report_and_test

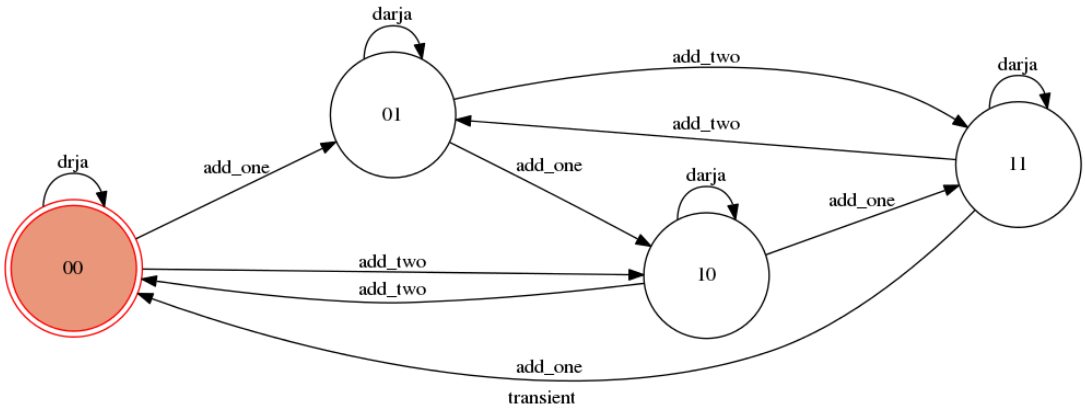
January 17, 2017

In [62]:
```
import os, sys, inspect

from Core.Diagrams.GraphMachine import GraphMachine

cmd_folder = os.path.realpath(
    os.path.dirname(
        os.path.abspath(os.path.split(inspect.getfile( inspect.currentframe() ))[0])))

if cmd_folder not in sys.path:
    sys.path.insert(0, cmd_folder)

from IPython.display import Image, display, display_png
from IPython.display import Image

class Bound(object):
    # create a model instance and boundle it to the machine so,
    # graph object is created by the machine
    def show_graph(self):
        self.graph.draw('state.png', prog='dot')
        display(Image('state.png'))
```

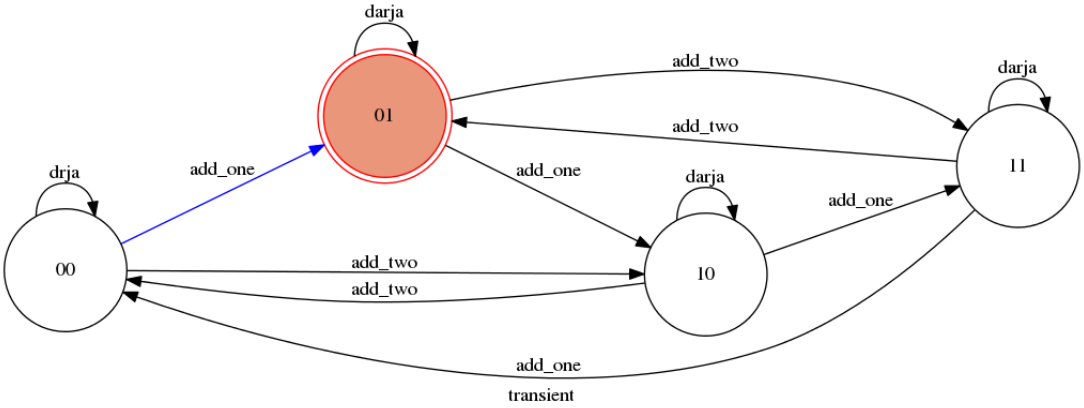
In [63]:
```
states=['00', '01', '10', '11']
transitions = [
    {'trigger': 'drja', 'source': '00', 'dest': '00'},
    {'trigger': 'add_one', 'source': '00', 'dest': '01'},
    {'trigger': 'add_two', 'source': '00', 'dest': '10'},
    {'trigger': 'darja', 'source': '01', 'dest': '01'},
    {'trigger': 'add_one', 'source': '01', 'dest': '10'},
    {'trigger': 'add_two', 'source': '01', 'dest': '11'},
    {'trigger': 'darja', 'source': '10', 'dest': '10'},
    {'trigger': 'add_one', 'source': '10', 'dest': '11'},
    {'trigger': 'add_two', 'source': '10', 'dest': '00'},
    {'trigger': 'darja', 'source': '11', 'dest': '11'},
    {'trigger': 'add_one', 'source': '11', 'dest': '00'},
    {'trigger': 'add_two', 'source': '11', 'dest': '01'}
]

model = Bound()
machine = GraphMachine(model=model,
                       states=states,
                       transitions=transitions,
                       auto_transitions=False,
                       initial='00',
                       title="transient",
```

```
                                  show_conditions=True)
```

