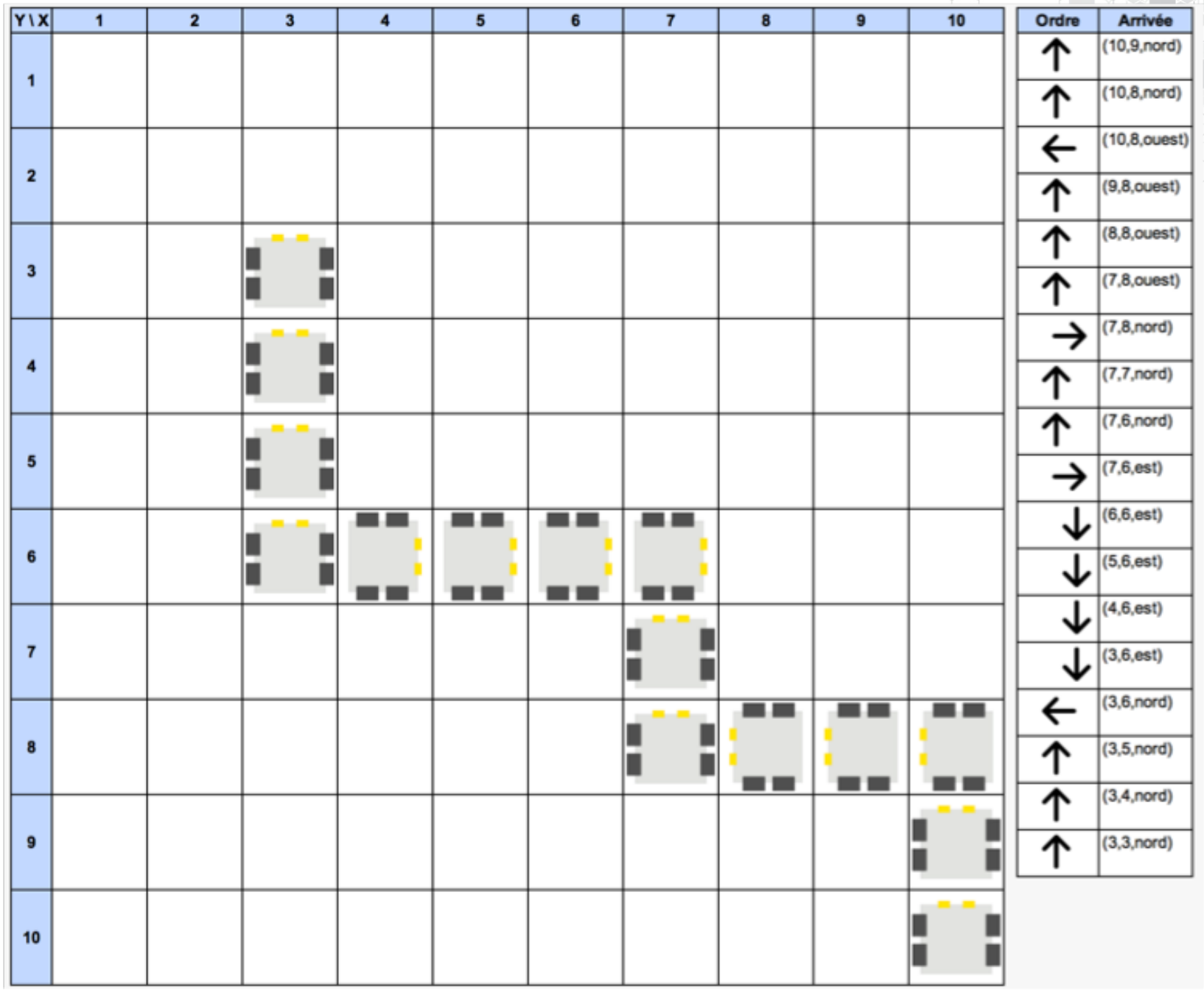
Test structurel
Exemple du robot

Fabrice AMBERT, Fabrice BOUQUET, Fabien PEUREUX,
Jean-Marie GAUTHIER, Alexandre VERNOTTE
prenom.nom@femto-st.fr

ENVOL 2014
La Londe-les-Maures
18-21 novembre 2014

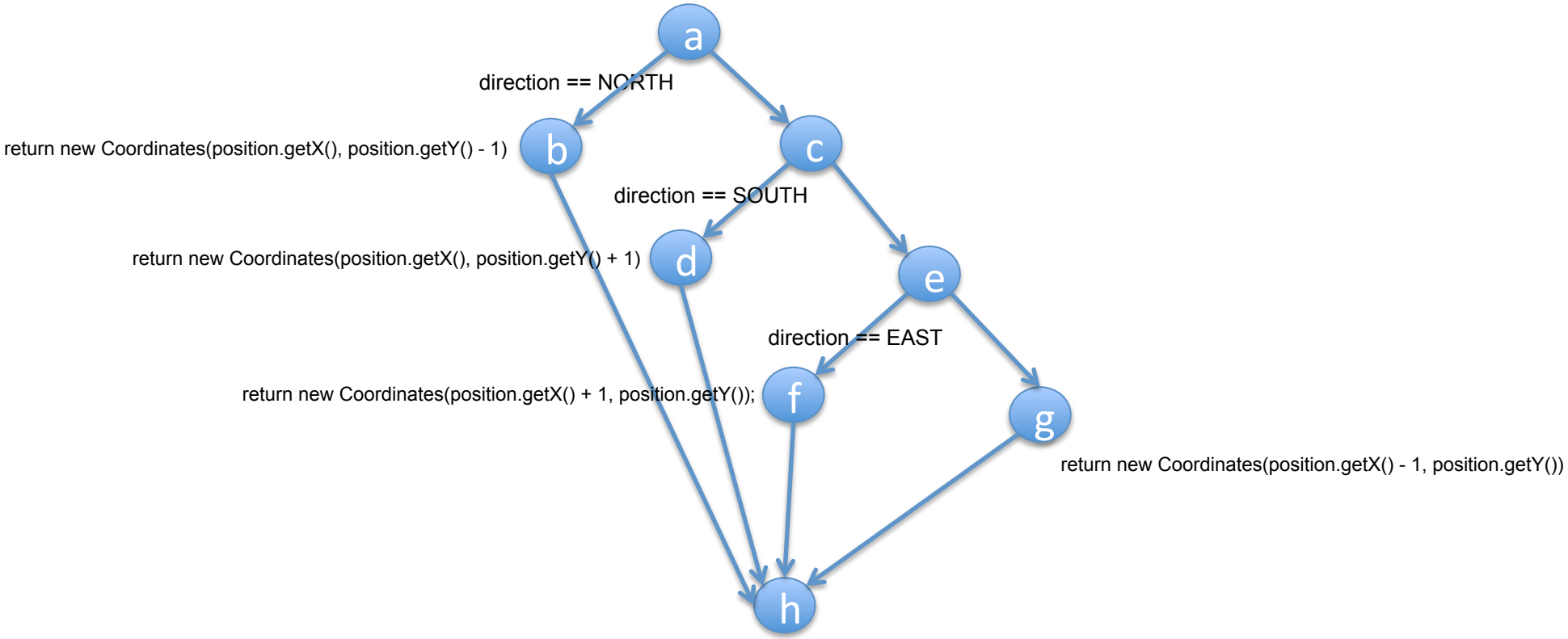

Y \ X
1
2
3
4
5
6
7
8
9
10
Ordre
Arrivée
(10,9,nord)
(10,8,nord)
(10,8,ouest)
(9,8,ouest)
(8,8,ouest)
(7,8,ouest)
(7,8,nord)
(7,7,nord)
(7,6,nord)
(7,6,est)
(6,6,est)
(5,6,est)
(4,6,est)
(3,6,est)
(3,6,nord)
(3,5,nord)
(3,4,nord)
(3,3,nord)

nextForwardPosition

```
public static Coordinates nextForwardPosition(Coordinates position, Direction direction) {
        if (direction == NORTH)
            return new Coordinates(position.getX(), position.getY() - 1);
        if (direction == SOUTH)
            return new Coordinates(position.getX(), position.getY() + 1);
        if (direction == EAST)
            return new Coordinates(position.getX() + 1, position.getY());
        return new Coordinates(position.getX() - 1, position.getY());
 }
```

