On the Complexity of pure 2D Context-free Grammars

Achille Frigeri

Dipartimento di Matematica “Francesco Brioschi”
Politecnico di Milano

joint work Marcello M. Bersani & Alessandra Cherubini

September 21, 2012

Initial motivations

- One-dimensional languages - consolidated theory
- Two-dimensional languages
  - The theory is getting consolidated
  - Various classes are known
  - Regular languages are not really defined TS best candidate
  - Expressiveness hierarchy is getting to be complete
  - Sometimes, classes are not comparable

Motivations

- Extend classes of languages
- Study new closures extending the theory
- Need for comparisons among classes - hierarchy
- Algorithm for parsing - complexity

What is a two-dimensional language

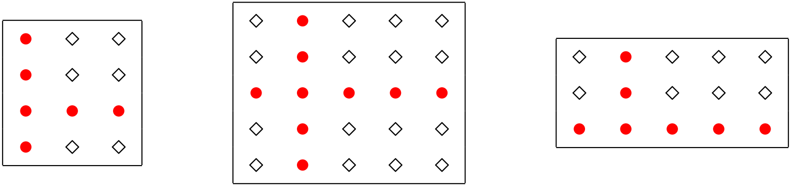
Alphabet Σ of finite symbols

Definition

A bidimensional language $\mathrm{L} \subseteq \Sigma^{**}$ is a set of two-dimensional arrays over alphabet Σ

$\mathrm{L} =$ cross of $\bullet$ over $\diamond$

$$\Sigma = \{\bullet, \diamond\}$$

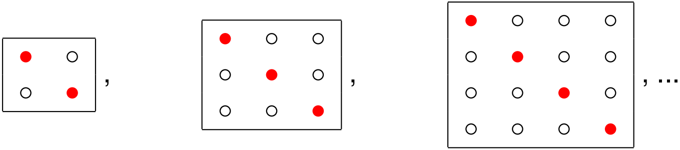


Formalism defining languages

- **Automata**
- **Grammars**
- Expressions
- Logic
- Algebra

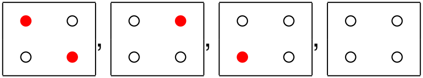
Local languages

L = squares with diagonal of ●

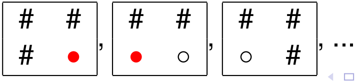


Alphabet: $\Gamma = \{\circ, \bullet\} \cup \{\#\}$

Local factors to build pictures:



Boundaries (removed after construction):



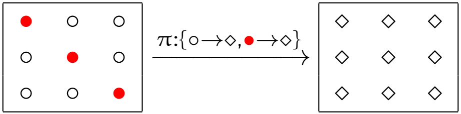
Tiling Systems

Projection of Local languages:

- Local alphabet: Γ
- Final alphabet: Σ
- Projection: $\pi : \Gamma \to \Sigma$

$L =$ squares of $\diamond$

$$\Gamma = \{\bullet, \circ\}, \Sigma = \{\diamond\}$$



Cannot be defined by local construction (π is needed).

Comparison Tiling Systems - Local

	LOC	**Tiling System**
Closures	$\cup, \cap$ R-C concat rotation complement	$\cup, \cap$ R-C concat rotation NOT complement
Membership	P	NP-complete
Emptiness	D	U
Other		EMSO, 2-way OTA, C-free REG exp

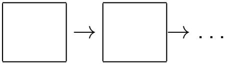
How to classify grammars

Set of rules defined by

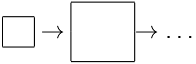
- *terminals* - alphabet constituting final pictures; $T = \{a, b\}$
- *non terminals* - intermediate alphabet realizing projection; $N = \{A, B\}$
- set of rules; $A \to aBa$, ...

Two classes:

- **Isometric**: rules do not modify the dimensions of the area on which they are applied



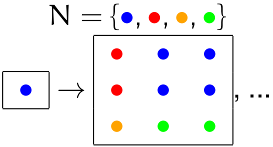
- **Non-isometric**: rules transform a starting axiom by means of successive expansions of its sub-pictures.



