Artur Jeż | Alexander Okhotin

Unambiguous conjunctive grammars over a one-letter alphabet

Turku Centre *for* Computer Science

TUCS Technical Report
No 1043, April 2012

Unambiguous conjunctive grammars over a one-letter alphabet

Artur Jeż
Institute of Computer Science,
University of Wrocław,
ul. Joliot-Curie 15, 50–383 Wroclaw, Poland
aje@cs.uni.wroc.pl
Alexander Okhotin
Department of Mathematics, University of Turku, *and*
Turku Centre for Computer Science
Turku FI–20014, Finland
alexander.okhotin@utu.fi

TUCS Technical Report
No 1043, April 2012

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 $k \geqslant 11$ and for every one-way real-time cellular automaton operating over the alphabet of base-k digits $\left\{\lceil\frac{k+9}{4}\rceil, \ldots, \lfloor\frac{k+1}{2}\rfloor\right\}$, the language of all strings a^n with the base-k notation of the form $\mathtt{1}w\mathtt{1}$, where w is accepted by the automaton, is generated 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.

Keywords: Keywords: conjunctive grammars, ambiguity, language equations, unary languages.

TUCS Laboratory
FUNDIM, Fundamentals of Computing and Discrete Mathematics

1 Introduction

Conjunctive grammars, introduced by Okhotin [9], are an extension of the context-free grammars, which allows the use of a conjunction operation in any rules, in addition to the implicit disjunction already present in context-free grammars. These grammars maintain the main principle behind the context-free grammars—that of inductive definition of the membership of strings in the language—inherit their parsing techniques and subcubic time complexity [13], and augment their expressive power in a meaningful way.

Conjunctive grammars over a one-letter alphabet $\Sigma = \{a\}$ were proved non-trivial by Jeż [4], who constructed a grammar for the language $\{a^{4^n} \mid n \geqslant 0\}$. Subsequent work on such grammars revealed their high expressive power and a number of undecidable properties [5]. Testing whether a given string a^n is generated by a grammar G can be done in time $|G| \cdot n(\log n)^3 \cdot 2^{O(\log^* n)}$ [14], and if n is given in binary notation, this problem is EXPTIME-complete already for a fixed grammar G [6]. These results also had impact on the study of language equations [8, 12], essential to understanding their computational completeness over a unary alphabet [7].

Unambiguous conjunctive grammars [11] are an important subclass of conjunctive grammars defined by analogy with unambiguous context-free grammars, and representing grammars that assign a unique syntactic structure to every well-formed sentence. Little is known about their properties, besides a parsing algorithm with $|G| \cdot O(n^2)$ running time, where n is the length of the input [11]; for a unary alphabet, the running time can be improved to $|G| \cdot n(\log n)^2 \cdot 2^{O(\log^* n)}$ [14]. However, all the known results on the expressive power of conjunctive grammars over a unary alphabet [4, 5, 6, 15] rely upon ambiguous grammars, and it is not even known whether unambiguous grammars can generate anything non-regular.

This paper sets off by presenting the first example of an unambiguous conjunctive grammar that generates a nonregular unary language. This is the same language $\{a^{4^n} \mid n \geqslant 0\}$, yet the grammar generating it, given in Section 3, is more involved than the known ambiguous grammar. 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 [3, 10, 17] over an input alphabet $\Sigma_k = \{\mathtt{0}, \mathtt{1}, \ldots, k-1\}$ 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 [5] essentially uses concatenations of highly populated sets, and hence the resulting grammars are ambiguous. 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 its subset of size at most around $\frac{k}{4}$. This restriction allows simulating the automaton, so that all concatenations in the grammar remain unambiguous.

The simulation of a cellular automaton presented in Section 5 produces languages that grow exponentially fast; these languages have *density 0*, in the sense that the fraction of strings of length up to n 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. However, the next Section 6 nevertheless succeeds in representing non-regular unary languages of density 1 (that is, containing almost all strings) by an unambiguous conjunctive grammar, and extends the simulation of cellular automata to this kind of unary languages.

These constructions yield undecidability results for unambiguous conjunctive grammars, presented in the last Section 7. For every fixed language L_0 generated by some unambiguous conjunctive grammar, it is proved that testing whether a given unambiguous conjunctive grammar generates L_0 is undecidable. This is compared to the known decidable properties of the unambiguous case of standard context-free grammars [16].

2 Conjunctive grammars and ambiguity

Conjunctive grammars generalize context-free grammars by allowing an explicit conjunction operation in the rules. This is more of a variant of the definition of the context-free grammars than something entirely new. In particular, it leaves the context-freeness intact, and only extends the set of logical connectives used to combine syntactical conditions.

Definition 1 (Okhotin [9])**.** *A conjunctive grammar is a quadruple $G = (\Sigma, N, P, S)$, in which Σ and N are disjoint finite non-empty sets of terminal and nonterminal symbols respectively; P is a finite set of grammar rules, each of the form*

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

while $S \in N$ is a nonterminal designated as the start symbol.

A rule (*) informally means that every string generated by each *conjunct* α_i is therefore generated by A. This understanding may be formalized either by term rewriting [9], or, equivalently, by a system of language equations. According to the definition by language equations, conjunction is interpreted as intersection of languages as follows.

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

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

where 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 $L(G) := L_G(S)$.

To see that such a system always has a least solution, it is sufficient to know that the operations in its right-hand side (in this case, union, intersection and concatenation) are monotone and continuous with respect to the partial ordering of componentwise inclusion on vectors of languages. This is known to be the case. Therefore, the right-hand side of the equation is a monotone and continuous operator φ on vectors of languages, and the least solution is given by the least upper bound of the iterative application of this operator, beginning with a vector of empty languages:

$$\bigsqcup_{k \geqslant 0} \varphi^k(\varnothing, \ldots, \varnothing)$$

(the notation $\bigsqcup$ refers to a componentwise union of vectors).

An equivalent definition of conjunctive grammars is given via *term rewriting*, which generalizes the string rewriting used by Chomsky to define context-free grammars.

Definition 3 ([9])**.** *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 $L_G(\varphi) = \{w \,|\, w \in \Sigma^*,\ \varphi \Longrightarrow^* w\}$. The language generated by the grammar is $L(G) = L_G(S) = \{w \,|\, w \in \Sigma^*,\ S \Longrightarrow^* w\}$.

Examples of conjunctive grammars for such non-context-free languages as $\{a^n b^n c^n \,|\, n \geqslant 0\}$ and $\{wcw \,|\, w \in \{a, b\}^*\}$ can be found in the literature [9].

This paper concentrates on a subclass of conjunctive grammars defined by analogy with unambiguous context-free grammars.

Definition 4 ([11])**.** *A conjunctive grammar $G = (\Sigma, N, P, S)$ is said to be*

I. with unambiguous choice of a rule*, 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. with unambiguous concatenation, *if for every conjunct $\alpha = s_1 \dots s_\ell$ and for every string w there exists at most one factorization $w = u_1 \dots u_\ell$ with $u_i \in L_G(s_i)$ for all i.*

If both conditions are satisfied, the grammar is called unambiguous. *A conjunctive language L can be called* inherently ambiguous *if every conjunctive grammar generating it is ambiguous.*

3 Representing powers of four

Consider the following grammar generating 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 these grammars since this example, it still remains the smallest and the easiest to understand.

Example 1 (Jeż [4])**.** *The conjunctive grammar*

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

with the start symbol A_1 generates the language $L(G) = \{a^{4^n} \mid n \geqslant 0\}$. In particular, $L_G(A_i) = \{a^{i \cdot 4^n} \mid n \geqslant 0\}$ for $i = 1, 2, 3, 6$.

The grammar is best explained in terms of base-4 notation of the lengths of the strings. Let $\Sigma_4 = \{\mathtt{0}, \mathtt{1}, \mathtt{2}, \mathtt{3}\}$ be the alphabet of base-4 digits, and for every $w \in \Sigma_4^*$, 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 generated by the nonterminals of the above grammar are $a^{(\mathtt{10}^*)_4}$, $a^{(\mathtt{20}^*)_4}$, $a^{(\mathtt{30}^*)_4}$ and $a^{(\mathtt{120}^*)_4}$.

Consider the system of language equations corresponding to the grammar: the equation for A_1 is

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

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

$$\begin{aligned} &a^{(\mathtt{10}^*)_4}a^{(\mathtt{30}^*)_4} \cap a^{(\mathtt{20}^*)_4}a^{(\mathtt{20}^*)_4} = \\ &\qquad = \left(a^{(\mathtt{10}^+)_4} \cup a^{(\mathtt{10}^*\mathtt{30}^*)_4} \cup a^{(\mathtt{30}^*\mathtt{10}^*)_4}\right) \cap \left(a^{(\mathtt{10}^+)_4} \cup a^{(\mathtt{20}^*\mathtt{20}^*)_4}\right) = a^{(\mathtt{10}^+)_4}. \end{aligned}$$

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, one can prove that the system has a unique ε-free solution [1, Thm. 2.3].

The grammar in Example 1 is ambiguous, because of the concatenations A_2A_2, A_1A_1, A_6A_6 and A_3A_3: 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 using such concatenations, though that requires representing 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 conjunctive grammar*

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

is unambiguous and generates the language $\{a^{4^n} \mid n \geqslant 0\}$. *Each nonterminal* A_i *generates 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

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

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 $a^{4^{m-1}}$ to $a^{3\cdot 4^{n-1}}$; 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, while terminating rules generate shorter strings than the non-terminating rule for the same nonterminal.

4 Representing powers of k

In this section, unambiguous grammars for the languages $L_k = \{a^{k^n} \mid n \geqslant 1\}$, with $k \geqslant 9$, are constructed. The nonterminal symbols of each grammar generate the languages $\{a^{c\cdot k^n} \mid n \geqslant 0\}$, for all $c \in \{k, k+1, \ldots, k^2-1\}$. In other words, these are the languages $a^{(ij0^*)_k}$, for some base-k digits i, j with $i \neq 0$. All these languages are thus proved to be unambiguous conjunctive.

Lemma 1. *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 start symbol $A_{1,0}$ generates the language $a^{(k0^+)_k}$:*

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

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

First, it has to be proved that the nonterminals of the constructed grammar indeed generate the desired languages.

Claim 1. *The system of language equations corresponding to the grammar in Lemma 1 has a unique solution in ε-free languages, with $A_{i,j} = a^{(ij0^*)_k}$ for each i,j. Furthermore, the grammar is with unambiguous choice of rules.*

Proof. The intended solution $A_{i,j} = a^{(ij0^*)_k}$ is ε-free, and one can prove by a standard argument [1] 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 $A_{i,j} = a^{(ij0^*)_k}$.

To show that the choice of a rule in the grammar is unambiguous, note, that each of the nonterminals has only two rules, one terminating and one non-terminating. It is later shown, that the language generated by the non-terminating rule consists of strings strictly longer than the string generated by the terminating rule. This will be enough to conclude that the grammar has unambiguous choice of a rule.

