Artur Jeż

Conjunctive grammars can generate non-regular unary languages

Turku Centre *for* Computer Science

TUCS Technical Report
No 842, October 2007

Conjunctive grammars can generate non-regular unary languages

Artur Jeż
Institute of Computer Science, University of Wrocław
ul. Joliot-Curie 15, 50–383 Wrocław, Poland
aje@ii.uni.wroc.pl

TUCS Technical Report
No 842, October 2007

Abstract

Conjunctive grammars, introduced by Okhotin, extend context-free grammars by an additional operation of intersection in the body of any production of the grammar. Several theorems and algorithms for context-free grammars generalize to the conjunctive case. Okhotin posed nine open problems concerning those grammars. One of them was a question, whether a conjunctive grammar over unary alphabet can generate only regular languages. We give a negative answer, contrary to the conjectured positive one, by constructing a conjunctive grammar for the language $\{a^{4^n} : n \in \mathbb{N}\}$. We also generalize this result: for every set of natural numbers L we show that $\{a^n : n \in L\}$ is a conjunctive unary language, whenever the set of representations in base-k system of elements of L is regular, for arbitrary k. We also investigate the unambiguity of the constructed grammars.

Keywords: Language equations, conjunctive grammars, unary languages, regular languages

TUCS Laboratory
Discrete Mathematics for Information Technology

1 Introduction

1.1 Background

Okhotin [1] introduced conjunctive grammars as a simple and powerful extension of context-free grammars. Informally speaking, conjunctive grammars allow intersections in the body of any rule of the grammar. More formally, conjunctive grammar is a quadruple $\langle \Sigma, N, P, S \rangle$ where Σ is a finite alphabet, N is a set of nonterminal symbols, $S \in N$ is a starting symbol and P is a set of productions of the form:

$$A \to \alpha_1 \& \alpha_2 \& \ldots \& \alpha_k, \quad \text{where } \alpha_i \in (\Sigma \cup N)^* \,. \tag{1}$$

Word w is derived by rule (1) if and only if (iff) it is derived from every string α_i for $i = 1, \ldots, k$, and $\alpha_i = N_1 \ldots N_l$ derives word w if $w = w_1 \ldots w_l$ and N_j derives word w_j for $j = 1, \ldots, l$.

We can also give semantics of conjunctive grammars with resolved language equations that use sum, intersection and concatenation. Language generated by conjunctive grammar is a component of the least solution of such equations.

The usage of intersection allows us to define many natural languages that are not context-free. On the other hand [1] conjunctive languages are computationally easy, that is they are in class $DTIME(n^3) \cap DSPACE(n)$.

We give an example of a conjunctive grammar and its language equations here and a formal definition in Section 2. For detailed results on conjunctive grammars see Okhotin [1], for shorter overview [3]. Work on the Boolean grammars [4], which extend conjunctive grammars by use of negation, is also suggested.

Example 1. *Let us consider conjunctive grammar* $\langle \Sigma, N, P, S \rangle$ *with* $\Sigma = \{a, b, c\}$, $N = \{S, B, C, E, A\}$. *The rules, corresponding language equations and their least solutions are as follows:*

$S \to (AE) \& (BC)$	$L_S = (L_A L_E) \cap (L_B L_C)$	$\{a^n b^n c^n : n \in \mathbb{N}\}$,
$A \to aA \mid \epsilon$	$L_A = \{a\} L_A \cup \{\epsilon\}$	a^*,
$B \to aBb \mid \epsilon$	$L_B = \{a\} L_B \{b\} \cup \{\epsilon\}$	$\{a^n b^n : n \in \mathbb{N}\}$,
$C \to Cc \mid \epsilon$	$L_C = \{c\} L_C \cup \{\epsilon\}$	c^*,
$E \to bEc \mid \epsilon$	$L_E = \{b\} L_E \{c\} \cup \{\epsilon\}$	$\{b^n c^n : n \in \mathbb{N}\}$.

Conjunctive grammars inherit many natural techniques and properties of context-free grammars: existence of the Chomsky normal form, parsing using a modification of CYK algorithm *etc.* On the other hand there is no Pumping Lemma for conjunctive grammars, they do not have bounded growth property, non-emptiness is undecidable. No technique for showing that a language is not conjunctive is known. We are not even capable of separating conjunctive languages from context-sensitive languages.