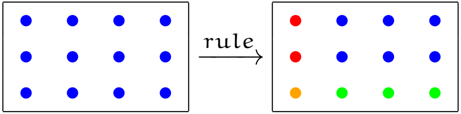
Regional Tile Grammars

Isometric grammars

- A sub-picture is substituted with an isometric sub-picture according to a rule of the grammar



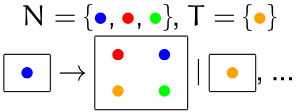
A “blue” area of • is replaced with an isometric subpicture of the form defined on the right.



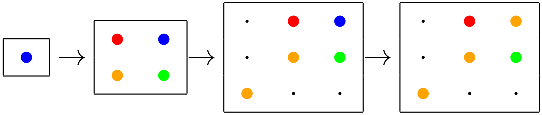
Průša grammar

Non-Isometric grammars

- A non-terminal is substituted with a sub-picture according to a rule of the grammar



Each non-terminal is substituted with the subpicture defined on the right.



(R)P2DCFL

(Regularly controlled) Pure 2D Context-free Grammar

- Pure: non-terminals not allowed
- Tables of row/column rules
  - strictly separated
  - arbitrarily applied on the picture
  - sequence of rule can be controlled by a regular language

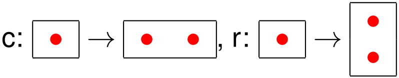
$$c: \begin{array}{l} \boxed{y} \to \boxed{x \; y \; x} \\ \boxed{z} \to \boxed{b \; z \; b} \end{array}, \quad r: \begin{array}{l} \boxed{x} \to \boxed{\begin{array}{c} x \\ b \end{array}} \mid \boxed{\begin{array}{c} x \\ c \end{array}} \\ \boxed{y} \to \boxed{\begin{array}{c} y \\ z \end{array}} \end{array}$$

Derivation:

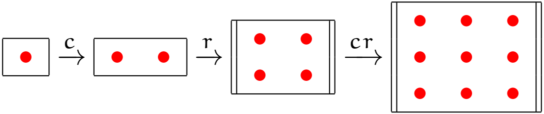
$$\boxed{y} \xrightarrow{c} \boxed{x \; y \; x} \xrightarrow{r} \boxed{\begin{array}{ccc} x & y & x \\ b & z & c \end{array}} \xrightarrow{c} \boxed{\begin{array}{ccccc} x & x & y & x & x \\ b & b & z & b & c \end{array}}$$

(R)P2DCFL

Language of squares of •



- Since each derivation is legal, we need a control language over the alphabet of rules
- $L = (cr)^*$



- Note that a P2DCFG is a RP2DCFG with control Γ^*

Comparison Regional - Průša - (R)P2DCFL

	RTG	Průša	(R)P2DCFL
Closures	$\cup$ R/C $\bullet$ Rotation Projection	$\cup, \cap$ R-C concat rotation NOT complement	Transposition Reflection NOT $\cup^P$ and R/C $\circ$ $\cup^R$, NOT $\cap^P$
$\in$	P	P	$\mathbf{P}^{P,1}$, $\mathbf{P}^{R,1}$ $\mathbf{NP}^{P,5}$, $\mathbf{NP}^{R,2}$
Emptiness	?	?	trivial
Other	Normal form	Normal form	**Normal form**

Normal form

Definition

A (R)P2DCFG is in normal form if all productions have the form $a \to \alpha$ or $a \to {}^t\beta$ with $|\alpha| = |\beta| = 2$.

Remark! Pure grammars do not have normal form!

Theorem

Each P2DCF and RP2DCF grammar is equivalent to a RP2DCFG in normal form.

