Theoretical Computer Science 665 (2017) 13–39

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Unambiguous conjunctive grammars over a one-symbol alphabet ☆

Artur Jeż [a,*], Alexander Okhotin [b]

[a] Institute of Computer Science, University of Wrocław, ul. Joliot-Curie 15, 50-383 Wrocław, Poland
[b] St. Petersburg State University, 14th Line V.O., 29B, Saint Petersburg 199178, Russia

ARTICLE INFO

Article history:
Received 10 September 2015
Received in revised form 2 December 2016
Accepted 6 December 2016
Available online 2 January 2017
Communicated by M. Crochemore

Keywords:
Conjunctive grammars
Ambiguity
Language equations
Undecidability
Unary languages

ABSTRACT

It is demonstrated that unambiguous conjunctive grammars over a unary alphabet $\Sigma = \{a\}$ have non-trivial expressive power, and that their basic properties are undecidable. The key result is that for every base of positional notation, $k \geqslant 11$, and for every one-way real-time cellular automaton operating over the alphabet of base-k digits between $\lfloor \frac{k+9}{4} \rfloor$ and $\lfloor \frac{k+1}{2} \rfloor$, the language of all strings a^n with the base-k representation of the form $1w1$, where w is accepted by the automaton, is described by an unambiguous conjunctive grammar. Another encoding is used to simulate a cellular automaton in a unary language containing almost all strings. These constructions are used to show that for every fixed unambiguous conjunctive language L_0, testing whether a given unambiguous conjunctive grammar generates L_0 is undecidable.

1. Introduction

In ordinary formal grammars, called "context-free" in the literature, the available operations are concatenation and disjunction: each rule defines a concatenation, whereas disjunction is implicit in having multiple rules for a single symbol. *Conjunctive grammars* [21] are an extension of ordinary grammars, which further allows a conjunction operation in any rules. Conjunctive grammars are more of a variant of the definition of grammars than something entirely new, as they maintain the main principle behind the context-free grammars—that of defining the structure of shorter strings by concatenating longer strings with already defined properties—and only extend the set of logical connectives used to combine syntactical conditions. In spite of the increased expressive power of conjunctive grammars, they inherit the subcubic upper bound on the parsing complexity [28], as well as other parsing techniques originally developed for ordinary grammars, such as the *generalized LR* and, recently, the *deterministic LR* [1]. These results make conjunctive grammars suitable for practical use. The work on conjunctive grammars has also led to more general grammar models further equipped with negation [23, 16] or with context operators [4].

Conjunctive grammars over a one-symbol alphabet $\Sigma = \{a\}$ were proved non-trivial by Jeż [10], who constructed a grammar for the language $\{a^{4^n} \mid n \geqslant 0\}$ and extended this construction to describe every so-called *automatic set* [2], that is, a unary language with a regular base-k representation. Subsequent work on such grammars revealed their relatively high expressive power and a number of undecidable properties [11]. Testing whether a given string a^n is generated by a grammar

☆ A preliminary version of this paper was presented at the conference on Developments in Language Theory (DLT 2013, Paris, France, 18–21 June 2013), and its extended abstract appeared in the conference proceedings.

* Corresponding author.
E-mail addresses: aje@cs.uni.wroc.pl (A. Jeż), alexander.okhotin@utu.fi (A. Okhotin).

http://dx.doi.org/10.1016/j.tcs.2016.12.009
0304-3975/

G can be done in time $|G| \cdot n(\log n)^3 \cdot 2^{O(\log^* n)}$ [29], and if n is given in binary notation, this problem is EXPTIME-complete already for a fixed grammar G [12]. Conjunctive grammars over a unary alphabet remain non-trivial even in the special case of grammars with one nonterminal symbol [13]. These results had impact on the study of language equations [17,26], being crucial to understanding their computational completeness over a unary alphabet [15,18]. They are also related to the complexity results for circuits over sets of numbers [20], as conjunctive grammars over a unary alphabet may be regarded as a generalization of those circuits.

Unambiguous conjunctive grammars [24] are an important subclass of conjunctive grammars defined by analogy with ordinary unambiguous grammars, which represent the idea of assigning a unique syntactic structure to every well-formed sentence. Little is known about their properties, besides a parsing algorithm with running time $|G| \cdot O(n^2)$, where n is the length of the input [24]; for a unary alphabet, the running time can be improved to $|G| \cdot n(\log n)^2 \cdot 2^{O(\log^* n)}$ [29]. However, all the known results on the expressive power of conjunctive grammars over a unary alphabet [10–12,30] rely upon ambiguous grammars, and it is not even known whether unambiguous grammars can describe anything non-regular.

This paper sets off by presenting the first example of an unambiguous conjunctive grammar that describes a non-regular unary language. This is the same language $\{a^{4^n} \mid n \geqslant 0\}$, yet the grammar describing it, given in Section 3, must operate more carefully than the known ambiguous grammar. This example is then extended to a construction of unambiguous conjunctive grammars generating all languages of the form $L_{k,c} = \{a^{c \cdot k^n} \mid n \geqslant 0\}$, with $k \geqslant 2$ and $c \in \{k, k+1, \ldots, k^2-1\}$; in other words, these are languages of all strings a^n with the base-k notation of n of the form $(ij0^*)_k$, where i and j are base-k digits. These languages serve as building blocks in the subsequent constructions.

Then the paper proceeds with reimplementing, using unambiguous grammars, the main general method for constructing conjunctive grammars over a unary alphabet. This method involves simulating a one-way real-time cellular automaton [6,9,22] over an input alphabet $\Sigma_k = \{0, 1, \ldots, k-1\}$ of base-k digits, by a grammar generating all strings a^n with the base-k representation of n accepted by the cellular automaton. The known construction of such conjunctive grammars [11] always produces ambiguous concatenations. This paper defines a different simulation, under the assumption that the input alphabet of the cellular automaton is not the entire set of base-k digits, but rather some subset of size around $\frac{k}{4}$. With this restriction, the automaton can be simulated, so that all concatenations in the grammar remain unambiguous.

The simulation of a cellular automaton presented in Section 4 produces languages that grow exponentially fast, that is, the length of the n-th shortest string in the language is exponential in n. These languages have *density 0*, in the sense that the fraction of strings of length up to ℓ belonging to these languages tends to 0. As the concatenation of any two unary languages of non-zero density is always ambiguous, this limitation of the given construction might appear to be inherent to unambiguous conjunctive grammars. Nevertheless, a method for describing unary languages of non-zero density by an unambiguous conjunctive grammar is established in the next Section 5. The construction is based on the representation of a^* as an unambiguous concatenation of several languages of density 0, given by Enflo et al. [8]. In particular, some unary languages of density 1 can be described.

The results in Sections 4–5 are weaker than the original results on ambiguous conjunctive grammars over a one-symbol alphabet [11]. It remains unknown whether every unary language with its base-k representation recognized by a trellis automaton is therefore defined by an unambiguous conjunctive grammar. Nevertheless, the constructions in this paper turn out to be sufficient for proving uniform undecidability results for unambiguous conjunctive grammars, presented in Section 6. Consider that for ordinary grammars, testing for equality to the empty set is decidable, whereas equality to the set of all strings over a multi-symbol alphabet is undecidable; for the unambiguous case of ordinary grammars, testing equality to any regular language is decidable [31]. In contrast, for unambiguous conjunctive grammars, for every fixed language L_0 (over an arbitrary alphabet) described by some unambiguous conjunctive grammar, it is proved that testing whether a given unambiguous conjunctive grammar describes L_0 is undecidable.

In the last Section 7, it is shown that unambiguous conjunctive grammars can describe some sparse unary languages that grow arbitrarily fast—the same result as for the general case of conjunctive grammars [11].

2. Conjunctive grammars and ambiguity

In ordinary formal grammars (Chomsky's "context-free"), each rule defines all strings representable as a *concatenation* of substrings with the given properties. In conjunctive grammars, a rule is a *conjunction* of such concatenations, meaning all strings that can be represented as each of the listed concatenations at the same time.

Definition 1 *(Okhotin [21]).* A conjunctive grammar is a quadruple $G = (\Sigma, N, R, S)$, in which Σ and N are disjoint finite non-empty sets of terminal and nonterminal symbols respectively; R is a finite set of rules of the form

$$A \to \alpha_1 \& \ldots \& \alpha_n, \qquad \text{with } A \in N, n \geqslant 1 \text{ and } \alpha_1, \ldots, \alpha_n \in (\Sigma \cup N)^*, \tag{$*$}$$

and $S \in N$ is a nonterminal designated as the initial symbol. A grammar is called *linear conjunctive*, if each α_i in each rule $(*)$ contains at most one nonterminal symbol.

Informally, a rule $(*)$ in a conjunctive grammar means that every string described by each *conjunct* α_i is therefore described by A. One way of formalizing this understanding is by *language equations*, in which the conjunction is interpreted as intersection of languages, as follows.

Definition 2. Let $G = (\Sigma, N, R, S)$ be a conjunctive grammar. The *associated system of language equations* is the following system in variables N.

$$A = \bigcup_{A \to \alpha_1 \& \ldots \& \alpha_n \in R} \bigcap_{i=1}^{n} \alpha_i \quad \text{(for all } A \in N\text{)}$$

Here each α_i in the equation is a concatenation of variables and constant languages $\{a\}$ representing terminal symbols (or constant $\{\varepsilon\}$ if α_i is the empty string). Let $(\ldots, L_A, \ldots)$ be its least solution (that is, such a solution that every other solution $(\ldots, L'_A, \ldots)$ has $L_A \subseteq L'_A$ for all $A \in N$) and denote $L_G(A) := L_A$ for each $A \in N$. Define $L(G) := L_G(S)$.

As in the case of ordinary grammars [3], every such system of equations has a least solution expressible through fixpoint iteration, because the right-hand sides of the system are monotone and continuous.

An equivalent definition of conjunctive grammars is given by *term rewriting*, generalizing the most common definition of ordinary grammars by string rewriting.

Definition 3. Given a conjunctive grammar G, consider terms over concatenation and conjunction with symbols from $\Sigma \cup N$ as atomic terms. The relation $\Longrightarrow$ of immediate derivability on the set of terms is defined as follows.

- Using a rule $A \to \alpha_1 \& \ldots \& \alpha_n$, a subterm $A \in N$ of any term $\varphi(A)$ can be rewritten as $\varphi(A) \Longrightarrow \varphi(\alpha_1 \& \ldots \& \alpha_n)$.
- A conjunction of several identical strings can be rewritten by one such string: $\varphi(w \& \ldots \& w) \Longrightarrow \varphi(w)$, for every $w \in \Sigma^*$.

The language generated by a term φ is $L_G(\varphi) = \{w \mid w \in \Sigma^*,\ \varphi \Longrightarrow^* w\}$. The language generated by the grammar is $L(G) = L_G(S) = \{w \mid w \in \Sigma^*,\ S \Longrightarrow^* w\}$.

Any finite intersection of ordinary context-free languages, such as $\{a^n b^n c^n \mid n \geqslant 0\}$, can be directly represented by a conjunctive grammar. The expressive power of conjunctive grammars actually goes beyond such intersections: for instance, they can represent the language $\{wcw \mid w \in \{a, b\}^*\}$ [21], which is known to be outside the intersection closure of the ordinary context-free languages.

This paper concentrates on a subclass of conjunctive grammars defined by an analogy with the unambiguous case of ordinary grammars. Let a concatenation $L_1 \cdot \ldots \cdot L_k$ be called *unambiguous* if every string $w \in L_1 \cdot \ldots \cdot L_k$ has a unique partition $w = u_1 \ldots u_k$ with $u_i \in L_i$.

Definition 4 *(Okhotin [24]).* Let $G = (\Sigma, N, R, S)$ be a conjunctive grammar.

I. The choice of a rule in G is unambiguous, if different rules for every single nonterminal A generate disjoint languages, that is, for every string w there exists at most one rule $A \to \alpha_1 \& \ldots \& \alpha_m$ with $w \in L_G(\alpha_1) \cap \ldots \cap L_G(\alpha_m)$.
II. Concatenation in G is said to be unambiguous, if for every conjunct $\alpha = X_1 \ldots X_\ell$, the concatenation $L_G(X_1) \cdot \ldots \cdot L_G(X_\ell)$ is unambiguous.

If both conditions are satisfied, then G is called an *unambiguous conjunctive grammar*.

Definition 4 implies that every string in $L(G)$ has a unique parse tree. The converse is untrue: some grammars define unique parse trees, but condition II does not hold.

For more information about conjunctive grammars, an interested reader is directed to a recent survey [27].

3. Representing powers of *k*

Consider the following grammar describing the language $\{a^{4^n} \mid n \geqslant 0\}$, which was the first example of a conjunctive grammar over a unary alphabet representing a non-regular language. Even though much was learned about those grammars since this example, it still remains the smallest and the easiest to understand.

Example 1 *(Jeż [10]).* The following conjunctive grammar with the initial symbol A_1 describes the language $L(G) = \{a^{4^n} \mid n \geqslant 0\}$.

$$\begin{aligned} A_1 &\to A_1A_3 \,\&\, A_2A_2 \mid a \\ A_2 &\to A_1A_1 \,\&\, A_2A_6 \mid aa \\ A_3 &\to A_1A_2 \,\&\, A_6A_6 \mid aaa \\ A_6 &\to A_1A_2 \,\&\, A_3A_3 \end{aligned}$$

Each nonterminal symbol A_i describes the language $L_G(A_i) = \{a^{i \cdot 4^n} \mid n \geqslant 0\}$.

The grammar is best explained in terms of base-4 notation of the lengths of the strings. Let $\Sigma_4 = \{0, 1, 2, 3\}$ be the alphabet of base-4 digits, and for every string $w \in \Sigma_4^*$, let $(w)_4$ denote the integer with base-4 notation w. For any $L \subseteq \Sigma_4^*$, denote $a^{(L)_4} = \{a^{(w)_4} \mid w \in L\}$. Then the languages described by the nonterminals of the above grammar are $a^{(10^*)_4}$, $a^{(20^*)_4}$, $a^{(30^*)_4}$ and $a^{(120^*)_4}$.

Consider the system of language equations corresponding to the grammar. For instance, the equation for A_1 is as follows.

$$A_1 = (A_1A_3 \cap A_2A_2) \cup \{a\}$$

Substituting the given four languages into the intersection $A_1A_3 \cap A_2A_2$ in the first equation, one obtains the following language.

$$a^{(10^*)_4}a^{(30^*)_4} \cap a^{(20^*)_4}a^{(20^*)_4} = \left(a^{(10^+)_4} \cup a^{(10^*30^*)_4} \cup a^{(30^*10^*)_4}\right) \cap \left(a^{(10^+)_4} \cup a^{(20^*20^*)_4}\right) = a^{(10^+)_4}$$

That is, both concatenations contain some garbage, yet the garbage in the concatenations is disjoint, and is accordingly filtered out by the intersection. Finally, the union with $\{a\}$ yields the language $\{a^{4^n} \mid n \geqslant 0\}$, and thus the first equation turns into an equality. The rest of the equations are verified similarly, and hence the given four languages form a solution. By a standard argument [3, Thm. 2.3], one can prove that the system has a unique ε-free solution. Therefore, the four languages substituted above must be this solution, and this must be the least solution of the system.

The grammar in Example 1 is ambiguous, because of the concatenations A_1A_1, A_2A_2, A_3A_3 and A_6A_6: indeed, since concatenation of unary strings is commutative, a concatenation of a language with itself is unambiguous only if this language is empty or a singleton. However, it is possible to remake the above grammar without ever concatenating a nonterminal to itself, though that requires defining a larger collection of languages. The following grammar becomes the first evidence of non-triviality of unambiguous conjunctive grammars over a unary alphabet.

Example 2. The below conjunctive grammar is unambiguous and describes the language $\{a^{4^n} \mid n \geqslant 0\}$.

$$\begin{aligned}
A_1 &\to A_1A_3 \,\&\, A_7A_9 \mid a \mid a^4\\
A_2 &\to A_1A_7 \,\&\, A_2A_6 \mid a^2\\
A_3 &\to A_1A_2 \,\&\, A_3A_9 \mid a^3\\
A_6 &\to A_1A_2 \,\&\, A_9A_{15} \mid a^6\\
A_7 &\to A_1A_3 \,\&\, A_1A_6\\
A_9 &\to A_1A_2 \,\&\, A_2A_7\\
A_{15} &\to A_6A_9 \,\&\, A_2A_7
\end{aligned}$$

Each nonterminal A_i describes the language $L_G(A_i) = \{a^{i \cdot 4^n} \mid n \geqslant 0\}$.

The correctness is established in the same way as in Example 1. For instance, the first equation is checked as follows.

$$\begin{aligned}
&a^{(10^*)_4}a^{(30^*)_4} \cap a^{(130^*)_4}a^{(210^*)_4}\\
&\quad = \left(a^{(10^+)_4} \cup a^{(10^*30^*)_4} \cup a^{(30^*10^*)_4}\right) \cap \left(a^{(10^{\geqslant 2})_4} \cup a^{(2110^*)_4} \cup a^{(2230^*)_4} \cup a^{(130^*210^*)_4} \cup a^{(210^*130^*)_4}\right)\\
&\quad = a^{(10^{\geqslant 2})_4}
\end{aligned}$$

Furthermore, the form of both concatenations is simple enough to see that they are unambiguous. For example, in the concatenation A_1A_3, each string of the form $a^{(10^n)_4}$ is produced in a unique way by concatenating $a^{4^{n-1}} \in L_G(A_1)$ to $a^{3 \cdot 4^{n-1}} \in L_G(A_3)$; every string $a^{(10^{m-n-1}30^n)_4}$ is produced uniquely by concatenating a^{4^m} to $a^{3 \cdot 4^n}$; the same argument applies to all strings in $a^{(30^*10^*)_4}$. The choice of a rule is always unambiguous, because there is only one non-terminating rule for each nonterminal, and all strings it generates are longer than any strings generated by the terminating rules.

More generally, for every alphabet $\Sigma_k = \{0, 1, \ldots, k-1\}$ of base-k digits, let $(w)_k$ denote the number with the base-k notation w, and let $a^{(L)_k} = \{a^{(w)_k} \mid w \in L\}$ for every language $L \subseteq \Sigma_k^*$. Then the following lemma states that the concatenation of a language $a^{(ij0^*)_k}$ with a language $a^{(i'j'0^*)_k}$ is *in most cases* unambiguous, and none of the concatenations actually used in the grammar are among the few exceptions to this rule.

Lemma 1. *Let $k \geqslant 2$, and consider any two different languages of the form $K = a^{(ij0^*)_k}$ and $L = a^{(i'j'0^*)_k}$, with $i, i' \in \Sigma_k \setminus \{0\}$ and $j, j' \in \Sigma_k$, except those with $i = j = i'$ and $j' = 0$, or vice versa. Then the concatenation KL is unambiguous.*

Proof. The goal is to show that if $a^{(ij0^\ell)_k}a^{(i'j'0^{\ell'})_k} = a^{(ij0^m)_k}a^{(i'j'0^{m'})_k}$, or, in other words, if

$$(k \cdot i + j)(k^\ell - k^m) = (k \cdot i' + j')(k^{m'} - k^{\ell'}) \tag{1}$$

for i, j, i', j' as in the statement, then $\ell = m$ and $\ell' = m'$.

