The Separability Problem for Presburger Definable Sets and Parikh Automata

Workshop on Recent Developments in Arithmetic Theories and Applications @ ICLA 2025, Kolkata

Elias Rojas Collins[1] Chris Köcher[2] Georg Zetzsche[2]

[1] Massachusetts Institute of Technology, Cambridge
[2] Max Planck Institute for Software Systems, Kaiserslautern

February 1, 2025

Section 1

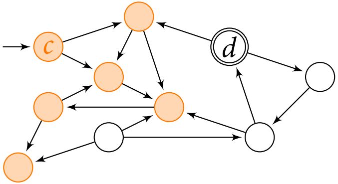
Verification Problems

Reachability Problem (1)

Reachability Problem

Given: A program $\mathfrak{P}$ and two configurations c and d of $\mathfrak{P}$

Question: Is there a run of $\mathfrak{P}$ starting in c and eventually arriving in d, i.e. $c \rightarrow^*_{\mathfrak{P}} d$?



- Classical correctness checks:
  - Run program $\mathfrak{P}$ with input x. Is the output y, i.e. $[\![\mathfrak{P}]\!](x) = y$?
  - Run program $\mathfrak{P}$ with input x. Will $\mathfrak{P}$ throw the exception e?
- Generalized problem:

  Given: A program $\mathfrak{P}$ and two sets of configurations C and D of $\mathfrak{P}$

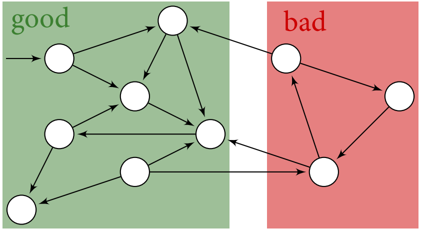
  Question: Is there a run of $\mathfrak{P}$ starting in $c \in C$ and eventually arriving in $d \in D$?

Safety Problem

Safety Problem

Input: Given a program $\mathfrak{P}$, a configuration c, and a set of configurations S of $\mathfrak{P}$

Question: Does any run of $\mathfrak{P}$ starting in c stay in the set S?



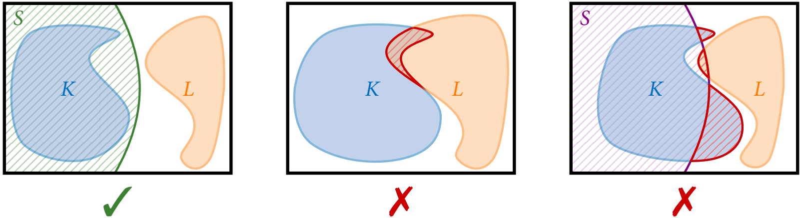
- Can we avoid reaching an undesired configuration (like an unhandled exception)?
- The complementary problem of Reachability:
  - Check whether there is no run starting from c eventually arriving in a $d \in \mathrm{Conf}_{\mathfrak{P}} \setminus S$.

Separability Problem

Separability Problem (of two sets from a class $\mathcal{C}$ by a set from a class $\mathcal{D}$)

Input: Two sets $K, L \in \mathcal{C}$.

Question: Is there a set $S \in \mathcal{D}$ such that $K \subseteq S$ and $L \cap S = \emptyset$?



- Certifies safety of programs:
  - K is the set of reachable configurations in $\mathfrak{P}$
  - L is the set of undesired configurations in $\mathfrak{P}$

Section 2

Separability in Presburger Definable Sets

Presburger Logic

Definition

Presburger logic is the first order logic of the structure $\mathcal{N} = (\mathbb{N}, +, \leq, 0, 1)$. A set $S \subseteq \mathbb{N}^d$ is Presburger definable if there is a Presburger formula $\phi(\vec{x})$ with d free variables $\vec{x}$ such that

$$S = \{\vec{v} \in \mathbb{N}^d \mid \mathcal{N}, \vec{v} \vDash \phi(\vec{x})\} .$$

Satisfiability Problem

Given A Presburger sentence ϕ (a formula without free variables)

Question Does $\mathcal{N} \vDash \phi$ hold?

Theorem (Presburger 1929)

Satisfiability of Presburger formulas is decidable.
Satisfiability of existential Presburger formulas is in NP.

