Computational completeness of equations over sets of natural numbers

Artur Jeż   Alexander Okhotin

Wrocław, Poland

Turku, Finland

July 7, 2008

Language equations

$$\begin{cases} \varphi_1(X_1, \ldots, X_n) & = & \psi_1(X_1, \ldots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \ldots, X_n) & = & \psi_m(X_1, \ldots, X_n) \end{cases}$$

Language equations

$$\begin{cases} \varphi_1(X_1, \dots, X_n) & = & \psi_1(X_1, \dots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \dots, X_n) & = & \psi_m(X_1, \dots, X_n) \end{cases}$$

- X_i: subset of Σ^*.

Language equations

$$\begin{cases} \varphi_1(X_1, \ldots, X_n) & = & \psi_1(X_1, \ldots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \ldots, X_n) & = & \psi_m(X_1, \ldots, X_n) \end{cases}$$

- X_i: subset of Σ^*.
- φ_i: variables, constants, operations on sets.

Language equations

$$\begin{cases} \varphi_1(X_1, \dots, X_n) & = & \psi_1(X_1, \dots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \dots, X_n) & = & \psi_m(X_1, \dots, X_n) \end{cases}$$

- X_i: subset of Σ^*.
- φ_i: variables, constants, operations on sets.

Solutions: least, greatest, unique.

Language equations

$$\begin{cases} \varphi_1(X_1, \ldots, X_n) & = & \psi_1(X_1, \ldots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \ldots, X_n) & = & \psi_m(X_1, \ldots, X_n) \end{cases}$$

- X_i: subset of Σ^*.
- φ_i: variables, constants, operations on sets.

Solutions: least, greatest, unique.

Example

$X = XX \cup \{a\}X\{b\} \cup \{\varepsilon\}$

Language equations

$$\begin{cases} \varphi_1(X_1, \dots, X_n) & = & \psi_1(X_1, \dots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \dots, X_n) & = & \psi_m(X_1, \dots, X_n) \end{cases}$$

- X_i: subset of Σ^*.
- φ_i: variables, constants, operations on sets.

Solutions: least, greatest, unique.

Example

$X = XX \cup \{a\}X\{b\} \cup \{\varepsilon\}$

Least solution: the Dyck language.

Language equations

$$\begin{cases} \varphi_1(X_1, \ldots, X_n) & = & \psi_1(X_1, \ldots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \ldots, X_n) & = & \psi_m(X_1, \ldots, X_n) \end{cases}$$

- X_i: subset of Σ^*.
- φ_i: variables, constants, operations on sets.

Solutions: least, greatest, unique.

Example

$X = XX \cup \{a\}X\{b\} \cup \{\varepsilon\}$

Least solution: the Dyck language.

Greatest solution: Σ^*.

Research on language equations

- Ginsburg and Rice (1962): representation of CFGs.

Research on language equations

- Ginsburg and Rice (1962): representation of CFGs.
- Conway (1971); Karhumäki et al. (2000–); Kunc (2005): $LX = XL$.

Research on language equations

- Ginsburg and Rice (1962): representation of CFGs.
- Conway (1971); Karhumäki et al. (2000–); Kunc (2005): $LX = XL$.
- Brzozowski/Leiss (1980): alternating finite automata.

Research on language equations

- Ginsburg and Rice (1962): representation of CFGs.
- Conway (1971); Karhumäki et al. (2000–); Kunc (2005): $LX = XL$.
- Brzozowski/Leiss (1980): alternating finite automata.
- Charatonik (1994): undecidability for equations with $\{\cdot, \cup, \cap, \sim\}$.

Research on language equations

- Ginsburg and Rice (1962): representation of CFGs.
- Conway (1971); Karhumäki et al. (2000–); Kunc (2005): $LX = XL$.
- Brzozowski/Leiss (1980): alternating finite automata.
- Charatonik (1994): undecidability for equations with $\{\cdot, \cup, \cap, \sim\}$.
- Baader/Narendran (1998), Bala (2004): complexity.

Research on language equations

- Ginsburg and Rice (1962): representation of CFGs.
- Conway (1971); Karhumäki et al. (2000–); Kunc (2005): $LX = XL$.
- Brzozowski/Leiss (1980): alternating finite automata.
- Charatonik (1994): undecidability for equations with $\{\cdot, \cup, \cap, \sim\}$.
- Baader/Narendran (1998), Bala (2004): complexity.
- Okhotin (2001–present): equations with Boolean operations.

Research on language equations

- Ginsburg and Rice (1962): representation of CFGs.
- Conway (1971); Karhumäki et al. (2000–); Kunc (2005): $LX = XL$.
- Brzozowski/Leiss (1980): alternating finite automata.
- Charatonik (1994): undecidability for equations with $\{\cdot, \cup, \cap, \sim\}$.
- Baader/Narendran (1998), Bala (2004): complexity.
- Okhotin (2001–present): equations with Boolean operations.
- Leiss (1995), Okhotin/Yakimova (2006), Jeż (2007), Jeż/Okhotin (2007–present): equations over $\{a\}$.

Computational completeness of language equations

- Language equations over Σ, with $|\Sigma| \geqslant 2$.

Computational completeness of language equations

- Language equations over Σ, with $|\Sigma| \geqslant 2$.

Theorem (Okhotin, ICALP 2003)

$L \subseteq \Sigma^*$ is given by unique solution of a system with $\{\cup, \cap, \sim, \cdot\}$
if and only if
L is recursive.

Computational completeness of language equations

- Language equations over Σ, with $|\Sigma| \geqslant 2$.

Theorem (Okhotin, ICALP 2003)

$L \subseteq \Sigma^*$ is given by unique solution of a system with $\{\cup, \cap, \sim, \cdot\}$
if and only if
L is recursive.

- $X \subseteq Y$: written as $X \cup Y = Y$.

Computational completeness of language equations

- Language equations over Σ, with $|\Sigma| \geqslant 2$.

Theorem (Okhotin, ICALP 2003)

$L \subseteq \Sigma^*$ is given by unique solution of a system with $\{\cup, \cap, \sim, \cdot\}$
if and only if
L is recursive.

- $X \subseteq Y$: written as $X \cup Y = Y$.
- $\mathrm{VALC}(T) = \{w\sharp C_T(w) | w \in L(T)\}$: accepting computations.

Computational completeness of language equations

- Language equations over Σ, with $|\Sigma| \geqslant 2$.

Theorem (Okhotin, ICALP 2003)

$L \subseteq \Sigma^*$ is given by unique solution of a system with $\{\cup, \cap, \sim, \cdot\}$
if and only if
L is recursive.