It is sufficient to show any of the two equalities $\ell = m$ and $\ell' = m'$, because each of them implies the other. Indeed, if $\ell = m$, then the left-hand side of Eq. (1) is zero, and therefore its right-hand side is zero as well, which holds only if $\ell' = m'$, as claimed. Similarly, $\ell' = m'$ implies $\ell = m$.

Assume that $\ell \neq m$ and $\ell' \neq m'$; by the symmetry, it may be further assumed that $\ell > m$. Then the left-hand side of Eq. (1) is non-negative, and hence its right-hand side is non-negative as well, which implies that $m' > \ell'$. Thus, the equality (1) can be rewritten as

$$(k \cdot i + j)k^m(k^{\ell-m} - 1) = (k \cdot i' + j')k^{\ell'}(k^{m'-\ell'} - 1). \tag{2}$$

In the following, the analysis splits, depending on whether $j = 0$ or $j' = 0$.

Consider first the case, when both digits j and j' are non-zero. Then the left-hand side of (2) is divisible by k^m, but not by k^{m+1}, whereas the number on the right-hand side is divisible by $k^{\ell'}$, but not by $k^{\ell'+1}$. Therefore, $\ell' = m$, and Eq. (2) is simplified to

$$(k \cdot i + j)(k^{\ell-m} - 1) = (k \cdot i' + j')(k^{m'-\ell'} - 1). \tag{3}$$

Since $K \neq L$, it holds that $i \neq i'$ or $j \neq j'$. Together with the assumption that $0 < i, j, i', j' < k$, this implies $ki + j \neq ki' + j'$. By the symmetry of Eq. (3), it may be assumed that $k \cdot i + j > k \cdot i' + j'$. Then, for the equality (3) to hold, $k^{m'-\ell'}$ should be strictly greater than $k^{\ell-m}$, and since both numbers are powers of k, this requires $m' - \ell' - 1 \geqslant \ell - m$. This is enough to estimate both sides of Eq. (3) with contradictory values.

$$(k \cdot i + j)(k^{\ell-m} - 1) < k^2(k^{m'-\ell'-1} - 1) = k^{m'-\ell'+1} - k^2$$
$$(k \cdot i' + j')(k^{m'-\ell'} - 1) \geqslant k(k^{m'-\ell'} - 1) \geqslant k^{m'-\ell'+1} - k$$

Together, those two estimations imply that $k^{m'-\ell'+1} - k^2 > k^{m'-\ell'+1} - k$, which is not possible, as $k > 1$. This contradiction shows that $\ell = m$ and $\ell' = m'$, as desired.

The second case, of $j = j' = 0$, follows by a similar argument as in the first case: the difference is that the left-hand side of (2) in this case is divisible by k^{m+1}, but not by k^{m+2}, whereas the number on the right-hand side is divisible by $k^{\ell'+1}$, but not by $k^{\ell'+2}$, and therefore $\ell' = m$. Then the equality (2) can be represented as

$$i(k^{\ell-m} - 1) = i'(k^{m'-\ell'} - 1).$$

Since $\ell > m$, the left-hand side is equivalent to $-i$ modulo k and, as $m' > \ell'$, the right-hand side is equivalent to $-i'$ modulo k. Since $0 < i, i' < k$, it is concluded that $i = i'$, contradiction.

It is left to consider the case, in which exactly one of j and j' is zero. Assume that $j \neq 0$ and $j' = 0$. Then the number on the left-hand side of (2) is divisible by k^m, but not by k^{m+1} and the number on the right-hand side is divisible by $k^{\ell'+1}$, but not by $k^{\ell'+2}$. Then $m = \ell' + 1$, and the equality (2) is simplified to

$$(k \cdot i + j)(k^{\ell-m} - 1) = i'(k^{m'-\ell'} - 1).$$

Modulo k, the left-hand side is equivalent to $-j$, whereas the right-hand is $-i'$, and therefore $j = i'$. Since the case $i = i' = j$ and $j' = 0$ is excluded in the statement of the lemma, this is only possible if $i \neq i'$. The equation can be thus further simplified to

$$i \cdot k^{\ell-m+1} + i' \cdot k^{\ell-m} = i' \cdot k^{m'-\ell'} + i \cdot k.$$

The base-k notation of the number on the left-hand side is $(ii'0^{\ell-m})_k$, and the sum of its digits is $i + i'$. For the number on the right-hand side, if $m' - \ell' = 1$, then there is a carry in the addition of $i'k$ and ik, and the sum of digits in the sum is $i + i' - k + 1 < i + i'$, and hence it is not the same number as $(ii'0^{\ell-\ell'})_k$. If $m' > \ell' + 1$, then the base-k notation of the number on the right-hand side is in $(i'0^*i0)_k$, and so the leading digit of the number on the right-hand side must be i', contradiction. This leaves the only possibility that the number is $(ii')_k$, and $\ell = m$ and $\ell' = m'$. □

Using concatenations of the form defined in Lemma 1, Example 2 can be generalized to construct unambiguous grammars for all languages $L_k = \{a^{k^n} \mid n \geqslant 1\}$, with $k \geqslant 9$.

Lemma 2. *For every $k \geqslant 9$, the following conjunctive grammar with the set of nonterminals $N = \{A_{i,j} \mid i, j \in \{0, \ldots, k-1\},\ i \neq 0\}$ and with the initial symbol $A_{1,0}$ is unambiguous and describes the language $a^{(10^+)_k}$.*

$$A_{1,j} \to A_{k-1,0}A_{j+1,0} \,\&\, A_{k-2,0}A_{j+2,0} \mid a^{(1j)_k} \qquad \text{for } j < \tfrac{k}{3} + 2$$
$$A_{i,j} \to A_{i-1,k-1}A_{j+1,0} \,\&\, A_{i-1,k-2}A_{j+2,0} \mid a^{(ij)_k} \qquad \text{for } i \geqslant 2,\ j < \tfrac{k}{3} + 2$$
$$A_{i,j} \to A_{i,j-1}A_{1,0} \,\&\, A_{i,j-2}A_{2,0} \mid a^{(ij)_k} \qquad \text{for } i \geqslant 1,\ j \geqslant \tfrac{k}{3} + 2$$

In particular, each nonterminal $A_{i,j}$ generates the language $a^{(ij0^*)_k}$.

Proof. The first claim is that the system of language equations corresponding to the grammar has a unique solution in ε-free languages, with $A_{i,j} = a^{(ij0^*)_k}$ for each i, j. The intended solution $A_{i,j} = a^{(ij0^*)_k}$ is ε-free, and one can prove by a standard argument [3] that this system has a unique solution in ε-free languages. So it is enough to check that the given values are a solution. To this end, the system will be evaluated under the substitution $A_{i,j} = a^{(ij0^*)_k}$.

The substitution involves a lot of manipulations with positional notation of numbers, and therefore it is more convenient to represent the system of language equations as a system of equations over sets of natural numbers, with unknowns $X_{i,j} \subseteq \mathbb{N}$. The concatenation of languages is replaced by the following addition operation on sets of numbers: $S + T = \{m + n \mid m \in S, n \in T\}$. Then the system of equations takes the following form.

$$X_{1,j} = \big[\big(X_{k-1,0} + X_{j+1,0}\big) \cap \big(X_{k-2,0} + X_{j+2,0}\big)\big] \cup \{(1j)_k\} \qquad \text{for } j < \tfrac{k}{3} + 2 \tag{4a}$$

$$X_{i,j} = \big[\big(X_{i-1,k-1} + X_{j+1,0}\big) \cap \big(X_{i-1,k-2} + X_{j+2,0}\big)\big] \cup \{(ij)_k\} \qquad \text{for } i \geqslant 2,\ j < \tfrac{k}{3} + 2 \tag{4b}$$

$$X_{i,j} = \big[\big(X_{i,j-1} + X_{1,0}\big) \cap \big(X_{i,j-2} + X_{2,0}\big)\big] \cup \{(ij)_k\} \qquad \text{for } i \geqslant 1,\ j \geqslant \tfrac{k}{3} + 2 \tag{4c}$$

An equation of the first type (4a), with $j < \frac{k}{3} + 2$, is verified by checking the following equality.

$$(1j0^+)_k = \big(((k-1)0^+)_k + ((j+1)0^+)_k\big) \cap \big(((k-2)0^+)_k + ((j+2)0^+)_k\big). \tag{5}$$

The two sets of numbers to be intersected are calculated separately as follows.

$$((k-1)0^+)_k + ((j+1)0^+)_k = ((k-1)0^*(j+1)0^+)_k \cup (1j0^+)_k \cup ((j+1)0^*(k-1)0^+)_k$$

$$((k-2)0^+)_k + ((j+2)0^+)_k = ((k-2)0^*(j+2)0^+)_k \cup (1j0^+)_k \cup ((j+2)0^*(k-2)0^+)_k$$

Their intersection obviously contains $(1j0^+)_k$, and it remains to see that no other numbers get into it, that is, that intersections of all other components are empty.

$$((k-1)0^*(j+1)0^+)_k \cap ((k-2)0^*(j+2)0^+)_k = \varnothing$$

$$((k-1)0^*(j+1)0^+)_k \cap ((j+2)0^*(k-2)0^+)_k = \varnothing$$

$$((j+1)0^*(k-1)0^+)_k \cap ((k-2)0^*(j+2)0^+)_k = \varnothing$$

$$((j+1)0^*(k-1)0^+)_k \cap ((j+2)0^*(k-2)0^+)_k = \varnothing$$

In each pair of sets being intersected, either set has all elements with the same leading digit, and those digits are different for the two sets: $k-1 \neq k-2$ in the first line; in the second line, $k-1 \neq j+2$, because $j+2 < \frac{k}{3}+4 < k-1$ for $k \geqslant 9$; then, $j+1 \neq k-2$ for the same reason; and $j+1 \neq j+2$. Thus (4a) holds, and the subsequent union with $(1j)_k$ on the right-hand side of (4a) produces the set $(1j0^*)_k$, and the equation holds true.

Similar calculations are performed for each equation (4b) with $i \geqslant 2$ and $j < \frac{k}{3} + 2$. The following equality is claimed.

$$\big(\underbrace{((i-1)(k-1)0^*)_k + ((j+1)0^+)_k}_{S_1}\big) \cap \big(\underbrace{((i-1)(k-2)0^*)_k + ((j+2)0^+)_k}_{S_2}\big) = (ij0^+)_k \tag{6}$$

Denote the two sums on the left-hand side of (6) by $S_1 = ((i-1)(k-1)0^*)_k + ((j+1)0^+)_k$ and $S_2 = ((i-1)(k-2)0^*)_k + ((j+2)0^+)_k$. Consider the first sum. The summand sets contain numbers with digits $i-1$ and $j+1$, which can be added to each other. Thus the form of the sum depends on whether $i + j < k$, in which case a digit $i + j$ is obtained, or $i + j \geqslant k$, when a digit $i + j - k$ with a carry appears instead.

$$\begin{aligned} S_1 = {} & ((i-1)(k-1)0^*(j+1)0^+)_k \cup (ij0^+)_k \cup \\ & ((i+j)(k-1)0^*)_k \cup ((j+1)0^*(i-1)(k-1)0^*)_k \qquad (\text{if } i+j<k) \end{aligned}$$

$$\begin{aligned} S_1 = {} & ((i-1)(k-1)0^*(j+1)0^+)_k \cup (ij0^+)_k \cup \\ & (1(i+j-k)(k-1)0^*)_k \cup ((j+1)0^*(i-1)(k-1)0^*)_k \qquad (\text{if } i+j\geqslant k) \end{aligned}$$

Similarly, the form of the second sum depends on whether $i + j + 1 < k$ or not.

$$\begin{aligned} S_2 = {} & ((i-1)(k-2)0^*(j+2)0^+)_k \cup (ij0^+)_k \cup \\ & ((i+j+1)(k-2)0^*)_k \cup ((j+2)0^*(i-1)(k-2)0^*)_k \qquad (\text{if } i+j+1<k) \end{aligned}$$

$$\begin{aligned} S_2 = {} & ((i-1)(k-2)0^*(j+2)0^+)_k \cup (ij0^+)_k \cup \\ & (1(i+j-k+1)(k-2)0^*)_k \cup ((j+2)0^*(i-1)(k-2)0^*)_k \qquad (\text{if } i+j+1\geqslant k) \end{aligned}$$

Both S_1 and S_2 contain $(ij0^*)_k$ in each case, and therefore their intersection also contains $(ij0^*)_k$. It remains to show that all other intersections of any two components in the above representations of S_1 and of S_2 is empty. This time, the numbers are distinguished by their *last (least significant) non-zero digits*: all numbers in S_1 (except for numbers from $(ij0^*)_k$)

have $j+1$ or $k-1$, whereas all numbers in S_2 (other than the numbers from $(ij0^*)_k$) have $j+2$ or $k-2$. As in the previous case, these numbers are distinct, which proves (6), and thus the equation (4b) is checked.

The last case of an equation (4c), with $i \geqslant 1$ and $j \geqslant \frac{k}{3}+2$, is proved by a similar argument. It following equality is to be shown.

$$\Big(\underbrace{(i(j-1)0^*)_k + (10^+)_k}_{S'_1}\Big) \cap \Big(\underbrace{(i(j-2)0^*)_k + (20^+)_k}_{S'_2}\Big) = (ij0^+)_k \tag{7}$$

Denote $S'_1 = (i(j-1)0^*)_k + (10^+)_k$ and $S'_2 = (i(j-2)0^*)_k + (20^+)_k$. The form of the first sum on the left-hand side of (7) is different for $i < k-1$ and for $i = k-1$.

$$S'_1 = (i(j-1)0^*10^+)_k \cup (ij0^+)_k \cup ((i+1)(j-1)0^*)_k \cup (10^*i(j-1)0^*)_k \qquad (\text{if } i < k-1) \tag{8a}$$

$$S'_1 = (i(j-1)0^*10^+)_k \cup (ij0^+)_k \cup (1(i+1-k)(j-1)0^*)_k \cup (10^*i(j-1)0^*)_k \qquad (\text{if } i = k-1) \tag{8b}$$

The second sum on the right-hand side of (7) is similarly expanded.

$$S'_2 = (i(j-2)0^*20^+)_k \cup (ij0^+)_k \cup ((i+2)(j-2)0^*)_k \cup (20^*i(j-2)0^*)_k \qquad (\text{if } i < k-2) \tag{8c}$$

$$S'_2 = (i(j-2)0^*20^+)_k \cup (ij0^+)_k \cup (1(i+2-k)(j-2)0^*)_k \cup (20^*i(j-2)0^*)_k \qquad (\text{if } i \geqslant k-2) \tag{8d}$$

Regardless of the value of i, both sums contain the subset $(ij0^+)_k$, which hence belongs to the intersection. To see that there is nothing else in the intersection, one has to show that any other two sets in the above representations of S'_1 and of S'_2 are disjoint. These sets are again distinguished by their last non-zero digit: the numbers in S'_1 (other than in $(ij0^+)_k$) have either 1 or $j-1$ as the last non-zero digit, whereas the numbers from S'_2 (again except those in $(ij0^+)_k$) have either 2 or $j-2$. By the case assumption, $j \geqslant \frac{k}{3}+2 \geqslant 5$, and so $1 < 2 < j-2 < j-1$. Therefore, the numbers in S'_1 are different those in S'_2, and Eq. (7) holds.

To see that the choice of a rule in the grammar is unambiguous, note that each of the nonterminals has only two rules, one terminating and the other non-terminating. For the first type of rules, that is, for $i = 1$ and $j < \frac{k}{3}+2$, observe that the set $(1j0^+)_k$ represents the lengths of strings generated by the non-terminating rule, as demonstrated in (5); this is strictly larger than the length of the constant string $(1j)_k$ in this case. Similar argument applies in the other two cases, with (6) and (7) giving the lengths of the strings generated by the non-terminating rules in these cases.

Thus it was proved that each nonterminal $A_{i,j}$ describes the language $a^{(ij0^*)_k}$ and that the choice of a rule in the grammar is unambiguous. By Lemma 1, the concatenation $A_{i,j}A_{i',j'}$ is unambiguous, unless

1. $i = i'$ and $j = j'$, or
2. $i = i' = j$ and $j' = 0$, or
3. $i = i' = j'$ and $j = 0$.

So it is enough to show that none of the forbidden cases takes place in any rules of each of the three types.

- Consider the rules for $A_{1,j}$ with $j < \frac{k}{3}+2$. Then there are two concatenations in this rule: $A_{k-1,0}A_{j+1,0}$ and $A_{k-2,0}A_{j+2,0}$. By the case assumption $j < \frac{k}{3}+2$ and $k \geqslant 9$ and thus $j+2 < \frac{k}{3}+4 \leqslant k-2$. Consequently $k-1 > k-2 > k+2 > j+1$ and so both concatenations in this rule are unambiguous.
- Let $i \geqslant 2$ and $j < \frac{k}{3}+2$. In this case there are two concatenations in the rule: $A_{i-1,k-1}A_{j+1,0}$ and $A_{i-1,k-2}A_{j+2,0}$. Consider the former. Since $k-1 \neq 0$ the only case, in which this concatenation could be unambiguous, is when $i-1 = j+1 = k-1$. However, as in the previous case, $j+1 < k-1$ and so this cannot hold. Similarly, the concatenation $A_{i-1,k-2}A_{j+2,0}$ is unambiguous, as $k-2 > j+2$.
- In the last case, when $i \geqslant 1$ and $j \geqslant \frac{k}{3}+2$ there are also two concatenations in the rule: $A_{i,j-1}A_{1,0}$ and $A_{i,j-2}A_{2,0}$. Consider the latter: since $j-2 > 0$, the only case in which this concatenation may be unambiguous, is when $i = 2 = j-2$. However, by the case assumption $j \geqslant \frac{k}{3}+2 \geqslant 5$. Hence $j-2 \geqslant 3$. Similar analysis shows also that the concatenation $A_{i,j-1}A_{1,0}$ is unambiguous.

Hence, Lemma 1 asserts that all concatenations in the grammar are unambiguous. □

4. Simulating trellis automata

In this section, the known general method for constructing conjunctive grammars over a unary alphabet [11] is extended to the case of unambiguous conjunctive grammars. The overall idea is to simulate a *one-way real-time cellular automaton* operating on base-k representations of numbers, by a grammar describing unary representations of the same numbers.

A one-way real-time cellular automaton, also known as a *trellis automaton* [6,7,22], processes an input string of length $n \geqslant 1$ using a uniform array of $\frac{n(n+1)}{2}$ nodes, as presented in Fig. 1. Each node computes a value from a fixed finite set Q. The nodes in the bottom row obtain their values directly from the input symbols using a function $I\colon \Sigma \to Q$. The rest of

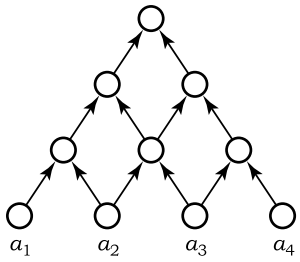

Fig. 1. Computation of a trellis automaton.

the nodes compute the function $\delta\colon Q \times Q \to Q$ of the values in their predecessors. The string is accepted if and only if the value computed by the topmost node belongs to the set of accepting states $F \subseteq Q$.

