Mini models

EECS 4315

wiki.eecs.yorku.ca/course/4315/

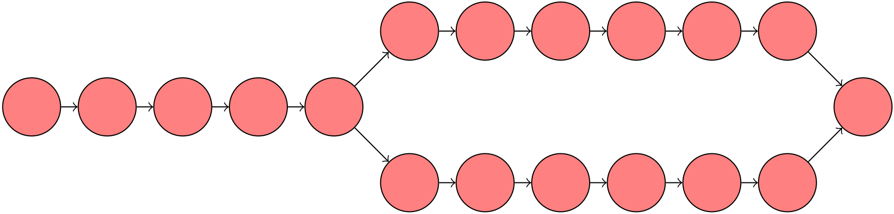
Mini

source: Keld Gydum

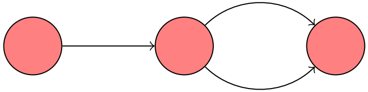
Mini

source: Mike Bird

Model

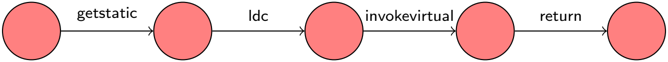


Mini model

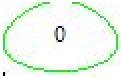


Hello World!

```
public class HelloWorld {
  public static void main(String[] args) {
    System.out.println("Hello World");
  }
}
```



Hello World!



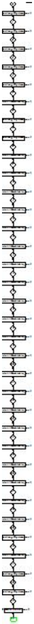
Hello World!

```
target=HelloWorld
classpath=.
listener=gov.nasa.jpf.listener.StateSpaceDot
vm.max_transition_length=1
```

Hello World!

```
====================================== stat
elapsed time:      00:00:02
states:            new=35,visited=0,backtracked=35,end=1
search:            maxDepth=35,constraints=0
choice generators: thread=35 (signal=0,lock=1,sharedRef=0,t
heap:              new=348,released=11,maxLive=331,gcCycles=
instructions:      3198
max memory:        61MB
loaded code:       classes=56,methods=1220
====================================== sear
```

Hello World!



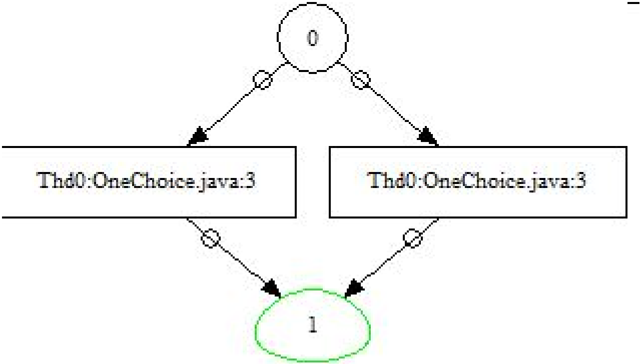
One choice

```
Random random = new Random();
if (random.nextBoolean()) {
  System.out.println("1");
} else {
  System.out.println("2");
}
```

One choice

```
target=OneChoice
classpath=.
cg.enumerate_random=true
listener=gov.nasa.jpf.listener.StateSpaceDot
```

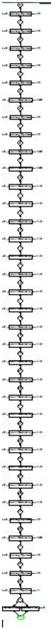
One choice



One choice

```
target=OneChoice
classpath=.
cg.enumerate_random=true
listener=gov.nasa.jpf.listener.StateSpaceDot
vm.max_transition_length=1
```

One choice



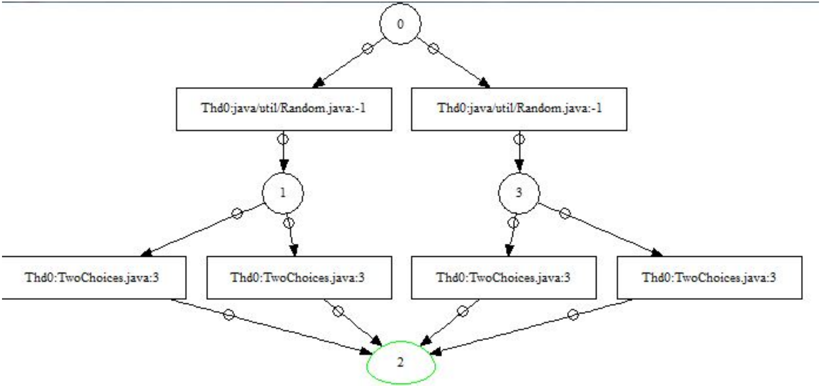
Two choices

```
Random random = new Random();
if (random.nextBoolean()) {
  if (random.nextBoolean()) {
    System.out.println("1");
  } else {
    System.out.println("2");
  }
} else {
  if (random.nextBoolean()) {
    System.out.println("3");
  } else {
    System.out.println("4");
  }
}
```

Two choices

```
target=TwoChoices
classpath=.
cg.enumerate_random=true
listener=gov.nasa.jpf.listener.StateSpaceDot
```

