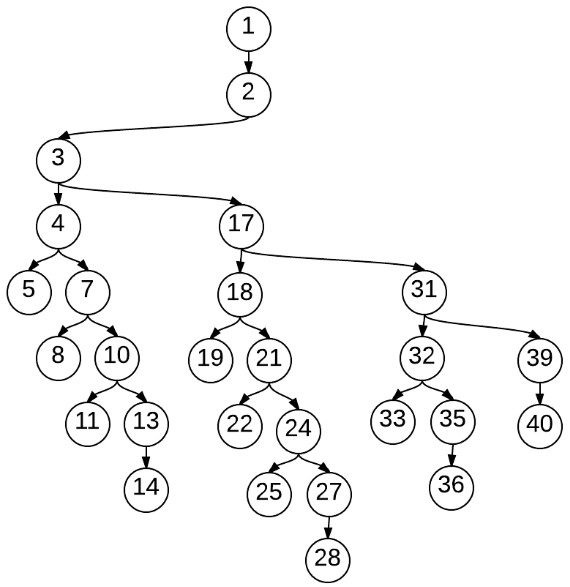
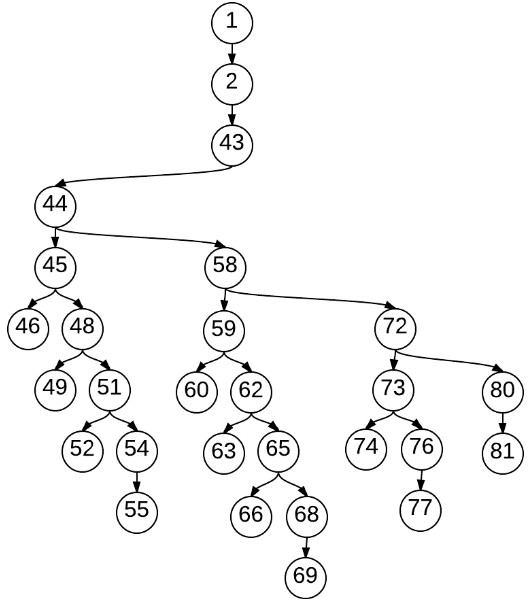
**Java code of Vending Machine**

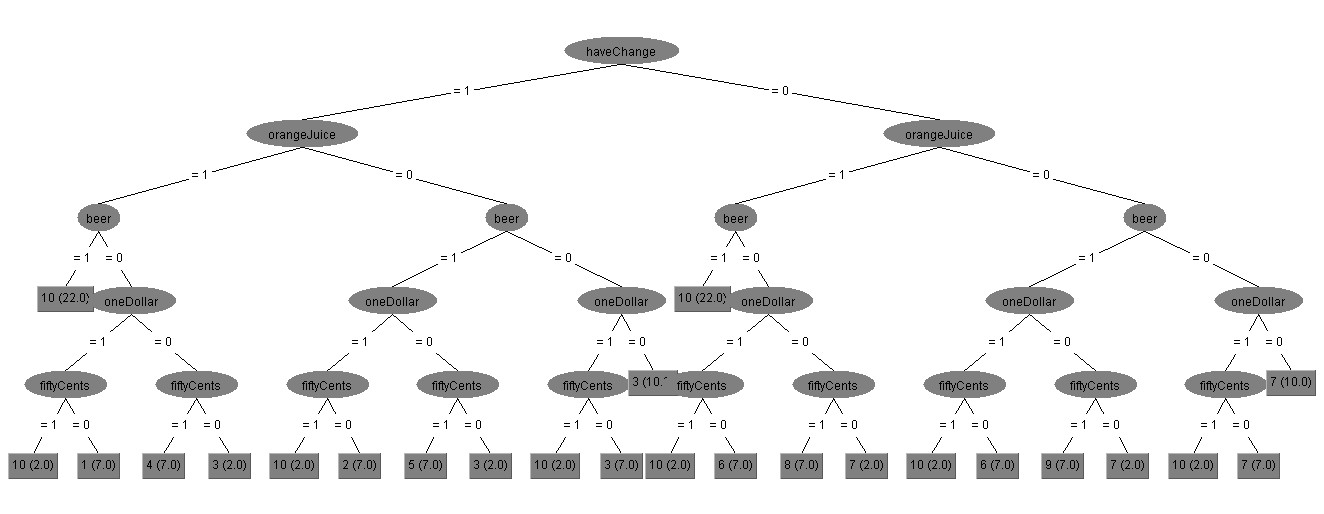
1. public int vTest(int haveChange, int oneDollar, int fiftyCents, int orangeJuice, int beer){
2. if(haveChange == 1){
3. if(oneDollar == 1 && fiftyCents == 0){
4. if(orangeJuice == 1 && beer == 0){
5. return 1; //light off, 50 cents return, get orange juice
6. }
7. else if(orangeJuice == 0 && beer == -1){
8. return 2; //light off, 50 cents return, get beer
9. }
10. else if(orangeJuice == 0 && beer == 0){
11. return 3; //light off, no money return, get nothing
12. }
13. else{
14. return 10;//invalid
15. }
16. }
17. else if(fiftyCents == 1 && oneDollar == 0){
18. if(orangeJuice == 1 && beer == 0){
19. return 4; //light off, no money return, get orange juice
20. }
21. else if(orangeJuice == 0 && beer != 1){
22. return 5; //light off, no money return, get beer
23. }
24. else if(orangeJuice == 0 && beer == 0){
25. return 3; //light off, no money return, get nothing
26. }
27. else{
28. return 10;
29. }
30. }
31. else if(fiftyCents == 0 && oneDollar == 0){
32. if(orangeJuice == 1 && beer == 1){
33. return 10;
34. }
35. else{
36. return 3;
37. }
38. }
39. else{
40. return 10;
41. }
42. }
43. else{
44. if(oneDollar == 1 && fiftyCents == 0){
45. if(orangeJuice == 1 && beer == 0){
46. return 6; //light on, 1 dollar return, get nothing
47. }
48. else if(orangeJuice == 0 && beer == 1){
49. return 6; //light on, 1 dollar return, get nothing
50. }
51. else if(orangeJuice == 0 && beer == 0){
52. return 7; //light on, no money return, get nothing
53. }
54. else{
55. return 10;
56. }
57. }
58. else if(oneDollar == 0 && fiftyCents == 1){
59. if(orangeJuice == 1 && beer == 0){
60. return 8; //light on, no money return, get orangejuice
61. }
62. else if(orangeJuice == 0 && beer == 1){
63. return 9; //light on, no money return, get beer
64. }
65. else if(orangeJuice == 0 && beer == 0){
66. return 7;
67. }
68. else{
69. return 10;
70. }
71. }
72. else if(oneDollar == 0 && fiftyCents == 0){
73. if(orangeJuice == 1 && beer == 1){
74. return 10;
75. }
76. else{
77. return 7;
78. }
79. }
80. else{
81. return 10;
82. }
83. }
84. }

Program Graph





**Decision Tree Model**

****

**C1: Cover each DD-path**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | DD-Path Coverage | Input | Output(code) | Output(model) |
| 1 | P1 | 1,2,3,4,5 | 1 1 0 1 0 | 1 | 1 |
| P2 | 1,2,3,4,7,8 | 1 1 0 0 1 | 2 | 2 |
| P3 | 1,2,3,4,7,10,11 | 1 1 0 0 0 | 3 | 3 |
| P4 | 1,2,3,4,7,10,13,14 | 1 1 0 1 1 | 10 | 10 |
| P5 | 1,2,3,17,18,19 | 1 0 1 1 0 | 4 | 4 |
| P6 | 1,2,3,17,18,21,22 | 1 0 1 0 0 | 5 | 3 |
| P7 |  |  |  |  |
| P8 |  |  |  |  |
| P9 |  |  |  |  |
| P10 |  |  |  |  |
| P11 |  |  |  |  |
| P12 |  |  |  |  |
| P13 |  |  |  |  |
| P14 |  |  |  |  |
| P15 |  |  |  |  |
| P16 |  |  |  |  |