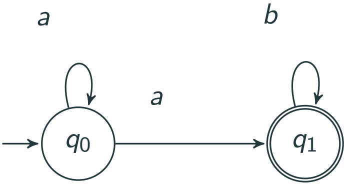
Remarks on Parikh-recognizable ω-languages

Mario Grobler, Leif Sabellek, Sebastian Siebertz

October 4, 2023

Parikh automata

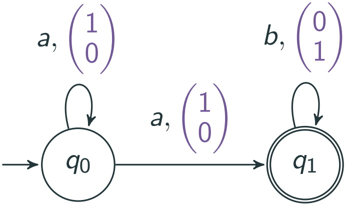
Consider the NFA $\mathcal{A}$



with input a a b b

Parikh automata

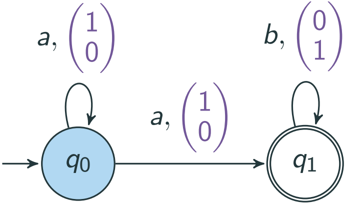
Consider the NFA $\mathcal{A}$



with input a a b b

Parikh automata

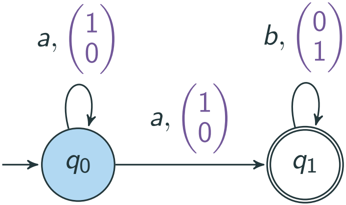
Consider the NFA $\mathcal{A}$



with input a a b b Run: q_0

Parikh automata

Consider the NFA $\mathcal{A}$

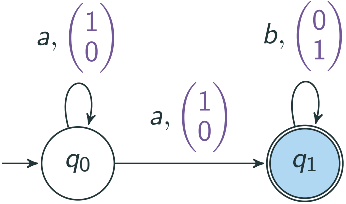


with input a a b b Run: $q_0 q_0$

$$v = \begin{pmatrix}1\\0\end{pmatrix}$$

Parikh automata

Consider the NFA $\mathcal{A}$

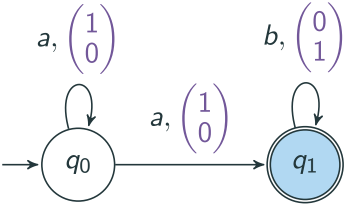


with input $\quad a \quad a \quad b \quad b \qquad$ Run: $q_0 q_0 q_1$

$$v = \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}1\\0\end{pmatrix}$$

Parikh automata

Consider the NFA $\mathcal{A}$

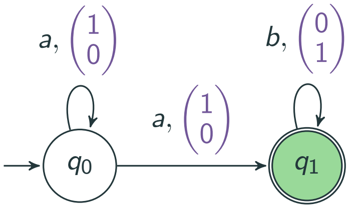


with input a a b b Run: $q_0 q_0 q_1 q_1$

$$v = \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix}$$

Parikh automata

Consider the NFA $\mathcal{A}$

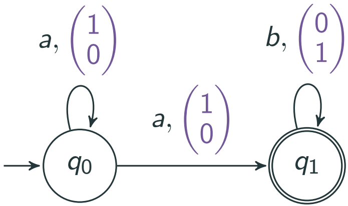


with input $\quad a \quad a \quad b \quad b \qquad$ Run: $q_0 q_0 q_1 q_1 q_1$

$$v = \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}2\\2\end{pmatrix}$$

Parikh automata

Consider the NFA $\mathcal{A}$



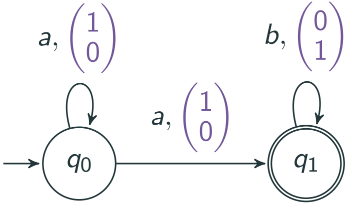
with input $a \quad a \quad b \quad b$ Run: $q_0q_0q_1q_1q_1$

$$v = \begin{pmatrix}1\\0\end{pmatrix}+\begin{pmatrix}1\\0\end{pmatrix}+\begin{pmatrix}0\\1\end{pmatrix}+\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}2\\2\end{pmatrix}$$

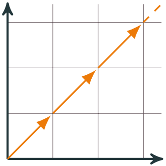
Check membership of v in given semi-linear set C; e.g

- Are both counter values equal?
- Is the first value at least twice the second value?

A Parikh automaton (PA) is a tuple $(\mathcal{A}, C)$.

