Complexity of Parikh's Theorem

Anthony Widjaja Lin

Yale-NUS College, Singapore

Classical automata theory

$$w = abaabba$$

$$T = \begin{array}{ccccc} & & a & & \\ & \swarrow & & \searrow & \\ & b & & & b \\ \swarrow & & \searrow & & \\ a & & & a & \end{array}$$

Introduction

Classical automata theory

$$w = abaabba$$

$$T = \begin{array}{ccc} & a & \\ b & & b \\ a & & a \end{array}$$

Parikh (1961) suggested to remove ordering

$$\mathcal{P}(w) = (4, 3)$$

$$\mathcal{P}(T) = (3, 2)$$

Introduction

Classical automata theory

$$w = abaabba$$

$$T = \begin{array}{ccccc} & & a & & \\ & \swarrow & & \searrow & \\ & b & & & b \\ \swarrow & & \searrow & & \\ a & & & a & \end{array}$$

Parikh (1961) suggested to remove ordering

$$\mathcal{P}(w) = (4, 3) \qquad \mathcal{P}(T) = (3, 2)$$

Q: What's the expressive power of standard automata models "modulo Parikh mapping $\mathcal{P}$" (i.e. treated as sets of vectors)?

Parikh's Theorem

Parikh's Theorem: Parikh images of regular and context-free languages are effectively semilinear.

Parikh's Theorem

Parikh's Theorem: Parikh images of regular and context-free languages are effectively semilinear.

Semilinear sets = Presburger-definable subsets of $\mathbb{N}^k$.

Parikh's Theorem Almost Everywhere

- Verification of concurrent systems
  - Bounded context-switch analysis [Esparza-Ganty'11, Hague-Lin'12]
  - Asynchronous programs [Ganty-Majumdar'10]
  - Message-passing programs [Abdulla-Atig-Cederberg'13]
- Verification of (restrictions of) counter machines:
  - Reversal-bounded verification [Ibarra'79]
  - Flat counter machines [Fribourg-Olsen'97, Comon-Jurski'98]
  - Flattable counter machines [Bardin-Finkel-Leroux-Schnoebelen'05, Leroux-Sutre'05]
- Path logics over graph databases [Barcelo-Libkin-Lin-Woods'12]
- Cryptographic Analysis of C programs [Verma *et al.*'06]

Complexity of Parikh's Theorem

Descriptional Complexity: How succinct are the different automata models for representing semilinear sets?

Complexity of Parikh's Theorem

Descriptional Complexity: How succinct are the different automata models for representing semilinear sets?

i.e. the size of the smallest semilinear sets for NFA, CFG, ...?

Complexity of Parikh's Theorem

Descriptional Complexity: How succinct are the different automata models for representing semilinear sets?

i.e. the size of the smallest semilinear sets for NFA, CFG, ...?

Computational complexity: Can we compute semilinear sets for Parikh images of these models efficiently?

Parikh's Theorem (more precisely)

Parikh Images

- Parikh Image $\mathcal{P}(L)$ of a language L over $\Sigma = \{a_1, \ldots, a_k\}$ is a subset of $\mathbb{N}^k$

Parikh Images

- Parikh Image $\mathcal{P}(L)$ of a language L over $\Sigma = \{a_1, \ldots, a_k\}$ is a subset of $\mathbb{N}^k$
- $L = \{a^n b^n : n \in \mathbb{N}\}$

Parikh Images

- Parikh Image $\mathcal{P}(L)$ of a language L over $\Sigma = \{a_1, \ldots, a_k\}$ is a subset of $\mathbb{N}^k$
- $L = \{a^n b^n : n \in \mathbb{N}\}$
  - $\mathcal{P}(L) = \{(0,0), (1,1), (2,2), (3,3), \ldots\}$

Parikh Images

- Parikh Image $\mathcal{P}(L)$ of a language L over $\Sigma = \{a_1, \ldots, a_k\}$ is a subset of $\mathbb{N}^k$
- $L = \{a^n b^n : n \in \mathbb{N}\}$
  - $\mathcal{P}(L) = \{(0,0), (1,1), (2,2), (3,3), \ldots\}$
- **Note**: $L' = (ab)^*$ and $\mathcal{P}(L) = \mathcal{P}(L')$.

Semilinear Sets

- A linear set (over $\mathbb{N}^k$) is a set of the form

$$L(\mathbf{v}_0; \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}) := \left\{ \mathbf{v}_0 + \sum_{i=1}^{m} a_i \mathbf{v}_i : a_1, \ldots, a_m \in \mathbb{N} \right\}$$

for some offset $\mathbf{v}_0 \in \mathbb{N}^k$ and periods $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{N}^k$ and $m \in \mathbb{N}$.

Semilinear Sets

- A linear set (over $\mathbb{N}^k$) is a set of the form

$$L(\mathbf{v}_0; \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}) := \left\{ \mathbf{v}_0 + \sum_{i=1}^{m} a_i \mathbf{v}_i : a_1, \ldots, a_m \in \mathbb{N} \right\}$$

  for some offset $\mathbf{v}_0 \in \mathbb{N}^k$ and periods $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{N}^k$ and $m \in \mathbb{N}$.
- **Example**: $\{(i, i) : i \in \mathbb{N}\} = L((0, 0); \{(1, 1)\})$

Semilinear Sets

- A linear set (over $\mathbb{N}^k$) is a set of the form

$$L(\mathbf{v}_0; \{\mathbf{v}_1, \ldots, \mathbf{v}_m\}) := \left\{ \mathbf{v}_0 + \sum_{i=1}^{m} a_i \mathbf{v}_i : a_1, \ldots, a_m \in \mathbb{N} \right\}$$

  for some offset $\mathbf{v}_0 \in \mathbb{N}^k$ and periods $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{N}^k$ and $m \in \mathbb{N}$.
- **Example**: $\{(i, i) : i \in \mathbb{N}\} = L((0, 0); \{(1, 1)\})$
- A semilinear set over $\mathbb{N}^k$ is a finite union of linear sets.

Complexity of Parikh's Theorem

Theorem (Parikh): Parikh images of context-free and regular languages are effectively semilinear.

- Descriptional complexity: size of smallest description
- Computational complexity: how efficient to compute

Complexity of Parikh's Theorem

Theorem (Parikh): Parikh images of context-free and regular languages are effectively semilinear.

- Descriptional complexity: size of smallest description
- Computational complexity: how efficient to compute

Parikh's original proof gives exponential complexity bound with linearly many periods for each linear set!

CFG Case

Esparza's Theorem

$$S \rightarrow A \tag{1}$$
$$A \rightarrow aBb \tag{2}$$
$$B \rightarrow aAb \tag{3}$$
$$B \rightarrow \varepsilon \tag{4}$$

Esparza's Theorem

$$S \rightarrow A \quad (1)$$
$$A \rightarrow aBb \quad (2)$$
$$B \rightarrow aAb \quad (3)$$
$$B \rightarrow \varepsilon \quad (4)$$

(Esparza'97): a sufficient and necessary condition for a multiset $X \subseteq \{1, \ldots, 4\}$ to be realisable based on (C1) flow condition, and (C2) connectivity condition.

Esparza's Theorem

$$S \rightarrow A \quad (1)$$
$$A \rightarrow aBb \quad (2)$$
$$B \rightarrow aAb \quad (3)$$
$$B \rightarrow \varepsilon \quad (4)$$

(Esparza'97): a sufficient and necessary condition for a multiset $X \subseteq \{1, \ldots, 4\}$ to be realisable based on (C1) flow condition, and (C2) connectivity condition.

The following are not realisable:

- $\{1^2, 2, 4\}$ — need at least two S (C1)

Esparza's Theorem

$$S \rightarrow A \quad (1)$$
$$A \rightarrow aBb \quad (2)$$
$$B \rightarrow aAb \quad (3)$$
$$B \rightarrow \varepsilon \quad (4)$$

(Esparza'97): a sufficient and necessary condition for a multiset $X \subseteq \{1, \ldots, 4\}$ to be realisable based on (C1) flow condition, and (C2) connectivity condition.

The following are not realisable:

- $\{1^2, 2, 4\}$ — need at least two S (C1)
- $\{1, 2, 3^7, 4\}$ — need at least seven B (C1)

Esparza's Theorem

$$S \rightarrow A \quad (1)$$
$$A \rightarrow aBb \quad (2)$$
$$B \rightarrow aAb \quad (3)$$
$$B \rightarrow \varepsilon \quad (4)$$

(Esparza'97): a sufficient and necessary condition for a multiset $X \subseteq \{1, \ldots, 4\}$ to be realisable based on (C1) flow condition, and (C2) connectivity condition.

The following are not realisable:

- $\{1^2, 2, 4\}$ — need at least two S (C1)
- $\{1, 2, 3^7, 4\}$ — need at least seven B (C1)
- $\{1, 3, 4\}$ — 3 cannot be fired without 2 (C2)

Esparza's Theorem

- Flow and connectivity conditions are expressible as an exponential sized semilinear set (Esparza'97)

Esparza's Theorem

- Flow and connectivity conditions are expressible as an exponential sized semilinear set (Esparza'97)
- Flow and connectivity conditions are expressible as a linear sized existential Presburger formula (Verma *et al.*'05)

Esparza's Theorem

- Flow and connectivity conditions are expressible as an exponential sized semilinear set (Esparza'97)
- Flow and connectivity conditions are expressible as a linear sized existential Presburger formula (Verma *et al.*'05)

n.b. checking existential Presburger formulas are NP-complete: can use fast SMT solvers.

Esparza's Theorem

- Flow and connectivity conditions are expressible as an exponential sized semilinear set (Esparza'97)
- Flow and connectivity conditions are expressible as a linear sized existential Presburger formula (Verma *et al.*'05)

n.b. checking existential Presburger formulas are NP-complete: can use fast SMT solvers.

Note: This has been successfully used in many applications in infinite-state verification.

Lower bound for semilinear sets for CFGs

Proposition: There is an infinite sequence $\{G_n\}_{n\in\mathbb{N}}$ of CFGs over $\Sigma = \{a\}$ s.t. $\mathcal{P}(L(G_n))$ must have at least $2^{\Omega(|G_n|)}$ linear sets.

Lower bound for semilinear sets for CFGs

Proposition: There is an infinite sequence $\{G_n\}_{n\in\mathbb{N}}$ of CFGs over $\Sigma = \{a\}$ s.t. $\mathcal{P}(L(G_n))$ must have at least $2^{\Omega(|G_n|)}$ linear sets.

The CFG G_n generates $\{a^j : j \in [0, 2^n - 1]\}$:

Lower bound for semilinear sets for CFGs

Proposition: There is an infinite sequence $\{G_n\}_{n\in\mathbb{N}}$ of CFGs over $\Sigma = \{a\}$ s.t. $\mathcal{P}(L(G_n))$ must have at least $2^{\Omega(|G_n|)}$ linear sets.

The CFG G_n generates $\{a^j : j \in [0, 2^n - 1]\}$: ($2^n$ linear sets!!)

Lower bound for semilinear sets for CFGs

Proposition: There is an infinite sequence $\{G_n\}_{n\in\mathbb{N}}$ of CFGs over $\Sigma = \{a\}$ s.t. $\mathcal{P}(L(G_n))$ must have at least $2^{\Omega(|G_n|)}$ linear sets.

