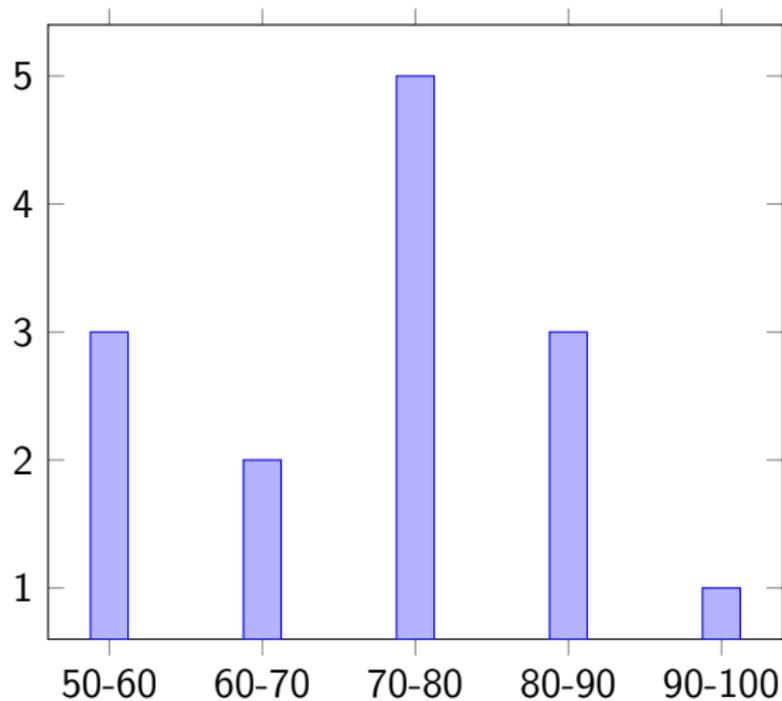


Quiz 2: grade distribution



Average: 73%

Quiz 2: Questions with low average scores

Question 2b: 36%

How is this entity (an annotation) used in the class?

Question 4c: 21%

Give an estimate of the (yearly world-wide) cost of debugging.

Question 8a: 46%

Mention two advantages of model checking (in comparison with theorem proving).

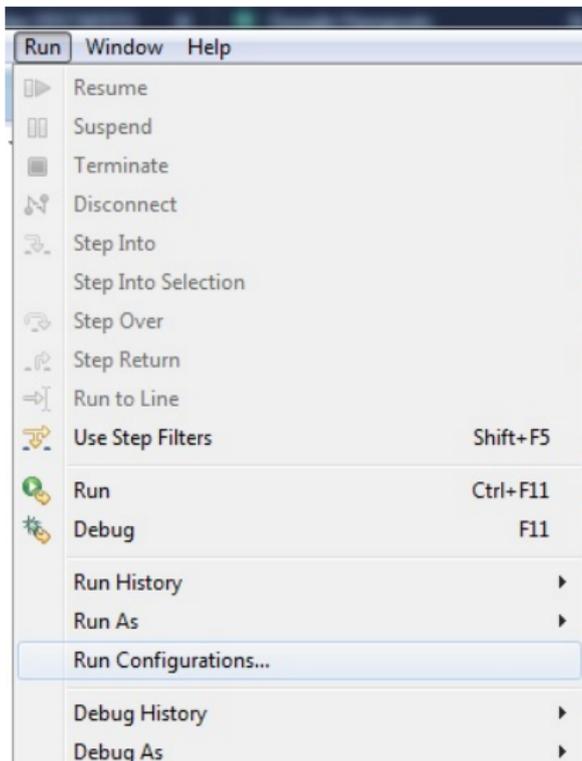
Question 8b: 54%

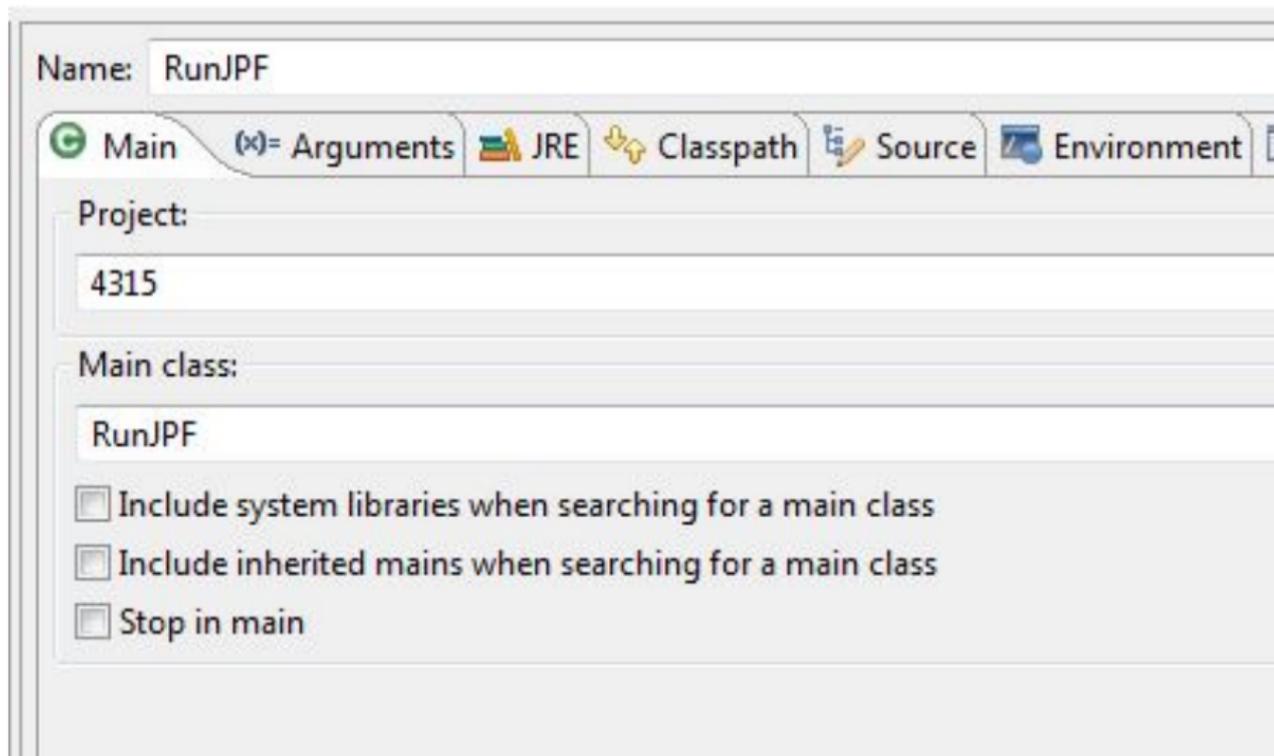
Mention two disadvantages of model checking (in comparison with theorem proving).