Two choices

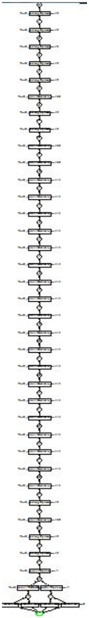


0
Thd0:java/util/Random.java:-1
Thd0:java/util/Random.java:-1
1
3
Thd0:TwoChoices.java:3
Thd0:TwoChoices.java:3
Thd0:TwoChoices.java:3
Thd0:TwoChoices.java:3
2

Two choices

```
target=TwoChoices
classpath=.
cg.enumerate_random=true
listener=gov.nasa.jpf.listener.StateSpaceDot
vm.max_transition_length=1
```

Two choices



Many choices

```java
Random random = new Random();
byte value = 0;
while (random.nextBoolean()) {
  value++;
}
System.out.println(value);
```

Many choices

Question

How many different executions does the app `ManyChoices` have?

Many choices

Question

How many different executions does the app `ManyChoices` have?

Answer

Infinitely many.

Many choices

Question

How many different executions does the app `ManyChoices` have?

Answer

Infinitely many.

Question

How many different states does JPF encounter?

Many choices

Question

How many different executions does the app `ManyChoices` have?

Answer

Infinitely many.

Question

How many different states does JPF encounter?

Answer

257.

Not so many choices

```
Random random = new Random();
byte value = 0;
while (random.nextBoolean()) {
  value = (byte) ((value + 1) % 5);
}
System.out.println(value);
```

Not so many choices

Question

How many different executions does the app `ManyChoices` have?

Not so many choices

Question

How many different executions does the app `ManyChoices` have?

Answer

Infinitely many.

Not so many choices

Question

How many different executions does the app `ManyChoices` have?

Answer

Infinitely many.

Question

How many different states does JPF encounter?

Not so many choices

Question

How many different executions does the app `ManyChoices` have?

Answer

Infinitely many.

Question

How many different states does JPF encounter?

Answer

6.

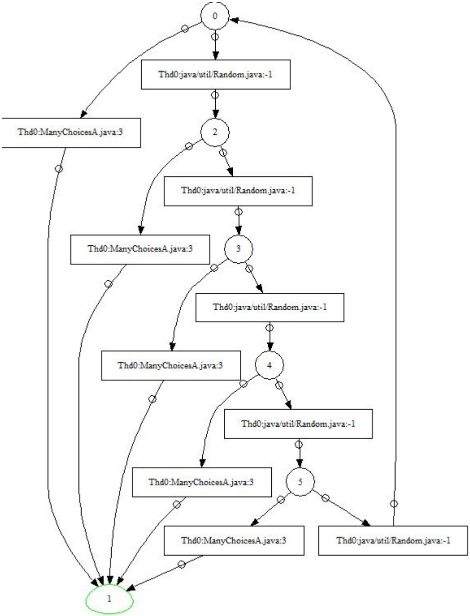
Not so many choices

```
target=NotSoManyChoices
classpath=.
cg.enumerate_random=true
listener=gov.nasa.jpf.listener.StateSpaceDot
```

Not so many choices

```
=========================================================== sear
0
1
2
3
4
```

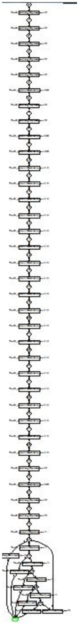
Not so many choices



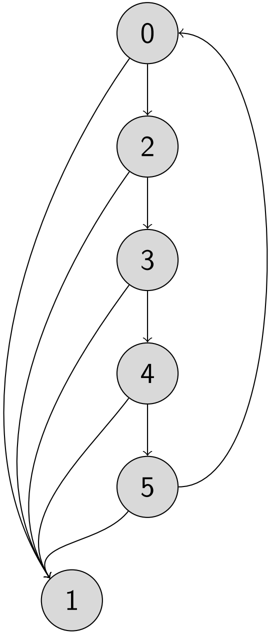
Not so many choices

```
target=NoSoManyChoices
classpath=.
cg.enumerate_random=true
listener=gov.nasa.jpf.listener.StateSpaceDot
vm.max_transition_length=1
```

