Crosswords of formal languages

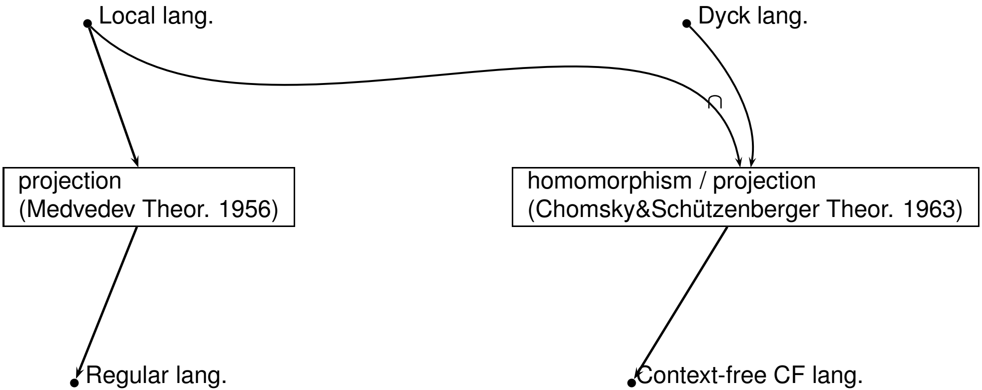
Stefano Crespi Reghizzi[1]

[1] Politecnico di Milano and CNR - IEIIT, Italy

FLAT Seminar, 18th December 2024

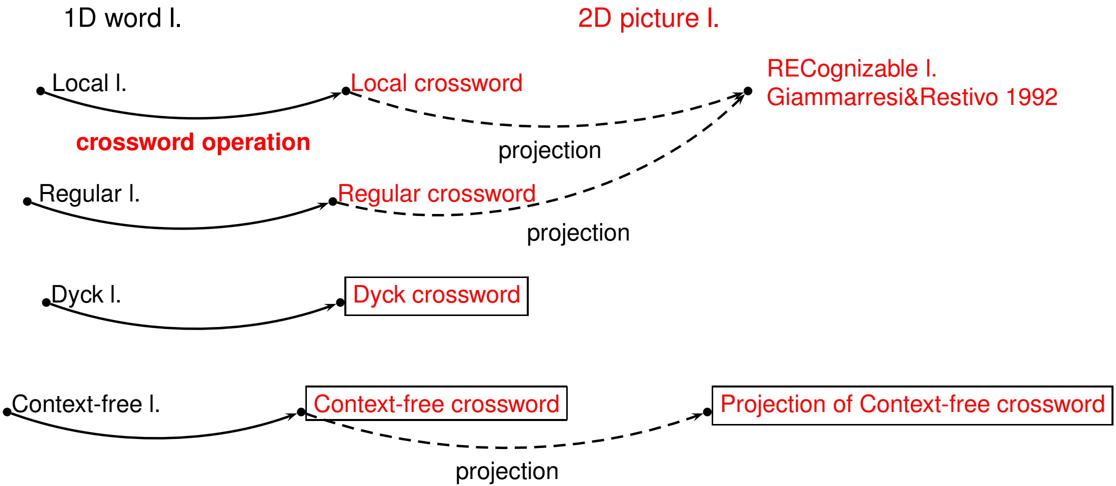
from recent research with Antonio Restivo and Pierluigi San Pietro

Historical inspirations from word (1D) language theorems



- NB Alphabetic transformation: projection $\stackrel{def}{=}$ non-erasing letter-to-letter homomorphism

From 1D to 2D by the crossword approach



Pictures, Crosswords

- picture $p \in \Sigma^{++}$: rectangular array of characters in alphabet Σ
- row resp. column concatenation: $p \varobar p'$ resp. , $p \ominus p'$ (partial operations) and closures $p^{\varobar *}$, $p^{\ominus *}$
- row language of $p \subseteq \Sigma^{++}$: $ROW(p) = \{w \in \Sigma^+ \mid \exists p \in P \text{ having } w \text{ as a row}\}$ similarly $COL(p)$
- For (word) languages L_1 , $L_2 \subseteq \Sigma^+$ N.B. over the same alphabet
- **Define** : crossword CW a.k.a. *Row-Column Combination* :

$$\boxed{L_1 \boxplus L_2 \stackrel{def}{=} L_1^{\ominus *} \cap L_2^{\varobar *}}$$

  as the set of pictures s.t. each row-word and column-word is resp. in L_1 and in L_2
- A more flexible definition of the crossword operation allowing different alphabets for rows and columns will be needed for the Chomsky-Schützenberger Theor.

Regular Crosswords

- **Example**: a single 1 in every row and column

$$\text{Let } L = 0^*10^*, \qquad L \boxplus L = \boxed{\begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{matrix}}$$

- Regular CWs are known since long. Lately:
  Fenner, Padé, and Thierauf *The complexity of regex crosswords*, 2022
  and the game *RegexCrosswords.com*

Recognizable pictures REC : *projection of regular CW*

- Originally [Giammarresi & Restivo]: projection of picture languages defined by 2×2 tiles
- **Equivalent Definition**: [Latteux & Simplot]
- $\boxed{\textit{REC} = \textit{projection of local CW} = \textit{projection of regular CW}}$
- **Example:** regular CW $P = b^* d\, a^* \boxplus a^* d\, b^*$,

e.g.

$$\begin{array}{cccc} d & a & a & a \\ b & d & a & a \\ b & b & d & a \\ b & b & b & d \end{array}$$

ROW $= b^* d\, a^*$ defined by *domino tiles* : $\boxed{\#d}, \boxed{\#b}, \boxed{da}, \boxed{bd}, \boxed{bb}, \boxed{a\#}, \boxed{d\#}$

COL $= a^* d\, b^*$ defined by domino tiles:

$$\begin{vmatrix} \# \\ d \end{vmatrix}, \begin{vmatrix} \# \\ a \end{vmatrix}, \begin{vmatrix} d \\ b \end{vmatrix}, \begin{vmatrix} a \\ d \end{vmatrix}, \begin{vmatrix} a \\ a \end{vmatrix}, \begin{vmatrix} b \\ b \end{vmatrix}, \begin{vmatrix} b \\ \# \end{vmatrix}, \begin{vmatrix} d \\ \# \end{vmatrix}$$

- the *square pictures* over a *unary* alphabet $\begin{array}{cccc} s & s & s & s \\ s & s & s & s \\ s & s & s & s \\ s & s & s & s \end{array}$ are not the CW of regular languages but are the projection $\left\{ \begin{array}{l} a \to s \\ b \to s \\ d \to s \end{array} \right\}$ of the example

Recognizable pictures REC : *projection of regular CW*

- Originally [Giammarresi & Restivo]: projection of picture languages defined by 2×2 tiles
- **Equivalent Definition**: [Latteux & Simplot]
- $\boxed{REC = \text{projection of local CW} = \text{projection of regular CW}}$
- **Example:** regular CW $P = b^* d\, a^* \boxplus a^* d\, b^*$,

e.g.

$$\begin{array}{|cccc|}\hline d & a & a & a \\ b & d & a & a \\ b & b & d & a \\ b & b & b & d \\ \hline\end{array}$$

$ROW = b^* d\, a^*$ defined by *domino tiles* : $\boxed{\#d}, \boxed{\#b}, \boxed{da}, \boxed{bd}, \boxed{bb}, \boxed{a\#}, \boxed{d\#}$

$COL = a^* d\, b^*$ defined by domino tiles:

$$\left|\begin{matrix}\# \\ d\end{matrix}\right|, \left|\begin{matrix}\# \\ a\end{matrix}\right|, \left|\begin{matrix}d \\ b\end{matrix}\right|, \left|\begin{matrix}a \\ d\end{matrix}\right|, \left|\begin{matrix}a \\ a\end{matrix}\right|, \left|\begin{matrix}b \\ b\end{matrix}\right|, \left|\begin{matrix}b \\ \#\end{matrix}\right|, \left|\begin{matrix}d \\ \#\end{matrix}\right|$$

- the *square pictures* over a *unary* alphabet $\begin{array}{|cccc|}\hline s & s & s & s \\ s & s & s & s \\ s & s & s & s \\ s & s & s & s \\ \hline\end{array}$ are not the CW of regular languages but

are the projection $\left\{\begin{matrix} a \to s \\ b \to s \\ d \to s \end{matrix}\right\}$ of the example

Crossword closure of a picture language

- **Theorem:** for all picture p and picture language L

$$\begin{cases} \overbrace{\mathrm{ROW}(p) \boxplus \mathrm{COL}(p)}^{\textit{crossword closure}} \supseteq \{p\} \\ \mathrm{ROW}(L) \boxplus \mathrm{COL}(L) = L \iff L \text{ is the CW of two word languages} \end{cases}$$

- This permits to prove family non-inclusion : $\boxed{\mathit{REC} \subsetneq \textit{regular CrossWord}}$

- **Witness:** the REC language $L = (a^{++} \oplus b^{++}) \ominus b^{++}$ e.g., $\boxed{\begin{matrix} a & a & a & b \\ a & a & a & b \\ b & b & b & b \end{matrix}}$ but $b^{++} \notin L$

- L is not a Cross Word since $ROW(L) = COL(L) = a^* b^+$ and $b^{++} \in a^* b^+ \boxplus a^* b^+$

- For the same reason $\boxed{\mathit{REC} \subsetneq \text{context-free CW}}$

Crossword closure of a picture language

- **Theorem:** for all picture p and picture language L

$$\begin{cases} \overbrace{\mathrm{ROW}(p) \boxplus \mathrm{COL}(p)}^{\textit{crossword closure}} \supseteq \{p\} \\ \mathrm{ROW}(L) \boxplus \mathrm{COL}(L) = L \iff L \text{ is the CW of two word languages} \end{cases}$$

- This permits to prove family non-inclusion : $\boxed{REC \subsetneq \textit{regular CrossWord}}$

- **Witness:** the REC language $L = (a^{++} \oplus b^{++}) \ominus b^{++}$ e.g., $\boxed{\begin{matrix} a & a & a & b \\ a & a & a & b \\ b & b & b & b \end{matrix}}$ but $b^{++} \notin L$

- L is not a Cross Word since $ROW(L) = COL(L) = a^* b^+$ and $b^{++} \in a^* b^+ \boxplus a^* b^+$

- For the same reason $\boxed{REC \subsetneq \text{context-free CW}}$

Context-Free (CF) Crosswords : points of interest

- a gallery of pictures
- Closure properties of the CF CW family
- Formalize and prove a Chomsky-Schützenberger Theorem (CST) for pictures
- Recall: Chomsky-Schützenberger [1963] :
  a language is CF $\iff$ it is the homomorphic image $L = h\,(D \cap R)$
  with D a Dyck language and R a regular (local) language
- the Dyck alphabet is *bipartite* $\{a_1, a_2, \ldots a_k;\ a'_1, a'_2, \ldots a'_k\}$ with the *coupling* relation
  $a_1 a'_1 \to \varepsilon,\ a_2 a'_2 \to \varepsilon, \ldots, a_k a'_k \to \varepsilon$ a.k.a. cancellation rules
  $w \in D \iff w$ reduces to null by application of cancellation rules
- To generalize CST to context-free CW we need a "good" generalization of the *Dyck language concept for pictures*

Context-Free (CF) Crosswords : points of interest

- a gallery of pictures
- Closure properties of the CF CW family
- Formalize and prove a Chomsky-Schützenberger Theorem (CST) for pictures
- Recall: Chomsky-Schützenberger [1963] :
  a language is CF $\iff$ it is the homomorphic image $L = h\,(D \cap R)$
  with D a Dyck language and R a regular (local) language
- the Dyck alphabet is *bipartite* $\{a_1, a_2, \ldots a_k;\ a'_1, a'_2, \ldots a'_k\}$ with the *coupling* relation
  $a_1 a'_1 \to \varepsilon,\ a_2 a'_2 \to \varepsilon, \ldots,\ a_k a'_k \to \varepsilon$ a.k.a. cancellation rules
  $w \in D \iff w$ reduces to null by application of cancellation rules
- To generalize CST to context-free CW we need a “good” generalization of the *Dyck language concept for pictures*

1st 2D generalization of Dyck languages: Dyck CrossWords

- **Most natural definition to investigate**:

  $$\text{Dyck CW} \stackrel{def}{=} D_1 \boxplus D_2$$

  with D_1 and D_2 Dyck languages over same alphabet

- the tree *syntax structure* of Dyck words shows up in Dyck CWs
- surprising rich patterns are possible in Dyck CWs that are worth studying
- the *cancellation rule* of Dyck words becomes *neutralization rule* for Dyck CW
- however for generalizing CST to context-free CWs, the above definition is *problematic*

The language DC_1 over 4-letter alphabet

- **Definition:** Dyck CW language $DC_1 \subseteq \{a, b, c, d\}^{++}$ with the couplings

$$\begin{cases} Rows: & \text{couplings: } ab \to \varepsilon, cd \to \varepsilon \\ Cols: & \text{couplings: } ac \to \varepsilon, bd \to \varepsilon \end{cases}$$

- **Example :** two pictures in DC_1

a	a	b	b
a	b	a	b
c	d	c	d
c	c	d	d

a	a	a	b	b	b
c	a	b	d	a	b
a	c	d	b	c	d
c	c	c	d	d	d

- Similarly, define DC_k over an alphabet $a_1, b_1, c_1, d_1, \ldots, a_k, b_k, c_k, d_k$ of cardinality $4k$, $k \geq 1$.

- We just focus on DC_1 since properties of DC_k and DC_1 are essentially the same : analogy to Dyck words

First properties of Dyck CW

- **Theorem :** alphabet of Dyck CWs
  1. Dyck CW lang. is non empty only if its alphabet has cardinality ≥ 4
  2. For alphabet $\{a, b, c, d\}$ only one coupling (up to isomorphism) defines a non-empty CW
  3. For every $k \geq 1$, there is a non-empty Dyck CW DC_k with alphabet of cardinality $4k$

- Proof of (1) *Why two letters* $\{a, b\}$ *are not enough*
  Rows: coupling: $ab \to \varepsilon$. *Cols*: coupling: either $ab \to \varepsilon$ or $ba \to \varepsilon$:

case $ab \to \varepsilon$

a	b	...
b	?	...
...	...	

case $ba \to \varepsilon$

a	b	...
?	a	...
...	...	

Both FAIL!

- **Theorem:** Dyck Crosswords $DC_1 \notin REC$. Witness $\overbrace{\begin{array}{cccccc} a & \dots & a & b & \dots & b \\ c & \dots & c & d & \dots & d \end{array}}^{\{a^n\, b^n \geq 1\} \text{ is not regular}}$

First properties of Dyck CW

- **Theorem :** alphabet of Dyck CWs
  1. Dyck CW lang. is non empty only if its alphabet has cardinality ≥ 4
  2. For alphabet $\{a, b, c, d\}$ only one coupling (up to isomorphism) defines a non-empty CW
  3. For every $k \geq 1$, there is a non-empty Dyck CW DC_k with alphabet of cardinality $4k$

- Proof of (1) *Why two letters* $\{a, b\}$ *are not enough*
  Rows: coupling: $ab \to \varepsilon$. *Cols*: coupling: either $ab \to \varepsilon$ or $ba \to \varepsilon$:

case $ab \to \varepsilon$

a	b	$\dots$
b	?	$\dots$
$\dots$	$\dots$	

case $ba \to \varepsilon$

a	b	$\dots$
?	a	$\dots$
$\dots$	$\dots$	

Both FAIL!

- **Theorem:** Dyck Crosswords $DC_1 \notin REC$. Witness

$\{a^n\, b^n \geq 1\}$ is not regular

a	$\dots$	a	b	$\dots$	b
c	$\dots$	c	d	$\dots$	d

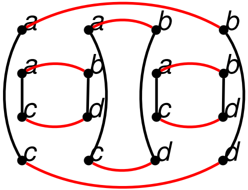
Matching-graph, circuits and 2D syntax structures

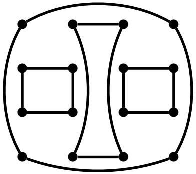
- **Definition:** the *matching graph* of a Dyck CW picture P has one node per pixel
- each node has one horizontal arc and one vertical arc, resp. connecting to the matching letter in the row/column

Theorem : the matching graph is partitioned into ≥ 1 subgraphs, each one a *simple undirected matching circuit*.

Example : Dyck CW picture and its matching graph, partitioned into 4 length 4 circuits

a	*a*	*b*	*b*
a	*b*	*a*	*b*
c	*d*	*c*	*d*
c	*c*	*d*	*d*





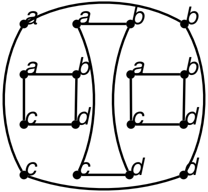
circuits make letters redundant

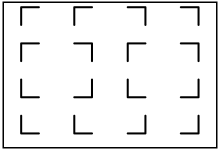
Alternative typography for the alphabet

- the well-nested layout of matching circuits is more apparent with ”corner“ symbols

-

	column open	column closed
row open	$a \to$ ┌	$b \to$ ┐
row closed	$c \to$ └	$d \to$ ┘

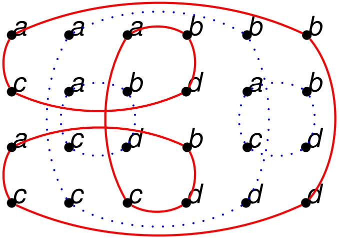




Longer non-rectangular matching circuits

- **Definition:** *quaternate picture* : a Dyck CW picture just containing rectangular matching circuits

- Is it possible to have longer circuits? Yes! E.g., matching graph with a circuit of length 12 and 3 rectangles

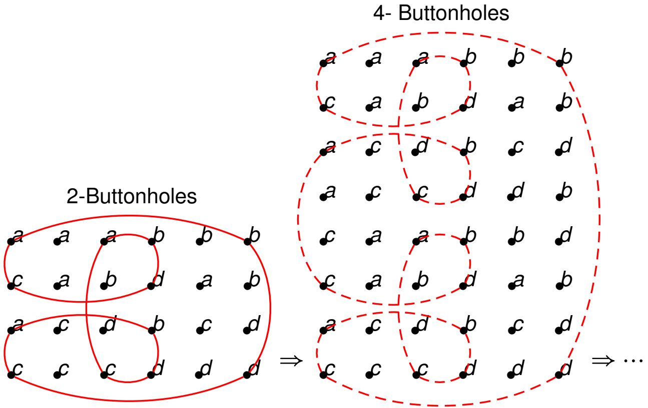


"2-Buttonholes"

- **Theorem**: Circuit length and label
  The length of a matching circuit is multiple of 4 and its label is a circular word in $(abdc)^+$

A family of Dyck CWs with unbounded-length circuits

- The *buttonhole* language is iteratively defined
- Vertically concatenate a 2-buttonhole to itself then adjust some nodes and arcs to connect the length 12 circuits into a length 20 circuit with 4 buttonholes, etc.
- All pictures have 6 columns.
- perhaps surprising **Theorem :** the *buttonhole* language is in REC



Generalizing the Dyck cancellation rule to Dyck CW

- Are there other ways for defining a “natural” family of 2D Dyck languages?
- Dyck word language is defined by cancellation rule such as $ab \to \varepsilon$
- Cancellation inside a picture would create a “hole”, producing an object that is not a picture
- Just rephrase Dyck cancellation rule as *neutralization rule*:

  $ab \to NN$, where N is a new “neutral” letter
- Thus a Dyck word is mapped to a word in N^+ by a series of neutralization steps
- Neutralization can be generalized to quaternate pictures (their circuits are rectangular)!
  Is the result a Dyck CW?

Neutralizable Dyck languages: a “2D definition”

- Neutralization operates on the four outermost matching corners of a (rectangular) *subpicture*
- Neutralization of innermost rectangle:

$$\begin{array}{cc} \ulcorner & \urcorner \\ \llcorner & \lrcorner \end{array} \longrightarrow \begin{array}{cc} N & N \\ N & N \end{array}$$

- Neutralization of a rectangle containing just neutral symbol:

$$\begin{array}{ccc} \ulcorner & N\dots N & \urcorner \\ N & \dots & N \\ \vdots & N\dots N & \vdots \\ N & \dots & N \\ \llcorner & N\dots N & \lrcorner \end{array} \longrightarrow \begin{array}{ccc} N & N\dots N & N \\ N & \dots & N \\ \vdots & N\dots N & \vdots \\ N & \dots & N \\ N & N\dots N & N \end{array}$$

- A picture is *neutralizable* if the procedure halts with the result in N^{++}.
- Neutralizable pictures nicely generalize Dyck: contain only well-nested rectangular patterns

Example of neutralizable picture

$$
p = \begin{array}{cccccc}
\ulcorner & \ulcorner & \urcorner & \ulcorner & \urcorner & \urcorner \\
\ulcorner & \ulcorner & \urcorner & \llcorner & \lrcorner & \urcorner \\
\llcorner & \llcorner & \lrcorner & \ulcorner & \urcorner & \lrcorner \\
\llcorner & \llcorner & \lrcorner & \llcorner & \lrcorner & \lrcorner
\end{array}
\xrightarrow{\nu}
\begin{array}{cccccc}
\ulcorner & \ulcorner & \urcorner & \ulcorner & \urcorner & \urcorner \\
\ulcorner & N & N & \llcorner & \lrcorner & \urcorner \\
\llcorner & N & N & \ulcorner & \urcorner & \lrcorner \\
\llcorner & \llcorner & \lrcorner & \llcorner & \lrcorner & \lrcorner
\end{array}
\xrightarrow{\nu}
\begin{array}{cccccc}
\ulcorner & N & N & \ulcorner & \urcorner & \urcorner \\
\ulcorner & N & N & \llcorner & \lrcorner & \urcorner \\
\llcorner & N & N & \ulcorner & \urcorner & \lrcorner \\
\llcorner & N & N & \llcorner & \lrcorner & \lrcorner
\end{array}
$$

$$
\xrightarrow{\nu}
\begin{array}{cccccc}
\ulcorner & N & N & N & N & \urcorner \\
\ulcorner & N & N & N & N & \urcorner \\
\llcorner & N & N & \ulcorner & \urcorner & \lrcorner \\
\llcorner & N & N & \llcorner & \lrcorner & \lrcorner
\end{array}
\xrightarrow{\nu}
\begin{array}{cccccc}
\ulcorner & N & N & N & N & \urcorner \\
\ulcorner & N & N & N & N & \urcorner \\
\llcorner & N & N & N & N & \lrcorner \\
\llcorner & N & N & N & N & \lrcorner
\end{array}
\xrightarrow{\nu}
\begin{array}{cccccc}
\ulcorner & N & N & N & N & \urcorner \\
N & N & N & N & N & N \\
N & N & N & N & N & N \\
\llcorner & N & N & N & N & \lrcorner
\end{array}
\xrightarrow{\nu}
\begin{array}{cccccc}
N & N & N & N & N & N \\
N & N & N & N & N & N \\
N & N & N & N & N & N \\
N & N & N & N & N & N
\end{array}
$$

- Every row and column is neutralized, i.e., rows and columns are Dyck words. Since Dyck CW contains also longer circuits:
- **Theorem:** neutralizable Dyck pictures $\subsetneq$ Dyck CW

Quaternate vs. neutralizable pictures

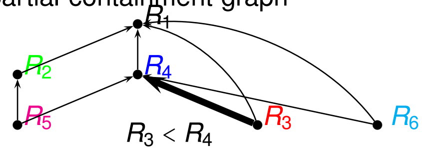
- A *quaternate* picture is a Dyck CW s.t. all its circuits are rectangles.
  All neutralizable pictures are quaternate. Does the vice versa hold?

- **Definition:** the binary partial-containment relation $R_1 < R_2$ holds between two rectangles, s.t. some nodes of R_1 are inside or on a side of R_2.

- **Example** : the rectangles are identified by colors/numbering:

⌜1	⌜2	⌝2	⌜**3**	⌝**3**	⌝1
⌜**4**	⌜5	⌝5	⌞**3**	⌟**3**	⌝**4**
⌞**4**	⌞5	⌟5	⌜6	⌝6	⌟**4**
⌞1	⌞2	⌟2	⌞6	⌟6	⌟1

partial-containment graph

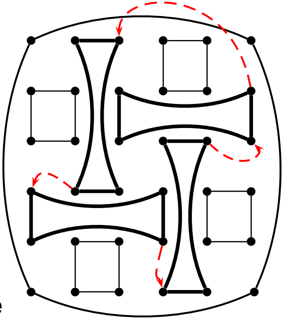


- **Definition:** *A quaternate picture is partially-ordered if its partial-containment relation is acyclic*
  The above picture is partially ordered

Theorem: A quaternate picture is neutralizable $\iff$ it is partially ordered

Examples: The preceding picture is neutralizable; the following picture is not:

Quaternate picture with
cyclic partial-inclusion
relation hence non-neutralizable



Good closure properties of context-free CW with projection

1. In general the CW of a family $\mathcal{F}$ of word languages does not preserve closure under concatenations and union
2. The application of projection after the CW operation $\boxplus$ preserves more closure properties
   Let $\mathcal{F}$ be closed under *disjoint union* –e.g. *local, regular and context-free families*
   1. **Theorem :** the projection of the CW of $\mathcal{F}$, for brevity $Proj \circ CW(\mathcal{F})$, is closed under **union**
   2. if $\mathcal{F}$ is closed under disjoint concatenation then $Proj \circ CW(\mathcal{F})$ is closed under both row and column **concatenations**
   3. if $\mathcal{F}$ is closed under disjoint concatenation closure, then $Proj \circ CW(\mathcal{F})$ is closed under both column and row **concatenation closures**
   4. if $\mathcal{F}$ is closed under inverse alphabetic homomorphism and under intersection with local languages, then $Proj \circ CW(\mathcal{F})$ is closed under **intersection** with the *REC* family;
   5. if $\mathcal{F}$ is closed under disjoint finite substitution, then $Proj \circ CW(\mathcal{F})$ is closed under picture homomorphism
      (the picture blow-up obtained by mapping each pixel on a finite picture with uniform size)

Chomsky-Schützenberger Theorem CST for pictures

Background
Chomsky-Schützenberger Theor. CST 1963:

Non-erasing CST variants 1996,2012,2019

generalizes to

CST for the projection of context-free CW

Context-free CW

Dyck CW is INADEQUATE for CST

replace by Dyck Cartesian CW

Dyck CW is unsuitable as generator for CST

- Intuitively we would like to formalize and then prove a statement such as:

$$\text{a picture language } L \subseteq \Sigma^{++} \textit{ is a context-free CW } \iff$$

$$L = h\Big(\overbrace{[2D \textit{ Dyck language}]}^{D} \cap \overbrace{[\textit{local} \mid \textit{slt picture language}]}^{R} \Big)$$

- Let us try with the natural choices: $\begin{cases} D \subseteq \Delta^{++} & \text{where } D \in \text{ Dyck } CW \\ R \subseteq \Delta^{++} & \text{where } R \in \text{ local } CW \\ \text{projection} h : \Delta \to \Sigma \end{cases}$

- in particular, for any choice of (non-empty) languages D, R and for any projection h, the language $h(D \cap R)$ should be a non-empty context-free CW

- this fails if we take: $\begin{cases} \text{Dyck word languages } D_1, D_2 \subseteq \{a, a'\}^* \\ D = D_1 \boxplus D_2 \text{ Dyck CW} \\ R = \Delta^{++} \text{ and the identity function for } h \end{cases}$

  since we know $D_1 \boxplus D_2$ is empty for any alphabet of size less then 4

From Cross-Words to Cartesian Cross-Words, CCW

Σ_1, Σ_2 (possibly not distinct) alphabets

$\Sigma_1 \times \Sigma_2$ the Cartesian alphabet

Cartesian Product $P_1 \times P_2$ of two pictures

$P_1 \subseteq \Sigma_1^{++}, P_2 \subseteq \Sigma_2^{++}$ of same size

e.g. : $\boxed{\begin{matrix} a & b \\ c & d \end{matrix}} \times \boxed{\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}} \stackrel{\text{def}}{=} \boxed{\begin{matrix} a,1 & b,2 \\ c,3 & d,4 \end{matrix}}$

- Let $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$
- **Definition:** the *Cartesian crossword*, denoted as $L_1 \boxtimes L_2 \subseteq (\Sigma_1 \times \Sigma_2)^{++}$, is the language

$$L_1 \boxtimes L_2 \stackrel{\text{def}}{=} (L_1)^{\ominus +} \times (L_2)^{\oplus +}$$

i.e. the *Cartesian product* of the row-concatenation-closure of L_1 and the column-concatenation-closure of L_2

Cartesian Cross-Words of Dyck languages, Dyck CCW

- Let D_1, D_2 be Dyck languages over alphabets Δ_1 , Δ_2
  The product $\Gamma = \Delta_1 \times \Delta_2$ is called *Cartesian Dyck alphabet*.

- **Example** D_1 over $\Delta_1 = \{ \overbracket{a, a'} , \overbracket{b, b'} \}$ $\qquad$ D_2 over $\Delta_2 = \{ \overbracket{a, a'} , \overbracket{b, b'} , \overbracket{c, c'} , \overbracket{d, d'} \}$

a picture in $D_1 \boxtimes D_2$

ac	$a'b$	bd	$b'a$	aa	$a'b$
ac	aa	aa	$a'b$	$a'b$	$a'd$
ac'	ba'	ba'	$b'b'$	$b'b'$	$a'd'$
ac'	ab'	$a'd'$	$a'a'$	ba'	$b'b'$

corners are explanatory: e.g. $\llcorner$ is open on row and closed on column

$\ulcorner ac$	$\urcorner$ **$a'b$**	$\ulcorner bd$	$\urcorner b'a$	$\ulcorner aa$	$\urcorner$ **$a'b$**
$\ulcorner ac$	$\ulcorner aa$	$\ulcorner aa$	$\urcorner a'b$	$\urcorner a'b$	$\urcorner a'd$
$\llcorner ac'$	$\llcorner ba'$	$\llcorner ba'$	$\lrcorner b'b'$	$\lrcorner b'b'$	$\lrcorner a'd'$
$\llcorner ac'$	$\llcorner$ **ab'**	$\lrcorner a'd'$	$\lrcorner a'a'$	$\llcorner ba'$	$\lrcorner$ **$b'b'$**

- the red columns violate the Dyck condition on Cartesian Dyck alphabet since (a', b) matches both (a, b') and (b', b') !

- N.B. The row and column languages (over alphabet Γ) are *Visibly Push-Down*

Essential properties of CCW

$$\begin{tikzcd}[column sep=small] (& a \arrow[d, "\pi_1"'] & , & b \arrow[d, "\pi_2"] &) \\ & a & & b & \end{tikzcd}$$

- Define projections from Cartesian alphabets to components
- **Identity:** $L_1 \boxtimes L_2 = \pi_1^{-1}(L_1) \boxplus \pi_2^{-1}(L_2)$
  i.e. the crossword of the inverse projections of the two word languages
- since the lang. family, say *context-free*, of L_1 and L_2 is closed under inverse homomorphism, *$L_1 \boxtimes L_2$ is a context-free CW over the Cartesian alphabet* $\Sigma_1 \times \Sigma_2$

Essential difference

Cartesian CW	CW
$\pi_1(ROW(L_1 \boxtimes L_2)) = L_1$ and $\pi_2(COL(L_1 \boxtimes L_2)) = L_2$	$ROW(L_1 \boxplus L_2) \subseteq L_1$ and $COL(L_1 \boxplus L_2) \subseteq L_2$
CCW of non-empty languages never empty	CW may be empty

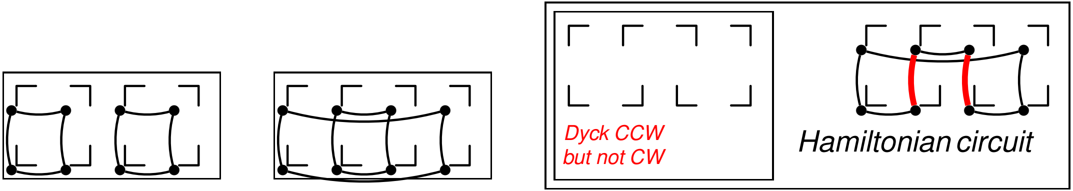
Matching-circuit graphs of Dyck Cartesian CW vs Dyck CW

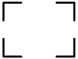
- Matching circuits of Dyck CW naturally generalize to Dyck Cartesian CW:

$$(a,\ b)\ \ (a',\ x)$$

$$(y,\ b')$$

- Valid 2×4 pictures in Dyck CW and Dyck Cartesian CW



- **Theorem:** the only Hamiltonian picture in Dyck CW is 

Properties of Cartesian Cross-Words needed to prove CST

1. **Distributive property of Cartesian CW operation over intersection**
   Let $L_1, L_3 \subseteq \Sigma_1^*$ and $L_2, L_4 \subseteq \Sigma_2^*$, then
   $\boxed{(L_1 \boxtimes L_2) \cap (L_3 \boxtimes L_4) = (L_1 \cap L_3) \boxtimes (L_2 \cap L_4)}$ When intersecting Dyck with a regular language remains context-free!

2. **Restricting the Cartesian alphabet to the diagonal reduces CCW to CW**

   Let $\Sigma_1 = \Sigma_2 = \Sigma$. The *diagonal alphabet* is $\Sigma_{1=2} \stackrel{def}{=} \{ (a, a) \mid a \in \Sigma \}$

   $\boxed{L_1 \boxplus L_2 = \pi((L_1 \boxtimes L_2) \cap \Sigma_{1=2}^{++})}$ where $\pi : \Sigma_{1=2} \to \Sigma$ is the projection $\pi((a, a)) = a$

Non-erasing variants of Chomsky-Schützenberger Theorem CST

- *Original CST [1963]* $L \subseteq \Sigma^*$ is CF $\iff$ $L = h(D \cap R)$, $h : \Delta^* \to \Sigma^*$ with $h(a) \in \Sigma \cup \{\varepsilon\}$ erasing homomorphism
- erasing makes a picture undefined, we need: *non-erasing CST* [Berstel 1996, Okhotin 2012]
- In such erasing CST statements the Dyck alphabet size depends on the language (i.e. grammar) complexity, which is inconvenient for proving CST for pictures.
- We use the non-erasing CST [CR-SP 2019] where the Dyck alphabet size just depends on the alphabet size $|\Sigma|$
- obviously non-erasing CSTs only apply to words of even length. For odd-length , we add to the Dyck alphabet a finite set of *neutral symbols*. We omit for brevity

