REGULAR EXPRESSIONS OVER ALPHABET
ASSUME Σ

SYNTAX r	SEMANTICS $L(r)$
ϵ	$L(\epsilon) = \{\varepsilon\}$
ϕ	$L(\phi) = \phi$
$a \in \Sigma$ a	$L(a) = \{a\}$
$(r_1 + r_2)$	$L(r_1+r_2) = L(r_1) \cup L(r_2)$
$(r_1 r_2)$	$L(r_1 r_2) = L(r_1) L(r_2)$
r^*	$L(r^*) = (L(r))^*$

ANALOGY

POLYNOMIALS SYNTACTIC OBJECT THAT DENOTES FUNCTIONS

FINITE REPRESENTATION
OF
INFINITE OBJECT

$$L((0+1)^*) = (L(0+1))^*$$
$$= (L(0) \cup L(1))^*$$
$$= (\{0\} \cup \{1\})^*$$
$$= \{0,1\}^*$$

SIPSER EXAMPLES

page 65
EX 1.53

$0^*1 0^*$ CONTAINS SINGLE 1

$(0+1)^* 1 (0+1)^*$ CONTAINS AT LEAST ONE 1

$(0+1)^* 001 (0+1)^*$ CONTAINS 001 AS SUBSTRING

$((0+1)(0+1))^*$ EVEN LENGTH

$(0+1)((0+1)(0+1))^*$ ODD LENGTH

$1^* (0 \overbrace{1 1^*}^{1^+})^*$ #4 OF EXAMPLE 1.53 pg 65

EVERY 0 FOLLOWED BY AT LEAST ONE 1

$\emptyset^* = \{\epsilon\}$

BEGINS WITH 0 AND ENDS WITH 11

$$0(0+1)^*11$$

CONTAINS AT LEAST TWO 1s

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

CONTAINS 111

$$(0+1)^*111(0+1)^*$$

d. DO NOT END WITH 01

00
01
10
11 ✓

$$\varepsilon + 1 + (0+1)^* 0 + (0+1)^* 11$$

ANY THING ENDING IN 0

ANYTHING ENDING IN 11

CONTAINS BOTH 01 AND 10 AS DISTINCT SUBSTRINGS

$$(0+1)^* 01 (0+1)^* 10 (0+1)^* + (0+1)^* 10 (0+1)^* 01 (0+1)^*$$

$L_3 = \{ w \mid$ 3 RD SYMBOL FROM THE END IS A 0 $\}$

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

3.9 a NOT CONTAINING 00

$$(1+01)^*(\varepsilon+0) \quad \text{OR} \quad (\varepsilon+0)(1+10)^*$$

---

g <u>NO MORE THAN ONE OCCURENCE OF 00</u>

NOTE $(1+01)^*$ DOESN'T CONTAIN 00 AND DOESN'T END 0

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

EXACTLY ONE OCCURENCE OF 00

$$(1+01)^*00(1+10)^*$$

AND SO R.E. IS

$$(1+01)^*(\varepsilon+0) + (1+01)^*00(1+10)^*$$

1.6 f page 84

f

DOES NOT CONTAIN SUBSTRING

110

~~000~~

~~001~~

~~1^*~~

~~$0^*(100^*)^*1^*$~~

~~00~~

~~+010~~

IF THERE ARE EVER TWO 1s REST OF STRING MUST BE 1s

$$0^*(\underbrace{100^*}_{\text{substrings of 1 followed by one or more 0s}})^*1^*$$

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

$\alpha \equiv \beta$ means $L(\alpha) = L(\beta)$

Let α, β, γ be regular expressions

$$\alpha + (\beta + \gamma) \equiv (\alpha + \beta) + \gamma$$
$$\alpha + \beta = \beta + \alpha$$
$$\alpha + \emptyset = \alpha$$
$$\alpha + \alpha = \alpha$$
$$\alpha(\beta\gamma) = (\alpha\beta)\gamma$$
$$\epsilon\alpha = \alpha\epsilon = \alpha$$
$$\alpha(\beta + \gamma) = \alpha\beta + \alpha\gamma$$
$$(\alpha + \beta)\gamma = \alpha\gamma + \beta\gamma$$
$$\emptyset\alpha = \alpha\emptyset = \emptyset$$
$$\epsilon + \alpha\alpha^* = \alpha^*$$
$$\epsilon + \alpha^*\alpha = \alpha^*$$