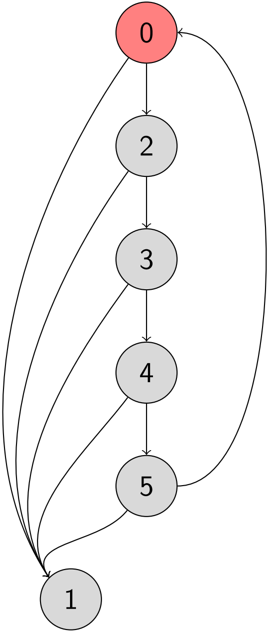
Not so many choices



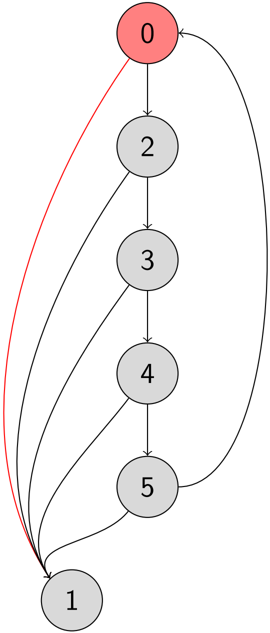
Not so many choices



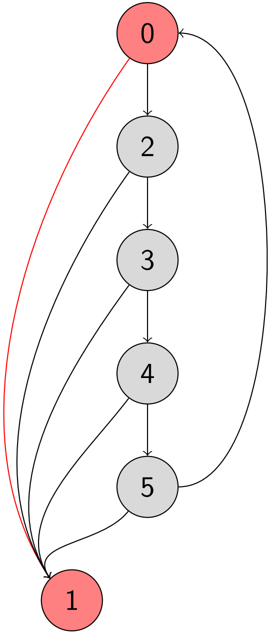
Not so many choices



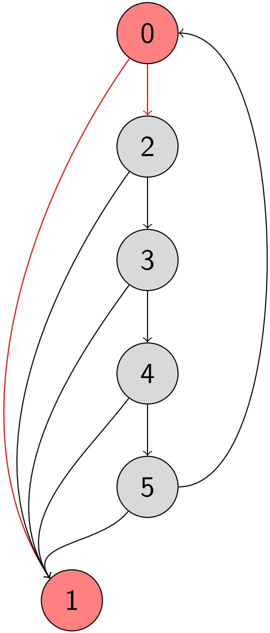
Not so many choices



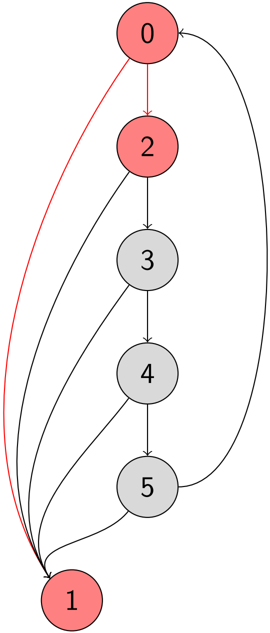
Not so many choices



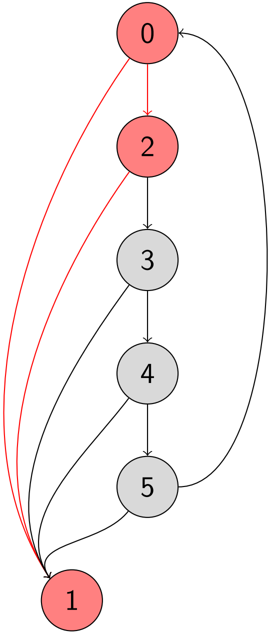
Not so many choices



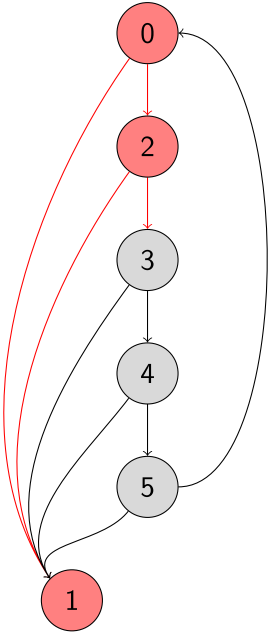
Not so many choices



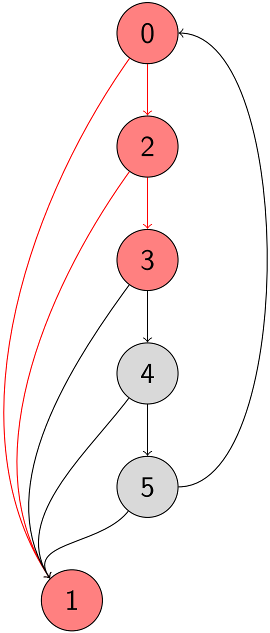
Not so many choices



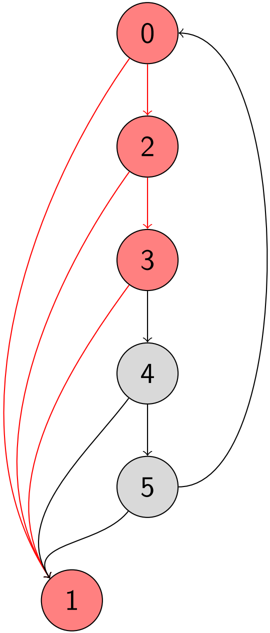
Not so many choices



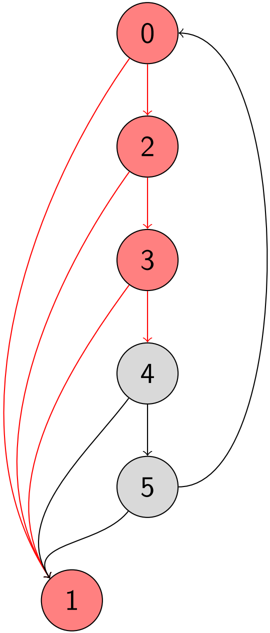
Not so many choices



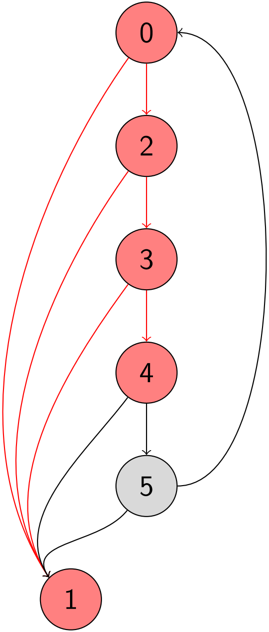
Not so many choices



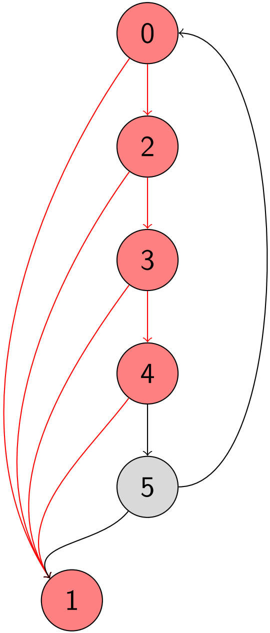
Not so many choices



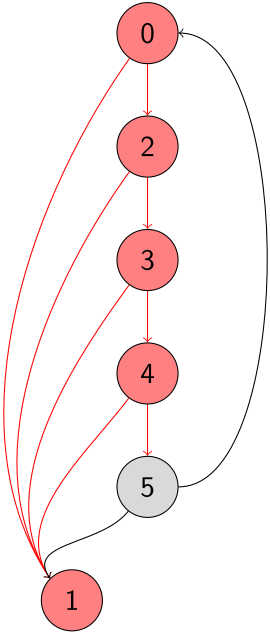
Not so many choices



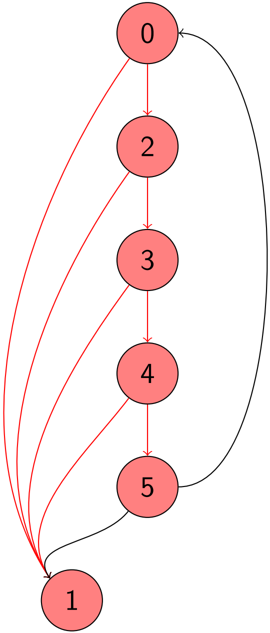
Not so many choices



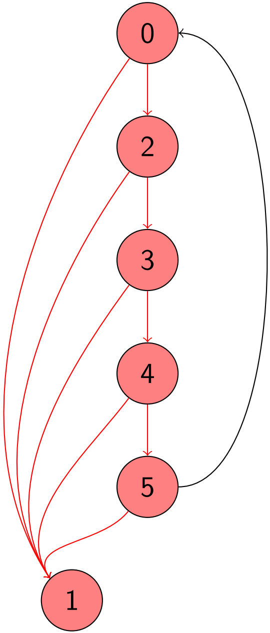
Not so many choices



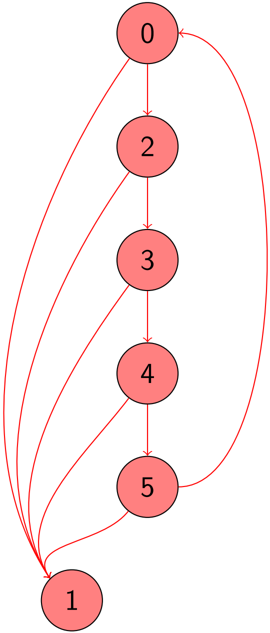
Not so many choices



Not so many choices



Not so many choices



Not so many choices

Question