Okhotin [5] gathered nine most important open problems for conjunctive and Boolean grammars. One of them was a question, whether unary conjunctive languages are always regular. As this holds for context-free grammars, the same result was conjectured for conjunctive grammars. We disprove this conjecture by giving conjunctive grammar for a language $\{a^{4^n} : n \in \mathbb{N}\}$.

The set $\{4^n : n \in \mathbb{N}\}$ written in binary is a regular language. This leads to a natural question, what is the relation between regular (over arbitrary base-k alphabet) languages and unary conjunctive languages. We prove that every regular language (written in some base-k system) interpreted as a set of numbers can be represented by a conjunctive grammar over a unary alphabet.

The conjunctive grammar is unambiguous if, informally speaking, for every nonterminal A deriving word w there is exactly one rule for A that derives w and for each concatenation $N_1 \dots N_i$ that derives a word w there is a unique factorization $w = w_1 \dots w_i$ such that N_j derives w_j. We investigate the unambiguity of the constructed grammars and prove, that for some subclasses of the languages the grammars can be improved to unambiguous ones. We fail to give an answer in case of all constructed grammars.

1.2 Outline of the paper

Section 2 contains the definition of conjunctive grammars, their semantics and facts about language equations. In Section 3 we present a conjunctive grammar for language $\{a^{4^n} : n \in \mathbb{N}\}$, as an example for our technique. In Section 4 we consider the question of the number of nonterminals required to generate non-regular language. In Section 5 we give the main result—for any set of natural numbers L, such that the representation in base-k system of L is regular, language $\{a^n : n \in L\}$ is a conjunctive unary language. In Section 6 we investigate the unambiguity of the constructed grammars. We prove that for languages $\{a^n : n \in ij0^*\}$ there is an unambiguous grammar for them. We fail to give an answer for n in arbitrary regular set. In Section 7 we summarize the results and state open problems.

2 Definitions and Notation

2.1 Conjunctive grammars

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

$$A \to \alpha_1 \& \dots \& \alpha_n \quad (\textit{where } A \in N,\ n \geqslant 1 \textit{ and } \alpha_1, \dots, \alpha_n \in (\Sigma \cup N)^*) \quad (2)$$

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

There are many ways of defining the semantics of conjunctive grammar, here we choose the formalism of language equations:

Definition 2 (Okhotin [2])**.** *For every conjunctive grammar* $\langle \Sigma, N, P, S \rangle$*, the* associated system of language equations *is a system of equations in variables* N*, in which variables assume values of languages over* Σ*, and which contains an equation:*

$$A = \bigcup_{A \to \alpha_1 \& \ldots \& \alpha_m \in P} \bigcap_{i=1}^{m} \alpha_i$$

for every variable A*. Symbols* $a \in \Sigma$ *in such a system define a language* $\{a\}$*, while empty strings denote a language* $\{\epsilon\}$*. A solution of such a system is a vector of languages* $(\ldots, L_C, \ldots)_{C \in N}$*, such that the substitution of* L_C *for* C*, for all* $C \in N$*, turns each equation form the system into an equality.*

Fact 1. *For every system of language equations of the form*

$$A = \bigcup_{i} \bigcap_{j=1}^{m} \alpha_{i,j} \quad \text{, where } \alpha_{i,j} \in (A \cup \Sigma)^* \tag{3}$$

with variables $A \in N$ *there exists a conjunctive grammar* G *with nonterminals* N*, such that the given system is a system of language equations associated with* G*.*

In the rest of the papers we use language equations of the form (3) instead of conjunctive grammars.

2.2 Language equations

We gather the basic definitions and facts about language equations in this section.

Definition 3. *Let* $(X_1, \ldots, X_n)$ *be language variables. A resolved system of language equations is a system of a form*

$$X_i = \varphi_i(X_1, \ldots, X_n) \quad \text{for } i = 1, \ldots, n \ .$$

We abbreviate such systems into vector form $(\ldots, X_i, \ldots) = \varphi(\ldots, X_j, \ldots)$. We identify system of resolved language equations with its operator φ and say about solution of an operator. Since solutions of such systems are vectors of languages we write $(\ldots, A_i, \ldots) \subseteq (\ldots, B_i, \ldots)$, meaning, that $A_i \subseteq B_i$ for $i = 1, \ldots, n$.

Language operation θ is *monotone* if

$$(\ldots, X_i, \ldots) \subseteq (\ldots, Y_i, \ldots) \quad \text{implies} \quad \theta(\ldots, X_i, \ldots) \subseteq \theta(\ldots, Y_i, \ldots) \ .$$

Language operation θ is *continuous* if for converging sequence of vectors of sets

$$(\ldots, L^{(n)}{}_i, \ldots) \to (\ldots, L_i, \ldots) \quad \text{implies} \quad \theta((\ldots, L^{(n)}{}_i, \ldots)) \to \theta((\ldots, L_i, \ldots)) \; .$$

For example intersection, sum and concatenation are continuous and monotone, complementation is continuous but not monotone.