Definition 5. A trellis automaton is a quintuple $M = (\Sigma, Q, I, \delta, F)$, in which:

- Σ is the input alphabet,
- Q is a finite non-empty set of states,
- $I\colon \Sigma \to Q$ is a function that sets the initial states,
- $\delta\colon Q \times Q \to Q$ is the transition function, and
- $F \subseteq Q$ is the set of final states.

The state computed on a string $w \in \Sigma^+$ is denoted by $\Delta(w)$ and defined inductively as $\Delta(a) = I(a)$ and $\Delta(awb) = \delta(\Delta(aw), \Delta(wb))$, for all $a, b \in \Sigma$, $w \in \Sigma^*$. Define $L_M(q) = \{w \mid \Delta(w) = q\}$ and $L(M) = \{w \mid \Delta(w) \in F\}$.

Trellis automata have been studied by numerous authors. In particular, important results on their expressive power and complexity were obtained by Ibarra and Kim [9] and by Terrier [32,33]. Trellis automata are known to be equivalent to linear conjunctive grammars [22], as stated in the following theorem.

Theorem A *(Okhotin [22]). A language $L \subseteq \Sigma^+$ is recognized by a trellis automaton if and only if it is described by a linear conjunctive grammar. The transformation is effective in both directions.*

The family of languages defined by these two equivalent models shall be called the *linear conjunctive languages*.

Consider a trellis automaton with the input alphabet $\Sigma_k = \{0, 1, \ldots, k-1\}$ of base-k digits, and assume that it does not accept any strings beginning with 0. Then, every string of digits accepted by the automaton denotes a certain non-negative integer, and thus the automaton defines a set of numbers. The goal is to represent the same set of numbers in unary notation by a conjunctive grammar. For conjunctive grammars of the general form, without the unambiguity condition, this is always possible.

Theorem B *(Jeż, Okhotin [11]). For every $k \geqslant 2$ and for every trellis automaton M over the alphabet Σ_k, with $L(M) \cap 0\Sigma_k^* = \varnothing$, there exists a conjunctive grammar that describes the language $\{a^{(w)_k} \mid w \in L(M)\}$.*

The grammar simulates the computation of a trellis automaton $M = (\Sigma_k, Q, I, \delta, F)$ in the nonterminal symbols C_q, with $q \in Q$, which describe the languages $L(C_q) = \{a^{(1w10^\ell)_k} \mid \Delta(w) = q,\ \ell \geqslant 0\}$, so that each string of digits $w \in \Sigma_k^*$ is represented in unary notation by the strings $a^{(1w1)_k}$, $a^{(1w10)_k}$, $a^{(1w100)_k}$, etc. The definition is inductive on the length of w. As the basis of induction, each C_q should describe all strings of the form $a^{(1j10^\ell)_k}$, with $j \in \Sigma$, $I(j) = q$ and $\ell \geqslant 0$; this is a language similar to the one in Lemma 2.

The grammar implements a step of induction as follows. A string $a^{(1w10^\ell)_k}$ with $|w| \geqslant 2$ should be generated by C_q if and only if $\Delta(w) = q$, which, according to the definition of a trellis automaton, holds if and only if $w = iuj$ for some $i, j \in \Omega$ and $u \in \Omega^*$ with $p = \Delta(iu)$, $r = \Delta(uj)$ and $q = \delta(p, r)$. Then, by the induction hypothesis, the nonterminal C_p generates the string $a^{(1iu10^{\ell+1})_k}$, which is one of the unary encodings of iu, whereas C_r generates $a^{(1uj10^\ell)_k}$, an encoding of uj. The rules of the grammar perform a series of concatenations and intersections on these strings, and ultimately generate $a^{(1iuj10^\ell)_k}$ by C_q.

However, the grammar produced by Theorem B is always ambiguous, and there is no apparent way of expressing the same languages $L(C_q) \subseteq a^*$ unambiguously, for the following reason. The construction of unambiguous grammars, as in Lemma 2, relies on concatenating *exponentially growing* languages, that is, those in which the length of the n-th longest string is exponential in n. The sparsity of such languages in some cases allows their concatenation to be unambiguous. On

the other hand, the languages $L(C_q)$, as defined above, may be denser than that, and then their concatenation with any infinite language is always ambiguous.

Thus, the first step towards simulating a trellis automaton by an unambiguous conjunctive grammar is to define a unary encoding of the languages $L_M(q)$ that always grows exponentially, regardless of the form of $L_M(q)$. This is done by choosing the base k to be larger than the cardinality of the input alphabet Ω of M, and then identifying the symbols in Ω with certain digits, so that Ω becomes a subset of the set of all digits.

Theorem 1. *For every trellis automaton M over an input alphabet Ω containing d symbols, let $c \geqslant \max(5, d+2)$ and assume that $\Omega = \{c, \ldots, c+d-1\}$. Then, for every base $k \geqslant 2c+2d-3$, there exists an unambiguous conjunctive grammar that describes the language $\{a^{(1w1)_k} \mid w \in L(M)\}$.*

If a base $k \geqslant 11$ is fixed, then, for instance, one can use the values $c = \lfloor \frac{k+9}{4} \rfloor$ and $d = \lfloor \frac{k-1}{4} \rfloor$, which induce the alphabet $\Omega = \{\lfloor \frac{k+9}{4} \rfloor, \ldots, \lfloor \frac{k+1}{2} \rfloor\}$. If the goal is to have an alphabet Ω with $d = 2$ symbols, then the smallest values of c and k are $c = 5$ and $k = 11$, so that $\Omega = \{5, 6\}$.

The construction developed in this paper to prove Theorem 1 is generally analogous to the one used in Theorem B; in particular, it adopts a very similar unary representation of strings over Ω. Let $M = (\Omega, Q, I, \delta, F)$ be a trellis automaton. For every state $q \in Q$ and for all $s, t \in \{1, 2\}$, the grammar has a nonterminal $C_q^{s,t}$, which defines the following language.

$$L(C_q^{s,t}) = \{a^{(swt0^\ell)_k} \mid \ell \geqslant 0,\ w \in \Omega^+,\ \Delta(w) = q\}$$

In this construction, the digits s and t surrounding the string w may be `1` or `2`, whereas Theorem B uses only `1` for that purpose; this is an insignificant technical detail. The crucial difference with Theorem B is that each string w processed by M uses only digits from a small subset of Σ_k, and hence each $C_q^{s,t}$ describes an exponentially growing unary language.

The strings $a^{(swt0^\ell)_k}$, with $\Delta(w) = q$, are generated by the corresponding nonterminals $C_q^{s,t}$ inductively on the length of w. The base case is that, for each digit $j \in \Omega$, all strings $a^{(sjt0^\ell)_k}$, with $\ell \geqslant 0$, must be in the language $L(C_q^{s,t})$, where $q = I(j)$. This means describing languages of the form $a^{(sjt0^*)_k}$, which can be represented as the following intersection of two concatenations, similar to those in Example 2.

$$a^{(sjt0^*)_k} = a^{(1t0^*)_k} a^{(s(j-1)0^*)_k} \cap a^{(2t0^*)_k} a^{(s(j-2)0^*)_k}$$

These four languages can be described by the grammar defined in Lemma 1, in which every nonterminal $A_{i,j}$, with $i, j \in \Sigma_k$ and $i \neq 0$, describes the language $L(A_{i,j}) = a^{(ij0^*)_k}$. Using those nonterminals, the desired strings $a^{(sjt0^\ell)_k}$ are obtained in the new grammar by the following rules, defined for all $s, t \in \{1, 2\}$, $j \in \Omega$, and $q = I(j)$.

$$C_q^{s,t} \to A_{1,t} A_{s,j-1} \,\&\, A_{2,t} A_{s,j-2} \tag{9a}$$

Any string $a^{(swt0^\ell)_k}$, with $|w| \geqslant 2$, is generated by the corresponding nonterminal $C_q^{s,t}$ with $q = \Delta(w)$, as follows. Let $w = iuj$, where $i, j \in \Omega$ and $u \in \Omega^*$. In the trellis automaton, $\Delta(iu) = p$, $\Delta(uj) = r$ and $\delta(p, r) = q$. In the grammar, the four strings $a^{(1ujt0^\ell)_k} \in L(C_r^{1,t})$, $a^{(2ujt0^\ell)_k} \in L(C_r^{2,t})$, $a^{(siu10^{\ell+1})_k} \in L(C_1^{s,t'})$ and $a^{(siu20^{\ell+1})_k} \in L(C_2^{s,t'})$ are assumed to be already defined, and the grammar uses them to produce the string $a^{(siujt0^\ell)_k}$ by $C_q^{s,t}$. This is done by the following rules, defined for all $s, t \in \{1, 2\}$, $i, j \in \Omega$, $p, r \in Q$ and $q = \delta(p, r)$.

$$C_q^{s,t} \to C_r^{1,t} A_{s,i-1} \,\&\, C_r^{2,t} A_{s,i-2} \,\&\, C_p^{s,1} A_{j-1,t} \,\&\, C_p^{s,2} A_{j-2,t} \tag{9b}$$

Such a rule represents the desired string as the following four concatenations.

$$\begin{aligned} a^{(siujt0^\ell)_k} &= a^{(1ujt0^\ell)_k} a^{(s(i-1)0^{|ujt|+\ell})_k} = a^{(2ujt0^\ell)_k} a^{(s(i-2)0^{|ujt|+\ell})_k} = \\ &= a^{(siu10^{\ell+1})_k} a^{((j-1)t0^\ell)_k} = a^{(siu20^{\ell+1})_k} a^{((j-2)t0^\ell)_k} \end{aligned}$$

The first two conjuncts of this rule transform unary encodings of the string uj to a unary encoding of iuj. More precisely, one has to transform the two strings $a^{(1ujt0^\ell)_k}$ and $a^{(2ujt0^\ell)_k}$, which are defined by $C_r^{1,t}$ and by $C_r^{2,t}$, respectively, into the string $a^{(siujt0^\ell)_k}$. This is done by concatenating the string $a^{(1ujt0^\ell)_k}$ to $a^{(s(i-1)0^{|ujt|+\ell})_k} \in L(A_{s,i-1})$, and similarly, $a^{(2ujs'0^\ell)_k}$ is concatenated to $a^{(s(i-2)0^{|ujt|+\ell})_k} \in L(A_{s,i-2})$. At the first glance, these two conjuncts do the same thing, but this double representation is necessary in order to filter out the garbage in the concatenations (similarly to what is done in Example 1, and in other grammars as well). It shall be proved below that the intersection of the first two conjuncts in a rule (9b) defines the set of all strings $a^{(siujt0^\ell)_k}$ with $\Delta(uj) = r$ and with arbitrary $i \in \Omega$.

The last two conjuncts of the rule similarly transform any two strings $a^{(siu10^{\ell+1})_k}$ and $a^{(siu20^{\ell+1})_k}$ into the string $a^{(siujt0^\ell)_k}$. It shall be proved that, for each $j \in \Omega$, the conjunction $C_p^{s,1} A_{j-1,t} \,\&\, C_p^{s,2} A_{j-2,t}$ defines the language of all $a^{(siujt0^\ell)_k}$ with $\Delta(iu) = p$. Once these four conjuncts are intersected in a single rule, it accordingly generates all $a^{(siujt0^\ell)_k}$ with $\Delta(iuj) = q$, as desired.

Finally, the desired language $\{a^{(1w1)_k} \mid w \in L(M)\}$ is obtained from these variables $C_q^{s,t}$ as follows. First, one should use only the variables $C_q^{s,t}$ with $s = t = 1$ and with accepting states $q \in F$. Together, these variables define all strings $(1w10^\ell)_k$ with $w \in L(M)$ and $\ell \geqslant 0$. It remains to ensure that there are no trailing zeroes in the base-k notation of the number, which is done by intersecting this union with the set of strings of length 1 modulo k.

The rules (9b) defined above are formed of two groups of two conjuncts each, which append the digits to the left and to the right of strings generated by other nonterminals. Explicitly naming these groups and determining their properties makes the proof easier. The rules (9a) can also be presented in this way. For that reason, the grammar defined above is equivalently restated as follows. First, denote the pairs of conjuncts occurring in the rules as the following expressions, $\lambda_i^s(X, Y)$ for appending a digit i to the left of the base-k notation of all numbers in X and Y, and similarly, $\rho_j^t(X, Y)$ for appending j to the right.

$$\lambda_i^s(X, Y) = XA_{s,i-1} \,\&\, YA_{s,i-2} \qquad \text{for } s \in \{1, 2\},\ i \in \Omega \tag{10a}$$

$$\rho_j^t(X, Y) = XA_{j-1,t} \,\&\, YA_{j-2,t} \qquad \text{for } t \in \{1, 2\},\ j \in \Omega \tag{10b}$$

Here $s, t \in \{1, 2\}$ are the digits delimiting the encoding in the resulting strings, and X and Y are assumed to encode the same language using different delimiting digits. Then, all rules of the above grammar can be formulated in terms of these expressions.

$$C_q^{s,t} \to \lambda_j^s(A_{1,t}, A_{2,t}) \qquad \text{for } j \in \Omega \text{ with } I(j) = q \tag{10c}$$

$$C_{\delta(p,r)}^{s,t} \to \lambda_i^s(C_r^{1,t}, C_r^{2,t}) \,\&\, \rho_j^t(C_p^{s,1}, C_p^{s,2}) \qquad \text{for } i, j \in \Omega \text{ and } p, r \in Q \tag{10d}$$

The rules for the initial symbol S use the variables $C_q^{s,t}$ with $s = t = 1$, and intersect them with an extra symbol O that generates all strings of length 1 modulo k.

$$S \to C_q^{1,1} \,\&\, O \qquad \text{for } q \in F \tag{10e}$$

$$O \to a \mid a^k O \tag{10f}$$

The first step of the proof is determining the effect of the expressions λ_i^s and ρ_j^t on any languages of the form described in the grammar. Consider the expression $\lambda_i^s(X, Y)$: for some fixed $t \in \{1, 2\}$, its first argument language X is assumed to contain strings of the form $a^{(1wt0^\ell)_k}$, for some strings $w \in \Omega^*$ and for all $\ell \geqslant 0$, whereas the second argument Y should contain strings of the form $a^{(2wt0^\ell)_k}$, which differ from those in X only in their first digit. Then, according to the lemma below, the expression λ_i^s appends the digit i to the left side of w in each of these strings, obtaining strings of the form $a^{(siwt0^\ell)_k}$.

Lemma 3. *Let $K \subseteq \Omega^+$ or $K = \{\varepsilon\}$, and let $t \in \{1, 2\}$. Define $L_1 = a^{(1Kt0^*)_k}$ and $L_2 = a^{(2Kt0^*)_k}$. Then, for each $i \in \Omega$ and $s \in \{1, 2\}$,*

$$\lambda_i^s(L_1, L_2) = a^{(siKt0^*)_k}. \tag{11}$$

Proof. Consider the case of $K \subseteq \Omega^+$. The goal is to establish the following equality.

$$a^{(1Kt0^*)_k} \cdot a^{(s(i-1)0^*)_k} \cap a^{(2Kt0^*)_k} \cdot a^{(s(i-2)0^*)_k} = a^{(siKt0^*)_k}$$

It is easy to show that every string in $a^{(siKt0^*)_k}$, which is of the form $a^{(siwt0^\ell)_k}$, with $w \in K$ and $\ell \geqslant 0$, must belong to the left-hand side expression. Indeed, its length can be represented in two ways as follows, where the given four numbers belong to the required sets $(1Kt0^*)_k$, $(s(i-1)0^*)_k$, $(2Kt0^*)_k$ and $(s(i-2)0^*)_k$.

$$(siwt0^\ell)_k = (1wt0^\ell)_k + (s(i-1)0^{\ell+|wt|})_k = (2wt0^\ell)_k + (s(i-2)0^{\ell+|wt|})_k \tag{12}$$

The remaining non-trivial part of the argument is that every number representable as two sums $x_1 + y_1 = x_2 + y_2$, with the form of the four numbers defined below, must therefore be in $(siKt0^*)_k$.

$$x_1 = (1w_1t0^{\ell_1})_k \in (1Kt0^*)_k$$

$$y_1 = (s(i-1)0^{m_1})_k \in (s(i-1)0^*)_k$$

$$x_2 = (2w_2t0^{\ell_2})_k \in (2Kt0^*)_k$$

$$y_2 = (s(i-2)0^{m_2})_k \in (s(i-2)0^*)_k$$

The possible cases of base-k addition of x_1 and y_1 depending on the relation between ℓ_1 and m_1 are illustrated in Fig. 2. There are seven main cases (a–g), where no carry occurs in any positions. Furthermore, under some conditions, there are two cases (d^1 and e^1) with a carry: this may only occur if $k = 2c + 2d - 3$ and $\omega' = i = c + d - 1$. When adding x_2 to y_2, the seven main cases are the same, with the only difference in the use of digit 2 instead of 1, as well as $i - 2$ instead of

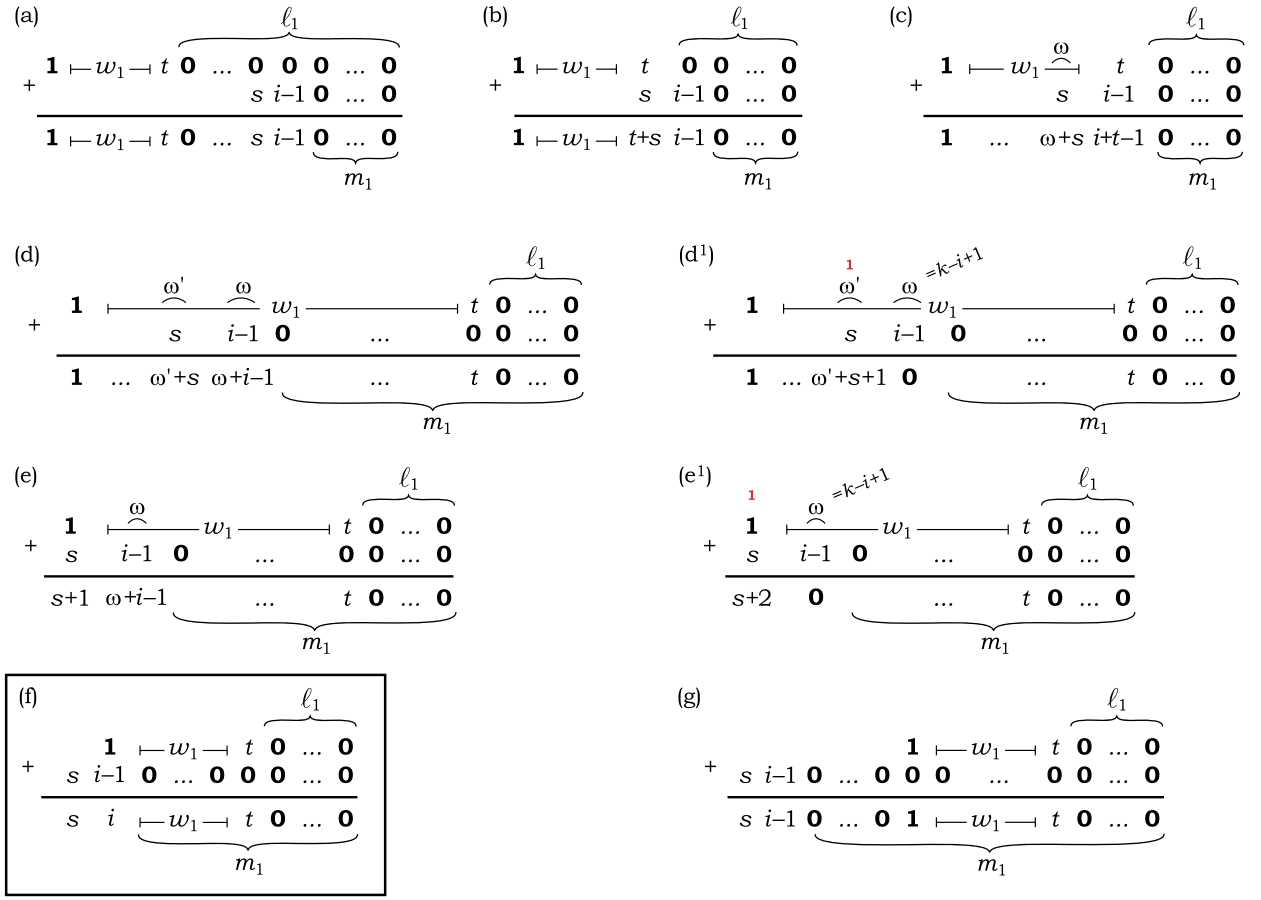

Fig. 2. Possible cases in the addition of $x_1 = (1w_1t0^{\ell_1})_k$ to $y_1 = (s(i-1)0^{m_1})_k$: (a) $m_1 < \ell_1 - 1$; (b) $m_1 = \ell_1 - 1$; (c) $m_1 = \ell_1$; (d) $\ell_1 < m_1 \leqslant |w_1t| + \ell_1 - 2$; (d¹) the same, with a carry, only if $\omega' = i = c + d - 1$ and $k = 2c + 2d - 3$; (e) $m_1 = |w_1t| + \ell_1 - 1$; (e¹) the same, with a carry; (f) $m_1 = |w_1t| + \ell_1$; (g) $m_1 > |w_1t| + \ell_1$. The same cases apply to $x_2 + y_2$, but carry is not possible (no cases d^1 and e^1).

