Assignment

Construct a DFA that recognizes signed integers and decimals. The two lexeme groups can be exemplified as:

Item	Valid Expressions	Erroneous Expressions
Integers	34, 0, −12346	02, +67, −0
Decimals	3.14, 0.02, .47, −12.23, −0.002	02.45, 23. , +2.1, −0.00, 0.0, 0. , .0

Notice: As a decimal we allow .47 for 0.47 but not 23. for 23.0. At the same time, the only proper way to represent zero in this assignment is "0" (only integer, no signs or decimal points allowed).

1. Construct regular expressions p_1, p_2 for each one of the items above. You can use regular definitions, character classes (e.g. $d \rightarrow [1 - 9]$ and the shortcut operators ? and +). Hint: Use 0 and [1 − 9] as two separate entities rather than [0 − 9].
2. Use Thompson's construction (Algorithm 2) to construct two NFA:s N_1, N_2 corresponding to each one of the regular expressions p_1, p_2. Put these two NFA:s together to a single large NFA N having final states corresponding to integers and floats.
3. Use the subset construction algorithm (Algorithm 3) to construct an equivalent deterministic finite automaton (DFA).

Solution:

Part 1:

a) Regular expression for Int part:

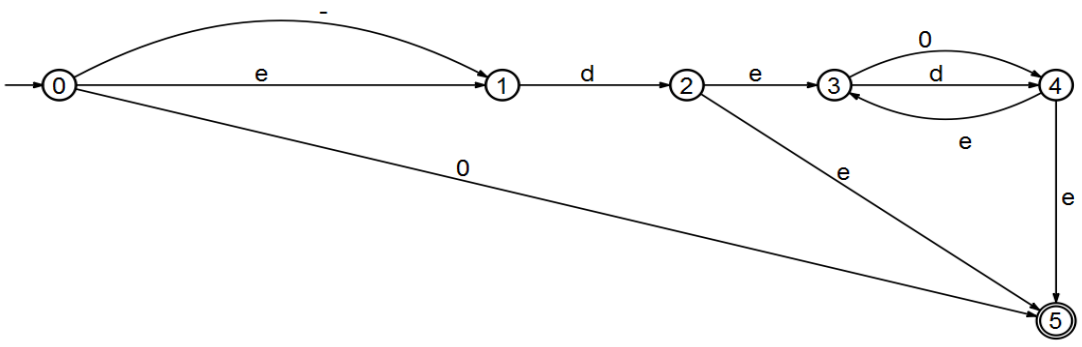
P1 = (0|((ε|-)d(0|d)*))

b) Regular expression for Decimal part:

P2 = ((.|(-|ε)(d(0|d)*|0+).)(0+d|d)(0|d)*)

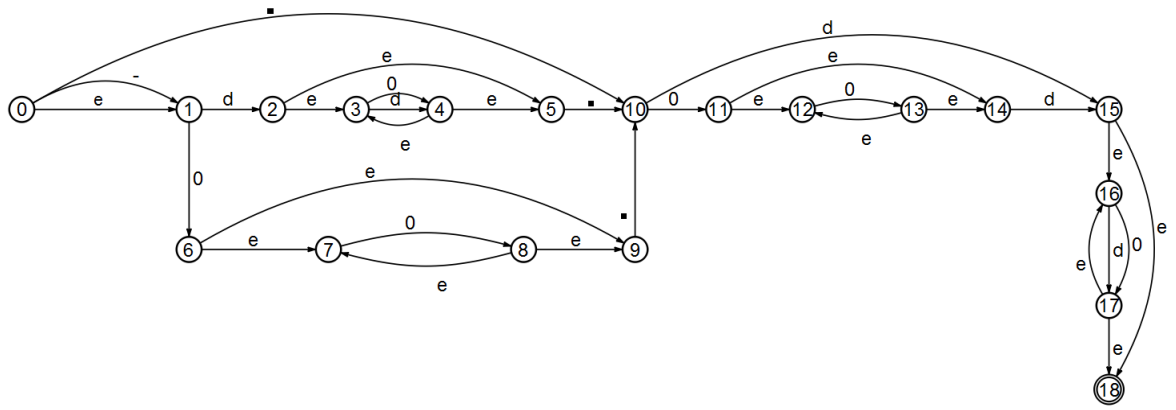
Part 2:

P1 = (0|((ε|-)d(0|d)*))