Getting back to substituting the intended solution into the equations, note, that it 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:

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

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

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

To verify an equation of the first type (1a), with $j < \frac{k}{3}+2$, one should check that

$$(1j0^+)_k = \big(((k-1)0^+)_k + ((j+1)0^+)_k\big) \cap \big(((k-2)0^+)_k + ((j+2)0^+)_k\big). \quad (2)$$

The two intersected sets of numbers are calculated separately as follows:

$$\begin{aligned}
&((k-1)0^+)_k + ((j+1)0^+)_k = \\
&\qquad ((k-1)0^*(j+1)0^+)_k \cup (1j0^+)_k \cup ((j+1)0^*(k-1)0^+)_k, \\
&((k-2)0^+)_k + ((j+2)0^+)_k = \\
&\qquad ((k-2)0^*(j+2)0^+)_k \cup (1j0^+)_k \cup ((j+2)0^*(k-2)0^+)_k.
\end{aligned}$$

Their intersection obviously contains $(1j0^+)_k$, and it remains to see that no other numbers get into it, that is, that

$$\begin{aligned}
&((k-1)0^*(j+1)0^+)_k \cap ((k-2)0^*(j+2)0^+)_k = \varnothing, \\
&((k-1)0^*(j+1)0^+)_k \cap ((j+2)0^*(k-2)0^+)_k = \varnothing, \\
&((j+1)0^*(k-1)0^+)_k \cap ((k-2)0^*(j+2)0^+)_k = \varnothing, \\
&((j+1)0^*(k-1)0^+)_k \cap ((j+2)0^*(k-2)0^+)_k = \varnothing.
\end{aligned}$$

Each of the sets being intersected has all elements with the same leading digit, and those digits are different for each pair of 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 (1a) holds, and the subsequent union with $(1j)_k$ on the right-hand side of (1a) produces the set $(1j0^*)_k$, and the equation holds true.

Similar calculations are performed for each equation (1b) with $i \geqslant 2$ and $j < \frac{k}{3}+2$. It is to be shown that

$$(ij0^+)_k = \big(((i-1)(k-1)0^*)_k + ((j+1)0^+)_k\big) \cap \big(((i-1)(k-2)0^*)_k + ((j+2)0^+)_k\big). \tag{3}$$

Each of the two sums on the right-hand side of (3) is now evaluated. Consider the first sum. The added sets contain numbers with digits $i-1$ and $j+1$, which can be added to each other. Thus the value of the sum depends on whether $i+j < k$, in which case a digit $i+j$ is obtained, or $i+j \geqslant k$, in which case a digit $i+j-k$ with a carry is obtained. To be more precise, if $i+j < k$, then

$$\begin{aligned}
&((i-1)(k-1)0^*)_k + ((j+1)0^+)_k = \\
&\qquad ((i-1)(k-1)0^*(j+1)0^+)_k \cup (ij0^+)_k \cup \\
&\qquad ((i+j)(k-1)0^*)_k \cup ((j+1)0^*(i-1)(k-1)0^*)_k,
\end{aligned} \tag{4a}$$

and if $i+j \geqslant k$, then

$$\begin{aligned}
&((i-1)(k-1)0^*)_k + ((j+1)0^+)_k = \\
&\qquad ((i-1)(k-1)0^*(j+1)0^+)_k \cup (ij0^+)_k \cup \\
&\qquad (1(i+j-k)(k-1)0^*)_k \cup ((j+1)0^*(i-1)(k-1)0^*)_k.
\end{aligned} \tag{4b}$$

Similarly, the value of the second sum depends on whether $i+j+1<k$ or not. That is, if $i+j+1<k$, then

$$\begin{aligned}(&(i-1)(k-2)\mathtt{0}^*)_k+((j+2)\mathtt{0}^+)_k=\\ &\qquad((i-1)(k-2)\mathtt{0}^*(j+2)\mathtt{0}^+)_k\cup(ij\mathtt{0}^+)_k\cup \qquad (5a)\\ &\qquad((i+j+1)(k-2)\mathtt{0}^*)_k\cup((j+2)\mathtt{0}^*(i-1)(k-2)\mathtt{0}^*)_k,\end{aligned}$$

and if $i+j+1\geqslant k$, then

$$\begin{aligned}(&(i-1)(k-2)\mathtt{0}^*)_k+((j+2)\mathtt{0}^+)_k=\\ &\qquad((i-1)(k-2)\mathtt{0}^*(j+2)\mathtt{0}^+)_k\cup(ij\mathtt{0}^+)_k\cup \qquad (5b)\\ &\qquad(\mathtt{1}(i+j-k+1)(k-2)\mathtt{0}^*)_k\cup((j+2)\mathtt{0}^*(i-1)(k-2)\mathtt{0}^*)_k.\end{aligned}$$

Each right-hand side of (4a)–(5b) contains $(ij\mathtt{0}^*)_k$, thus also their intersection on the right-hand side of (3) contains $(ij\mathtt{0}^*)_k$, and it remains to show that the intersection of any other component of (4a) or (4b) with any other component of (5a) or (5b) is empty. This time, the numbers are distinguished by their *last (least significant) non-zero digits*: all numbers in (4a) and (4b) (except for numbers from $(ij\mathtt{0}^*)_k$) have $j+1$ or $k-1$, while all numbers in (5a) and (5b) (other than the numbers from $(ij\mathtt{0}^*)_k$) have $j+2$ or $k-2$. As in the previous case, these numbers are distinct, which proves (3), and thus the equation (1b) is checked.

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

$$(ij\mathtt{0}^+)_k=\big((i(j-1)\mathtt{0}^*)_k+(\mathtt{10}^+)_k\big)\cap\big((i(j-2)\mathtt{0}^*)_k+(\mathtt{20}^+)_k\big). \qquad (6)$$

The value of the first sum on the right-hand side of (6) is different for $i<k-1$ and for $i=k-1$. In the first case it is equal to

$$\begin{aligned}&(i(j-1)\mathtt{0}^*)_k+(\mathtt{10}^+)_k=\\ &(i(j-1)\mathtt{0}^*\mathtt{10}^+)_k\cup(ij\mathtt{0}^+)_k\cup((i+1)(j-1)\mathtt{0}^*)_k\cup(\mathtt{10}^*i(j-1)\mathtt{0}^*)_k, \qquad (7a)\end{aligned}$$

while for $i=k-1$ the sum equals

$$\begin{aligned}&(i(j-1)\mathtt{0}^*)_k+(\mathtt{10}^+)_k=\\ &(i(j-1)\mathtt{0}^*\mathtt{10}^+)_k\cup(ij\mathtt{0}^+)_k\cup(\mathtt{1}(i+1-k)(j-1)\mathtt{0}^*)_k\cup(\mathtt{10}^*i(j-1)\mathtt{0}^*)_k. \qquad (7b)\end{aligned}$$

The second sum on the right-hand side of (6) is similarly expanded. For $i<k-2$ it is represented as

$$\begin{aligned}&(i(j-2)\mathtt{0}^*)_k+(\mathtt{20}^+)_k=\\ &(i(j-2)\mathtt{0}^*\mathtt{20}^+)_k\cup(ij\mathtt{0}^+)_k\cup((i+2)(j-2)\mathtt{0}^*)_k\cup(\mathtt{20}^*i(j-2)\mathtt{0}^*)_k, \qquad (8a)\end{aligned}$$

and for $i \geqslant k-2$ as

$$\begin{aligned}&(i(j-2)\mathtt{0}^*)_k + (\mathtt{2}0^+)_k = \\ &(i(j-2)\mathtt{0}^*\mathtt{2}0^+)_k \cup (ij\mathtt{0}^+)_k \cup (\mathtt{1}(i+2-k)(j-2)\mathtt{0}^*)_k \cup (\mathtt{2}0^* i(j-2)\mathtt{0}^*)_k.\end{aligned} \tag{8b}$$

Both sums (regardless of the value of i) contain the subset $(ij\mathtt{0}^+)_k$, which hence belongs to the intersection. To see that there is nothing else in the intersection, one has to show that the intersection of any other set on the right-hand side of (7a) or (7b) with any set from the right-hand side of (8a) or (8b) is empty. These sets are again distinguished by their last non-zero digit: the numbers from sets on the right-hand side of (7a) or (7b) (other than $(ij\mathtt{0}^+)_k$) have either $\mathtt{1}$ or $j-1$ as the last non-zero digit, while the numbers from the sets on the right-hand side of (8a) or (8b) (other than $(ij\mathtt{0}^+)_k$) have either $\mathtt{2}$ or $j-2$. By the case assumption, $j \geqslant \frac{k}{3}+2 \geqslant 5$ and so it holds that $\mathtt{1} < \mathtt{2} < j-2 < j-1$. Therefore, the numbers appearing on the right-hand side of (7a) or (7b) are different than the numbers appearing on the right-hand side of (8a) or (8b). And so (6) holds.

It is left to show that the grammar has unambiguous choice of a rule. As already mentioned, this is done by comparing the lengths of the strings generated by the terminating and non-terminating rules. For the first case, that is of $i = 1$ and $j < \frac{k}{3}+2$, observe that the set $(\mathtt{1}j\mathtt{0}^+)_k$ describes the lengths of strings generated by the non-terminating rule, as demonstrated in (2); this is strictly larger than the length of the constant string $(\mathtt{1}j)_k$ in this case. Similar argument applies to the other cases, with (3) and (6) giving the lengths of the strings generated by the non-terminating rules in these cases. This shows that the choice of a rule in the grammar is unambiguous. □

In order to show that the constructed grammar has unambiguous concatenation, it is proved that the concatenation of a language $a^{(ij\mathtt{0}^*)_k}$ with a language $a^{(i'j'\mathtt{0}^*)_k}$ is *in most cases* unambiguous, and that none of the concatenations actually used in the grammar are among the few exceptions to this rule.

Claim 2. *Let $k \geqslant 2$, and consider any two different languages of the form $K = a^{(ij\mathtt{0}^*)_k}$ and $L = a^{(i'j'\mathtt{0}^*)_k}$, with $i, i' \in \Sigma_k \setminus \{\mathtt{0}\}$ 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 $a^{(ij\mathtt{0}^\ell)_k} a^{(i'j'\mathtt{0}^m)_k} = a^{(ij\mathtt{0}^{\ell'})_k} a^{(i'j'\mathtt{0}^{m'})_k}$, or, in other words, if

$$(k \cdot i + j)(k^\ell - k^{\ell'}) = (k \cdot i' + j')(k^{m'} - k^m) \tag{9}$$

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

It is first shown that $\ell = \ell'$ or $m = m'$ implies the claim: if $\ell = \ell'$, then the left-hand side of (9) is zero, and therefore its right-hand side is zero as well, which holds only if $m' = m$, as claimed. Similarly, $m = m'$ implies $\ell = \ell'$. Assume that $\ell \neq \ell'$ and $m \neq m'$; by the symmetry, it may be further assumed that $\ell > \ell'$, which shows that both sides of (9) are non-negative, hence $k^{m'} > k^m$ and it can be concluded that $m' > m$. Then (9) can be equally written as

$$(k \cdot i + j)k^{\ell'}(k^{\ell-\ell'} - 1) = (k \cdot i' + j')k^m(k^{m'-m} - 1). \tag{10}$$

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

Consider first the case, when both j and j' are nonzero. Then the left-hand side of (10) is divisible by $k^{\ell'}$, but not by $k^{\ell'+1}$, while the number on the right-hand side is divisible by k^m, but not by k^{m+1}. Therefore, $\ell' = m$ and the equation (10) is simplified to

$$(k \cdot i + j)(k^{\ell-\ell'} - 1) = (k \cdot i' + j')(k^{m'-m} - 1). \tag{11}$$

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 yields that $ki + j \neq ki' + j'$. Again by the symmetry it may be assumed that $k \cdot i + j > k \cdot i' + j'$. Then (11) implies that $(k^{m'-m} - 1) > (k^{\ell-\ell'} - 1)$, that is, $(k^{m'-m-1} - 1) \geqslant (k^{\ell-\ell'} - 1)$. This is enough to estimate both sides of (11) with contradictory values. The left-hand side of (11) is at most

$$\begin{aligned}(k \cdot i + j)(k^{\ell-\ell'} - 1) \leqslant (k \cdot (k-1) + k - 1)(k^{m'-m-1} - 1) < \\ < k^2(k^{m'-m-1} - 1) = k^{m'-m+1} - k^2,\end{aligned}$$

while the right-hand side of (11) is at least

$$\begin{aligned}(k \cdot i' + j')(k^{m'-m} - 1) \geqslant (k+1)(k^{m'-m} - 1) = \\ = k^{m'-m+1} - k + k^{m'-m} - 1 \geqslant k^{m'-m+1} - 1.\end{aligned}$$