- - - - - - - - - - - -

$(\alpha\beta)^*\alpha = \alpha(\beta\alpha)^*$ SHIFT RULE

$(\alpha^*\beta)^*\alpha^* = (\alpha + \beta)^*$ DENESTING RULE

$$\alpha^*(\beta\alpha^*)^* = (\alpha + \beta)^*$$
$$(\epsilon + \alpha)^* = \alpha^*$$
$$\alpha\alpha^* = \alpha^*\alpha$$
$$\alpha^*\alpha^* = \alpha^*$$
$$\alpha^{**} = \alpha^*$$

KLEENE'S THEOREM

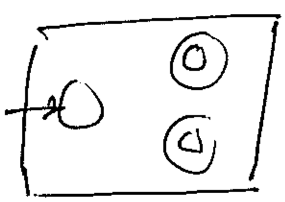
PART I

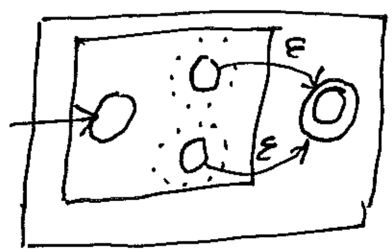
THM FOR EVERY REGULAR EXPRESSION THERE IS AN EQUIVALENT ϵ-NFA.

PROOF: USE FACT THAT R.E.s ARE RECURSIVELY DEFINED.

AND ~~use~~ USE CONSTRUCTIVE INDUCTION TO GET AN EQUIVALENT ϵ-NFA.

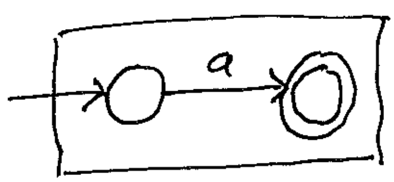
FOR CONVENIENCE ALL OUR FAs WILL HAVE A UNIQUE ACCEPTING STATE.
ANY ϵ-NFA CAN BE SO TRANSFORMED, SIMPLY ADD THE NEW UNIQUE ACCEPTING STATE AND ϵ MOVES FROM ALL THE OLD ACCEPTING STATE AND THEN MAKE THE OLD ACCEPTING STATES REJECTING STATES.



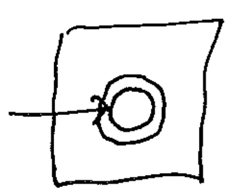


BASE $M = 1$ LENGTH OF R.E.