Create the following app.

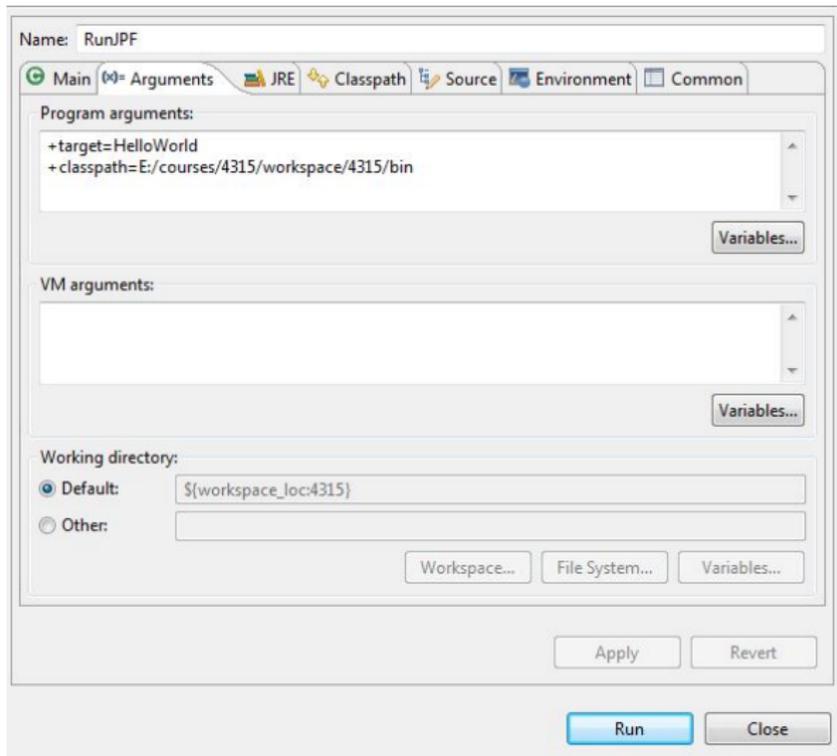
```
public class RunJPF {  
    public static void main(String[] args) {  
        // location of .jpf directory  
        final String HOME = "/courses/4315/";  
        System.setProperty("user.home", HOME);  
        gov.nasa.jpf.tool.RunJPF.main(args);  
    }  
}
```

Eclipse: Run → Run Configurations ...





Eclipse: Run → Run Configurations ... → Arguments



`https://www.graphviz.org/`

- January 25: install JPF (5%)
- February 15: draft proposal (2%)
- February 25: proposal (3%)
- March 8: first progress report (5%)
- March 22: second progress report (5%)
- Exam period: deliverables (20%)

Very brief descriptions of the last three years' projects can be found [here](#).

Students can work alone or in groups of two on their project.

Students are expected to work on average two hours per week on their projects from now on.

Mini models

EECS 4315

www.eecs.yorku.ca/course/4315/

Question

What do the model and the mini model have in common?

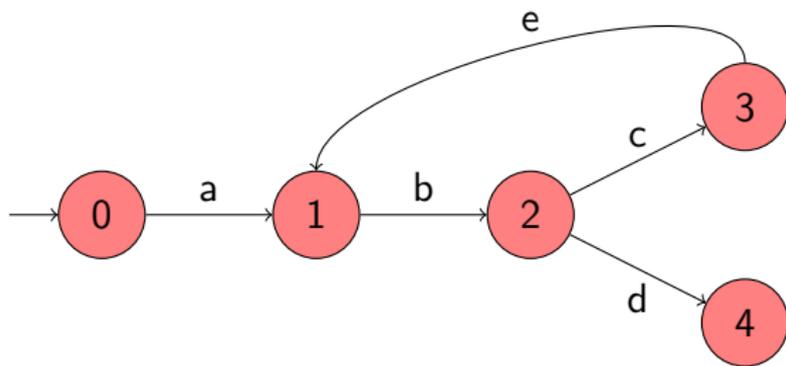
Question

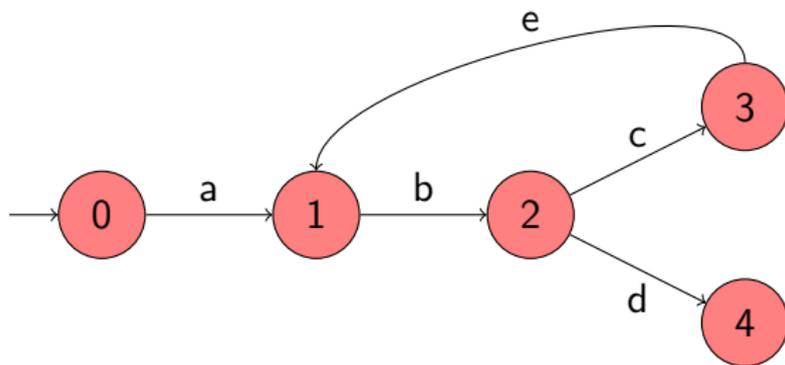
What do the model and the mini model have in common?

Answer

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

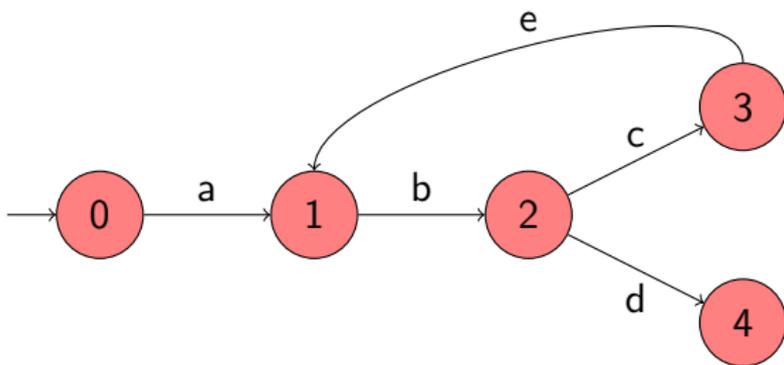
^aSimilar to the language accepted by a finite automaton, as discussed in EECS 2001 Introduction to Theory of Computation.





Question

Which is the initial state?

