Slice closures of indexed languages and word equations with counting constraints

Laura Ciobanu[1] and Georg Zetzsche[2]

[1]Hariot-Watt University, Edinburgh

[2]Max Planck Institute for Software Systems (MPI-SWS)

MOSCA 2025

Word equations with counting constraints

Word equation: $YabX = XbaZ$.

Word equations with counting constraints

Word equation: $YabX = XbaZ$. Formally: (U, V) with $U, V \in (A \cup \mathcal{X})^*$.

Word equations with counting constraints

Word equation: $YabX = XbaZ$. Formally: (U, V) with $U, V \in (A \cup \mathcal{X})^*$.

Solution: homomorphism $\sigma\colon \mathcal{X}^* \to A^*$ with $\sigma(U) = \sigma(V)$.

Word equations with counting constraints

Word equation: $YabX = XbaZ$. Formally: (U, V) with $U, V \in (A \cup \mathcal{X})^*$.
Solution: homomorphism $\sigma \colon \mathcal{X}^* \to A^*$ with $\sigma(U) = \sigma(V)$.

Theorem (Makanin 1977)

Solvability of word equations is decidable.

Word equations with counting constraints

Word equation: $YabX = XbaZ$. Formally: (U, V) with $U, V \in (A \cup \mathcal{X})^*$.

Solution: homomorphism $\sigma \colon \mathcal{X}^* \to A^*$ with $\sigma(U) = \sigma(V)$.

Theorem (Makanin 1977)

Solvability of word equations is decidable.

Büchi & Senger 1988

Word equations with counting constraints

Word equation: $YabX = XbaZ$. Formally: (U, V) with $U, V \in (A \cup \mathcal{X})^*$.
Solution: homomorphism $\sigma\colon \mathcal{X}^* \to A^*$ with $\sigma(U) = \sigma(V)$.

Theorem (Makanin 1977)

Solvability of word equations is decidable.

Büchi & Senger 1988

- Adding constraints "$|X|_a = |Y|_b$" → undecidable.

Word equations with counting constraints

Word equation: $YabX = XbaZ$. Formally: (U, V) with $U, V \in (A \cup \mathcal{X})^*$.
Solution: homomorphism $\sigma\colon \mathcal{X}^* \to A^*$ with $\sigma(U) = \sigma(V)$.

Theorem (Makanin 1977)

Solvability of word equations is decidable.

Büchi & Senger 1988

- Adding constraints "$|X|_a = |Y|_b$" → undecidable.
- Open problem: What about length constraints "$|X| = |Y|$"?

Word equations with counting constraints

Word equation: $YabX = XbaZ$. Formally: (U, V) with $U, V \in (A \cup \mathcal{X})^*$.
Solution: homomorphism $\sigma \colon \mathcal{X}^* \to A^*$ with $\sigma(U) = \sigma(V)$.

Theorem (Makanin 1977)

Solvability of word equations is decidable.

Büchi & Senger 1988

- Adding constraints "$|X|_a = |Y|_b$" → undecidable.
- Open problem: What about length constraints "$|X| = |Y|$"?

For $\mathcal{X} = \{X_1, \ldots, X_k\}$, we encode a solution $\sigma \colon \mathcal{X} \to A^*$

Word equations with counting constraints

Word equation: $YabX = XbaZ$. Formally: (U, V) with $U, V \in (A \cup \mathcal{X})^*$.
Solution: homomorphism $\sigma\colon \mathcal{X}^* \to A^*$ with $\sigma(U) = \sigma(V)$.

Theorem (Makanin 1977)

Solvability of word equations is decidable.

Büchi & Senger 1988

- Adding constraints "$|X|_a = |Y|_b$" → undecidable.
- Open problem: What about length constraints "$|X| = |Y|$"?

For $\mathcal{X} = \{X_1, \ldots, X_k\}$, we encode a solution $\sigma\colon \mathcal{X} \to A^*$ as

$$\mathrm{enc}(\sigma) = \sigma(X_1)\# \cdots \#\sigma(X_k)$$

Word equations with counting constraints

Word equation: $YabX = XbaZ$. Formally: (U, V) with $U, V \in (A \cup \mathcal{X})^*$.
Solution: homomorphism $\sigma\colon \mathcal{X}^* \to A^*$ with $\sigma(U) = \sigma(V)$.

Theorem (Makanin 1977)

Solvability of word equations is decidable.

Büchi & Senger 1988

- Adding constraints "$|X|_a = |Y|_b$" → undecidable.
- Open problem: What about length constraints "$|X| = |Y|$"?

For $\mathcal{X} = \{X_1, \ldots, X_k\}$, we encode a solution $\sigma\colon \mathcal{X} \to A^*$ as

$$\mathrm{enc}(\sigma) = \sigma(X_1)\# \cdots \#\sigma(X_k)$$

Theorem (Ciobanu, Diekert, Elder, ICALP 2015)

Given word equation with rational constraints, the language $\{\mathrm{enc}(\sigma) \mid \sigma$ is a solution$\}$ is an EDT0L language, in particular indexed.

Word equations with counting constraints

Word equation: $YabX = XbaZ$. Formally: (U, V) with $U, V \in (A \cup \mathcal{X})^*$.
Solution: homomorphism $\sigma\colon \mathcal{X}^* \to A^*$ with $\sigma(U) = \sigma(V)$.

Theorem (Makanin 1977)

Solvability of word equations is decidable.

Büchi & Senger 1988

- Adding constraints "$|X|_a = |Y|_b$" → undecidable.
- Open problem: What about length constraints "$|X| = |Y|$"?

For $\mathcal{X} = \{X_1, \ldots, X_k\}$, we encode a solution $\sigma\colon \mathcal{X} \to A^*$ as

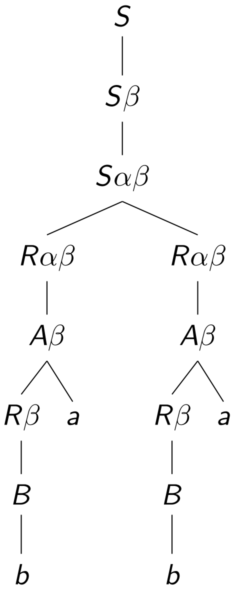
$$\mathrm{enc}(\sigma) = \sigma(X_1)\#\cdots\#\sigma(X_k)$$

Theorem (Ciobanu, Diekert, Elder, ICALP 2015)

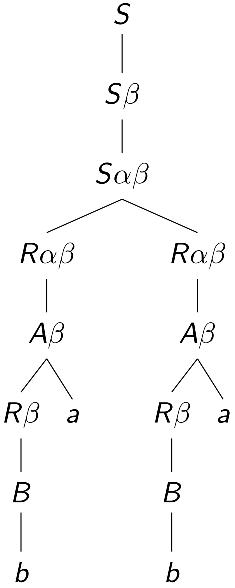
Given word equation with rational constraints, the language $\{\mathrm{enc}(\sigma) \mid \sigma$ *is a solution*$\}$ *is an EDT0L language, in particular indexed.*

Useful for decidability of any counting constraints?

Indexed grammars

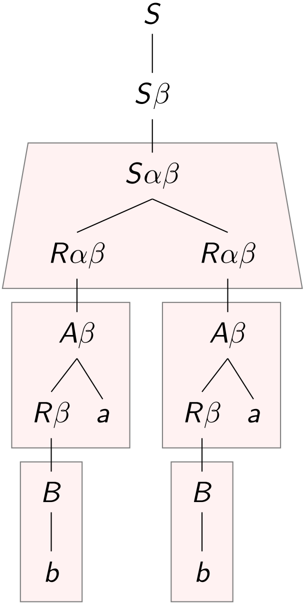


Indexed grammars



Context-free $X \to YZ$, $X \to x$ etc.

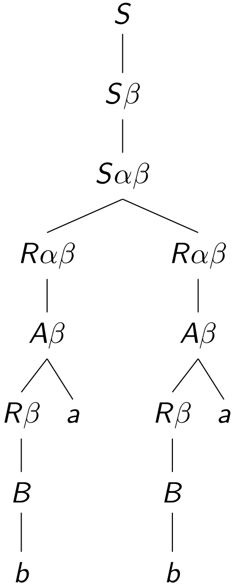
Indexed grammars



Context-free $X \rightarrow YZ$, $X \rightarrow x$ etc.