Together those two estimations yield $k^{m'-m+1} - k^2 > k^{m'-m+1} - 1$, which is not possible, as $k > 1$. The obtained contradiction shows that $\ell = \ell'$ and $m = m'$, 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 (10) in this case is divisible by $k^{\ell'+1}$, but not by $k^{\ell'+2}$, and the number on the right-hand side is divisible by k^{m+1}, but not by k^{m+2}, which yields the equality $\ell' = m$. Then the equations (10) can be represented as

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

Since $\ell > \ell'$, the left-hand side is equal to $-i$ modulo k and, as $m' > m$, the right-hand side is equal 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 the following holds: $j = 0$ or $j' = 0$. By symmetry it may be assumed that $j \neq 0$ and $j' = 0$. Then the number on the left-hand side of (10) is divisible by $k^{\ell'}$, but not by $k^{\ell'+1}$ and the number on the right-hand side is divisible by k^{m+1}, but not by k^{m+2}. Thus $\ell' = m + 1$. Consider the original equality $a^{(ij0^{\ell})_k} a^{(i'0^{m+1})_k} = a^{(ij0^{\ell'})_k} a^{(i'0^{m'+1})_k}$, which implies that $(ij0^{\ell})_k + (i'0^{m+1})_k = (ij0^{\ell'})_k + (i'0^{m'+1})_k$. Consider the positional notations of numbers represented by both these sums. Since $\ell > \ell' = m + 1$, the k-positional of $(ij0^{\ell})_k + (i'0^{m+1})_k$ has three non-zero digits: i, j and i', which appear in this order. The analysis for $(ij0^{\ell'})_k + (i'0^{m'+1})_k$ is more complicated: it is already known that $m' + 1 > m + 1 = \ell'$, that is $m' + 1 \geqslant \ell' + 1$. In the following, it is distinguished, whether $m' + 1 > \ell' + 1$ or $m' + 1 = \ell' + 1$:

- If $m' + 1 > \ell' + 1$ then the number $(ij0^{\ell'})_k + (i'0^{m'+1})_k$ has three non-zero digits in its k-positional notation: i', i and j, which appear in this order. Recall, that the order of non-zero digits in k-positional notation of $(ij0^{\ell})_k + (i'0^{m+1})_k$ is i, j and i'. Since these two numbers are equal, $i = i'$, $j = i$ and $i' = j$, that is, $i = i' = j$. This is exactly the combination excluded explicitly by the assumptions of the lemma, contradiction.

- If $m' + 1 = \ell' + 1$, the i and i' in the positional notation of $(ij0^{\ell'})_k$ and $(i'0^{m'+1})_k$ are added to each other. If $i + i' < k$ then $(ij0^{\ell'})_k + (i'0^{m'+1})_k = ((i + i')j0^{\ell'})_k$, which has only two non-zero digits in its base-k positional notation. This contradicts the fact that $(ij0^{\ell})_k + (i'0^{m+1})_k$ has three non-zero digits in its k-positional notation. If $i + i' \geqslant k$ then $(ij0^{\ell'})_k + (i'0^{m'+1})_k = (1(i + i' - k)j0^{\ell'})_k$. The sum of digits in k-positional notation of this number is $i + i' + j - k$, while the sum of digits in k-positional notation of $(ij0^{\ell})_k + (i'0^{m+1})_k$ is $i + i' + j$, contradiction.

The obtained contradiction completes the proof. □

Now the proof of Lemma 1 is inferred from both Claims 1–2.

Proof of Lemma 1. By Claim 1, each nonterminal $A_{i,j}$ generates the language $a^{(ij0^*)_k}$. Furthermore, by the same claim, the choice of a rule in the grammar is unambiguous. By Claim 2 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. There are three types of rules in the grammar, let us consider them separately.

- rules for $A_{1,j}$ for $j < \frac{k}{3} + 2$. 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.

- rules for $i \geqslant 2$ 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 $i = j + 1 = k - 1$. 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 $A_{i,j-2}A_{2,0}$ is unambiguous.

Hence, concatenation in the grammar is unambiguous. □

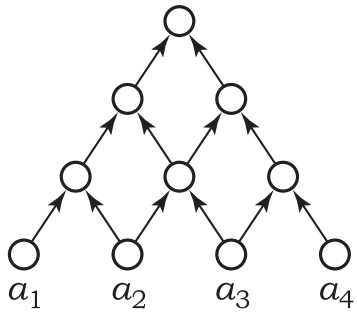
5 Simulating trellis automata

This section extends the main general method for constructing conjunctive grammars over a unary alphabet to the case of unambiguous conjunctive grammars. The overall idea is to simulate a *one-way real-time cellular automaton*, also known as a *trellis automaton*, operating on base-k representations of numbers, by a grammar generating unary representations of the same numbers.

A trellis automaton [2, 10], defined as a quintuple $(\Omega, Q, I, \delta, F)$, processes an input string of length $n \geqslant 1$ using a uniform array of $\frac{n(n+1)}{2}$ nodes, as presented in the figure below. 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 $I : \Omega \to Q$. The rest of the nodes compute the function $\delta : Q \times Q \to Q$ 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 $M = (\Omega, Q, I, \delta, F)$, in which:*

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



Extend δ *to a function* $\delta : Q^+ \to Q$ *by* $\delta(q) = q$ *and*

$$\delta(q_1, \ldots, q_n) = \delta(\delta(q_1, \ldots, q_{n-1}), \delta(q_2, \ldots, q_n)),$$

while I *is extended to a homomorphism* $I : \Omega^* \to Q^*$. *Let* $L_M(q) = \{w \mid \delta(I(w)) = q\}$ *and define* $L(M) = \bigcup_{q \in F} L_M(q)$.

Consider a trellis automaton with the input alphabet $\Omega = \Sigma_k = \{\mathtt{0}, \mathtt{1}, \ldots, k-1\}$ of base-k digits, and assume that it does not accept any strings beginning with $\mathtt{0}$. Then, every string of digits accepted by the automaton defines a certain number, and thus the automaton defines a set of non-negative integers. 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 A (Jeż, Okhotin [5])**.** *For every* $k \geqslant 2$ *and for every trellis automaton* M *over the alphabet* Σ_k, *with* $L(M) \cap \mathtt{0}\Sigma_k^* = \varnothing$, *there exists a conjunctive grammar generating the language* $\{a^{(w)_k} \mid w \in L(M)\}$.

The grammar simulates the computation of a trellis automaton $M = (\Omega, Q, I, \delta, F)$ using the nonterminal symbols A_q with $q \in Q$, which generate the languages $L_G(A_q) = \{a^{(\mathtt{1}w\mathtt{10}^\ell)_k} \mid \delta(w) = q,\ \ell \geqslant 0\}$, so that each string of digits $w \in \Sigma_k^*$ is represented in unary notation by the strings $a^{(\mathtt{1}w\mathtt{1})_k}$, $a^{(\mathtt{1}w\mathtt{10})_k}$, $a^{(\mathtt{1}w\mathtt{100})_k}$, etc.[1] The recursive dependence of L_q on each other simulates the definition of the trellis automata in terms of these unary encodings as follows. Consider a string $w = iuj$ with $i, j \in \Omega$ and $u \in \Omega^*$. Then $\delta(w) = q$ if and only if $q = \delta(q', q'')$, where $q' = \delta(iu)$ and $q'' = \delta(uj)$. In terms of unary encodings, $a^{(\mathtt{1}iuj\mathtt{10}^\ell)_k} \in L_q$ if and only if $a^{(\mathtt{1}iu\mathtt{10}^{\ell+1})_k} \in L_{q'}$ and $a^{(\mathtt{1}uj\mathtt{10}^\ell)_k} \in L_{q''}$ (note the additional $\mathtt{0}$ in the first case), and this dependence is expressed in the rules of a conjunctive grammar. The desired language $\{a^{(w)_k} | w \in L(M)\}$ is expressed through A_q with $q \in F$ by an additional series of rules.

However, the grammar produced by Theorem A is always ambiguous, and there is no general way of expressing the same languages $L_G(A_q) \subseteq a^*$ in an unambiguous grammar, for the following reason. The construction of

[1] For the more obvious encoding as $\{a^{(w)_k} \mid \delta(w) = q\}$, the constructions would not work [5].

unambiguous grammars, as presented in Lemma 1, relies on concatenating *exponentially growing* languages, and the sparsity of such languages in some cases allows their concatenation to be unambiguous. However, the languages $L_G(A_q)$, as defined above, may be denser than that, and their concatenation with any infinite language is ambiguous.

Thus, the first step of 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 assuming that Ω is a subset of the set of all digits.

Theorem 1. *For every trellis automaton over a d-letter input alphabet Ω, 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 generating the language $\{a^{(\mathtt{1}w\mathtt{1})_k} \mid w \in L\}$.*

If a base $k \geqslant 11$ is fixed, then, for instance, the following values of c and d are suitable: $c = \frac{k+9}{4}$, $d = \frac{k-3}{4}$. They induce the alphabet $\Omega = \{\frac{k+9}{4}, \ldots, \frac{k+1}{2}\}$.

The smallest values of c and k for an alphabet with $d = 2$ letters are $c = 5$ and $k = 11$, so that $\Omega = \{\mathtt{5}, \mathtt{6}\}$.

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

$$L_q^{s,s'} = \{a^{(sws'\mathtt{0}^\ell)_k} \mid \ell \geqslant 0,\ \delta(w) = q\}. \tag{12}$$

In this construction, the symbols s and s' surrounding the string w may be $\mathtt{1}$ or $\mathtt{2}$, whereas Theorem A uses only $\mathtt{1}$ for that purpose; this is an insignificant technical detail. The crucial difference with Theorem A is that each string w processed by M uses only digits from a small subset of Σ_k, which makes each $X_q^{s,s'}$ generate an exponentially growing language.

In terms of this encoding, computations of a trellis automaton are simulated as follows. In a trellis automaton, if a string $w = iuj$ with $i, j \in \Omega$ and $u \in \Omega^*$ has $\delta(iu) = q_1$, $\delta(uj) = q_2$ and $\delta(q_1, q_2) = q$, then $\delta(w) = q$. A grammar simulating this single step of computation has to generate the four strings $a^{(siujs'\mathtt{0}^\ell)_k}$ with $s, s' \in \{\mathtt{1}, \mathtt{2}\}$ by the corresponding nonterminals $X_q^{s,s'}$, using the eight strings $a^{(sius'\mathtt{0}^{\ell+1})_k} \in L_{q_1}^{s,s'}$ and $a^{(sujs'\mathtt{0}^\ell)_k} \in L_{q_2}^{s,s'}$, for all $s, s' \in \{\mathtt{1}, \mathtt{2}\}$.

Fix any $s, s' \in \{\mathtt{1}, \mathtt{2}\}$. The string $a^{(siujs'\mathtt{0}^\ell)_k}$ is generated in several phases. The first phase expresses all strings $a^{(siujs'\mathtt{0}^\ell)_k}$ with $\delta(uj) = q_2$ and with arbitrary $i \in \Omega$. That is, using two strings of the form $a^{(s''ujs'\mathtt{0}^\ell)_k}$, where $s'' \in \{\mathtt{1}, \mathtt{2}\}$, a string $a^{(siujs'\mathtt{0}^\ell)_k}$ is defined. On a high level this is done by appending

the digit i to the left of the encoded string uj. In terms of the encodings, $a^{(s(i-s'')0^{|ujs'|+\ell})_k}$ should be concatenated to $a^{(s''ujs'0^\ell)_k}$. The operations on $a^{(1ujs'0^\ell)_k}$ and $a^{(2ujs'0^\ell)_k}$, which correspond to two possible values 1 and 2 of s'', are performed simultaneously in the expression λ_i^s, defined as follows:

$$\lambda_i^s(x,y) = (x \cdot a^{(s(i-1)0^*)_k}) \cap (y \cdot a^{(s(i-2)0^*)_k}) \tag{13}$$

and it can be shown that on arguments $a^{(1ujs'0^\ell)_k}$ and $a^{(2ujs'0^\ell)_k}$ its value is $a^{(siujs'0^\ell)_k}$.

