Single-Cage Elevator in Six-Story Building

EE6226 Discrete Event System Assignment 2020

Jiaxiang CHENG
School of Electrical and Electronic Engineering
Nanyang Techological University
Singapore
JCHENG029@E.NTU.EDU.SG

***Abstract*—The simulation program is built for a single-cage elevator in a *6*-story building. And the models are described by automaton. The code of simulation program, as well as other relevant files in this project, are available on *Github*: Github - Lawrence Cheng Jiaxiang - EE6226_DES_ASSIGNMENT.**

***Index Terms*—Single-Cage Elevator, Regular Languages, Finite Automaton, Simulation**

I. General Description

As how general elevator runs, there are *"0"-"5"* number keys inside the elevator (*Internal Calls*) in the *6*-story building. On the *0*th floor, there is only the *Up* key externally, and only the *Down* key on the *5*th floor, while on the other floors it is provided with *Up* and *Down* keys.

The general principle of how the elevator works is illustrated as follows (set 3 variables: i, j, k, where $1 \leq i < j < k \leq 6$.):

- For the elevator stopping on i^{th} floor, if j^{th} and k^{th} floors are called internally, the elevator will arrive at floor *j* and then floor *k* in sequence;
- For the elevator ascending on the i^{th} floor and k^{th} floor already called, the elevator will arrive at floor *j* firstly and then floor *k* if k^{th} floor is called at this time;
- For the elevator ascending on the j^{th} floor and k^{th} floor already called, the elevator will arrive at floor *k* and stop, not coming back to i^{th} floor;
- For the elevator descending, the principles are the same as above.

And in general, when the elevator is ascending, the external calls to go up above the current story will be served, while the external calls to go down will not be responded, except for the end story, i.e. the top and bottom floors. As for the descending elevator, it works with the same principle.

II. Modelling

A. Pre-processing of Input Call

Above all, a projection table is constructed to generate the *unserviced call* after reading the elevator status and input calls, as the next input, shown in Fig. 1.

B. Generating Queue from Unserviced Calls

A six-bit binary string *Queue* is used to indicate the queue to be responded to. $Queue_i = 1$ means that the call to i^{th} story has not been responded. $Queue_i = 0$ means that there is no request for i^{th} floor or the request has been responded. *"0"-"5"* in the input signal indicates that the there is a call to i^{th} story, which comes from the unserviced call generated by the previous step. And *"S0"-"S5"* indicate that the call for corresponding floor has been responded.

Therefore, a finite automaton is constructed as M_1 = *(Q, Σ, δ, q_0, F)* to describe the transition of the above *Queue* state, where *Q* is the finite set of states of the queue to be responded {000000, ..., 111111}, Σ = {0, 1, 2, 3, 4, 5, S0, S1, S2, S3, S4, S5} is the finite set of alphabet, q_0 is the initial state 000000, which means that no call needs to be served on any floor of the building, and *F* is final status 000000, indicating that there are no more calls in the building to respond to as well. The transition function is shown in Fig. 2.

	F0	F1	F2	F3	F4	F5	U0	U1	U2	U3	U4	D1	D2	D3	D4	D5
IN0	0	0	0	0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0
IN1	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	1	1	1	1
IN2	2	2	2	2	2	2	2	2	-1	-1	-1	-1	-1	2	2	2
IN3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	3	3
IN4	4	4	4	4	4	4	4	4	4	4	-1	-1	-1	-1	-1	4
IN5	5	5	5	5	5	5	5	5	5	5	5	-1	-1	-1	-1	-1
UP0	0	0	0	0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0
UP1	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1
UP2	2	2	2	2	2	2	2	2	-1	-1	-1	-1	-1	-1	-1	-1
UP3	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1
UP4	4	4	4	4	4	4	4	4	4	4	-1	-1	-1	-1	-1	-1
DW1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	1	1	1	1
DW2	2	2	2	2	2	2	-1	-1	-1	-1	-1	-1	-1	2	2	2
DW3	3	3	3	3	3	3	-1	-1	-1	-1	-1	-1	-1	-1	3	3
DW4	4	4	4	4	4	4	-1	-1	-1	-1	-1	-1	-1	-1	-1	4
DW5	5	5	5	5	5	5	5	5	5	5	5	-1	-1	-1	-1	-1

Fig. 1. Projection table to generate *unserviced call* from input according to the current state. The row indicates the input calls, while *"UP0"* indicates the *Up* key is called externally on the *0*th floor, *"DW5"* indicates the *Down* key is called externally on the *5*th floor, and *"IN0"* indicates the internal *0* key is called inside the elevator. The columns indicate the elevator status. The generated result is the unserviced calls, and "-1" means that the elevator will not respond to the call at this situation.

C. Command Generation and State Transition

Before coming to the transition of elevator states, the final control command needs to be generated from previous projection and generation. As shown in Fig. 3, the control command is generated after obtaining the updated *Queue* and according to the current state of the elevator.

By obtaining the final control command, the finite automaton is constructed as M_2 = *(Q, Σ, δ, q_0, F)* to describe the transition of the states of elevator, where *Q* is the finite set of states of the queue to be responded {F0, F1, F2, F3, F4, F5, U0, U1, U2, U3, U4, D1, D2, D3, D4, D5}, Σ = {0, 1, 2, 3, 4,