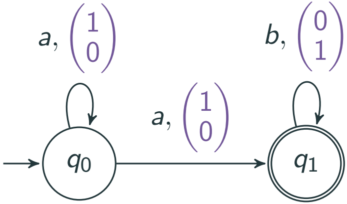


$\mathcal{A}$

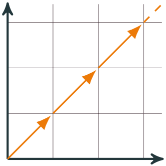


$$C = \left\{ \begin{pmatrix}1\\1\end{pmatrix} z \mid z \in \mathbb{N} \right\}$$

A Parikh automaton (PA) is a tuple $(\mathcal{A}, C)$.



$\mathcal{A}$



$$C = \left\{ \begin{pmatrix}1\\1\end{pmatrix} z \mid z \in \mathbb{N} \right\}$$

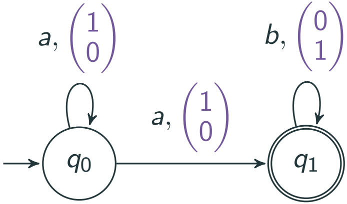
Accept if

- there is an accepting run, *and*
- summed up vector v is contained in C.

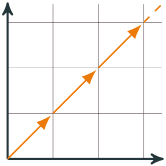
Recognized language: $L(\mathcal{A}, C) = ?$

Parikh automata

A Parikh automaton (PA) is a tuple $(\mathcal{A}, C)$.



$\mathcal{A}$



$$C = \left\{ \begin{pmatrix}1\\1\end{pmatrix} z \mid z \in \mathbb{N} \right\}$$

Accept if

- there is an accepting run, *and*
- summed up vector v is contained in C.

Recognized language: $L(\mathcal{A}, C) = \{a^n b^n \mid n \geq 1\}$.

History of Parikh automata

Studied as a tool to decide an existential fragment of (W)MSO with cardinality constraints [Klaedtke & Ruess, 2003].

Equivalent models:

- Blind-counter automata [Greibach, 1978].
- Weighted automata over $(\mathbb{Z}^k, +, \mathbf{0})$ [Mitrana & Stiebe, 2001].
- Reversal-bounded multicounter machines [Ibarra, 1978].

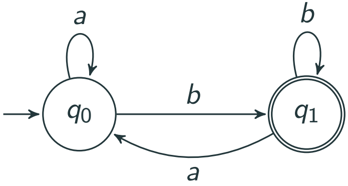
These models can be translated into each other in logspace

[Baumann et al., 2023].

To infinity and beyond

Reminder: Büchi automata

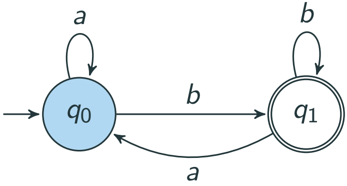
Recall Büchi automata which operate on infinite words.



Input:	*infinite* words $\alpha_1\alpha_2\alpha_3\ldots$
Acceptance:	Visit accepting state infinitely often.

Reminder: Büchi automata

Recall Büchi automata which operate on infinite words.



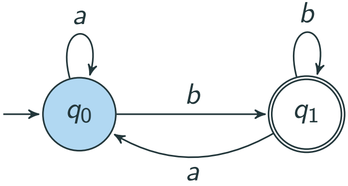
Input: *infinite* words $\alpha_1\alpha_2\alpha_3\ldots$

Acceptance: Visit accepting state infinitely often.

Input: *ababab* . . .   Run: q_0

Reminder: Büchi automata

Recall Büchi automata which operate on infinite words.

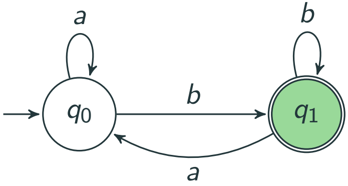


Input:	*infinite* words $\alpha_1\alpha_2\alpha_3\ldots$
Acceptance:	Visit accepting state infinitely often.

Input: *ababab*... Run: $q_0 q_0$

Reminder: Büchi automata

Recall Büchi automata which operate on infinite words.

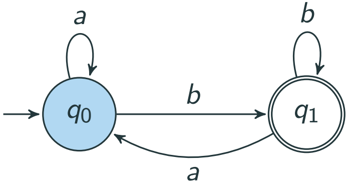


Input:	*infinite* words $\alpha_1\alpha_2\alpha_3\ldots$
Acceptance:	Visit accepting state infinitely often.

Input: *ababab*... Run: $q_0q_0q_1$

Reminder: Büchi automata

Recall Büchi automata which operate on infinite words.