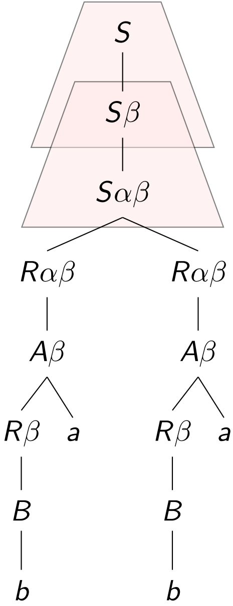
stack gets copied

Indexed grammars



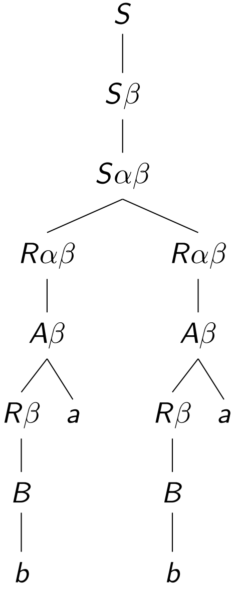
Context-free	$X \to YZ$, $X \to x$ etc. stack gets copied
Push productions	$X \to Y\gamma$

Indexed grammars



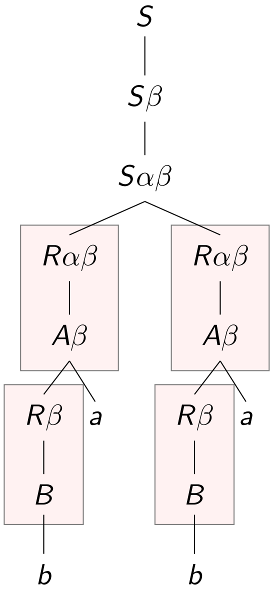
Context-free	$X \rightarrow YZ$, $X \rightarrow x$ etc. stack gets copied
Push productions	$X \rightarrow Y\gamma$ push letter on stack

Indexed grammars



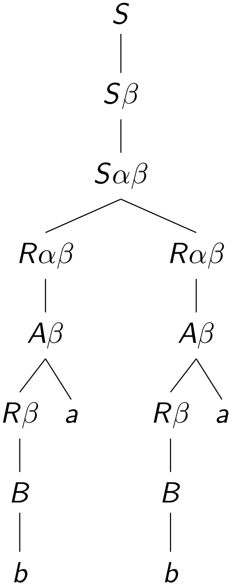
Context-free	$X \to YZ$, $X \to x$ etc.
	stack gets copied
Push productions	$X \to Y\gamma$
	push letter on stack
Pop productions	$X\gamma \to Y$

Indexed grammars



Context-free	$X \to YZ$, $X \to x$ etc. stack gets copied
Push productions	$X \to Y\gamma$ push letter on stack
Pop productions	$X\gamma \to Y$ pop letter from stack

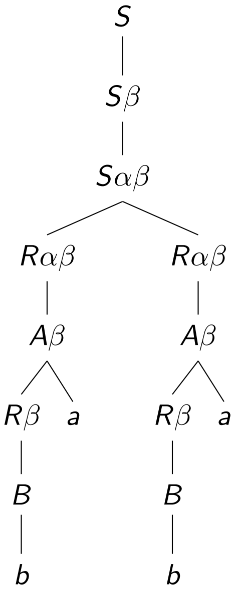
Indexed grammars



Context-free	$X \to YZ$, $X \to x$ etc. stack gets copied
Push productions	$X \to Y\gamma$ push letter on stack
Pop productions	$X\gamma \to Y$ pop letter from stack

- introduced: Aho (1968)

Indexed grammars



Context-free	$X \to YZ$, $X \to x$ etc. stack gets copied
Push productions	$X \to Y\gamma$ push letter on stack
Pop productions	$X\gamma \to Y$ pop letter from stack

- introduced: Aho (1968)
- equivalent to Higher-Order Recursion Schemes (HORS) of order 2

Counting problems for indexed languages

Raised by Kobayashi (J. ACM 2013)

Challenge: Find decidable counting properties

Counting problems for indexed languages

Raised by Kobayashi (J. ACM 2013)

Challenge: Find decidable counting properties

Parikh image

$\Psi\colon \{a_1, \ldots, a_n\}^* \to \mathbb{N}^n, \quad \Psi(w) = (|w|_{a_1}, \ldots, |w|_{a_n})$

Counting problems for indexed languages

Raised by Kobayashi (J. ACM 2013)

Challenge: Find decidable counting properties

Parikh image

$\Psi\colon \{a_1, \ldots, a_n\}^* \to \mathbb{N}^n, \quad \Psi(w) = (|w|_{a_1}, \ldots, |w|_{a_n})$

For context-free L: $\Psi(L)$ is semilinear

Counting problems for indexed languages

Raised by Kobayashi (J. ACM 2013)

Challenge: Find decidable counting properties

Parikh image

$\Psi\colon \{a_1, \ldots, a_n\}^* \to \mathbb{N}^n, \quad \Psi(w) = (|w|_{a_1}, \ldots, |w|_{a_n})$

For context-free L: $\Psi(L)$ is semilinear

Sets definable in first-order logic
in $(\mathbb{N}, +, \leqslant, 0, 1)$
Presburger arithmetic: decidable logic!

Counting problems for indexed languages

Raised by Kobayashi (J. ACM 2013)

Challenge: Find decidable counting properties

Parikh image

$\Psi\colon \{a_1, \dots, a_n\}^* \to \mathbb{N}^n, \quad \Psi(w) = (|w|_{a_1}, \dots, |w|_{a_n})$

For context-free L: $\Psi(L)$ is semilinear → many decidability properties

Sets definable in first-order logic
in $(\mathbb{N}, +, \leqslant, 0, 1)$
Presburger arithmetic: decidable logic!

Counting problems for indexed languages

Raised by Kobayashi (J. ACM 2013)

Challenge: Find decidable counting properties

Parikh image

$\Psi\colon \{a_1, \ldots, a_n\}^* \to \mathbb{N}^n, \quad \Psi(w) = (|w|_{a_1}, \ldots, |w|_{a_n})$

For context-free L: $\Psi(L)$ is semilinear → many decidability properties

For indexed languages (Uezato & Minamide, Kobayashi, Z.)

Counting problems for indexed languages

Raised by Kobayashi (J. ACM 2013)

Challenge: Find decidable counting properties

Parikh image

$\Psi\colon \{a_1, \dots, a_n\}^* \to \mathbb{N}^n, \quad \Psi(w) = (|w|_{a_1}, \dots, |w|_{a_n})$

For context-free L: $\Psi(L)$ is semilinear → many decidability properties

For indexed languages (Uezato & Minamide, Kobayashi, Z.)

- $L \cap \{a^n b^n \mid n \geqslant 0\} \overset{?}{=} \varnothing$ undecidable

Counting problems for indexed languages

Raised by Kobayashi (J. ACM 2013)

Challenge: Find decidable counting properties

Parikh image

$\Psi\colon \{a_1, \dots, a_n\}^* \to \mathbb{N}^n, \quad \Psi(w) = (|w|_{a_1}, \dots, |w|_{a_n})$

For context-free L: $\Psi(L)$ is semilinear → many decidability properties

For indexed languages (Uezato & Minamide, Kobayashi, Z.)

- $L \cap \{a^n b^n \mid n \geqslant 0\} \stackrel{?}{=} \emptyset$, $L \cap \{a^m b^n \mid m \geqslant n\} \stackrel{?}{=} \emptyset$ both undecidable

Counting problems for indexed languages

Raised by Kobayashi (J. ACM 2013)

Challenge: Find decidable counting properties

Parikh image

$\Psi\colon \{a_1,\ldots,a_n\}^* \to \mathbb{N}^n, \quad \Psi(w) = (|w|_{a_1},\ldots,|w|_{a_n})$
For context-free L: $\Psi(L)$ is semilinear → many decidability properties

For indexed languages (Uezato & Minamide, Kobayashi, Z.)

