COMPILER CONSTRUCTION

LAB 2

GROUP MEMBERS

109459 - Imali Susan Lung'aho

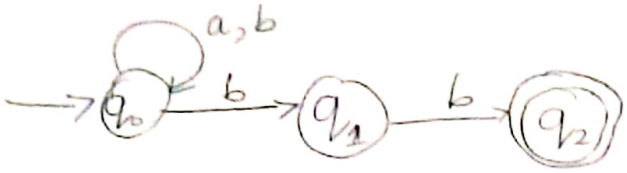
123038 - Ng’ang’a Susan Wanjiku

115238 - Brian Munene Muriithi

118242 - Brian Muriithi Koome

Question One

Conversion of NFA to DFA

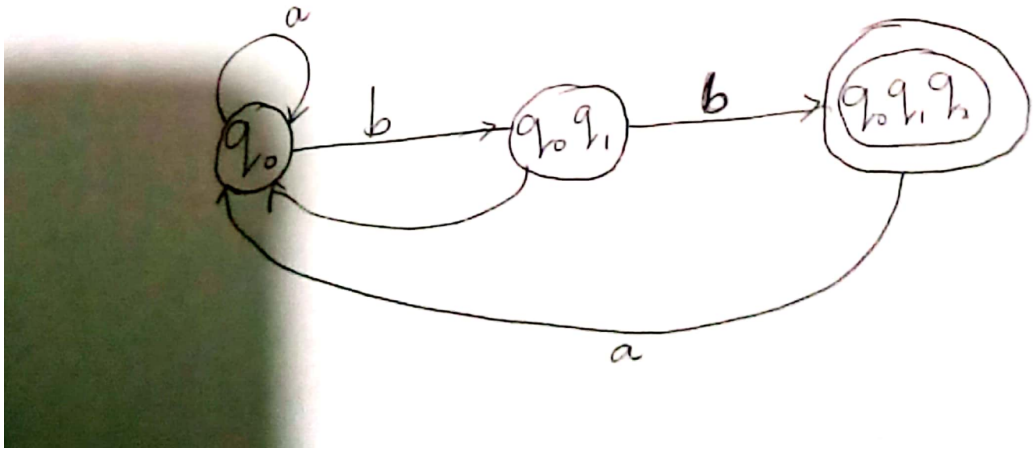


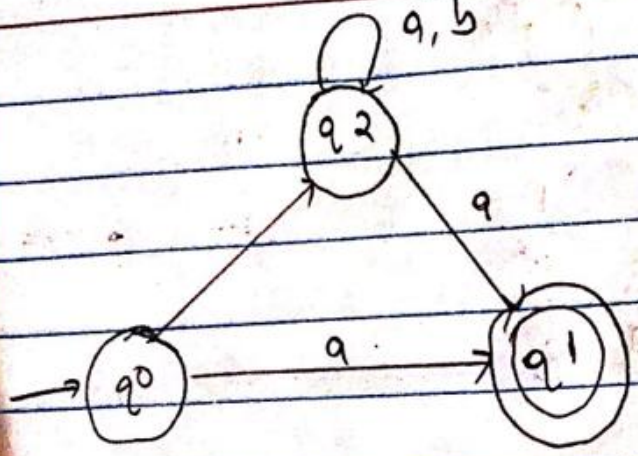
NFA Transition Table

	a	b
→ q_0	q_0	$\{q_0, q_1\}$
q_1	—	q_2
q_2	—	—

DFA Transition Table.

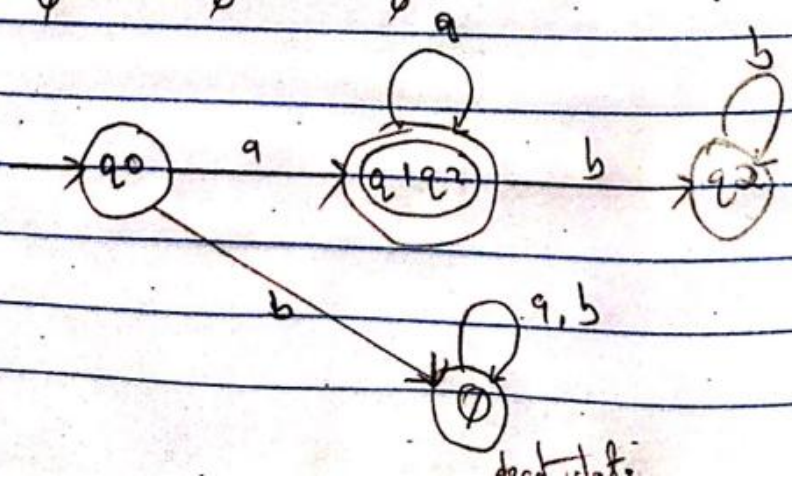
δ	a	b
→ q_0	q_0	$[q_0 q_1]$
$[q_0 q_1]$	$[q_0]$	$[q_0 q_1 q_2]$
$[q_0 q_1 q_2]$	$[q_0]$	$[q_0 q_1 q_2]$





	a	b
q0	q1,q2	∅
q1	∅	∅
q2	q1,q2	q2

	a	b
q0	q1q2	∅
q1 q2	q1q2	q2
q2	q1q2	q2
∅	∅	∅



Question Two

You invoke flex on a .l file and it creates lex.yy.c, a source file containing a pile of unrecognizable C code that implements an FA encoding all your rules and including the code for the actions you specified. The file provides an external function yylex() that will scan one token. You compile that C file normally, link with the lex library, and you have built a scanner! The scanner reads from stdin and writes to stdout by default.whenever the scanner reads an input sequence that matches a pattern, it executes the action to process it. As the scanner reads characters from the file, it will gather them until it forms the longest possible match for any of the available patterns. If two or more patterns match an equally long sequence, the pattern listed first in the file is used.