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, $\mathrm{EXPTIME}$ algorithm finding all solutions

  (Makanin 1977, Plandowski 2006)
- satisfiability-equivalent to language equations with singleton constants and concatenation as the only operation:

  $\mathrm{shortlex}$-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: $u \cdot v = \{uv\} \qquad (K \cdot L = \{\, uv \mid u \in K,\ v \in L \,\})$

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: $u \sqcup\!\sqcup v = \{\, u_1 v_1 \ldots u_k v_k \mid u_1 \ldots u_k = u,\ v_1 \ldots v_k = v \,\}$

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

$$\varphi(X_1, \ldots, X_n) = \psi(X_1, \ldots, X_n)$$

φ, ψ ... 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:

$$(K_1, \ldots, K_n) \leq (L_1, \ldots, L_n) \quad \Longleftrightarrow \quad K_1 \subseteq L_1, \ldots, K_n \subseteq L_n$$

The Basic Property

$f\colon \wp(A^*)^n \to \wp(A^*)$ 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.

$\rightsquigarrow$ describing languages as largest and smallest solutions of systems of equations

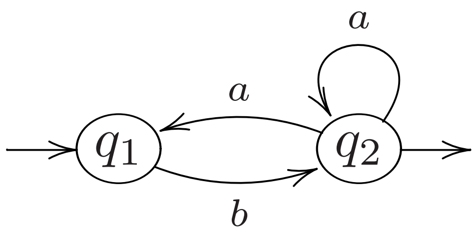
Main questions to study:

- 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:



$$X_1 = \{\varepsilon\} \cup X_2 \cdot a \qquad X_2 = X_1 \cdot b \cup X_2 \cdot a$$

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

$$X_i = K_i \cup \bigcup_{j=1}^{n} X_j \cdot L_{j,i} \qquad i = 1, \ldots, n$$

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

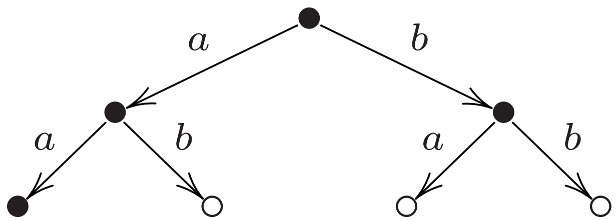
Example: $\{b\} \cup Xa \subseteq X \cup Xba$

$$X \text{ is a solution } \iff X(b) \wedge \big(\forall x\colon X(x) \implies (X(xa) \vee \exists y\colon X(y) \wedge x = yb)\big)$$

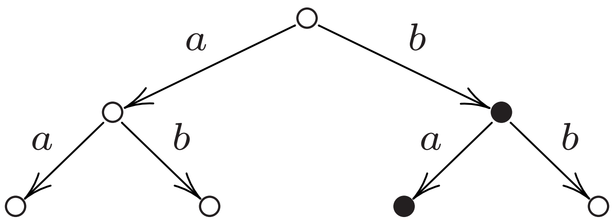
$$X \text{ minimal } \iff \forall Y\colon (Y \text{ is a solution} \wedge \forall x\colon Y(x) \implies X(x)) \implies$$
$$\implies (\forall x\colon X(x) \implies Y(x))$$

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

$a^* \cup b$:



ba^* :



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 $\mathrm{EXPTIME}$-complete

(Aiken & Kozen & Vardi & Wimmers 1994,

Baader & Küsters & Narendran & Okhotin 2001–2006)

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

- $r\colon A^* \to Q$ run of $\mathcal{A}$: $\quad r(\varepsilon) \in I, \quad (r(w), a, r(wa)) \in \delta$
- 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: (MK 2005)

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

Explicit Systems of Equations

Explicit Systems of Equations

$$X_1 = \varphi_1(X_1, \dots, X_n)$$
$$\vdots$$
$$X_n = \varphi_n(X_1, \dots, X_n)$$

notation:

$X = (X_1, \dots, X_n), \quad \varphi = (\varphi_1, \dots, \varphi_n)$

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

φ_i monotone and continuous $\Longrightarrow$ system possesses the least and the greatest solution

$\lim_{k\to\infty} \varphi^k(\emptyset, \dots, \emptyset) \qquad \lim_{k\to\infty} \varphi^k(A^*, \dots, A^*)$