Does this remind you of an algorithm you have seen in the course EECS 2011 Fundamentals of Data Structures and EECS 3101 Design and Analysis of Algorithms?

Not so many choices

Question

Does this remind you of an algorithm you have seen in the course EECS 2011 Fundamentals of Data Structures and EECS 3101 Design and Analysis of Algorithms?

Answer

Depth-first search of a directed graph.

Not so many choices

Question

Does this remind you of an algorithm you have seen in the course EECS 2011 Fundamentals of Data Structures and EECS 3101 Design and Analysis of Algorithms?

Answer

Depth-first search of a directed graph.

A labelled transition system is similar to a directed graph.

state	vertex
transition	edge

Question

Why do we have to keep track of the vertices that have been visited in depth-first search?

Depth-first search

Question

Why do we have to keep track of the vertices that have been visited in depth-first search?

Answer

To ensure that the traversal terminates.

Depth-first search

Question

Why do we have to keep track of the vertices that have been visited in depth-first search?

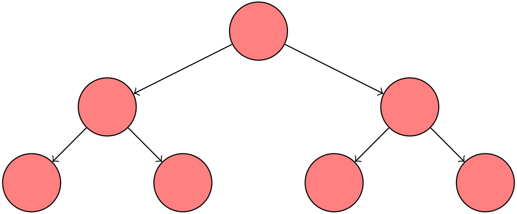
Answer

To ensure that the traversal terminates.

