

Optimal Transformations of Muller Conditions

Antonio Casares

Journées annuelles GT ALGA 2021

Joint work with Thomas Colcombet (IRIF) and Nathanaël Fijalkow (LaBRI).

To appear at ICALP 2021

We are interested in non-terminating reactive systems:

- Deterministic ω -automata.
- Nondeterministic ω -automata.
- Infinite duration two-players games with ω -regular conditions.

} Transition systems

Question

Transition system using
a complex acceptance
condition



Transition system using a
simpler acceptance
condition

We are interested in non-terminating reactive systems:

- **Deterministic ω -automata.** (This talk)
- Nondeterministic ω -automata.
- Infinite duration two-players games with ω -regular conditions.

} Transition systems
(In the paper)

Question

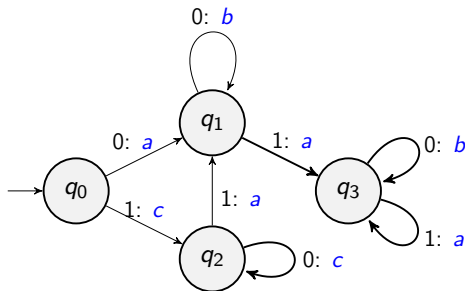
Transition system using
a complex acceptance
condition



Transition system using a
simpler acceptance
condition

Deterministic automata over infinite words

- Input alphabet $\Sigma (= \{0, 1\})$.
- Set of states Q (one of them initial).
- Transition function $\delta : Q \times \Sigma \rightarrow Q \times \Gamma$.
- Output alphabet $\Gamma (= \{a, b, c\})$.
- **Acceptance condition** $Acc \subseteq \Gamma^\omega$.



Input = 10101000010... $\in \Sigma^\omega$

Output = cababbbbab... $\in \Gamma^\omega$

Output $\in Acc$?

$\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^\omega : \text{the (unique) run produced by reading } w \text{ is accepted}\}.$

We are interested in conditions that specify the “asymptotic behaviour” of the system.

Given a word $u \in \Gamma^\omega$ we write:

$$\text{Inf}(u) = \{\alpha \in \Gamma : \alpha \text{ appears infinitely often in } u\}.$$

Let Γ be a finite alphabet.

A **Muller condition** is given by a family of sets

$$\mathcal{F} \subseteq 2^\Gamma.$$

The words accepted by this condition are:

$$\text{Acc}_{\mathcal{F}} = \{u \in \Gamma^\omega : \text{Inf}(u) \in \mathcal{F}\}.$$

Let $\Gamma \subseteq \mathbb{N}$.

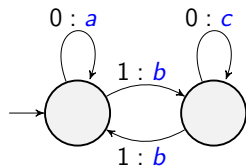
The **parity condition** is:

$$Acc_{Parity} = \{u \in \Gamma^\omega : \min Inf(u) \text{ is even} \}.$$

We call the elements of Γ **priorities**.

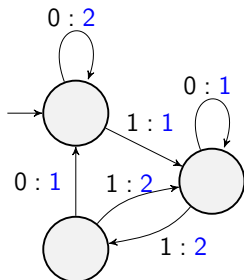
Remark: We can reformulate parity conditions as Muller ones.

Example



Deterministic Muller automaton.

$$\mathcal{F}_1 = \{\{a\}, \{b\}\}.$$



Deterministic parity automaton.

$L = \{w \in \{0, 1\}^\omega : \text{eventually } w \text{ only contains } 1, \\ \text{or it only contains } 0 \text{ after an even number of occurrences of } 1\}.$

- ▶ Muller conditions are more general than parity ones.
- ▶ Parity conditions are easier to deal with:
 - Quasi-polynomial time algorithms for solving parity games.
 - Parity games are positionally determined.
- ▶ Every ω -regular language can be recognized by a deterministic parity automaton, and using a parity condition is “the best we can do”.
(Strict hierarchy depending on the *number of priorities*).

Objective

Given a Muller automaton, transform it into a parity automaton

- as small as possible.
- using the least possible number of priorities.

The classical approach: product by an automaton

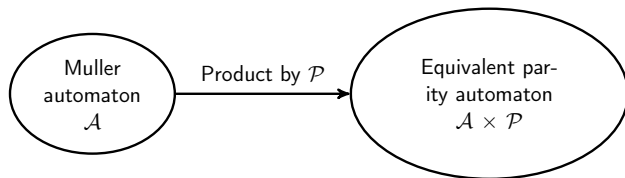
Let $\mathcal{F} \subseteq 2^\Gamma$ define a Muller condition. We can find a deterministic parity automaton \mathcal{P} *recognising the condition*, i.e., for $u \in \Gamma^\omega$:

$$u \in \mathcal{L}(\mathcal{P}) \Leftrightarrow \text{Inf}(u) \in \mathcal{F}.$$

Then, given a Muller automaton \mathcal{A} using condition \mathcal{F} , we can build

$$\mathcal{A} \times \mathcal{P},$$

a parity automaton recognizing $\mathcal{L}(\mathcal{A})$ and using the condition from \mathcal{P} .



The classical approach: product by an automaton

This approach is not optimal (even if we have a minimal automaton \mathcal{P}).

