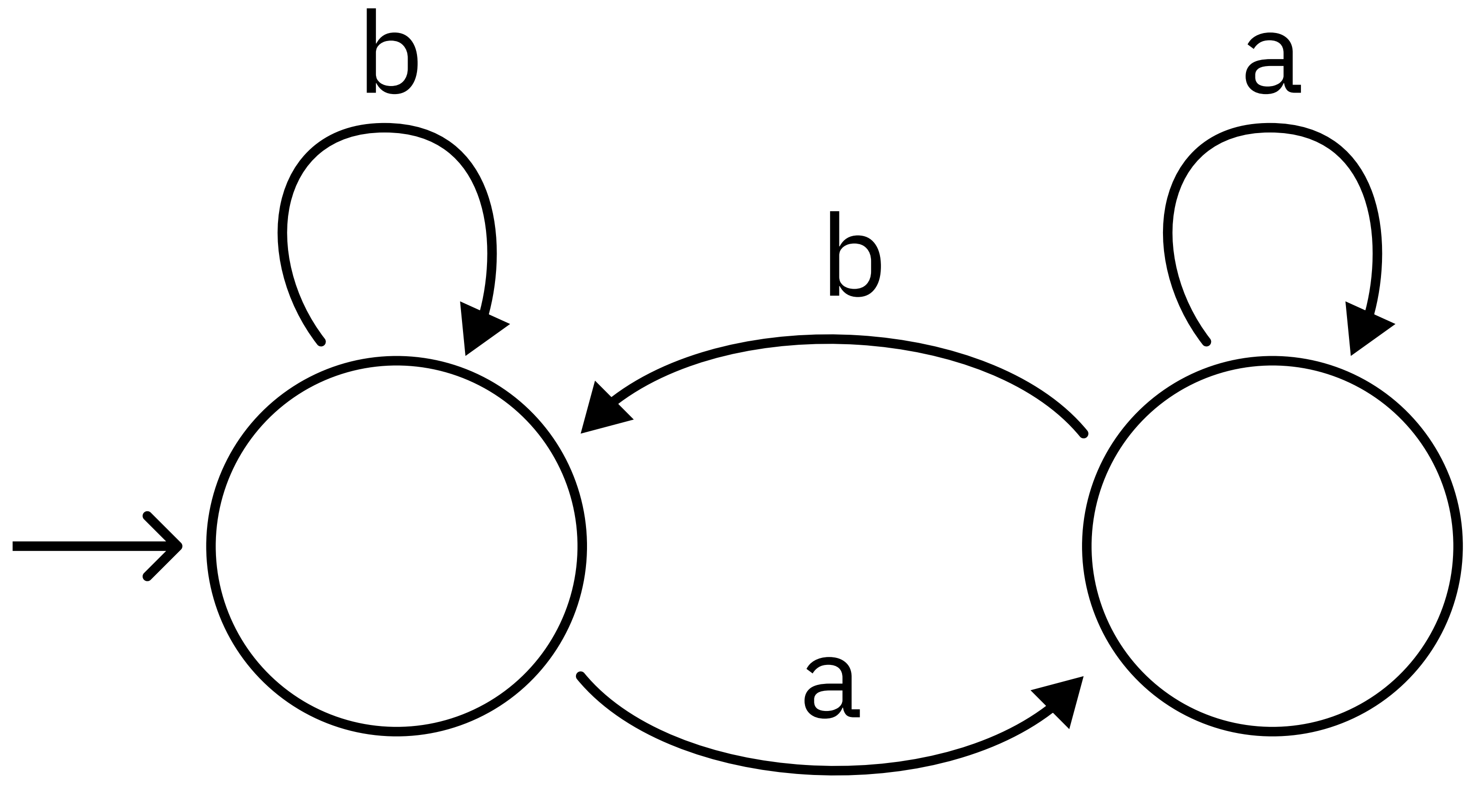
Simplifying ω-Automata through the Alternating Cycle Decomposition

Antonio Casares · University of Warsaw & Corto Mascle · LaBRI, University of Bordeaux

Automata over infinite words (ω-automata)

Automata over infinite words (ω-automata)

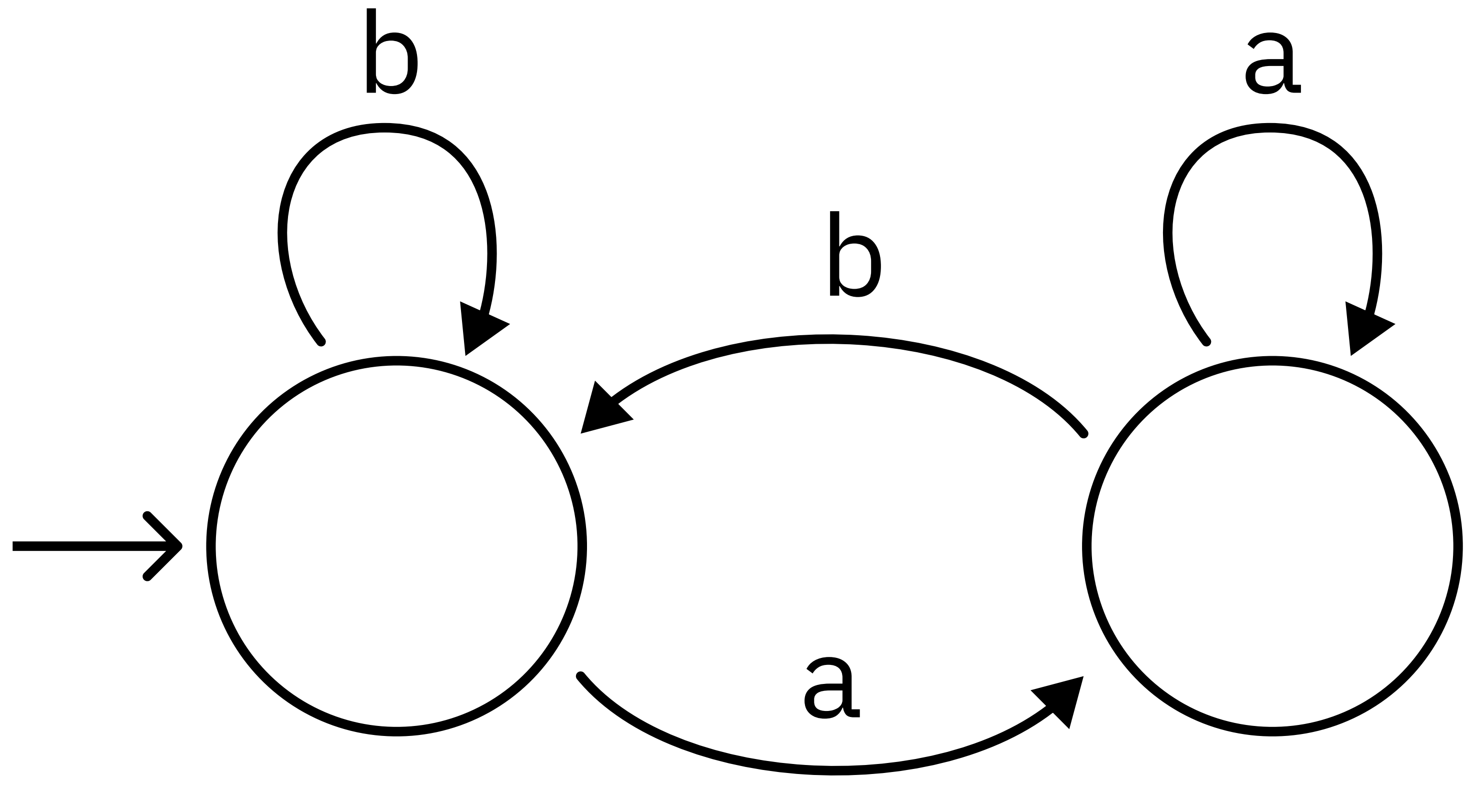
$\mathcal{A}$



Automata over infinite words (ω-automata)

Automata are deterministic
in this talk

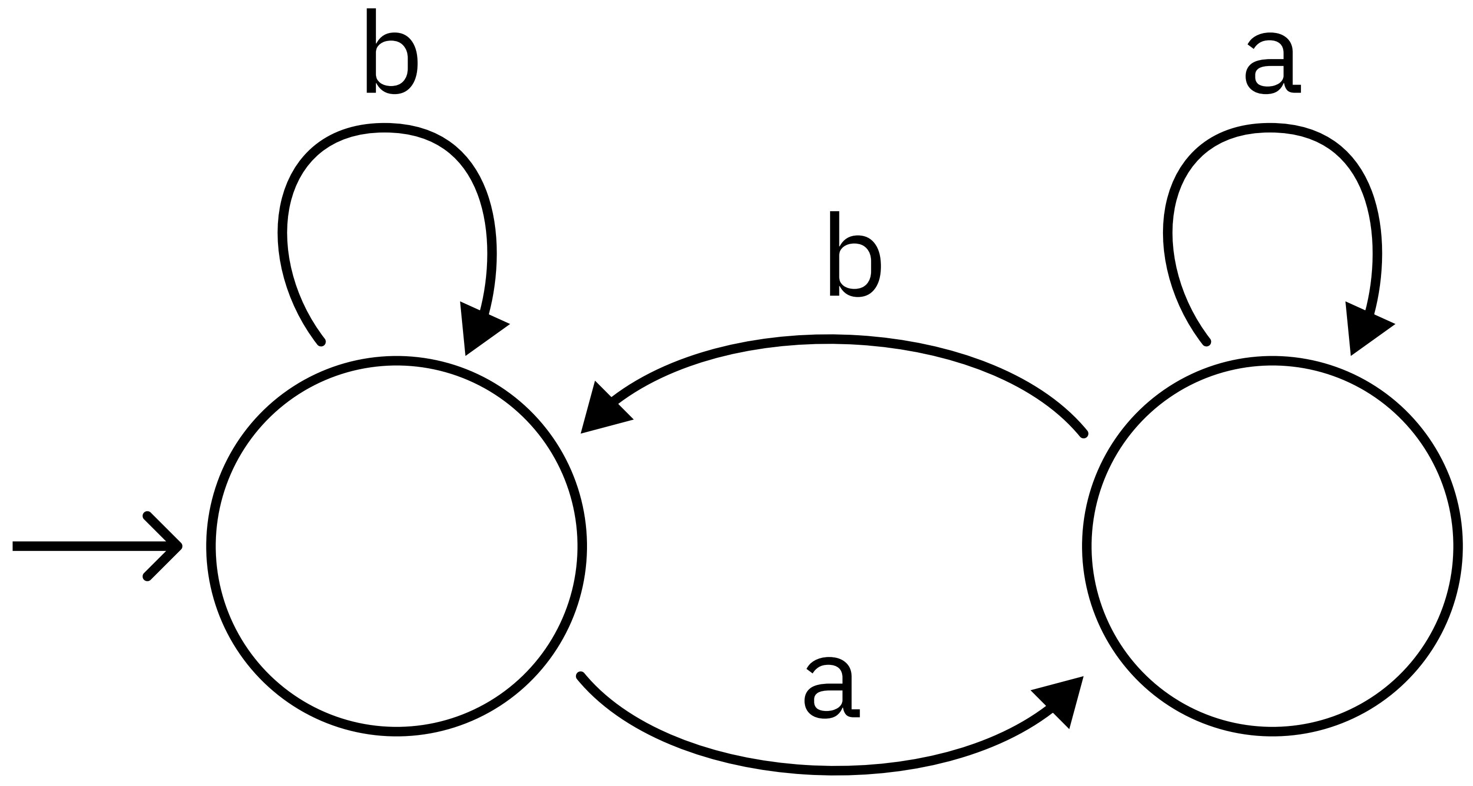
$\mathcal{A}$



Automata over infinite words (ω-automata)

Automata are deterministic in this talk

$\mathcal{A}$

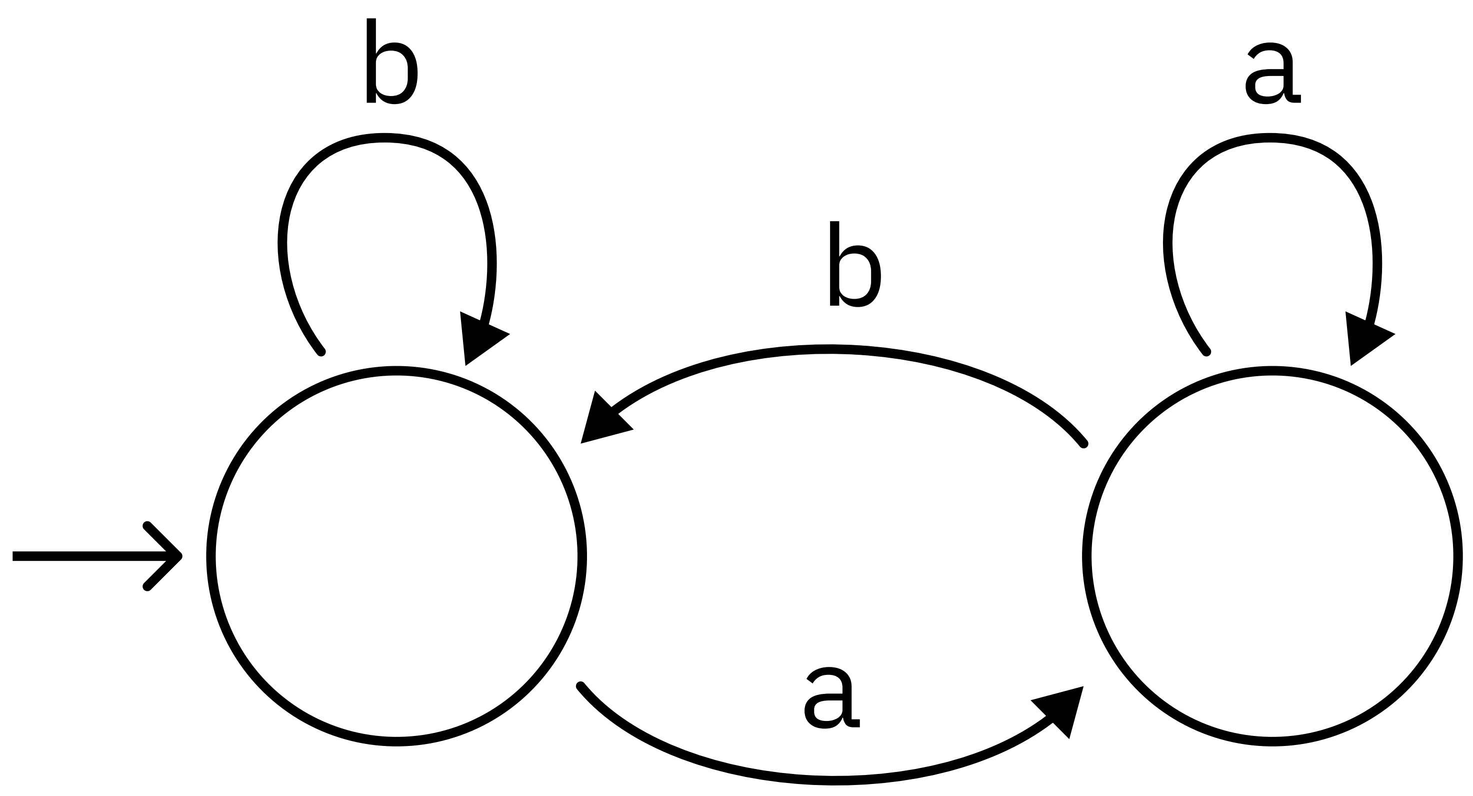


Input $w \in \Sigma^\omega$ $\longrightarrow$ Infinite path ρ

Automata over infinite words (ω-automata)

Automata are deterministic in this talk

$\mathcal{A}$



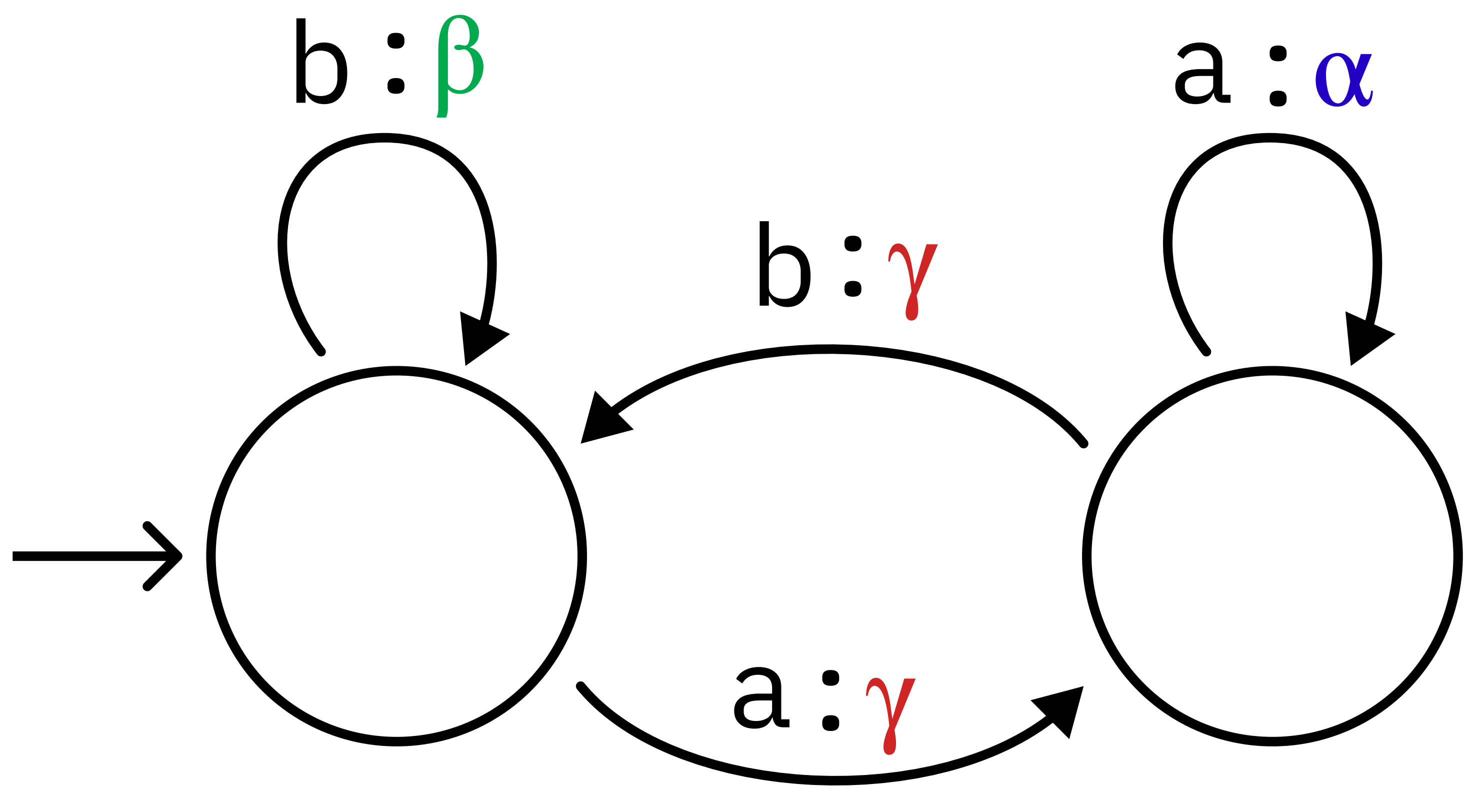
Acceptance Condition?

Input $w \in \Sigma^\omega$ $\longrightarrow$ Infinite path ρ

Automata over infinite words (ω-automata)

Automata are deterministic
in this talk

$\mathcal{A}$



Acceptance Condition?

$C = \{\alpha, \beta, \gamma\}$

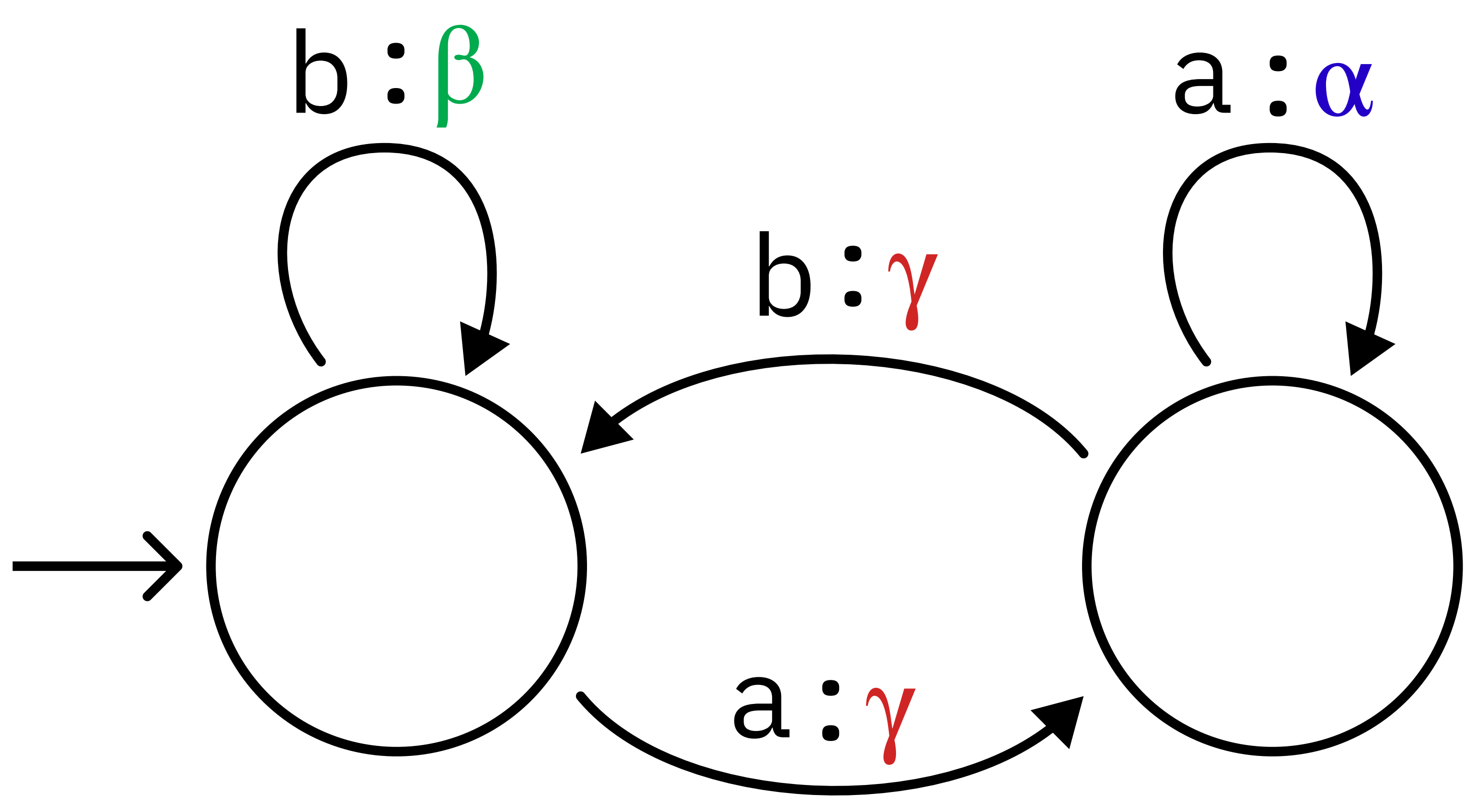
output colours

Input $w \in \Sigma^\omega$ $\longrightarrow$ Infinite path ρ

Automata over infinite words (ω-automata)

Automata are deterministic in this talk

$\mathcal{A}$



Acceptance Condition?

$C = \{\alpha, \beta, \gamma\}$

output colours

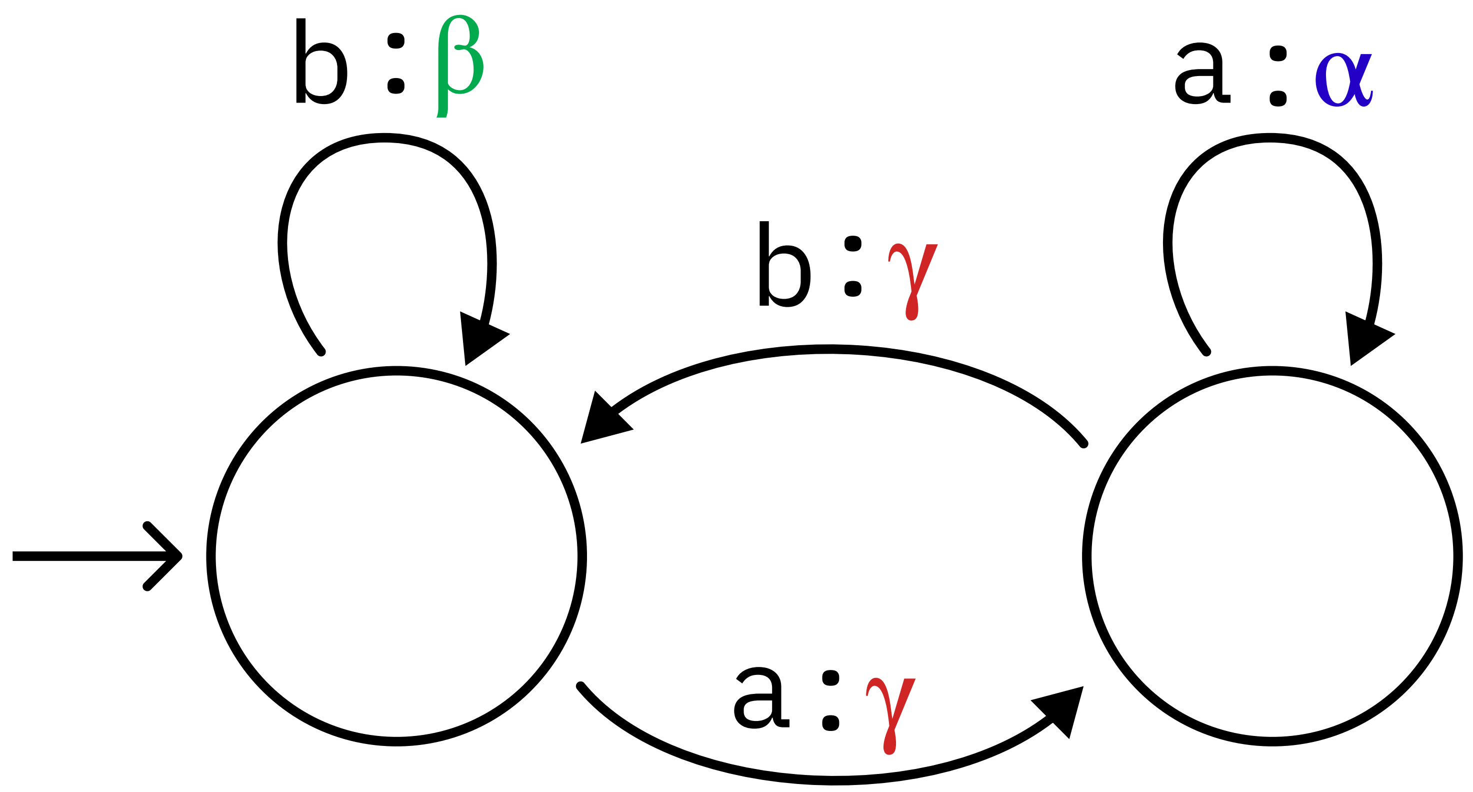
Input $w \in \Sigma^{\omega}$ $\longrightarrow$ Infinite path ρ

$\mathrm{Inf}(\rho) = \{$Colours appearing infinitely often in $\rho\}$

Automata over infinite words (ω-automata)

Automata are deterministic in this talk

$\mathcal{A}$



Input $w \in \Sigma^\omega$ $\longrightarrow$ Infinite path ρ

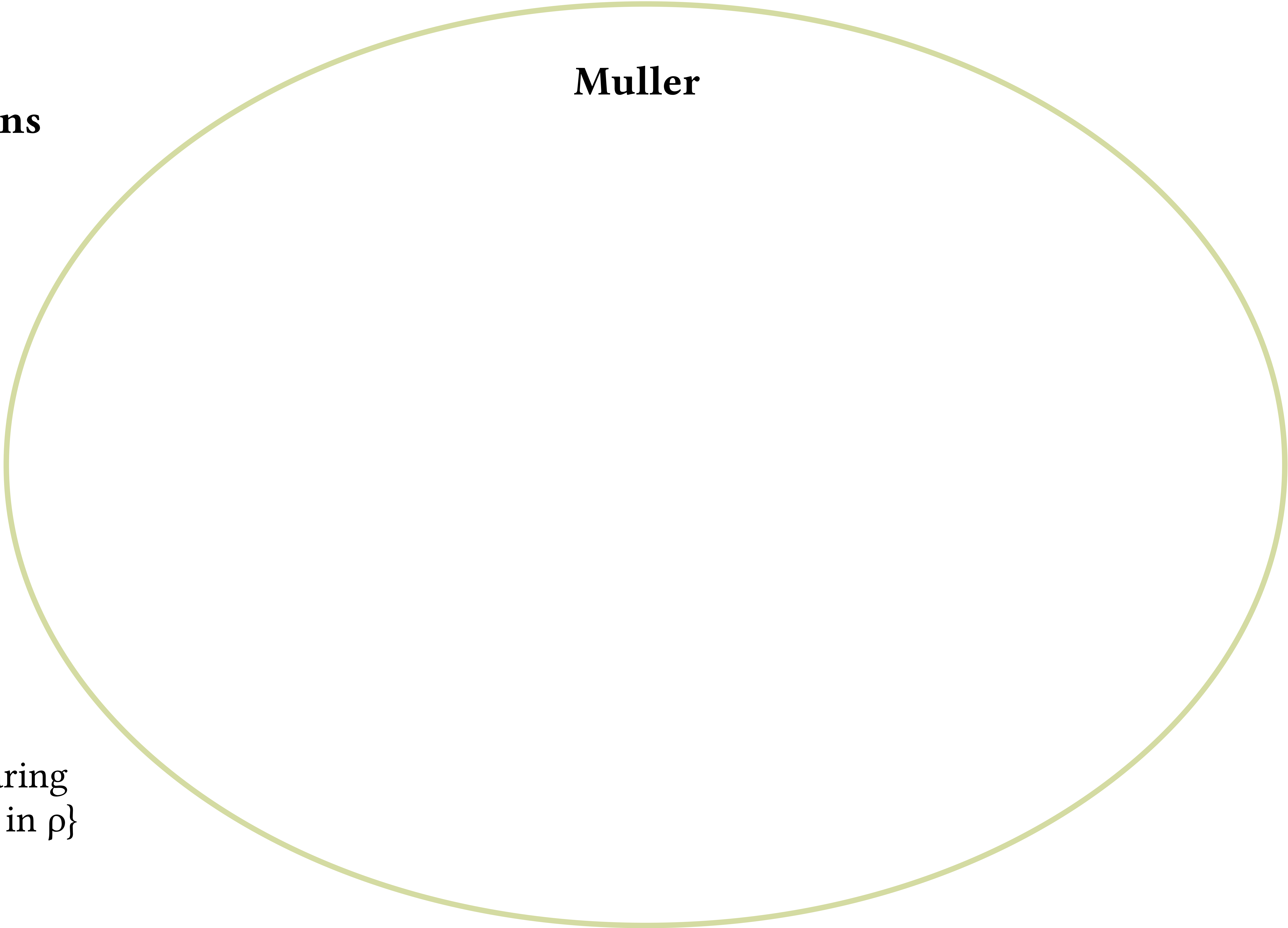
Different Acceptance Conditions

$C = \{\alpha, \beta, \gamma\}$

output colours

$\mathrm{Inf}(\rho) = \{$Colours appearing infinitely often in $\rho\}$

Different Acceptance Conditions



$C = \{ {\color{blue}\alpha}, {\color{green}\beta}, {\color{red}\gamma} \}$

output colours

$\mathrm{Inf}(\rho) =$ {Colours appearing infinitely often in ρ}

Muller

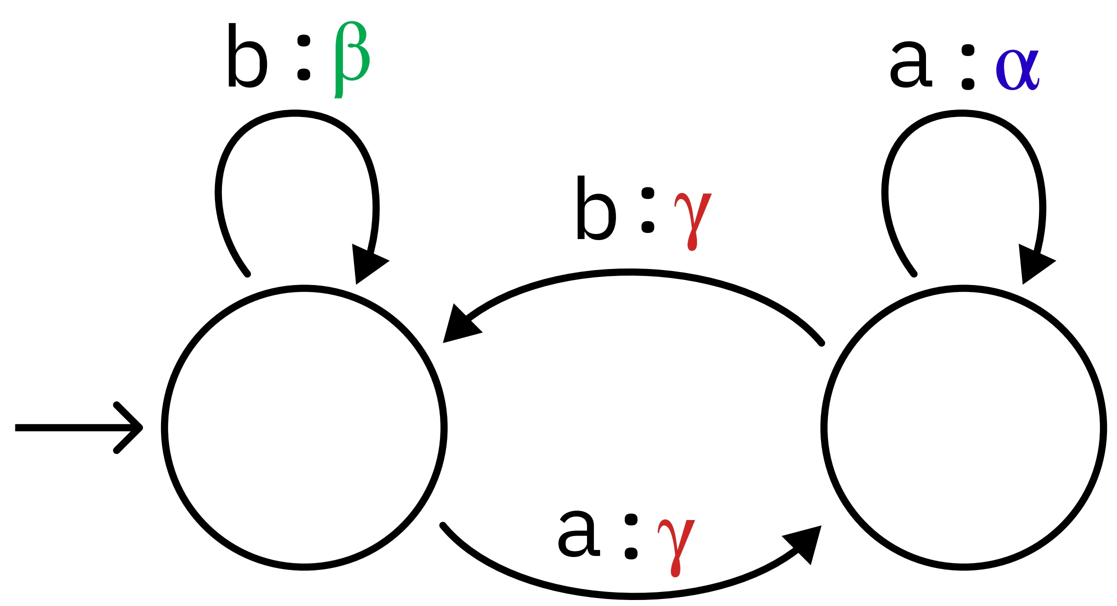
$\mathcal{F} \subseteq 2^{C}$ family of subsets