Concatenation and Union — Context-Free Languages

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

context-free grammar: $X_1 \longrightarrow \varepsilon \mid X_2 X_1 \qquad X_2 \longrightarrow (X_1)$

system of language equations: $X_1 = \{\varepsilon\} \cup X_2 \cdot X_1 \qquad X_2 = \{(\} \cdot X_1 \cdot \{)\}$

Ginsburg & Rice 1962:

context-free languages $=$ components of smallest (largest, unique) solutions of explicit systems

$$X_i = S_{i,1} \cup \cdots \cup S_{i,k_i} \qquad i = 1, \ldots, n$$

of polynomial equations with $S_{i,j} \in (A \cup \mathcal{V})^*$

Concatenation, Union and Intersection — Conjunctive Languages

Okhotin 2001–today:

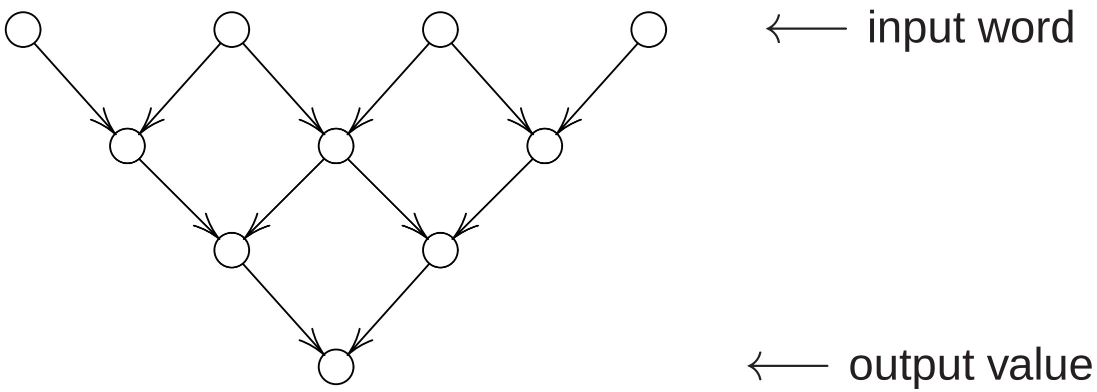
- analogy of alternating machines for context-free grammars
- we can specify that a word satisfies certain syntactic conditions simultaneously
- parsing using standard techniques
- $\subseteq \mathrm{DTIME}(n^3) \cap \mathrm{DSPACE}(n)$

Linear Concatenation, Union and Intersection

$X_i = \varphi_i$ $\qquad$ φ_i constructed from elements of A^* and $A^* \mathcal{V} A^*$ 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 $L \subseteq \{a\}^*$ 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^*
- smallest solutions are (as sets of numbers) in $\mathrm{EXPTIME}$ and
  can be $\mathrm{EXPTIME}$-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 $\Longrightarrow$ 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_ε

final states X_w, where $w \in L$

Universal automaton of a language L

$=$ smallest non-deterministic automaton admitting morphism from every automaton accepting L

state $=$ 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

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

maximal solutions: (Conway 1971)

- finitely many, all of them regular
- for context-free expressions $\bigcup P_i$: algorithmically regular
- $\sigma : A^* \to M$ homomorphism recognizing all languages L_i

  (i.e. $L_i = \sigma^{-1}(F_i)$ for some $F_i \subseteq M$)

$\implies$ σ recognizes all components of maximal solutions

Systems of equations with constant sides:

$\varphi_i(X_1, \ldots, X_n) = L_i \qquad L_i \subseteq A^*$ regular constant, φ_i regular expression

- satisfiability by arbitrary (finite) languages is $\mathrm{EXPSPACE}$-complete (Bala 2006)
- Is satisfiability decidable if φ_i can contain intersection?

Implicit Equations

First Something Simple — Checking Validity for All Languages

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):

$XK = LX, A^*X = A^* \qquad K, L \subseteq A^*$ 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: $XXK = XXL, XM = N, \quad K, L$ 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: (MK 2005)

$XK \subseteq LX$, $X \subseteq M$

$XK \subseteq LX$, $XM \subseteq NX$

$XL = LX$, with L finite

Game corresponding to equation $XL = LX$:

position: $w \in A^*$

attacker: chooses $u \in L$

plays either $w \longrightarrow wu$ or $w \longrightarrow uw$

defender: chooses $v \in L$ so that $wu = v\tilde{w}$, $uw = \tilde{w}v$, respectively

plays $wu \longrightarrow \tilde{w}$, $uw \longrightarrow \tilde{w}$, respectively

largest solution $=$ all winning positions of the defender

Commutation — Example of Non-regular Solution

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

$$L = \{c, ef, ga, e, fg, \hat{f}\hat{e}, a\hat{g}, \hat{e}, \hat{g}\hat{f}, fgba\hat{g}\} \cup cM \cup Mc \cup$$
$$\cup\, A^*bA^*bA^* \cup (A \setminus \{c\})^*b(A \setminus \{c\})^* \setminus N$$

$$M = efga^+ba^* \cup ga^*ba^*\hat{g}\hat{f} \cup a^*ba^*\hat{g}\hat{f}\hat{e} \cup fga^*ba^*\hat{g}$$

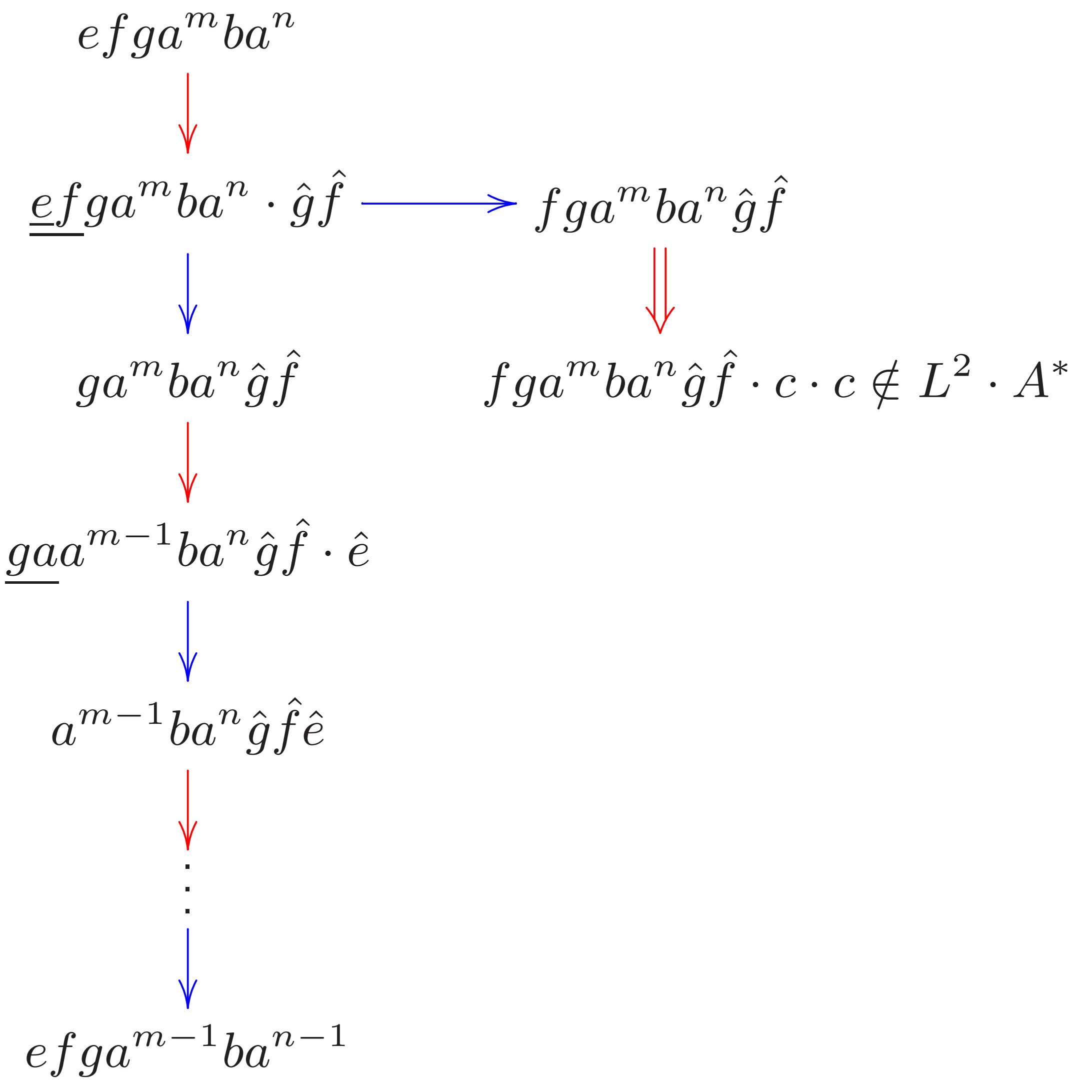
$$N = \{efg, fg, g, \varepsilon\} \cdot a^*ba^* \cdot \{\varepsilon, \hat{g}, \hat{g}\hat{f}, \hat{g}\hat{f}\hat{e}\}$$