$i-1$. The carry can no longer occur here—that is, cases (d^1) and (e^1) are impossible—because the sum $\omega' + (i-2)$ is at most $2c + 2d - 4$, which is smaller than k.

The point is to show that for the two sums $x_1 + y_1$ and $x_2 + y_2$ to be equal, each of them must be of the intended form, shown in Fig. 2(f).

First, consider the last (that is, least significant) non-zero digit in the base-k notation of the number $x_1 + y_1$. The last non-zero digit of the base-k notation of x_1 is t and the last non-zero digit of y_1 is $i-1$. If one of x_1, y_1 has fewer terminating zeroes than the other, then $x_1 + y_1$ inherits the last non-zero digit from that number: that is, if $\ell_1 > m_1$, as in Fig. 2(a,b), then the last non-zero digit in $x_1 + y_1$ comes from y_1 and equals $i-1$, and if $\ell_1 < m_1$, then it comes from x_1 and is t, see Fig. 2(d–g). If the number of terminating zeroes in x_1 and y_1 is the same, that is, if $\ell_1 = m_1$, then this digit is $t + i - 1$, obtained as the sum of the last non-zero digits of x_1 and y_1, which is illustrated in Fig. 2(c).

A similar analysis can be made for $x_2 + y_2$. The last non-zero digit in the base-k notation of x_2 is t, whereas for y_2 it is $i-2$. Therefore, if $\ell_2 > m_2$, then the last non-zero digit of $x_2 + y_2$ is $i-2$; if $\ell_2 < m_2$, then it is t; and if $\ell_2 = m_2$, then the digit is $t + i - 2$.

Since $x_1 + y_1 = x_2 + y_2$, both sums have the same base-k representation with the same last non-zero digit. There are three possibilities for the last non-zero digit of $x_1 + y_1$ and three possibilities for $x_2 + y_2$, which give rise to the following nine cases.

	$\ell_2 > m_2$ digit $i-2$	$\ell_2 = m_2$ digit $t+i-2$	$\ell_2 < m_2$ digit t
$\ell_1 > m_1$ digit $i-1$	$i-1 > i-2$	potentially equal	$i-1 > t$
$\ell_1 = m_1$ digit $t+i-1$	$t+i-1 > i-2$	$t+i-1 > t+i-2$	$t+i-1 > t$
$\ell_1 < m_1$ digit t	$t < i-2$	$t < t+i-2$	equal

As demonstrated in the table, in seven of these cases, these digits cannot be the same, which leaves only two possibilities: either $\ell_1 > m_1$ and $\ell_2 = m_2$, or $\ell_1 < m_1$ and $\ell_2 < m_2$.

The former case is ruled out by considering the digit to the left of the last non-zero digit in $x_1 + y_1$ and in $x_2 + y_2$, as follows. In $x_2 + y_2$, as $\ell_2 = m_2$, the numbers x_2 and y_2 have the same number of terminating zeroes, and the digit to the left of the last non-zero digit comes from adding a digit s from y_2 to a digit $\omega \in \Omega$ from x_2, as illustrated in Fig. 2(c). In the other sum $x_1 + y_1$, there are two subcases, depicted in Fig. 2(a,b): if $\ell_1 = m_1 + 1$, then the digit to the left of the last non-zero digit $(i-1)$ is $t+s$, which is less than $\omega + s$; and if $\ell_1 > m_1 + 1$, then the digit in question is s, which is again less than $\omega + s$.

The contradiction obtained proves that if $x_1 + y_1 = x_2 + y_2$, then $\ell_1 = \ell_2$, $\ell_1 < m_1$ and $\ell_2 < m_2$, and the last non-zero digit of $x_1 + y_1$ is t. In other words, the first three cases of addition shown in Fig. 2(a–c) are impossible both in the sum $x_1 + y_1$ and in the sum $x_2 + y_2$. The next goal is to show that $m_1 = \ell_1 + |w_1 t|$ and $m_2 = \ell_2 + |w_2 t|$, as in Fig. 2(f). This, together with the already established equality $\ell_1 = \ell_2$, would imply that Eq. (12) is the only possible form of the double representation $x_1 + y_1 = x_2 + y_2$.

With this in mind, the proof proceeds by characterizing the base-k representation of $x_1 + y_1$ depending on the relation between m_1 and $\ell_1 + |w_1 t|$ (in Claim 1 below), and by similarly characterizing the base-k notation of $x_2 + y_2$ (Claim 2). The rest of the proof will be a case analysis based of these two characterizations of the same number $x_1 + y_1 = x_2 + y_2$.

First, there are the following three possible cases of the sum $x_1 + y_1$.

Claim 1. *Let $x_1 = (1 w_1 t 0^{\ell_1})_k$ and $y_1 = (s(i-1)0^{m_1})_k$, with $\ell_1 < m_1$. Then, depending on m_1, the base-k notation of the number $x_1 + y_1$ is of one of the three mutually exclusive forms.*

i. *If $\ell_1 < m_1 < |w_1 t| + \ell_1$, then $x_1 + y_1$ has a digit at least $c + d + 1$ at position $m_1 + 1$ (as illustrated in Fig. 2(d,e)), or digit 0 at the same position (Fig. 2(d^1,e^1)).*
ii. *If $m_1 = |w_1 t| + \ell_1$, as in Fig. 2(f), then $x_1 + y_1 \in (siKt0^*)_k$.*
iii. *If $m_1 > |w_1 t| + \ell_1$, as in Fig. 2(g), then $x_1 + y_1 \in (s(i-1)0^*1\Omega^* t 0^*)_k$.*

Furthermore, in case (i) the digit at position $m_1 + 1$ is either at least $c + d$, and there is no carry, or this digit is 0 (this possibility is illustrated in Fig. 2(d^1)), and there is a unique carry in $x_1 + y_1$, from position $m_1 + 1$ to $m_1 + 2$. Either way, the digit at position $m_1 + 1$ is followed by a string of digits from $\Omega^* t 0^*$.

Proof. The second summand y_1 has the digit $i-1$ in the position $m_1 + 1$. The value of the corresponding digit in $x_1 = (1 w_1 t 0^{\ell_1})_k$ depends on the relation between m_1 and ℓ_1. Since $m_1 > \ell_1$, there are three possibilities, corresponding to the three cases in the statement of the claim.

- If $\ell_1 < m_1 < |w_1 t| + \ell_1$, as in Fig. 2(d,e,$d^1$,$e^1$), then it is one of the digits from Ω, say ω. In this case, the sum of digits at position $m_1 + 1$ in $x_1 + y_1$ is $\omega + i - 1$. Since both ω and i are in Ω, the sum $\omega + i - 1$ is at least $2c - 1$ and at most $2c + 2d - 3 \leqslant k$, in particular it can cause a carry.
  - If there is no carry, then, since $c \geqslant d + 2$, this digit is at least $c + d + 1$, as illustrated in Fig. 2(d,e).
  - If there is a carry, then the digit at position $m_1 + 1$ is 0, shown in Fig. 2(d^1,e^1).

  There can be no other carries, because y_1 has only two non-zero digits (s at position $m_1 + 2$ and $i - 1$ at position $m_1 + 1$), and since $s \in \{1, 2\}$, it cannot cause a carry.

  It is left to show that the string of digits in positions m_1, $m_1 - 1$, …, 1 in $x_1 + y_1$ belongs to $\Omega^* t 0^*$. Observe that y_1 has only zeroes in those positions, and hence the digits in the sum are the same as the digits in the corresponding positions in x_1. Since $x_1 \in (1\Omega^* t 0^*)_k$ and the digit at position $m_1 + 1$ in x_1 is from Ω, it follows that the string of digits in positions 1, 2, …, m_1 is from $(\Omega^* t 0^*)_k$.
- If $m_1 = |w_1 t| + \ell_1$, that is, in the case depicted in Fig. 2(f), then the digit in position $m_1 + 1$ in x_1 is the leading digit 1. In this case, $x_1 + y_1 = (1 w_1 t 0^{\ell_1})_k + (s(i-1)0^{|w_1 t| + \ell_1})_k = (siw_1 t 0^{\ell_1})_k$.
- If $m_1 > |1 w_1| + \ell_1$, see Fig. 2(g), then there is no digit at position $m_1 + 1$ in base-k positional notation of x_1. Then $x_1 + y_1 = (1 w_1 t 0^{\ell_1})_k + (s(i-1)0^{m_1})_k = (s(i-1)0^{m_1 - |1 w_1 t| - \ell_1} w_1 t 0^{\ell_1})_k$.

This ends the case inspection. To conclude the proof, it is left to show that the three cases (i–iii) of the base-k notation of $x_1 + y_1$ are indeed mutually disjoint. Observe that if a number belongs to the set from case (i) and has a digit at least $c + d$, then the numbers described in cases (ii–iii) do not contain any digits that large. If the number is of the form described in case (i) and has a digit 0 that is followed by a string of digits from $\Omega^* t 0^*$, then the numbers described in cases (ii–iii) do not have a substring of digits from $0\Omega^* t 0^*$. If a number belongs to a set $(siKt0^*)_k$ in case (ii), then its second leading digit is i, and if it belongs to a set $(s(i-1)0^*1\Omega^* t 0^*)_k$ in case (iii), then such a digit is $i - 1$. Thus, the sets described in cases (ii) and (iii) are disjoint as well. □

A similar, slightly simpler characterization holds for the sum $x_2 + y_2$.

Claim 2. *Let $x_2 = (2 w_2 t 0^{\ell_2})_k$ and $y_2 = (s(i-2)0^{m_2})_k$ with $\ell_2 < m_2$. Then, depending on m_2, the base-k notation of the number $x_2 + y_2$ is of one of the three mutually exclusive forms.*

i. *If* $\ell_2 < m_2 < |w_2 t| + \ell_2$ *(as in Fig. 2(d,e), with digit* 1 *in the first number replaced with* 2*), then* $x_2 + y_2$ *has a digit at least* $c + d$*, which is at position* $m_2 + 1$*. This digit is followed by a string of digits from* $\Omega^* t 0^*$*. There is no carry in the addition* $x_2 + y_2$*.*
ii. *If* $m_2 = |w_2 t| + \ell_2$ *(see Fig. 2(f)), then* $x_2 + y_2 \in (siKt0^*)_k$*.*
iii. *If* $m_2 > |w_2 t| + \ell_2$ *(see Fig. 2(g)), then* $x_2 + y_2 \in (s(i-2)0^*2\Omega^* t 0^*)_k$*.*

Proof. The only difference between the proof of this claim and Claim 1 is in the first case, when $\ell_2 < m_2 < |w_2 t| + \ell_2$: then the sum of the digits $i - 2$ and ω is at least $2c - 2$ and at most $2c + 2d - 4 < k$, and so it cannot cause a carry. So only the subcase with a digit with value at least $c + d$ needs to be considered. The rest of the proof is the same. □

Resuming the proof of Lemma 3, Claims 1 and 2 identify three cases for the sum $x_1 + y_1$, and three cases for $x_2 + y_2$. This gives in total nine cases to consider. The relation between ℓ_1, m_1, ℓ_2 and m_2 is determined in the following case analysis.

First, consider the case of $m_1 = |w_1 t| + \ell_1$, as in Fig. 2(f). Then, by Claim 1 for the number $x_1 + y_1$, this number is in $(siKt0^*)_k$, as claimed. Similarly, if $m_2 = |w_2 t| + \ell_2$, as depicted in Fig. 2(f), then, by Claim 2, the sum $x_2 + y_2$ is in $(siKt0^*)_k$, again as claimed.

The remaining cases are arranged according to the relation between ℓ_2 and m_2.

- Suppose that $\ell_2 < m_2 < |w_2 t| + \ell_2$, see Fig. 2(d,e). Then, by Claim 2, there is a digit at least $c + d$ in base-k notation of $x_2 + y_2$ (at position $m_2 + 1$), which is followed by a string of digits from $\Omega^* t 0^*$. This digit and the following string of digits also must occur in the base-k notation of $x_1 + y_1$, and Claim 1 asserts that such a large digit can be there only when $\ell_1 < m_1 < |w_1 t| + \ell_1$ (so again this is a situation depicted in Fig. 2(d,e)), and it is at position $m_1 + 1$. As this digit needs to be at the same position in both $x_2 + y_2$ and $x_1 + y_1$, it can be concluded that $m_1 = m_2$, and so a common name m is used for both of them. Furthermore, Claim 1(i) implies that, since there is such a large digit in the sum, there is no carry in the base-k addition $x_1 + y_1$. Lastly, as $m < |w_2 t| + \ell_2$, the length of the positional notation of $x_2 + y_2$ is $|w_2 t| + \ell_2 + 1$, and similarly the length of positional notation of $x_1 + y_1$ is $|w_1 t| + \ell_1 + 1$. Those lengths are equal, because they refer to the same number, and so $|w_1 t| + \ell_1 = |w_2 t| + \ell_2$.
  Further analysis investigates the leading digits of base-k representations of $x_2 + y_2$ and $x_1 + y_1$. By Claim 2(i), there is no carry in $x_2 + y_2$, and it is known that there is no carry in the base-k addition $x_1 + y_1$. Hence, there are the following two subcases.
  - If $m < |w_2 t| + \ell_2 - 1$, as shown in Fig. 2(d), then the leading digit of $x_2 + y_2$ comes from x_2, and it is 2. Since $|w_2 t| + \ell_2 - 1 = |w_1 t| + \ell_1 - 1$, similarly the leading digit of $x_1 + y_1$ is 1, contradiction.
  - If $m = |w_2 t| + \ell_2 - 1$ (see Fig. 2(e)), then it is $2 + s$, obtained by summing the leading digits of x_2 and y_2, which are 2 and s, respectively. As $|w_2 t| + \ell_2 - 1 = |w_1 t| + \ell_1 - 1$, in the same way the leading digit of $x_1 + y_1$ is $1 + s$, contradiction.

  The contradictions obtained rule out this case.
- Suppose that $m_2 > |w_2 t| + \ell_2$, see Fig. 2(g), and consider the relation between ℓ_1 and m_1 in $x_1 + y_1$.
  - Suppose that $m_1 > |w_1 t| + \ell_1$, which is also shown in Fig. 2(g). Then, by Claim 1, the second leading digit in $x_1 + y_1$ is $i - 1$, whereas Claim 2 implies that the second leading digit in $x_2 + y_2$ is $i - 2$, contradiction.
  - So the only remaining case is $m_1 < |w_1 t| + \ell_1$, as shown in Fig. 2(d,e). In such a case, by Claim 1, the base-k notation of $x_1 + y_1$ either has a digit of value at least $c + d + 1$, or it has a suffix from the set $0\Omega^* t 0^*$. However, by Claim 2, neither of these can happen when $m_2 > |w_2 t| + \ell_2$.

This ends the case inspection, to the conclusion that $x_1 + y_1 = x_2 + y_2$ implies $x_1 + y_1 \in (siKt0^*)_k$, for any $K \subseteq \Omega^+$.

It is left to consider the case of $K = \{\varepsilon\}$. Then the claim of the lemma—Eq. (11)—reduces to

$$a^{(1t0^*)_k} a^{(s(i-1)0^*)_k} \cap a^{(2t0^*)_k} a^{(s(i-2)0^*)_k} = a^{(sit0^*)_k},$$

which can be shown as in Lemma 2. This argument is omitted due to no novelty in the method. □

The other crucial property of the expression λ_i^s is that, under the substitution of languages of the intended form, the concatenations therein are unambiguous.

Lemma 4. *Let* $K = \{\varepsilon\}$ *or* $K \subseteq \Omega^+$ *and let* $t \in \{1, 2\}$*. Define* $L_1 = a^{(1Kt0^*)_k}$ *and* $L_2 = a^{(2Kt0^*)_k}$*. Then, for each* $s \in \{1, 2\}$ *and* $i \in \Omega$*, both concatenations in* $\lambda_i^s(L_1, L_2)$ *are unambiguous.*

Proof. Consider the case of $K \subseteq \Omega^+$. The expression $\lambda_i^s(L_1, L_2)$ is a conjunction of two concatenations $a^{(rKt0^*)_k} a^{(s(i-r)0^*)_k}$, with $r \in \{1, 2\}$, and the goal is to prove that all concatenations of this form, for any $r, s, t \in \{1, 2\}$, are unambiguous. Consider any four numbers of the following form, and assume that $x_1 + y_1 = x_2 + y_2$.

$$x_1 = (rw_1 t 0^{\ell_1})_k \in (rKt0^*)_k$$
$$y_1 = (s(i-r)0^{m_1})_k \in (s(i-r)0^*)_k$$

$$x_2 = (rw_2t0^{\ell_2})_k \in (rKt0^*)_k$$
$$y_2 = (s(i-r)0^{m_2})_k \in (s(i-r)0^*)_k$$

If $(x_1, y_1) \neq (x_2, y_2)$, these partitions would witness the ambiguity of this concatenation. To see that the partition is unique and the concatenation is thus unambiguous, it is enough to show that $m_1 = m_2$, as this will imply $y_1 = y_2$ and consequently also $x_1 = x_2$.

