Ausarbeitung Hauptseminar

Complementation of Büchi Automata

Johannes Bauer

February 2, 2008

Supervisor: PD Carsten Lutz

1 Introduction

J.R. Büchi proposed already in [1] a complementation method for his nondeterministic Büchi automata. However, not only did this construction involve a non-trivial combinatorial argument[7], but also a blow-up of the state space to $2^{2^{O(n)}}$. This is far from the lower bound of $2^{O(n\ log\ n)}$ established by Michel[5]. Only in [6] Safra came up with a construction that allowed a complementation in $2^{O(n\ log\ n)}$ states.

This article is a reformulation of (parts of) [2] by Felix Klaedte and presents two constructions, the first one due to Klarlund [3], using progress measures on run graphs, and the second one taking a detour over Weak Alternating Parity Automata. Both have the same complexity as the construction by Safra and in fact can be shown to be isomorphic to each other.

1.1 Preliminaries

Obviously the most central concept of this article are Büchi automata, reason enough to first introduce them in some depth:

Definition 1.1. Structurally identical to nondeterministic finite automata (NFA), nondeterministic Büchi automata[1] are tuples $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$ in which Q is a finite set of states, Σ is a finite alphabet, $\delta : Q \to 2^Q$ the transition function, q_i an initial state and $F \subseteq Q$ a set of accepting states. What distinguishes a Büchi automaton from an NFA is the acceptance condition: while the latter accept finite Words $\alpha_f \in \Sigma^*$ over the alphabet, the former accept infinite words $\alpha_b \in \Sigma^\omega$, the formal definition being as follows:

[1] 'Büchi automata' for short, as this article does not consider deterministic Büchi automata.

The Büchi automaton $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$ accepts the word $\alpha \in \Sigma^\omega$ iff there is a sequence $\rho = r_0 r_1 r_2 \ldots$ of elements of Q such that, $r_0 = q_i$ and, for $k \geq 0$, $r_{k+1} \in \delta(r_k)$, and there are infinitely many indeces $l \in \omega$ such that $r_l \in F$.

Any sequence over Q fulfilling the first condition, ie. adhering to the transition function, is called a *run*. Any such sequence containing infinitely many states in F is called an *accepting run*.

The language $\mathcal{L}(\mathcal{B})$ of a Büchi automaton is the set of words it accepts. As the title of this article suggests, Büchi automata can be complemented: For every Büchi automaton $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$ another Büchi automaton $\overline{\mathcal{B}}$, called its *complement*, can be found, accepting — little surprise here — the *complement language* $\overline{\mathcal{L}(\mathcal{B})}$:

$$\overline{\mathcal{L}(\mathcal{B})} = \Sigma^\omega \backslash \mathcal{L}(\mathcal{B})$$

In its presentation of two methods to construct complemented Büchi automata, this article will make extensive use of graphs and operations on them, which is why they are defined here, along with some non-standard extensions and the relevant notation:

Definition 1.2. A *graph* is a 2-tuple of the form (V, E), where V is the set of vertices and $E \subseteq V \times V$ the set of edges.

For a graph $\mathcal{G}$ and a set S the graph $\mathcal{G}' = (V \backslash S, E \backslash (S \times S))$ is denoted by $\mathcal{G} \setminus S$.

A graph (V, E) is called a *sliced graph* over some finite set Q, iff for $i, j \in \omega$ and $p, q \in Q$:

$$V = \bigcup_i S_i$$

and

$$((p, i), (q, j)) \in E \Rightarrow j = i + 1.$$

where $S_i \subseteq Q \times i$ are called its *slices*.

A *marked graph* is a 3-tuple (V, E, C), where the additional $C \subseteq V$ can be understood as a binary labeling of the vertices.

A graph (V, E) contains *paths* which are finite or infinite sequences $\pi = p_0 p_1 p_2 \ldots$ such that $(p_i, p_{i+1}) \in E$ for $i \geq 0$. A *suffix* of a path $\pi = p_0 p_1 p_2 \ldots$ is a sequence $o_0, o_1, o_2 \ldots$ where for some fixed $i_0 \in \omega$ and $i \geq 0$: $o_i = p_{i_0 + i}$. A *suffix* of a path π is finite, if π is, and infinite otherwise. The notation $\pi(i)$ is used for the i^{th} element of a path or any other sequence. $Occ(s)$ is the set of elements of a sequence: $Occ(s) := \{s(i) \mid i \geq 0\}$.

We say that some vertice q is *non-trivially reachable* from another vertice p in a graph, formally $q \in R(p)$ iff there is a path of non-zero length in $\mathcal{G}$ starting at p and terminating at q.

2 A Run Graph Characterization

While it suffices to find one accepting run of a Büchi automaton $\mathcal{B}$ on an infinite word α to decide whether or not $\alpha \in \mathcal{L}(\mathcal{B})$, proving that $\alpha \in \Sigma^\omega \setminus \mathcal{L}(\mathcal{B})$ is equivalent to proving that there is no accepting run of $\mathcal{B}$ on α, ie. that all runs of $\mathcal{B}$ on α are non-accepting.

The most central definition in this section, thus, is the one of the *run graph*, which is a representation of all possible runs of a Büchi automaton on a word, and which can be used to argue about them:

Definition 2.1. A *run graph* $\mathcal{G} = (V, E, C)$ for a Büchi automaton $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$ and an infinite word $\alpha = a_0a_1a_2\dots$ is a sliced, marked graph where the infinite sequence of slices $S_0, S_1, S_2\dots$ is inductively defined by

$$\begin{aligned} S_0 \quad &= \{(q_i, 0)\} \\ S_{k+1} &= \{(q, k+1) \mid q \in \delta(p, a_k), (p, k) \in S_k\} \end{aligned}$$

The set of edges E of $\mathcal{G}$ is defined according to the transition relation δ and the definition of the slices:

$$E = \{((p, k), (q, k+1)) \mid (p, k) \in S_k, q \in \delta(p, a_k)\}$$

Finally, C is the set of vertices whose first component is an accepting state:

$$C = \{(q, k) \in S_k \mid k \geq 0, q \in F\}$$

It is easy to see that the run graph indeed is a representation of all possible runs of $\mathcal{B}$ on α in that for every run ρ of $\mathcal{B}$ on α there is a path π in $\mathcal{G}$ such that for $i \geq 0$ it holds that $\pi(i) = (\rho(i), i)$ and vice versa. Furthermore, the question of whether $\mathcal{B}$ accepts α is equivalent to the question of whether there is a path $\pi = p_0p_1p_2\dots$ in $\mathcal{G}$ such that for infinitely many $i \in \omega : p_i \in C$. Such a path and the run graph containing it are called *accepting*.

