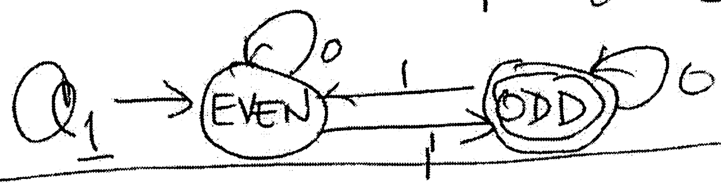
WEEKS #2 + #1

SAMPLES

1 OR 9a

ODD PARITY $\Sigma = \{0,1\}$ $Q = \{EVEN, ODD\}$

$F = \{ODD\}$ $q_0 = EVEN$



CHECK FOR 2's COMPLEMENT

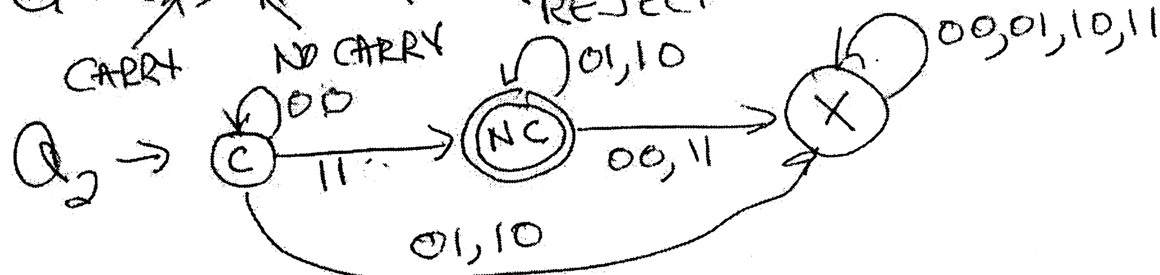
$\Sigma = \{0,1\} \times \{0,1\} = \{00, 01, 10, 11\}$

THINK OF THIS AS TWO SYCHRONIZED CHANNELS OF INPUT (TOP IS #, BOTTOM IS 2's COMPLEMENT)

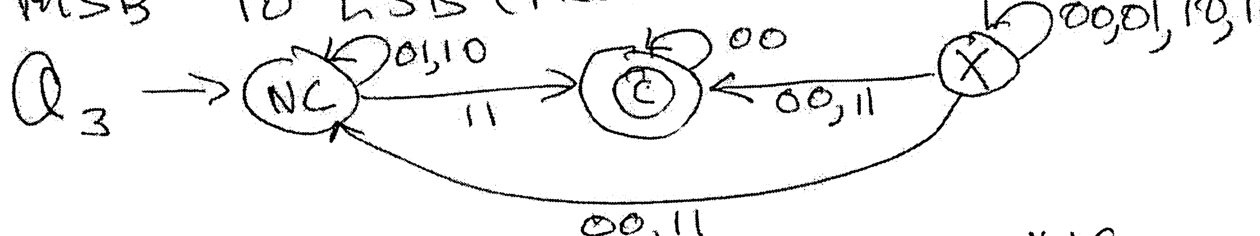
WE WILL DO LEAST SIGNIFICANT TO MOST SIGNIFICANT BIT (LSB TO MSB) FIRST

$Q = \{C, NC, X\}$ $q_0 = C$ $F = \{NC\}$

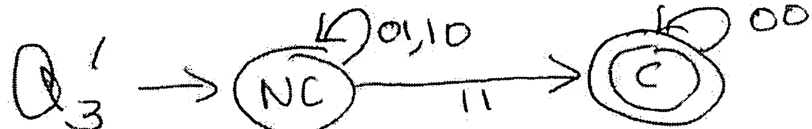
CARRY, NO CARRY, REJECT



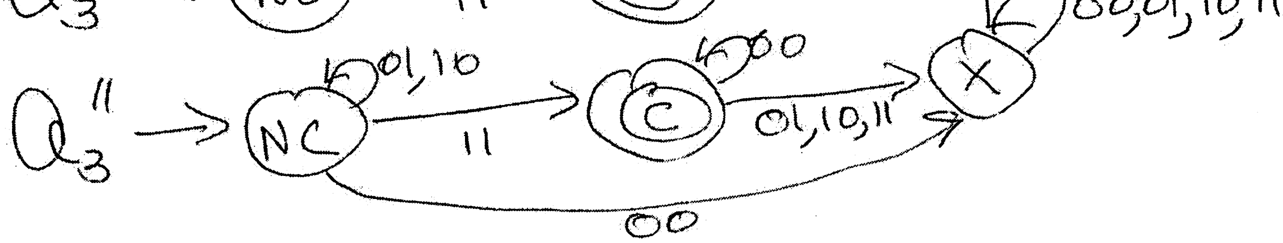
MSB TO LSB (FROM Q2)



