

# Master's Thesis Presentation Notes

*Daniel Weibel*

*August 19, 2015*

## Outline

### Slide 2

1. Introduction
2. Implementation
  - Implementation of the complementation construction a part of the project
  - Includes demo
3. Study setup
  - Setup of the empirical investigation
4. Results

## Introduction

### Slide 3

- Büchi automata are like ordinary finite state automata (FSA)
- But Büchi automata run on infinite words (omega-words)
- Difference to FSA: acceptance condition
- Complementation is defined similarly as for FSA

### Slide 4

- Problem of Büchi complementation: state complexity
- State complexity defines the performance of a complementation construction

### Slide 5

- Common way to investigate the state complexity of a construction
- Maximum number of states that a construction can produce
  - Subset construction, the output automaton cannot have maximally  $2^n$  states (because there are  $2^n$  subsets of  $n$  states)
- Worst-case state complexity often used to assess general performance
  - Schewe would be “better” than Vardi and Wilke, which would be “better” than Piterman, etc.

### Slide 6

- Worst-case state complexity interesting from a theoretical point of view
- Not necessarily relevant for a practical point of view
- Empirical performance investigation
  - These three steps summarise the content of this thesis
  - And outline the rest of this presentation

## Slide 7

- Omit to describe the Fribourg construction in this presentation
- Optimisations also described in paper [Allred and Ultes-Nitsche, 2014]
- Assigned short names to the optimisations
- States of Fribourg construction consist of tuple (sequence) of subsets of input-states (compare subset construction where states consist of a single subset of input states), each set is called a component, and each component has a colour
- Optimisations can be added to basic construction and combined
- M2 dependent on M1

## Implementation

### Slide 8 (Roadmap)

- As mentioned, I implemented the Fribourg construction in order to empirically investigate its performance

### Slide 9

- Implementation as part of an existing tool: GOAL
- GOAL: dedicated omega-automata tool
  - Can also handle logic (e.g translation of temporal logic formulas to omega-automata, and games)
- Available for free, Java program
- Provides a large amount of operations on omega-automata
- Everything accessible through a graphical and a command line interface
  - Graphical: interactive use
  - Command line: automated use
- Will demonstrate the capabilities of GOAL in some minutes

### Slide 10

- Main reason to implement the Fribourg construction as a part of GOAL: contains implementations of several other complementation constructions
- Allows to compare the performance of the Fribourg construction with other constructions

### Slide 11

- How is it possible to integrate the Fribourg construction with GOAL?
- GOAL is designed for extensibility
- GOAL provides an extension point for complementation constructions
- Plugin seamlessly integrates the Fribourg construction with GOAL
  - In the GUI and in the command line interface

### Slide 12

Demo

# Study Setup

## Slide 13 (Roadmap)

### Slide 14

- Test data
  - Which automata to run the construction on
- Test scenarios
  - Which constructions and construction versions to test
- Execution
  - Execution environment, restrictions
- Outline of the slides in this section

### Slide 15

- Two different test sets: GOAL test set and Michel test set
- This slide describes the GOAL test set, the next slide the Michel test set
- Created and used for another study (by creators of GOAL)
- Automata are in GOAL File Format, hence the name GOAL test set
- Transition density:  $t \times n$  = number of transitions per alphabet symbol
- Acceptance density: ratio of accepting states
- Analysis of test set for special properties
  - May have an influence on our results

### Slide 16

- Michel test set: automata from a family of automata used by Michel
- Very “difficult” automata
- “Stress test” for constructions
- Our test set includes the four smallest of these automata
- Example: Michel 4
- Reason to include only the smallest four of these automata: complementation becomes so computation intensive, that we cannot handle it

### Slide 17

- Two basic test scenarios
- Goal of internal tests: compare different versions of the Fribourg construction
  - Unreachable states: cannot be reached from an initial state
  - Dead states: cannot reach an accepting state
- Goal of external tests: compare Fribourg construction with other constructions
  - Possible thanks to GOAL
  - Reasons for only these three other constructions
    - \* Implementations of other existing constructions are not performant enough
    - \* Representatives of three of the four complementation approaches

### Slide 18

- Both, internal and external, are run on both test sets
- This results in four test scenarios
- Which exact constructions are used in each test scenario?

- IG
  - M2 brings no overall improvement
- IM
  - Rationale same as IG, but
    - \* Michel automata are complete, thus no C
    - \* M2 brings improvement
- EG
  - Piterman, Rank, and Slice with default options (did not investigate)
  - Best version of Fribourg construction is Fribourg+M1+R2C
- EM
  - Same versions of other constructions
  - Best version of Fribourg construction is Fribourg+M1+R2C

## Slide 19

Shorthand names for test scenarios

## Slide 20

- Resource limits similar to [Tsai et al., 2011], which also used the GOAL test set
- Limited memory, limited time
- Abortion of complementation tasks require introduction of effective samples

## Slide 21

- Due to number and complexity of complementation tasks, and efficiency of implementations, execution on a HPC cluster
  - Thanks to the Joint master students of UNIFR can get access to UBELIX
- As mentioned, individual tasks executed through command line interface of GOAL
  - Glueing together of these commands by shell scripts
- Made sure to use nodes with identical specifications on cluster

