1. **Precondition for startProject of a department is that it has at least 5 employees, including a manager**

    context Department::startProject(p: Project)

    pre:

      self.employees->size() >= 5 and

      self.employees->exists(e | e.isManager = true)

2. **The SSN of each employee is unique**

    context Employee

    inv:

      Employee.allInstances()->forAll(e1, e2 | e1 <> e2

                                              implies e1.SSN <> e2.SSN)

3. **Total number of employees in a department equals the sum of employees working on different projects controlled by that department.**

    context Department

    Inv:

      self.employees->size() = self.projects->collect(p | p.employees)->flatten()->asSet()->size()

4. **Post condition for join(Department) is that nbrEmployees increases by 1 and dEmployees is updated**

    context Employee::join(d: Department)

    post:

      d.nbrEmployees = d.nbrEmployees@pre + 1 and

      d.employees->**includes**(self)

Note: Use ***includes***, not ***including***, because we're checking membership (true/false), not adding to the collection.

5. **The location of the project belongs to the locations of its concerned departments**

    context Project

    inv:

      self.department.locations->includes(self.location)