Language Equations

Michal Kunc
Masaryk University Brno
First Something Different — Word Equations

- operation: concatenation
- constants: letters
- variables stand for words
- for instance, solutions of equation $xba = abx$ are exactly $x = a(ba)^n$, where $n \in \mathbb{N}_0$
- PSPACE algorithm deciding satisfiability, EXPTIME algorithm finding all solutions (Makanin 1977, Plandowski 2006)
- satisfiability-equivalent to language equations with singleton constants and concatenation as the only operation:
  - shortlex-minimal words of an arbitrary language solution form a word solution
Overview

• General properties.
• Equations with one-sided concatenation.
• Explicit systems of equations. (basic models of computation, semantics of grammars)
• Inequalities with constant sides.
• General implicit equations. (surprising computational completeness)
Language Equations — Basic Elements

set of variables $\mathcal{V} = \{X_1, \ldots, X_n\}$

finite alphabet $A = \{a, b, \ldots\}$

$A^*$ ... the monoid of finite words over $A$ with the operation of concatenation

$L \subseteq A^*$ ... language over $A$

$\wp(A^*)$ ... the set of all languages over $A$

operations: usually extended from operations $A^* \times A^* \to \wp(A^*)$ defined on words

concatenation: $u \cdot v = \{uv\}$ \quad \(K \cdot L = \{uv \mid u \in K, v \in L\}\)

union: $u \cup v = \{u, v\}$

intersection: $u \cap v = \begin{cases} \{u\} & \text{if } u = v \\ \emptyset & \text{if } u \neq v \end{cases}$

shuffle: $u \bigshuffle v = \{u_1v_1 \ldots u_kv_k \mid u_1 \ldots u_k = u, v_1 \ldots v_k = v\}$

all such $n$-ary operations $f$ are monotone:

\[ K_1 \subseteq L_1 \& \ldots \& K_n \subseteq L_n \implies f(K_1, \ldots, K_n) \subseteq f(L_1, \ldots, L_n) \]
Language Equations — Definition

\[ \varphi(X_1, \ldots, X_n) = \psi(X_1, \ldots, X_n) \]

\( \varphi, \psi \) … expressions using variables, constant languages and language operations

solutions: \((L_1, \ldots, L_n) \in \wp(A^*)^n\) such that \(\varphi(L_1, \ldots, L_n) = \psi(L_1, \ldots, L_n)\)

ordering of solutions by componentwise inclusion:

\((K_1, \ldots, K_n) \leq (L_1, \ldots, L_n) \iff K_1 \subseteq L_1, \ldots, K_n \subseteq L_n\)
The Basic Property

\[ f: \wp(A^*)^n \rightarrow \wp(A^*) \text{ continuous:} \]
\[ \forall \ell \in \mathbb{N} \exists m \in \mathbb{N} \forall K_1, \ldots, K_n, L_1, \ldots, L_n \subseteq A^*: \]
\[ K_i \cap A^{\leq m} = L_i \cap A^{\leq m} \implies f(K_1, \ldots, K_n) \cap A^{\leq \ell} = f(L_1, \ldots, L_n) \cap A^{\leq \ell} \]

**Continuous operations:** Boolean operations, concatenation, shuffle, 

**Non-continuous operations:** Typically erasing operations, e.g. erasing homomorphisms

**Proposition:** If all operations are continuous, then every solution is contained in a maximal solution and contains a minimal solution.

\[ \rightsquigarrow \text{describing languages as largest and smallest solutions of systems of equations} \]

**Main questions to study:**
- expressive power, properties of solutions
- decidability of existence and uniqueness of solutions
- algorithms for finding minimal and maximal solutions
Equations with One-Sided Concatenation
One-Sided Concatenation — Explicit Systems

Example:

\[
X_1 = \{\varepsilon\} \cup X_2 \cdot a \quad X_2 = X_1 \cdot b \cup X_2 \cdot a
\]

regular languages = components of smallest (largest, unique) solutions of explicit systems

\[
X_i = K_i \cup \bigcup_{j=1}^{n} X_j \cdot L_{j,i} \quad i = 1, \ldots, n
\]

of left-linear equations with finite constants \(K_i\) and \(L_{j,i}\)

Systems correspond to non-deterministic automata with arcs labelled with constant languages.