A run ρ is accepting $\Leftrightarrow$ $\mathrm{Inf}(\rho) \in \mathcal{F}$

Muller

$\mathcal{F} \subseteq 2^C$ family of subsets

A run ρ is accepting $\Leftrightarrow$ $\mathrm{Inf}(\rho) \in \mathcal{F}$

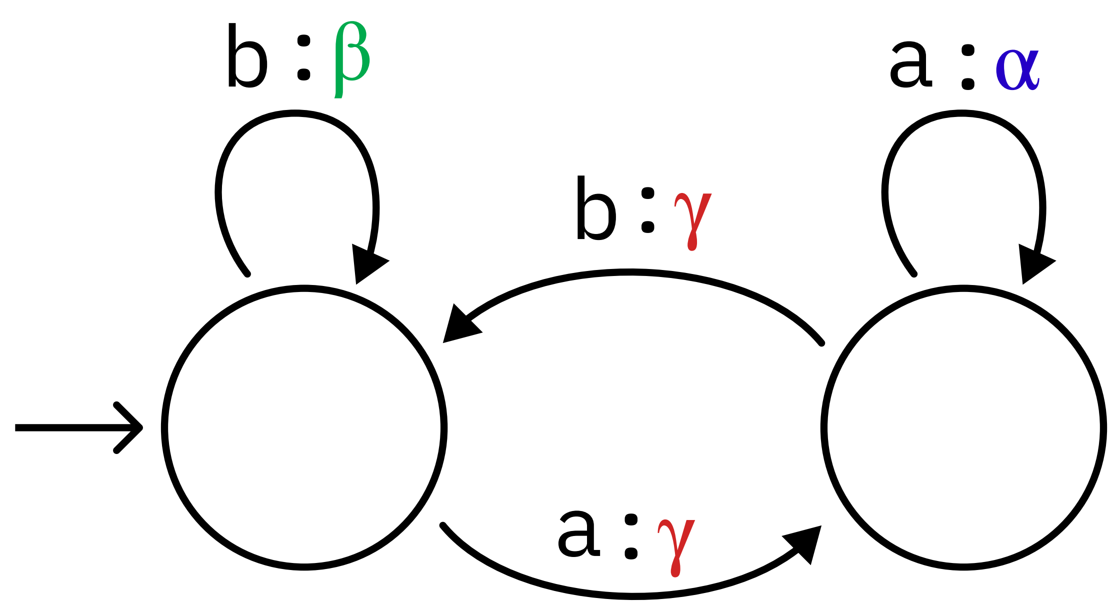


$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

Muller

$\mathcal{F} \subseteq 2^{C}$ family of subsets

A run ρ is accepting $\Leftrightarrow$ $\mathrm{Inf}(\rho) \in \mathcal{F}$



$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

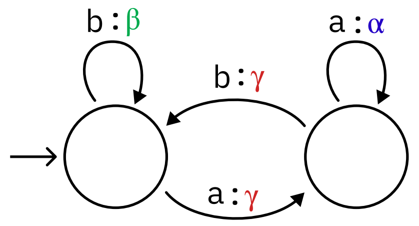
$$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$$

Different Acceptance Conditions

Muller

$\mathcal{F} \subseteq 2^C$ family of subsets

A run ρ is accepting $\Leftrightarrow$ $\mathrm{Inf}(\rho) \in \mathcal{F}$



$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

$$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$$

$C = \{\alpha, \beta, \gamma\}$

output colours

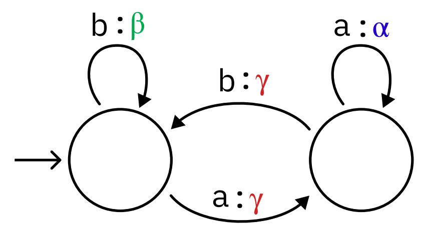
$\mathrm{Inf}(\rho) = \{$Colours appearing infinitely often in $\rho\}$

Different Acceptance Conditions

Muller

$\mathcal{F} \subseteq 2^{C}$ family of subsets

A run ρ is accepting $\Leftrightarrow$ $\mathrm{Inf}(\rho) \in \mathcal{F}$



$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$

Rabin

$C = \{\alpha, \beta, \gamma\}$

output colours

$\mathrm{Inf}(\rho) = \{$Colours appearing infinitely often in $\rho\}$

Rabin

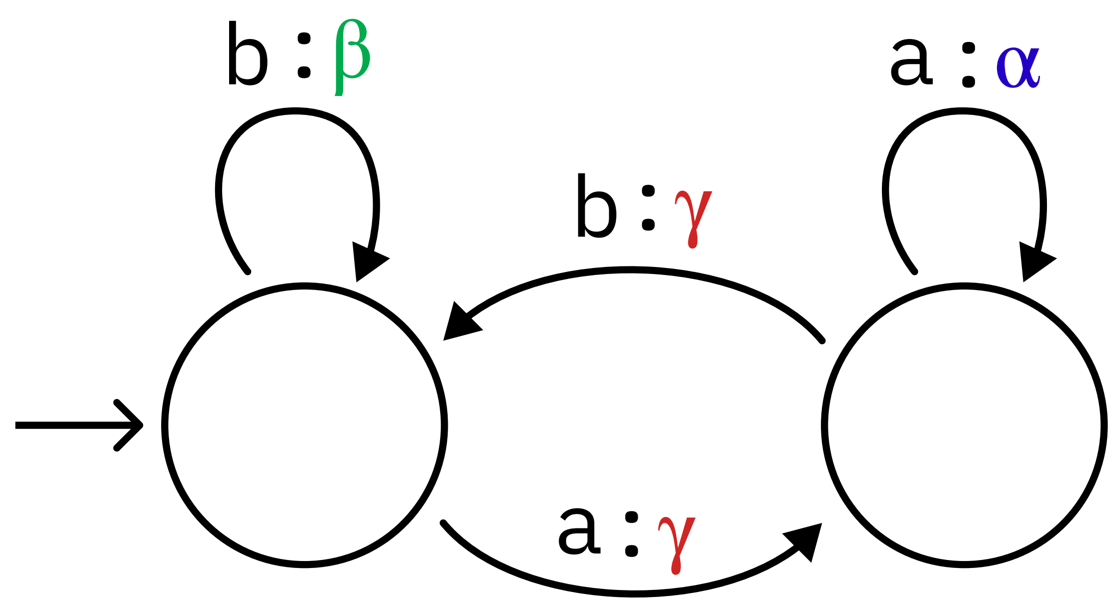
k Rabin pairs: $(G_1,R_1), \ldots, (G_k,R_k)$, $\quad G_i, R_i \subseteq C$

A run ρ is accepting $\Leftrightarrow$ $\bigvee \mathrm{Inf}(\rho) \cap G_i \neq \emptyset \wedge \mathrm{Inf}(\rho) \cap R_i = \emptyset$

Rabin

k Rabin pairs: $(G_1,R_1), \dots , (G_k,R_k)$, $G_i, R_i \subseteq C$

A run ρ is accepting $\Leftrightarrow$ $\bigvee \mathrm{Inf}(\rho) \cap G_i \neq \emptyset \wedge \mathrm{Inf}(\rho) \cap R_i = \emptyset$



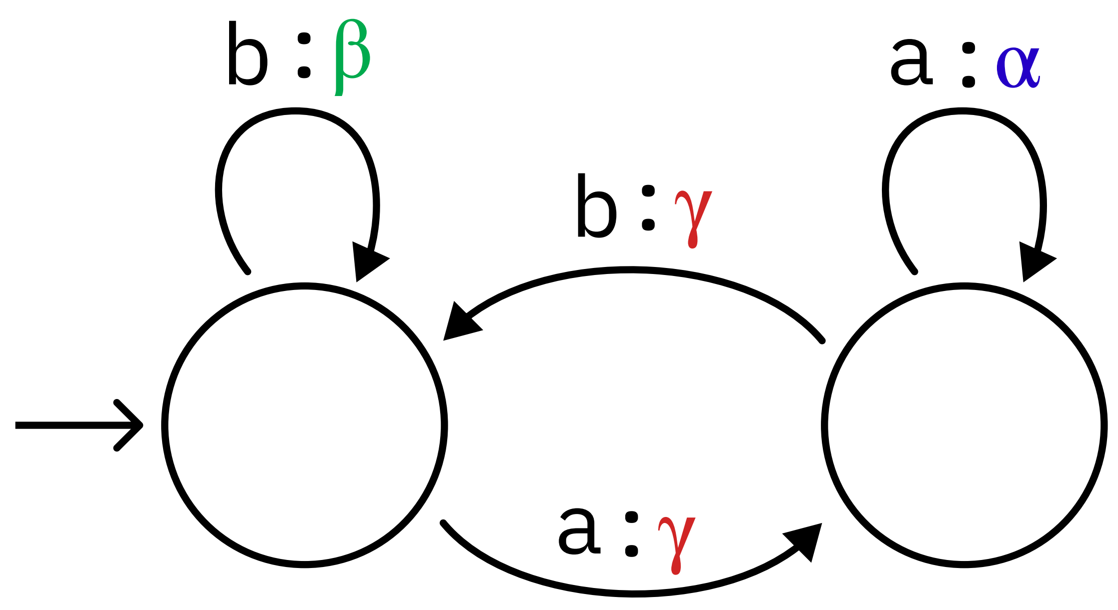
$G_1 = \{\alpha\}, R_1 = \{\gamma\}$

$G_2 = \{\beta\}, R_2 = \{\gamma\}$

Rabin

k Rabin pairs: $(G_1,R_1), \ldots, (G_k,R_k)$, $G_i, R_i \subseteq C$

A run ρ is accepting $\Leftrightarrow$ $\bigvee \mathrm{Inf}(\rho) \cap G_i \neq \emptyset \wedge \mathrm{Inf}(\rho) \cap R_i = \emptyset$



$G_1 = \{\alpha\}$, $R_1 = \{\gamma\}$

$G_2 = \{\beta\}$, $R_2 = \{\gamma\}$

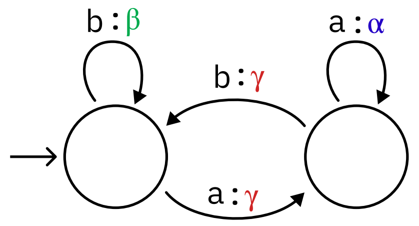
$$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$$

Different Acceptance Conditions

Muller

$\mathcal{F} \subseteq 2^{C}$ family of subsets

A run ρ is accepting ⇔ $\mathrm{Inf}(\rho) \in \mathcal{F}$



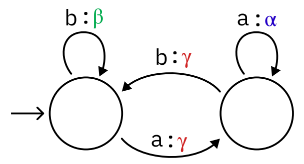
$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

$$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$$

Rabin

k Rabin pairs: $(G_1,R_1), \dots , (G_k,R_k)$, $\quad G_i, R_i \subseteq C$

A run ρ is accepting ⇔ $\bigvee \mathrm{Inf}(\rho) \cap G_i \neq \emptyset \wedge \mathrm{Inf}(\rho) \cap R_i = \emptyset$



$G_1 = \{\alpha\}, R_1 = \{\gamma\}$
$G_2 = \{\beta\}, R_2 = \{\gamma\}$

$$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$$

$C = \{\alpha, \beta, \gamma\}$

output colours

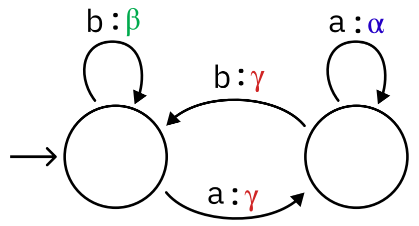
$\mathrm{Inf}(\rho) =$ {Colours appearing infinitely often in ρ}

Different Acceptance Conditions

Muller

$\mathcal{F} \subseteq 2^{C}$ family of subsets

A run ρ is accepting ⇔ $\mathrm{Inf}(\rho) \in \mathcal{F}$



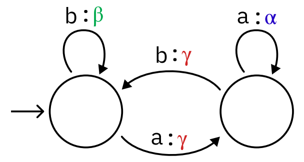
$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$

Rabin

k Rabin pairs: $(G_1,R_1), \dots, (G_k,R_k)$, $G_i, R_i \subseteq C$

A run ρ is accepting ⇔ $\bigvee \mathrm{Inf}(\rho) \cap G_i \neq \emptyset \wedge \mathrm{Inf}(\rho) \cap R_i = \emptyset$



$G_1 = \{\alpha\}, R_1 = \{\gamma\}$
$G_2 = \{\beta\}, R_2 = \{\gamma\}$

$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$

Streett

$C = \{\alpha, \beta, \gamma\}$

output colours

$\mathrm{Inf}(\rho) = \{$Colours appearing infinitely often in $\rho\}$

Different Acceptance Conditions

$C = \{\alpha, \beta, \gamma\}$

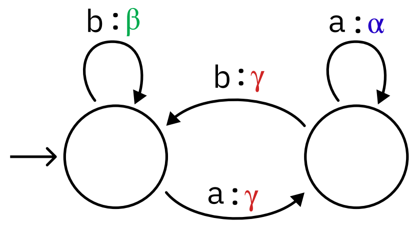
output colours

$\mathrm{Inf}(\rho) = \{$Colours appearing infinitely often in $\rho\}$

Muller

$\mathcal{F} \subseteq 2^C$ family of subsets

A run ρ is accepting $\Leftrightarrow$ $\mathrm{Inf}(\rho) \in \mathcal{F}$



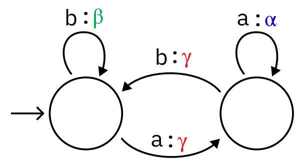
$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$

Rabin

k Rabin pairs: $(G_1,R_1), \dots, (G_k,R_k)$, $G_i, R_i \subseteq C$

A run ρ is accepting $\Leftrightarrow$ $\bigvee \mathrm{Inf}(\rho) \cap G_i \neq \emptyset \wedge \mathrm{Inf}(\rho) \cap R_i = \emptyset$



$G_1 = \{\alpha\}$, $R_1 = \{\gamma\}$
$G_2 = \{\beta\}$, $R_2 = \{\gamma\}$

$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$

Streett

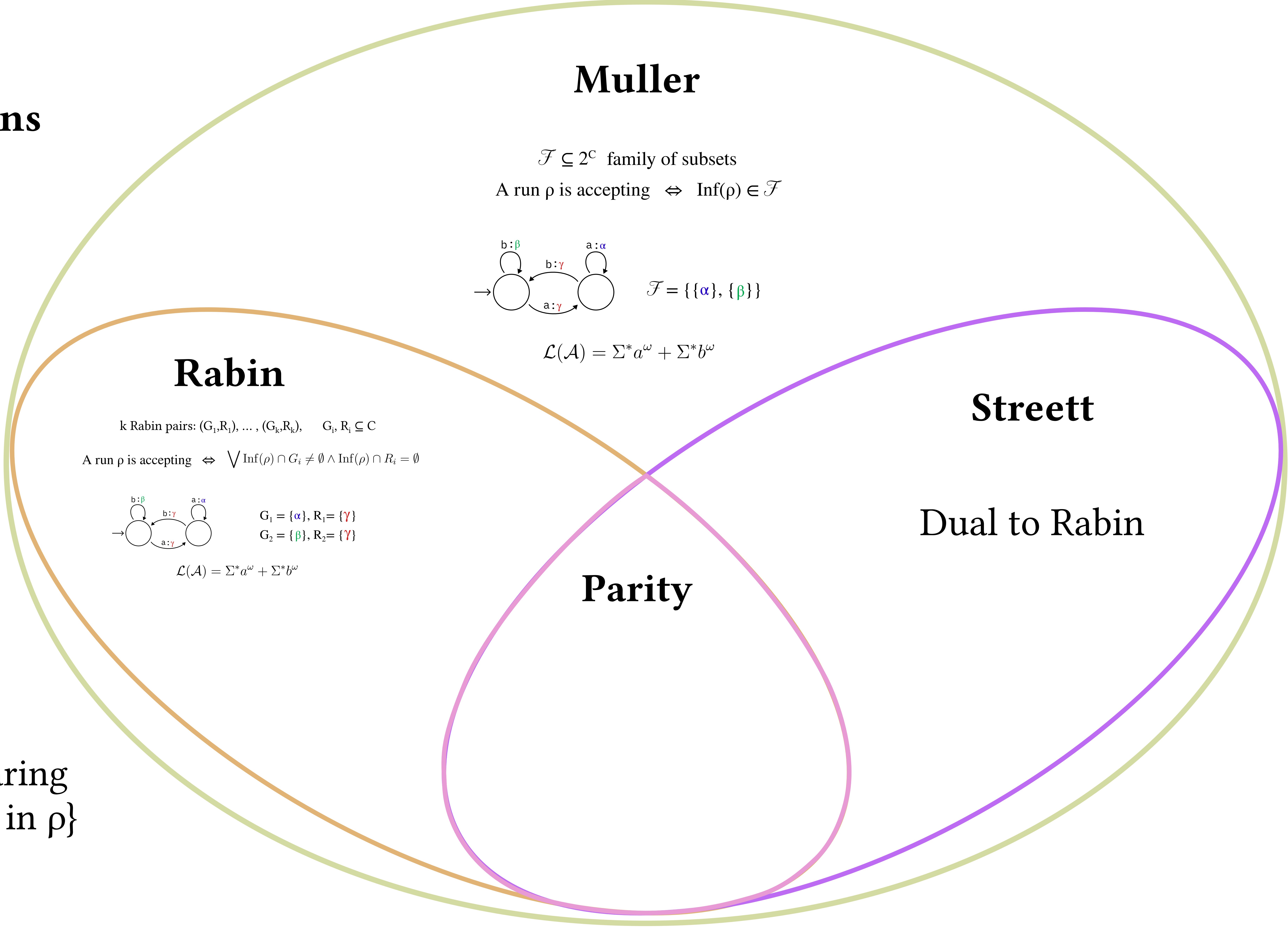
Dual to Rabin

Different Acceptance Conditions

$C = \{\alpha, \beta, \gamma\}$

output colours

$\mathrm{Inf}(\rho) = \{$Colours appearing infinitely often in $\rho\}$



Parity

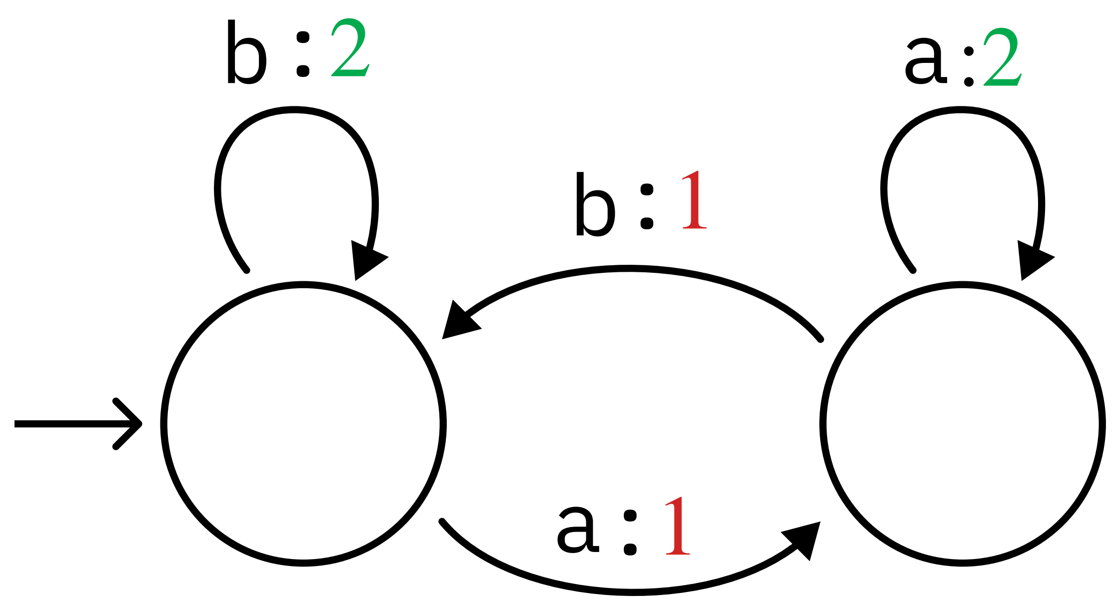
We assume $C \subseteq \mathbb{N}$

ρ accepting $\Leftrightarrow$ $\min \mathrm{Inf}(\rho)$ is even

Parity

We assume $C \subseteq \mathbb{N}$

ρ accepting $\Leftrightarrow$ min Inf(ρ) is even



$$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$$

Different Acceptance Conditions

$C = \{\alpha, \beta, \gamma\}$

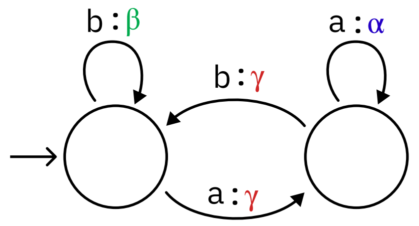
output colours

$\mathrm{Inf}(\rho) = \{$Colours appearing infinitely often in $\rho\}$

Muller

$\mathcal{F} \subseteq 2^C$ family of subsets

A run ρ is accepting $\Leftrightarrow$ $\mathrm{Inf}(\rho) \in \mathcal{F}$



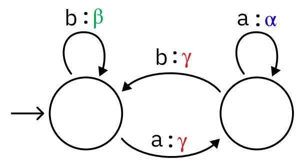
$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$

Rabin

k Rabin pairs: $(G_1,R_1), \dots, (G_k,R_k)$, $G_i, R_i \subseteq C$

A run ρ is accepting $\Leftrightarrow$ $\bigvee \mathrm{Inf}(\rho) \cap G_i \neq \emptyset \wedge \mathrm{Inf}(\rho) \cap R_i = \emptyset$



$G_1 = \{\alpha\}$, $R_1 = \{\gamma\}$

$G_2 = \{\beta\}$, $R_2 = \{\gamma\}$

$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$

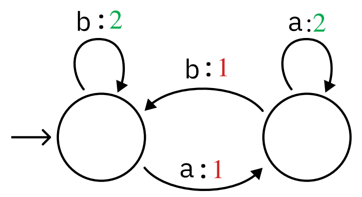
Streett

Dual to Rabin

Parity

We assume $C \subseteq \mathbb{N}$

ρ accepting $\Leftrightarrow$ min $\mathrm{Inf}(\rho)$ is even



$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$

Different Acceptance Conditions

$C = \{\alpha, \beta, \gamma\}$

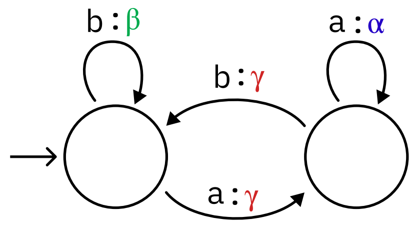
output colours

$\mathrm{Inf}(\rho) = \{$Colours appearing infinitely often in $\rho\}$

Muller

$\mathcal{F} \subseteq 2^C$ family of subsets

A run ρ is accepting $\Leftrightarrow$ $\mathrm{Inf}(\rho) \in \mathcal{F}$



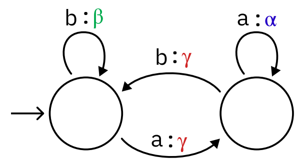
$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

$$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$$

Rabin

k Rabin pairs: $(G_1,R_1), \ldots, (G_k,R_k)$, $G_i, R_i \subseteq C$

A run ρ is accepting $\Leftrightarrow$ $\bigvee \mathrm{Inf}(\rho) \cap G_i \neq \emptyset \wedge \mathrm{Inf}(\rho) \cap R_i = \emptyset$



$G_1 = \{\alpha\}, R_1 = \{\gamma\}$
$G_2 = \{\beta\}, R_2 = \{\gamma\}$

$$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$$

WHY SO MANY?

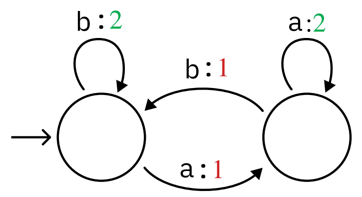
Streett

Dual to Rabin

Parity

We assume $C \subseteq \mathbb{N}$

ρ accepting $\Leftrightarrow$ min $\mathrm{Inf}(\rho)$ is even



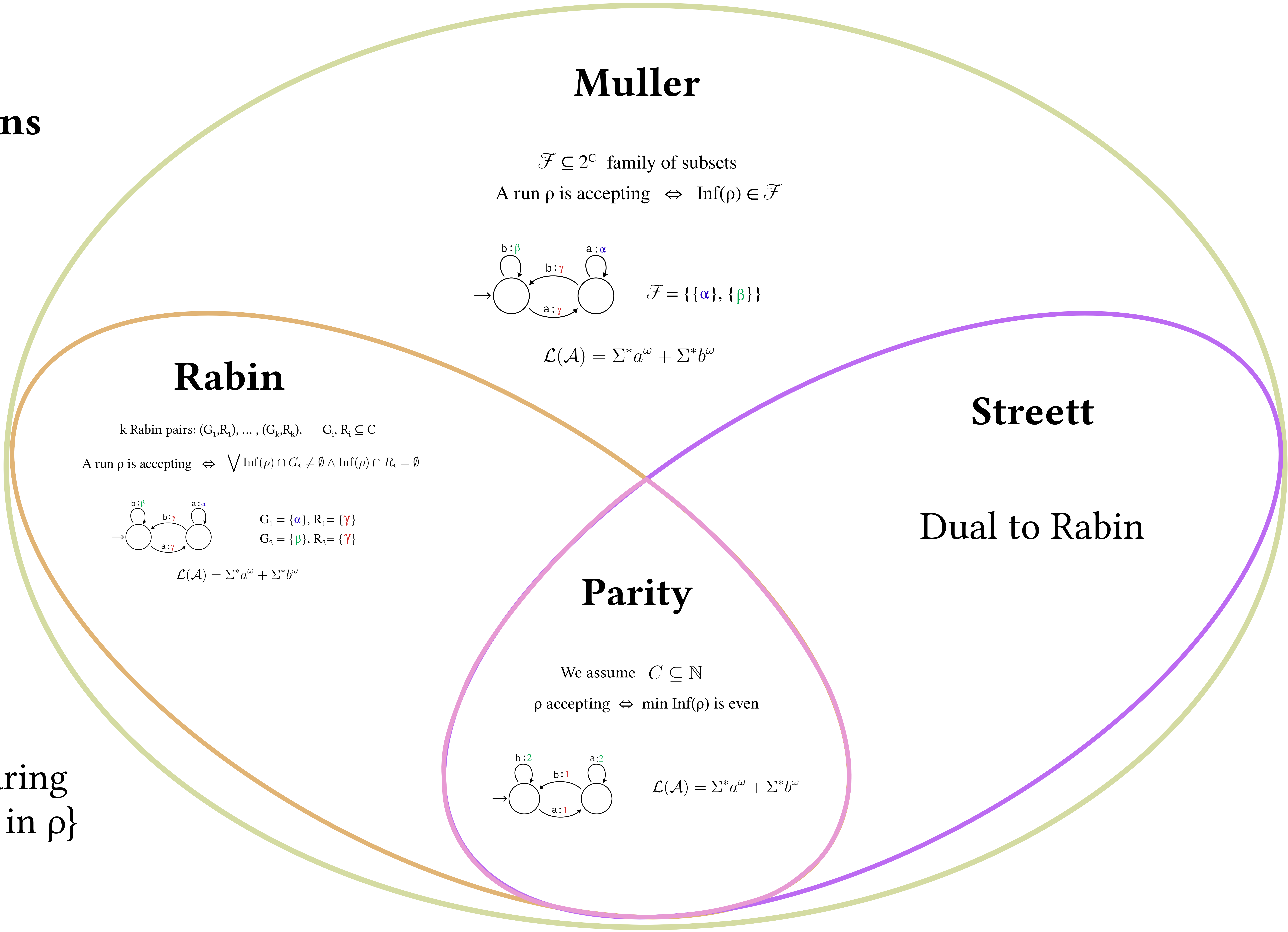
$$\mathcal{L}(\mathcal{A}) = \Sigma^* a^\omega + \Sigma^* b^\omega$$

Different Acceptance Conditions

$C = \{ \alpha, \beta, \gamma \}$

output colours

$\mathrm{Inf}(\rho) = \{$Colours appearing infinitely often in $\rho\}$



More general conditions → Automata are simpler to produce

More general conditions → Automata are simpler to produce

- LTL $\longrightarrow$ Det. Muller automata
- ND Büchi $\longrightarrow$ Det. Rabin automata

More general conditions → Automata are simpler to produce

- LTL $\longrightarrow$ Det. Muller automata
- ND Büchi $\longrightarrow$ Det. Rabin automata

Simpler conditions → More efficient algorithms

More general conditions $\rightarrow$ Automata are simpler to produce

- LTL $\longrightarrow$ Det. Muller automata
- ND Büchi $\longrightarrow$ Det. Rabin automata

Simpler conditions $\rightarrow$ More efficient algorithms

Complexity of solving 2-player games. $n =$ Size of input graph

More general conditions → Automata are simpler to produce

- LTL $\longrightarrow$ Det. Muller automata
- ND Büchi $\longrightarrow$ Det. Rabin automata

Simpler conditions → More efficient algorithms

Complexity of solving 2-player games. n = Size of input graph

Muller	PSPACE-complete	$\mathcal{O}(k^{5k}n^5)$

k = #Colours

More general conditions → Automata are simpler to produce

- LTL $\longrightarrow$ Det. Muller automata
- ND Büchi $\longrightarrow$ Det. Rabin automata