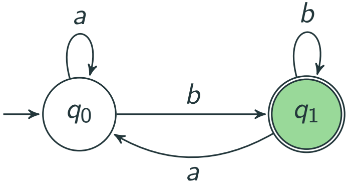
Input: *infinite* words $\alpha_1\alpha_2\alpha_3\ldots$

Acceptance: Visit accepting state infinitely often.

Input: $ababab\ldots$ Run: $q_0 q_0 q_1 q_0$

Reminder: Büchi automata

Recall Büchi automata which operate on infinite words.



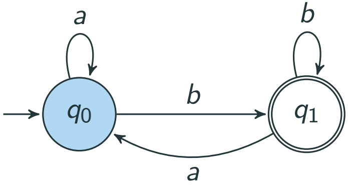
Input: *infinite* words $\alpha_1\alpha_2\alpha_3\ldots$

Acceptance: Visit accepting state infinitely often.

Input: *ababab*... Run: $q_0q_0q_1q_0q_1$

Reminder: Büchi automata

Recall Büchi automata which operate on infinite words.



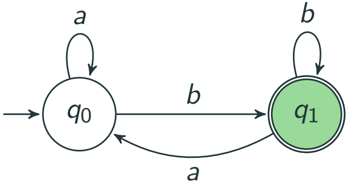
Input: *infinite* words $\alpha_1\alpha_2\alpha_3\ldots$

Acceptance: Visit accepting state infinitely often.

Input: $ababab\ldots$ Run: $q_0q_0q_1q_0q_1q_0$

Reminder: Büchi automata

Recall Büchi automata which operate on infinite words.



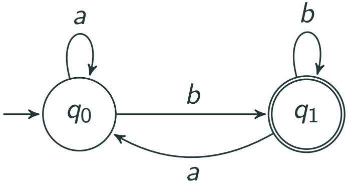
Input: *infinite* words $\alpha_1\alpha_2\alpha_3\ldots$

Acceptance: Visit accepting state infinitely often.

Input: $ababab\ldots$ Run: $q_0q_0q_1q_0q_1q_0q_1\ldots$ ✓

Reminder: Büchi automata

Recall Büchi automata which operate on infinite words.



Input: *infinite* words $\alpha_1\alpha_2\alpha_3\ldots$

Acceptance: Visit accepting state infinitely often.

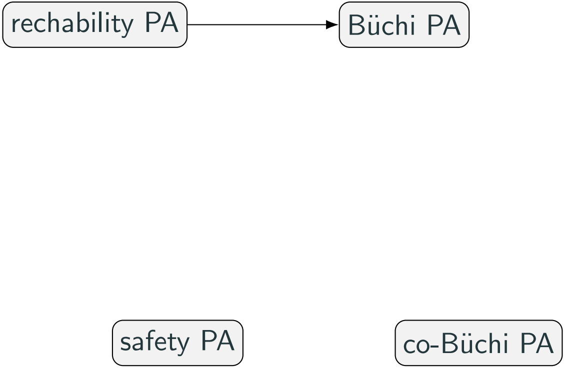
Input: $ababab\ldots$ Run: $q_0q_0q_1q_0q_1q_0q_1\ldots$ ✓

Büchi automata recognize the ω-regular languages.

PA on infinite words

How to define PA on infinite words?

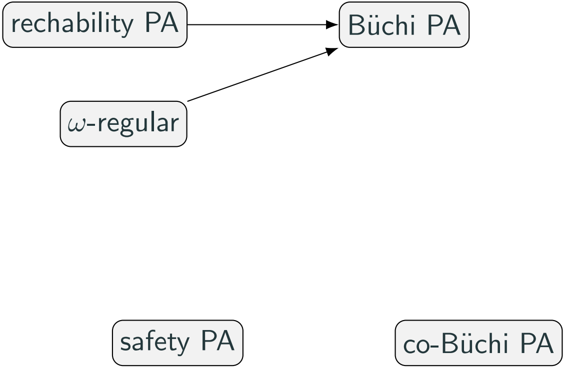
- Answered independently by Guha et al., 2022.
- They proposed the following models:



PA on infinite words

How to define PA on infinite words?

- Answered independently by Guha et al., 2022.
- They proposed the following models:



Büchi's Theorem and friends?