- $L \cap \{a^n b^n \mid n \geqslant 0\} \overset{?}{=} \varnothing$, $L \cap \{a^m b^n \mid m \geqslant n\} \overset{?}{=} \varnothing$ both undecidable
- $L \overset{?}{\subseteq}$ well-nested words over [and] undecidable

Counting problems for indexed languages

Raised by Kobayashi (J. ACM 2013)

Challenge: Find decidable counting properties

Parikh image

$\Psi\colon \{a_1, \ldots, a_n\}^* \to \mathbb{N}^n, \quad \Psi(w) = (|w|_{a_1}, \ldots, |w|_{a_n})$

For context-free L: $\Psi(L)$ is semilinear → many decidability properties

For indexed languages (Uezato & Minamide, Kobayashi, Z.)

- $L \cap \{a^n b^n \mid n \geqslant 0\} \stackrel{?}{=} \emptyset$, $L \cap \{a^m b^n \mid m \geqslant n\} \stackrel{?}{=} \emptyset$ both undecidable
- $L \overset{?}{\subseteq}$ well-nested words over [and] undecidable
- $L \overset{?}{\subseteq}$ same number of a's and b's decidability open (Kobayashi 2019)

Semilinear sets

A set $S \subseteq \mathbb{N}^d$ is *linear* if it is of the form

$$S = \{u \ + \ \lambda_1 u_1 \ + \ \cdots \ + \ \lambda_n u_n \mid \lambda_1, \ldots, \lambda_n \in \mathbb{N}\}$$

with $u, u_1, \ldots, u_n \in \mathbb{N}^d$.

Semilinear sets

A set $S \subseteq \mathbb{N}^d$ is *linear* if it is of the form

$$S = \{u \; + \; \lambda_1 u_1 \; + \; \cdots \; + \; \lambda_n u_n \mid \lambda_1, \ldots, \lambda_n \in \mathbb{N}\}$$

with $u, u_1, \ldots, u_n \in \mathbb{N}^d$. A *semilinear set* is a finite union of linear sets.

Semilinear sets

A set $S \subseteq \mathbb{N}^d$ is *linear* if it is of the form

$$S = \{u \ + \ \lambda_1 u_1 \ + \ \cdots \ + \ \lambda_n u_n \mid \lambda_1, \ldots, \lambda_n \in \mathbb{N}\}$$

with $u, u_1, \ldots, u_n \in \mathbb{N}^d$. A *semilinear set* is a finite union of linear sets.

Presburger arithmetic

First-order logic over the structure $\langle \mathbb{N}; +, \leqslant, 0, 1 \rangle$. For example:

$$\exists y \colon x = y + y \quad \exists y \colon \exists z \colon \varphi(y) \wedge \psi(z) \wedge x \geqslant y + z$$

A formula φ with d free variables *defines* the set $\{x \in \mathbb{N}^d \mid \varphi(x)\}$.

Semilinear sets

A set $S \subseteq \mathbb{N}^d$ is *linear* if it is of the form

$$S = \{u \ + \ \lambda_1 u_1 \ + \ \cdots \ + \ \lambda_n u_n \mid \lambda_1, \ldots, \lambda_n \in \mathbb{N}\}$$

with $u, u_1, \ldots, u_n \in \mathbb{N}^d$. A *semilinear set* is a finite union of linear sets.

Presburger arithmetic

First-order logic over the structure $\langle \mathbb{N}; +, \leqslant, 0, 1 \rangle$. For example:

$$\exists y \colon x = y + y \quad \exists y \colon \exists z \colon \varphi(y) \wedge \psi(z) \wedge x \geqslant y + z$$

A formula φ with d free variables *defines* the set $\{x \in \mathbb{N}^d \mid \varphi(x)\}$.

Theorem (Presburger 1929)

Presburger arithmetic is decidable.

Semilinear sets

A set $S \subseteq \mathbb{N}^d$ is *linear* if it is of the form

$$S = \{u \ + \ \lambda_1 u_1 \ + \ \cdots \ + \ \lambda_n u_n \mid \lambda_1, \ldots, \lambda_n \in \mathbb{N}\}$$

with $u, u_1, \ldots, u_n \in \mathbb{N}^d$. A *semilinear set* is a finite union of linear sets.

Presburger arithmetic

First-order logic over the structure $\langle \mathbb{N}; +, \leqslant, 0, 1 \rangle$. For example:

$$\exists y \colon x = y + y \quad \exists y \colon \exists z \colon \varphi(y) \wedge \psi(z) \wedge x \geqslant y + z$$

A formula φ with d free variables *defines* the set $\{x \in \mathbb{N}^d \mid \varphi(x)\}$.

Theorem (Presburger 1929)

Presburger arithmetic is decidable.

Theorem (Ginsburg & Spanier 1966)

The definable sets are exactly the semilinear sets.

Semilinear sets

A set $S \subseteq \mathbb{N}^d$ is *linear* if it is of the form

$$S = \{u \ + \ \lambda_1 u_1 \ + \ \cdots \ + \ \lambda_n u_n \mid \lambda_1, \ldots, \lambda_n \in \mathbb{N}\}$$

with $u, u_1, \ldots, u_n \in \mathbb{N}^d$. A *semilinear set* is a finite union of linear sets.

Presburger arithmetic

First-order logic over the structure $\langle \mathbb{N}; +, \leqslant, 0, 1 \rangle$. For example:

$$\exists y \colon x = y + y \quad \exists y \colon \exists z \colon \varphi(y) \wedge \psi(z) \wedge x \geqslant y + z$$

A formula φ with d free variables *defines* the set $\{x \in \mathbb{N}^d \mid \varphi(x)\}$.

Theorem (Presburger 1929)

Presburger arithmetic is decidable.

Theorem (Ginsburg & Spanier 1966)

The definable sets are exactly the semilinear sets. This is effective.

Semilinear sets

A set $S \subseteq \mathbb{N}^d$ is *linear* if it is of the form

$$S = \{u + \lambda_1 u_1 + \cdots + \lambda_n u_n \mid \lambda_1, \ldots, \lambda_n \in \mathbb{N}\}$$

with $u, u_1, \ldots, u_n \in \mathbb{N}^d$. A *semilinear set* is a finite union of linear sets.

Presburger arithmetic

First-order logic over the structure $\langle \mathbb{N}; +, \leqslant, 0, 1 \rangle$. For example:

Unfortunately, for indexed languages L,
the Parikh image $\Psi(L)$ is not always semilinear.

A formula φ $)\}$.

Theorem (Presburger 1929)

Presburger arithmetic is decidable.

Theorem (Ginsburg & Spanier 1966)

The definable sets are exactly the semilinear sets. This is effective.

Semilinear sets

A set $S \subseteq \mathbb{N}^d$ is *linear* if it is of the form

$$S = \{u \ + \ \lambda_1 u_1 \ + \ \cdots \ + \ \lambda_n u_n \mid \lambda_1, \ldots, \lambda_n \in \mathbb{N}\}$$

with $u, u_1, \ldots, u_n \in \mathbb{N}^d$. A *semilinear set* is a finite union of linear sets.

Presburger arithmetic

First-order logic over the structure $\langle \mathbb{N}; +, \leqslant, 0, 1 \rangle$. For example:

A formula φ $)\}$.

Unfortunately, for indexed languages L,
the Parikh image $\Psi(L)$ is not always semilinear.
E.g.: $\{a^{n^2} \mid n \in \mathbb{N}\}$ is an indexed language.

Theorem (Presburger 1929)

Presburger arithmetic is decidable.

Theorem (Ginsburg & Spanier 1966)

The definable sets are exactly the semilinear sets. This is effective.

Slices

A subset $S \subseteq \mathbb{N}^d$ is a *slice* if $u, u + v, u + w \in S$ implies $u + v + w \in S$.

