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) tesselation 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 Σ of the form

$$p = \begin{matrix} p(1,1) & \cdots & p(1,n) \\ \vdots & \ddots & \vdots \\ p(m,1) & \cdots & p(m,n) \end{matrix}$$

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

- Σ^{**} = set of all rectangular arrays over Σ (λ 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 $\pi \colon \Gamma \to \Sigma$, where Γ, Σ are alphabets.
- Column concatenation of two pictures ($p \oplus q$) requires the same number of rows.
- Row concatenation of two pictures ($p \ominus q$) requires the same number of columns.
- Column/Row closure $L^{*\oplus}$ and $L^{*\ominus}$ such that $L^{**} = (L^{*\oplus})^{*\ominus} = (L^{*\ominus})^{*\oplus}$
- Clock-wise rotation of a picture (p^R)

Definitions and Examples

2D Regular Expressions

Recursive definition over alphabet Σ

- Atomic languages: the empty language $\emptyset$, {a} with $a \in \Sigma$.
- 2D Regular operations $\mathcal{R} = \{\ominus, \oplus, *\ominus, *\oplus, \cup, \cap, {}^c\}$.
- The result of $\odot \in \mathcal{R}$ applied to regular 2D languge is a regular 2D language.

- Family: RE
- Modifications: complement-free RE (CFRE), star-free RE (SFRE), projection of CFRE (PCFRE)

2D Regular Expressions - Example

- Let $\Sigma = \{\blacksquare, \square\}$
- 2D regular expression over Σ: $\left(\left((\blacksquare \ominus \square)^{*\ominus}\right) \varobar \left((\square \ominus \blacksquare)^{*\ominus}\right)\right)^{*\varobar}$

2D Regular Expressions - Example

- Let $\Sigma = \{\blacksquare, \square\}$
- 2D regular expression over Σ: $(((\blacksquare \ominus \square)^{*\ominus}) \obar ((\square \ominus \blacksquare)^{*\ominus}))^{*\obar}$

■	□	■	□	■	□
□	■	□	■	□	■
■	□	■	□	■	□
□	■	□	■	□	■

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;
- $\delta\colon Q - \{q_a, q_r\} \times \Sigma \to 2^{Q \times \Delta}$ ($\delta\colon Q - \{q_a, q_r\} \times \Sigma \to Q \times \Delta$) 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”

4-way Automata - Example

Example 2.

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

4-way Automata - Example

Example 2.

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

- Moves along the diagonal until the bottom-right corner $\Rightarrow$ square.
- Checks that all positions contain a symbol from Σ.

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:

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:

- 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.

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:

- 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 Σ 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*:
  $V \to AV'$ or $V \to A$ where $V, V' \in V_h$ and $A \in \Sigma_I$;
- R_v is a finite set of *vertical rules*:
  $A \to aA'$ or $A \to a$ where $A, A' \in V_v$ 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)

$B_{h,k}(p)$ = the set of all blocks of p of size (h,k), where $h \le m$, $k \le n$.

Definition 6.

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

- Φ is the set of *allowed blocks* or *representation by tiles* including #.
- $\lambda \in L(\Phi)$ iff $\begin{matrix} \# & \# \\ \# & \# \end{matrix} \in \Phi$
- The family: LOC

Local 2D Languages (LOC) - Example

Example 7.

$$\Phi = \left\{ \begin{array}{cccccc} \begin{matrix}1 & 0\\0 & 1\end{matrix}, & \begin{matrix}0 & 0\\1 & 0\end{matrix}, & \begin{matrix}0 & 1\\0 & 0\end{matrix}, & \begin{matrix}0 & 0\\0 & 0\end{matrix}, & \begin{matrix}\# & 1\\\# & 0\end{matrix}, & \begin{matrix}\# & 0\\\# & 0\end{matrix}, \\ \begin{matrix}0 & 0\\\# & \#\end{matrix}, & \begin{matrix}0 & 1\\\# & \#\end{matrix}, & \begin{matrix}0 & \#\\1 & \#\end{matrix}, & \begin{matrix}0 & \#\\0 & \#\end{matrix}, & \begin{matrix}\# & \#\\0 & 0\end{matrix}, & \begin{matrix}\# & \#\\1 & 0\end{matrix}, \\ & \begin{matrix}\# & \#\\\# & 1\end{matrix}, & \begin{matrix}\# & \#\\0 & \#\end{matrix}, & \begin{matrix}\# & 0\\\# & \#\end{matrix}, & \begin{matrix}1 & \#\\\# & \#\end{matrix} & \end{array} \right\}$$

- $L(\Phi)$ 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 $\mathcal{T} = (\Sigma, \Gamma, \Phi, \pi)$, where

- Σ and Γ are two alphabets;
- Φ is finite set of *tiles* over $\Gamma \cup \#$;
- $\pi : \Gamma \to \Sigma$ is a *projection*.