Büchi's Theorem

A language $L \subseteq \Sigma^\omega$ is ω-regular if and only if there are regular languages $U_1, V_1, \ldots, U_n, V_n \subseteq \Sigma^*$ with $L = U_1 V_1^\omega \cup \cdots \cup U_n V_n^\omega$.

What happens if we plug in PA-recognizable languages?

- $\mathcal{L}^\omega_{\text{Reg},\text{Reg}}$: regular U_i and V_i.
- $\mathcal{L}^\omega_{\text{Reg},\text{PA}}$: regular U_i and PA-recognizable V_i.
- $\mathcal{L}^\omega_{\text{PA},\text{Reg}}$: PA-recognizable U_i and regular V_i.
- $\mathcal{L}^\omega_{\text{PA},\text{PA}}$: PA-recognizable U_i and V_i.

Büchi's Theorem and friends?

Büchi's Theorem

A language $L \subseteq \Sigma^\omega$ is ω-regular if and only if there are regular languages $U_1, V_1, \dots, U_n, V_n \subseteq \Sigma^*$ with $L = U_1 V_1^\omega \cup \dots \cup U_n V_n^\omega$.

What happens if we plug in PA-recognizable languages?

- $\mathcal{L}^\omega_{\text{Reg,Reg}}$: regular U_i and V_i.
- $\mathcal{L}^\omega_{\text{Reg,PA}}$: regular U_i and PA-recognizable V_i.
- $\mathcal{L}^\omega_{\text{PA,Reg}}$: PA-recognizable U_i and regular V_i.
- $\mathcal{L}^\omega_{\text{PA,PA}}$: PA-recognizable U_i and V_i.

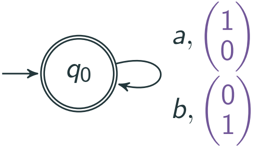
Lemma [Guha et al., 2022]

Büchi PA recognize a strict subset of $\mathcal{L}^\omega_{\text{PA,PA}}$.

Limit PA

First, we define limit PA:

- Here, we consider semi-linear sets over $(\mathbb{N} \cup \{\infty\})^d$.

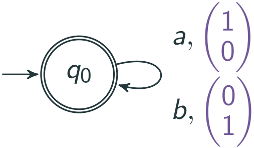


$$C = \left\{ \begin{pmatrix} \infty \\ n \end{pmatrix} \mid n \in \mathbb{N} \right\}$$

Limit PA

First, we define limit PA:

- Here, we consider semi-linear sets over $(\mathbb{N} \cup \{\infty\})^d$.



$$C = \left\{ \begin{pmatrix} \infty \\ n \end{pmatrix} \mid n \in \mathbb{N} \right\}$$

We compute the sum of vectors over the whole infinite word:

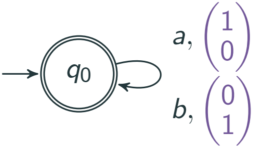
- Component is ∞ if the series diverges.
- Example: run on $abbaba^\omega$ yields $(\infty, 3)$.

Accept if an acc. state is seen infinitely often and vector is in C.

Limit PA

First, we define limit PA:

- Here, we consider semi-linear sets over $(\mathbb{N} \cup \{\infty\})^d$.



$$C = \left\{ \begin{pmatrix} \infty \\ n \end{pmatrix} \mid n \in \mathbb{N} \right\}$$

We compute the sum of vectors over the whole infinite word:

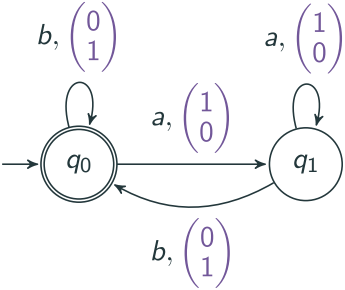
- Component is ∞ if the series diverges.
- Example: run on $abbaba^\omega$ yields $(\infty, 3)$.

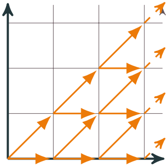
Accept if an acc. state is seen infinitely often and vector is in C.

- Here: accepts α if $|\alpha|_b < \infty$.

Rechability-regular PA

Next, we define reachability-regular PA:





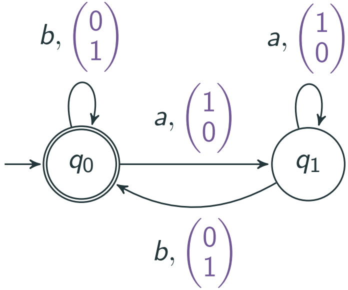
$$C = \left\{ \begin{pmatrix} z \\ z' \end{pmatrix} \mid z \geq z' \right\}$$

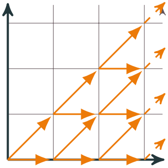
Accept an infinite word if

- it has a finite prefix that is accepted by the underlying PA and
- an infinite state is seen infinitely often.

Rechability-regular PA

Next, we define reachability-regular PA:





$$C = \left\{ \begin{pmatrix} z \\ z' \end{pmatrix} \mid z \geq z' \right\}$$

Accept an infinite word if

- it has a finite prefix that is accepted by the underlying PA and
- an infinite state is seen infinitely often.

Here: accepts $w\beta$ if $|\beta|_b = \infty$, $|w|_a \geq |w|_b$ and w ends with b.

Bringing them together

Lemma

The following are equivalent for all ω-languages $L \subseteq \Sigma^\omega$.

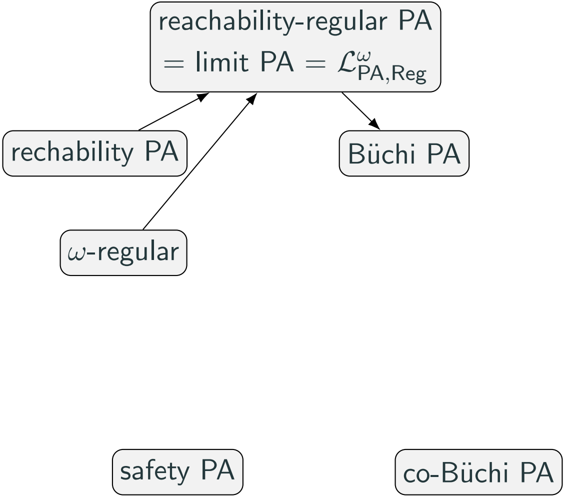
1. L is limit PA-recognizable.
2. L is reachability-regular.
3. L is in $\mathcal{L}^\omega_{\text{PA,Reg}}$.

Side-product: for every limit PA there is an equivalent limit PA whose semi-linear set C does not use ∞, that is $C \subseteq \mathbb{N}^d$.

We observe:

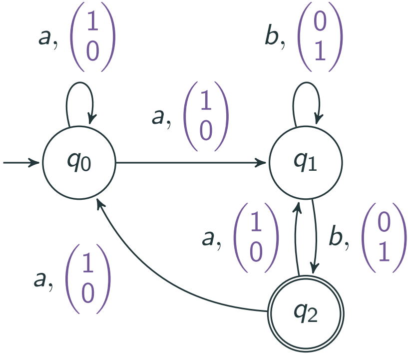
- If L is reachability PA-recognizable, then L is in $\mathcal{L}^\omega_{\text{PA,Reg}}$.
- If L is in $\mathcal{L}^\omega_{\text{PA,Reg}}$, then L is Büchi PA-recognizable.

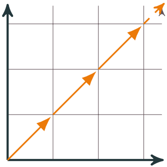
Overview


reachability-regular PA
$=$ limit PA $= \mathcal{L}^{\omega}_{\mathrm{PA,Reg}}$
rechability PA
Büchi PA
ω-regular
safety PA
co-Büchi PA

Reset PA

Finally, we define reset PA:



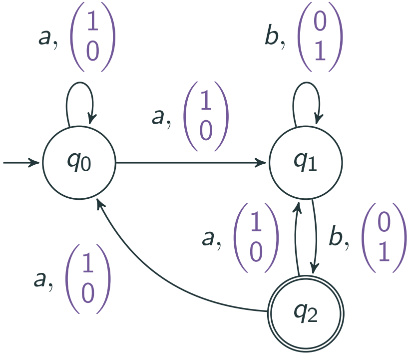


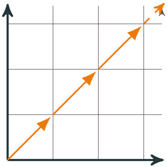
$$C = \left\{ \begin{pmatrix}1\\1\end{pmatrix} z \mid z \in \mathbb{N} \right\}$$

- Whenever an acc. state is visited, the value *must* be in C.
- Then the counters are reset. Accept if there are ∞ resets.

Reset PA

Finally, we define reset PA:





$$C = \left\{ \begin{pmatrix}1\\1\end{pmatrix} z \mid z \in \mathbb{N} \right\}$$

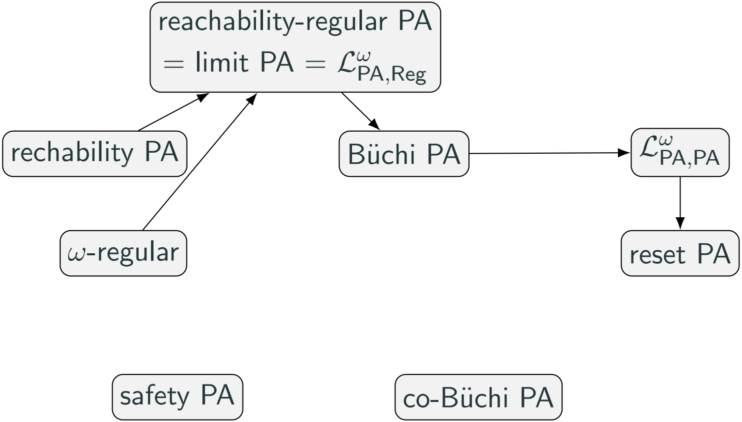
- Whenever an acc. state is visited, the value *must* be in C.
- Then the counters are reset. Accept if there are ∞ resets.

Here: recognizes $\{a^n b^n \mid n \geq 1\}^\omega$

Reset PA

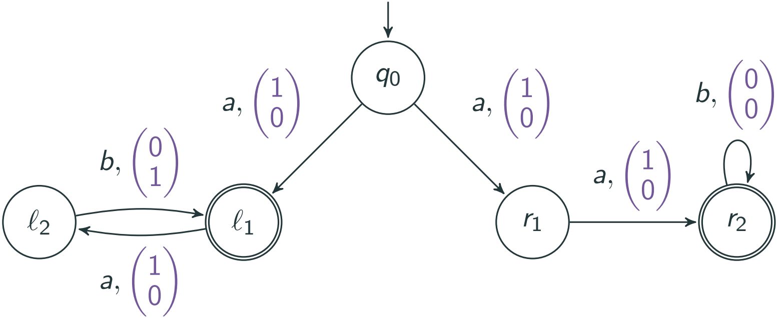
Reset PA are closed under $\cdot^\omega$.

- Reset PA recognize a strict superset of $\mathcal{L}^\omega_{\mathsf{PA},\mathsf{PA}}$.
- Still NP-complete emptiness problem.



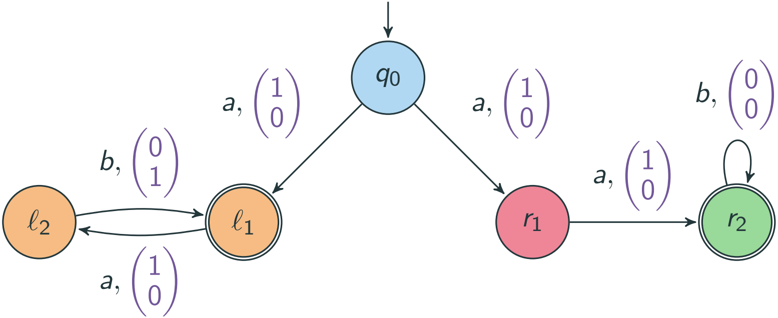
Automata characterizations of $\mathcal{L}^{\omega}_{\mathrm{PA,PA}}$ and $\mathcal{L}^{\omega}_{\mathrm{Reg,PA}}$

Restricting reset PA yield an automata characterization of $\mathcal{L}^{\omega}_{\mathrm{PA,PA}}$.



Automata characterizations of $\mathcal{L}^{\omega}_{\mathsf{PA},\mathsf{PA}}$ and $\mathcal{L}^{\omega}_{\mathsf{Reg},\mathsf{PA}}$

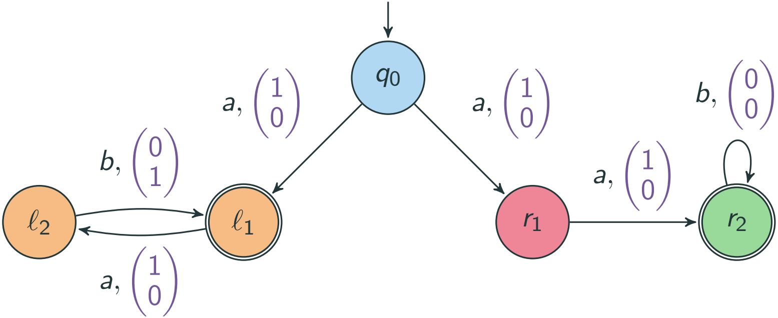
Restricting reset PA yield an automata characterization of $\mathcal{L}^{\omega}_{\mathsf{PA},\mathsf{PA}}$.