BUT X IS NOT REACHABLE FOM NC



OR



WEEK#2

NON-DETERMINISTIC AUTOMATON (NFA)

REALLY VIEW AS $\{q_0\}$

$$Q = (Q, \Sigma, \delta, q_0, F)$$

$$\delta : Q \times \Sigma_\epsilon \to 2^Q = \mathcal{P}(Q)$$

$$\Sigma_\epsilon = \Sigma \cup \{\epsilon\} = \Sigma \cup \{\lambda\}$$

$$\lambda\text{-CLOSURE}(S) = \{t \mid t \in \delta^*(S, \lambda)\},\ S \in \mathcal{P}(Q)$$

HERE, WE USED EXTENDED δ THAT OPERATES ON SETS $\delta(S,a) = \bigcup_{q \in S} \delta(q,a)$ WHERE $a \in \Sigma_\epsilon$

THIS ACCOMMODATES STATE CHANGES WITHOUT READING ANY INPUT.

$$\delta^*(S, \lambda) = \lambda\text{-CLOSURE}(S)$$

$$\delta^*(S, ax) = \bigcup_{q \in S} \bigcup_{p \in \text{CLOSURE}(\delta(q,a))} \delta^*(p, x),\quad a \in \Sigma,\ x \in \Sigma^*,\ S \in \mathcal{P}(Q)$$

δ^+ AS BEFORE

$$L(Q) = \{w \mid \delta^*(\{q_0\}, w) \cap F \neq \emptyset\}$$

ACTUALLY, ~~BETTER~~ TO USE

$$L(Q) = \{w \mid \delta^*(\lambda\text{-CLOSURE}(\{q_0\}), w) \cap F \neq \emptyset\}$$

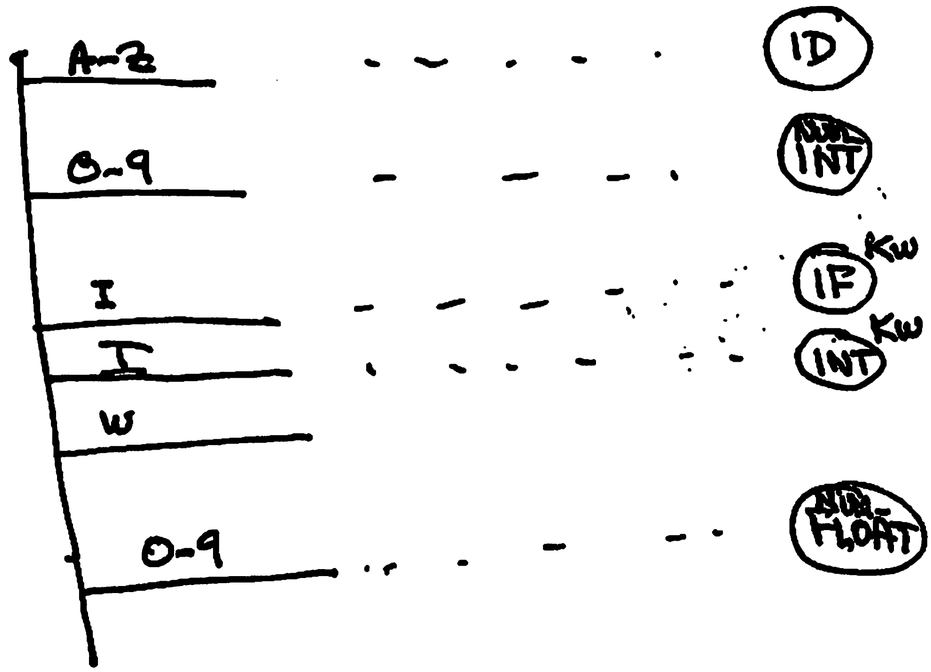
OR EVEN CHANGE START TO A SET

λ-CLOSURE$(\{q_0\})$

WEEK #2

CONSIDER LEXICAL ANALYSIS

ASSUME FIRST CHARACTER READ IN ADVANCE



WEEK #2

COMPLEMENT OF REGULAR LANGUAGE

1. REALLY BEGS FOR DFA VS NFA

$\mathcal{Q} = (Q, \Sigma, \delta, q_0, F)$ A DFA

$\mathcal{L}(\mathcal{Q}) = \{w \mid \delta^*(q_0, w) \in F\}$

$\overline{\mathcal{L}(\mathcal{Q})} = \{w \mid \delta^*(q_0, w) \notin F\}$

$= \{w \mid \delta^*(q_0, w) \in Q - F\}$

Thus

$\overline{\mathcal{L}(\mathcal{Q})} = \mathcal{L}(\mathcal{Q}^c)$ WHERE

$\mathcal{Q}^c = (Q, \Sigma, \delta, q_0, Q - F)$

WHICH IS A DFA

2. $\mathcal{Q} = (Q, \Sigma, \delta, q_0, F)$ AN NFA

$\mathcal{L}(\mathcal{Q}) = \{w \mid \delta^*(q_0, w) \cap F \neq \emptyset\}$

BUT $\mathcal{L}(\mathcal{Q}') = \{w \mid \delta^*(q_0, w) \cap (Q - F) \neq \emptyset\}$