In general: Components of smallest solutions are rational combinations of constant languages.

Additionally intersection allowed: alternating finite automata.
One-Sided Concatenation — Implicit Systems

Inequalities with one-sided concatenation, Boolean operations and regular constants:

basic properties can be expressed using formulae of monadic second-order theory
of the infinite $|A|$-ary tree

Example: $\{b\} \cup X a \subseteq X \cup X ba$

$$X \text{ is a solution } \iff X(b) \land (\forall x: X(x) \implies (X(xa) \lor \exists y: X(y) \land x = yb))$$

$$X \text{ minimal } \iff \forall Y: (Y \text{ is a solution } \land \forall x: Y(x) \implies X(x)) \implies$$

$$\implies (\forall x: X(x) \implies Y(x))$$

minimal solutions: $\bullet = \text{"X holds"}$ $\circ = \text{"X does not hold"}$

$$a^* \cup b :$$

$$a^* \cup b :$$

Rabin 1969 $\implies$ algorithmically solvable using tree automata
One-Sided Concatenation — Complexity of Decision Problems

Inequalities with one-sided concatenation, Boolean operations and regular constants:

basic decision problems are EXPTIME-complete

(Aiken & Kozen & Vardi & Wimmers 1994,

The set of all solutions represented by an NFA $A = (Q, I, F, \delta)$ computable in EXPTIME:

- $r: A^* \rightarrow Q$ run of $A$: $r(\varepsilon) \in I$, $(r(w), a, r(wa)) \in \delta$
- solutions are exactly languages $L(r) = \{ w \in A^* \mid r(w) \in F \}$
One-Sided Concatenation — Non-regular Constants

\[ K_0 \cup X_1 K_1 \cup \cdots \cup X_n K_n \subseteq L_0 \cup X_1 L_1 \cup \cdots \cup X_n L_n \]
\( K_j \) arbitrary, \( L_j \) regular

largest solution:

- regular
- for context-free \( K_j \): algorithmically regular
- direct construction of the automaton accepting the solution

(MK 2005)
Explicit Systems of Equations
Explicit Systems of Equations

\[ X_1 = \varphi_1(X_1, \ldots, X_n) \]

\[ \vdots \]

\[ X_n = \varphi_n(X_1, \ldots, X_n) \]

notation:

\[ X = (X_1, \ldots, X_n), \quad \varphi = (\varphi_1, \ldots, \varphi_n) \]

system of equations \( X = \varphi(X) \)

\( \varphi_i \) monotone and continuous \( \implies \) system possesses the least and the greatest solution

\[ \lim_{k \to \infty} \varphi^k(\emptyset, \ldots, \emptyset) \quad \lim_{k \to \infty} \varphi^k(A^*, \ldots, A^*) \]
Concatenation and Union — Context-Free Languages

Example: Dyck language of correct bracketings over \( A = \{ (, ) \} \):

- context-free grammar:
  \[ X_1 \to \varepsilon | X_2 X_1 \quad X_2 \to (X_1) \]

- system of language equations:
  \[ X_1 = \{ \varepsilon \} \cup X_2 \cdot X_1 \quad X_2 = \{ (\} \cdot X_1 \cdot \{) \} \]

Ginsburg & Rice 1962:
context-free languages = components of smallest (largest, unique) solutions of explicit systems
\[ X_i = S_{i,1} \cup \cdots \cup S_{i,k_i} \quad i = 1, \ldots, n \]

of polynomial equations with \( S_{i,j} \in (A \cup V)^* \)
Concatenation, Union and Intersection — Conjunctive Languages

Okhotin 2001–today:

• analogy of alternating machines for context-free grammars
• we can specify that a word satisfies certain syntactic conditions simultaneously
• parsing using standard techniques
• $\subseteq \text{DTIME}(n^3) \cap \text{DSPACE}(n)$
Linear Concatenation, Union and Intersection

$X_i = \varphi_i$  \hspace{1em} $\varphi_i$ constructed from elements of $A^*$ and $A^* \cup A^*$ using union and intersection

Okhotin 2004:
systems define exactly languages accepted by one-way real-time cellular automata

Examples:

\{ wcw \mid w \in \{a, b\}^* \}, \{ a^n b^n c^n \mid n \in \mathbb{N} \}, all computations of a Turing machine
Conjunctive Languages over Unary Alphabet

alphabet \( A = \{a\} \)

Language \( L \subseteq \{a\}^\ast \) represents the set \( \{ k \mid a^k \in L \} \) of non-negative integers.

concatenation = elementwise addition

Context-free unary languages are regular, i.e. ultimately periodic.