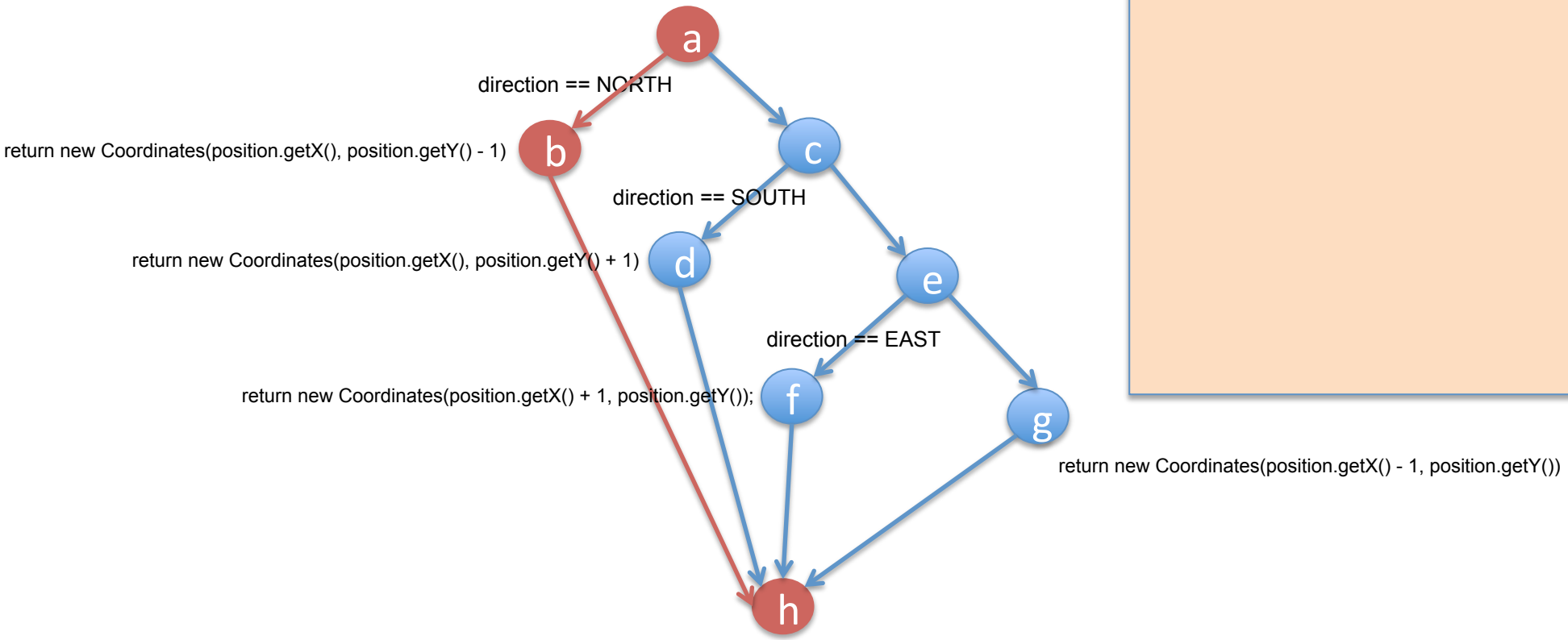
nextForwardPosition

```
public static Coordinates nextForwardPosition(Coordinates position, Direction direction) {
        if (direction == NORTH)
            return new Coordinates(position.getX(), position.getY() - 1);
        if (direction == SOUTH)
            return new Coordinates(position.getX(), position.getY() + 1);
        if (direction == EAST)
            return new Coordinates(position.getX() + 1, position.getY());
        return new Coordinates(position.getX() - 1, position.getY());
    }
```

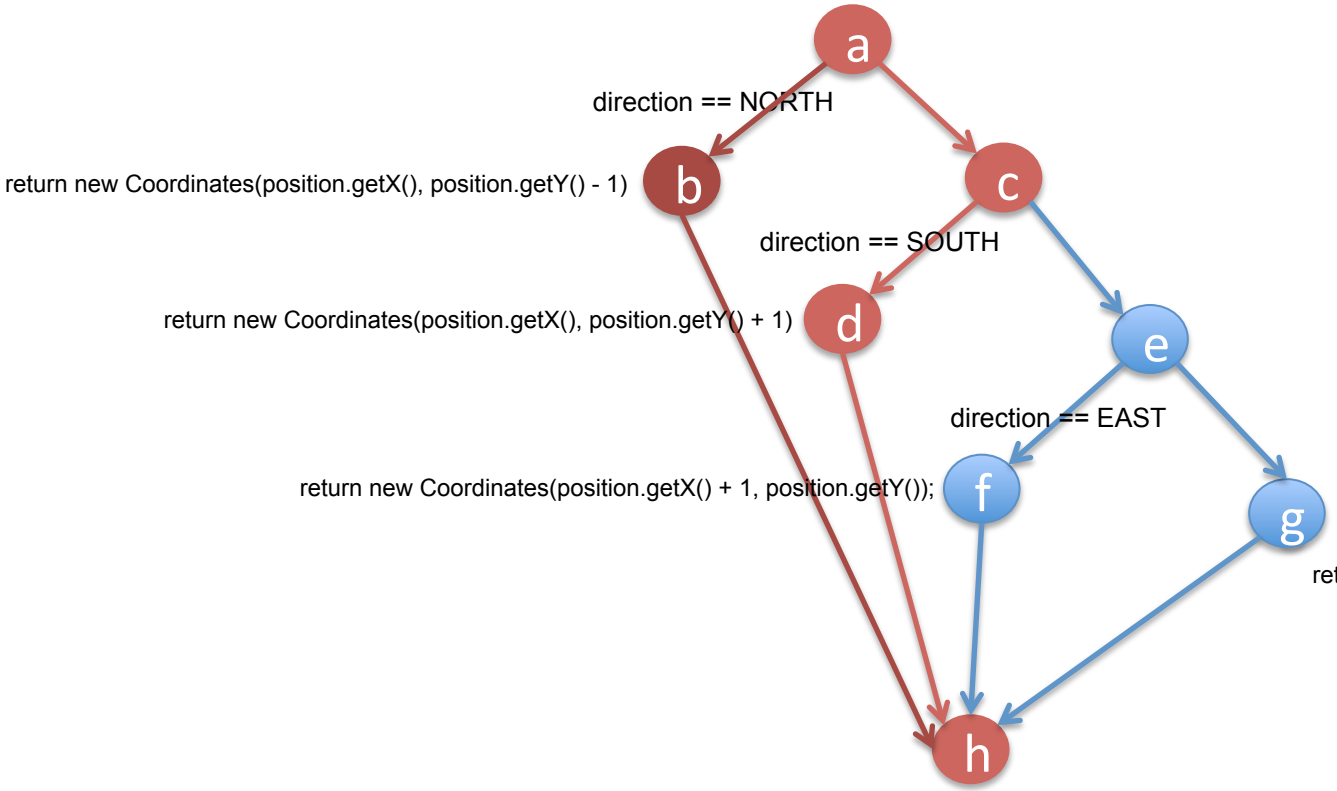


```
public static Coordinates nextForwardPosition(Coordinates position, Direction direction) {
        if (direction == NORTH)
            return new Coordinates(position.getX(), position.getY() - 1);
        if (direction == SOUTH)
            return new Coordinates(position.getX(), position.getY() + 1);
        if (direction == EAST)
            return new Coordinates(position.getX() + 1, position.getY());
        return new Coordinates(position.getX() - 1, position.getY());
    }
```

Cas 1 : <(1,1), NORTH>
oracle : (1,0)



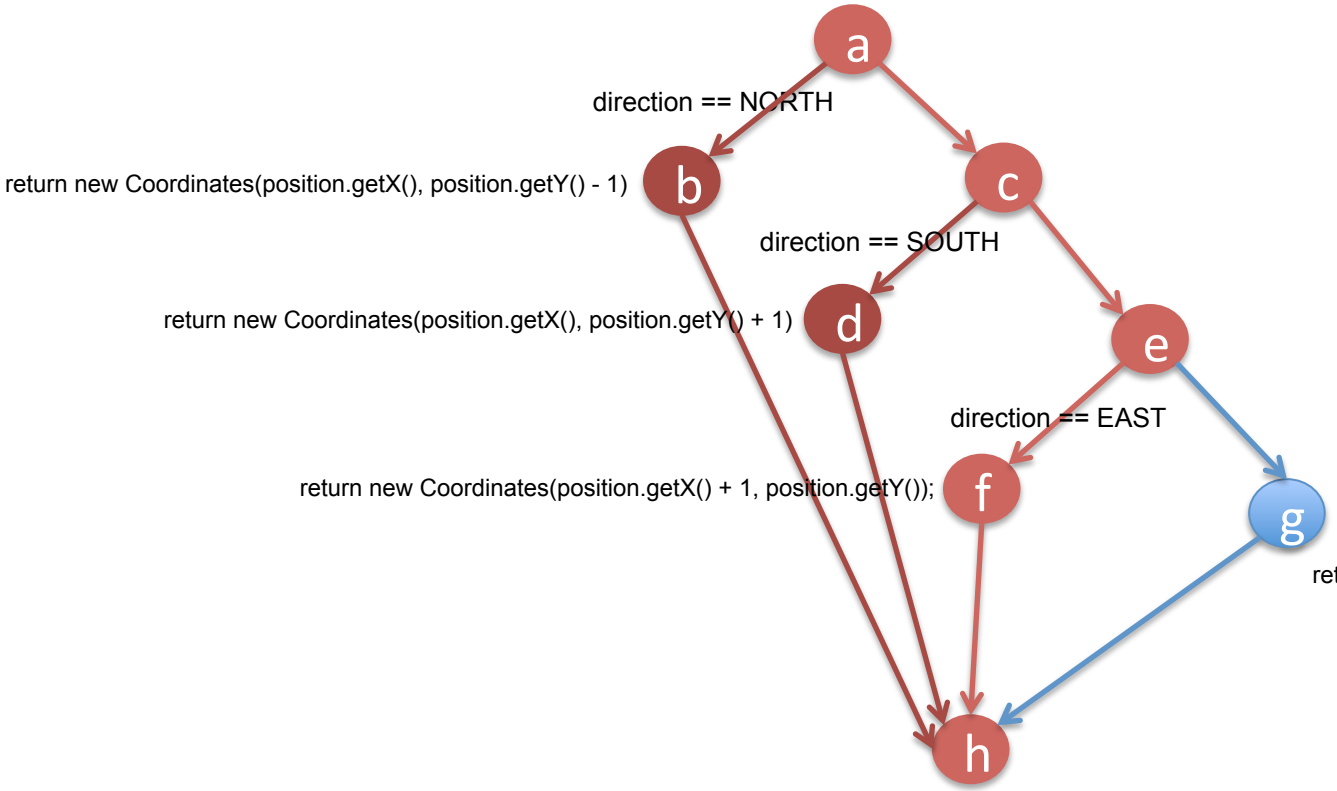
```
public static Coordinates nextForwardPosition(Coordinates position, Direction direction) {
        if (direction == NORTH)
            return new Coordinates(position.getX(), position.getY() - 1);
        if (direction == SOUTH)
            return new Coordinates(position.getX(), position.getY() + 1);
        if (direction == EAST)
            return new Coordinates(position.getX() + 1, position.getY());
        return new Coordinates(position.getX() - 1, position.getY());
    }
```