IS NOT $\overline{\mathcal{L}(\mathcal{Q})}$

WEEK #2

WHEN SHOWING CLOSURE, WANT RIGHT MODEL

3. UNION

FOR NFA's

$Q_1 = (Q_1, \Sigma, \delta_1, q_{10}, F_1)$

$Q_2 = (Q_2, \Sigma, \delta_2, q_{20}, F_2)$

CAN DO UNION OF $L(Q_1)$ AND $L(Q_2)$

BY

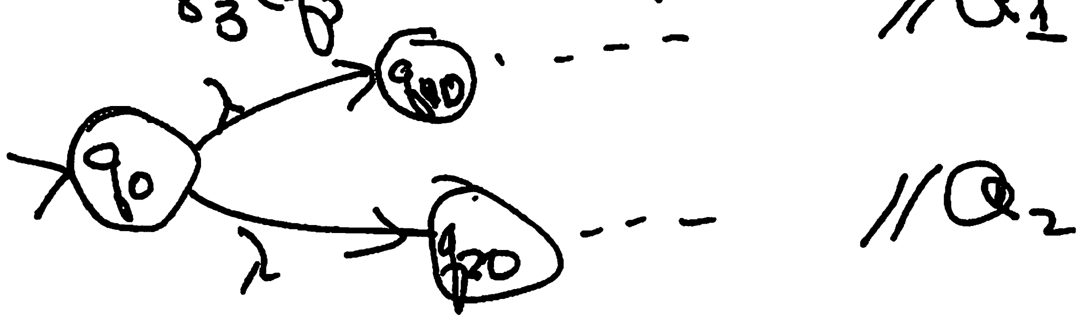
$Q_3 = (\{q_0\} \cup Q_1 \cup Q_2, \delta_3, q_0, F_1 \cup F_2)$

WHERE

$\delta_3(q_0, \lambda) = \{q_{10}, q_{20}\}$

$\delta_3(q, a) = \delta_1(q, a)$ WHEN $q \in Q_1$

$\delta_3(q, a) = \delta_2(q, a)$ WHEN $q \in Q_2$



WEEK #2

4. UNION CAN BE DONE WITH DFA BY

$\mathcal{Q}_1 = (Q_1, \Sigma, \delta_1, q_{10}, F_1)$; $\mathcal{Q}_2 = (Q_2, \Sigma, \delta_2, q_{20}, F_2)$

$\mathcal{Q}_3 = (Q_1 \times Q_2, \Sigma, \delta_3, \langle q_{10}, q_{20} \rangle, F_3)$

WHERE

$Q_1 \times Q_2 = \{ \langle p, q \rangle \mid p \in Q_1, q \in Q_2 \}$

$\delta_3(\langle p, q \rangle, a) = \langle \delta_1(p, a), \delta_2(q, a) \rangle$

SYNCHRONIZED PARALLEL (MISD - MULTIPLE INSTRUCTION, SINGLE DATA)

$F_3 = F_1 \times Q_2 \cup Q_1 \times F_2$

ACCEPTS IF FIRST OR SECOND OR BOTH ACCEPT

5. CAN USE THIS CONSTRUCT FOR MANY SET OPERATIONS, SOME OF WHICH, LIKE COMPLEMENT, CANNOT BE DIRECTLY PROVEN WITH NFA MODEL

$\mathcal{L}(\mathcal{Q}_1) \cap \mathcal{L}(\mathcal{Q}_2)$ SET $F_3 = F_1 \times F_2$

$\mathcal{L}(\mathcal{Q}_1) - \mathcal{L}(\mathcal{Q}_2)$ SET $F_3 = F_1 \times (Q_2 - F_2)$

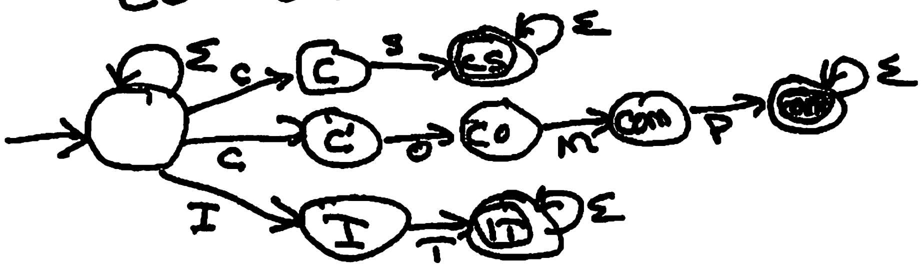
$\mathcal{L}(\mathcal{Q}_1) \oplus \mathcal{L}(\mathcal{Q}_2)$ SET $F_3 = F_1 \times (Q_2 - F_2) \cup (Q_1 - F_1) \times Q_2$

$= F_1 \times Q_2 \cup Q_1 \times F_2 - F_1 \times F_2$

6. OF COURSE, IF YOU HAVE UNION & COMPLEMENT, YOU CAN BUILD ALL OF ABOVE, BUT IT'S NOT AS INSTRUCTIVE OR INTUITIVE AS CONSTRUCTION

7. NFA'S

IF WANT ANY STRING CONTAINING "CS" OR "COMP" OR "IT"



GENEALLY EASIER TO WRITE NFA WHEN MULTIPLE PATHWAYS TO SUCCESS.

