Aranich, Marcel - 251453
Quera, Clàudia - 231197
Villarino, Jorge - 231351
Solans, Arnau - 216530
Garcia, David - 251587
Prat, Ariadna - 251281

1.Alphabet definition

Following the specification of the lexemes described in the statement, we have derived the alphabet:

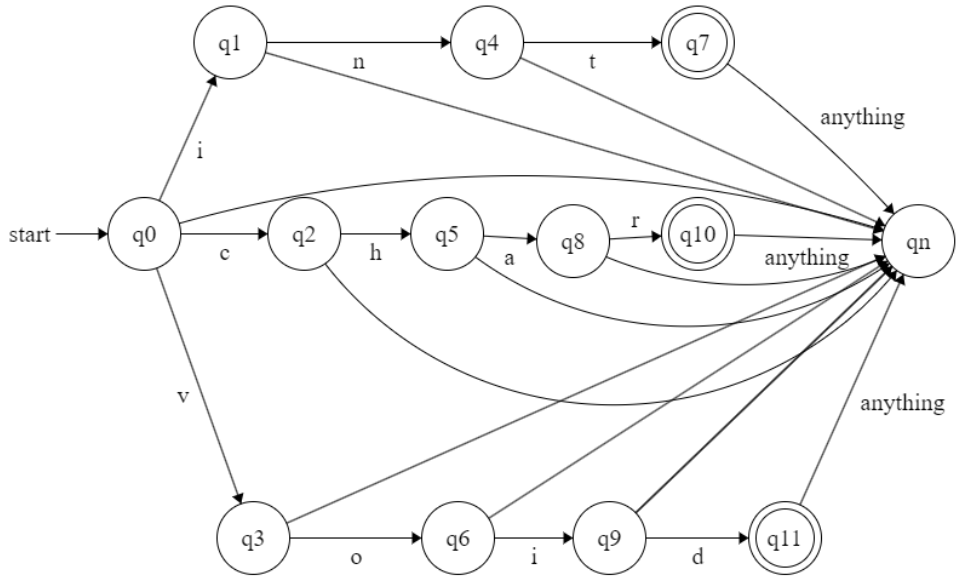
Σ = { **a-z**, **A-Z**, **0-9**, ‘**,**’, ‘**;**’, ‘**(**‘, ‘**)**’, ‘**[**‘, ‘**]**’, ‘**{**‘, ‘**}**’, =, +, >, *, ‘ **“** ‘}

2.DFA definitions

DFA for types:

- Accepts “int”, “char” and “void”:

	i	n	t	c	h	a	r	v	o	d	other
S	q1	qn	qn	q2	qn	qn	qn	q3	qn	qn	qn
q1	qn	q4	qn	qn	qn	qn	qn	qn	qn	qn	qn
q2	qn	qn	qn	qn	q5	qn	qn	qn	qn	qn	qn
q3	qn	qn	qn	qn	qn	qn	qn	qn	q6	qn	qn
q4*	qn	qn	q7	qn	qn	qn	qn	qn	qn	qn	qn
q5	qn	qn	qn	qn	qn	q8	qn	qn	qn	qn	qn
q6	q9	qn	qn	qn	qn	qn	qn	qn	qn	qn	qn
q7*	qn	qn	qn	qn	qn	qn	qn	qn	qn	qn	qn
q8	qn	qn	qn	qn	qn	qn	q10	qn	qn	qn	qn
q9	qn	qn	qn	qn	qn	qn	qn	qn	qn	q11	qn
q10*	qn	qn	qn	qn	qn	qn	qn	qn	qn	qn	qn
q11*	qn	qn	qn	qn	qn	qn	qn	qn	qn	qn	qn
qn	qn	qn	qn	qn	qn	qn	qn	qn	qn	qn	qn



(empty lines represent any other input)

DFA for identifiers

Rules for identifiers: only letters (both uppercase and lowercase) and numbers. Underscore (‘_’) is not allowed since it's not part of the alphabet. Also, the first character must NOT be a digit.

