CONTEXTUAL ARRAY GRAMMARS

Rudolf Freund[1], Gheorghe Păun[2], Grzegorz Rozenberg[3]
[1]Technical University Wien, Institute for Computer Languages
Resselgasse 3, 1040 Wien, Austria
email: freund@csdec1.tuwien.ac.at, tel. and fax: ++43 1 5041589
[2]Institute of Mathematics of the Romanian Academy of Sciences
Grivitei Str. 21, PO Box 1-764, 70700 Bucureşti
email: gpaun@imar.ro
[3]University of Leiden,Department of Computer Science
Niels Bohrweg 1, 2333 CA Leiden, The Netherlands

Abstract. We consider the (n-dimensional) array contenant of string contextual grammars in the sense of Marcus and investigate the power of such grammars, also with respect to the degree of connectedness, mainly in comparison with context-free array grammars. The main result is that every array language generated by a context-free array grammar is the coding of a language generated by a contextual array grammar.

1 Introduction

For the string case, the contextual grammars were introduced in [3] with motivations arising from descriptive linguistics. Basically, such a contextual grammar consists of a finite set of strings (*axioms*) and a finite set of productions of the form (s, c), where s is a string (the *selector*) and c is a pair of strings (the *context*). Starting from an axiom, iteratively contexts are added as it is indicated by the productions, which yields new strings. There are two modes of adjoining contexts: In the *external* case, contexts are added at the ends [3]; usually infinitely many productions (s, c), with finitely many contexts are used, otherwise the obtained language is finite. In the *internal* case, contexts are added in the interior of strings by bracketting one occurence of a selector with the associated context [4]; the generated language consists of all strings obtained in this way, by finitely but arbitrarily many context adjoinings.

Recently, several variants of these grammars were considered (e. g. [2], [5]). One variant we shall also consider here is the maximal mode of derivation (called t-mode in cooperating grammar systems [1]), i. e. we only retain the set of strings produced derivations which cannot be continued.

The fundamental difference between contextual grammars and Chomsky grammars or Lindenmayer systems is that in contextual grammars we cannot *rewrite* symbols, we only can *adjoin* symbols to the current string; once introduced, a symbol cannnot be modified any more.

2 Definitions

The contextual style of generating strings can be extended to arrays in a natural way: We start from a given finite set of arrays (axioms) and apply to them productions (s, c) where both s and c are finite patterns. If in the current array we can identify a subarray identical with s in such a way that the places corresponding to c are empty, then c is adjoined, producing a new array. In the generated language we may retain either all the arrays produced in this way or only the arrays to which no productions are applicable (maximal derivations).

The family of n-dimensional array languages of k-connected arrays generated by contextual array grammars with at most i symbols in the mode $f \in \{*, t\}$ is denoted by $L(n, cont, k, f, i)$. When no bound on the number of symbols is considered, we replace i by ∞, thus obtaining $L(n, cont, k, f, \infty)$; when no bound on the degree of connectedness is considered, we replace k by ∞, thus obtaining $L(n, cont, \infty, f, i)$.

In the same way, for all $n, k \in N$ (where N is the set of positive integers), the family of n-dimensional array languages of k-connected arrays generated by monotonic respectively context-free array grammars G is denoted by $L(n, mon, k)$ respectively $L(n, cf, k)$; when no bound on the degree of connectedness is considered, we replace k by ∞, thus obtaining $L(n, mon, \infty)$ respectively $L(n, cf, \infty)$. Obviously, for all $n, k \in N$ we obtain

a) $L(n, mon, k) \subsetneqq L(n, mon, k+1)$ and $L(n, cf, k) \subsetneqq L(n, cf, k+1)$;

b) $L(n, cf, k) \subsetneqq L(n, mon, k)$.

Two n-dimensional arrays are called *shape-equivalent,* if they coincide in the positions of the n-dimensional space that they occupy by non-blank symbols; we say they have the same shape. Two n-dimensional array languages are called *shape-equivalent,* if the n-dimensional arrays in these two array languages induce the same set of shapes.