8. CONVERSION TO DFA

FIRST WANT TO ACCOMMODATE λ-TRANSITIONS

FOR $q \in Q$,

$$\lambda\text{-CLOSURE}(q) = \{t \mid \delta^*(q,\lambda) \supseteq \{t\}\}$$
$$= \{t \mid t \in \delta^*(q,\lambda)\}$$

FOR $S \subseteq Q$

$$\lambda\text{-CLOSURE}(S) = \{t \mid t \in \bigcup_{q \in S} \lambda\text{-CLOSURE}(q)\}$$

LET $A = (Q, \Sigma, \delta, q_0, F)$ BE AN NFA

DEFINE

$$A' = (\langle P(Q)\rangle, \Sigma, \delta', \langle \lambda\text{-CLOSURE}(q_0)\rangle, F')$$

WHERE

$$\delta'(\langle S\rangle, a) = \langle \lambda\text{-CLOSURE}(\delta(S,a))\rangle$$
$$= \langle \bigcup_{q \in S} \lambda\text{-CLOSURE}(\delta(q,a))\rangle, a \in \Sigma, S \in P(Q)$$

$$F' = \{\langle S\rangle \in \langle P(Q)\rangle \mid (S \cap F) \neq \emptyset\}$$

THIS LEADS TO $2^{|Q|}$ STATES, MOST OF WHICH ARE UNREACHABLE FROM START.

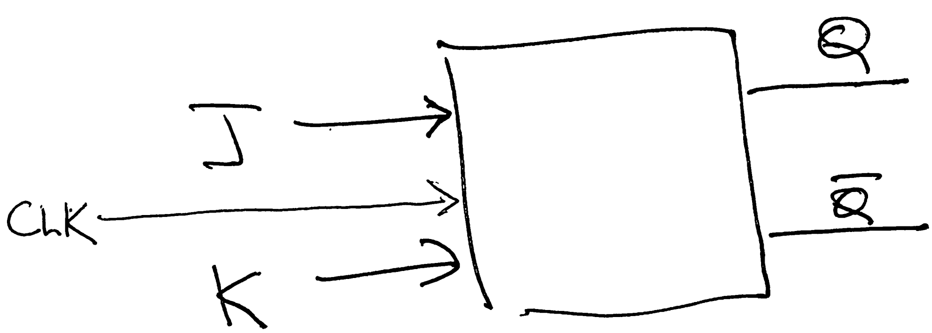
WEEK #2

WHERE DID FSA (DFA) COME FROM?

MATHEMATICAL/READABLE DESCIPTION OF SEQUENTIAL CIRCUITS

BOOLEAN + FLIP/FLOPS (2 VALUES)

PARITY CHECKER HAS JUST ONE FLIP/FLOP



$J=0, K=0$ NO CHANGE

$J=1, K=0$ $Q=1, \overline{Q}=0$

$J=0, K=1$ $Q=0, \overline{Q}=1$

$J=1, K=1$ FLIP VALUES

ODD PARITY JUST STARTS AT 0 AND EACH BIT IS REPLICATED ON J AND K. 1'S FLIP STATE; 0'S MAKE NO CHANGE

WEEK #2

9. COMPACT DFA FROM NFA

BUILD NEEDED SUBSETS FROM STATES ACCESSIBLE FROM START

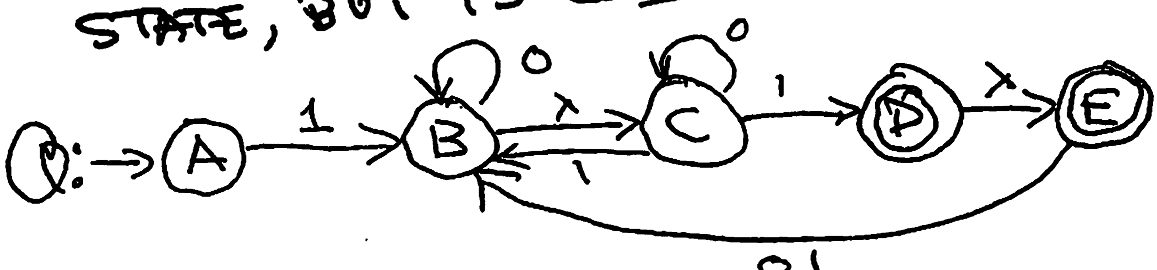
PROCESS
COMPUTE λ-CLOSURE OF EACH $q \in Q$.

