Szegedi Tudományegyetem
Informatikai Tanszékcsoport
Számitástudomány Alapjai Tanszék
2006. május 8.

Identities of Two-Dimensional Languages

Igor Dolinka

Department of Mathematics and Informatics
University of Novi Sad, Serbia & Mont.

What is a two-dimensional language?

What is a two-dimensional language?

Well, it depends on your personal view of ordinary (string) languages...

What is a two-dimensional language?

Well, it depends on your personal view of ordinary (string) languages...

Version A. (Combinatorial)

WORD = a finite sequence of letters

What is a two-dimensional language?

Well, it depends on your personal view of ordinary (string) languages...

Version A. (Combinatorial)

WORD = a finite sequence of letters

Version B. (Algebraic)

WORD = an element of a free monoid

What is a two-dimensional language?

A

A **two-dimensional word** is a matrix of letters – a **picture**:

$$P = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix},$$

where $a_{ij} \in \Sigma$ for some alphabet Σ.

What is a two-dimensional language?

A

A **two-dimensional word** is a matrix of letters – a **picture**:

$$P = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix},$$

where $a_{ij} \in \Sigma$ for some alphabet Σ.

A **picture language** is a set of pictures.

Operations on pictures and picture languages

Let $P = [a_{ij}]_{m \times n}$ and $Q = [b_{ij}]_{k \times \ell}$ be pictures.

Operations on pictures and picture languages

Let $P = [a_{ij}]_{m \times n}$ and $Q = [b_{ij}]_{k \times \ell}$ be pictures.

The **column product** $P \rightarrow Q$ is defined only if $m = k$, and its result is

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} & b_{11} & \ldots & b_{1\ell} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} & b_{m1} & \cdots & b_{m\ell} \end{bmatrix}.$$

Operations on pictures and picture languages

Let $P = [a_{ij}]_{m \times n}$ and $Q = [b_{ij}]_{k \times \ell}$ be pictures.

The **column product** $P \to Q$ is defined only if $m = k$, and its result is

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} & b_{11} & \ldots & b_{1\ell} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} & b_{m1} & \cdots & b_{m\ell} \end{bmatrix}.$$

The **row product** $P \downarrow Q$ is defined only if $n = \ell$, and its result is

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \\ b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{k1} & \cdots & b_{kn} \end{bmatrix}.$$

Operations on pictures and picture languages

Let L_1, L_2, L be picture languages.

Operations on pictures and picture languages

Let L_1, L_2, L be picture languages.

Products

$$L_1 \rightarrow L_2 = \{P_1 \rightarrow P_2 : \ P_i \in L_i, \ i = 1, 2, \ P_1 \rightarrow P_2 \text{ exists}\},$$

$$L_1 \downarrow L_2 = \{P_1 \downarrow P_2 : \ P_i \in L_i, \ i = 1, 2, \ P_1 \downarrow P_2 \text{ exists}\}.$$

Operations on pictures and picture languages

Let L_1, L_2, L be picture languages.

Products

$$L_1 \rightarrow L_2 = \{P_1 \rightarrow P_2 : \; P_i \in L_i, \; i = 1, 2, \; P_1 \rightarrow P_2 \text{ exists}\},$$
$$L_1 \downarrow L_2 = \{P_1 \downarrow P_2 : \; P_i \in L_i, \; i = 1, 2, \; P_1 \downarrow P_2 \text{ exists}\}.$$

Iterations

$$L^{>} = \bigcup_{n \geqslant 0} L^{\overset{n}{\rightarrow}}, \qquad L^{\vee} = \bigcup_{n \geqslant 0} L^{\downarrow n},$$

where $L^{\overset{0}{\rightarrow}} = L^{\downarrow 0} = \{\epsilon\}$.

What is a two-dimensional language?

B

A **two-dimensional word** is an element of a **free binoid** over Σ.

What is a two-dimensional language?

B

A **two-dimensional word** is an element of a **free binoid** over Σ.

Free binoid = the free object in the variety of all algebras with two binary associative operations and a common 1 (to be denoted by ϵ).

What is a two-dimensional language?

B

A **two-dimensional word** is an element of a **free binoid** over Σ.

Free binoid = the free object in the variety of all algebras with two binary associative operations and a common 1 (to be denoted by ϵ).

A **binoid language** (or **bi-language**) is a subset of a free binoid.

How to represent elements of a free binoid? Take 1.

Z.Ésik (2000): sp-biposets

How to represent elements of a free binoid? Take 1.

Z.Ésik (2000): sp-biposets

Σ-labelled biposets: a set with two strict orders $\mathcal{A} = (A, <_1, <_2)$ and a labelling function $\lambda_{\mathcal{A}} : A \to \Sigma$.

How to represent elements of a free binoid? Take 1.

Z.Ésik (2000): sp-biposets

Σ-labelled biposets: a set with two strict orders $\mathcal{A} = (A, <_1, <_2)$ and a labelling function $\lambda_{\mathcal{A}} : A \to \Sigma$.

$x \in \Sigma$ is identified with the singleton poset S_x, labelled by x.

How to represent elements of a free binoid? Take 1.

New biposets are obtained by two binary operations $\circ_1, \circ_2$, where $\mathcal{A} \circ_i \mathcal{B}$ $(i = 1, 2)$ is defined on $A \cup B$ by

$$<_j^{\mathcal{A} \circ_i \mathcal{B}} = \begin{cases} <_j^{\mathcal{A}} \cup <_j^{\mathcal{B}} & \text{if } j \neq i, \\ <_j^{\mathcal{A}} \cup <_j^{\mathcal{B}} \cup (A \times B) & \text{if } j = i. \end{cases}$$

How to represent elements of a free binoid? Take 1.

New biposets are obtained by two binary operations $\circ_1, \circ_2$, where $\mathcal{A} \circ_i \mathcal{B}$ $(i = 1, 2)$ is defined on $A \cup B$ by

$$<_j^{\mathcal{A} \circ_i \mathcal{B}} = \begin{cases} <_j^{\mathcal{A}} \cup <_j^{\mathcal{B}} & \text{if } j \neq i, \\ <_j^{\mathcal{A}} \cup <_j^{\mathcal{B}} \cup (A \times B) & \text{if } j = i. \end{cases}$$

A biposet is series-parallel (**sp** for short) if it is generated from the singletons by the two product operations.

How to represent elements of a free binoid? Take 2.

I prefer to think about the (nonempty) elements of a free binoid as trees labelled by $\Sigma \cup \{\rightarrow, \downarrow\}$, called **bi-words**.

How to represent elements of a free binoid? Take 2.

I prefer to think about the (nonempty) elements of a free binoid as trees labelled by $\Sigma \cup \{\rightarrow, \downarrow\}$, called **bi-words**.

(1) the leaves are labelled by the letters from Σ,

How to represent elements of a free binoid? Take 2.

I prefer to think about the (nonempty) elements of a free binoid as trees labelled by $\Sigma \cup \{\rightarrow, \downarrow\}$, called **bi-words**.

(1) the leaves are labelled by the letters from Σ,

(2) the labels $\rightarrow, \downarrow$ of the non-leaves alternate, depending on the parity of the distance from the root,

How to represent elements of a free binoid? Take 2.

I prefer to think about the (nonempty) elements of a free binoid as trees labelled by $\Sigma \cup \{\rightarrow, \downarrow\}$, called **bi-words**.

(1) the leaves are labelled by the letters from Σ,

(2) the labels $\rightarrow, \downarrow$ of the non-leaves alternate, depending on the parity of the distance from the root,

(3) each non-leaf has $\geqslant 2$ successors.

How to represent elements of a free binoid? Take 2.

I prefer to think about the (nonempty) elements of a free binoid as trees labelled by $\Sigma \cup \{\rightarrow, \downarrow\}$, called **bi-words**.

(1) the leaves are labelled by the letters from Σ,

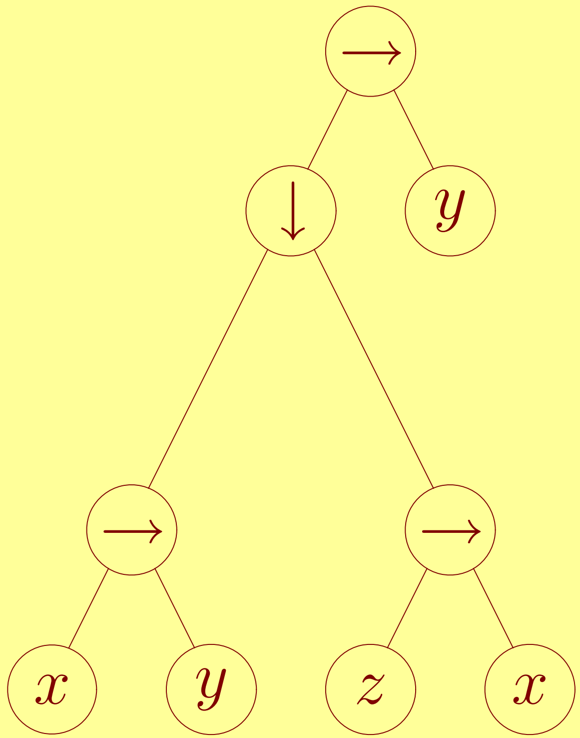
(2) the labels $\rightarrow, \downarrow$ of the non-leaves alternate, depending on the parity of the distance from the root,

(3) each non-leaf has $\geqslant 2$ successors.

The set of all bi-words over Σ: BW_{Σ}

How to represent elements of a free binoid? Take 2.

Example. $b(x,y,z) = ((x \to y) \downarrow (z \to x)) \to y$



How to represent elements of a free binoid? Take 2.

We distinguish between three kinds of bi-words:

How to represent elements of a free binoid? Take 2.

We distinguish between three kinds of bi-words:

(1) horizontal $=$ the root is labelled by $\longrightarrow$,

How to represent elements of a free binoid? Take 2.

We distinguish between three kinds of bi-words:

(1) horizontal = the root is labelled by $\rightarrow$,

(2) vertical = the root is labelled by $\downarrow$,

How to represent elements of a free binoid? Take 2.

We distinguish between three kinds of bi-words:

(1) horizontal = the root is labelled by $\rightarrow$,

(2) vertical = the root is labelled by $\downarrow$,

(3) neutral = singletons + ϵ.

How to represent elements of a free binoid? Take 2.

We distinguish between three kinds of bi-words:

(1) horizontal = the root is labelled by $\longrightarrow$,

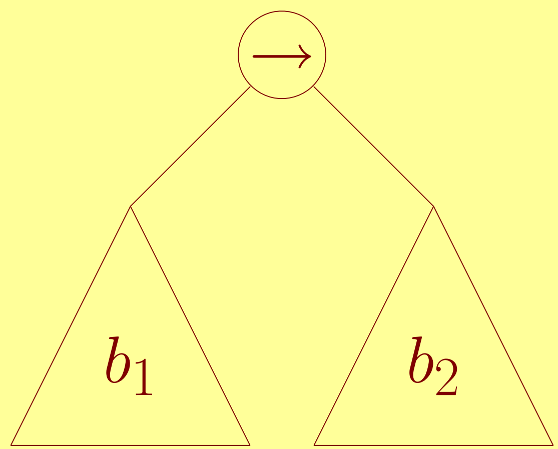
(2) vertical = the root is labelled by $\downarrow$,

(3) neutral = singletons + ϵ.

As an example, we show how the horizontal product works. We have three cases.

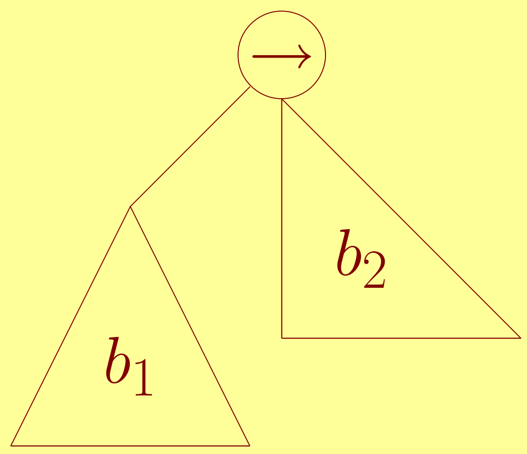
How to represent elements of a free binoid? Take 2.

Case 1: b_1, b_2 are vertical/neutral



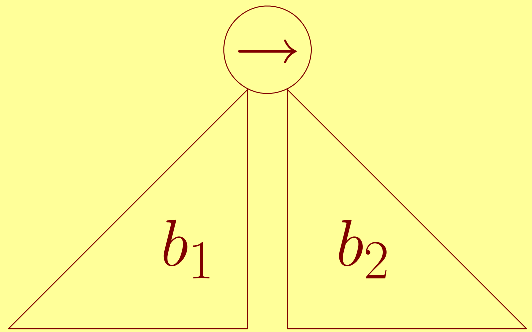
How to represent elements of a free binoid? Take 2.

Case 2: b_1 is vertical/neutral, b_2 is horizontal



How to represent elements of a free binoid? Take 2.

Case 3: b_1, b_2 are horizontal



Operations on binoid languages

Let L_1, L_2, L be binoid languages.

Operations on binoid languages

Let L_1, L_2, L be binoid languages.

Products

$$L_1 \rightarrow L_2 = \{b_1 \rightarrow b_2 : \; b_i \in L_i, \; i = 1, 2\},$$

$$L_1 \downarrow L_2 = \{b_1 \downarrow b_2 : \; b_i \in L_i, \; i = 1, 2\}.$$

Operations on binoid languages

Let L_1, L_2, L be binoid languages.

Products

$$L_1 \rightarrow L_2 = \{b_1 \rightarrow b_2 : \ b_i \in L_i, \ i = 1, 2\},$$
$$L_1 \downarrow L_2 = \{b_1 \downarrow b_2 : \ b_i \in L_i, \ i = 1, 2\}.$$

Iterations

$$L^{>} = \bigcup_{n \geqslant 0} L^{\overset{n}{\rightarrow}}, \qquad L^{\vee} = \bigcup_{n \geqslant 0} L^{\downarrow n},$$

where $L^{\overset{0}{\rightarrow}} = L^{\downarrow 0} = \{\epsilon\}$.

Algebras

Algebra of bi-languages over Σ:

$$\mathbf{BiLang}_\Sigma = (\mathcal{P}(\mathsf{BW}_\Sigma), +, \rightarrow, \downarrow, {}^{>}, {}^{\vee}, \varnothing, \{\epsilon\})$$

Algebras

Algebra of bi-languages over Σ:

$$\mathbf{BiLang}_\Sigma = (\mathcal{P}(\mathsf{BW}_\Sigma), +, \rightarrow, \downarrow, {}^{>}, {}^{\vee}, \varnothing, \{\epsilon\})$$

Algebra of picture languages over Σ:

$$\mathbf{Pict}_\Sigma = (\mathcal{P}(\Sigma^{**}), \cup, \rightarrow, \downarrow, {}^{>}, {}^{\vee}, \varnothing, \{\epsilon\})$$

Algebras

Algebra of bi-languages over Σ:

$$\mathbf{BiLang}_\Sigma = (\mathcal{P}(\mathsf{BW}_\Sigma), +, \rightarrow, \downarrow, {}^{>}, {}^{\vee}, \varnothing, \{\epsilon\})$$

Algebra of picture languages over Σ:

$$\mathbf{Pict}_\Sigma = (\mathcal{P}(\Sigma^{**}), \cup, \rightarrow, \downarrow, {}^{>}, {}^{\vee}, \varnothing, \{\epsilon\})$$

A word of caution: Recognizable picture languages (REC) require, besides the above operations, the intersection and the so-called alphabetic projection.

A result ($\sim$, 2005)

Theorem. Identities satisfied by all algebras $\mathbf{BiLang}_{\Sigma}$ = identities satisfied by all algebras $\mathbf{Pict}_{\Sigma}$.

I.Dolinka, A note on identities of two-dimensional languages, *Discrete Applied Mathematics* **146** (2005), 43–50.

A result ($\sim$, 2005)

Theorem. Identities satisfied by all algebras $\mathbf{BiLang}_\Sigma$ = identities satisfied by all algebras $\mathbf{Pict}_\Sigma$.

I.Dolinka, A note on identities of two-dimensional languages, *Discrete Applied Mathematics* **146** (2005), 43–50.

In the sequel, we denote the above equational theory by Θ.

A result ($\sim$, 2005)

Idea:

Proposition.

A result ($\sim$, 2005)

Idea:

Proposition. For each bi-word $b = b(x_1, \ldots, x_n)$ there are:

A result ($\sim$, 2005)

Idea:

Proposition. For each bi-word $b = b(x_1, \ldots, x_n)$ there are:

- an alphabet Γ,

A result ($\sim$, 2005)

Idea:

Proposition. For each bi-word $b = b(x_1, \ldots, x_n)$ there are:

- an alphabet Γ,
- a picture $P_b \in \Gamma^{**}$ (the "witness" picture), and

A result ($\sim$, 2005)

Idea:

Proposition. For each bi-word $b = b(x_1, \ldots, x_n)$ there are:

- an alphabet Γ,
- a picture $P_b \in \Gamma^{**}$ (the "witness" picture), and
- finite picture languages $L_1, \ldots, L_n \subseteq \Gamma^{**}$

A result ($\sim$, 2005)

Idea:

Proposition. For each bi-word $b = b(x_1, \ldots, x_n)$ there are:

- an alphabet Γ,
- a picture $P_b \in \Gamma^{**}$ (the "witness" picture), and
- finite picture languages $L_1, \ldots, L_n \subseteq \Gamma^{**}$
  (consisting of homogeneous pictures = rectangles filled with a single kind of letter)

A result ($\sim$, 2005)

Idea:

Proposition. For each bi-word $b = b(x_1, \ldots, x_n)$ there are:

- an alphabet Γ,
- a picture $P_b \in \Gamma^{**}$ (the "witness" picture), and
- finite picture languages $L_1, \ldots, L_n \subseteq \Gamma^{**}$
  (consisting of homogeneous pictures = rectangles filled with a single kind of letter)

such that for any bi-word $b' = b'(x_1, \ldots, x_n)$ we have

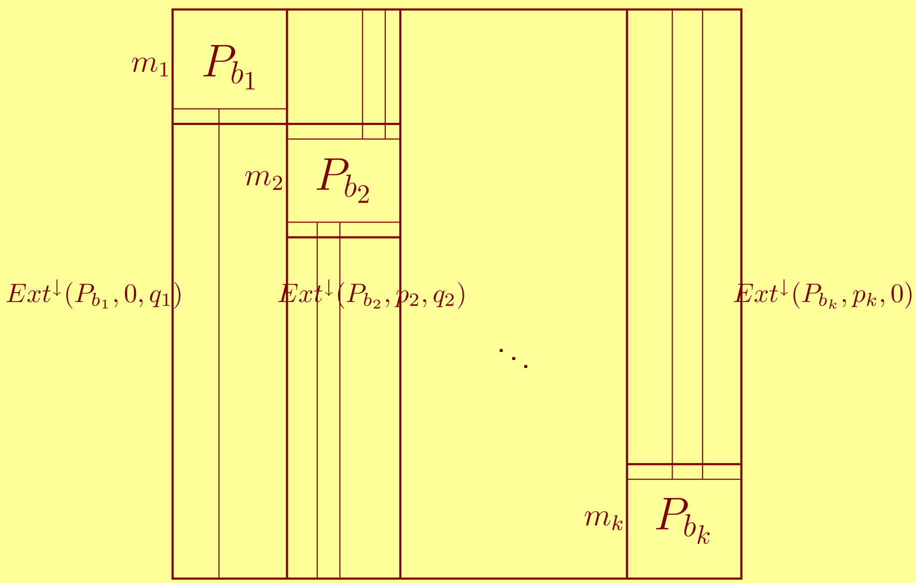
$$P_b \in b'(L_1, \ldots, L_n) \iff b' \equiv b.$$

A result ($\sim$, 2005)

Idea: Suppose we have witness pictures P_{b_i} for b_i, $1 \leqslant i \leqslant k$.

A result ($\sim$, 2005)

Idea: Suppose we have witness pictures P_{b_i} for b_i, $1 \leqslant i \leqslant k$.
The witness for $b_1 \to b_2 \to \ldots \to b_k$ is:



A result ($\sim$, 2005)

Example: $b(x, y, z) = ((x \rightarrow y) \downarrow (z \rightarrow x)) \rightarrow y$

A result ($\sim$, 2005)

Example: $b(x, y, z) = ((x \to y) \downarrow (z \to x)) \to y$

The algorithm from the proof of **Proposition** gives $\Gamma = \{1, 2, 3, 4, 5\}$.

A result ($\sim$, 2005)

Example: $b(x, y, z) = ((x \to y) \downarrow (z \to x)) \to y$

The algorithm from the proof of **Proposition** gives $\Gamma = \{1, 2, 3, 4, 5\}$.

The witness picture is:

$$P_b = \begin{bmatrix} 1 & 2 & 2 & 2 & 5 \\ 1 & 2 & 2 & 2 & 5 \\ 3 & 3 & 3 & 4 & 5 \\ 3 & 3 & 3 & 4 & 5 \\ 3 & 3 & 3 & 4 & 5 \end{bmatrix}.$$

A problem

What are the axioms for the equational theory Θ ?

A problem

What are the axioms for the equational theory Θ ?

Conjecture. The identities of ordinary string languages in the 'horizontal' signature $\{+, \rightarrow, {}^{>}, \emptyset, \epsilon\}$ & the same identities in the 'vertical' signature $\{+, \downarrow, {}^{\vee}, \emptyset, \epsilon\}$ will do.

A problem

What are the axioms for the equational theory Θ ?

Conjecture. The identities of ordinary string languages in the 'horizontal' signature $\{+, \rightarrow, ^{>}, \varnothing, \epsilon\}$ & the same identities in the 'vertical' signature $\{+, \downarrow, ^{\vee}, \varnothing, \epsilon\}$ will do.

Recently, I succeeded in proving that this conjecture is true.

A short summary of the proof follows.

Definitions #1

Birational expression = term in the signature $\{+, \rightarrow, \downarrow, {}^{>}, {}^{\vee}, \varnothing, \epsilon\}$.

Definitions #1

Birational expression = term in the signature $\{+, \rightarrow, \downarrow, {}^{>}, {}^{\vee}, \varnothing, \epsilon\}$.

$\rightarrow$-rational ($\downarrow$-rational) expression = birational expression which contains only $+$, the constants, and the horizontal (vertical) operation symbols.

Definitions #1

Birational expression $=$ term in the signature $\{+, \rightarrow, \downarrow, {}^{>}, {}^{\vee}, \varnothing, \epsilon\}$.

$\rightarrow$-rational ($\downarrow$-rational) expression $=$ birational expression which contains only $+$, the constants, and the horizontal (vertical) operation symbols.

Value of a birational expression α, $\mathcal{B}(\alpha)$ $=$ value of the term α under $x \mapsto \{x\}$, $x \in \Sigma$.

Definitions #1

Birational expression = term in the signature $\{+, \rightarrow, \downarrow, {}^{>}, {}^{\vee}, \varnothing, \epsilon\}$.

$\rightarrow$-rational ($\downarrow$-rational) expression = birational expression which contains only $+$, the constants, and the horizontal (vertical) operation symbols.

Value of a birational expression α, $\mathcal{B}(\alpha)$ = value of the term α under $x \mapsto \{x\}$, $x \in \Sigma$.

Birational bi-language = bi-language of the form $\mathcal{B}(\alpha)$

Definitions #1

Z.Ésik & Z.L.Németh (2004): every birational bi-language consists of bi-words of bounded depth ($\subseteq \mathsf{BW}_{\Sigma}^{\leqslant d}$). The least such d is the depth $\delta(\alpha)$ of the corresponding expression α.

Definitions #1

Z.Ésik & Z.L.Németh (2004): every birational bi-language consists of bi-words of bounded depth ($\subseteq \mathrm{BW}_{\Sigma}^{\leqslant d}$). The least such d is the depth $\delta(\alpha)$ of the corresponding expression α.

