Constraint Based Analysis

Question 1. Let `edge` be the input relation containing all edges in a directed graph. Let the relation `path` be defined as follows:

```
path(n1, n2) :- edge(n1, n2).
path(n1, n2) :- path(n1, n3), path(n3, n2).
```

a. Is the following Datalog relation `path2` equivalent to the above relation `path` (do they always produce the same output given the same input edge relation)?

```
path2(n1, n2) :- edge(n1, n2).
path2(n1, n2) :- edge(n1, n3), path2(n3, n2).
```

Answer: Yes

b. Write a Datalog relation `R` that outputs each node `n` that has at least one incoming edge.

```
R(n) :- ?
```

Answer: `R(n) :- edge(_, n).`

c. Write a Datalog relation `S` that outputs each pair of nodes `n1`, `n2` that occur in the same cycle. Your answer may use `edge` and/or `path` as necessary.

```
S(n1, n2)   :- ?
```

Answer: `S(n1, n2) :- path(n1, n2), path(n2, n1).`

Question 2. State two benefits of writing a static analysis in Datalog instead of Java.

Answer:

1) Datalog allows the analysis writer to separate the analysis specification from the analysis implementation. This separation of concerns simplifies maintaining the analysis, ensuring its correctness, etc.

2) Datalog allows the analysis writer to only specify the analysis logic, and use off-the-shelf solvers to execute it efficiently. As better solvers are introduced, the analysis performance improves for "free".