Prof. Dr. Peter Thiemann
Stephan Arlt
Martin Wehrle                                               Sommersemester 2011

# Softwaretechnik

## Exercise Sheet 7

## Exercise 1 (8 points)

Consider the input string $s=$"xxmxoxnxtxxaxgxx". Apply the delta debugging algorithm $dd_{Min}(s, 2)$ as presented in the lecture to identify a minimal failing input, where an input $c$ fails if the substring "montag" is contained in $c$. More precisely, the test function $test(c)$ returns *FAIL* if "montag" is contained in $c$, and *PASS* otherwise. For each step of the algorithm, describe the input and the test outcome.

## Exercise 2 (8 points)

In this exercise, you are supposed to write a simple Java program that simplifes failure-inducing input. The idea is to use delta debugging to find a minimal input that causes an XML parser to fail. Download "ex07-parser.zip", which contains a binary of the parser for Windows, Linux and Mac OS, and "ex07-example.xml" from the website. The program "parser" has a small defect. The input file "ex07-example.xml" causes the parser to fail.

### Exercise 2.1

Write a *test* function. Start with a Java program that invokes the parser, as described above, and assesses the result. You may use the Java method *Runtime.getRuntime().exec()* for this. Differentiate the following three outcomes:

- The parser succesfully parses the file.

- The parser fails as in the original failure.

- The parser has another outcome, in particular parse errors.

### Exercise 2.2

Implement the delta debugging algorithm $dd_{Min}$ from the lecture.

### Exercise 2.3

Run your program. It should record all tests and the corresponding outcomes. How many tests did delta debugging take? What is the simplified failure-inducing input your program extracted from "ex07-example.xml"?

**Exercise 3 (8 points)**

The paper "Holmes: Effective Statistical Debugging via Efficient Path Profiling" written by Trishul Chilimbi, Ben Liblit, Krishna Mehra, Aditya V. Nori, and Kapil Vaswani, considers *statistical debugging*. Read this paper and answer the following questions.

- What is statistical debugging? What is the aim, how does it work?

- What is branch profiling, what is path profiling? What is the conceptual difference?

- Under which circumstances is *Holmes* expected to work well?