Cas 1 : <(1,1), NORTH>
oracle : (1,0)

Cas 2 : <(1,1), SOUTH>
oracle : (1,2)

```
public static Coordinates nextForwardPosition(Coordinates position, Direction direction) {
        if (direction == NORTH)
            return new Coordinates(position.getX(), position.getY() - 1);
        if (direction == SOUTH)
            return new Coordinates(position.getX(), position.getY() + 1);
        if (direction == EAST)
            return new Coordinates(position.getX() + 1, position.getY());
        return new Coordinates(position.getX() - 1, position.getY());
    }
```

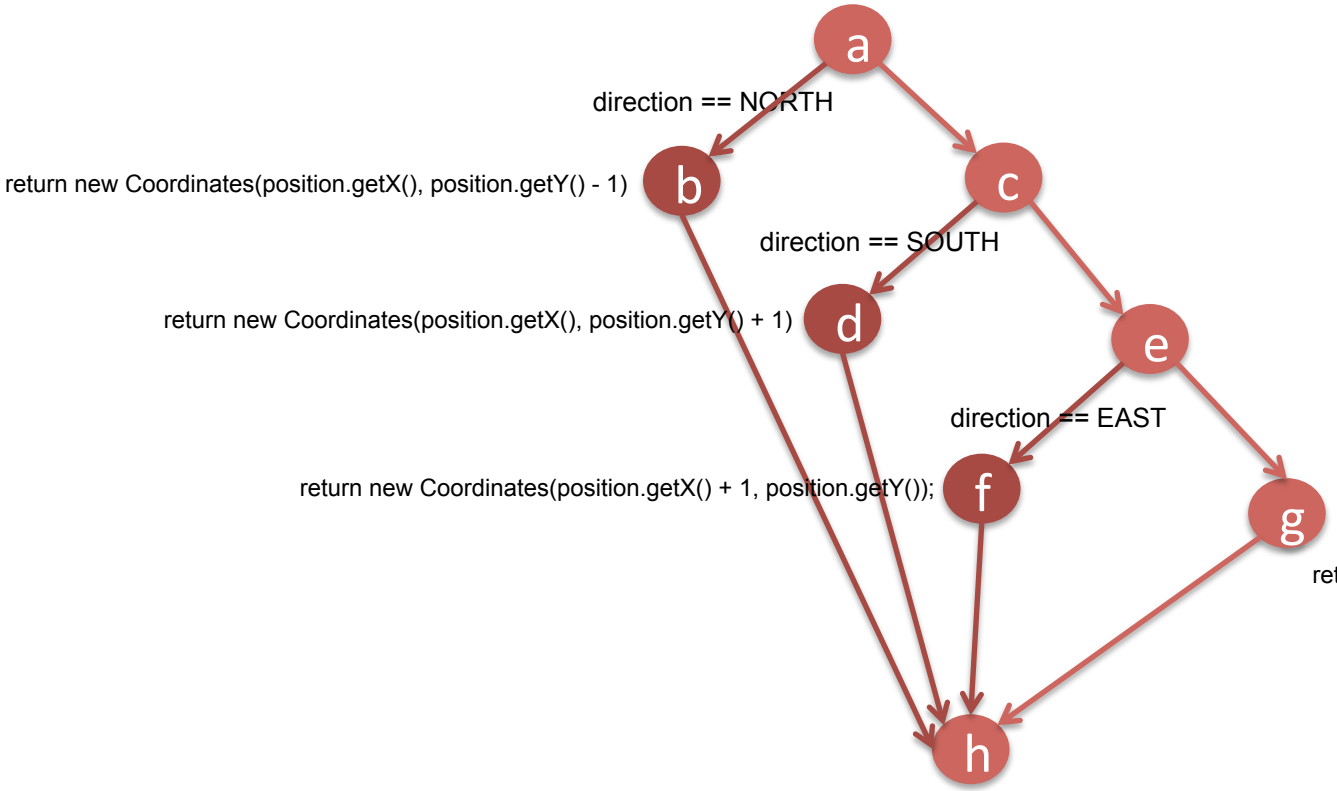


Cas 1 : <(1,1), NORTH>
oracle : (1,0)

Cas 2 : <(1,1), SOUTH>
oracle : (1,2)

Cas 3 : <(1,1), EAST>
oracle : (2,1)

```
public static Coordinates nextForwardPosition(Coordinates position, Direction direction) {
        if (direction == NORTH)
            return new Coordinates(position.getX(), position.getY() - 1);
        if (direction == SOUTH)
            return new Coordinates(position.getX(), position.getY() + 1);
        if (direction == EAST)
            return new Coordinates(position.getX() + 1, position.getY());
        return new Coordinates(position.getX() - 1, position.getY());
    }
```



Cas 1 : <(1,1), NORTH>
oracle : (1,0)

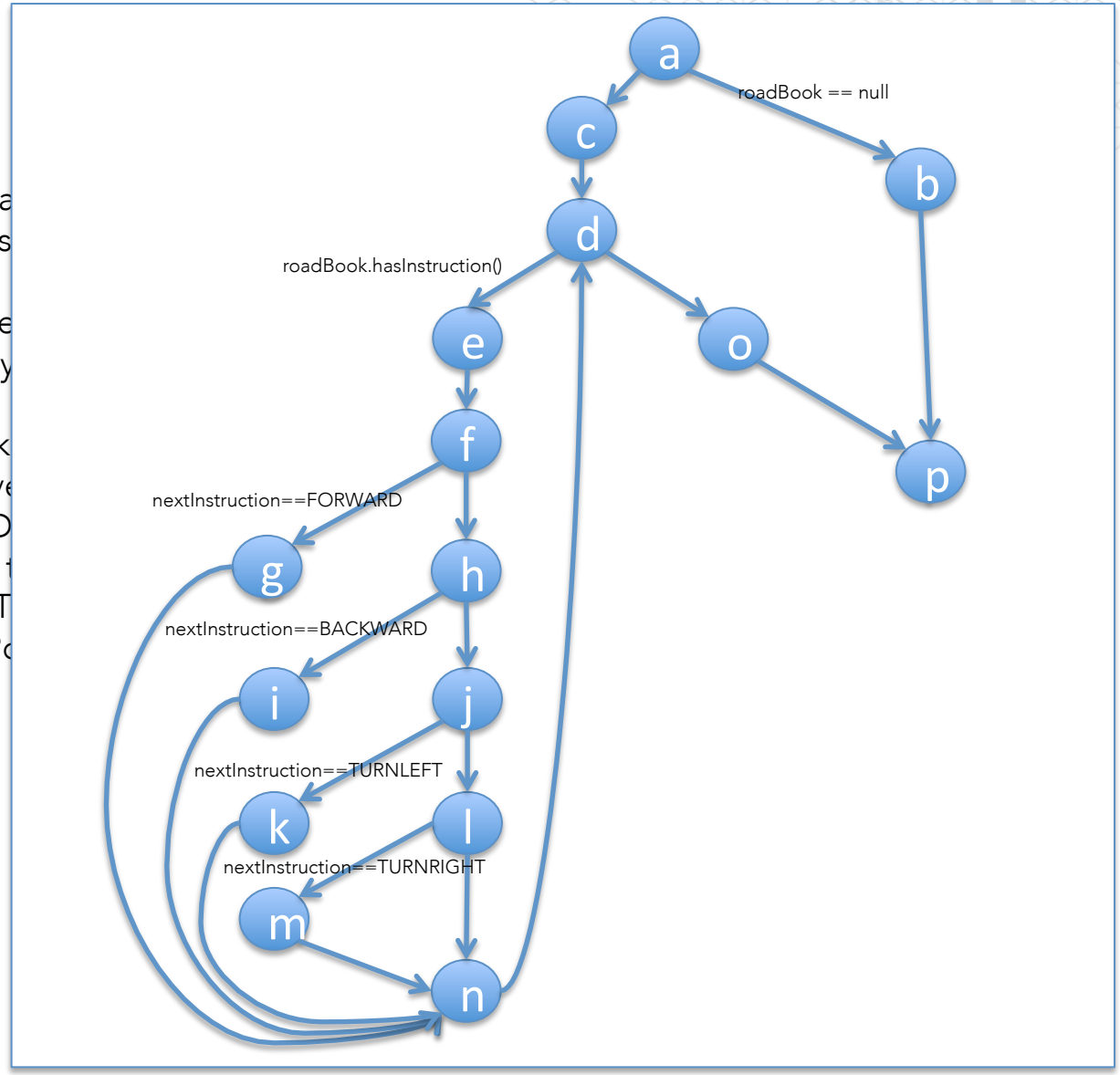
Cas 2 : <(1,1), SOUTH>
oracle : (1,2)

Cas 3 : <(1,1), EAST>
oracle : (2,1)