Systems of equations with addition, union and intersection:

- allow manipulating integers in positional notation
  e.g. binary notation of \( \{ a^{2^n} \mid n \in \mathbb{N} \} \) is regular \( 10^\ast \)
- smallest solutions are (as sets of numbers) in EXPTIME and can be EXPTIME-complete (Jeż & Okhotin 2008)
- unary notation of any linear conjunctive language can be represented (Jeż & Okhotin 2010)
  (in particular, unary representation of valid computations of a Turing machine)
Explicit Systems with Concatenation and All Boolean Operations

In general, powerful enough to express implicit equations $\implies$ computationally universal.

Boolean grammars (Okhotin 2004–2007):
- semantics defined only for some systems
- generalization of conjunctive languages
- standard parsing techniques still available
- used to give a formal specification of a simple programming language

Equations with concatenation and any clone of Boolean operations:
- Okhotin 2007: exactly seven classes of languages

Largest and smallest solutions w.r.t. lexicographical ordering:
- Okhotin 2005: number of variables corresponds to the levels of arithmetical hierarchy
Equations with Constant Sides
Inequalities with Constant Sides — Examples

Minimal deterministic automaton of a language $L$:

state reached by $w \in A^* =$ largest solution of the inequality $w \cdot X_w \subseteq L$

$X_w \xrightarrow{a} X_{wa}$

initial state $X_\varepsilon$

final states $X_w$, where $w \in L$

Universal automaton of a language $L$

$= \text{smallest non-deterministic automaton admitting morphism from every automaton accepting } L$

state $= \text{maximal solution of the inequality } X \cdot Y \subseteq L$

$(X, Y) \xrightarrow{a} (X', Y') \iff aY' \subseteq Y \iff Xa \subseteq X'$

$(X, Y)$ initial state $\iff \varepsilon \in X$

$(X, Y)$ final state $\iff \varepsilon \in Y$
Systems of Inequalities with Constant Sides — General Results

\[ \bigcup P_i \subseteq L_i \qquad L_i \subseteq A^* \text{ regular constant, } P_i \subseteq (A \cup \mathcal{V})^* \text{ arbitrary} \]

maximal solutions:

- finitely many, all of them regular
- for context-free expressions \( \bigcup P_i \): algorithmically regular
- \( \sigma : A^* \rightarrow M \) homomorphism recognizing all languages \( L_i \)
  
  \[ (\text{i.e. } L_i = \sigma^{-1}(F_i) \text{ for some } F_i \subseteq M) \]

  \[ \implies \sigma \text{ recognizes all components of maximal solutions} \]

(Conway 1971)

Systems of equations with constant sides:

\[ \varphi_i(X_1, \ldots, X_n) = L_i \qquad L_i \subseteq A^* \text{ regular constant, } \varphi_i \text{ regular expression} \]

- satisfiability by arbitrary (finite) languages is EXPSPACE-complete (Bala 2006)
- Is satisfiability decidable if \( \varphi_i \) can contain intersection?
Implicit Equations
Does $\varphi(L_1, \ldots, L_n) = \psi(L_1, \ldots, L_n)$ hold for arbitrary (regular) languages $L_1, \ldots, L_n$?

- trivially *decidable* with union, concatenation, Kleene iteration and regular constants:
  treat variables as letters and compare regular languages
- decidable also with the shuffle operation (Meyer & Rabinovich 2002)
- open problems for expressions with intersection
Implicit Equations — Undecidability of Solvability

Equations with finite constants, union and concatenation:

Context-free languages $X$ and $Y$ defined by explicit systems.
Add equation $X = Y$ to test for equivalence.