encodes simultaneous decrementation of two counters and zero-test

Configuration: $[[[e]f]g]a^mba^n[\hat{g}[\hat{f}[\hat{e}]]]$

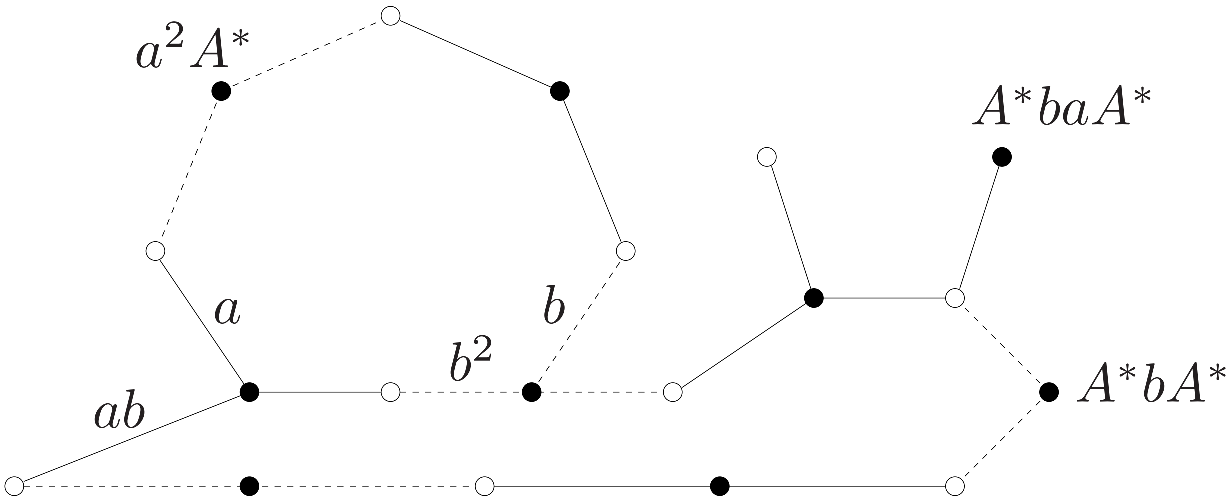
Commutation — Simultaneous Decrementation of Both Counters

Attacker forces defender to remove one a on each side:



Commutation — Encoding Games (Jeandel & Ollinger 2008)

Example:



● = attacker should play ——— modification on the left

○ = defender should play – – – 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

$\Longrightarrow$ satisfiability decidable. (Culik II & Karhumäki 1983, Albert & Lawrence 1985, Guba 1986)

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

$\ldots \subseteq \ldots XLY \ldots$

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 $\bullet\ \bullet\ \bullet \cdots$

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

$\sigma\colon A^* \to M$ ... homomorphism recognizing constant languages on the right