Slices

A subset $S \subseteq \mathbb{N}^d$ is a *slice* if $u, u+v, u+w \in S$ implies $u+v+w \in S$.

$$\begin{tikzcd}
S \ni & u+v & & u+w & \in S \\
& & u \arrow[ul, "v"'] \arrow[ur, "w"] & \in S &
\end{tikzcd}$$

Slices

A subset $S \subseteq \mathbb{N}^d$ is a *slice* if $u, u+v, u+w \in S$ implies $u+v+w \in S$.

$$\begin{tikzcd}
 & & u+v+w & \in S & \\
S \ni & u+v \arrow[ur, dashed, "w"] & & u+w \arrow[ul, dashed, "v"'] & \in S \\
 & & u \arrow[ul, "v"] \arrow[ur, "w"'] & \in S &
\end{tikzcd}$$

Slices

A subset $S \subseteq \mathbb{N}^d$ is a *slice* if $u, u+v, u+w \in S$ implies $u+v+w \in S$.

$$\begin{tikzcd}
 & & u+v+w & \in S & \\
S \ni & u+v \arrow[ur, dashed, "w"] & & u+w \arrow[ul, dashed, "v"'] & \in S \\
 & & u \arrow[ul, "v"] \arrow[ur, "w"'] & \in S &
\end{tikzcd}$$

Examples

Slices

A subset $S \subseteq \mathbb{N}^d$ is a *slice* if $u, u+v, u+w \in S$ implies $u+v+w \in S$.

$$\begin{tikzcd}
 & & u+v+w & \in S & \\
S \ni & u+v \arrow[ur, dashed, "w"] & & u+w \arrow[ul, dashed, "v"'] & \in S \\
 & & u \arrow[ul, "v"] \arrow[ur, "w"'] & \in S &
\end{tikzcd}$$

Examples

- Upward closed sets (w.r.t. component-wise ordering)

Slices

A subset $S \subseteq \mathbb{N}^d$ is a *slice* if $u, u + v, u + w \in S$ implies $u + v + w \in S$.

$$\begin{tikzcd}
 & & u + v + w & \in S & \\
S \ni & u + v \arrow[ur, dashed, "w"] & & u + w \arrow[ul, dashed, "v"'] & \in S \\
 & & u \arrow[ul, "v"] \arrow[ur, "w"'] & \in S &
\end{tikzcd}$$

Examples

- Upward closed sets (w.r.t. component-wise ordering)
- Solution sets to linear equation systems: $\{x \in \mathbb{N}^d \mid Ax = b\}$

Slices

A subset $S \subseteq \mathbb{N}^d$ is a *slice* if $u, u+v, u+w \in S$ implies $u+v+w \in S$.

$$\begin{tikzcd}
 & & u+v+w & \in S & \\
S \ni & u+v \arrow[ur, dashed, "w"] & & u+w \arrow[ul, dashed, "v"'] & \in S \\
 & & u \arrow[ul, "v"] \arrow[ur, "w"'] & \in S &
\end{tikzcd}$$

Examples

- Upward closed sets (w.r.t. component-wise ordering)
- Solution sets to linear equation systems: $\{x \in \mathbb{N}^d \mid Ax = b\}$
- Congruence relations in $\mathbb{N}^d$

Slices

A subset $S \subseteq \mathbb{N}^d$ is a *slice* if $u, u+v, u+w \in S$ implies $u+v+w \in S$.

$$\begin{tikzcd}
 & & u+v+w & \in S & \\
S \ni & u+v \arrow[ur, dashed, "w"] & & u+w \arrow[ul, dashed, "v"'] & \in S \\
 & & u \arrow[ul, "v"] \arrow[ur, "w"'] & \in S &
\end{tikzcd}$$

Examples

- Upward closed sets (w.r.t. component-wise ordering)
- Solution sets to linear equation systems: $\{x \in \mathbb{N}^d \mid Ax = b\}$
- Congruence relations in $\mathbb{N}^d$

Theorem (Eilenberg & Schützenberger 1969)

Every slice is semilinear.

Slices

A subset $S \subseteq \mathbb{N}^d$ is a *slice* if $u, u+v, u+w \in S$ implies $u+v+w \in S$.

$$\begin{tikzcd}
 & & u+v+w & \in S & \\
S \ni & u+v \arrow[ur, dashed, "w"] & & u+w \arrow[ul, dashed, "v"'] & \in S \\
 & & u \arrow[ul, "v"] \arrow[ur, "w"'] & \in S &
\end{tikzcd}$$

Examples

- Upward closed sets (w.r.t. component-wise ordering)
- Solution sets to linear equation systems: $\{x \in \mathbb{N}^d \mid Ax = b\}$
- Congruence relations in $\mathbb{N}^d$

Theorem (Eilenberg & Schützenberger 1969)

Every slice is semilinear.

Note: This is not constructive!

Slice closure

Slice closure

For $M \subseteq \mathbb{N}^d$, denote by $\mathsf{Slice}(M)$ the *smallest* slice containing M.

Slice closure

Slice closure

For $M \subseteq \mathbb{N}^d$, denote by $\mathsf{Slice}(M)$ the *smallest* slice containing M.

Theorem (Ciobanu & Z., LICS 2024)

For indexed L, its slice closure $\mathsf{Slice}(\Psi(L))$ is effectively semilinear.

Slice closure

Slice closure

For $M \subseteq \mathbb{N}^d$, denote by $\mathrm{Slice}(M)$ the *smallest* slice containing M.

Theorem (Ciobanu & Z., LICS 2024)

For indexed L, its slice closure $\mathrm{Slice}(\Psi(L))$ is effectively semilinear.

$L \subseteq$ same number of a's and b's $\iff \Psi(L) \qquad\qquad \subseteq \{(n,n) \mid n \geqslant 0\}$

Slice closure

Slice closure

For $M \subseteq \mathbb{N}^d$, denote by $\mathrm{Slice}(M)$ the *smallest* slice containing M.

Theorem (Ciobanu & Z., LICS 2024)

For indexed L, its slice closure $\mathrm{Slice}(\Psi(L))$ is effectively semilinear.

$$\begin{aligned} L \subseteq \text{same number of } a\text{'s and } b\text{'s} \iff &\ \Psi(L) && \subseteq \{(n,n) \mid n \geqslant 0\} \\ \iff &\ \mathrm{Slice}(\Psi(L)) && \subseteq \{(n,n) \mid n \geqslant 0\}. \end{aligned}$$

Slice closure

Slice closure

For $M \subseteq \mathbb{N}^d$, denote by $\mathrm{Slice}(M)$ the *smallest* slice containing M.

Theorem (Ciobanu & Z., LICS 2024)

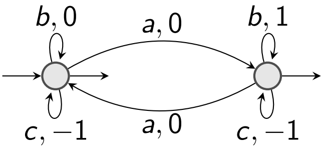
For indexed L, its slice closure $\mathrm{Slice}(\Psi(L))$ is effectively semilinear.

$$\begin{aligned} L \subseteq \text{same number of } a\text{'s and } b\text{'s} &\iff \Psi(L) \subseteq \{(n,n) \mid n \geqslant 0\} \\ &\iff \mathrm{Slice}(\Psi(L)) \subseteq \{(n,n) \mid n \geqslant 0\}. \end{aligned}$$

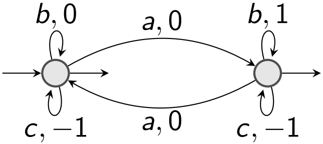
Slice closures encode more

Affine hull of $\Psi(L)$, upward closure $\Psi(u)\uparrow = \{u \in \mathbb{N}^d \mid \exists v \in \Psi(L) \colon u \geqslant v\}$.