The second phase defines a symmetric operation $\rho_j^{s'}$, which appends js' to the right in the encoded string. Given two strings $a^{(siu10^{\ell+1})_k}$ and $a^{(siu20^{\ell+1})_k}$ its value is $a^{(siujs'0^\ell)_k}$.

$$\rho_j^{s'}(x,y) = (x \cdot a^{((j-1)s'0^*)_k}) \cap (y \cdot a^{((j-2)s'0^*)_k}) \tag{14}$$

The third phase corresponds to the intersection of both defined operations: note that $a^{(siujs'0^\ell)_k}$ is expressible by both (13) and (14) applied to appropriate strings, and so $a^{(siujs'0^\ell)_k}$ belongs to their intersection.

The following lemma establishes that these operations work as intended.

Lemma 2. *Let $K \subseteq \Omega^+$ or $K = \{\varepsilon\}$ and let $s' \in \{1,2\}$. Define $L_1 = a^{(1Ks'0^*)_k}$ and $L_2 = a^{(2Ks'0^*)_k}$. Then for each $i \in \Omega$ and $s \in \{1,2\}$*

$$\lambda_i^s(L_1, L_2) = a^{(siKs'0^*)_k}. \tag{15}$$

Proof. The case of $K = \{\varepsilon\}$ can be shown in a similar fashion as Lemma 1, moreover, the proof of the general case for $K \subseteq \Omega^+$ can be adapted also for $K = \{\varepsilon\}$. Thus only the proof for the general case of $K \subseteq \Omega^+$ is considered.

Substituting the languages into (15) yields that the following equality is to be shown:

$$a^{(1Ks'0^*)_k} \cdot a^{(s(i-1)0^*)_k} \cap a^{(2Ks'0^*)_k} \cdot a^{(s(i-1)0^*)_k} = a^{(siKs'0^*)_k}$$

The proof of this equality is done it terms of equations over sets of numbers, i.e., it is shown that

$$((1Ks'0^*)_k + (s(i-1)0^*)_k) \cap ((2Ks'0^*)_k + (s(i-1)0^*)_k) = (siKs'0^*)_k \tag{16}$$

Denote the set on the right-hand side of (16) by S.

Consider arbitrary $x_1 = (1w_1s'0^{\ell_1})_k \in (1Ks'0^*)_k$ and $y_1 = (s(i-1)0^{m_1})_k \in (s(i-1)0^*)_k$. It is shown that the form of base-k positional notation of $x_1 + y_1$ reflects the relation between m_1 and ℓ_1.

Claim 3. *Consider the sum $x_1 + y_1$ as above. Then:*

i. the last non-zero digit of $x_1 + y_1$ is $i-1$ if and only if $\ell_1 > m_1$;

ii. the last non-zero digit of $x_1 + y_1$ is $i - 1 + s'$ if and only if $\ell_1 = m_1$;

iii. One of the following two sets of conditions holds if and only if $\ell_1 < m_1 < |w_1 s'| + \ell_1$*:*

- *the last non-zero digit of* $x_1 + y_1$ *is* s' *and* $x_1 + y_1$ *has a unique digit at least* $c + d$*, which is at position* $m_1 + 1$ *from the right, and all* $\mathtt{0}$*s in this notation form a suffix, or*
- *the positional notation of* $x_1 + y_1$ *is in the set* $(\Sigma_k^* \mathtt{0} \Omega^* s' \mathtt{0}^*)_k$ *and the first* $\mathtt{0}$ *to the left of* s' *is at position* $m_1 + 1$ *from the right*

iv. the sum $x_1 + y_1$ *is in* S *if and only if* $m_1 = |w_1 s'| + \ell_1$

v. the sum $x_1 + y_1$ *is in the set* $(s(i-1)\mathtt{0}^*\mathtt{1}\Omega^* s' \mathtt{0}^*)_k$ *if and only if* $m_1 > |w_1 s'| + \ell_1$*.*

Proof. Consider the last non-zero digit of $x_1 + y_1$. If $\ell_1 > m_1$, then this digit comes from y_1 and is equal to $i - 1$; if $\ell_1 = m_1$, then this digit is obtained as the sum of last non-zero digits of x_1 and y_1 and is equal to $s' + i - 1$, which is at most $2 + c + d - 2 = c + d < k$, so it is a valid digit. Lastly, when $m_1 > \ell_1$, this digit is obtained from last non-zero digit of x_1 and is equal to s'. Since $i \geqslant \mathtt{5}$ and $s' \leqslant \mathtt{2}$, it can be concluded that $s' < i - 1 < i$, and this shows (i) and (ii).

It is left to show (iii–v). Assume that $m_1 > \ell_1$, and that the last non-zero digit of $x_1 + y_1$ comes from x_1 and is equal to s'. Observe that the sets of numbers described in (iii–v) are disjoint: indeed, the numbers in S contain neither any digits of value $c + d$ or more, nor any digit $\mathtt{0}$ to the left of s', and hence the sets described in (iii) and in (iv) are disjoint. Similar argument applies to the numbers in $(s(i-1)\mathtt{0}^*\mathtt{1}\Omega^* s' \mathtt{0}^*)_k$ and the numbers defined in (iii): clearly, all digits in the latter are smaller than $c + d$, and if it had a zero to the left of s', then there would be a digit from $\{\mathtt{1}, \mathtt{2}\}$ between this zero and s'; hence, the sets from (iii) and (v) are also disjoint. Lastly, the sets S and $(s(i-1)\mathtt{0}^*\mathtt{1}\Omega^* s' \mathtt{0}^*)_k$ are disjoint, as the elements of the former have only two digits from set $\{\mathtt{1}, \mathtt{2}\}$ in their base-k positional notation, while the numbers from the latter set have exactly three such digits. Thus sets described in (iv) and (v) are also disjoint. Hence, in order to show (iii–v), it is enough to demonstrate, that

- if $\ell_1 < m_1 < |w_1 s'| + \ell_1$, then the base-$k$ positional notation of $x_1 + y_1$ has a last non-zero digit s', and either
  - it has a digit at least $c + d$ and all zeroes in the notation form a suffix, or
  - it is in the set $(\Sigma_k^* \mathtt{0} \Omega^* s' \mathtt{0}^*)_k$
- if $m_1 = |w_1 s'| + \ell_1$, then $x_1 + y_1 \in S$
- if $m_1 > |w_1 s'| + \ell_1$, then $x_1 + y_1 \in (s(i-1)\mathtt{0}^*\mathtt{1}\Omega^* s' \mathtt{0}^*)_k$.

Consider what is the digit on the $m_1 + 1$ position from the right in $x_1 = (\mathtt{1}w_1 s' \mathtt{0}^{\ell_1})_k$, recall that y_1 has $i-1$ at this position. Since $m_1 > \ell_1$, there are three possibilities:

- it is one of the digit's from Ω, say ω, which happens when $\ell_1 < m_1 < |w_1 s'| + \ell_1$. 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 $2c + 2d - 3 \leqslant k$. This means that the sum may cause a carry. If there is no carry, the digit at position $m_1 + 1$ is at least $c + d$; and if there is a carry, the digit at position $m_1 + 1$ is $\mathtt{0}$, and furthermore, between this $\mathtt{0}$ and s_1 there are only digits from Ω.

- it is the leading digit $\mathtt{1}$, which happens when $m_1 = |w_1 s'| + \ell_1$. In this case $x_1 + y_1 = (\mathtt{1}w_1 s' \mathtt{0}^{\ell_1})_k + (s(i-1)\mathtt{0}^{|w_1 s'| + \ell_1})_k = (siw_1 s' \mathtt{0}^{\ell_1})_k$.

- there is no digit on position $m_1 + 1$ in base-k positional notation of x_1, as it is too short. This happens when $m_1 > |\mathtt{1}w_1| + \ell_1$. Then $x_1 + y_1 = (\mathtt{1}w_1 s' \mathtt{0}^{\ell_1})_k + (s(i-1)\mathtt{0}^{m_1})_k = (s(i-1)\mathtt{0}^{m_1 - |\mathtt{1}w_1| - \ell_1} w_1 s' \mathtt{0}^{\ell_1})_k$.

This ends the case inspection and the proof. □

A similar characterisation can be obtained for addition of x_2 and y_2, where $x_1 = (\mathtt{2}w_2 s' \mathtt{0}^{\ell_2})_k \in (\mathtt{2}Ks' \mathtt{0}^*)_k$ and $y_2 = (s(i-2)\mathtt{0}^{m_2})_k \in (s(i-2)\mathtt{0}^*)_k$.

Claim 4. *Consider the sum $x_2 + y_2$ as above. Then*

- *the last non-zero digit of $x_2 + y_2$ is $i - 2$ if and only if $\ell_2 > m_2$;*

- *the last non-zero digit of $x_2 + y_2$ is $i - 2 + s'$ if and only if $\ell_2 = m_2$;*

- *the last non-zero digit of $x_2 + y_2$ is s', all zeroes form a suffix of the notation, and there is a digit that is at least $c + d$ if and only if $\ell_2 < m_2 < |w_2 s'| + \ell_2$, and furthermore, the digit that is at least $c + d$ is at position $m_2 + 1$ from the right;*

- *the sum $x_2 + y_2$ is in S if and only if $m_2 = |w_2 s'| + \ell_2$*

- *the sum $x_2 + y_2$ is in the set $(s(i-2)\mathtt{0}^* 2 \Omega^* s' \mathtt{0}^*)_k$ if and only if $m_2 > |w_2 s'| + \ell_2$.*

The proof of Claim 4 is similar to the proof of Claim 3, and so it is omitted. Observe that the statement of Claim 4 is simplified, as compared to Claim 3: this is because the sum of the highest digit from Ω (i.e., $c+d-1$) with the greatest value of $i - 1$, i.e., $c + d - 2$, is k, which causes a carry; on the other hand, the sum of $c + d - 1$ and $i - 2$ is always at most $k - 1$, and hence does not cause a carry.

Suppose now that $x_1 + y_1 = x_2 + y_2$. It will be shown that this implies that $x_1 + y_1 \in S$. The analysis depends on the relation between ℓ_2 and m_2

- Suppose that $\ell_2 > m_2$. Then by Claim 4 the last non-zero digit of $x_2 + y_2$ is $i - 2$. On the other hand, by Claim 3 the last non-zero digit of $x_1 + y_1$ is either at least $i - 1$ or at most $\mathtt{2}$, so it cannot be $i - 2$

- Suppose that $\ell_2 = m_2$. Then by Claim 4 the last non-zero digit of $x_2 + y_2$ is $i - 2 + s'$. On the other hand, by Claim 3, the last non-zero digit of $x_1 + y_1$ can be s', which is clearly smaller than $i - 2 + s'$, $i - 1 + s'$, which is clearly larger, or $i - 1$, which is the only feasible case and happens if and only if $\ell_1 > m_1$. Consider then the second-last non-zero digit of $x_2 + y_2 = x_1 + y_1$: since $\ell_1 > m_1$ the second-last digit of $x_1 + y_1$ is either s (if $\ell_1 > m_1 + 1$) or $s' + s$ (if $\ell_1 = m_1 + 1$); on the other hand, as $\ell_2 = m_2$ and $\varepsilon \notin K$, there is a digit from Ω on this position in x_2 and so the second-last non-zero digit of $x_2 + y_2$ is at least $c + s'$. Since $s < s' + s' \leq 4 < c + s$, this rules out this case.

- Suppose now that $\ell_2 < m_2 < |w_2 s'| + \ell_2 - 1$. Then the leading digit of $x_2 + y_2$ is $\mathtt{2}$, which comes from x_2. Furthermore, Claim 4 yields that $x_2 + y_2$ has a digit at least $c + d$, which is moreover at position $m_2 + 1$. Also, all zeroes in the base-k notation of $x_2 + y_2$ form a suffix. The digit with value at least $c + d$ also appears in $x_1 + y_1$ and furthermore all zeroes in this notation form a suffix. From Claim 3 it can be inferred that $\ell_1 < m_1 < |w_1 s'| + \ell_1$ and that the digit at least $c + d$ is at position $m_1 + 1$. Hence, $m_1 = m_2$. Consider the leading digit in $x_1 + y_1$. It is either $\mathtt{1}$ (when $m_1 < |w_1 s'| + \ell_1 - 1$) or $\mathtt{1} + s$ (when $m_1 = |w_1 s'| + \ell_1 - 1$). As $\mathtt{1} < \mathtt{2}$, this rules out the former case. In the latter case, the length of the positional notation of $x_1 + y_1$ is $m_1 + 2$, on the other hand, the length of the positional notation of $x_2 + y_2$ is at least $m_2 + 3$; since $m_1 = m_2$, this is a contradition.

- Consider $m_2 = |w_2 s'| + \ell_2 - 1$. Then an analysis similar to the one in the previous case yields that the leading digit in $x_2 + y_2$ is $\mathtt{2} + s$ and $x_2 + y_2$ has a digit at least $c + d$, which is at position $m_2 + 1$ and furthermore all zeroes form a suffix. Then, as $x_1 + y_1$ also has a digit that is at least $c + d$ and all its zeroes form a suffix, it is concluded, that $m_1 < |w_1 s'| + \ell_1$. Then the leading digit of $x_1 + y_1$ is either $\mathtt{1}$, if the inequality is tight, or $\mathtt{1} + s$ if it is not. In both cases, this is less than $\mathtt{2} + s$.

- If $m_2 = |w_2 s'| + \ell_2$ then $x_2 + y_2 \in S$, by Claim 4, and there is nothing more to prove.

- Suppose now that $m_2 > |w_2 s'| + \ell_2$. By Claim 4 the base-k positional notation of $x_2 + y_2$ has 3 digits from the set $\{\mathtt{1}, \mathtt{2}\}$, and so clearly also $x_1 + y_1$ has them. Using Claim 3 it follows that this is possible only in two cases: when $m_1 > |w_1 s'| + \ell_1$ or when $m_1 > |w_1 s'| + \ell_1$ or when $\ell_1 < m < |w_1 s'| + \ell_1$. However, in the former case $x_1 + y_1$ has a digit $\mathtt{0}$

which is to the left of s' and there are only digits from Ω between this s' and $\mathtt{0}$, which does not hold for $x_2 + y_2$. Thus $m_1 > |w_1 s'| + \ell_1$, and consequently the second-leading digit of $x_1 + y_1$ is $i - 1$. On the other hand, from Claim 4 guarantees that the second-leading digit of $x_2 + y_2$ is $i - 2$, contradiction.

The case analysis above yields that the left-hand side of (16) is a subset of the right-hand side. To show the converse, consider any $(siw\mathtt{0}^\ell)_k \in S$. Then, by definition, $x_1 = (\mathtt{1}w\mathtt{0}^\ell)_k \in (\mathtt{1}Ks'\mathtt{0}^*)_k$ and $y_1 = (s(i-1)\mathtt{0}^{\ell+|w|})_k \in (s(i-1)\mathtt{0}^*)_k$. Clearly, $x_1 + y_1 = (siw\mathtt{0}^\ell)_k$. One can give a similar representation of $x_2 \in (\mathtt{2}Ks'\mathtt{0}^*)_k$ and $y_2 \in (s(i-2)\mathtt{0}^*)_k$, such that $x_2 + y_2 = (siw\mathtt{0}^\ell)_k$. Therefore, $(siw\mathtt{0}^\ell)_k \in \big((\mathtt{1}Ks'\mathtt{0}^*)_k + (s(i-1)\mathtt{0}^*)_k\big) \cap \big((\mathtt{2}Ks'\mathtt{0}^*)_k + (s(i-2)\mathtt{0}^*)_k\big)$, and thus the stated equality (16) holds. □

Similarly to Lemma 2, the following property of $\rho_j^{s'}$ shows that it works as intended.

Lemma 3. *Let $K \subseteq \Omega^+$ and let $s \in \{\mathtt{1}, \mathtt{2}\}$. Define $L_1 = a^{(sK\mathtt{10}^*)_k}$ and $L_2 = a^{(sK\mathtt{20}^*)_k}$. Then, for every $j \in \Omega$ and $s' \in \{\mathtt{1}, \mathtt{2}\}$,*

$$\rho_j^{s'}(L_1, L_2) = a^{(sKjs'\mathtt{0}^*)_k}. \tag{17}$$

The other crucial property of the expressions λ_i^s and ρ_s^i 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 $s' \in \{\mathtt{1}, \mathtt{2}\}$. Define $L_1 = a^{(\mathtt{1}Ks'\mathtt{0}^*)_k}$ and $L_2 = a^{(\mathtt{2}Ks'\mathtt{0}^*)_k}$. Then, for each $s \in \{\mathtt{1}, \mathtt{2}\}$ and $i \in \Omega$, both concatenations in $\lambda_i^s(L_1, L_2)$ are unambiguous.*

Proof. The case for $K = \{\varepsilon\}$ follows easily from Lemma 2, so in the rest of the proof only the case of $K \subseteq \Omega^+$ is considered.

Consider first the $\lambda_i^{\mathtt{1}}$, i.e., $s = \mathtt{1}$. Let $x_1 = (\mathtt{1}w_1 s'\mathtt{0}^{\ell_1})_k \in (\mathtt{1}Ks'\mathtt{0}^*)_k$, $y_1 = (s(i-1)\mathtt{0}^{m_1})_k \in (s(i-1)\mathtt{0}^*)_k$ and $x_2 = (\mathtt{1}w_2 s'\mathtt{0}^{\ell_2})_k \in (s(i-1)\mathtt{0}^*)_k$, $y_2 = (s(i-1)\mathtt{0}^{m_2})_k \in (s(i-1)\mathtt{0}^*)_k$ be such that $x_1 + y_1 = x_2 + y_2$. Claim 3 is employed to show that $x_1 = x_2$ and $y_1 = y_2$. Observe, that it is enough to show that $m_1 = m_2$, as this yields $y_1 = y_2$ and consequently also $x_1 = x_2$. Consider the relation between ℓ_1 and m_1

- if $\ell_1 > m_1$ then by Claim 3 the last non-zero digit of $x_1 + y_1$ is $i - 1$ and it comes from y_1, in particular, it is on position $m_1 + 1$ from the right. Clearly, $i - 1$ is also the last non-zero digit of $x_2 + y_2$, and again from Claim 3 it is obtained that the $\ell_2 > m_2$. Hence, the last non-zero digit of $x_2 + y_2$ is on position $m_2 + 1$. Consequently, $m_1 = m_2$, which ends the proof in this case.

- Suppose that $\ell_1 = m_2$. Using Claim 3 is is obtained that the last non-zero digit of x_1+y_1 is $i-1+s'$ and it is on position $m_1+1 = \ell_1+1$ from the right. By an argument as in the previous case it can be deduced that $\ell_2 = m_2$ and the last non-zero digit of x_2+y_2 is on position m_2+1. Hence, $m_1 = m_2$, as claimed.

- Let now $\ell_1 < m_1 < |sw_1| + \ell_1$. By Claim 3, the number $x_1 + y_1$ in its base-k notation either has a unique digit which is at least $c+d$ and all 0s form a suffix, or has a digit 0 to the left of s' and there are only digits from Ω between them. In the former case the digit that is at least $c+d$ is on position m_1+1 from the right, in the second case this 0 is as well at position m_1+1. Consider the former case. Clearly, x_2+y_2 has the same digit on this position and, by Claim 3, it can be concluded that this digit is on position m_2+1 from the right and so $m_2 = m_1$, which ends the proof in this case. If x_1+y_1 has 0 to the left of s', then so does x_2+y_2, and, by Claim 3, such a digit 0 is at position m_2+1 from the right. Hence $m_2 = m_1$, which ends also this subcase.

- Let now $m_1 = |sw_1| + \ell_1$. Then, by Claim 3, the number $x_1 + y_1$ is in S. Moreover the length of the base-k notation of $x_1 + y_1$ is $m_1 + 2$. On the other hand, $x_2 + y_2$ also belongs to S and, by Claim 3, it holds that $m_2 = |sw_2| + \ell_2$. In particular, the length of the base-k notation of $x_2 + y_2$ is $m_2 + 2$. Thus $m_1 = m_2$, as desired.

- Consider the last case, when $m_1 > |sw_1|+\ell_1$. Then by Claim 3, x_1+y_1 is in the set $(s(i-1)0^*1\Omega^*s'0^*)_k$, and so clearly so is x_2+y_2. Using Claim 3 again yields that $m_2 > |sw_2| + \ell_2$. Observe that the length of the base-k notation of x_1+y_1 is m_1+2 and the length of such notation for x_2+y_2 is m_2+2. Thus $m_1 = m_2$, which ends the proof in this case.

The case inspection above yields that $x_1 + y_1 = x_2 + y_2$ implies $x_1 = x_2$ and $y_1 = y_2$. Thus λ_i^1 is with unambiguous concatenation. The proof for λ_i^2 is similar and is thus omitted. □

Similarly, the concatenations in $\rho_j^{s'}$ are unambiguous under the right substitution.

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 each $j \in \Omega$ and $s' \in \{1,2\}$, both concatenations in $\rho_j^{s'}(L_1, L_2)$ are unambiguous.*

Using the expressions λ_s^i and $\rho_j^{s'}$, each representing a conjunction of two concatenations, the main part of the grammar for the proof of Theorem 1 is

constructed as follows:

$$X_q^{s,s'} \to \lambda_b^s(A_{1,s'}, A_{2,s'}) \qquad \text{for } b \in \Omega \text{ such that } \delta(b) = q \tag{18a}$$

$$X_{\delta(q',q'')}^{s,s'} \to \lambda_i^s(X_{q''}^{1,s'}, X_{q''}^{2,s'}) \,\&\, \rho_j^{s'}(X_{q'}^{s,1}, X_{q'}^{2,2}) \qquad \text{for } i,j \in \Omega \text{ and } q', q'' \in Q \tag{18b}$$

$$X^{s,s'} \to X_q^{s,s'} \qquad \text{for } q \in F \tag{18c}$$

Here, the nonterminals $A_{i,j}$ are from Lemma 1, and each of them generates the corresponding language $a^{(ij0^*)_k}$. Whenever the expressions λ_s^i and $\rho_j^{s'}$ use constant languages of the form $a^{(ij0^*)_k}$, the grammar accordingly refers to the appropriate $A_{i,j}$.

Proof of Theorem 1. The proof follows by showing the claim:

Main Claim. *The given conjunctive grammar* (18) *is unambiguous, and each nonterminal* $X_q^{s,s'}$ *generates* $\{a^{(sws'0^\ell)_k} \mid \delta(w) = q,\ \ell \geqslant 0\}$*, while each* $X^{s,s'}$ *generates* $\{a^{(sws'0^\ell)_k} \mid a^{(w)_k} \in L(M), \ell \geqslant 0\}$*, for all* $s, s' \in \Omega$ *and* $q \in Q$*.*

Observe, that with the main claim established, it is enough to restrict $L(X^{1,1})$ to strings of odd length, which is done by adding the following additional rules

$$X \to X^{1,1} \,\&\, O, \tag{19}$$

$$O \to a \mid aaO \tag{20}$$

to the grammar (18). Then X generates $\{a^{(1w1)_k} \mid a^{(w)_k} \in L(M)\}$.

Now the Main Claim is shown.

Consider the resolved system of equations associated with the grammar (18). Observe that all constants in (18) come from the grammar defining the nonterminals $A_{i,j}$, see Lemma 1, and all these constants are ε-free. Hence, the associated resolved system uses only constants that are ε-free and thus has a unique ε-free solution, which coincides with the least solution. Since the desired solution consists of sets that are all ε-free, it is enough to check that $X_q^{s,s'} = \{a^{(sws'0^\ell)_k} \mid \delta(b) = q,\ \ell \geqslant 0\}$, $X^{s,s'} = \{a^{(sws'0^\ell)_k} \mid w \in L(M),\ \ell \geqslant 0\}$ is indeed a solution.

Consider first the rule (18a) and the expressions corresponding to its right-hand side. Lemma 2 is used to determine the value of this expression under substitution of $a^{(1s'0^*)_k}$ for $A_{1,s'}$ and $a^{(2s'0^*)_k}$ for $A_{2,s'}$. In the terminology of the lemma, $K = \{\varepsilon\}$. Then $L_1 = a^{(1Ks'0^*)_k}$ and $L_2 = a^{(2Ks'0^*)_k}$, and, by the lemma,

$$\begin{aligned}\lambda_b^s(a^{(1s'0^*)_k}, a^{(2s'0^*)_k}) &= a^{(sbKs'0^*)_k} \\ &= a^{(sbs'0^*)_k}.\end{aligned}$$

Taking the union over $b \in \Omega$ such that $\delta(b) = q$ yields that

$$\bigcup_{\substack{b\in\Omega\\ \delta(b)=q}} \lambda_b^s(a^{(1s'0^*)_k}, a^{(2s'0^*)_k}) = \{a^{(sbs'0^\ell)_k} \mid b \in \Omega,\ \delta(b) = q,\ \ell \geqslant 0\}.$$

Furthermore observe, that as M is deterministic, the union is taken over disjoint sets.

Turning to the rules (18b), Lemmata 2 and 3 are used to determine the value of the corresponding expression under the appropriate substitution: Lemma 2 for $\lambda_i^s(X_{q''}^{1,s'}, X_{q''}^{2,s'})$ and Lemma 3 for $\rho_j^{s'}(X_{q'}^{s,1}, X_{q'}^{2,2})$. In the first case, using the terminology of Lemma 2, let $K_{q''} = \{w \mid \delta(w) = q''\}$, then $L_1 = a^{(1K_{q''}s'0^*)_k}$ and $L_2 = a^{(2K_{q''}a'0^*)_k}$; then the lemma states that

$$\lambda_i^s(a^{(1K_{q''}s'0^*)_k}, a^{(2K_{q''}s'0^*)_k}) = a^{(siK_{q''}s'0^*)_k}.$$

Applying Lemma 3 to the second expression, with the values $K_{q'} = \{w \mid \delta(w) = q'\}$, $L_1 = a^{(sK_{q''}10^*)_k}$ and $L_2 = a^{(sK_{q'}20^*)_k}$, yields

$$\rho_j^{s'}(a^{(sK_{q'}10^*)_k}, a^{(sK_{q'}20^*)_k}) = a^{(sK_{q'}js'0^*)_k}.$$

Then the conjunction in (18b) defines

$$\begin{aligned}\lambda_i^s(a^{(1K_{q''}s'0^*)_k}, a^{(2K_{q''}s'0^*)_k}) \cap \rho_j^{s'}((a^{(sK_{q'}10^*)_k}, (a^{(sK_{q'}20^*)_k}) &= \\ = a^{(siK_{q''}s'0^*)_k} \cap a^{(sK_{q'}js'0^*)_k} &= \\ = \{a^{(siwjs'0^\ell)_k} \mid \delta(iw) = q',\ \delta(wj) = q'',\ \ell \geqslant 0\}.&\end{aligned}$$

Taking the union over q', q'' such that $\delta(q', q'') = q$ yields

$$\begin{aligned}\bigcup_{\substack{q',q''\in Q,\\ \delta(q',q'')=q}} \{a^{(siwjs'0^\ell)_k} \mid \delta(iw) = q',\ \delta(wj') = q'',\ \ell \geqslant 0\} = \\ = \{a^{(siwjs'0^\ell)_k} \mid \delta(iwj) = q,\ \ell \geqslant 0\}.\end{aligned}$$

Notice that as M is deterministic, the above union is over disjoint sets. Lastly, taking the union over $i, j, \in \Omega$ gives

$$\begin{aligned}\bigcup_{i,j\in\Omega} \{a^{(siwjs'0^\ell)_k} \mid \delta(iwj) = q,\ \ell \geqslant 0\} = \\ = \{a^{(sw's'0^\ell)_k} \mid \delta(w') = q,\ |w'| \geqslant 2,\ \ell \geqslant 0\}.\end{aligned}$$

The union is again over disjoint sets. The variable $X_q^{s,s'}$ is defined as the union of all rules (18a) and (18b), which is

$$\begin{aligned}X_q^{s,s'} &= \{a^{(sw's'0^\ell)_k} \mid \delta(w') = q,\ |w'| \geqslant 2,\ \ell \geqslant 0\} \cup \\ &\quad \cup \{a^{(sw's'0^\ell)_k} \mid \delta(w') = q,\ |w'| = 1,\ \ell \geqslant 0\} \\ &= \{a^{(sw's'0^\ell)_k} \mid \delta(w') = q,\ \ell \geqslant 0\},\end{aligned}$$

as desired. 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 $X_q^{s,s'}$ is unambiguous.

Lastly, consider the equation for $X^{s,s'}$, which corresponds to the rules (18c). Since the value of the variables $X_q^{s,s'}$ is already known, it easily follows that

$$\begin{aligned} X^{s,s'} &= \bigcup_{q \in F} \{a^{(sws'0^\ell)_k} \mid \delta(w) = q,\ \ell \geqslant 0\} \\ &= \{a^{(sw0^\ell)_k} \mid \delta(w) \in F,\ \ell \geqslant 0\} \\ &= \{a^{(sw0^\ell)_k} \mid \delta(w) \in L(M),\ \ell \geqslant 0\}, \end{aligned}$$

as desired. Furthermore, as M is deterministic, this is a union of disjoint sets.

Turning to the unambiguity of the grammar (18), Lemma 1 supplies an unambiguous conjunctive grammar for the languages $a^{(ij0^*)_k}$, and it remains to show that each concatenation and union in (18) is unambiguous. The only concatenations that appear in (18) are in the subexpressions λ_i^s and $\rho_j^{s'}$, and in each case, the assumptions of Lemmata 4 and 5 are met; consequently, all concatenations in the system (18) are unambiguous. The unambiguity of the choice of a rule has already been shown above. □

6 Dense encoding of trellis automata

For a language $L \subseteq a^*$, consider the number

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

called the *density* of L [15]. This limit, if it exists, always lies within the bounds $0 \leqslant d(L) \leqslant 1$. Let a language be called *sparse*, if $d(L) = 0$, and *dense*, if $d(L) = 1$.

All unambiguous conjunctive grammars constructed so far generate only sparse unary languages (actually, only exponentially-growing languages). Using only sparse languages in the constructions is, to some extent, 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. Of course, this does not mean that non-sparse sets cannot be represented at all—for instance, it is easy to modify the grammar in Example 2 to represent the language $\{a^{4^n} \mid n \geqslant 0\} \cup a(aa)^*$ of density $\frac{1}{2}$—but only that, once represented, non-sparse sets cannot be non-trivially concatenated to anything.

This section develops a method of simulating the computation of a trellis automaton in an unambiguous conjunctive grammar generating a unary language of density 1. This result parallels that of Theorem 1, which simulates

a trellis automaton in a grammar generating a unary language of density 0. The proof of the new result is actually inferred from Theorem 1 by developing constructions on top of it.

The general idea of the new construction is based on the following representation of a^* by an unambiguous concatenation:

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

The correctness of this example is established in the following lemma.

Lemma 6. *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 $n_c \in (c\{0, c\}^*)_k \cup \{\varepsilon\}$.*

Proof. The argument is based on the fact that each number $d \in \{0, \dots, k-1\}$ is uniquely representable as a sum $d = d_1 + d_2 + d_4 + d_8 + \ldots + d_{\frac{k}{2}}$ with $d_c \in \{0, c\}$; the choice of nonzero summands is determined by the binary notation of d.

Now let $n \geqslant 0$ be any integer and consider its representation $n = n_1 + n_2 + n_4 + n_8 + \ldots + n_{\frac{k}{2}}$ with $n_c \in (\{0, c\}^*)_k$. Each i-th base-k digit of n is a obtained as a sum of the i-th digits of $n_1, \dots, n_{\frac{k}{2}}$. This sum is at most $k-1$ in every position, and hence there is no carry anywhere. Since each i-th digit of n is uniquely representable as a sum of digits that may occur in the base-k notation of $n_1, \dots, n_{\frac{k}{2}}$, this uniquely determines each digit of each n_c.

To construct the actual representation, define the homomorphisms $h_c \colon \Sigma_k \to \{0, c\}$ with $c \in \{1, 2, 4, 8, \dots, \frac{k}{2}\}$ by $h_c(i) = 0$ if $\lfloor \frac{i}{c} \rfloor$ is even and $h_c(i) = c$ if $\lfloor \frac{i}{c} \rfloor$ is odd. Let $n = (w)_k$. Then the desired representation is given by $(h_1(w))_k + (h_2(w))_k + (h_4(w))_k + \ldots + (h_{\frac{k}{2}}(w))_k$. For example, $(12345670)_8 = (10101010)_8 + (2200220)_8 + (44440)_8$. □

Consider the concatenation in Example 3. Let one of the languages L_i (in the actual construction, this shall be L_2) be replaced with a language $L'_i \subseteq L_i$, which encodes a computation of a trellis automaton operating on the two-letter input alphabet $\{0, i\}$ similarly to the encoding in Theorem 1. Then the concatenation $L = L_1 \dots L_{i-1} L'_i L_{i+1} \dots L_{k/2}$ is still unambiguous, and the density of the language L is controlled by the given linear conjunctive language, and can be set to any desired value. This construction leads to representing the following languages by unambiguous conjunctive grammars.

Theorem 2. *Let L be a linear conjunctive language over a two-letter 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 $h \colon \Sigma_k^* \to \Gamma^*$ by $h(4i) = h(4i+1) = e$ and $h(4i+2) = h(4i+3) = f$ for all $i \in \{0, \dots, \frac{k}{4} - 1\}$. Then the language $\{a^{(w)_k} \mid h(w) \in e^* L\}$ is generated by an unambiguous conjunctive*

grammar. Given a trellis automaton recognizing L, this grammar can be effectively constructed.

In order to prove the theorem according to the above general idea, there are several independent claims to be established. First, one needs to represent the sets $L_1, L_2, \ldots, L_{\frac{k}{2}}$ by unambiguous grammars. Next, a representation L'_2 of a linear conjunctive language is obtained by modifying the representation from Theorem 1 using additional rules.

Lemma 7. *For $k \geqslant 16$ and $c \leqslant \frac{k}{2}$, where both k and c are powers of two, there is an unambiguous conjunctive grammar generating the language $a^{(c\{0,c\}^*)_k}$.*

Proof. Observe, that the language $c\{0,c\}^*$ clearly is linear conjunctive, and thus (for some values of c), the Theorem 1 yields a construction of languages $a^{1c\{0,c\}^*1}$, which is "similar" to desired language $a^{(c\{0,c\}^*)_k}$. However, the construction employed in Theorem 1 greatly simplifies, when it is applied to the particular language $c\{0,c\}^*$. In particular, the construction can be refactored, so that it works for every $c \leqslant k/2$.

The proof proceeds by first constructing grammars for the languages $a^{(s\{0,c\}^*)_k}$ for $s \in \{\frac{k}{2}+2, \frac{k}{2}+5\}$, then the grammar for $a^{(c\{0,c\}^*)_k}$ is obtained out of them. The two aforementioned languages are constructed using a recursive dependence similar to the one described by expression λ_i^s, see (13), and the obtained rules are simplifications of the rules used in λ_i^s.

Consider the following grammar, it uses some of the nonterminals $A_{i,j}$ defined in Section 4, as well as new nonterminals $X_{\frac{k}{2}+2}$, $X_{\frac{k}{2}+5}$ and C.

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

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

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

As the nonterminals $A_{i,j}$ are supplied with the rules defined in Lemma 1, they define the languages $a^{(ij0^*)_k}$; the other nonterminals generate the languages

$$L(X_s) = a^{(s\{0,c\}^*)_k} \tag{21a}$$

$$L(C) = a^{(c\{0,c\}^*)_k} \tag{21b}$$

As already noted, this grammar is a simple variation of the construction for λ_i^s, defined in (13). In particular, it can be shown in the same way as for λ_i^s that the constructed grammar is unambiguous and generates the desired languages. To be more precise: as in Lemma 2 it can be shown that the grammar generates the intended languages, i.e., that (21) holds. The unambiguity of concatenation is shown as in Lemma 4; the nonterminal C

has only one rule and so there is no ambiguity of rule choice for it, furthermore both $X_{\frac{k}{2}+2}$ and $X_{\frac{k}{2}+5}$ have only two rules and one of them is terminating, hence the unambiguity of choice easily follows. □

The second step is the encoding of the language accepted by a TA in $L'_2 \subseteq L_2 = a^{(\mathtt{2}\{\mathtt{0},\mathtt{2}\}^*)_k}$. Theorem 1 gives a construction that encodes a language $L(M)$ for M working over alphabet $\{c, c+1, \ldots, c+d-1\}$ as $a^{(\mathtt{1}L(M)\mathtt{1})_k}$. If d is set to 3 and M is restricted to use only two digits $\{c, c+2\}$ out of $\{c, c+1, c+2\}$, the encoding provided by Theorem 1 is a subset of $a^{(\mathtt{1}\{c,c+2\}^*\mathtt{1})_k}$. Thus, a natural approach is to, on one hand, modify M into M', which works exactly the same as M but it uses digits $\{\mathtt{0}, \mathtt{2}\}$ instead of $\{c, c+2\}$; and on the other hand modify the encoding $a^{(\mathtt{1}L(M)\mathtt{1})_k}$ so that every $a^{(\mathtt{1}(c+d_m)(c+d_{m-1})\ldots(c+d_0)\mathtt{1})_k}$ is replaced with $a^{(\mathtt{2}d_m d_{m-1}\ldots d_0\mathtt{0})_k}$. In this way, $a^{(\mathtt{1}L(M)\mathtt{1})_k}$ is changed into into $a^{(\mathtt{2}L(M')\mathtt{0})_k} \subseteq L_2$.

Changing M into M' is done by simple syntactic modifications in the definition trellis automaton M, the changes of the encoding $a^{(\mathtt{1}L(M)\mathtt{1})_k}$ are done on each of its string separately. Consider a string $a^{(\mathtt{1}(c+d_m)(c+d_{m-1})\ldots(c+d_0)\mathtt{1})_k}$. Then $\mathtt{1}(c+d_m)(c+d_{m-1})\ldots(c+d_0)\mathtt{1}$ should be replaced with $\mathtt{2}d_m d_{m-1}\ldots d_0\mathtt{0}$, the replacement is done position by position. On the technical level this is done by concatenating an appropriate string from $a^{((k-c+1)^+)_k}$:

$$a^{(\mathtt{1}(c+d_m)(c+d_{m-1})\ldots(c+d_0)\mathtt{1})_k} \cdot a^{((k-c-1)^{m+1})_k} \cdot a^c = a^{(\mathtt{2}d_m d_{m-1}\ldots d_0\mathtt{0})_k}$$

The correctness of this construction is one of the key features of proof of Theorem 3. However, first one needs to make sure that the language $a^{((k-c-1)^+)_k}$ can be represented by an unambiguous conjunctive grammar; this is shown in the below Lemma 8.

Lemma 8. *For $k \geqslant 16$ and $c \leqslant \frac{k}{2}$, where both k and c are powers of two, there is an unambiguous conjunctive grammar generating the language $a^{((k-c-1)^+)_k}$.*

Proof. The construction is a simple variant of the construction of languages in Lemma 7: nonterminals Y_1 and Y_2 should generate the languages $a^{(\mathtt{1}(k-c-1)^*)_k}$ and $a^{(\mathtt{2}(k-c-1)^*)_k}$, respectively. These languages can be represented through each other by a recursive definition. Then a nonterminal D is defined using them, it produces the language $a^{(k-c-1^+)_k}$.

Consider the grammar

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

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

in which the noterminals $A_{i,j}$ are supplied with the rules as in Lemma 1. Then the new nonterminalsof this grammar generate the languages

$$L(Y_s) = a^{(s(k-c-1)^*)_k}$$

$$L(D) = a^{((k-c-1)^+)_k}$$

The proof follows in the same way as the proof in Lemma 7. Furthermore, as in Lemma 7, it is shown that this grammar is unambiguous. □

Theorem 3. *For every linear conjunctive language* $L \subseteq \{0,2\}^* \setminus 0\{0,2\}^*$ *and for every base* $k \geqslant 16$ *that is a power of two, there is an unambiguous conjunctive grammar generating the language* $\{a^{(w)_k} \mid w \in L\}$.

Proof. Consider the languages $L^{(0)} = (\{2\}^{-1}L\{0\}^{-1}) \setminus \{\varepsilon\}$ and $L^{(2)} = (\{2\}^{-1}L\{2\}^{-1})\setminus\{\varepsilon\}$. By known closure properties of linear conjunctive grammars, both of these languages are linear conjunctive. Since they moreover are ε-free, they are also recognised by some trellis automata, i.e., there are trellis automata M_0 and M_2 such that $L(M^0) = L^{(0)}$ and $L(M^2) = L^{(2)}$. Consider a trellis automaton M'_0 which looks exactly like M_0 except that it works over alphabet $\{c, c+2\}$ instead of $\{0,2\}$. By Theorem 1 there is an unambiguous conjunctive grammar with a nonterminal Z defining a language $\{a^{(1w1)_k} \mid w \in L(M'_0)\}$. Add to this grammar a rule

$$Z^0 \to ZDa^c \,\&\, C,$$

and the rules for nonterminals C and D are as in Lemma 7 and Lemma 8. It is shown that Z^0 generates the language

$$\{a^{(2b_mb_{m-1}\ldots b_10)_k} \mid a^{(1(b_m+c)(b_{m-1}+c)\ldots(b_1+c)1)_k} \in L(Z)\}$$
$$= \{a^{(2b_mb_{m-1}\ldots b_10)_k} \mid (b_m+c)(b_{m-1}+c)\ldots(b_1+c) \in L(M'_0)\}.$$

Furthermore, the rule for Z^0 has unambiguous concatenation.

Consider any $a^{(1(b_m+c)(b_{m-1}+c)\ldots(b_1+c)1)_k} \in L(Z)$ and $a^{((k-c-1)^{m+2})_k} \in L(D)$ and their concatenation. The digits of $1(b_m+c)(b_{m-1}+c)\ldots(b_1+c)1$ are processed one by one: a $k-c-1$ and a carry of 1 is added to every digit, which is either c or $c+2$. The former is turned to 0 and the latter to 2. The following intersection with $a^{(2\{0,2\}^*)_k}$ ensures that every digit is processed.

A formal proof that the rule indeed generates the given language follows in a similar way as in Lemma 13. Also the unambiguity of concatenation follows in the same way as in Lemma 4. By definition $(b_m+c)(b_{m-1}+c)\ldots(b_1+c) \in L(M'_0)$ if and only if $b_mb_{m-1}\ldots b_1 \in L(M^0)$. Hence,

$$L(Z^0) = \{a^{(2w0)_k} \mid w \in L(M^0)\} = \{a^{(2w0)_k} \mid 2w0 \in L\} = a^{(L\cap 2\{0,2\}^*0)_k}.$$

Now, a similar construction is applied for the automaton M_2 recognising the language $L^{(2)}$: consider the automaton M'_2 which looks like M_2 except that it uses the alphabet $\{c, c+2\}$ instead of $\{0,2\}$. Then, similarly to the construcion of the nonterminal Z^0, a grammar for a nonterminal Z^2 can be constructed so that

$$L(Z^2) = \{a^{(2w0)_k} \mid w \in L(M^2)\} = \{a^{(2w0)_k} \mid 2w2 \in L\}.$$

Then

$$L(a^2Z^2) = a^{(L\cap 2\{0,2\}^*2)_k}.$$

Add a nonterminal Z' to the grammar, let it have the rules

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

Then Z' generates the language

$$a^{(L\cap 2\{0,2\}^*0)_k} \cup a^{(L\cap 2\{0,2\}^*2)_k} \cup a^{(L\cap \Sigma^{\leqslant 2})_k} = a^{(L)_k}.$$

Concerning the unambiguity of the constructed grammar: the grammar for nonterminal Z, which generates the language $\{a^{(1w1)_k} \mid w \in L\}$, is unambiguous by Theorem 1. It was already shown that the rules for nonterminals $Z^{(0)}$ and $Z^{(2)}$ have unambiguous concatenation. Lastly, the rule for Z' has only trivial concatenation with a^2, which is unambiguous, and different rules for Z' clearly generate disjoint languages, hence the unambiguity of choice follows. □

Proof of Theorem 2. Observe that $h(0) = e$ and $h(2) = f$ and consider the language $L' = h^{-1}(L) \cap \{0,2\}^*$. It is intuitively obtained by taking a string from L and replacing each of its es by 0 and each of its fs by 2. Observe that $h(L') = L$. Furthermore, since none of the strings in L begins with e, $L' \subseteq \{0,2\}^* \setminus 0\{0,2\}^*$.

Let k be a power of 2 (be it 16). By Lemma 7, the languages $a^{(c\{0,c\}^*)_k}$ for $c \in \{1,2,4,8,\ldots,\frac{k}{2}\}$ are all unambiguous conjunctive, and hence so are the languages $L_c = a^{(c\{0,c\}^*)_k} \cup \{\varepsilon\}$. Then, by Lemma 6, the concatenation $L_1L_2L_4\cdots L_{\frac{k}{2}}$ is unambiguous and equal to a^*.

The theorem is proved by replacing L_2 in this concatenation with a language given by Theorem 3 for the language L' denote this language by L'_2. To be more precise, $L'_2 = \{a^{(w)_k} \mid w \in L'\}$; rewriting this in terms of L instead of L' yields

$$L'_2 = \{a^{(w)_k} \mid h(w) \in L, w \in \{0,2\}^*\}. \tag{22}$$

As obviously $L'_2 \subseteq L_2$, the concatenation $L_1L'_2L_4\cdots L_{\frac{k}{2}}$ is unambiguous, it remains to be shown that it is equal to $\{a^{(w)_k} \mid h(w) \in e^*L\}$.

Consider any a^n, where $n = (w)_k$. Take the unique representation of n, provided by Lemma 6, i.e., $n = (w_1)_k + (w_2)_k + (w_4)_k + (w_8)_k + \ldots + (w_{\frac{k}{2}})_k$, where $w_c \in \{c(0,c)^*\}$ for each $c \in \{1,2,4,\ldots,\frac{k}{2}\}$. Observe first that

$$a^n \in L_1L'_2L_4\cdots L_{\frac{k}{2}} \iff a^{(w_2)_k} \in L'_2. \tag{23}$$

Indeed, by Lemma 6 it holds that $a^n \in L_1L_2L_4\cdots L_{\frac{k}{2}}$ and its factorisation into elements of $a^{(w_1)_k} \in L_1$, $a^{(w_2)_k}$, …, $a^{(w_{\frac{k}{2}})_k} \in L_{\frac{k}{2}}$ is unique. So either

this is also a factorisation into elements of L_1, L'_2, L_4, ..., $L_{\frac{k}{2}}$, in which case $a^{(w_2)_k} \in L'_2$, or $a^{(w_2)_k} \notin L'_2$ and so $a^n \notin L_1 L'_2 L_4 \cdots L_{\frac{k}{2}}$.

Then observe that

$$h(w) = h(\mathtt{0}^{|w|-|w_2|} w_2). \tag{24}$$

Indeed, fix i and consider the i-th digits d_c in base-k positional notation of $(w_c)_k$, for $c \in \{1, 2, \ldots, \frac{k}{2}\}$. Some of these numbers may be shorter than i, extend their notation by leading zeroes up to position $|w|$. Then d_c is in $\{\mathtt{0}, c\}$. Then the sum $d_1 + d_2 + \ldots + d_{\frac{k}{2}}$ is at most $k-1$ and so there are no carries in the sum $(w_1)_k + (w_2)_k + (w_4)_k + (w_8)_k + \ldots + (w_{\frac{k}{2}})_k$. Hence, the i-th digit d in w is equal to $d_1 + d_2 + \ldots + d_{\frac{k}{2}}$. It is easy to check that straight from the definition that $h(d) = h(d_2)$. As this holds for an arbitrary position i, this holds for $\mathtt{0}^{|w|-|w_2|} w_2$ as well.

Now, it can be concluded that $a^n \in L_1 L'_2 L_4 \cdots L_{\frac{k}{2}}$ if and only if $h(w) \in L$.

$\Rightarrow$ Suppose that $a^n \in L_1 L'_2 L_4 \cdots L_{\frac{k}{2}}$. Then by (23), $a^{(w_2)_k} \in L'_2$. From (22) it can be then concluded that $w_2 \in \{\mathtt{0}, \mathtt{2}\}^*$ and $h(w_2) \in L$. The latter, by (24), yields that $h(w) \in e^{|w|-|w_2|} L$, and so consequently $h(w) \in e^* L$.

$\Leftarrow$ So suppose that $h(w) \in e^* L$, hence there is m such that $h(w) \in e^m L$. By (24), $h(w) = h(\mathtt{0}^{|w|-|w_2|} w_2)$. Since the leading symbol in w_2 is $\mathtt{2}$ (or there is no symbol at all) and $h(\mathtt{2}) = f$, it can be concluded that the e-prefix of $h(w)$ is of length $|w|-|w_2|$, and so $m = |w|-|w_2|$. Hence, $h(w_2) \in L$. By (22) this means that $a^{(w_2)_k} \in L'_2$ and then (23) implies $a^n \in L_1 L'_2 L_4 \cdots L_{\frac{k}{2}}$, which concludes the proof. □

With Theorem 2 established, one can infer from it the following encoding of an arbitrary trellis automaton over a two-letter alphabet in a dense unary language.

Corollary 1. *For every linear conjunctive language L over $\Gamma = \{e, f\}$ and for every base $k \geqslant 256$ that is an even power of two, there exists an unambiguous conjunctive language $K \subseteq a^*$ of density 1, with $w \in L$ if and only if $a^{(g(w))_k} \in K$, where $g\colon \Gamma^* \to \Sigma_k^*$ is a letter-to-letter homomorphism defined by $g(e) = 2\sqrt{k}$ and $g(f) = 2\sqrt{k} + 2$.*

Given a trellis automaton recognizing L, a grammar for K can be effectively constructed.

To prove this corollary, construct another linear conjunctive language $\widetilde{L} = h_0(L) \cup (\{e, f\}^* \setminus \{fe, ff\}^*)$, where $h_0\colon \Gamma^* \to \Gamma^*$ is a homomorphism with $h_0(e) = fe$, $h_0(f) = ff$. Applying Theorem 2 to $\widetilde{L}$ yields the desired unary language.

7 Decision problems

Already for standard context-free grammars, many basic decision problems are undecidable, such as testing whether two given grammars generate the

same language (the equivalence problem), or even testing whether a given grammar generates the fixed language $\{a, b\}^*$. A few problems are known to be decidable: for instance, one can test in polynomial time whether a given context-free grammar generates the empty set. In contrast, for conjunctive grammars, there is a uniform undecidability result: for every language L_0 generated by some conjunctive grammar, testing whether a given conjunctive grammar generates L_0 is undecidable [5].

Turning to unambiguous subclasses, the decidability status of the equivalence problem for unambiguous context-free grammars is among the major unsolved questions in formal language theory. On the other hand, as proved by Salomaa and Soittola [16, Thm. 5.5], testing whether a given unambiguous context-free grammar generates 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 the main decision problems for unambiguous conjunctive grammars over a unary alphabet. The underlying idea is the same as in the previous results for ambiguous conjunctive grammars [5]: the language of computation histories of a Turing machine 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 notations of these numbers are represented by a conjunctive grammar [5]. However, Theorems 1–2 proved in this paper for the unambiguous case are more restricted than the known constructions of ambiguous conjunctive grammars [5], 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 language

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

over the alphabet Ω is the language of valid accepting computations of T. 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 linear context-free languages, and hence recognized by a trellis automaton [10].

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 9. *It is undecidable to determine whether a given unambiguous conjunctive grammar over a unary alphabet generates $\varnothing$.*

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 cardinality of this alphabet, let $c = \max(5, d+2)$ and $k = 2c + 2d - 3$; renaming the symbols of this alphabet 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 generating 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 10. *It is undecidable to determine whether a given unambiguous conjunctive grammar over a unary alphabet generates a^*.*

Proof. The proof elaborates that of Lemma 9.

Let M 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 $\ell \geqslant 1$, beginning with f. Consider the language $L \subseteq \{e, f\}^*$ defined by

$$L = \big(f\{e, f\}^* \setminus L_0\big) \cup \big(L_0 \setminus g(\mathrm{VALC}(T))\big),$$

where

$$L_0 = \big(g(\Omega)\big)^*.$$

Since linear conjunctive languages are closed under codes 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 generating 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. Then its image $g(x) \in f\{e, f\}^*$ is not in L. Let $w \in \Sigma_{16}^*$ 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 9) and the equality to a^* (Lemma 10), one can extend these results to the problem of equality to any fixed language, as follows.

Theorem 4. *For every alphabet Σ and for every language $L_0 \subseteq \Sigma^*$ generated by an unambiguous conjunctive grammar, it is undecidable whether a given unambiguous conjunctive grammar generates L_0.*

Proof. 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 generates the language L_0. It is claimed that then one can decide whether a given unambiguous conjunctive grammar G generates 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 the grammar G' that generates $L_0 \cup w \cdot L(G)$. 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 9.

If L_0 is **infinite**, then, again, suppose that one can decide whether a grammar generates L_0. The claim is that there is an algorithm for testing whether a given unambiguous conjunctive grammar G over a one-letter alphabet generates the language a^*. For a given G, construct a related grammar $\widetilde{G} = (\Sigma, N, P, S)$ over the alphabet Σ, which generates the language $L(\widetilde{G}) = \{w \mid w \in \Sigma^*,\ a^{|w|} \in L(G)\}$. Let $G_0 = (\Sigma, N_0, P_0, S_0)$ be an unambiguous conjunctive grammar generating L_0. Construct a new grammar G' over the alphabet Σ

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

If $L(G) = a^*$, then $L(\widetilde{G}) = \Sigma^*$, hence the rules for A in G' generate the language Σ^*, and 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 generates L_0, this algorithm would answer the question of whether $L(G)$ is a^*, which is undecidable by Lemma 10. □

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), and in a "dense" unary language (Theorem 2). Though these are rather restricted representations, as

compared to those constructed earlier for ambiguous conjunctive grammars over the unary alphabet [5], they were sufficient to establish uniform undecidability results for the problem of testing equivalence to a fixed language (Theorem 4).

The paper leaves the following key question unsettled: do there exist any *inherently ambiguous* conjunctive languages, that is, those generated only by ambiguous grammars? In particular, can any such examples be found in the domain of unary languages?

Finally, the undecidability results of Section 7 bring to mind a long-standing open problem in formal language theory: is the equivalence problem for unambiguous context-free grammars decidable? Already for the subclass of unambiguous linear context-free grammars, its decidability status is unknown.

References

[1] J. Autebert, J. Berstel, L. Boasson, "Context-free languages and push-down automata", in: Rozenberg, Salomaa (Eds.), *Handbook of Formal Languages*, Vol. 1, Springer-Verlag, 1997, 111–174.

[2] K. Culik II, J. Gruska, A. Salomaa, "Systolic trellis automata", I and II, *International Journal of Computer Mathematics*, 15 (1984), 195–212; 16 (1984), 3–22.

[3] O. H. Ibarra, S. M. Kim, "Characterizations and computational complexity of systolic trellis automata", *Theoretical Computer Science*, 29 (1984), 123–153.

[4] A. Jeż, "Conjunctive grammars can generate non-regular unary languages", *International Journal of Foundations of Computer Science*, 19:3 (2008), 597–615.

[5] A. Jeż, A. Okhotin, "Conjunctive grammars over a unary alphabet: undecidability and unbounded growth", *Theory of Computing Systems*, 46:1 (2010), 27–58.

[6] A. Jeż, A. Okhotin, "Complexity of equations over sets of natural numbers", *Theory of Computing Systems*, 48:2 (2011), 319–342.

[7] A. Jeż, A. Okhotin, "On the computational completeness of equations over sets of natural numbers" *35th International Colloquium on Automata, Languages and Programming* (ICALP 2008, Reykjavik, Iceland, July 7–11, 2008), 63–74.

[8] M. Kunc, "What do we know about language equations?", *Developments in Language Theory* (DLT 2007, Turku, Finland, July 3–6, 2007), LNCS 4588, 23–27.

[9] A. Okhotin, "Conjunctive grammars", *Journal of Automata, Languages and Combinatorics*, 6:4 (2001), 519–535.

[10] A. Okhotin, "On the equivalence of linear conjunctive grammars to trellis automata", *Informatique Théorique et Applications*, 38:1 (2004), 69–88.

[11] A. Okhotin, "Unambiguous Boolean grammars", *Information and Computation*, 206 (2008), 1234–1247.

[12] A. Okhotin, "Decision problems for language equations", *Journal of Computer and System Sciences*, 76:3–4 (2010), 251–266.

[13] A. Okhotin, "Fast parsing for Boolean grammars: a generalization of Valiant's algorithm", *Developments in Language Theory* (DLT 2010, London, Ontario, Canada, August 17–20, 2010), LNCS 6224, 340–351.

[14] A. Okhotin, C. Reitwießner, "Parsing unary Boolean grammars using online convolution", *Advances and Applications of Automata on Words and Trees* (Dagstuhl seminar 10501, 12–17 December 2010).

[15] A. Okhotin, P. Rondogiannis, "On the expressive power of univariate equations over sets of natural numbers", *Information and Computation*, 212 (2012), 1–14.

[16] A. Salomaa, M. Soittola, *Automata-Theoretic Aspects of Formal Power Series*, Springer-Verlag, 1978.

[17] V. Terrier, "On real-time one-way cellular array", *Theoretical Computer Science*, 141 (1995), 331–335.

Turku
Centre *for*
Computer
Science

Lemminkäisenkatu 14 A, 20520 Turku, Finland | www.tucs.fi

University of Turku
- Department of Information Technology
- Department of Mathematical Sciences

Åbo Akademi University
- Department of Computer Science
- Institute for Advanced Management Systems Research

Turku School of Economics and Business Administration
- Institute of Information Systems Sciences

ISBN 978-952-12-2733-2
ISSN 1239-1891