Similarly, when model checking we need to keep track of the states that have already been visited.

Exercise

Write a recursive method that for a given depth d chooses an integer in the range $1 - 2^d$ uniformly at random using `random.nextBoolean`. Hint: provide the method with an additional parameter.



Number of states

```
target=Choice
target.args=2
classpath=.
```

Number of states

d	number of states
0	35
1	36
2	38
3	42
4	50
5	66
10	1,058
20	1,048,610

Number of states

d	number of states
0	35
1	36
2	38
3	42
4	50
5	66
10	1,058
20	1,048,610

Question

Can you express the number of states in terms of d?

Number of states

d	number of states
0	35
1	36
2	38
3	42
4	50
5	66
10	1,058
20	1,048,610

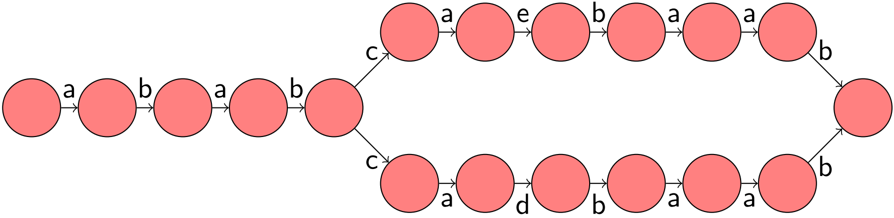
Question

Can you express the number of states in terms of d?

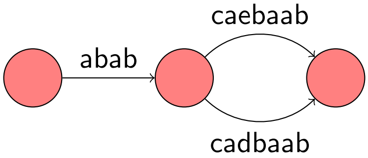
Answer

$2^d + 34$.

Model



Mini model



Question

What do the model and the mini model in common?

Question

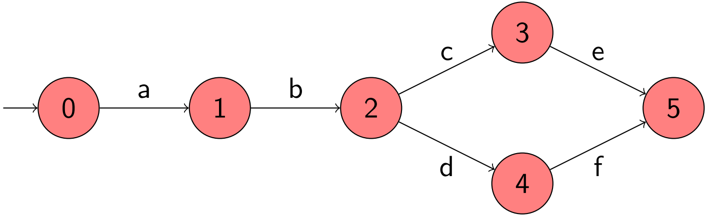
What do the model and the mini model in common?

Answer

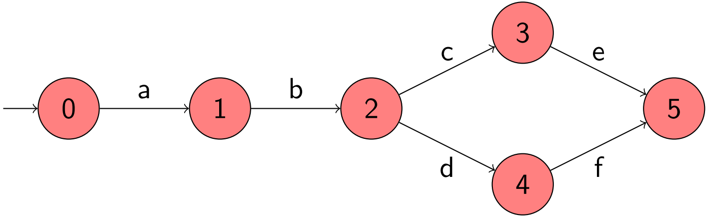
- The initial state.
- The final states.
- The branching structure.
- The language: (finite and infinite) sequences of actions.[a]

[a] Similar to the language accepted by a finite automaton, as discussed in EECS 2001 Introduction to Theory of Computation.

Model



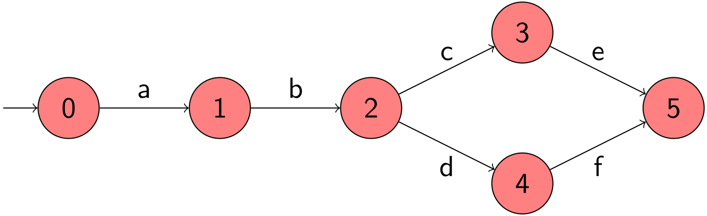
Model



Question

Which is the initial state?

Model



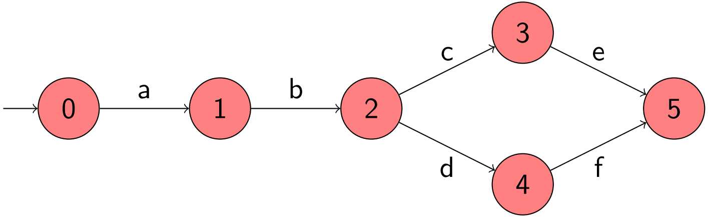
Question

Which is the initial state?

Answer

State 0.

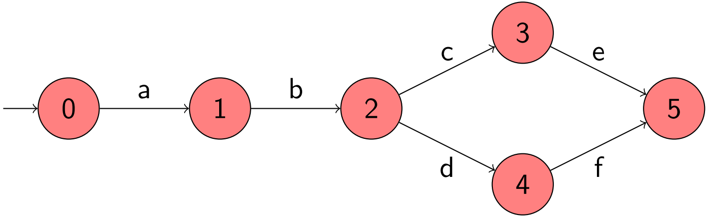
Model



Question

Which are the final states?