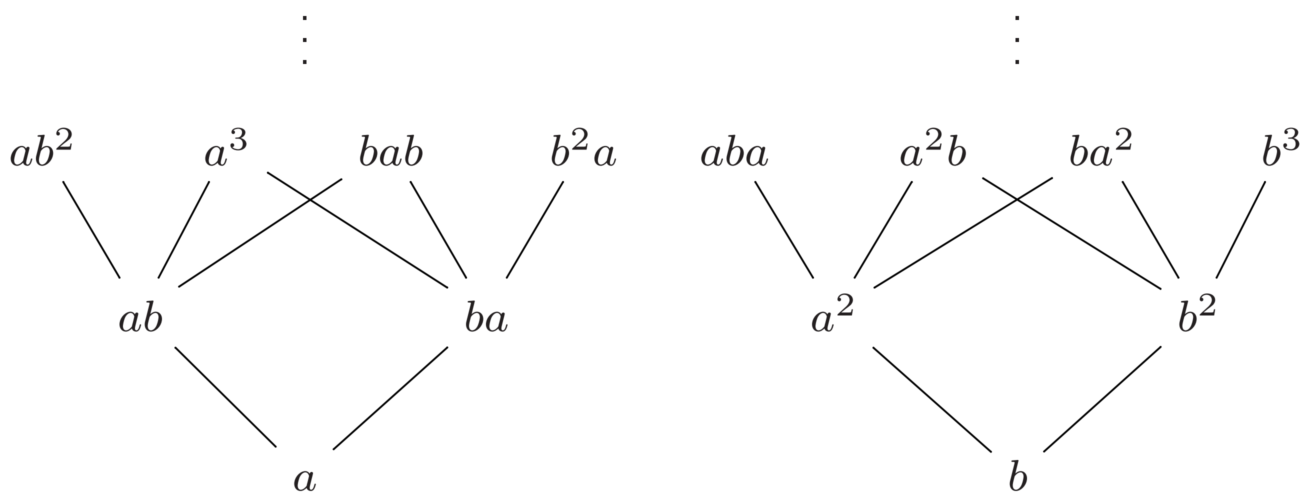
Definition (Bucher & Ehrenfeucht & Haussler 1985):

$$w \leq_\sigma v \iff \begin{aligned} & w = a_1 \cdots a_m, \quad a_j \in A, \\ & v = v_1 \cdots v_m, \quad v_j \in A^+, \\ & \sigma(a_1) = \sigma(v_1), \ldots, \sigma(a_m) = \sigma(v_m) \end{aligned}$$

$\leq_\sigma$ is the derivation relation of the rewriting system

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

Example: $\sigma\colon \{a,b\}^* \to (\{0,1\},+)$ (two-element group) $\sigma(a) = 1, \sigma(b) = 0$



Implicit Inequalities with Restrictions on Constants

Theorem: (MK 2005)

$\sigma \colon A^* \to M$ homomorphism

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

- all operations monotone
- in φ_i all K-ary operations $f\colon (\wp(A^*))^K \to \wp(A^*)$ 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 ψ_i all K-ary operations $f\colon (\wp(A^*))^K \to \wp(A^*)$ satisfy:
  $f((\langle L_k\rangle_{\leq_\sigma})_{k\in K}) \supseteq \langle f((L_k)_{k\in K})\rangle_{\leq_\sigma}$ for all $L_k \subseteq A^*$

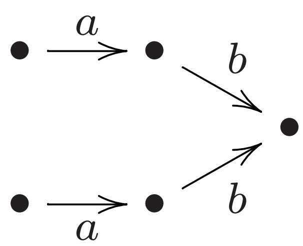
Then all maximal solutions are recognized by $\leq_\sigma$.

Examples of admissible operations:

- anywhere: concatenation, Kleene iteration, shuffle, (infinitary) union,
  constants recognized by σ, 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



$\Longrightarrow$ $\leq_\sigma$ is a wqo $\Longrightarrow$ all maximal solutions are regular

Example: L admissible constant language $\Longrightarrow$ 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

$XK \subseteq LX$ $\qquad$ K arbitrary, L 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 $w \longrightarrow wu$