The CFG G_n generates $\{a^j : j \in [0, 2^n - 1]\}$: ($2^n$ linear sets!!)

$$
\begin{aligned}
S &\to A_0 \dots A_{n-1} \\
A_i &\to \varepsilon \quad \text{for each } 0 \le i < n \\
A_i &\to B_i \quad \text{for each } 0 \le i < n \\
B_i &\to B_{i-1}B_{i-1} \quad \text{for each } 0 < i < n \\
B_0 &\to a
\end{aligned}
$$

Lower bound for semilinear sets for CFGs

Proposition: There is an infinite sequence $\{G_n\}_{n\in\mathbb{N}}$ of CFGs over $\Sigma = \{a\}$ s.t. $\mathcal{P}(L(G_n))$ must have at least $2^{\Omega(|G_n|)}$ linear sets.

The CFG G_n generates $\{a^j : j \in [0, 2^n - 1]\}$: ($2^n$ linear sets!!)

$$
\begin{array}{rcl}
S & \to & A_0 \dots A_{n-1} \\
A_i & \to & \varepsilon \quad \text{for each } 0 \leq i < n \\
A_i & \to & B_i \quad \text{for each } 0 \leq i < n \\
B_i & \to & B_{i-1} B_{i-1} \quad \text{for each } 0 < i < n \\
B_0 & \to & a
\end{array}
$$

n.b. this kind of encoding for CFG is from (Stockmeyer-Meyer'73)

NFA Case

Exponential lower bound for DFAs

Proposition: There is an infinite sequence $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ of DFAs s.t. $\mathcal{P}(L(\mathcal{A}_n))$ must have at least $2^{\Omega(|\mathcal{A}_n|)}$ linear sets.

Exponential lower bound for DFAs

Proposition: There is an infinite sequence $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ of DFAs s.t. $\mathcal{P}(L(\mathcal{A}_n))$ must have at least $2^{\Omega(|\mathcal{A}_n|)}$ linear sets.

$\mathcal{A}_n$ is over $\Sigma_n := \{a_1, \ldots, a_{n+1}\}$:

$$\underbrace{\Sigma_n \cdot \Sigma_n \cdot \cdots \cdot \Sigma_n}_{n \text{ copies}}$$

Exponential lower bound for DFAs

Proposition: There is an infinite sequence $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ of DFAs s.t. $\mathcal{P}(L(\mathcal{A}_n))$ must have at least $2^{\Omega(|\mathcal{A}_n|)}$ linear sets.

$\mathcal{A}_n$ is over $\Sigma_n := \{a_1, \ldots, a_{n+1}\}$:

$$\underbrace{\Sigma_n \cdot \Sigma_n \cdot \ldots \cdot \Sigma_n}_{n \text{ copies}}$$

$\mathcal{P}(L(\mathcal{A}_n))$ contains each $(r_1, \ldots, r_{n+1})$ s.t. $\sum_{i=1}^{n+1} r_i = n$.

Exponential lower bound for DFAs

Proposition: There is an infinite sequence $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ of DFAs s.t. $\mathcal{P}(L(\mathcal{A}_n))$ must have at least $2^{\Omega(|\mathcal{A}_n|)}$ linear sets.

$\mathcal{A}_n$ is over $\Sigma_n := \{a_1, \ldots, a_{n+1}\}$:

$$\underbrace{\Sigma_n \cdot \Sigma_n \cdot \ldots \cdot \Sigma_n}_{n \text{ copies}}$$

$\mathcal{P}(L(\mathcal{A}_n))$ contains each $(r_1, \ldots, r_{n+1})$ s.t. $\sum_{i=1}^{n+1} r_i = n$. There are $\binom{2n}{n} \geq \frac{2^{2n-1}}{\sqrt{n}}$ of these.

Exponential lower bound for DFAs

Proposition: There is an infinite sequence $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ of DFAs s.t. $\mathcal{P}(L(\mathcal{A}_n))$ must have at least $2^{\Omega(|\mathcal{A}_n|)}$ linear sets.

$\mathcal{A}_n$ is over $\Sigma_n := \{a_1, \ldots, a_{n+1}\}$:

$$\underbrace{\Sigma_n \cdot \Sigma_n \cdot \ldots \cdot \Sigma_n}_{n \text{ copies}}$$

$\mathcal{P}(L(\mathcal{A}_n))$ contains each $(r_1, \ldots, r_{n+1})$ s.t. $\sum_{i=1}^{n+1} r_i = n$. There are $\binom{2n}{n} \geq \frac{2^{2n-1}}{\sqrt{n}}$ of these.

Note: Σ_n grows with n

What about a fixed alphabet size?

Unary alphabet case: Chrobak-Martinez Theorem

Let us assume that the alphabet is unary, i.e., $\Sigma = \{a\}$.

Unary alphabet case: Chrobak-Martinez Theorem

Let us assume that the alphabet is unary, i.e., $\Sigma = \{a\}$.

Theorem (Chrobak-Martinez): Descriptional and computational complexity of Parikh Images of unary regular languages are polynomial.

Unary alphabet case: Chrobak-Martinez Theorem

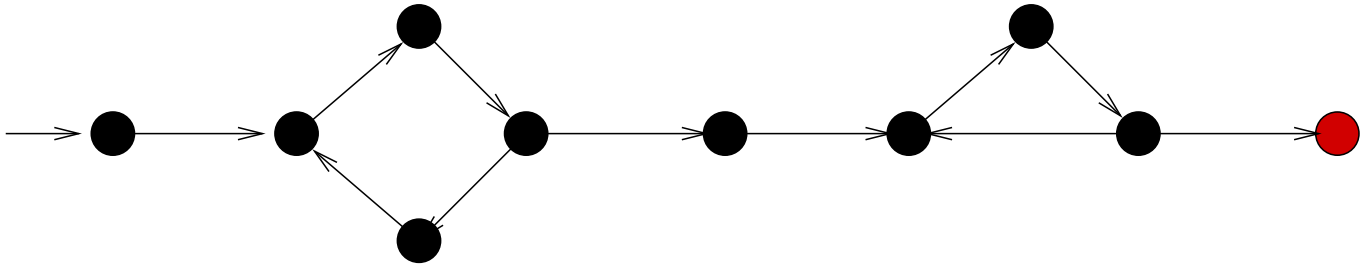
Let us assume that the alphabet is unary, i.e., $\Sigma = \{a\}$.

Theorem (Chrobak-Martinez): Descriptional and computational complexity of Parikh Images of unary regular languages are polynomial.

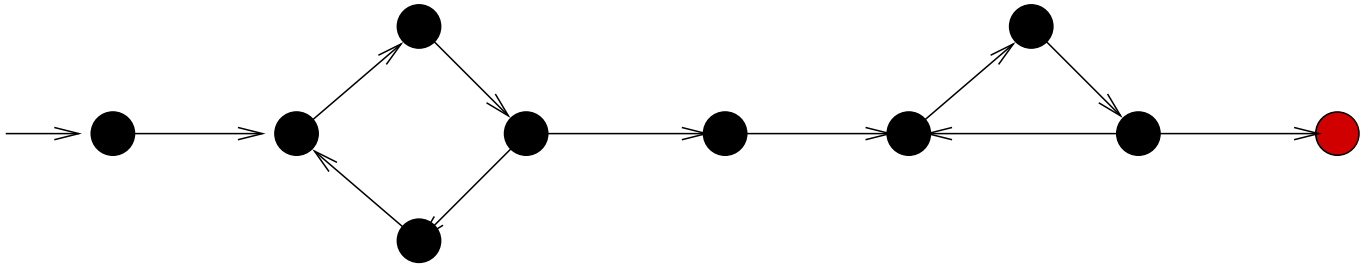
Note:

- quadratically many union of arithmetic progressions with periods of linear size suffice.

Chrobak-Martinez Theorem in Action

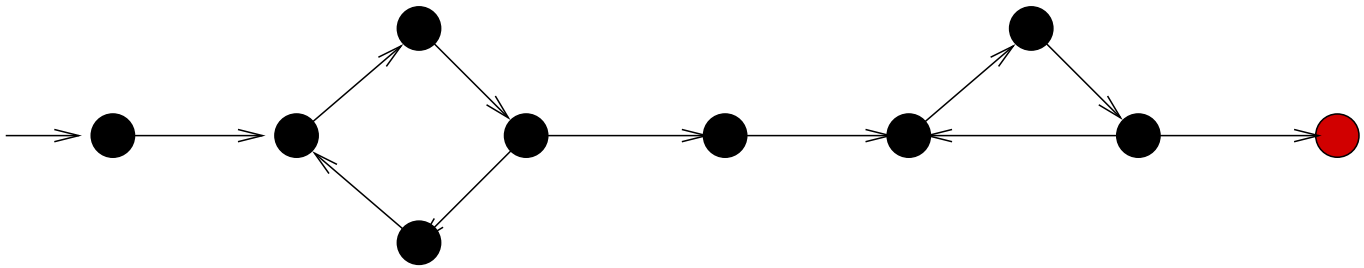


Chrobak-Martinez Theorem in Action



Parikh image of $L(\mathcal{A})$ is $8 + 4\mathbb{N} + 3\mathbb{N}$

Chrobak-Martinez Theorem in Action



Parikh image of $L(\mathcal{A})$ is $8 + 4\mathbb{N} + 3\mathbb{N}$ which is equal to

$$(8 + 4\mathbb{N}) \cup (11 + 4\mathbb{N}) \cup (14 + 4\mathbb{N}) \cup (17 + 4\mathbb{N})$$

Generalised Chrobak-Martinez Theorem

Let $\Sigma := \{a_1, \ldots, a_k\}$ for fixed $k \in \mathbb{Z}_{>0}$.

Generalised Chrobak-Martinez Theorem

Let $\Sigma := \{a_1, \ldots, a_k\}$ for fixed $k \in \mathbb{Z}_{>0}$.

Theorem (Kopczynski & Lin'10): Descriptional and computational complexity of Parikh Images of NFAs are polynomial.

- union of polynomially many linear sets with at most k polynomially-bounded periods
- Complexities are exponential in k
- Generalizes Chrobak-Martinez Theorem (case $k = 1$).