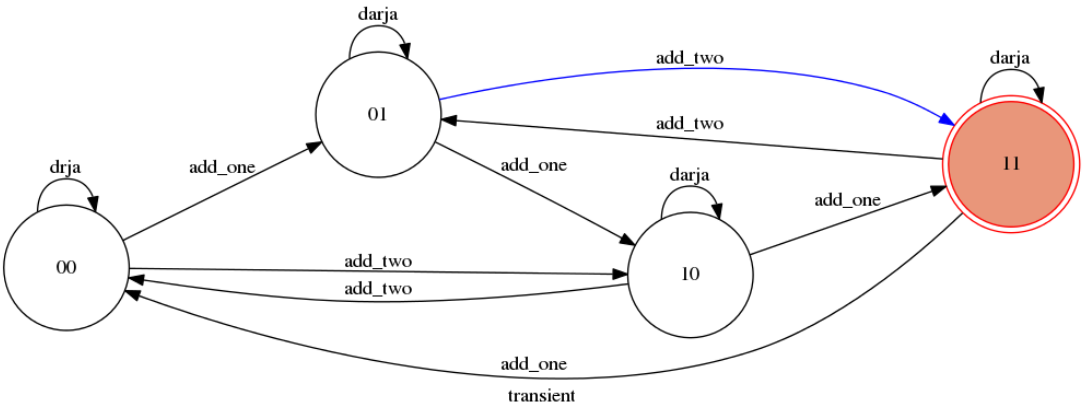
```
model.show_graph()
```



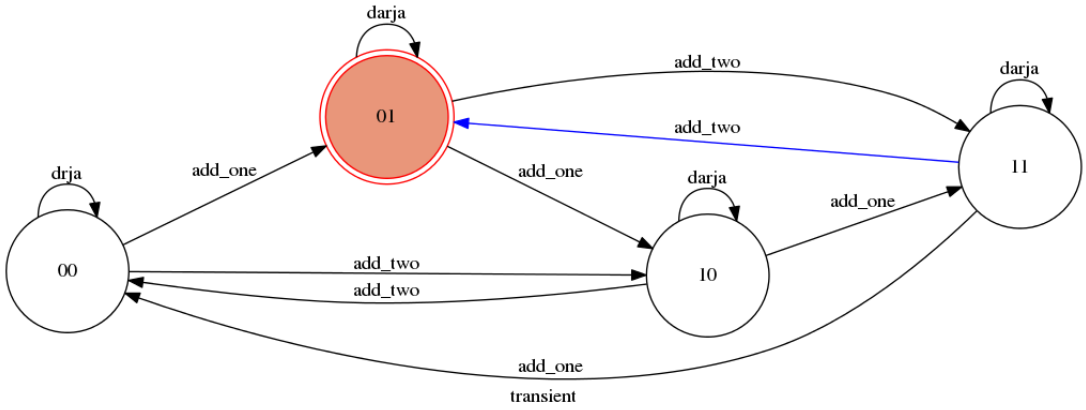
In [64]:
```
model.add_one()
model.show_graph()
```