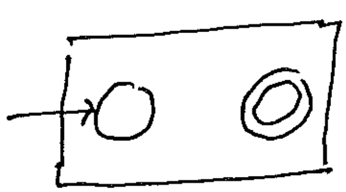
a WHERE $a \in \Sigma$



ε



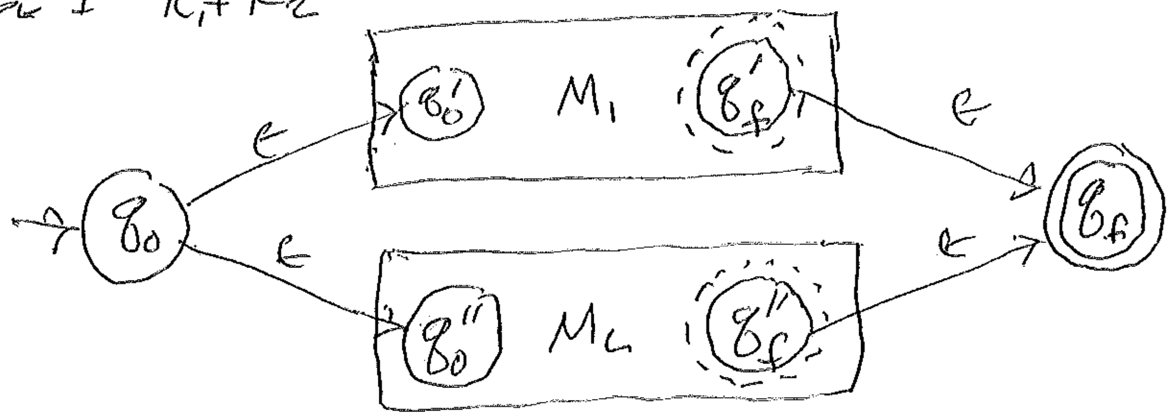
$\emptyset$



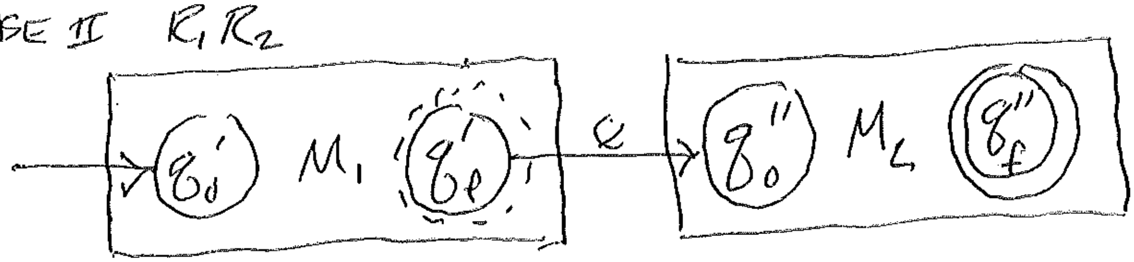
INDUCTIVE STEP IF R IS NOT A BASE CASE THEN R MUST BE $R_1 + R_2$ OR $R_1 R_2$ OR R_1^* WHERE R_1, R_2 ARE REG. EXPS WHOSE LENGTHS IS LESS THAN THE LENGTH OF R.

R_1 AND R_2 HAVE EQUIVALENT ε-NFAs BY THE I.H. SAY THEY ARE M_1 AND M_2

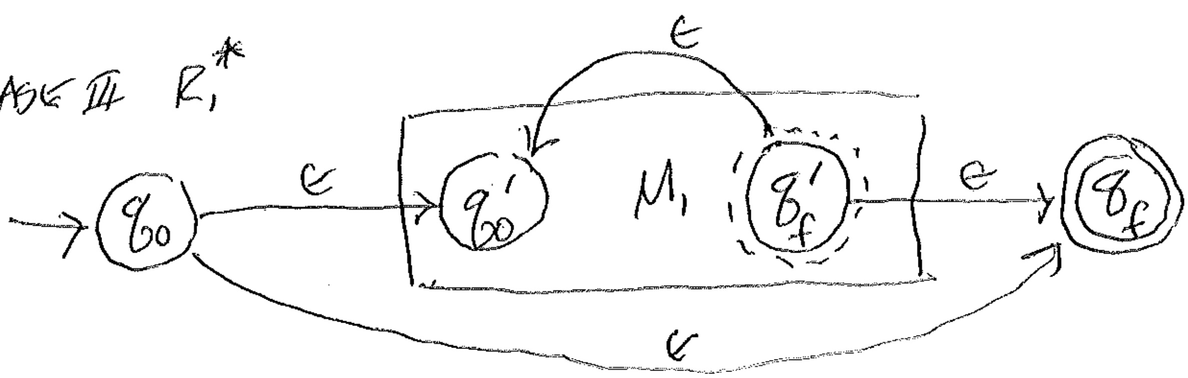
CASE I $R_1 + R_2$



CASE II $R_1 R_2$



CASE III R_1^*



SO WE HAVE BUILT ε-NFAs FOR ALL ~~THE~~ THREE CASES. AND THE INDUCTIVE STEP PROOF IS COMPLETED.

Q.E.D.

KLEENE'S THM

PART II.

GIVEN M, DFA CAN CONSTRUCT RE THAT DENOTES L(M)

CAN PICK ANY TYPE TO WORK WITH DFA, NFA, ϵ-NFA
SINCE ALL EQUIVALENT. DFA IS BEST FOR THIS.

INDUCTIVE PROOF - MECH. CONSTRUCTION.

NEED GENERAL WAY TO TALK ABOUT PATHS

⇨ AIM TO DESCRIBE ALL ACCEPTING PATHS TROUGH M WITH R.E.

LOOPS ⇒ PATHS CAN BE ARBITRARILY LARGE
MOST MACHINES HAVE ∞ # OF ACCEPTING PATHS.

(INDUCTING ON LENGTH OR # OF PATHS, DOES NOT WORK.)
BECAUSE ∞ UNION OF r.e.s IS NOT AN r.e.

NUMBER OF STATES IS CONSTANT FOR A MACH.
REGARDLESS OF LENGTH OR # OF PATHS.

GIVEN DFA $M = (Q, \Sigma, \delta, q_0, F)$, THERE IS A REG. EXPRESSION r S.T. $L(r) = L(M)$.

IDEA OF PROOF:

DECOMPOSE $L(M)$ INTO "SIMPLER" LANGUAGES.

SHOW EACH SIMPLER LANGUAGE CAN BE DENOTED BY A REG. EXPRESSION.