Horizontal (vertical) birational expression $\alpha = \mathcal{B}(\alpha)$ consists entirely of horizontal (vertical) and neutral bi-words.

Definitions #1

Z.Ésik & Z.L.Németh (2004): every birational bi-language consists of bi-words of bounded depth ($\subseteq \mathsf{BW}_{\Sigma}^{\leqslant d}$). The least such d is the depth $\delta(\alpha)$ of the corresponding expression α.

Horizontal (vertical) birational expression $\alpha = \mathcal{B}(\alpha)$ consists entirely of horizontal (vertical) and neutral bi-words.

Γ_1 (Γ_2) $=$ all identities of string languages in the horizontal (vertical) signature.

Decomposition Lemma

For any birational expression α, there are birational expressions α^h and α^v such that

- $\alpha = \alpha^h + \alpha^v$ follows from $\Gamma_1 \cup \Gamma_2$,
- $\mathcal{B}(\alpha_h)$ ($\mathcal{B}(\alpha_v)$) consists precisely of all horizontal (vertical) and neutral bi-words from $\mathcal{B}(\alpha)$.

Decomposition Lemma

For any birational expression α, there are birational expressions α^h and α^v such that

- $\alpha = \alpha^h + \alpha^v$ follows from $\Gamma_1 \cup \Gamma_2$,
- $\mathcal{B}(\alpha_h)$ ($\mathcal{B}(\alpha_v)$) consists precisely of all horizontal (vertical) and neutral bi-words from $\mathcal{B}(\alpha)$.

Lemma. Let α_1, α_2 be birational expressions, and let α_i^h, α_i^v $(i = 1, 2)$ have the same meaning as above. Then $\alpha_1 = \alpha_2$ belongs to Θ if and only if both $\alpha_1^h = \alpha_2^h$ and $\alpha_1^v = \alpha_2^v$ belong to Θ.

Definitions #2 & a lemma

A possible problem: α is a horizontal expression $\Rightarrow \alpha \downarrow \epsilon$ is horizontal (in spite of being of the form $\_\_ \downarrow \_\_$)

Definitions #2 & a lemma

A possible problem: α is a horizontal expression $\Rightarrow \alpha \downarrow \epsilon$ is horizontal (in spite of being of the form $\_\_ \downarrow \_\_$)

An expression α is trimmed if it is either

Definitions #2 & a lemma

A possible problem: α is a horizontal expression $\Rightarrow \alpha \downarrow \epsilon$ is horizontal (in spite of being of the form $\_\_ \downarrow \_\_$)

An expression α is trimmed if it is either

- graphically equal to $\varnothing$, or

Definitions #2 & a lemma

A possible problem: α is a horizontal expression $\Rightarrow \alpha \downarrow \epsilon$ is horizontal (in spite of being of the form $\_\_ \downarrow \_\_$)

An expression α is trimmed if it is either

- graphically equal to $\varnothing$, or
- has no subterm equivalent to $\varnothing$ or ϵ, except

Definitions #2 & a lemma

A possible problem: α is a horizontal expression $\Rightarrow \alpha \downarrow \epsilon$ is horizontal (in spite of being of the form $\_\_ \downarrow \_\_$)

An expression α is trimmed if it is either

- graphically equal to $\varnothing$, or
- has no subterm equivalent to $\varnothing$ or ϵ, except
  - a possible single summand graphically equal to ϵ

Definitions #2 & a lemma

A possible problem: α is a horizontal expression $\Rightarrow \alpha \downarrow \epsilon$ is horizontal (in spite of being of the form $\_\_ \downarrow \_\_$)

An expression α is trimmed if it is either

- graphically equal to $\varnothing$, or
- has no subterm equivalent to $\varnothing$ or ϵ, except
  - a possible single summand graphically equal to ϵ

Lemma. For each α there is a trimmed expression α_0 such that $\Gamma_1 \cup \Gamma_2 \vdash \alpha = \alpha_0$.

Linearization Lemma

Let α be a horizontal birational expression.

Linearization Lemma

Let α be a horizontal birational expression.

(i) There exist a linear ($=$ each variable occurs exactly once) $\longrightarrow$-rational expression $\alpha'(x_1, \ldots, x_n)$ and vertical expressions $\beta_1, \ldots, \beta_n$ such that

$$\alpha \equiv \alpha'(\beta_1, \ldots, \beta_n).$$

In such a case, if $\delta(\alpha) \geqslant 1$, we have $\delta(\alpha) = \max(\delta(\beta_1), \ldots, \delta(\beta_n)) + 1$.

Linearization Lemma

Let α be a horizontal birational expression.

(i) There exist a linear (= each variable occurs exactly once) $\longrightarrow$-rational expression $\alpha'(x_1, \ldots, x_n)$ and vertical expressions $\beta_1, \ldots, \beta_n$ such that

$$\alpha \equiv \alpha'(\beta_1, \ldots, \beta_n).$$

In such a case, if $\delta(\alpha) \geqslant 1$, we have $\delta(\alpha) = \max(\delta(\beta_1), \ldots, \delta(\beta_n)) + 1$.

(ii) There exist a horizontal birational expression $\hat{\alpha}$, a linear $\longrightarrow$-rational expression $\alpha''(x_1, \ldots, x_k)$ and vertical expressions $\beta'_1, \ldots, \beta'_k$ such that

(a) the identity $\alpha = \hat{\alpha}$ follows from $\Gamma_1 \cup \Gamma_2$,

(b) $\hat{\alpha} \equiv \alpha''(\beta'_1, \ldots, \beta'_k)$, and

(c) $\epsilon \notin \mathcal{B}(\beta'_i)$ and $\mathcal{B}(\beta'_i) \neq \emptyset$ for all $1 \leqslant i \leqslant k$.