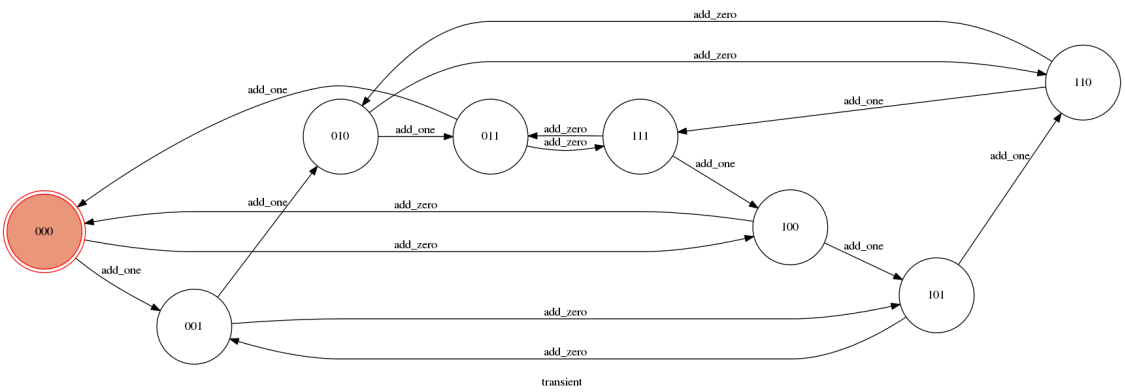
In [65]:
```
model.add_two()
model.show_graph()
```



In [66]:
```
model.add_two()
model.show_graph()
```



In [67]:
```
states=['000', '001', '010', '011', '100', '101', '110', '111']
transitions = [
    {'trigger': 'add_zero', 'source': '000', 'dest': '100'},
    {'trigger': 'add_one', 'source': '000', 'dest': '001'},
    {'trigger': 'add_zero', 'source': '001', 'dest': '101'},
    {'trigger': 'add_one', 'source': '001', 'dest': '010'},
    {'trigger': 'add_zero', 'source': '010', 'dest': '110'},
    {'trigger': 'add_one', 'source': '010', 'dest': '011'},
    {'trigger': 'add_zero', 'source': '011', 'dest': '111'},
    {'trigger': 'add_one', 'source': '011', 'dest': '000'},
    {'trigger': 'add_zero', 'source': '100', 'dest': '000'},
    {'trigger': 'add_one', 'source': '100', 'dest': '101'},
    {'trigger': 'add_zero', 'source': '101', 'dest': '001'},
    {'trigger': 'add_one', 'source': '101', 'dest': '110'},
    {'trigger': 'add_zero', 'source': '110', 'dest': '010'},
    {'trigger': 'add_one', 'source': '110', 'dest': '111'},
    {'trigger': 'add_zero', 'source': '111', 'dest': '011'},
    {'trigger': 'add_one', 'source': '111', 'dest': '100'},
]

model = Bound()
machine = GraphMachine(model=model,
                       states=states,
                       transitions=transitions,
                       auto_transitions=False,
                       initial='000',
                       title="transient",
                       show_conditions=True)

model.show_graph()
```




```
In [51]: r1 = input()
         r2 = input()
         r3 = input()
         r4 = input()
         r5 = input()
         r6 = input()
         r7 = input()
         r8 = input()
         r9 = input()
         r10 = input()
         r11 = input()
         r12 = input()
         r13 = input()
         r14 = input()
         r15 = input()
         r16 = input()
```