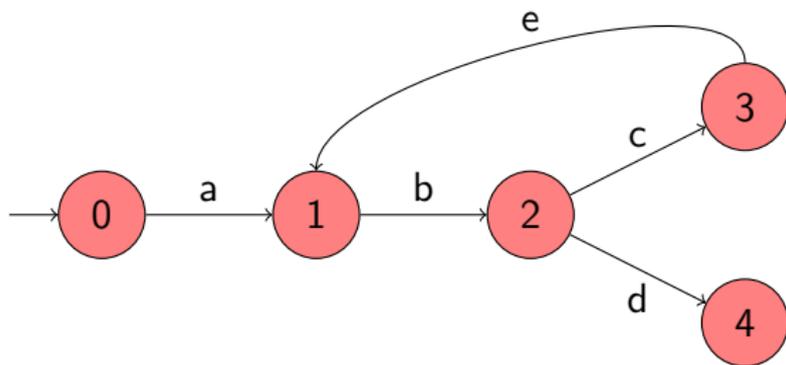


Question

Which is the initial state?

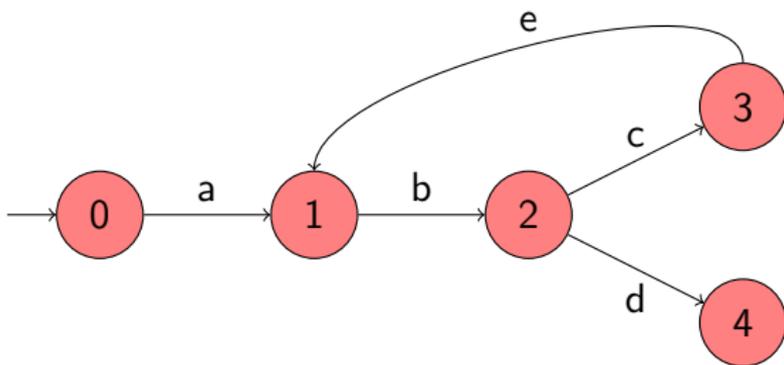
Answer

State 0.



Question

Which are the final states?

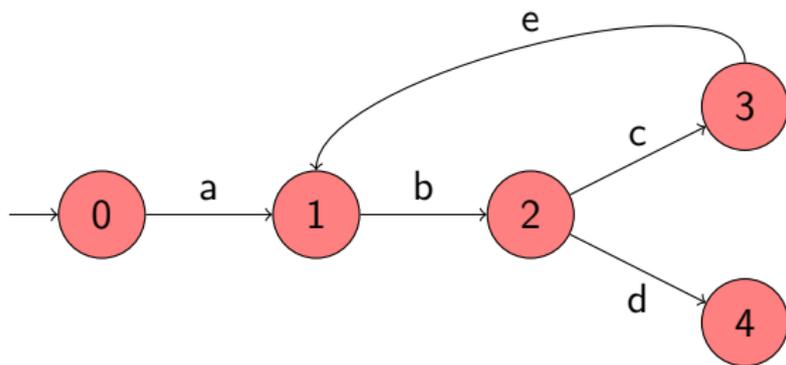


Question

Which are the final states?

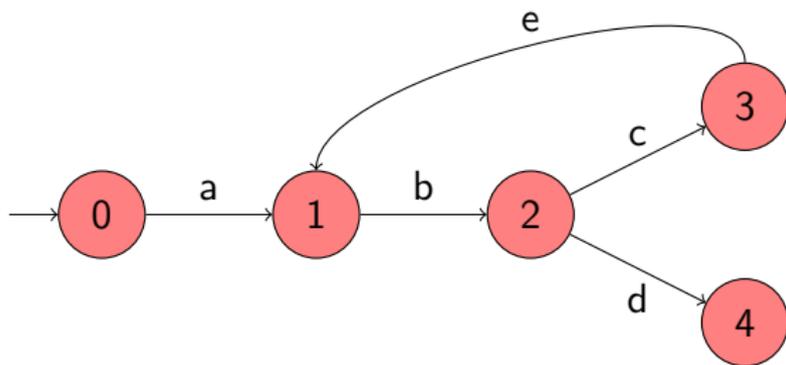
Answer

State 4.



Question

Which are the branching states?

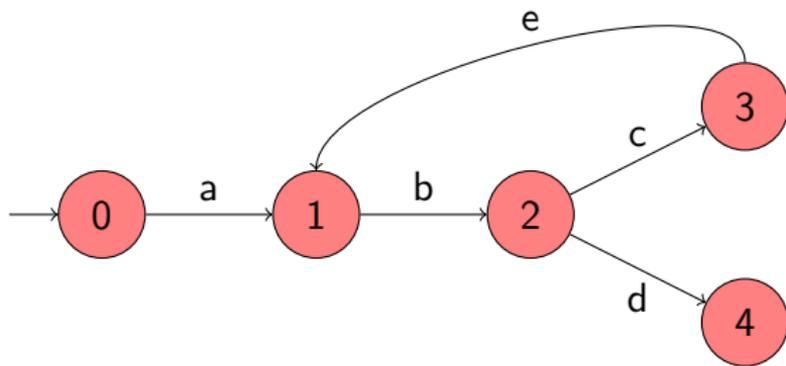


Question

Which are the branching states?

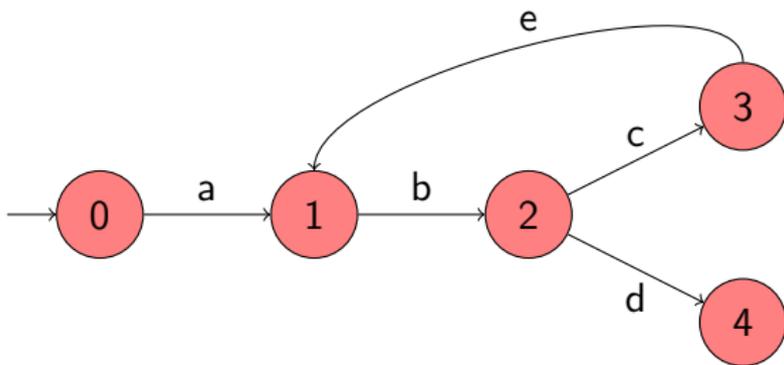
Answer

State 2.



Question

What is the language?



Question

What is the language?

Answer

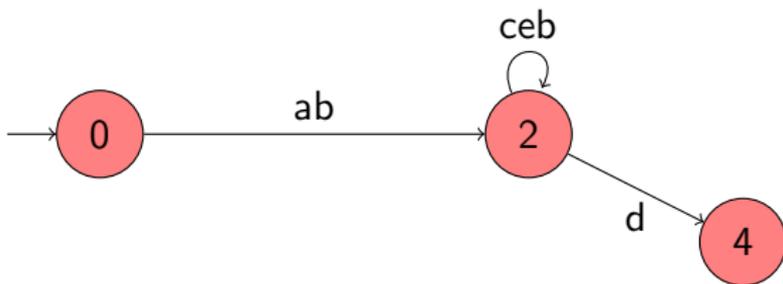
$\{abd, abcebd, abcebcebd, \dots, abcebcebcce \dots\}$.