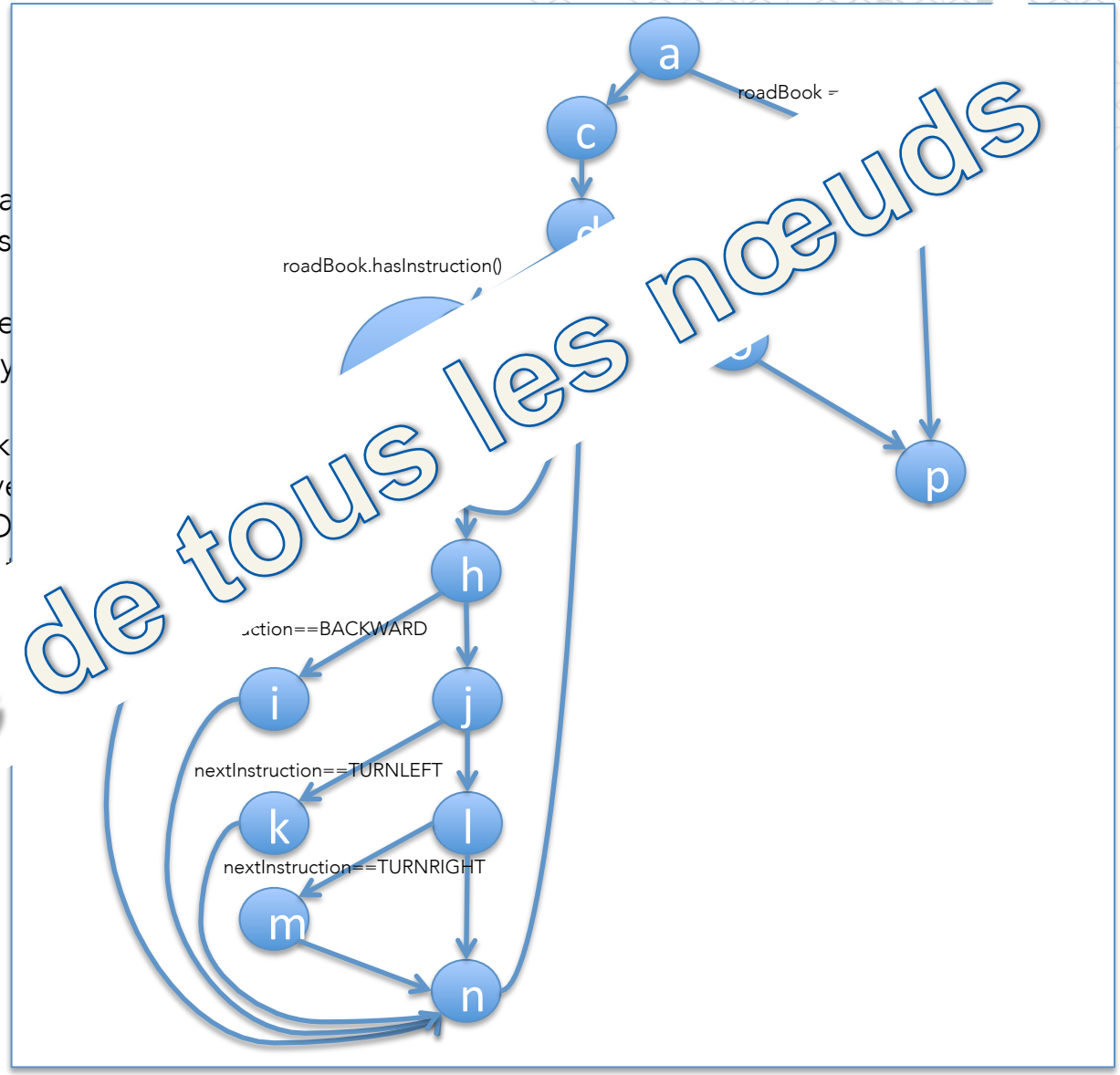
Cas 4 : <(1,1), WEST>
oracle : (0,1)

```
public List<CheckPoint> letsGo() throws UnlandedRobotException, UndefinedRoadbookException,
        InsufficientChargeException, LandSensorDefaillance, InaccessibleCoordinate {

    if (roadBook == null) throw new UndefinedRoadbookException();
    List<CheckPoint> mouchard = new ArrayList<CheckPoint>();
    while (roadBook.hasInstruction()) {
        Instruction nextInstruction = roadBook.next();
        if (nextInstruction == FORWARD) moveForward();
        else if (nextInstruction == BACKWARD) moveBackward();
        else if (nextInstruction == TURNLEFT) turnLeft();
        else if (nextInstruction == TURNRIGHT) turnRight();
        CheckPoint checkPoint = new CheckPoint(position, direction, false);
        mouchard.add(checkPoint);
        blackBox.addCheckPoint(checkPoint);
    }
    return mouchard;
}
```

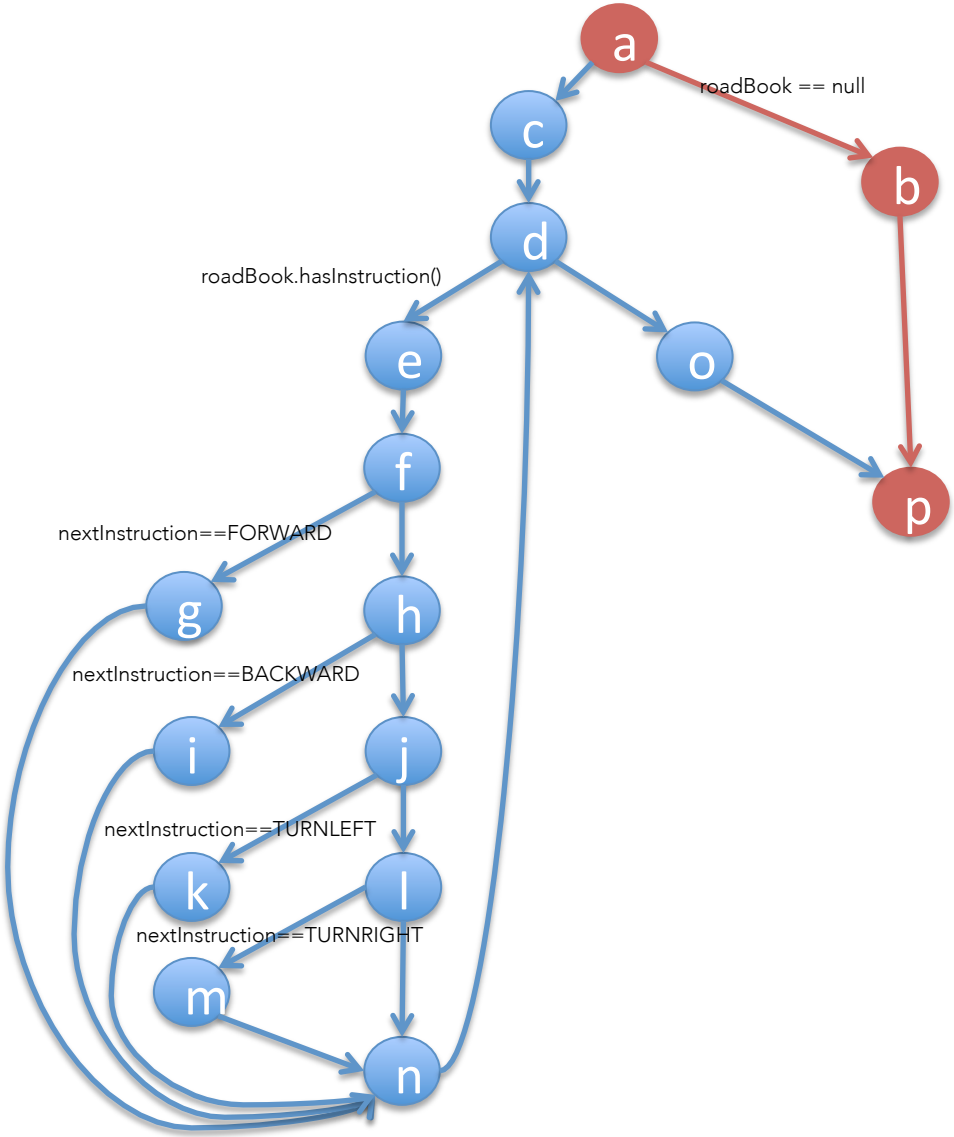
```
public List<CheckPoint> letsGo() throws Unla
        InsufficientChargeException, LandSens

    if (roadBook == null) throw new Undefine
    List<CheckPoint> mouchard = new Array
    while (roadBook.hasInstruction()) {
        Instruction nextInstruction = roadBook
        if (nextInstruction == FORWARD) move
        else if (nextInstruction == BACKWARD
        else if (nextInstruction == TURNLEFT)
        else if (nextInstruction == TURNRIGHT
        CheckPoint checkPoint = new CheckPo
        mouchard.add(checkPoint);
        blackBox.addCheckPoint(checkPoint);
    }
    return mouchard;
}
```