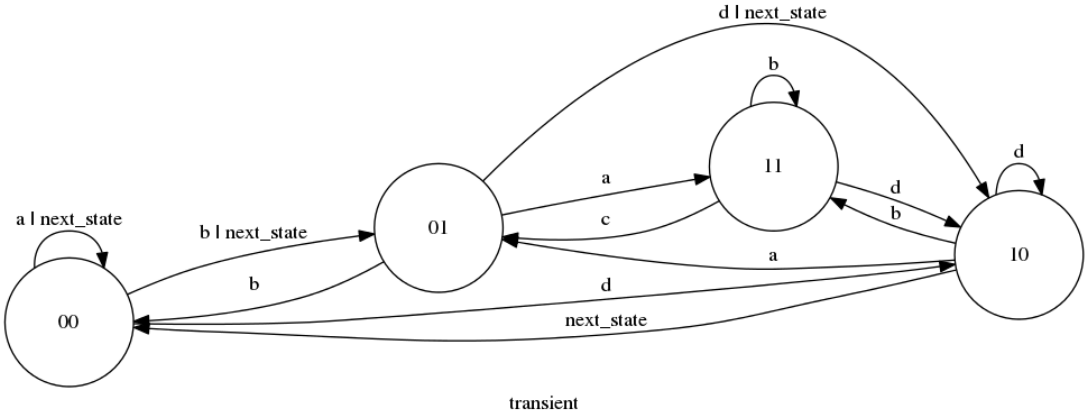
```
a
b
d

b

d
a

a
d
b

c
d
b
```

```
In [68]: string = '----------------------\n'+\
                  '|  State   |00|01|10|11|\n'+\
                  '----------------------\n'+\
                  '|    00    |{} |{} |{} |{} |\n'.format(r1, r2, r3, r4)+\
                  '|    01    |{} |{} |{} |{} |\n'.format(r5, r6, r7, r8)+\
```

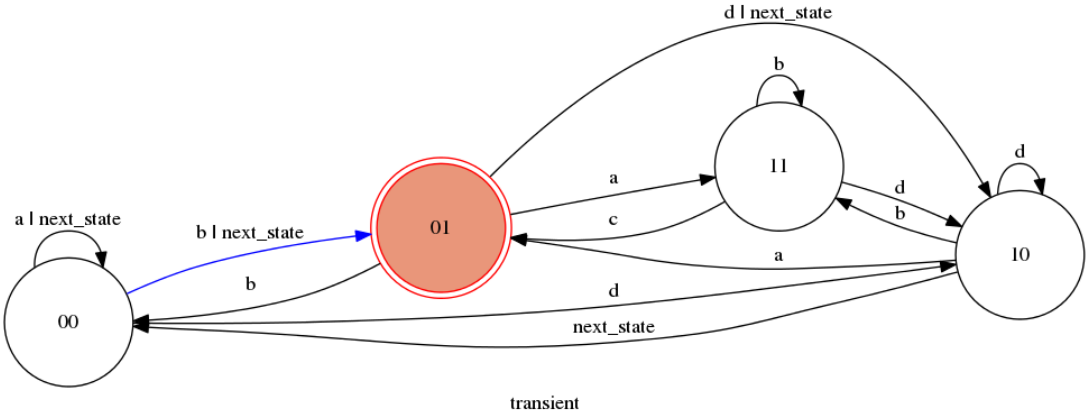
```
                '|    10    |{} |{} |{} |{} |\n'.format(r9, r10, r11, r12)+\
                '|    11    |{} |{} |{} |{} |\n'.format(r13, r14, r15, r16)

        print(string)

        states=['00', '01', '10', '11']

        transitions = [
            {'trigger': r1, 'source': '00', 'dest': '00'},
            {'trigger': r2, 'source': '00', 'dest': '01'},
            {'trigger': r3, 'source': '00', 'dest': '10'},
            {'trigger': r4, 'source': '00', 'dest': '11'},
            {'trigger': r5, 'source': '01', 'dest': '00'},
            {'trigger': r6, 'source': '01', 'dest': '01'},
            {'trigger': r7, 'source': '01', 'dest': '10'},
            {'trigger': r8, 'source': '01', 'dest': '11'},
            {'trigger': r9, 'source': '10', 'dest': '00'},
            {'trigger': r10, 'source':'10', 'dest': '01'},
            {'trigger': r11, 'source':'10', 'dest': '10'},
            {'trigger': r12, 'source':'10', 'dest': '11'},
            {'trigger': r13, 'source':'11', 'dest': '00'},
            {'trigger': r14, 'source':'11', 'dest': '01'},
            {'trigger': r15, 'source':'11', 'dest': '10'},
            {'trigger': r16, 'source':'11', 'dest': '11'},
        ]

        for el in transitions:
            if el['trigger'] is "":
                print(el)
                transitions.remove(el)

        model = Bound()
        machine = GraphMachine(model=model,
                               states=states,
                               transitions=transitions,
                               auto_transitions=False,
                               initial='00',
                               title="transient",
                               # ordered_transitions=True,
                               show_conditions=True)

        l = ['00', '01', '10', '00']
        machine.add_ordered_transitions(l)

        model.show_graph()
```

```
-----------------------
|  State   |00|01|10|11|
-----------------------
|    00    |a |b |d | |
|    01    |b | |d |a |
|    10    | |a |d |b |
|    11    | |c |d |b |
```

```
{'dest': '11', 'trigger': '', 'source': '00'}
{'dest': '01', 'trigger': '', 'source': '01'}
{'dest': '00', 'trigger': '', 'source': '10'}
{'dest': '00', 'trigger': '', 'source': '11'}
```