Outline of proof

- Normal-Form Theorem for Semilinear sets.
- Normal-Form Theorem for Parikh images of NFA

Normal Form Theorem for Semilinear Sets

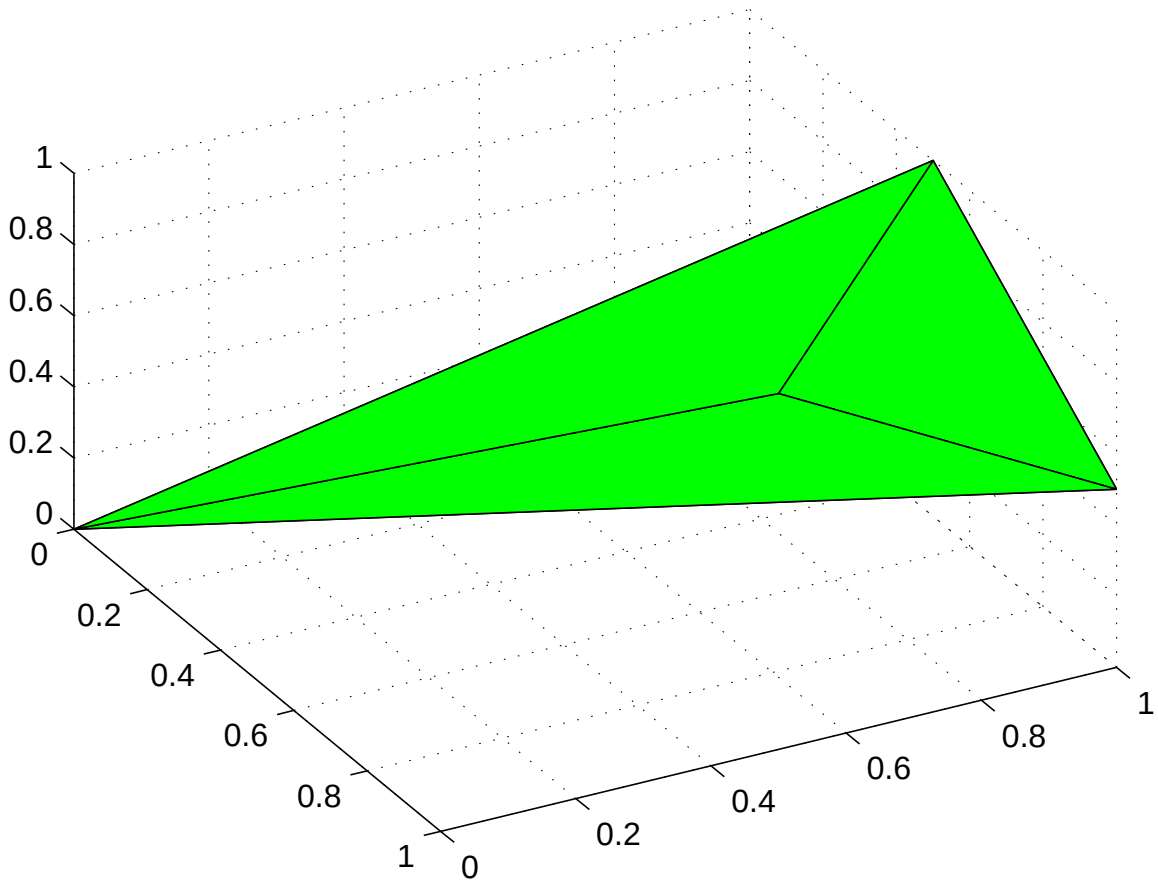
Digression to Convex Geometry: Real Cones

Given $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \subseteq \mathbb{R}^k$, define the real cone over V:

$$\textbf{cone}(V) := \{\Sigma_{i=1}^{n} a_i \mathbf{v}_i : a_i \in \mathbb{R}_{\geq 0}\}.$$

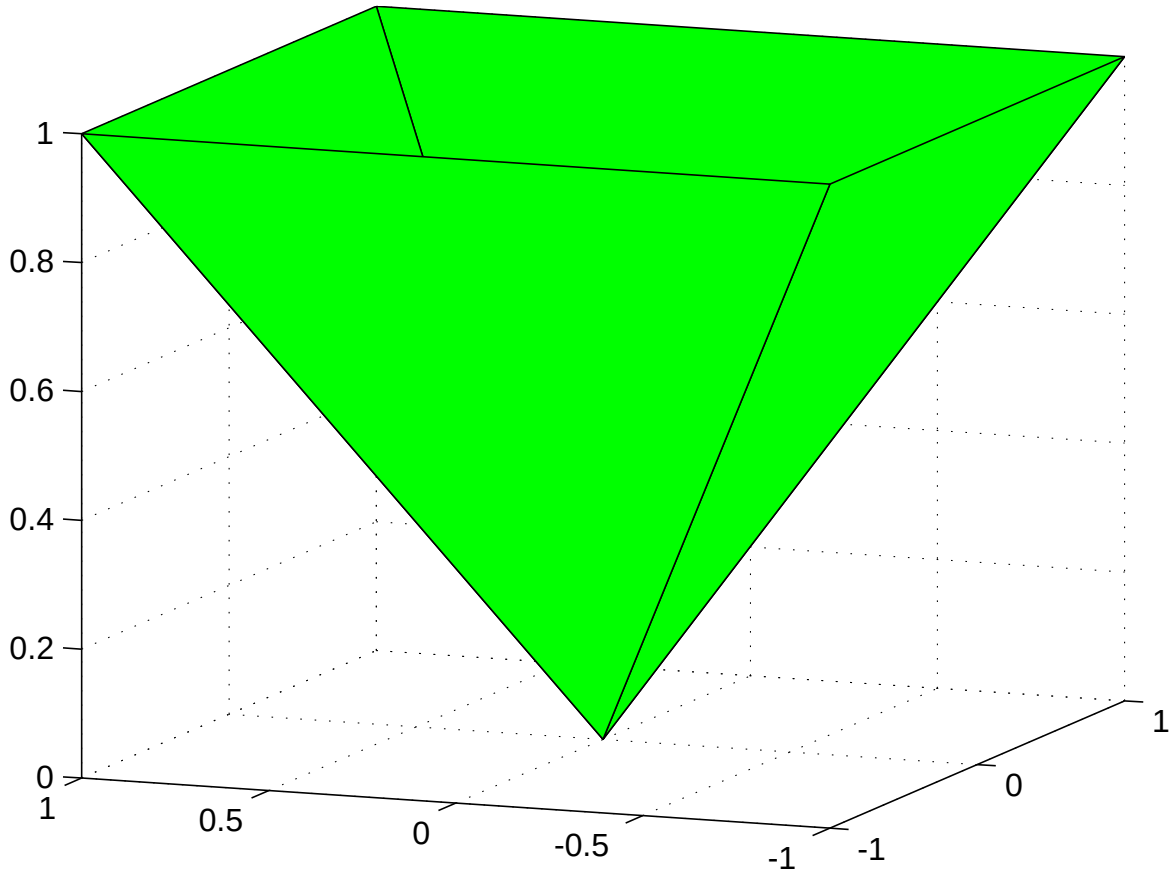
Note: akin to definition of vector subspace of $\mathbb{R}^k$ "spanned" by some set V (but ...)

Examples



Real cone of $(1, 1, 0.5), (1, 0.5, 1), (0.5, 1, 1)$

Examples



Real cone over $\{1, -1\} \times \{1, -1\} \times \{1\}$

Caratheodory's theorem

Theorem: Given a finite $V \subseteq \mathbb{R}^k$ of rank $d \leq k$, we have

$$\textbf{cone}(V) = \bigcup_{V' \subseteq V, |V'| = \|V'\| = d} \textbf{cone}(V').$$

- *Cones over $\mathbb{R}^k$ can be decomposed into smaller subcones with $\leq k$ vertices*

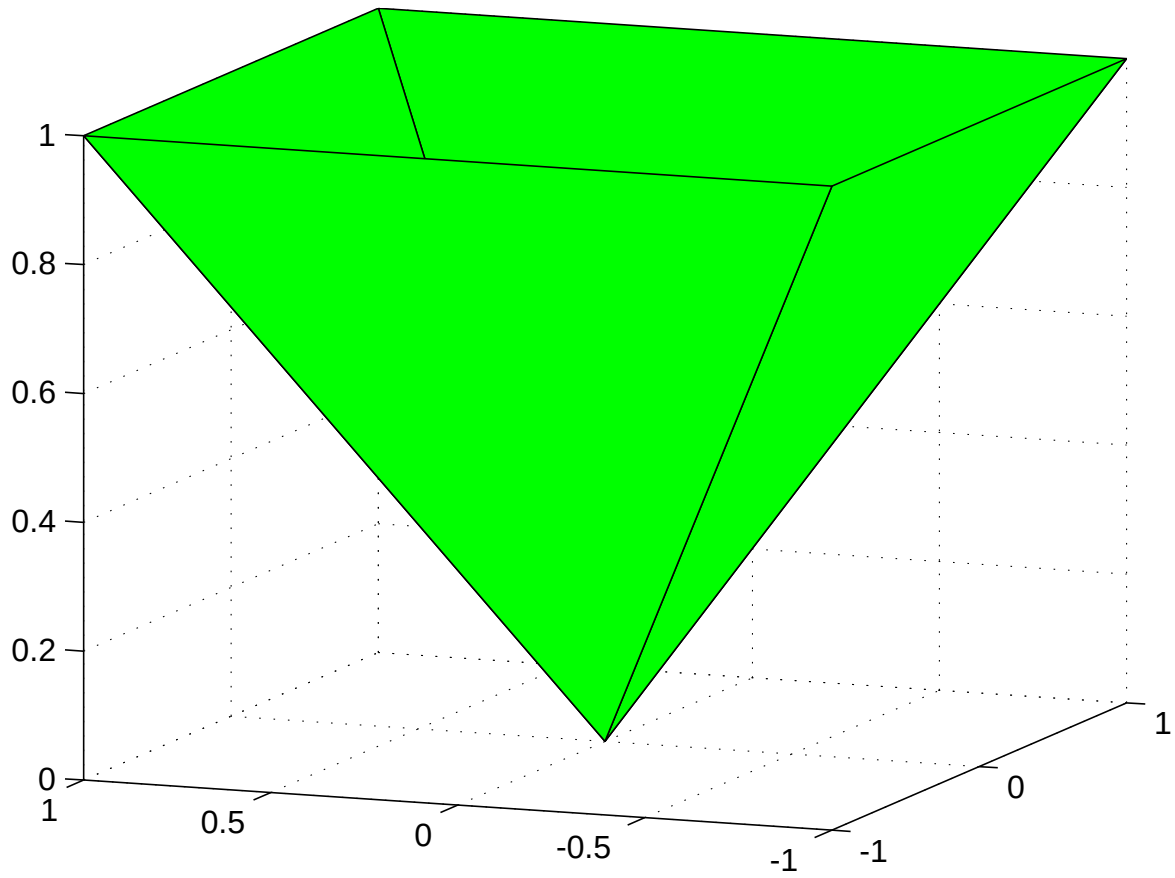
Caratheodory's theorem

Theorem: Given a finite $V \subseteq \mathbb{R}^k$ of rank $d \leq k$, we have

$$\textbf{cone}(V) = \bigcup_{V' \subseteq V, |V'| = \|V'\| = d} \textbf{cone}(V').$$

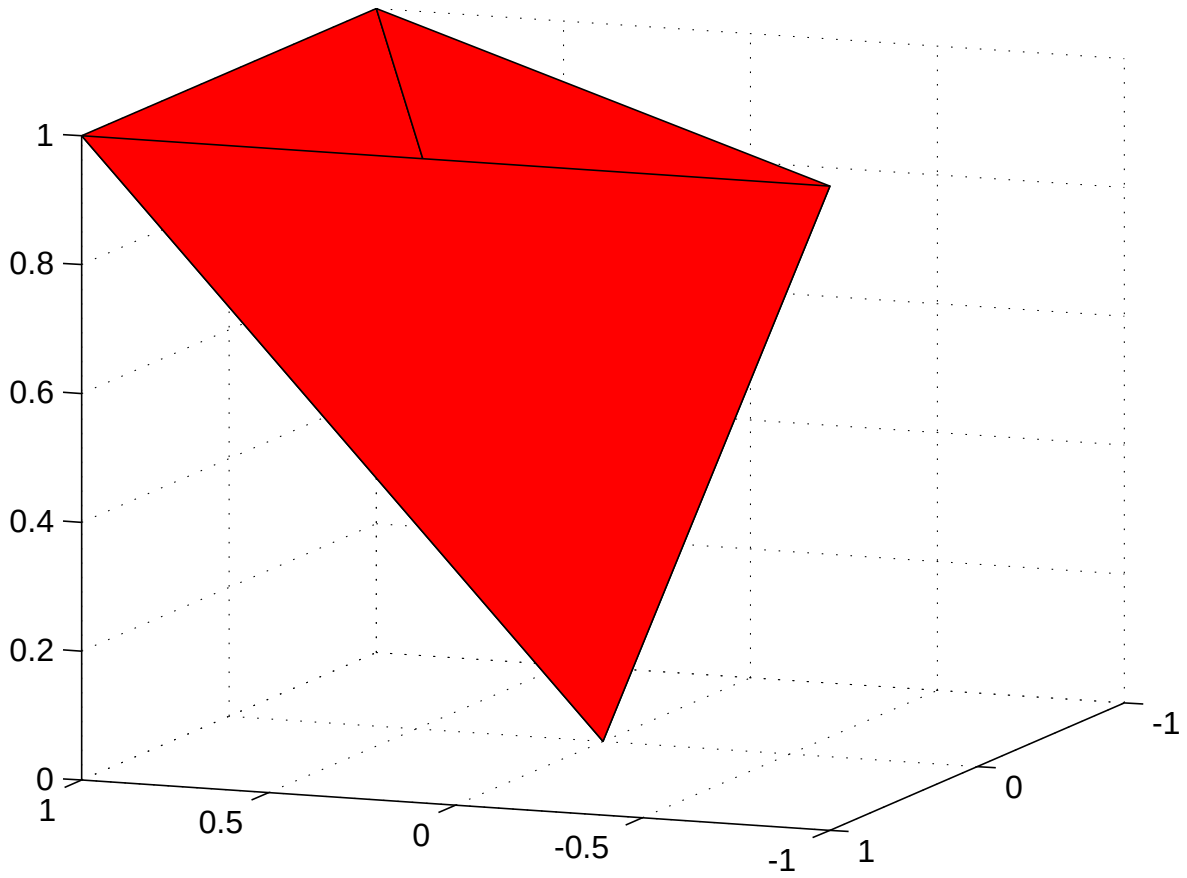
- *Cones over $\mathbb{R}^k$ can be decomposed into smaller subcones with $\leq k$ vertices*
- ***Note**: if k is fixed, there are only polynomially many such subcones.*