Question

What is the corresponding mini model?

Question

What is the corresponding mini model?



Definition

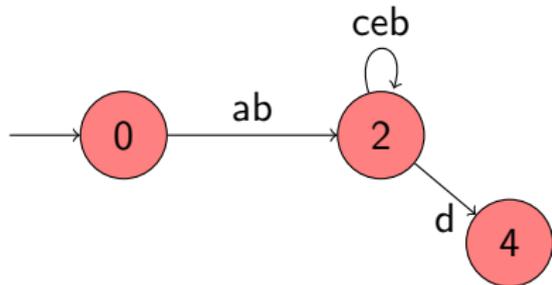
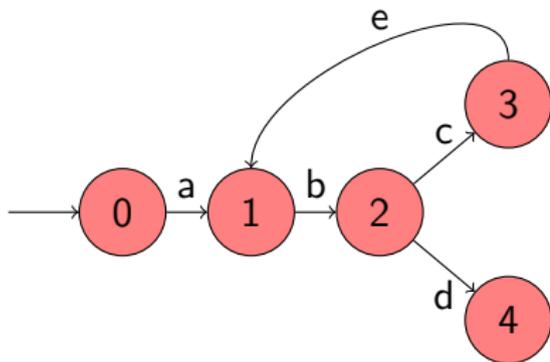
A labelled transition system is a tuple $\langle S, A, \rightarrow, s \rangle$ consisting of

- a set S of states,
- a set A of actions,
- a set of transitions $\rightarrow \subseteq S \times A \times S$, and
- a start state $s \in S$.

Problem

Given a model, expressed as a labelled transition system, construct the corresponding mini model, also expressed as a labelled transition system.

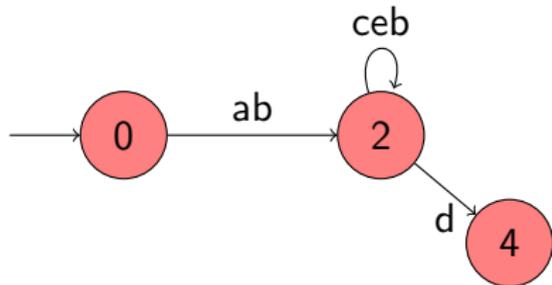
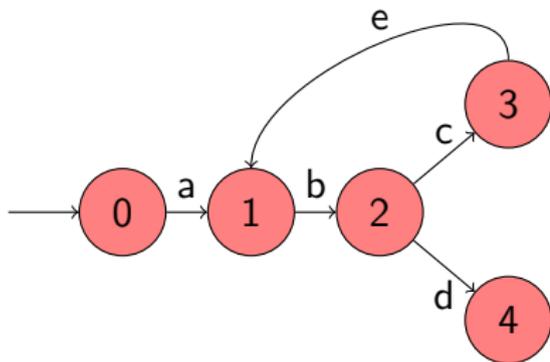
States



Question

Which states do we keep?

States



Question

Which states do we keep?

Answer

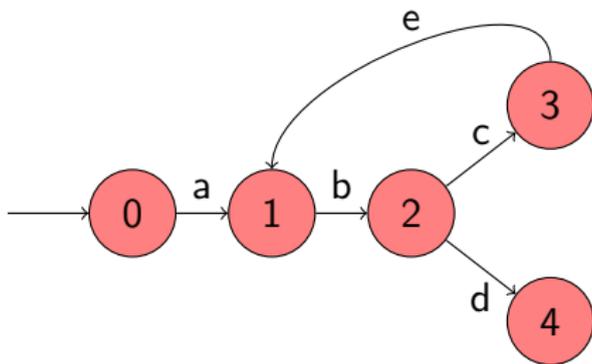
Initial state, final states, and all branching states.

Definition

The set $\text{succ}(s)$ of successors of the state s is defined by

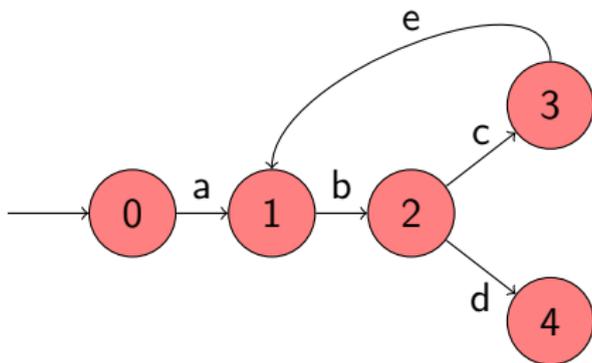
$$\text{succ}(s) = \{ t \in S \mid \exists a \in A : s \xrightarrow{a} t \}.$$