Systems of equations with regular constants and concatenation (MK 2007):

$XXK = LX, A^*X = A^*$
$K, L \subseteq A^*$ regular

(conjugacy via languages containing the empty word)
Implicit Equations — Computational Universality

Components of unique (smallest, largest) solutions =
= recursive (recursively enumerable, co-recursively enumerable) languages.

Universality of simple systems of equations:

**Unary alphabet, concatenation, union and finite constants (Jeż & Okhotin 2008):**
- Computations of a Turing machine encoded in the unary notation in a very special way and the accepted language extracted using language equations.

**Unary alphabet, concatenation and regular constants (Jeż & Okhotin 2009):**
- Encoding of languages, which allows using concatenation to compute both concatenation and union
- **Lehtinen & Okhotin 2009:** $XXK = XXL, XM = N$, $K, L$ finite, $M, N$ regular

**Two-letter alphabet, concatenation and finite constants (MK 2007):**
- $XL = LX$, with $L$ finite

$\rightsquigarrow$ All basic decision problems are undecidable for very simple equations.
Commutation — Example of Computational Universality

Every co-recursively enumerable language can be encoded into the largest solution of a system of any of the following forms, with regular constants $K$, $L$, $M$ and $N$:  

\[
\begin{align*}
XK & \subseteq LX, \ X \subseteq M \\
XK & \subseteq LX, \ XM \subseteq NX \\
XL & = LX, \text{ with } L \text{ finite}
\end{align*}
\]

**Game** corresponding to equation $XL = LX$:

- **position**: $w \in A^*$
- **attacker**: chooses $u \in L$
  - plays either $w \rightarrow wu$ or $w \rightarrow uw$
- **defender**: chooses $v \in L$ so that $wu = v\tilde{w}, uw = \tilde{w}v$, respectively
  - plays $wu \rightarrow \tilde{w}, uw \rightarrow \tilde{w}$, respectively

largest solution $= \text{all winning positions of the defender}$
Commutation — Example of Non-regular Solution

\[ A = \{a, b, c, e, \hat{e}, f, \hat{f}, g, \hat{g}\} \]

\[ L = \{c, e f, g a, e, f g, \hat{f} \hat{e}, a \hat{g}, \hat{e}, \hat{g} \hat{f}, f g b a \hat{g}\} \cup c M \cup M c \cup \]
\[ \cup A^* b A^* b A^* \cup (A \setminus \{c\})^* b (A \setminus \{c\})^* \setminus N \]

\[ M = e f g a^+ b a^* \cup g a^* b a^* \hat{g} \hat{f} \cup a^* b a^* \hat{g} \hat{f} \hat{e} \cup f g a^* b a^* \hat{g} \]

\[ N = \{e f g, f g, g, \varepsilon\} \cdot a^* b a^* \cdot \{\varepsilon, \hat{g}, \hat{g} \hat{f}, \hat{g} \hat{f} \hat{e}\} \]

encodes simultaneous decrementation of two counters and zero-test

Configuration: \[ [[[e]f]g]a^m b a^n [\hat{g} [\hat{f} [\hat{e}]]] \]
Commutation — Simultaneous Decrementation of Both Counters

Attacker forces defender to remove one $a$ on each side:

\[
e f g a^m b a^n  \\
\downarrow  \\
e f g a^m b a^n \cdot \hat{g} \hat{f} \quad \rightarrow \quad f g a^m b a^n \hat{g} \hat{f}  \\
\downarrow  \\
g a^m b a^n \hat{g} \hat{f}  \\
\downarrow  \\
g a a^{m-1} b a^n \hat{g} \hat{f} \cdot \hat{e}  \\
\downarrow  \\
a^{m-1} b a^n \hat{g} \hat{f} \hat{e}  \\
\downarrow  \\
\vdots  \\
\downarrow  \\
e f g a^{m-1} b a^{n-1}
\]
Commutation — Encoding Games  

Example:

- ● = attacker should play
- ○ = defender should play
- modification on the left
- modification on the right