	0	1	2	3	4	5	S0	S1	S2	S3	S4	S5
000000	000001	000010	000100	001000	010000	100000	000000	000000	000000	000000	000000	000000
000001	000001	000011	000101	001001	010001	100001	000000	000001	000001	000001	000001	000001
000010	000011	000010	000110	001010	010010	100010	000010	000000	000010	000010	000010	000010
000011	000011	000011	000111	001011	010011	100011	000010	000001	000011	000011	000011	000011
000100	000101	000110	000100	001100	010100	100100	000100	000100	000000	000100	000100	000100
000101	000101	000111	000101	001101	010101	100101	000100	000101	000001	000101	000101	000101
000110	000111	000110	000110	001110	010110	100110	000110	000100	000010	000110	000110	000110
000111	000111	000111	000111	001111	010111	100111	000110	000101	000011	000111	000111	000111
001000	001001	001010	001100	001000	011000	101000	001000	001000	001000	000000	001000	001000
001001	001001	001011	001101	001001	011001	101001	001000	001001	001001	000001	001001	001001
001010	001011	001010	001110	001010	011010	101010	001010	001000	001010	000010	001010	001010
001011	001011	001011	001111	001011	011011	101011	001010	001001	001011	000011	001011	001011
001100	001101	001110	001100	001100	011100	101100	001100	001100	001000	000100	001100	001100
001101	001101	001111	001101	001101	011101	101101	001100	001101	001001	000101	001101	001101
001110	001111	001110	001110	001110	011110	101110	001110	001100	001010	000110	001110	001110
001111	001111	001111	001111	001111	011111	101111	001110	001101	001011	000111	001111	001111
010000	010001	010010	010100	011000	010000	110000	010000	010000	010000	010000	000000	010000
010001	010001	010011	010101	011001	010001	110001	010000	010001	010001	010001	000001	010001
010010	010011	010010	010110	011010	010010	110010	010010	010000	010010	010010	000010	010010
010011	010011	010011	010111	011011	010011	110011	010010	010001	010011	010011	000011	010011
010100	010101	010110	010100	011100	010100	110100	010100	010100	010000	010100	000100	010100
010101	010101	010111	010101	011101	010101	110101	010100	010101	010001	010101	000101	010101
010110	010111	010110	010110	011110	010110	110110	010110	010100	010010	010110	000110	010110
010111	010111	010111	010111	011111	010111	110111	010110	010101	010011	010111	000111	010111
011000	011001	011010	011100	011000	011000	111000	011000	011000	011000	010000	001000	011000
011001	011001	011011	011101	011001	011001	111001	011000	011001	011001	010001	001001	011001
011010	011011	011010	011110	011010	011010	111010	011010	011000	011010	010010	001010	011010
011011	011011	011011	011111	011011	011011	111011	011010	011001	011011	010011	001011	011011
011100	011101	011110	011100	011100	011100	111100	011100	011100	011000	010100	001100	011100
011101	011101	011111	011101	011101	011101	111101	011100	011101	011001	010101	001101	011101
011110	011111	011110	011110	011110	011110	111110	011110	011100	011010	010110	001110	011110
011111	011111	011111	011111	011111	011111	111111	011110	011101	011011	010111	001111	011111
100000	100001	100010	100100	101000	110000	100000	100000	100000	100000	100000	100000	000000
100001	100001	100011	100101	101001	110001	100001	100000	100001	100001	100001	100001	000001
100010	100011	100010	100110	101010	110010	100010	100010	100000	100010	100010	100010	000010
100011	100011	100011	100111	101011	110011	100011	100010	100001	100011	100011	100011	000011
100100	100101	100110	100100	101100	110100	100100	100100	100100	100000	100100	100100	000100
100101	100101	100111	100101	101101	110101	100101	100100	100101	100001	100101	100101	000101
100110	100111	100110	100110	101110	110110	100110	100110	100100	100010	100110	100110	000110
100111	100111	100111	100111	101111	110111	100111	100110	100101	100011	100111	100111	000111
101000	101001	101010	101100	101000	111000	101000	101000	101000	101000	100000	101000	001000
101001	101001	101011	101101	101001	111001	101001	101000	101001	101001	100001	101001	001001
101010	101011	101010	101110	101010	111010	101010	101010	101000	101010	100010	101010	001010
101011	101011	101011	101111	101011	111011	101011	101010	101001	101011	100011	101011	001011
101100	101101	101110	101100	101100	111100	101100	101100	101100	101000	100100	101100	001100
101101	101101	101111	101101	101101	111101	101101	101100	101101	101001	100101	101101	001101
101110	101111	101110	101110	101110	111110	101110	101110	101100	101010	100110	101110	001110
101111	101111	101111	101111	101111	111111	101111	101110	101101	101011	100111	101111	001111
110000	110001	110010	110100	111000	110000	110000	110000	110000	110000	110000	100000	010000
110001	110001	110011	110101	111001	110001	110001	110000	110001	110001	110001	100001	010001
110010	110011	110010	110110	111010	110010	110010	110010	110000	110010	110010	100010	010010
110011	110011	110011	110111	111011	110011	110011	110010	110001	110011	110011	100011	010011
110100	110101	110110	110100	111100	110100	110100	110100	110100	110000	110100	100100	010100
110101	110101	110111	110101	111101	110101	110101	110100	110101	110001	110101	100101	010101
110110	110111	110110	110110	111110	110110	110110	110110	110100	110010	110110	100110	010110
110111	110111	110111	110111	111111	110111	110111	110110	110101	110011	110111	100111	010111
111000	111001	111010	111100	111000	111000	111000	111000	111000	111000	110000	101000	011000
111001	111001	111011	111101	111001	111001	111001	111000	111001	111001	110001	101001	011001
111010	111011	111010	111110	111010	111010	111010	111010	111000	111010	110010	101010	011010
111011	111011	111011	111111	111011	111011	111011	111010	111001	111011	110011	101011	011011
111100	111101	111110	111100	111100	111100	111100	111100	111100	111000	110100	101100	011100
111101	111101	111111	111101	111101	111101	111101	111100	111101	111001	110101	101101	011101
111110	111111	111110	111110	111110	111110	111110	111110	111100	111010	110110	101110	011110
111111	111111	111111	111111	111111	111111	111111	111110	111101	111011	110111	101111	011111

Fig. 2. Transition function of *Queue*. The row indicates the current unserviced *Queue*, while the column represents the input calls. The transition function is, when the call for a certain story is received, the corresponding bit of *Queue* is set to 1, or when the call is served, i.e. the elevator reaches the called floor, the corresponding bit in *Queue* is set to 0.

