Resit Exam Testing Techniques
192170015
1 July 2015
8:45 — 11:45

To make this exam:

- You are allowed to have 1 A4 sheet with your notes and nothing else.
- **Make each exercise on a separate page.**
- Write your name on each separate page that you hand in.
- Hand in the exam as well.

We wish you a lot of success!

1 What is testing?

1. Describe the purpose of testing. *(1 points)*

2. In his guest lecture, Machiel van der Bijl presented how Axini uses model-based testing in practice. Describe their main reasons for using model-based testing. *(1 points)*

2 Blackbox Testing.

The C function *saved* calculates the total sum of money that results after saving for years a certain fixed amount per year with a fixed interest rate. More precisely, on Januari 1st of each year a certain amount `amount` is put in a bank account. Each year on December 31st, the bank sents out an account summary indicating the total amount of money in the account. The bank uses the function `saved` below to compute the total amount of money after `years` years of saving.

The three inputs must be greater than or equal to 0; the output is a real value (double in C).

```
double saved(int amount, double rate, int years)
{
  int j;
  double s;

  j = 1;
  s = amount * (1.0 + rate/100);

  do
  {
    s = (s + amount) * (1.0 + rate/100.0);
    j = j+1;
  }
  while (j<years);

  return s;
}
```

1. Give a formal specification for the function `saved` as pre- and postconditions, based on the informal description. *(2 points)*

2. Use the equivalence partitioning technique to divide the input suitable equivalence classes. *(2 points)*

3. Give a test set that covers all equivalence classes. *(1 points)*

4. Extend the test set following the principle of boundary value analysis. *(2 points)*

5. Use the principle of error guessing to extend your test suite. *(2 points)*

6. Give a test case that fails on the implementation above. *(1 points)*

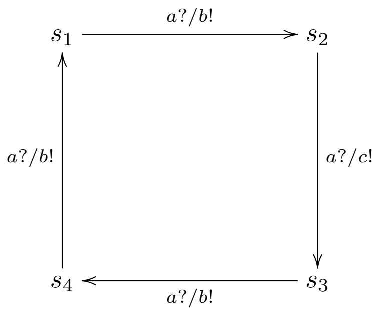

Figure 1: The FSM A, where s_1 is initial.

3 FSM testing

1. Describe the notion of soundness for FSM testing. *(2 points)*
2. Prove that the state tour method is sound. *(2 points)*
3. Prove that the state tour method is not complete. *(2 points)*
4. Use the transition test method to derive a test for the transition $s_2 \xrightarrow{a?/c!} s_3$. Describe the steps that you performed to obtain this test. *(2 points)*
5. Suppose you have tested a system implementation using FSM transition testing, and it passed all tests. Give four reasons why the implementation could not work as desired, despite the fact that FSM transition testing is complete. *(2 points)*

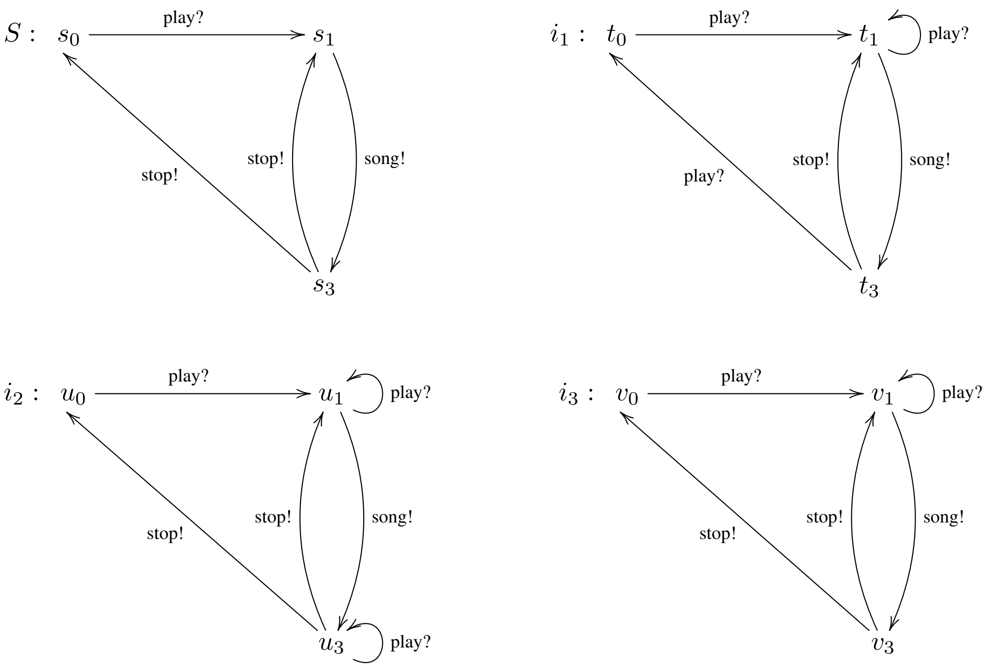

Figure 2: Transition systems S, i_1, i_2, and i_3; s_0, t_0, u_0, v_0 are initial states.

4 Ioco

The specification S in Figure 2 represents a simplified specification of an MP3 player. After pushing the play button, a song is started. When the song is over, the MP3 player either moves to a state where it starts a new song, or (if the play list is finished), it stops playing and waits for the user to push the play button again.

1. Add δ-transitions to the transition systems whenever required. *(1 points)*
2. Describe the role of quiescence in QTSs. *(1 points)*
3. Which of the IOLTSs i_1, i_2, i_3 in Figure 2 are ioco-correct implementations of S? If an implementation is incorrect provide a test case that fails on this implementation. *(8 points)*

5 Proofs

Are the following statements true or false? For a true statement give a proof, for a false statement, give a counter example.

1. Suppose that we have a sound test suite for S. Suppose that we change one of the pass verdicts in a fail. Statement: the resulting test suite is still sound. *(3 points)*

2. Suppose that we have a sound test suite for S. Suppose that we change one of the fail verdicts in a pass. Statement: the resulting test suite is still sound? *(3 points)*

3. Suppose that we have a complete test suite for S. Suppose that we change one of the fail verdicts in a pass. Statement: the resulting test suite still sound? *(3 points)*