Generally, there are vertices in a run graph which are not part of an accepting path, and which can be removed without changing the run graph's property of being accepting or not. We will successively remove them in order to reduce the problem of proving the non-existence of an accepting path in the original run graph to proving its non-existence in a smaller graph.

Let us first consider those vertices that do not have any successors in C. They are elements of a set called the *unmarked boundary* of a run graph:

Definition 2.2. For a graph (V, E, C), we define the *unmarked boundary*:

$$U(\mathcal{G}) = \{q \in V \mid \ R(q) \cap C = \emptyset\}$$

As suggested above, nodes in the unmarked boundary can be removed from the run graph and it will still be accepting iff it was, before removing them:

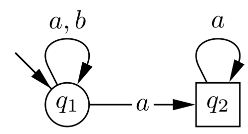

Figure 1: Büchi Automaton $\mathcal{B}_1$

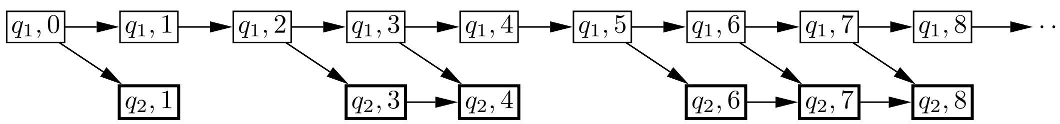

Figure 2: Run Graph $\mathcal{G}_1$

Proposition 2.3. *A run graph $\mathcal{G} = (V, E, C)$ for a Büchi automaton $\mathcal{B}$ and a word α is accepting iff $\mathcal{G} \setminus U(\mathcal{G})$ contains an accepting path π.*

Proof. "$\Leftarrow$": Any path that is in $\mathcal{G} \setminus U(\mathcal{G})$ is also in $\mathcal{G}$.

"$\Rightarrow$": Assume to the contrary that there is an accepting path $\pi = p_0 p_1 p_2 \ldots$ in $\mathcal{G}$ which is not a path in $\mathcal{G} \setminus U(\mathcal{G})$. Then for some $i \geq 0$ we have that $p_i \in U(\mathcal{G})$. But then $R(p_i) \cap C = \emptyset$ in $\mathcal{G}$ and thus $p_k \notin C, k > i$ which is a contradition to the assumption that π is accepting. □

An empty run graph does not contain any paths at all and is therefore non-accepting. Hence it seems natural to iterate Proposition 2.3 and check whether repeatedly removing the unmarked boundary will always result in an empty graph for a non-accepting run graph. Unfortunately it does not:

To see this, consider the Büchi automaton $\mathcal{B}_1$ depicted in Figure 1[2]. It accepts the language given by the ω-regular expression $\{a, b\}^* a^\omega$. The run graph $\mathcal{G}_1$ for $\mathcal{B}_1$[3] and the word $\alpha = abaabaaabaaaab \ldots \notin \mathcal{L}(\mathcal{B}_1)$ is sketched in Figure 2. Note that, for every number $n \in \omega$, there is a finite path in $\mathcal{G}_1$ containing n vertices in C.

Let U^n be defined inductively:

$$\begin{aligned} U^0(\mathcal{G}) &= \emptyset \\ U^{i+1}(\mathcal{G}) &= U^i(\mathcal{G}) \cup U(\mathcal{G} \setminus U^i(\mathcal{G})) \end{aligned}$$

It is clear, then, that $\mathcal{G}_1 \setminus U^n(\mathcal{G}_1)$ is not the empty graph regardless of n.

However, all vertices in $\mathcal{G}_1$ which are elements of C have only finitely many successors and can therefore not be part of an infinite path, and thus of an accepting path. Removing all vertices with only a finite number of successors and then the unmarked

[2] Rectangular nodes indicate states in F

[3] Boldly outlined boxes indicate vertices in C

boundary of the resulting graph in fact leaves $\mathcal{G}_1$ empty. This motivates the definition of another set of vertices which can be removed from a run graph to yield a smaller, equivalent graph:

Definition 2.4. Let the *finite boundary* be defined as:

$$F(\mathcal{G}) = \{q \in V \mid \ |R(q)| = k, k \in \omega\}$$

The following proposition shows that removing the finite boundary from a run graph indeed preserves acceptance:

Proposition 2.5. *A run graph $\mathcal{G} = (V, E, C)$ is accepting iff $\mathcal{G} \setminus F(\mathcal{G})$ contains an accepting path π.*

Proof. **"⇒"** Any path in $\mathcal{G} \setminus F(\mathcal{G})$ is also a path in $\mathcal{G}$. Thus, if $\mathcal{G} \setminus F(\mathcal{G})$ contains some π containing infinitely many vertices in C, then so does $\mathcal{G}$.

"⇐" Suppose there is some accepting path $\pi = p_0p_1p_2\dots$ in $\mathcal{G}$. Then π is infinite and therefore, for every $p_i, i \geq 0$ it holds that $p_i \notin F(\mathcal{G})$ and thus, π is also a path of $\mathcal{G} \setminus F(\mathcal{G})$.

□

We will see that repeatedly removing finite boundary and unmarked boundary from a non-accepting graph always results in an empty graph. But first, the following definition will be needed:

Definition 2.6. The *width* of a run graph $\mathcal{G}$, denoted by $||\mathcal{G}||$, is defined as the limes superior of the sequence of slices $(|S_n|)$, ie. the greatest cardinality of slices occurring infinitely often.