Example

$$c_i : a \to abcd \rightsquigarrow \begin{cases} c_i^1 : a \to a\alpha_1 \\ c_i^2 : \alpha_1 \to b\alpha_2 \\ c_i^3 : \alpha_2 \to cd \end{cases}$$

Parsing P2DCFL

Theorem

The general problem of the membership of a picture into a language generated by a P2DCFG is NP-complete.

$$\Phi = (x_1 \vee x_2 \vee \neg x_4) \wedge (x_2 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee \neg x_3 \vee x_4)$$

$$p_\Phi = \begin{array}{|ccc|} \hline \mathtt{P} & \mathtt{M} & \mathtt{N} \\ \mathtt{P} & \mathtt{P} & \mathtt{M} \\ \mathtt{M} & \mathtt{N} & \mathtt{N} \\ \mathtt{N} & \mathtt{N} & \mathtt{P} \\ \hline \end{array} \xrightarrow{x_1=1, x_3=0, x_4=1} \begin{array}{|ccc|} \hline 1 & \mathtt{M} & 0 \\ - & - & \mathtt{M} \\ \mathtt{M} & 1 & 1 \\ 0 & 0 & 1 \\ \hline \end{array}$$

Also, $x_1 = 0$, $x_2 = 0$, $x_4 = 1$ and ...

We define a P2DCF grammar G generating the smallest set of pictures over the alphabet $\{P, N, M\}$ which correspond to satisfiable CNF formulae:

$$p_\Phi \in L(G) \text{ iff } \Phi \text{ is satisfiable}$$

- generate the truth table of a satisfiable formula with n clauses over m literals/variables
  - each column has at least one 1

$$p_{\{0,1,M\}} = \begin{array}{r|cccc|} & C_1 & C_2 & \dots & C_n \\ \hline l(x_1) & \cdot & M & & \cdot \\ l(x_2) & 1 & \cdot & & 1 \\ \vdots & & & & \\ l(x_m) & \cdot & 1 & & M \\ \hline \end{array}$$

- build a formula which has $p_{\{0,1,M\}}$ as truth table

	C_1	C_2	$\dots$	C_n
$l(x_1)$	·	M		·
$l(x_2)$	P	·		P
$\vdots$				
$l(x_m)$	·	·		M

$x_2 \to 1$

	C_1	C_2	$\dots$	C_n
x_1	·	M		·
x_2	N	·	·	N
$\vdots$				
x_m	·	·		M

$\neg x_2 \to 1$

Parsing P2DCFL over unary alphabet

Theorem

The parsing of a language generated by a pure 2D context free grammar with unary alphabet is in P.

Parsing problem $\rightsquigarrow$ system of two Diophantine equations.

$$\boxed{h_1 x_1 + h_2 x_2 + \cdots + h_t x_t = n}$$

If $\gcd(h_1, \ldots, h_t) = 1$, every $n \geqslant C$ can be written as a conical combination of $h_1, \ldots, h_t$

Example

$G = (\{a\}, \{c : a \to a^3 \mid a^6\}, \{r : a \to (a^2)^t \mid (a^7)^t\}, a^{(2,3)})$.
Deciding $a^{(n,m)} \in L(G)$ amounts to verifying

$$\begin{cases} x_{c_1}(3-1) + x_{c_2}(6-1) = n-3 \\ y_{r_1}(2-1) + y_{r_2}(7-1) = m-2 \end{cases}$$

has solutions $x_{c_i}, y_{r_i} \geqslant 0$.

Parsing RP2DCFL over (at least) binary alphabet

Theorem

The general problem of the membership of a picture to a language generated by a RP2DCFG with (at least) two symbols is NP-complete.

Definition

Let $\{S_1, \ldots S_n\}$ be a family of finite sets and $C \subseteq \{S_1, \ldots S_n\}$. We say that C is a k-set-covering for $\bigcup_{i=1}^{n} S_i$ when $\bigcup_{S_j \in C} S_j = \bigcup_{i=1}^{n} S_i$ and $|C| \leqslant k$. The *set-covering problem* is defined for a family $\{S_1, \ldots S_n\}$ and with respect to a positive integer k and it requires to check whether there exists a k-set-covering for $\{S_1, \ldots S_n\}$.