CST representation and characterization theorems for *context-free Cross Words*

Representation theorem

Let a picture $P \subseteq \Sigma^{++}$ be a *context-free CW* $\Longrightarrow$

Characterization theorem

A picture language $P \subseteq \Sigma^{++}$ is the *projection of a context-free CW* $\Longleftrightarrow$

there exist

- a Cartesian Dyck alphabet Γ
- a *Dyck CCW* language D_{CCW} over Γ
- a *strictly-locally-testable CW* language R over Γ
- a projection $h : \Gamma \to \Sigma$

such that

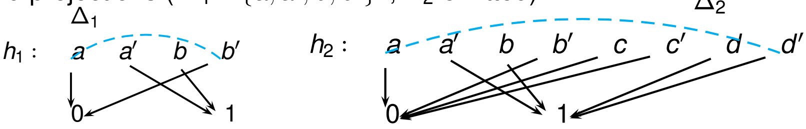
$$P = h(D_{CCW} \cap R)$$

Steps of Proof of Representation Theorem

- For $i = 1\,,\,2$: $L_i = h_i(D_i \cap R_i)$, with L_i, over Σ and D_i, R_i over Δ. Let $\Sigma_{1=2} \stackrel{def}{=} \{(a,a) \mid a \in \Sigma\}$
- $L_1 \boxplus L_2 = \pi\left((L_1 \boxtimes L_2) \cap \Sigma^{++}_{1=2}\right)$
  - $L_1 \boxtimes L_2 = h_1(D_1 \cap R_1) \boxtimes (h_2(D_2 \cap R_2)$
  - $= \rho\left((D_1 \cap R_1) \boxtimes (D_2 \cap R_2)\right)$ where $\rho(\langle a,b\rangle) \stackrel{def}{=} \langle h_1(a), h_2(b)\rangle$. By distributivity:
  - $L_1 \boxtimes L_2 = \rho\left((D_1 \boxtimes D_2) \cap (R_1 \boxtimes R_2)\right)$
- $L_1 \boxplus L_2 = \pi\left(\rho\left((D_1 \boxtimes D_2) \cap (R_1 \boxtimes R_2)\right) \cap \Sigma^{++}_{1=2}\right)$ recall $\pi\left((a,a)\right) = a$
- $= \ldots\ldots$ $L_1 \boxplus L_2 = \pi\left(\rho\left((D_1 \boxtimes D_2) \cap (R_1 \boxtimes R_2)\right) \cap \Gamma^{++}_{agree}\right)$ where $\Gamma_{agree} \stackrel{def}{=} \{(a,b) \mid h_1(a) = h_2(b)\}$ i.e. the subset of $\Delta \times \Delta$ where projections agree
- The theor. follows with $D_{CCW} = D_1 \boxtimes D_2$, the composition $h = \pi \circ \rho$ and $R = (R_1 \boxtimes R_2) \cap \Gamma^{++}_{agree}$

Example. CST construction for context-free CW $L = L_1 \boxplus L_2$

- $$L = \overbrace{\{x \mid \text{ number of 1 } = \text{ number of 0 in } x\}}^{L_1 = h_1(D_1 \cap R_1)} \boxplus \overbrace{(\{y \mid \text{ number of 1 } \geq \text{ number of 0 in } y\} \cup}^{L_2 = h_2(D_2 \cap R_2)}$$
- Alphabets and projections ($R_1 = \{a, a', b, b'\}^*$, R_2 omitted)

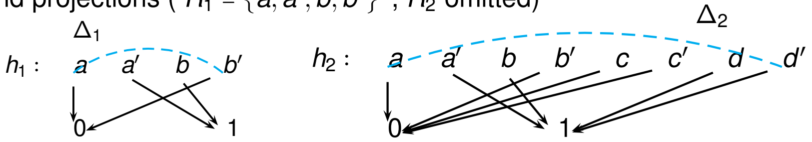


0	1	1	0	0	1
0	0	0	1	1	1
0	1	1	0	0	1
0	0	1	1	1	0

picture over $\Sigma = \{0, 1\}$

Example. CST construction for context-free CW $L = L_1 \boxplus L_2$

- Alphabets and projections ($R_1 = \{a, a', b, b'\}^*$, R_2 omitted)

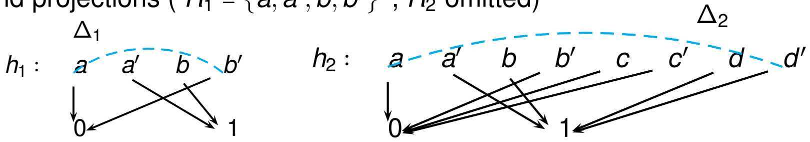


a	c	0	a'	b	1	b	d	1	b'	a	0	a	a	0	a'	b	1
a	c'	0	a	a	0	a	a	0	a'	b	1	a'	b	1	a'	d	1
a	c'	0	b	a'	1	b	a'	1	b'	b'	0	b'	b'	0	a'	d'	1
a	c'	0	a	b'	0	a'	d'	1	a'	a'	1	b	a'	1	b'	b'	0

picture preimage over $\Delta_1 \times \Delta_2$.
N.B. couple (a, b) may not occur since $h_1(a) \neq h_2(b)$

Example. CST construction for context-free CW $L = L_1 \boxplus L_2$

- Alphabets and projections ($R_1 = \{a, a', b, b'\}^*$, R_2 omitted)