	F0	F1	F2	F3	F4	F5	U0	U1	U2	U3	U4	D1	D2	D3	D4	D5
000000	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
000001	0	0	0	0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0
000010	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	1	1	1	1
000011	0	1	1	1	1	1	1	-1	-1	-1	-1	0	1	1	1	1
000100	2	2	2	2	2	2	2	2	-1	-1	-1	-1	-1	2	2	2
000101	0	2	2	2	2	2	2	2	-1	-1	-1	0	0	2	2	2
000110	1	1	2	2	2	2	1	2	-1	-1	-1	-1	1	2	2	2
000111	0	1	2	2	2	2	1	2	-1	-1	-1	0	1	2	2	2
001000	3	3	3	3	3	3	3	3	3	-1	-1	-1	-1	-1	3	3
001001	0	0	3	3	3	3	3	3	3	-1	-1	0	0	0	3	3
001010	1	1	3	3	3	3	1	3	3	-1	-1	-1	1	1	3	3
001011	0	1	1	3	3	3	1	3	3	-1	-1	0	1	1	3	3
001100	2	2	2	3	3	3	2	2	3	-1	-1	-1	-1	2	3	3
001101	0	2	2	3	3	3	2	2	3	-1	-1	0	0	2	3	3
001110	1	1	2	3	3	3	1	2	3	-1	-1	-1	1	2	3	3
001111	0	1	2	3	3	3	1	2	3	-1	-1	0	1	2	3	3
010000	4	4	4	4	4	4	4	4	4	4	-1	-1	-1	-1	-1	4
010001	0	0	4	4	4	4	4	4	4	4	-1	0	0	0	0	4
010010	1	1	1	4	4	4	1	4	4	4	-1	-1	1	1	1	4
010011	0	1	1	4	4	4	1	4	4	4	-1	0	1	1	1	4
010100	2	2	2	4	4	4	2	2	4	4	-1	-1	-1	2	2	4
010101	0	2	2	2	4	4	2	2	4	4	-1	0	0	2	2	4
010110	1	1	2	2	4	4	1	2	4	4	-1	-1	1	2	2	4
010111	0	1	2	2	4	4	1	2	4	4	-1	0	1	2	2	4
011000	3	3	3	3	4	4	3	3	3	4	-1	-1	-1	-1	3	4
011001	0	3	3	3	4	4	3	3	3	4	-1	0	0	0	3	4
011010	1	1	3	3	4	4	1	3	3	4	-1	-1	1	1	3	4
011011	0	1	3	3	4	4	1	3	3	4	-1	0	1	1	3	4
011100	2	2	2	3	4	4	2	2	3	4	-1	-1	-1	2	3	4
011101	0	2	2	3	4	4	2	2	3	4	-1	0	0	2	3	4
011110	1	1	2	3	4	4	1	2	3	4	-1	-1	1	2	3	4
011111	0	1	2	3	4	4	1	2	3	4	-1	0	1	2	3	4
100000	5	5	5	5	5	5	5	5	5	5	5	-1	-1	-1	-1	-1
100001	0	0	0	5	5	5	5	5	5	5	5	0	0	0	0	0
100010	1	1	1	5	5	5	1	5	5	5	5	-1	1	1	1	1
100011	0	1	1	1	1	5	1	5	5	5	5	0	1	1	1	1
100100	2	2	2	2	5	5	2	2	5	5	5	-1	-1	2	2	2
100101	0	2	2	2	2	5	2	2	5	5	5	0	0	2	2	2
100110	1	1	2	2	2	5	1	2	5	5	5	-1	1	2	2	2
100111	0	1	2	2	2	5	1	2	5	5	5	0	1	2	2	2
101000	3	3	3	3	5	5	3	3	3	5	5	-1	-1	-1	3	3
101001	0	0	3	3	3	5	3	3	3	5	5	0	0	0	3	3
101010	1	1	3	3	3	5	1	3	3	5	5	-1	1	1	3	3
101011	0	1	1	3	3	5	1	3	3	5	5	0	1	1	3	3
101100	2	2	2	3	3	5	2	2	3	5	5	-1	-1	2	3	3
101101	0	2	2	3	3	5	2	2	3	5	5	0	0	2	3	3
101110	1	1	2	3	3	5	1	2	3	5	5	-1	1	2	3	3
101111	0	1	2	3	3	5	1	2	3	5	5	0	1	2	3	3
110000	4	4	4	4	4	5	4	4	4	4	5	-1	-1	-1	-1	4
110001	0	0	4	4	4	5	4	4	4	4	5	0	0	0	0	4
110010	1	1	1	4	4	5	1	4	4	4	5	-1	1	1	1	4
110011	0	1	1	4	4	5	1	4	4	4	5	0	1	1	1	4
110100	2	2	2	4	4	5	2	2	4	4	5	-1	-1	2	2	4
110101	0	2	2	2	4	5	2	2	4	4	5	0	0	2	2	4
110110	1	1	2	2	4	5	1	2	4	4	5	-1	1	2	2	4
110111	0	1	2	2	4	5	1	2	4	4	5	0	1	2	2	4
111000	3	3	3	3	4	5	3	3	3	4	5	-1	-1	-1	3	4
111001	0	3	3	3	4	5	3	3	3	4	5	0	0	0	3	4
111010	1	1	3	3	4	5	1	3	3	4	5	-1	1	1	3	4
111011	0	1	3	3	4	5	1	3	3	4	5	0	1	1	3	4
111100	2	2	2	3	4	5	2	2	3	4	5	-1	-1	2	3	4
111101	0	2	2	3	4	5	2	2	3	4	5	0	0	2	3	4
111110	1	1	2	3	4	5	1	2	3	4	5	-1	1	2	3	4
111111	0	1	2	3	4	5	1	2	3	4	5	0	1	2	3	4

Fig. 3. Command generation table as the final input alphabet. The row indicates the unserviced *Queue*, while the column represents the current state of the elevator.

5} is the finite set of alphabet, i.e. the final command generated previously, q_0 is the initial state F0, which means that the elevator is located on the 0^{th} floor at the very beginning, and F is final status, including F0, F1, F2, F3, F4, F5, indicating that the elevator can finally stop at any floor in this building. The transition function is shown in Fig. 4.

Then according to the designed final automaton, the illustration can be generated automatically using Automaton Debugger, as shown in Fig. 6.

III. Simulation

The simulation program is developped using *Matlab*. By entering calls randomly, the simulation will react correspondingly. As shown in Fig. 7, user can enter number from *"0"*-*"15"*, where *"0"*-*"5"* represent the internal call to corresponding floors, *"6"*-*"10"* the external calls for going up on the 0^{th} - 4^{th} floors, and *"11"*-*"15"* the external calls for going down on the 1^{th} - 5^{th} floors.

	F0	F1	F2	F3	F4	F5	U0	U1	U2	U3	U4	D1	D2	D3	D4	D5
0	F0	D1	D2	D3	D4	D5	F1	F2	F3	F4	F5	F0	D1	D2	D3	D4
1	U0	F1	D2	D3	D4	D5	F1	F2	F3	F4	F5	F0	F1	D2	D3	D4
2	U0	U1	F2	D3	D4	D5	U1	F2	F3	F4	F5	F0	F1	F2	D3	D4
3	U0	U1	U2	F3	D4	D5	U1	U2	F3	F4	F5	F0	F1	F2	F3	D4
4	U0	U1	U2	U3	F4	D5	U1	U2	U3	F4	F5	F0	F1	F2	F3	F4
5	U0	U1	U2	U3	U4	F5	U1	U2	U3	U4	F5	F0	F1	F2	F3	F4

Fig. 4. Transition function of six-floor elevator state. The row indicates the final control command, while the column represents the current state of the elevator.

The current cage location can be shown accordingly, and when internal and external calls will be shown if the calls will be responded based on the principles described above.