Lemma 2.7. *Let $\mathcal{G} = (V, E, C)$ be a non-accepting run graph with $||\mathcal{G}|| > 0$. Let $\mathcal{G}_f = \mathcal{G} \setminus F(\mathcal{G})$ and $\mathcal{G}_u = \mathcal{G}_u \setminus U(\mathcal{G}_b)$. Then*

$$||\mathcal{G}|| > ||\mathcal{G}_u||.$$

Proof. Since $||\mathcal{G}|| > 0$, there are infinitely many slices S_i of $\mathcal{G}$ such that $|S_i| > 0$. By the structure of a run graph, there must be an infinite path π in $\mathcal{G}$ starting at its root v_0. Since $F(\mathcal{G})$ does not contain any nodes belonging to an infinite path, π must also be in $\mathcal{G}_f$. Also, $p_{i_0+i} \notin C$ for $i \geq 0$ and some suffix $s = p_{i_0}, p_{i_0+1}, p_{i_0+2}\dots$ of π, because $\mathcal{G}$ is non-accepting. Then $p_{i_0+i} \in S_{i_0+i} \setminus S_{u,i_0+i}$ for the slices $S_{u,k}$ of $\mathcal{G}_u$. Therefore $|S_{i_0+i}| > |S_{u,i_0+i}|$ and $||\mathcal{G}|| > ||\mathcal{G}_u||$. □

Theorem 2.8. *Let $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$ be a Büchi automaton, $\alpha \in \Sigma^\omega$ and $\mathcal{G}$ the corresponding run graph. Define the sequence of run graphs $\mathcal{G}_0, \mathcal{G}_1 \dots$ by*

$$\begin{array}{ll} \mathcal{G}_0 & := \mathcal{G} \\ \mathcal{G}_{2k+1} & := \mathcal{G}_{2k} \setminus F(\mathcal{G}_{2k}) \\ \mathcal{G}_{2k+2} & := \mathcal{G}_{2k+1} \setminus U(\mathcal{G}_{2k+1}). \end{array}$$

Then $\mathcal{B}$ does not accept α iff $\mathcal{G}_{(2|Q|+1)}$ is the empty graph.

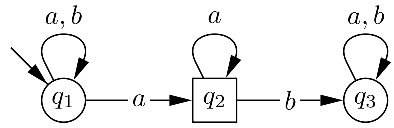

Figure 3: Büchi Automaton $\mathcal{B}_2$

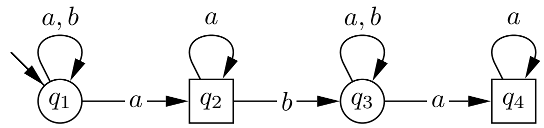

Figure 4: Büchi Automaton $\mathcal{B}_3$

Proof. **"⇒"** $\mathcal{B}$ does not accept α iff $\mathcal{G}$ is non-accepting. By Lemma 2.7, $||\mathcal{G}_i|| > ||\mathcal{G}_{i+2}||$ while $||\mathcal{G}_i|| > 0$. Since $S_k \subseteq Q \times \{k\}$ for every slice S_k of $\mathcal{G}$, and $||\mathcal{G}|| = |S_j|$ for infinitely many j, $||\mathcal{G}|| \leq |Q|$. Therefore $||\mathcal{G}_{2|Q|}|| = 0$ and

$$||\mathcal{G}_{2|Q|}|| = 0 \Rightarrow \mathcal{G}_{2|Q|+1} = \mathcal{G}_{2|Q|} \setminus F(\mathcal{G}_{2|Q|}) \text{ is the empty graph.}$$

"⇐" By Propositions 2.3 and 2.5, for every $i \in \omega$ we have that $\mathcal{G}_i$ is accepting iff $\mathcal{G}_{i+1}$ is. The empty graph does not contain any infinite paths and therefore is non-accepting. Thus, $\mathcal{G}_{2|Q|+1}$ being empty entails that $\mathcal{G}$ is non-accepting and therefore $\mathcal{B}$ does not accept α.

□

A Remark On The Sequence $(\mathcal{G}_i)$ When looking at the example given above by the Büchi automaton $\mathcal{B}_1$, one might be tempted to think that taking turns in removing the finite boundary and the unmarked boundary in the sequence $(\mathcal{G}_i)$ is not necessary. To understand why it is, it is helpful to take a look at Figures 3-4: Note that the difference between $\mathcal{B}_1$ and $\mathcal{B}_2$ is the additional state q_3 and that $\mathcal{B}_3$ is $\mathcal{B}_2$ plus the state q_4. $\mathcal{G}_3$ is the run graph for $\mathcal{B}_3$ and the same word $\alpha = abaabaaabaaaab\ldots \notin \mathcal{L}(\mathcal{B}_3)$ as for $\mathcal{G}_1$.

It is clear that, just as for $\mathcal{G}_1$, there is no n such that $\mathcal{G}_3 \setminus U^n(\mathcal{G}_3)$ is the empty graph. But unlike for $\mathcal{G}_1$, also $\mathcal{G}_3 \setminus F(\mathcal{G}_3) \setminus U(\mathcal{G}_3 \setminus F(\mathcal{G}_3))$ is not empty. In fact $\mathcal{G}_3 \setminus F(\mathcal{G}_3)$ is the run graph for $\mathcal{B}_2$ and α, and $\mathcal{G}_3 \setminus F(\mathcal{G}_3) \setminus U(\mathcal{G}_3 \setminus F(\mathcal{G}_3))$ is $\mathcal{G}_1 \setminus U(\mathcal{G}_1)$.

This game can be played ad infinitum: per added state to the Büchi automaton, one more step has to be taken before the run graph is empty.