START WITH λ-CLOSURE OF q_0 AND COMPUTE ALL SUCCESSOR STATES OF ONE STEP, ALWAYS DOING λ-CLOSURE OF THESE STATE SETS.

NEVER ADD AN UNNEEDED STATE

NOTE: <∅> IS COMMON AND IS "DEAD" STATE

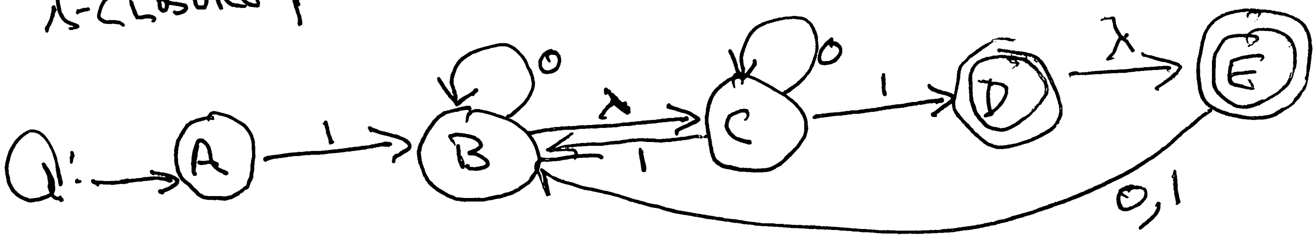
EVEN THIS DOES NOT GET MINIMAL STATE, BUT IS GOOD START

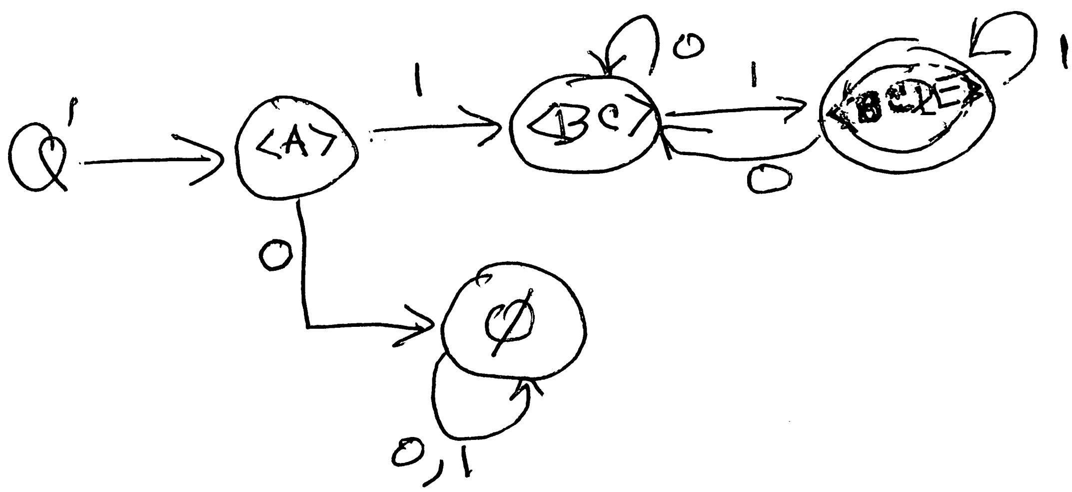
10. 

STATE	A	B	C	D	E
λ-CLOSURE	{A}	{B,C}	{C}	{DE}	{E}

WEEK#2

STATE	A	B	C	D	E
λ-CLOSURE	{A}	{B,C}	{C}	{D,E}	{E}





WHAT IS ASSOCIATED REG. EXPR.

$$10^*1(1^*+0^+1)^*$$

BUT HOW TO DO IN GENERAL?

OH, AND $10^*1(1^*+0^+1)^*$

$$=10^*1(1+0^+1)^*$$

$$=10^*1(0^*1)^*$$

$$=1(0^*1)^+$$

WEEK #2

1. REGULAR EXPRESSIONS

PRIMITIVE REG. EXPR. OVER Σ

EXPRESSION	SET
$\emptyset$	$\emptyset$
λ	$\{\lambda\}$
$a,\ a \in \Sigma$	$\{a\}$

CLOSURE OF EXPRESSIONS R, S DENOTING SETS R, S

$R+S$	$R \cup S = \{x \mid x \in R \text{ or } x \in S\}$
$R \cdot S$	$R \cdot S = \{xy \mid x \in R \text{ and } y \in S\}$
R^*	$R^* = \{\lambda\} \cup R \cup R^2 \cup \cdots$

