On the Complexity of Pure 2D Context-free Grammars

Marcello M. Bersani, Achille Frigeri, Alessandra Cherubini

Politecnico di Milano
bersani@elet.polimi.it,{achille.frigeri,alessandra.cherubini}@polimi.it

1 Introduction

Picture languages generalize classical string languages to two-dimensional arrays. Several approaches have been proposed during the years; consequently, a general classification and a detailed comparison of the classes proposed turns to be necessary. We studied in detail closure properties of (regular) pure 2D context-free grammars (R)P2DCFG [1], and the complexity of the membership problem [2].

2 Preliminaries

General definitions Let Σ be a finite alphabet. A two-dimensional array of elements of Σ is a picture over Σ. The set of all pictures over Σ is denoted by Σ^{++}. For $h, k \geq 0$, $\Sigma^{(h,k)}$ denotes the set of pictures of size (h, k), and $\Sigma^{**} = \Sigma^{++} \cup \lambda$, where λ is the empty picture. Conversely, if $p \in \Sigma^{**}$, we denote by $|p|_{row}$ and $|p|_{col}$, the number of rows and columns of p, respectively. The size of p is the pair $|p| = (|p|_{row}, |p|_{col})$. As in the one-dimensional case, a *picture language* is a subset of Σ^{**}. For $1 \leq i \leq |p|_{row}$, $1 \leq j \leq |p|_{col}$, the element of p in the i-th row and j-th column is called a *pixel* and denoted by $p(i, j)$.

Operations on picture languages Let Γ and Σ be two finite alphabets and $\pi : \Gamma \to \Sigma$ a function between them, if $p \in \Gamma^{(h,k)}$, the *projection* of p by π is the picture $p' \in \Sigma^{(h,k)}$ such that $p'(i, j) = \pi(p(i, j))$, for all $1 \leq i \leq |p|_{row}$, $1 \leq j \leq |p|_{col}$. Projection naturally extends to languages. *Row and column concatenations* are partial operations on pictures denoted $\ominus$ and $\oplus$, respectively. If $p, q \in \Sigma^{(h,*)}$ (resp. $p, q \in \Sigma^{(*,k)}$) $p \oplus q$ (resp. $p \ominus q$) is the horizontal (resp. vertical) juxtaposition of p and q. With $p^{n\oplus}$ (resp. $p^{n\ominus}$) is denoted the horizontal (resp. vertical) juxtaposition of n copies of p; $p^{+\oplus}$ (resp. $p^{+\ominus}$) is the corresponding closure. Concatenations also extend to languages.

Pure 2D Context-free Grammars Context free grammars which make use of only terminal symbols (i.e., *pure grammars*) have been well investigated in the theory of string languages. Pure 2D context-free grammars [3], unlike Matrix grammars ([4,5]), admit rewriting any row/column of pictures with no priority of columns and rows. Row/column sub-arrays of pictures are rewritten in parallel by equal length strings and by using only terminal symbols.

Definition 1. *A* pure 2D context-free grammar (P2DCFG) *is a 4-tuple* $G = (\Sigma, P^c, P^r, S')$ *where:*

1. *Σ is a finite set of symbols;*
2. *$P^c = \{c_i \mid 1 \leq i \leq m\}$ is the set of* column rule tables*, where a table c_i is a set of context-free rules of the form $a \to \alpha$, $a \in \Sigma$, $\alpha \in \Sigma^+$ s.t. for any two rules $a \to \alpha$, $b \to \beta$ in c_i, $|\alpha| = |\beta|$ where $|\alpha|$ denotes the length of α.*
3. *$P^r = \{r_i \mid 1 \leq i \leq n\}$ is the set of* row rule tables*, where a table r_i is a set of context-free rules of the form $a \to {}^t\alpha$, $a \in \Sigma$, $\alpha \in \Sigma^+$ s.t. for any two rules $a \to {}^t\alpha$, $b \to {}^t\beta$ in r_i, $|\alpha| = |\beta|$.*
4. *$S' \subseteq \Sigma^{++}$ is a finite set of* axioms*.*