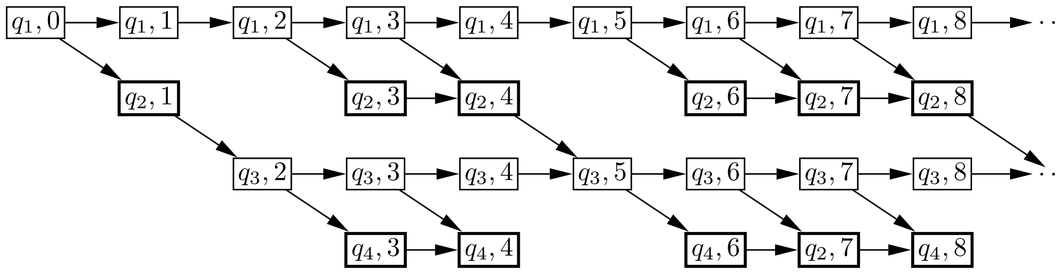

Figure 5: Run Graph $\mathcal{G}_3$

3 Progress Measures

In this section, functions on the vertices of run graphs called progress measures will be developed by abstracting the run graph characterization of Section 2. It will be shown that finding a progress measure for a run graph is equivalent to proving that it is non-accepting and that Büchi automata can be constructed to check for the existence of progress measures. This will yield a construction for the complementation of Büchi automata in $2^{O(|Q| \, log|Q|)}$ states.

3.1 From the Run Graph Characterization to Progress Measures

In Section 2 it was shown that a Büchi automaton accepts a word α iff for the run graph $\mathcal{G}$ and the sequence $(\mathcal{G}_i), i \in \omega$, defined in Theorem 2.8, the graph $\mathcal{G}_{(2|Q|+1)}$ is the empty graph. In this sequence, $\mathcal{G}_{i+1}$ is a subgraph of $\mathcal{G}_i$ for $i \geq 0$, and thus, if $\mathcal{G}_{(2|Q|+1)}$ is empty, every vertice in the run graph is a vertice of some $\mathcal{G}_{k_0}$ and not of $\mathcal{G}_{k_0+k}$ for $k_0 \leq 2|Q|$ and every $k \geq 1$.

Thus, Theorem 2.8 lends itself to the following corollary:

Corollary 3.1. *Let $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$ be a Büchi automaton, α a word in Σ^ω and $\mathcal{G} = (V, E, C)$ the run graph for $\mathcal{B}$ and α. Then $\mathcal{B}$ rejects α iff there is a function $\mu : V \to \{1, \dots, 2|Q| + 1\}$ such that*

$$\mu(v) = k_0 + 1 \quad | \; v \in V$$

Where $k_0 \in \omega$ is the unique index for which v is a node in $\mathcal{G}_{k_0}$ and not in $\mathcal{G}_{k_0+1}$ in the sequence $(\mathcal{G}_i)$ defined in Theorem 2.8.

We will see that the existence of any function $\mu_m : V \to \{0, \dots, 2m + 1\}$ proves that the run graph is non-accepting, provided μ_m shares the following properties with the function μ defined in Corollary 3.1:

- it induces a monotonically decreasing sequence on any path in the graph,

- no sequence induced by it on a path in the run graph has an infinite suffix $kkk\ldots$, where $k \in \omega$ is odd,
- for any path $\pi = p_0p_1p_2\ldots$ it is true that $\mu_m(p_i) = \mu_m(p_{i+1})$ only if $\mu_m(p_i)$ is odd or $p_{i+1} \notin C$

We call such a function a *progress measure of size m.*

First, we show that μ really does have these properties:

Lemma 3.2. *For a non-accepting run graph $\mathcal{G} = (V, E, C)$, the function*

$$\mu(v) = k_0 + 1 \quad | \; v \in V$$

Where $k_0 \in \omega$ is the unique index for which v is a node in $\mathcal{G}_{k_0}$ and not in $\mathcal{G}_{k_0+1}$ in the sequence $(\mathcal{G}_i)$ defined in Theorem 2.8 is a progress measure of size $|Q|$.

Proof. μ is a function from V to $\{1, \ldots, 2|Q| + 1\}$.

For any path $\pi = p_0p_1p_2\ldots$ in $\mathcal{G}$, the sequence $(\mu(p_i))$ is monotonically decreasing: Assume it is not. Then there is some $i \in \omega$ such that $\mu(p_i) < \mu(p_{i+1})$. Let $\mu(p_i) = k$. Then, in $\mathcal{G}_k$, p_i does not have a successor in C or only finitely many successors, but, since $\mu(p_{i+1}) > k$, p_{i+1} does. But this is impossible because all successors of p_{k+1} are also successors of p_k.

For any path $\pi = p_0p_1p_2\ldots$ in $\mathcal{G}$, the sequence $(\mu(p_i))$ does not have an infinite suffix $(\mu(p_{i_0+i})) = kkk\ldots$ where k is odd, for some $i_0 \in \omega$ and $i \geq 0$. Suppose it does. Then p_{i_0} has infinitely many successors in $\mathcal{G}_{k-1}$, leading to a contradiction because, by the definition of μ and the sequence $(\mathcal{G}_i)$, $\mu(p_{i_0}) = k$ entails that $p_{i_0} \in F(\mathcal{G}_{k-1})$ if k is odd.

For any path $\pi = p_0p_1p_2\ldots$ in $\mathcal{G}$, $k = \mu(p_i) = \mu(p_{i+1})$ only if k is odd or $p_{i+1} \notin C$. This is certainly true because $p_i, p_{i+1} \in U(\mathcal{G}_{k-1})$ and thus p_i cannot have a successor in C in $\mathcal{G}_{k-1}$.

□

Next on the agenda is to prove that the existence of a progress measure on a run graph entails that it is non-accepting. Together with Corollary 3.1 and Lemma 3.2, this will give us the result of this section: that a progress measure of size $|Q|$ can be found for a run graph for some Büchi automaton $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$ and some word α iff the run graph is non-accepting.

