Karlsruher Institut für Technologie
Institut für Theoretische Informatik
Prof. Dr. B. Beckert
Dr. Vladimir Klebanov

Applying Formal Verification, SS 2012

Modeling and Analysis with Maude

Assignment 1

Solve the following Pseudo-Sudoku problem with Maude. Given: a list of elements from the set $\mathbb{N} \cup \{*\}$ of length 10. The $*$ symbol is a wild card that can be replaced with any natural number. Question: Is it possible to instantiate the wild cards in the given input such that every natural number between 0 and 9 appears in the list exactly once?

Assignment 2

Consider the Java program given on the next page. The program simulates a simultaneous transfer of funds between several bank accounts. The program has a bug: Its execution can end in deadlock.

1. Simulate this program in JavaFAN.
2. Check the program for presence of deadlock with JavaFAN.
3. Fix the bug leading to deadlock.
4. Check the program for presence of deadlock with JavaFAN.

Submit the output of JavaFAN in step 2 and the fixed program.

```
public class Account {

    int balance;
    int number;

    public Account (int nr, int bal) {
        this.number = nr;
        this.balance = bal;
    }

    public static void main(String[] argv) {
        Account a1 = new Account (1, 100);
        Account a2 = new Account (2, 100);

        Transaction t1 = new Transaction (a1, a2, 50);
        Transaction t2 = new Transaction (a2, a1, 50);
        t1.start();
        t2.start();
    }

    public String toString() {
        return "Account␣#"+number+"␣has␣balance␣"+balance;
    }
}


class Transaction extends Thread {

    private Account a1;
    private Account a2;
    private int amount;

    Transaction(Account a1, Account a2, int amount) {
        this.a1 = a1;
        this.a2 = a2;
        this.amount = amount;
    }

    public void run() {

        synchronized (a1) {
            synchronized (a2) {
                if (a1.balance >= amount) {
                    a1.balance -= amount;
                    a2.balance += amount;
                }
                System.out.println(a1.balance);
                System.out.println(a2.balance);
            }
        }
    }
}
```