Word equations and counting functions

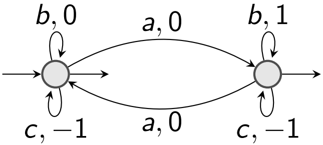


Word equations and counting functions



$f(w) =$ number of b's after a's, minus c's

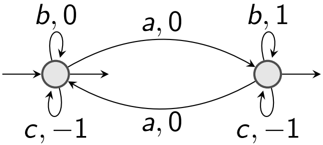
Word equations and counting functions



$f(w)$ = number of b's after a's, minus c's

Automaton with labels in $A^* \times \mathbb{Z}^n$ $\rightsquigarrow$ counting function $f\colon A^* \to \mathbb{Z}^n$

Word equations and counting functions

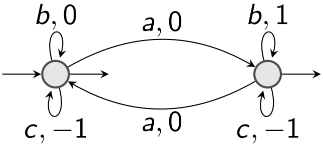


$f(w)$ = number of b's after a's, minus c's

Automaton with labels in $A^* \times \mathbb{Z}^n$ ⇝ counting function $f : A^* \to \mathbb{Z}^n$

Word equations with counting constraints

Word equations and counting functions



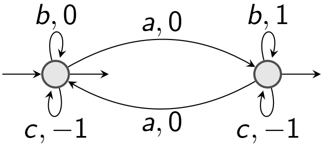
$f(w)$ = number of b's after a's, minus c's

Automaton with labels in $A^* \times \mathbb{Z}^n$ $\rightsquigarrow$ counting function $f \colon A^* \to \mathbb{Z}^n$

Word equations with counting constraints

Given Word equation (U, V) over $(A \cup \mathcal{X})^*$, counting function $f \colon (A \cup \{\#\})^* \to \mathbb{Z}^n$

Word equations and counting functions



$f(w) =$ number of b's after a's, minus c's

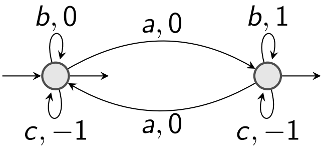
Automaton with labels in $A^* \times \mathbb{Z}^n$ $\rightsquigarrow$ counting function $f\colon A^* \to \mathbb{Z}^n$

Word equations with counting constraints

Given Word equation (U, V) over $(A \cup \mathcal{X})^*$, counting function $f\colon (A \cup \{\#\})^* \to \mathbb{Z}^n$

Question Is there a solution σ with $f(\mathrm{enc}(\sigma)) = 0$?

Word equations and counting functions



$f(w)$ = number of b's after a's, minus c's

Automaton with labels in $A^* \times \mathbb{Z}^n$ $\rightsquigarrow$ counting function $f\colon A^* \to \mathbb{Z}^n$

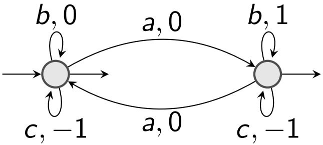
Word equations with counting constraints

Given Word equation (U, V) over $(A \cup \mathcal{X})^*$, counting function $f\colon (A \cup \{\#\})^* \to \mathbb{Z}^n$

Question Is there a solution σ with $f(\mathrm{enc}(\sigma)) = 0$?

- Length functions: $f(\mathrm{enc}(\sigma)) = |\sigma(X)|$ (→ open problem)

Word equations and counting functions



$f(w)$ = number of b's after a's, minus c's

Automaton with labels in $A^* \times \mathbb{Z}^n$ ⇝ counting function $f\colon A^* \to \mathbb{Z}^n$

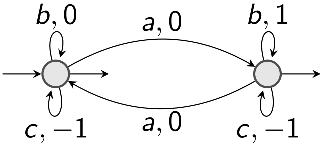
Word equations with counting constraints

Given Word equation (U, V) over $(A \cup \mathcal{X})^*$, counting function $f\colon (A \cup \{\#\})^* \to \mathbb{Z}^n$

Question Is there a solution σ with $f(\mathrm{enc}(\sigma)) = 0$?

- Length functions: $f(\mathrm{enc}(\sigma)) = |\sigma(X)|$ (→ open problem)
- Counting letters: $f(\mathrm{enc}(\sigma)) = |\sigma(X)|_a$ (→ undecidable)

Word equations and counting functions



$f(w) =$ number of b's after a's, minus c's

Automaton with labels in $A^* \times \mathbb{Z}^n$ $\rightsquigarrow$ counting function $f\colon A^* \to \mathbb{Z}^n$

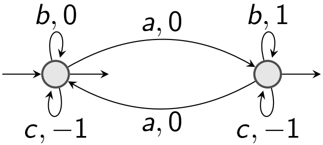
Word equations with counting constraints

Given Word equation (U, V) over $(A \cup \mathcal{X})^*$, counting function $f\colon (A \cup \{\#\})^* \to \mathbb{Z}^n$

Question Is there a solution σ with $f(\mathrm{enc}(\sigma)) = 0$?

- Length functions: $f(\mathrm{enc}(\sigma)) = |\sigma(X)|$ (→ open problem)
- Counting letters: $f(\mathrm{enc}(\sigma)) = |\sigma(X)|_a$ (→ undecidable)
- Counting positions satisfying MSO properties

Word equations and counting functions



$f(w)$ = number of b's after a's, minus c's

Automaton with labels in $A^* \times \mathbb{Z}^n$ $\rightsquigarrow$ counting function $f\colon A^* \to \mathbb{Z}^n$

Word equations with counting constraints

Given Word equation (U, V) over $(A \cup \mathcal{X})^*$, counting function $f\colon (A \cup \{\#\})^* \to \mathbb{Z}^n$

Question Is there a solution σ with $f(\mathrm{enc}(\sigma)) = 0$?

- Length functions: $f(\mathrm{enc}(\sigma)) = |\sigma(X)|$ (→ open problem)
- Counting letters: $f(\mathrm{enc}(\sigma)) = |\sigma(X)|_a$ (→ undecidable)
- Counting positions satisfying MSO properties
- Linear combinations: $f(w) = \lambda_1 f_1(w) + \cdots + \lambda_n f_n(w)$

Corollary (to our main result)

Given a word equation, rational constraints, and counting function f, one can decide if there is a solution σ with $f(\mathsf{enc}(\sigma)) \neq 0$.

Corollary (to our main result)

Given a word equation, rational constraints, and counting function f, one can decide if there is a solution σ with $f(\mathsf{enc}(\sigma)) \neq 0$.

(First decidable extension of word equations by counting constraints)

Corollary (to our main result)

Given a word equation, rational constraints, and counting function f, one can decide if there is a solution σ with $f(\mathrm{enc}(\sigma)) \neq 0$.

(First decidable extension of word equations by counting constraints)

1. Given $f\colon (A \cup \{\#\})^* \to \mathbb{Z}$, and a word equation (U, V) with rational constraints, construct indexed language

$$L = \{\mathrm{enc}(\sigma) \mid \sigma \text{ is a solution}\}.$$

Corollary (to our main result)

Given a word equation, rational constraints, and counting function f, one can decide if there is a solution σ with $f(\mathrm{enc}(\sigma)) \neq 0$.

(First decidable extension of word equations by counting constraints)

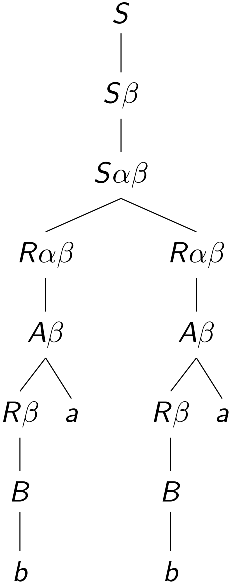
1. Given $f\colon (A \cup \{\#\})^* \to \mathbb{Z}$, and a word equation (U, V) with rational constraints, construct indexed language
$$L = \{\mathrm{enc}(\sigma) \mid \sigma \text{ is a solution}\}.$$
2. From L, construct indexed language K with
$$\{f(\mathrm{enc}(\sigma)) \mid \sigma \text{ is a solution}\} = \{|w|_a - |w|_b \mid w \in K\}$$

Corollary (to our main result)

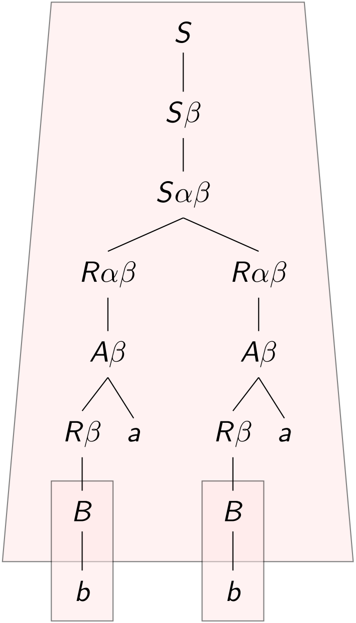
Given a word equation, rational constraints, and counting function f, one can decide if there is a solution σ with $f(\mathrm{enc}(\sigma)) \neq 0$.