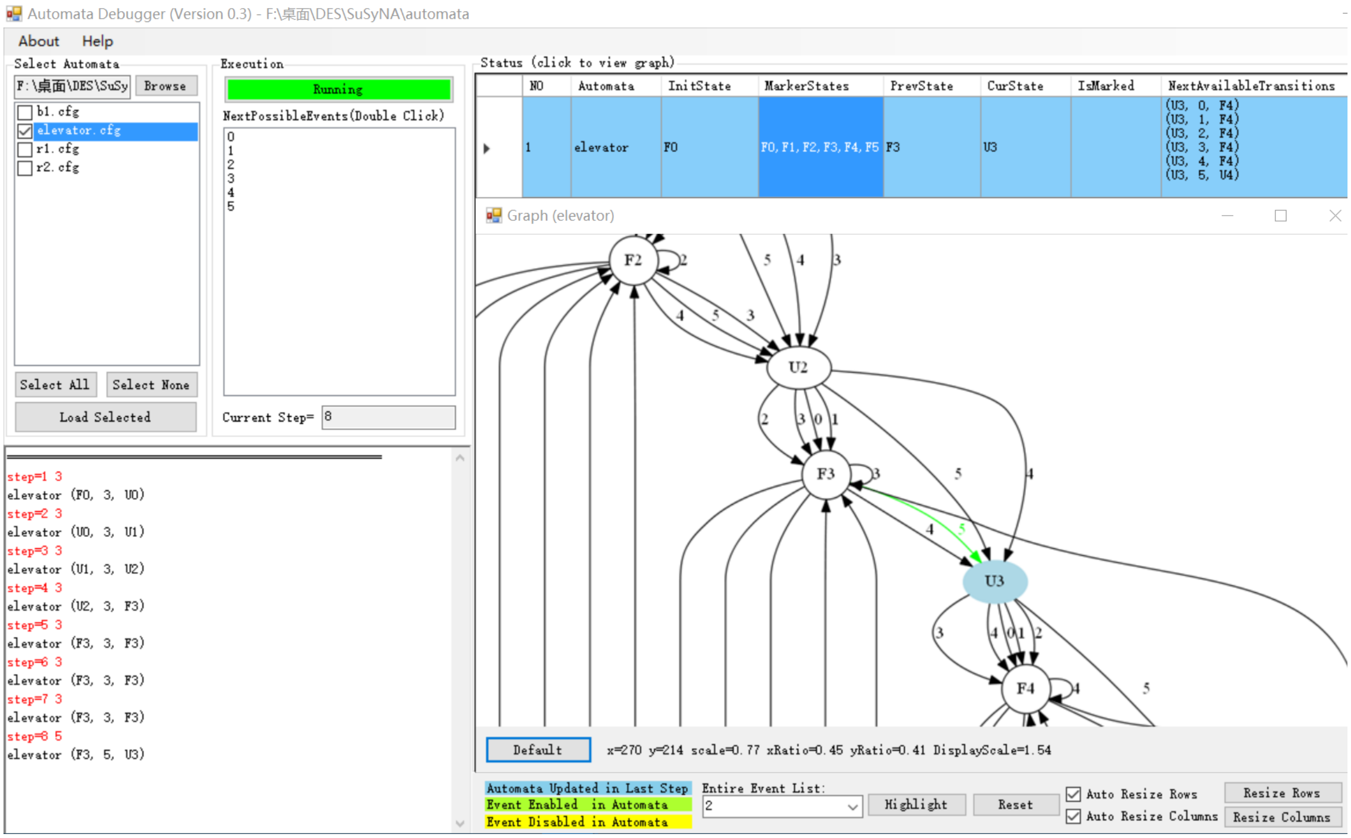

Fig. 5. Simulation display of the six-story elevator using Automaton Debugger. By using this simulation with input of the final control command, the result shows that the automaton works the same way as the simulation program via *Matlab*.

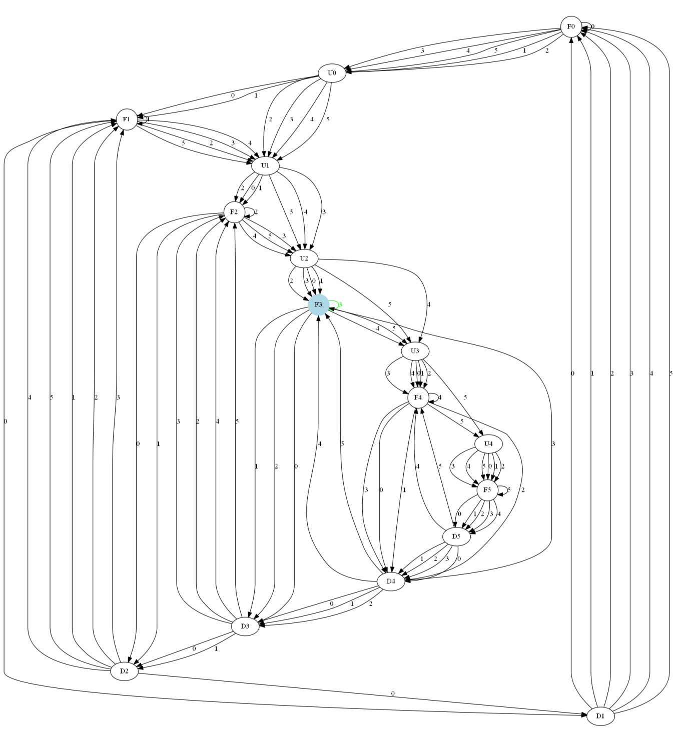

Fig. 6. Illustration of the elevator automaton, including the states and corresponding transition function between the states.

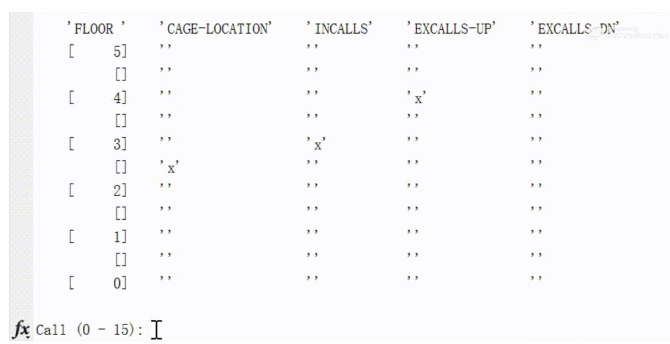

Fig. 7. Simulation display of the six-story elevator. By using *"x"*, the location and unserviced calls can be shown in the table. The program is available on Github - Lawrence Cheng Jiaxiang - EE6226_DES_ASSIGNMENT.

IV. Conclusion

The simulation program on Github - Lawrence Cheng Jiaxiang - EE6226_DES_ASSIGNMENT works sensibly with consideration of internal and external calls. By prior processing with projection tables, the final automaton can be constructed much more lightly.

Acknowledgment

Many thanks for the great lectures given by Prof. Su. I've learned a lot through the teaching and this assignment.