- $X \subseteq Y$: written as $X \cup Y = Y$.
- $\mathrm{VALC}(T) = \{w\sharp C_T(w) | w \in L(T)\}$: accepting computations.
- $\mathrm{VALC}^{rej}(T) = \{w\sharp C_T(w) | w \notin L(T)\}$: rejecting computations.

Computational completeness of language equations

- Language equations over Σ, with $|\Sigma| \geqslant 2$.

Theorem (Okhotin, ICALP 2003)

$L \subseteq \Sigma^*$ is given by unique solution of a system with $\{\cup, \cap, \sim, \cdot\}$
if and only if
L is recursive.

- $X \subseteq Y$: written as $X \cup Y = Y$.
- $\text{VALC}(T) = \{w\sharp C_T(w) | w \in L(T)\}$: accepting computations.
- $\text{VALC}^{rej}(T) = \{w\sharp C_T(w) | w \notin L(T)\}$: rejecting computations.
- $L(T)$ is a unique solution of

$$\text{VALC}(T) \subseteq X\sharp\Sigma^*$$
$$X\sharp\Sigma^* \subseteq \overline{\text{VALC}^{rej}(T)}$$

Computational completeness of language equations

- Language equations over Σ, with $|\Sigma| \geqslant 2$.

Theorem (Okhotin, ICALP 2003)

$L \subseteq \Sigma^*$ is given by unique solution of a system with $\{\cup, \cap, \sim, \cdot\}$
if and only if
L is recursive.

- $X \subseteq Y$: written as $X \cup Y = Y$.
- $\mathrm{VALC}(T) = \{w\sharp C_T(w) | w \in L(T)\}$: accepting computations.
- $\mathrm{VALC}^{rej}(T) = \{w\sharp C_T(w) | w \notin L(T)\}$: rejecting computations.
- $L(T)$ is a unique solution of

$$\mathrm{VALC}(T) \subseteq X\sharp\Sigma^*$$
$$X\sharp\Sigma^* \subseteq \overline{\mathrm{VALC}^{rej}(T)}$$

- Multiple-letter alphabet essentially used.

Computational completeness of language equations

- Language equations over Σ, with $|\Sigma| \geqslant 2$.

Theorem (Okhotin, ICALP 2003)

$L \subseteq \Sigma^*$ is given by unique solution of a system with $\{\cup, \cap, \sim, \cdot\}$
if and only if
L is recursive.

- $X \subseteq Y$: written as $X \cup Y = Y$.
- $\mathrm{VALC}(T) = \{w \sharp C_T(w) | w \in L(T)\}$: accepting computations.
- $\mathrm{VALC}^{rej}(T) = \{w \sharp C_T(w) | w \notin L(T)\}$: rejecting computations.
- $L(T)$ is a unique solution of

$$\mathrm{VALC}(T) \subseteq X \sharp \Sigma^*$$
$$X \sharp \Sigma^* \subseteq \overline{\mathrm{VALC}^{rej}(T)}$$

- Multiple-letter alphabet essentially used.

✓ Remaking the argument for the unary case!

Unary languages as sets of numbers

- $\Sigma = \{a\}$.

Unary languages as sets of numbers

- $\Sigma = \{a\}$.
- a^n $\longleftrightarrow$ number n

Unary languages as sets of numbers

- $\Sigma = \{a\}$.
- a^n $\longleftrightarrow$ number n
- Language $\longleftrightarrow$ set of numbers

Unary languages as sets of numbers

- $\Sigma = \{a\}$.
- $a^n \quad \longleftrightarrow \quad$ number n
- Language $\quad \longleftrightarrow \quad$ set of numbers
- $K \cdot L \quad \longleftrightarrow \quad X + Y = \{x + y \mid x \in X, y \in Y\}$

Unary languages as sets of numbers

- $\Sigma = \{a\}$.
- a^n $\longleftrightarrow$ number n
- Language $\longleftrightarrow$ set of numbers
- $K \cdot L$ $\longleftrightarrow$ $X + Y = \{x + y \mid x \in X, y \in Y\}$
- Regular $\longleftrightarrow$ ultimately periodic

Unary languages as sets of numbers

- $\Sigma = \{a\}$.
- a^n $\longleftrightarrow$ number n
- Language $\longleftrightarrow$ set of numbers
- $K \cdot L$ $\longleftrightarrow$ $X + Y = \{x + y \mid x \in X, y \in Y\}$
- Regular $\longleftrightarrow$ ultimately periodic

✓ Equations over sets of numbers.

Unary languages as sets of numbers

- $\Sigma = \{a\}$.
- a^n $\longleftrightarrow$ number n
- Language $\longleftrightarrow$ set of numbers
- $K \cdot L$ $\longleftrightarrow$ $X + Y = \{x + y \mid x \in X, y \in Y\}$
- Regular $\longleftrightarrow$ ultimately periodic

✓ Equations over sets of numbers.

$$\begin{cases} \varphi_1(X_1, \ldots, X_n) & = & \psi_1(X_1, \ldots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \ldots, X_n) & = & \psi_m(X_1, \ldots, X_n) \end{cases}$$

Unary languages as sets of numbers

- $\Sigma = \{a\}$.
- a^n $\longleftrightarrow$ number n
- Language $\longleftrightarrow$ set of numbers
- $K \cdot L$ $\longleftrightarrow$ $X + Y = \{x + y \mid x \in X, y \in Y\}$
- Regular $\longleftrightarrow$ ultimately periodic

✓ Equations over sets of numbers.

$$\begin{cases} \varphi_1(X_1, \ldots, X_n) & = & \psi_1(X_1, \ldots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \ldots, X_n) & = & \psi_m(X_1, \ldots, X_n) \end{cases}$$

- X_i: subset of $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$.

Unary languages as sets of numbers

- $\Sigma = \{a\}$.
- a^n $\longleftrightarrow$ number n
- Language $\longleftrightarrow$ set of numbers
- $K \cdot L$ $\longleftrightarrow$ $X + Y = \{x + y \mid x \in X, y \in Y\}$
- Regular $\longleftrightarrow$ ultimately periodic

✓ Equations over sets of numbers.

$$\begin{cases} \varphi_1(X_1, \ldots, X_n) & = & \psi_1(X_1, \ldots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \ldots, X_n) & = & \psi_m(X_1, \ldots, X_n) \end{cases}$$

- X_i: subset of $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$.
- φ_i: variables, singleton constants, operations on sets.