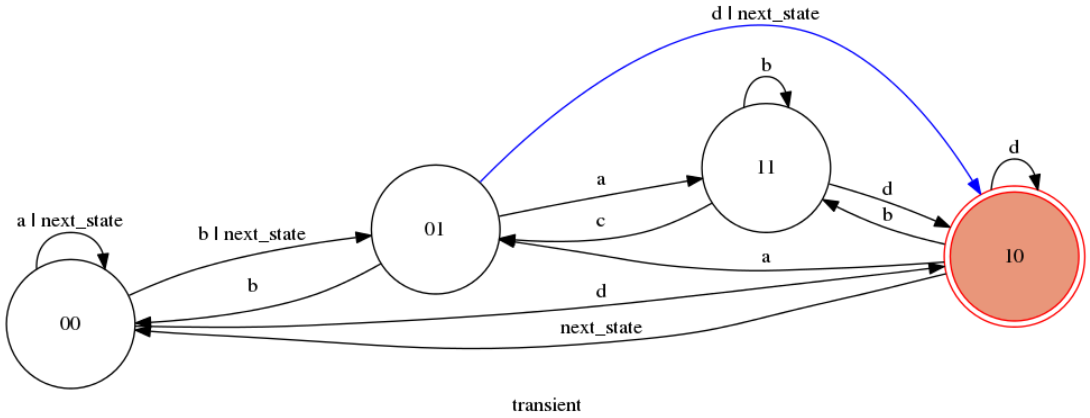



```
In [69]: model.next_state()
         model.show_graph()
         print(model.state)
```



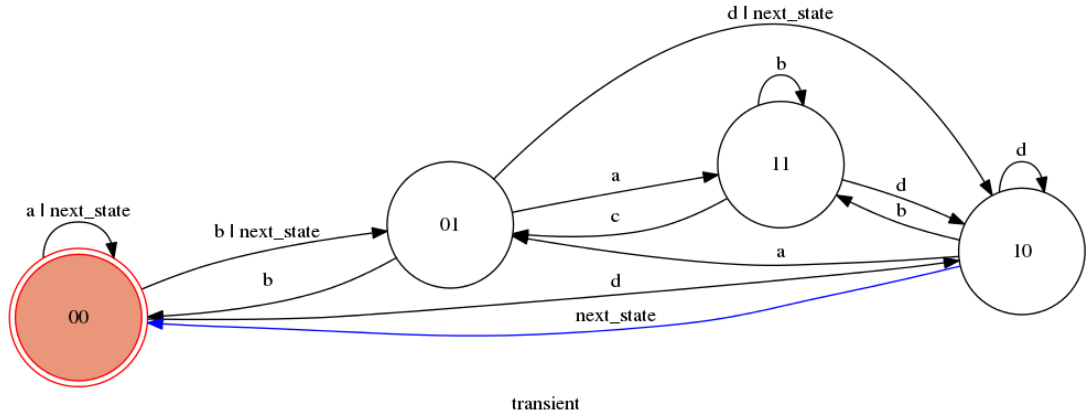

```
01
```

```
In [70]: model.next_state()
         model.show_graph()
         print(model.state)
```