SIMULATION CODE - DES_1.M

```
%*************************************%
% by CHENG JIAXIANG                   %
%                                     %
% NANYANG TECHNOLOGICAL UNIVERSITY    %
% EE6226 - DISCRETE EVENT SYSTEM      %
% March 3rd, 2020                     %
%*************************************%

clear all

%% Initialization

% Initialize the queue and state:
queue = [0 0 0 0 0 0];
current_state = 0;
next_state = 0;

% Initialize the display board:
load display
for m = 2 : 12
    for n = 2 : 5
        display{m,n} = '';
    end
end

%% Generate the simulation:
while 1
    % Receive the call from both internal
    % and external:
    call = input("Call (0 - 15):");
    if call == -1
        break
    end % Input -1 to stop the iteration.

    % Use the projection table to generate
    % the service call:
    load table1
    sig_1 = table1(call+1, current_state+1);
    if call > -1
        unser_call = call;
    end % Make a record only when the call
        % is valid.

    if ~isempty(sig_1) && (unser_call < 6)
                       && (sig_1 ~= -1)
        switch unser_call
        % Display valid call from internal.
            case 0
                display{12,3} = 'x';
            case 1
                display{10,3} = 'x';
            case 2
                display{8,3} = 'x';
            case 3
                display{6,3} = 'x';
            case 4
                display{4,3} = 'x';
            case 5
                display{2,3} = 'x';
            otherwise
                disp('other value')
        end
    elseif ~isempty(sig_1) && (unser_call < 11)
                           && (sig_1 ~= -1)
        switch unser_call
        % Display valid UP-call from external.
            case 6
                display{12,4} = 'x';
            case 7
                display{10,4} = 'x';
            case 8
                display{8,4} = 'x';
            case 9
                display{6,4} = 'x';
            case 10
                display{4,4} = 'x';
        end
    elseif ~isempty(sig_1) && (unser_call < 16)
                           && (sig_1 ~= -1)
        switch unser_call
        % Display valid DOWN-call from external.
            case 11
                display{10,5} = 'x';
            case 12
                display{8,5} = 'x';
            case 13
                display{6,5} = 'x';
            case 14
                display{4,5} = 'x';
            case 15
                display{2,5} = 'x';
            otherwise
                disp('other value')
        end
    end

    current_state = next_state;
    % Mark the unserviced call as serviced
    % when reach the floor:
        switch current_state
        case 0
            display{12,3} = '';
            display{12,4} = '';
        case 1
            display{10,3} = '';
            display{10,4} = '';
            display{10,5} = '';
        case 2
            display{8,3} = '';
            display{8,4} = '';
```

```
            display{8,5} = '';
        case 3
            display{6,3} = '';
            display{6,4} = '';
            display{6,5} = '';
        case 4
            display{4,3} = '';
            display{4,4} = '';
            display{4,5} = '';
        case 5
            display{2,3} = '';
            display{2,5} = '';
        otherwise
            disp('other value')
    end
% Mark the unserviced and serviced calls in queue:
    sig_S = current_state;
    for i = 0 : 5
        if sig_1 == i
            queue(1,6-i) = 1;
        end
        if sig_S == i
            queue(1,6-i) = 0;
        end
    end

% Generate the queue from binary to decimal:
    queue_label = 0;
    for k = 0 : 5
        queue_label = queue_label +
            queue(1,k+1)*(2^(5-k));
    end

% Use projection to generate control command:
    load table2;
    command = table2(queue_label+1,
                     current_state+1);

% Use command and transition matrix to
% get the next state:
    if command ~= -1
        load state_transition;
        next_state = state_transition(command+1,
                     current_state+1);
    else
        next_state = current_state;
    end

% Display the next state (current cage location):
    switch next_state
        case 0
            display{12,2} = 'x';
        case 1
            display{10,2} = 'x';
        case 2
            display{8,2} = 'x';
        case 3
            display{6,2} = 'x';
        case 4
            display{4,2} = 'x';
        case 5
            display{2,2} = 'x';
        case 6
            display{11,2} = 'x';
        case 7
            display{9,2} = 'x';
        case 8
            display{7,2} = 'x';
        case 9
            display{5,2} = 'x';
        case 10
            display{3,2} = 'x';
        case 11
            display{11,2} = 'x';
        case 12
            display{9,2} = 'x';
        case 13
            display{7,2} = 'x';
        case 14
            display{5,2} = 'x';
        case 15
            display{3,2} = 'x';
        otherwise
            disp('other value')
    end

% Erase the past state (past cage location):
switch current_state
    case 0
        display{12,2} = '';
    case 1
        display{10,2} = '';
    case 2
        display{8,2} = '';
    case 3
        display{6,2} = '';
    case 4
        display{4,2} = '';
    case 5
        display{2,2} = '';
    case 6
        display{11,2} = '';
    case 7
        display{9,2} = '';
    case 8
        display{7,2} = '';
    case 9
        display{5,2} = '';
    case 10
        display{3,2} = '';
    case 11
        display{11,2} = '';
```

```
        case 12
            display{9,2} = '';
        case 13
            display{7,2} = '';
        case 14
            display{5,2} = '';
        case 15
            display{3,2} = '';
        otherwise
            disp('other value')
    end

    disp(display) % Show the simulation!

end
```

AUTOMATON - QUEUE.CFG

[automaton]

initial-state = 000000

marker-states = 000000 states = 000000, 000001, 000010, 000011, 000100, 000101, 000110, 000111, 001000, 001001, 001010, 001011, 001100, 001101, 001110, 001111, 010000, 010001, 010010, 010011, 010100, 010101, 010110, 010111, 011000, 011001, 011010, 011011, 011100, 011101, 011110, 011111, 100000, 100001, 100010, 100011, 100100, 100101, 100110, 100111, 101000, 101001, 101010, 101011, 101100, 101101, 101110, 101111, 110000, 110001, 110010, 110011, 110100, 110101, 110110, 110111, 111000, 111001, 111010, 111011, 111100, 111101, 111110, 111111