FOR CONVENIENCE WE ALSO ALLOW

R^+	$R^+ = R \cup R^2 \cup R^3 \cup \cdots$

PRECEDENCE

$* > \cdot > +$

$\cdot$ AND $+$ ARE ASSOCIATED LEFT TO RIGHT

$*$ IS NON-ASSOCIATIVE OR JUST REALIZE $(R^*)^* = R^*$

PARENTHESES ARE USED TO OVERRIDE DEFAULT PARSE OR FOR READABILITY

2. REGULAR SETS

EVERY <u>REGULAR EXPRESSION</u> DENOTES A <u>REGULAR SET</u>

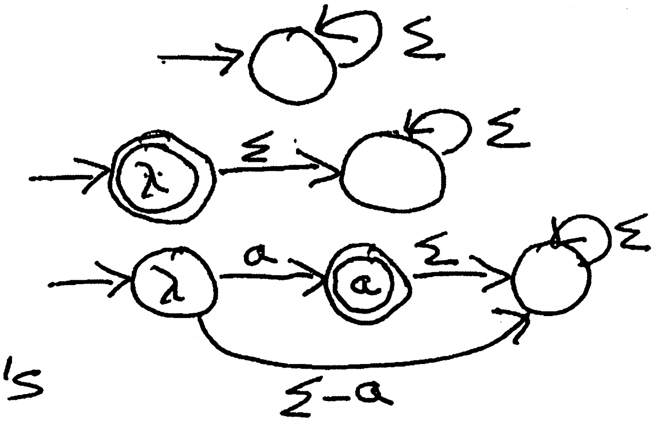
2. EVERY REGULAR SET IS A REGULAR LANG.

REGULAR EXPRESSION | DFA

∅

λ

a



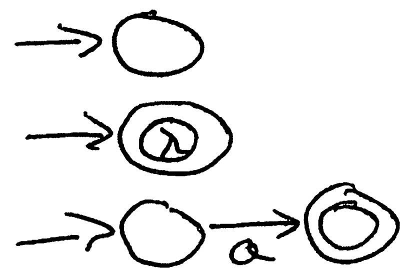
OR COULD DO NFA'S

REG EXP | NFA

∅

λ

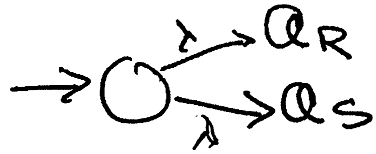
a



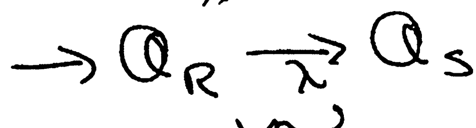
AS REGULAR LANGUAGES CLOSED UNDER ∪, ∘, *

THEN GET ALL REG. SETS AS REG. LANGUAGES

R+S

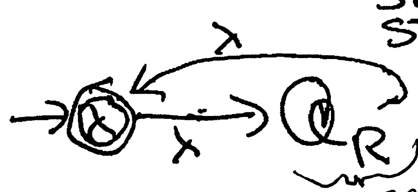


R∘S



FROM EACH FINAL STATE OF Q_R TO START STATE OF Q_S

R*



FROM EACH FINAL STATE OF Q_R TO START STATE OF ALTERED AUTOMATON

WEEK #2

(B)

3. EVERY REGULAR LANGUAGES IS A REG. SET

a. APPROACH #1: R_{ij}^{k} SETS (REG. EXPR.)

LET $L = \mathcal{L}(Q)$ WHERE

$Q = (Q, \Sigma, \delta, q_1, F)$, $Q = \{q_1, \ldots, q_n\}$

WANT TO BUILD EXPRESSIONS

R_{ij}^{k}, WHERE $1 \le i, j \le n$; $0 \le k \le n$

[EXAMPLE ON 88/89]

SEMANTICALLY,

$R_{ij}^{k} = \{ w \mid \delta^*(q_i, w) = q_j$ AND PATH FROM q_i TO q_j INVOLVES NO STATE WITH INDEX $> k \}$

ACTUALLY R_{ij}^{k} IS REG. EXPR FOR THIS SET

BASIS: $R_{ij}^{0} = \emptyset$ IF $i \ne j$ AND $\delta(q_i, a) \ne q_j \; \forall a \in \Sigma$

$R_{ij}^{0} = a$ IF $\delta(q_i, a) = q_j$ AND $i \ne j$

$R_{ij}^{0} = \lambda$ IF $i = j$ AND $\delta(q_i, a) \ne q_j \; \forall a \in \Sigma$

$R_{ij}^{0} = \lambda + a$ IF $i = j$ AND $\delta(q_i, a) = q_i$

INDUCTIVE HYP: ASSUME R_{ij}^{m} REG EXPR (SET) $0 \le m \le k$, $1 \le i, j \le n$