Model



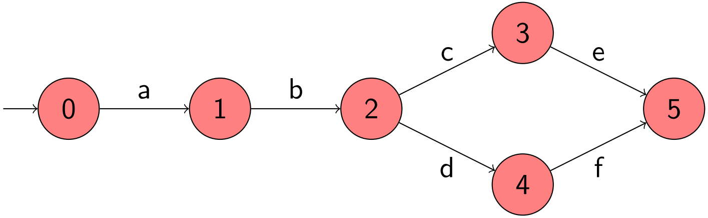
Question

Which are the final states?

Answer

State 5.

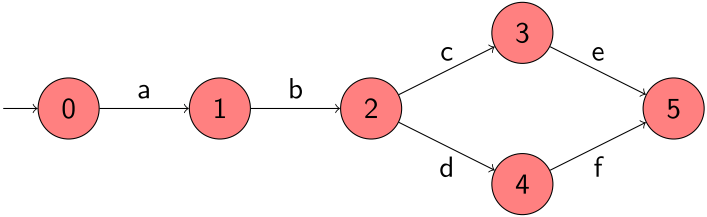
Model



Question

Which are the branching states?

Model



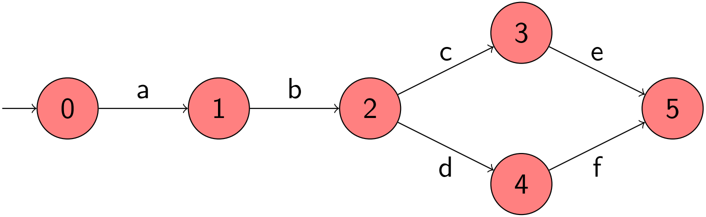
Question

Which are the branching states?

Answer

State 2.

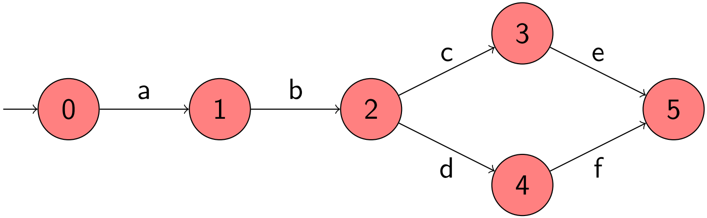
Model



Question

What is the language?

Model



Question

What is the language?

Answer

$\{abce, abdf\}$.

Mini model

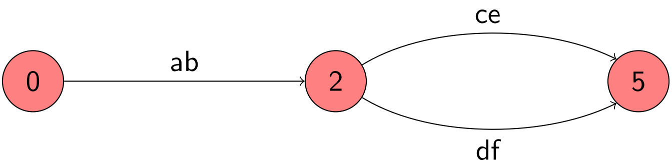
Question

What is the corresponding mini model?

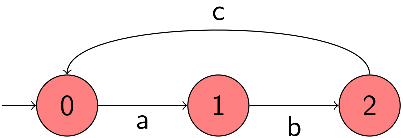
Mini model

Question

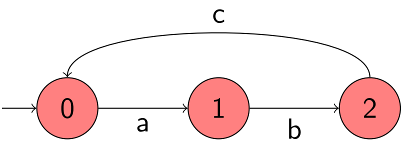
What is the corresponding mini model?



Model



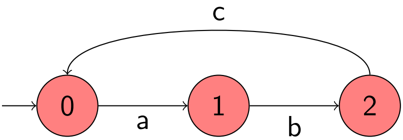
Model



Question

Which is the initial state?

Model



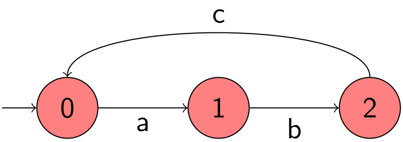
Question

Which is the initial state?

Answer

State 0.

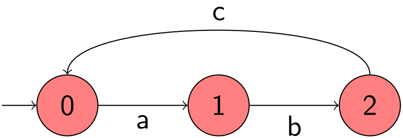
Model



Question

Which are the final states?

Model



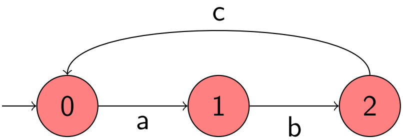
Question

Which are the final states?

Answer

There are none.

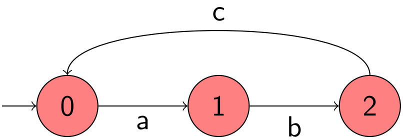
Model



Question

Which are the branching states?

Model

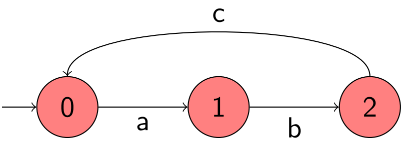


Question

Which are the branching states?

Answer

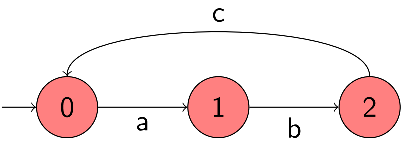
There are none.



Question

What is the language?

Model



Question

What is the language?

Answer

$\{abcabcabc \ldots\}$.