We consider the condensation (dag of connected components).

Automata characterizations of $\mathcal{L}^\omega_{\mathrm{PA,PA}}$ and $\mathcal{L}^\omega_{\mathrm{Reg,PA}}$

Restricting reset PA yield an automata characterization of $\mathcal{L}^\omega_{\mathrm{PA,PA}}$.

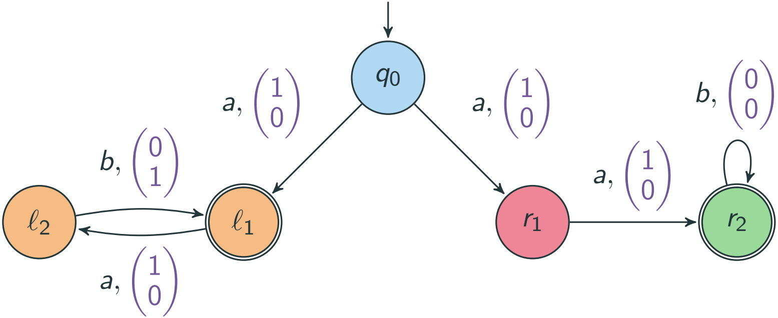


We consider the condensation (dag of connected components).

- If accepting states appear only in leaves, and there is at most one accepting state per leaf, we exactly capture $\mathcal{L}^\omega_{\mathrm{PA,PA}}$.

Automata characterizations of $\mathcal{L}^{\omega}_{\mathrm{PA,PA}}$ and $\mathcal{L}^{\omega}_{\mathrm{Reg,PA}}$

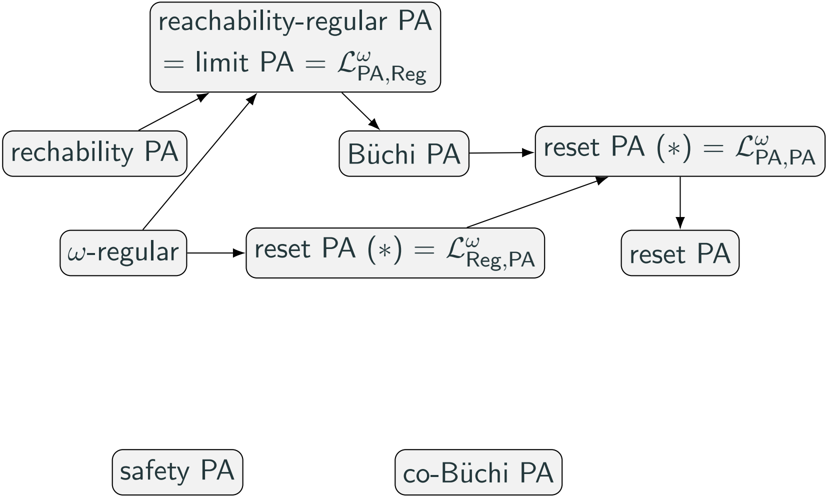
Restricting reset PA yield an automata characterization of $\mathcal{L}^{\omega}_{\mathrm{PA,PA}}$.



We consider the condensation (dag of connected components).

- If accepting states appear only in leaves, and there is at most one accepting state per leaf, we exactly capture $\mathcal{L}^{\omega}_{\mathrm{PA,PA}}$.
- A further restrictions yields a characteriztion of $\mathcal{L}^{\omega}_{\mathrm{Reg,PA}}$.

Overview



$(*)$ suitable graph theoretical restrictions

Blind counter machines vs Büchi PA

Blind counter machines on infinite words [Fernau & Stiebe, 2007]

- Transitions are equipped with (possibly negative) integers.
- Accept if an acc. state is seen ∞ often while counters are 0.
- Silent transitions (ε-transitions) are allowed.

Blind counter machines vs Büchi PA

Blind counter machines on infinite words [Fernau & Stiebe, 2007]

- Transitions are equipped with (possibly negative) integers.
- Accept if an acc. state is seen ∞ often while counters are 0.
- Silent transitions (ε-transitions) are allowed.