Position of the game: a node of the graph and a word
Labels of attacker’s nodes: allowed words
Labels of edges: words to be added by attacker or removed by defender
  - when attacker modifies on one side, defender has to modify on the other
  - bipartite graph for each type of edges
  - at most one common node for any two connected components of different types
  - only one type of edges leading from each of attacker’s nodes
  - non-empty labels of edges only around one attacker’s node for each type of edges
Implicit Equations — Rational Infinite Systems of Equations

rational system = defined by a finite transducer

Every rational system of word equations is algorithmically equivalent to its finite subsystem


Do given finite languages form a solution of the system \( \{ X^n Z = Y^n Z \mid n \in \mathbb{N} \} \)?

undecidable (Lisovik 1997, Karhumäki & Lisovik 2003, MK 2007)
Implicit Equations — Tractable Cases

\[ \subseteq \subseteq XLY \ldots \]

We need to classify words according to their decompositions with respect to constant languages on the right.
Well-quasiorders (wqo) — Powerful Tool for Proving Regularity

Quasiorder $\leq$ on $A^*$ is a wqo, if it contains neither infinite descending chains nor infinite antichains.

Equivalent definitions:
• Every upward closed language over $A$ is finitely generated.
• There is no infinite ascending sequence of upward closed languages.

Example: “scattered subword” ordering

Ehrenfeucht & Haussler & Rozenberg 1983:
$L \subseteq A^*$ is regular $\iff$ $L$ is upward closed with respect to a monotone wqo on $A^*$.

Generalizes recognition by finite monoids:
• Congruence of finite index is a monotone wqo.
• upward closed = recognized by the congruence

Applying wqos to language inequalities:
Construct a wqo on $A^*$ such that every solution is contained in an upward closed solution.
Quasiorder Classifying Words According to Their Decompositions

\( \sigma : A^* \to M \) ... homomorphism recognizing constant languages on the right

Definition (Bucher & Ehrenfeucht & Haussler 1985):

\( w \leq_{\sigma} v \iff w = a_1 \cdots a_m, \ a_j \in A, \ v = v_1 \cdots v_m, \ v_j \in A^+, \)

\( \sigma(a_1) = \sigma(v_1), \ldots, \sigma(a_m) = \sigma(v_m) \)

\( \leq_{\sigma} \) is the derivation relation of the rewriting system

\[ \{ a \to v \mid a \in A, \ v \in A^*, \ \sigma(a) = \sigma(v) \} \]

Example: \( \sigma : \{a, b\}^* \to (\{0, 1\}, +) \) (two-element group) \( \sigma(a) = 1, \sigma(b) = 0 \)
Implicit Inequalities with Restrictions on Constants

Theorem: (MK 2005)

\[ \sigma : A^* \to M \] homomorphism

\[ \varphi_i(X_1, \ldots, X_n) \subseteq \psi_i(X_1, \ldots, X_n) \] (infinite) system of inequalities

- all operations monotone
- in \( \varphi_i \) all \( K \)-ary operations \( f : (\varphi(A^*))^K \to \varphi(A^*) \) satisfy:
  \[ f((\langle L_k \rangle_{\leq \sigma})_{k \in K}) \subseteq \langle f((L_k)_{k \in K}) \rangle_{\leq \sigma} \] for all \( L_k \subseteq A^* \)
  \( (\langle L \rangle_{\leq \sigma} \) upward closure)
- in \( \psi_i \) all \( K \)-ary operations \( f : (\varphi(A^*))^K \to \varphi(A^*) \) satisfy:
  \[ f((\langle L_k \rangle_{\leq \sigma})_{k \in K}) \supseteq \langle f((L_k)_{k \in K}) \rangle_{\leq \sigma} \] for all \( L_k \subseteq A^* \)

Then all maximal solutions are recognized by \( \leq_{\sigma} \).

Examples of admissible operations:

- anywhere: concatenation, Kleene iteration, shuffle, (infinitary) union,
  constants recognized by \( \sigma \), constants \( A^{\geq n} \) and \( \{ \varepsilon \} \).
- on the right: (infinitary) intersection.
- on the left: arbitrary constants.
Implicit Inequalities — Regularity of Maximal Solutions  

(MK 2005)

minimal deterministic automata of constant languages do not contain the pattern

\[
\begin{array}{ccc}
\bullet & \xrightarrow{a} & \bullet \\
\bullet & \xrightarrow{b} & \bullet \\
\bullet & \xrightarrow{a} & \bullet \\
\bullet & \xrightarrow{b} & \bullet
\end{array}
\]

\[
\Rightarrow \leq_{\sigma} \text{ is a wqo} \Rightarrow \text{all maximal solutions are regular}
\]

Example: \( L \) admissible constant language \( \Rightarrow \) every union of powers of \( L \) is regular.