Caratheodory's theorem (example)



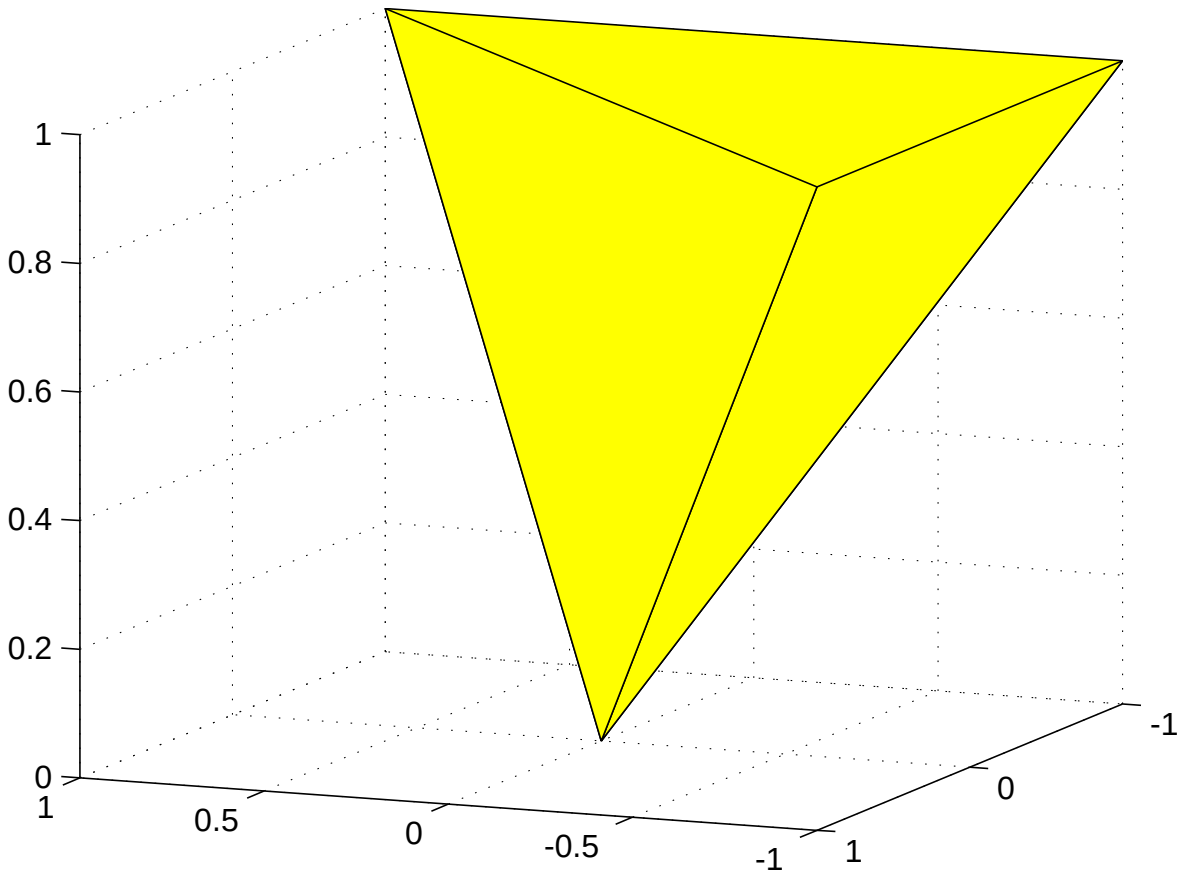
Real cone over $\{1, -1\} \times \{1, -1\} \times \{1\}$

Caratheodory's theorem (example)



Real cone over $(1, 1, 1), (1, -1, 1), (-1, 1, 1)$

Caratheodory's theorem (example)



Real cone over $(-1,-1,1),(1,-1,1),(-1,1,1)$

Caratheodory-like theorem for linear sets

- **Question**: Does the integer version of Caratheodory's theorem hold?

Caratheodory-like theorem for linear sets

- **Question**: Does the integer version of Caratheodory's theorem hold?
- Unfortunately, no!

Caratheodory-like theorem for linear sets

- **Question**: Does the integer version of Caratheodory's theorem hold?
- Unfortunately, no!
- Fortunately, it does if you allow nonzero offsets :)

Caratheodory-like theorem for linear sets

Theorem: Fix $k \in \mathbb{N}$. Given a finite $V \subseteq \mathbb{Z}^k$ of rank $d \leq k$, we can compute in pseudopolynomial time linear sets $L(\mathbf{w}_1; S_1), \ldots, L(\mathbf{w}_s; S_s)$ s.t. each $S_i \subseteq V$ and $|S_i| = d$

$$L(\mathbf{0}; V) = \bigcup_{i=1}^{s} L(\mathbf{w}_i; S_i).$$

Each number in $\mathbf{w}_i$ is polynomially large (in unary)

Caratheodory-like theorem for linear sets

Theorem: Fix $k \in \mathbb{N}$. Given a finite $V \subseteq \mathbb{Z}^k$ of rank $d \leq k$, we can compute in pseudopolynomial time linear sets $L(\mathbf{w}_1; S_1), \ldots, L(\mathbf{w}_s; S_s)$ s.t. each $S_i \subseteq V$ and $|S_i| = d$

$$L(\mathbf{0}; V) = \bigcup_{i=1}^{s} L(\mathbf{w}_i; S_i).$$

Each number in $\mathbf{w}_i$ is polynomially large (in unary)

When $V \subseteq \mathbb{N}^k$, each $\mathbf{w}_i \in \mathbb{N}^k$

Caratheodory-like theorem for linear sets

Theorem: Fix $k \in \mathbb{N}$. Given a finite $V \subseteq \mathbb{Z}^k$ of rank $d \leq k$, we can compute in pseudopolynomial time linear sets $L(\mathbf{w}_1; S_1), \ldots, L(\mathbf{w}_s; S_s)$ s.t. each $S_i \subseteq V$ and $|S_i| = d$

$$L(\mathbf{0}; V) = \bigcup_{i=1}^{s} L(\mathbf{w}_i; S_i).$$

Each number in $\mathbf{w}_i$ is polynomially large (in unary)

When $V \subseteq \mathbb{N}^k$, each $\mathbf{w}_i \in \mathbb{N}^k$

The same holds when the initial offset is $\mathbf{v} \neq \mathbf{0}$

Caratheodory-like theorem for linear sets

Theorem: Fix $k \in \mathbb{N}$. Given a finite $V \subseteq \mathbb{Z}^k$ of rank $d \leq k$, we can compute in pseudopolynomial time linear sets $L(\mathbf{w}_1; S_1), \ldots, L(\mathbf{w}_s; S_s)$ s.t. each $S_i \subseteq V$ and $|S_i| = d$

$$L(\mathbf{0}; V) = \bigcup_{i=1}^{s} L(\mathbf{w}_i; S_i).$$

Each number in $\mathbf{w}_i$ is polynomially large (in unary)

When $V \subseteq \mathbb{N}^k$, each $\mathbf{w}_i \in \mathbb{N}^k$

The same holds when the initial offset is $\mathbf{v} \neq \mathbf{0}$

Caratheodory-like theorem for linear sets

Theorem: Fix $k \in \mathbb{N}$. Given a finite $V \subseteq \mathbb{Z}^k$ of rank $d \leq k$, we can compute in pseudopolynomial time linear sets $L(\mathbf{w}_1; S_1), \ldots, L(\mathbf{w}_s; S_s)$ s.t. each $S_i \subseteq V$ and $|S_i| = d$

$$L(\mathbf{0}; V) = \bigcup_{i=1}^{s} L(\mathbf{w}_i; S_i).$$

Use Caratheodory's theorem

Caratheodory-like theorem for linear sets

Theorem: Fix $k \in \mathbb{N}$. Given a finite $V \subseteq \mathbb{Z}^k$ of rank $d \leq k$, we can compute in pseudopolynomial time linear sets $L(\mathbf{w}_1; S_1), \ldots, L(\mathbf{w}_s; S_s)$ s.t. each $S_i \subseteq V$ and $|S_i| = d$

$$L(\mathbf{0}; V) = \bigcup_{i=1}^{s} L(\mathbf{w}_i; S_i).$$

Use Caratheodory's theorem

Use bounds from integer programming (Papadimitriou'83) for estimating the biggest entries in $\mathbf{w}_i$'s

Caratheodory-like theorem for linear sets

Theorem: Fix $k \in \mathbb{N}$. Given a finite $V \subseteq \mathbb{Z}^k$ of rank $d \leq k$, we can compute in pseudopolynomial time linear sets $L(\mathbf{w}_1; S_1), \ldots, L(\mathbf{w}_s; S_s)$ s.t. each $S_i \subseteq V$ and $|S_i| = d$