The value and position of the last non-zero digit in the base-k notation of $x_1 + y_1$ are established by the following case analysis, which is illustrated in Fig. 2, with 1 replaced by r, and $i-1$ by $i-r$. If $m_1 < \ell_1$, as shown in Fig. 2(a,b), then the last non-zero digit of $x_1 + y_1$ is the last non-zero digit of y_1, which is $i-r$ at the position m_1+1 from the right. If $m_1 > \ell_1$, as in Fig. 2(d–g), then this digit is t, the last non-zero digit of x_1 at the position ℓ_1+1 from the right. Lastly, if $m_1 = \ell_1$, illustrated in Fig. 2(c), then the last non-zero digit of $x_1 + y_1$ is the sum of last non-zero digits of x_1 and y_1, that is, $i-r+t$ at the position m_1+1.

The same analysis applies to $x_2 + y_2$, so that if $m_2 < \ell_2$, then its last non-zero digit is $i-r$ at position m_2+1, and if $m_2 > \ell_2$, then it is t at position ℓ_2+1, and if $m_2 = \ell_2$, the digit is $i-r+t$ at position m_2+1. Since $x_1 + y_1 = x_2 + y_2$, the last non-zero digits in $x_1 + y_1$ and in $x_2 + y_2$ are the same and are at the same position, which can take place for the following combinations of these cases.

		$\ell_2 > m_2$ $i-r$	$\ell_2 = m_2$ $t+i-r$	$\ell_2 < m_2$ t
$\ell_1 > m_1$	digit $i-r$	equal	$i-r < t+i-r$	$i-r > t$
$\ell_1 = m_1$	digit $t+i-r$	$t+i-r > i-r$	equal	$t+i-r > t$
$\ell_1 < m_1$	digit t	$t < i-r$	$t < t+i-r$	equal

This leaves several options for the last non-zero digit of $x_1 + y_1$ and $x_2 + y_2$.

- It can be $i-r$, which is at position m_1+1 in $x_1 + y_1$ and at position m_2+1 in $x_2 + y_2$. This shows that $m_1 = m_2$ and so ends the proof.
- For $t+i-r$ at position m_1+1, as in the case above, this implies $m_1+1 = m_2+1$ and thus completes the proof.
- If it is t at position ℓ_1+1 in $x_1 + y_1$ and at position ℓ_2+1 in $x_2 + y_2$, this implies that $\ell_1 < m_1$, $\ell_2 < m_2$ and $\ell_1 = \ell_2$. This case is considered in the rest of the proof.

Assume that $\ell_1 < m_1$, $\ell_2 < m_2$ and $\ell_1 = \ell_2$, and accordingly denote $\ell := \ell_1 = \ell_2$; the cases $\ell < m_1$ and $\ell < m_2$ are shown in Fig. 2(d–g). If $r = 1$, then the rest of the argument is based on Claim 1, which is applied to both $x_1 + y_1$ and $x_2 + y_2$. For $r = 2$, Claim 2 is similarly used for both numbers. Besides using two different claims, the arguments for $r = 1$ and for $r = 2$ are identical, and hence only the case of $r = 1$ is considered in the following.

First, consider the case of m_1 lying within the bounds $\ell < m_1 < |w_1t| + \ell$, which is depicted in Fig. 2(d,e,d^1,e^1). Then the base-k notation of the number $x_1 + y_1$ is described in Claim 1(i). Consider $x_2 + y_2$ and its form, which is the same as the form of $x_1 + y_1$. As the forms of numbers in different cases of Claim 1 are disjoint, it can be concluded that $x_2 + y_2$ must be of the form described in Claim 1(i). In particular, $\ell < m_2 < |w_2t| + \ell$.

Consider the digit at position m_1+1 in $x_1 + y_1$. It is either zero or at least $c+d+1$, and to the right of its position there is a string of digits from $\Omega^* t 0^*$. As none of the digits $c+d+1, c+d+2, \ldots, k-1, 0$ is in Ω, this is also the longest suffix of the base-k notation of $x_1 + y_1$ from the set $\Omega^* t 0^*$, and it is of the length m_1. The same analysis of $x_2 + y_2$ yields that the analogous longest suffix of $x_2 + y_2$ has length m_2. Hence, $m_1 = m_2$, which ends the proof for this case.

In the symmetric case, when $\ell < m_2 < |w_2t| + \ell$, the same argument as in the previous paragraph implies that $\ell < m_1 < |w_1t| + \ell$ and $m_1 = m_2$. Therefore, the only remaining case is $m_1 \geqslant |w_1t| + \ell$ and $m_2 \geqslant |w_2t| + \ell$, which is shown in Fig. 2(f,g). In such a case, the number of digits in the base-k notation of $x_1 + y_1$ is $m_1 + 2$, whereas the length of the base-k notation of $x_2 + y_2$ is $m_2 + 2$. Thus, $m_1 = m_2$ also in this case, which completes the proof of the lemma for $K \subseteq \Omega^+$.

The case of $K = \{\varepsilon\}$ reduces to showing that the concatenation $a^{(1t0^*)_k} a^{(s(i-1)0^*)_k}$ is unambiguous. This follows from Lemma 1, since all digits 1, t, s and $i-1$ are non-zero. □

The above Lemmata 3 and 4 state all the necessary properties of the expression λ_s^i. Now the task is to establish a similar characterization of the other expression ρ_j^t.

Lemma 5. *Let $K \subseteq \Omega^+$ and let $s \in \{1, 2\}$. Define $L_1 = a^{(sK10^*)_k}$ and $L_2 = a^{(sK20^*)_k}$. Then, for every $j \in \Omega$ and $t \in \{1, 2\}$,*

$$\rho_j^t(L_1, L_2) = a^{(sKjt0^*)_k}. \tag{13}$$

Sketch of a proof. The lemma asserts the following equality.

$$a^{(sK10^*)_k} \cdot a^{((j-1)t0^*)_k} \cap a^{(sK20^*)_k} \cdot a^{((j-2)t0^*)_k} = a^{(sKjt0^*)_k}$$

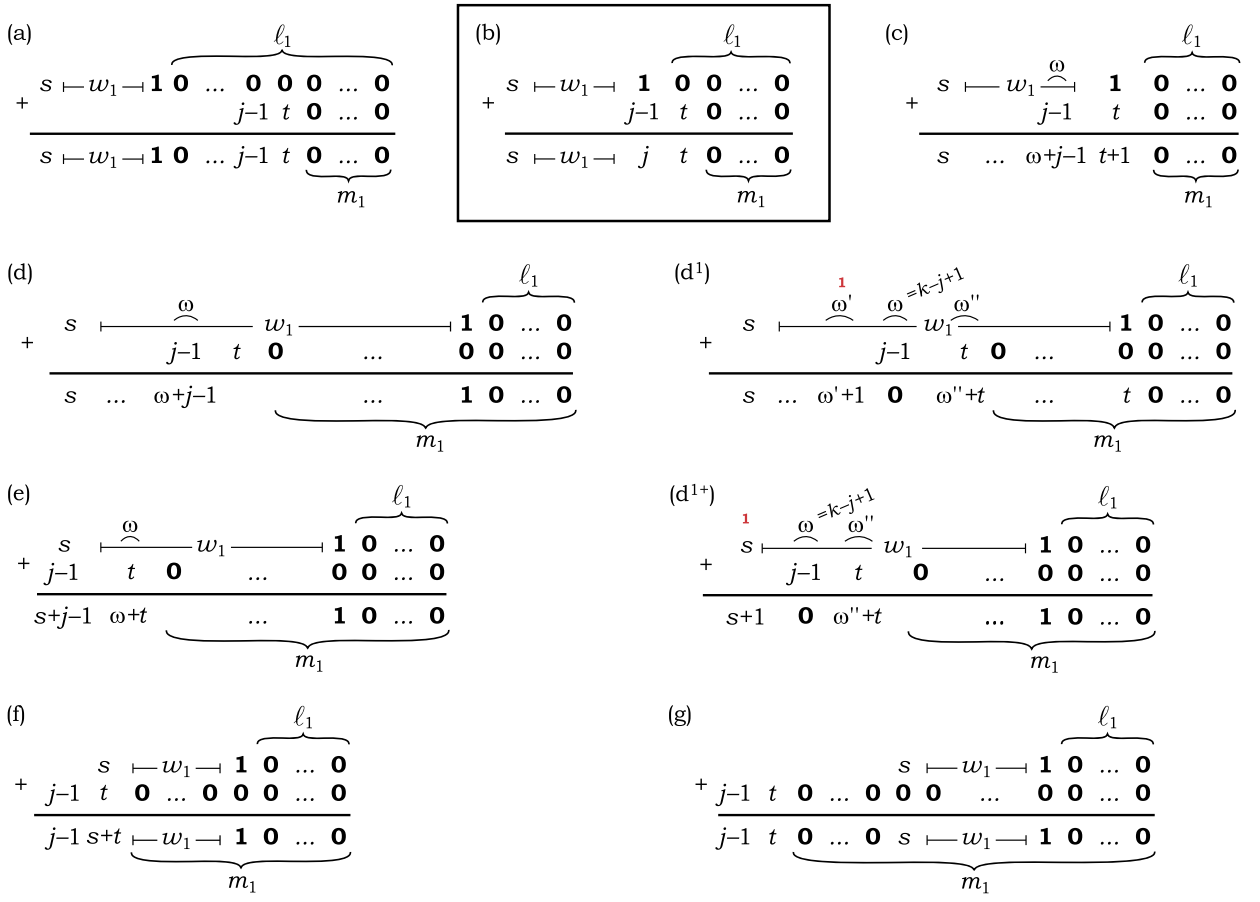

Fig. 3. Possible cases in the addition of $x_1 = (sw_1 1 0^{\ell_1})_k$ to $y_1 = ((j-1)t0^{m_1})_k$: (a) $m_1 < \ell_1 - 1$; (b) $m_1 = \ell_1 - 1$; (c) $m_1 = \ell_1$; (d) $\ell_1 < m_1 \leqslant |w_1 1| + \ell_1 - 2$; (d^1) $\ell_1 < m_1 \leqslant |w_1 1| + \ell_1 - 3$, with a carry; (d^{1+}) $m_1 = |w_1 1| + \ell_1 - 2$, with a carry; (e) $m_1 = |w_1 1| + \ell_1 - 1$; (e^1) the same, with a carry; (f) $m_1 = |w_1 1| + \ell_1$; (g) $m_1 > |w_1 1| + \ell_1$. The same cases apply to $x_2 + y_2$, although carry is not possible (no cases d^1 and e^1).

The proof proceeds similarly to Lemma 3. Every string from $a^{(sKjt0^*)_k}$ belongs to the intersection on left-hand side, because its length can be represented in the following two ways.

$$(swjt0^{\ell})_k = (sw10^{\ell})_k + ((j-1)t0^{\ell-1})_k = (sw20^{\ell})_k + ((j-2)t0^{\ell-1})_k \tag{14}$$

The task is to show that if a number is represented as $x_1 + y_1$ and as $x_2 + y_2$, with x_1, y_1, x_2, y_2 as below, then it is indeed in $(sKjt0^*)_k$.

$$x_1 = (sw_1 10^{\ell_1})_k \in (sK10^*)_k$$
$$y_1 = ((j-1)t0^{m_1})_k \in ((j-1)t0^*)_k$$
$$x_2 = (sw_2 20^{\ell_2})_k \in (sK20^*)_k$$
$$y_2 = ((j-2)t0^{m_2})_k \in ((j-2)t0^*)_k$$

The case analysis is mostly symmetrical to the one in Lemma 3. Observe that the order of non-zero digits in x_1 above is reversed, as compared to the number x_1 in Lemma 3. The same applies to x_2, y_1 and y_2. What is not symmetric, is the carry (if present), as the extra 1 is always carried over to the right.

Possible cases of addition of x_1 to y_1 are depicted in Fig. 3. Again, the analysis is similar to the one in Lemma 3, though this time it is the leading digit that is investigated, instead of the last non-zero digit (due to the symmetry). The possible cases are listed in the table below.

	$m_2 < \ell_2 + \lvert w_2 \rvert$ digit s	$m_2 = \ell_2 + \lvert w_2 \rvert$ digit $s+j-2$	$m_2 > \ell_2 + \lvert w_2 \rvert$ digit $j-2$
$m_1 < \ell_1 + \lvert w_1 \rvert$ digit s or $s+1$	may be equal	$s+1 < s+j-2$	$s+1 < j-2$
$m_1 = \ell_1 + \lvert w_1 \rvert$ digit $s+j-1$	$s+j-1 > s$	$s+j-1 > s+j-2$	$s+j-1 > j-2$
$m_1 > \ell_1 + \lvert w_1 \rvert$ digit $j-1$	$j-1 > s$	potentially equal	$j-1 > j-2$

The "potentially equal" case is excluded similarly to how it was done in Lemma 3, but this time using the second leading digit. Thus, the rest of the proof deals with the case when $m_1 < \ell_1 + |w_1|$ and $m_2 < \ell_2 + |w_2|$. The following Claim describes the possible forms of numbers $x_1 + y_1$, depending on the relation between m_1 and ℓ_1.

Claim 3. *Let $x_1 = (sw_1 1 0^{\ell_1})_k$ and $y_1 = ((j-1)t0^{m_1})_k$, with $m_1 < \ell_1 + |w_1|$. Then, depending on m_1, the base-k notation of the number $x_1 + y_1$ is of one of the following three mutually exclusive forms.*

i. *If $m_1 > \ell_1 - 1$, then $x_1 + y_1$ has a digit at least $c+d+1$ at position $m_1 + 2$ (as illustrated in Fig. 3(c,d)), or, under some conditions, digit 0 at this position (see Fig. 3(d^1,d^{1+})).*
ii. *If $m_1 = \ell_1 - 1$, as in Fig. 3(b), then $x_1 + y_1 \in (sKjt0^*)_k$.*
iii. *If $m_1 < \ell_1 - 1$, as in Fig. 3(a), then $x_1 + y_1 \in (s\Omega^* 1 0^* (j-1)t0^*)_k$.*

Furthermore, if in case (i) the digit at position $m_1 + 2$ is at least $c + d + 1$, then there is no carry. If this digit is 0, then there is a unique carry in $x_1 + y_1$: from position $m_1 + 2$ to $m_1 + 3$. Moreover, either the two leading digits of $x_1 + y_1$ are $s + 1$ and 0, when $m_1 + 1 = |w_1| + \ell_1$, or $x_1 + y_1$ has a zero preceded by a digit at least $c + 1$, in which case the leading digit is s.

The proof follows by a case inspection symmetrical to the proof of Claim 1.
A similar statement can be shown also for x_2 and y_2, except that there can be no carry in addition of $x_2 + y_2$.

Claim 4. *Let $x_2 = (sw_2 2 0^{\ell_2})_k$ and $y_2 = ((j-2)t0^{m_2})_k$, with $m_2 < \ell_2 + |w_2|$. Then, depending on m_2, the base-k notation of the number $x_2 + y_2$ is of one of the three mutually exclusive forms.*

i. *If $m_1 > \ell_2 - 1$, then $x_2 + y_2$ has a digit at least $c + d + 1$ at position $m_2 + 2$ (as illustrated in Fig. 3(c,d)).*
ii. *If $m_2 = \ell_2 - 1$, as in Fig. 3(b), then $x_2 + y_2 \in (sKjt0^*)_k$.*
iii. *If $m_2 < \ell_2 - 1$, as in Fig. 3(a), then $x_2 + y_2 \in (s\Omega^* 2 0^* (j-2)t0^*)_k$.*

The rest of the proof of Lemma 5 follows the same plan as in the proof of Lemma 3. Investigations of possible forms of numbers $x_1 + y_1$ and $x_2 + y_2$ yield that the equality $x_1 + y_1 = x_2 + y_2$ given in Claim 3 and Claim 4 is possible only when this number belongs to $(sKjt0^*)_k$. The case analysis is similar to the one following Claims 1 and 2 in Lemma 3. □

The next claim is that the concatenations in ρ_j^t are unambiguous under the right substitution.

Lemma 6. *Let $K \subseteq \Omega^*$ and let $s \in \{1, 2\}$. Define $L_1 = a^{(sK1 0^*)_k}$ and $L_2 = a^{(sK2 0^*)_k}$. Then, for each $j \in \Omega$ and $t \in \{1, 2\}$, both concatenations in $\rho_j^t(L_1, L_2)$ are unambiguous.*

Sketch of a proof. The proof is similar to the proof of Lemma 4: consider first the concatenation $a^{(sK1 0^*)_k} \cdot a^{((j-1)t0^*)_k}$ from $\rho_j^t(L_1, L_2)$. To see that it is unambiguous, it is enough to show that for $x_1, x_2 \in (sKr0^*)_k$ and $y_1, y_2 \in ((j-r)t0^*)_k$, the equality $x_1 + y_1 = x_2 + y_2$ implies $x_1 = x_2$ and $y_1 = y_2$, for any $r, s, t \in \{1, 2\}$. Let these four numbers be of the following form.

$$x_1 = (sw_1 r 0^{\ell_1})_k \in (sKr0^*)_k$$
$$y_1 = ((j-r)t0^{m_1})_k \in ((j-r)t0^*)_k$$
$$x_2 = (sw_2 r 0^{\ell_2})_k \in (sKr0^*)_k$$
$$y_2 = ((j-r)t0^{m_2})_k \in ((j-r)t0^*)_k$$

It is enough to show that $m_1 = m_2$, as this implies $y_1 = y_2$ and consequently $x_1 = x_2$. The following table gives the possible leading digits of $x_1 + y_1$ and $x_2 + y_2$.

		$m_2 < \ell_2 + \|w_2\|$ digit s or $s+1$	$m_2 = \ell_2 + \|w_2\|$ digit $s + j - r$	$m_2 > \ell_2 + \|w_2\|$ digit $j - r$
$m_1 < \ell_1 + \|w_1\|$	digit s or $s+1$	may be equal	$s+1 < s+j-r$	$s+1 < j-r$
$m_1 = \ell_1 + \|w_1\|$	digit $s+j-r$	$s+j-r > s+1$	equal	$s+j-r > j-r$
$m_1 > \ell_1 + \|w_1\|$	digit $j-r$	$j-r > s$	$j-r < s+j-r$	equal

The investigation yields three possible cases, in which it can hold that $x_1 + y_1 = x_2 + y_2$. In one case, when $m_1 > \ell_1 + |w_1|$ and $m_2 > \ell_2 + |w_2|$, the resulting number $x_1 + y_1$ has base-k representation of length $m_1 + 2$, whereas for $x_2 + y_2$ it is of length $m_2 + 2$; thus, $m_1 = m_2$. The same argument applies to $m_1 = \ell_1 + |w_1|$ and $m_2 = \ell_2 + |w_2|$.

The remaining case to be considered is $m_1 < \ell_1 + |w_1|$ and $m_2 < \ell_2 + |w_2|$. In this case, the form of numbers $x_1 + y_1$ and $x_2 + y_2$ is given by Claim 3 (for $r = 1$) and Claim 4 (for $r = 2$). According to the case inspection given there, either

$m_1 > \ell_1 - 1$ and $m_2 > \ell_2 - 1$, *or* $m_1 = \ell_1 - 1$ and $m_2 = \ell_2 - 1$, *or* $m_1 < \ell_1 - 1$ and $m_2 < \ell_2 - 1$. In the first case, in $x_1 + y_1$ and $x_2 + y_2$, there are distinctive digits at positions $m_1 + 2$ and $m_2 + 2$, and so $m_1 = m_2$, as claimed. In the second and third cases, the last non-zero digits in $x_1 + y_1$ and $x_2 + y_2$ are at positions $m_1 + 1$ and $m_2 + 1$, respectively, and therefore $m_1 = m_2$. □

With the properties of λ_i^s and ρ_j^t established, the theorem is proved as follows.

