# What Do We Know About Language Equations?

Michal Kunc

Masaryk University Brno

## What are we going to deal with?

- equations over algebras of formal languages
- concatenation operation, and possibly Boolean operations or Kleene star
- very different from formal power series (unambiguous operations)
- long ago: explicit systems of polynomial equations – context-free languages
- today: renewed interest, surprising recent results

## What are we interested in?

- expressive power, properties of solutions
- decidability of existence and uniqueness of solutions
- algorithms for finding (minimal and maximal) solutions

## What do we need?

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

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

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

concatenation of languages $K \cdot L = \{\, uv \mid u \in K,\ v \in L \,\}$

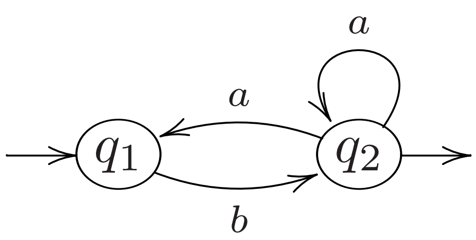
finite set of variables $\mathcal{V} = \{X_1, \dots, X_n\}$

We know . . .

# . . . that they are natural and useful.

## Description of regular languages:

Example:




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

In general:

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

regular languages $=$ components of smallest (largest, unique) solutions of explicit systems of left-linear equations with finite constants $K_i$ and $L_{j,i}$

Matrix notation: union instead of summation

row vectors $X = (X_i)$ and $S = (K_i)$, matrix $R = (L_{j,i})$

$$X = S + XR$$

# Solving Explicit Systems of Left-Linear Equations

Theorem:

Components of the smallest solution of the system $X = S + XR$ belong to the rational closure of entries of $R$ and $S$. (one direction of Kleene theorem)

The system as an automaton:

- language $R_{j,i}$ labels the transition from state $j$ to state $i$
- a word from $S_i$ is read when entering the automaton at state $i$

Proof:

The smallest solution of $X = S + XR$ is $SR^*$, where $R^* = E + R + R^2 + \cdots$.

Inductive formula for computing $R^*$ as a block matrix:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^* = \begin{pmatrix} (A + BD^*C)^* & A^*B(D + CA^*B)^* \\ D^*C(A + BD^*C)^* & (D + CA^*B)^* \end{pmatrix}$$

# Description of Context-Free Languages

Example: Dyck language

$$S \to \varepsilon \mid TS \qquad\qquad X_1 = \{\varepsilon\} \cup X_2 \cdot X_1$$

$$T \to aSb \qquad\qquad X_2 = a \cdot X_1 \cdot b$$

In general:

$$X_i = P_i \qquad i = 1, \ldots, n$$

Ginsburg & Rice 1962:

context-free languages $=$ components of smallest (largest, unique) solutions of explicit systems of polynomial equations with finite $P_i \subseteq (A \cup \mathcal{V})^*$

elegant matrix notation for certain normal forms

Rosenkrantz 1967: construction of quadratic Greibach normal form (right-hand sides of rules belong to $A\mathcal{V}^2 \cup A\mathcal{V} \cup A$)

# Generalizations of Context-Free Languages

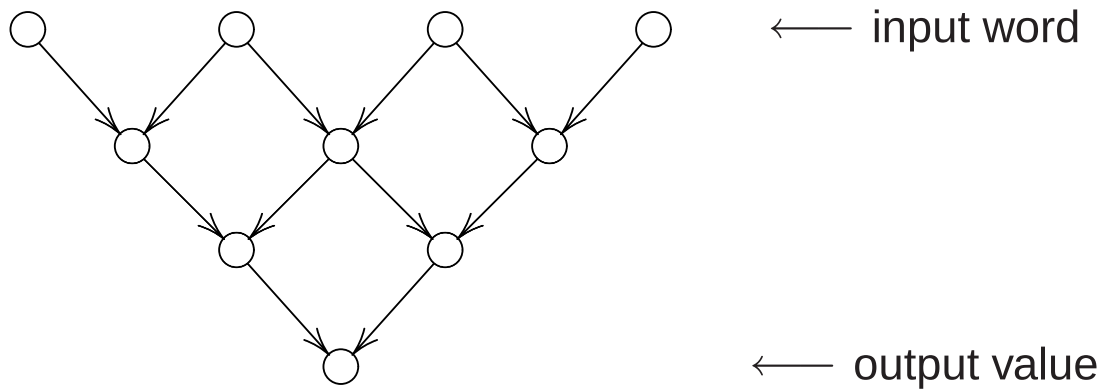
Conjunctive languages (Okhotin 2001):