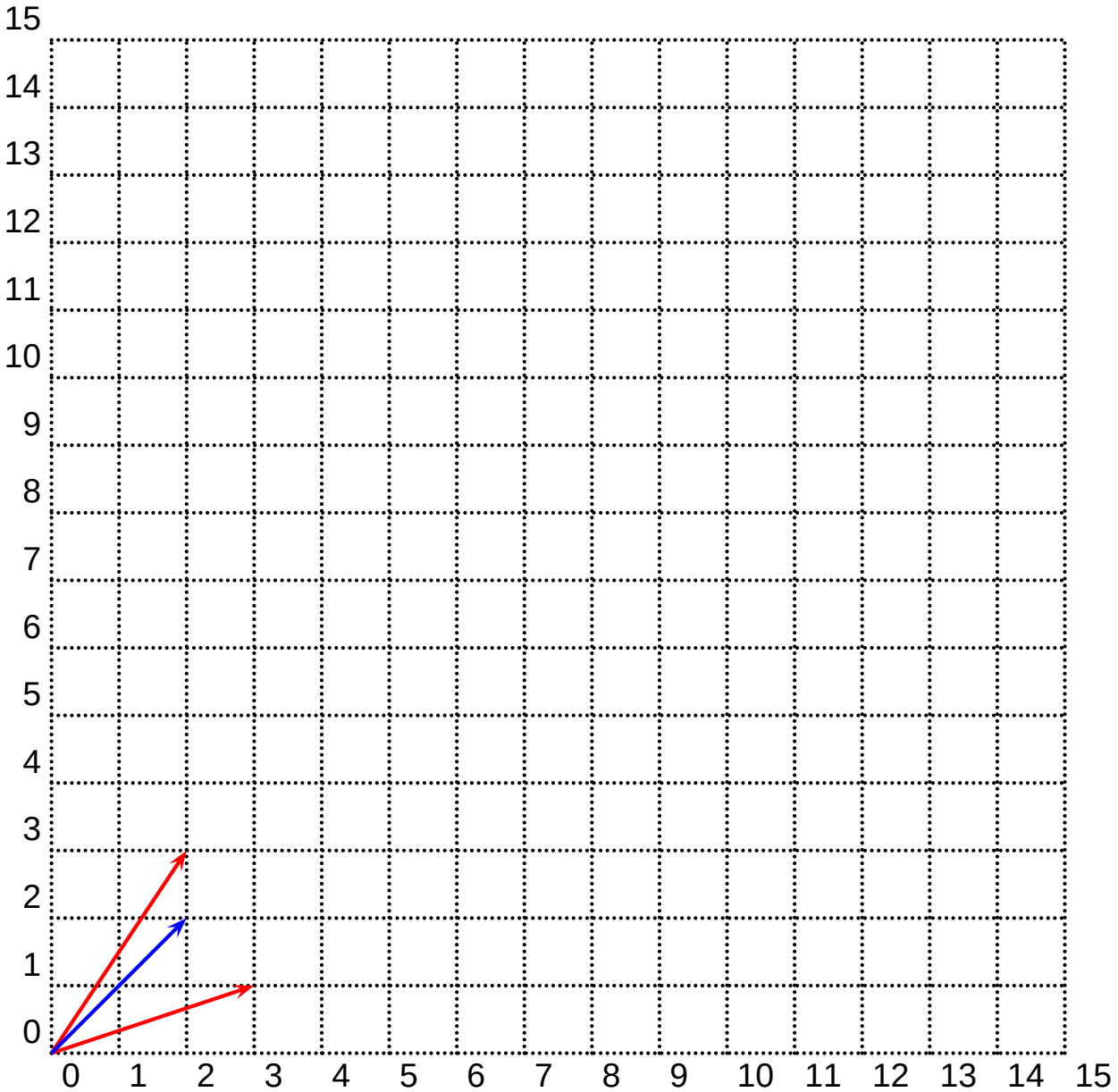
$$L(\mathbf{0}; V) = \bigcup_{i=1}^{s} L(\mathbf{w}_i; S_i).$$

Use Caratheodory's theorem

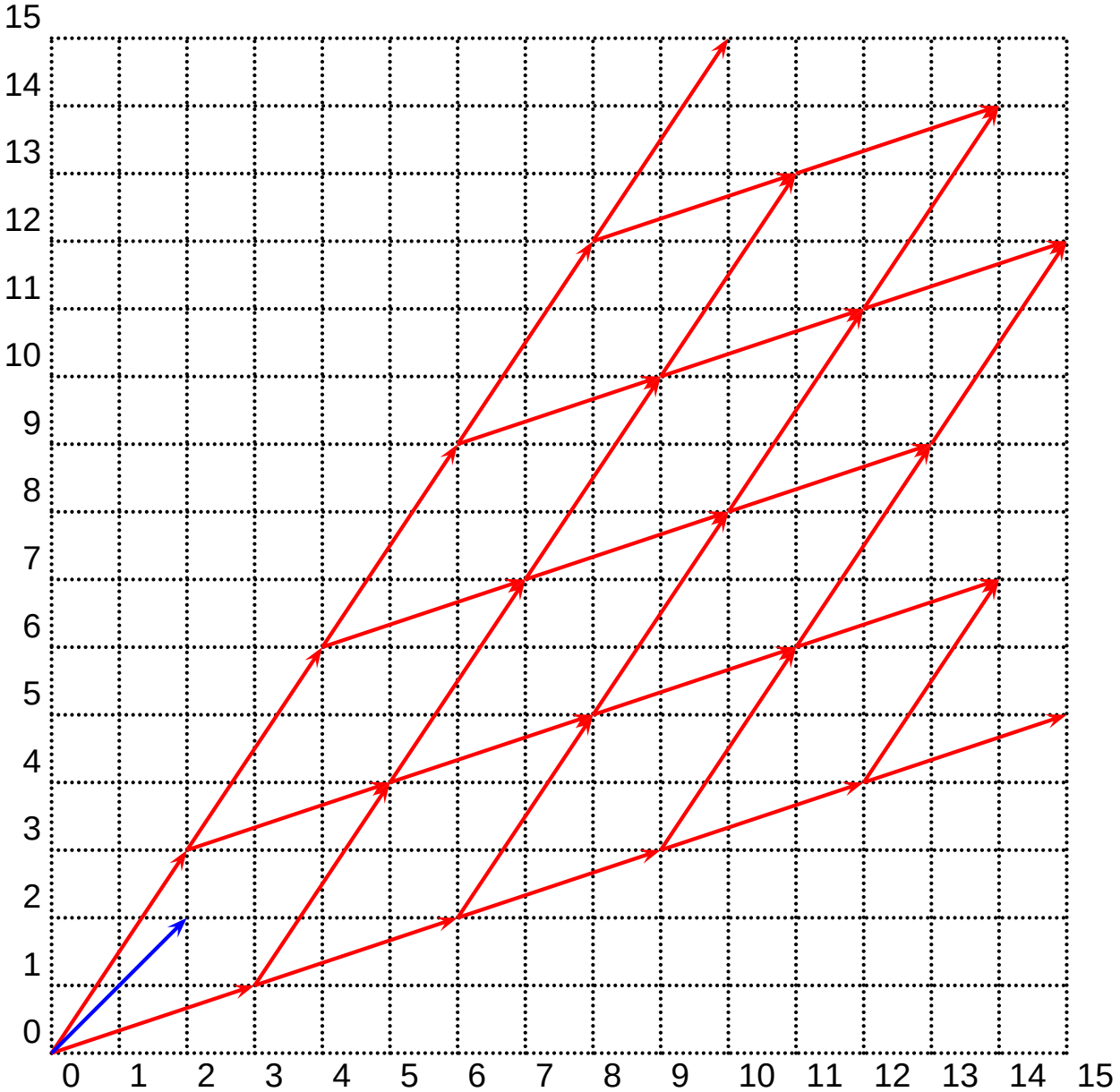
Use bounds from integer programming (Papadimitriou'83) for estimating the biggest entries in $\mathbf{w}_i$'s