- L recognizable by TS $\mathcal{T}$: $L(\mathcal{T}) = \pi(L')$ 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*

Pure 2D Context-Free Grammars

Definition 11.
A pure 2D context-free grammar ($P2DCFG$, [2]) is a 4-tuple

$$G = (\Sigma, P_1, P_2, \mathcal{M}_0)$$

where

i) Σ 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) $\mathcal{M}_0 \subseteq \Sigma^{**} - \{\lambda\}$ is a finite set of axiom arrays.

Pure 2D Context-Free Grammars - Derivation

A derivation in a *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: $L(G) = \{M \in \Sigma^{**} | \, M_0 \Rightarrow^* M$ for some $M_0 \in \mathcal{M}_0\}$.
- 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

$$c = \{a \to bab, e \to aea\}, r = \left\{ e \to \begin{matrix} e \\ a \end{matrix}, a \to \begin{matrix} a \\ b \end{matrix} \right\}, M_0 = \begin{matrix} a & e & a \\ b & a & b \end{matrix}$$

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

$$\begin{matrix} a & a & a & e & a & a & a \\ b & b & b & a & b & b & b \\ b & b & b & a & b & b & b \\ b & b & b & a & b & b & b \\ b & b & b & a & b & b & b \end{matrix}$$

Figure: A picture in $L(G_1)$

Controlled Pure 2D Context-Free Grammars

Definition 13.
A Controlled $P2DCFG$ is $G^c = (G, C)$ where

- $G = (\Sigma, P_1, P_2, \mathcal{M}_0)$ 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 $w = l_1 l_2 \ldots l_m$, then the tables of rules with labels l_1, l_2, …, and l_m are successively applied starting with M_1 to finally yield M_2.
- The families: $(R)P2DCFL$ and $(CF)P2DCFL$

Leftmost/Uppermost Pure 2D Context-Free Grammars

Definition 14.

- A $(l/u)P2DCFG$ is $P2DCFG$ $G = (\Sigma, P_1, P_2, \mathcal{M}_0)$ 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: $(l/u)P2DCFL$

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

$$c = \{a \to ab, b \to ba\}, r = \left\{a \to \begin{matrix} a \\ b \end{matrix}, b \to \begin{matrix} b \\ a \end{matrix}\right\} M_0 = \begin{matrix} b & a \\ a & b \end{matrix}$$

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

$$M_0 = \begin{matrix} b & a \\ a & b \end{matrix} \Rightarrow_{(l/u)}$$

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

$$c = \{a \to ab, b \to ba\}, r = \left\{a \to \begin{matrix} a \\ b \end{matrix}, b \to \begin{matrix} b \\ a \end{matrix}\right\} M_0 = \begin{matrix} b & a \\ a & b \end{matrix}$$

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

$$M_0 = \begin{matrix} b & a \\ a & b \end{matrix} \Rightarrow_{(l/u)} \begin{matrix} b & a & a \\ a & b & b \end{matrix} \Rightarrow_{(l/u)}$$

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

$$c = \{a \to ab, b \to ba\}, r = \left\{ a \to \begin{matrix} a \\ b \end{matrix}, b \to \begin{matrix} b \\ a \end{matrix} \right\} M_0 = \begin{matrix} b & a \\ a & b \end{matrix}$$

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

$$M_0 = \begin{matrix} b & a \\ a & b \end{matrix} \Rightarrow_{(l/u)} \begin{matrix} b & a & a \\ a & b & b \end{matrix} \Rightarrow_{(l/u)} \begin{matrix} b & a & a \\ a & b & b \\ a & b & b \end{matrix} \Rightarrow_{(l/u)}$$

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

$$c = \{a \to ab, b \to ba\}, r = \left\{a \to \begin{matrix} a \\ b \end{matrix}, b \to \begin{matrix} b \\ a \end{matrix}\right\} M_0 = \begin{matrix} b & a \\ a & b \end{matrix}$$

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

$$M_0 = \begin{matrix} b & a \\ a & b \end{matrix} \Rightarrow_{(l/u)} \begin{matrix} b & a & a \\ a & b & b \end{matrix} \Rightarrow_{(l/u)} \begin{matrix} b & a & a \\ a & b & b \\ a & b & b \end{matrix} \Rightarrow_{(l/u)}$$

$$\begin{matrix} b & a & a & a \\ a & b & b & b \\ a & b & b & b \end{matrix} \Rightarrow_{(l/u)}$$