alphabet = 0, 1, 2, 3, 4, 5, S0, S1, S2, S3, S4, S5

controllable = 0, 1, 2, 3, 4, 5, S0, S1, S2, S3, S4, S5

observable = 0, 1, 2, 3, 4, 5, S0, S1, S2, S3, S4, S5

transitions = (000000, 000001, 0), (000000, 000010, 1), (000000, 000100, 2), (000000, 001000, 3), (000000, 010000, 4), (000000, 000000, S0), (000000, 000000, S1), (000000, 100000, 5), (000000, 000000, S2), (000000, 000000, S3), (000000, 000000, S4), (000000, 000000, S5), (000001, 000001, 0), (000001, 000011, 1), (000001, 000101, 2), (000001, 001001, 3), (000001, 010001, 4), (000001, 000000, S0), (000001, 000001, S1), (000001, 100001, 5), (000001, 000001, S2), (000001, 000001, S3), (000001, 000001, S4), (000001, 000001, S5), (000010, 000011, 0), (000010, 000010, 1), (000010, 000110, 2), (000010, 001010, 3), (000010, 010010, 4), (000010, 000010, S0), (000010, 000000, S1), (000010, 100010, 5), (000010, 000010, S2), (000010, 000010, S3), (000010, 000010, S4), (000010, 000010, S5), (000011, 000011, 0), (000011, 000011, 1), (000011, 000111, 2), (000011, 001011, 3), (000011, 010011, 4), (000011, 000010, S0), (000011, 000001, S1), (000011, 100011, 5), (000011, 000011, S2), (000011, 000011, S3), (000011, 000011, S4), (000011, 000011, S5), (000100, 000101, 0), (000100, 000110, 1), (000100, 000100, 2), (000100, 001100, 3), (000100, 010100, 4), (000100, 000100, S0), (000100, 000100, S1), (000100, 100100, 5), (000100, 000000, S2), (000100, 000100, S3), (000100, 000100, S4), (000100, 000100, S5), (000101, 000101, 0), (000101, 000111, 1), (000101, 000101, 2), (000101, 001101, 3), (000101, 010101, 4), (000101, 000100, S0), (000101, 000101, S1), (000101, 100101, 5), (000101, 000001, S2), (000101, 000101, S3), (000101, 000101, S4), (000101, 000101, S5), (000110, 000111, 0), (000110, 000110, 1), (000110, 000110, 2), (000110, 001110, 3), (000110, 010110, 4), (000110, 000110, S0), (000110, 000100, S1), (000110, 100110, 5), (000110, 000010, S2), (000110, 000110, S3), (000110, 000110, S4), (000110, 000110, S5), (000111, 000111, 0), (000111, 000111, 1), (000111, 000111, 2), (000111, 001111, 3), (000111, 010111, 4), (000111, 000110, S0), (000111, 000101, S1), (000111, 100111, 5), (000111, 000011, S2), (000111, 000111, S3), (000111, 000111, S4), (000111, 000111, S5), (001000, 001001, 0), (001000, 001010, 1), (001000, 001100, 2), (001000, 001000, 3), (001000, 011000, 4), (001000, 001000, S0), (001000, 001000, S1), (001000, 101000, 5), (001000, 001000, S2), (001000, 000000, S3), (001000, 001000, S4), (001000, 001000, S5), (001001, 001001, 0), (001001, 001011, 1), (001001, 001101, 2), (001001, 001001, 3), (001001, 011001, 4), (001001, 001000, S0), (001001, 001001, S1), (001001, 101001, 5), (001001, 001001, S2), (001001, 000001, S3), (001001, 001001, S4), (001001, 001001, S5), (001010, 001011, 0), (001010, 001010, 1), (001010, 001110, 2), (001010, 001010, 3), (001010, 011010, 4), (001010, 001010, S0), (001010, 001000, S1), (001010, 101010, 5), (001010, 001010, S2), (001010, 000010, S3), (001010, 001010, S4), (001010, 001010, S5), (001011, 001011, 0), (001011, 001011, 1), (001011, 001111, 2), (001011, 001011, 3), (001011, 011011, 4), (001011, 001010, S0), (001011, 001001, S1), (001011, 101011, 5), (001011, 001011, S2), (001011, 000011, S3), (001011, 001011, S4), (001011, 001011, S5), (001100, 001101, 0), (001100, 001110, 1), (001100, 001100, 2), (001100, 001100, 3), (001100, 011100, 4), (001100, 001100, S0), (001100, 001100, S1), (001100, 101100, 5), (001100, 001000, S2), (001100, 000100, S3), (001100, 001100, S4), (001100, 001100, S5), (001101, 001101, 0), (001101, 001111, 1), (001101, 001101, 2), (001101, 001101, 3), (001101, 011101, 4), (001101, 001100, S0), (001101, 001101, S1), (001101, 101101, 5), (001101, 001001, S2), (001101, 000101, S3), (001101, 001101, S4), (001101, 001101, S5), (001110, 001111, 0), (001110, 001110, 1), (001110, 001110, 2), (001110, 001110, 3), (001110, 011110, 4), (001110, 001110, S0), (001110, 001100, S1), (001110, 101110, 5), (001110, 001010, S2), (001110, 000110, S3), (001110, 001110, S4), (001110, 001110, S5), (001111, 001111, 0), (001111, 001111, 1), (001111, 001111, 2), (001111, 001111, 3), (001111, 011111, 4), (001111, 001110, S0), (001111, 001101, S1), (001111, 101111, 5), (001111, 001011, S2), (001111, 000111, S3), (001111, 001111, S4), (001111, 001111, S5), (010000, 010001, 0), (010000, 010010, 1), (010000, 010100, 2), (010000, 011000, 3), (010000, 010000, 4), (010000, 010000, S0), (010000, 010000, S1), (010000, 110000, 5), (010000, 010000, S2), (010000, 010000, S3), (010000, 000000, S4), (010000, 010000, S5), (010001, 010001, 0), (010001, 010011, 1), (010001, 010101, 2), (010001, 011001, 3), (010001, 010001, 4), (010001, 010000, S0), (010001, 010001, S1), (010001, 110001, 5), (010001, 010001, S2), (010001, 010001, S3), (010001, 000001, S4), (010001, 010001, S5), (010010, 010011, 0), (010010, 010010, 1), (010010, 010110, 2), (010010, 011010, 3), (010010, 010010, 4), (010010, 010010, S0), (010010, 010000, S1), (010010, 110010, 5), (010010, 010010, S2), (010010, 010010, S3), (010010, 000010, S4), (010010, 010010, S5), (010011, 010011, 0), (010011, 010011, 1), (010011, 010111, 2), (010011, 011011, 3), (010011, 010011, 4), (010011, 010010, S0), (010011, 010001, S1), (010011, 110011, 5), (010011, 010011, S2), (010011, 010011, S3), (010011, 000011, S4), (010011, 010011, S5), (010100, 010101, 0), (010100, 010110, 1), (010100, 010100, 2), (010100, 011100, 3), (010100, 010100, 4), (010100, 010100, S0), (010100, 010100, S1), (010100, 110100, 5), (010100, 010000, S2), (010100, 010100, S3), (010100, 000100, S4), (010100, 010100, S5), (010101, 010101, 0), (010101, 010111, 1), (010101, 010101, 2), (010101, 011101, 3), (010101, 010101, 4), (010101, 010100, S0), (010101, 010101, S1), (010101, 110101, 5), (010101, 010001, S2), (010101, 010101, S3), (010101, 000101, S4), (010101, 010101, S5), (010111, 010111, 0), (010111, 010111, 1), (010111, 010111, 2), (010111, 011111, 3), (010111, 010111, 4), (010111, 010110, S0), (010111, 010101, S1), (010111, 110111, 5), (010111, 010011, S2), (010111, 010111, S3), (010111, 000111, S4), (010111, 010111, S5), (010110, 010111, 0), (010110, 010110, 1), (010110, 010110, 2), (010110, 011110, 3), (010110, 010110, 4), (010110, 010110, S0), (010110, 010100, S1), (010110, 110110, 5), (010110, 010010, S2), (010110, 010110, S3), (010110, 000110, S4), (010110, 010110, S5), (011000, 011001, 0), (011000, 011010, 1), (011000, 011100, 2), (011000, 011000, 3), (011000, 011000, 4), (011000, 011000, S0), (011000, 011000, S1), (011000, 111000, 5), (011000, 011000, S2), (011000, 010000, S3), (011000, 001000, S4), (011000, 011000, S5), (011001, 011001, 0), (011001, 011011, 1), (011001, 011101, 2), (011001, 011001, 3), (011001, 011001, 4), (011001, 011000, S0), (011001, 011001, S1), (011001, 111001, 5), (011001, 011001, S2), (011001, 010001, S3), (011001, 001001, S4), (011001, 011001, S5), (011010, 011011, 0), (011010, 011010, 1), (011010, 011110, 2), (011010, 011010, 3), (011010, 011010, 4), (011010, 011010, S0), (011010, 011000, S1), (011010, 111010, 5), (011010, 011010, S2), (011010, 010010, S3), (011010, 001010, S4), (011010, 011010, S5), (011011, 011011, 0), (011011, 011011, 1), (011011, 011111, 2), (011011, 011011, 3), (011011, 011011, 4), (011011, 011010, S0), (011011, 011001, S1), (011011, 111011, 5), (011011, 011011, S2), (011011, 010011, S3), (011011, 001011, S4), (011011, 011011, S5), (011100, 011101, 0), (011100, 011110, 1), (011100, 011100, 2), (011100, 011100, 3), (011100, 011100, 4), (011100, 011100, S0), (011100, 011100, S1), (011100, 111100, 5), (011100, 011000, S2), (011100, 010100, S3), (011100, 001100, S4), (011100, 011100, S5), (011101, 011101, 0), (011101, 011111, 1), (011101, 011101, 2), (011101, 011101, 3), (011101, 011101, 4), (011101, 011100, S0), (011101, 011101, S1), (011101, 111101, 5), (011101, 011001, S2), (011101, 010101, S3), (011101, 001101, S4), (011101, 011101, S5), (011110, 011111, 0), (011110, 011110, 1), (011110, 011110, 2), (011110, 011110, 3), (011110, 011110, 4), (011110, 011110, S0), (011110, 011100, S1), (011110, 111110, 5), (011110, 011010, S2), (011110, 010110, S3), (011110, 001110, S4), (011110, 011110, S5), (011111, 011111, 0), (011111, 011111, 1), (011111, 011111, 2), (011111, 011111, 3), (011111, 011111, 4), (011111, 011110, S0), (011111, 011101, S1), (011111, 111111, 