Use dynamic programming to compute all $\mathbf{w}_i$'s

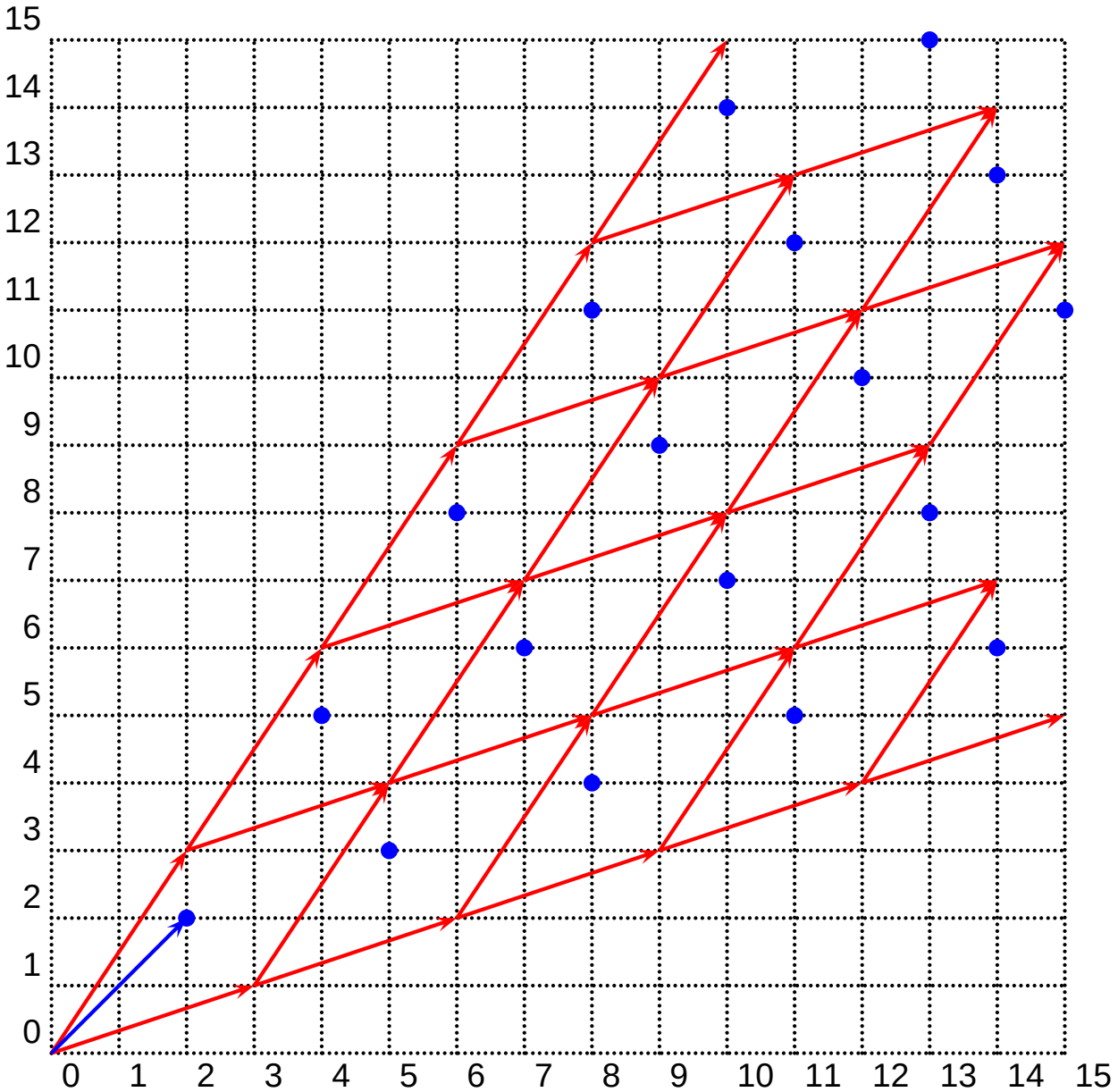
Intuition for dimension 2



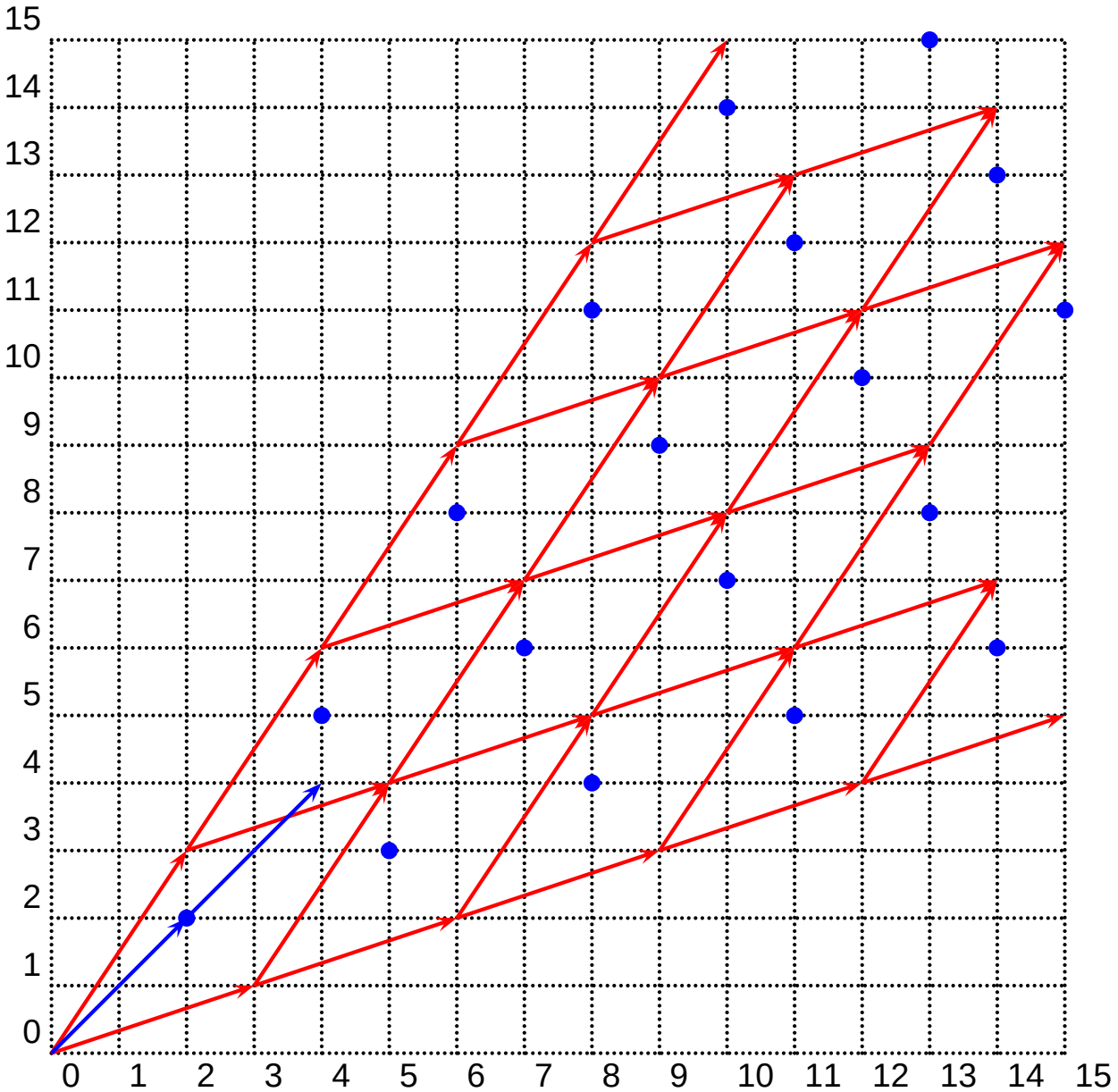
Intuition for dimension 2



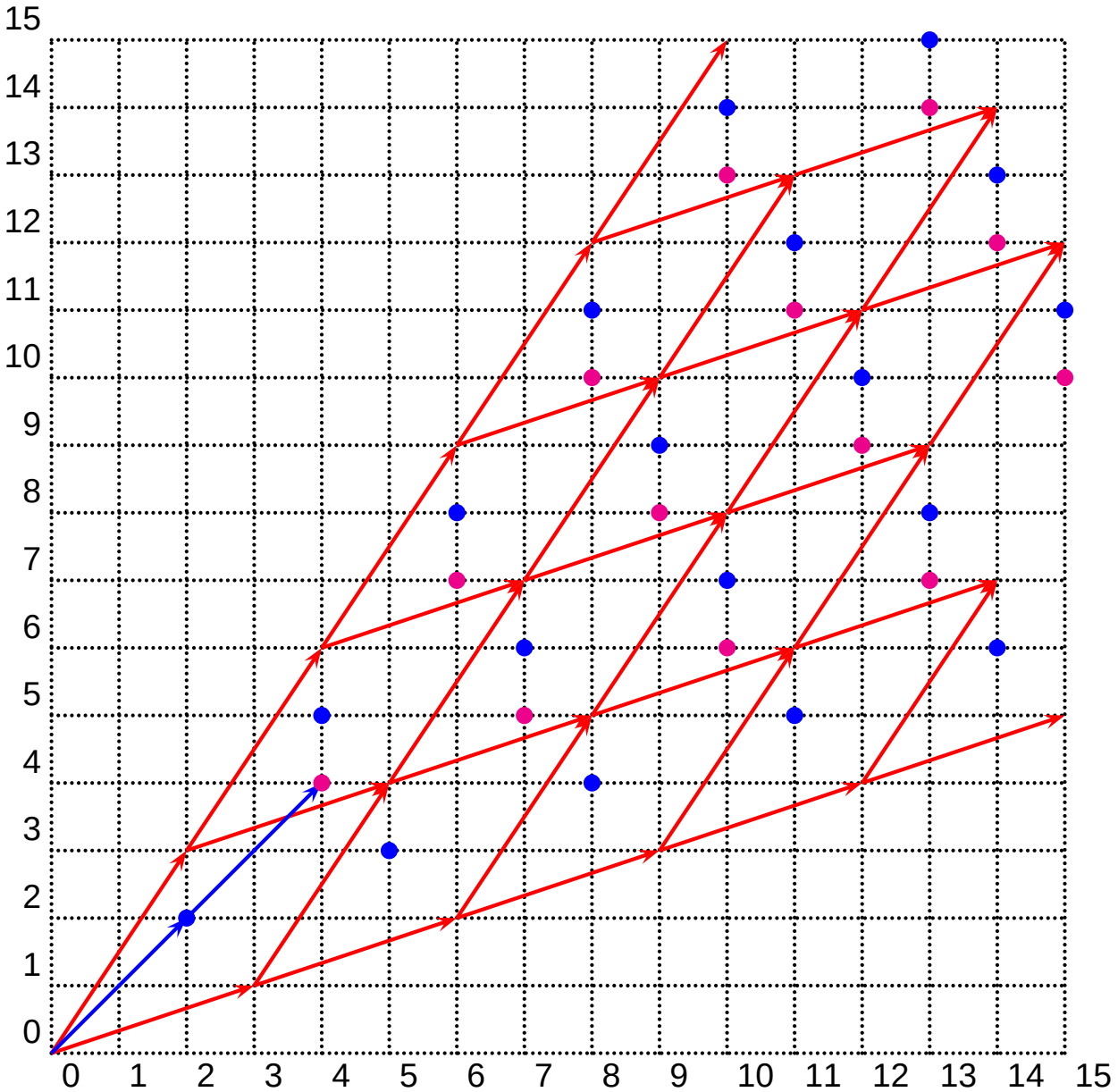
Intuition for dimension 2



Intuition for dimension 2



Intuition for dimension 2



Intuition for dimension 2

Key observations:

- Each parallelogram has a quadratically many integer points

Intuition for dimension 2

Key observations:

- Each parallelogram has a quadratically many integer points
- There is a natural ordering on these parallelograms:
  - The parallelogram above or to the right of a parallelogram is larger.

Intuition for dimension 2

Key observations:

- Each parallelogram has a quadratically many integer points
- There is a natural ordering on these parallelograms:
  - The parallelogram above or to the right of a parallelogram is larger.
- Once a point “appears”, it stays in the larger parallelograms.

Intuition for dimension 2

Key observations:

- Each parallelogram has a quadratically many integer points
- There is a natural ordering on these parallelograms:
  - The parallelogram above or to the right of a parallelogram is larger.
- Once a point "appears", it stays in the larger parallelograms.
- So, only need to keep track of minimal representatives $\mathcal{M}$, i.e.,

$$L(\mathbf{0}; \{(3,1),(2,3),(2,2)\}) = \mathcal{M} + (3,1)\mathbb{N} + (2,3)\mathbb{N}$$

where:

$$\mathcal{M} = \{(2,2),(4,4),(6,6),(8,8),(10,10),(12,12)\}$$

Intuition for dimension 2

Key observations:

- Each parallelogram has a quadratically many integer points
- There is a natural ordering on these parallelograms:
  - The parallelogram above or to the right of a parallelogram is larger.
- Once a point "appears", it stays in the larger parallelograms.
- So, only need to keep track of minimal representatives $\mathcal{M}$, i.e.,

$$L(\mathbf{0}; \{(3,1),(2,3),(2,2)\}) = \mathcal{M} + (3,1)\mathbb{N} + (2,3)\mathbb{N}$$

where:

$$\mathcal{M} = \{(2,2),(4,4),(6,6),(8,8),(10,10),(12,12)\}$$

- $|\mathcal{M}|$ and all numbers in $\mathcal{M}$ are "not big".

Case of dimension > 2

- Need to use linear sets with different periods (unlike $d = 2$)
- Replace finding two outermost vectors with Caratheodory's
- The rest are similar:
  - Dynamic programming,
  - Bounds from integer programming

Normal Form Theorem for Parikh Images of NFA

Statement of the Theorem

Let $\Sigma := \{a_1, \ldots, a_k\}$ for fixed $k \in \mathbb{Z}_{>0}$.

Theorem (Kopczynski & Lin, LICS 2010): Descriptional and computational complexity of Parikh Images of NFAs are polynomial.

- poly many union of linear sets with at most k periods with poly-bounded numbers
- Complexities are exponential in k .
- Generalizes Chrobak-Martinez Theorem (case $k = 1$).

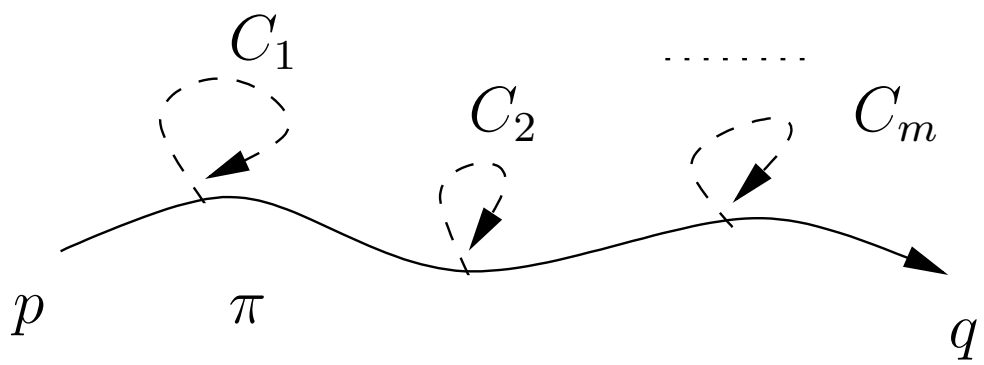
Proof idea: Path types

Notation: $Q = \{q_0, \ldots, q_{n-1}\}$ with initial state q_0 and final state q_{n-1}.

Proof idea: Path types

Notation: $Q = \{q_0, \ldots, q_{n-1}\}$ with initial state q_0 and final state q_{n-1}.