Resolved systems with $\{\cup, +\}$ and $\{\cup, \cap, +\}$

$$\begin{cases} X_1 & = & \varphi_1(X_1, \ldots, X_n) \\ & \vdots & \\ X_n & = & \varphi_n(X_1, \ldots, X_n) \end{cases}$$

Resolved systems with $\{\cup, +\}$ and $\{\cup, \cap, +\}$

$$\begin{cases} X_1 &= \varphi_1(X_1, \ldots, X_n) \\ &\vdots \\ X_n &= \varphi_n(X_1, \ldots, X_n) \end{cases}$$

Example

$X = \big(X + \{2\}\big) \cup \{0\}$

Resolved systems with $\{\cup, +\}$ and $\{\cup, \cap, +\}$

$$\begin{cases} X_1 & = & \varphi_1(X_1, \dots, X_n) \\ & \vdots & \\ X_n & = & \varphi_n(X_1, \dots, X_n) \end{cases}$$

Example

$X = \big(X + \{2\}\big) \cup \{0\}$ Unique solution: the even numbers

Resolved systems with $\{\cup, +\}$ and $\{\cup, \cap, +\}$

$$\begin{cases} X_1 & = & \varphi_1(X_1, \ldots, X_n) \\ & \vdots & \\ X_n & = & \varphi_n(X_1, \ldots, X_n) \end{cases}$$

Example

$X = \big(X + \{2\}\big) \cup \{0\}$ Unique solution: the even numbers

$S \rightarrow aaS \mid \varepsilon$

Resolved systems with $\{\cup, +\}$ and $\{\cup, \cap, +\}$

$$\begin{cases} X_1 = \varphi_1(X_1, \ldots, X_n) \\ \vdots \\ X_n = \varphi_n(X_1, \ldots, X_n) \end{cases}$$

Example

$X = (X + \{2\}) \cup \{0\}$ Unique solution: the even numbers

$S \to aaS \mid \varepsilon$

- Representing sets by unique solutions.

Resolved systems with $\{\cup, +\}$ and $\{\cup, \cap, +\}$

$$\begin{cases} X_1 = \varphi_1(X_1, \ldots, X_n) \\ \vdots \\ X_n = \varphi_n(X_1, \ldots, X_n) \end{cases}$$

Example

$X = (X + \{2\}) \cup \{0\}$ Unique solution: the even numbers

$S \to aaS \mid \varepsilon$

- Representing sets by unique solutions.
- With $\{\cup, +\}$: context-free grammars.

Resolved systems with $\{\cup, +\}$ and $\{\cup, \cap, +\}$

$$\begin{cases} X_1 = \varphi_1(X_1, \dots, X_n) \\ \vdots \\ X_n = \varphi_n(X_1, \dots, X_n) \end{cases}$$

Example

$X = (X + \{2\}) \cup \{0\}$ Unique solution: the even numbers

$S \to aaS \mid \varepsilon$

- Representing sets by unique solutions.
- With $\{\cup, +\}$: context-free grammars.

Theorem (Bar-Hillel et al., 1961)

Every context-free language over $\{a\}$ is regular.

Resolved systems with $\{\cup, +\}$ and $\{\cup, \cap, +\}$

$$\begin{cases} X_1 &= \varphi_1(X_1, \ldots, X_n) \\ &\vdots \\ X_n &= \varphi_n(X_1, \ldots, X_n) \end{cases}$$

Example

$X = (X + \{2\}) \cup \{0\}$ Unique solution: the even numbers

$S \to aaS \mid \varepsilon$

- Representing sets by unique solutions.
- With $\{\cup, +\}$: context-free grammars.

Theorem (Bar-Hillel et al., 1961)

Every context-free language over $\{a\}$ is regular.

- With $\{\cup, \cap, +\}$: conjunctive grammars.

Resolved systems with $\{\cup, +\}$ and $\{\cup, \cap, +\}$

$$\begin{cases} X_1 &= \varphi_1(X_1, \ldots, X_n) \\ &\vdots \\ X_n &= \varphi_n(X_1, \ldots, X_n) \end{cases}$$

Example

$X = (X + \{2\}) \cup \{0\}$ Unique solution: the even numbers

$S \to aaS \mid \varepsilon$

- Representing sets by unique solutions.
- With $\{\cup, +\}$: context-free grammars.

Theorem (Bar-Hillel et al., 1961)

Every context-free language over $\{a\}$ is regular.

- With $\{\cup, \cap, +\}$: conjunctive grammars.
- The power of conjunctive grammars over $\{a\}$?

Conjunctive grammars

Quadruple $G = (\Sigma, N, P, S)$, where...

Context-free grammars: Rules of the form

$$A \to \alpha$$

"If w is generated by α, then w is generated by A".

Conjunctive grammars

Quadruple $G = (\Sigma, N, P, S)$, where...

Context-free grammars: Rules of the form

$$A \to \alpha$$

"If w is generated by α, then w is generated by A".

✓ Multiple rules for A: *disjunction*.

Conjunctive grammars

Quadruple $G = (\Sigma, N, P, S)$, where...

Context-free grammars: Rules of the form

$$A \to \alpha$$

"If w is generated by α, then w is generated by A".

✓ Multiple rules for A: *disjunction*.

Conjunctive grammars (Okhotin, 2000) Rules of the form

$$A \to \alpha_1 \& \ldots \& \alpha_m$$

"If w is generated by each α_i, then w is generated by A".

Definition of conjunctive grammars

- Semantics by language equations:

$$A = \bigcup_{A \to \alpha_1 \& \ldots \& \alpha_m \in P} \bigcap_{i=1}^{m} \alpha_i$$

Definition of conjunctive grammars

- Semantics by language equations:

$$A = \bigcup_{A \to \alpha_1 \& \ldots \& \alpha_m \in P} \bigcap_{i=1}^{m} \alpha_i$$

  - $L_G(A)$ is the A-component of the least solution.

Definition of conjunctive grammars

- Semantics by language equations:

$$A = \bigcup_{A \to \alpha_1 \& \ldots \& \alpha_m \in P} \bigcap_{i=1}^{m} \alpha_i$$

  - $L_G(A)$ is the A-component of the least solution.
- Equivalent semantics by term rewriting.

Definition of conjunctive grammars

- Semantics by language equations:

$$A = \bigcup_{A \to \alpha_1 \& \ldots \& \alpha_m \in P} \bigcap_{i=1}^{m} \alpha_i$$

  - $L_G(A)$ is the A-component of the least solution.
- Equivalent semantics by term rewriting.
- Generated languages are in $DTIME(n^3) \cap DSPACE(n)$.

Definition of conjunctive grammars

- Semantics by language equations:

$$A = \bigcup_{A \to \alpha_1 \& \ldots \& \alpha_m \in P} \bigcap_{i=1}^{m} \alpha_i$$

  - $L_G(A)$ is the A-component of the least solution.
- Equivalent semantics by term rewriting.
- Generated languages are in $DTIME(n^3) \cap DSPACE(n)$.
- Efficient parsing: Generalized LR, recursive descent.