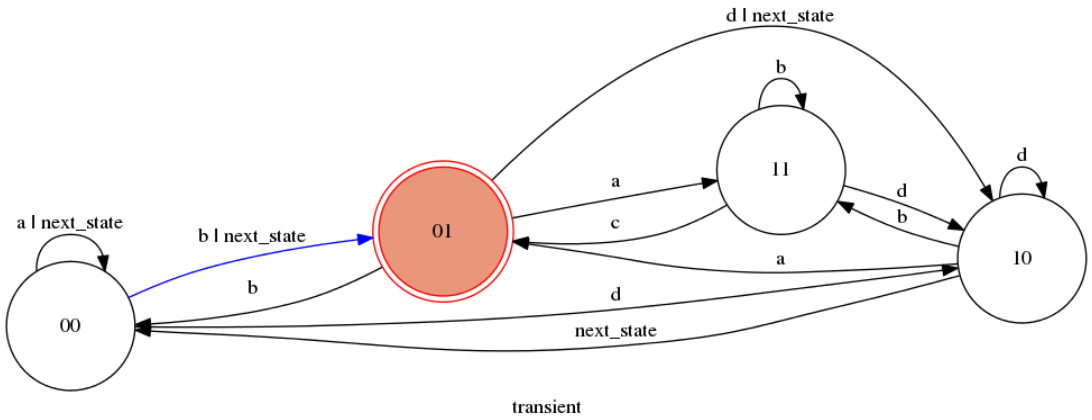
10

In [71]:
```
model.next_state()
model.show_graph()
print(model.state)
```



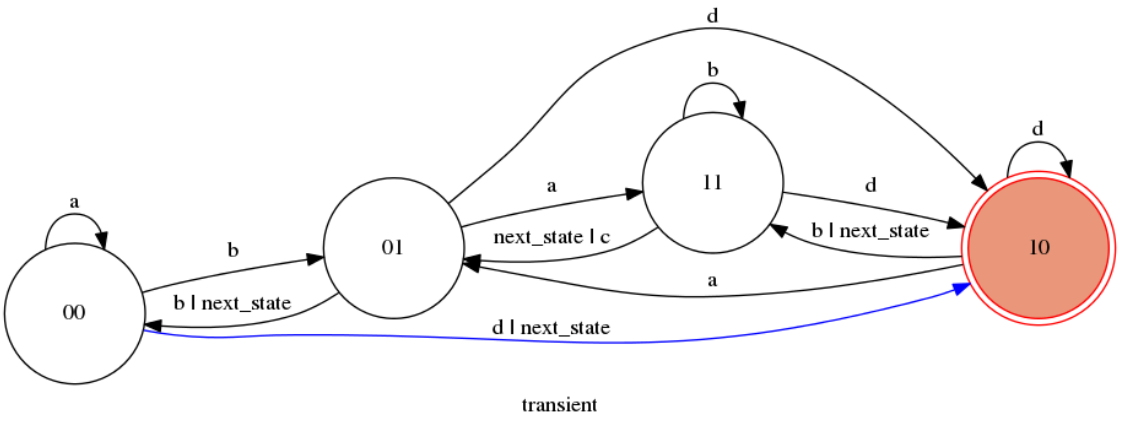
00

In [72]:
```
model.next_state()
model.show_graph()
print(model.state)
```



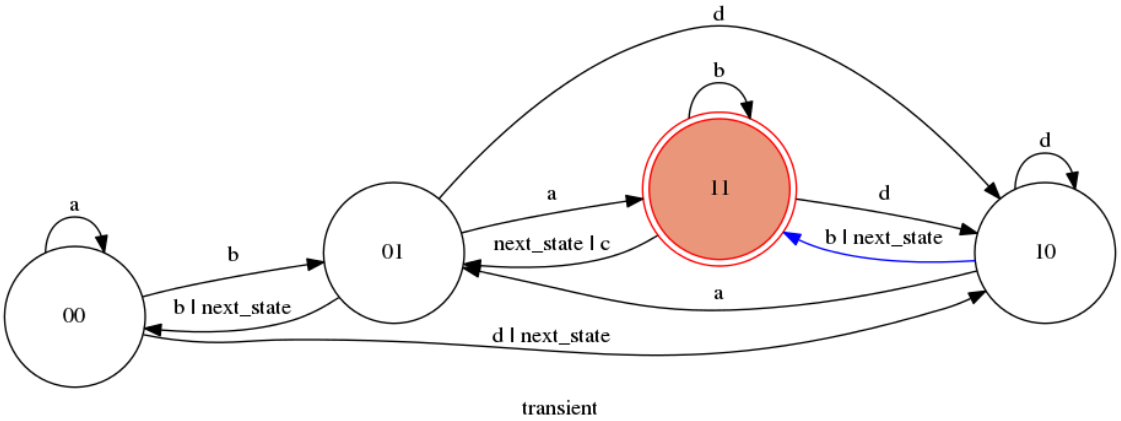
01

In [73]:
```
model = Bound()
machine = GraphMachine(model=model,
                       states=states,
                       transitions=transitions,
                       auto_transitions=False,
                       initial='00',
                       title="transient",
                       # ordered_transitions=True,
                       show_conditions=True)

l = ['00', '10', '11', '01']
machine.add_ordered_transitions(l)
model.next_state()
model.show_graph()
print(model.state)
model.next_state()
model.show_graph()
print(model.state)
model.next_state()
model.show_graph()
print(model.state)
model.next_state()
model.show_graph()
print(model.state)
model.next_state()
model.show_graph()
print(model.state)
model.next_state()
model.show_graph()
print(model.state)
```