Lemma 3.3. *Let $\mathcal{G} = (V, E, C)$ be a run graph for some Büchi automaton $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$ and some word $\alpha \in \Sigma^\omega$ and $\mu_m : Q \to \{1, \ldots, 2m+1\}$. Then, if μ_m is a progress measure, $\mathcal{G} = (V, E, C)$ is non-accepting.*

Proof. Assume $\mathcal{G}$ is accepting. Then $\mathcal{G}$ contains some accepting path $\pi = p_0p_1p_2\ldots$. Since the sequence $(\mu_m(p_i)), i \in \omega$ is monotonically decreasing and bounded from below by 1, it must have an infinite suffix $(\mu_m(p_{i_0+i})) = kkk\ldots$, for some $i_0 \in \omega$ and $i \geq 0$. k, then, cannot be odd by the definition of progress measures. π being an accepting path, however, implies that there are infinitely many $i \in \omega$ for which

$p_i \in C$. But then there are also infinitely many $i \geq 1$ such that $p_{i_0+i} \in C$, which yields a contradiction, because μ_m is a progress measure and thus, for $n \geq 1$, $\mu_m(p_{i_0+n}) = k$, $p_{i_0+n} \in C$ and k being even implies that $\mu_m(p_{i_0+n-1}) \neq k$. □

Now everything is in place to establish the main result of this section: the equivalence of finding a progress measure of size $|Q|$ for the run graph for some Büchi automaton $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$ and some word α and proving that it is non-accepting.

Theorem 3.4. *A Büchi automaton $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$ rejects a word α iff there exists a progress measure of size $|Q|$ for the run graph of $\mathcal{B}$ and α.*

Proof. By Corollary 3.1 and Lemma 3.2 there is a progress measure of size $|Q|$ if $\mathcal{B}$ rejects α and by Lemma 3.3 the run graph for $\mathcal{B}$ and α is non-accepting, if there is progress measure of size $m, m \in \omega$. □

3.2 Complementing Büchi Automata

The last section introduced progress measures and showed that there exists a progress measure for a run graph for a Büchi automaton $\mathcal{B}$ and a word α iff $\mathcal{B}$ rejects α. With this result, containment of a word in the complement language $\overline{\mathcal{L}(\mathcal{B})} = \Sigma^\omega \setminus \mathcal{L}(\mathcal{B})$ of a Büchi automaton $\mathcal{B}$ can be decided on the level of run graphs. However, we are after a construction for the complementation of Büchi automata, ie. a Büchi automaton which decides the word problem for the complement of some other Büchi automaton's language.

Intuitively, the construction in this section yields, for every Büchi automaton $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$, another Büchi automaton $\overline{\mathcal{B}}_m$ which, on input of a word α, simulates going slice by slice through the run graph $\mathcal{G}$ for $\mathcal{B}$ and α and, if there exists a progress measure of size m for $\mathcal{G}$, guesses its value for every vertex in every slice. $\overline{\mathcal{B}}_m$ accepts α if it successfully guesses the correct progress measure, and otherwise rejects it. $\overline{\mathcal{B}} := \overline{\mathcal{B}}_{|Q|}$ will then be the complement automaton of $\mathcal{B}$.

States and transition function of $\overline{\mathcal{B}}_m$ for $\mathcal{B}$ will be defined such that the state $\overline{q}$ of $\overline{\mathcal{B}}_m$, after reading n input letters of α, represents the slice S_n of the run graph for $\mathcal{B}$ and α and, for every vertice $(q, n) \in S_n$, the value $\mu((q, n))$ of the guessed progress measure.

Let Ψ denote the set of all partial functions $\psi : Q \to \{1, \ldots, 2m + 1\}$. Then, more specifically, every state $\overline{q}$ of $\overline{\mathcal{B}}_m$ is a pair (ψ, P) such that $\psi \in \Psi$ and $P \subseteq Q$. The transition function will ensure that, if $\overline{\mathcal{B}}_m$ is in state $\overline{q} = (\psi, P)$ after n steps, then the domain of ψ agrees with the n^{th} slice of the run graph in the sense that $dom(\psi) \times \{n\} = S_n$. The value $\psi(q)$ for $q \in Q$ will be the value $\mu((q, n))$ of the guessed progress measure for the vertex $(q, n) \in S_n$.

All states $(\psi, \emptyset)$ will be accepting states and the transition function will enforce that an accepting run generates a valid progress measure.

Theorem 3.5. *Let $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$ be a Büchi automaton accepting $\mathcal{L}(\mathcal{B})$. Then $\overline{\mathcal{B}}_m = (\Psi \times 2^Q, \Sigma, \overline{\delta}_m, q_I, \Psi \times \{\emptyset\})$ accepts a word α iff there exists a progress measure of size m for the run graph of $\mathcal{B}$ and α, where*

$q_I = (\{(q_i, 2m+1)\}, \emptyset)$

$(\psi', P') \in \overline{\delta}_m((\psi, P), a)$ iff

1. $q' \in dom(\psi') \iff \exists q \in dom(\psi) : q' \in \delta(q, a)$
2. If $q' \in \delta(q, a)$ then $\psi'(q') \leq \psi(q)$ and $\psi'(q') < \psi(q)$ if $\psi(q)$ is even and $q' \in F$.
3. If $P = \emptyset$, then $P' = \{q' \in dom(\psi') \mid \psi'(q') \text{ is odd}\}$
4. If $P \neq \emptyset$, then $P' = \{q' \mid \exists q \in P : q' \in \delta(q, a), \psi(q) = \psi'(q') \text{ is odd }\}$

Proof. **"⇒"** We show that a progress measure of size m for the run graph $\mathcal{G} = (V, E, C)$ of $\mathcal{B}$ and α can be extracted from an accepting run ρ of $\overline{\mathcal{B}}_m$ on α.

Let $\mu : V \Rightarrow \{1, \ldots, 2|Q|+1\}$ and $\mu(q, k) = \psi_k(q)$ if $\rho(k) = (\psi_k, P_k)$.

Then μ is a progress measure of size m by conditions 1-4 defining the transition function $\overline{\delta}_m$:

By 1. and induction over $k \in \omega$,

$$(q, k) \in V \iff q \in dom(\psi_k) \text{ for } \rho(k) = (\psi_k, P_k)$$

and thus μ is well defined.