Definition of conjunctive grammars

- Semantics by language equations:

$$A = \bigcup_{A \to \alpha_1 \& \ldots \& \alpha_m \in P} \bigcap_{i=1}^{m} \alpha_i$$

  - $L_G(A)$ is the A-component of the least solution.
- Equivalent semantics by term rewriting.
- Generated languages are in $DTIME(n^3) \cap DSPACE(n)$.
- Efficient parsing: Generalized LR, recursive descent.
- Greater expressive power.

Definition of conjunctive grammars

- Semantics by language equations:

$$A = \bigcup_{A \to \alpha_1 \& \ldots \& \alpha_m \in P} \bigcap_{i=1}^{m} \alpha_i$$

  - $L_G(A)$ is the A-component of the least solution.
- Equivalent semantics by term rewriting.
- Generated languages are in $DTIME(n^3) \cap DSPACE(n)$.
- Efficient parsing: Generalized LR, recursive descent.
- Greater expressive power.
  - Conjunctive grammar for $\{a^{4^n} \mid n \geqslant 0\}$.

Using positional notation

- Numbers in base-k notation: strings over $\Sigma_k = \{0, 1, \ldots, k-1\}$.

Using positional notation

- Numbers in base-k notation: strings over $\Sigma_k = \{0, 1, \ldots, k-1\}$.
- Set of numbers $\leftrightarrow$ formal language over Σ_k

Using positional notation

- Numbers in base-k notation: strings over $\Sigma_k = \{0, 1, \ldots, k-1\}$.
- Set of numbers $\leftrightarrow$ formal language over Σ_k

Example (Jeż, DLT 2007)

$$X_1 = (X_2 + X_2 \cap X_1 + X_3) \cup \{1\}$$
$$X_2 = (X_{12} + X_2 \cap X_1 + X_1) \cup \{2\}$$
$$X_3 = (X_{12} + X_{12} \cap X_1 + X_2) \cup \{3\}$$
$$X_{12} = X_3 + X_3 \cap X_1 + X_2$$

Using positional notation

- Numbers in base-k notation: strings over $\Sigma_k = \{0, 1, \dots, k-1\}$.
- Set of numbers $\leftrightarrow$ formal language over Σ_k

Example (Jeż, DLT 2007)

$$X_1 = (X_2 + X_2 \cap X_1 + X_3) \cup \{1\}$$
$$X_2 = (X_{12} + X_2 \cap X_1 + X_1) \cup \{2\}$$
$$X_3 = (X_{12} + X_{12} \cap X_1 + X_2) \cup \{3\}$$
$$X_{12} = X_3 + X_3 \cap X_1 + X_2$$

$$((10^*)_4, (20^*)_4, (30^*)_4, (120^*)_4)$$

Using positional notation

- Numbers in base-k notation: strings over $\Sigma_k = \{0, 1, \dots, k-1\}$.
- Set of numbers $\leftrightarrow$ formal language over Σ_k

Example (Jeż, DLT 2007)

$$\begin{aligned}
X_1 &= (X_2 + X_2 \cap X_1 + X_3) \cup \{1\} \\
X_2 &= (X_{12} + X_2 \cap X_1 + X_1) \cup \{2\} \\
X_3 &= (X_{12} + X_{12} \cap X_1 + X_2) \cup \{3\} \\
X_{12} &= X_3 + X_3 \cap X_1 + X_2
\end{aligned}$$

$$((10^*)_4, (20^*)_4, (30^*)_4, (120^*)_4)$$

- $X_2 + X_2 = (20^*)_4 + (20^*)_4 =$

Using positional notation

- Numbers in base-k notation: strings over $\Sigma_k = \{0, 1, \dots, k-1\}$.
- Set of numbers $\leftrightarrow$ formal language over Σ_k

Example (Jeż, DLT 2007)

$$\begin{aligned}
X_1 &= (X_2 + X_2 \cap X_1 + X_3) \cup \{1\} \\
X_2 &= (X_{12} + X_2 \cap X_1 + X_1) \cup \{2\} \\
X_3 &= (X_{12} + X_{12} \cap X_1 + X_2) \cup \{3\} \\
X_{12} &= X_3 + X_3 \cap X_1 + X_2
\end{aligned}
\qquad ((10^*)_4, (20^*)_4, (30^*)_4, (120^*)_4)$$

- $X_2 + X_2 = (20^*)_4 + (20^*)_4 = (10^+)_4 \cup$

Using positional notation

- Numbers in base-k notation: strings over $\Sigma_k = \{0, 1, \dots, k-1\}$.
- Set of numbers $\leftrightarrow$ formal language over Σ_k

Example (Jeż, DLT 2007)

$$\begin{aligned}
X_1 &= (X_2 + X_2 \cap X_1 + X_3) \cup \{1\} \\
X_2 &= (X_{12} + X_2 \cap X_1 + X_1) \cup \{2\} \\
X_3 &= (X_{12} + X_{12} \cap X_1 + X_2) \cup \{3\} \\
X_{12} &= X_3 + X_3 \cap X_1 + X_2
\end{aligned}$$

$$((10^*)_4, (20^*)_4, (30^*)_4, (120^*)_4)$$

- $X_2 + X_2 = (20^*)_4 + (20^*)_4 = (10^+)_4 \cup (20^*20^*)_4$

Using positional notation

- Numbers in base-k notation: strings over $\Sigma_k = \{0, 1, \dots, k-1\}$.
- Set of numbers $\leftrightarrow$ formal language over Σ_k

Example (Jeż, DLT 2007)

$$\begin{aligned}
X_1 &= (X_2 + X_2 \cap X_1 + X_3) \cup \{1\} \\
X_2 &= (X_{12} + X_2 \cap X_1 + X_1) \cup \{2\} \\
X_3 &= (X_{12} + X_{12} \cap X_1 + X_2) \cup \{3\} \\
X_{12} &= X_3 + X_3 \cap X_1 + X_2
\end{aligned}$$

$$((10^*)_4, (20^*)_4, (30^*)_4, (120^*)_4)$$

- $X_2 + X_2 = (20^*)_4 + (20^*)_4 = (10^+)_4 \cup (20^*20^*)_4$
- $X_1 + X_3 = (10^*)_4 + (30^*)_4 =$

Using positional notation

- Numbers in base-k notation: strings over $\Sigma_k = \{0, 1, \dots, k-1\}$.
- Set of numbers $\leftrightarrow$ formal language over Σ_k