Labelled transition systems: successors



$\text{succ}(0) =$

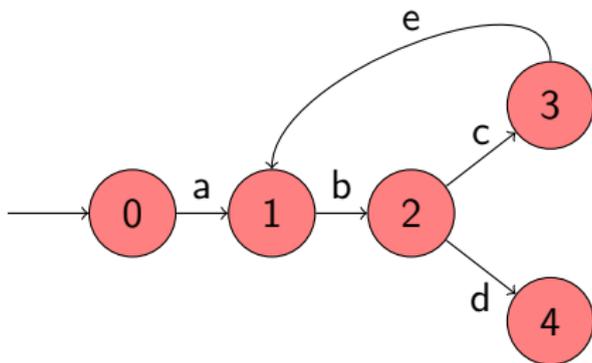
Labelled transition systems: successors



$$\text{succ}(0) = \{1\}$$

$$\text{succ}(1) =$$

Labelled transition systems: successors

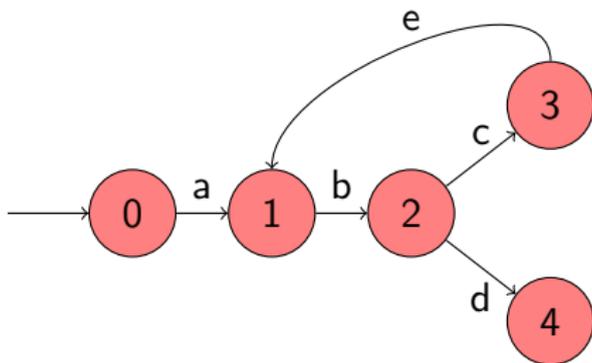


$$\text{succ}(0) = \{1\}$$

$$\text{succ}(1) = \{2\}$$

$$\text{succ}(2) =$$

Labelled transition systems: successors



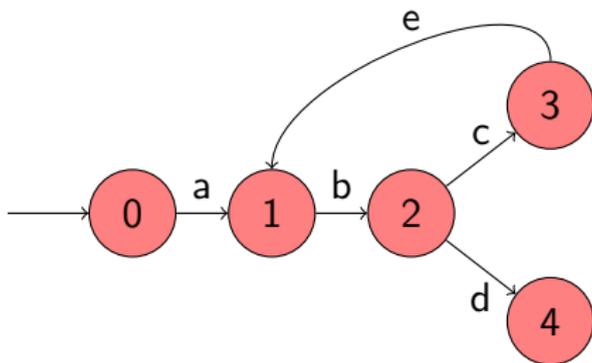
$$\text{succ}(0) = \{1\}$$

$$\text{succ}(1) = \{2\}$$

$$\text{succ}(2) = \{3, 4\}$$

$$\text{succ}(3) =$$

Labelled transition systems: successors



$$\text{succ}(0) = \{1\}$$

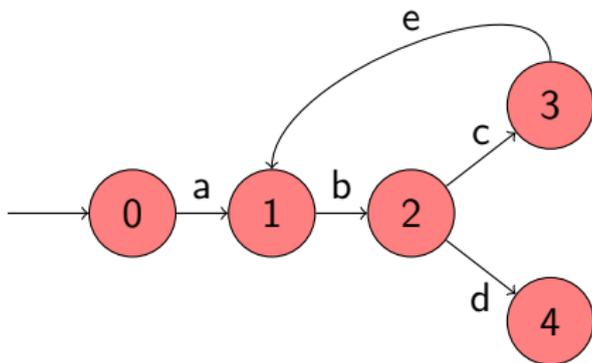
$$\text{succ}(1) = \{2\}$$

$$\text{succ}(2) = \{3, 4\}$$

$$\text{succ}(3) = \{1\}$$

$$\text{succ}(4) =$$

Labelled transition systems: successors



$$\text{succ}(0) = \{1\}$$

$$\text{succ}(1) = \{2\}$$

$$\text{succ}(2) = \{3, 4\}$$

$$\text{succ}(3) = \{1\}$$

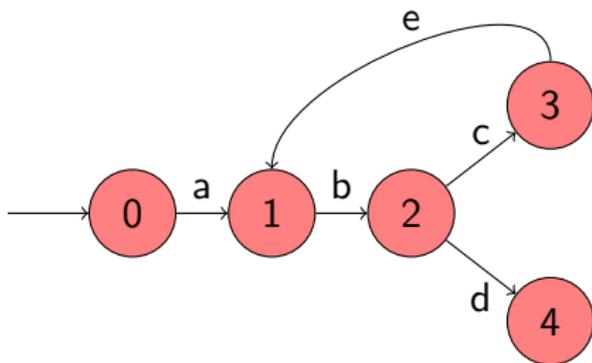
$$\text{succ}(4) = \emptyset$$

Definition

The set $pred(s)$ of predecessors of the state s is defined by

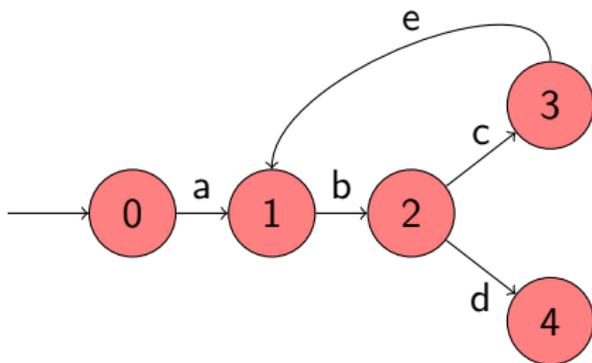
$$pred(s) = \{ t \in S \mid \exists a \in A : t \xrightarrow{a} s \}.$$

Labelled transition systems: predecessors



$pred(0) =$

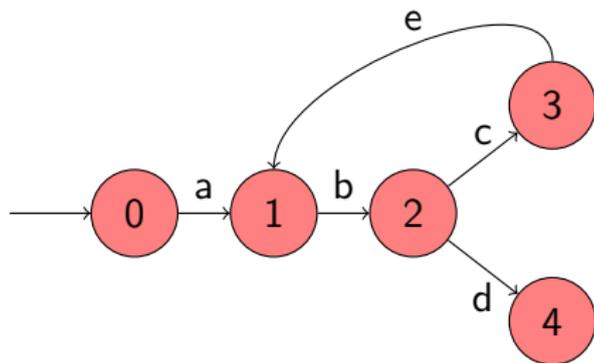
Labelled transition systems: predecessors