By 2. for $(p, p') \in E$:

$$\mu(p) \geq \mu(p'),$$

$$\mu(p) > \mu(p'), \text{ if } p' \in C \text{ and } \mu(p) \text{ is even.}$$

By 3. and 4., infinitely often an accepting state $(\psi_k, \emptyset)$ is reached, iff for the run $\overline{\rho} = (\psi_0, P_0)(\psi_1, P_1)(\psi_2, P_2)\ldots$ of $\overline{\mathcal{B}}_m$ there is no run $\rho = r_0 r_1 r_2 \ldots$ of $\mathcal{B}$ on α such that $r_i \in dom(\psi_i)$ for $i \in \omega$, and for some fixed $i_0, b \in \omega$ and every $i \geq i_0$: $\psi_i(r_i, i) = b$ and b is odd. This implies that, by the definition of μ there is no path $\pi = p_0 p_1 p_2 \ldots$ in $\mathcal{G}$ such that for some $i_0, b \in \omega$ and every $i \geq i_0$: $\mu(p_i) = b$ is odd.

"⇐" Assume there is a progress measure μ of size m for the run graph of the Büchi automaton $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$ and the word α. Then let $\rho = r_0 r_1 r_2 \ldots$ be the run for the Büchi automaton $\overline{\mathcal{B}}_m$ defined inductively by:

$$r_0 = q_I = (\{(q_i, 2m+1)\}, \emptyset)$$

$$r_{k+1} = (\psi_{k+1}, P_{k+1})$$

such that

$$\psi_{k+1}(q') = \begin{cases} \mu_m(q', k+1) & \text{if } \exists q \in dom(\psi_k) \wedge q' \in \delta(q, \alpha(k)) \\ undefined & \text{otherwise} \end{cases}$$

and p_{k+1} is defined according to conditions 3,4 of the definition of $\overline{\delta}_m$. Then all requirements for a valid run are met and ρ is accepting as we will see:

Assume it is not. Then there is some k_0 such that $P_k \neq \emptyset$ for $k \geq k_0$. But then there are $q_{k_0}, q_{k_0+1}, q_{k_0+2}, \ldots$ such that for all $k \geq k_0$, $q_{k+1} \in \delta(q_k, \alpha(k))$, $q_k \in P_k$, and $\psi_k(q_k) = \psi_{k+1}(q_{k+1})$ where $\psi_k(q_k)$ is odd. But then, for all $k \geq k_0$, $\mu(q_k, k) = \mu(q_{k+1}, k+1)$ is odd which is in contradiction to the assumption that μ is a progress measure.

□

3.3 Size Estimation

The automaton $\overline{\mathcal{B}}_{|Q|}$ for $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$ defined in Theorem 3.5 accepts a word α iff $\mathcal{B}$ does not, ie. $\mathcal{L}(\overline{\mathcal{B}}_{|Q|}) = \Sigma^\omega \setminus \mathcal{L}(\mathcal{B})$. But is the number of states of $\overline{\mathcal{B}}_{|Q|}$ really bounded by $2^{O(|Q|\, log|Q|)}$ as promised in the beginning of this section? Indeed it is:

The set of states in $\overline{\mathcal{B}}_{|Q|}$ is defined as $\Psi \times 2^Q$ where Ψ is the set of partial functions $\psi : Q \to \{1, \ldots, 2|Q| + 1\}$. The number of such functions is $|Q|^{O(|Q|)} = 2^{O(|Q| log|Q|)}$ and thus

$$\begin{aligned} |\Psi \times 2^Q| &= |\Psi| \cdot |2^Q| \\ &= 2^{O(|Q| log|Q|)} \cdot 2^{|Q|} \\ &= 2^{O(|Q| log|Q|)} \end{aligned}$$

4 The Detour over WAPA

As was suggested in the beginning of Section 2, the difficulty of complementing Büchi automata lies in the fact that one only has to prove the existence of an accepting path to decide the word problem for some Büchi automaton, but to decide that some word does not belong to the language of a Büchi automaton one has to prove that there is no accepting path.

This section will use another type of automaton, the Weak Alternating Parity Automaton (WAPA). Although conceptually they seem a bit less intuitive than Büchi automata at first glance, they have two advantages: first, there are translations of Büchi automata into WAPA and of WAPA into Büchi automata, and second the process of complementing them is rather simple. Together, these two properties yield another complementation construction for Büchi automata, and again the size of the complement Büchi automaton will be $2^{O(n\, log\, n)}$ after some tweaking.

4.1 Definitions

For the definition of WAPA, more precisely for the definition of their transition function, the set of *positive Boolean formulas* is needed:

Definition 4.1. The *set of positive Boolean formulas* $B^+(X)$ over some set X is defined as follows:

$X \subset B^+(X)$

$t,\ f \in B^+(X)$

$x, y \in B^+(X) \Rightarrow (x \wedge y), (x \vee y) \in B^+(X)$

A *model* for some $\theta \in B^+(X)$ is a subset M of X such that θ evaluates to *true* under the interpretation I, defined as

$$\forall p \in X : \ \ p^I = true \iff p \in M \vee p = t.$$

M is *minimal* if no proper subset of M is a model for θ.

The set $vars(\theta)$ is the set of elements of X occurring in θ.

With this we can introduce the type of automaton we will be using throughout this section:

Definition 4.2. For a *WAPA* $\mathcal{A} = (Q, \Sigma, \delta, q_i, c)$, Q, Σ, and q_i are defined as for Büchi automata, $\delta : Q \times \Sigma \to B^+(Q)$ is the transition function, and $c : Q \to \omega$ is called a *parity function*.

Before formally defining the semantics of transition and parity function, I will first explain the intuition behind them:

At any given time a Büchi automaton has exactly one state and goes from one into another as it reads a letter from the input word, choosing nondeterministically from a set of successor states given by the transition function. A WAPA, on the other hand, does not choose one successor state from a set, but one minimal model for the positive Boolean formula returned by its transition function for its current state and the input letter. A copy of the automaton then goes into each one of the states from the model and continues the computation.

Example. Let $\mathcal{A} = (Q, \Sigma, \delta, q_i, c)$ be a WAPA and $\alpha = ab\dots$ be a word in Σ^ω. Then $\mathcal{A}$ starts in state q_i and reads letter a. Suppose $\delta(q_i, a) = (q_1 \vee q_2)$. Then the minimal models for $\delta(q_i, a)$ are $\{q_1\}$ and $\{q_2\}$, so the WAPA has the nondeterministic choice of going into either of the two states. Suppose it goes into q_1 and $\delta(q_1, b) = (q_3 \wedge q_4)$. Then it has to make two copies of itself and send one copy into state q_3 and one into q_4 where both of the copies will independently continue consuming input letters.