Example (Jeż, DLT 2007)

$$X_1 = (X_2 + X_2 \cap X_1 + X_3) \cup \{1\}$$
$$X_2 = (X_{12} + X_2 \cap X_1 + X_1) \cup \{2\}$$
$$X_3 = (X_{12} + X_{12} \cap X_1 + X_2) \cup \{3\}$$
$$X_{12} = X_3 + X_3 \cap X_1 + X_2$$

$$((10^*)_4, (20^*)_4, (30^*)_4, (120^*)_4)$$

- $X_2 + X_2 = (20^*)_4 + (20^*)_4 = (10^+)_4 \cup (20^*20^*)_4$
- $X_1 + X_3 = (10^*)_4 + (30^*)_4 = (10^+)_4 \cup$

Using positional notation

- Numbers in base-k notation: strings over $\Sigma_k = \{0, 1, \ldots, k-1\}$.
- Set of numbers $\leftrightarrow$ formal language over Σ_k

Example (Jeż, DLT 2007)

$$X_1 = (X_2 + X_2 \cap X_1 + X_3) \cup \{1\}$$
$$X_2 = (X_{12} + X_2 \cap X_1 + X_1) \cup \{2\}$$
$$X_3 = (X_{12} + X_{12} \cap X_1 + X_2) \cup \{3\}$$
$$X_{12} = X_3 + X_3 \cap X_1 + X_2$$

$$((10^*)_4, (20^*)_4, (30^*)_4, (120^*)_4)$$

- $X_2 + X_2 = (20^*)_4 + (20^*)_4 = (10^+)_4 \cup (20^*20^*)_4$
- $X_1 + X_3 = (10^*)_4 + (30^*)_4 = (10^+)_4 \cup (10^*30^*)_4 \cup$

Using positional notation

- Numbers in base-k notation: strings over $\Sigma_k = \{0, 1, \ldots, k-1\}$.
- Set of numbers $\leftrightarrow$ formal language over Σ_k

Example (Jeż, DLT 2007)

$$\begin{aligned} X_1 &= (X_2 + X_2 \cap X_1 + X_3) \cup \{1\} \\ X_2 &= (X_{12} + X_2 \cap X_1 + X_1) \cup \{2\} \\ X_3 &= (X_{12} + X_{12} \cap X_1 + X_2) \cup \{3\} \\ X_{12} &= X_3 + X_3 \cap X_1 + X_2 \end{aligned}$$

$$((10^*)_4, (20^*)_4, (30^*)_4, (120^*)_4)$$

- $X_2 + X_2 = (20^*)_4 + (20^*)_4 = (10^+)_4 \cup (20^*20^*)_4$
- $X_1 + X_3 = (10^*)_4 + (30^*)_4 = (10^+)_4 \cup (10^*30^*)_4 \cup (30^*10^*)_4$

Using positional notation

- Numbers in base-k notation: strings over $\Sigma_k = \{0, 1, \ldots, k-1\}$.
- Set of numbers $\leftrightarrow$ formal language over Σ_k

Example (Jeż, DLT 2007)

$$X_1 = (X_2 + X_2 \cap X_1 + X_3) \cup \{1\}$$
$$X_2 = (X_{12} + X_2 \cap X_1 + X_1) \cup \{2\}$$
$$X_3 = (X_{12} + X_{12} \cap X_1 + X_2) \cup \{3\}$$
$$X_{12} = X_3 + X_3 \cap X_1 + X_2$$

$$((10^*)_4, (20^*)_4, (30^*)_4, (120^*)_4)$$

- $X_2 + X_2 = (20^*)_4 + (20^*)_4 = (10^+)_4 \cup (20^*20^*)_4$
- $X_1 + X_3 = (10^*)_4 + (30^*)_4 = (10^+)_4 \cup (10^*30^*)_4 \cup (30^*10^*)_4$
- $(X_2 + X_2) \cap (X_1 + X_3) =$

Using positional notation

- Numbers in base-k notation: strings over $\Sigma_k = \{0, 1, \dots, k-1\}$.
- Set of numbers $\leftrightarrow$ formal language over Σ_k

Example (Jeż, DLT 2007)

$$X_1 = (X_2 + X_2 \cap X_1 + X_3) \cup \{1\}$$
$$X_2 = (X_{12} + X_2 \cap X_1 + X_1) \cup \{2\}$$
$$X_3 = (X_{12} + X_{12} \cap X_1 + X_2) \cup \{3\}$$
$$X_{12} = X_3 + X_3 \cap X_1 + X_2$$

$$((10^*)_4, (20^*)_4, (30^*)_4, (120^*)_4)$$

- $X_2 + X_2 = (20^*)_4 + (20^*)_4 = (10^+)_4 \cup (20^*20^*)_4$
- $X_1 + X_3 = (10^*)_4 + (30^*)_4 = (10^+)_4 \cup (10^*30^*)_4 \cup (30^*10^*)_4$
- $(X_2 + X_2) \cap (X_1 + X_3) = (10^+)_4$

Automata recognizing positional notation

Theorem (Jeż, Okhotin, CSR 2007)

$\forall$ *trellis automaton* M *over* Σ_k *with* $L(M) \subseteq \Sigma_k^* \setminus 0\Sigma_k^*$,
$\exists$ *a system with* $\{\cup, \cap, +\}$ *representing* $(L(M))_k$.

Automata recognizing positional notation

Theorem (Jeż, Okhotin, CSR 2007)

$\forall$ *trellis automaton* M *over* Σ_k *with* $L(M) \subseteq \Sigma_k^* \setminus 0\Sigma_k^*$,
$\exists$ *a system with* $\{\cup, \cap, +\}$ *representing* $(L(M))_k$.

Definition (Culik, Gruska, Salomaa, 1981)

A trellis automaton is a
$M = (\Sigma, Q, I, \delta, F)$ where:

Automata recognizing positional notation

Theorem (Jeż, Okhotin, CSR 2007)

$\forall$ *trellis automaton* M *over* Σ_k *with* $L(M) \subseteq \Sigma_k^* \setminus 0\Sigma_k^*$,
$\exists$ *a system with* $\{\cup, \cap, +\}$ *representing* $(L(M))_k$.

Definition (Culik, Gruska, Salomaa, 1981)

A trellis automaton is a
$M = (\Sigma, Q, I, \delta, F)$ where:

- Σ: input alphabet;

Automata recognizing positional notation

Theorem (Jeż, Okhotin, CSR 2007)

$\forall$ *trellis automaton* M *over* Σ_k *with* $L(M) \subseteq \Sigma_k^* \setminus 0\Sigma_k^*$,
$\exists$ *a system with* $\{\cup, \cap, +\}$ *representing* $(L(M))_k$.