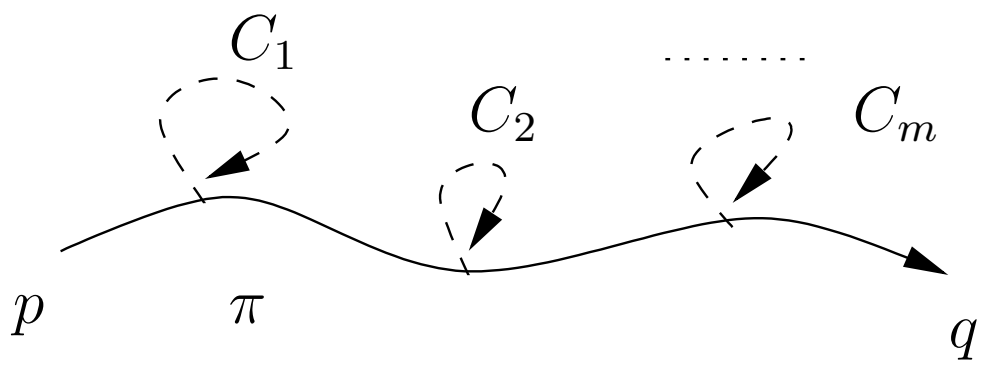
Given a path π and simple cycles $C_1, \ldots, C_m$ meeting with π:



Proof idea: Path types

Notation: $Q = \{q_0, \ldots, q_{n-1}\}$ with initial state q_0 and final state q_{n-1}.

Given a path π and simple cycles $C_1, \ldots, C_m$ meeting with π:

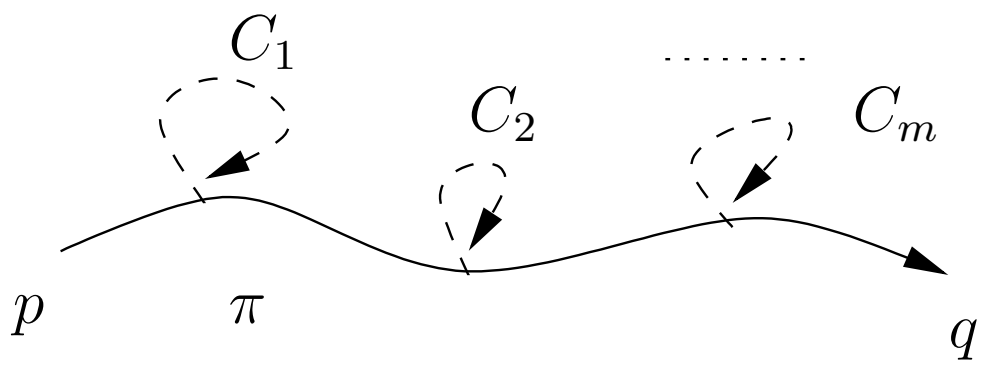


The path type T_π of π is the linear set $L(\mathcal{P}(\pi); \{\mathcal{P}(C_i)\}_{i=1}^m)$.

Proof idea: Path types

Notation: $Q = \{q_0, \ldots, q_{n-1}\}$ with initial state q_0 and final state q_{n-1}.

Given a path π and simple cycles $C_1, \ldots, C_m$ meeting with π:



The path type T_π of π is the linear set $L(\mathcal{P}(\pi); \{\mathcal{P}(C_i)\}_{i=1}^m)$.

There are at most $n^{k^{O(1)}}$ many periods.

Proof idea

Characterization of Parikh images of $L(\mathcal{A})$

Lemma: $\mathcal{P}(L(\mathcal{A})) = \bigcup_\pi T_\pi$, where π ranges over paths from initial to final state of length at most $O(n^2)$.

Proof idea

Characterization of Parikh images of $L(\mathcal{A})$

Lemma: $\mathcal{P}(L(\mathcal{A})) = \bigcup_\pi T_\pi$, where π ranges over paths from initial to final state of length at most $O(n^2)$.

Problem: There are $2^{(n+k)^{O(1)}}$ such π.

Proof idea

Characterization of Parikh images of $L(\mathcal{A})$

Lemma: $\mathcal{P}(L(\mathcal{A})) = \bigcup_\pi T_\pi$, where π ranges over paths from initial to final state of length at most $O(n^2)$.

Problem: There are $2^{(n+k)^{O(1)}}$ such π.

BUT: can apply Caratheodory-like theorem on each T_π:

Proof idea

Characterization of Parikh images of $L(\mathcal{A})$

Lemma: $\mathcal{P}(L(\mathcal{A})) = \bigcup_\pi T_\pi$, where π ranges over paths from initial to final state of length at most $O(n^2)$.

Problem: There are $2^{(n+k)^{O(1)}}$ such π.

BUT: can apply Caratheodory-like theorem on each T_π:

Corollary: $\mathcal{P}(L(\mathcal{A}))$ coincides with a union of polynomially many linear sets with polynomially large offsets and at most k polynomially large periods.

Computational complexity

Theorem: $\mathcal{P}(L(\mathcal{A}))$ coincides with a union of polynomially many linear sets with polynomially large offsets and at most k polynomially large periods.

A naive implementation takes exponential time.

Can improve to PTIME using dynamic programming!

Parikh images of extensions of NFA

Linear grammars

$$S \rightarrow A \tag{5}$$
$$A \rightarrow aBb \tag{6}$$
$$B \rightarrow aAb \tag{7}$$
$$B \rightarrow \varepsilon \tag{8}$$

Linear grammars

$$S \rightarrow A \tag{5}$$
$$A \rightarrow aBb \tag{6}$$
$$B \rightarrow aAb \tag{7}$$
$$B \rightarrow \varepsilon \tag{8}$$

The r.h.s. of a rule has at most 1 variable.

Linear grammars

$$S \rightarrow A \tag{5}$$
$$A \rightarrow aBb \tag{6}$$
$$B \rightarrow aAb \tag{7}$$
$$B \rightarrow \varepsilon \tag{8}$$

The r.h.s. of a rule has at most 1 variable.

Proposition: Generalised Chrobak-Martinez Theorem extends to Linear Grammars.

Linear grammars

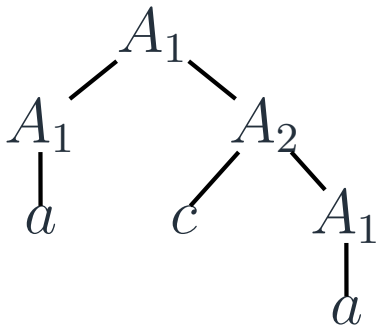
$$S \rightarrow A \tag{5}$$
$$A \rightarrow aBb \tag{6}$$
$$B \rightarrow aAb \tag{7}$$
$$B \rightarrow \varepsilon \tag{8}$$

The r.h.s. of a rule has at most 1 variable.

Proposition: Generalised Chrobak-Martinez Theorem extends to Linear Grammars.

Proof: Linear grammars are essentially NFA ...

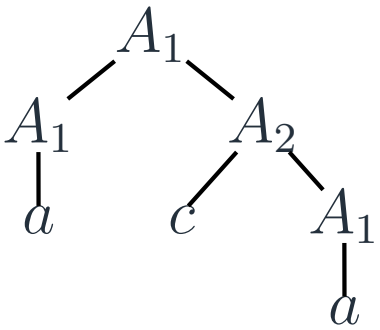
CFG of fixed “dimensions”



Theorem (Esparza-Ganty-Kiefer-Luttenberger’11): Generalised Chrobak-Martinez Theorem extends to CFG of fixed dimensions.

Dimensions measure how many times “doubling tricks” possibly get used in a CFG.

CFG of fixed “dimensions”

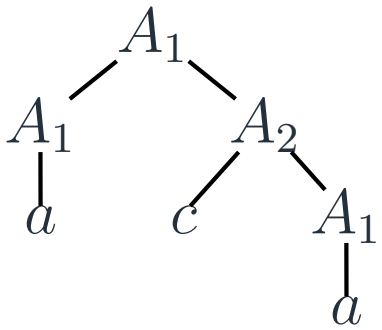


Theorem (Esparza-Ganty-Kiefer-Luttenberger’11): Generalised Chrobak-Martinez Theorem extends to CFG of fixed dimensions.

Dimensions measure how many times “doubling tricks” possibly get used in a CFG.

Linear grammars have dimension 0.

CFG of fixed “dimensions”



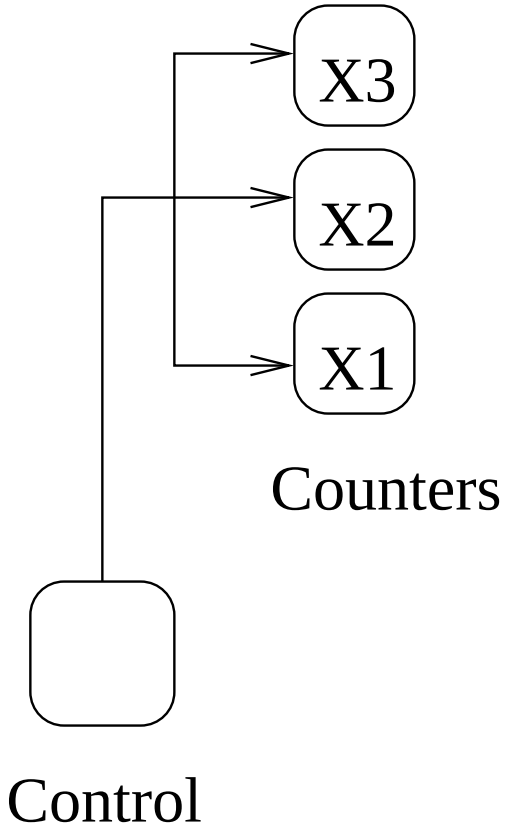
Theorem (Esparza-Ganty-Kiefer-Luttenberger’11): Generalised Chrobak-Martinez Theorem extends to CFG of fixed dimensions.

Dimensions measure how many times “doubling tricks” possibly get used in a CFG. Linear grammars have dimension 0.

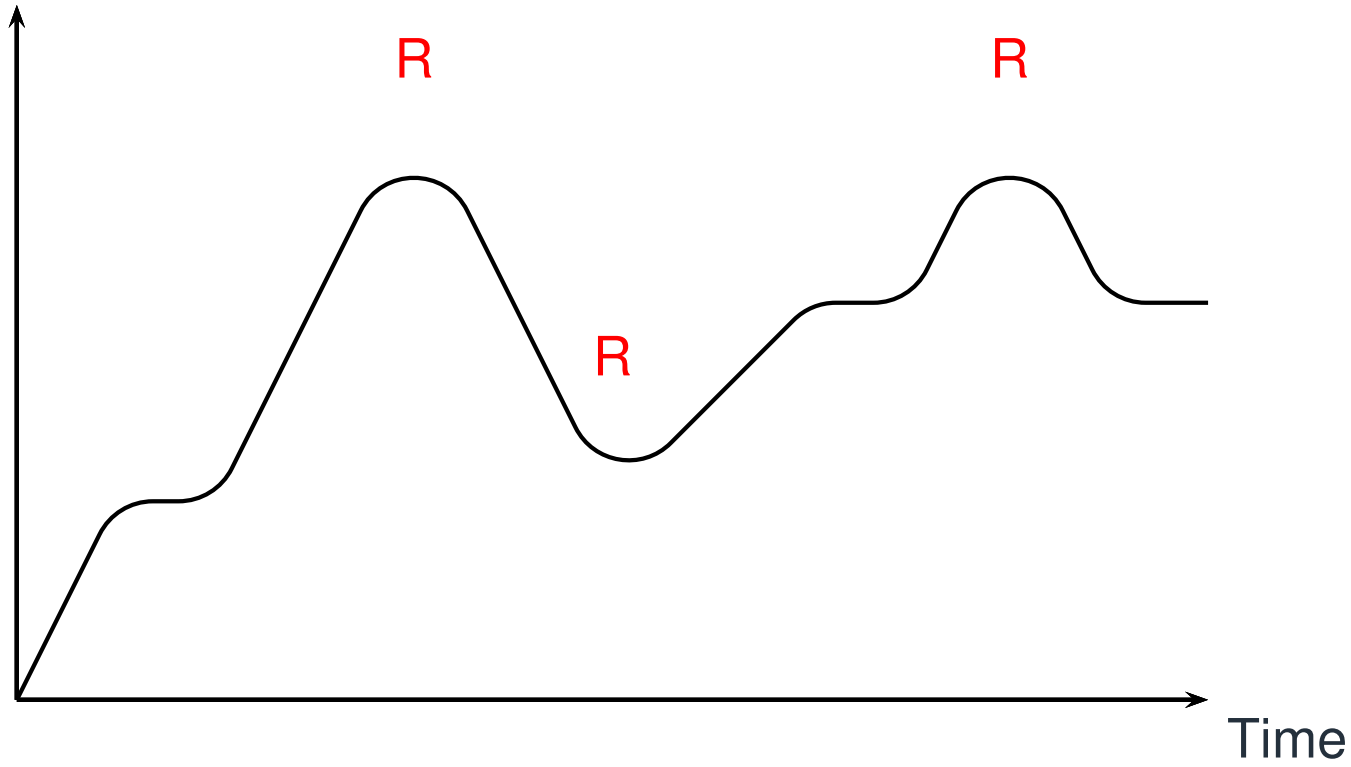
Proof: Compute an equivalent NFA of polynomial size and use Generalised Chrobak-Martinez Theorem.