- analogy of alternating finite automata and Turing machines for context-free grammars
- additionally intersection allowed in equations
- we can specify that a word satisfies certain syntactic conditions simultaneously
- unary languages can be non-regular: regular in positional notation (Jeż 2007), e.g. $a^{2^n}$

Linear conjunctive languages:

Okhotin 2004:

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

# All Boolean Operations

Okhotin 2003:

components of unique (smallest, largest) solutions $=$

$=$ recursive (recursively enumerable, co-recursively enumerable) languages

Boolean grammars (Okhotin 2004):

- restriction to systems with naturally reachable solution (undecidable property)
- generalization of conjunctive languages (in particular, context-free)
- parsing using standard techniques
- $\subseteq \mathrm{DTIME}(n^3) \cap \mathrm{DSPACE}(n)$
- used for formal specification of a simple programming language
- other approaches to defining semantics

Okhotin 2007:

equations with concatenation and any clone of Boolean operations

(concatenation and symmetric difference: universal)

Arithmetical hierarchy:

- components of largest and smallest solutions with respect to lexicographical ordering
- characterized by the number of variables in equations (Okhotin 2005)

# . . . that words are not enough.

Equations over words:

- constants are letters, for variables only words are substituted
- for instance, solutions of equation $xba = abx$ are exactly $x = a(ba)^n$, where $n \in \mathbb{N}_0$
- term unification modulo associativity
- $\mathrm{PSPACE}$ algorithm deciding satisfiability, $\mathrm{EXPTIME}$ algorithm finding all solutions (Makanin 1977, Plandowski 2006)
- Conjecture: Satisfiability problem is $\mathrm{NP}$-complete.
- satisfiability-equivalent to language equations with only letters as constants and concatenation: $\mathrm{shortlex}$-minimal words of an arbitrary language solution form a word solution

Satisfiability of language equations by arbitrary languages is undecidable for

- equations with finite constants, union and concatenation
- systems of equations with regular constants and concatenation (MK 2007)

# Conjugacy of Languages

$KM = ML$ ... languages $K$ and $L$ are conjugated via a language $M$

Words $u$ and $v$ are conjugated $\iff$ $v$ can be obtained from $u$ by cyclic shift.

MK 2007:

Conjugacy of regular languages via any language containing $\varepsilon$ is undecidable.

Corollary:

Satisfiability of systems $KX = XL, A^*X = A^*$ is undecidable for regular languages $K$, $L$.

Cassaigne & Karhumäki & Salmela 2007:

Conjugacy of finite bifix codes via any non-empty language is decidable.

Open questions:

- removal of the requirement on $\varepsilon$
- conjugacy of finite languages (satisfiability of equations with finite constants)
- conjugacy via regular or finite languages (satisfiability by regular or finite languages)

## Identity problem for regular expressions:

$f$, $g$ regular expressions with variables $X_1, \ldots, X_n$ (union, concatenation, Kleene star, letters)

Does $f(L_1, \ldots, L_n) = g(L_1, \ldots, L_n)$ hold for arbitrary (regular) languages $L_1, \ldots, L_n$?

- trivially decidable (treat variables as letters and compare regular languages)
- decidable also with the shuffle operation (Meyer & Rabinovich 2002)
- open problems for expressions with intersection

## Rational systems:

Satisfiability of rational systems of word equations is decidable (thanks to compactness).

(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)

## . . . that they can be often encountered as inequalities.

Minimal automaton of a language $L$:

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

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

initial state $X_\varepsilon$

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 $\iff \varepsilon \in Y$

## . . . that they can be studied in general.

Example: Minimal solutions of $X \cup Y = L$ are precisely disjoint decompositions of $L$.