Definition (Culik, Gruska, Salomaa, 1981)

A trellis automaton is a
$M = (\Sigma, Q, I, \delta, F)$ where:

- Σ: input alphabet;
- Q: finite set of states;

Automata recognizing positional notation

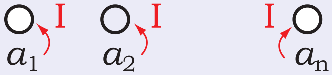
Theorem (Jeż, Okhotin, CSR 2007)

$\forall$ trellis automaton M over Σ_k with $L(M) \subseteq \Sigma_k^* \setminus 0\Sigma_k^*$,
$\exists$ a system with $\{\cup, \cap, +\}$ representing $(L(M))_k$.

Definition (Culik, Gruska, Salomaa, 1981)

A trellis automaton is a
$M = (\Sigma, Q, I, \delta, F)$ where:

- Σ: input alphabet;
- Q: finite set of states;
- $I : \Sigma \to Q$ sets initial states;



Automata recognizing positional notation

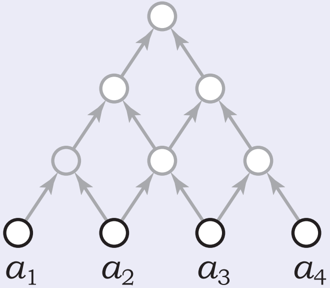
Theorem (Jeż, Okhotin, CSR 2007)

$\forall$ *trellis automaton* M *over* Σ_k *with* $L(M) \subseteq \Sigma_k^* \setminus 0\Sigma_k^*$,
$\exists$ *a system with* $\{\cup, \cap, +\}$ *representing* $(L(M))_k$.

Definition (Culik, Gruska, Salomaa, 1981)

A trellis automaton is a $M = (\Sigma, Q, I, \delta, F)$ where:

- Σ: input alphabet;
- Q: finite set of states;
- $I : \Sigma \to Q$ sets initial states;



Automata recognizing positional notation

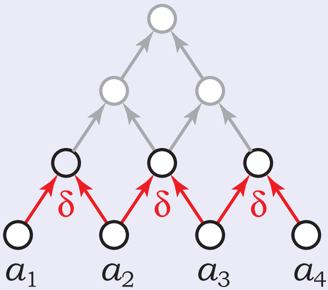
Theorem (Jeż, Okhotin, CSR 2007)

$\forall$ *trellis automaton* M *over* Σ_k *with* $L(M) \subseteq \Sigma_k^* \setminus 0\Sigma_k^*$,
$\exists$ *a system with* $\{\cup, \cap, +\}$ *representing* $(L(M))_k$.

Definition (Culik, Gruska, Salomaa, 1981)

A trellis automaton is a $M = (\Sigma, Q, I, \delta, F)$ where:

- Σ: input alphabet;
- Q: finite set of states;
- $I : \Sigma \to Q$ sets initial states;
- $\delta : Q \times Q \to Q$, transition function;



Automata recognizing positional notation

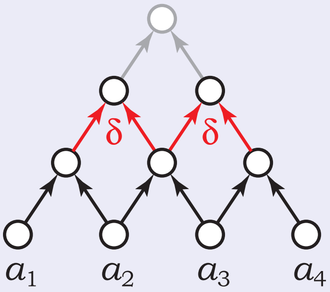
Theorem (Jeż, Okhotin, CSR 2007)

$\forall$ *trellis automaton* M *over* Σ_k *with* $L(M) \subseteq \Sigma_k^* \setminus 0\Sigma_k^*$,
$\exists$ *a system with* $\{\cup, \cap, +\}$ *representing* $(L(M))_k$.

Definition (Culik, Gruska, Salomaa, 1981)

A trellis automaton is a
$M = (\Sigma, Q, I, \delta, F)$ where:

- Σ: input alphabet;
- Q: finite set of states;
- $I : \Sigma \to Q$ sets initial states;
- $\delta : Q \times Q \to Q$, transition function;



Automata recognizing positional notation

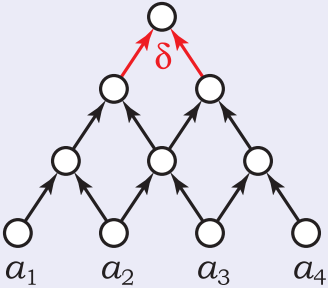
Theorem (Jeż, Okhotin, CSR 2007)

$\forall$ trellis automaton M over Σ_k with $L(M) \subseteq \Sigma_k^* \setminus 0\Sigma_k^*$,
$\exists$ a system with $\{\cup, \cap, +\}$ representing $(L(M))_k$.

Definition (Culik, Gruska, Salomaa, 1981)

A trellis automaton is a $M = (\Sigma, Q, I, \delta, F)$ where:

- Σ: input alphabet;
- Q: finite set of states;
- $I : \Sigma \to Q$ sets initial states;
- $\delta : Q \times Q \to Q$, transition function;



Automata recognizing positional notation

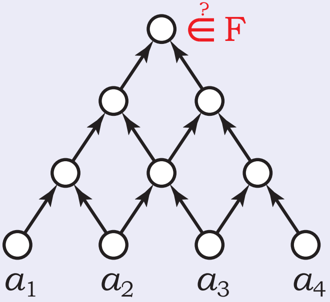
Theorem (Jeż, Okhotin, CSR 2007)

$\forall$ *trellis automaton M over Σ_k with $L(M) \subseteq \Sigma_k^* \setminus 0\Sigma_k^*$,*
$\exists$ *a system with $\{\cup, \cap, +\}$ representing $(L(M))_k$.*

Definition (Culik, Gruska, Salomaa, 1981)

A trellis automaton is a $M = (\Sigma, Q, I, \delta, F)$ where:

- Σ: input alphabet;
- Q: finite set of states;
- $I : \Sigma \to Q$ sets initial states;
- $\delta : Q \times Q \to Q$, transition function;
- $F \subseteq Q$: accepting states.



Automata recognizing positional notation

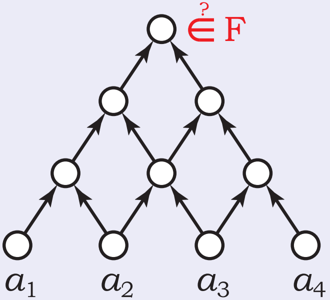
Theorem (Jeż, Okhotin, CSR 2007)

$\forall$ *trellis automaton* M *over* Σ_k *with* $L(M) \subseteq \Sigma_k^* \setminus 0\Sigma_k^*$,
$\exists$ *a system with* $\{\cup, \cap, +\}$ *representing* $(L(M))_k$.

Definition (Culik, Gruska, Salomaa, 1981)

A trellis automaton is a $M = (\Sigma, Q, I, \delta, F)$ where:

- Σ: input alphabet;
- Q: finite set of states;
- $I : \Sigma \to Q$ sets initial states;
- $\delta : Q \times Q \to Q$, transition function;
- $F \subseteq Q$: accepting states.



- Can recognize $\{wcw\}$, $\{a^n b^n c^n\}$, $\{a^n b^{2^n}\}$, VALC.

Outline of the construction

- Turing machine (Turing, 1936)

Outline of the construction

- Turing machine (Turing, 1936)
- VALC(T): intersection of two LinCFLs (Hartmanis, 1967; Baker, Book, 1978)

$$\mathrm{VALC}(T) = \{C_T(w)\natural w \mid w \in L(T)\}$$
$$C_T(w) = q_0 w \# u_1 q_1 a_1 v_1 \# \dots \# u_\ell q_\ell a_\ell v_\ell$$

Outline of the construction

- Turing machine (Turing, 1936)
- VALC(T): intersection of two LinCFLs (Hartmanis, 1967; Baker, Book, 1978)

$$\mathrm{VALC}(T) = \{C_T(w)\natural w \mid w \in L(T)\}$$
$$C_T(w) = q_0 w \# u_1 q_1 a_1 v_1 \# \dots \# u_\ell q_\ell a_\ell v_\ell$$

- Trellis automata (1970s–80s)

Outline of the construction

- Turing machine (Turing, 1936)
- VALC(T): intersection of two LinCFLs (Hartmanis, 1967; Baker, Book, 1978)

$$\mathrm{VALC}(T) = \{ C_T(w) \natural w \mid w \in L(T) \}$$
$$C_T(w) = q_0 w \# u_1 q_1 a_1 v_1 \# \ldots \# u_\ell q_\ell a_\ell v_\ell$$

- Trellis automata (1970s–80s)
- Extracting $L(T)$ from VALC(T) (Okhotin, ICALP 2003)

Outline of the construction

- Turing machine (Turing, 1936)
- VALC(T): intersection of two LinCFLs (Hartmanis, 1967; Baker, Book, 1978)

$$\mathrm{VALC}(T) = \{C_T(w)\natural w \mid w \in L(T)\}$$
$$C_T(w) = q_0 w \# u_1 q_1 a_1 v_1 \# \ldots \# u_\ell q_\ell a_\ell v_\ell$$

- Trellis automata (1970s–80s)
- Extracting $L(T)$ from VALC(T) (Okhotin, ICALP 2003)
- Trellis automata → equations over sets of numbers (Jeż, Okhotin, CSR 2007)

Outline of the construction

- Turing machine (Turing, 1936)
- VALC(T): intersection of two LinCFLs (Hartmanis, 1967; Baker, Book, 1978)

$$\mathrm{VALC}(T) = \{C_T(w)\natural w \mid w \in L(T)\}$$
$$C_T(w) = q_0 w \# u_1 q_1 a_1 v_1 \# \dots \# u_\ell q_\ell a_\ell v_\ell$$

- Trellis automata (1970s–80s)
- Extracting $L(T)$ from VALC(T) (Okhotin, ICALP 2003)
- Trellis automata → equations over sets of numbers (Jeż, Okhotin, CSR 2007)
- Extracting numbers with notation $L(T)$ from numbers with notation VALC(T)

Arithmetization of Turing machines

- R: recursive set of numbers recognized by TM T.

Arithmetization of Turing machines

- R: recursive set of numbers recognized by TM T.
- $\Sigma_6 = \{0, 1, 2, 3, 4, 5\}$, using base-6 notation.

Arithmetization of Turing machines

- R: recursive set of numbers recognized by TM T.
- $\Sigma_6 = \{0, 1, 2, 3, 4, 5\}$, using base-6 notation.
- Computation of T on numbers $(1w)_6$.

Arithmetization of Turing machines

- R: recursive set of numbers recognized by TM T.
- $\Sigma_6 = \{0, 1, 2, 3, 4, 5\}$, using base-6 notation.
- Computation of T on numbers $(1w)_6$.
- ... encoded by $C_T^1 \in \{30, 300\}^*$.

Arithmetization of Turing machines

- R: recursive set of numbers recognized by TM T.
- $\Sigma_6 = \{0, 1, 2, 3, 4, 5\}$, using base-6 notation.
- Computation of T on numbers $(1w)_6$.
- ... encoded by $C_T^1 \in \{30, 300\}^*$.
- $\text{VALC}_1(T) = \{C_T^1(iw)1w \mid 1w \in L(T)\}$.

$$(\underbrace{30300300\ldots30300}_{C_T^1(123450)}123450)_6 \in \text{VALC}_1(T)$$

Arithmetization of Turing machines

- R: recursive set of numbers recognized by TM T.
- $\Sigma_6 = \{0, 1, 2, 3, 4, 5\}$, using base-6 notation.
- Computation of T on numbers $(1w)_6$.
- ... encoded by $C_T^1 \in \{30, 300\}^*$.
- $\mathrm{VALC}_1(T) = \{C_T^1(iw)1w \mid 1w \in L(T)\}$.

$$(\underbrace{30300300\ldots30300}_{C_T^1(123450)}123450)_6 \in \mathrm{VALC}_1(T)$$

- As formal language: recognized by trellis automaton.

Arithmetization of Turing machines

- R: recursive set of numbers recognized by TM T.
- $\Sigma_6 = \{0, 1, 2, 3, 4, 5\}$, using base-6 notation.
- Computation of T on numbers $(1w)_6$.
- ... encoded by $C_T^1 \in \{30, 300\}^*$.
- $\text{VALC}_1(T) = \{C_T^1(iw)1w \mid 1w \in L(T)\}$.

$$(\underbrace{30300300\ldots30300}_{C_T^1(123450)}123450)_6 \in \text{VALC}_1(T)$$

- As formal language: recognized by trellis automaton.
- As set of numbers: given by equations.

Constructing the equations

- For TM T recognizing L:

$$\mathrm{VALC}_1(T) = \{\underbrace{C_T^1(iw)}_{\{30,300\}^*} 1w \mid 1w \in L(T)\}$$

Constructing the equations

- For TM T recognizing L:

$$\mathrm{VALC}_1(T) = \{\underbrace{C_T^1(iw)}_{\{30,300\}^*} 1w \mid 1w \in L(T)\}$$

- Two equations:

$$\begin{aligned} Y_1 &\subseteq (1\Sigma_6^+)_6 \\ \mathrm{VALC}_1(T) &\subseteq (\{30, 300\}^* 3000^*)_6 + Y_1 \end{aligned}$$

Constructing the equations

- For TM T recognizing L:

$$\mathrm{VALC}_1(T) = \{\underbrace{C_T^1(iw)}_{\{30,300\}^*}1w \mid 1w \in L(T)\}$$

- Two equations:

$$\begin{aligned} Y_1 &\subseteq (1\Sigma_6^+)_6 \\ \mathrm{VALC}_1(T) &\subseteq (\{30, 300\}^* 3000^*)_6 + Y_1 \end{aligned}$$

- Equivalent to:

$$\{(1w)_6 \mid (1w)_6 \in L(T)\} \subseteq Y_1 \subseteq (1\Sigma_6^+)_6$$

Constructing the equations

- For TM T recognizing L:

$$\mathrm{VALC}_1(T) = \{\underbrace{C_T^1(iw)}_{\{30,300\}^*} 1w \mid 1w \in L(T)\}$$

- Two equations:

$$Y_1 \subseteq (1\Sigma_6^+)_6$$
$$\mathrm{VALC}_1(T) \subseteq (\{30, 300\}^* 3000^*)_6 + Y_1$$

- Equivalent to:

$$\{(1w)_6 \mid (1w)_6 \in L(T)\} \subseteq Y_1 \subseteq (1\Sigma_6^+)_6$$

- Least solution: $Y_1 = \{1w \mid 1w \in L(T)\}$.

