ACME: Automata with Counters, Monoids and Equivalence

Nathanaël Fijalkow[1,2] and Denis Kuperberg[2]

[1] LIAFA, Paris 7
[2] University of Warsaw

The dedicated webpage where the tool ACME can be downloaded is

http://www.liafa.univ-paris-diderot.fr/~nath/acme.htm .

1 Installation

Linux: type “make linux” to compile, it produces the executable acme.

Windows: you need the make utility package, which you can get here:

http://gnuwin32.sourceforge.net/packages/make.htm

Type “make win” to compile, it produces acme.exe.

2 Input Files

We describe it line by line:

- the first line is the size of the automaton
- the second line is the type of the automaton:
  - c: classical
  - p: probabilistic
  - n (int): a B-automaton with n counters
- the third line is the alphabet. Each character is a letter, they should not be separated
- the fourth line is the initial states. Each state should be separated by spaces
- the fifth line is the final states. Each state should be separated by spaces
- the next lines are the transition matrices, one for each letter in the input order. A transition matrix is given by actions (like IE,RI, OO, __) separated by spaces. Each matrix is preceded by a single character line, the letter (for readability and checking purposes).
  For classical automata (without counters), the transition matrix contains only 1 and _.

  The blank lines and lines starting with "%" are ignored.

3 Available Functions

3.1 Computing the Stabilization Monoid of a B-Automaton

In text verbose mode:

```
acme -sm Examples/test_sm.txt -text
```

To visualize it using Graphviz:

```
acme -sm Examples/test_sm.txt -dotty
```

3.2 Checking the Equivalence of two B-Automata

In text verbose mode:

```
acme -equ Examples/test_equ.txt -text
```

To visualize them using Graphviz:

```
acme -equ Examples/test_equ.txt -dotty
```

3.3 Running the Markov Monoid Algorithm on a Probabilistic Automaton

In text verbose mode:

```
acme -mma Examples/test_mma.txt -text
```

To visualize it using Graphviz:

```
acme -mma Examples/test_mma.txt -dotty
```

3.4 Checking whether a Classical Automaton has the Finite Power Property

In text verbose mode:

```
acme -fpp Examples/test_fpp_true.txt -text
```

To visualize it using Graphviz:

```
acme -fpp Examples/test_fpp_true.txt -dotty
```

3.5 Drawing an automaton using Graphviz

```
acme -dotty Examples/test_sm.txt -text
```