Reversal-bounded counter automata

Minsky's counter automata

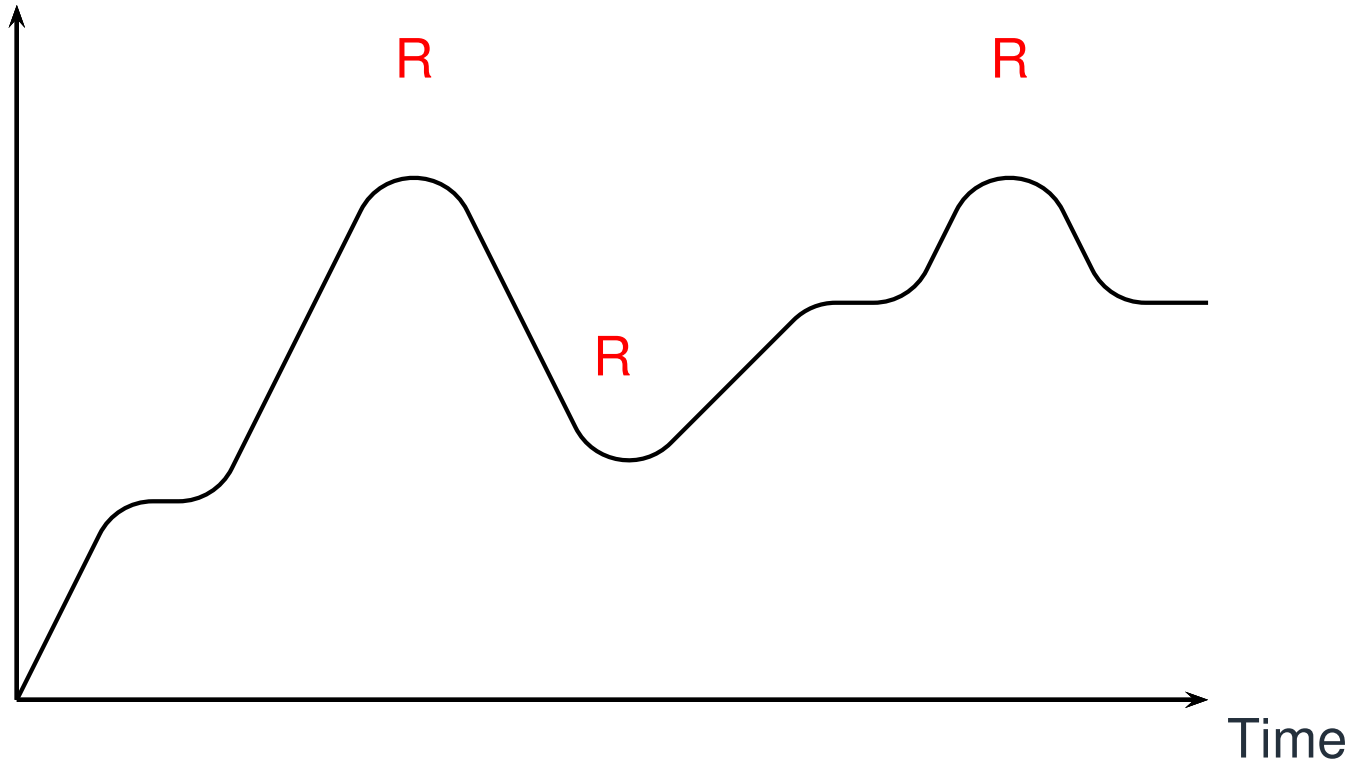


Reversal-bounded counter automata



This variable has 3 reversals

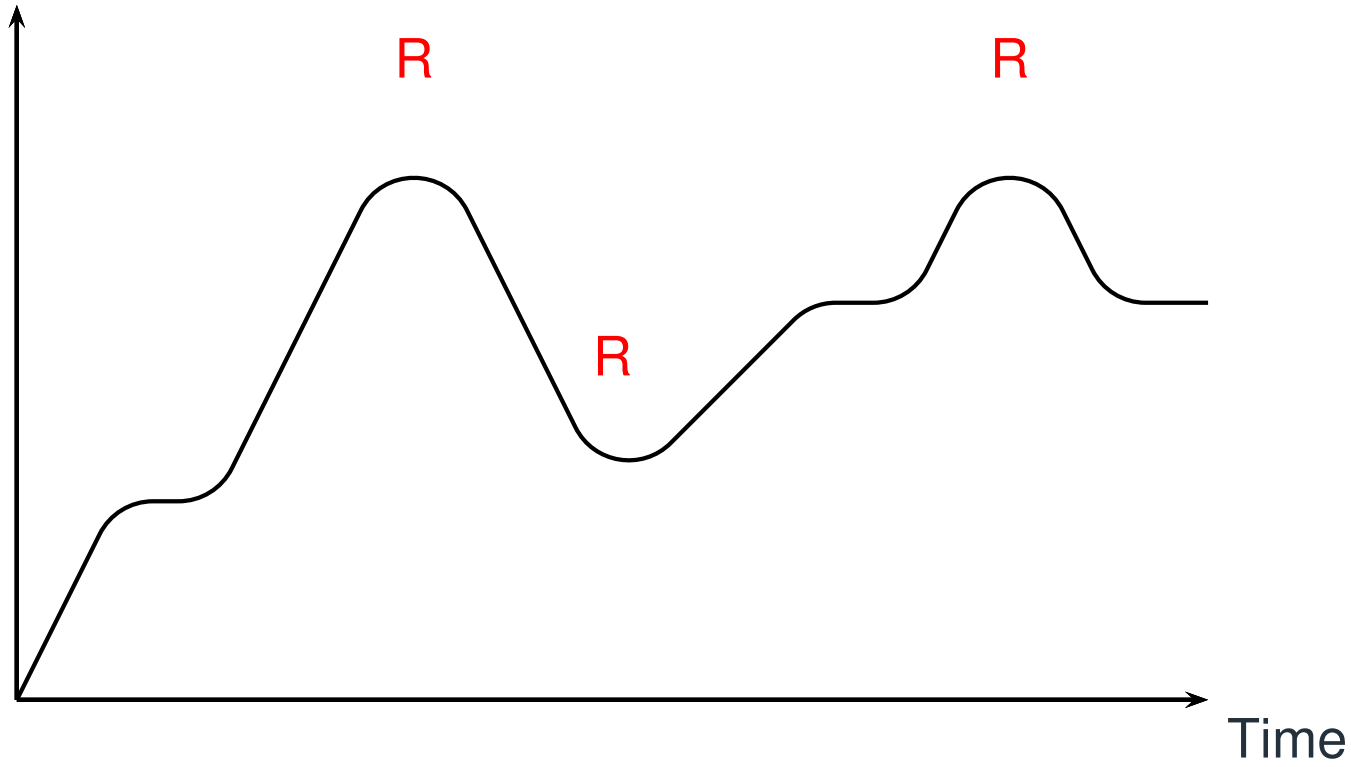
Reversal-bounded counter automata



This variable has 3 reversals

Restricted problem: examine paths with $r \in \mathbb{N}$ reversals for all variables

Reversal-bounded counter automata



This variable has 3 reversals

Restricted problem: examine paths with $r \in \mathbb{N}$ reversals for all variables

No finite bounds on length of 0-reversal-bounded paths!

Reversal-bounded counter automata

Theorem: Generalised Chrobak-Martinez Theorem extends to reversal-bounded counter automata with a fixed number of reversals and a fixed number of counters.

Reversal-bounded counter automata

Theorem: Generalised Chrobak-Martinez Theorem extends to reversal-bounded counter automata with a fixed number of reversals and a fixed number of counters.

Proof idea: follow Ibarra's proof, but use Generalised Chrobak-Martinez for Parikh images of NFA.

Reversal-bounded counter automata

Theorem: Generalised Chrobak-Martinez Theorem extends to reversal-bounded counter automata with a fixed number of reversals and a fixed number of counters.

Proof idea: follow Ibarra's proof, but use Generalised Chrobak-Martinez for Parikh images of NFA.

Note: This result can be used to derive optimal complexity for reversal-bounded verification counter automata.

Reversal-bounded counter automata

Theorem: Generalised Chrobak-Martinez Theorem extends to reversal-bounded counter automata with a fixed number of reversals and a fixed number of counters.

Proof idea: follow Ibarra's proof, but use Generalised Chrobak-Martinez for Parikh images of NFA.

Note: This result can be used to derive optimal complexity for reversal-bounded verification counter automata.

Theorem (Hague-Lin'11): This does not work if # reversals/# counters are non-fixed, but you can compute an existential Presburger formula in polynomial time (even if a pushdown stack is added).

Alternative notion of Complexity

Study the complexity of decision problems (e.g. membership, universality, ...) over different models.

Alternative notion of Complexity

Study the complexity of decision problems (e.g. membership, universality, ...) over different models.

There is also a stark difference, e.g., for emptiness:

- NFA: P (f.a.), NP (u.a.) [Kopczynski-Lin'10]
- CFG: NP (f.a.), NP (u.a.) [Hyunh'83]

Alternative notion of Complexity

Study the complexity of decision problems (e.g. membership, universality, ...) over different models.

There is also a stark difference, e.g., for emptiness:

- NFA: P (f.a.), NP (u.a.) [Kopczynski-Lin'10]
- CFG: NP (f.a.), NP (u.a.) [Hyunh'83]

Can be proven by first constructing Parikh images (as semilinear set representation or existential Presburger formulas).

Conclusion

Challenges

- Does generalised Chrobak-Martinez Theorem extend to one-counter automata?

Challenges

- Does generalised Chrobak-Martinez Theorem extend to one-counter automata?
- How do we compare the descriptional complexity of alternating finite-state automata vs. CFG?

Challenges

- Does generalised Chrobak-Martinez Theorem extend to one-counter automata?
- How do we compare the descriptional complexity of alternating finite-state automata vs. CFG?
- Study succinctness hierarchy in Parikh's Theorem.

Challenges

- Does generalised Chrobak-Martinez Theorem extend to one-counter automata?
- How do we compare the descriptional complexity of alternating finite-state automata vs. CFG?
- Study succinctness hierarchy in Parikh's Theorem.

THANKS!!