(largest solution of the inequality \( X \subseteq \bigcup_{n \in N} L^n \), for \( N \subseteq \mathbb{N} \))

Corollary:

The class of polynomials of group languages is closed under taking maximal solutions of all such systems.
Semi-commutation Inequalities

\[ XK \subseteq LX \quad K \text{ arbitrary, } L \text{ regular} \]

largest solution:
- always regular \((MK 2005)\)
- for context-free \(K\): algorithmically recursive
- if \(K\) and \(L\) finite and all words in \(K\) longer than all in \(L\): algorithmically regular \((Ly 2007)\)

**Game:** position: \(w \in A^*\)

attacker: chooses \(u \in K\)
- plays \(w \rightarrow wu\)

defender: chooses \(v \in L\) so that \(wu = v\tilde{w}\)
- plays \(wu \rightarrow \tilde{w}\)

largest solution = all winning positions of the defender
Semi-commutation — Encoding Defender’s Strategies

\[ w \in A^* \ldots \text{initial word of the game} \]

Labelled tree:
- defender moves along the edges = removes prefixes of \( w \)
- label = \( \sigma \)-image of the current remainder of \( w \), where \( \sigma : A^* \rightarrow M \) recognizes \( L \)

Example: \( w = abcd, L = \{a, ab, abcde, bc, c, cd, da\} \)

\[ \sigma(abcd) \leftarrow \ldots (ab) \rightarrow \sigma(cd) \]

\[ \sigma(bcd) \leftarrow \ldots (a) \]

\[ \sigma(d) \leftarrow \ldots (a, bc) \]

\[ \sigma(d) \leftarrow \ldots (ab, c) \]

\[ (ab, cd) \rightarrow 1 \]
Semi-commutation — Well-quasiordering Labelled Trees

\( w \leq v \ldots \) winning strategies of the defender for \( w \) can be used also for \( v \)

Example:

\[
\begin{array}{c}
s \\
\downarrow \\
t \\
\downarrow \\
p \\
\end{array} \quad \begin{array}{c}
t \\
\downarrow \\
q \\
\downarrow \\
p \\
\end{array} \quad \begin{array}{c}
t \\
\downarrow \\
s \\
\end{array}
\]

Largest solution is upward closed with respect to \( \leq \).

Kruskal 1960: \( \leq \) is wqo.
Implicit Equations — Tractable Cases for “Simple” Equations

Positive results for commutation equations $XL = LX$:

- three-element languages, regular codes (Karhumäki & Latteux & Petre 2005)
- binary languages closed under factors (Frid 2009)

Open questions for commutation:

- Conjecture: (Ratoandromanana 1989)
  Among codes, equation $XY = YX$ has only solutions of the form $X = L^m$, $Y = L^n$.
  Equivalently: Every code has a primitive root.

Decidability results for conjugacy equations $XK = LX$:

- conjugacy of finite bifix codes via any non-empty language
  (Cassaigne & Karhumäki & Salmela 2007)

Open decision problems for conjugacy:

- existence of a non-empty solution
- solvability with finite constants
- existence of a regular or finite solution
Open Questions

Explicit systems:
- methods for proving non-representability of languages by context-free, conjunctive and Boolean grammars
- closure of conjunctive languages under complementation

General solvability questions:
- equations with concatenation and finite constants
- equations with concatenation (and union) over finite or regular languages

Simple implicit systems:
- regularity of solutions of other simple systems, for example:
  \[ K X L \subseteq M X \]
  \[ K X \subseteq L X, \ X M \subseteq X N \]
- existence of algorithms for finding solutions, which are already known to be regular

Other operations:
- existence of non-trivial shuffle decompositions \( X \uplus Y = L \) of a regular language \( L \)
- existence of non-trivial unambiguous decompositions of regular languages