A run of a WAPA on some word is therefore not represented by a sequence of states, but by a rooted, sliced graph, the *run dag*.

Definition 4.3. The *slices* S_i and edges of a *run dag* (V, E) for a WAPA $\mathcal{A} = (Q, \Sigma, \delta, q_i, c)$ and a word α are defined by means of the transition function:

$$S_0 = \{(q_i, 0)\}$$

Let $S_i = \{(q_0, i), (q_1, i), \dots, (q_n, i)\}$ and let for each (q, i) in S_i some minimal model $M_{q,i}$ for $\theta_{q,i} = \delta(q, \alpha(i))$ then

$$S_{i+1} = \bigcup_{(q,i) \in S_i} M_{q,i} \times \{i + 1\}$$

$$((q,i),(q',i+1)) \in E \iff q' \in M_{q,i}$$

An important detail here is that there is no model for f or any equivalent formula and hence there can be no run dag containing an edge $(f,(p,i))$ or a node $f \in V$. The only minimal model for t or any equivalent formula on the other hand is $\emptyset$, so the edge $((p,i),t)$ leads to the leaf t in the dag[4].

A run dag then is *accepting*, iff there is no infinite path π in the dag such that the minimum of the set $\{c(q) \mid (q,i) \in Occ(\pi)\}$ is odd. A WAPA $\mathcal{A}$ accepts a word α, iff there is an accepting run dag for $\mathcal{A}$ and α.

4.2 The Construction

Due to its length, the proof for the following lemma will not be given here. It is carried out in [2, 40pp.].

Lemma 4.4. *Let the* dual $\overline{\theta}$ *of a formula* $\theta \in B^+(X)$ *be the formula obtained by exchanging* t *and* f *as well as* $\vee$ *and* $\wedge$*. The* dual automaton $\overline{\mathcal{A}}$ *for some WAPA* $\mathcal{A} = (Q, \Sigma, \delta, q_i, c)$ *is then defined as* $\overline{\mathcal{A}} = (Q, \Sigma, \overline{\delta}, q_i, \overline{c})$ *where* $\overline{\delta}(q,a) := \overline{\delta(q,a)}$ *for all* $q \in Q, a \in \Sigma$ *and* $\overline{c}(q) := c(q) + 1$ *for all* $q \in Q$*. Then, for any given WAPA* $\mathcal{A}$*,* $\overline{\mathcal{A}}$ *accepts the complement language, ie.* $\mathcal{L}(\overline{\mathcal{A}}) = \Sigma^\omega \setminus \mathcal{L}(\mathcal{A})$*.*

The first step in this construction for complementation of Büchi automata is finding a translation from Büchi automata to WAPA. In other words, we have to find a WAPA for every Büchi automaton that accepts precisely the same language. The following lemma shows one such construction:

Lemma 4.5. *Let* $\mathcal{B} = (Q, \Sigma, \delta, q_i, F)$ *be a Büchi automaton. Then* $\mathcal{B}$ *and the WAPA* $\mathcal{A} = (Q', \Sigma, \delta', q_i', c)$ *accept the same language over* Σ *if*

- $Q' = Q \times \{0, \ldots, 2|Q|\}$
- $q_i' = (q_i, 2|Q|)$
- $c(q', k) = k$
- $\delta'((p,k),a) = \bigvee_{q \in \delta(p,a)} (\Delta(q,k))$
- $\Delta(q,k) = \begin{cases} \bigwedge_{0 \le j \le k} (q,j) & \textit{if } k \textit{ is even or } q \in F \\ \bigwedge_{0 \le j < k} (q,j) & \textit{otherwise} \end{cases}$

Proof. In two steps:

$\mathcal{L}(\mathcal{B}) \subseteq \mathcal{L}(\mathcal{A})$: Assume there is some $\alpha \in \mathcal{L}(\mathcal{B})$. Then there is an accepting run $\rho = r_0 r_1 r_2 \ldots$ for $\mathcal{B}$ on α. From this run, the slices of a run dag $\mathcal{D} = (V, E)$ for $\mathcal{A}$ on α can be constructed as follows:

$$\begin{aligned} S_0 &= \{(q_i, 2|Q|)\} \times \{0\} \\ S_k &= \{(r_k, l) \mid l \le 2|Q|\} \times \{k\},\ k \ge 1 \end{aligned}$$

and the edges are defined according to the transition function. Then every vertex $((r_k, 2j+1), i)$ eventually reaches a vertex $((r_l, 2j), m)$, because ρ contains infinitely many states in F and thus the run graph is accepting.

$\mathcal{L}(\mathcal{B}) \supseteq \mathcal{L}(\mathcal{A})$: Consider the dual WAPA $\overline{\mathcal{A}} = (Q', \Sigma, \overline{\delta}', q_i', \overline{c}')$. Q', Σ and q_i' are defined as for $\mathcal{A}$ and

- $\overline{c}(q,i) = i + 1$.
- $\overline{\delta}'((p,i),a) = \bigwedge_{q\in\delta(p,a)}(\overline{\Delta}(q,i))$
- $\overline{\Delta}(q,i)) = \begin{cases} \bigvee_{0\leq j\leq i}(q,j) & \text{if } i \text{ is even or } q \in F \\ \bigvee_{0\leq j< i}(q,j) & \text{otherwise} \end{cases}$

We show that, for $\alpha \notin \mathcal{L}(\mathcal{B})$, there is an accepting run dag $\mathcal{D} = (V, E)$ for $\overline{\mathcal{A}}$ and α. Since $\mathcal{L}(\mathcal{A}) = \Sigma^\omega \setminus \mathcal{L}(\overline{\mathcal{A}})$, this implies that $\alpha \notin \mathcal{L}(\mathcal{A})$

Let $\alpha \notin \mathcal{L}(\mathcal{B})$.