For any two arrays $p_1, p_2 \in \Sigma^{**}$, p_2 is derived from p_1 in G, in symbols $p_1 \Rightarrow p_2$, if p_2 is obtained from p_1 by either rewriting a column of p_1 by applying to each letter of the column a rule in a table $c_i \in P^c$, or rewriting a row of p_1 by applying to each letter of the row a rule in a table $r_i \in P^r$. The set of symbols occurring in the column (resp. row) that will be rewritten by $c_i \in P^c$ (resp. $r_i \in P^r$) must be a subset of $\{a \mid a \to \alpha \in c_i\}$ (resp. $\{a \mid a \to {}^t\alpha \in r_i\}$). Otherwise, the derivation can not be achieved because there are some symbols for which c_i (resp. r_i) does not provide a rewriting rule. Derivation $\Rightarrow$ is a binary relation over Σ^{**} and its reflexive and transitive closure is denoted by $\overset{*}{\Rightarrow}$. The language $\mathcal{L}(G)$ generated by the P2DCF grammar G is the set $\{p \mid S \overset{*}{\Rightarrow} p \in \Sigma^{++}$ for some $S \in S'\}$. The family of languages generated by some P2DCF grammar is denoted by P2DCFL. It is worth noticing that all pictures derived at each step by applying a rewriting rule from the set P^c or P^r are legal pictures. Since non-terminals are not admitted by P2DCF grammars, each derivation consists of characters of Σ. To augment the expressive power of P2DCF grammars, the sequence of rules to be used can be led by a *control language*.

Definition 2. *A* pure 2D context-free grammar with regular control (RP2DCFG) *is a tuple* $G_r = (G, \Gamma, \mathcal{C})$ *where:*

1. *G is a $P2DCF$ grammar;*
2. *Γ is the control alphabet, actually the set of labels of the rule tables in $P^c \cup P^r$;*
3. *$\mathcal{C} \subseteq \Gamma^*$ is the regular control associated to the grammar.*

If $p \in \Sigma^{**}$ and $S \in S'$, p is derived from S in G_r by means of a control word $w = w_1 w_2 \ldots w_n \in \mathcal{C}$, in symbols $S \Rightarrow_w p$, if p is obtained from S by applying the column/row rules defined by w. The language $\mathcal{L}(G)$ generated by the RP2DCF grammar G_r is the set of pictures $\{p \mid S \Rightarrow_w p \in \Sigma^{++}$ for some $w \in C\}$. The family of languages generated by some RP2DCF grammar is denoted by RP2DCFL. The family P2DCFL is strictly included in RP2DCFL, indeed each P2DCF language is a RP2DCF language with control $C = \Gamma^*$. On the other hand, the language of squares over the symbol a is not a P2DCF language but can be generated by the RP2DCF grammar $(G, \{c, r\}, (cr)^*)$ where $G =$

$(\{a\}, \{c\}, \{r\}, S)$ and $S \to a$, $c = \{a \to aa\}$, $r = \{a \to {}^t(aa)\}$. In order to refine the given definition of this class of grammars, we consider RP2DCFG whose alphabet is $\Sigma = \Sigma_T \cup \Sigma_C$ where Σ_T is the alphabet of final symbol defining the pictures, and Σ_C is a set of auxiliary characters, or *control symbols*, which are involved only in the process of derivation. Yet, control symbols can not be considered as proper non-terminal symbols since they have to be rewritten by means of derivations guided by the control language, so that no control symbols appear in the final picture and the generating device can still be seen as a pure grammar. We showed that the use of control symbols is needed to reach the full expressiveness of RP2DCFG, i.e., there exist RP2DCF languages that can not be defined without the use of control symbols.

3 Normal form and parsing complexity

Normal forms of generating grammars are useful tools to get in a easier way properties of languages and make comparisons between different generating devices. Normal forms, in general, force some constraints on the size and on the alphabet of the strings/pictures occurring in the left and right parts of the productions. Since the model we are considering is a pure grammar, and productions in the row/column tables always rewrite a single character into the right part that is a string (or the transpose of a string), the normal form we ask for has to fix the length of the strings in the right part of each production: formally a (R)P2DCFG is in normal form if all productions have the form $a \to \alpha$ or $a \to {}^t\beta$ with $|\alpha| = |\beta| = 2$. Pure 2D context-free grammars do not have a normal form. Indeed, the language $L_3(a)$ of pictures of size (hn, kn) (where h, k are positive integers) on the alphabet $\{a\}$ are generated by the P2DCFG $(\{a\}, \{c_1\}, \{r_1\}, S)$, where c_1 and r_1 are, respectively, composed by the unique rules $a \to a^n$, $a \to {}^t(a^n)$ and S is the square of a of size (n, n), but no P2DCFG with column and row productions of length 2. However, we have the following.

Proposition 1. *Each (R)P2DCF grammar is equivalent to a RP2DCFG in normal form.*

Actually, a more general result holds: each pure 2D context-free grammar with a control language belonging to a class of languages closed with respect to finite substitutions admits a normal form.

Theorem 1. *The general problem of the membership of a picture into a language generated by a P2DCFG is NP-complete.*

We proved the NP-hardness by providing a (polynomial-time) reduction of SAT to the membership problem for a P2DCFL with an alphabet of at least 5 symbols. So, the question concerning the parsing complexity for pure 2D grammars with smaller alphabets is quite natural. We have the following results.

Theorem 2. *The parsing of a language generated by (R)P2DCF grammars with unary alphabet is in P.*

On the other hand, the NP-completeness of the membership for RP2DCFL characterizes all the languages with at least two symbols. The proof consists of a reduction of the set-covering problem to the membership for RP2DCFG.

Theorem 3. *The general problem of the membership of a picture to a language generated by a RP2DCFG with (at least) two symbols is NP-complete.*

4 Closure properties

In this section, we present some closure properties of the class of $(R)P2DCFL$. Some of them are known from [3] but here we provide new results. First we considered projections:

Proposition 2. *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 $\pi(L(G)) = L(\overline{G}) \cap \Delta^{++}$.*

Proposition 3. *Let $G_{reg} = (G, \Gamma, \mathcal{C})$ with $G = (\Sigma, P^c, P^r, S)$ be a RP2DCFG 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 RP2DCFG $\overline{G}_{reg} = (\overline{G}, \overline{\Gamma} \cup \{c_\pi\}, \overline{\mathcal{C}}\{c_\pi\}^*)$ such that $\pi(L(G_{reg})) = L(\overline{G}_{reg}) \cap \Delta^{++}$.*

The two previous propositions show how projection may change the expressiveness of the class of languages of P2DCFG. A similar result is obtained also for Tiling Systems which are the projection of Local languages. In [3] the authors proved that P2DCFL are not closed under union and under row/column concatenation and proved that the closure under union can be retained when a regular control language is added to the grammars. Yet, no results is provided concerning the closure under intersection. We have the following:

Proposition 4. *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 $\mathcal{L}(G_r^1) \cup \mathcal{L}(G_r^2)$ is RP2DCFL.*

Proposition 5. *The family of P2DCFL is not closed under intersection.*

The family of P2DCFL was shown not to be closed under row/colum concatenation in [3]. We conjecture that this holds also for the family of RP2DCFL.

References

1. Bersani, M.M., Frigeri, A., Cherubini, A.: On Some Classes of 2D Languages and Their Relations. In: IWCIA. (2011) 222–234
2. Bersani, M.M., Frigeri, A., Cherubini, A.: Expressiveness and Complexity Of Regular Pure 2D Context-free Languages (RP2DCFL). International Journal of Computer Mathematics (to appear)
3. Subramanian, K.G., Nagar, A.K., Geethalakshmi, M.: Pure 2D picture grammars (P2DPG) and P2DPG with regular control. In: IWCIA. (2008) 330–341
4. Siromoney, G., Siromoney, R., Krithivasan, K.: Abstract families of matrices and picture languages. Computer Graphics and Image Processing **1** (1972) 284–307
5. Siromoney, G., Siromoney, R., Krithivasan, K.: Picture languages with array rewriting rules. Information and Control **23(5)** (1973) 447–470