(First decidable extension of word equations by counting constraints)

1. Given $f\colon (A \cup \{\#\})^* \to \mathbb{Z}$, and a word equation (U, V) with rational constraints, construct indexed language
$$L = \{\mathrm{enc}(\sigma) \mid \sigma \text{ is a solution}\}.$$
2. From L, construct indexed language K with
$$\{f(\mathrm{enc}(\sigma)) \mid \sigma \text{ is a solution}\} = \{|w|_a - |w|_b \;\mid\; w \in K\}$$
3. Check if $K \subseteq$ same number of a's and b's.

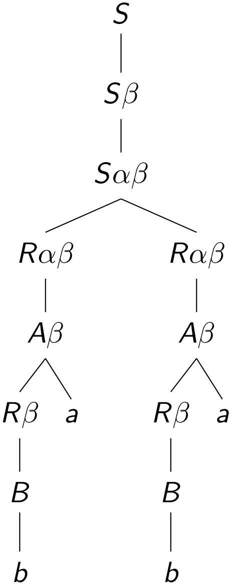


Building trees



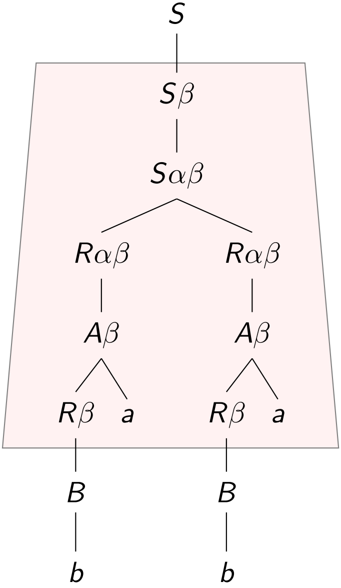
Building trees

- Plugging



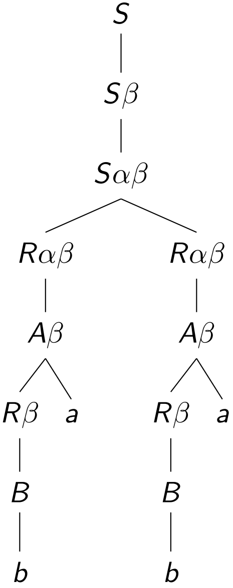
Building trees

- Plugging
- Nesting



Building trees

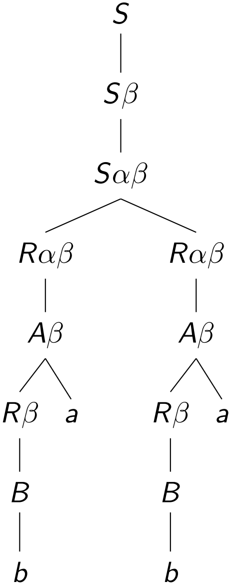
- Plugging
- Nesting



Building trees

- Plugging
- Nesting

$\rightsquigarrow$ operator $\mathcal{D}$ on subsets of $\mathbb{N}^{N \cup T}$

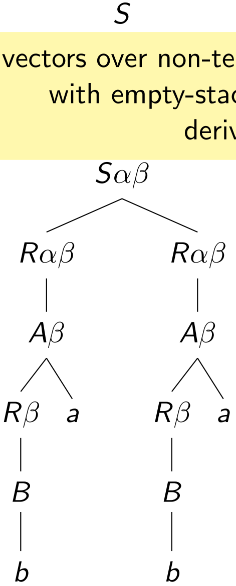


Building trees

- Plugging
- Nesting

$\rightsquigarrow$ operator $\mathcal{D}$ on subsets of $\mathbb{N}^{N \cup T}$ with

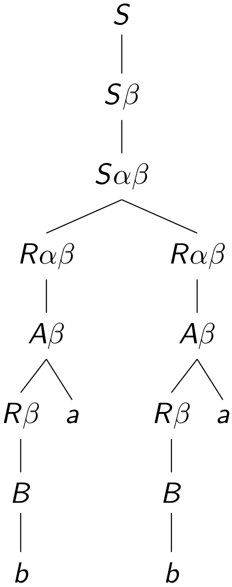
$$\text{derivable vectors} = \bigcup_{i \geqslant 0} \mathcal{D}^i(\varnothing)$$



vectors over non-terminals and terminals with empty-stack to empty-stack derivations

$\rightsquigarrow$ operator $\mathcal{D}$ on subsets of $\mathbb{N}^{N \cup T}$ with

$$\text{derivable vectors} = \bigcup_{i \geqslant 0} \mathcal{D}^i(\varnothing)$$



Building trees

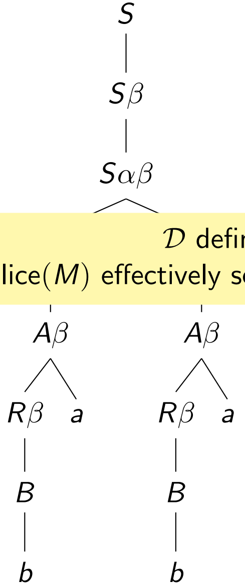
- Plugging
- Nesting

$\rightsquigarrow$ operator $\mathcal{D}$ on subsets of $\mathbb{N}^{N \cup T}$ with

$$\text{derivable vectors} = \bigcup_{i \geqslant 0} \mathcal{D}^i(\varnothing)$$

Show that

$$\text{Slice}(\bigcup_{i \geqslant 0} \mathcal{D}^i(\varnothing)) = \bigcup_{i \geqslant 0} (\text{Slice} \circ \mathcal{D})^i(\varnothing))$$



Building trees

- Plugging
- Nesting

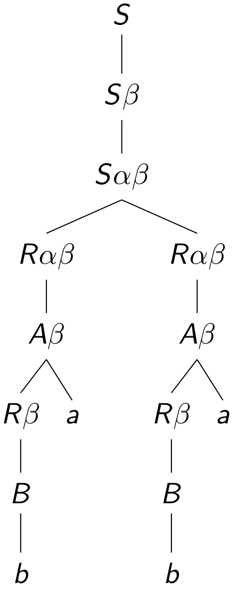
⇝ operator $\mathcal{D}$ on subsets of $\mathbb{N}^{N \cup T}$ with

$\mathcal{D}$ definable in Presburger arithmetic

Slice(M) effectively semilinear for semilinear M (Grabowski 1981)

Show that

$$\mathrm{Slice}\Big(\bigcup_{i \geqslant 0} \mathcal{D}^i(\varnothing)\Big) = \bigcup_{i \geqslant 0} (\mathrm{Slice} \circ \mathcal{D})^i(\varnothing))$$



Building trees

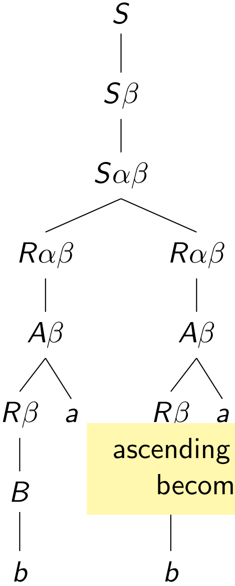
- Plugging
- Nesting

$\rightsquigarrow$ operator $\mathcal{D}$ on subsets of $\mathbb{N}^{N \cup T}$ with

$$\text{derivable vectors} = \bigcup_{i \geqslant 0} \mathcal{D}^i(\varnothing)$$

Show that

$$\begin{aligned} \text{Slice}\Big(\bigcup_{i \geqslant 0} \mathcal{D}^i(\varnothing)\Big) &= \bigcup_{i \geqslant 0} (\text{Slice} \circ \mathcal{D})^i(\varnothing)) \\ &= \bigcup_{i=0}^{n} (\text{Slice} \circ \mathcal{D})^i(\varnothing) \end{aligned}$$