3 Results

Without proofs, we state some of the results we obtained.

Lemma 1. $L(n, cont, k, f, i) \subsetneqq L(n, cont, k+1, f, i)$ for all $f \in \{*, t\}$, $k \in N$, and $i \in N \cup \{\infty\}$.

Lemma 2. For every $n \in N$ as well as all $k \in N \cup \{\infty\}$ and $i \in N$,

$$L(n, cont, k, *, i) \subsetneqq L(n, cont, k, *, i+1), \ L(n, cont, k, t, i) \subsetneqq L(n, cont, k, t, i+1).$$

Theorem 1. For every $n \in N$ as well as $k \in N \cup \{\infty\}$ and $i \in N$, each language in $L(n, cont, k, *, i)$ is shape-equivalent with a language in $L(n, cont, k, t, \infty)$.

Example 1. For $n \geq 2$, the converse of Theorem 1 is not true, e.g. the two-dimensional array language Sq of solid squares is in $L(2, cont, 2, t, 1)$, but no language in $L(2, cont, \infty, *, \infty)$ is shape-equivalent with Sq: In any contextual grammar we start from a finite set of axioms, and in order to generate an arbitrarily large square we must pass through stages when the current array cannot be a square; yet such arrays would belong to the n-dimensional array language generated in the arbitrary derivation mode for any contextual array grammar.

Lemma 3. Every one-dimensional array language in $L(1, cont, 1, t, 1)$ is finite. For $n \geq 2$, array languages in $L(n, cont, 1, t, 1)$ need not be finite.

Theorem 2. Let L be a set of 1-connected one-dimensional arrays over the single letter alphabet $\{a\}$. Then L is regular if and only if $L \in L(1, cont, \infty, *, 1)$.

Lemma 4. For every $n \geq 1$, there exists a language in $L(n, cont, 1, *, 1) - L(n, cont, 1, t, \infty)$.

Lemma 5. There exists an array language L in $L(2, cont, 2, *, 2)$ such that there is no array language L' in $L(2, cont, \infty, *, 1)$ that is shape-equivalent with L.

Lemma 6. There is an array language L in $L(2, cont, 2, t, 2)$ such that there is no context-free array grammar L' which is shape-equivalent with L.

Theorem 3. For every $n \in N$ and all $k \in N \cup \{\infty\}$, $L(n, cont, k, *, \infty) \subsetneqq L(n, mon, k)$.

Theorem 4. For every $n \in N$ and all $k \in N \cup \{\infty\}$, $L(n, cont, k, t, \infty) \subsetneqq L(n, mon, k)$.

The following is the main result of our paper:

Theorem 5. Every context-free array language in $L(n, cf, k)$ is the coding of a contextual array language in $L(n, cont, k, t, \infty)$, hence every context-free array language in $L(n, cf, k)$ is shape–equivalent with a contextual array language in $L(n, cont, k, t, \infty)$.

References

[1] E. Csuhaj-Varjù, J. Dassow, J. Kelemen, and Gh. Păun, *Grammar Systems* (Gordon and Breach, London, 1994).

[2] A. Ehrenfeucht, A. Mateescu, Gh. Păun, G. Rozenberg, and A. Salomaa, On representing RE languages by one-sided internal contextual languages, Technical Report 95-04, Leiden University, 1995.

[3] S. Marcus, Contextual grammars, *Rev. Roum. Math. Pures Appl.,* 14 (1969) pp. 1525-1534.

[4] Gh. Păun and X. M. Nguyen, On the inner contextual grammars, *Rev. Roum. Math. Pures Appl.,* 25 (1980) pp. 641-651.

[5] Gh. Păun, G. Rozenberg, and A. Salomaa, Contextual grammars: erasing, determinism, one-sided contexts. In: G. Rozenberg and A. Salomaa (eds.), *Developments in Language Theory (*World Scientific Publ., Singapore, 1994) pp. 370-388.