## Results

### Slide 22 (Roadmap)

Remainig slides present results of the four test scenarios

### Slide 23 (Overview)

First test scenario: IG

### Slide 24

- As mentioned, analysis is based only on the effective samples (10,939)
- Here we see the mean and median complement sizes of these 10939 eff. samples
- Mean is higher than median → right-skewed (positive-skewed, right-tailed)
- Constructions from left to right:
  - Fribourg: median ca. 750 states, mean ca. 2000 states
  - Fribourg+R2C: mean and median decrease (9% complete automata)
  - Fribourg+R2C+C: mean increases, median decreases

- Fribourg+M1: mean decreases significantly
- Fribourg+M1+M2: no improvement
- Fribourg+M1+R2C: small improvement (compared to Fribourg+M1)
- Fribourg+M1+R2C+C: same picture, mean up, median down
- Fribourg+R: very small automata (61.8% universal automata)

## Slide 25

- Median complement size for each of the 110 dt/da classes
- Each class contains 100 automata
- Show acceptance and transition densities on x and y-axis
- Distribution very uneven, transition densities between 1.2 and 2.0 have higher medians
- Peak at (1.6, 0.3)
- Medians decreasing with increasing acceptance density
- Fribourg+R2C+C: high areas higher, low areas lower

## Slide 26

- Remaining versions
- First three plots: effect of M1 optimisation (switch slide back and forth)

## Slide 27

- Attempt to categorise the dt/da classes into easy medium and hard
- Calculated for each class the average of the median complement sizes of all constructions
- Results specific to GOAL test set and the Fribourg construction (not generalisable)

## Slide 28 (Overview)

Second test scenario: IM

## Slide 29

- Plot of the complement sizes of the four Michel automata for each construction
- Significant difference between Michel 1–3 and Michel 4
- Complements of Michel 4 very large (50,000 and more) for an automaton with 6 states
- Results of Michel 4 are most informative
- R2C and M1 have large impact
- M2 has positive impact (unlike for the GOAL test set)

## Slide 30

- Plot of execution times for the four Michel automata
- Difference between Michel 1–3 and Michel 4 even more significant
- But also differences between constructions more significant
- Times:
  - Fribourg: 100,976 sec, 28 hours
  - Fribourg+R2C: 27,938 sec, 7.75 hours
  - Fribourg+M1: 6,508 sec, 1 hour 50 min
  - Fribourg+M1+M2: 2,707 sec, 45 min
  - Fribourg+M1+M2+R2C: 2,332 sec, 40 min
  - Fribourg+R: like Fribourg

## Slide 31 (Overview)

Third test scenario: EG

## Slide 32

- Only 7,204 effective samples
- Mean for Rank extremely high
  - But median of Rank similar to other constructions
  - Rank has some very big complements
- Piterman clear winner
- Fribourg and Slice similar, but Slice slightly better
- Almost all of the excluded automata are provoked by Rank
- Therefore, another analysis by excluding Rank

## Slide 33

- Without Rank, 10,998 effective samples (only to aborted automata)
- Similar picture as before
  - Piterman clear winner
  - Fribourg and Slice similar, but Slice slightly better

## Slide 34

- Median complement sizes for each of the 110 dt/da classes
- Based on the 10,998 eff. samples without Rank
  - With Rank, there would be many “empty” classes, thus per-class analysis is less useful
- Other constructions show a similar pattern as the Fribourg construction
  - The same automata are hard or easy for all constructions

## Slide 35 (Overview)

Fourth test scenario: EM

## Slide 36

- Plot of complement sizes of the four Michel automata for each construction
- Regarding Michel 4: reverse of EG
- Piterman worst, and Rank best
- Fribourg better than Slice
- However, Piterman still better than Fribourg (without opt.) in IM (287,907)

## Slide 37

- Plot of execution times for the four Michel automata for each construction
- Differences between constructions even more pronounced
- Times:
  - Piterman: 75,917 sec, 21 hours
  - Slice: 159 sec. 2 min 40 sec
  - Fribourg: 2,332 sec, 39 min
  - Rank: 30 sec

- Complement of Fribourg is smaller than complement of Slice, but exec. time of Fribourg is considerably higher
- Piterman still faster than Fribourg in IM (around 100,976 sec)

## Conclusions

### Slide 38

- Regarding performance of Fribourg construction
  - R2C large impact (can be seen in IM)
  - M1 large impact
  - M2 no improvement for GOAL test set, but remarkable improvement for Michel test set
  - R2C+C: makes hard automata harder and easy automata easier
  - Fribourg can compete with Slice, not with Piterman, and is much better than Rank
- Regarding Büchi complementation in general
  - Worst-case complexities do not reflect actual performance
    - \* E.g. Rank has best worst-case complexity, Piterman has worst
    - \* Fribourg+M1+M2 has better worst-case complexity than Fribourg+M1
    - \* Actual performance is much more multifaceted
  - Surprising behaviour
    - \* Piterman very good on GOAL test set, but very bad on Michel test set
    - \* Rank very bad on GOAL test set, but very good on Michel test set
  - No overall best construction
    - \* All constructions have individual strenghts and weaknesses
- Future work
  - More profound statistical analysis