Proof of Theorem 1. The claim is that the given conjunctive grammar (10) is unambiguous, and each nonterminal $C_q^{s,t}$, with $s, t \in \{1, 2\}$ and $q \in Q$, describes a language $\{a^{(swt0^\ell)_k} \mid \Delta(w) = q,\ \ell \geqslant 0\}$, whereas S describes $\{a^{(1w10^\ell)_k} \mid a^{(w)_k} \in L(M), \ell \geqslant 0\}$.

Consider the system of language equations associated with the grammar (10). The languages described by the nonterminals $A_{i,j}$ have already been determined in Lemma 2, and can be treated as constants in this argument. All these constants are ε-free. Hence, the associated resolved system uses only constants that are ε-free and therefore has a unique ε-free solution [3, Thm. 2.3], which is also the least solution. Accordingly, it is enough to check that $C_q^{s,t} = \{a^{(swt0^\ell)_k} \mid \Delta(w) = q,\ \ell \geqslant 0\}$ is indeed a solution.

For every nonterminal $C_q^{s,t}$, consider first any rules of the form (10c); it has as many of those as there are symbols $j \in \Omega$ with $I(j) = q$. Every such rule has the right-hand side $\lambda_j^s(A_{1,t}, A_{2,t})$, and should define the corresponding language $a^{(sjt0^*)_k}$. The value of this expression under the substitution of $a^{(1t0^*)_k}$ for $A_{1,t}$ and $a^{(2t0^*)_k}$ for $A_{2,t}$ is given by Lemma 3, which is used with $K = \{\varepsilon\}$ and with the arguments $L_1 = a^{(1Kt0^*)_k}$ and $L_2 = a^{(2Kt0^*)_k}$. The lemma asserts that this is the desired language.

$$\lambda_j^s(a^{(1t0^*)_k}, a^{(2t0^*)_k}) = a^{(sjKt0^*)_k} = a^{(sjt0^*)_k}$$

Taking the union over all rules of the form (10c) for $C_q^{s,t}$, that is, over all digits $j \in \Omega$ with $\Delta(j) = q$, yields the following language.

$$\bigcup_{\substack{j \in \Omega \\ \Delta(j) = q}} \lambda_j^s(a^{(1t0^*)_k}, a^{(2t0^*)_k}) = \{a^{(sjt0^\ell)_k} \mid j \in \Omega,\ \Delta(j) = q,\ \ell \geqslant 0\}$$

This union is disjoint, because each set $\lambda_j^s(a^{(1t0^*)_k}, a^{(2t0^*)_k})$ consists of numbers with the second most significant digit j.

Each nonterminal symbol $C_q^{s,t}$ may also have one or more rules of the form (10d), corresponding to different pairs of states $p, r \in Q$, for which $\delta(p, r) = q$, and to various digits $i, j \in \Omega$ concatenated on the sides. Every such rule is a conjunction of two expressions, $\lambda_i^s(C_r^{1,t}, C_r^{2,t})$ and $\rho_j^t(C_p^{s,1}, C_p^{s,2})$, and their values under the desired substitution are determined by Lemmata 3 and 5. For $\lambda_i^s(C_r^{1,t}, C_r^{2,t})$, it is given by Lemma 3, with $K_r = \{w \mid \Delta(w) = r\}$, $L_1 = a^{(1K_rt0^*)_k}$ and $L_2 = a^{(2K_rt0^*)_k}$, which states the following equality.

$$\lambda_i^s(a^{(1K_rt0^*)_k}, a^{(2K_rt0^*)_k}) = a^{(siK_rt0^*)_k}$$

Applying Lemma 5 to the second expression, with the values $K_p = \{w \mid \Delta(w) = p\}$, $L_1 = a^{(sK_p10^*)_k}$ and $L_2 = a^{(sK_p20^*)_k}$, gives the value of the other representation.

$$\rho_j^t(a^{(sK_p10^*)_k}, a^{(sK_p20^*)_k}) = a^{(sK_pjt0^*)_k}$$

Then the conjunction in (10d) defines the following language.

$$\begin{aligned}&\lambda_i^s(a^{(1K_rt0^*)_k}, a^{(2K_rt0^*)_k}) \cap \rho_j^t(a^{(sK_p10^*)_k}, a^{(sK_p20^*)_k}) = \\ &\quad = a^{(siK_rt0^*)_k} \cap a^{(sK_pjt0^*)_k} = \{a^{(siwjt0^\ell)_k} \mid \Delta(iw) = p,\ \Delta(wj) = r,\ \ell \geqslant 0\}\end{aligned}$$

The union of these expressions over all rules of the form (10d) for the symbol $C_q^{s,t}$ reflects all possible strings of length 2, on which the trellis automaton reaches the state q.

$$\bigcup_{i,j \in \Omega} \bigcup_{\substack{p,r \in Q, \\ \delta(p,r) = q}} \{a^{(siwjt0^\ell)_k} \mid w \in \Omega^*,\ \Delta(iw) = p,\ \Delta(wj) = r,\ \ell \geqslant 0\} = \{a^{(siwjt0^\ell)_k} \mid w \in \Omega^*,\ \Delta(iwj) = q,\ \ell \geqslant 0\}$$

This union is disjoint, because each string $a^{(siwjt0^\ell)_k}$ may appear only in one particular component, the one with the digits i and j in the first union, and with the states $p = \Delta(iw)$ and $r = \Delta(wj)$ in the second union.

The entire equation for the variable $C_q^{s,t}$ is defined as the union over all rules (10c) and (10d), and it holds true as follows.

$$C_q^{s,t} = \{a^{(sw't0^\ell)_k} \mid w' \in \Omega^*,\ \Delta(w') = q,\ |w'| \geqslant 2,\ \ell \geqslant 0\} \cup \{a^{(sw't0^\ell)_k} \mid w' \in \Omega^*,\ \Delta(w') = q,\ |w'| = 1,\ \ell \geqslant 0\} =$$
$$= \{a^{(sw't0^\ell)_k} \mid w' \in \Omega^*,\ \Delta(w') = q,\ \ell \geqslant 0\}$$

Since the conditions $|w'| = 1$ and $|w'| \geqslant 2$ are mutually exclusive, this is again a union of disjoint sets, and consequently, the choice of a rule for $C_q^{s,t}$ is unambiguous.

Lastly, consider the equations for S and O. Clearly, O describes the language $\{a^{km+1} \mid m \geqslant 0\}$, and the two rules for O define disjoint sets. Then, as the value of the variable $C_q^{1,1}$ is already known, the intersection in the rule for S defines exactly the desired set.

$$S = \bigcup_{q \in F} \{a^{(1w10^\ell)_k} \mid \Delta(w) = q,\ \ell \geqslant 0\} \cap \{a^{km+1} \mid m \geqslant 0\} = \{a^{(1w1)_k} \mid \Delta(w) \in F\} = \{a^{(1w1)_k} \mid w \in L(M)\}$$

Turning to the unambiguity of the grammar (10), Lemma 2 supplies an unambiguous conjunctive grammar for the languages $a^{(ij0^*)_k}$, and it remains to show that each concatenation and union in (10) is unambiguous. The only concatenations that appear in (10) are in the subexpressions λ_i^s and ρ_j^t, and in each case, the assumptions of Lemmata 4 and 6 are met; consequently, all concatenations in the system (10) are unambiguous. The unambiguity of the choice of a rule has already been shown above. □

5. A non-sparse encoding of trellis automata

For a language $L \subseteq \Sigma^*$, consider the fraction of strings of length up to n that belong to L, as a function of n. The limit of this fraction, denoted by $d(L)$, is called *the density of* L [5].

$$d(L) = \lim_{n \to \infty} \frac{\left|L \cap \Sigma^{<n}\right|}{|\Sigma^{<n}|}$$

This limit, if it exists, always lies within the bounds $0 \leqslant d(L) \leqslant 1$.

In this paper, it is mostly the density of *unary languages* that is investigated, and for $L \subseteq a^*$ the definition simplifies to the following.

$$d(L) = \lim_{n \to \infty} \frac{|L \cap \{\varepsilon, a, a^2, \ldots, a^{n-1}\}|}{n}.$$

The general upper and lower bounds are the same: $0 \leqslant d(L) \leqslant 1$. Let a unary language be called *sparse*, if $d(L) = 0$, and *dense*, if $d(L) = 1$ [30].

All unambiguous conjunctive grammars constructed so far describe sparse unary languages. Using only sparse languages in the constructions is actually a necessity, because languages are expressed in the grammar by concatenating them to each other, and a concatenation of a non-sparse unary language with any infinite language is bound to be ambiguous.

Lemma C *(Jeż and Okhotin [14, Lemma 3], proof in an upcoming full version). Let $K \subseteq a^*$ be any infinite language, and let $L \subseteq a^*$ be a language that is not of density 0, that is, $d(L)$ is either undefined or greater then zero. Then the concatenation KL is ambiguous.*

Of course, this does not mean that non-sparse sets cannot be described at all—for instance, one can modify the grammar in Example 2 to describe the language $L_{1/2} = \{a^{4^n} \mid n \geqslant 0\} \cup a(aa)^*$ of density $\frac{1}{2}$. However, once represented, non-sparse sets cannot be non-trivially concatenated to anything, and therefore cannot be used in any further constructions.

The above language $L_{1/2}$ is a union of a sparse language and a regular language. In this way, one can represent any union of sparse subsets of disjoint regular languages, with any extra regular language—but nothing beyond that.

Another method for constructing unambiguous conjunctive grammars that describe non-sparse languages shall be developed in this section. Those non-sparse languages shall be obtained through another simulation of a trellis automaton recognizing base-k notation of numbers. This time, different base-k strings accepted by the automaton shall contribute disjoint infinite subsets of a^* to the resulting unary language. In particular, if the given trellis automaton accepts a language of density 0, then the resulting unary language shall also have density 0, and if the trellis automaton accepts almost all strings, then the unary language shall have density 1.

The proof of the new result is done on top of the construction in Theorem 1, so that a given trellis automaton is first simulated using only languages of density 0, and at the last step, several languages of density 0 are concatenated to obtain the desired language. The latter concatenation of languages of density 0 is modeled upon the following known representation of the language of all unary strings as an unambiguous concatenation.

Example 3 *(Enflo et al. [8]).* Let the base of positional notation $k \geqslant 2$ be any power of two, and consider the languages L_1, L_2, L_4, L_8, …, $L_{\frac{k}{2}}$, defined by $L_i = a^{(i\{i,0\}^*)_k} \cup \{\varepsilon\}$. Then $L_1 L_2 L_4 L_8 \ldots L_{\frac{k}{2}} = a^*$, and this concatenation is unambiguous.

The correctness of Example 3 is based on the fact that every number $n \geqslant 0$ has a unique representation as a sum $n = n_1 + n_2 + n_4 + n_8 + \ldots + n_{\frac{k}{2}}$, where the base-$k$ notation for each number n_t uses only the digits 0 and t. Furthermore, each digit in the base-k representation of each n_t can be determined from the corresponding digit of n, as follows. Let $i \in \{0, \ldots, k-1\}$ be a base-k digit; its binary notation uniquely represents it as a sum of powers of two. Whenever the binary representation of a digit i in n contains some power of two, t, the corresponding number n_t has t in the same position; otherwise, it has the zero there.

The mapping of digits mentioned above is denoted by $h_t \colon \Sigma_k \to \{0, t\}$, defined as follows.

$$h_t(i) = \begin{cases} 0, & \text{if } \lfloor \frac{i}{t} \rfloor \text{ is even} \\ t, & \text{if } \lfloor \frac{i}{t} \rfloor \text{ is odd} \end{cases}$$

The mapping is extended to a homomorphism $h_t \colon \Sigma_k^* \to \{0, t\}^*$. Then, the correctness of Example 3 is formalized as follows.

Lemma 7. *Let $k \geqslant 2$ be any power of two. Then, every integer $n \geqslant 0$ is uniquely representable as a sum $n = n_1 + n_2 + n_4 + n_8 + \ldots + n_{\frac{k}{2}}$, with $n_t \in (t\{0,t\}^*)_k \cup \{0\}$. Furthermore, if w is the base-k notation of n, then $h_t(w)$ is the base-k notation of n_t, possibly with zero digits pre-pended.*

Leading zeroes appear in $h_t(w)$ for the reason that h_t is applied digit by digit, and the projection of a non-zero digit by h_t may still be 0. For example, in base-8 notation, the number $(12345670)_8$ is uniquely represented as follows.

$$(12345670)_8 = (10101010)_8 + (2200220)_8 + (44440)_8$$

Direct digit mapping gives $h_1(12345670) = 10101010$, $h_2(12345670) = 02200220$ and $h_4(12345670) = 00044440$, with leading zeroes pre-pended in the latter two cases.

Let one of the languages L_i in the concatenation $L_1 \ldots L_{\frac{i}{2}} L_i L_{2i} \ldots L_{\frac{k}{2}}$ be replaced with any subset $L'_i \subseteq L_i$, which encodes the computation of a trellis automaton operating on the two-symbol input alphabet $\{0, i\}$, similarly to the encoding in Theorem 1. Then the concatenation $L = L_1 \ldots L_{\frac{i}{2}} L'_i L_{2i} \ldots L_{\frac{k}{2}}$ is still unambiguous, and the composition of the language L is determined by the given linear conjunctive language. In particular, if the trellis automaton accepts all strings, then $L'_i = L_i$ and $L = a^*$; and if the automaton accepts nothing, then $L'_i = \varnothing$ and $L = \varnothing$.

This construction, with $i = 2$, allows the following languages to be described by unambiguous conjunctive grammars.

Theorem 2. *Let K be a linear conjunctive language over a two-symbol alphabet $\Gamma = \{e, f\}$, which does not contain any strings beginning with e. Let $k \geqslant 16$ be any power of two, and define a homomorphism $h \colon \Sigma_k^* \to \Gamma^*$ that maps each base-k digit to e or to f, by setting $h(4i) = h(4i+1) = e$ and $h(4i+2) = h(4i+3) = f$ for all $i \in \{0, \ldots, \frac{k}{4} - 1\}$. Then there exists an unambiguous conjunctive grammar describing the language $\{a^{(w)_k} \mid h(w) \in e^*K,\ w \in \Sigma_k^* \setminus 0\Sigma_k^*\}$. Given a trellis automaton recognizing K, this grammar can be effectively constructed.*

The first step towards proving the theorem according to the above general plan is to describe the necessary constant languages L_1, L_2, L_4, L_8, …, $L_{\frac{k}{2}}$.

Lemma 8. *For all $k \geqslant 16$ and $t \in \{1, \ldots, \frac{k}{2}\}$, where k is a power of two, there is an unambiguous conjunctive grammar that describes the language $a^{(t\{0,t\}^*)_k}$.*

Sketch of a proof. The grammar reuses the nonterminals $A_{i,j}$ given in Lemma 2, which describe the languages $a^{(ij0^*)_k}$. There are three new nonterminals: $X_{\frac{k}{2}+1}$, $X_{\frac{k}{2}+5}$ and the initial symbol C. Each X_s, with $s \in \{\frac{k}{2}+1, \frac{k}{2}+5\}$, describes an auxiliary language $L(X_s) = a^{(s\{0,t\}^*)_k}$. The membership of strings in these languages is defined inductively of their length. The base rules put the shortest strings in $X_{\frac{k}{2}+1}$ and in $X_{\frac{k}{2}+5}$.

$$X_s \to a^{(s)_k} \qquad \text{for } s \in \{\tfrac{k}{2}+1, \tfrac{k}{2}+5\}$$

Every next string $a^{(sjw)_k}$, with $s \in \{\frac{k}{2}+1, \frac{k}{2}+5\}$, $j \in \{0, t\}$ and $w \in \{0, t\}^*$, is defined by the corresponding X_s using the following rule.

$$X_s \to \mathop{\&}_{i \in \{\frac{k}{2}+1, \frac{k}{2}+5\}} A_{s-1, k+j-i} X_i \qquad \text{for } j \in \{0, t\} \text{ and } s \in \{\tfrac{k}{2}+1, \tfrac{k}{2}+5\}$$

This rule expresses the number $(sjw)_k$ in two ways: as a sum of $((s-1)(\frac{k}{2}+j-1)0^{|w|})_k$ and $((\frac{k}{2}+1)w)_k$, and as a sum of $((s-1)(\frac{k}{2}+j-5)0^{|w|})_k$ and $((\frac{k}{2}+5)w)_k$. A conjunction of these two representation filters out the garbage.

The initial symbol C defines the desired language $L(C) = a^{(t\{0,t\}^*)_k}$ using rules very similar to those for X_s. For each digit $j \in \{0, t\}$, there is a rule that defines all strings of the form $a^{(tjw)_k}$, with $w \in \{0, t\}^*$.

$$C \to \&_{i \in \{\frac{k}{2}+1, \frac{k}{2}+5\}} A_{t-1,k+j-i} X_i \qquad \text{for } j \in \{0, t\}$$

The shortest string in the desired language is defined by a special rule.

$$C \to a^{(t)_k}$$

Thus, each of the three nonterminal symbols $X_{\frac{k}{2}+1}$, $X_{\frac{k}{2}+5}$ and C has three rules, which generate disjoint languages. The unambiguity of concatenation could be proved by the same methods as in Lemmata 2 and 4. □

Lemma 9. *For $k \geqslant 16$ and $\omega \in \{\frac{k}{2}-1, \ldots, k-2, k-1\}$, where k is a power of two, there is an unambiguous conjunctive grammar that describes the language $a^{(\omega^+)_k}$.*

Sketch of a proof. The construction is a variant of the construction of languages in Lemma 2; it is different only in using the digit ω instead of the zero.

Let the nonterminals $A_{i,j}$ define the languages $a^{(ij0^*)_k}$ using the rules given in Lemma 2. Two new nonterminals, Y_1 and Y_2, should describe the languages $a^{(1\omega^*)_k}$ and $a^{(2\omega^*)_k}$, respectively. These languages are represented through each other by the following recursive definition, which expresses a number $(s\omega^{\ell+1})_k$ as a sum of $(1\omega^\ell)_k$ and $(s(\omega-1)0^\ell)_k$, and as a sum of $(2\omega^\ell)_k$ and $(s(\omega-2)0^\ell)_k$.

$$Y_s \to \&_{i \in \{1,2\}} A_{s,\omega-i} Y_i \mid a^{(s)_k} \qquad \text{for } s \in \{1, 2\}$$

The garbage in these two representations is disjoint, and is eliminated out by the intersection. Then a third nonterminal D defines the desired language $a^{(\omega^+)_k}$ by the following rule.

$$D \to \&_{i \in \{1,2\}} A_{\omega-i,0} Y_i$$

The correctness proof and the argument for unambiguity follow in the same way as for Lemma 2. □

Then, a linear conjunctive language is encoded within a subset of L_2 as follows.

Lemma 10. *For every linear conjunctive language $K \subseteq \{0,2\}^* \setminus 0\{0,2\}^*$, and for every base $k \geqslant 16$ that is a power of two, there is an unambiguous conjunctive grammar that describes the language $\{a^{(w)_k} \mid w \in K\}$.*

Sketch of a proof. Consider the following two languages, obtained by removing the first and the last digits in all strings in K.

$$K^{(0)} = \{w \mid w \in \{0,2\}^+,\ 2w0 \in K\}$$
$$K^{(2)} = \{w \mid w \in \{0,2\}^+,\ 2w2 \in K\}$$

By the known closure properties of linear conjunctive languages, both $K^{(0)}$ and $K^{(2)}$ are linear conjunctive. Let M_0 and M_2 be such trellis automata, that $L(M_0) = K^{(0)}$ and $L(M_2) = K^{(2)}$.

In the first part of the argument, the automaton M_0 is used to construct a grammar describing the language $a^{(K \cap 2\{0,2\}^*0)_k}$. Construct a trellis automaton M_0' by modifying M_0 to use the input alphabet $\{5,7\}$ instead of $\{0,2\}$. Then M_0' satisfies the assumptions of Theorem 1 for $c=5$ and $d=3$; these values $c=5$, $d=3$ are fixed for the rest of the proof.

By Theorem 1, there is an unambiguous conjunctive grammar with a nonterminal Z that defines the language $\{a^{(1w1)_k} \mid w \in L(M_0')\}$. Next, the goal is to modify this encoding by concatenating it with an auxiliary language $a^{((k-c-1)^*)_k}$, designed to shift the digits 5 and 7 in the strings accepted by M_0' back to 0 and 2. Malformed sums shall be filtered out by intersecting this concatenation with another auxiliary language $a^{(2\{0,2\}^*)_k}$. The grammars describing these two auxiliary languages are already known. First, consider the grammar given in Lemma 9, for $\omega = k-c-1$, and let D be the nonterminal with $L(D) = a^{((k-c-1)^+)_k}$. The second grammar is given in Lemma 8, with $t=2$, and it has a nonterminal C describing the language $L(C) = a^{(2\{0,2\}^*)_k}$.

Now all the above grammars are combined into one. Let $Z^{(0)}$ be a new nonterminal with the following single rule.

$$Z^{(0)} \to ZDa^c \,\&\, C$$

This rule operates as follows. Consider any unary string $a^{(1(b_m+c)(b_{m-1}+c)\ldots(b_1+c)1)_k}$ in $L(Z)$, where $(b_m+c)\ldots(b_1+c) \in \{c, c+2\}^*$ is a string of length m accepted by M_0', and $b_m \ldots b_1 \in \{0,2\}^*$ is the corresponding unmodified string accepted by M_0. Then, concatenating the above unary string to the string $a^{((k-c-1)^{m+2})_k}$ from $L(D)$, and further appending a^c, produces the desired unary representation of a string in $L(M_0)$.

$$a^{(1(b_m+c)(b_{m-1}+c)\ldots(b_1+c)1)_k} \cdot a^{((k-c-1)^{m+2})_k} \cdot a^c = a^{(2b_m b_{m-1}\ldots b_1 0)_k} \tag{15}$$

$$\begin{array}{cccccc} & {\scriptstyle 1} & {\scriptstyle 1} & & {\scriptstyle 1} & \\ & \mathbf{1} & b_m{+}c & \dots & b_1{+}c & \mathbf{1} \\ + & & k{-}c{-}1 & \dots & k{-}c{-}1 & k{-}c{-}1 \\ + & & & & & c \\ \hline & \mathbf{2} & b_m & \dots & b_1 & \mathbf{0} \end{array}$$

Fig. 4. Addition carried out in the concatenation $Z^{(0)}Da^c$ in Lemma 10.

The addition of these three numbers is illustrated in Fig. 4. The ensuing intersection with the nonterminal C, which describes the language $a^{(2\{0,2\}^*)_k}$, ensures that every element of ZDa^c that is not of the form (15) is filtered out by the intersection. A formal proof that the rule indeed describes the given language can be carried out similarly to Lemma 3. Also the unambiguity of concatenation follows in the same way as in Lemma 4.

Altogether, $Z^{(0)}$ defines the following language.

$$L(Z^{(0)}) = \{a^{(2w0)_k} \mid w \in L(M_0)\} = \{a^{(2w0)_k} \mid w \in 2\{0,2\}^+0,\ 2w0 \in K\} = a^{(K \cap 2\{0,2\}^+0)_k}$$

An unambiguous conjunctive grammar describing the language $a^{(L \cap 2\{0,2\}^*2)_k}$ is constructed by applying the same construction to the automaton M_2 recognizing the language $L^{(2)}$. First, consider the automaton M_2' that replicates M_2 over the alphabet $\{c, c+2\}$. Then, applying Theorem 1 to this automaton and adding a rule similar to the one for $Z^{(0)}$, one can construct a grammar with a new nonterminal $Z^{(2)}$ that describes the following language.

$$L(Z^{(2)}) = \{a^{(2w2)_k} \mid w \in L(M_2)\} = \{a^{(2w2)_k} \mid w \in 2\{0,2\}^+0,\ 2w2 \in K\} = a^{(K \cap 2\{0,2\}^+2)_k}$$

The languages defined by $Z^{(0)}$ and $Z^{(2)}$ are disjoint, and together they constitute all but finitely many of the strings in $a^{(K)_k}$. It remains to add a new nonterminal symbol Z' to the grammar, and let it have the following rules.

$$\begin{aligned} Z' &\to Z^{(0)} \mid Z^{(2)} \\ Z' &\to a^m \qquad\qquad \text{for } m \in (K \cap \Sigma^{\leqslant 2})_k \end{aligned}$$

Then Z' represents the desired language as the following disjoint union.

$$a^{(K \cap 2\{0,2\}^+0)_k} \cup a^{(K \cap 2\{0,2\}^+2)_k} \cup a^{(K \cap \Sigma^{\leqslant 2})_k} = a^{(K)_k} \qquad \square$$

Using the specially modified unary simulation of a trellis automaton given in Lemma 10, the theorem on representing languages of various density is proved as follows.

Proof of Theorem 2. For the proof, it is convenient to rename e to 0 and f to 2, so that the alphabet of K becomes a subset of the alphabet of base-k digits, and Lemma 10 becomes applicable to K. Then the homomorphism $h\colon \Sigma_k \to \{0,2\}$ is exactly h_2 in Lemma 7. That is, for each digit $d \in \Sigma_k$, consider its representation as a sum $d_1 + d_2 + d_4 + d_8 + \ldots + d_{\frac{k}{2}}$, with $d_t \in \{0, t\}$; then the image of d is d_2.

$$h(d_1 + d_2 + d_4 + d_8 + \ldots + d_{\frac{k}{2}}) = d_2 \qquad (d_t \in \{0,t\},\ \text{for each } i)$$

By Lemma 8, each language $a^{(t\{0,t\}^*)_k}$, with $t \in \{1, 2, 4, 8, \ldots, \frac{k}{2}\}$, is described by an unambiguous conjunctive grammar, and hence so are the languages $L_t = a^{(t\{0,t\}^*)_k} \cup \{\varepsilon\}$. Then, by Lemma 7, the concatenation $L_1 L_2 L_4 \ldots L_{\frac{k}{2}}$ is unambiguous and equal to a^*.

The theorem is proved by replacing L_2 in this concatenation with the unary language given by Lemma 10 for the language K. The lemma yields an unambiguous conjunctive grammar that describes the language $L_2' = \{a^{(w)_k} \mid w \in K\}$. As $L_2' \subseteq L_2$, the concatenation $L_1 L_2' L_4 \ldots L_{\frac{k}{2}}$ is unambiguous, and it remains to show that this is exactly the desired language.

Claim. $L_1 L_2' L_4 \ldots L_{\frac{k}{2}} = \{a^{(w)_k} \mid h(w) \in 0^*K,\ w \in \Sigma_k^* \setminus 0\Sigma_k^*\}$.

$\Rightarrow$ Consider any string $a^{(w)_k}$ in $L_1 L_2' L_4 \ldots L_{\frac{k}{2}}$. Then, the number $a^{(w)_k}$ can be represented as a sum $(w)_k = (w_1)_k + (w_2)_k + (w_4)_k + \ldots + (w_{\frac{k}{2}})_k$, where each number $(w_i)_k$, with $i \in \{1, 2, 4, \ldots, \frac{k}{2}\}$, corresponds to a string $a^{(w_i)_k} \in L_i$, and $a^{(w_2)_k} \in L_2'$ for $i = 2$. By the definition of L_2', the string of digits w_2 must be in K. Since the representation of $(w)_k$ as a sum of $(w_t)_k$ is unique by Lemma 7, the representation given above is the same as the one given in the lemma. Then, in particular, $h(w)$ is of the form $0^m w_2$, for some $m \geqslant 0$, and is therefore in 0^*K.

$\Leftarrow$ Conversely, consider a string $a^{(w)_k}$, with $w \in \Sigma_k^* \setminus 0\Sigma_k^*$ and $h(w) \in 0^*K$. According to Lemma 7, the number $(w)_k$ is represented as $(w)_k = (w_1)_k + (w_2)_k + (w_4)_k + \ldots + (w_{\frac{k}{2}})_k$, where $w_t \in t\{0,t\}^*$ for each t, and, in particular, $h(w) = 0^m w_2$,

for some $m \geqslant 0$. Since neither w_2 nor any strings in K have leading zeroes, this implies that w_2 must be in K. Then, by the construction of L'_2, it must contain the string $a^{(w_2)_k}$. This is exactly what is needed for the string $a^{(w)_k}$ to belong to the desired concatenation.

$$a^{(w)_k} = a^{(w_1)_k} a^{(w_2)_k} a^{(w_4)_k} \dots a^{(w_{\frac{k}{2}})_k} \in L_1 L'_2 L_4 \dots L_{\frac{k}{2}}$$

Indeed, for any other summand $(w_t)_k$, its base-k notation is in $t\{0,t\}^*$, and this is enough for $a^{(w_t)_k}$ to be in L_t. □

Since the concatenation $L_1 L_2 L_4 \dots L_{\frac{k}{2}}$ has been presented as a concatenation of languages of density 0 that itself has density 1, it is natural to ask, what is the density of the concatenation $L = L_1 L'_2 L_4 \dots L_{\frac{k}{2}}$ constructed in Theorem 2, where $L'_2 = \{a^{(w)_k} \mid w \in K\}$. A certain connection between the densities of the base-k language K and of the resulting unary language L is established in the following lemma.

Lemma 11. *In Theorem 2, if the resulting language $L = \{a^{(w)_k} \mid h(w) \in e^*K,\ w \in \Sigma_k^* \setminus 0\Sigma_k^*\}$ has a density, then $d(K)$ is defined and it equals $\frac{1}{2}d(L)$.*

Furthermore, if K has density 0, then L has density 0 as well, and if K has density $\frac{1}{2}$, then L has density 1.

The scaling factor of one half has a trivial explanation: it is only caused by the fact that binary notation never begins with a zero, and then the language of all valid binary representations of numbers has density $d(1\{0,1\}^*) = \frac{1}{2}$. The statement of the lemma should be understood in the sense that the density stays the same.

Proof. All claims of the Lemma require a correspondence between the number of strings of length up to m in K and the number of strings up to a^{k^m} in L. These numbers are related as follows.

Claim 5. *Let L encode the language K as in Theorem 2 and m be a natural number. Then*

$$\frac{\left|L \cap a^{<k^m}\right|}{k^m} = \frac{\left|K \cap \{e,f\}^{<m+1}\right|}{2^m}$$

According to Theorem 2, for every string $u \in K$, with $|u| \leqslant m$, the language L contains all strings $w \in \Sigma_k^{<m+1}$ not beginning with 0 that have homomorphic images of the form $h(w) \in e^*u$. The number of such strings w is exactly the same as the number of padded strings $0^{|w|-m}w$ with homomorphic images $h(0^{|w|-m}w) \in e^*u$, for the reason that $h(0) = e$. As h maps $k/2$ digits to f and $k/2$ to e, there are $(k/2)^m$ such strings w in L. For different $u \in K$, the corresponding subsets of L are disjoint. Therefore, $|L \cap a^{<k^m}| = (k/2)^m \cdot |K \cap \{e,f\}^{<m+1}|$, as desired.

Turning to the proof of the lemma, assuming that $d(L)$ is defined, the limit defining the density of K is calculated as follows.

$$d(K) = \lim_{m\to\infty} \frac{\left|K \cap \{e,f\}^{<m}\right|}{\{e,f\}^{<m}} = \lim_{m\to\infty} \frac{2^{m-1}}{2^m - 1} \cdot \frac{\left|K \cap \{e,f\}^{<m}\right|}{2^{m-1}} \overset{\text{Claim 5}}{=} \lim_{m\to\infty} \frac{2^{m-1}}{2^m - 1} \cdot \frac{\left|L \cap a^{<k^{m-1}}\right|}{k^{m-1}} = \frac{1}{2} \cdot d(L)$$

The last equality follows from the fact that the sequence in question is a subsequence defining the density $d(L)$.

Now assume that $d(K) = 0$, it is shown that $d(L) = 0$ as well. Consider the sequences $x_n = \frac{|L \cap a^{<n}|}{n}$ and $y_m = \frac{\left|L \cap a^{<k^m}\right|}{k^m}$. First, it is proved that y_m converges to zero.

$$\lim_{m\to\infty} y_m = \lim_{m\to\infty} \frac{\left|L \cap a^{<k^m}\right|}{k^m} \overset{\text{Claim 5}}{=} \lim_{m\to\infty} \frac{\left|K \cap \{e,f\}^{<m+1}\right|}{2^m} = \lim_{m\to\infty} \frac{2^{m+1}-1}{2^m} \cdot \frac{\left|K \cap \{e,f\}^{<m+1}\right|}{\left|\{e,f\}^{<m+1}\right|} = 2 \cdot d(K) = 0$$

To see that x_n converges to 0 as well, fix n and let m be such that $k^m \leqslant n < k^{m+1}$. Then

$$\frac{1}{k} \cdot y_m = \frac{\left|L \cap a^{<k^m}\right|}{k^{m+1}} < x_n \leqslant \frac{\left|L \cap a^{<k^{m+1}}\right|}{k^m} = k \cdot y_{m+1}$$

Since y_m converges to zero, so does x_n.

Let now $d(K) = \frac{1}{2}$. Consider also its complement K' in $f\{e,f\}^* \cup \{\varepsilon\}$, note that $d(K') = 0$; let L and L' be their encodings as in Theorem 2; as the definitions of those encodings refer to sets of the form e^*K and e^*K', observe first that

$$e^*K \cup eK' = e^*(K \cup K') = e^*(f\{e,f\}^* \cup \{\varepsilon\}) = \{e,f\}^*,$$

where the first equality follows from the fact that K and K' have no strings beginning with e. Similarly,

$$e^*K \cap e^*K' = e^*(K \cap K') = \varnothing.$$

As a consequence, similar equalities hold also for the corresponding base-k languages L and L':

$$L \cup L' = \{a^{(w)_k} \mid h(w) \in e^*K \cup e^*K',\ w \in \Sigma_k^* \setminus 0\Sigma_k^*\} = \{a^{(w)_k} \mid h(w) \in \{e, f\}^*,\ w \in \Sigma_k^* \setminus 0\Sigma_k^*\} = a^*$$

and similar calculations yield that $L \cap L' = \varnothing$. Hence

$$d(L) = d(a^* \setminus L') = d(a^*) - d(L') = 1 - 0 = 1,$$

which ends the proof of the lemma. □

The density of the resulting unary language is determined by the density of K *if* it is defined—but it is not always defined. This is demonstrated in the following example.

Example 4. The language $K = fe\{e, f\}^* \cup \{e\}$ has density $\frac{1}{4}$. However, for the corresponding language L defined as in Theorem 2, for $k = 16$, its density is not defined.

Proof. For each length $m \geqslant 2$, the language $K = fe\{e, f\}^* \cup \{e\}$ contains one quarter of all base-k strings of length m: that is, exactly 2^{m-2} of them. In total, it contains 2^{m-1} of all strings of length up to m. For that reason, $d(K) = \frac{1}{4}$.

If the density $d(L)$ is defined, then, by Lemma 11, it should be equal to $\frac{1}{2}$. Then, to see that it is not defined, that is, that the following limit does not exist, it is sufficient to demonstrate a subsequence that converges to a different value.

$$d(L) = \lim_{n \to \infty} \frac{|L \cap \{\varepsilon, a, a^2, \ldots, a^{n-1}\}|}{n}.$$

The desired subsequence is $\{2 \cdot 16^{m-1} + 16^{m-2}\}_{m \geqslant 2}$. The number of strings in $L \cap \{\varepsilon, a, a^2, \ldots, a^{2 \cdot 16^{m-1} + 16^{m-2}}\}$ is estimated in the following.

Three blocks of strings are enumerated separately. The first block contains all strings between ε and $a^{16^{m-1}-1}$. Concerning the fraction of such strings that belong to L, Claim 5 yields that this is the same as the fraction $\frac{|K \cap \{e,f\}^{<m}|}{2^{m-1}}$, which is $\frac{1}{2}$.

Strings in L that are in the subsequent block between $a^{16^{m-1}}$ and $a^{2 \cdot 16^{m-1}-1}$ have base-16 representations of the form $1w$, where $w \in \Sigma_{16}^*$ and $a^{(w)_{16}} \in L$. According to Theorem 2, $a^{(1w)_{16}} \in L$ if and only if $a^{(w)_{16}} \in L$, and for that reason, the set $L \cap \{a^{16^{m-1}}, \ldots, a^{2 \cdot 16^{m-1}-1}\}$ contains as many strings as $L \cap \{\varepsilon, \ldots, a^{16^{m-1}-1}\}$, that is, 2^{4m-5} out of $16^{m-1} = 2^{4m-4}$.

The situation is different in the next block of 16^{m-2} strings, in the range between $a^{2 \cdot 16^{m-1}}$ and $a^{2 \cdot 16^{m-1} + 16^{m-2} - 1}$. For any string $a^{(w)_{16}}$ in this range, its base-16 representation is of the form $w \in 20\Sigma_{16}^{m-2}$, and therefore $h(w) \in fe\{f, e\}^{m-2} \subseteq K$. Then, by Theorem 2, this string is in L. Hence, L contains 16^{m-2} out of 16^{m-2} of these strings.

Altogether, L contains $2 \cdot 2^{4m-5} + 2^{4m-8} = 17 \cdot 2^{4m-8}$ out of the first $2 \cdot 2^{4m-4} + 2^{4m-8} = 33 \cdot 2^{4m-8}$ strings, that is, a fraction of $\frac{17}{33}$, for every m. This forms a subsequence with a limit $\frac{17}{33}$, and therefore $d(L)$ does not exist. □

Even though the unary languages, defined according to Theorem 2, do not necessarily have their density defined, they turn out to be sufficient to establish several undecidability results for unambiguous conjunctive grammars, which are presented in the next section.

6. Decision problems

Already for ordinary grammars, many basic decision problems are undecidable, such as testing whether two given grammars describe the same language (the equivalence problem), or even testing whether a given grammar describes the language of all strings over the alphabet $\{a, b\}$. A few problems are known to be decidable: for instance, one can test in polynomial time whether a given ordinary grammar describes the empty set. In contrast, for conjunctive grammars, there is a uniform undecidability result: for every language L_0 described by some conjunctive grammar, testing whether a given conjunctive grammar describes L_0 is undecidable [11].