In the presence of union and concatenation, interesting properties are demonstrated by maximal solutions.

# Systems of Inequalities with Constant Right-Hand Sides

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

maximal solutions (Conway 1971):

- finitely many, all of them regular
- for context-free expressions $P_i$: algorithmically regular
- every solution is contained in a maximal one
- all components are recognized by the syntactic congruence $\sim$ of the languages $L_i$

  $u \sim v \implies (\forall x, y\colon xuy \in L_i \iff xvy \in L_i)$

Analogy: preservation of regularity by arbitrary inverse substitutions:

Largest solution of the inequality $\varphi(X) \subseteq A^* \setminus L$ is $X = A^* \setminus (\varphi^{-1}(L))$.

## Systems of equations with constant right-hand sides:

$P_i = L_i$ $\qquad L_i \subseteq A^*$ regular, $P_i \subseteq (A \cup \mathcal{V})^*$ regular expression

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

# General Left-Linear Inequalities

$$K_0 \cup X_1 K_1 \cup \cdots \cup X_n K_n \subseteq L_0 \cup X_1 L_1 \cup \cdots \cup X_n L_n$$

$K_j, L_j$ regular $\implies$ basic properties of the inequality can be expressed using formulae of monadic second-order theory of infinite $|A|$-ary tree

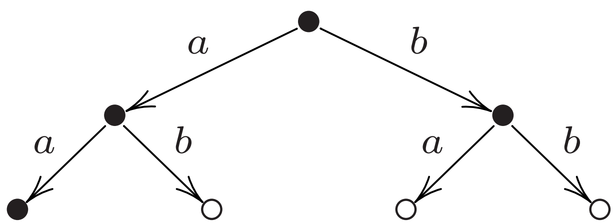
Example: $b \cup Xa \subseteq X \cup Xba$

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

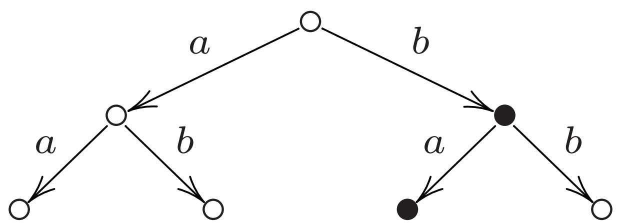
$$X \text{ minimal } \iff \forall Y : (Y \text{ is a solution} \wedge \forall x\colon Y(x) \implies X(x)) \implies$$
$$\implies (\forall x\colon X(x) \implies Y(x))$$

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

$a^* \cup b$ :




$ba^*$ :




Rabin 1969 $\implies$ algorithmically solvable using tree automata

very special case of set constraints (letters as unary functions)

$\mathrm{EXPTIME}$-complete (even when complementation is allowed) (1994–2006)

# Yet More General Left-Linear Inequalities

$$K_0 \cup X_1K_1 \cup \cdots \cup X_nK_n \subseteq L_0 \cup X_1L_1 \cup \cdots \cup X_nL_n$$

$K_j$ arbitrary, $L_j$ regular

MK 2005:

largest solution:

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

# Concatenations on the Right

Previous cases:

| | |
|---|---|
| $\ldots \subseteq L$ | constants on the right fix the context |
| $XK \cup \ldots \subseteq XL \cup \ldots$ | local modifications on one side |

Next task:

| | |
|---|---|
| $\ldots \subseteq XLY$ | general concatenations on the right |

We need to classify words according to their decompositions with respect to constant languages.

# Well-quasiorder (wqo)

Quasiorder $\leq$ on $A^*$ is a wqo, if it contains neither  nor $\bullet\ \bullet\ \bullet \cdots$

Equivalent definitions:

- Every upward closed language over $A$ is finitely generated.
- There is no infinite ascending sequence of upward closed languages.

Monotone: $u \leq v \;\&\; \tilde{u} \leq \tilde{v} \implies u\tilde{u} \leq v\tilde{v}$

Example: “scattered subword” relation

Ehrenfeucht & Haussler & Rozenberg 1983:

$L \subseteq A^*$ is regular $\iff$ $L$ is upward closed with respect to a monotone wqo on $A^*$.