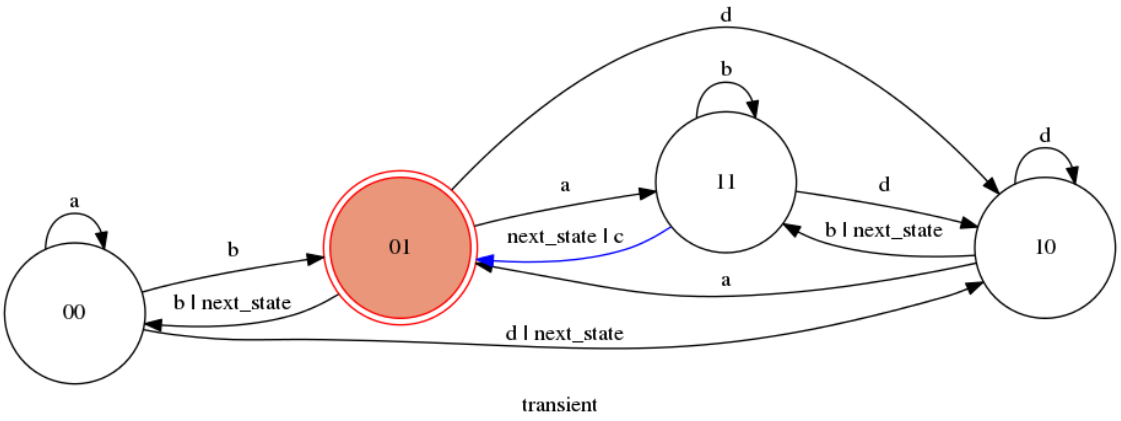
transient

10



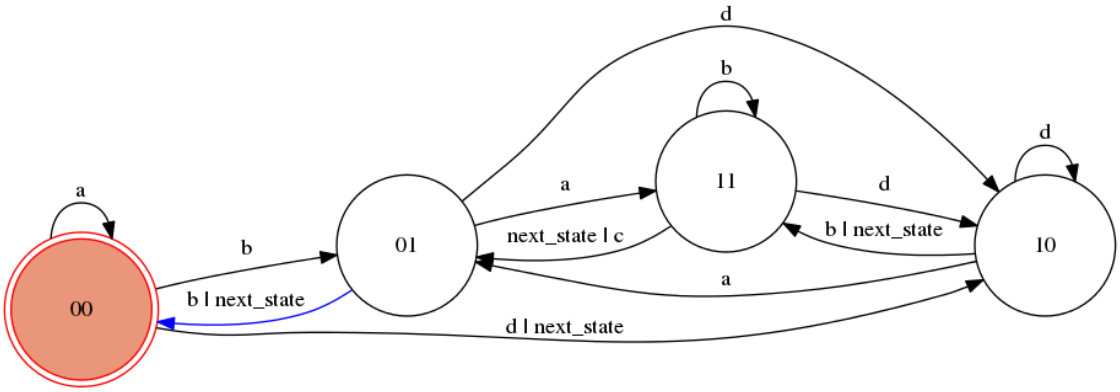
transient

11



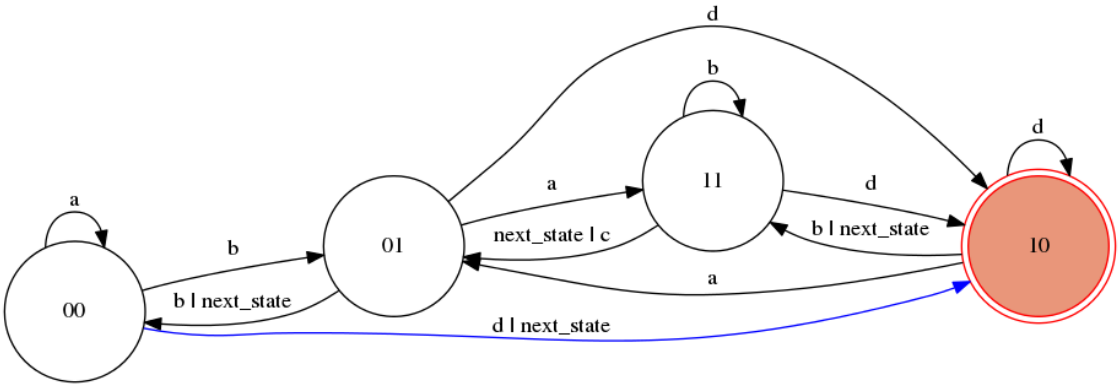