Building trees

- Plugging
- Nesting

$\rightsquigarrow$ operator $\mathcal{D}$ on subsets of $\mathbb{N}^{N\cup T}$ with

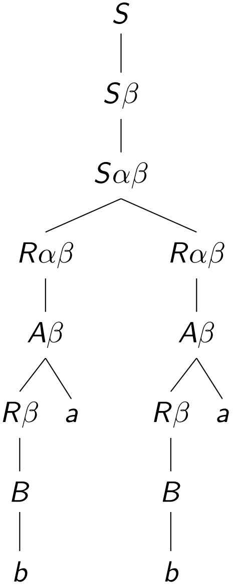
$$\text{derivable vectors} = \bigcup_{i\geqslant 0} \mathcal{D}^i(\varnothing)$$

Show that

$$\mathrm{Slice}(\bigcup_{i\geqslant 0} \mathcal{D}^i(\varnothing)) = \bigcup_{i\geqslant 0} (\mathrm{Slice}\circ\mathcal{D})^i(\varnothing))$$

$$= \bigcup_{i=0}^{n} (\mathrm{Slice}\circ\mathcal{D})^i(\varnothing)$$

ascending chains of slices become stationary



Building trees

- Plugging
- Nesting

⇝ operator $\mathcal{D}$ on subsets of $\mathbb{N}^{N \cup T}$ with

$$\text{derivable vectors} = \bigcup_{i \geqslant 0} \mathcal{D}^i(\varnothing)$$

Show that

$$\mathrm{Slice}\Big(\bigcup_{i \geqslant 0} \mathcal{D}^i(\varnothing)\Big) = \bigcup_{i \geqslant 0} (\mathrm{Slice} \circ \mathcal{D})^i(\varnothing))$$

$$= \bigcup_{i=0}^{n} (\mathrm{Slice} \circ \mathcal{D})^i(\varnothing)$$

Key property of slices

Slice(derivable vectors $\cap\ \mathbb{N}^T$) = Slice(derivable vectors) $\cap\ \mathbb{N}^T$

Ingredient I: Ascending chains

Theorem (Eilenberg & Schützenberger 1969)

If $S_1 \subseteq S_2 \subseteq \cdots$ *are slices, then there is an* $i \geqslant 1$ *with* $S_i = S_{i+1} = \cdots$.

Ingredient I: Ascending chains

Theorem (Eilenberg & Schützenberger 1969)

If $S_1 \subseteq S_2 \subseteq \cdots$ *are slices, then there is an* $i \geqslant 1$ *with* $S_i = S_{i+1} = \cdots$.

Lemma

Slices are finitely generated (as slices).

Ingredient I: Ascending chains

Theorem (Eilenberg & Schützenberger 1969)

If $S_1 \subseteq S_2 \subseteq \cdots$ are slices, then there is an $i \geqslant 1$ with $S_i = S_{i+1} = \cdots$.

Lemma

Slices are finitely generated (as slices).

Proof.

For a slice $S \subseteq \mathbb{N}^d$, consider a linear set

$$L_i = \{u + \lambda_1 u_1 + \cdots + \lambda_n u_n \mid \lambda_1, \ldots, \lambda_n \in \mathbb{N}\}$$

in S and consider the finite set $F_i = \{u, u + u_1, u + u_2, \ldots, u + u_n\}$. Then S is generated by $\bigcup_i F_i$. □

Ingredient I: Ascending chains

Theorem (Eilenberg & Schützenberger 1969)

If $S_1 \subseteq S_2 \subseteq \cdots$ *are slices, then there is an* $i \geqslant 1$ *with* $S_i = S_{i+1} = \cdots$.

Lemma

Slices are finitely generated (as slices).

Proof.

For a slice $S \subseteq \mathbb{N}^d$, consider a linear set

$$L_i = \{u + \lambda_1 u_1 + \cdots + \lambda_n u_n \mid \lambda_1, \ldots, \lambda_n \in \mathbb{N}\}$$

in S and consider the finite set $F_i = \{u, u + u_1, u + u_2, \ldots, u + u_n\}$. Then S is generated by $\bigcup_i F_i$. □

If $S_1 \subseteq S_2 \subseteq \cdots$, then $\bigcup_j S_j$ is a slice, and thus finitely generated.

Ingredient I: Ascending chains

Theorem (Eilenberg & Schützenberger 1969)

If $S_1 \subseteq S_2 \subseteq \cdots$ are slices, then there is an $i \geqslant 1$ with $S_i = S_{i+1} = \cdots$.

Lemma

Slices are finitely generated (as slices).

Proof.

For a slice $S \subseteq \mathbb{N}^d$, consider a linear set

$$L_i = \{u + \lambda_1 u_1 + \cdots + \lambda_n u_n \mid \lambda_1, \ldots, \lambda_n \in \mathbb{N}\}$$

in S and consider the finite set $F_i = \{u, u + u_1, u + u_2, \ldots, u + u_n\}$. Then S is generated by $\bigcup_i F_i$. □

If $S_1 \subseteq S_2 \subseteq \cdots$, then $\bigcup_j S_j$ is a slice, and thus finitely generated. These finitely many vectors must occur in some S_i, hence $S_i = S_{i+1} = \cdots$.

Ingredient II: Slice closures of semilinear sets

Theorem ($\approx$ Grabowski 1981)

Given a semilinear $M \subseteq \mathbb{N}^d$, the set $\mathrm{Slice}(M)$ is effectively semilinear.

Ingredient II: Slice closures of semilinear sets

Theorem ($\approx$ Grabowski 1981)

Given a semilinear $M \subseteq \mathbb{N}^d$, the set $\mathsf{Slice}(M)$ is effectively semilinear.

Observation

Given semilinear $M \subseteq \mathbb{N}^d$ and $u \in \mathbb{N}^d$, one can decide whether $u \in \mathsf{Slice}(M)$.

Ingredient II: Slice closures of semilinear sets

Theorem ($\approx$ Grabowski 1981)

Given a semilinear $M \subseteq \mathbb{N}^d$, the set $\mathsf{Slice}(M)$ *is effectively semilinear.*

Observation

Given semilinear $M \subseteq \mathbb{N}^d$ and $u \in \mathbb{N}^d$, one can decide whether $u \in \mathsf{Slice}(M)$.

Dynamic programming:

Ingredient II: Slice closures of semilinear sets

Theorem ($\approx$ Grabowski 1981)

Given a semilinear $M \subseteq \mathbb{N}^d$, the set $\mathsf{Slice}(M)$ *is effectively semilinear.*

Observation

Given semilinear $M \subseteq \mathbb{N}^d$ and $u \in \mathbb{N}^d$, one can decide whether $u \in \mathsf{Slice}(M)$.

Dynamic programming:
a vector can only be obtained from smaller vectors.

Ingredient II: Slice closures of semilinear sets

Theorem ($\approx$ Grabowski 1981)

Given a semilinear $M \subseteq \mathbb{N}^d$, the set $\mathsf{Slice}(M)$ *is effectively semilinear.*

Observation

Given semilinear $M \subseteq \mathbb{N}^d$ and $u \in \mathbb{N}^d$, one can decide whether $u \in \mathsf{Slice}(M)$.

Dynamic programming:
a vector can only be obtained from smaller vectors.

Computing $\mathsf{Slice}(M)$:

Ingredient II: Slice closures of semilinear sets

Theorem (≈ Grabowski 1981)

Given a semilinear $M \subseteq \mathbb{N}^d$, the set $\mathsf{Slice}(M)$ is effectively semilinear.

Observation

Given semilinear $M \subseteq \mathbb{N}^d$ and $u \in \mathbb{N}^d$, one can decide whether $u \in \mathsf{Slice}(M)$.

Dynamic programming:
a vector can only be obtained from smaller vectors.

Computing $\mathsf{Slice}(M)$:
Enumerate semilinear sets R. For each R:

Ingredient II: Slice closures of semilinear sets

Theorem ($\approx$ Grabowski 1981)

Given a semilinear $M \subseteq \mathbb{N}^d$, the set $\mathsf{Slice}(M)$ is effectively semilinear.