5), (011111, 011011, S2), (011111, 010111, S3), (011111, 001111, S4), (011111, 011111, S5), (100000, 100001, 0), (100000, 100010, 1), (100000, 100100, 2), (100000, 101000, 3), (100000, 110000, 4), (100000, 100000, S0), (100000, 100000, S1), (100000, 100000, 5), (100000, 100000, S2), (100000, 100000, S3), (100000, 100000, S4), (100000, 000000, S5), (100001, 100001, 0), (100001, 100011, 1), (100001, 100101, 2), (100001, 101001, 3), (100001, 110001, 4), (100001, 100000, S0), (100001, 100001, S1), (100001, 100001, 5), (100001, 100001, S2), (100001, 100001, S3), (100001, 100001, S4), (100001, 000001, S5), (100010, 100011, 0), (100010, 100010, 1), (100010, 100110, 2), (100010, 101010, 3), (100010, 110010, 4), (100010, 100010, S0), (100010, 100000, S1), (100010, 100010, 5), (100010, 100010, S2), (100010, 100010, S3), (100010, 100010, S4), (100010, 000010, S5), (100011, 100011, 0), (100011, 100011, 1), (100011, 100111, 2), (100011, 101011, 3), (100011, 110011, 4), (100011, 100010, S0), (100011, 100001, S1), (100011, 100011, 5), (100011, 100011, S2), (100011, 100011, S3), (100011, 100011, S4), (100011, 000011, S5), (100100, 100101, 0), (100100, 100110, 1), (100100, 100100, 2), (100100, 101100, 3), (100100, 110100, 4), (100100, 100100, S0), (100100, 100100, S1), (100100, 100100, 5), (100100, 100000, S2), (100100, 100100, S3), (100100, 100100, S4), (100100, 000100, S5), (100101, 100101, 0), (100101, 100111, 1), (100101, 100101, 2), (100101, 101101, 3), (100101, 110101, 4), (100101, 100100, S0), (100101, 100101, S1), (100101, 100101, 5), (100101, 100001, S2), (100101, 100101, S3), (100101, 100101, S4), (100101, 000101, S5), (100110, 100111, 0), (100110, 100110, 1), (100110, 100110, 2), (100110, 101110, 3), (100110, 110110, 4), (100110, 100110, S0), (100110, 100100, S1), (100110, 100110, 5), (100110, 100010, S2), (100110, 100110, S3), (100110, 100110, S4), (100110, 000110, S5), (100111, 100111, 0), (100111, 100111, 1), (100111, 100111, 2), (100111, 101111, 3), (100111, 110111, 4), (100111, 100110, S0), (100111, 100101, S1), (100111, 100111, 5), (100111, 100011, S2), (100111, 100111, S3), (100111, 100111, S4), (100111, 000111, S5), (101000, 101001, 0), (101000, 101010, 1), (101000, 101100, 2), (101000, 101000, 3), (101000, 111000, 4), (101000, 101000, S0), (101000, 101000, S1), (101000, 101000, 5), (101000, 101000, S2), (101000, 100000, S3), (101000, 101000, S4), (101000, 001000, S5), (101001, 101001, 0), (101001, 101011, 1), (101001, 101101, 2), (101001, 101001, 3), (101001, 111001, 4), (101001, 101000, S0), (101001, 101001, S1), (101001, 101001, 5), (101001, 101001, S2), (101001, 100001, S3), (101001, 101001, S4), (101001, 001001, S5), (101010, 101011, 0), (101010, 101010, 1), (101010, 101110, 2), (101010, 101010, 3), (101010, 111010, 4), (101010, 101010, S0), (101010, 101000, S1), (101010, 101010, 5), (101010, 101010, S2), (101010, 100010, S3), (101010, 101010, S4), (101010, 001010, S5), (101011, 101011, 0), (101011, 101011, 1), (101011, 101111, 2), (101011, 101011, 3), (101011, 111011, 4), (101011, 101010, S0), (101011, 101001, S1), (101011, 101011, 5), (101011, 101011, S2), (101011, 100011, S3), (101011, 101011, S4), (101011, 001011, S5), (101100, 101101, 0), (101100, 101110, 1), (101100, 101100, 2), (101100, 101100, 3), (101100, 111100, 4), (101100, 101100, S0), (101100, 101100, S1), (101100, 101100, 5), (101100, 101000, S2), (101100, 100100, S3), (101100, 101100, S4), (101100, 001100, S5), (101101, 101101, 0), (101101, 101111, 1), (101101, 101101, 2), (101101, 101101, 3), (101101, 111101, 4), (101101, 101100, S0), (101101, 101101, S1), (101101, 101101, 5), (101101, 101001, S2), (101101, 100101, S3), (101101, 101101, S4), (101101, 001101, S5), (101110, 101111, 0), (101110, 101110, 1), (101110, 101110, 2), (101110, 101110, 3), (101110, 111110, 4), (101110, 101110, S0), (101110, 101100, S1), (101110, 101110, 5), (101110, 101010, S2), (101110, 100110, S3), (101110, 101110, S4), (101110, 001110, S5), (101111, 101111, 0), (101111, 101111, 1), (101111, 101111, 2), (101111, 101111, 3), (101111, 111111, 4), (101111, 101110, S0), (101111, 101101, S1), (101111, 101111, 5), (101111, 101011, S2), (101111, 100111, S3), (101111, 101111, S4), (101111, 001111, S5), (110000, 110001, 0), (110000, 110010, 1), (110000, 110100, 2), (110000, 111000, 3), (110000, 110000, 4), (110000, 110000, S0), (110000, 110000, S1), (110000, 110000, 5), (110000, 110000, S2), (110000, 110000, S3), (110000, 100000, S4), (110000, 010000, S5), (110001, 110001, 0), (110001, 110011, 1), (110001, 110101, 2), (110001, 111001, 3), (110001, 110001, 4), (110001, 110000, S0), (110001, 110001, S1), (110001, 110001, 5), (110001, 110001, S2), (110001, 110001, S3), (110001, 100001, S4), (110001, 010001, S5), (110010, 110011, 0), (110010, 110010, 1), (110010, 110110, 2), (110010, 111010, 3), (110010, 110010, 4), (110010, 110010, S0), (110010, 110000, S1), (110010, 110010, 5), (110010, 110010, S2), (110010, 110010, S3), (110010, 100010, S4), (110010, 010010, S5), (110011, 110011, 0), (110011, 110011, 1), (110011, 110111, 2), (110011, 111011, 3), (110011, 110011, 4), (110011, 110010, S0), (110011, 110001, S1), (110011, 110011, 5), (110011, 110011, S2), (110011, 110011, S3), (110011, 100011, S4), (110011, 010011, S5), (110100, 110101, 0), (110100, 110110, 1), (110100, 110100, 2), (110100, 111100, 3), (110100, 110100, 4), (110100, 110100, S0), (110100, 110100, S1), (110100, 110100, 5), (110100, 110000, S2), (110100, 110100, S3), (110100, 100100, S4), (110100, 010100, S5), (110101, 110101, 0), (110101, 110111, 1), (110101, 110101, 2), (110101, 111101, 3), (110101, 110101, 4), (110101, 110100, S0), (110101, 110101, S1), (110101, 110101, 5), (110101, 110001, S2), (110101, 110101, S3), (110101, 100101, S4), (110101, 010101, S5), (110110, 110111, 0), (110110, 110110, 1), (110110, 110110, 2), (110110, 111110, 3), (110110, 110110, 4), (110110, 110110, S0), (110110, 110100, S1), (110110, 110110, 5), (110110, 110010, S2), (110110, 110110, S3), (110110, 100110, S4), (110110, 010110, S5), (110111, 110111, 0), (110111, 110111, 1), (110111, 110111, 2), (110111, 111111, 3), (110111, 110111, 4), (110111, 110110, S0), (110111, 110101, S1), (110111, 110111, 5), (110111, 110011, S2), (110111, 110111, S3), (110111, 100111, S4), (110111, 010111, S5), (111000, 111001, 0), (111000, 111010, 1), (111000, 111100, 2), (111000, 111000, 3), (111000, 111000, 4), (111000, 111000, S0), (111000, 111000, S1), (111000, 111000, 5), (111000, 111000, S2), (111000, 110000, S3), (111000, 101000, S4), (111000, 011000, S5), (111001, 111001, 0), (111001, 111011, 1), (111001, 111101, 2), (111001, 111001, 3), (111001, 111001, 4), (111001, 111000, S0), (111001, 111001, S1), (111001, 111001, 5), (111001, 111001, S2), (111001, 110001, S3), (111001, 101001, S4), (111001, 011001, S5), (111010, 111011, 0), (111010, 111010, 1), (111010, 111110, 2), (111010, 111010, 3), (111010, 111010, 4), (111010, 111010, S0), (111010, 111000, S1), (111010, 111010, 5), (111010, 111010, S2), (111010, 110010, S3), (111010, 101010, S4), (111010, 011010, S5), (111011, 111011, 0), (111011, 111011, 1), (111011, 111111, 2), (111011, 111011, 3), (111011, 111011, 4), (111011, 111010, S0), (111011, 111001, S1), (111011, 111011, 5), (111011, 111011, S2), (111011, 110011, S3), (111011, 101011, S4), (111011, 011011, S5), (111100, 111101, 0), (111100, 111110, 1), (111100, 111100, 2), (111100, 111100, 3), (111100, 111100, 4), (111100, 111100, S0), (111100, 111100, S1), (111100, 111100, 5), (111100, 111000, S2), (111100, 110100, S3), (111100, 101100, S4), (111100, 011100, S5), (111101, 111101, 0), (111101, 111111, 1), (111101, 111101, 2), (111101, 111101, 3), (111101, 111101, 4), (111101, 111100, S0), (111101, 111101, S1), (111101, 111101, 5), (111101, 111001, S2), (111101, 110101, S3), (111101, 101101, S4), (111101, 011101, S5), (111110, 111111, 0), (111110, 111110, 1), (111110, 111110, 2), (111110, 111110, 3), (111110, 111110, 4), (111110, 111110, S0), (111110, 111100, S1), (111110, 111110, 5), (111110, 111010, S2), (111110, 110110, S3), (111110, 101110, S4), (111110, 011110, S5), (111111, 111111, 0), (111111, 111111, 1), (111111, 111111, 2), (111111, 111111, 3), (111111, 111111, 4), (111111, 111110, S0), (111111, 111101, S1), (111111, 111111, 5), (111111, 111011, S2), (111111, 110111, S3), (111111, 101111, S4), (111111, 011111, S5)