transient

01



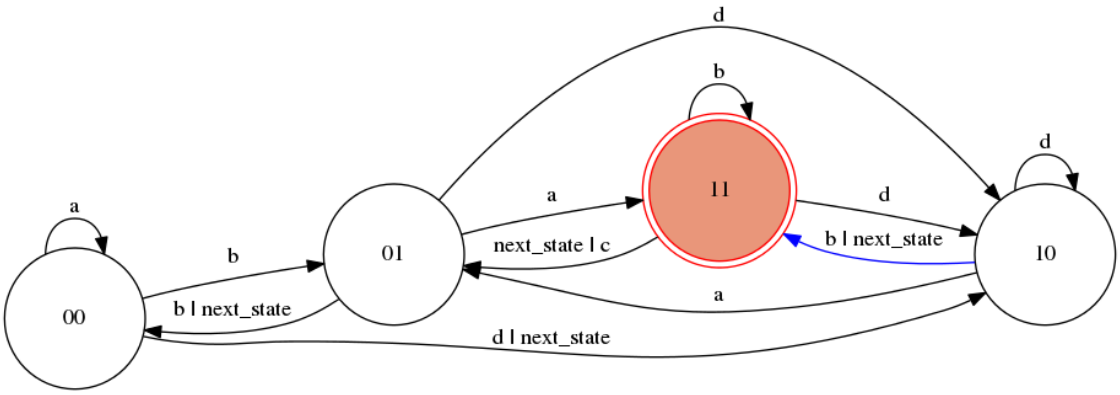
transient

00



transient

10



transient

```
11
```

In [ ]: