Model the traffic light system as CSP in Boolean constraint domain.
Traffic Light System

Right Turn
Forward
Left Turn

“Stop” Sensor

“Go” Sensor

Stop
Sensor

Go
Sensor

Traffic Light System

Stop
Sensor

Go
Sensor

Stop
Sensor

Go
Sensor
Someone was murdered last night, and you are summoned to investigate the murder. The objects found on the spot that do not belong to the victim include: a pistol, an umbrella, a cigarette, a diary, and a threatening letter. There are also witnesses who testify that someone had argued with the victim, someone left the house, someone rang the victim, and someone walked past the house several times about the time the murder occurred. The suspects are: Miss Linda Ablaze, Mr. Tom Burner, Ms. Lana Curious, Mrs. Suzie Dulles, and Mr. Jack Evilson. Each suspect has a different motive for the murder, including: being harassed, abandoned, sacked, promotion and hate. Other clues are given below.
The cigarette belongs to Mrs. Burner.

Neither Ms. Curious nor the person who was sacked by the victim is the author of the threatening letter.

Also, Ms. Curious does not own the pistol and she did not hate the victim.

In fact, the person who hated the victim is the one who owns the diary that disclosed this information.

The person who owns the umbrella is the one who left the victim's house without it.

It is Mrs. Dulles who walked past the house several times.
The person who argued with the victim is the man who stands a good chance of being promoted to the victim's position.

As for Miss Ablaze, she had been often harassed by the victim, but she did not write the threatening letter and did not commit the murder.

Finally, it is established that the people heard or seen by the witnesses are different people among the suspects and that they did not commit the murder;

also each evidence-object belongs to a different suspect.

You are asked to find the motive, the evidence-object, and the activity associated with each suspect; in particular, the person who committed the murder.
- JSolver is an ILOG Solver in Java

```java
public class Murder {
    public static void main(String argv[]) throws FailException{

        // refer to IlcManager
        Solver solver = new Solver();

        // to be continued
    }
}
```
The Evidence Objects

//...
// refer to IlcIntVar
Var pistol = solver.var(0, 4, "pistol");
Var umbrella = solver.var(0, 4, "umbrella");
Var cigarette = solver.var(0, 4, "cigarette");
Var diary = solver.var(0, 4, "diary");
Var letter = solver.var(0, 4, "letter");

// refer to IlcIntVarArray
VarVector allObjects = solver.varVector(pistol,
                                       umbrella, cigarette,
                                       diary, letter);

// to be continued
//...
Var argue = solver.var(0, 4, "argue");
Var leave = solver.var(0, 4, "leave");
Var rang = solver.var(0, 4, "rang");
Var walk = solver.var(0, 4, "walk");
Var murder = solver.var(0, 4, "murder");
VarVector allActions = solver.varVector(argue, leave, rang, walk, murder);

// to be continued
// ...
Var harassed = solver.var(0, 4, "harassed");
Var abandoned = solver.var(0, 4, "abandoned");
Var sacked = solver.var(0, 4, "sacked");
Var promotion = solver.var(0, 4, "promotion");
Var hate = solver.var(0, 4, "hate");
VarVector allMotives = solver.varVector(harassed,
                        abandoned, sacked,
                        promotion, hate);

// to be continued
//...  
// refer to IlcManager::add(...)  
solver.post(cigarette.eq(BURNER));  
solver.post(letter.neq(CURIOUS));  
solver.post(sacked.neq(letter));  
solver.post(pistol.neq(CURIOUS));  
solver.post(hate.neq(CURIOUS));  
solver.post(hate.eq(diary));  
solver.post(umbrella.eq(leave));  
solver.post(walk.eq(DULLES));  
solver.post(argue.eq(promotion));  
solver.post(harassed.eq(ABLAZE));  
solver.post(letter.neq(ABLAZE));  
solver.post(murder.neq(ABLAZE));  
// to be continued
Additional Clues

// ...

solver.post(allActions.allDiff());
solver.post(allObjects.allDiff());
solver.post(allMotives.allDiff());

// to be continued
//...
solver.activate(solver.generate(allObjects));
solver.activate(solver.generate(allActions));
solver.activate(solver.generate(allMotives));

while (solver.nextSolution()) {
    System.out.println(allObjects);
    System.out.println(allActions);
    System.out.println(allMotives);
}

System.out.println("Done");

private final static int ABLAZE = 0, BURNER = 1,
    CURIOUS = 2, DULLES = 3, EVILSON = 4;
Any Questions...