2D Picture Languages

Zbyněk Křivka
krivka@fit.vutbr.cz
Brno University of Technology
Faculty of Information Technology
Czech Republic

Talk at Formal Model Research Group Seminar, FIT, BUT
March 27, 2014
Outline

Introduction

Definitions and Examples

Survey

Results
Introduction
Motivation

- **Picture** = rectangular two-dimensional (2D) array of symbols
- picture analysis (structure), picture recognition
- tiling patterns, floor designs
Picture-defining Devices

- Language/picture properties/operations
  - 2D regular expressions
  - Logic formulas (first-order and monadic second-order)
- Accepting devices
  - Four-way automata
  - 2D (on-line) tessellation automata (variant of cellular automata)
- 2D grammars
  - Isometric - geometric shape of the rewritten portion is preserved
    - Array grammars (replaces block of the same size)
  - Non-isometric - can alter the geometric shape
    - Siromoney Matrix Grammars
    - "Image Grammars"
**Picture**

**Picture** (2D array, picture array) \( p \) is a rectangular \( m \times n \) array over \( \Sigma \) of the form

\[
p = \begin{bmatrix}
p(1, 1) & \cdots & p(1, n) \\
\vdots & \ddots & \vdots \\
p(m, 1) & \cdots & p(m, n)
\end{bmatrix}
\]

- where each \( p(i, j) \in \Sigma \) (pixel), \( 1 \leq i \leq m, 1 \leq j \leq n \).
- \( |p|_{\text{row}}, |p|_{\text{col}} \) denote the number of rows/columns of \( p \).

- \( \Sigma^{**} = \) set of all rectangular arrays over \( \Sigma \) (\( \lambda \) for empty picture).
- \( \Sigma^{**} = \Sigma^{**} - \{\lambda\} \)
- A picture language \( L \subseteq \Sigma^{**} \)
Operations

- Block (sub-picture)
- Boundary symbol \( \# \notin \Sigma \).

Picture/Language Operations

- Projection by mapping \( \pi : \Gamma \rightarrow \Sigma \), where \( \Gamma, \Sigma \) are alphabets.
- Column concatenation of two pictures \( (p \ominus q) \) requires the same number of rows.
- Row concatenation of two pictures \( (p \Theta q) \) requires the same number of columns.
- Column/Row closure \( L^{\ominus} \) and \( L^{\Theta} \) such that \( L^{**} = (L^{\ominus})^{\Theta} = (L^{\Theta})^{\ominus} \)
- Clock-wise rotation of a picture \( (p^R) \)
Definitions and Examples
2D Regular Expressions

Recursive definition over alphabet $\Sigma$

- Atomic languages: the empty language $\emptyset$, $\{a\}$ with $a \in \Sigma$.
- 2D Regular operations $\mathcal{R} = \{\Theta, \bigoplus, \ast\Theta, \ast\bigoplus, \cup, \cap, c\}$.
- The result of $\odot \in \mathcal{R}$ applied to regular 2D language is a regular 2D language.
- Family: RE
- Modifications: complement-free RE (CFRE), star-free RE (SFRE), projection of CFRE (PCFRE)
Let $\Sigma = \{\blacksquare, \square\}$

2D regular expression over $\Sigma$: $\left((\blacksquare \oplus \square)^{*}\right) \uplus \left((\square \oplus \blacksquare)^{*}\right)^{*} \ominus$
2D Regular Expressions - Example

- Let $\Sigma = \{■, □\}$
- 2D regular expression over $\Sigma$: $((\text{■} \ominus \text{□})^* \ominus \text{□} \ominus (\text{□} \ominus \text{■})^*)^\oplus$

Figure: A rectangular "chessboard" with even side-length
4-way Automata

Extension of finite automata for 2D (Blum, Hewitt 1967)

**Definition 1.**

Non-deterministic (deterministic) 4-way finite automaton (4NFA, 4DFA) is a 7-tuple \( \mathcal{A} = (\Sigma, Q, \Delta, q_0, q_a, q_r, \delta) \) where

- \( \Delta = \{R, L, U, D\} \) is a set of directions;
- \( q_a, q_r \in Q \) are accepting and rejecting state;
- \( \delta: Q \setminus \{q_a, q_r\} \times \Sigma \to 2^{Q \times \Delta} \) (\( \delta: Q \setminus \{q_a, q_r\} \times \Sigma \to Q \times \Delta \)) is the transition function.

- Starting at position \((1,1)\) in \(q_0\), finishing in \(q_a\) or \(q_r\) (need not to read whole picture)
- "Border sensitive"
Example 2.

Let $\Sigma = \{0, 1\}$, $L_1 \subseteq \Sigma^{**}$ consists of square pictures.  
4DFA $\mathcal{A}_1$ works in the following way:
Example 2.

