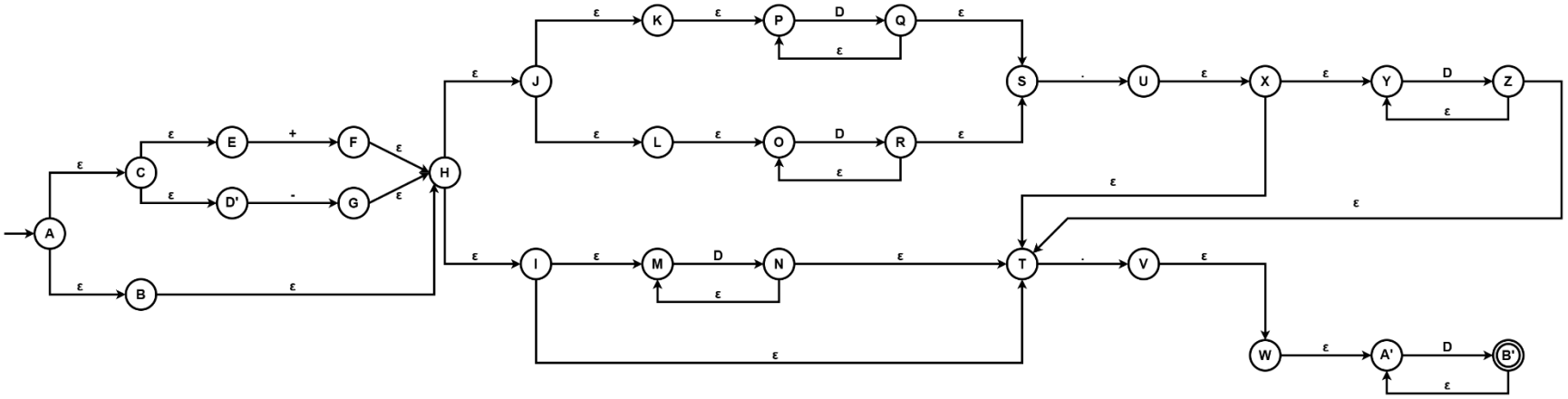
Answer to the Question No. **(a)**

Following regular expression can be used to represent the language of all valid numbers,

$$(+ \cup - \cup \varepsilon)\ (D^{+} \cup D^{+} . D^{*} \cup D^{*} . D^{+})$$ Where D = {0,1,2,3,4,5,6,7,8,9}