$$\text{pred}(0) = \emptyset$$

$$\text{pred}(1) =$$

Labelled transition systems: predecessors

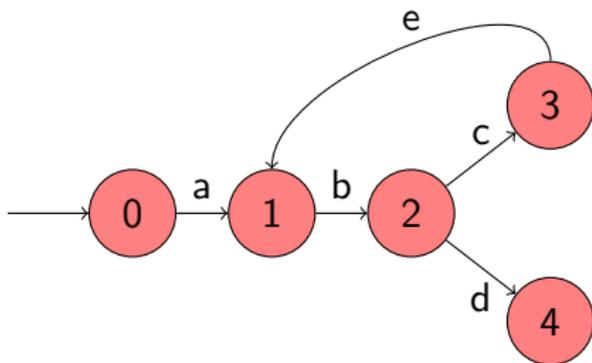


$$\text{pred}(0) = \emptyset$$

$$\text{pred}(1) = \{0, 3\}$$

$$\text{pred}(2) =$$

Labelled transition systems: predecessors



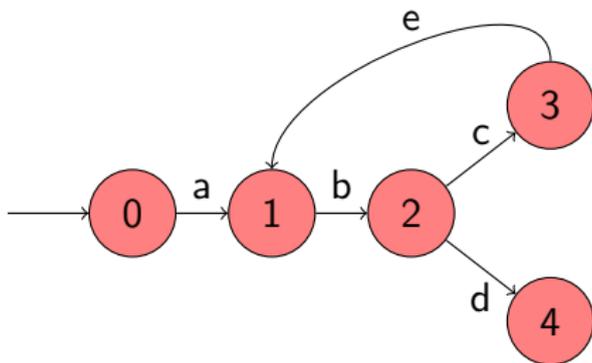
$$\text{pred}(0) = \emptyset$$

$$\text{pred}(1) = \{0, 3\}$$

$$\text{pred}(2) = \{1\}$$

$$\text{pred}(3) =$$

Labelled transition systems: predecessors



$$\text{pred}(0) = \emptyset$$

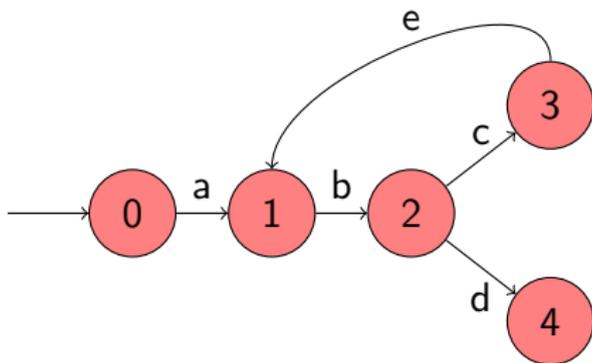
$$\text{pred}(1) = \{0, 3\}$$

$$\text{pred}(2) = \{1\}$$

$$\text{pred}(3) = \{2\}$$

$$\text{pred}(4) =$$

Labelled transition systems: predecessors



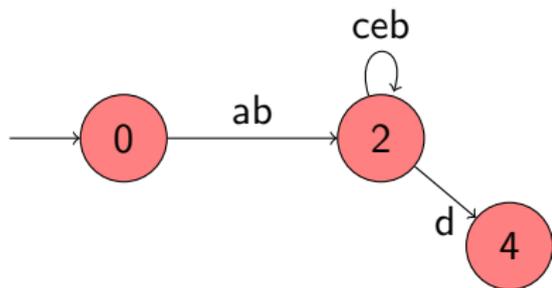
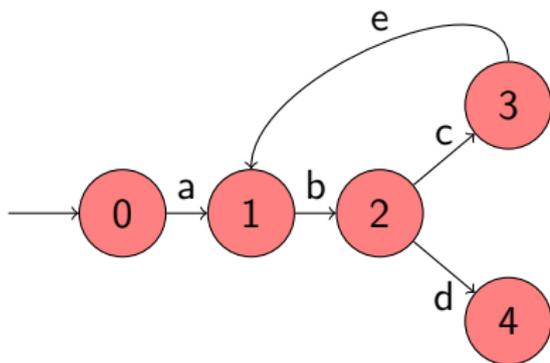
$$\text{pred}(0) = \emptyset$$

$$\text{pred}(1) = \{0, 3\}$$

$$\text{pred}(2) = \{1\}$$

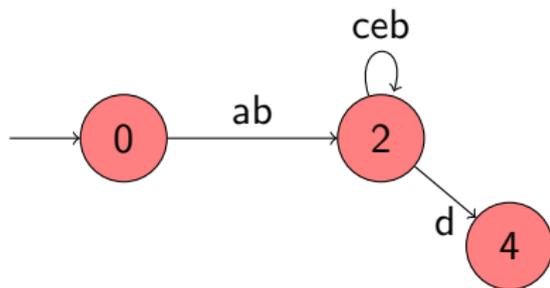
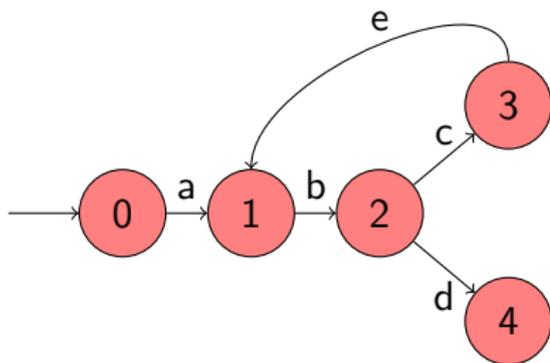
$$\text{pred}(3) = \{2\}$$

$$\text{pred}(4) = \{2\}$$



Question

Given a labelled transition system $\langle S, A, \rightarrow, s_0 \rangle$, which states do we keep?

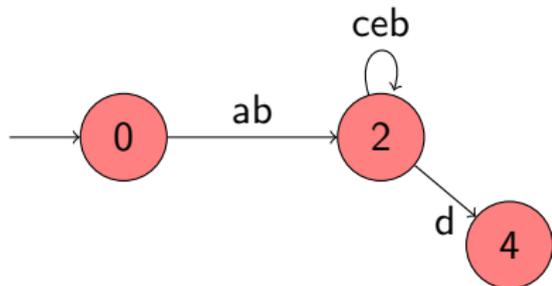
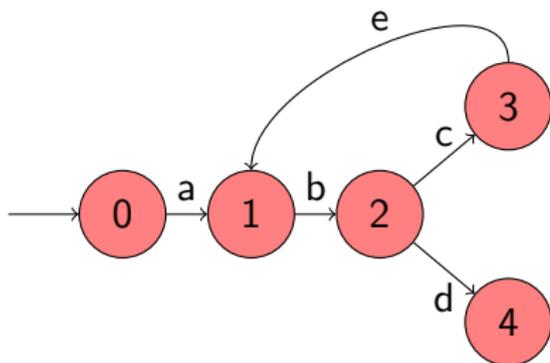


Question

Given a labelled transition system $\langle S, A, \rightarrow, s_0 \rangle$, which states do we keep?

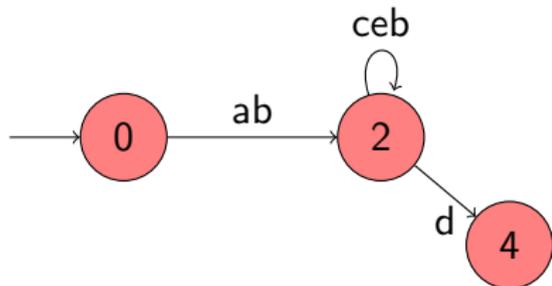
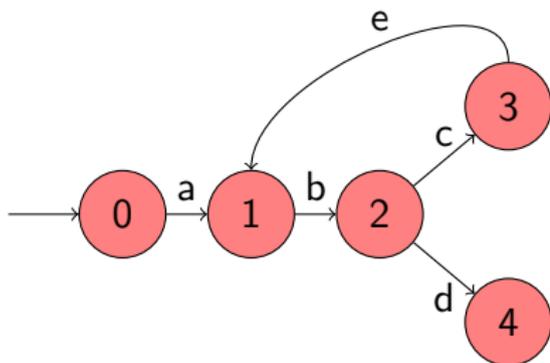
Answer

$$S^+ = \{s_0\} \cup \{s \in S \mid |\text{succ}(s)| \neq 1\}.$$



Question

Given a labelled transition system $\langle S, A, \rightarrow, s_0 \rangle$, what are the actions of the labelled transition system of the corresponding mini model?