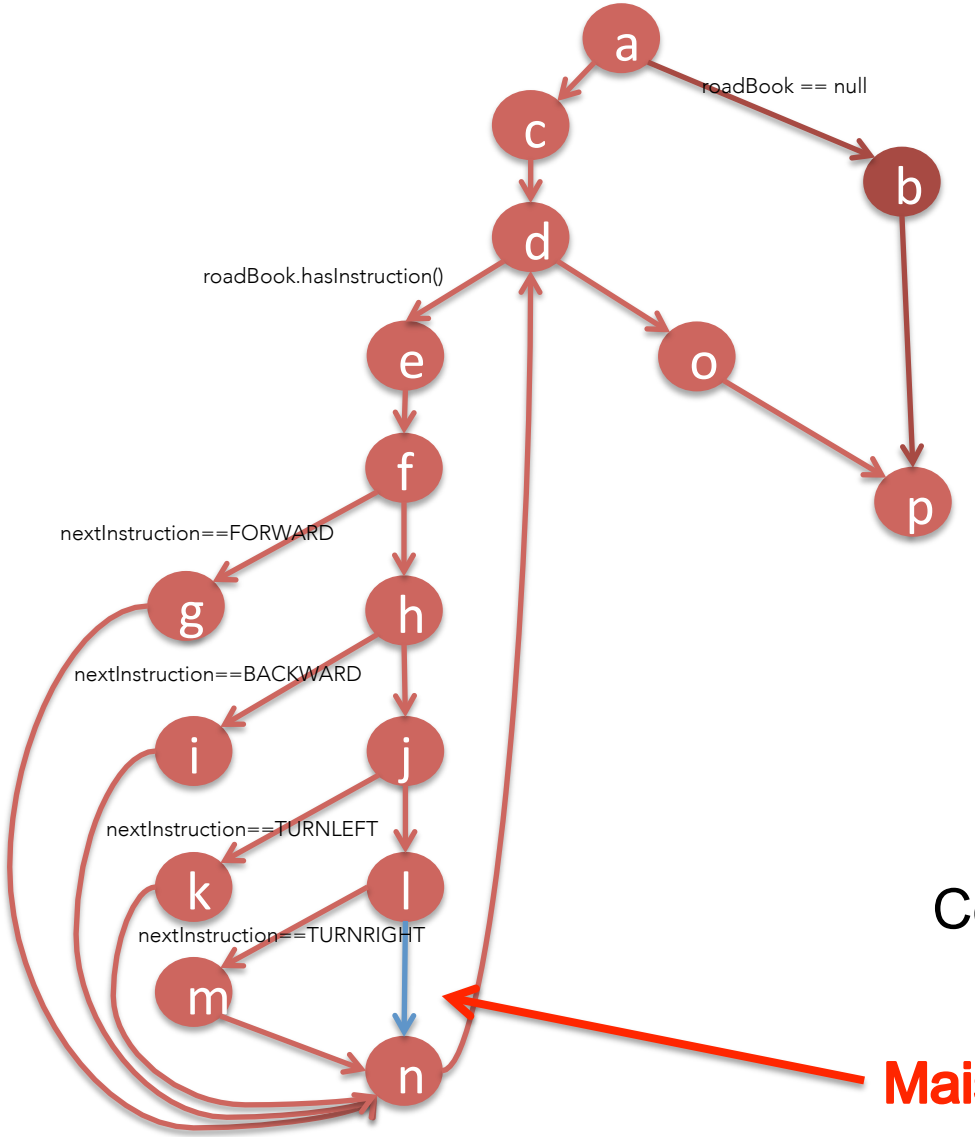
```
public List<CheckPoint> letsGo() throws Unla
        InsufficientChargeException, LandSens

    if (roadBook == null) throw new Undefine
    List<CheckPoint> mouchard = new Array
    while (roadBook.hasInstruction()) {
        Instruction nextInstruction = roadBook
        if (nextInstruction == FORWARD) move
        else if (nextInstruction == BACKWARD
        else if (nextInstruction == TURNLEFT)
        else if (nextInstruction == TURNR
        CheckPoint checkPoint = ne
        mouchard.add(checkPo
        blackBox.addChec
    }
    return mou
}
```



Couverture de tous les nœuds



Cas 1 :
<roadBook = null, position = (0,0), direction = NORTH>
Oracle : UndefinedRoadBookException



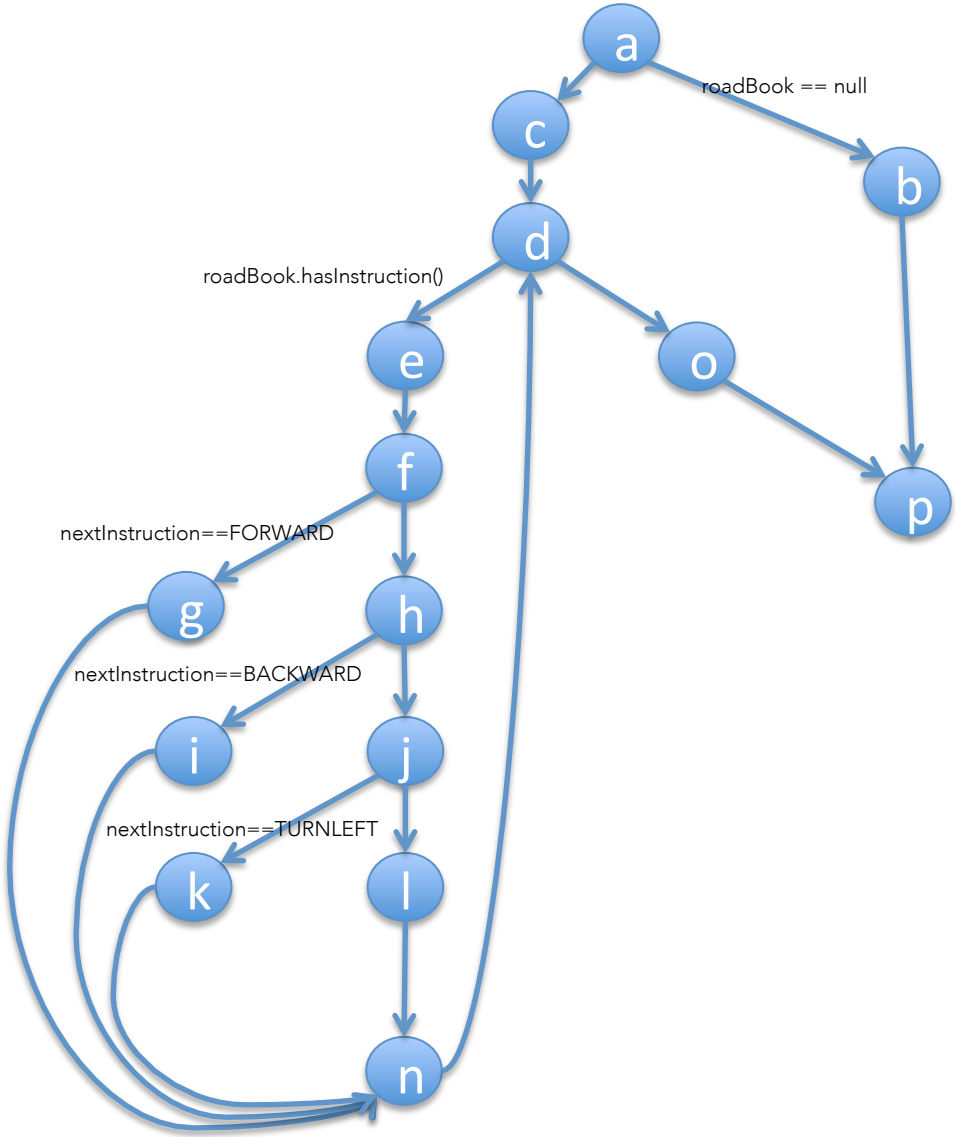
Cas 1 :
<roadBook = null, position = (0,0), direction = NORTH>
Oracle : UndefinedRoadBookException

Cas 2 :
<roadBook = [FORWARD, TURNRIGHT, BACKWARD, TURNLEFT],
position = (0,0), direction = NORTH>
Oracle : [<(0,-1),NORTH,false>, <(0,-1),EAST,false>,
<(-1,-1),EAST,false>, <(-1,-1),NORTH,false>]