Special case:

Congruence of finite index is a monotone well-quasiorder.

upward closed $=$ recognized by the congruence

Applying well-quasiorders to inequalities:

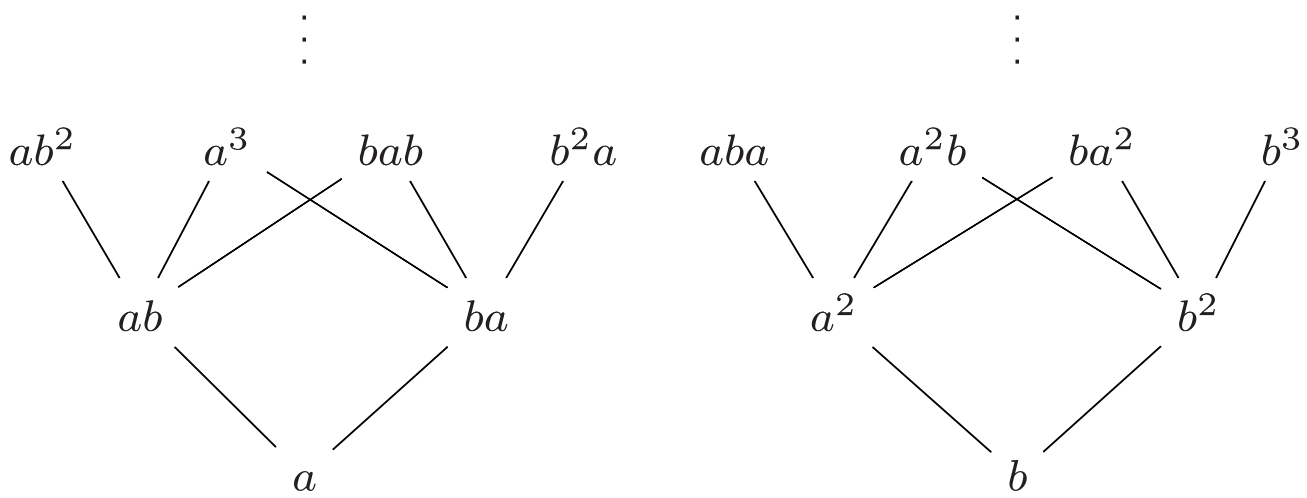
Construct a wqo on $A^*$ such that every solution is contained in an upward closed solution.

# A Quasiorder for Dealing with Concatenations on the Right

$\sim$ . . . syntactic congruence of constant languages on the right side of inequalities

$$w \leq v \iff \begin{aligned} & w = a_1 \cdots a_m,\, a_j \in A, \\ & v = v_1 \cdots v_m,\, v_j \in A^+, \\ & a_j \sim v_j,\, j = 1, \ldots, m \end{aligned}$$

Example: $\{a,b\}^+ / \sim \;\cong \mathbb{Z}_2 \qquad 1 = [a]_\sim,\, 0 = [b]_\sim$

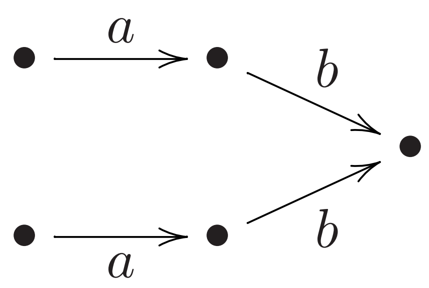


# Restrictions on Constants

## Systems of inequalities $P_i \subseteq Q_i$

$P_i \subseteq (A \cup \mathcal{V})^*$ arbitrary

$Q_i$ ... regular expressions over variables and languages, whose minimal automaton does not contain




MK 2005: all maximal solutions are regular

Corollary:

The class of polynomials of group languages is closed under taking maximal solutions of such systems.

