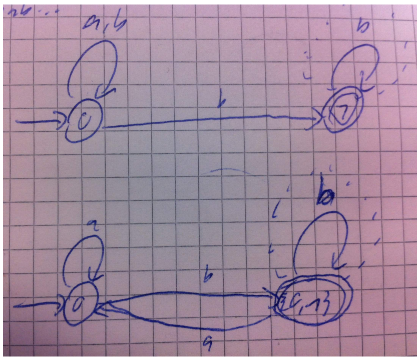
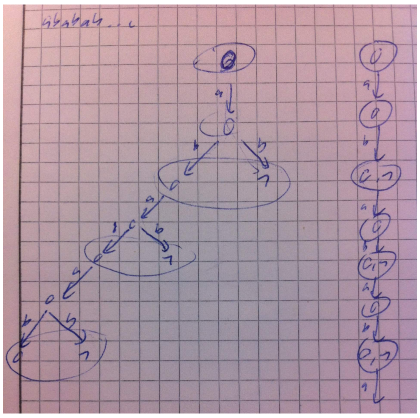
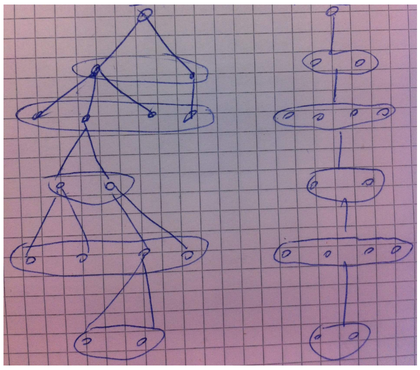
Master's Thesis Notes

Daniel Weibel, 16.04.2014

1 Theoretical Part

1.1 Introduction

Subset construction on Büchi automaton	Run trees of word ababab... on nondet. and det. automaton	Impact of subset construction on run tree
		

The subset construction merges several nondeterministic runs to a single run. All the information about specific nondeterministic runs get lost: their state sequences, if and where they die, the number of nondeterministic runs until a given point, etc.

For automata on finite words this is not a problem, because all we are interested in are the states the automaton is in after reading all the symbols of the input word.

With ω-automata, however, we are interested in the behaviour of *specific runs*. In the case of Büchi automata we need to know whether a specific run visited an accepting state infinitely often.

However, after the subset construction, this information is lost in the determinized automaton. All that we can say after the subset construction is that an *accepting state q* has been visited *infinitely often*. This can have three reasons:

1. A finite number of runs visited q *infinitely* often
2. An infinite number of runs visited q *infinitely* often
3. An infinite number of runs visited q *finitely* often

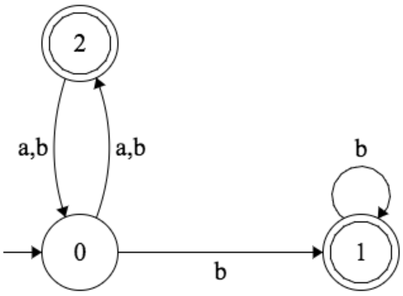
Cases 1 and 2 satisfy the Büchi acceptance condition, because there must be at least one single run that visited q infinitely often. In Case 3 however, there is no specific run that visited the q infinitely often, and thus, the Büchi acceptance conditon is not met. This is the case in the above example.

This means, we cannot distinguish the invalid Case 3 from the valid Cases 1 and 2 anymore. Hence, a "subset-constructed" automaton additionally recognises the words that lead to instances of Case 3, which are not accepted by the nondeterministic automaton.

What if in the subset construction accepting and non-accepting states are never mixed?
Lemma (informal): if accepting and non-accepting states are not mixed in subsets, then the above problem with the subset construction does not occur.
Proof (informal): subset state q = {q1,q2,..,qn}, with q1,..,qn accepting, visited infinitely often → there exists a specific run that visited at least one of q1,...,qn infinitely often.
There exists an infinite path (run) in the run tree (König's Lemma). This run necessarily visits subset state q infinitely often (one component of q at a time). Since the number of component states of q (q1,...,qn) is finite, the run must visit at least one of them infinitely often.

1.2 Questions

1.2.1 Mixing of accepting and non-accepting states



Example: mix states 0 and 2 into state q = {0,2}. In the end it must hold that if q is visited infinitely often (implying that both 0 and 2 are visited infinitely often), then there is a single run that visited 2 infinitely often. Which approach should we take for formulating the condition(s) that must hold in order to mix states?

1.2.2 Complementation

Standard subset construction on Büchi automaton A:

	Nondeterministic		Deterministic
Language	L(A)	$\subseteq$	L(D)
	||		||
Complement language	L(A')	$\supseteq$	L(D')

From the information between A and D, how do we finally arrive at A'?

2 Practical Part

- Existing Ruby program
- Web-based application
- More functionality
  - For NFA, drawing run tree of a given input word