Definition: Doppelgänger (as in "Twin Peaks")

Let α_1, α_2 be two horizontal birational expressions (of depth $d \geqslant 1$). **Linearization Lemma** $\Rightarrow$

$$\begin{aligned} \alpha_1 &= \alpha_1''(\beta_1, \ldots, \beta_n), \\ \alpha_2 &= \alpha_2''(\beta_{n+1}, \ldots, \beta_m), \end{aligned}$$

where α_i'' are linear, and $\epsilon \notin \mathcal{B}(\beta_i) \neq \varnothing$.

Definition: Doppelgänger (as in "Twin Peaks")

Let α_1, α_2 be two horizontal birational expressions (of depth $d \geqslant 1$). **Linearization Lemma** $\Rightarrow$

$$\begin{aligned} \alpha_1 &= \alpha_1''(\beta_1, \ldots, \beta_n), \\ \alpha_2 &= \alpha_2''(\beta_{n+1}, \ldots, \beta_m), \end{aligned}$$

where α_i'' are linear, and $\epsilon \notin \mathcal{B}(\beta_i) \neq \varnothing$.

Let $Y_i = \mathcal{B}(\beta_i)$ $(1 \leqslant i \leqslant m)$.

Definition: Doppelgänger (as in "Twin Peaks")

Let α_1, α_2 be two horizontal birational expressions (of depth $d \geqslant 1$). **Linearization Lemma** $\Rightarrow$

$$\begin{aligned} \alpha_1 &= \alpha_1''(\beta_1, \ldots, \beta_n), \\ \alpha_2 &= \alpha_2''(\beta_{n+1}, \ldots, \beta_m), \end{aligned}$$

where α_i'' are linear, and $\epsilon \notin \mathcal{B}(\beta_i) \neq \varnothing$.

Let $Y_i = \mathcal{B}(\beta_i)$ $(1 \leqslant i \leqslant m)$.

All there languages are (nonempty) subsets of E – the set of all neutral and vertical bi-words of depth $\leqslant d - 1$.

Definition: Doppelgänger (as in "Twin Peaks")

For convenience, let $Y_i^0 = Y_i$ and $Y_i^1 = E \setminus Y_i$.

Definition: Doppelgänger (as in “Twin Peaks”)

For convenience, let $Y_i^0 = Y_i$ and $Y_i^1 = E \setminus Y_i$.

For a binary sequence $\sigma \in \{0,1\}^m$, let

$$X_\sigma = \bigcap_{i=1}^{m} Y_i^{\sigma(i)}.$$

Definition: Doppelgänger (as in "Twin Peaks")

For convenience, let $Y_i^0 = Y_i$ and $Y_i^1 = E \setminus Y_i$.

For a binary sequence $\sigma \in \{0,1\}^m$, let

$$X_\sigma = \bigcap_{i=1}^{m} Y_i^{\sigma(i)}.$$

What the heck is this?

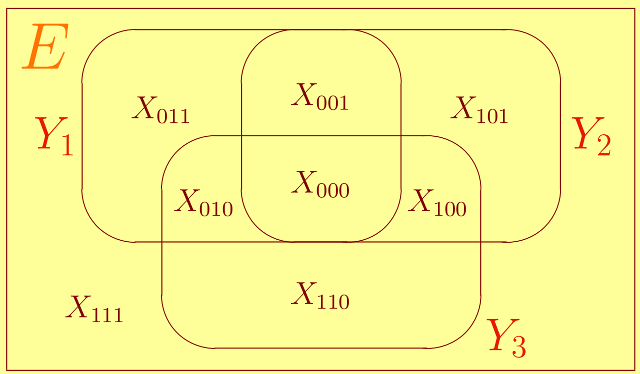
Definition: Doppelgänger (as in "Twin Peaks")

For convenience, let $Y_i^0 = Y_i$ and $Y_i^1 = E \setminus Y_i$.

For a binary sequence $\sigma \in \{0,1\}^m$, let

$$X_\sigma = \bigcap_{i=1}^{m} Y_i^{\sigma(i)}.$$

What the heck is this?



Definition: Doppelgänger (as in "Twin Peaks")

For $1 \leqslant i \leqslant m$, define the sets $\Lambda_i \subseteq \{0,1\}^m$ by

$$\sigma \in \Lambda_i \quad \text{if and only if} \quad \sigma(i) = 0 \text{ and } X_\sigma \neq \emptyset.$$

Definition: Doppelgänger (as in "Twin Peaks")

For $1 \leqslant i \leqslant m$, define the sets $\Lambda_i \subseteq \{0,1\}^m$ by

$$\sigma \in \Lambda_i \quad \text{if and only if} \quad \sigma(i) = 0 \text{ and } X_\sigma \neq \emptyset.$$

The 'horizontal' identity

$$\alpha_1''\left(\sum_{\sigma\in\Lambda_1} x_\sigma, \ldots, \sum_{\sigma\in\Lambda_n} x_\sigma\right) = \alpha_2''\left(\sum_{\sigma\in\Lambda_{n+1}} x_\sigma, \ldots, \sum_{\sigma\in\Lambda_m} x_\sigma\right)$$

is an adjoined string identity (or doppelgänger) for $\alpha_1 = \alpha_2$.

Definition: Doppelgänger (as in "Twin Peaks")

For $1 \leqslant i \leqslant m$, define the sets $\Lambda_i \subseteq \{0,1\}^m$ by

$$\sigma \in \Lambda_i \quad \text{if and only if} \quad \sigma(i) = 0 \text{ and } X_\sigma \neq \emptyset.$$

The 'horizontal' identity

$$\alpha_1''\left(\sum_{\sigma\in\Lambda_1} x_\sigma, \dots, \sum_{\sigma\in\Lambda_n} x_\sigma\right) = \alpha_2''\left(\sum_{\sigma\in\Lambda_{n+1}} x_\sigma, \dots, \sum_{\sigma\in\Lambda_m} x_\sigma\right)$$

is an adjoined string identity (or doppelgänger) for $\alpha_1 = \alpha_2$.

The idea behind this identity is that the above sums of letters (from $\Xi_m = \{x_\sigma : \sigma \in \{0,1\}^m\}$) indexed by Λ_i's record the **set-theoretical configuration** of the bi-languages Y_i.