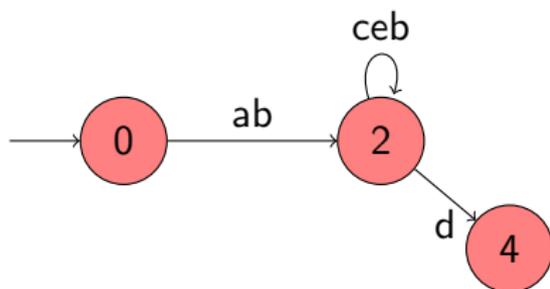
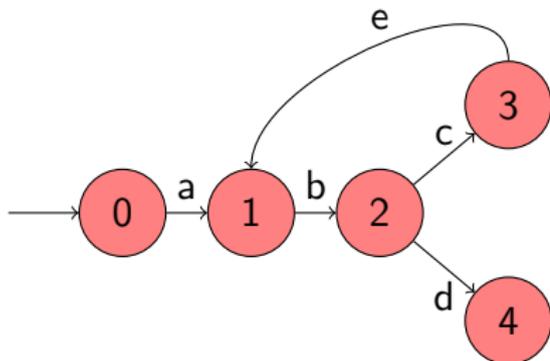
Question

Given a labelled transition system $\langle S, A, \rightarrow, s_0 \rangle$, what are the actions of the labelled transition system of the corresponding mini model?

Answer

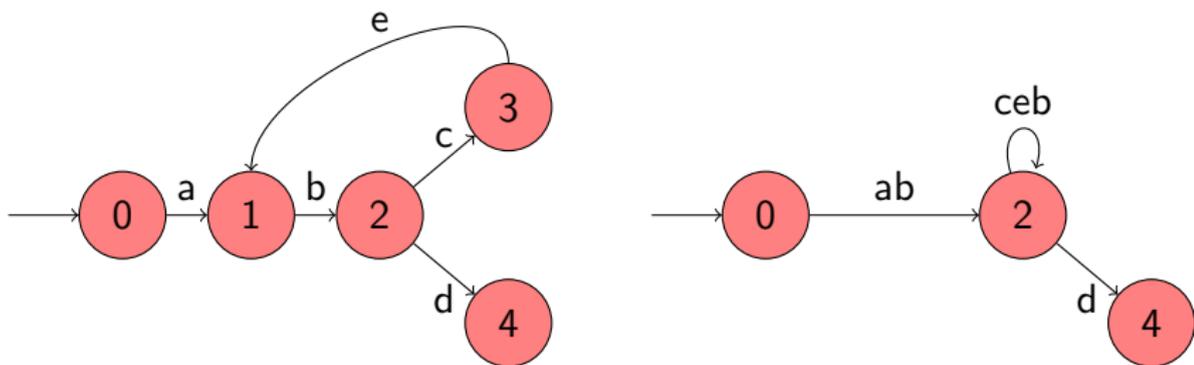
A^+ : nonempty and finite sequences of actions.

Transitions



Question

Given a labelled transition system $\langle S, A, \rightarrow, s_0 \rangle$, what are the transitions of the labelled transition system of the corresponding mini model?



Answer

$s_1 \xrightarrow{a_1 \dots a_n}^+ s_{n+1}$ if

$$s_1 \in S^+ \wedge s_{n+1} \in S^+ \wedge$$

$$\exists s_2, \dots, s_n \in S \setminus S^+ : \forall 1 \leq i < n : s_i \xrightarrow{a_i} s_{i+1} \wedge$$

$$\forall 1 \leq i, j \leq n : s_i = s_j \Rightarrow i = j$$

Problem

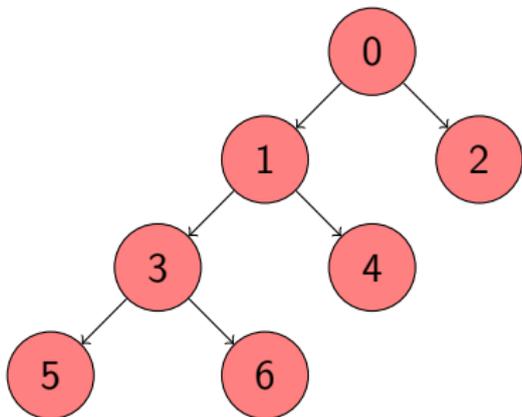
Given a model, expressed as a labelled transition system $\langle S, A, \rightarrow, s_0 \rangle$, construct the corresponding mini model, also expressed as a labelled transition system.

Problem

Given a model, expressed as a labelled transition system $\langle S, A, \rightarrow, s_0 \rangle$, construct the corresponding mini model, also expressed as a labelled transition system.

Solution

$\langle S^+, A^+, \rightarrow^+, s_0 \rangle$.



Task 1

Develop a Java app that prints some output. When checking the Java app by JPF with depth-first search (DFS) the output should be different from the output for breadth-first search (BFS).

```
Random random = new Random();
System.out.println("0");
if (random.nextBoolean()) {
    System.out.println("2");
} else {
    System.out.println("1");
    if (random.nextBoolean()) {
        System.out.println("4");
    } else {
        System.out.println("3");
        if (random.nextBoolean()) {
            System.out.println("6");
        } else {
            System.out.println("5");
        }
    }
}
}
```

Task 2

Verify your program using JPF with BFS and DFS. To do that, you need to create an application properties file (.jpf file) for your Java app developed in Task 1. Configure the search property to be `gov.nasa.jpf.search.heuristic.BFSHeuristic` or `gov.nasa.jpf.search.heuristic.DFSHeuristic`.

```
target=Traversal  
classpath=.  
cg.enumerate_random=true  
search=gov.nasa.jpf.search.heuristic.BFSHeuristic
```

```
===== syst
Traversal.main()
===== sear
0
1
3
5
6
4
2
=====
```

That is not breadth first search!

Question

Have we set the search property correctly? How can we check that?

That is not breadth first search!

Question

Have we set the search property correctly? How can we check that?