Simpler conditions → More efficient algorithms

Complexity of solving 2-player games. n = Size of input graph

Muller	PSPACE-complete	$\mathcal{O}(k^{5k}n^5)$
Rabin	NP-complete	$\mathcal{O}(n^{k+3}kk!)$

k = #Colours

k = #Rabin pair

More general conditions → Automata are simpler to produce

- LTL $\longrightarrow$ Det. Muller automata
- ND Büchi $\longrightarrow$ Det. Rabin automata

Simpler conditions → More efficient algorithms

Complexity of solving 2-player games. n = Size of input graph

Muller	PSPACE-complete	$\mathcal{O}(k^{5k}n^5)$	k = #Colours
Rabin	NP-complete	$\mathcal{O}(n^{k+3}kk!)$	k = #Rabin pair
Parity	NP$\bigcap$coNP. In P?	$\mathcal{O}(n^{\log(k)})$	k = #Colours

More general conditions → Automata are simpler to produce

- LTL $\longrightarrow$ Det. Muller automata
- ND Büchi $\longrightarrow$ Det. Rabin automata

Simpler conditions → More efficient algorithms

Complexity of solving 2-player games. n = Size of input graph

Muller	PSPACE-complete	$\mathcal{O}(k^{5k}n^5)$	k = #Colours
Rabin	NP-complete	$\mathcal{O}(n^{k+3}kk!)$	k = #Rabin pair
Parity	NP$\bigcap$coNP. In P?	$\mathcal{O}(n^{\log(k)})$	k = #Colours

Goal: Simplify the acceptance condition of automata

More general conditions → Automata are simpler to produce

- LTL $\longrightarrow$ Det. Muller automata
- ND Büchi $\longrightarrow$ Det. Rabin automata

Simpler conditions → More efficient algorithms

Complexity of solving 2-player games. n = Size of input graph

Muller	PSPACE-complete	$\mathcal{O}(k^{5k}n^5)$	k = #Colours
Rabin	NP-complete	$\mathcal{O}(n^{k+3}kk!)$	k = #Rabin pair
Parity	NP$\bigcap$coNP. In P?	$\mathcal{O}(n^{\log(k)})$	k = #Colours

Goal: Simplify the acceptance condition of automata

a) Simplify the type (Muller → Rabin → Parity)

More general conditions → Automata are simpler to produce

- LTL ⟶ Det. Muller automata
- ND Büchi ⟶ Det. Rabin automata

Simpler conditions → More efficient algorithms

Complexity of solving 2-player games. n = Size of input graph

Muller	PSPACE-complete	$\mathcal{O}(k^{5k}n^5)$	k = #Colours
Rabin	NP-complete	$\mathcal{O}(n^{k+3}kk!)$	k = #Rabin pair
Parity	NP$\bigcap$coNP. In P?	$\mathcal{O}(n^{\log(k)})$	k = #Colours

Goal: Simplify the acceptance condition of automata

a) Simplify the type (Muller → Rabin → Parity)

b) Minimise #Colours / #Rabin pairs

Approach 1

Add states to the automaton to maximize simplification

Approach 1

Add states to the automaton to maximize simplification

✓ Well understood

Approach 1

Add states to the automaton to maximize simplification

✓ Well understood

✦ Optimal transformation to parity automata
(C. Colcombet, Fijalkow '21)

Approach 1

Add states to the automaton to
maximize simplification

√ Well understood

✦ Optimal transformation to parity automata
(C. Colcombet, Fijalkow '21)

✘ Exponential blowup in the worst case
(Löding '99)

Approach 1

Add states to the automaton to maximize simplification

Well understood

Optimal transformation to parity automata
(C. Colcombet, Fijalkow '21)

Exponential blowup in the worst case
(Löding '99)

In this paper

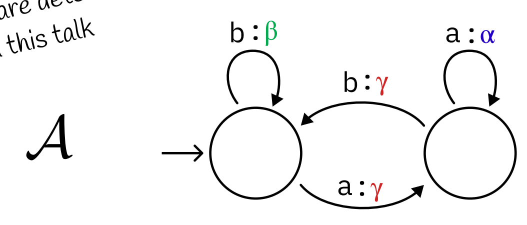
Simplify the condition **without** adding further states

Simplifying ω-Automata through the Alternating Cycle Decomposition

Antonio Casares · University of Warsaw & Corto Mascle · LaBRI, University of Bordeaux

Automata over infinite words (ω-automata)

Automata are deterministic in this talk



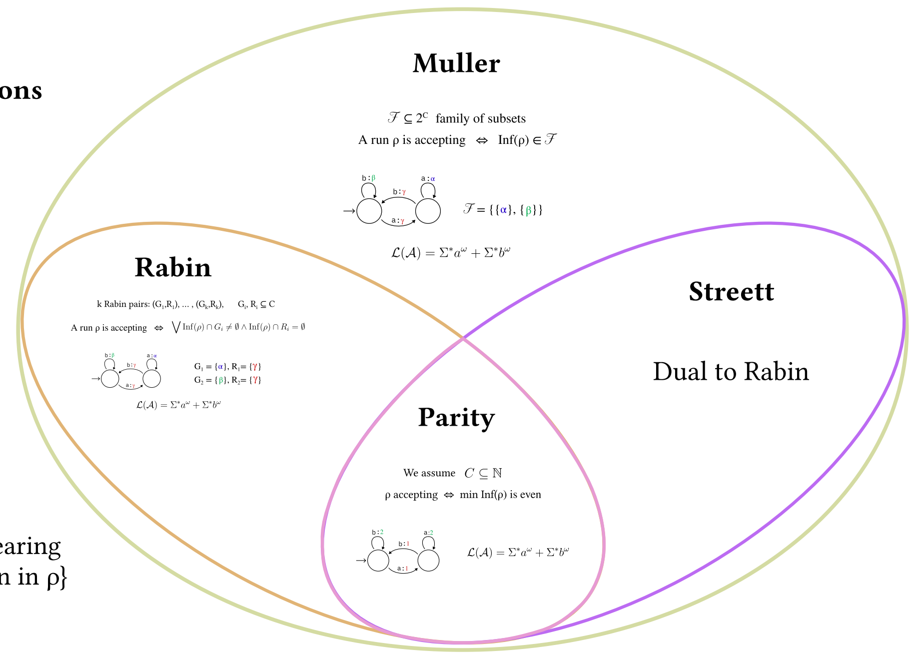
Input w ∈ Σ^ω ⟶ Infinite path ρ

Different Acceptance Conditions

C = {α, β, γ}

output colours

Inf(ρ) = {Colours appearing infinitely often in ρ}



More general conditions → Automata are simpler to produce

- LTL ⟶ Det. Muller automata
- ND Büchi ⟶ Det. Rabin automata

Simpler conditions → More efficient algorithms

Complexity of solving 2-player games. n = Size of input graph

Muller	PSPACE-complete	$\mathcal{O}(k^{5k}n^5)$	k = #Colours
Rabin	NP-complete	$\mathcal{O}(n^{k+3}kk!)$	k = #Rabin pair
Parity	NP∩coNP. In P?	$\mathcal{O}(n^{\log(k)})$	k = #Colours

Goal: Simplify the acceptance condition of automata

a) Simplify the type (Muller → Rabin → Parity)

b) Minimise #Colours / #Rabin pairs

Approach 1

Add states to the automaton to maximize simplification

√ Well understood

✦ Optimal transformation to parity automata *(C. Colcombet, Fijalkow '21)*

✗ Exponential blowup in the worst case *(Löding '99)*

In this paper

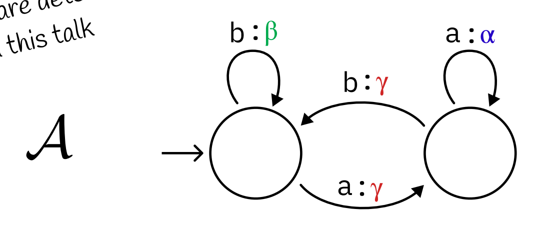
Simplify the condition **without** adding further states

Simplifying ω-Automata through the Alternating Cycle Decomposition

Antonio Casares · University of Warsaw & Corto Mascle · LaBRI, University of Bordeaux

Automata over infinite words (ω-automata)

Automata are deterministic in this talk

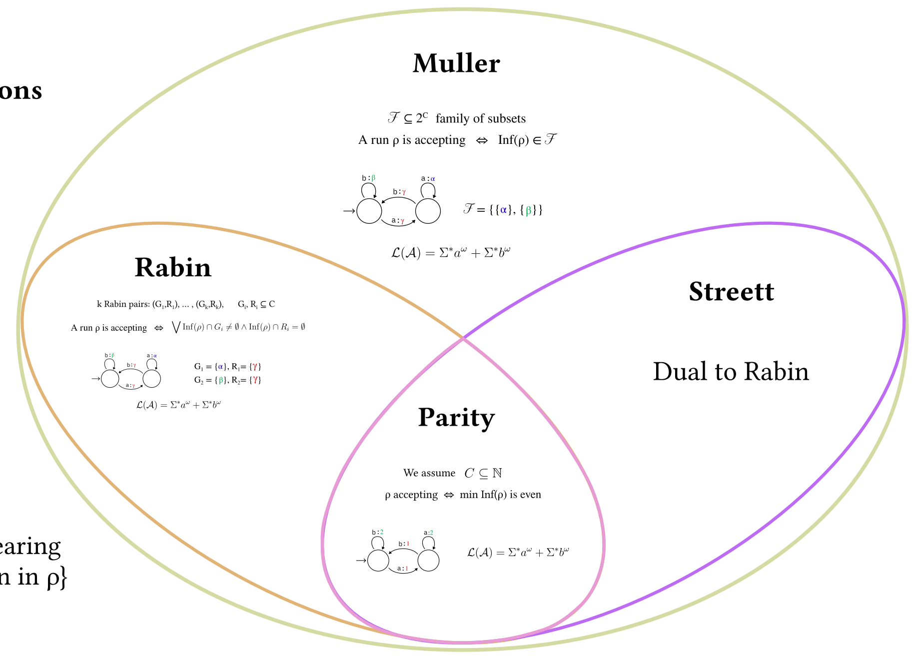


Input $w \in \Sigma^\omega$ ⟶ Infinite path ρ

Different Acceptance Conditions

$C = \{\alpha, \beta, \gamma\}$
output colours

Inf(ρ) = {Colours appearing infinitely often in ρ}



More general conditions → Automata are simpler to produce

- LTL → Det. Muller automata
- ND Büchi → Det. Rabin automata

Simpler conditions → More efficient algorithms

Complexity of solving 2-player games. n = Size of input graph

Muller	PSPACE-complete	$\mathcal{O}(k^{5k}n^5)$	k = #Colours
Rabin	NP-complete	$\mathcal{O}(n^{k+3}kk!)$	k = #Rabin pair
Parity	NP∩coNP. In P?	$\mathcal{O}(n^{\log(k)})$	k = #Colours

Goal: Simplify the acceptance condition of automata

a) Simplify the type (Muller → Rabin → Parity)

b) Minimise #Colours / #Rabin pairs

Approach 1

Add states to the automaton to maximize simplification

√ Well understood

✦ Optimal transformation to parity automata *(C. Colcombet, Fijalkow '21)*

✗ Exponential blowup in the worst case *(Löding '99)*

In this paper

Simplify the condition **without** adding further states

Pure Conditions

Pure Conditions

Reduction of conditions

A reduction of a condition $\mathcal{F} \subseteq 2^C$ to $\mathcal{G} \subseteq 2^{C'}$ is a mapping

$$\varphi : C \rightarrow C' \qquad \text{such that}$$

$$X \in \mathcal{F} \quad \Leftrightarrow \quad \varphi(X) \in \mathcal{G}, \quad \text{for all } X \subseteq C$$

Pure Conditions

Reduction of conditions

A reduction of a condition $\mathcal{F} \subseteq 2^C$ to $\mathcal{G} \subseteq 2^{C'}$ is a mapping

$$\varphi : C \longrightarrow C' \qquad \text{such that}$$

$$X \in \mathcal{F} \quad \Leftrightarrow \quad \varphi(X) \in \mathcal{G}, \qquad \text{for all } X \subseteq C$$

Given a condition $\mathcal{F}$, can we reduce it to

- a Muller condition with $\leq$ k colours?
- a Rabin condition? With $\leq$ k Rabin pairs?
- a parity condition? With $\leq$ k colours?

Main tool: Zielonka DAG *(Zielonka '98, Hunter-Dawar '06)*

Main tool: Zielonka DAG *(Zielonka '98, Hunter-Dawar '06)*

$C = \{\alpha, \beta, \gamma, \delta, \lambda\}$

$\mathcal{F} = \{\{\alpha, \beta, \gamma\}, \{\alpha, \gamma\}, \{\alpha, \beta\}, \{\beta, \gamma, \delta, \lambda\}, \{\beta, \delta, \lambda\},$
$\{\gamma, \delta, \lambda\}, \{\beta, \gamma, \lambda\}, \{\beta, \gamma, \delta\}, \{\beta, \delta\}, \{\beta, \lambda\}, \{\gamma, \delta\}, \{\gamma, \lambda\}\}$

Main tool: Zielonka DAG *(Zielonka '98, Hunter-Dawar '06)*

$C = \{\alpha, \beta, \gamma, \delta, \lambda\}$

$\mathcal{F} = \{\{\alpha,\beta,\gamma\}, \{\alpha,\gamma\}, \{\alpha,\beta\}, \{\beta,\gamma,\delta,\lambda\}, \{\beta,\delta,\lambda\}, \{\gamma,\delta,\lambda\}, \{\beta,\gamma,\lambda\}, \{\beta,\gamma,\delta\}, \{\beta,\delta\}, \{\beta,\lambda\}, \{\gamma,\delta\}, \{\gamma,\lambda\}\}$

Root $\longrightarrow$ All colours $\boxed{\alpha \;\; \beta \;\; \gamma \;\; \delta \;\; \lambda}$

Main tool: Zielonka DAG *(Zielonka '98, Hunter-Dawar '06)*

$C = \{\alpha, \beta, \gamma, \delta, \lambda\}$

$\mathcal{F} = \{\{\alpha, \beta, \gamma\}, \{\alpha, \gamma\}, \{\alpha, \beta\}, \{\beta, \gamma, \delta, \lambda\}, \{\beta, \delta, \lambda\}, \{\gamma, \delta, \lambda\}, \{\beta, \gamma, \lambda\}, \{\beta, \gamma, \delta\}, \{\beta, \delta\}, \{\beta, \lambda\}, \{\gamma, \delta\}, \{\gamma, \lambda\}\}$

Root $\longrightarrow$ All colours $\boxed{\alpha \quad \beta \quad \gamma \quad \delta \quad \lambda}$

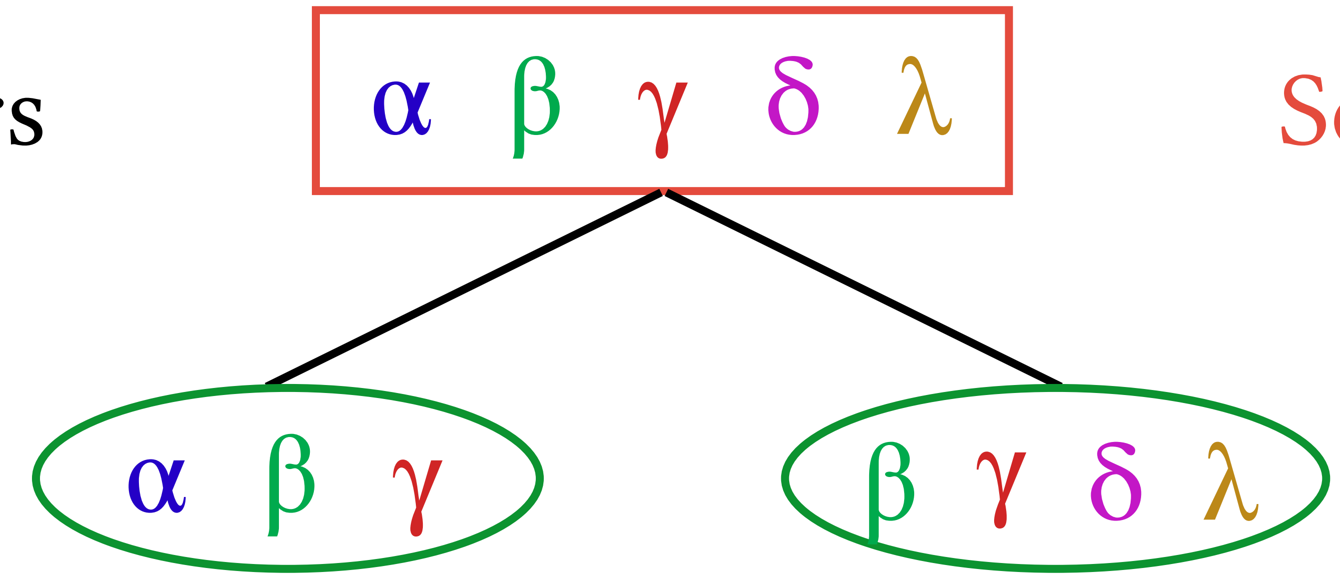
Square if it is a rejecting set

Main tool: Zielonka DAG *(Zielonka '98, Hunter-Dawar '06)*

$C = \{\alpha, \beta, \gamma, \delta, \lambda\}$

$\mathcal{F} = \{\{\alpha, \beta, \gamma\}, \{\alpha, \gamma\}, \{\alpha, \beta\}, \{\beta, \gamma, \delta, \lambda\}, \{\beta, \delta, \lambda\},$
$\{\gamma, \delta, \lambda\}, \{\beta, \gamma, \lambda\}, \{\beta, \gamma, \delta\}, \{\beta, \delta\}, \{\beta, \lambda\}, \{\gamma, \delta\}, \{\gamma, \lambda\}\}$

Root $\longrightarrow$ All colours



Square if it is a rejecting set

Children: Maximal accepting subsets

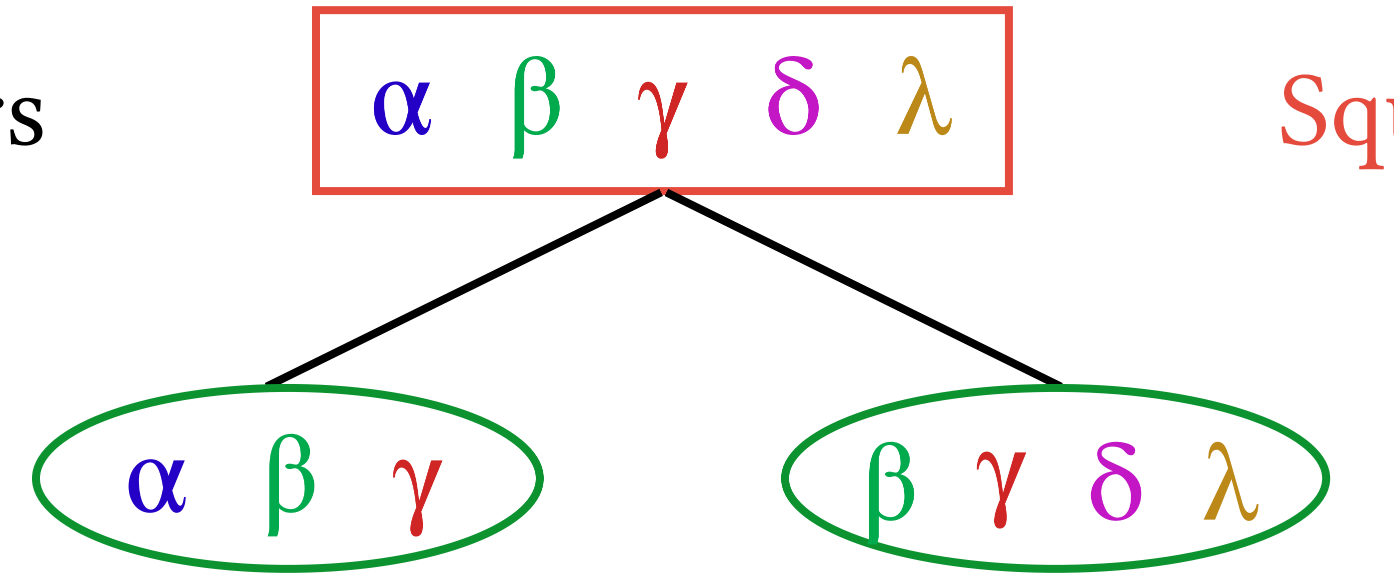
Main tool: Zielonka DAG *(Zielonka '98, Hunter-Dawar '06)*

$C = \{\alpha, \beta, \gamma, \delta, \lambda\}$

$\mathcal{F} = \{\{\alpha, \beta, \gamma\}, \{\alpha, \gamma\}, \{\alpha, \beta\}, \{\beta, \gamma, \delta, \lambda\}, \{\beta, \delta, \lambda\}, \{\gamma, \delta, \lambda\}, \{\beta, \gamma, \lambda\}, \{\beta, \gamma, \delta\}, \{\beta, \delta\}, \{\beta, \lambda\}, \{\gamma, \delta\}, \{\gamma, \lambda\}\}$

Root ⟶ All colours

Square if it is a rejecting set



Children: Maximal accepting subsets

Round because they are accepting sets

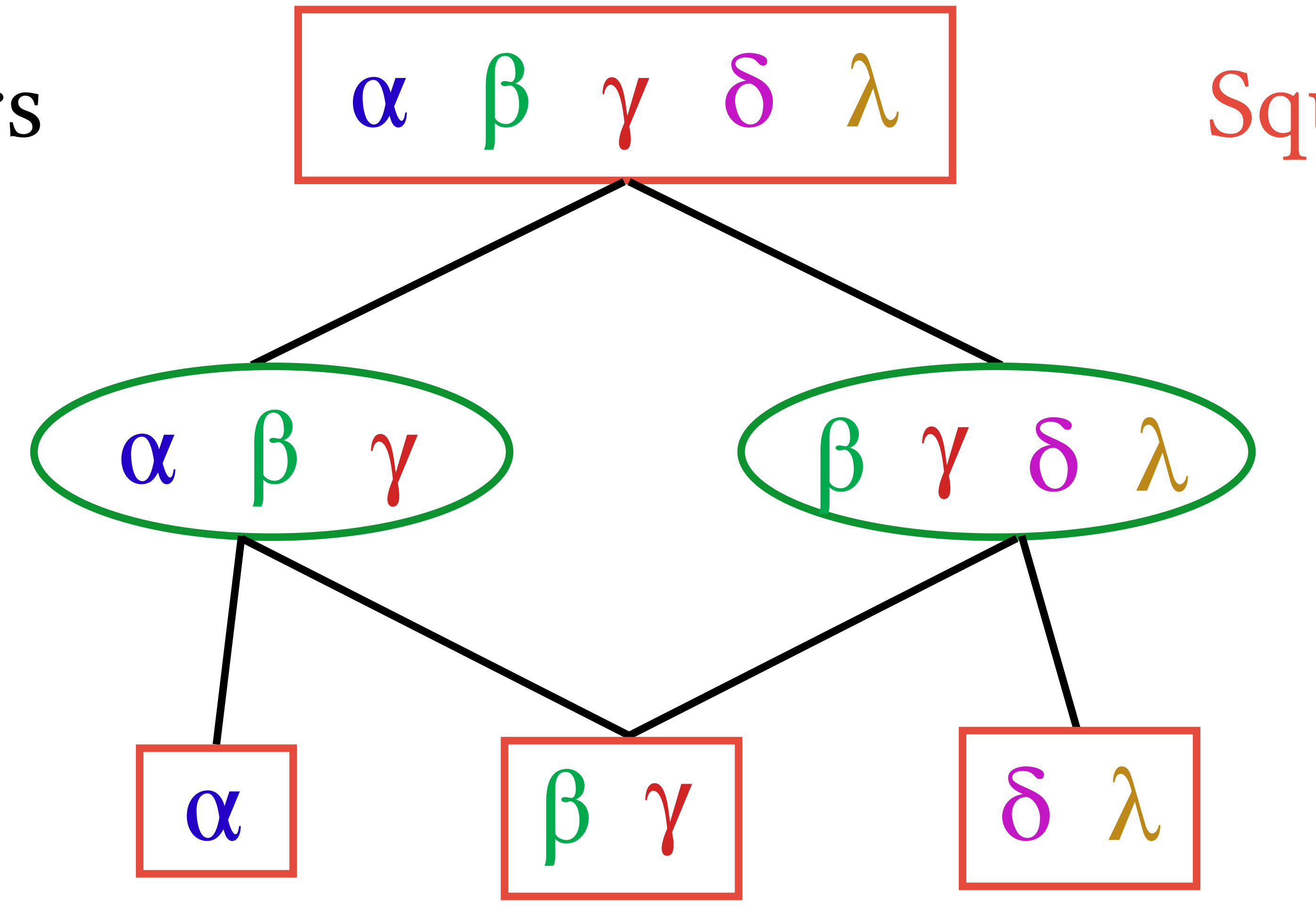
Main tool: Zielonka DAG *(Zielonka '98, Hunter-Dawar '06)*

$C = \{\alpha, \beta, \gamma, \delta, \lambda\}$

$\mathcal{F} = \{\{\alpha, \beta, \gamma\}, \{\alpha, \gamma\}, \{\alpha, \beta\}, \{\beta, \gamma, \delta, \lambda\}, \{\beta, \delta, \lambda\}, \{\gamma, \delta, \lambda\}, \{\beta, \gamma, \lambda\}, \{\beta, \gamma, \delta\}, \{\beta, \delta\}, \{\beta, \lambda\}, \{\gamma, \delta\}, \{\gamma, \lambda\}\}$

Root ⟶ All colours

Square if it is a rejecting set



Children: Maximal accepting subsets

Round because they are accepting sets

Children: Maximal rejecting subsets

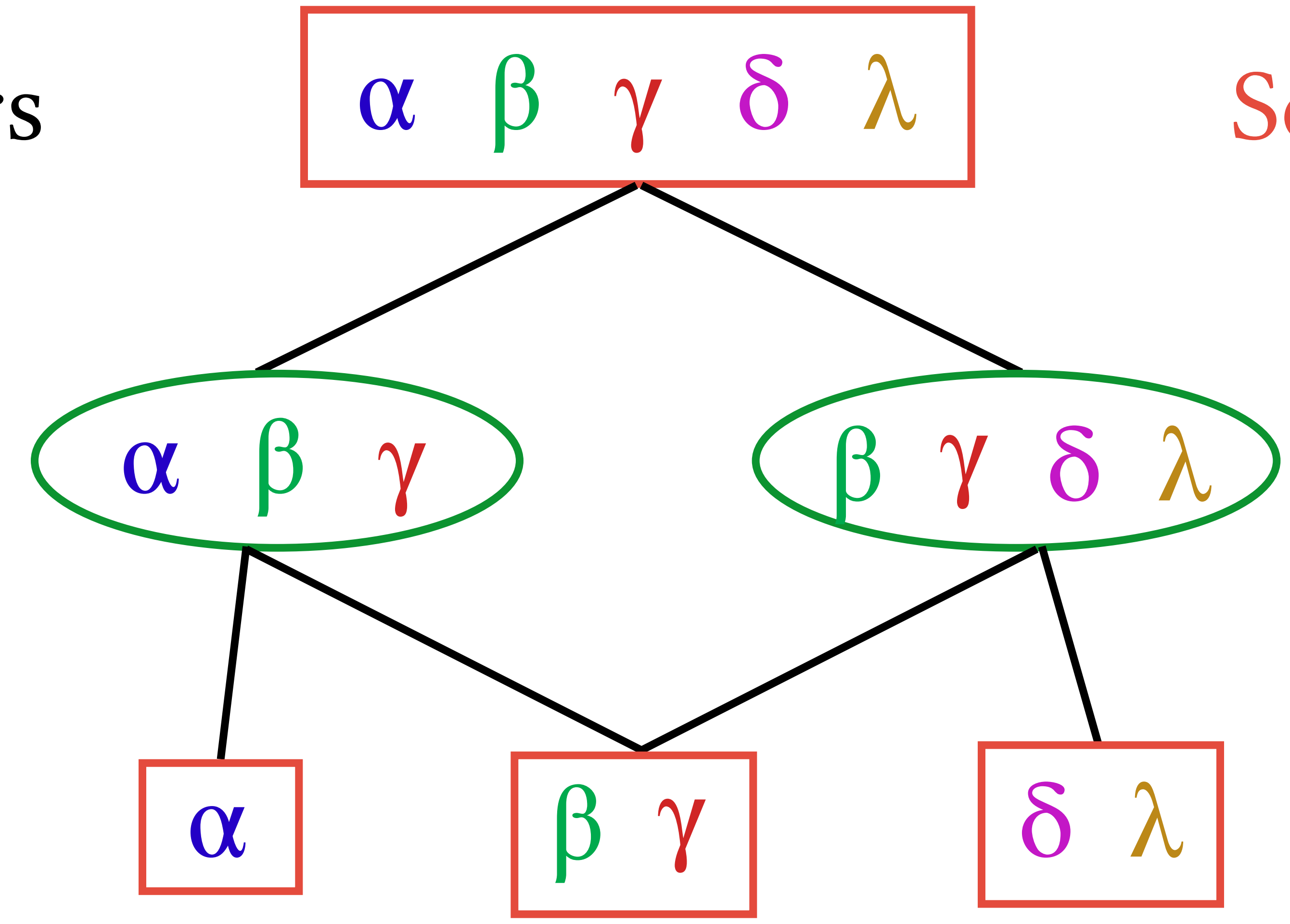
Main tool: Zielonka DAG *(Zielonka '98, Hunter-Dawar '06)*

$C = \{\alpha, \beta, \gamma, \delta, \lambda\}$

$\mathcal{F} = \{\{\alpha, \beta, \gamma\}, \{\alpha, \gamma\}, \{\alpha, \beta\}, \{\beta, \gamma, \delta, \lambda\}, \{\beta, \delta, \lambda\}, \{\gamma, \delta, \lambda\}, \{\beta, \gamma, \lambda\}, \{\beta, \gamma, \delta\}, \{\beta, \delta\}, \{\beta, \lambda\}, \{\gamma, \delta\}, \{\gamma, \lambda\}\}$

Root ⟶ All colours

Square if it is a rejecting set



Children: Maximal accepting subsets

Round because they are accepting sets

Children: Maximal rejecting subsets

PROPOSITION *(Hunter-Dawar '06)*

The Zielonka DAG can be computed in polynomial time in $|\mathcal{F}|$.

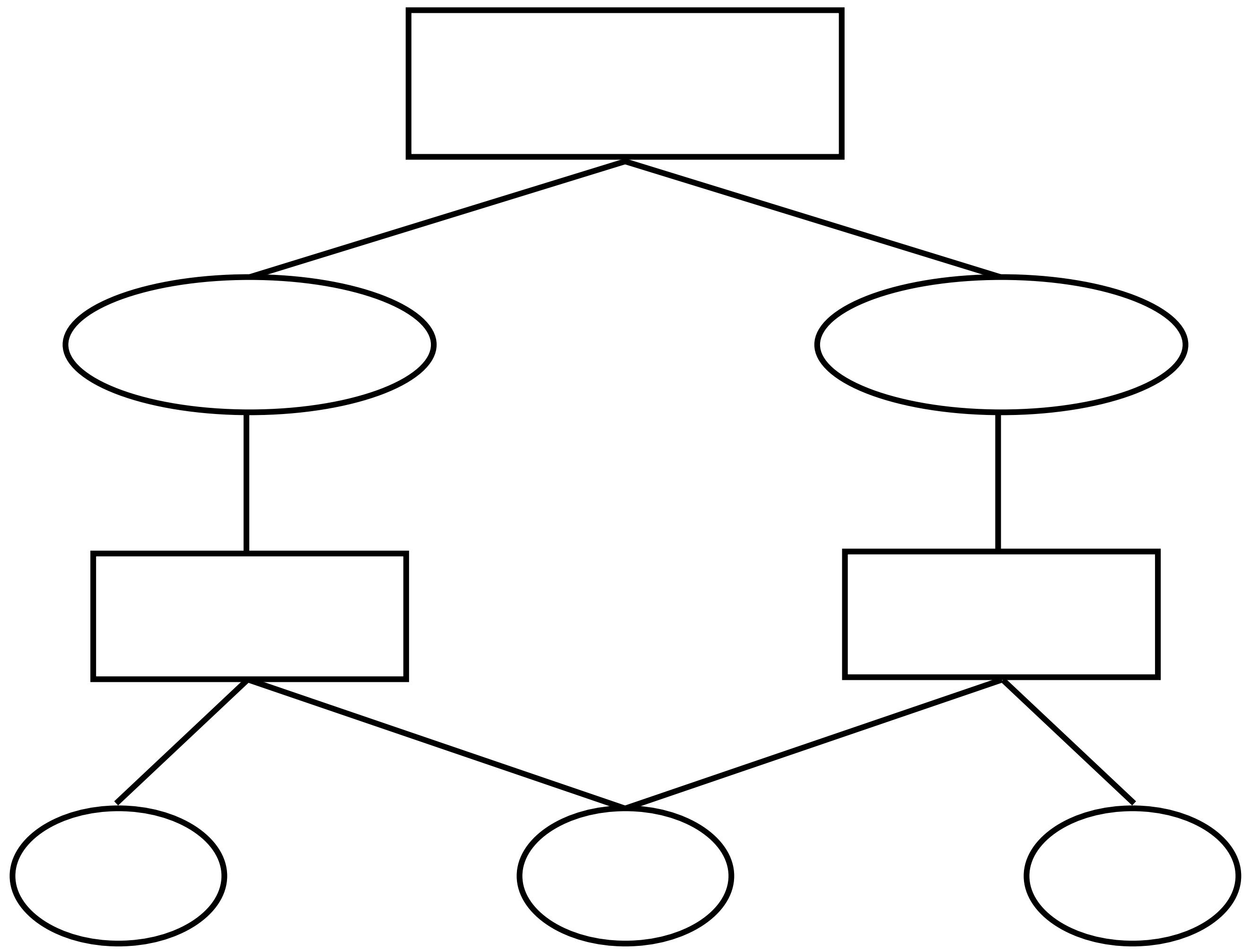
Typeness: Can we reduce $\mathcal{F}$ to a Rabin/parity condition?

Typeness: Can we reduce $\mathcal{F}$ to a Rabin/parity condition?

Just look to the Zielonka DAG! *(Zielonka '98)*

Typeness: Can we reduce $\mathcal{F}$ to a Rabin/parity condition?

Just look to the Zielonka DAG! *(Zielonka '98)*



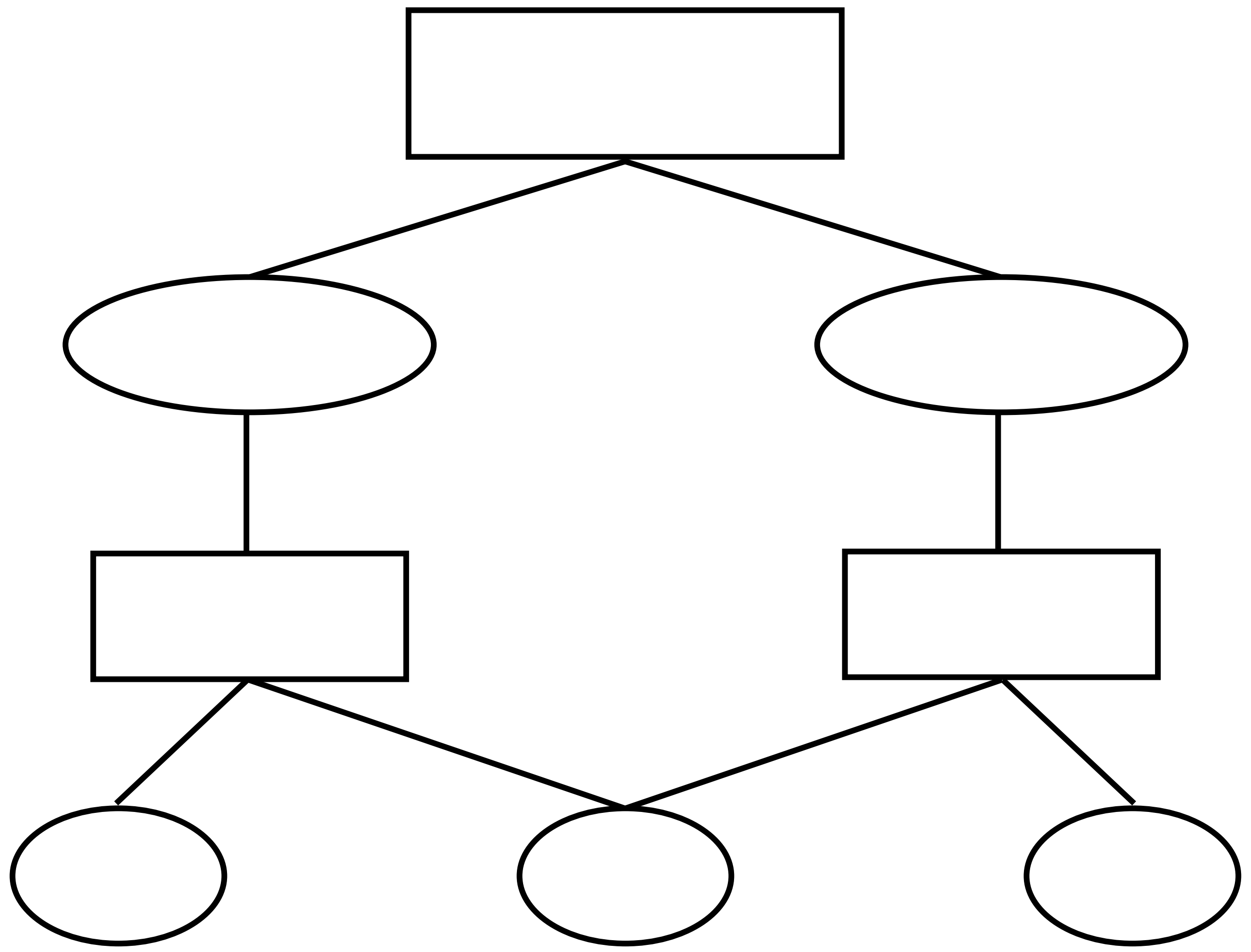
Rabin

$\Leftrightarrow$

Round nodes have a single child

Typeness: Can we reduce $\mathscr{F}$ to a Rabin/parity condition?

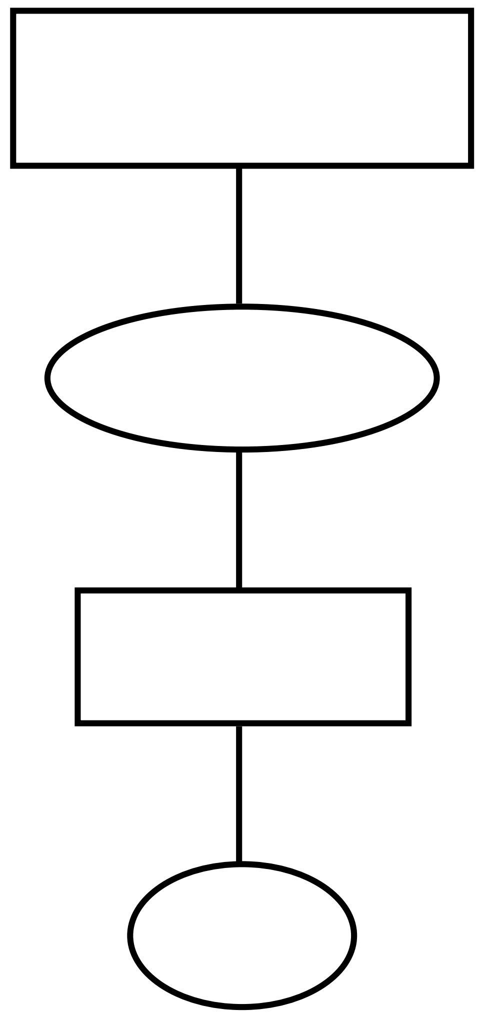
Just look to the Zielonka DAG! *(Zielonka '98)*



Rabin

$\Leftrightarrow$

Round nodes have a single child



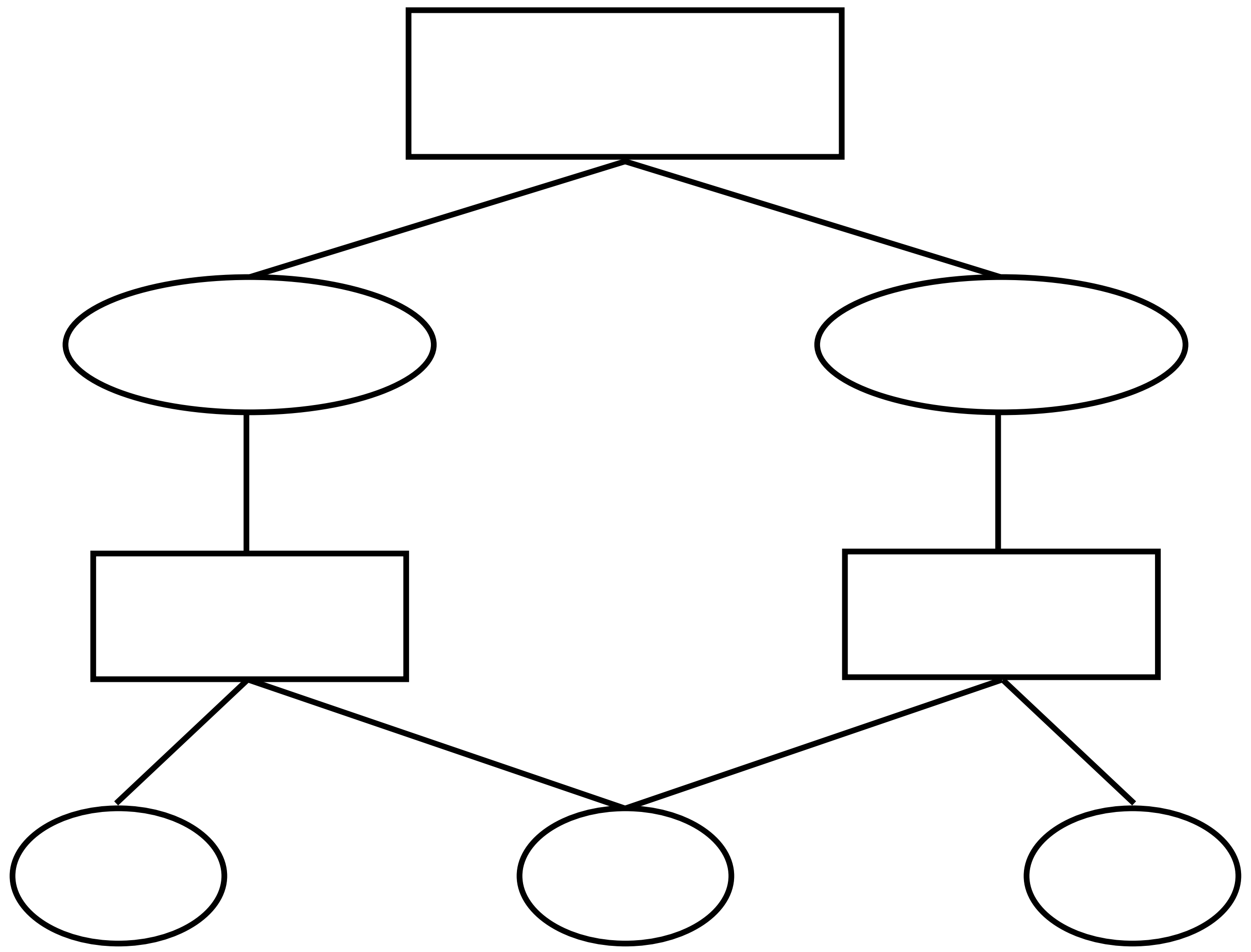
Parity

$\Leftrightarrow$

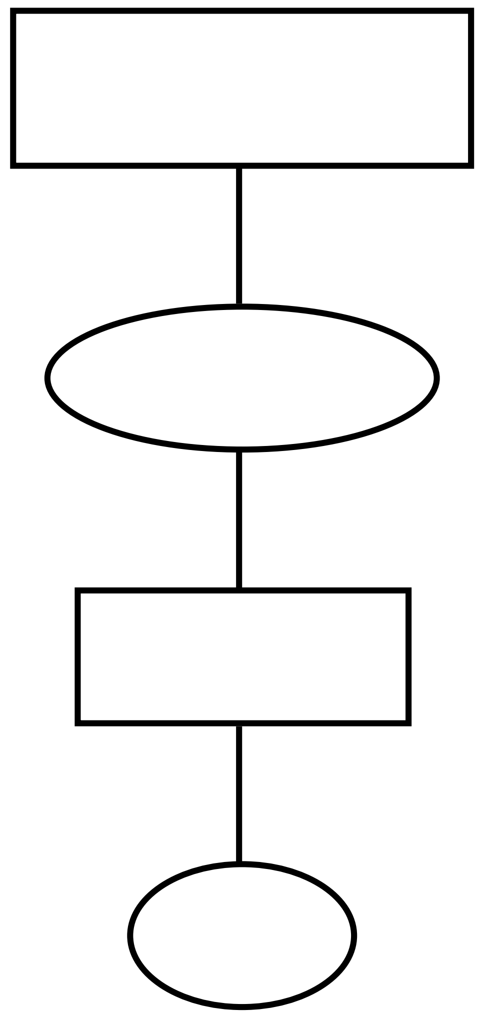
A single branch

Typeness: Can we reduce $\mathcal{F}$ to a Rabin/parity condition?

Just look to the Zielonka DAG! *(Zielonka '98)*



Rabin
$\Leftrightarrow$
Round nodes have a single child



Parity
$\Leftrightarrow$
A single branch

+ Minimal #Colours
=
Height of the branch

Minimisation of colours and Rabin pairs

Minimisation of colours and Rabin pairs

Theorem

We can minimise the #Colours (resp. #Rabin pairs) in polynomial time.

Minimisation of colours and Rabin pairs

THEOREM

We can minimise the #Colours (resp. #Rabin pairs) in polynomial time.

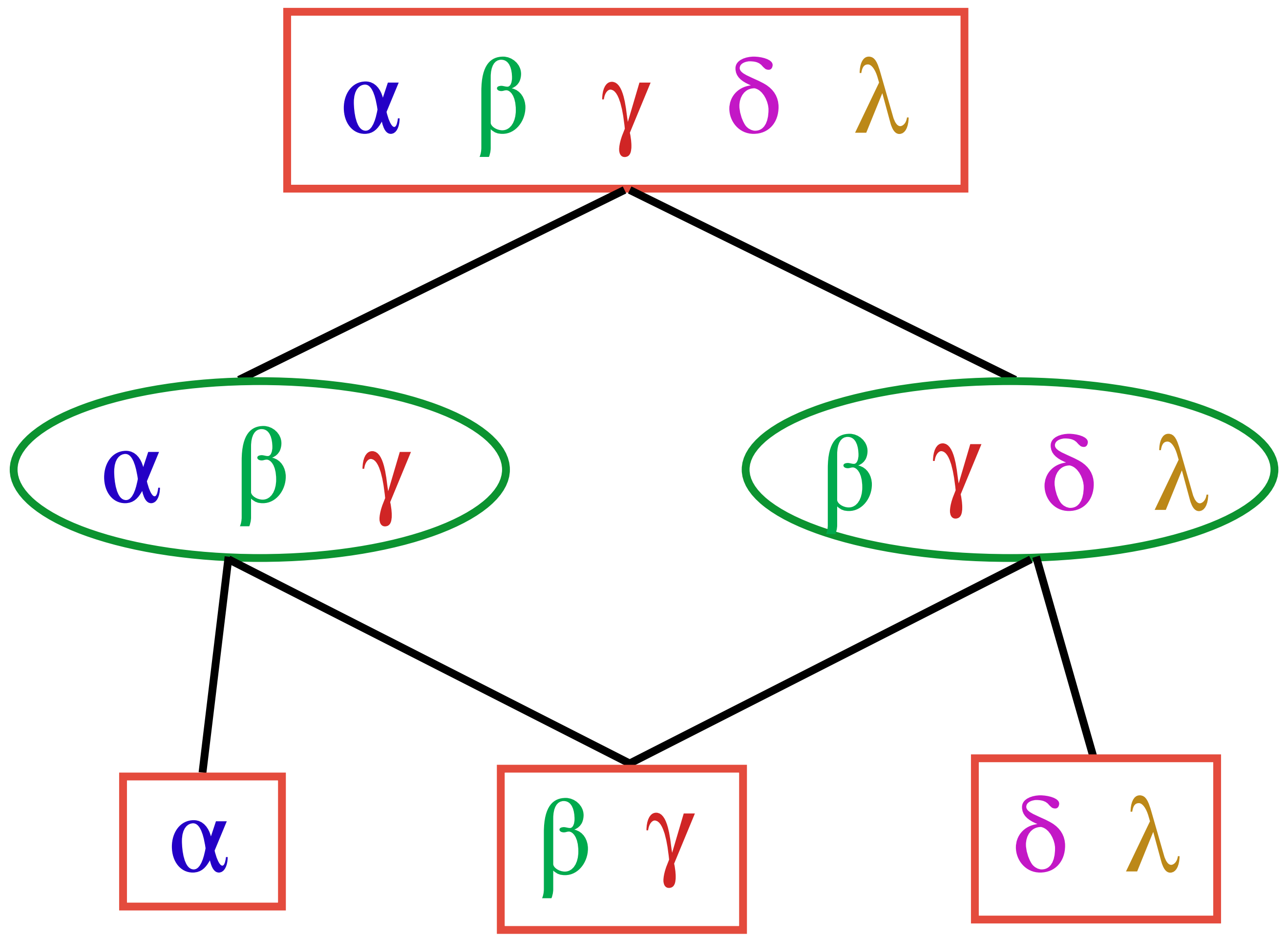
Idea (for colours):

Minimisation of colours and Rabin pairs

THEOREM

We can minimise the #Colours (resp. #Rabin pairs) in polynomial time.

Idea (for colours):

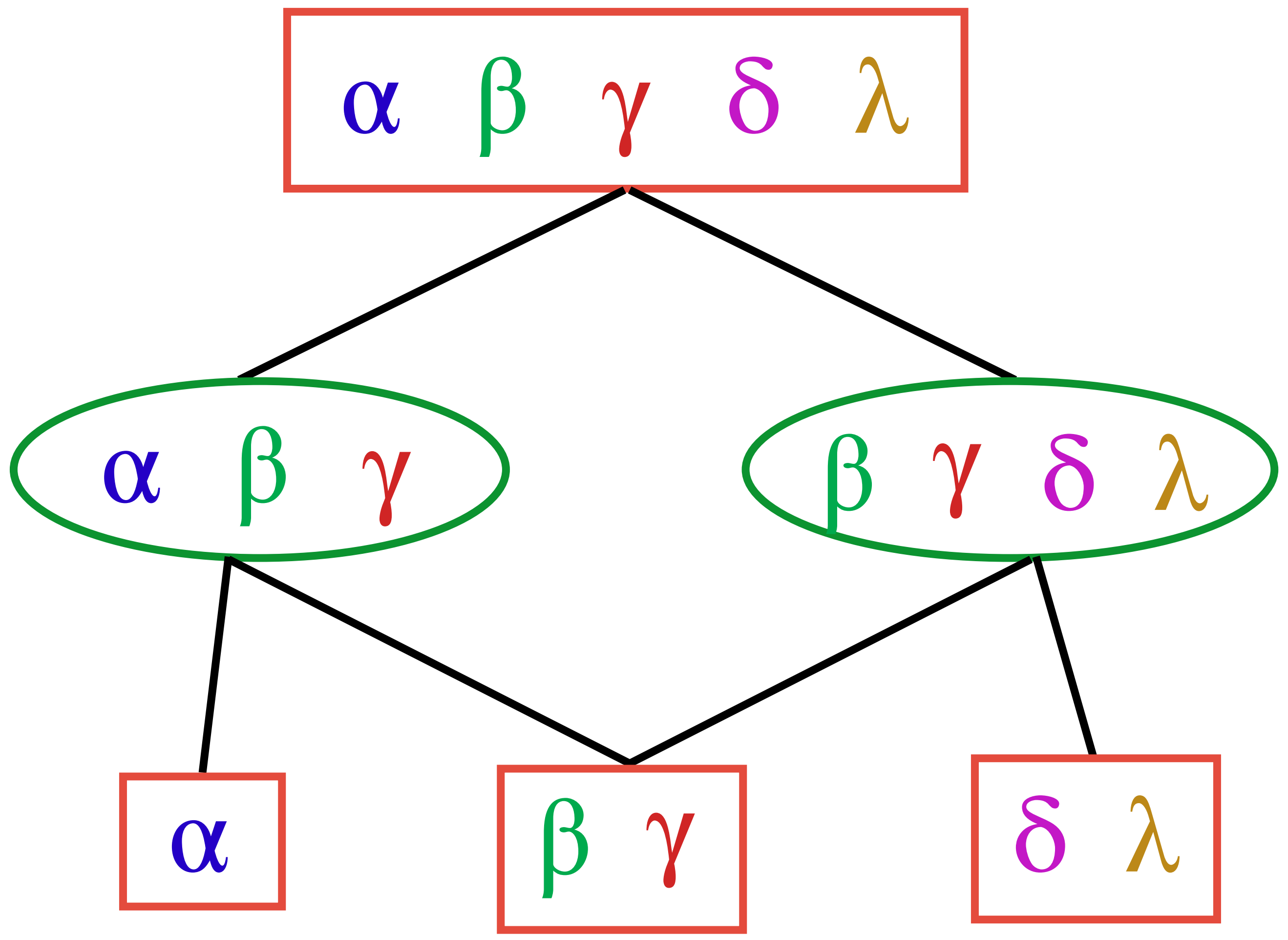


Minimisation of colours and Rabin pairs

Theorem

We can minimise the #Colours (resp. #Rabin pairs) in polynomial time.

Idea (for colours):



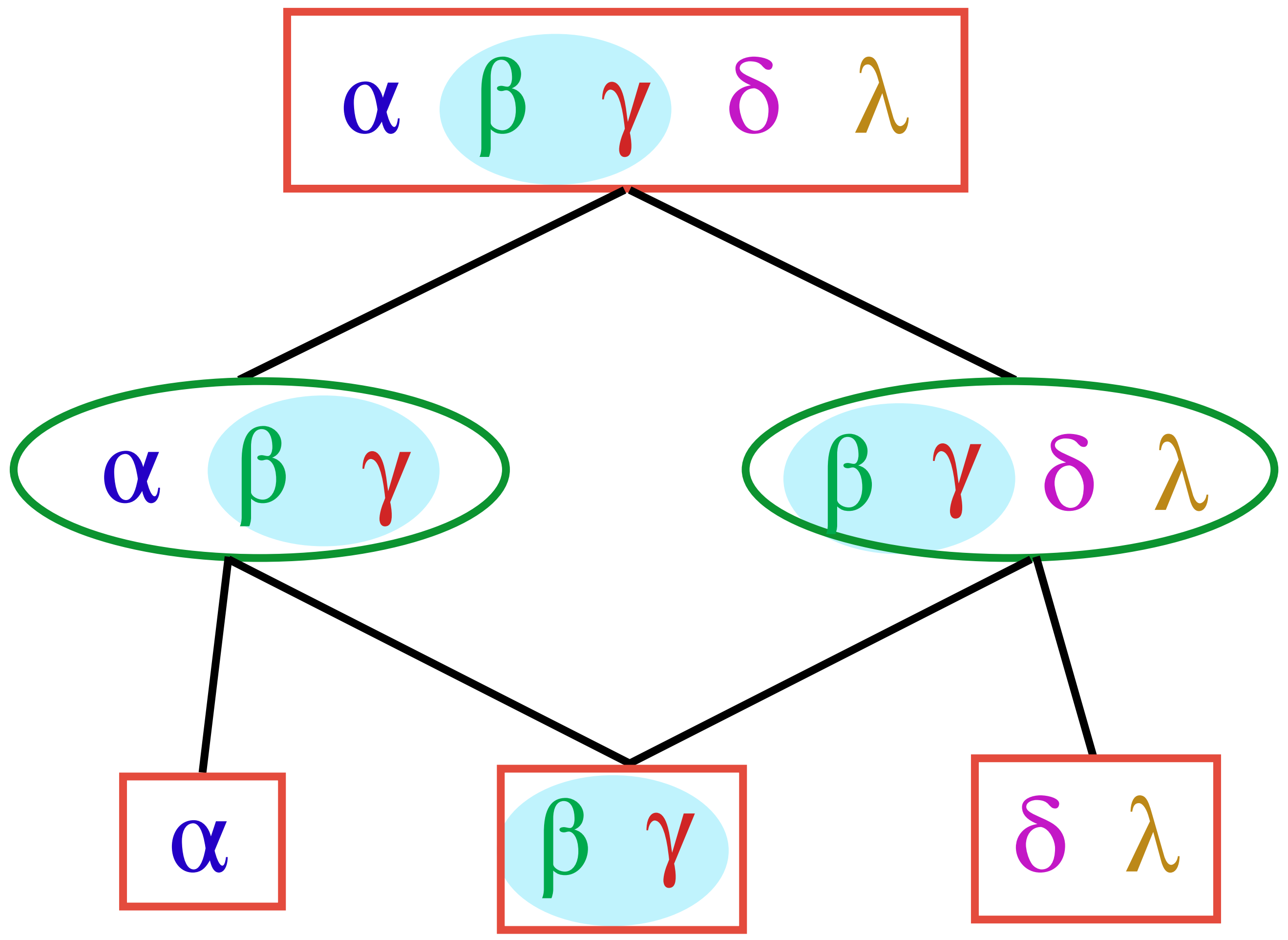
Two colours are equivalent if they appear in the same nodes

Minimisation of colours and Rabin pairs

THEOREM

We can minimise the #Colours (resp. #Rabin pairs) in polynomial time.

Idea (for colours):



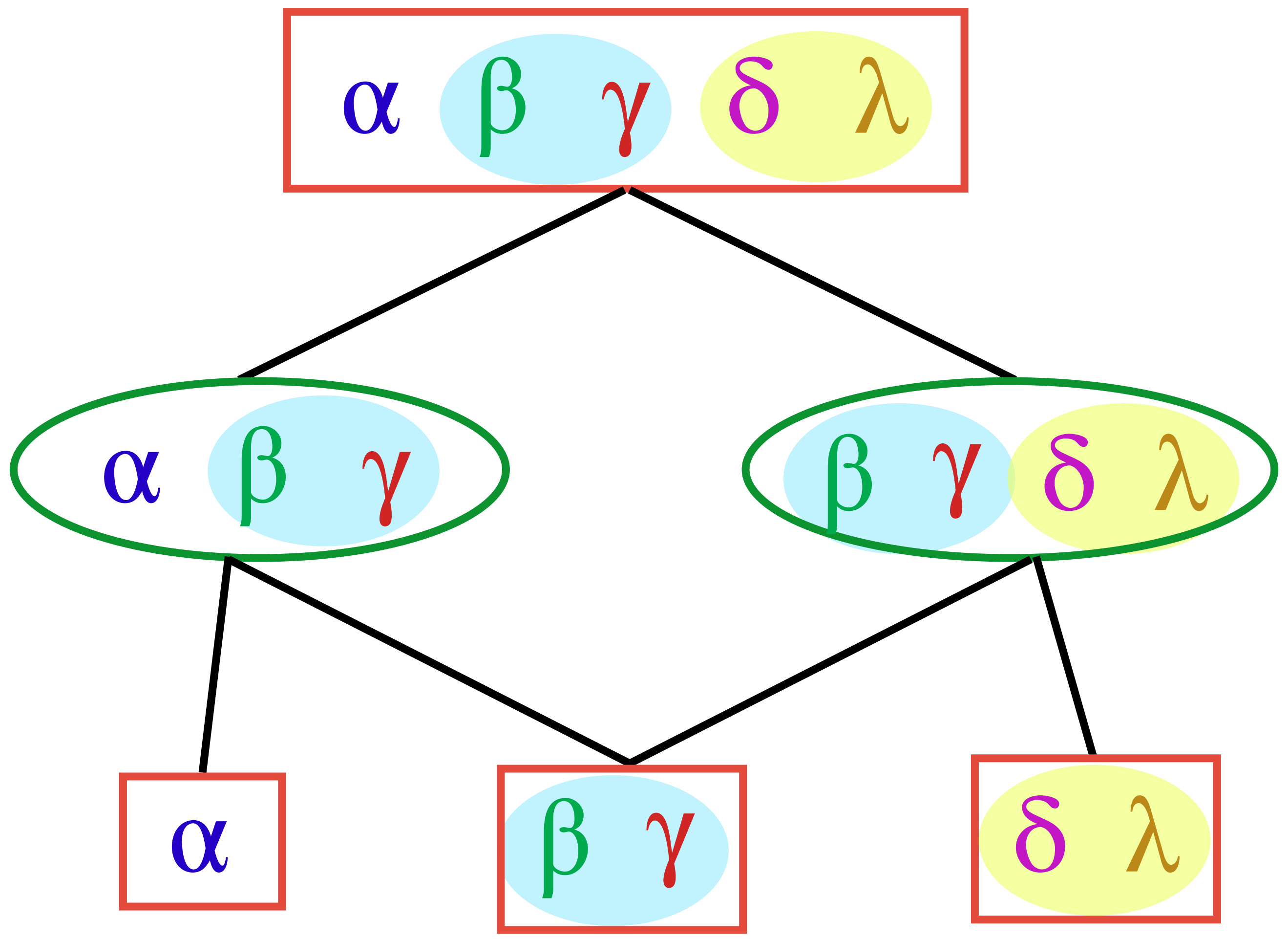
Two colours are equivalent if they appear in the same nodes

Minimisation of colours and Rabin pairs

THEOREM

We can minimise the #Colours (resp. #Rabin pairs) in polynomial time.

Idea (for colours):



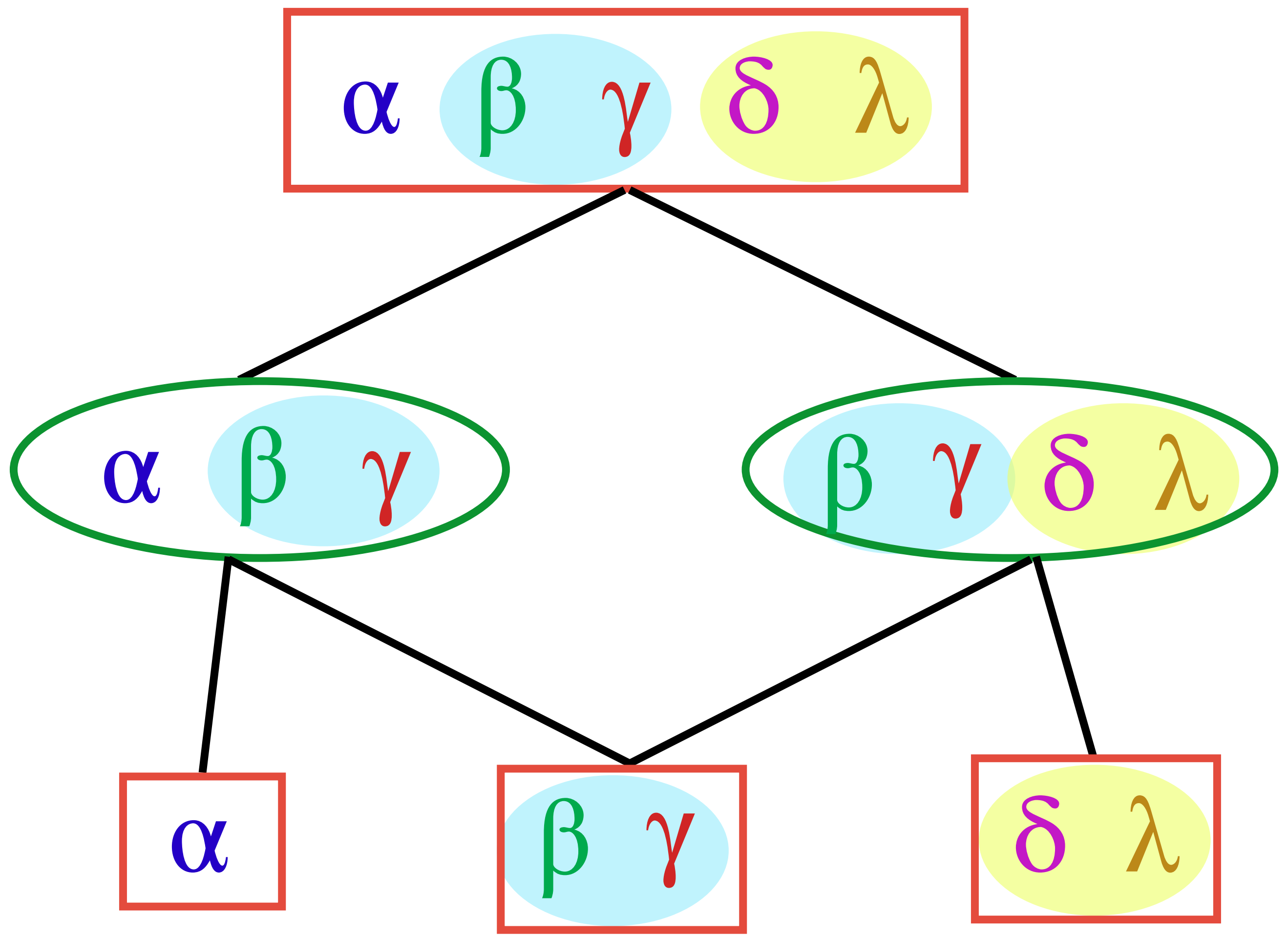
Two colours are equivalent if they appear in the same nodes

Minimisation of colours and Rabin pairs

THEOREM

We can minimise the #Colours (resp. #Rabin pairs) in polynomial time.

Idea (for colours):



Two colours are equivalent if they appear in the same nodes

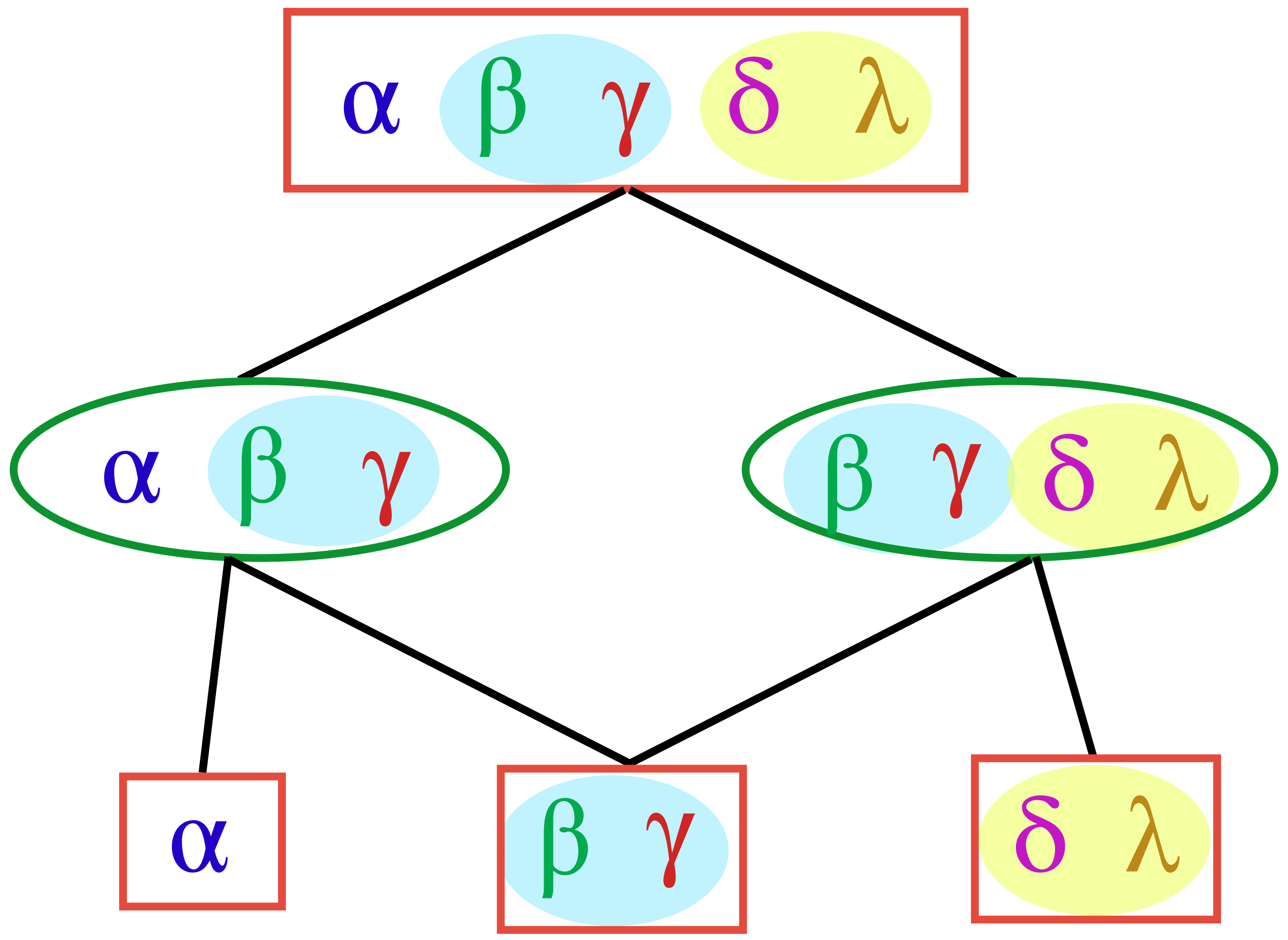
Merge equivalent colours

Minimisation of colours and Rabin pairs

THEOREM

We can minimise the #Colours (resp. #Rabin pairs) in polynomial time.

Idea (for colours):



Two colours are equivalent if they appear in the same nodes

Merge equivalent colours

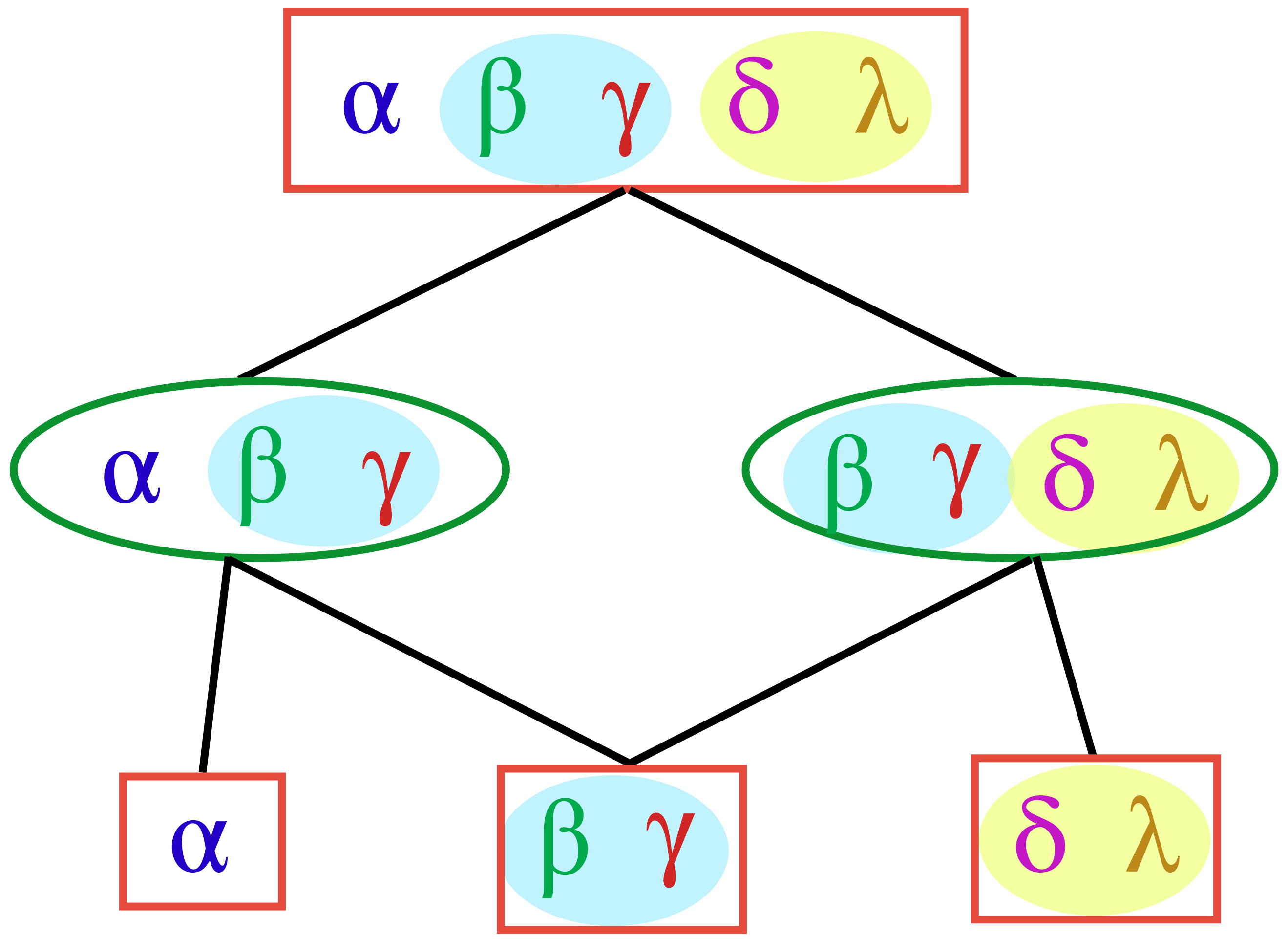
We show that the obtained condition is optimal

Minimisation of colours and Rabin pairs

THEOREM

We can minimise the #Colours (resp. #Rabin pairs) in polynomial time.

Idea (for colours):



Two colours are equivalent if they appear in the same nodes

Merge equivalent colours

We show that the obtained condition is optimal

- Minimisation of Rabin pairs is quite involved.

Pure Conditions

Reduction of conditions

A reduction of a condition $\mathcal{F} \subseteq 2^{C}$ to $\mathcal{G} \subseteq 2^{C'}$ is a mapping

$$\varphi : C \to C' \qquad \text{such that}$$

$$X \in \mathcal{F} \iff \varphi(X) \in \mathcal{G}, \quad \text{for all } X \subseteq C$$

Given a condition $\mathcal{F}$, can we reduce it to

- a Muller condition with $\leq k$ colours?
- a Rabin condition? With $\leq k$ Rabin pairs?
- a parity condition? With $\leq k$ colours?

Main tool: Zielonka DAG *(Zielonka '98, Hunter-Dawar '06)*

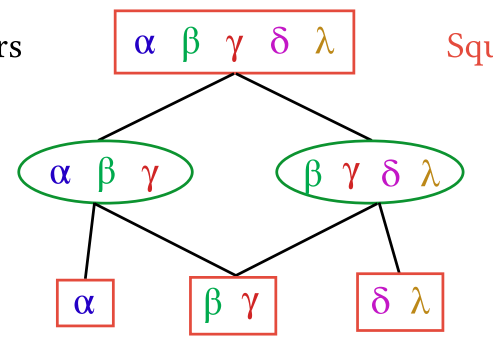
$C = \{\alpha, \beta, \gamma, \delta, \lambda\}$

$\mathcal{F} = \{\{\alpha,\beta,\gamma\}, \{\alpha,\gamma\}, \{\alpha,\beta\}, \{\beta,\gamma,\delta,\lambda\}, \{\beta,\delta,\lambda\}, \{\gamma,\delta,\lambda\}, \{\beta,\gamma,\lambda\}, \{\beta,\gamma,\delta\}, \{\beta,\delta\}, \{\beta,\lambda\}, \{\gamma,\delta\}, \{\gamma,\lambda\}\}$

Root → All colours

Children: Maximal accepting subsets

Children: Maximal rejecting subsets



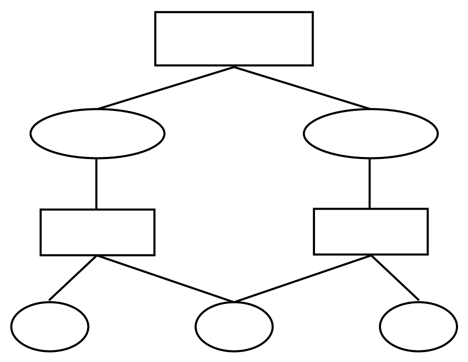
Square if it is a rejecting set

Round because they are accepting sets

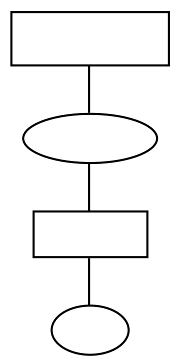
PROPOSITION *(Hunter-Dawar '06)*
The Zielonka DAG can be computed in polynomial time in $|\mathcal{F}|$.

Typeness: Can we reduce $\mathcal{F}$ to a Rabin/parity condition?

Just look to the Zielonka DAG! *(Zielonka '98)*



Rabin
⇔
Round nodes have a single child



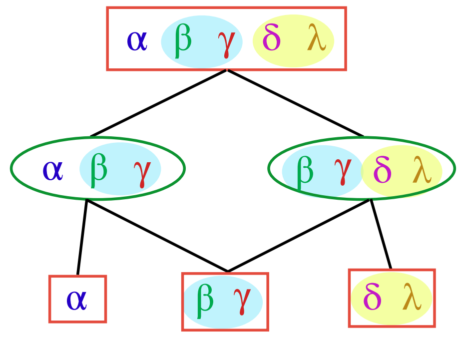
Parity
⇔
A single branch

+ Minimal #Colours
=
Height of the branch

Minimisation of colours and Rabin pairs

THEOREM
We can minimise the #Colours (resp. #Rabin pairs) in polynomial time.

Idea (for colours):



Two colours are equivalent if they appear in the same nodes

Merge equivalent colours

We show that the obtained condition is optimal

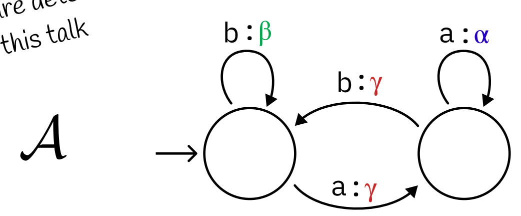
- Minimisation of Rabin pairs is quite involved.

Simplifying ω-Automata through the Alternating Cycle Decomposition

Antonio Casares · University of Warsaw & Corto Mascle · LaBRI, University of Bordeaux

Automata over infinite words (ω-automata)

Automata are deterministic in this talk



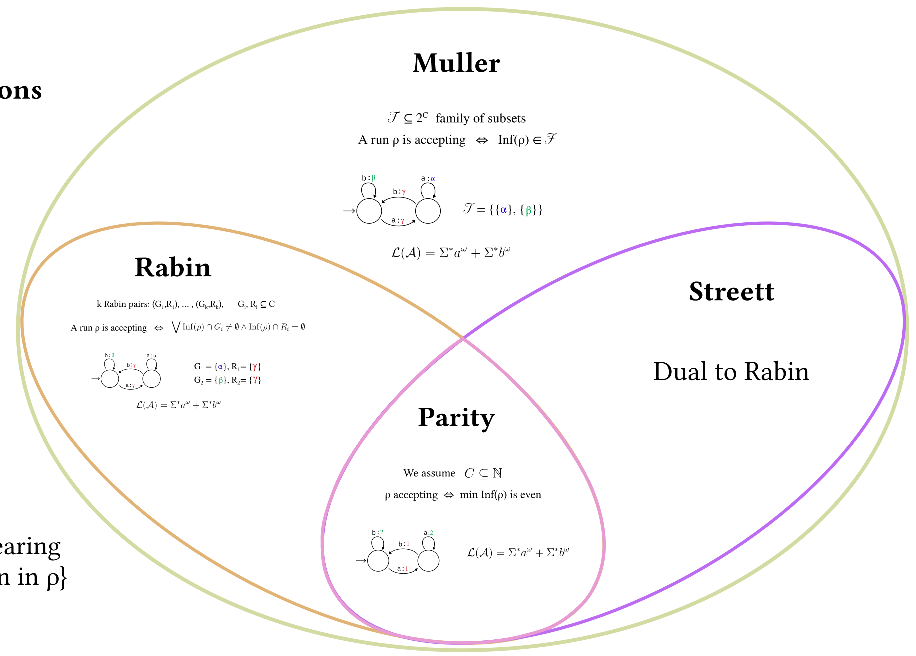
Input $w \in \Sigma^\omega$ ⟶ Infinite path ρ

Different Acceptance Conditions

C = {α, β, γ}

output colours

Inf(ρ) = {Colours appearing infinitely often in ρ}



More general conditions → Automata are simpler to produce

- LTL → Det. Muller automata
- ND Büchi → Det. Rabin automata

Simpler conditions → More efficient algorithms

Complexity of solving 2-player games. n = Size of input graph

Muller	PSPACE-complete	$\mathcal{O}(k^{5k}n^5)$	k = #Colours
Rabin	NP-complete	$\mathcal{O}(n^{k+3}kk!)$	k = #Rabin pair
Parity	NP⋂coNP. In P?	$\mathcal{O}(n^{\log(k)})$	k = #Colours

Goal: Simplify the acceptance condition of automata

a) Simplify the type (Muller → Rabin → Parity)

b) Minimise #Colours / #Rabin pairs

Approach 1

Add states to the automaton to maximize simplification

√ Well understood

✦ Optimal transformation to parity automata *(C. Colcombet, Fijalkow '21)*

✗ Exponential blowup in the worst case *(Löding '99)*

In this paper

Simplify the condition **without** adding further states

Pure Conditions

Reduction of conditions

A reduction of a condition $\mathcal{F} \subseteq 2^C$ to $\mathcal{G} \subseteq 2^{C'}$ is a mapping

$$\varphi : C \longrightarrow C' \quad \text{such that}$$

$$X \in \mathcal{F} \iff \varphi(X) \in \mathcal{G}, \quad \text{for all } X \subseteq C$$

Given a condition $\mathcal{F}$, can we reduce it to

- ✦ a Muller condition with ≤ k colours?
- ✦ a Rabin condition? With ≤ k Rabin pairs?
- ✦ a parity condition? With ≤ k colours?

Main tool: Zielonka DAG *(Zielonka '98, Hunter-Dawar '06)*

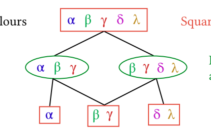
C = {α, β, γ, δ, λ}

$\mathcal{F}$ = {{α,β,γ},{α, γ}, {α, β},{β,γ,δ,λ}, {β,δ,λ}, {γ,δ,λ},{β,γ,λ},{β,γ,δ},{β, δ},{β, λ},{γ,δ},{γ, λ}}

Root → All colours

Children: Maximal accepting subsets

Children: Maximal rejecting subsets



Square if it is a rejecting set

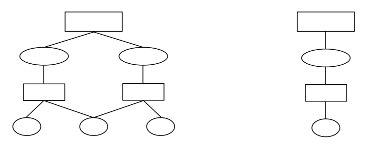
Round because they are accepting sets

Proposition *(Hunter-Dawar '06)*

The Zielonka DAG can be computed in polynomial time in $|\mathcal{F}|$.

Typeness: Can we reduce $\mathcal{F}$ to a Rabin/parity condition?

Just look to the Zielonka DAG! *(Zielonka '98)*



Rabin ⇔ Round nodes have a single child

Parity ⇔ A single branch

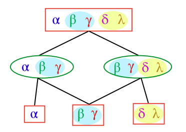
+ Minimal #Colours = Height of the branch

Minimisation of colours and Rabin pairs

Theorem

We can minimise the #Colours (resp. #Rabin pairs) in polynomial time.

Idea (for colours):



Two colours are equivalent if they appear in the same nodes

Merge equivalent colours

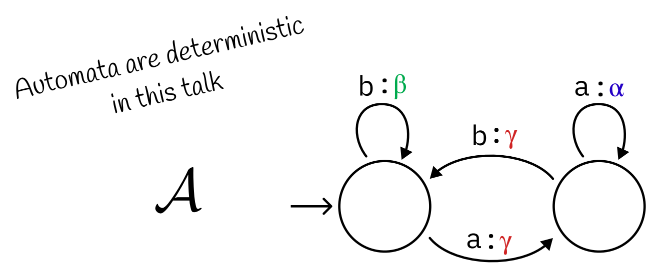
We show that the obtained condition is optimal

- Minimisation of Rabin pairs is quite involved.

Simplifying ω-Automata through the Alternating Cycle Decomposition

Antonio Casares · University of Warsaw & Corto Mascle · LaBRI, University of Bordeaux

Automata over infinite words (ω-automata)

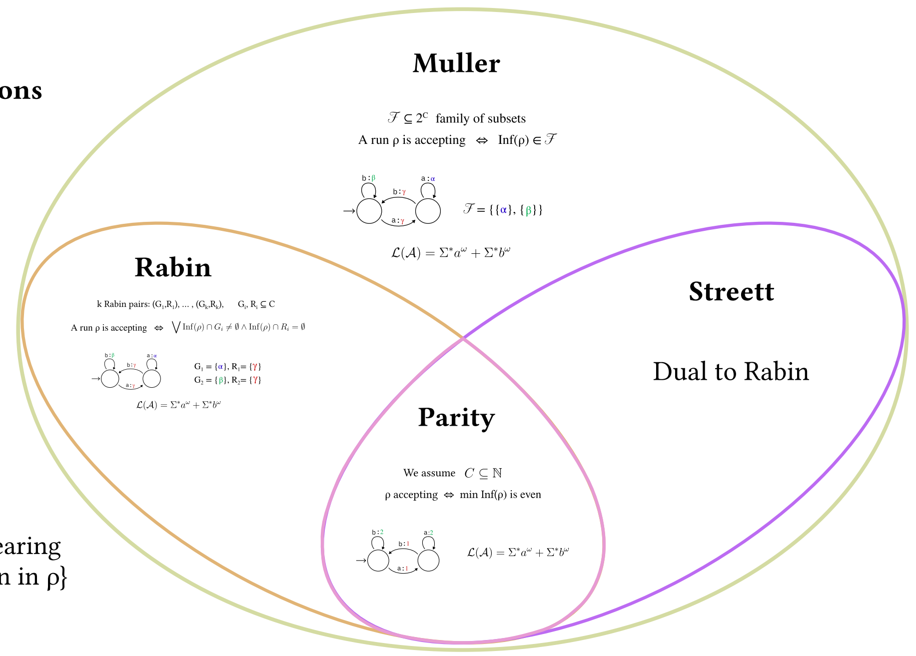


Input $w \in \Sigma^\omega$ ⟶ Infinite path ρ

Different Acceptance Conditions

C = {α, β, γ} output colours

Inf(ρ) = {Colours appearing infinitely often in ρ}



More general conditions → Automata are simpler to produce

- LTL ⟶ Det. Muller automata
- ND Büchi ⟶ Det. Rabin automata

Simpler conditions → More efficient algorithms

Complexity of solving 2-player games. n = Size of input graph

Muller	PSPACE-complete	$\mathcal{O}(k^{5k} n^5)$	k = #Colours
Rabin	NP-complete	$\mathcal{O}(n^{k+3} k k!)$	k = #Rabin pair
Parity	NP∩coNP. In P?	$\mathcal{O}(n^{\log(k)})$	k = #Colours

Goal: Simplify the acceptance condition of automata

a) Simplify the type (Muller → Rabin → Parity)

b) Minimise #Colours / #Rabin pairs

Approach 1

Add states to the automaton to maximize simplification

√ Well understood

✦ Optimal transformation to parity automata *(C. Colcombet, Fijalkow '21)*

✗ Exponential blowup in the worst case *(Löding '99)*

In this paper

Simplify the condition **without** adding further states

Pure Conditions

Reduction of conditions

A reduction of a condition $\mathcal{F} \subseteq 2^C$ to $\mathcal{G} \subseteq 2^{C'}$ is a mapping

$$\varphi : C \longrightarrow C' \quad \text{such that}$$

$$X \in \mathcal{F} \iff \varphi(X) \in \mathcal{G}, \quad \text{for all } X \subseteq C$$

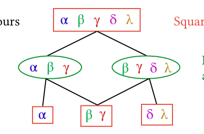
Given a condition $\mathcal{F}$, can we reduce it to

- ✦ a Muller condition with ≤ k colours?
- ✦ a Rabin condition? With ≤ k Rabin pairs?
- ✦ a parity condition? With ≤ k colours?

Main tool: Zielonka DAG *(Zielonka '98, Hunter-Dawar '06)*

C = {α, β, γ, δ, λ}

$\mathcal{F}$ = {{α, β, γ}, {α, γ}, {α, β}, {β, γ, δ, λ}, {β, δ, λ}, {γ, δ, λ}, {β, γ, λ}, {β, γ, δ}, {β, δ}, {β, λ}, {γ, δ}, {γ, λ}}

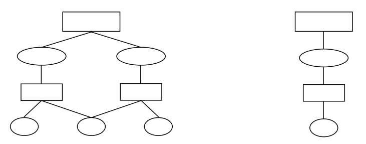


Proposition *(Hunter-Dawar '06)*

The Zielonka DAG can be computed in polynomial time in $|\mathcal{F}|$.

Typeness: Can we reduce $\mathcal{F}$ to a Rabin/parity condition?

Just look to the Zielonka DAG! *(Zielonka '98)*



Rabin ⇔ Round nodes have a single child

Parity ⇔ A single branch

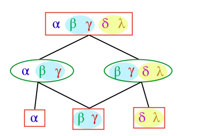
+ Minimal #Colours = Height of the branch

Minimisation of colours and Rabin pairs

Theorem

We can minimise the #Colours (resp. #Rabin pairs) in polynomial time.

Idea (for colours):



Two colours are equivalent if they appear in the same nodes

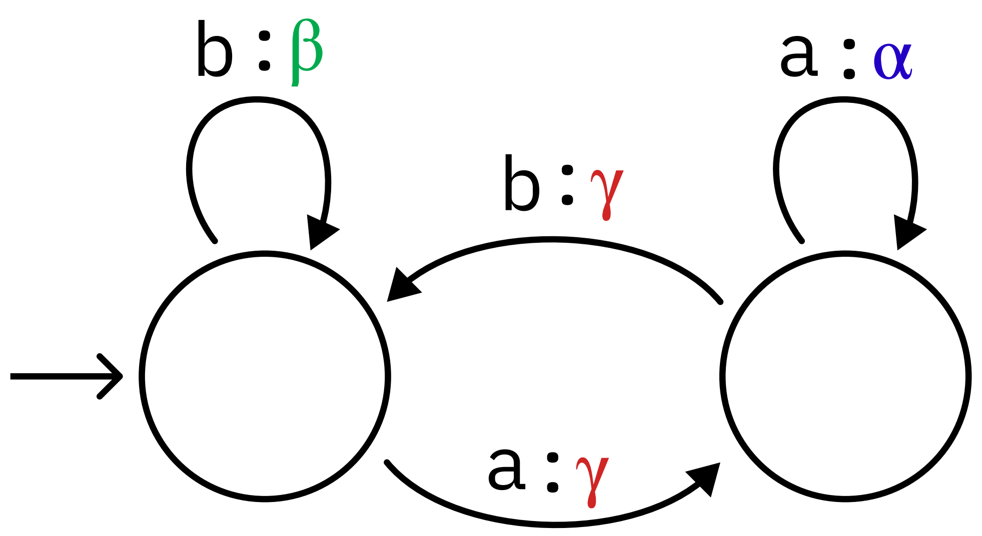
Merge equivalent colours

We show that the obtained condition is optimal

- Minimisation of Rabin pairs is quite involved.

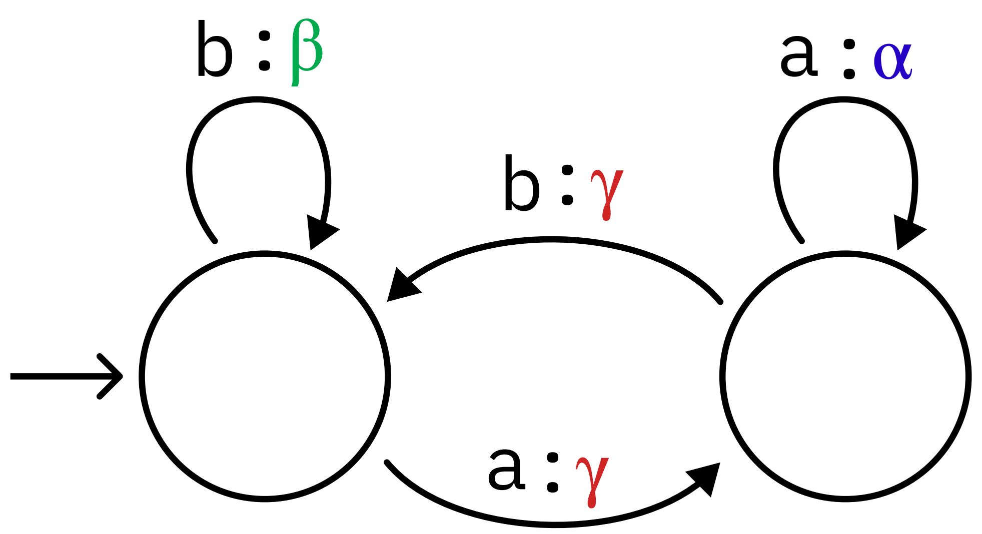
Automata

Automata



$$\mathscr{F} = \{\{\alpha\}, \{\beta\}\}$$

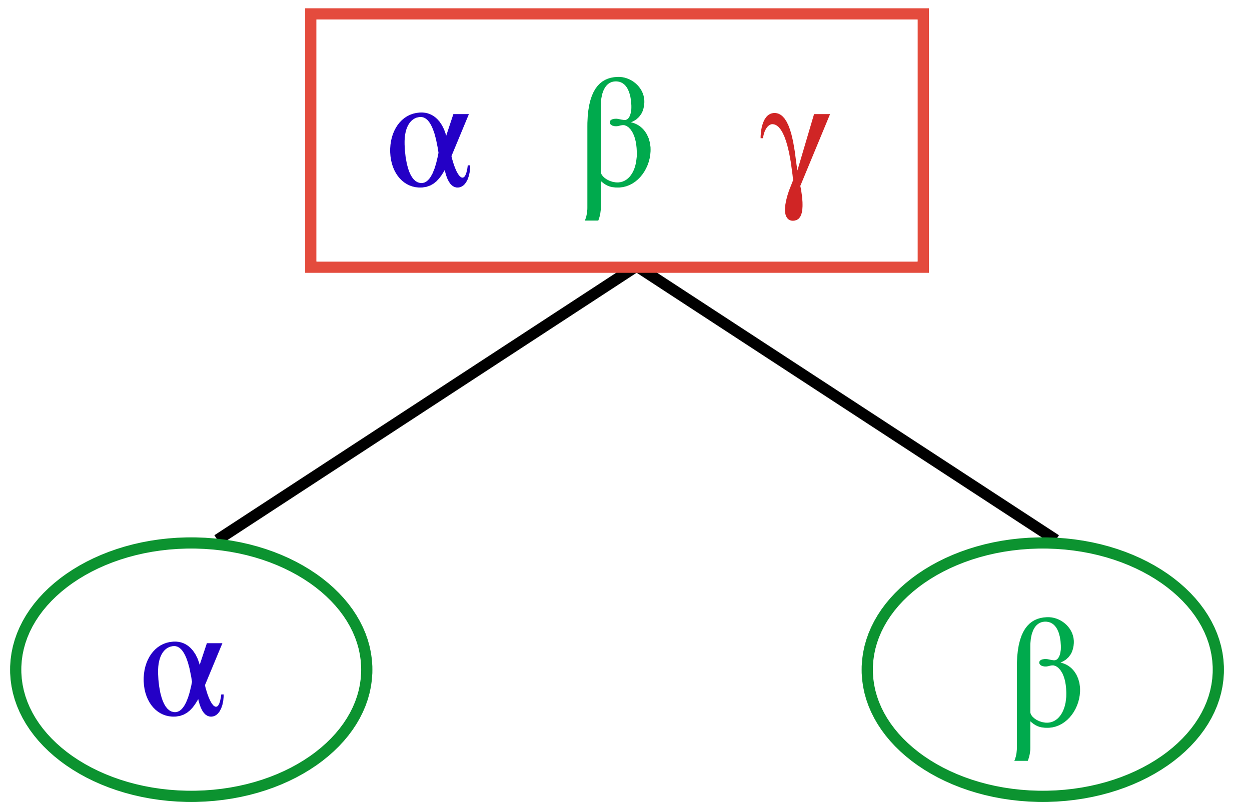
Automata

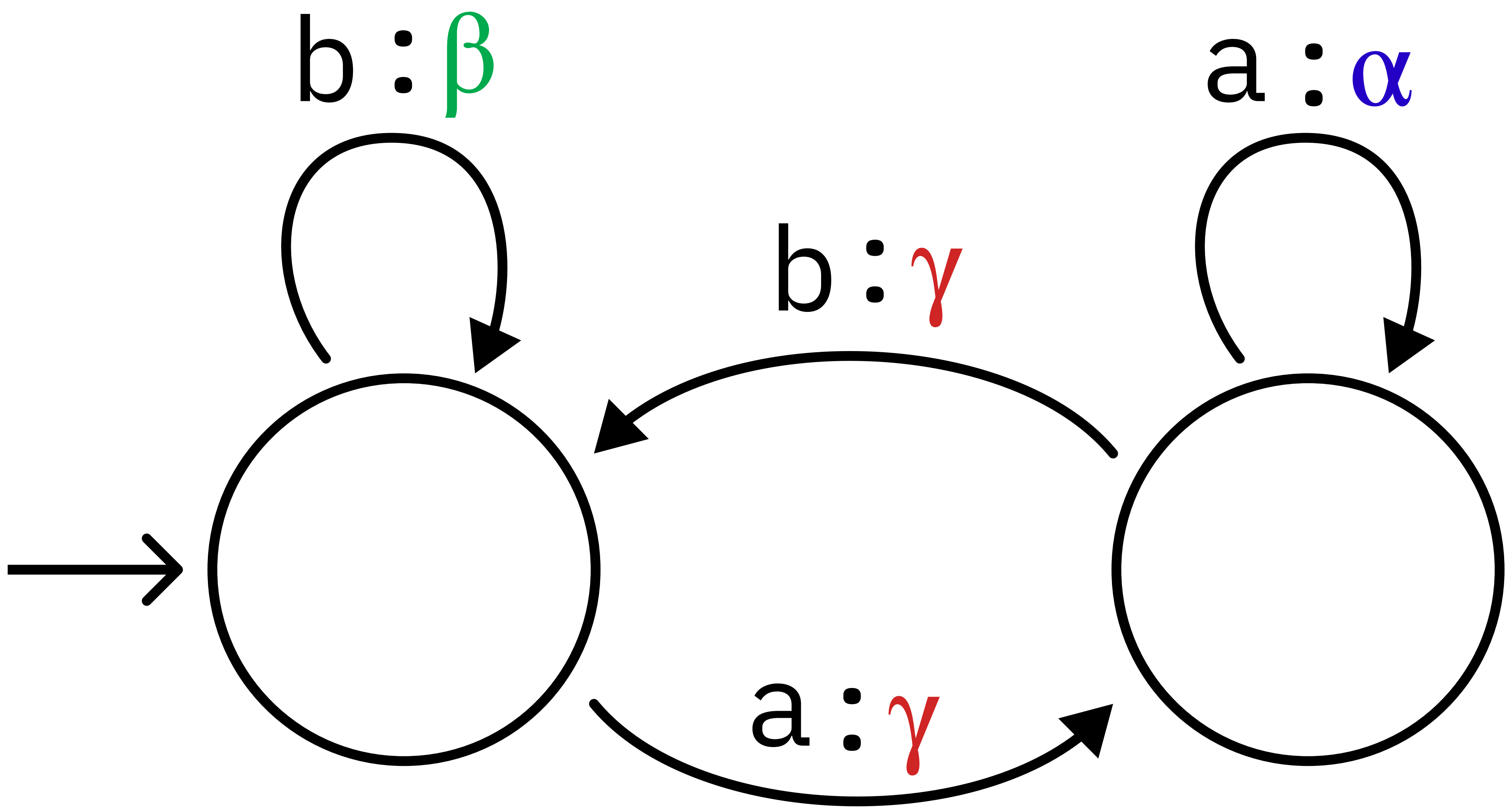


$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

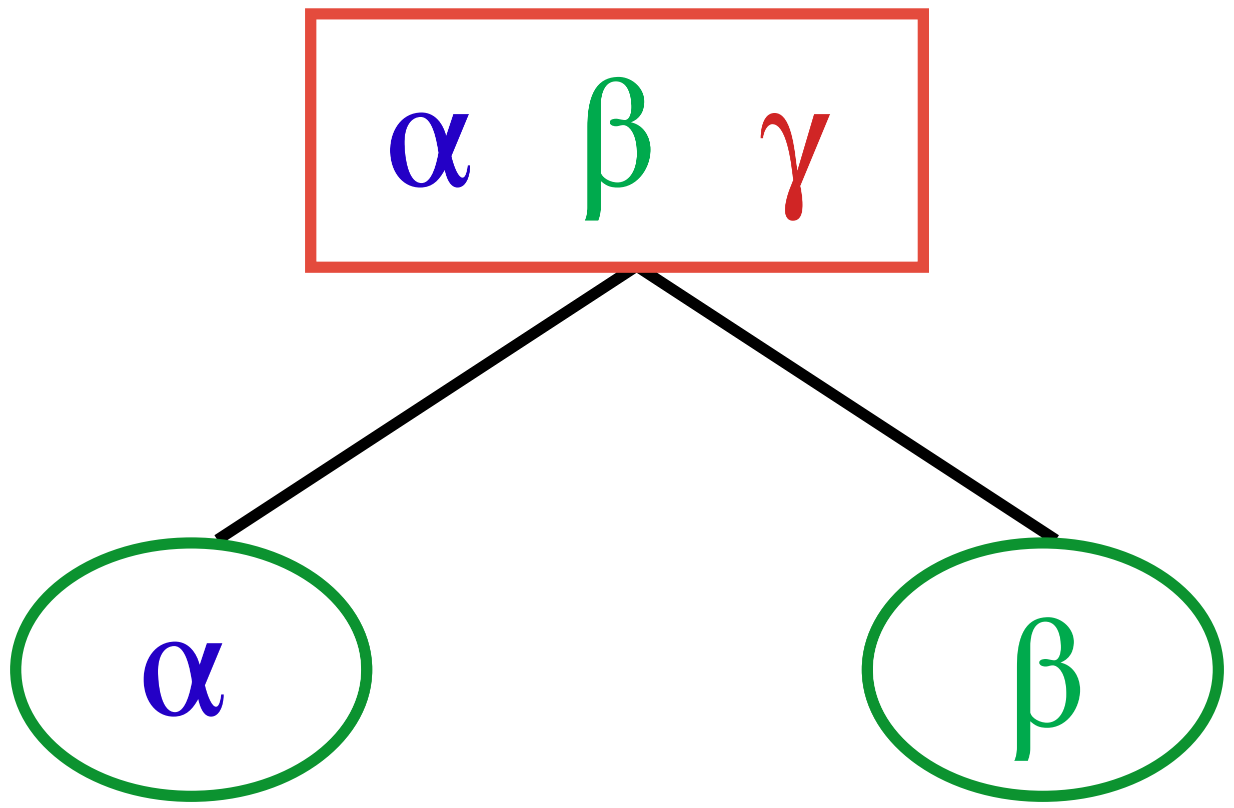
Can we relabel $\mathcal{A}$ with

- a parity/Rabin condition?
- $\leq$ k colours / Rabin pairs?



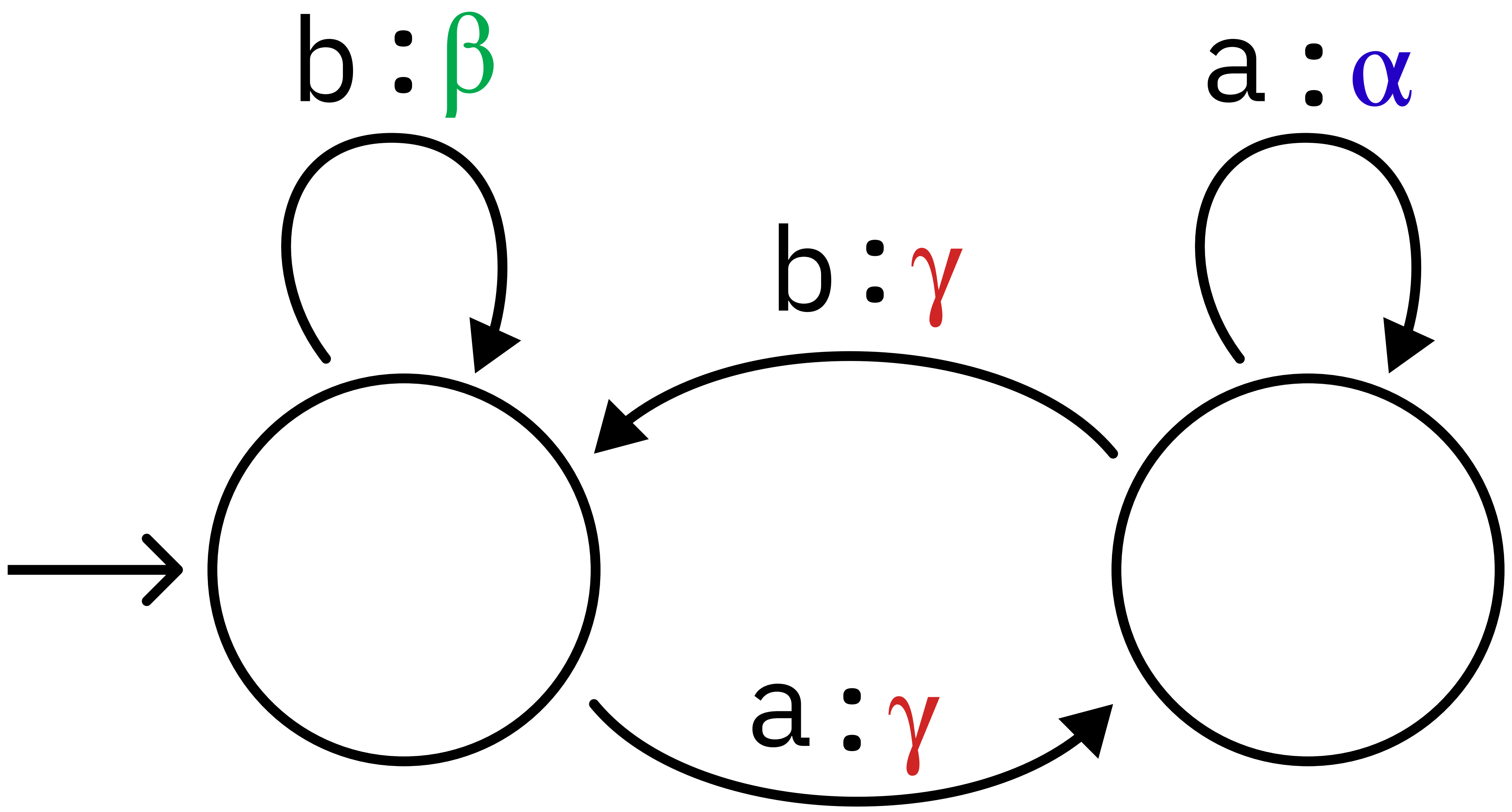


$$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$$

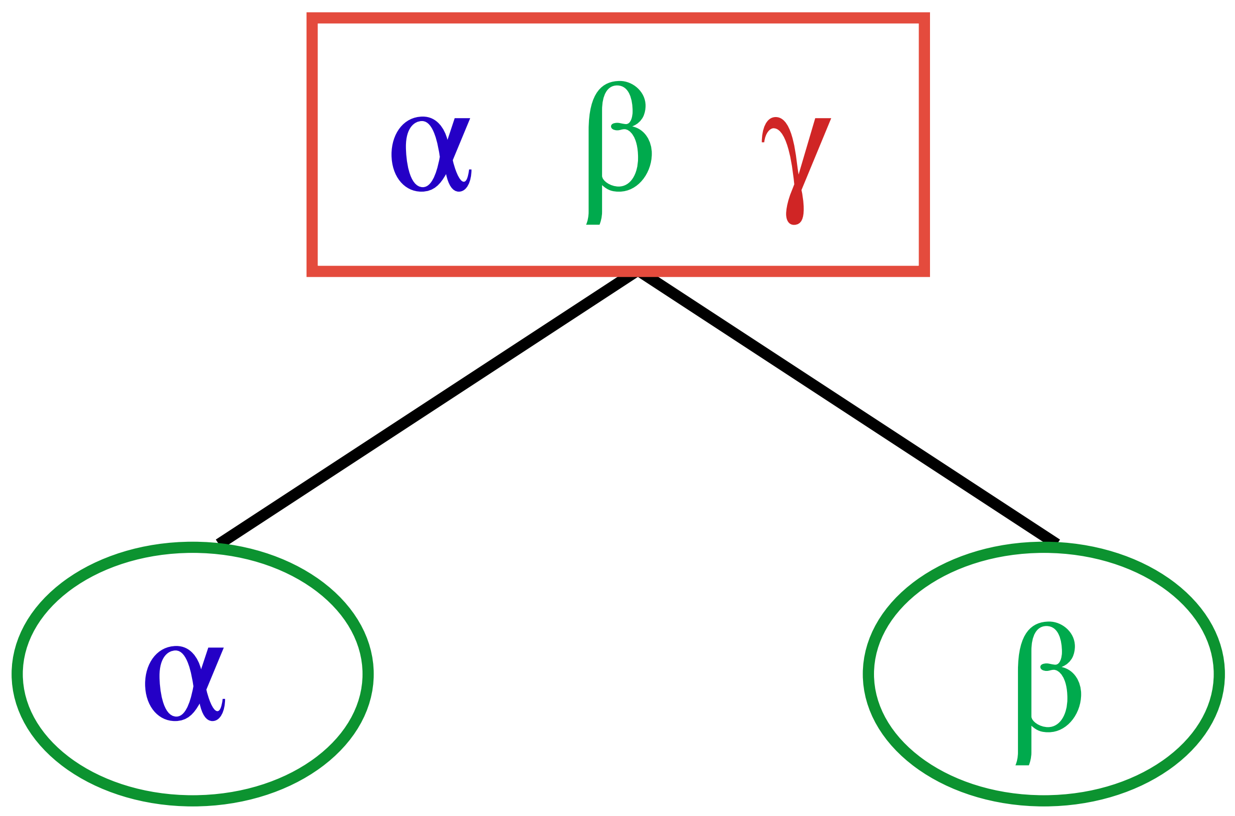


Not parity

Requires 3 colours

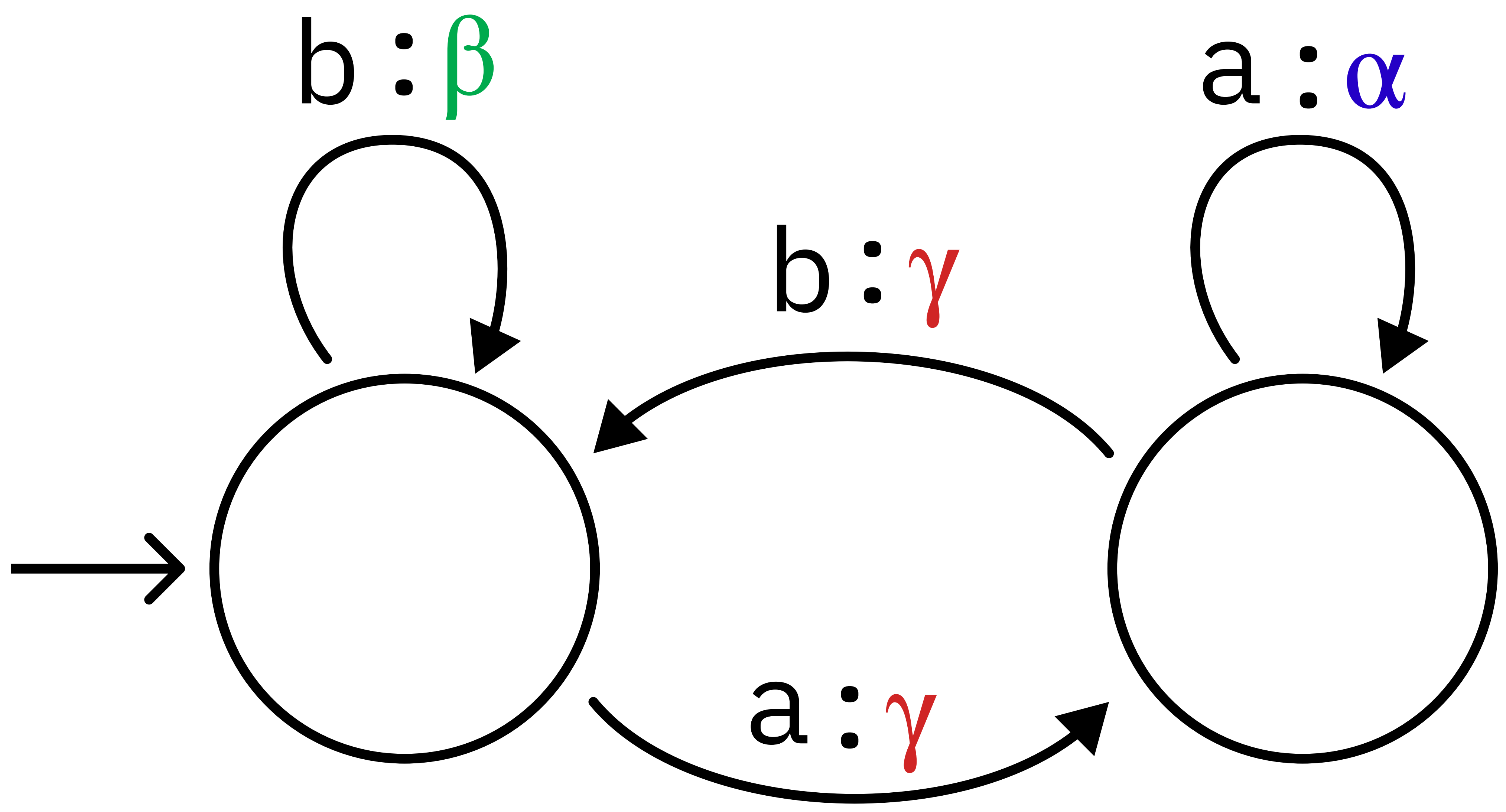


$$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$$



Not parity

Requires 3 colours

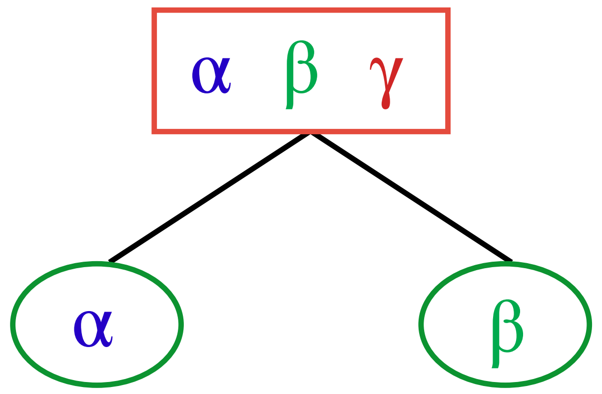


$$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$$

But we can relabel the automaton with

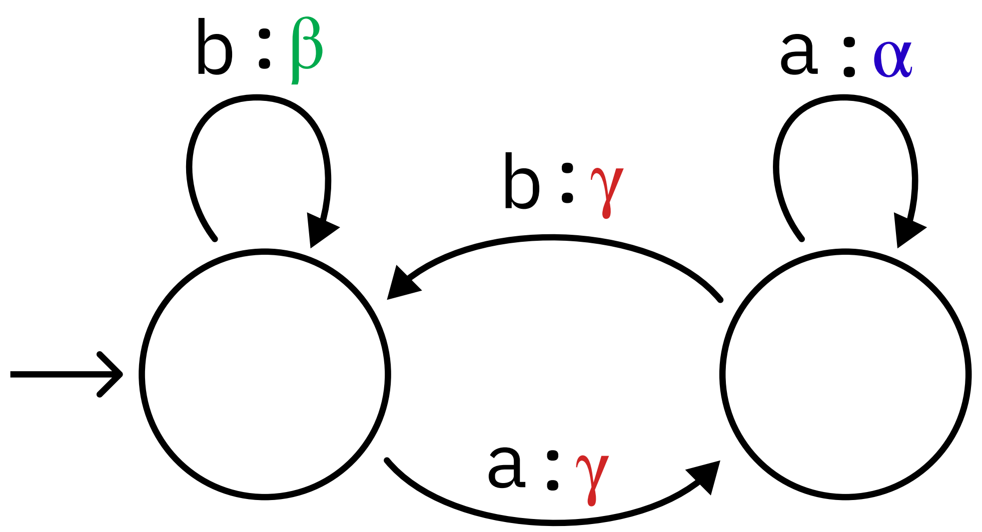
a parity condition with 2 colours!

Automata



Not parity

Requires 3 colours



$$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$$

But we can relabel the automaton with
a parity condition with 2 colours!

Can we relabel $\mathcal{A}$ with

- a parity/Rabin condition?
- $\leq$ k colours / Rabin pairs?

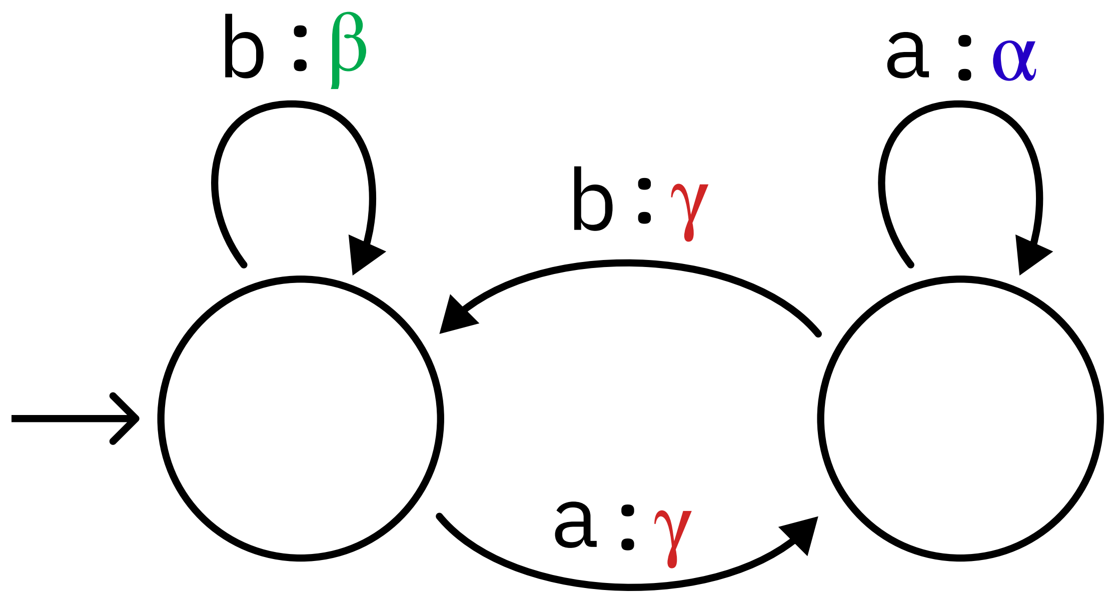
Main tool: Alternating Cycle Decomposition (ACD) *(C. Colcombet Fijalkow '21)*

Main tool: Alternating Cycle Decomposition (ACD) *(C. Colcombet Fijalkow '21)*

Generalisation of the Zielonka DAG

Main tool: Alternating Cycle Decomposition (ACD) *(C. Colcombet Fijalkow '21)*

Generalisation of the Zielonka DAG

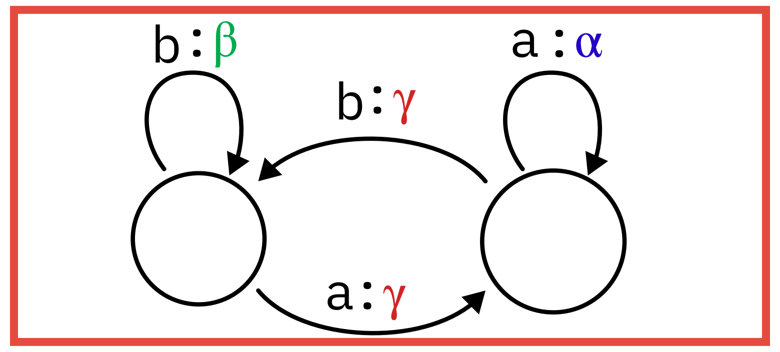


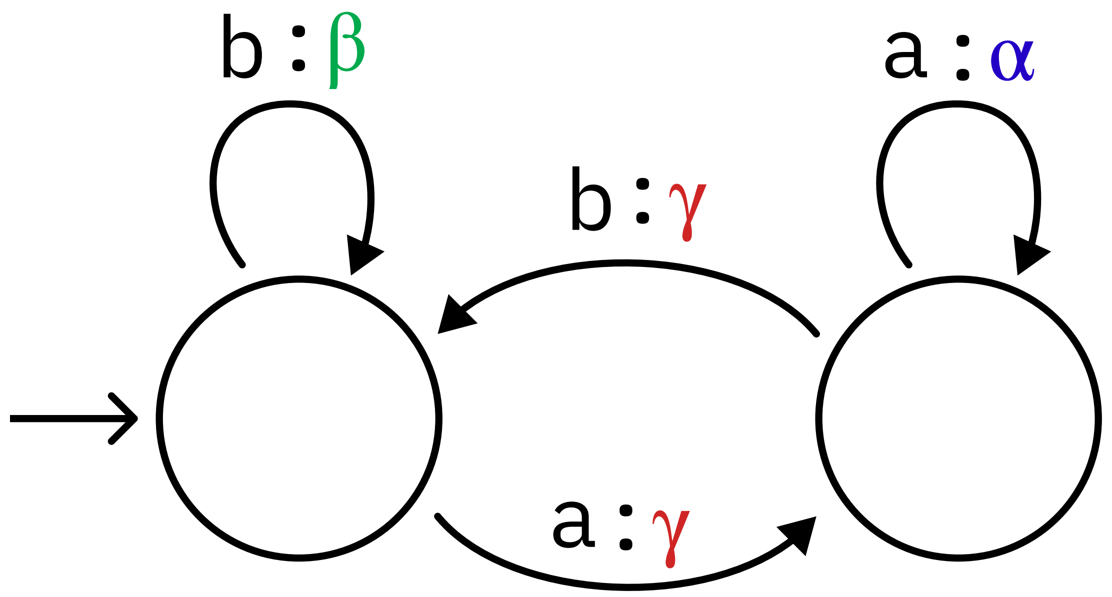
$$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$$

Main tool: Alternating Cycle Decomposition (ACD) *(C. Colcombet Fijalkow '21)*

Generalisation of the Zielonka DAG

Root $\longrightarrow$ All edges



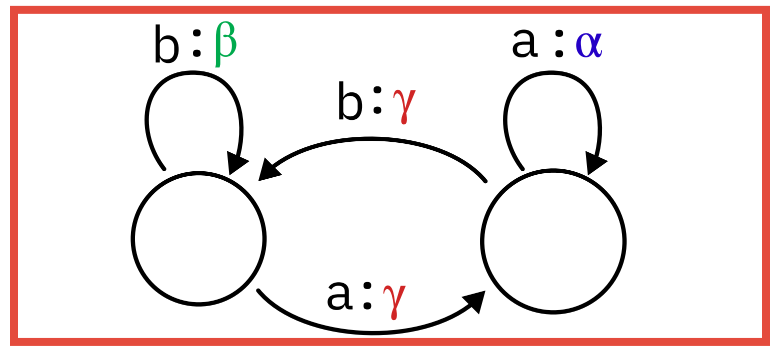


$$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$$

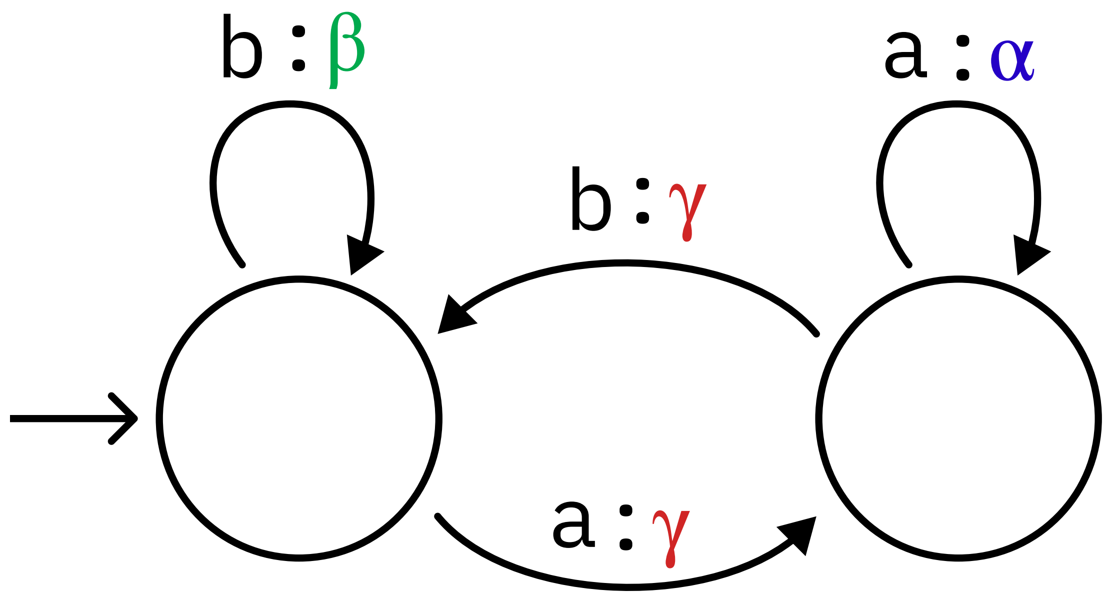
Main tool: Alternating Cycle Decomposition (ACD) *(C. Colcombet Fijalkow '21)*

Generalisation of the Zielonka DAG

Root $\longrightarrow$ All edges



Square if they form a rejecting cycle



$$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$$

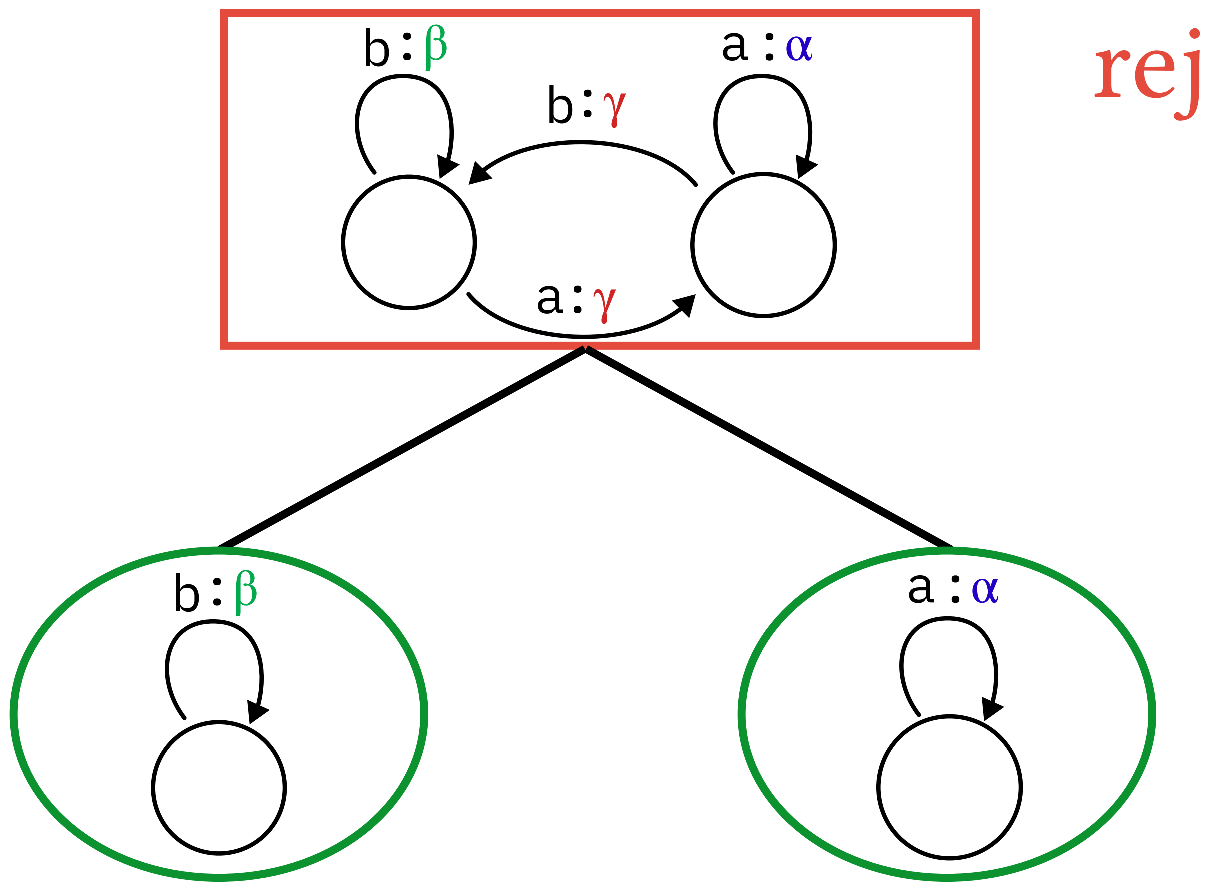
Main tool: Alternating Cycle Decomposition (ACD) *(C. Colcombet Fijalkow '21)*

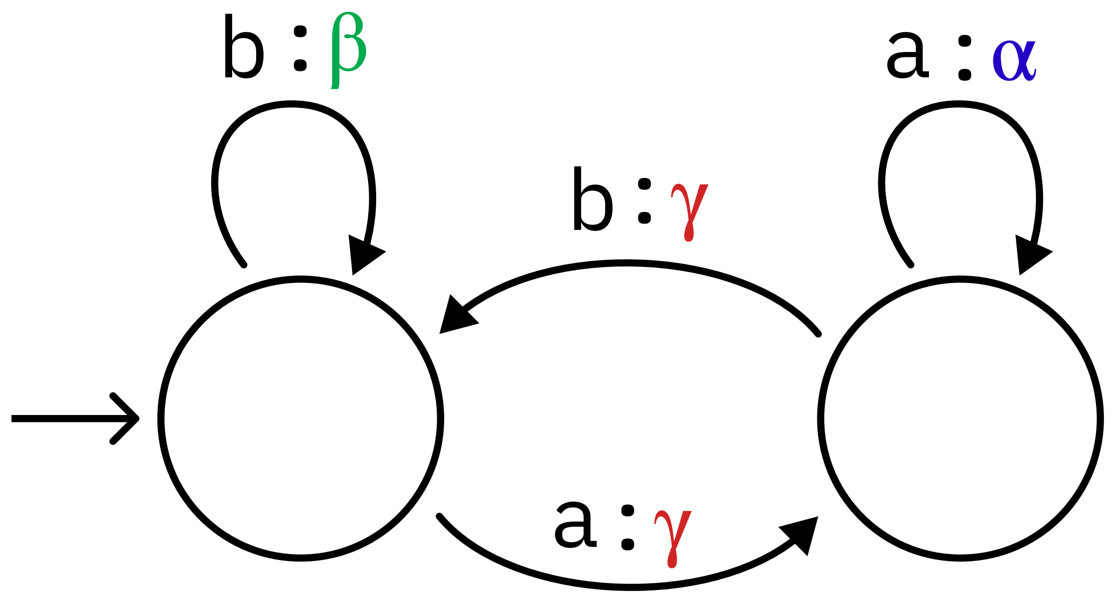
Generalisation of the Zielonka DAG

Root $\longrightarrow$ All edges

Square if they form a rejecting cycle

Children: Maximal accepting subcycles



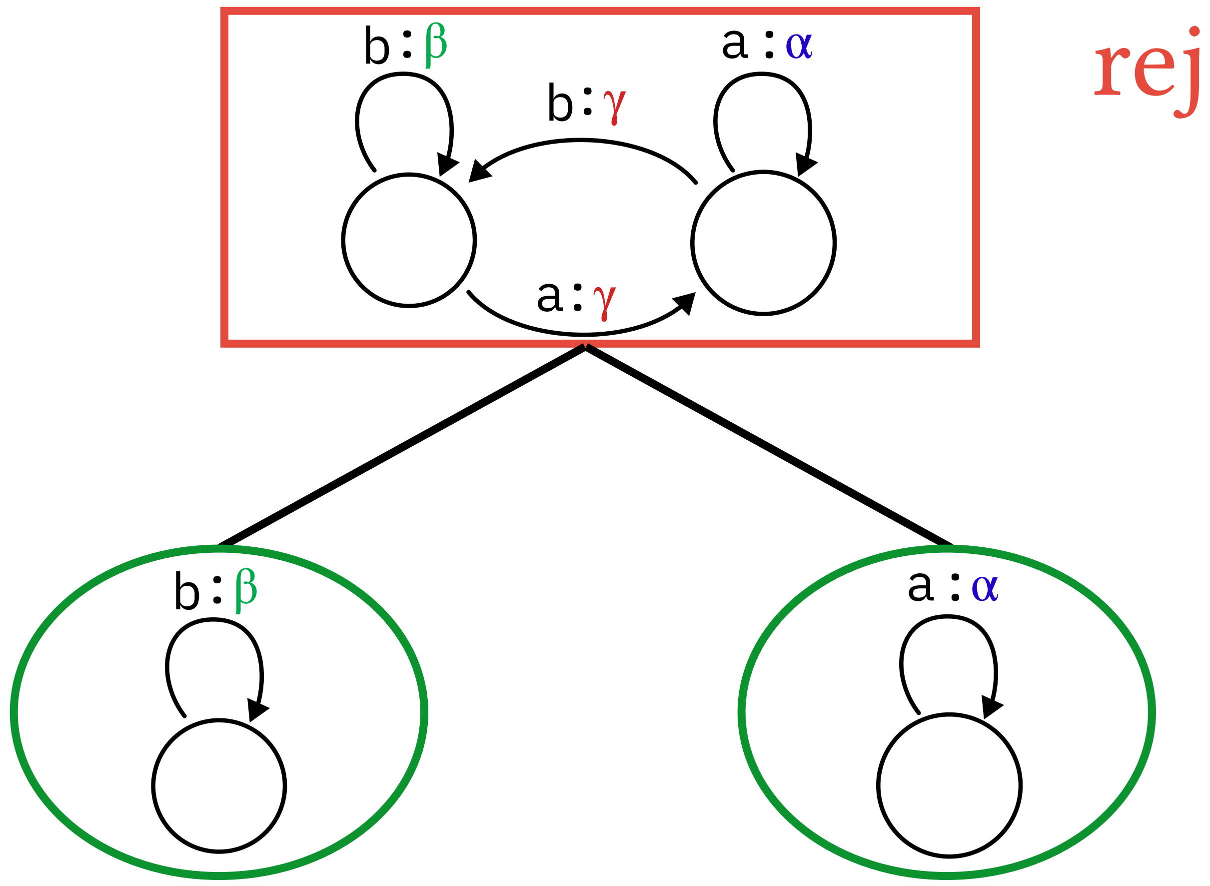


$$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$$

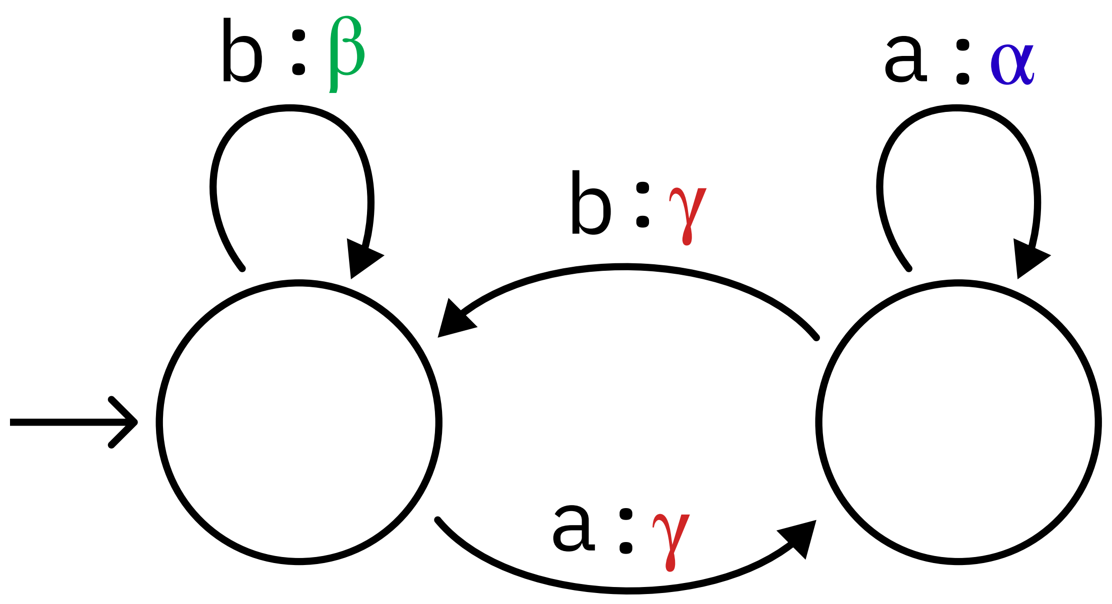
Main tool: Alternating Cycle Decomposition (ACD) *(C. Colcombet Fijalkow '21)*

Generalisation of the Zielonka DAG

Root $\longrightarrow$ All edges



Square if they form a rejecting cycle



$$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$$

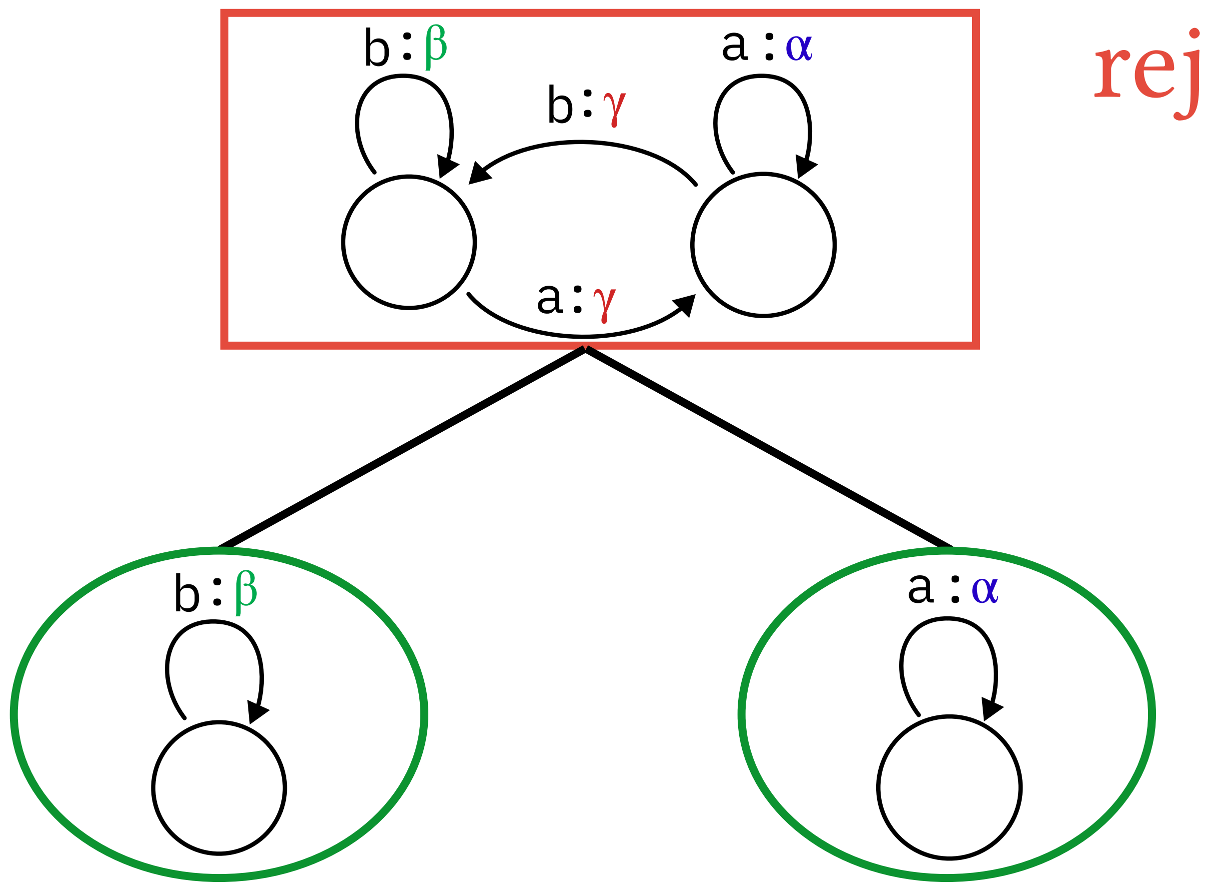
Children: Maximal accepting subcycles

Round because they are accepting

Main tool: Alternating Cycle Decomposition (ACD) *(C. Colcombet Fijalkow '21)*

Generalisation of the Zielonka DAG

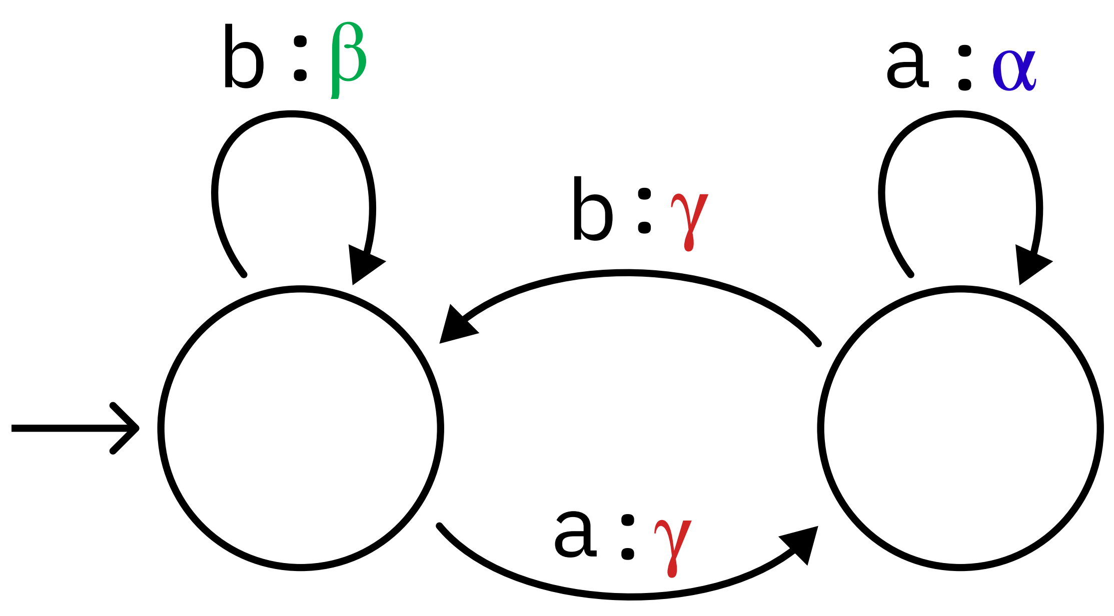
Root $\longrightarrow$ All edges



Children: Maximal accepting subcycles

Square if they form a rejecting cycle

Round because they are accepting



$$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$$

THEOREM (main technical contribution)

The Alternating Cycle Decomposition can be computed in **polynomial time** in $|\mathcal{A}| + |\mathcal{F}|$.

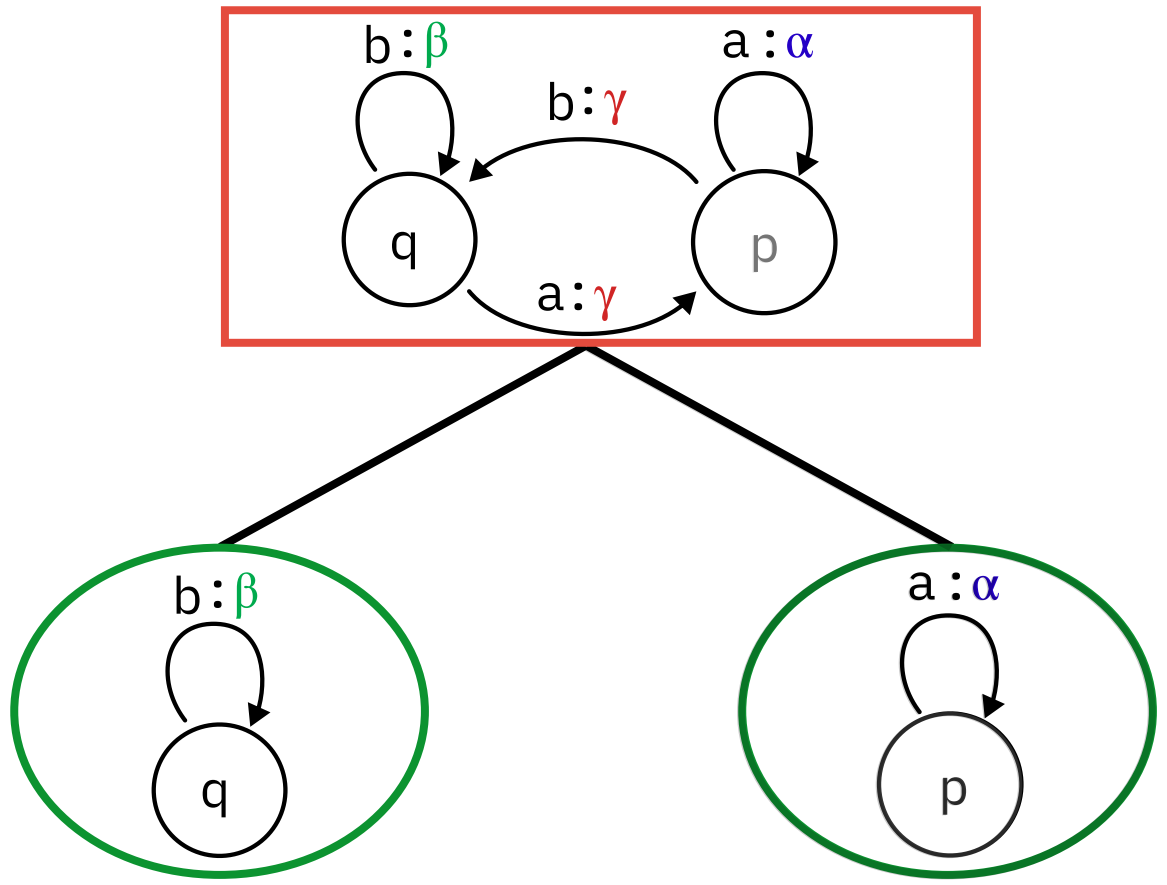
Typeness: Can we relabel $\mathcal{A}$ with a Rabin/parity condition?

Typeness: Can we relabel $\mathcal{A}$ with a Rabin/parity condition?

Just look to the Altenating Cycle Decompositon! *(C. Colcombet Fijalkow '21)*

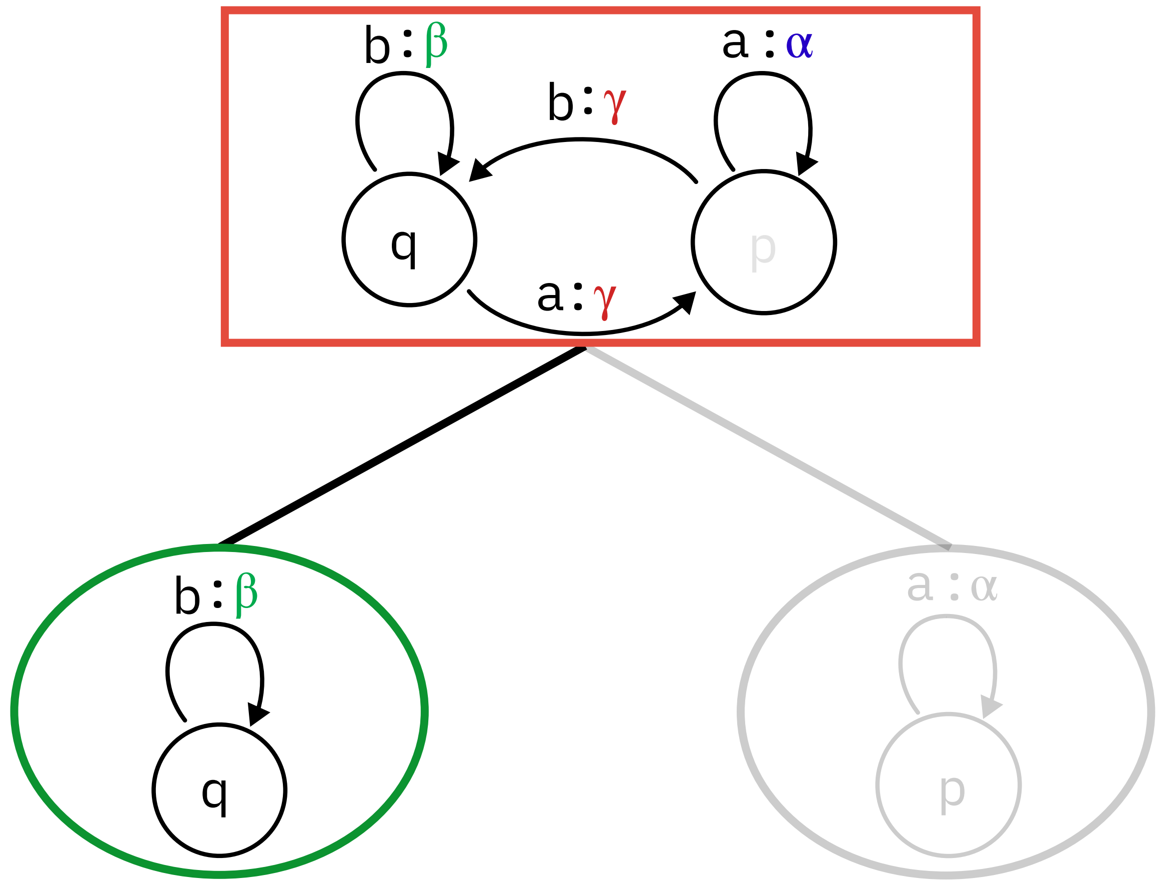
Typeness: Can we relabel $\mathcal{A}$ with a Rabin/parity condition?

Just look to the Altenating Cycle Decompositon! *(C. Colcombet Fijalkow '21)*



Typeness: Can we relabel $\mathcal{A}$ with a Rabin/parity condition?

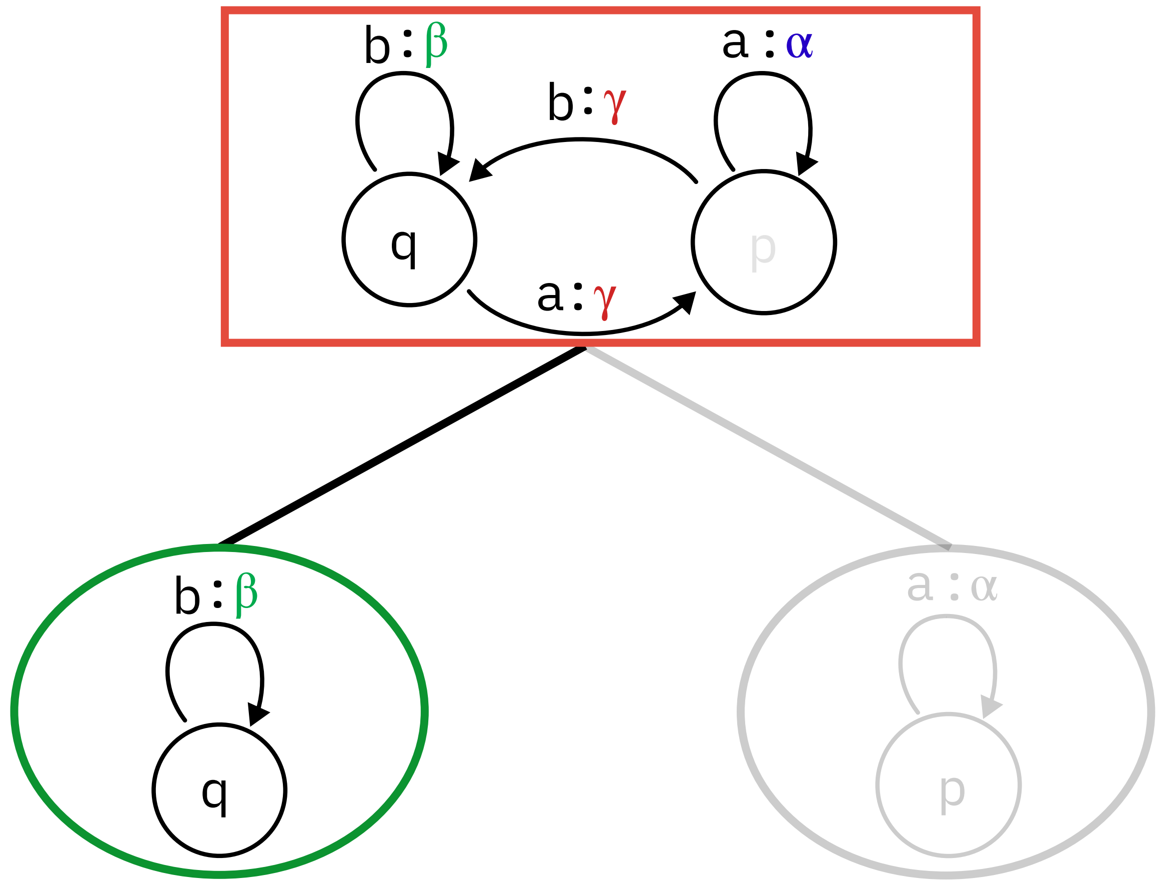
Just look to the Altenating Cycle Decompositon! *(C. Colcombet Fijalkow '21)*



Restriction to the tree of a state q

Typeness: Can we relabel $\mathcal{A}$ with a Rabin/parity condition?

Just look to the Altenating Cycle Decompositon! *(C. Colcombet Fijalkow '21)*



Restriction to the tree of a state q

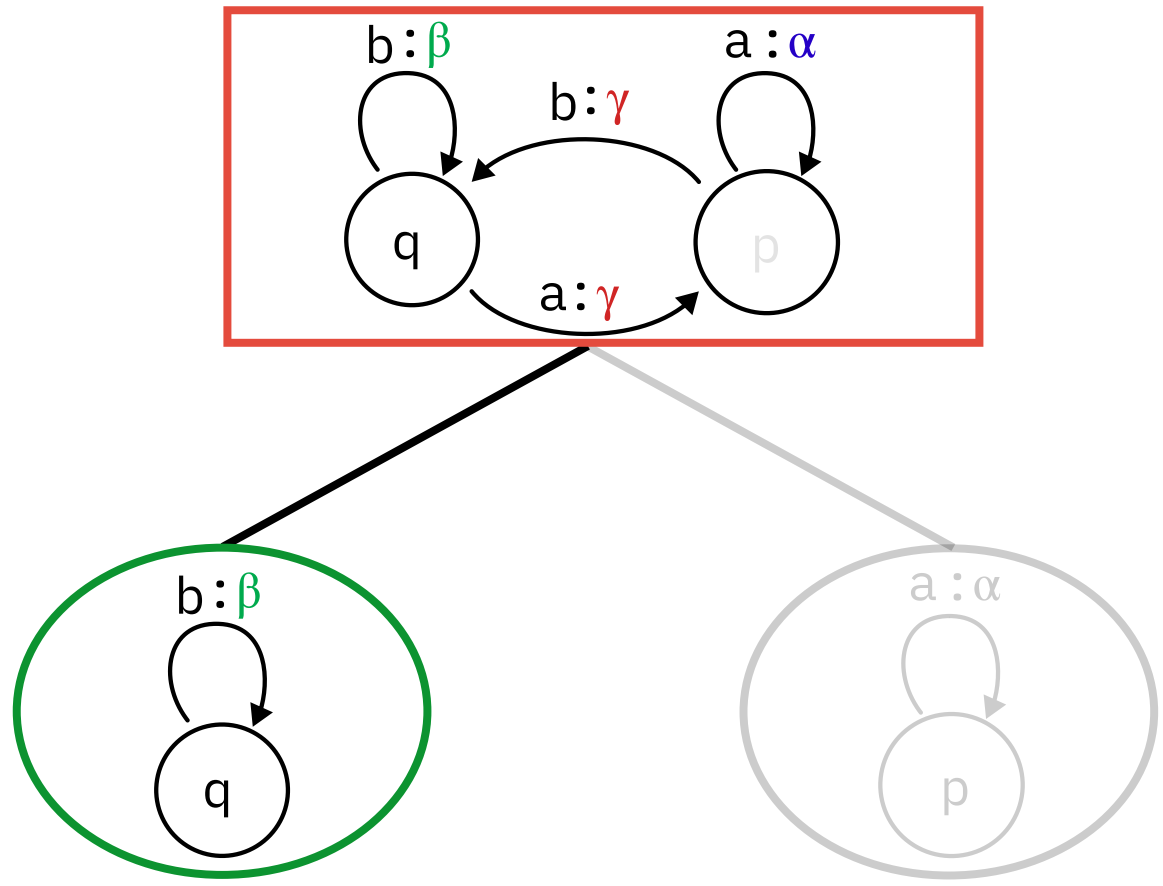
Rabin

$\Leftrightarrow$

In the restriction-tree of every state round nodes have a single child

Typeness: Can we relabel $\mathcal{A}$ with a Rabin/parity condition?

Just look to the Altenating Cycle Decompositon! *(C. Colcombet Fijalkow '21)*



Restriction to the tree of a state q

Rabin

$\Leftrightarrow$

In the restriction-tree of every state round nodes have a single child

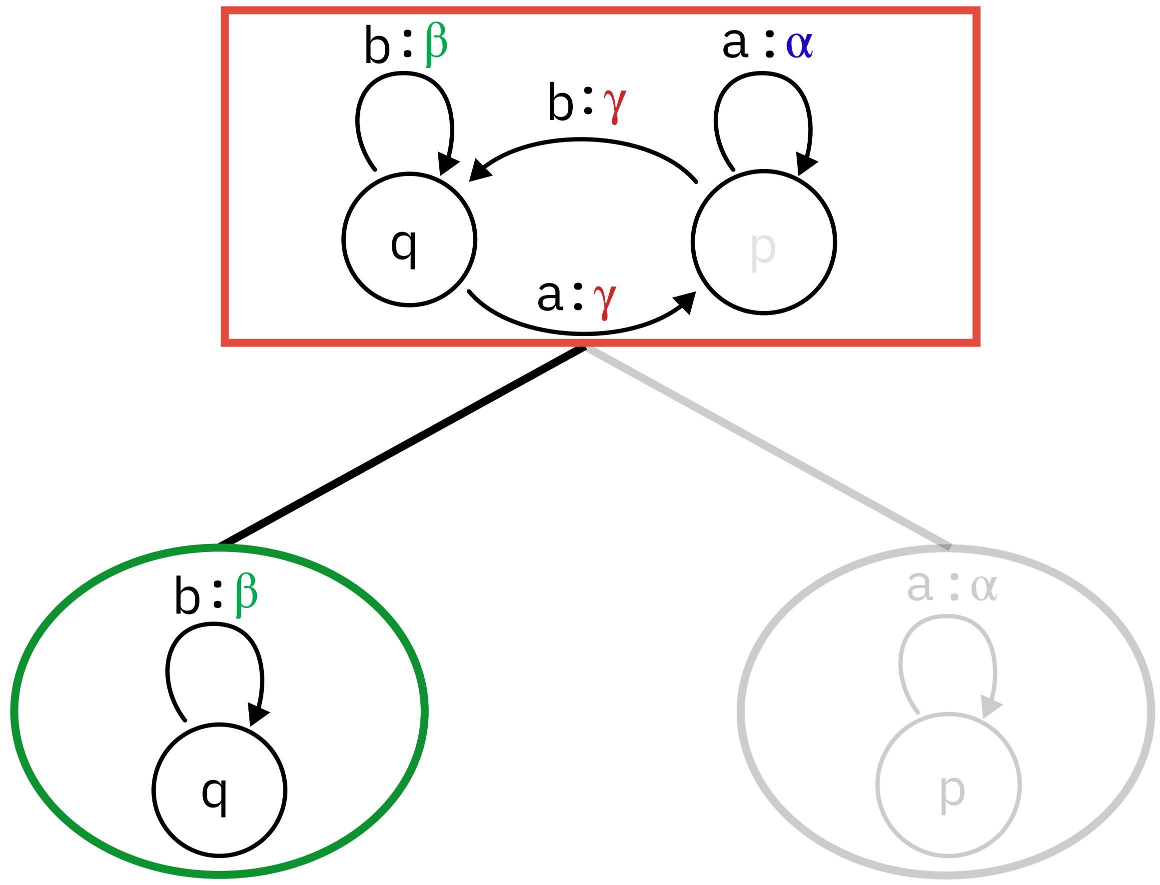
Parity

$\Leftrightarrow$

The restriction-tree of every state consists of a single branch.

Typeness: Can we relabel $\mathcal{A}$ with a Rabin/parity condition?

Just look to the Altenating Cycle Decompositon! *(C. Colcombet Fijalkow '21)*



Restriction to the tree of a state q

Rabin

$\Leftrightarrow$

In the restriction-tree of every state round nodes have a single child

Parity

$\Leftrightarrow$

The restriction-tree of every state consists of a single branch.

+ Minimal #Colours

=

Height of the ACD

Minimization of colours and Rabin pairs

Minimization of colours and Rabin pairs

First idea: Do as with the Zielonka DAG!

Minimization of colours and Rabin pairs

First idea: Do as with the Zielonka DAG!

But... It does not work!

😢

Minimization of colours and Rabin pairs

First idea: Do as with the Zielonka DAG!

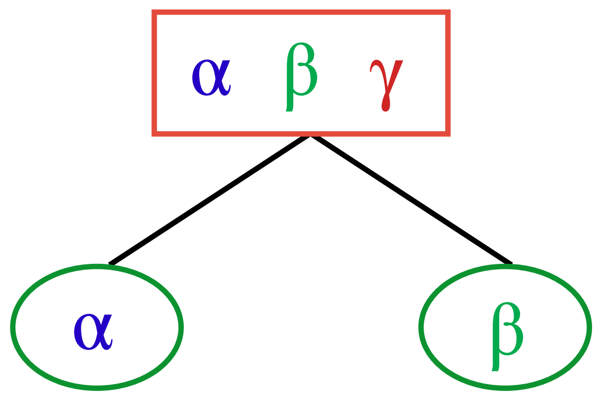
But... It does not work!

THEOREM

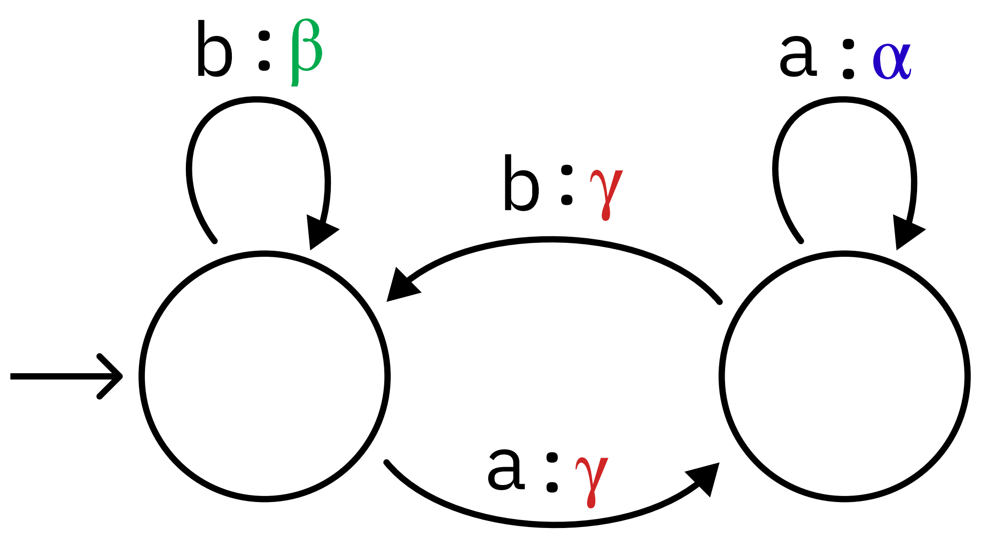
Given an automaton $\mathcal{A}$ and its ACD, it is **NP-complete** to determine if we can relabel it with a condition using:

- $\leq$ k colours.
- $\leq$ k Rabin pairs.

Automata



Not parity
Requires 3 colours



$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

Can we relabel $\mathcal{A}$ with

- a parity/Rabin condition?
- $\leq$ k colours / Rabin pairs?

But we can relabel the automaton with
a parity condition with 2 colours!

Main tool: Alternating Cycle Decomposition (ACD) *(C. Colcombet Fijalkow '21)*

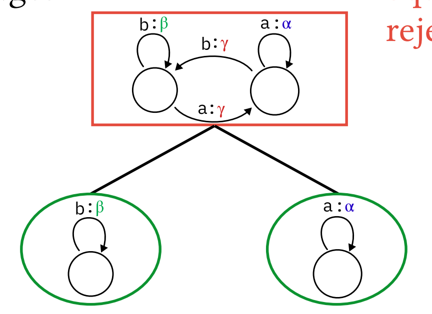
Generalisation of the Zielonka DAG

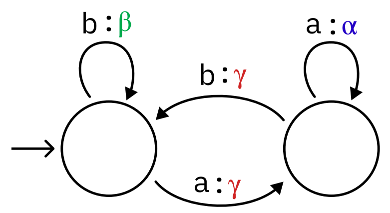
Root → All edges

Square if they form a rejecting cycle

Children: Maximal accepting subcycles

Round because they are accepting





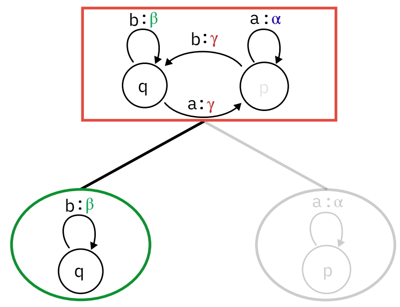
$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

THEOREM (main technical contribution)

The Alternating Cycle Decomposition can be computed in **polynomial time** in $|\mathcal{A}| + |\mathcal{F}|$.

Typeness: Can we relabel $\mathcal{A}$ with a Rabin/parity condition?

Just look to the Altenating Cycle Decompositon! *(C. Colcombet Fijalkow '21)*



Restriction to the tree of a state q

Rabin
⇔
In the restriction-tree of every state
round nodes have a single child

Parity
⇔
The restriction-tree of every state
consists of a single branch.

+ Minimal #Colours
=
Height of the ACD

Minimization of colours and Rabin pairs

First idea: Do as with the Zielonka DAG!

But... It does not work!

THEOREM

Given an automaton $\mathcal{A}$ and its ACD, it is **NP-complete** to determine if we can relabel it with a condition using:

- $\leq$ k colours.
- $\leq$ k Rabin pairs.

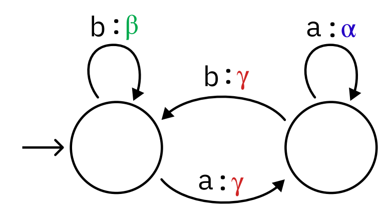
Simplifying ω-Automata through the Alternating Cycle Decomposition

Antonio Casares · University of Warsaw & Corto Mascle · LaBRI, University of Bordeaux

Automata over infinite words (ω-automata)

Automata are deterministic in this talk

$\mathcal{A}$



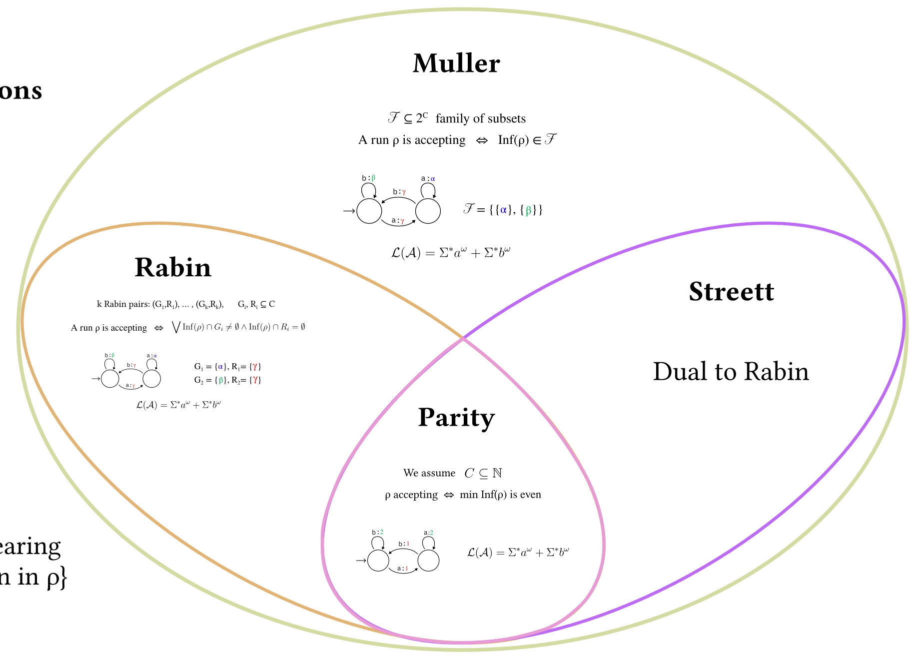
Input $w \in \Sigma^\omega$ ⟶ Infinite path ρ

Different Acceptance Conditions

$C = \{\alpha, \beta, \gamma\}$

output colours

$\text{Inf}(\rho)$ = {Colours appearing infinitely often in ρ}



More general conditions → Automata are simpler to produce

- LTL ⟶ Det. Muller automata
- ND Büchi ⟶ Det. Rabin automata

Simpler conditions → More efficient algorithms

Complexity of solving 2-player games. n = Size of input graph

Muller	PSPACE-complete	$\mathcal{O}(k^{5k}n^5)$	k = #Colours
Rabin	NP-complete	$\mathcal{O}(n^{k+3}kk!)$	k = #Rabin pair
Parity	NP⋂coNP. In P?	$\mathcal{O}(n^{\log(k)})$	k = #Colours

Goal: Simplify the acceptance condition of automata

a) Simplify the type (Muller → Rabin → Parity)

b) Minimise #Colours / #Rabin pairs

Approach 1

Add states to the automaton to maximize simplification

√ Well understood

✦ Optimal transformation to parity automata *(C. Colcombet, Fijalkow '21)*

✗ Exponential blowup in the worst case *(Löding '99)*

In this paper

Simplify the condition **without** adding further states

Pure Conditions

Reduction of conditions

A reduction of a condition $\mathcal{F} \subseteq 2^C$ to $\mathcal{G} \subseteq 2^{C'}$ is a mapping

$$\varphi : C \longrightarrow C' \quad \text{such that}$$

$$X \in \mathcal{F} \iff \varphi(X) \in \mathcal{G}, \quad \text{for all } X \subseteq C$$

Given a condition $\mathcal{F}$, can we reduce it to

- ✦ a Muller condition with ≤ k colours?
- ✦ a Rabin condition? With ≤ k Rabin pairs?
- ✦ a parity condition? With ≤ k colours?

Main tool: Zielonka DAG *(Zielonka '98, Hunter-Dawar '06)*

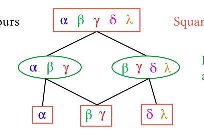
$C = \{\alpha, \beta, \gamma, \delta, \lambda\}$

$\mathcal{F} = \{\{\alpha,\beta,\gamma\}, \{\alpha,\gamma\}, \{\alpha,\beta\}, \{\beta,\gamma,\delta,\lambda\}, \{\beta,\delta,\lambda\}, \{\gamma,\delta,\lambda\}, \{\beta,\gamma,\lambda\}, \{\beta,\gamma,\delta\}, \{\beta,\delta\}, \{\beta,\lambda\}, \{\gamma,\delta\}, \{\gamma,\lambda\}\}$

Root ⟶ All colours

Children: Maximal accepting subsets

Children: Maximal rejecting subsets



Square if it is a rejecting set

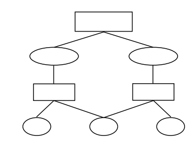
Round because they are accepting sets

Proposition *(Hunter-Dawar '06)*

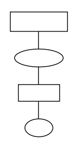
The Zielonka DAG can be computed in polynomial time in $|\mathcal{F}|$.

Typeness: Can we reduce $\mathcal{F}$ to a Rabin/parity condition?

Just look to the Zielonka DAG! *(Zielonka '98)*



Rabin
⇔
Round nodes have a single child



Parity
⇔
A single branch

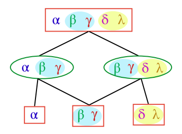
+ Minimal #Colours
=
Height of the branch

Minimisation of colours and Rabin pairs

Theorem

We can minimise the #Colours (resp. #Rabin pairs) in polynomial time.

Idea (for colours):



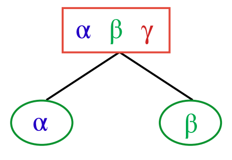
Two colours are equivalent if they appear in the same nodes

Merge equivalent colours

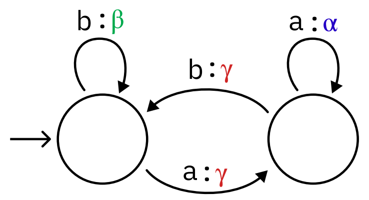
We show that the obtained condition is optimal

- Minimisation of Rabin pairs is quite involved.

Automata



Not parity
Requires 3 colours



$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

But we can relabel the automaton with a parity condition with 2 colours!

Can we relabel $\mathcal{A}$ with

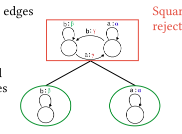
- ✦ a parity/Rabin condition?
- ✦ ≤ k colours / Rabin pairs?

Main tool: Alternating Cycle Decomposition (ACD) *(C. Colcombet Fijalkow '21)*

Generalisation of the Zielonka DAG

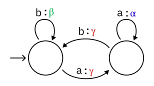
Root ⟶ All edges

Children: Maximal accepting subcycles



Square if they form a rejecting cycle

Round because they are accepting



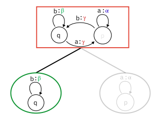
$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

Theorem (main technical contribution)

The Alternating Cycle Decomposition can be computed in **polynomial time** in $|\mathcal{A}| + |\mathcal{F}|$.

Typeness: Can we relabel $\mathcal{A}$ with a Rabin/parity condition?

Just look to the Altenating Cycle Decompositon! *(C. Colcombet Fijalkow '21)*



Restriction to the tree of a state q

Rabin
⇔
In the restriction-tree of every state round nodes have a single child

Parity
⇔
The restriction-tree of every state consists of a single branch.

+ Minimal #Colours
=
Height of the ACD

Minimization of colours and Rabin pairs

First idea: Do as with the Zielonka DAG!

But... It does not work!

Theorem

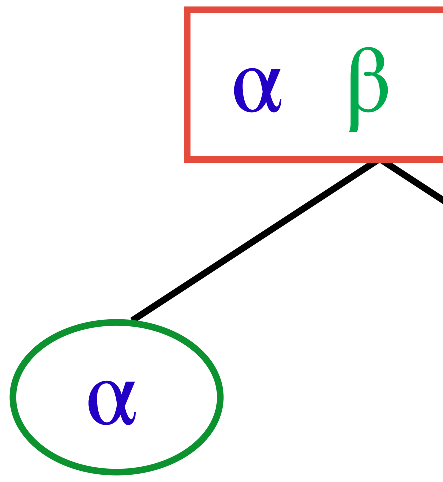
Given an automaton $\mathcal{A}$ and its ACD, it is **NP-complete** to determine if we can relabel it with a condition using:

- ≤ k colours.
- ≤ k Rabin pairs.

?

ion

	Pure Conditions	On top of automata
Typeness		
Min. colours/ Rabin pairs		



Not parity

Requires 3 co

Main tool: Alter

Generalisation of

Root ⟶ All ed

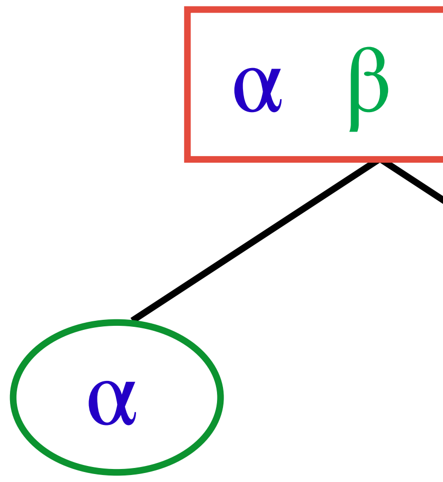
Children: Maximal accepting subcycles

Tr

?

	Pure Conditions	On top of automata
Typeness	PTIME	
Min. colours/ Rabin pairs	PTIME	

ion



Not parity

Requires 3 co

Main tool: Alter

Generalisation of

Root → All ed

Children: Maximal accepting subcycles

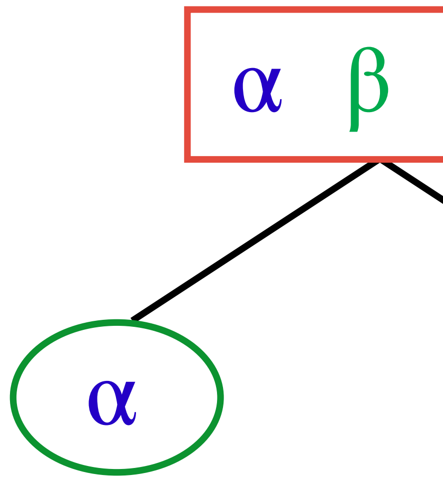
Tr

?

	Pure Conditions	On top of automata
Typeness	PTIME	
Min. colours/ Rabin pairs	PTIME	

Zielonka tree

ion



Not parity

Requires 3 co

Main tool: Alter

Generalisation of

Root → All ed

Children: Maximal accepting subcycles

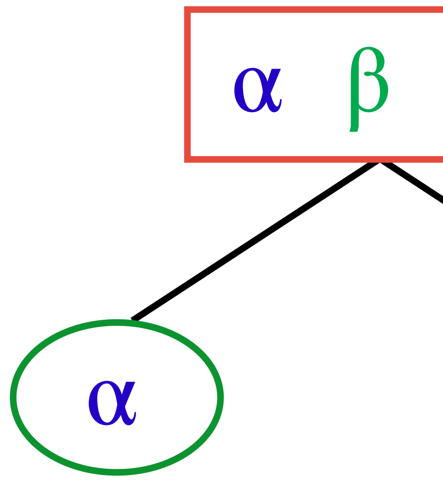
Tr

?

ion

	Pure Conditions	On top of automata
Typeness	PTIME	PTIME
Min. colours/ Rabin pairs	PTIME	

Zielonka tree



Not parity

Requires 3 co

Main tool: Alter

Generalisation of

Root ⟶ All ed

Children: Maximal accepting subcycles

Tr

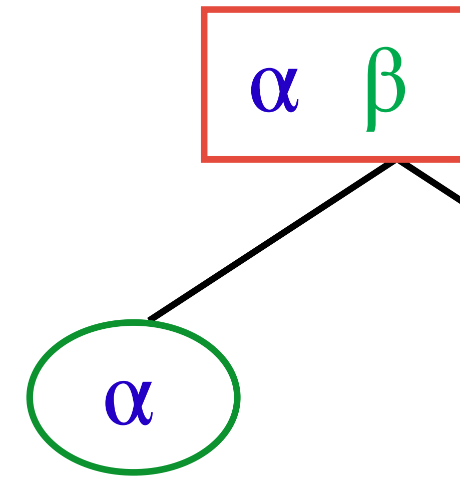
?

	Pure Conditions	On top of automata
Typeness	PTIME	PTIME
Min. colours/ Rabin pairs	PTIME	

Zielonka tree

Alternating Cycle Decomposition

ion



Not parity

Requires 3 co

Main tool: Alter

Generalisation of

Root ⟶ All ed

Children: Maximal accepting subcycles

Tr

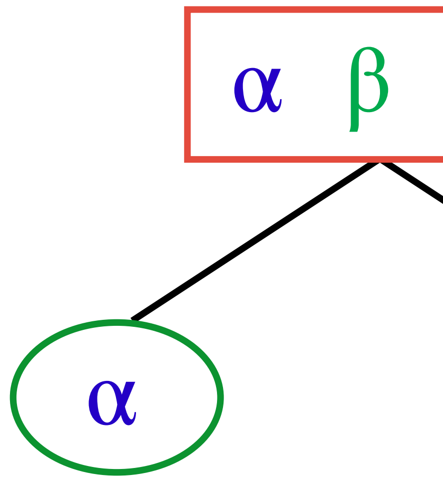
s?

	Pure Conditions	On top of automata
Typeness	PTIME	PTIME
Min. colours/ Rabin pairs	PTIME	NP-complete

Zielonka tree

Alternating Cycle Decomposition

ion



Not parity

Requires 3 co

Main tool: Alter

Generalisation of

Root → All ed

Children: Maximal accepting subcycles

Tr

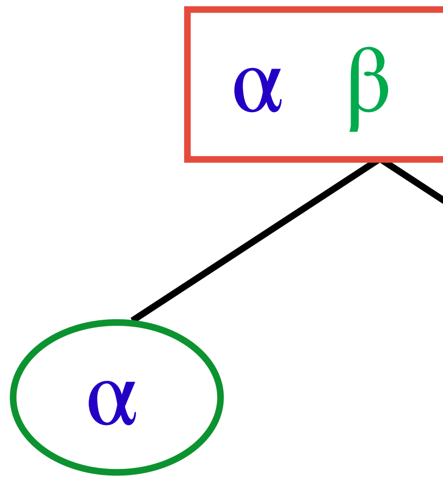
	Pure Conditions	On top of automata
Typeness	PTIME	PTIME
Min. colours/ Rabin pairs	PTIME	NP-complete

	Zielonka tree	*Alternating Cycle Decomposition*

THANK YOU!

?

ion



Not parity

Requires 3 co

Main tool: Alter

Generalisation of

Root → All ed

Children: Maximal accepting subcycles

Tr

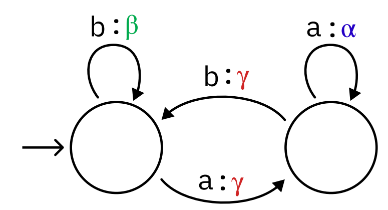
Simplifying ω-Automata through the Alternating Cycle Decomposition

Antonio Casares · University of Warsaw & Corto Mascle · LaBRI, University of Bordeaux

Automata over infinite words (ω-automata)

Automata are deterministic in this talk

$\mathcal{A}$

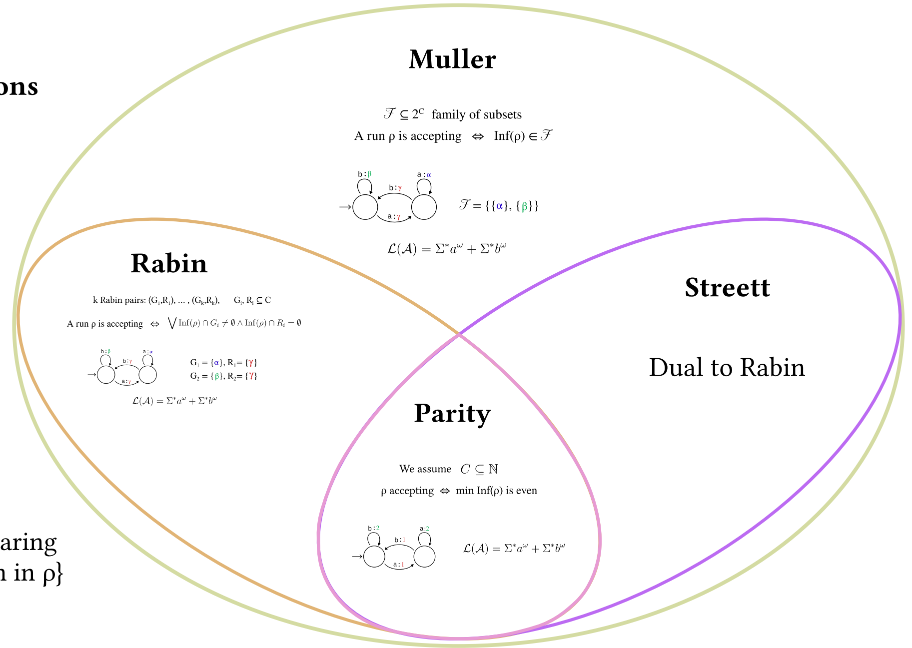


Input w ∈ Σ^ω ⟶ Infinite path ρ

Different Acceptance Conditions

C = {α, β, γ}
output colours

Inf(ρ) = {Colours appearing infinitely often in ρ}



More general conditions → Automata are simpler to produce

- LTL → Det. Muller automata
- ND Büchi → Det. Rabin automata

Simpler conditions → More efficient algorithms

Complexity of solving 2-player games. n = Size of input graph

Muller	PSPACE-complete	$\mathcal{O}(k^{5k}n^5)$	k = #Colours
Rabin	NP-complete	$\mathcal{O}(n^{k+3}kk!)$	k = #Rabin pair
Parity	NP∩coNP. In P?	$\mathcal{O}(n^{\log(k)})$	k = #Colours

Goal: Simplify the acceptance condition of automata

a) Simplify the type (Muller → Rabin → Parity)

b) Minimise #Colours / #Rabin pairs

Approach 1

Add states to the automaton to maximize simplification

✓ Well understood

✦ Optimal transformation to parity automata *(C. Colcombet, Fijalkow '21)*

✗ Exponential blowup in the worst case *(Löding '99)*

In this paper

Simplify the condition **without** adding further states

Pure Conditions

Reduction of conditions

A reduction of a condition $\mathcal{F} \subseteq 2^C$ to $\mathcal{G} \subseteq 2^{C'}$ is a mapping

$$\varphi : C \to C' \quad \text{such that}$$

$$X \in \mathcal{F} \iff \varphi(X) \in \mathcal{G}, \quad \text{for all } X \subseteq C$$

Given a condition $\mathcal{F}$, can we reduce it to

- ✦ a Muller condition with ≤ k colours?
- ✦ a Rabin condition? With ≤ k Rabin pairs?
- ✦ a parity condition? With ≤ k colours?

Main tool: Zielonka DAG *(Zielonka '98, Hunter-Dawar '06)*

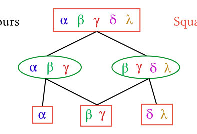
C = {α, β, γ, δ, λ}

$\mathcal{F}$ = {{α,β,γ},{α,γ},{α,β},{β,γ,δ,λ},{β,δ,λ},{γ,δ,λ},{β,γ,λ},{β,γ,δ},{β,δ},{β,λ},{γ,δ},{γ,λ}}

Root → All colours

Children: Maximal accepting subsets

Children: Maximal rejecting subsets



Square if it is a rejecting set

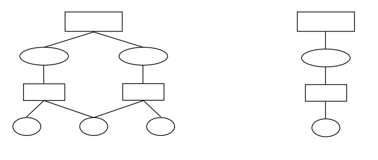
Round because they are accepting sets

Proposition *(Hunter-Dawar '06)*

The Zielonka DAG can be computed in polynomial time in $|\mathcal{F}|$.

Typeness: Can we reduce $\mathcal{F}$ to a Rabin/parity condition?

Just look to the Zielonka DAG! *(Zielonka '98)*



Rabin ⇔ Round nodes have a single child

Parity ⇔ A single branch

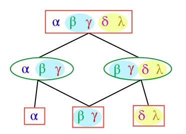
+ Minimal #Colours = Height of the branch

Minimisation of colours and Rabin pairs

Theorem

We can minimise the #Colours (resp. #Rabin pairs) in polynomial time.

Idea (for colours):



Two colours are equivalent if they appear in the same nodes

Merge equivalent colours

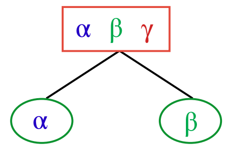
We show that the obtained condition is optimal

- Minimisation of Rabin pairs is quite involved.

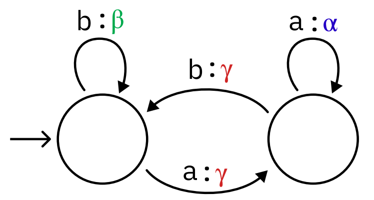
	Pure Conditions	On top of automata
Typeness	PTIME	PTIME
Min. colours/ Rabin pairs	PTIME	NP-complete
	Zielonka tree	*Alternating Cycle Decomposition*

THANK YOU!

Automata



Not parity
Requires 3 colours



$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

But we can relabel the automaton with a parity condition with 2 colours!

Can we relabel $\mathcal{A}$ with

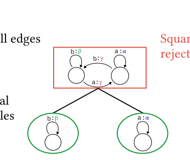
- ✦ a parity/Rabin condition?
- ✦ ≤ k colours / Rabin pairs?

Main tool: Alternating Cycle Decomposition (ACD) *(C. Colcombet Fijalkow '21)*

Generalisation of the Zielonka DAG

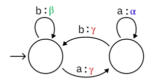
Root → All edges

Children: Maximal accepting subcycles



Square if they form a rejecting cycle

Round because they are accepting



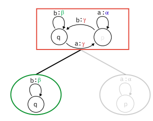
$\mathcal{F} = \{\{\alpha\}, \{\beta\}\}$

Theorem (main technical contribution)

The Alternating Cycle Decomposition can be computed in **polynomial time** in $|\mathcal{A}| + |\mathcal{F}|$.

Typeness: Can we relabel $\mathcal{A}$ with a Rabin/parity condition?

Just look to the Altenating Cycle Decompositon! *(C. Colcombet Fijalkow '21)*



Restriction to the tree of a state q

Rabin ⇔ In the restriction-tree of every state round nodes have a single child

Parity ⇔ The restriction-tree of every state consists of a single branch.

+ Minimal #Colours = Height of the ACD

Minimization of colours and Rabin pairs

First idea: Do as with the Zielonka DAG!

But... It does not work!

Theorem

Given an automaton $\mathcal{A}$ and its ACD, it is **NP-complete** to determine if we can relabel it with a condition using:

- ≤ k colours.
- ≤ k Rabin pairs.