SWE-4806: Software Verification and Validation Lab

Lab-02: An Example of Alloy-"I'm my own grandpa"

Software Requirements Specification (SRS)

Requirements of a system:

- There are two types of **persons** in a system, such as **men** and **women**. No **person** is both a **man** and a **woman**. In other words, all **persons** are either **men** or **women**.
- Every **person** has at most one **father** and at most one **mother**. **Fathers** are always **men** and **mothers** are always **women**.
- Every **man** has at most one **wife**. Wives are always **women**. In the same way, every **woman** has at most one **husband**. **Husbands** are always **men**.

Constraints:

- No person can be his/her own ancestor.
- A man's wife has that man as a husband. In the same way, a woman's husband has that woman as a wife. In mathematical notion, wife set is equal to the transpose of husband set.

Questions and Discussions:

1. Define an ***assertion*** that says the following-
    a. ***noSelfFather:*** No man is his own father.
    b. ***noSelfmother:*** No woman is her own mother.
    c. ***noSelfParent:*** Nobody is his/her own parent.
2. Define a ***function*** (***grandpas***) that takes a person as input and returns a set of grandpas of that person as output.

```
fun grandpas[p:Person]: set Man{
        ….//implement it
}
```

3. Define a ***predicate*** (***ownGrandpa***) that takes a person as input and finds a set of grandpas of that person as an example. Then run the following ***run*** command to show the graphical output and explain it.

```
pred ownGrandpa (p:Person) {
        ….// implement it
}
run ownGrandpa for 10 Person
```

4. As you may have already seen that "no instance found" has been shown in your console, which means there are no solutions within this scope (10 Person). At this time, we can change the scope with larger values. But, we have an alternative. We can define an ***assertion*** to find a solution (instance).
    a. Now, define an ***assertion*** (***noSelfGrandpa***) and check the ***assertion***.

```
assert noSelfGrandpa {
        ...//implement it
}
check noSelfGrandpa for 10 Person
```

    b. Is there any counterexample found from ***noSelfGrandpa***? What can we infer by checking this?
5. Here, we want to extend the **grandpa** notion beyond biological grandpas to include grandpas by marriage. So, we define **parent** relation in such a way that maps a person to his/her mother, father, father's wife and mother's husband. In other words, our grandpa is any man who is our parent's parent, where "**parent**" now includes stepparents.
    a. Change the ***function grandpas*** according to the above requirements.
    b. Check the ***noSelfGrandpa assertion*** and explain the counterexample.
    c. Add another ***fact***, "No person has a spouse who is also an ancestor".
    d. Finally, again check the ***noSelfGrandpa assertion*** and explain the counterexample.