Mini model

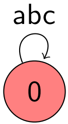
Question

What is the corresponding mini model?

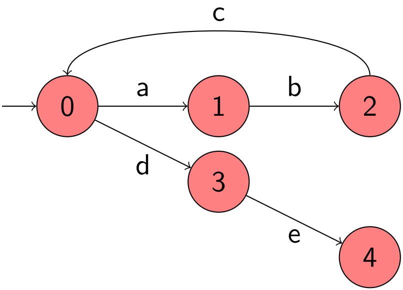
Mini model

Question

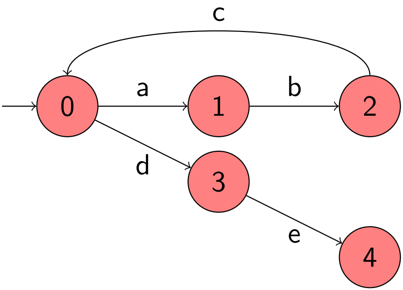
What is the corresponding mini model?



Model



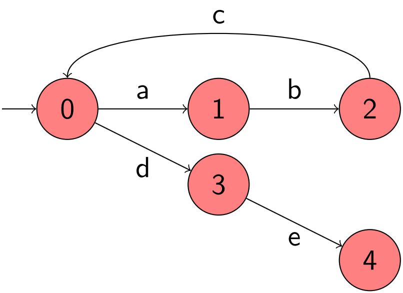
Model



Question

Which is the initial state?

Model



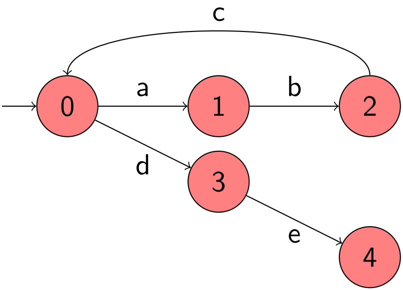
Question

Which is the initial state?

Answer

State 0.

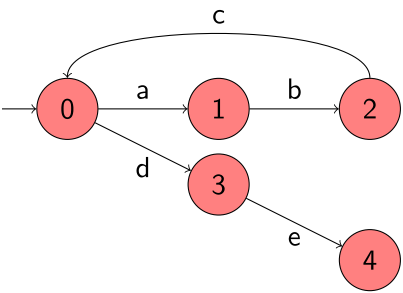
Model



Question

Which are the final states?

Model



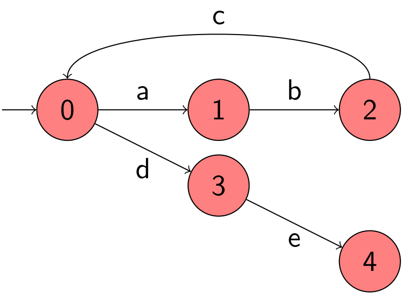
Question

Which are the final states?

Answer

State 4.

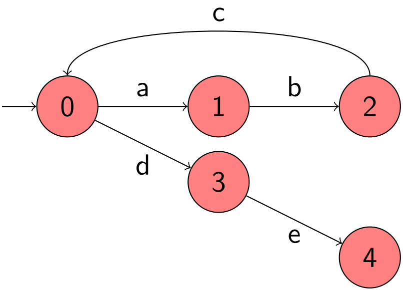
Model



Question

Which are the branching states?

Model



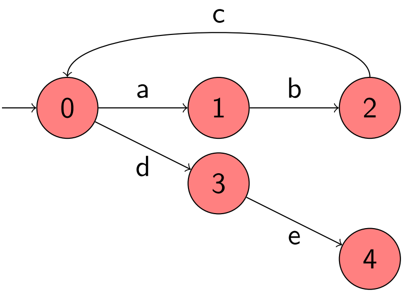
Question

Which are the branching states?

Answer

State 0.

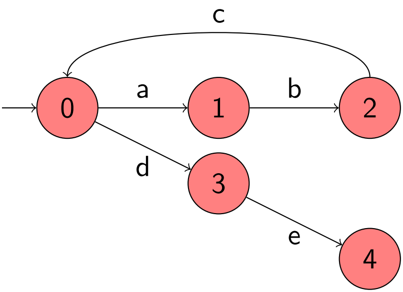
Model



Question

What is the language?

Model



Question

What is the language?

Answer

$\{de, abcde, abcabcde, \ldots, abcabcabc \ldots\}$.

Mini model

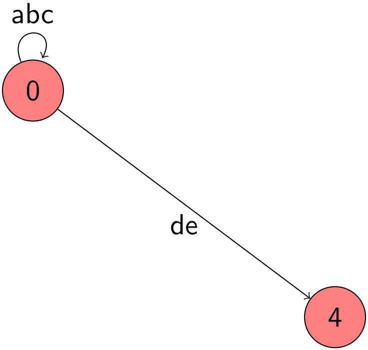
Question

What is the corresponding mini model?