Example

$$x^{>} + (x^{\vee})^{>} = (x^{\vee})^{>}$$

Example

$$x^{>} + (x^{\vee})^{>} = (x^{\vee})^{>}$$

Linearization yields $\beta_1^{>} + \beta_2^{>} = \beta_3^{>}$, where

$$\begin{aligned} \beta_1 &\equiv x \\ \beta_2 \equiv \beta_3 &\equiv x^{\vee}. \end{aligned}$$

Example

$$x^{>} + (x^{\vee})^{>} = (x^{\vee})^{>}$$

Linearization yields $\beta_1^{>} + \beta_2^{>} = \beta_3^{>}$, where

$$\beta_1 \equiv x$$
$$\beta_2 \equiv \beta_3 \equiv x^{\vee}.$$

To get rid of ϵ from $\mathcal{B}(\beta_2) = \mathcal{B}(\beta_3)$, we make use of

$$x^{\vee} = \epsilon + x \downarrow x^{\vee}$$

and proceed with $x \downarrow x^{\vee}$ instead of $x^{\vee}$.

Example

Now we have $Y_1 \subset Y_2 = Y_3$, thus $\Lambda_1 = \{000\}$ and $\Lambda_2 = \Lambda_3 = \{000, 100\}$.

Example

Now we have $Y_1 \subset Y_2 = Y_3$, thus $\Lambda_1 = \{000\}$ and $\Lambda_2 = \Lambda_3 = \{000, 100\}$.

For simplicity, write x for x_{000} and y for x_{100}. So, our doppelgänger is just

$$x^> + (x+y)^> = (x+y)^>,$$

a familiar law telling that the Kleene star is monotone.

Doppelgänger Lemma

Assume $\alpha_1 = \alpha_2$ belongs to Θ (i.e. it is a valid bi-langauge identity). Then its doppelgänger is a valid string identity.

The main proof (outlined)

Goal: to prove that a valid identity $\alpha_1 = \alpha_2$ is a consequence of $\Gamma_1 \cup \Gamma_2$.

The main proof (outlined)

Goal: to prove that a valid identity $\alpha_1 = \alpha_2$ is a consequence of $\Gamma_1 \cup \Gamma_2$.

Plan: induction on $\delta(\alpha_1) = \delta(\alpha_2) = d$ (case $d \leqslant 1$ is trivial...).

The main proof (outlined)

Goal: to prove that a valid identity $\alpha_1 = \alpha_2$ is a consequence of $\Gamma_1 \cup \Gamma_2$.

Plan: induction on $\delta(\alpha_1) = \delta(\alpha_2) = d$ (case $d \leqslant 1$ is trivial...).

Decomposition Lemma $\Rightarrow$ $\alpha_i = \alpha_i^h + \alpha_i^v$ $(i = 1, 2)$ follows from $\Gamma_1 \cup \Gamma_2$.

The main proof (outlined)

Goal: to prove that a valid identity $\alpha_1 = \alpha_2$ is a consequence of $\Gamma_1 \cup \Gamma_2$.

Plan: induction on $\delta(\alpha_1) = \delta(\alpha_2) = d$ (case $d \leqslant 1$ is trivial...).

Decomposition Lemma $\Rightarrow \alpha_i = \alpha_i^h + \alpha_i^v$ $(i = 1, 2)$ follows from $\Gamma_1 \cup \Gamma_2$.

$\alpha_1 = \alpha_2$ holds if and only if both $\alpha_1^h = \alpha_2^h$ and $\alpha_1^v = \alpha_2^v$ are valid.

The main proof (outlined)

Goal: to prove that a valid identity $\alpha_1 = \alpha_2$ is a consequence of $\Gamma_1 \cup \Gamma_2$.

Plan: induction on $\delta(\alpha_1) = \delta(\alpha_2) = d$ (case $d \leqslant 1$ is trivial...).

Decomposition Lemma $\Rightarrow \alpha_i = \alpha_i^h + \alpha_i^v$ $(i = 1, 2)$ follows from $\Gamma_1 \cup \Gamma_2$.

$\alpha_1 = \alpha_2$ holds if and only if both $\alpha_1^h = \alpha_2^h$ and $\alpha_1^v = \alpha_2^v$ are valid.

So, we may assume that both α_1 and α_2 are e.g. horizontal.

The main proof (outlined)

Decomposition Lemma $\Rightarrow$ there are horizontal birational expressions $\hat{\alpha}_1, \hat{\alpha}_2$ such that

$$\Gamma_1 \cup \Gamma_2 \vdash \quad \alpha_1 = \hat{\alpha}_1, \quad \alpha_2 = \hat{\alpha}_2,$$

The main proof (outlined)

Decomposition Lemma $\Rightarrow$ there are horizontal birational expressions $\hat{\alpha}_1, \hat{\alpha}_2$ such that

$$\Gamma_1 \cup \Gamma_2 \vdash \quad \alpha_1 = \hat{\alpha}_1, \quad \alpha_2 = \hat{\alpha}_2,$$

while the identity $\hat{\alpha}_1 = \hat{\alpha}_2$ has the form

$$\alpha_1''(\beta_1', \ldots, \beta_k') = \alpha_2''(\beta_{k+1}', \ldots, \beta_m'),$$

The main proof (outlined)

Decomposition Lemma $\Rightarrow$ there are horizontal birational expressions $\hat{\alpha}_1, \hat{\alpha}_2$ such that

$$\Gamma_1 \cup \Gamma_2 \vdash \quad \alpha_1 = \hat{\alpha}_1, \quad \alpha_2 = \hat{\alpha}_2,$$

while the identity $\hat{\alpha}_1 = \hat{\alpha}_2$ has the form

$$\alpha_1''(\beta_1', \ldots, \beta_k') = \alpha_2''(\beta_{k+1}', \ldots, \beta_m'),$$

where α_1'', α_2'' are linear $\longrightarrow$-rational expressions (involved later in the course of forming a doppelgänger identity), and $\beta_1', \ldots, \beta_m'$ are vertical expressions, all of them having depth at most $d-1$, whose values $Y_1, \ldots, Y_m$ satisfy $\epsilon \notin Y_i \neq \varnothing$, $1 \leqslant i \leqslant m$.