Büchi PA [Guha et al., 2022]

- Syntactically equivalent to PA.
- Accept if an acc. state is seen ∞ often while counters are in C.
- Silent transitions (ε-transitions) are not allowed.

Blind counter machines vs Büchi PA

Blind counter machines on infinite words [Fernau & Stiebe, 2007]

- Transitions are equipped with (possibly negative) integers.
- Accept if an acc. state is seen ∞ often while counters are 0.
- Silent transitions (ε-transitions) are allowed.

Büchi PA [Guha et al., 2022]

- Syntactically equivalent to PA.
- Accept if an acc. state is seen ∞ often while counters are in C.
- Silent transitions (ε-transitions) are not allowed.

We show that they are equivalent, answering an open question.

- Technical part: removal of ε-transitions (thanks Georg!)

Elimination of ε-transitions in other models

Natural question: which models do admit ε-elimination?

- Almost all, the exception being safety and co-Büchi PA.
- Often a direct consequence of characterization lemmas (finite word PA admit ε-elimination).
- Sometimes more technical, e.g. for reset PA.

Elimination of ε-transitions in other models

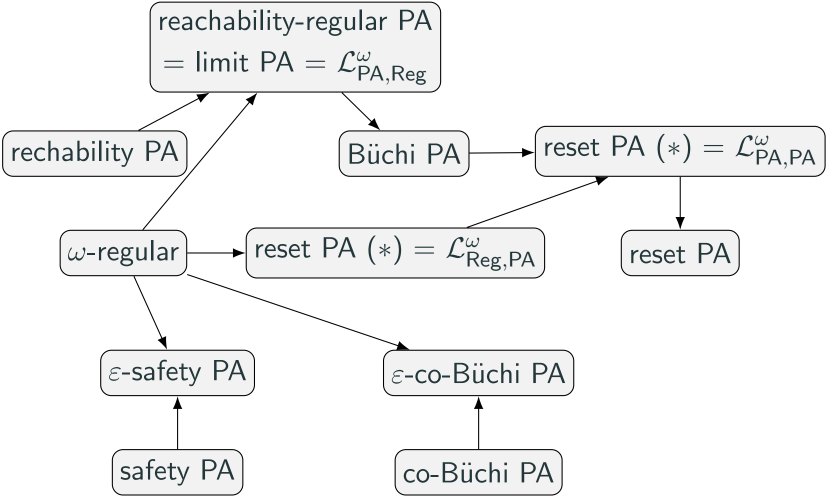
Natural question: which models do admit ε-elimination?

- Almost all, the exception being safety and co-Büchi PA.
- Often a direct consequence of characterization lemmas (finite word PA admit ε-elimination).
- Sometimes more technical, e.g. for reset PA.

For safety and co-Büchi PA, we can use ε-transitions to encode the acceptance condition of Büchi automata.

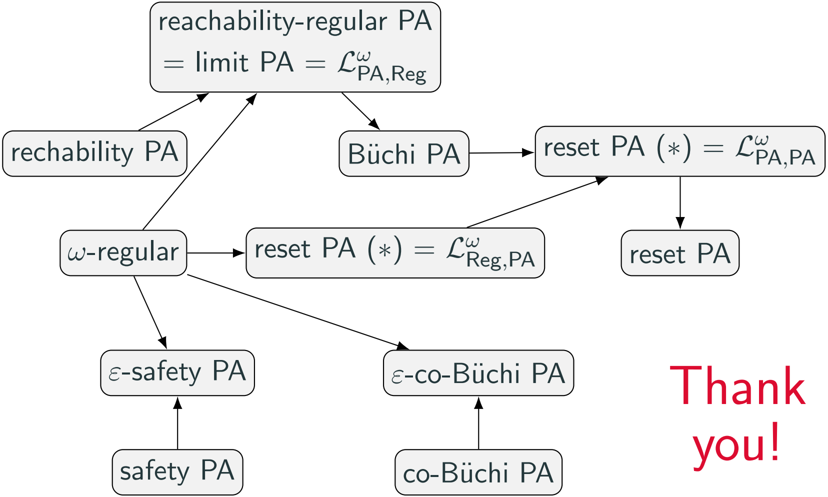
- Hence, these models recognize all ω-regular languages.

Conclusion



$(*)$ suitable graph theoretical restrictions

Conclusion



Thank you!

$(*)$ suitable graph theoretical restrictions