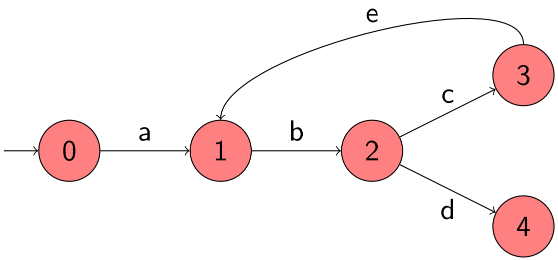
Mini model

Question

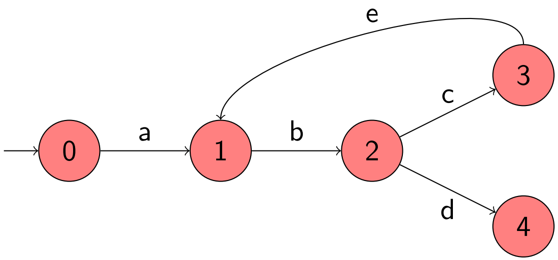
What is the corresponding mini model?



Model



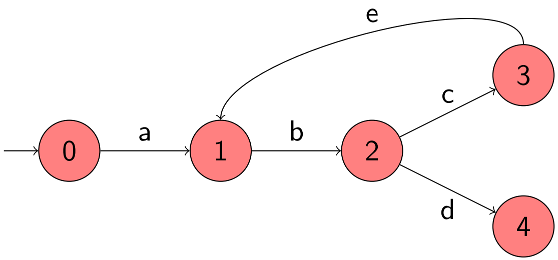
Model



Question

Which is the initial state?

Model



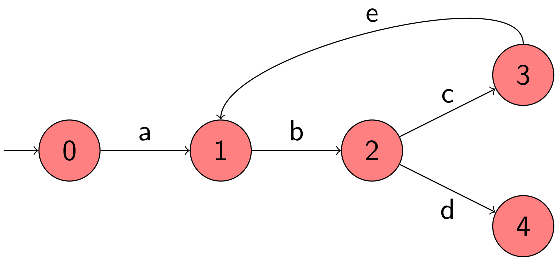
Question

Which is the initial state?

Answer

State 0.

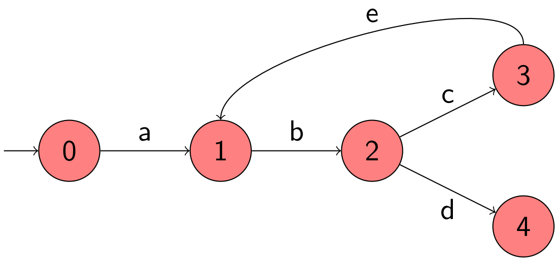
Model



Question

Which are the final states?

Model



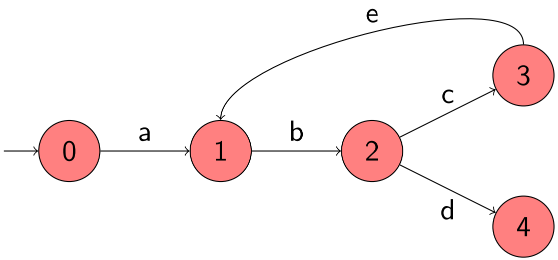
Question

Which are the final states?

Answer

State 4.

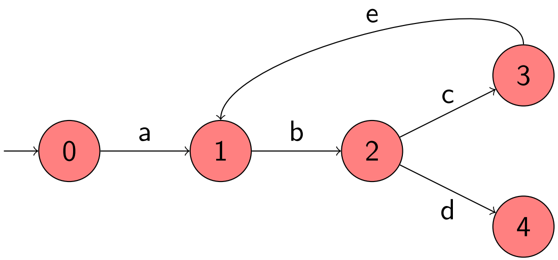
Model



Question

Which are the branching states?

Model



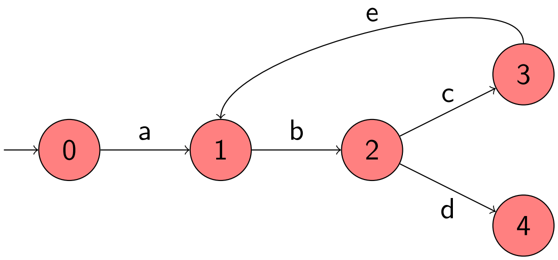
Question

Which are the branching states?

Answer

State 2.

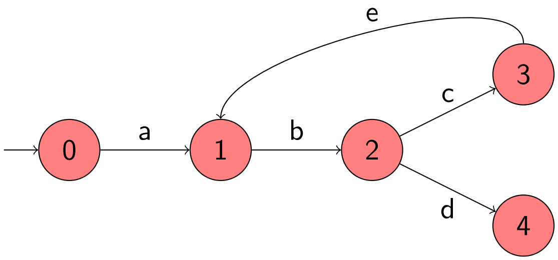
Model



Question

What is the language?

Model



Question

What is the language?

Answer

$\{abd, abcebd, abcebcebd, \ldots, abcebcebce \ldots\}$.

Mini model

Question

What is the corresponding mini model?

Mini model

Question

What is the corresponding mini model?