States/symbols	{0 - 9}	Uppercase U Lowercase	Anything else
0 (reject)	0	0	0
START	0	Accepting	0
Accepting	Accepting	Accepting	0

{0 - 9} = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

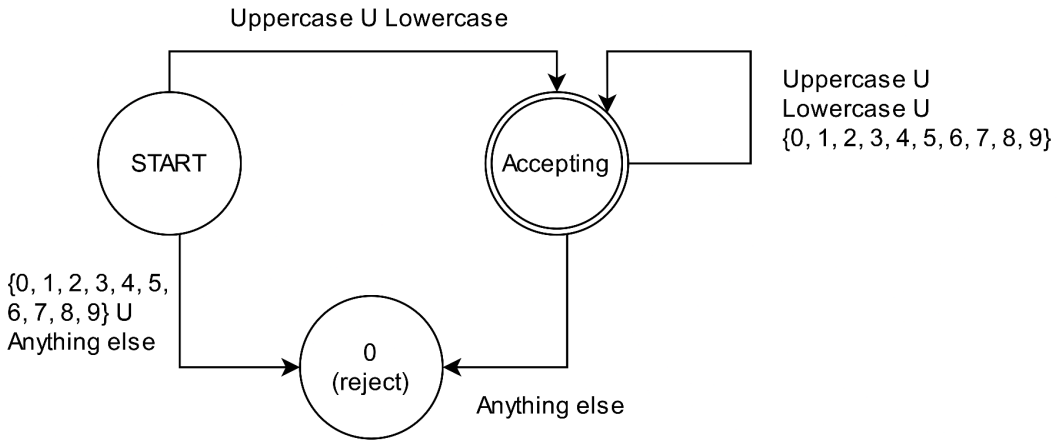
Uppercase = {A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z}

Lowercase = {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z}

Anything else is any other symbol that does not belong to {0 - 9} U Uppercase U Lowervase

0 represents the rejecting state. And START is the starting state.

Accepting is the only accepting state.

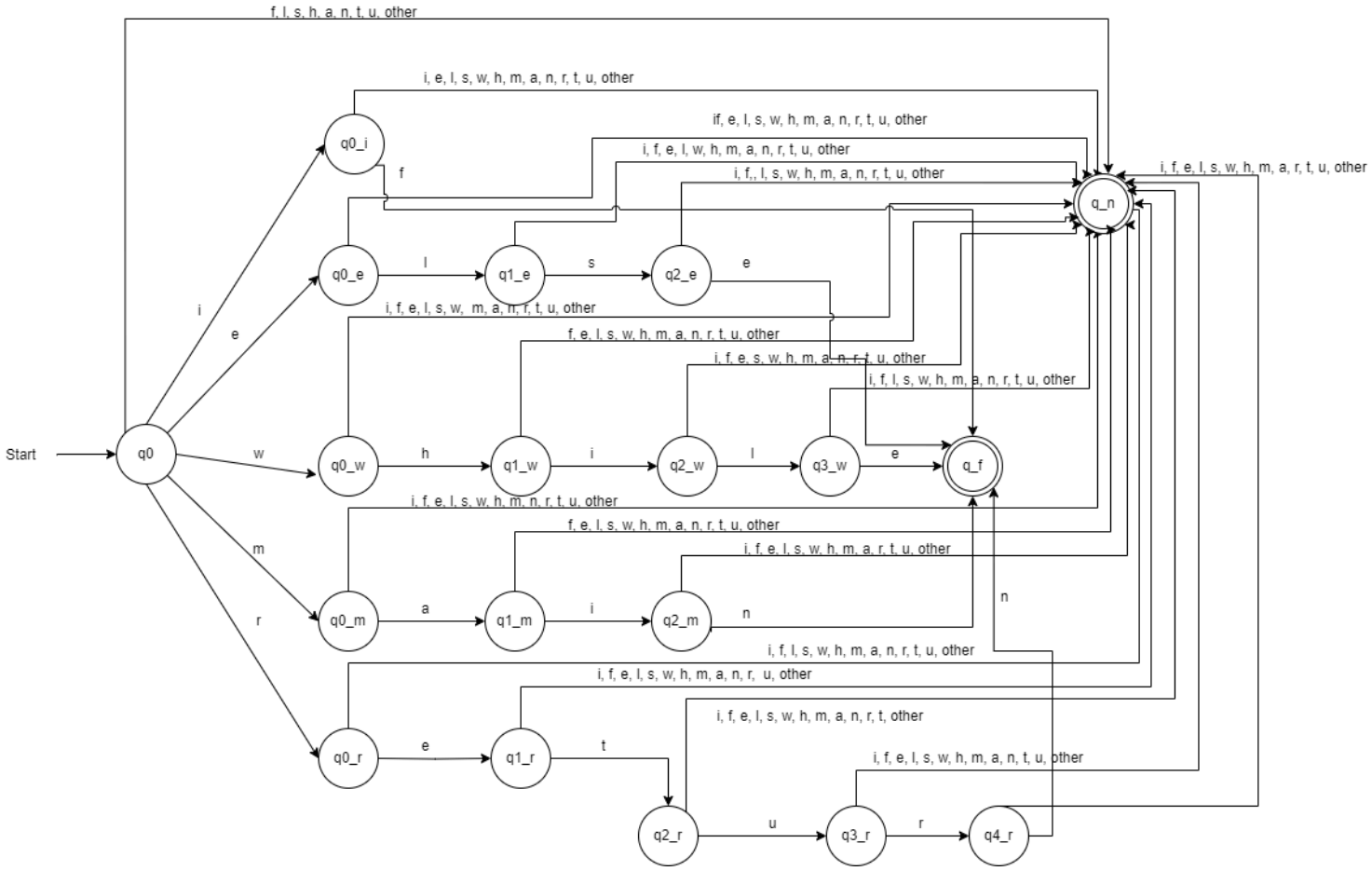


DFA Keywords: if, else, while, main, return

The starting state is q0, and the accepting states are q1_i, q3_e, q4_w, q3_m, and q5_r. When the current state is one of the previous states, the automata end, and the string is accepted and kept in a buffer. The state q_n is the rejecting state, when we are in this state, we reject the string. In this automata, we only accept the strings "if, else, while, main, return". This is the alphabet "i, f, e, l, s, w, h, m, a, n, r, t, u, other".

	i	**f**	**e**	**l**	**s**	**w**	**h**	**m**	**a**	**n**	**r**	**t**	**u**	**other**
q0	q0_i	q_n	q0_e	q_n	q_n	q0_w	q_n	q0_m	q_n	q_n	q0_r	q_n	q_n	q_n
q0_i	q_n	q_f	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n
q0_e	q_n	q_n	q_n	q1_e	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n
q1_e	q_n	q_n	q_n	q_n	q2_e	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n
q2_e	q_n	q_n	q_f	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n
q0_w	q_n	q_n	q_n	q_n	q_n	q_n	q1_w	q_n	q_n	q_n	q_n	q_n	q_n	q_n
q1_w	q2_w	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n
q2_w	q_n	q_n	q_n	q3_w	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n
q3_w	q_n	q_n	q_f	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n
q0_m	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q1_m	q_n	q_n	q_n	q_n	q_n
q1_m	q2_m	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n
q2_m	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_f	q_n	q_n	q_n	q_n

q0_r	q_n	q_n	q1_r	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n
q1_r	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q2_r	q_n	q_n
q2_r	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q3_r	q_n
q3_r	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q4_r	q_n	q_n	q_n
q4_r	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_f	q_n	q_n	q_n	q_n
q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n
q_f	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n	q_n



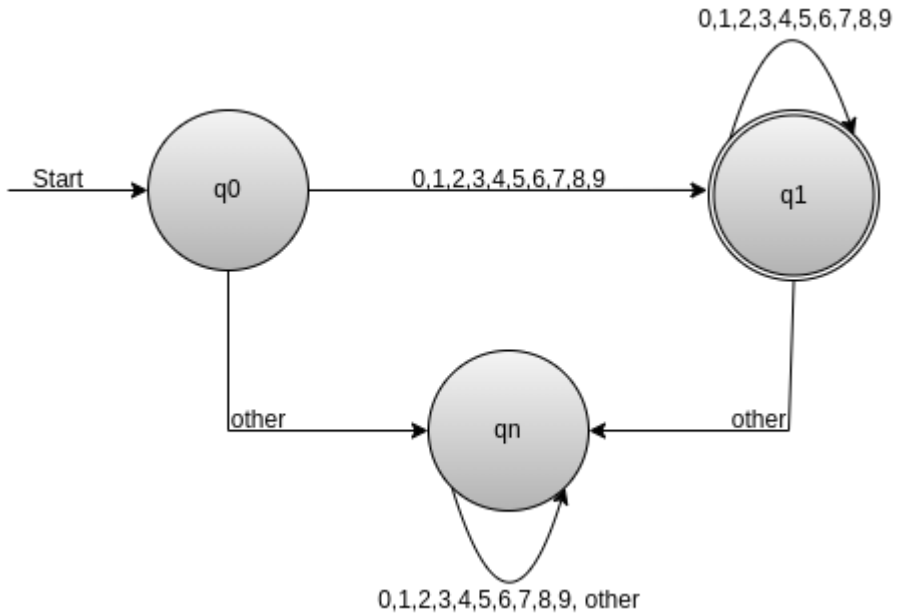
DFA for numbers {0-9}*:

q0 represents the starting state, any input being a number from 0 to 9 leads to state q1, any other input leads to state qn.

State qn is the rejecting state, indicating that the input read is not a number if the automata reaches this state, any input received keeps the automata in this state.

State q1 is the accepting state, it keeps looping itself when the input is a number between 0 to 9, if any other input is received it goes to state qn.

	0	1	2	3	4	5	6	7	8	9	other
q0	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	qn
q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	qn
qn	qn	qn	qn	qn	qn	qn	qn	qn	qn	qn	qn



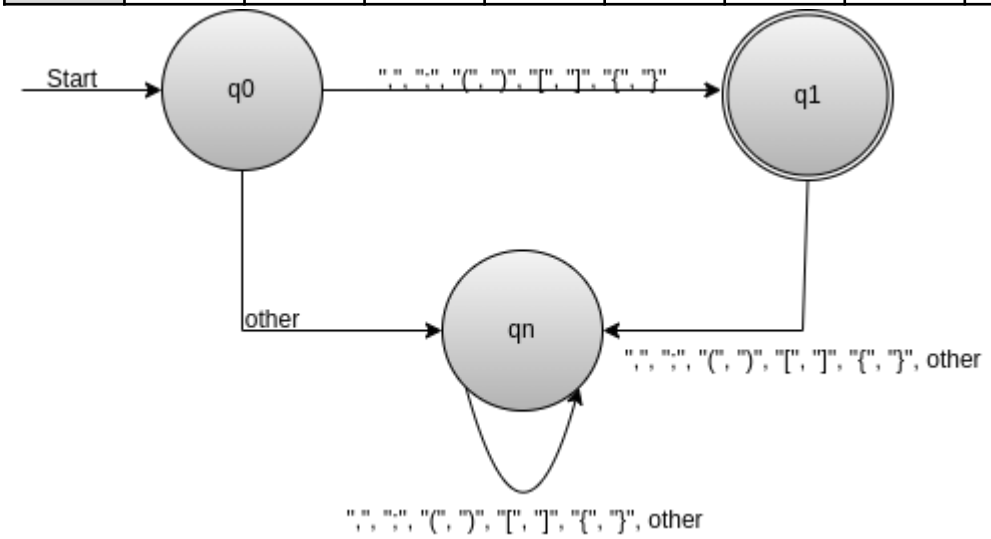
DFA for special characters (“;”, “,”, “[“, “]”, “{“ , “}”, “(“, “)”):

q0 represents the starting state, any input of “,”, “;”, “(“, “)”, “[“, “]”, “{“, “}” leads to state q1, any other input to state qn.