Let $\Sigma = \{0, 1\}$, $L_1 \subseteq \Sigma^{**}$ consists of square pictures. 4DFA $A_1$ works in the following way:

- Moves along the diagonal until the bottom-right corner $\Rightarrow$ square.
- Checks that all positions contain a symbol from $\Sigma$. 
4-way Automata - Example

Example 3.
Let $\Sigma = \{0, 1\}$, $L_2 \subseteq \Sigma^{**}$ consists of square pictures of odd side-length with "1" in the central position.

4NFA $\mathcal{A}_2$ works in the following way:
Example 3.

Let $\Sigma = \{0, 1\}$, $L_2 \subseteq \Sigma^{**}$ consists of square pictures of odd side-length with "1" in the central position.

$4\text{NFA } \mathcal{A}_2$ works in the following way:

- Moves along the diagonal (one step right, one step down).
- It non-deterministically chooses a point where a symbol is checked to be 1.
- Continue downwards but to the bottom-left corner.
Example 3.
Let $\Sigma = \{0, 1\}$, $L_2 \subseteq \Sigma^{**}$ consists of square pictures of odd side-length with "1" in the central position.
4NFA $\mathcal{A}_2$ works in the following way:
- Moves along the diagonal (one step right, one step down).
- It non-deterministically chooses a point where a symbol is checked to be 1.
- Continue downwards but to the bottom-left corner.

Theorem 4.
The family of 4DFA is strictly included in 4NFA.
2D Right-Linear Grammar

Definition 5.
A 2D right-linear grammar (2DRLIN, [1]) is a 7-tuple

\[ G = (V_h, V_v, \Sigma_I, \Sigma, S, R_h, R_v) \]

where

- \( V_h \) and \( V_v \) is a finite set of horizontal and vertical nonterminals;
- \( \Sigma_I \subseteq V_v \) and \( \Sigma \) is a finite set of intermediates and terminals;
- \( S \in V_h \) is a starting symbol;
- \( R_h \) is a finite set of horizontal rules:
  \[ V \rightarrow AV' \text{ or } V \rightarrow A \text{ where } V, V' \in V_h \text{ and } A \in \Sigma_I; \]
- \( R_v \) is a finite set of vertical rules:
  \[ A \rightarrow aA' \text{ or } A \rightarrow a \text{ where } A, A' \in V_v \text{ and } a \in \Sigma. \]

First, generate string \( w \in \Sigma_I \) by \( R_h \).
Second, build a picture by \( R_v \) in the downward direction.
Local 2D Languages (LOC)

\[ B_{h,k}(p) = \text{the set of all blocks of } p \text{ of size } (h, k), \text{ where } h \leq m, k \leq n. \]

**Definition 6.**
Let \( \Gamma \) be an alphabet. A 2D language \( L \subseteq \Gamma^{**} \) is **local** if there exists a finite set \( \Phi \) of **tiles** over \( \Gamma \cup \{\#\} \) s.t. \( L = \{p \in \Gamma^{**} | B_{2,2}(p) \subseteq \Phi\} \).

- \( \Phi \) is the set of **allowed blocks** or **representation by tiles** including \( \# \).
- \( \lambda \in L(\Phi) \) iff \( \begin{array}{cccc} \# & \# \\ \# & \# \end{array} \in \Phi \)
- The family: LOC
Local 2D Languages (LOC) - Example

Example 7.

\[
\Phi = \begin{cases}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & # & 1 & # & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & # & 0 & # & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & # & 0 & # & # & # & # & # \\
# & # & # & # & 1 & # & 0 & # & 1 & 0 & # & # \\
# & # & # & # & # & 0 & 1 & # & # & # & # & # \\
# & 1 & 1 & 0 & # & # & # & # & # & # & # & # & # 
\end{cases}
\]

- \( L(\Phi) \) contains squares with 1s on the main diagonal positions; otherwise 0.
- Observe that no square language is a local 2D language over unary alphabet.
- Generalization: \((h, k)\)-local 2D languages, i.e. LOC is \((2, 2)\)-local 2D language.
Tiling Recognizable Languages

Definition 8.
A tiling system \((TS)\) is 4-tuple \(T = (\Sigma, \Gamma, \Phi, \pi)\), where

- \(\Sigma\) and \(\Gamma\) are two alphabets;
- \(\Phi\) is finite set of tiles over \(\Gamma \cup \#\);
- \(\pi: \Gamma \rightarrow \Sigma\) is a projection.

- \(L\) recognizable by TS \(T\): \(L(T) = \pi(L')\) where \(L' = L(\Phi) \in LOC\).
- The family: TS or REC
- Domino system works with \(B_{1,2}(\hat{p})\) and \(B_{2,1}(\hat{p})\) but DS = TS.

Example 9.
Take previous example \(L(\Phi)\) with \(\Gamma = \{0, 1\}\) and \(\pi(0) = \pi(1) = a\).

Theorem 10.
\(LOC \subset TS\)
Definition 11.
A pure 2D context-free grammar \((P2DCFG, [2])\) is a 4-tuple

\[
G = (\Sigma, P_1, P_2, M_0)
\]

where

i) \(\Sigma\) is a finite alphabet of symbols;

ii) \(P_1 = \{c_i| 1 \leq i \leq s_c\}\), where \(c_i\) is called a column rule table, \(s_c \geq 0\); each \(c_i\) is a finite set of CF rules: \(a \rightarrow \alpha, a \in \Sigma, \alpha \in \Sigma^*\) s.t. for any \(a \rightarrow \alpha, b \rightarrow \beta\) in \(c_i\), \(|\alpha| = |\beta|\);

iii) \(P_2 = \{r_j| 1 \leq j \leq s_r\}\), where \(r_j\), is called a row rule table, \(s_r \geq 0\); each \(r_j\) is a finite set of CF rules: \(c \rightarrow \gamma^R, c \in \Sigma, \gamma \in \Sigma^*\) s.t. for any \(c \rightarrow \gamma^R, d \rightarrow \delta^R\) in \(r_j\), \(|\gamma| = |\delta|\);

iv) \(M_0 \subseteq \Sigma^{**} - \{\lambda\}\) is a finite set of axiom arrays.
Pure 2D Context-Free Grammars - Derivation

A derivation in a \textit{P2DCFG} \( G \) is defined as follows: Let \( p, q \in \Sigma^{**} \).

\[
p \Rightarrow q
\]

i) either by rewriting in parallel all the symbols in a column of \( p \), each symbol by a rule in some column rule table

ii) or rewriting in parallel all the symbols in a row of \( p \), each symbol by a rule in some row rule table.

All the rules used to rewrite a column (or row) have to belong to the same table.

- **Picture language:** \( L(G) = \{ M \in \Sigma^{**} | M_0 \Rightarrow^* M \text{ for some } M_0 \in M_0 \} \).
- **The family:** \( P2DCFL \).
**Pure 2D Context-Free Grammars - Example**

**Example 12.**

$P2DCFG \ G_1 = (\Sigma, P_1, P_2, \{M_0\})$ where $\Sigma = \{a, b, e\}$, $P_1 = \{c\}$, $P_2 = \{r\}$, where

\[ c = \{a \rightarrow bab, e \rightarrow aea\}, \quad r = \left\{ e \rightarrow \begin{array}{c} e \\ a \end{array}, a \rightarrow \begin{array}{c} a \\ b \end{array} \right\}, \quad M_0 = \begin{array}{ccc} a & e & a \\ b & a & b \end{array} \]

$L(G_1) =$ pictures of size $(m, 2n + 1)$, $m \geq 2$, $n \geq 1$.

![Figure: A picture in $L(G_1)$](image-url)
Definition 13.
A Controlled $P2DCFG$ is $G^c = (G, C)$ where

- $G = (\Sigma, P_1, P_2, M_0)$ is a $P2DCFG$,
- $C \subseteq (P_1 \cup P_2)^*$ is a control language (regular or context-free) consisting of control strings over labels of tables.

- Derivations $M_1 \Rightarrow_w M_2$ in $G^c$ as in $G$ except that if $w \in (P_1 \cup P_2)^*$ and $w = l_1l_2 \ldots l_m$, then the tables of rules with labels $l_1, l_2, \ldots, l_m$ are successively applied starting with $M_1$ to finally yield $M_2$.

- The families: $(R)P2DCFL$ and $(CF)P2DCFL$
Definition 14.

- A \((l/u)P2DCFG\) is \(P2DCFG\) \(G = (\Sigma, P_1, P_2, M_0)\) with \(\Rightarrow(l/u)\) derivations.
- \(M_1 \Rightarrow(l/u) M_2\) means only the leftmost column or the uppermost row of \(M_1\) is rewritten.
- The family: \((l/u)P2DCFL\)
Example 15.

\((l/u)\text{P2DCFG} \ G_2 = (\Sigma, P_1, P_2, \{M_0\})\) where \(\Sigma = \{a, b\}\), \(P_1 = \{c\}\), \(P_2 = \{r\}\) with

\[
c = \{a \to ab, b \to ba\}, \quad r = \left\{ \begin{array}{ll} a \to & a \\ b \to & b \end{array} \right\} \quad M_0 = \begin{array}{cc} b & a \\ a & b \end{array}
\]

\(L(G_2)\) consists of pictures \(p\) of size \((m, n), m \geq 2, n \geq 2\).

\[
M_0 = \begin{array}{cc} b & a \\ a & b \end{array} \Rightarrow (l/u)
\]
Leftmost/Uppermost P2DCFG - Example

Example 15.

\[(l/u)P2DCFG \ G_2 = (\Sigma, P_1, P_2, \{M_0\})\] where \(\Sigma = \{a, b\}, P_1 = \{c\}, P_2 = \{r\}\) with

\[
c = \{a \rightarrow ab, b \rightarrow ba\}, \quad r = \left\{a \rightarrow \begin{array}{c} a \\ b \end{array}, \quad b \rightarrow \begin{array}{c} b \\ a \end{array} \right\} \quad M_0 = \begin{array}{cc} b & a \\ a & b \end{array}
\]

\(L(G_2)\) consists of pictures \(p\) of size \((m, n), m \geq 2, n \geq 2.\)

\[
M_0 = \begin{array}{cc} b & a \\ a & b \end{array} \Rightarrow (l/u) \begin{array}{ccc} b & a & a \\ a & b & b \end{array} \Rightarrow (l/u)
\]
Example 15.