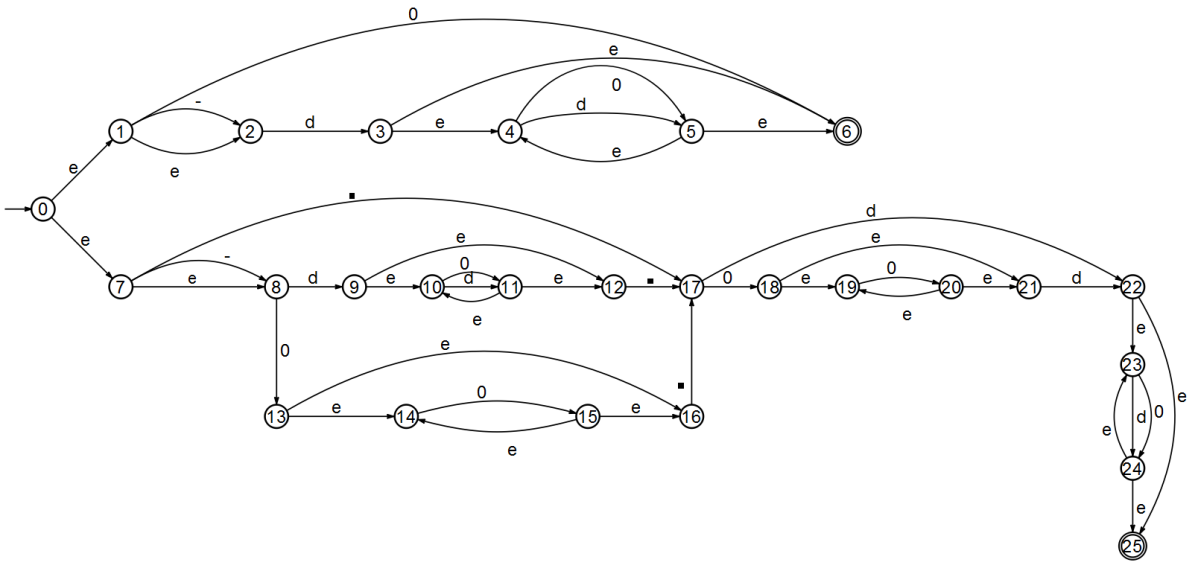
NFA for P1

P2 = ((.|(-|ε)(d(0|d)*|0+).)(0+d|d)(0|d)*)



NFA for P2

By combining NFA P1 and NFA P2



NFA accepting both int and decimal

NFA: {Q ,Σ,q0,F}

Q = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25}

Σ = {ε,0,d,.,-}

q0=0

F={6,25}

Transaction table for that NFA is below:

	-	**0**	**d**	**.**	**Accepting**
0	{1,2,7,8}	{1,2,6,7,8,13,16}	{1,2,3,4,6,7,8,9,10,12}	{1,2,7,8,17}	int
1	{2}	{6,2}	{2,3,6,4}	{2}	int
2	ø	ø	{3,6,4}	ø	int
3	{6,4}	{6,4,5}	{6,4,5}	{6,4}	int
4	ø	{5,4,6}	{5,4,6}	ø	int
5	{6}	{6}	{6}	{6}	int
6	ø	ø	ø	ø	int
7	{8}	{8,13,16,14}	{8,9,12,10}	{17,8}	Decimal
8	ø	{13,16,14}	{9,12,10}	ø	Decimal
9	{12,10}	{12,10,11}	{12,10,11}	{12,10}	Decimal
10	ø	{11,10,12}	{11,10,12}	ø	Decimal
11	{12,10}	{12,10,11}	{12,10,11}	{12,10,17}	Decimal
12	ø	ø	ø	{17}	Decimal
13	{16,14}	{16,14,15}	{16,14}	{16,17,14}	Decimal
14	ø	{15,14,16}	ø	ø	Decimal
15	{16,14}	{16,14,15}	{16,14}	{16,14,17}	Decimal
16	ø	ø	ø	{17}	Decimal
17	ø	{18,21,19}	{22,25,23}	ø	Decimal
18	{21,19]	{21,19,20}	{21,22,19}	ø	Decimal
19	ø	{20,19,21}	ø	ø	Decimal
20	{21}	{21}	{21,22,25,23}	{21}	Decimal
21	ø	ø	{22,25,23}	ø	Decimal
22	{25,23}	{25,23,24}	{25,23,24}	{25,23}	Decimal
23	ø	{24,23,25}	{24,23,25}	ø	Decimal
24	{25}	{25}	{25}	{25}	Decimal
25	ø	ø	ø	ø	Decimal

Part 3:

NFA to DFA conversion by subset construction algorithm:

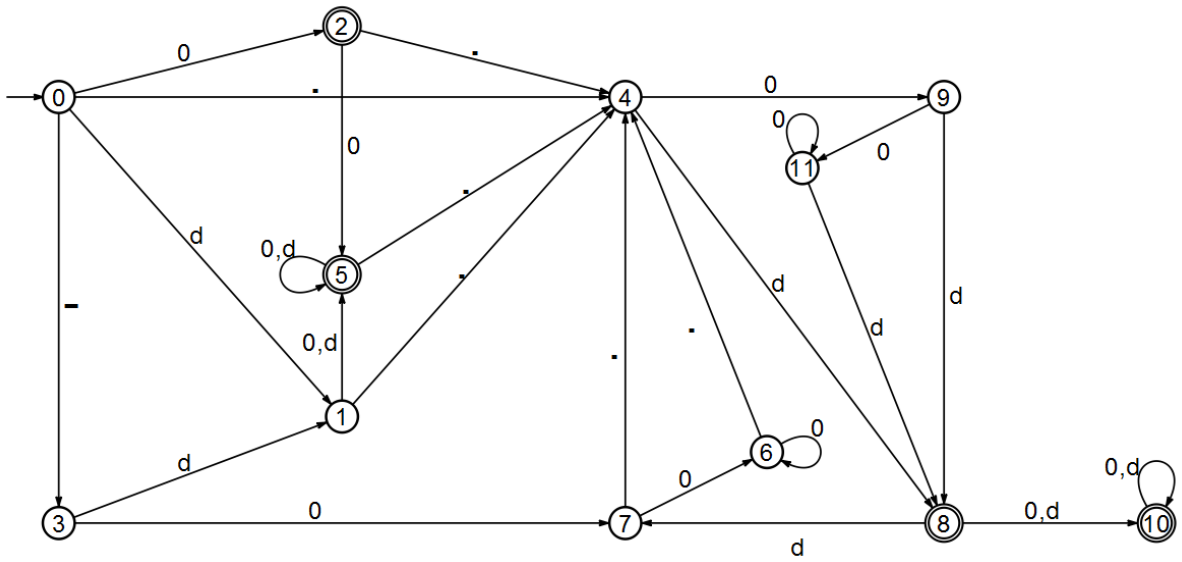
DFA	NFA	D	0	-	.
0	{ 0,1,2,7,8}	{ 3,6,4,9, 12,10}[1]	{6, 13,16,14}[2]	{2,8 }[3]	{17 }[4]
1	{ 3,4,6,9,10,12}	{ 5,6,4,11,12,10}[5]	{ 5,6,4,11,12,10}[5]	ø	{17 }[4]
2	{6,13,14,16 }	ø	{ 15,16,14}[6]	ø	{17 }[4]
3	{2,8 }	{ 3,6,4,9,12,10}[1]	{ 13,16,14}[7]	ø	ø
4	{17 }	{22,25,23 }[8]	{18,21,19}[9]	ø	ø
5	{4,5,6,10,11,12}	{ 5,6,4,11,12,10}[5]	{5,6,4,11,12,10}[5]	ø	{17 }[4]
6	{14,15,16 }	ø	{ 15,16,14}[6]	ø	{17 }[4]
7	{ 13,14,16}	ø	{15,16,14 }[6]	ø	{17}[4]
8	{22,23,25 }	{ 24,25,23}[10]	{ 24,25,23}[10]	ø	ø
9	{ 18,19,21}	{ 22,25,23}[8]	{20,21,19}[11]	ø	ø
10	{ 23,24,25}	{ 24,25,23}[10]	{ 24,25,23}[10]	ø	ø
11	{19,20,21 }	{ 22,25,23}[8]	{ 20,21,19}[11]	ø	ø

DFA: {Q ,Σ,q0,F}
Q = {0,1,2,3,4,5,6,7,8,9,10,11}
Σ = {0,d,.,-}
q0=0
F={2,5,8,10}

DFA Transaction table:

DFA	D	0	-	.	Initial/final	Accepting
0	1	2	3	4	Initial state	
1	5	5	err	4		
2	err	5	err	4	final state	int
3	1	7	err	err		
4	8	9	err	err		
5	5	5	err	4	final state	int
6	err	6	err	4		
7	err	6	err	4		
8	10	10	err	err	final state	decimal
9	8	11	err	err		
10	10	10	err	err	final state	decimal
11	8	11	err	err		

Graph:



DFA