Prof. Dr. Peter Thiemann
Manuel Geffken
Matthias Keil Summer Term 2013

Softwaretechnik

`http://proglang.informatik.uni-freiburg.de/teaching/swt/2013/`

Exercise Sheet 8

Exercise 1

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 string c fails if all characters in "montag" are contained in c. More precisely, the test function call $test(c)$ returns $FAIL$ if all characters in "montag" are contained in c, and $PASS$ otherwise. For each step of the algorithm, describe the input and the test outcome.

Exercise 2

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 "ex08-parser.zip", which contains a binary of the parser for Windows, Linux and Mac OS, and "ex08-example.xml" from the website. The program "parser" has a small defect. The input file "ex08-example.xml" causes the parser to fail.

Exercise 2.1

Before you start the implementation answer the following questions.

- What kind of input is XML compared to the input in the previous exercise?
- In which way does your program need to reflect this difference?
- How do you recognize invalid trees? I.e. one cannot remove a node without removing its child nodes.
  Hint: An addressing scheme where the root node has address ϵ (the empty string) in contrast to 0 *as in the examples in the lecture might turn out handy in the implementation.*

Exercise 2.2

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 2.3

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

Exercise 2.4

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 "ex08-example.xml"?

Exercise 3

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?

Exercise 4 (Optional)

In the lecture "Debugging I" on slide 12 you find an interactive sequence of interaction with Firefox. Explain how can one automate this interactive sequence.
Optional: Implement this automated sequence.