q1 is the accepting state, any input received kicks the automata out of this state to state qn.

qn is the rejecting state, indicating that the input read is not a special character, any input keeps the automata in this state.

	“(“	“)”	“[“	“]”	“{“	“}”	“,”	“;”	other
q0	q1	q1	q1	q1	q1	q1	q1	q1	qn
q1	qn	qn	qn	qn	qn	qn	qn	qn	qn
qn	qn	qn	qn	qn	qn	qn	qn	qn	qn



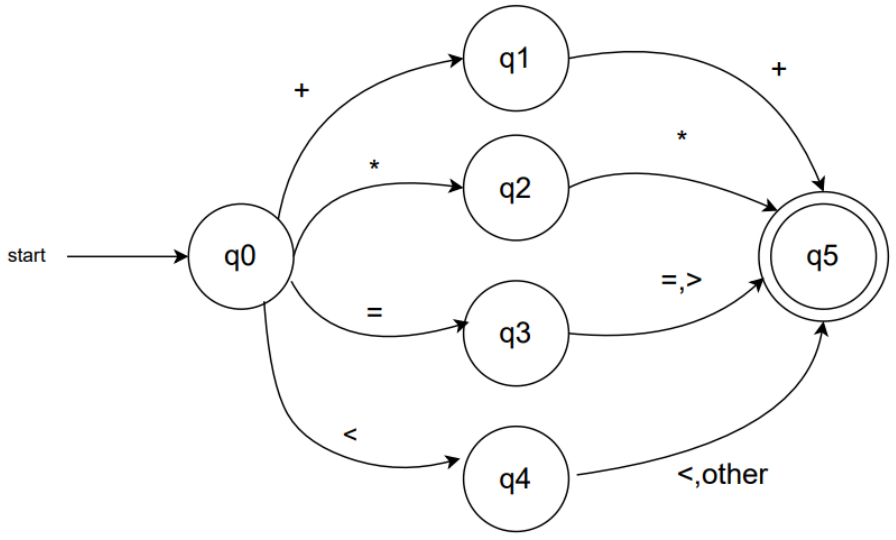
DFA for operators (+, *, =, >):

	+	*	=	>	<	other
q0	q1	q2	q3	-	q4	-
q1	q5	-	-	-	-	-
q2	-	q5	-	-	-	-
q3	-	-	q5	q5	-	-
q4	-	-	-	-	q5	q5
q5	-	-	-	-	-	-

	+	*	=	>	<	other
0 (reject)	0	0	0	0	0	0
START	q1	q1	q1	q1	q1	0
q1	0	0	0	0	0	0

//^accepta un únic operator i rebutja tota la resta. Si ens trobem "=<", ho identifiquem com a 2 tokens i que el parser fagi lu que toqui.

Where the, q1, q2, q3 and q4 states represent the accepting states for the +, *, =, < symbols, respectively, and q0 is the initial state. The q5 state is the accepting state, and indicates that the DFA identifies successfully an operator as valid, while the states marked as "-" indicate that the operator is not accepted as valid by the DFA.



DFA for literals (“ ”):

q0 represents the starting state, which changes to q1 with an input of “ (any other input leads to the rejecting state) and stays in q1 until an input “ is received, then changes to q2, the accepting state. Any additional input leads to the rejecting state, as the literal finishes with the second “.

	“	**other**
q0	q1	qn
q1	q2	q1
q2	qn	qn
qn (rejecting state)	qn	qn