INDUCTIVE STEP:

$$R_{ij}^{k+1} = R_{ij}^{k} + R_{i\,k+1}^{k} \left(R_{k+1,k+1}^{k}\right)^* R_{k+1,j}^{k}$$

$$\mathcal{L}(Q) = +_{q_f \in F} R_{1f}^{n}$$

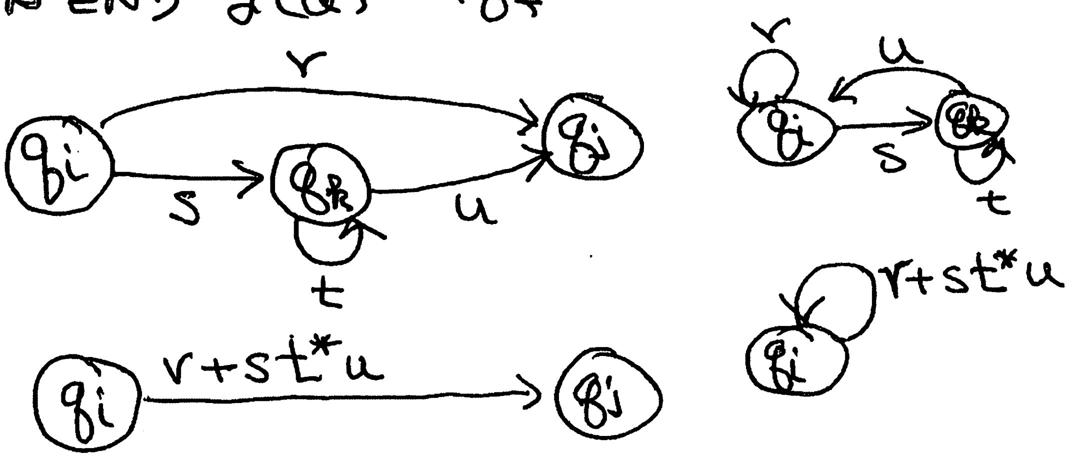
3. b. STATE RIPPING

LET $L = \mathcal{L}(\mathcal{Q})$ WHERE
$\mathcal{Q} = (Q, \Sigma, \delta, q_1, F)$, $Q = \{q_1, \ldots, q_n\}$

CONSTRUCT
$\mathcal{Q}' = (Q \cup \{q_0, q_f\}, \Sigma, \delta', q_0, \{q_f\})$ $q_0, q_f \notin Q$
$\delta'(q_0, \lambda) = \{q_1\}$ $\delta'(q_j, \lambda) \supseteq \{q_f\}$ $\forall q_j \in F$
r_{ij} INITIALLY IS REG. EXP THAT DIRECTLY GOES FROM q_i TO q_j
WE WILL COLLAPSE STATES IN Q, EXCEPT FOR q_0 + q_f. AS WE RIP A STATE q_k OUT, WE WILL EXTEND EXPRESSION r_{ij} BY ARCS THAT LED INTO q_k FROM q_i AND, TRANSITIVELY TO q_j, BY AN EXPR. THAT REPRESENTS $r_{ik} r_{kk}^* r_{kj} + r_{ij}$

IN END $\mathcal{L}(\mathcal{Q}) = r_{0f}$

[EXAMPLE ON 92-93]

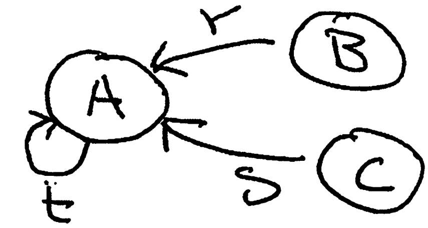


OF COURSE MUST DO FOR ALL PAIRS

WEEK #2

3.C. REGULAR EQUATIONS

IF



THEN $A = Br + Cs + At$

IF A IS START STATE THEN

$$A = \lambda + Br + Cs + At$$

OF COURSE, THERE IS A TERM FOR EVERY STATE THAT HAS AN ARC LEADING INTO A.

DO THIS FOR ALL STATES AND HAVE r EQUATIONS IN n UNKNOWNS

THAT SEEMS EASY BUT SOME MAY BE RECURSIVE IF AUTOMATA HAS LOOPS

WE CLAIM THAT IF $\lambda \notin P$ THEN

$$R = Q + RP$$

HAS UNIQUE SOLUTION

$$R = QP^*$$

3 C. CONTINUED

(1) QP^* IS A SOLUTION

$$R = Q + RP = Q + QP^*P = Q + QP^+ = QP^*$$

(2) ANY SOLUTION IS CONTAINED IN QP^*

$$R = Q + RP = Q + (Q+RP)P = Q + QP + RP^2$$
$$= Q + QP + (Q+RP)P^2 = Q + QP + QP^2 + RP^3$$
$$\cdots$$
$$= Q + QP + QP^2 + QP^3 + \cdots + QP^k + RP^{k+1}, \; k \geq 0$$
$$= Q(\lambda + P + P^2 + \cdots + P^k) + RP^{k+1}, \; k \geq 0$$