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

$$c = \{a \to ab, b \to ba\}, r = \left\{a \to \begin{matrix} a \\ b \end{matrix}, b \to \begin{matrix} b \\ a \end{matrix}\right\} M_0 = \begin{matrix} b & a \\ a & b \end{matrix}$$

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

$$M_0 = \begin{matrix} b & a \\ a & b \end{matrix} \Rightarrow_{(l/u)} \begin{matrix} b & a & a \\ a & b & b \end{matrix} \Rightarrow_{(l/u)} \begin{matrix} b & a & a \\ a & b & b \\ a & b & b \end{matrix} \Rightarrow_{(l/u)}$$

$$\begin{matrix} b & a & a & a \\ a & b & b & b \\ a & b & b & b \end{matrix} \Rightarrow_{(l/u)} \begin{matrix} b & a & a & a & a \\ a & b & b & b & b \\ a & b & b & b & b \end{matrix}$$

Figure: A sample derivation under (l/u) mode in G_2

Survey

Language Families Hierachy (Recognizing devices)

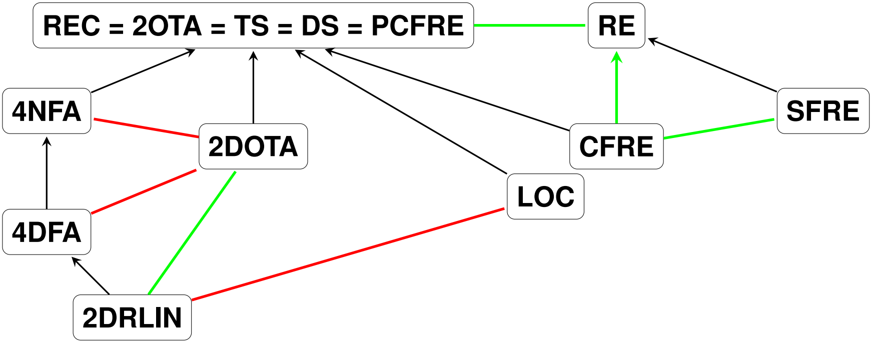

Figure: Red edge = incomparable, Green edge = open problem

Closure Properties (Recognizing devices)

Operations	4DFA	4NFA	2OTA	TS
Union	+	+		+
Intersection	+	+		+
Projection			+	+
Row concatenation	-	-	+	+
Column concatenation	-	-	+	+
Row/Column Closure	-	-	+	+
Complement	+	?		-
Clock-wise rotation				+

Table: Empty cell = unknown, ? = open problem

Closure Properties (Grammars)

Operations	TS	2DRLIN	P2DCFL	(R)P2DCFL	(CF)P2DCFL
Union	+		-	+	
Intersection	+		-		-
Projection	+	+	+	+	
Row concatenation	+		-	-	
Column concat.	+		-	-	
Row/Col. Closure	+				
Complement	-				
C-W rotation	+				

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.

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

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.

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

$$c_1 = \{b \to ba, a \to ad\},\ r_1 = \left\{ b \to \begin{matrix} b \\ a \end{matrix}, a \to \begin{matrix} a \\ d \end{matrix} \right\},\ \mathcal{M}_1 = \left\{ \begin{matrix} b & a \\ a & d \end{matrix} \right\}.$$

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

$$c_2 = \{b \to ba, a \to ae\},\ r_2 = \left\{ b \to \begin{matrix} b \\ a \end{matrix}, a \to \begin{matrix} a \\ e \end{matrix} \right\},\ \mathcal{M}_2 = \left\{ \begin{matrix} b & a \\ a & e \end{matrix} \right\}.$$

Closure Properties of (l/u)P2DCFL

Theorem 17.

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

Proof.

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

$$c_1 = \{b \to ba, a \to ad\},\ r_1 = \left\{ b \to \begin{matrix} b \\ a \end{matrix}, a \to \begin{matrix} a \\ d \end{matrix} \right\},\ \mathcal{M}_1 = \left\{ \begin{matrix} b & a \\ a & d \end{matrix} \right\}.$$

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

$$c_2 = \{b \to ba, a \to ae\},\ r_2 = \left\{ b \to \begin{matrix} b \\ a \end{matrix}, a \to \begin{matrix} a \\ e \end{matrix} \right\},\ \mathcal{M}_2 = \left\{ \begin{matrix} b & a \\ a & e \end{matrix} \right\}.$$