Turning to unambiguous subclasses, the decidability status of the equivalence problem for (ordinary) unambiguous grammars is among the major unsolved questions in formal language theory. At the same time, as proved by Semenov [31], testing whether a given unambiguous grammar describes a given regular language is decidable: this remarkable proof proceeds by reducing the decision problem to a statement of elementary analysis, and then using Tarski's algorithm to solve it.

This section establishes the undecidability of checking whether an unambiguous conjunctive grammar describes a fixed language, *for any fixed language*. The underlying idea is the same as in the earlier results for ambiguous conjunctive grammars [11]: the language of computation histories of a Turing machine (VALC) is represented by a trellis automaton, its alphabet is reinterpreted as an alphabet of digits, so that each computation history is associated to a number, and then the unary representations of these numbers are defined by a conjunctive grammar [11]. However, Theorems 1–2 proved in this

paper for the unambiguous case are more restricted than the known constructions of ambiguous conjunctive grammars [11], and the same undecidability methods require a careful re-implementation.

For a Turing machine T over an input alphabet Θ, its computations are represented as strings over an auxiliary alphabet Ω. For every $w \in L(T)$, let $C_T(w) \in \Omega^*$ denote some representation of the accepting computation of T on w. The following language over the alphabet Ω is known as the language of valid accepting computations of T.

$$\mathrm{VALC}(T) = \{C_T(w) \mid w \in \Theta^* \text{ and } C_T(w) \text{ is an accepting computation}\}$$

It is well-known that for a certain simple encoding $C_T : \Theta^* \to \Omega^*$, the language $\mathrm{VALC}(T)$ is an intersection of two languages described by linear grammars, and hence is recognized by a trellis automaton [22].

Consider the following first undecidability result for unambiguous conjunctive grammars, proved by embedding $\mathrm{VALC}(T)$ into a sparse unary language using Theorem 1.

Lemma 12. *It is undecidable to determine whether a given unambiguous conjunctive grammar over a unary alphabet describes the empty language $\varnothing$.*

Proof. Reduction from the Turing machine emptiness problem. Given a Turing machine T, consider the language $\mathrm{VALC}(T)$ defined over the alphabet Ω. Let d be the number of symbols in this alphabet, let $c = \max(5, d+2)$ and $k = 2c + 2d - 3$. Renaming the symbols of Ω to base-k digits $\{c, \ldots, c+d-1\}$, assume that $\mathrm{VALC}(T) \subseteq \Sigma_k^*$. Then, according to Theorem 1, one can construct an unambiguous conjunctive grammar that describes the language $L = a^{(1\mathrm{VALC}(T)1)_k}$. Since $L = \varnothing$ if and only if $\mathrm{VALC}(T) = \varnothing$, which holds if and only if $L(T) = \varnothing$, an algorithm solving the emptiness problem for unambiguous conjunctive grammars over a unary alphabet would test the emptiness of $L(T)$, which is known to be undecidable. □

By a similar argument, using Theorem 2 instead of Theorem 1 to embed $\mathrm{VALC}(T)$ in a dense unary language, one can prove the following second undecidability result.

Lemma 13. *It is undecidable to determine whether a given unambiguous conjunctive grammar over a unary alphabet describes the language of all strings a^*.*

Proof. The general line of the argument is the same as for Lemma 12, though several details are different.

Let T be a Turing machine, define the language $\mathrm{VALC}(T)$ over the alphabet Ω, and consider a code $g\colon \Omega \to \{e, f\}^*$, which maps each symbol to a string of a fixed length $\ell \geqslant 1$ that begins with f. For the image of $\mathrm{VALC}(T)$ under g, consider its complement and denote it by L.

$$L = \big(f\{e, f\}^* \setminus g(\mathrm{VALC}(T))\big) \cup \{\varepsilon\}$$

Since the linear conjunctive languages are closed under codes [25] and under all Boolean operations, the language L is linear conjunctive. Applying Theorem 2 with $k = 16$ to L yields an unambiguous conjunctive grammar G that describes the language $\{a^{(w)_k} \mid h(w) \in e^*L\}$, where $h\colon \Sigma_{16} \to \{e, f\}^*$ is some homomorphism. It is claimed that $L(G) = a^*$ if and only if $L(T) = \varnothing$.

If $L(T) = \varnothing$, then $\mathrm{VALC}(T) = \varnothing$, and hence $L = f\{e, f\}^* \cup \{\varepsilon\}$. Then $e^*L = \{e, f\}^*$, and therefore $L(G) = \{a^{(w)_k} \mid h(w) \in \{e, f\}^*\} = a^*$.

If $L(T) \neq \varnothing$, then there exists a string $x \in \mathrm{VALC}(T)$ representing an accepting computation of T. Therefore, its image $g(x) \in f\{e, f\}^*$ is not in L. Let $w \in \Sigma_{16}^* \setminus 0\Sigma_{16}^*$ be any string of digits with $h(w) \in e^*g(x)$. Then the string $a^{(w)_k}$ is missing from $L(G)$, and accordingly $L(G) \neq a^*$. □

Knowing the undecidability of both the equality to $\varnothing$ (Lemma 12) and the equality to a^* (Lemma 13), one can extend these results to the problem of equality to any fixed language, as follows.

Theorem 3. *For every alphabet Σ and for every language $L_0 \subseteq \Sigma^*$ described by an unambiguous conjunctive grammar, it is undecidable whether a given unambiguous conjunctive grammar describes L_0.*

Proof. The plan is to reduce the problem of equality to $\varnothing$ to the equality problem for any finite language L_0, whereas equality to a^* shall be reduced to equality to any infinite L_0. Accordingly, the proof splits into two cases, depending on whether L_0 is finite or infinite.

Let L_0 be **finite**, and suppose, for the sake of contradiction, that it is decidable whether a given unambiguous conjunctive grammar describes the language L_0. It is claimed that one can then decide whether a given unambiguous conjunctive grammar G describes the empty language. Let ℓ_0 be the length of the longest string in L_0 and choose any string $w_0 \in \Sigma^*$ with $|w_0| > \ell_0$. Construct a grammar G' that describes $L_0 \cup w_0 \cdot L(G)$. Then $L(G') = L_0$ if $L(G) = \varnothing$, and if $L(G) \neq \varnothing$, then $L(G')$ is a proper superset of L_0. Thus, any algorithm for testing whether $L(G')$ is equal to L_0 can decide the emptiness of $L(G)$, contradicting Lemma 12.

If L_0 is **infinite**, then, again, suppose that one can decide whether a grammar describes L_0. Then, an algorithm for testing whether a given unambiguous conjunctive grammar G over a one-symbol alphabet describes the language a^* is obtained as follows. For a given G, the algorithm first constructs a related grammar $\widetilde{G} = (\Sigma, N, R, S)$ over the alphabet Σ, which, for every string a^n in $L(G)$, defines all strings of length n over Σ. This grammar describes the language $L(\widetilde{G}) = \{w \mid w \in \Sigma^*,\ a^{|w|} \in L(G)\}$. Let $G_0 = (\Sigma, N_0, R_0, S_0)$ be a fixed unambiguous conjunctive grammar that describes L_0. Construct a new grammar G' over the alphabet Σ, with the following rules.

$$
\begin{aligned}
S' &\to A \,\&\, S_0 \\
A &\to bA \,\&\, bS && \text{(for all } b \in \Sigma\text{)} \\
A &\to \varepsilon \\
S_0 &\to \ldots && \text{(the rules describing } L_0\text{)} \\
S &\to \ldots && \text{(the rules describing } \{w \mid w \in \Sigma^*,\ a^{|w|} \in L(G)\}\text{)}
\end{aligned}
$$

If $L(G) = a^*$, then $L(\widetilde{G}) = \Sigma^*$, and hence the rules for A in G' describe the language Σ^*. Therefore, $L(G') = L_0$.

Let $L(G) \neq a^*$. Then, for some number $\ell \geqslant 0$, none of the strings in Σ^ℓ are in $L_{G'}(S)$, and, accordingly, none of the strings of length $\ell + 1$ are in $L_{G'}(A)$. The latter implies that all strings in $L_{G'}(A)$ are of length at most ℓ, and therefore, $L(G')$ is finite. Since L_0 is infinite by assumption, it follows that $L(G') \neq L_0$.

Now, if there existed an algorithm for testing whether an unambiguous conjunctive grammar describes L_0, this algorithm would answer the question of whether $L(G)$ is a^*, which is undecidable by Lemma 13. □

7. Growth rate

Another standard consequence of being able to represent the language of computational histories, is that one can construct an unambiguous conjunctive grammar for an arbitrarily fast growing language. Let the *growth rate function* $g_L : \mathbb{N} \to \mathbb{N}$ of a unary language $L = \{a^{n_1}, a^{n_2}, \ldots\}$, with $n_1 < n_2 < \ldots$, be defined by $g_L(i) = n_i$ for each i. For example, regular unary languages have their growth rate functions bounded by linear functions, whereas languages in Examples 1 and 2 have exponential growth rate.

Theorem 4 *(cf. Jeż and Okhotin [11, Thm. 6]). Let $n_1, n_2, \ldots$ be an increasing recursively enumerable sequence of natural numbers, and let $f(i) = n_i$. Then there exists such a sparse unary language described by an unambiguous conjunctive grammar, that its growth rate function is greater than f at any point.*

Sketch of a proof. The theorem is proved exactly as in the cited paper, taking a Turing machine enumerating the sequence, constructing a trellis automaton for the language of its computation histories, and then applying Theorem 1 as in the proof of Lemma 12. □

The corresponding theorem for the general case of conjunctive grammars [11, Thm. 6] also includes the second case on representing a dense unary language, with the growth rate function of its complement being greater than f at any point. The methods of this paper are insufficient to construct any such examples, because the growth rate of every infinite language defined in Theorem 2 cannot be faster than exponential, and the same applies to growth rates of their complements.

Proposition 1. *Let K be a linear conjunctive language over a two-symbol alphabet $\Gamma = \{e, f\}$, which does not contain any strings beginning with e, and let $L \subseteq a^+$ be the encoding of K defined in Theorem 2. Then the growth rate of its complement, $\overline{L}$, is at most exponential.*

Proof. Suppose that $K \neq f\{e, f\}^*$, as otherwise $e^*K = \{e, f\}^*$, and then $L = a^+$ by Theorem 2.

Let $x \in f\{e, f\}^*$ be the shortest string not in K. For any number $\ell \geqslant 0$, the string $e^\ell x$ has at least one pre-image with respect to h; let $w_\ell \in \Sigma_k^{\ell+|x|}$, with $h(w_\ell) = e^\ell x$, be any of these pre-images. Then, $e^\ell x \notin e^*K$, and so $a^{(w_\ell)_k} \notin L$. The lengths of strings $|w_1|, |w_2|, \ldots$ form a linear sequence, and therefore the corresponding numbers $(w_1)_k, (w_2)_k, \ldots$ form a sequence with exponential growth rate. Any further strings in $\overline{L}$ besides $w_1, w_2, \ldots$ may only slow down its growth rate. □

It remains an open problem whether any unambiguous conjunctive grammar over a unary alphabet can describe a language with its complement growing faster than exponentially.

8. Conclusion

The expressive power of unambiguous conjunctive grammars over a unary alphabet has been developed up to the point of simulating a cellular automaton in a "sparse" unary language (Theorem 1), as well as in a variety of "non-sparse" unary

languages, with some "dense languages" among them (Theorem 2). Although these representations are not as general as those constructed earlier for ambiguous conjunctive grammars over the unary alphabet [11], they were sufficient to establish uniform undecidability results for the problem of testing equivalence to a fixed language (Theorem 3). The results of this paper have already been used to investigate the properties of unambiguous conjunctive grammars with two nonterminal symbols [14].

Another possible application of these constructions is to establish the complexity of the *compressed membership problem* [19, Sect. 9] for unambiguous conjunctive grammars. For ambiguous conjunctive grammars over a unary alphabet, it is known to be EXPTIME-complete [12,13], with the hardness proved by presenting a unary language of the form $a^{(L)_k}$, for an EXPTIME-complete language L [12]. It might be possible to re-implement this argument, this time using an unambiguous conjunctive grammar.

The paper leaves the following key question unsettled: do there exist any *inherently ambiguous* conjunctive languages, that is, those that can be described only by ambiguous grammars? In particular, can any such examples be found in the domain of unary languages? Though, at the first glance, it seemed that there cannot be any unambiguous conjunctive grammars for unary languages of high density, Theorems 1–2 show that languages of various density can be represented. Then, the question is, are there any properties of unary languages that would rule out their representation by unambiguous conjunctive grammars, and what kind of properties these could be? At the moment, there are still no examples of unambiguous conjunctive grammars over a unary alphabet with fast-growing complements: could this be an essential limitation of the grammars themselves, or merely of the techniques devised by the authors?

This also reminds of the main research problem for conjunctive grammars, that of finding any non-representable languages [27]. No methods for proving such results are known, besides trivial arguments based on computational complexity upper bounds. Would it be possible at least to separate unary unambiguous conjunctive grammars from the deterministic linear space?

Acknowledgements

The work of the first author was supported by National Science Centre, Poland project number 2014/15/B/ST6/00615. The second author did most of the work on this paper during his appointment at the University of Turku (Finland), and, at the time, his research was supported by the Academy of Finland under Grant 257857.

The authors are grateful to an anonymous referee for pointing out some problems in the initial submission, which prompted the authors to include Lemma 11 and Example 4, as well as to correct Theorem 4. The authors would also like to thank another referee from DLT 2013 for careful reading and, in particular, for noticing several small mistakes in the arguments.

References

[1] T. Aizikowitz, M. Kaminski, LR(0) conjunctive grammars and deterministic synchronized alternating pushdown automata, J. Comput. System Sci. 82 (8) (2016) 1329–1359.
[2] J.-P. Allouche, J. Shallit, Automatic Sequences: Theory, Applications, Generalizations, Cambridge University Press, 2003.
[3] J. Autebert, J. Berstel, L. Boasson, Context-free languages and pushdown automata, in: Rozenberg, Salomaa (Eds.), Handbook of Formal Languages, vol. 1, Springer-Verlag, 1997, pp. 111–174.
[4] M. Barash, A. Okhotin, An extension of context-free grammars with one-sided context specifications, Inform. and Comput. 237 (2014) 268–293.
[5] J. Berstel, Sur la densité asymptotique des languages formels, in: Automata, Languages and Programming, ICALP, 1973, pp. 345–358.
[6] K. Čulík II, J. Gruska, A. Salomaa, Systolic trellis automata, I, Int. J. Comput. Math. 15 (1984) 195–212.
[7] C. Dyer, One-way bounded cellular automata, Inf. Control 44 (3) (1980) 261–281.
[8] P. Enflo, A. Granville, J. Shallit, S. Yu, On sparse languages L such that $LL = \Sigma^*$, Discrete Appl. Math. 52 (1994) 275–285.
[9] O.H. Ibarra, S.M. Kim, Characterizations and computational complexity of systolic trellis automata, Theoret. Comput. Sci. 29 (1984) 123–153.
[10] A. Jeż, Conjunctive grammars can generate non-regular unary languages, Internat. J. Found. Comput. Sci. 19 (3) (2008) 597–615.
[11] A. Jeż, A. Okhotin, Conjunctive grammars over a unary alphabet: undecidability and unbounded growth, Theory Comput. Syst. 46 (1) (2010) 27–58.
[12] A. Jeż, A. Okhotin, Complexity of equations over sets of natural numbers, Theory Comput. Syst. 48 (2) (2011) 319–342.
[13] A. Jeż, A. Okhotin, One-nonterminal conjunctive grammars over a unary alphabet, Theory Comput. Syst. 49 (2) (2011) 319–342.
[14] A. Jeż, A. Okhotin, On the number of nonterminal symbols in unambiguous conjunctive grammars, in: Descriptional Complexity of Formal Systems, DCFS 2012, Braga, Portugal, 23–25 July 2012, in: LNCS, vol. 7386, pp. 183–195.
[15] A. Jeż, A. Okhotin, Computational completeness of equations over sets of natural numbers, Inform. and Comput. 237 (2014) 56–94.
[16] V. Kountouriotis, Ch. Nomikos, P. Rondogiannis, Well-founded semantics for Boolean grammars, Inform. and Comput. 207 (9) (2009) 945–967.
[17] M. Kunc, What do we know about language equations?, in: Developments in Language Theory, DLT 2007, Turku, Finland, 3–6 July 2007, in: LNCS, vol. 4588, pp. 23–27.
[18] T. Lehtinen, On equations $X + A = B$ and $(X + X) + C = (X - X) + D$ over sets of numbers, in: Mathematical Foundations of Computer Science, MFCS 2012, Bratislava, Slovakia, 26–31 August 2012, in: LNCS, vol. 7464, pp. 615–629.
[19] M. Lohrey, Algorithmics on SLP-compressed strings: a survey, Groups Complex. Cryptol. 4 (2) (2012) 241–299.
[20] P. McKenzie, K.W. Wagner, The complexity of membership problems for circuits over sets of natural numbers, Comput. Complexity 16 (2007) 211–244.
[21] A. Okhotin, Conjunctive grammars, J. Autom. Lang. Comb. 6 (4) (2001) 519–535.
[22] A. Okhotin, On the equivalence of linear conjunctive grammars to trellis automata, RAIRO Inform. Théor. Appl. 38 (1) (2004) 69–88.
[23] A. Okhotin, Boolean grammars, Inform. and Comput. 194 (1) (2004) 19–48.
[24] A. Okhotin, Unambiguous Boolean grammars, Inform. and Comput. 206 (2008) 1234–1247.
[25] A. Okhotin, Homomorphisms preserving linear conjunctive languages, J. Autom. Lang. Comb. 13 (3–4) (2008) 299–305.
[26] A. Okhotin, Decision problems for language equations, J. Comput. System Sci. 76 (3–4) (2010) 251–266.
[27] A. Okhotin, Conjunctive and Boolean grammars: the true general case of the context-free grammars, Comput. Sci. Rev. 9 (2013) 27–59.

[28] A. Okhotin, Parsing by matrix multiplication generalized to Boolean grammars, Theoret. Comput. Sci. 516 (2014) 101–120.
[29] A. Okhotin, C. Reitwießner, Parsing Boolean grammars over a one-letter alphabet using online convolution, Theoret. Comput. Sci. 457 (2012) 149–157.
[30] A. Okhotin, P. Rondogiannis, On the expressive power of univariate equations over sets of natural numbers, Inform. and Comput. 212 (2012) 1–14.
[31] A.L. Semenov, Algorithmic problems for power series and for context-free grammars, Dokl. Akad. Nauk SSSR 212 (1973) 50–52.
[32] V. Terrier, On real-time one-way cellular array, Theoret. Comput. Sci. 141 (1995) 331–335.
[33] V. Terrier, Recognition of linear-slender context-free languages by real time one-way cellular automata, in: Cellular Automata and Discrete Complex Systems, AUTOMATA 2015, Turku, Finland, 8–10 June 2015, in: LNCS, vol. 9099, pp. 251–262.