# ...that they are nice to play with.

$XK \subseteq LX$ $\quad$ $K$ arbitrary, $L$ regular

largest solution: • always regular

- for context-free $K$: algorithmically recursive (MK 2005)
- 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: $u \in K$, $w \longrightarrow wu$

defender: $v \in L$, $wu = v\tilde{w}$, $wu \longrightarrow \tilde{w}$

largest solution $=$ all winning positions of the defender

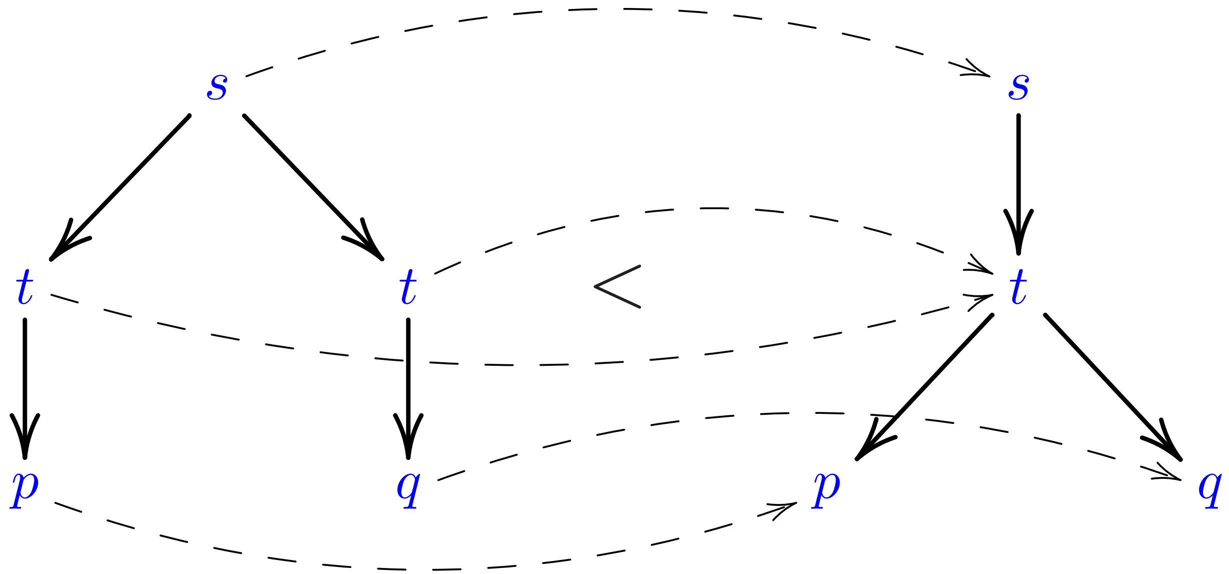
Example: $w = abcd$, $L = \{a, ab, abcde, bc, c, cd, da\}$, $\sim =$ syntactic congruence of $L$

$$
\begin{tikzcd}
 & & & & & {[abcd]_\sim} & (\,) \arrow[l, dotted] \arrow[dlll] \arrow[drr] & & & & & \\
 & & {[bcd]_\sim} & (a) \arrow[l, dotted] \arrow[dll] & & & & & (ab) \arrow[r, dotted] \arrow[dlll] \arrow[drr] & {[cd]_\sim} & & \\
{[d]_\sim} & (a, bc) \arrow[l, dotted] & & & {[d]_\sim} & (ab, c) \arrow[l, dotted] & & & & & (ab, cd) \arrow[r, dotted] & 1
\end{tikzcd}
$$

# Well-quasiordering Trees

$w \leq v \ldots$ 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.

## . . . that they can be surprisingly powerful.

MK 2005:

Every co-recursively enumerable language can be described as the largest solution of any of the following systems with regular constants $K$, $L$, $M$ and $N$.

$$\begin{array}{ccc} XK \subseteq LX & XK \subseteq LX & XK \subseteq LX \\ X \subseteq M & XM \subseteq NX & MX \subseteq XN \end{array}$$

Special case: $XL = LX$

- formulated by Conway 1971
- positive results:

  at most ternary languages, regular codes (Karhumäki & Latteux & Petre 2005)

MK 2007:

There exists a finite language $L$ such that the largest solution $\mathcal{C}(L)$ of $XL = LX$ is not recursively enumerable.