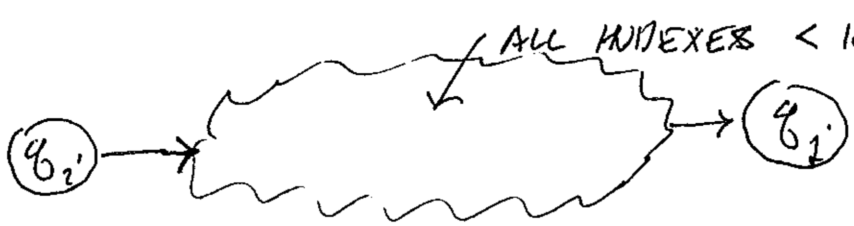
WRITE $L(M)$ AS THE FINITE UNION OF THE SIMPLER LANGUAGES. THIS CAN BE EXPRESSED USING "+".

PROOF

LET $Q = \{q_0, q_1, \ldots, q_n\}$

IDEA: USE THE ACCEPTING STATES (F) TO DETERMINE THE SIMPLER LANGUAGES.

LET $L_{ij}^{k} = \{ w \in \Sigma^* \mid \delta^*(q_i, w) = q_j$ AND ALL INTERMEDIATE STATES HAVE INDEXES LESS THAN $k \}$



NOTE: i AND/OR j CAN BE $\geq k$, THE RESTRICTION IS ON INTERMEDIATE STATES.

BY DEF. $L(M) = \{ w \mid \delta^*(q_0, w) \in F \}$

NOW WE CAN DECOMPOSE THIS TO:

$$L(M) = \bigcup_{q_j \in F} L_{0j}^{m+1}$$

WE NEED TO SHOW FOR EACH $q_j \in F$ THERE IS A REG. EXPRESSION DENOTING L_{0j}^{m+1}

CAN NOT USE INDUCTION DIRECTLY ON m BECAUSE L_{0j}^{m} DOES NOT IN GENERAL COME FROM SOME L_{0j}^{m-1}.

SOLUTION PROVE SOMETHING MORE GENERAL STRONGER (INDUCTION LOADING) THAT OUR DESIRED RESULT IS A SPECIAL CASE OF.

FOR ALL K, FOR ALL i, j
L_{ij}^{K} IS DENOTED BY A REG. EXPRESSION.

PROVE BY INDUCTION ON K.

r_{ij}^{K}

BASE CASE $k = 0$

PROVE FOR ALL i, j L^0_{ij} IS DENOTED BY A r.e.

$k = 0$ MEANS NO INTERMEDIATE STATES, I.E., DIRECT TRANSITIONS

FOR $i \neq j$ $L^0_{ij} = \{ a \in \Sigma \mid \delta(q_i, a) = q_j \}$

$r^0_{ij} = \sum_{a \in L^0_{ij}} a$

FOR $i = j$ WE HAVE ε

NOTE IF $L^0_{ij} = \emptyset$ THEN $r^0_{ij} = \emptyset$

INDUCTIVE STEP

~~ASSUMING FOR~~

GIVEN FOR

L^k_{ij} WE HAVE r.e. r^k_{ij} S.T. $L(r^k_{ij}) = L^k_{ij}$

PROVE THAT FOR

$L^{k+1}_{ij} = \{ w \mid \delta^*(q_i, w) = q_j$ AND ALL INTERMEDIATE STATES HAVE INDEXES LESS THAN $k+1 \}$

WE CAN WRITE A r.e.

TWO CASES FOR L_{ij}^{K+1}

1). PATHS THAT DO NOT INCLUDE K AND 2.) PATHS THAT DO.