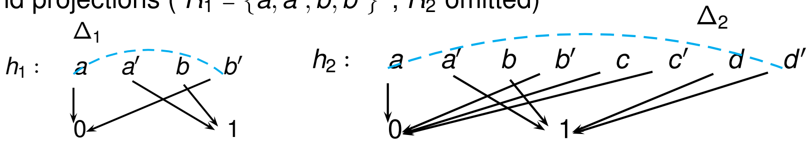


a	c	0	a'	b	1	b	d	1	b'	a	0	a	a	0	a'	b	1
a	c'	0	a	a	0	a	a	0	a'	b	1	a'	b	1	a'	d	1
a	c'	0	b	a'	1	b	a'	1	b'	b'	0	b'	b'	0	a'	d'	1
a	c'	0	a	b'	0	a'	d'	1	a'	a'	1	b	a'	1	b'	b'	0

some Dyck matching arcs of D_1 and D_2

Example. CST construction for context-free CW $L = L_1 \boxplus L_2$

- Alphabets and projections ($R_1 = \{a, a', b, b'\}^*$, R_2 omitted)



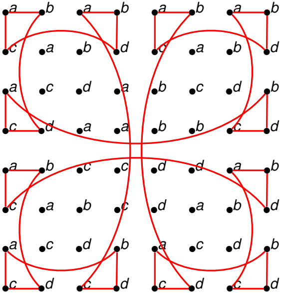
a	c	0	a'	b	1	b	d	1	b'	a	0	a	a	0	a'	b	1
a	c'	0	a	a	0	a	a	0	a'	b	1	a'	b	1	a'	d	1
a	c'	0	b	a'	1	b	a'	1	b'	b'	0	b'	b'	0	a'	d'	1
a	c'	0	a	b'	0	a'	d'	1	a'	a'	1	b	a'	1	b'	b'	0

All projections h_1 red and h_2 green agree on target. Then h projects e.g., (a, c) to 0

The theor. follows with $D_{CCW} = D_1 \boxtimes D_2$, the composition $h = \pi \circ \rho$ and $R = (R_1 \boxtimes R_2) \cap \Gamma^{++}_{agree}$

Some specular and palindromic symmetries in context-free CWs

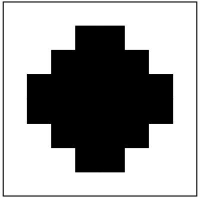
Iterated butterfy in Dyck CW



HV-palindrome [Mahalingam Pandoh 2022]
$L_{pal} \boxplus L_{pal}$ with $L_{pal} = \{ww^R \mid w \in \{0,1\}^+\}$

0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1
1	0	1	1	1	1	0	1
1	0	1	1	1	1	0	1
1	0	0	0	0	0	0	1
0	1	1	1	1	1	1	0

HV-convex polyomino [Barcucci et al. 1996]
$L_{HV} \boxplus L_{HV}$ with $L_{HV} = \{0^n 1^+ 0^n \mid n \geq 0\}$



- Through the CST, pictures inherit the matching graph, i.e., syntax structure, of Dyck CWs
- Patterns in (projective) CF crosswords are intriguing and remain to be classified

Hints to future work

- Cross Words of CF languages: another definition of *context-freeness* for pictures.
  Closer comparisons with past proposals often based on grammars [Siromoney, Matz, Prusa, Cherubini, Pradella, ...] remain to be done
- The CST nicely generalizes to pictures the duality *regular* vs *context-free* corresponding to the historical Medvedev vs Chomsky-Schützenberger theorems
  Other CF properties (beyond the closures investigated) are preserved?
- More research problems are easy to imagine , e.g. how CF subfamilies (say linear) maps on CW
- Dyck words represent the *Last-In-First-Out linear-time* model of procedure *calls* and *returns*.
  Which is the model induced by Dyck CWs? Is it a branching-time model?

Applications are non-inexistent!
Maung & Crawfis, Applying formal picture languages to procedural content generation, IEEE CGAMES 2015

Bibliography

Crespi-Reghizzi, Restivo, and San Pietro. Row-column combination of Dyck words. SOFSEM 2024: LNCS, 2024. [to appear on Acta Informatica]

Crespi-Reghizzi, Restivo, and San Pietro. From words to pictures: Row-column combinations and Chomsky-Schützenberger theorem. Theor. Comput. Sci., 2024.

Fenner, Padé, and Thierauf. The complexity of regex crosswords. Inf. Comput., 2022.

Giammarresi and Restivo. Two-dimensional languages. In G. Rozenberg and A. Salomaa, editors, Handbook of formal languages, vol. 3, 1997.

Latteux and Simplot. Recognizable picture languages and domino tiling. Theor. Comput. Sci., 1997.

Crespi-Reghizzi and San Pietro. Non-erasing Chomsky-Schützenberger theorem with grammar-independent alphabet. Inf. Comput., 2019.

Barcucci, Del Lungo, Nivat, and Pinzani. Reconstructing convex polyominoes from horizontal and vertical projections. Theor. Comput. Sci., 1996.

Mahalingam and Pandoh. HV-palindromes in two-dimensional words. Int. J. Found. Comput. Sci., 2022.

Many thanks to Luca and Rogério !

Ciao , Buone Feste !

abstract - Crosswords of formal languages

The definition of crosswords as row-column combinations of words over an alphabet is applied to regular and context-free (CF) languages, thus producing picture (2D) languages.
The letter-to-letter projection of regular crosswords coincides with the well-known family of pictures recognizable by tiling systems.
Recent results for the CF case, and especially for the Dyck languages, are presented, that culminate in a generalized Chomsky-Schützenberger Theorem (CST) for CF crosswords.
The CST represents the family of pictures defined by context-free crosswords, while it fully characterizes the more general family where the crossword is applied to CF languages over two alphabets, whose Cartesian product becomes the picture alphabet.
Dyck crosswords exhibit a rich spectrum of 2D patterns that combine the syntax trees of their CF components.
The simpler Dyck crosswords subfamilies generalize in 2D the well-nesting property of Dyck words.