Presburger Logic

Definition

Presburger logic is the first order logic of the structure $\mathcal{N} = (\mathbb{N}, +, \leq, 0, 1)$. A set $S \subseteq \mathbb{N}^d$ is Presburger definable if there is a Presburger formula $\phi(\vec{x})$ with d free variables $\vec{x}$ such that

$$S = \{\vec{v} \in \mathbb{N}^d \mid \mathcal{N}, \vec{v} \models \phi(\vec{x})\}.$$

Satisfiability Problem

Given A Presburger sentence ϕ (a formula without free variables)

Question Does $\mathcal{N} \models \phi$ hold?

$\exists \vec{x} \colon \psi(\vec{x}, \vec{y})$, where ψ contains no quantifiers

Satisfiability of Presburger formulas is decidable.

Satisfiability of existential Presburger formulas is in NP.

Semilinear Sets

Definition

A set $S \subseteq \mathbb{N}^d$ is

- linear if there is a vector $\vec{u} \in \mathbb{N}^d$ and a finite set $P \subseteq \mathbb{N}^d$ with $S = \vec{u} + P^*$.
- semilinear if it is a finite union of linear sets.

Theorem (Ginsburg & Spanier 1964)

Let $S \subseteq \mathbb{N}^d$ be a set of vectors. The following are equivalent:

1. *S is Presburger definable.*
2. *S is semilinear.*

The equivalence is effective.

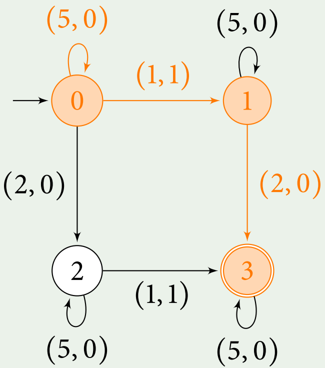
Recognizable Sets (1)

Definition

A set $S \subseteq \mathbb{N}^d$ is recognizable if S accepted by a DFA $\mathfrak{A}$ labeled with vectors from $\mathbb{N}^d$ such that $p \xrightarrow{\vec{v}} q \xrightarrow{\vec{w}} r$ implies the existence of q' with $p \xrightarrow{\vec{w}} q' \xrightarrow{\vec{v}} r$.

Example

$(8,1) \in L(\mathfrak{A})$?



Recognizable Sets (2)

Theorem

Let $S \subseteq \mathbb{N}^d$ be a set of vectors. The following are equivalent:

1. *S is recognizable.*
2. *S is definable by a monadic Presburger formula.*

each atom in $\phi(\vec{x})$ contains at most one variable

Main Theorem 1

Theorem

Given two semilinear sets defined by existential Presburger formulas, separability via recognizable sets is coNP*-complete.*

- coNP-hardness:
  - Reduction from emptiness of semilinear sets (given as existential Presburger formulas)
  - Let $K \subseteq \mathbb{N}^d$ be defined by a formula ϕ.
  - K is empty iff K is separable from $\mathbb{N}^d$ by a recognizable set.

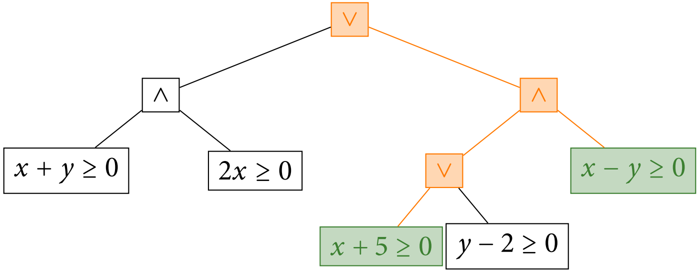
Proof Plan

- We will show that the inseparability problem is in NP.
- Given two existential Presburger formulas $\phi(\vec{x})$ and $\psi(\vec{x})$.
- We will construct another existential Presburger sentence χ such that

$$\mathcal{N} \models \chi \iff \text{the solution sets of } \phi \text{ and } \psi \text{ are inseparable}\,.$$

Simplifying Formulas

- Let $\phi = \exists \vec{y}\colon \xi_1(\vec{x}, \vec{y})$ and $\psi = \exists \vec{y}\colon \xi_2(\vec{x}, \vec{y})$ where
  - $\xi_i(\vec{x}, \vec{y})$ contains no quantifiers, no negation, and only atoms of the form $t \geq 0$
- There are formulas $\phi_1, \ldots, \phi_k, \psi_1, \ldots, \psi_\ell$ using only conjunctions such that $\xi_1 \equiv \phi_1 \vee \cdots \vee \phi_k$ and $\xi_2 \equiv \psi_1 \vee \cdots \vee \psi_\ell$.
  - i.e., we can transform ξ_1 and ξ_2 into disjunctive normal form.
  - Problem: k and ℓ are of exponential size!
- ϕ and ψ are inseparable if, and only if, there are i, j such that ϕ_i and ψ_j are inseparable.
- We can guess in polynomial time such ϕ_i and ψ_j:



Formulas to Diophantine Equations

- Now, we can assume that ϕ and ψ are finite conjunctions of atoms of the form $t \geq 0$.
- Adding further variables, we can replace each $t \geq 0$ by a $t' = 0$.
- We can turn ϕ and ψ into

$$R = \pi(\{\vec{x} \in \mathbb{N}^e \mid A\vec{x} = \vec{b}\}) \qquad \text{and} \qquad S = \pi(\{\vec{y} \in \mathbb{N}^e \mid C\vec{y} = \vec{d}\}) .$$

- R and S are hyperlinear sets.

R and S are of the form $A + U^*$ for finite sets $A, U \subseteq \mathbb{N}^d$

Intermezzo: An Idea for Simplification

- Assume R is bounded at coordinate j, i.e., there is a $M \in \mathbb{N}$ such that $\vec{v}[j] \leq M$ for all $\vec{v} \in R$.
- Then R and S are inseparable iff there is $x \in [0, M]$ such that $R[j \mapsto x]$ and $S[j \mapsto x]$ are inseparable
  - $R[j \mapsto x]$ contains all vectors $\vec{v} \in R$ with $\vec{v}[j] = x$, projected to coordinates $[1, d] \setminus \{j\}$.
- Guess such x and then continue with $R[j \mapsto x]$ and $S[j \mapsto x]$.

- Problem: Constructing $R[j \mapsto x]$ is expensive!

[Choffrut & Grigorieff 2006, Clemente et al. @ STACS 2017]

Strongly Unbounded Coordinates

- For a vector $\vec{v} \in \mathbb{N}^d$ the support is $\text{supp}(\vec{v}) = \{j \in [1, d] \mid \vec{v}[j] \neq 0\}$.
- Let $R = A + U^*$ and $S = B + V^*$.
- Repeat the following until stabilization. For each $j \in [1, d]$:
  - If S is bounded at j, remove all vectors $\vec{v} \in U$ with $j \in \text{supp}(\vec{v})$.
  - If R is bounded at j, remove all vectors $\vec{v} \in V$ with $j \in \text{supp}(\vec{v})$.
- The remaining (unbounded) coordinates are called strongly unbounded.
- Let $J \subseteq [1, d]$ be the set of all strongly unbounded coordinates.
- U_J and V_J are the sets of all remaining vectors in U resp. V after the procedure above.

A Technical Lemma

Lemma

Let $R = A + U^*$ *and* $S = B + V^*$ *be two hyperlinear sets. Then R and S are not separable by a recognizable set if, and only if, the intersection*

$$(A + U^* - U_J^*) \cap (B + V^* - V_J^*)$$

is not empty.

Lemma

Let $R = A + U^*$ *and* $S = B + V^*$ *be two hyperlinear sets. Then R and S are not separable by a recognizable set if, and only if, the intersection*

$$(A + U^* + V_J^*) \cap (B + V^* + U_J^*)$$

is not empty.

A Technical Lemma

Lemma

Let $R = A + U^*$ and $S = B + V^*$ be two hyperlinear sets. Then R and S are not separable by a recognizable set if, and only if, the intersection

$$(A + U^* - U_J^*) \cap (B + V^* - V_J^*)$$

is not empty.

R extended by the group generated by U_J

Lemma

Let $R = A + U^*$ and $S = B + V^*$ be two hyperlinear sets. Then R and S are not separable by a recognizable set if, and only if, the intersection

$$(A + U^* + V_J^*) \cap (B + V^* + U_J^*)$$

is not empty.

Diophantine Equations to Satisfiability

- Recall:
  - $R = \pi(\{\vec{x} \in \mathbb{N}^e \mid A\vec{x} = \vec{b}\})$ and $S = \pi(\{\vec{y} \in \mathbb{N}^e \mid C\vec{y} = \vec{d}\})$.
  - R and S are inseparable iff $(A + U^* + V_J^*) \cap (B + V^* + U_J^*) \neq \varnothing$.

Lemma

R and S are inseparable if, and only if, there are vectors $\vec{u}, \vec{v}, \vec{x}, \vec{y} \in \mathbb{N}^e$ with

1. $A\vec{u} = \vec{0}$, $C\vec{v} = \vec{0}$, $\mathrm{supp}(\pi(\vec{u})) = \mathrm{supp}(\pi(\vec{v}))$, *and*
2. $A\vec{x} = \vec{b}$, $C\vec{y} = \vec{d}$, *and* $\pi(\vec{x} + \vec{v}) = \pi(\vec{y} + \vec{u})$.