\((l/u)P2DCFG\) \(G_2 = (\Sigma, P_1, P_2, \{M_0\})\) where \(\Sigma = \{a, b\}, P_1 = \{c\}, P_2 = \{r\}\) with

\[
c = \{a \rightarrow ab, b \rightarrow ba\}, \quad r = \left\{ a \rightarrow \begin{array}{c} a \\ b \end{array}, \quad b \rightarrow \begin{array}{c} b \\ a \end{array}\right\}M_0 = \begin{array}{cc} b & a \\ a & b \end{array}
\]

\(L(G_2)\) consists of pictures \(p\) of size \((m, n), m \geq 2, n \geq 2.\)

\[
M_0 = \begin{array}{cc} b & a \\ a & b \end{array} \Rightarrow_{(l/u)} \begin{array}{ccc} b & a & a \\ a & b & b \end{array} \Rightarrow_{(l/u)} \begin{array}{ccc} b & a & a \\ a & b & b \end{array} \Rightarrow_{(l/u)} \begin{array}{ccc} b & a & a \\ a & b & b \end{array}
\]
Example 15.

\((l/u)P2DCFG\) \(G_2 = (\Sigma, P_1, P_2, \{M_0\})\) where \(\Sigma = \{a, b\}\), \(P_1 = \{c\}\), \(P_2 = \{r\}\) with

\[c = \{a \rightarrow ab, b \rightarrow ba\}, \quad r = \left\{a \rightarrow \frac{a}{b}, b \rightarrow \frac{b}{a}\right\}\]

\(M_0 = \begin{array}{cc}
    b & a \\
    a & b \\
\end{array}\)

\(L(G_2)\) consists of pictures \(p\) of size \((m, n)\), \(m \geq 2, n \geq 2\).

\[
M_0 = \begin{array}{ccc}
    b & a & a \\
    a & b & b \\
\end{array} \Rightarrow_{(l/u)} \begin{array}{ccc}
    b & a & a \\
    a & b & b \\
\end{array} \Rightarrow_{(l/u)} \begin{array}{ccc}
    b & a & a \\
    a & b & b \\
\end{array}
\]

\[
\begin{array}{cccc}
    b & a & a & a \\
    a & b & b & b \\
\end{array} \Rightarrow_{(l/u)} \begin{array}{cccc}
    b & a & a & a \\
    a & b & b & b \\
\end{array}
\]
Leftmost/Uppermost P2DCFG - Example

Example 15.

\((l/u)P2DCFG\ G_2 = (\Sigma, P_1, P_2, \{M_0\})\) where \(\Sigma = \{a, b\}\), \(P_1 = \{c\}\), \(P_2 = \{r\}\) with

\[c = \{a \rightarrow ab, b \rightarrow ba\},\ r = \left\{\begin{array}{c} a \rightarrow a \ b, \ b \rightarrow b \ a \end{array}\right\} M_0 = \begin{array}{c} b \ a \\ a \ b \end{array}\]

\(L(G_2)\) consists of pictures \(p\) of size \((m, n), m \geq 2, n \geq 2\).

\[M_0 = \begin{array}{c} b \ a \\ a \ b \end{array} \Rightarrow_{(l/u)} \begin{array}{c} b \ a \ a \\ a \ b \end{array} \Rightarrow_{(l/u)} \begin{array}{c} b \ a \ a \\ a \ b \end{array} \Rightarrow_{(l/u)} \begin{array}{c} b \ a \ a \\ a \ b \ b \end{array} \Rightarrow_{(l/u)} \begin{array}{c} b \ a \ a \\ a \ b \ b \end{array} \Rightarrow_{(l/u)} \begin{array}{c} b \ a \ a \ a \\ a \ b \ b \ b \end{array} \Rightarrow_{(l/u)} \begin{array}{c} b \ a \ a \ a \ a \\ a \ b \ b \ b \ b \end{array} \Rightarrow_{(l/u)} \begin{array}{c} b \ a \ a \ a \ a \ a \\ a \ b \ b \ b \ b \ b \end{array} \Rightarrow_{(l/u)} \begin{array}{c} b \ a \ a \ a \ a \ a \ a \\ a \ b \ b \ b \ b \ b \ b \end{array} \Rightarrow_{(l/u)} \begin{array}{c} b \ a \ a \ a \ a \ a \ a \ a \\ a \ b \ b \ b \ b \ b \ b \ b \end{array}

Figure: A sample derivation under \((l/u)\) mode in \(G_2\)
Survey
Language Families Hierachy (Recognizing devices)

Figure: Red edge = incomparable, Green edge = open problem
## Closure Properties (Recognizing devices)

<table>
<thead>
<tr>
<th>Operations</th>
<th>4DFA</th>
<th>4NFA</th>
<th>2OTA</th>
<th>TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Union</td>
<td>+</td>
<td>+</td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Intersection</td>
<td>+</td>
<td>+</td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Projection</td>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Row concatenation</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Column concatenation</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Row/Column Closure</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Complement</td>
<td>+</td>
<td>?</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>Clock-wise rotation</td>
<td></td>
<td></td>
<td></td>
<td>+</td>
</tr>
</tbody>
</table>

**Table:** Empty cell = unknown, ? = open problem
# Closure Properties (Grammars)

<table>
<thead>
<tr>
<th>Operations</th>
<th>TS</th>
<th>2DRLIN</th>
<th>P2DCFL</th>
<th>(R)P2DCFL</th>
<th>(CF)P2DCFL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Union</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Intersection</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Projection</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Row concatenation</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Column concat.</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Row/Col. Closure</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complement</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-W rotation</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table:** Empty cell = unknown, ? = open problem
Results
Comparison of P2DCFL and (l/u)P2DCFL

Theorem 16.  
P2DCFL and (l/u)P2DCFL with non-unary alphabet are incomparable but not disjoint.

Proof.
Comparison of P2DCFL and (l/u)P2DCFL

Theorem 16.

P2DCFL and (l/u)P2DCFL with non-unary alphabet are incomparable but not disjoint.

Proof.

1. $\{a, b\}** \in P2DCFL \cap (l/u)P2DCFL$

See Example 15:

$L(G_2) \in (l/u)P2DCFL - P2DCFL$ since we need to rewrite only the first column/row.

See Example 12:

$L(G_1) \in P2DCFL - (l/u)P2DCFL$ since we need to rewrite unique middle column and produce the same columns to the both sides.

□
Comparison of P2DCFL and (l/u)P2DCFL

**Theorem 16.**
P2DCFL and (l/u)P2DCFL with non-unary alphabet are incomparable but not disjoint.

**Proof.**

- \{a, b\}** ∈ P2DCFL ∩ (l/u)P2DCFL
- See Example 15: \(L(G_2) ∈ (l/u)P2DCFL − P2DCFL\) since we need to rewrite only the first column/row.
Comparison of $P2DCFL$ and $(l/u)P2DCFL$

**Theorem 16.**

$P2DCFL$ and $(l/u)P2DCFL$ with non-unary alphabet are incomparable but not disjoint.

**Proof.**

- $\{a, b\}^{**} \in P2DCFL \cap (l/u)P2DCFL$
- See Example 15: $L(G_2) \in (l/u)P2DCFL - P2DCFL$ since we need to rewrite only the first column/row.
- See Example 12: $L(G_1) \in P2DCFL - (l/u)P2DCFL$ since we need to rewrite unique middle column and produce the same columns to the both sides.

□
Comparison of P2DCFL and (l/u)P2DCFL

**Theorem 16.**
P2DCFL and (l/u)P2DCFL with non-unary alphabet are incomparable but not disjoint.

**Proof.**

- \( \{a, b\}^{**} \in P2DCFL \cap (l/u)P2DCFL \)
- See Example 15: \( L(G_2) \in (l/u)P2DCFL – P2DCFL \) since we need to rewrite only the first column/row.
- See Example 12: \( L(G_1) \in P2DCFL – (l/u)P2DCFL \) since we need to rewrite unique middle column and produce the same columns to the both sides.

P2DCFL and (l/u)P2DCFL with **unary** alphabet are equivalent.
Closure Properties of (l/u)P2DCFL

Theorem 17.
(l/u)P2DCFL is not closed under union.

Proof.
Let \( L(G_1) \subseteq \{a, b, d\}^{**} \):

\[
c_1 = \{b \rightarrow ba, a \rightarrow ad\}, \quad r_1 = \{b \rightarrow b, a \rightarrow a\}, \quad M_1 = \begin{bmatrix} b & a \\ a & d \end{bmatrix}.
\]

Let \( L(G_2) \subseteq \{a, b, e\}^{**} \):

\[
c_2 = \{b \rightarrow ba, a \rightarrow ae\}, \quad r_2 = \{b \rightarrow b, a \rightarrow a\}, \quad M_2 = \begin{bmatrix} b & a \\ a & e \end{bmatrix}.
\]
Closure Properties of (l/u)P2DCFL

**Theorem 17.**

(l/u)P2DCFL is not closed under union.

**Proof.**

Let \( L(G_1) \subseteq \{a, b, d\}^* \):

\[
c_1 = \{ b \to ba, a \to ad \}, \quad r_1 = \left\{ b \to \frac{b}{a}, a \to \frac{a}{d} \right\}, \quad M_1 = \left\{ \begin{array}{cc} b & a \\ a & d \end{array} \right\}.
\]

Let \( L(G_2) \subseteq \{a, b, e\}^* \):

\[
c_2 = \{ b \to ba, a \to ae \}, \quad r_2 = \left\{ b \to \frac{b}{a}, a \to \frac{a}{e} \right\}, \quad M_2 = \left\{ \begin{array}{cc} b & a \\ a & e \end{array} \right\}.
\]

\( M_{1 \cup 2} \subseteq M_1 \cup M_2 \), \( P_{1 \cup 2_{\text{column}}} \) requires \( a \to ad \cdots d \) and \( a \to ae \cdots e \).
Closure Properties of (l/u)P2DCFL

Theorem 17. (l/u)P2DCFL is not closed under union.

Proof.
Let \( L(G_1) \subseteq \{a, b, d\}^* \):

\[
c_1 = \{b \rightarrow ba, a \rightarrow ad\}, \quad r_1 = \left\{ b \rightarrow \begin{array}{c} b \\ a \\ \end{array}, \quad a \rightarrow \begin{array}{c} a \\ d \\ \end{array} \right\}, \quad M_1 = \left\{ \begin{array}{cc} b & a \\ a & d \\ \end{array} \right\}.
\]

Let \( L(G_2) \subseteq \{a, b, e\}^* \):

\[
c_2 = \{b \rightarrow ba, a \rightarrow ae\}, \quad r_2 = \left\{ b \rightarrow \begin{array}{c} b \\ a \\ \end{array}, \quad a \rightarrow \begin{array}{c} a \\ e \\ \end{array} \right\}, \quad M_2 = \left\{ \begin{array}{cc} b & a \\ a & e \\ \end{array} \right\}.
\]

\( M_1 \cup M_2 \subseteq M_1 \cup M_2, P_{1 \cup 2_{\text{column}}} \) requires \( a \rightarrow ad \cdots d \) and \( a \rightarrow ae \cdots e \).

But rule tables with these rules can be mixed and generate pictures not in \( L(G_1) \cup L(G_2) \).
Closure Properties of (l/u)P2DCFL

Theorem 18.

(l/u)P2DCFL is not closed under intersection.

Proof.

- Let $L(G_2)$ from Example 15 is denoted as $L_r$. 

Observe that $L \cap L_r = L_s$, but $L_s < (l/u)P^2DCFL$. □
Closure Properties of (l/u)P2DCFL

Theorem 18.
(l/u)P2DCFL is not closed under intersection.

Proof.

- Let $L(G_2)$ from Example 15 is denoted as $L_r$.
- $L_s \subseteq L_r$ s.t. all pictures are square sized.
Closure Properties of \((l/u)P2DCFL\)

**Theorem 18.**

\((l/u)P2DCFL\) is not closed under intersection.

**Proof.**

- Let \(L(G_2)\) from Example 15 is denoted as \(L_r\).
- \(L_s \subseteq L_r\) s.t. all pictures are square sized.
- Consider \(L\) consisting of sets
  1. square pictures with the first row \(xd\cdots d\), the first column \((xe\cdots e)^R\), otherwise \(bs\);
  2. rectangular picture with the first row \(yd\cdots d\), the first column \((ye\cdots e)^R\), otherwise \(bs\);
  3. pictures of \(L_s\)

\(L \cap L_r = L_s\), but \(L_s < (l/u)P2DCFL\).
Theorem 18.\( (l/u)P2DCFL \) is not closed under intersection.

Proof.

- Let \( L(G_2) \) from Example 15 is denoted as \( L_r \).
- \( L_s \subseteq L_r \) s.t. all pictures are square sized.
- Consider \( L \) consisting of sets
  1. square pictures with the first row \( xd \cdots d \), the first column \( (xe \cdots e)^R \), otherwise \( bs \);
  2. rectangular picture with the first row \( yd \cdots d \), the first column \( (ye \cdots e)^R \), otherwise \( bs \);
  3. pictures of \( L_s \)
- \( L \) can be generated by \( (l/u)P2DCFG \) \( G \):
  \[
  c_1 = \{ x \to yd, \ e \to eb \}, \ c_2 = \{ x \to b, \ e \to a \},
  r_1 = \left\{ y \to x, \ d \to \begin{array}{c} d \\ b \end{array} \right\}, \ r_2 = \{ b \to b, \ d \to a \}, \ M = \left\{ \begin{array}{cc} x & d \\ e & b \end{array} \right\}.
  \]
Closure Properties of \((l/u)P2DCFL\)

**Theorem 18.**

\((l/u)P2DCFL\) is not closed under intersection.

**Proof.**

- Let \(L(G_2)\) from Example 15 is denoted as \(L_r\).
- \(L_s \subseteq L_r\) s.t. all pictures are square sized.
- Consider \(L\) consisting of sets
  1. square pictures with the first row \(xd \cdots d\), the first column \((xe \cdots e)^R\), otherwise \(bs\);
  2. rectangular picture with the first row \(yd \cdots d\), the first column \((ye \cdots e)^R\), otherwise \(bs\);
  3. pictures of \(L_s\)
- \(L\) can be generated by \((l/u)P2DCFG\) \(G\):
  \(c_1 = \{x \rightarrow yd, e \rightarrow eb\}\), \(c_2 = \{x \rightarrow b, e \rightarrow a\}\),
  \(r_1 = \left\{y \rightarrow x\begin{array}{c}e\end{array}, d \rightarrow \begin{array}{c}d \ b\end{array}\right\}\), \(r_2 = \{b \rightarrow b, d \rightarrow a\}\), \(M = \left\{\begin{array}{ccc}x & d \\ e & b\end{array}\right\}\).
- Observe that \(L \cap L_r = L_s\), but \(L_s \notin (l/u)P2DCFL\).
Theorem 19.

\( (l/u)P2DCFL \subset (R)(l/u)P2DCFL \subset (CF)(l/u)P2DCFL \)

Proof.

- Consider \( L_s \) from Theorem 18. There is a \( (R)(l/u)P2DCFG \) with control language \((cr)^*\) generating \( L_s \).
Generative Power of Controlled (l/u)P2DCFL

Theorem 19.

\((l/u)P2DCFL \subset (R)(l/u)P2DCFL \subset (CF)(l/u)P2DCFL\)

Proof.

- Consider \(L_{s}\) from Theorem 18. There is a \((R)(l/u)P2DCFG\) with control language \((cr)^*\) generating \(L_{s}\).

- Consider \(L(G_1)\) from Example 12 but with sizes \((k + 1, 2k + 1), k \geq 1\). 

\[\square\]
Generative Power of Controlled (l/u)P2DCFL

**Theorem 19.**

\[(l/u)P2DCFL \subset (R)(l/u)P2DCFL \subset (CF)(l/u)P2DCFL\]

**Proof.**

- Consider \(L_s\) from Theorem 18. There is a \((R)(l/u)P2DCFG\) with control language \((cr)^*\) generating \(L_s\).
- Consider \(L(G_1)\) from Example 12 but with sizes \((k + 1, 2k + 1), k \geq 1\).
- It can be generated by \((CF)(l/u)P2DCFG\) \(G\) with \(\Sigma = \{a, b, e\}\):
  \[c_1 = \{e \rightarrow ea, a \rightarrow ab\}, c_2 = \{e \rightarrow ae, a \rightarrow ba\}, c_3 = \{a \rightarrow aa, b \rightarrow bb\},\]
  \[r = \{e \rightarrow a, a \rightarrow b\}, M = \begin{pmatrix} e & a \\ a & b \end{pmatrix}.\]
Generative Power of Controlled \((l/u)P2DCFL\)

**Theorem 19.**
\[(l/u)P2DCFL \subset (R)(l/u)P2DCFL \subset (CF)(l/u)P2DCFL\]

**Proof.**

- Consider \(L_s\) from Theorem 18. There is a \((R)(l/u)P2DCFG\) with control language \((cr)^*\) generating \(L_s\).
- Consider \(L(G_1)\) from Example 12 but with sizes \((k + 1, 2k + 1)\), \(k \geq 1\).
- It can be generated by \((CF)(l/u)P2DCFG\) \(G\) with \(\Sigma = \{a, b, e\}\):
  \[
c_1 = \{e \rightarrow ea, a \rightarrow ab\},
  c_2 = \{e \rightarrow ae, a \rightarrow ba\},
  c_3 = \{a \rightarrow aa, b \rightarrow bb\},
  r = \left\{ e \rightarrow \begin{array}{c} e \\ a \end{array}, a \rightarrow \begin{array}{c} a \\ b \end{array} \right\},
  M = \left\{ \begin{array}{cc} e & a \\ a & b \end{array} \right\}.
\]
- \(C = \{(c_1 r)^n c_2 c_3^n | n \geq 0\}\)
Generative Power of Controlled \((l/u)P2DCFL\)

**Theorem 19.**

\((l/u)P2DCFL \subset (R)(l/u)P2DCFL \subset (CF)(l/u)P2DCFL\)

**Proof.**

- Consider \(L_s\) from Theorem 18. There is a \((R)(l/u)P2DCFG\) with control language \((cr)^*\) generating \(L_s\).
- Consider \(L(G_1)\) from Example 12 but with sizes \((k + 1, 2k + 1), k \geq 1\).
- It can be generated by \((CF)(l/u)P2DCFG\) \(G\) with \(\Sigma = \{a, b, e\}\):
  \[c_1 = \{e \rightarrow ea, a \rightarrow ab\}, c_2 = \{e \rightarrow ae, a \rightarrow ba\}, c_3 = \{a \rightarrow aa, b \rightarrow bb\},\]
  \[r = \left\{ \begin{array}{c} e \rightarrow e \\ a \rightarrow a \\ b \rightarrow b \end{array} \right\}, M = \left\{ \begin{array}{cc} e & a \\ a & b \end{array} \right\}.\]
- \(C = \{(c_1r)^n c_2 c_3^n | n \geq 0\}\)
- Regular controlled language is not enough. We need to ”remember” the number of columns generated to the right of the middle one.
Expressiveness of Controlled (l/u)P2DCFL

Lemma 20.

$L_d = \{ p \in \{a, b\}^{++} \mid |p|_{col} = |p|_{row}, p(i,j) = b, \text{for } i = j, p(i,j) = a \text{ for } i \neq j \}$
can be generated by (R)(l/u)P2DCFG $G_d$ with one control symbol, but $L_d \not\in (l/u)P2DCFL$.

Proof.

Consider (l/u)P2DCFG of $G_d$ as $(\{0, 1, 2\}, \{c\}, \{r\}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix})$ where

$$c = \{1 \rightarrow 12, 0 \rightarrow 00\}, \quad r = \begin{cases} 1 \rightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix}, & 2 \rightarrow \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\ 0 \rightarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{cases},$$

and regular control language $(cr)^*$. 
Lemma 20.

\[ L_d = \{ p \in \{a, b\}^{++} \mid |p|_{col} = |p|_{row}, p(i, j) = b, \text{ for } i = j, p(i, j) = a \text{ for } i \neq j \} \]

can be generated by \((R)\)(l/u)P2DCFG \(G_d\) with one control symbol, but \(L_d \notin \)(l/u)P2DCFL.

Proof.

Consider \((l/u)P2DCFG\) of \(G_d\) as \((\{0, 1, 2\}, \{c\}, \{r\}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix})\) where

\[
c = \{1 \rightarrow 12, 0 \rightarrow 00\}, \quad r = \begin{cases} 1 \rightarrow 1, & 2 \rightarrow 0 \\ 0 \rightarrow 0 \end{cases}\]

and regular control language \((cr)^*\).

- \((R)\)(l/u)P2DCFG \(G_d\) generates \(L_d\).
Expressiveness of Controlled (l/u)P2DCFL

Lemma 20.

\[ L_d = \{ p \in \{a, b\}^+ | |p|_{col} = |p|_{row}, p(i, j) = b, \text{ for } i = j, p(i, j) = a \text{ for } i \neq j \} \]
can be generated by (R)(l/u)P2DCFG \( G_d \) with one control symbol, but \( L_d \notin (l/u)P2DCFL \).

Proof.

Consider \((l/u)P2DCFG\) of \( G_d \) as \( (\{0, 1, 2\}, \{c\}, \{r\}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}) \) where

\[ c = \{1 \rightarrow 12, 0 \rightarrow 00\}, \quad r = \begin{cases} 1 \rightarrow 1 \\ 2 \rightarrow 0 \\ 0 \rightarrow 0 \end{cases} \]

and regular control language \((cr)^*\).

- \((R)(l/u)P2DCFG\) \( G_d \) generates \( L_d \).
- 2 is the only control symbol.
Expressiveness of Controlled (l/u)P2DCFL

Lemma 20.

\[ L_d = \{ p \in \{a, b\}^+ \mid |p|_{\text{col}} = |p|_{\text{row}}, p(i, j) = b, \text{ for } i = j, p(i, j) = a \text{ for } i \neq j \} \]

can be generated by \((R)(l/u)P2DCFG\) \(G_d\) with one control symbol, but \(L_d \notin (l/u)P2DCFL\).

Proof.

Consider \((l/u)P2DCFG\) of \(G_d\) as \((\{0, 1, 2\}, \{c\}, \{r\}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix})\) where

\[ c = \{1 \rightarrow 12, 0 \rightarrow 00\}, \quad r = \left\{1 \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix}, 2 \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix}, 0 \rightarrow \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right\}, \]

and regular control language \((cr)^*\).

- \((R)(l/u)P2DCFG\) \(G_d\) generates \(L_d\).
- 2 is the only control symbol.
- From [4], there is no P2DCFG with regular control with less than two control symbols that generates \(L_d\).
Theorem 21.
(l/u)P2DCFL and LOC are incomparable but not disjoint.

Proof.

- $\{a\}^{**} \in (l/u)P2DCFL \cap LOC$
Generative Power of \((l/u)P2DCFL\)

**Theorem 21.**

\((l/u)P2DCFL\) and \(LOC\) are incomparable but not disjoint.

**Proof.**

- \(\{a\}^{**} \in (l/u)P2DCFL \cap LOC\)
- Languages with rectangular pictures with even number or rows and columns \(\in (l/u)P2DCFL - LOC\)
Theorem 21.

\((l/u)P2DCFL\) and \(LOC\) are incomparable but not disjoint.

Proof.

- \{a\}** \(\in (l/u)P2DCFL \cap LOC\)
- Languages with rectangular pictures with even number or rows and columns \(\in (l/u)P2DCFL - LOC\)
- \(L_d \in LOC - (l/u)P2DCFL\)
### Closure Properties (P2DCFL)

<table>
<thead>
<tr>
<th>Operations</th>
<th>TS</th>
<th>P2DCFL</th>
<th>(l/u)P2DCFL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Union</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Intersection</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Projection</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Row concatenation</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Column concatenation</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table:** Empty cell = unknown
Language Families Hierachy (Grammars)

Figure: Red edge = incomparable but not disjoint
Thanks for your attention!
References