Couverture tous nœuds obtenue

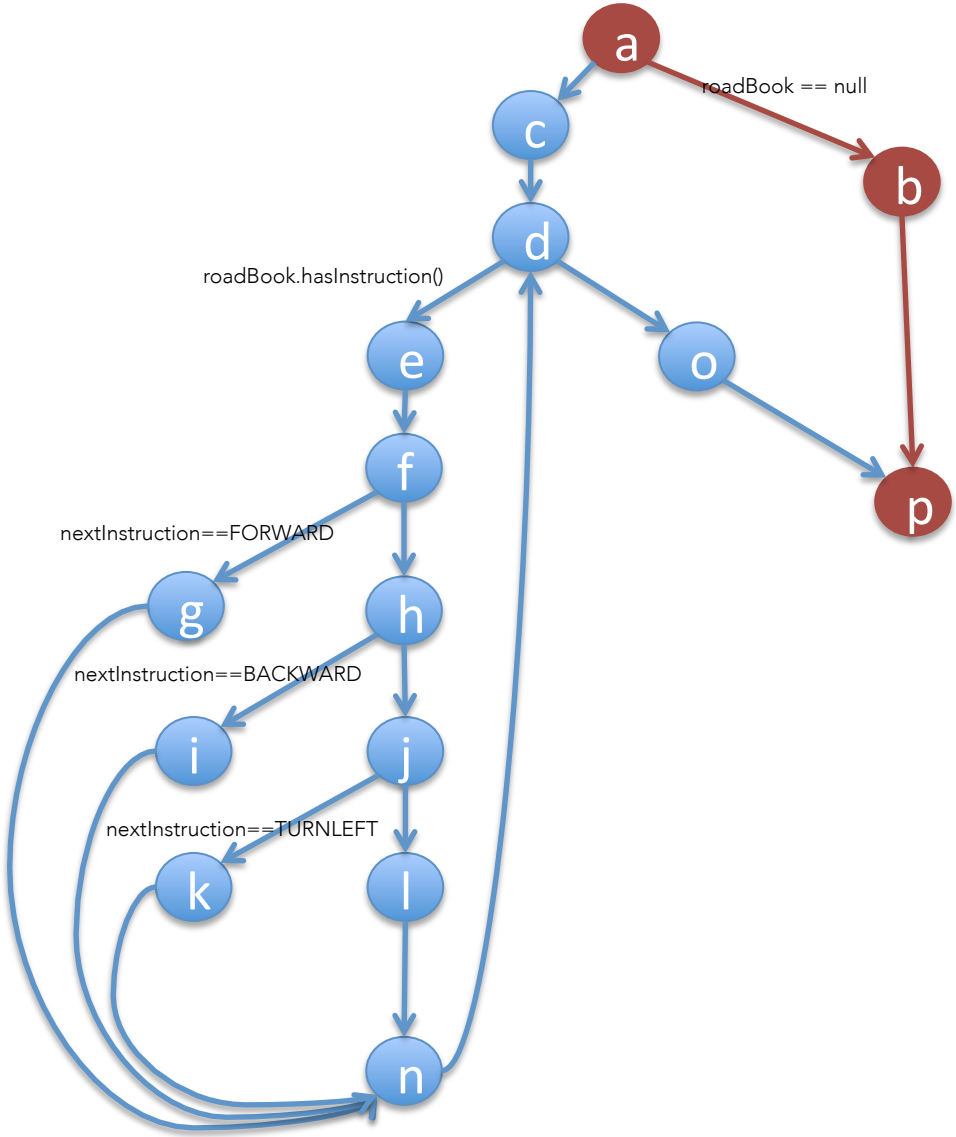
Mais pas tous arcs

Couverture de tous chemins indépendants



Mc Cabe : 6

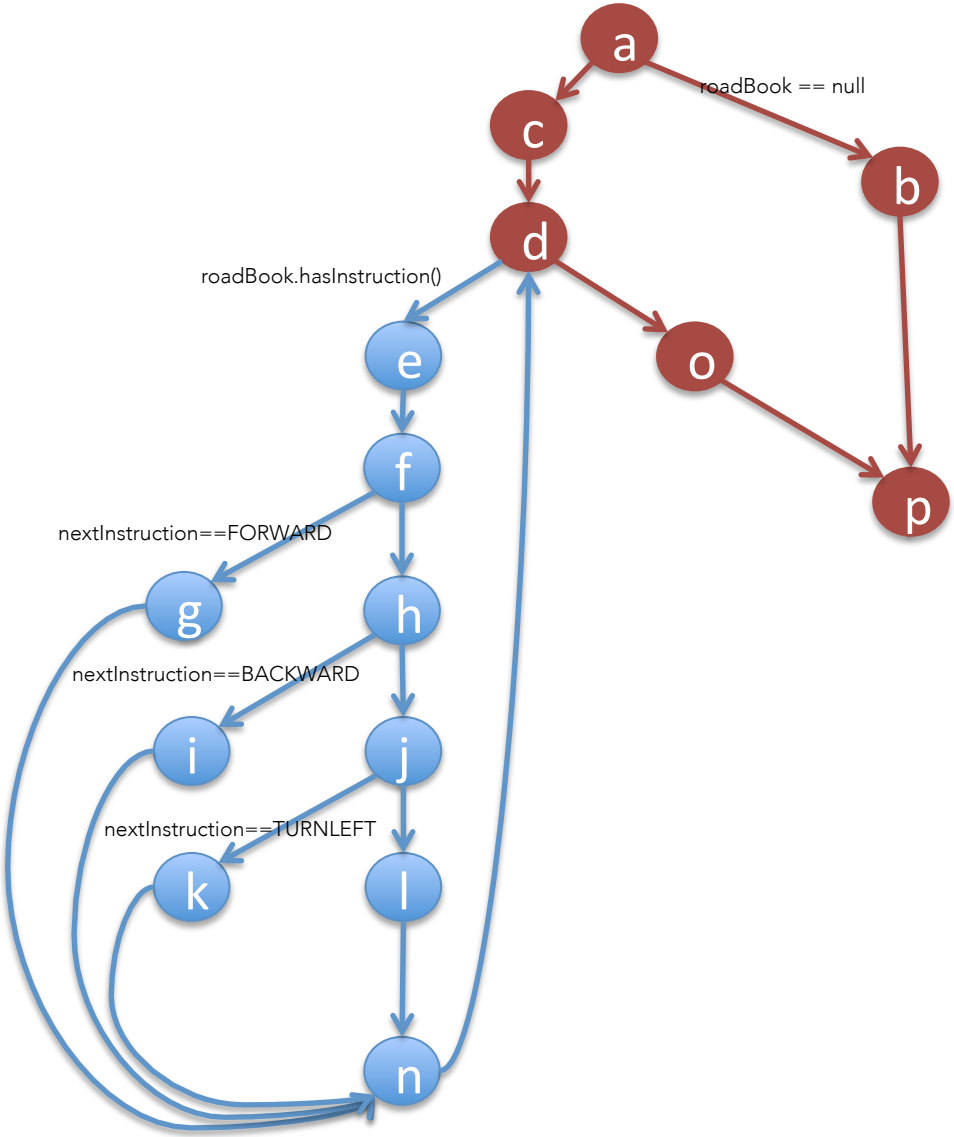
Couverture de tous chemins indépendants



Mc Cabe : 6

Cas 1 :
<roadBook = null, position = (0,0), direction = NORTH>
Oracle : UndefinedRoadBookException

Couverture de tous chemins indépendants

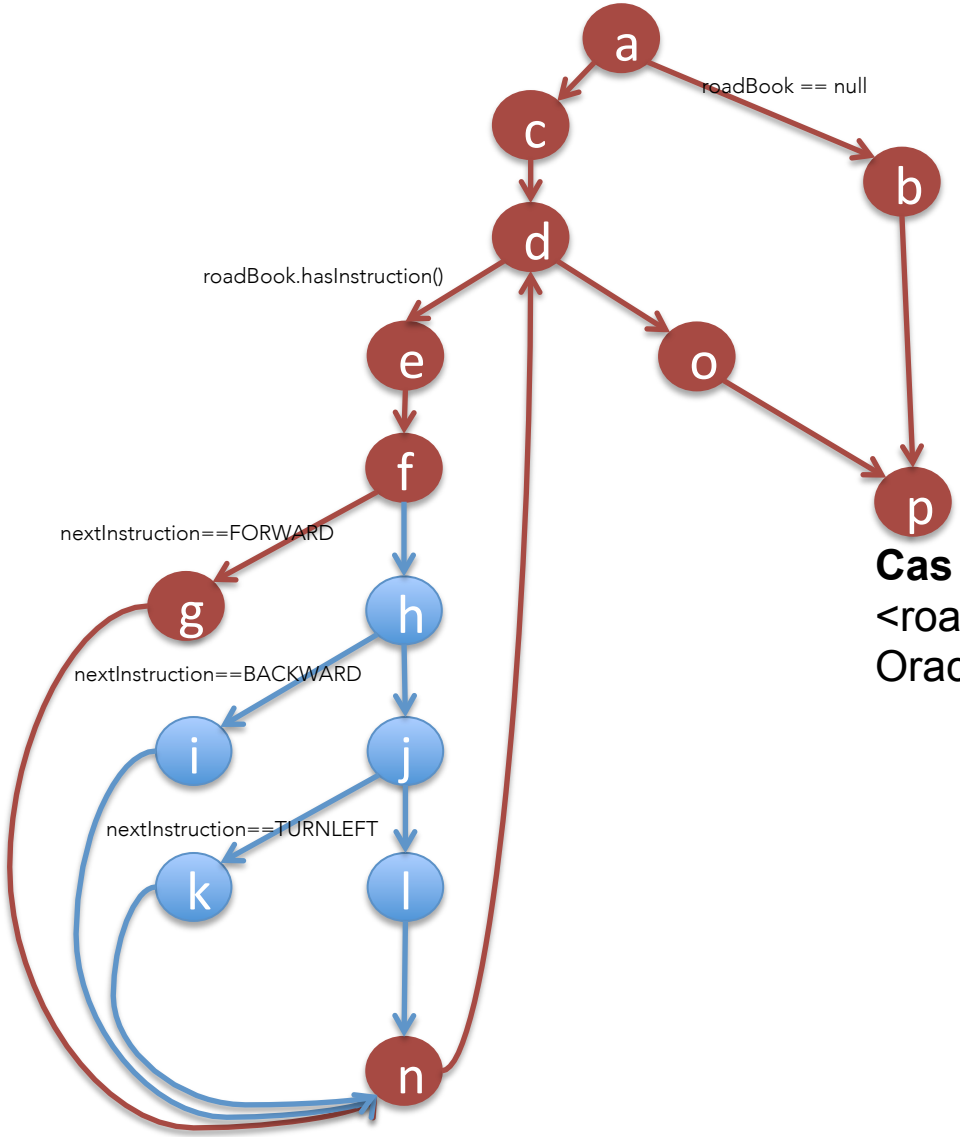


Mc Cabe : 6

Cas 1 :
<roadBook = null, position = (0,0), direction = NORTH>
Oracle : UndefinedRoadBookException

Cas 2 :
<roadBook = [], position = (0,0), direction = NORTH>
Oracle : []