Case 1: Assume there is no infinite run of $\mathcal{B}$ on α. Then there is no infinite path in the run dag $\mathcal{D}$ of $\overline{\mathcal{A}}$. However, for no $((p,i),a),(p,i) \in Q, a \in \Sigma$, $\overline{\delta}'((p,i),a) = \theta$ and θ is equivalent to f. Hence a run dag for $\overline{\mathcal{A}}$ and α exists, is finite and therefore accepting.

Case 2: There is an infinite, non-accepting run of $\mathcal{B}$ on α. Note that intuitively this WAPA simulates all runs of the Büchi automaton ensuring that, along every path $\pi = p_0p_1p_2\ldots$ in the run dag $\mathcal{D}$ of $\overline{\mathcal{A}}$ on α, $(c'(p_i))$ is a monotonically decreasing sequence. The WAPA is free to choose exactly how this sequence decreases along every path as long as it obeys the rule that it strictly decreases on occurrence of a vertex $((q,k),i)$, where $q \notin F \wedge k$ is odd. Since $\alpha \notin \mathcal{L}(\mathcal{B})$, there is a progress measure μ for $\mathcal{B}$ and α. If it chooses $k = \mu(q) + 1$ for every $((q,k),i) \in V$, then the resulting run dag is accepting by the properties of progress measures and the definition of c'.

□

Observation 4.6. *The WAPA $\mathcal{A} = (Q, \Sigma, \delta, q_i, c)$ and its complement resulting from the construction in Lemma 4.5 are* stratified*, ie. $c(q) \leq c(p)$, for all $p \in Q$ and $q \in vars(\delta(p,a))$, respectively, by the definition of Δ, and likewise for $\overline{\mathcal{A}}$.*

Having shown how to get from a Büchi automaton accepting $\mathcal{L}(\mathcal{B})$ to a WAPA accepting the complement language, next and last on our roadmap is to find a translation from the WAPA obtained so far back to a Büchi automaton. The construction will take advantage of the fact that the WAPA is stratified and it will lead to a Büchi automaton which simulates the run dag for the WAPA and the input word. To this end, it will, for each input letter it reads, guess the run dag's next slice and represent it in its state along with information about those paths in the guessed dag whose last vertices have an odd parity and need to encounter one with an even parity in order for the simulated run dag to be accepting.

Lemma 4.7. *Let* $\mathcal{A} = (Q, \Sigma, \delta, q_i, c)$ *be a stratified WAPA and*

$$\mathcal{B} = (2^Q \times 2^Q, \Sigma, \delta', (\{q_i\}, \emptyset), 2^Q \times \emptyset)$$

be a Büchi automaton where

$$\delta((S, O), a) := \{(S', O' \setminus E)\}$$

for

- $a \in \Sigma$,
- $O \subseteq S \subseteq Q$,
- E *the set of states in* Q *with an even parity,*
- $S' \in Mod(\bigwedge_{q \in S} \delta(q, a))$
- $\begin{array}{ll} O' \subseteq S', O' \in Mod(\bigwedge_{q \in O} \delta(q, a)) & \textit{if } O \neq \emptyset \\ O' = S' & \textit{else} \end{array}$

then $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$

Proof Sketch: First show that, from a run dag $\mathcal{D}$ of $\mathcal{A}$, an infinite run $\rho = r_0 r_1 r_2 \dots$ of $\mathcal{B}$ can be constructed and vice-versa. Then show that ρ is accepting iff $\mathcal{D}$ is: there are infinitely many states $r_i = (S, \emptyset)$ in ρ iff, infinitely often, every path in $\mathcal{D}$ goes through a state with an even parity. The latter, finally, means that the run dag is accepting, because $\mathcal{A}$ is stratified, $img(c)$ is bounded from below, and going through infinitely many states with an even parity means for a path $\pi = p_0 p_1 p_2 \dots$ that $Min(\{c(p) \mid p \in Occ(\pi)\})$ is even. □

4.3 Size Estimation

In this section, a translation was presented which yields, for a Büchi automaton $\mathcal{B}$ with n states, a Weak Alternating Parity Automaton with $O(n)$ states accepting the same language. Complementing this WAPA involves no blow-up at all, but the result of the translation from the complemented WAPA back to an equivalent Büchi automaton, the end product of the complementation process, does:

Generally, for a WAPA with n states, the Büchi automaton $\overline{\mathcal{B}}$ obtained by the construction in Lemma 4.7 has $2^{O(n^2)}$ states. However, this can be improved in our special case: Note that $Q = P \times \{0, \dots, 2|P|\}$ where P is the set of states of the original Büchi automaton $\mathcal{B}$. But for no state $r_i = (O, S)$ in any run $\rho = r_0 r_1 r_2 \dots$ of $\overline{\mathcal{B}}$ will the sets S and O ever contain pairs $(p, k), (p', l)$ where k and l are different. Thus the state space $2^Q \times 2^Q$ of $\overline{\mathcal{B}}$ contains many states that never occur in the transition relation. Removing these states leaves us with a set of states of size $2^{O(n\, log\, n)}$.

References

[1] J.R. Büchi. On a decision method in restricted second order arithmetic. In *Proc. International Congress on Logic, Method, and Philosophy of Science*, pages 1–12. Stanford University Press, New York, NY, USA, 1960.

[2] Felix Klaedtke. Complementation of büchi automata using alternation. In *Automata logics, and infinite games: a guide to current research*, pages 61–77. Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[3] Nils Klarlund. Progress measures for complementation of ω-automata with applications to temporal logic. In *Proceedings of the 32nd annual symposium on Foundations of computer science*, pages 358–367, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[4] Orna Kupferman and Moshe Y. Vardi. Weak alternating automata are not that weak. In *ACM Trans. Comput. Logic*, volume 2, pages 408–429. ACM, New York, NY, USA, 2001.

[5] M. Michel. Complementation is more difficult with automata on infinite words. 1988.

[6] S. Safra. On the complexity of ω-automata. In *Proc.* 29^{th} *Symposium on Foundations of Computer Science*, page 319. White Plains, 1988.

[7] Moshe Y. Vardi. Büchi complementation: A forty-year saga.