# Example: $L$ regular, but $\mathcal{C}(L)$ non-regular

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

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

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

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

encodes simultaneous decrementation of two counters and zero-test

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

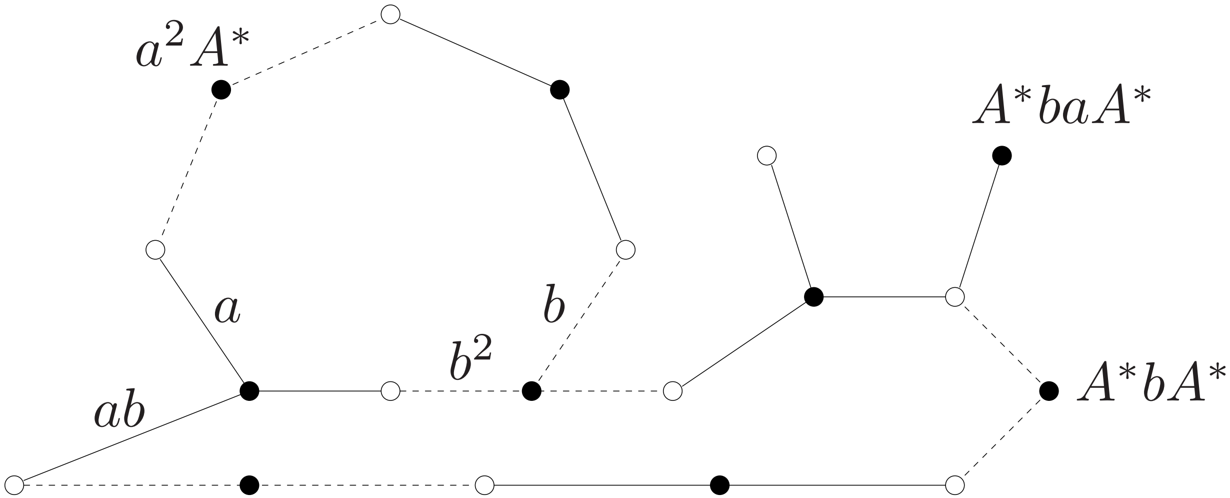
# Simultaneous Decrementation of Both Counters

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

$$
\begin{tikzcd}
efga^m ba^n \arrow[d] & \\
\underline{ef}ga^m ba^n \cdot \hat{g}\hat{f} \arrow[r] \arrow[d] & fga^m ba^n \hat{g}\hat{f} \arrow[d, Rightarrow] \\
ga^m ba^n \hat{g}\hat{f} \arrow[d] & fga^m ba^n \hat{g}\hat{f} \cdot c \cdot c \notin L^2 \cdot A^* \\
\underline{ga}a^{m-1} ba^n \hat{g}\hat{f} \cdot \hat{e} \arrow[d] & \\
a^{m-1} ba^n \hat{g}\hat{f}\hat{e} \arrow[d] & \\
\vdots \arrow[d] & \\
efga^{m-1} ba^{n-1} &
\end{tikzcd}
$$

# Games That Can Be Encoded (Jeandel & Ollinger)

Example:




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

○ = defender should play – – – modification on the right

position of the game: a vertex of the graph and a word

labels of attacker's vertices: 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 vertex for any two connected components of different types
- only one type of edges leading from each of attacker's vertices
- non-empty labels of edges only around one attacker's vertex for each type of edges

# ... that we do not understand their languages.

- satisfiability of equations with concatenation (and union) over finite or regular languages
- satisfiability of equations with concatenation and finite constants
- 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.
- regularity of solutions of other simple systems of inequalities, for example:

  $KXL \subseteq MX$

  $KX \subseteq LX,\ XM \subseteq XN$
- existence of algorithms for finding regular solutions
- methods for proving properties of conjunctive and Boolean grammars
- existence of non-trivial shuffle decomposition $X \sqcup\!\sqcup Y = L$ of a regular language $L$
- existence of non-trivial unambiguous decompositions of regular languages
- unary languages

$$X = TY = Z_1 Z_2$$

$$X^2 = Z_1 ank\ you Th Z_2$$