Couverture de tous chemins indépendants



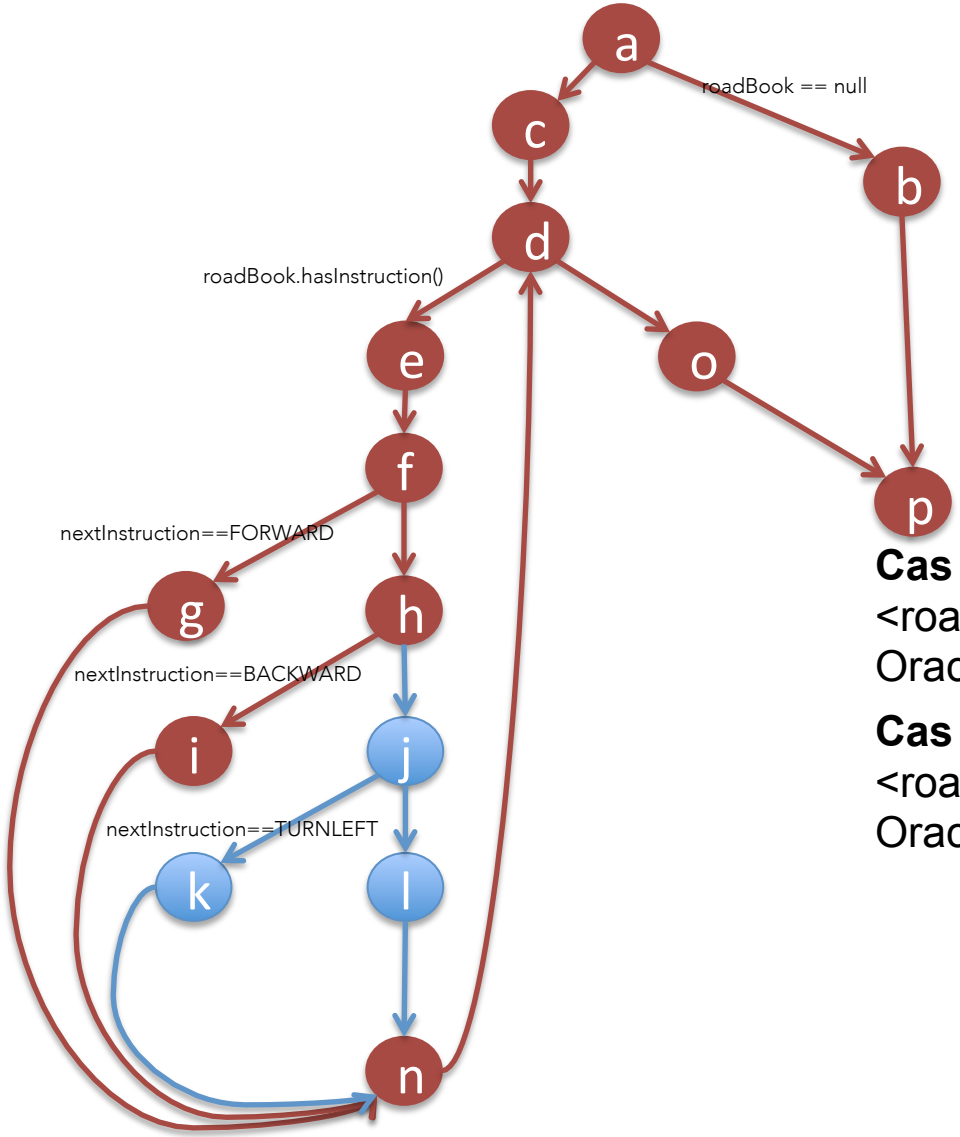
Mc Cabe : 6

Cas 1 :
<roadBook = null, position = (0,0), direction = NORTH>
Oracle : UndefinedRoadBookException

Cas 2 :
<roadBook = [], position = (0,0), direction = NORTH>
Oracle : []

Cas 3 :
<roadBook = [FORWARD], position = (0,0), direction = NORTH>
Oracle : [<(0,-1),NORTH,false>]

Couverture de tous chemins indépendants



Mc Cabe : 6

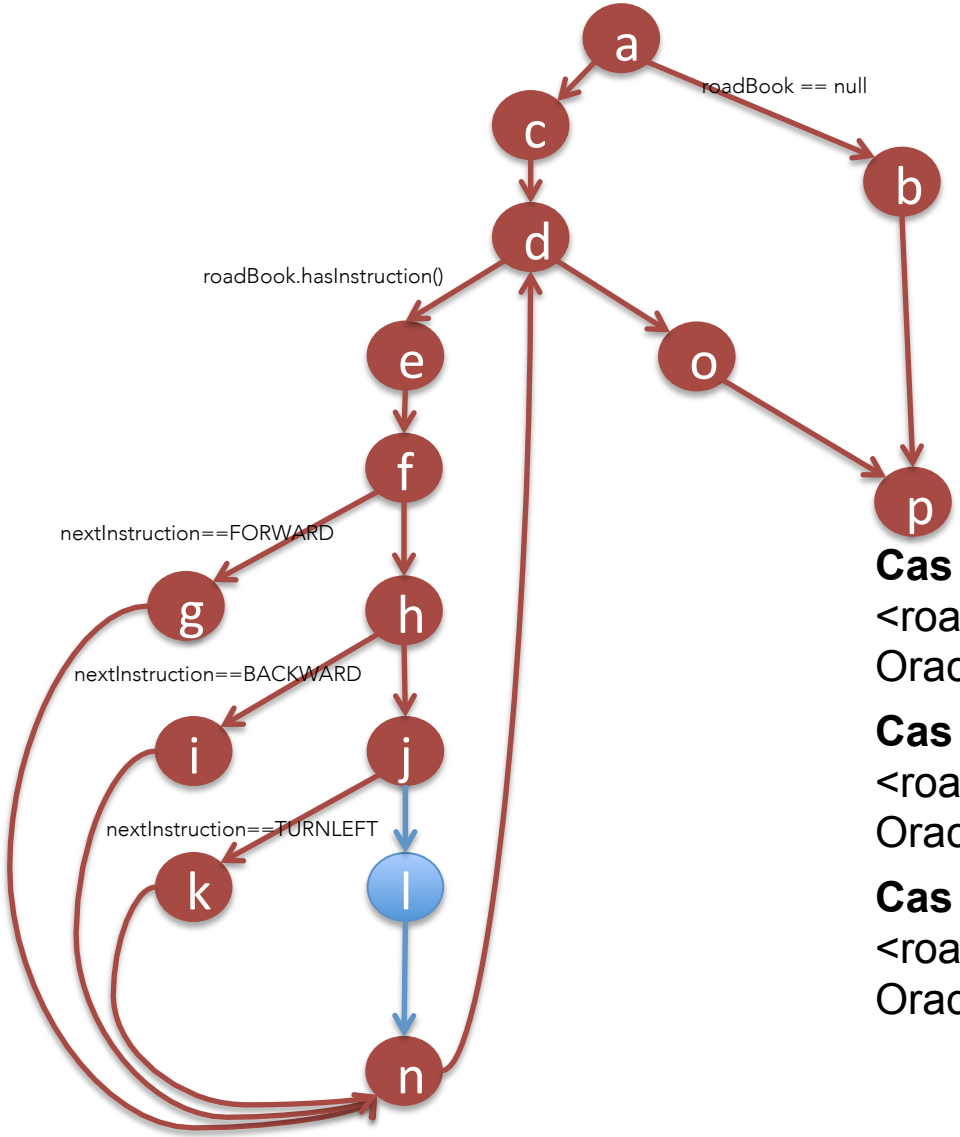
Cas 1 :
<roadBook = null, position = (0,0), direction = NORTH>
Oracle : UndefinedRoadBookException

Cas 2 :
<roadBook = [], position = (0,0), direction = NORTH>
Oracle : []

Cas 3 :
<roadBook = [FORWARD], position = (0,0), direction = NORTH>
Oracle : [<(0,-1),NORTH,false>]

Cas 4 :
<roadBook = [BACKWARD], position = (0,0), direction = NORTH>
Oracle : [<(0,1),NORTH,false>]

Couverture de tous chemins indépendants



Mc Cabe : 6

Cas 1 :
<roadBook = null, position = (0,0), direction = NORTH>
Oracle : UndefinedRoadBookException

Cas 2 :
<roadBook = [], position = (0,0), direction = NORTH>
Oracle : []