defender: chooses $v \in L$ so that $wu = v\tilde{w}$

plays $wu \longrightarrow \tilde{w}$

largest solution $=$ all winning positions of the defender

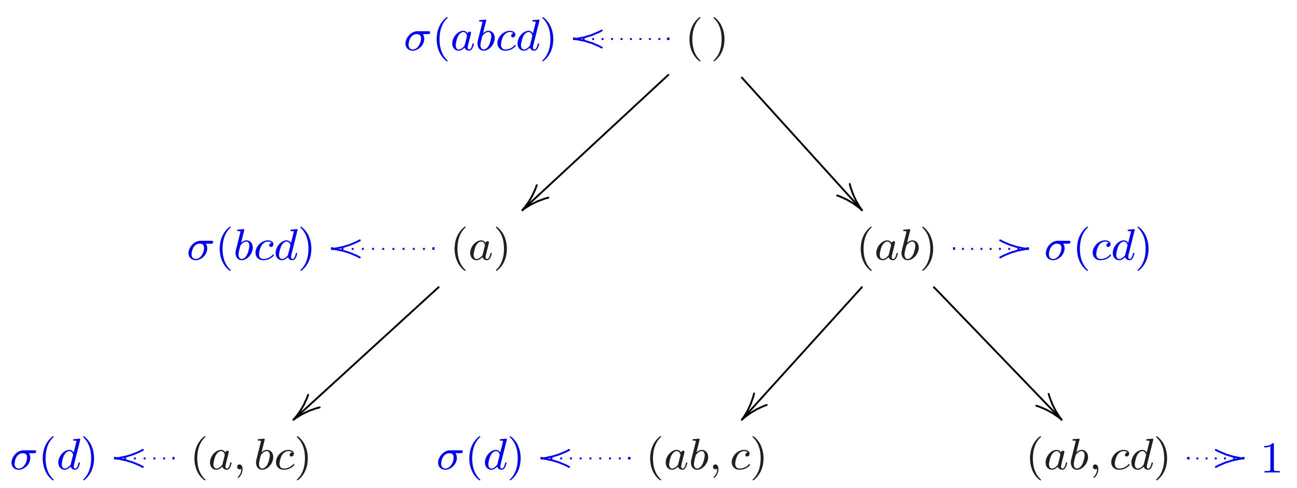
Semi-commutation — Encoding Defender's Strategies

$w \in A^*$ . . . initial word of the game

Labelled tree:

- defender moves along the edges $=$ removes prefixes of w
- label $=$ σ-image of the current remainder of w, where $\sigma : A^* \to M$ recognizes L

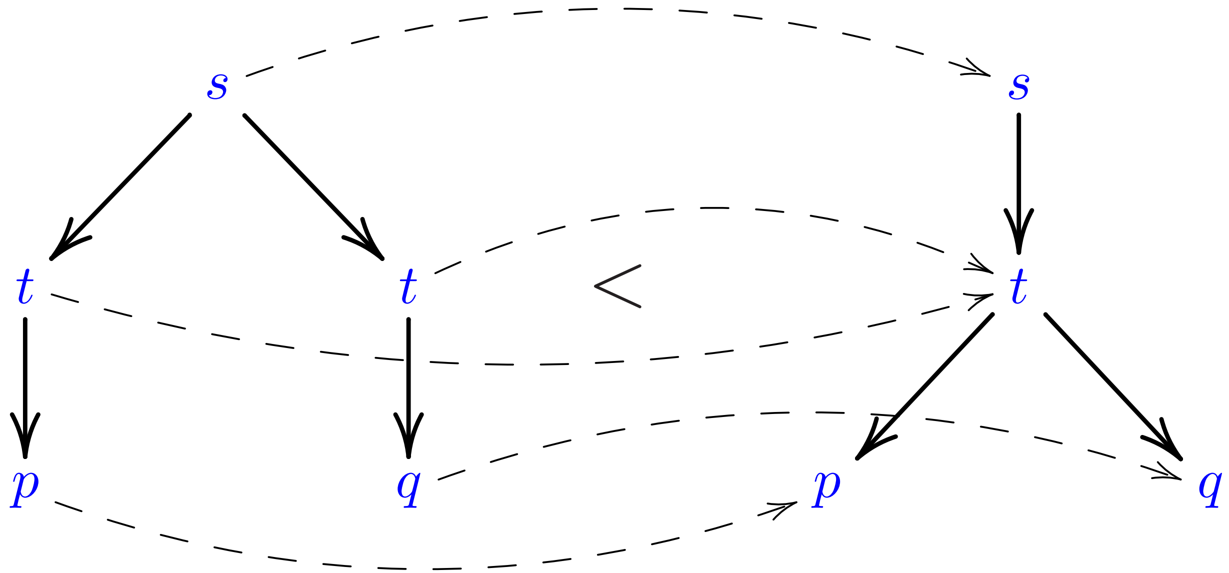
Example: $w = abcd$, $L = \{a, ab, abcde, bc, c, cd, da\}$



Semi-commutation — Well-quasiordering Labelled Trees

$w \leq v$ . . . winning strategies of the defender for w can be used also for v

Example:



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:

  $KXL \subseteq MX$

  $KX \subseteq LX, \; XM \subseteq XN$
- existence of algorithms for finding solutions, which are already known to be regular

Other operations:

- existence of non-trivial shuffle decompositions $X \sqcup\!\sqcup Y = L$ of a regular language L
- existence of non-trivial unambiguous decompositions of regular languages