CASE 1. L_{ij}^{K}

CASE 2. PATHS CAN BE DECOMPOSED INTO THREE CLASSES

I. GO FROM q_i TO FIRST OCCURANCE OF q_K

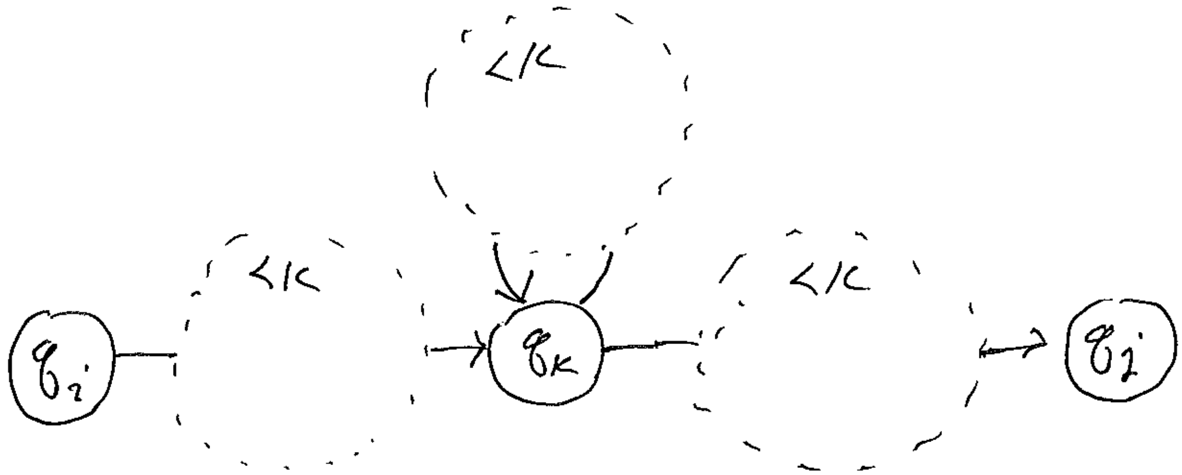
L_{iK}^{K}

II. GO FROM q_K TO q_K

L_{KK}^{K}

III GO FROM q_K TO q_j

L_{Kj}^{K}



$$L^{k}_{ik}\left(L^{k}_{kk}\right)^{*}L^{k}_{kj}$$

$$L^{k+1}_{ij} = L^{k}_{ij} \cup L^{k}_{ik}\left(L^{k}_{kk}\right)^{*}L^{k}_{kj}$$

r.e. FOR $L^{k+1}_{ij} = r^{k}_{ij} + r^{k}_{ik}\left(r^{k}_{kk}\right)^{*}r^{k}_{kj}$

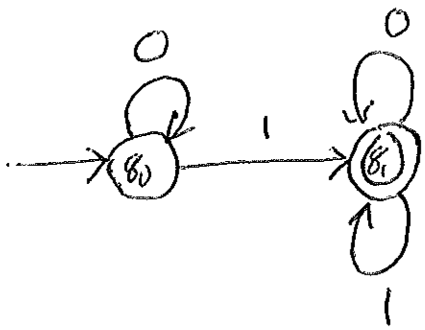
Q.E.D.

REG. EXPRESSION FOR WHOLE MACHINE

$$\sum_{j \in F} r_{0j}^{n+1}$$

i, j	K = 0	1	2	$\cdots$	$m+1$
0 0					
0 1					
$\vdots$					
0 m					
$\vdots$					
m 0					
m 1					
$\vdots$					
m m					

0



	0	1	2
0 0	$\epsilon + 0$	0^*	0^*
0 1	1	0^*1	$0^*1(0+1)^*$
1 0	$\emptyset$	$\emptyset$	
1 1	$\epsilon + 0 + 1$	$\epsilon + 0 + 1$	

$$r_{00}^1 = r_{00}^0 + r_{00}^0 (r_{00}^0)^* r_{00}^0$$

$$\underbrace{\epsilon+0 + (\epsilon+0)(\epsilon+0)^*(\epsilon+0)}_{0^*}$$

$$r_{01}^1 = r_{01}^0 + r_{00}^0 (r_{00}^0)^* r_{01}^0$$

$$1 + (\epsilon+0)(\epsilon+0)^* 1$$

$$1 + 0^* 1 = 0^* 1$$

$$r_{10}^1 = r_{10}^0 + r_{10}^0 (r_{00}^0)^* r_{00}^0$$

$$\emptyset + \emptyset = \emptyset$$

$$r_{11}^1 = r_{11}^0 + r_{10}^0 (r_{00}^0)^* r_{01}^0$$

$$(\epsilon+0+1) + \emptyset = \epsilon+0+1$$

$$r^2_{00} = r^1_{00} + r^1_{01}(r^1_{11})^* r^1_{10}$$

$$= 0^* + \cancel{0^*1(\epsilon+0+1)^*}\,\emptyset$$

$$= 0^*$$

$$\checkmark\; r^2_{01} = r^1_{01} + r^1_{01}(r^1_{11})^* r^1_{11}$$

$$0^*1 + 0^*1(\epsilon+0+1)^*(\epsilon+0+1)$$

$$0^*1 + 0^*1\,(0+1)^*$$

$$r^2_{10} = r^1_{10} + r^1_{11}(r^1_{11})\, r^1_{10}$$

$$= \emptyset \qquad\qquad \emptyset = \emptyset$$

$$r^2_{11} = r^1_{11} + r^1_{11}(r^1_{11})^* r^1_{11} = r^{1\,*}_{11}$$

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