Technische Universität München **Winter term 2012/13**
I7
Prof. J. Esparza / J. Křetínský **16.1.2013**

Automata and Formal Languages – Programming Assignment

Due 13.2.2013

Your task is to write a program in Java/C++ that executes operations on Büchi automata.

Grading

The same criteria apply as in the first part.

Boolean expressions

Let us define *Boolean expressions* using the following grammar:

$$\begin{aligned} r &::= r|r \mid r\&r \mid !r \mid (r) \mid \mathit{file} \\ \mathit{file} &::= a \mid \cdots \mid z \end{aligned}$$

The semantics is given by the following rules:

$$\begin{aligned} \mathcal{L}(r_1|r_2) =& \mathcal{L}(r_1) \cup \mathcal{L}(r_2) \\ \mathcal{L}(r_1\&r_2) =& \mathcal{L}(r_1) \cap \mathcal{L}(r_2) \\ \mathcal{L}(!r) =& \Sigma^* \setminus \mathcal{L}(r) \\ \mathcal{L}((r)) =& \mathcal{L}(r) \\ \mathcal{L}(\mathit{file}) =& \mathcal{L}(\mathcal{A}) \text{ where } \mathcal{A} \text{ is a Büchi automaton in the file } \mathit{file} \end{aligned}$$

where Σ is the union of alphabets of the automata referred in the in the expression. The precedence of the operators is again the following: | has the lowest priority, then &, then !.

The format of the files is the same as in the first part, i.e.

```
digraph G {
<initial state>
<list of accepting states>
<list of edges>
}
```

where:

- `<initial state>` is a line containing `<state>[shape=<diamond>];`
- `<list of accepting states>` is a sequence of lines each containing `<state>[peripheries=2];`
- `<list of edges>` is a sequence of lines each containing `<source> -> <target> [label=<label>];`
- `<source>`, `<target>`, `<state>` are decimal numbers
- `<label>` is a non-empty subsequence of $0, 1, a, b, \ldots, z$.

Functionality

Your jar file `buechi.jar` or executable `buechi` will be called with one parameter, namely a string with the name of the input file. The input file contains a Boolean expression. The corresponding files are in the same folder. The output file should contain the Büchi automaton in the same format as described above.

An example

Consider the following files:

- `bool` contains

  ```
  (!a|b)&b
  ```

- `a` contains

  ```
  digraph G {
  0[shape=diamond];
  0[peripheries=2];
  0 -> 0 [label="01"];
  }
  ```

- `b` contains

  ```
  digraph G {
  0[shape=diamond];
  1[peripheries=2];
  0 -> 0 [label="01"];
  0 -> 1 [label="0"];
  1 -> 1 [label="0"];
  }
  ```

Upon calling `java -jar buechi.jar bool` a file `bool.dot` is created. Here it has the same content as `b`.

What to hand in?

By February you have to hand in the following files into your svn from the first part (if you do not have any yet, please follow the instructions of the first part). The topmost folder shall contain a folder `buechi` with

- a compiled executable file `buechi` or executable jar file `buechi.jar`;
- subfolder `src` with source codes; the code should be well structured, easily readable and properly commented and documented, in particular every class and method should be (apart from comments in the code) immediately preceded by a comment on what it does;
- a file `description.txt` very briefly describing the structure of your source files, i.e. which file implements what;
- a file `howtocompile.txt` containing the command used to compile your files in order to get the executable; it may also contain any other comments from your side if necessary.

Note that the sources must be compilable on `lxhalle.in.tum.de` using the command written in `howtocompile.txt`. Otherwise, *no points will be awarded.*