AUTOMATON - ELEVATOR.CFG

[automaton]

initial-state = F0

marker-states = F0, F1, F2, F3, F4, F5

states = F0, F1, F2, F3, F4, F5, U0, U1, U2, U3, U4, D1, D2, D3, D4, D5

alphabet = 0, 1, 2, 3, 4, 5

controllable = 0, 1, 2, 3, 4, 5

observable = 0, 1, 2, 3, 4, 5

transitions = (F0, F0, 0), (F0, U0, 1), (F0, U0, 2), (F0, U0, 3), (F0, U0, 4), (F0, U0, 5), (F1, D1, 0), (F1, F1, 1), (F1, U1, 2), (F1, U1, 3), (F1, U1, 4), (F1, U1, 5), (F2, D2, 0), (F2, D2, 1), (F2, F2, 2), (F2, U2, 3), (F2, U2, 4), (F2, U2, 5), (F3, D3, 0), (F3, D3, 1), (F3, D3, 2), (F3, F3, 3), (F3, U3, 4), (F3, U3, 5), (F4, D4, 0), (F4, D4, 1), (F4, D4, 2), (F4, D4, 3), (F4, F4, 4), (F4, U4, 5), (F5, D5, 0), (F5, D5, 1), (F5, D5, 2), (F5, D5, 3), (F5, D5, 4), (F5, F5, 5), (U0, F1, 0), (U0, F1, 1), (U0, U1, 2), (U0, U1, 3), (U0, U1, 4), (U0, U1, 5), (U1, F2, 0), (U1, F2, 1), (U1, F2, 2), (U1, U2, 3), (U1, U2, 4), (U1, U2, 5), (U2, F3, 0), (U2, F3, 1), (U2, F3, 2), (U2, F3, 3), (U2, U3, 4), (U2, U3, 5), (U3, F4, 0), (U3, F4, 1), (U3, F4, 2), (U3, F4, 3), (U3, F4, 4), (U3, U4, 5), (U4, F5, 0), (U4, F5, 1), (U4, F5, 2), (U4, F5, 3), (U4, F5, 4), (U4, F5, 5), (D1, F0, 0), (D1, F0, 1), (D1, F0, 2), (D1, F0, 3), (D1, F0, 4), (D1, F0, 5), (D2, D1, 0), (D2, F1, 1), (D2, F1, 2), (D2, F1, 3), (D2, F1, 4), (D2, F1, 5), (D3, D2, 0), (D3, D2, 1), (D3, F2, 2) , (D3, F2, 3) , (D3, F2, 4) , (D3, F2, 5), (D4, D3, 0), (D4, D3, 1) , (D4, D3, 2) , (D4, F3, 3) , (D4, F3, 4) , (D4, F3, 5), (D5, D4, 0), (D5, D4, 1), (D5, D4, 2), (D5, D4, 3), (D5, F4, 4), (D5, F4, 5)