Workshop 10 (Week 11) – Formal Methods

The purpose of this workshop is to practice and develop an understanding of formal methods.

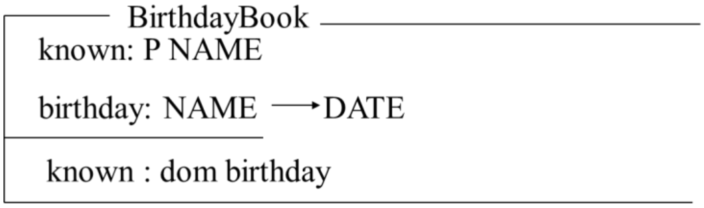
1. Concepts

1) Explain the limitations of informal specifications.
2) Explain what formal methods, formal specification, and formal verification are.
3) Explain how formal methods can facilitate software quality assurance.

2. The Z Language

Consider the BirthdayBook example described in the lecture.

1) Please describe the specification expressed by the schema FindBirthday
2) Please describe the specification expressed by the schema RBirthday:

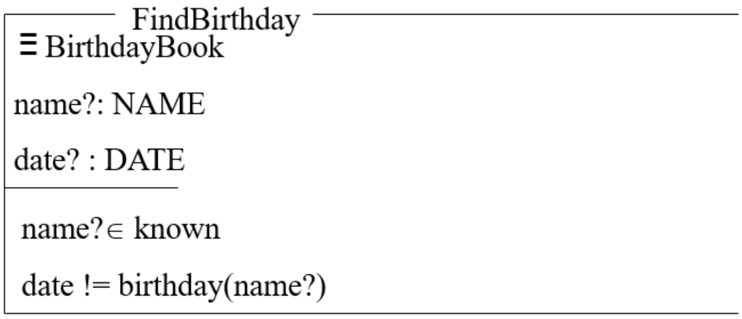


BirthdayBook

known: P NAME

birthday: NAME ⟶ DATE

known : dom birthday



FindBirthday

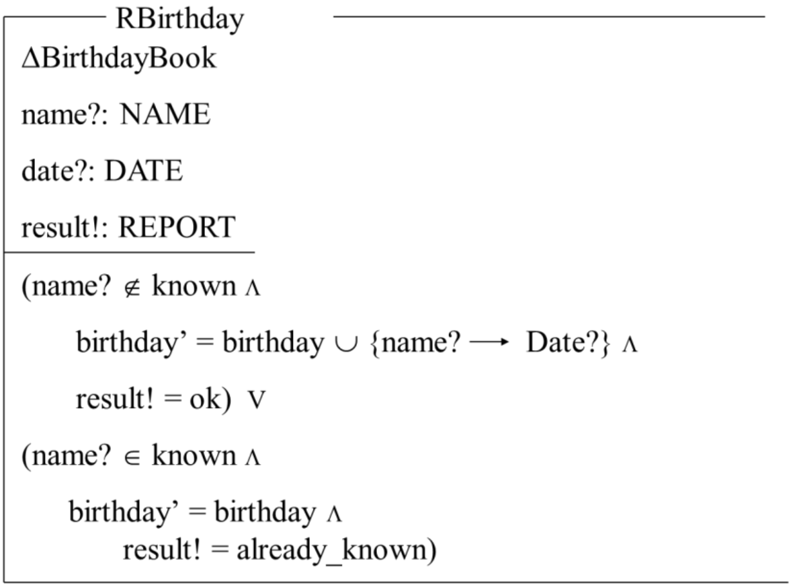
Ξ BirthdayBook

name?: NAME

date? : DATE

name?∈ known

date != birthday(name?)



RBirthday

ΔBirthdayBook

name?: NAME

date?: DATE

result!: REPORT

(name? ∉ known ∧

birthday’ = birthday ∪ {name? ⟶ Date?} ∧

result! = ok) ∨

(name? ∈ known ∧

birthday’ = birthday ∧

result! = already_known)

3. JML

As described in the lecture, Java Modeling Language (JML) can be used to write formal specification accessible to programmers. Can you describe what the preconditions and postconditions of the following myJML function are?

(The JML specification can be found at: http://www.eecs.ucf.edu/~leavens/JML/documentation.shtml)

```
//@ requires true;
//@ ensures \result == -1 ==> a.length == 0;
//@ ensures \result > -1 ==> (\forall int i; 0 <= i && i < a.length; a[\result] <= a[i]);
static public int myJML(int[] a) {
    if (a.length == 0) return -1;

    int index = 0;
    int s = 0;
    while (a.length - index > 0) {
        if (a[index] < a[s]) {
            s = index;
        }
        index = index + 1;
    }
    return s;
}
```

4. OpenJML

As described in the lecture, OpenJML is a program verification tool for Java programs that allows you to check the specifications of programs annotated in the Java Modeling Language (JML).

In this workshop you will download and try out OpenJML:

- Download and install the OpenJML tool from http://www.openjml.org/
- Apply OpenJML to detect bugs in the MaybeAdd program described in the lecture.