Observation

Given semilinear $M \subseteq \mathbb{N}^d$ and $u \in \mathbb{N}^d$, one can decide whether $u \in \mathsf{Slice}(M)$.

Dynamic programming:
a vector can only be obtained from smaller vectors.

Computing $\mathsf{Slice}(M)$:
Enumerate semilinear sets R. For each R:

- Check $R \subseteq \mathsf{Slice}(M)$: Build generating set F and check $F \subseteq \mathsf{Slice}(M)$

Ingredient II: Slice closures of semilinear sets

Theorem ($\approx$ Grabowski 1981)

Given a semilinear $M \subseteq \mathbb{N}^d$, the set $\mathsf{Slice}(M)$ *is effectively semilinear.*

Observation

Given semilinear $M \subseteq \mathbb{N}^d$ and $u \in \mathbb{N}^d$, one can decide whether $u \in \mathsf{Slice}(M)$.

Dynamic programming:
a vector can only be obtained from smaller vectors.

Computing $\mathsf{Slice}(M)$:
Enumerate semilinear sets R. For each R:

- Check $R \subseteq \mathsf{Slice}(M)$: Build generating set F and check $F \subseteq \mathsf{Slice}(M)$
- Check that R is a slice: can be stated in Presburger arithmetic

Ingredient II: Slice closures of semilinear sets

Theorem ($\approx$ Grabowski 1981)

Given a semilinear $M \subseteq \mathbb{N}^d$, the set $\mathsf{Slice}(M)$ is effectively semilinear.

Observation

Given semilinear $M \subseteq \mathbb{N}^d$ and $u \in \mathbb{N}^d$, one can decide whether $u \in \mathsf{Slice}(M)$.

Dynamic programming:
a vector can only be obtained from smaller vectors.

Computing $\mathsf{Slice}(M)$:
Enumerate semilinear sets R. For each R:

- Check $R \subseteq \mathsf{Slice}(M)$: Build generating set F and check $F \subseteq \mathsf{Slice}(M)$
- Check that R is a slice: can be stated in Presburger arithmetic

If both checks succeed, we know $R = \mathsf{Slice}(M)$.

Ingredient III: Slice closure of intersection

Key property of slices

For $M \subseteq \mathbb{N}^{N \cup T}$, we have $\mathsf{Slice}(M \cap \mathbb{N}^{T}) = \mathsf{Slice}(M) \cap \mathbb{N}^{T}$.

Ingredient III: Slice closure of intersection

Key property of slices

For $M \subseteq \mathbb{N}^{N \cup T}$, we have $\mathsf{Slice}(M \cap \mathbb{N}^{T}) = \mathsf{Slice}(M) \cap \mathbb{N}^{T}$.

$\subseteq$:

Ingredient III: Slice closure of intersection

Key property of slices

For $M \subseteq \mathbb{N}^{N \cup T}$, we have $\mathsf{Slice}(M \cap \mathbb{N}^{T}) = \mathsf{Slice}(M) \cap \mathbb{N}^{T}$.

$\subseteq$: $\mathbb{N}^{T}$ is closed under the slice operation

Ingredient III: Slice closure of intersection

Key property of slices

For $M \subseteq \mathbb{N}^{N \cup T}$, we have $\mathsf{Slice}(M \cap \mathbb{N}^T) = \mathsf{Slice}(M) \cap \mathbb{N}^T$.

$\subseteq$: $\mathbb{N}^T$ is closed under the slice operation

$\supseteq$:

Ingredient III: Slice closure of intersection

Key property of slices

For $M \subseteq \mathbb{N}^{N \cup T}$, we have $\mathsf{Slice}(M \cap \mathbb{N}^T) = \mathsf{Slice}(M) \cap \mathbb{N}^T$.

$\subseteq$: $\mathbb{N}^T$ is closed under the slice operation
$\supseteq$: $\mathbb{N}^T$ can only be produced from vectors in $\mathbb{N}^T$

Ingredient III: Slice closure of intersection

Key property of slices

For $M \subseteq \mathbb{N}^{N \cup T}$, we have $\mathsf{Slice}(M \cap \mathbb{N}^{T}) = \mathsf{Slice}(M) \cap \mathbb{N}^{T}$.

$\subseteq$: $\mathbb{N}^{T}$ is closed under the slice operation
$\supseteq$: $\mathbb{N}^{T}$ can only be produced from vectors in $\mathbb{N}^{T}$

Slices closure vs. other counting closures

Not true for existing closure operators (affine hull, Zariski closure)!

Conclusion

Theorem (Ciobanu & Z., LICS 2024)

For indexed L, its slice closure $\mathsf{Slice}(\Psi(L))$ is effectively semilinear.

Conclusion

Theorem (Ciobanu & Z., LICS 2024)

For indexed L, its slice closure $\mathrm{Slice}(\Psi(L))$ is effectively semilinear.

Corollary

Given a word equation, rational constraints, and counting function f, one can decide if there is a solution σ with $f(\mathrm{enc}(\sigma)) \neq 0$.

Conclusion

Theorem (Ciobanu & Z., LICS 2024)

For indexed L, its slice closure $\mathsf{Slice}(\Psi(L))$ is effectively semilinear.

Corollary

Given a word equation, rational constraints, and counting function f, one can decide if there is a solution σ with $f(\mathsf{enc}(\sigma)) \neq 0$.

Ongoing work with Corto Mascle & Richard Mandel

Conclusion

Theorem (Ciobanu & Z., LICS 2024)

For indexed L, its slice closure $\mathrm{Slice}(\Psi(L))$ is effectively semilinear.

Corollary

Given a word equation, rational constraints, and counting function f, one can decide if there is a solution σ with $f(\mathrm{enc}(\sigma)) \neq 0$.

Ongoing work with Corto Mascle & Richard Mandel

- Complexity bounds? (So far, not even Ackermann...)

Conclusion

Theorem (Ciobanu & Z., LICS 2024)

For indexed L, its slice closure $\mathrm{Slice}(\Psi(L))$ is effectively semilinear.

Corollary

Given a word equation, rational constraints, and counting function f, one can decide if there is a solution σ with $f(\mathrm{enc}(\sigma)) \neq 0$.

Ongoing work with Corto Mascle & Richard Mandel

- Complexity bounds? (So far, not even Ackermann...)
- Polynomial equations?

Conclusion

Theorem (Ciobanu & Z., LICS 2024)

For indexed L, its slice closure $\mathsf{Slice}(\Psi(L))$ is effectively semilinear.

Corollary

Given a word equation, rational constraints, and counting function f, one can decide if there is a solution σ with $f(\mathsf{enc}(\sigma)) \neq 0$.

Ongoing work with Corto Mascle & Richard Mandel

- Complexity bounds? (So far, not even Ackermann...)
- Polynomial equations?
- Higher order?

Conclusion

Theorem (Ciobanu & Z., LICS 2024)

For indexed L, its slice closure $\mathrm{Slice}(\Psi(L))$ is effectively semilinear.

Corollary

Given a word equation, rational constraints, and counting function f, one can decide if there is a solution σ with $f(\mathrm{enc}(\sigma)) \neq 0$.

Ongoing work with Corto Mascle & Richard Mandel

- Complexity bounds? (So far, not even Ackermann...)
- Polynomial equations?
- Higher order?

Exploit EDT0L

Here: relied on indexed languages (closure properties!)

Conclusion

Theorem (Ciobanu & Z., LICS 2024)

For indexed L, its slice closure $\mathsf{Slice}(\Psi(L))$ is effectively semilinear.

Corollary

Given a word equation, rational constraints, and counting function f, one can decide if there is a solution σ with $f(\mathsf{enc}(\sigma)) \neq 0$.

Ongoing work with Corto Mascle & Richard Mandel

- Complexity bounds? (So far, not even Ackermann...)
- Polynomial equations?
- Higher order?

Exploit EDT0L

Here: relied on indexed languages (closure properties!)
Can Ciobanu, Diekert & Elder 2015 be strengthened, to avoid detour?