Answer

Use the following command line arguments

- **-log**: lists the order in which properties files got loaded
- **-show**: prints all configuration entries after the initialization is complete

```
loading property file: /eecs/home/franck/.jpf/site.properties  
loading property file: /eecs/fac/pkg/jpf/jpf-core/jpf.properties  
loading property file: Example.jpf  
collected native_classpath=/eecs/fac/pkg/jpf/jpf-core/build/jpf.  
collected native_libraries=null
```

...

```
search = gov.nasa.jpf.search.heuristic.BFSHeuristic  
search.class = gov.nasa.jpf.search.DFSearch
```

...

```
target=Traversal  
classpath=.  
cg.enumerate_random=true  
search.class=gov.nasa.jpf.search.heuristic.BFSHeuristic
```

```
===== syst
Traversal.main()
===== sear
0
1
2
3
4
5
6
=====
```

```
target=Traversal  
classpath=.  
cg.enumerate_random=true  
search.class=gov.nasa.jpf.search.heuristic.DFSHeuristic
```

```
===== syst
Traversal.main()
===== sear
0
1
2
3
4
5
6
=====
```

That is not depth first search!

That is not depth first search!

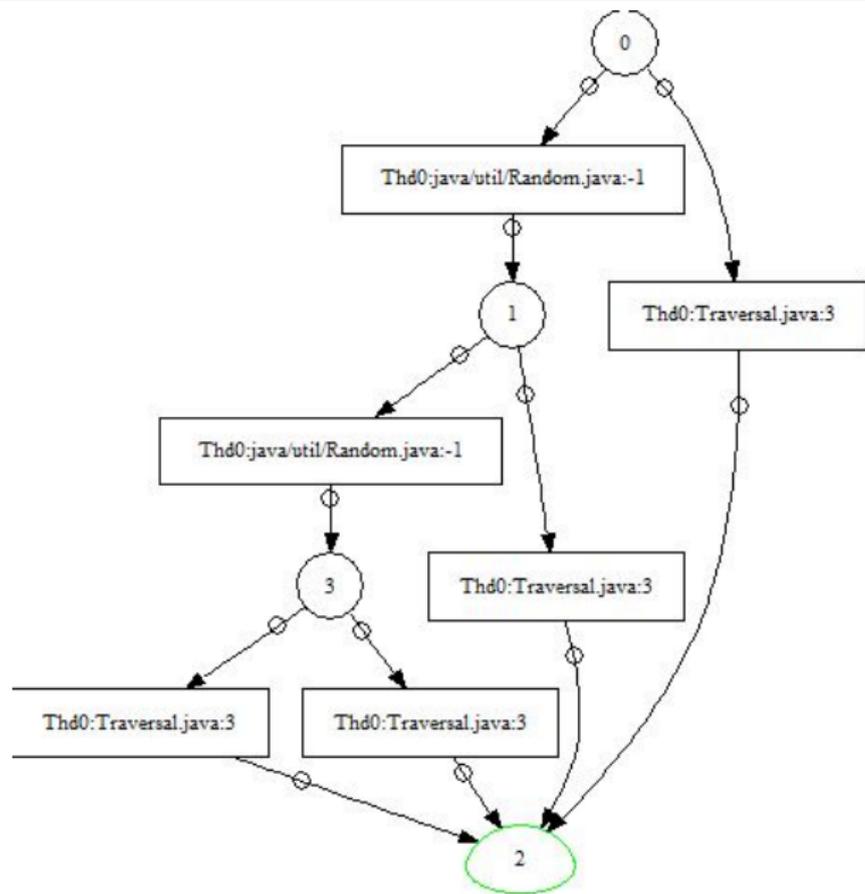
Let's try instead `gov.nasa.jpfn.search.DFSearch`.

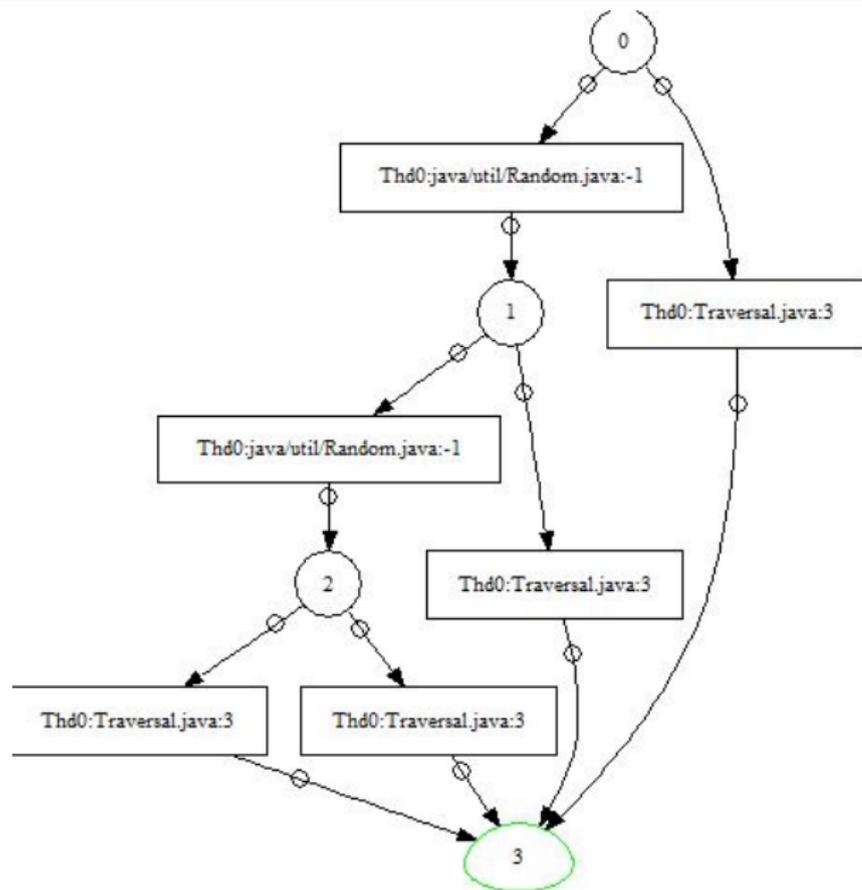
```
target=Traversal  
classpath=.  
cg.enumerate_random=true  
search.class=gov.nasa.jpf.search.DFSearch
```

```
===== syst
Traversal.main()
===== sear
0
1
3
5
6
4
2
=====
```

Task 3

Generate the state space diagram for BFS and DFS. To do this you need to set the listener to `StateSpaceDot`.





Task 4

Verify your program using RS. RS can explore several random executions and in JPF you have the freedom to set the maximum number of executions you would like RS to explore. Firstly, set your search strategy to `gov.nasa.jpf.search.RandomSearch`. Secondly, set the `search.RandomSearch.path_limit` property to be any integer larger than 0. Compare the resulting state space diagrams.

JavaPathfinder core system v8.0 (rev 2+) - (C) 2005-2014 U

===== syst

Traversal.main()

===== sear

0

1

3

5

No space diagram has been produced.

No space diagram has been produced.

Bugs are everywhere, even in JPF!