- $\mathcal{M}_{1\cup 2} \subseteq \mathcal{M}_1 \cup \mathcal{M}_2$, $P_{1\cup 2_{column}}$ 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 $L(G_1) \subseteq \{a,b,d\}^{**}$:

$$c_1 = \{b \to ba, a \to ad\},\ r_1 = \left\{ b \to \begin{matrix} b \\ a \end{matrix}, a \to \begin{matrix} a \\ d \end{matrix} \right\},\ \mathcal{M}_1 = \left\{ \begin{matrix} b & a \\ a & d \end{matrix} \right\}.$$

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

$$c_2 = \{b \to ba, a \to ae\},\ r_2 = \left\{ b \to \begin{matrix} b \\ a \end{matrix}, a \to \begin{matrix} a \\ e \end{matrix} \right\},\ \mathcal{M}_2 = \left\{ \begin{matrix} b & a \\ a & e \end{matrix} \right\}.$$

- $\mathcal{M}_{1\cup 2} \subseteq \mathcal{M}_1 \cup \mathcal{M}_2$, $P_{1\cup 2_{column}}$ requires $a \to ad \cdots d$ and $a \to ae \cdots e$.
- 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.

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

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 $(l/u)P2DCFG\ G$:

$$c_1 = \{x \to yd,\ e \to eb\},\ c_2 = \{x \to b,\ e \to a\},$$
$$r_1 = \left\{ y \to \begin{array}{c} x \\ e \end{array},\ d \to \begin{array}{c} d \\ b \end{array} \right\},\ r_2 = \{b \to b,\ d \to a\},\ \mathcal{M} = \left\{ \begin{array}{cc} x & d \\ e & b \end{array} \right\}.$$

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 $(l/u)P2DCFG\ G$:
  $c_1 = \{x \rightarrow yd,\ e \rightarrow eb\},\ c_2 = \{x \rightarrow b,\ e \rightarrow a\},$
  $r_1 = \left\{ y \rightarrow \begin{matrix} x \\ e \end{matrix},\ d \rightarrow \begin{matrix} d \\ b \end{matrix} \right\},\ r_2 = \{b \rightarrow b,\ d \rightarrow a\},\ \mathcal{M} = \left\{ \begin{matrix} x & d \\ e & b \end{matrix} \right\}.$
- Observe that $L \cap L_r = L_s$, but $L_s \notin (l/u)P2DCFL$.

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.

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 $(k+1, 2k+1)$, $k \geq 1$.

□

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 $\Sigma = \{a, b, e\}$:
  $c_1 = \{e \to ea, a \to ab\}, c_2 = \{e \to ae, a \to ba\}, c_3 = \{a \to aa, b \to bb\}$,
  $r = \left\{e \to \begin{matrix} e \\ a \end{matrix}, a \to \begin{matrix} a \\ b \end{matrix}\right\}, \mathcal{M} = \left\{\begin{matrix} e & a \\ a & b \end{matrix}\right\}$.

□

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 $\Sigma = \{a, b, e\}$:
  $c_1 = \{e \rightarrow ea, a \rightarrow ab\}, c_2 = \{e \rightarrow ae, a \rightarrow ba\}, c_3 = \{a \rightarrow aa, b \rightarrow bb\}$,
  $r = \left\{ e \rightarrow \begin{matrix} e \\ a \end{matrix}, a \rightarrow \begin{matrix} a \\ b \end{matrix} \right\}, \mathcal{M} = \left\{ \begin{matrix} e & a \\ a & b \end{matrix} \right\}.$
- $C = \{(c_1 r)^n c_2 c_3^n |\ n \geq 0\}$

□

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 $\Sigma = \{a, b, e\}$:
  $c_1 = \{e \to ea, a \to ab\}, c_2 = \{e \to ae, a \to ba\}, c_3 = \{a \to aa, b \to bb\},$
  $r = \left\{ e \to \begin{array}{c} e \\ a \end{array}, a \to \begin{array}{c} a \\ b \end{array} \right\}, \mathcal{M} = \left\{ \begin{array}{cc} e & a \\ a & b \end{array} \right\}.$
- $C = \{(c_1 r)^n c_2 c_3^n |\ n \geq 0\}$
- 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.

