r such that $L(r) = \{\varepsilon\}$:

$$\emptyset^*$$

$$L(\emptyset^*) := \{\varepsilon\} \cup \emptyset \cup \emptyset\emptyset \cup \cdots$$

$$= \{\varepsilon\}$$

Shorthand: Use $\varepsilon := \emptyset^*$ as a regex.

So — $r? := r + \varepsilon$

Optionally signed int const:

$$[+-]?[0-9]+$$

"abc" stands for $\{a, b, c\}$ (or the regex abc)

"" stands for ε

Ex: Identifiers

$$[A-Za-z\_][A-Za-z0-9\_]*$$

~~Float~~ Floating point constants in Pascal:

$$[0-9]+"."[0-9]+([Ee][+-]?[0-9]+)?$$

Use | for union: $0|1|2|\cdots|9 = [0-9]$