Constructing the equations

- For TM T recognizing L:

$$\text{VALC}_1(T) = \{\underbrace{C_T^1(iw)}_{\{30,300\}^*} 1w \mid 1w \in L(T)\}$$

- Two equations:

$$Y_1 \subseteq (1\Sigma_6^+)_6$$
$$\text{VALC}_1(T) \subseteq (\{30, 300\}^* 3000^*)_6 + Y_1$$

- Equivalent to:

$$\{(1w)_6 \mid (1w)_6 \in L(T)\} \subseteq Y_1 \subseteq (1\Sigma_6^+)_6$$

- Least solution: $Y_1 = \{1w \mid 1w \in L(T)\}$.
- Equation with greatest solution $Z_1 = \{1w \mid 1w \in L(T)\}$.

Results for unresolved equations with $\{\cup, +\}$ or $\{\cap, +\}$

$$\begin{cases} \varphi_1(X_1, \ldots, X_n) & = & \psi_1(X_1, \ldots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \ldots, X_n) & = & \psi_m(X_1, \ldots, X_n) \end{cases}$$

Results for unresolved equations with $\{\cup, +\}$ or $\{\cap, +\}$

$$\begin{cases} \varphi_1(X_1, \ldots, X_n) & = & \psi_1(X_1, \ldots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \ldots, X_n) & = & \psi_m(X_1, \ldots, X_n) \end{cases}$$

Theorem

$S \subseteq \mathbb{N}_0$ is given by unique/least/greatest solution of such a system
if and only if
S is recursive/r.e./co-r.e.

Results for unresolved equations with $\{\cup, +\}$ or $\{\cap, +\}$

$$\begin{cases} \varphi_1(X_1, \ldots, X_n) & = & \psi_1(X_1, \ldots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \ldots, X_n) & = & \psi_m(X_1, \ldots, X_n) \end{cases}$$

Theorem

$S \subseteq \mathbb{N}_0$ is given by unique/least/greatest solution of such a system
if and only if
S is recursive/r.e./co-r.e.

Theorem

Decision problems are undecidable, namely:

Results for unresolved equations with $\{\cup, +\}$ or $\{\cap, +\}$

$$\begin{cases} \varphi_1(X_1, \ldots, X_n) & = & \psi_1(X_1, \ldots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \ldots, X_n) & = & \psi_m(X_1, \ldots, X_n) \end{cases}$$

Theorem

$S \subseteq \mathbb{N}_0$ is given by unique/least/greatest solution of such a system
if and only if
S is recursive/r.e./co-r.e.

Theorem

Decision problems are undecidable, namely:

- *"Exists a solution?":* Π_1-*complete.*

Results for unresolved equations with $\{\cup, +\}$ or $\{\cap, +\}$

$$\begin{cases} \varphi_1(X_1, \dots, X_n) & = & \psi_1(X_1, \dots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \dots, X_n) & = & \psi_m(X_1, \dots, X_n) \end{cases}$$

Theorem

$S \subseteq \mathbb{N}_0$ is given by unique/least/greatest solution of such a system
if and only if
S is recursive/r.e./co-r.e.

Theorem

Decision problems are undecidable, namely:

- *"Exists a solution?":* Π_1-*complete.*
- *"Exists a unique solution?":* Π_2-*complete.*

Results for unresolved equations with $\{\cup, +\}$ or $\{\cap, +\}$

$$\begin{cases} \varphi_1(X_1, \dots, X_n) & = & \psi_1(X_1, \dots, X_n) \\ & \vdots & \\ \varphi_m(X_1, \dots, X_n) & = & \psi_m(X_1, \dots, X_n) \end{cases}$$

Theorem

$S \subseteq \mathbb{N}_0$ is given by unique/least/greatest solution of such a system
if and only if
S is recursive/r.e./co-r.e.

Theorem

Decision problems are undecidable, namely:

- *"Exists a solution?":* Π_1-*complete.*
- *"Exists a unique solution?":* Π_2-*complete.*
- *"Exist finitely many solutions?":* Σ_3-*complete.*

Conclusion

- Results on language equations over Σ with $|\Sigma| \geqslant 2$.

Conclusion

- Results on language equations over Σ with $|\Sigma| \geqslant 2$.
- ... extended to $\Sigma = \{a\}$.

Conclusion

- Results on language equations over Σ with $|\Sigma| \geqslant 2$.
- ... extended to $\Sigma = \{a\}$.
- cf. Diophantine equations.

Conclusion

- Results on language equations over Σ with $|\Sigma| \geqslant 2$.
- ... extended to $\Sigma = \{a\}$.
- cf. Diophantine equations.

Example

Let PRIMES be the set of all primes.

Conclusion

- Results on language equations over Σ with $|\Sigma| \geqslant 2$.
- ... extended to $\Sigma = \{a\}$.
- cf. Diophantine equations.

Example

Let PRIMES be the set of all primes.

1. A Diophantine equation with PRIMES as the range of x.

Conclusion

- Results on language equations over Σ with $|\Sigma| \geqslant 2$.
- ... extended to $\Sigma = \{a\}$.
- cf. Diophantine equations.

Example

Let PRIMES be the set of all primes.

1. A Diophantine equation with PRIMES as the range of x.
2. A system over sets of numbers with PRIMES as the unique value of X.

Conclusion

- Results on language equations over Σ with $|\Sigma| \geqslant 2$.
- ... extended to $\Sigma = \{a\}$.
- cf. Diophantine equations.

Example

Let PRIMES be the set of all primes.

1. A Diophantine equation with PRIMES as the range of x.
2. A system over sets of numbers with PRIMES as the unique value of X.

Problem

Construct any simple system using $\{\cup, +\}$ with a non-periodic solution.