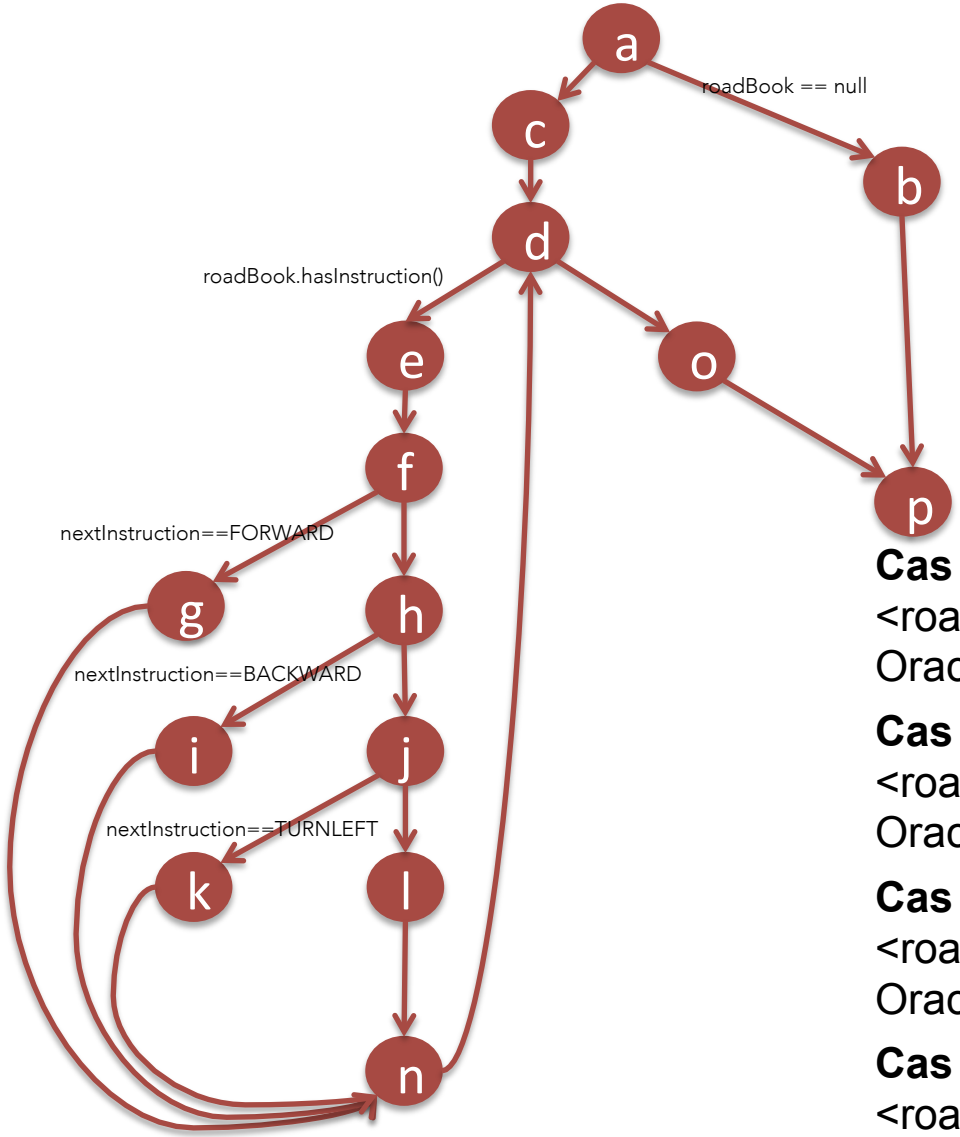
Cas 3 :
<roadBook = [FORWARD], position = (0,0), direction = NORTH>
Oracle : [<(0,-1),NORTH,false>]

Cas 4 :
<roadBook = [BACKWARD], position = (0,0), direction = NORTH>
Oracle : [<(0,1),NORTH,false>]

Cas 5 :
<roadBook = [TURNLEFT], position = (0,0), direction = NORTH>
Oracle : [<(0,0),EAST,false>]

19

Couverture de tous chemins indépendants



Mc Cabe : 6

Cas 1 :
<roadBook = null, position = (0,0), direction = NORTH>
Oracle : UndefinedRoadBookException

Cas 2 :
<roadBook = [], position = (0,0), direction = NORTH>
Oracle : []

Cas 3 :
<roadBook = [FORWARD], position = (0,0), direction = NORTH>
Oracle : [<(0,-1),NORTH,false>]

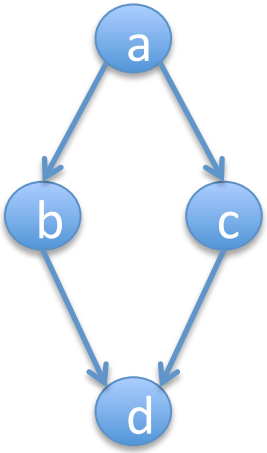
Cas 4 :
<roadBook = [BACKWARD], position = (0,0), direction = NORTH>
Oracle : [<(0,1),NORTH,false>]

Cas 5 :
<roadBook = [TURNLEFT], position = (0,0), direction = NORTH>
Oracle : [<(0,0),EAST,false>]

Cas 6 :
<roadBook = [TURNLEFT], position = (0,0), direction = NORTH>
Oracle : [<(0,0),EAST,false>]

Choix de la couverture

```
if (x>y)
      max=x;
else
      max=y;
return max
```

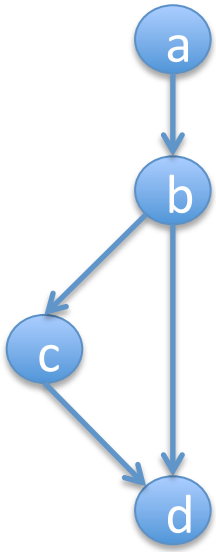


x=3, y=2 Oracle 3

x=1, y=2 Oracle 2

tous nœuds

```
read(x, y);
if (x<0)
      y=-y;
return y/x;
```

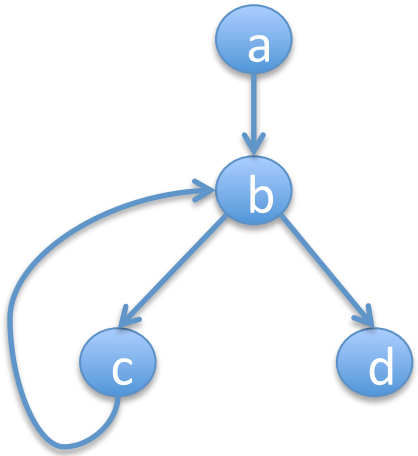


x=-2, y=4 Oracle 2

x=2, y=4 Oracle 2

tous arcs

```
read(x, y);
i=0;
while(i<x){
      y=2*y;
      i++; }
return y/x;
```



x=-2, y=4 Oracle 2

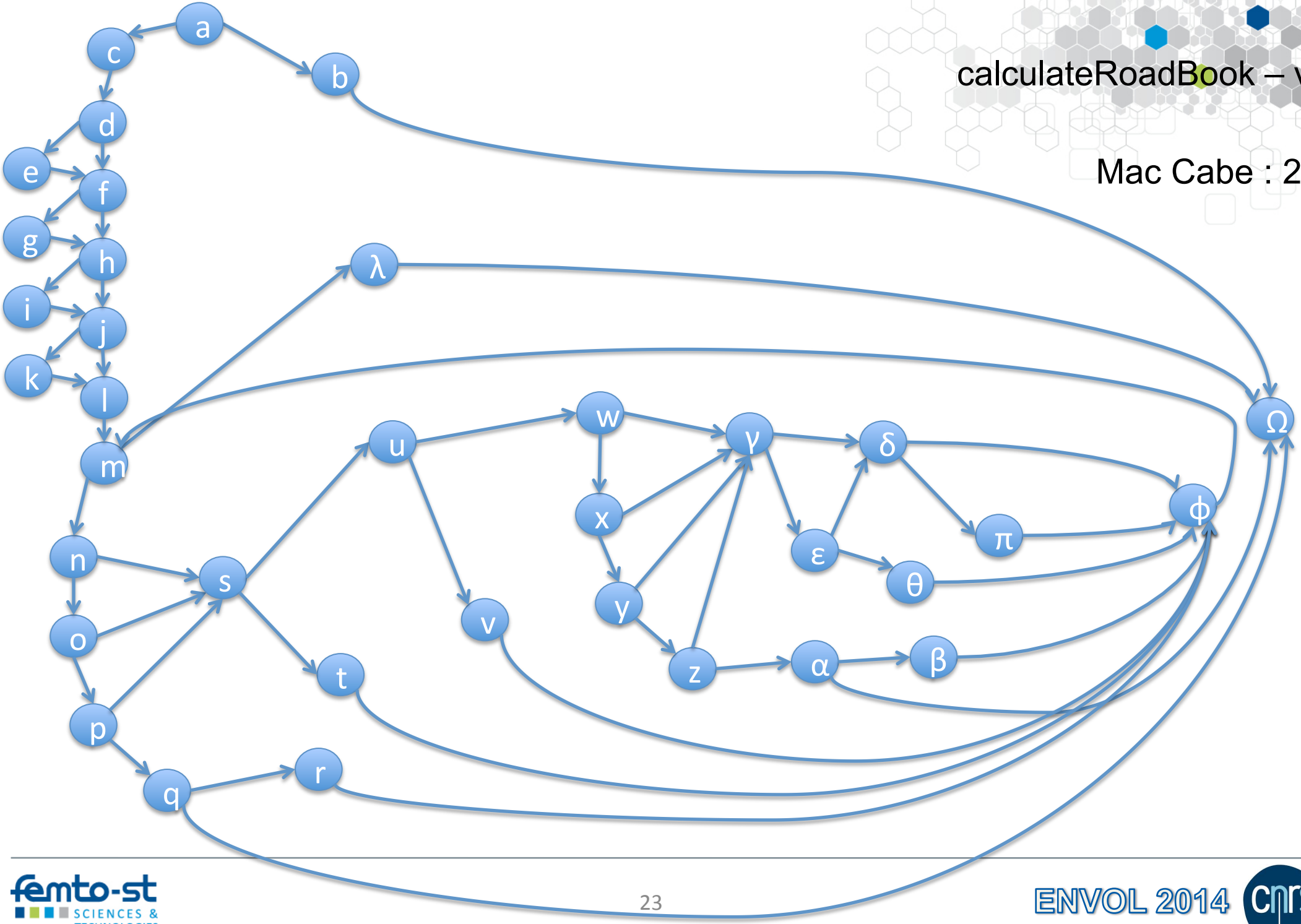
x=2, y=4 Oracle 2

tous chemins indépendants

Exercices

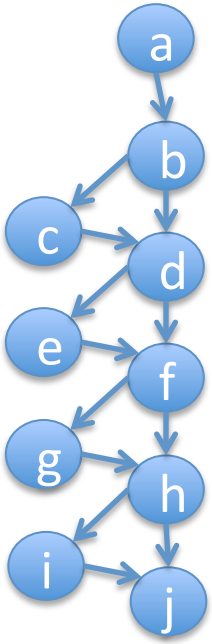
calculateRoadBook – v1

Mac Cabe : 21



calculateRoadBook – v2

Mac Cabe : 9



computePrivilegeDirection

Merci pour votre attention…