Answer to the Question No. **(b)**

The equivalent NFA,

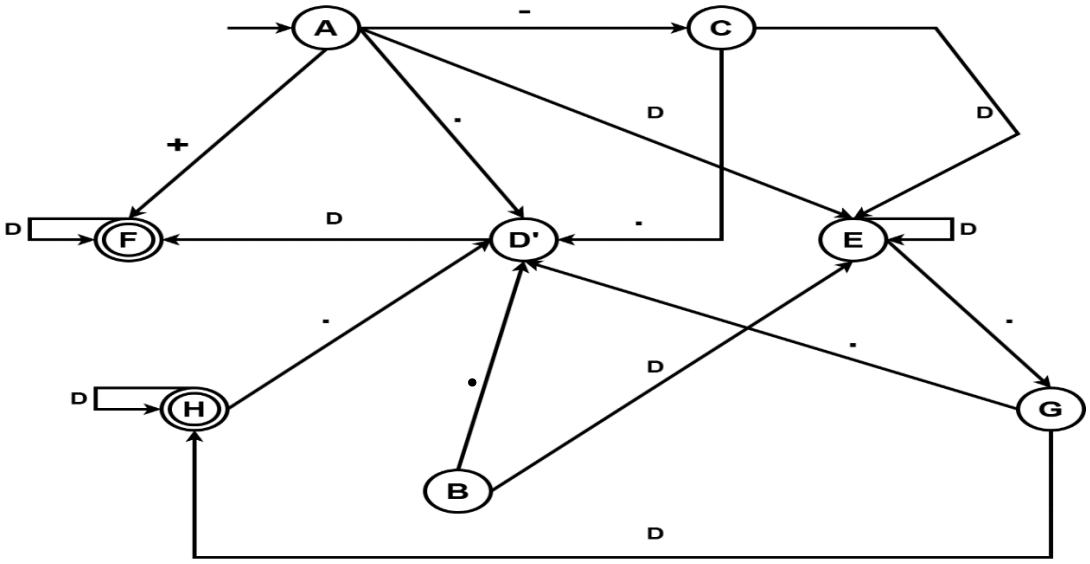


Answer to the Question No. **(c)**

DFA State	E-closure of	E-closure Outcome States
A	E-closure ({A})	A, B, C, D', E, H, I, J, K, L, M, O, P, T
B	E-closure ({F})	F, H, J, I, K, P, L, O, M, T
C	E-closure ({G})	G, H, J, I, K, P, L, O, M, T
D'	E-closure ({V})	V, W, A'
E	E-closure ({Q, R, N})	M, N, O, P, Q, R, S, T
F	E-closure ({B'})	A', B'
G	E-closure ({U, V})	T, U, V, W, X, Y, A'
H	E-closure ({Z, B'})	T, Y, Z, A', B'

NFA States	DFA State	+	-	.	D
A, B, C, D', E, H, I, J, K, L, M, O, P, T	A	F	C	D'	E
F, H, J, I, K, P, L, O, M, T	B	∅	∅	D'	E
G, H, J, I, K, P, L, O, M, T	C	∅	∅	D'	E
V, W, A'	D'	∅	∅	∅	F
M, N, O, P, Q, R, S, T	E	∅	∅	G	E
A', B'	F	∅	∅	∅	F
T, U, V, W, X, Y, A'	G	∅	∅	D'	H
T, Y, Z, A', B',	H	∅	∅	D'	H

The equivalent DFA,



Answer to the Question No. **(d)**

Equivalent Computer Program,

```
#include <iostream>
using namespace std;

int main()
{
    string dfa_current_input, dfa_current_state;
    string D[10] = {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9"};

    cin>>dfa_current_state;
    cin>>dfa_current_input;

    if(dfa_current_state== "A" && dfa_current_input=="+"){
        dfa_current_state= "F";
        cout<<"Current State : "<<dfa_current_state;
    }

    if(dfa_current_state== "A" && dfa_current_input=="."){
        dfa_current_state= "D'";
        cout<<"Current State : "<<dfa_current_state;
    }
```

```
if(dfa_current_state== "A" && dfa_current_input=="-"){
    dfa_current_state= "C";
    cout<<"Current State : "<<dfa_current_state;
}

if(dfa_current_state== "B" && dfa_current_input=="."){
    dfa_current_state= "D'";
    cout<<"Current State : "<<dfa_current_state;
}

if(dfa_current_state== "B" && dfa_current_input=="D"){
    dfa_current_state= "E";
    cout<<"Current State : "<<dfa_current_state;
}

if(dfa_current_state== "C" && dfa_current_input=="."){
    dfa_current_state= "D'";
    cout<<"Current State : "<<dfa_current_state;
}

if(dfa_current_state== "C" && dfa_current_input=="D"){
    dfa_current_state= "D'";
```

```
    cout<<"Current State : "<<dfa_current_state;
}

if(dfa_current_state== "D'" && dfa_current_input=="D"){
    dfa_current_state= "F";
    cout<<"Current State : "<<dfa_current_state;
}

if(dfa_current_state== "E" && dfa_current_input=="."){
    dfa_current_state= "G";
    cout<<"Current State : "<<dfa_current_state;
}

if(dfa_current_state== "E" && dfa_current_input=="D"){
    dfa_current_state= "E";
    cout<<"Current State : "<<dfa_current_state;
}

if(dfa_current_state== "F" && dfa_current_input=="D"){
    dfa_current_state= "F";
    cout<<"Current State : "<<dfa_current_state;
}
```

```
    if(dfa_current_state== "G" && dfa_current_input=="."){
        dfa_current_state= "D'";
        cout<<"Current State : "<<dfa_current_state;
    }

    if(dfa_current_state== "G" && dfa_current_input=="D"){
        dfa_current_state= "H";
        cout<<"Current State : "<<dfa_current_state;
    }

    if(dfa_current_state== "H" && dfa_current_input=="."){
        dfa_current_state= "D'";
        cout<<"Current State : "<<dfa_current_state;
    }

    if(dfa_current_state== "H" && dfa_current_input=="D"){
        dfa_current_state= "H";
        cout<<"Current State : "<<dfa_current_state;
    }
    return 0;
}
```