- **We do not take into account the structure of the input automaton \mathcal{A} .**

Intuitively, this is due to:

- \mathcal{A} might not use the full power of the considered Muller condition.
- Different parts of \mathcal{A} make use of the condition in different ways.

Alternating Cycle Decomposition (ACD)

A data structure (a tree) that captures the interplay between the structure of a Muller automaton and its acceptance condition.

→ We use it to define an “*optimal*” transformation of a Muller automaton into a parity one.

It is notably inspired by:

- The Zielonka tree of a Muller condition (Zielonka 98') and (Dziembowski, Jurdziński, Walukiewicz 97').
- The study of the alternating chains of accepting and rejecting cycles (Wagner 79').

But... What is an **optimal transformation**?

Definition (Morphism of automata)

Let \mathcal{A} and \mathcal{B} be two automata over an input alphabet Σ . A *morphism* is a mapping of states

$$\varphi: \mathcal{A} \rightarrow \mathcal{B}$$

that verifies two local conditions:

- 1 Preserves the initial state.
- 2 Preserves transitions ($\delta_{\mathcal{A}}(q, x) = q' \Rightarrow \delta_{\mathcal{B}}(\varphi(q), x) = \varphi(q')$).

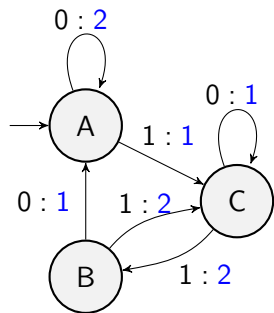
and the following global condition:

- 3 A run ϱ in \mathcal{A} is accepting if and only if $\varphi(\varrho)$ is accepting.

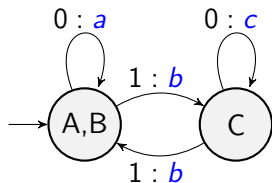
Proposition

If there exists a morphism $\varphi: \mathcal{A} \rightarrow \mathcal{B}$, then $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$.

Example of morphism



Parity automaton.



$$\mathcal{F}_1 = \{\{a\}, \{b\}\}.$$

Optimality of the ACD-transformation

Let \mathcal{A} be a Muller automaton and $\mathcal{P}_{ACD(\mathcal{A})}$ be the parity automaton obtained using the ACD.

Proposition (Correctness)

There is a morphism $\varphi: \mathcal{P}_{ACD(\mathcal{A})} \rightarrow \mathcal{A}$.

Theorem (Optimality)

Let \mathcal{P}' be a parity automaton such that there is a morphism of automata $\varphi: \mathcal{P}' \rightarrow \mathcal{A}$. Then:

- 1 $|\mathcal{P}_{ACD(\mathcal{A})}| \leq |\mathcal{P}'|$.
- 2 \mathcal{P}' uses at least as many priorities as $\mathcal{P}_{ACD(\mathcal{A})}$.

Remark (Locally bijective morphisms as witnesses of transformations)

*In order to state the correctness and optimality for nondeterministic automata and games we have to introduce the notion of **locally bijective morphism** (similar to bisimulation).*

→ Locally bijective morphisms preserve all the desired semantic properties.

The ACD allows us to prove the following result:

Theorem (Relabelling with parity conditions)

Let \mathcal{A} be a Muller automaton. The following are equivalent:

- We can define a parity condition on top of \mathcal{A} obtaining an equivalent automaton.*
- For any pair of loops ℓ_1, ℓ_2 with a common state, if both are accepting or rejecting, so is their union.*

Application: Relabelling automata with simpler conditions

The ACD allows us to prove the following results:

Theorem (Relabeling with Rabin conditions)

Let \mathcal{A} be a Muller automaton. The following are equivalent:

- We can define a Rabin condition on top of \mathcal{A} obtaining an equivalent automaton.*
- For any pair of loops ℓ_1, ℓ_2 with a common state, if both are rejecting, so is their union.*

Corollary (Boker, Kupferman, Steinitz 10')

A deterministic Muller automaton \mathcal{A} can be labelled with a parity condition if and only if it can be labeled with both Rabin and Streett conditions.

Corollary (C. 21')

Arena-independent memories for games using a given Muller condition coincide with Rabin automata for this condition.

Application: Determinisation of Büchi automata

→ The determinisation of Büchi automata is the bottleneck in LTL synthesis.

Theorem (Piterman 06')

Given a non-deterministic Büchi automaton \mathcal{B} , there is a construction that produces a deterministic parity automaton $\mathcal{P}_{\mathcal{B}}$ such that

$$\mathcal{L}(\mathcal{P}_{\mathcal{B}}) = \mathcal{L}(\mathcal{B}).$$

We can see this construction in two steps (Schewe 09'):

$$\mathcal{B} \longrightarrow \mathcal{M}_{\mathcal{B}} \longrightarrow \mathcal{P}_{\mathcal{B}}$$

where $\mathcal{M}_{\mathcal{B}}$ is a Muller automaton.

We can substitute the second step by the ACD-transformation:

- In many cases we obtain a strictly smaller automaton.
- We do not improve the worst case (already optimal).

- The ACD provides an optimal transformation of Muller automata into parity automata.
- The ACD captures the structure of automata and games and the interplay with the acceptance condition. This can be used to prove theoretical results about these transition systems. (See latest results about the chromatic memory requirements for Muller games on Arxiv).

Thank you!