We define a RP2DCF grammar defining the language of pictures representing family of sets $S = \{S_1, \ldots, S_n\}$ which are k-set-covering of $\bigcup_{i=1}^{n} S_i = s_1, s_2, \ldots, s_m$

Parsing RP2DCFL over (at least) binary alphabet

$$S = \{S_1, \ldots, S_n\}, \bigcup_{i=1}^{n} S_i = \sigma_1, \sigma_2, \ldots, \sigma_m$$

$$p_S = \begin{array}{c|cccc|} & \sigma_1 & \sigma_2 & \ldots & \sigma_m \\ \hline S_1 & * & X & & * \\ S_2 & X & * & & X \\ \vdots & & & & \\ S_n & X & * & & * \\ \hline \end{array}$$

Generating a picture is in two phase and ruled by $r^{k-1}(r_1 + r_2)$

- build a k-cover of $\bigcup_{i=1}^{n} S_i$

$$r : \{X \to {}^t\alpha, * \to {}^t\beta \mid \alpha \in \{*X \mid X* \mid X^2\}, \beta \in \Sigma^2\}$$

$$\boxed{\begin{matrix} X & X & X \end{matrix}} \xrightarrow{r} \boxed{\begin{matrix} X & * & X \\ * & X & X \end{matrix}}$$

- freely add rows without altering the k-cover by r_1, r_2

$$r_1 : \{X \to {}^t\alpha, * \to {}^t\beta \mid \alpha \in \{X* \mid X^2\}, \beta \in \Sigma^2\}$$

Summary on parsing

	P2DCF	RP2DCF
$\lvert\Sigma\rvert = 1$	**P**	**P**
$\lvert\Sigma\rvert = 2$	?	**NP**-c
$3 \leqslant \lvert\Sigma\rvert \leqslant 4$	?	**NP**-c
$\lvert\Sigma\rvert \geqslant 5$	**NP**-c	**NP**-c

Some closure for (R)P2DCFL - projection

Projection may change the expressiveness of the class of languages of P2DCFG (also RP2DCFG)

Theorem

Let $G = (\Sigma, P^c, P^r, S)$ *be a* $P2DCFG$ *and let* π *be a projection from the alphabet* Σ *to the alphabet* Δ*. Then* $\pi(L(G))$ *is a subset of the language generated by a P2DCFG* $\overline{G}$ *such that* $\pi(L(G)) = L(\overline{G}) \cap \Delta^{++}$*.*

In general, projection requires a non-regular control language.

Some closure for (R)P2DCFL - union

RP2DCFG is closed under union.

Theorem

Let $G_r^1 = (G_1, \Gamma_1, \mathcal{C}_1)$ *and* $G_r^2 = (G_2, \Gamma_2, \mathcal{C}_2)$ *be two RP2DCFG. Then, the language* $L(G_r^1) \cup L(G_r^2)$ *is RP2DCFL.*

- tag the productions tables
- the control language of $L(G_r^1) \cup L(G_r^2)$ is the union of $\mathcal{C}_1$, $\mathcal{C}_2$ (over tagged symbols)

Some closure for (R)P2DCFL - Intersection

Theorem

The family of P2DCFL is not closed under intersection.

$$L_{square(a)} = L_{rect(a)} \cap L$$

where $L = L_{square(a)} \cup L_{spurious}$ is a language containing

- squares of a
- rectangles defined over $\{a, b, c, d\}$ (b, c, d are symbols generating squares but not used along with a control language)

LOC, (R)P2DCFG, RTG & Průša

- LOC and P2DCFG: incomparable & nonempty intersection
- LOC∩RP2DCFG∩Průša$\neq \emptyset$
- RP2DCFG∩Průša$\subsetneq$LOC
- RP2DCFG, Průša (RTG): incomparable

Future works

- Complete membership hierarchy for (R)P2DCFL
  - when membership is in $\mathbf{P}$, RTG $\cap$ (R)P2DCFL $\subseteq$ PG?
- P2DCFL $\cap$ P2DCFL $\overset{?}{\sim}$ (R)P2DCFL
- Family of automata recognizing (R)P2DCFL
- Other closures
  - concatenation of RP2DCFL