$L_d = \{p \in \{a,b\}^{++} \mid |p|_{col} = |p|_{row}, p(i,j) = b$, for $i = j, p(i,j) = a$ 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 $(\{0,1,2\},\{c\},\{r\},\left\{\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right\})$ where

$$c = \{1 \rightarrow 12, 0 \rightarrow 00\}, r = \left\{1 \rightarrow \begin{array}{c} 1 \\ 0 \end{array}, 2 \rightarrow \begin{array}{c} 0 \\ 1 \end{array}, 0 \rightarrow \begin{array}{c} 0 \\ 0 \end{array}\right\},$$

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

Expressiveness of Controlled (l/u)P2DCFL

Lemma 20.
$L_d = \{p \in \{a,b\}^{++} \mid |p|_{col} = |p|_{row}, p(i,j) = b,$ *for* $i = j, p(i,j) = a$ *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 $(\{0,1,2\},\{c\},\{r\},\left\{\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}\right\})$ where

$$c = \{1 \to 12, 0 \to 00\}, r = \left\{1 \to \begin{matrix} 1 \\ 0 \end{matrix}, 2 \to \begin{matrix} 0 \\ 1 \end{matrix}, 0 \to \begin{matrix} 0 \\ 0 \end{matrix}\right\},$$

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

- (R)(l/u)P2DCFG G_d generates L_d.

Expressiveness of Controlled (l/u)P2DCFL

Lemma 20.

$L_d = \{p \in \{a,b\}^{++} \mid |p|_{col} = |p|_{row}, p(i,j) = b,$ for $i = j, p(i,j) = a$ 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 $(\{0,1,2\},\{c\},\{r\},\left\{\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right\})$ where

$$c = \{1 \to 12, 0 \to 00\}, r = \left\{1 \to \begin{array}{c} 1 \\ 0 \end{array}, 2 \to \begin{array}{c} 0 \\ 1 \end{array}, 0 \to \begin{array}{c} 0 \\ 0 \end{array}\right\},$$

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.
$L_d = \{p \in \{a,b\}^{++} \mid |p|_{col} = |p|_{row}, p(i,j) = b,$ for $i = j, p(i,j) = a$ 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 $(\{0,1,2\},\{c\},\{r\},\left\{\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}\right\})$ where

$$c = \{1 \rightarrow 12, 0 \rightarrow 00\}, r = \left\{1 \rightarrow \begin{matrix}1\\0\end{matrix}, 2 \rightarrow \begin{matrix}0\\1\end{matrix}, 0 \rightarrow \begin{matrix}0\\0\end{matrix}\right\},$$

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.

Generative Power of (l/u)P2DCFL

Theorem 21.
$(l/u)P2DCFL$ and LOC are incomparable but not disjoint.

Proof.

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

□

Generative Power of (l/u)P2DCFL

Theorem 21.
$(l/u)P2DCFL$ and LOC are incomparable but not disjoint.

Proof.

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

□

Generative Power of (l/u)P2DCFL

Theorem 21.
$(l/u)P2DCFL$ and LOC are incomparable but not disjoint.

Proof.

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

Operations	TS	P2DCFL	(l/u)P2DCFL
Union	+	-	-
Intersection	+	-	-
Projection	+	+	
Row concatenation	+	-	
Column concatenation	+	-	

Table: Empty cell = unknown

Language Families Hierachy (Grammars)

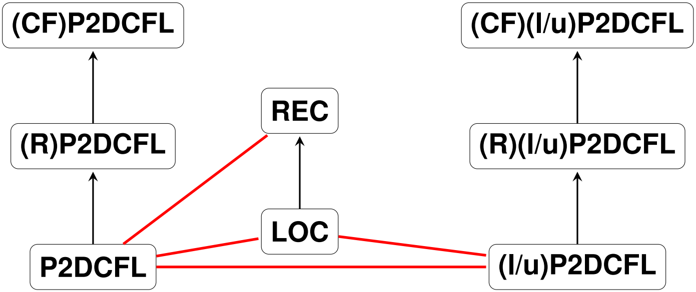

Figure: Red edge = incomparable but not disjoint

Thanks for your attention!

References

- G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages Vol. 3, Springer, Berlin, 1997.
- K. G. Subramanian, R. M. Ali, M. Geethalakshmi, A. K. Nagar, Pure 2D picture grammars and languages. Discrete Appl. Math. 157 (2009) 3401-3411.
- M. M. Bersani, A. Frigeri, A. Cherubini, On Some Classes of 2D Languages and Their Relations. In: IWCIA 2011, LNCS 6636, Springer-Verlag, 2011, 222–234.
- M. M. Bersani, A. Frigeri, A. Cherubini, Expressiveness and complexity of regular pure two-dimensional contextfree languages. International Journal of Computer Mathematics 90(8), 2013, 1708–1733.
- Z. Křivka, C. Martín-Vide, A. Meduna, K. G. Subramanian, A Variant of Pure Two-dimensional Context-free Grammars generating Picture Languages. In: IWCIA 2014, Brno, 11 pages (in press).