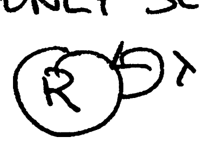
SINCE $\lambda \notin P$ THEN FOR ANY $w \in R$, $|w| = k$, $w \notin RP^{k+1}$ AND SO $w \in Q(\lambda + P + P^2 + \cdots + P^k) \subseteq QP^*$

THUS $R \subseteq QP^*$

(3) AS $QP^* \subseteq R$ AND $R \subseteq QP^*$ THEN $R = QP^*$

AGAIN PROVIDED $\lambda \notin P$

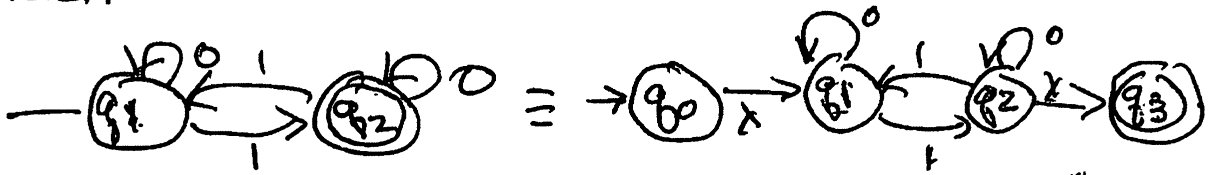
NOTE: WE WOULD ONLY SEE $\lambda \in P$ IF WE HAD



WHICH IS USELESS

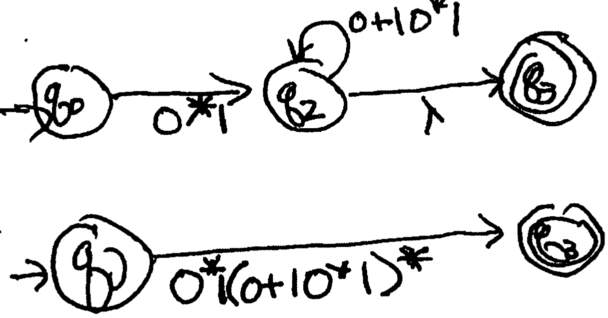
[EXAMPLES OF 98/99]

4. REALLY SIMPLE EXAMPLE



$$R_{11}^{0} = \lambda + 0 \qquad R_{22}^{0} = 0$$

$$R_{12}^{0} = 1 \qquad R_{21}^{0} = 1$$



$$R_{11}^{1} = (\lambda+0) + (\lambda+0)(\lambda+0)^*(\lambda+0) = 0^*$$

$$R_{12}^{1} = 1 + (\lambda+0)(\lambda+0)^* 1 = 0^* 1$$

$$R_{22}^{1} = 0 + 1(\lambda+0)^* 1 = 0 + 10^* 1$$

$$R_{21}^{1} = 1 + 1(\lambda+0)^*(\lambda+0) = 10^*$$

$$R_{12}^{2} = 0^*1 + 0^*1(0+10^*1)^*(0+10^*1)$$
$$= 0^*1(0+10^*1)^*$$

$$Q_1 = \lambda + Q_2 1 + Q_1 0$$

$$Q_2 = Q_1 1 + Q_2 0$$

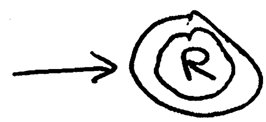
$$Q_1 = (\lambda + Q_2 1)0^*$$

$$Q_2 = (\lambda + Q_2 1)0^* 1 + Q_2 0$$
$$= 0^*1 + Q_2(0+10^*1)$$
$$= 0^*1(0+10^*1)^*$$

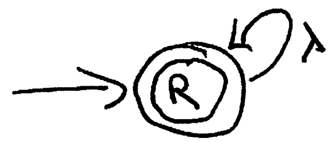
VERY RARE THAT ALL THREE REG EXPR IN SAME FORM. NOTES HAVE BETTER EXAMPLE.

WEEK #

WHAT COULD HAPPEN IF ALLOWED λ-TRANSITIONS IN REG. EQ. APPROACH

CONSIDER  OVER $\Sigma = \{a\}$

$R = \lambda$ IS ONLY EQUATION AND ALL IS WELL

NOW CONSIDER  OVER $\Sigma = \{a\}$

$$R = \lambda + R \cdot \lambda$$

CLEARLY $R = \lambda$ IS <u>A</u> SOLUTION

$$R = \lambda + \lambda \cdot \lambda = \lambda$$

BUT SO IS $R = \lambda + A$, WHERE A IS ANY LANGUAGE OVER $\{a\}$

$$R = \lambda + (\lambda + A) \cdot \lambda = \lambda + A$$

BUT THEN THERE ARE AN UNCOUNTABLY INFINITE NUMBER OF SOLUTIONS!