- We can express this in an existential Presburger formula χ.
- This formula is satisfiable if, and only if, R and S are inseparable.
- We can compute χ from ϕ and ψ in (non-deterministically) polynomial time. □

Section 3

Separability in Parikh Automata

Parikh's Theorem

Definition

Let $\Sigma = \{a_1, a_2, \ldots, a_d\}$ be an alphabet. The Parikh map is defined as

$$\Psi \colon \Sigma^* \to \mathbb{N}^d \colon w \mapsto (|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_d})$$

where $|w|_a$ is the number of occurrences of a in the word w.
The Parikh image of a language $L \subseteq \Sigma^*$ is $\Psi(L) = \{\Psi(w) \mid w \in L\}$

Theorem (Parikh 1966)

The Parikh image of context-free languages is semilinear.

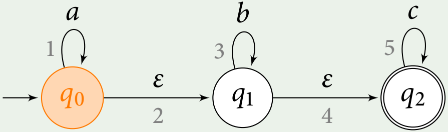
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: $aabbcc$



counters:

0	0	0	0	0

Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

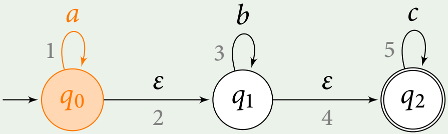
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: $aabbcc$



counters:

0	0	0	0	0

Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

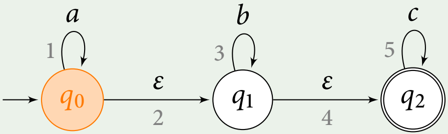
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: $aabbcc$



counters:

1	0	0	0	0

Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

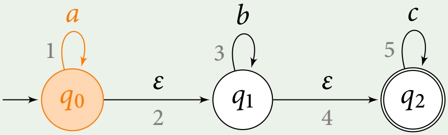
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: $aabbcc$



counters:

1	0	0	0	0

Theorem

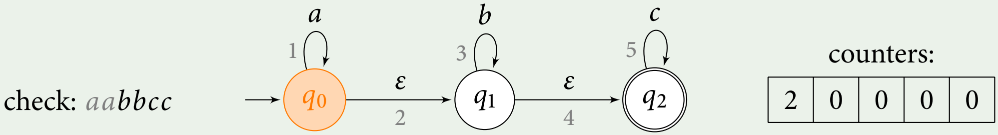
The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)



Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

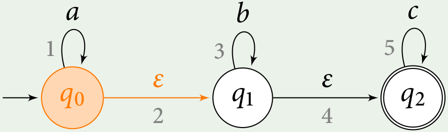
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: $aabbcc$



counters:

2	0	0	0	0

Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

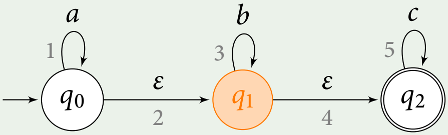
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: $aabbcc$



counters:

2	1	0	0	0

Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

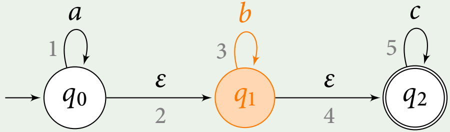
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: $aabbcc$



counters:

2	1	0	0	0

Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

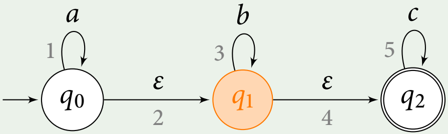
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: $aabbcc$



counters:

2	1	1	0	0

Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

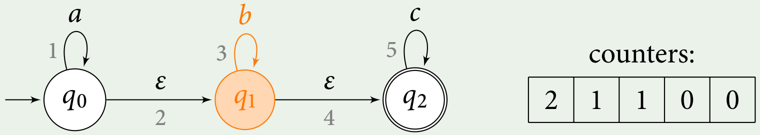
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: $aabbcc$



Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

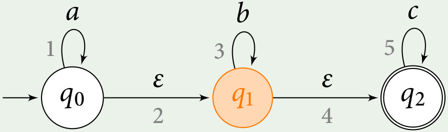
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: $aabbcc$



counters:

2	1	2	0	0

Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

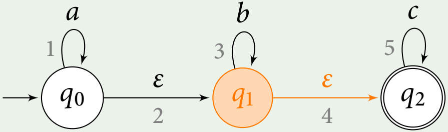
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: *aabbcc*



counters:

2	1	2	0	0

Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

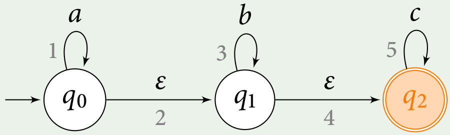
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: $aabbcc$



counters:

2	1	2	1	0

Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

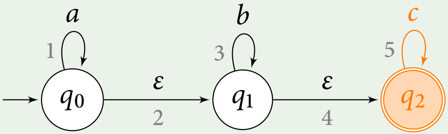
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: $aabbcc$



counters:

2	1	2	1	0

Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

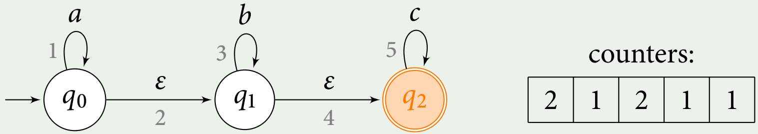
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: *aabbcc*



Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

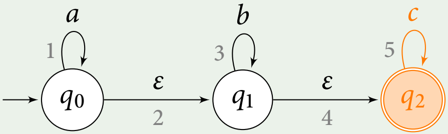
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: *aabbcc*



counters:

2	1	2	1	1

Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

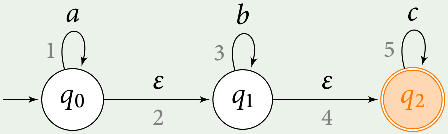
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: *aabbcc*



counters:

2	1	2	1	2

Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

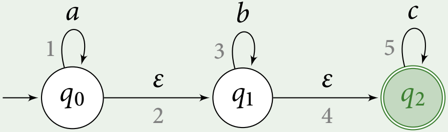
Parikh Automata

Definition

A Parikh automaton is a tuple $(\mathfrak{A}, C)$ where $\mathfrak{A} = (Q, \Sigma, T, q_0, F)$ is an ε-NFA and $C \subseteq \mathbb{N}^T$ is semilinear. A word $w \in \Sigma^*$ is in $L(\mathfrak{A}, C)$ if there is an accepting w-labeled run ρ in $\mathfrak{A}$ with $\Psi(\rho) \in C$.

Example ($C = (0,1,0,1,0) + \{(1,0,1,0,1)\}^*$)

check: *aabbcc*



counters:

2	1	2	1	2

Theorem

The emptiness problem for Parikh automata (where the target sets are given by existential Presburger formulas) is NP*-complete.*

Main Theorem 2

Theorem

The regular separability problem for Parikh automata is coNP*-complete.*

- Decidability was already known [Clemente et al. @ ICALP 2017], but complexity was unknown.
- coNP-hardness:
  - Reduction from emptiness problem for Parikh automata.
  - For a given Parikh automaton $(\mathfrak{A}, C)$, $L(\mathfrak{A}, C)$ is separable from Σ^* if, and only if, $L(\mathfrak{A}, C)$ is empty.

Proof Plan

- Again, we prove that regular inseparability is in NP.
- Let $(\mathfrak{A}, C)$ and $(\mathfrak{B}, D)$ be two Parikh automata.
  1. Construct in polynomial time a DFA $\mathfrak{C}$ and semilinear sets E_1, E_2 such that $L(\mathfrak{A}, C)$ and $L(\mathfrak{B}, D)$ are regularly separable if, and only if, $L(\mathfrak{C}, E_1)$ and $L(\mathfrak{C}, E_2)$ are regularly separable.
  2. There are hyperlinear sets R, S such that $L(\mathfrak{C}, E_1)$ and $L(\mathfrak{C}, E_2)$ are regularly separable if, and only if, R and S are separable by a recognizable set.
     - In R and S we count the simple cycles of $\mathfrak{C}$ on accepting runs.
     - Recall that $R = A + U^*$ and $S = B + V^*$ are inseparable iff $(A + U^* + V_J^*) \cap (B + V^* + U_J^*) \neq \emptyset$
     - Attention: The number of simple cycles in $\mathfrak{C}$ can be exponential!
  3. Guess and verify the set J (or actually the participating transitions) in polynomial time.
  4. Construct from $\mathfrak{C}, E_1, E_2$ in polynomial time a Parikh automaton $(\mathfrak{D}, F)$ such that $L(\mathfrak{D}, F) \neq \emptyset$ iff $(A + U^* + V_J^*) \cap (B + V^* + U_J^*) \neq \emptyset$. □

Thank you!