The main proof (outlined)

Let $\Lambda_1, \ldots, \Lambda_m$ and X_σ, $\sigma \in I$, be as in the definition of a doppelgänger. We already know that

$$Y_i = \bigcup_{\sigma \in \Lambda_i} X_\sigma$$

holds for all $1 \leqslant i \leqslant m$.

The main proof (outlined)

Let $\Lambda_1, \ldots, \Lambda_m$ and X_σ, $\sigma \in I$, be as in the definition of a doppelgänger. We already know that

$$Y_i = \bigcup_{\sigma \in \Lambda_i} X_\sigma$$

holds for all $1 \leqslant i \leqslant m$.

Ésik–Németh (2004) $\Rightarrow$ birational bi-languages closed for complements (and so, for intersections), so all X_σ's are birational,

$$X_\sigma = \mathcal{B}(\xi_\sigma).$$

The main proof (outlined)

Therefore, the following identities are valid:

$$\beta_i' = \sum_{\sigma \in \Lambda_i} \xi_\sigma, \quad (*)$$

for all $1 \leqslant i \leqslant m$.

The main proof (outlined)

Therefore, the following identities are valid:

$$\beta_i' = \sum_{\sigma \in \Lambda_i} \xi_\sigma, \quad (*)$$

for all $1 \leqslant i \leqslant m$.

This is an identity of depth $\leqslant d - 1$, so it follows from $\Gamma_1 \cup \Gamma_2$ by induction hypothesis.

The main proof (outlined)

Doppelgänger Lemma $\Rightarrow$ the adjoined string identity

$$\alpha_1''\left(\sum_{\sigma\in\Lambda_1} x_\sigma, \ldots, \sum_{\sigma\in\Lambda_n} x_\sigma\right) = \alpha_2''\left(\sum_{\sigma\in\Lambda_{n+1}} x_\sigma, \ldots, \sum_{\sigma\in\Lambda_m} x_\sigma\right)$$

is a valid one, thus it belongs to Γ_1.

The main proof (outlined)

Doppelgänger Lemma $\Rightarrow$ the adjoined string identity

$$\alpha_1''\left(\sum_{\sigma\in\Lambda_1} x_\sigma, \ldots, \sum_{\sigma\in\Lambda_n} x_\sigma\right) = \alpha_2''\left(\sum_{\sigma\in\Lambda_{n+1}} x_\sigma, \ldots, \sum_{\sigma\in\Lambda_m} x_\sigma\right)$$

is a valid one, thus it belongs to Γ_1.

Apply the substitution $x_\sigma \mapsto \xi_\sigma$.

The main proof (outlined)

Doppelgänger Lemma $\Rightarrow$ the adjoined string identity

$$\alpha_1''\left(\sum_{\sigma\in\Lambda_1} x_\sigma, \ldots, \sum_{\sigma\in\Lambda_n} x_\sigma\right) = \alpha_2''\left(\sum_{\sigma\in\Lambda_{n+1}} x_\sigma, \ldots, \sum_{\sigma\in\Lambda_m} x_\sigma\right)$$

is a valid one, thus it belongs to Γ_1.

Apply the substitution $x_\sigma \mapsto \xi_\sigma$.

By combining $(*)$ and the above doppelgänger, we obtain the required formal proof for $\alpha_1 = \alpha_2$.

Example (continued)

$$x^{>} + (x^{\vee})^{>} = (x^{\vee})^{>}$$

Example (continued)

$$x^{>} + (x^{\vee})^{>} = (x^{\vee})^{>}$$

This is first transformed into

$$x^{>} + (x \downarrow x^{\vee})^{>} = (x \downarrow x^{\vee})^{>}.$$

Example (continued)

$$x^{>} + (x^{\vee})^{>} = (x^{\vee})^{>}$$

This is first transformed into

$$x^{>} + (x \downarrow x^{\vee})^{>} = (x \downarrow x^{\vee})^{>}.$$

As we have argued, a doppelgänger is

$$x^{>} + (x + y)^{>} = (x + y)^{>}.$$

Example (continued)

$$x^{>} + (x^{\vee})^{>} = (x^{\vee})^{>}$$

This is first transformed into

$$x^{>} + (x \downarrow x^{\vee})^{>} = (x \downarrow x^{\vee})^{>}.$$

As we have argued, a doppelgänger is

$$x^{>} + (x + y)^{>} = (x + y)^{>}.$$

So, the nonempty X_σ's are $X_{000} = \{x\}$ and $X_{100} = \{x \downarrow x, x \downarrow x \downarrow x, \ldots\}$.

Example (continued)

$$x^{>} + (x^{\vee})^{>} = (x^{\vee})^{>}$$

This is first transformed into

$$x^{>} + (x \downarrow x^{\vee})^{>} = (x \downarrow x^{\vee})^{>}.$$

As we have argued, a doppelgänger is

$$x^{>} + (x + y)^{>} = (x + y)^{>}.$$

So, the nonempty X_σ's are $X_{000} = \{x\}$ and $X_{100} = \{x \downarrow x, x \downarrow x \downarrow x, \ldots\}$.

Thus, we have $\xi_{000} \equiv x$ and $\xi_{100} = x \downarrow x \downarrow x^{\vee}$.

Example (continued)

$$x^{>} + (x^{\vee})^{>} = (x^{\vee})^{>}$$

This is first transformed into

$$x^{>} + (x \downarrow x^{\vee})^{>} = (x \downarrow x^{\vee})^{>}.$$

As we have argued, a doppelgänger is

$$x^{>} + (x + y)^{>} = (x + y)^{>}.$$

So, the nonempty X_σ's are $X_{000} = \{x\}$ and $X_{100} = \{x \downarrow x, x \downarrow x \downarrow x, \ldots\}$.

Thus, we have $\xi_{000} \equiv x$ and $\xi_{100} = x \downarrow x \downarrow x^{\vee}$.

Now, our identity follows from the above doppelgänger and

$$x + x \downarrow x \downarrow x^{\vee} = x \downarrow x^{\vee}.$$

THANK YOU!

All questions and comments to:

dockie@im.ns.ac.yu

A preprint may be found at:

www.im.ns.ac.yu/personal/dolinkai