Fact 2. *Every resolved system of equations using only monotone and continuous operations has the least (with respect to $\subset$) solution $(\ldots, S_i, \ldots)$ given by*

$$(\ldots, S_i, \ldots) = \bigcup_{i=0}^{\infty} \varphi^i(\ldots, \emptyset, \ldots) \; .$$

We use a following alternative characterization of the least solution:

Lemma 1. *Let φ be an operator using only monotone and continuous operations. Let $(\ldots, S_i, \ldots)$ be its least solution. If a vector of languages $(\ldots, X_i, \ldots)$ satisfies*

$$\varphi(\ldots, X_i, \ldots) \subseteq (\ldots, X_i, \ldots) \subseteq (\ldots, S_i, \ldots)$$

then

$$(\ldots, X_i, \ldots) = (\ldots, S_i, \ldots) \; .$$

Proof. Clearly

$$(\ldots, \emptyset, \ldots) \subseteq (\ldots, X_i, \ldots) \subseteq (\ldots, S_i, \ldots) \; .$$

Since φ is monotone:

$$\varphi^k(\ldots, \emptyset, \ldots) \subseteq \varphi^k(\ldots, X_i, \ldots) \subseteq \varphi^k(\ldots, S_i, \ldots) \; .$$

As φ is continuous:

$$\bigcup_{k=0}^{\infty} \varphi^k(\ldots, \emptyset, \ldots) \subseteq \bigcup_{k=0}^{\infty} \varphi^k(\ldots, X_i, \ldots) \subseteq \bigcup_{k=0}^{\infty} \varphi^k(\ldots, S_i, \ldots) \; .$$

Since $\varphi(\ldots, X_i, \ldots) \subseteq (\ldots, X_i, \ldots)$ and by the definition of $(\ldots, S_i, \ldots)$ we obtain

$$(\ldots, S_i, \ldots) \subseteq (\ldots, X_i, \ldots) \subseteq (\ldots, S_i, \ldots) \; .$$

□

Note, that language equations emerging from conjunctive grammars use only monotone and continuous operations.

2.3 Unary languages and sets of natural numbers

In case of unary alphabet we identify word a^n with number n and work with sets of integers rather than with sets of words. The allowed operations are (set-theoretical) sum, intersection and 'concatenation', which interpreted in terms of numbers is an addition:

$$XY := \{x + y : x \in X, y \in Y\} .$$

Still we use words 'grammar' and 'language', as this is the main interest of this paper. In many technical proofs we are interested in the multiset of non-zero digits in some base-k notation of natural numbers (for fixed k). Therefore we introduce notation of $\mathrm{Dig}_k(n)$—the multiset of non-zero digit of n and $\Sigma\mathrm{Dig}_k(n)$ for the sum of those digits. If the value of k is clear for the context we sometimes omit it. We use Dig and ΣDig for sets of natural numbers with obvious meaning.

3 Toy Example

Let us define the following sets of integers:

$$A_i = \{1 \cdot 4^n : n \in \mathbb{N}\}, \quad \text{for } i = 1, 2, 3 \qquad A_{12} = \{6 \cdot 4^n : n \in \mathbb{N}\}$$

The indices reflect the fact that these sets consists of numbers that written in base-4 positional system begin with digits $1, 2, 3, 12$, respectively and have only 0'es afterwards. We show that those sets are the least solution of the equations:

$$\begin{aligned} B_1 &= (B_2B_2 \cap B_1B_3) \cup \{1\} , && (4)\\ B_2 &= (B_{12}B_2 \cap B_1B_1) \cup \{2\} , && (5)\\ B_3 &= (B_{12}B_{12} \cap B_1B_2) \cup \{3\} , && (6)\\ B_{12} &= (B_3B_3 \cap B_1B_2) . && (7) \end{aligned}$$

Note, that none of those sets is regular.

Lemma 2. *Every solution (S_1, S_2, S_3, S_{12}) of (4)–(7) satisfies:*

$$(A_1, A_2, A_3, A_{12}) \subseteq (S_1, S_2, S_3, S_{12}).$$

Proof. We prove by induction on m, that $m \in A_i$ implies $m \in S_i$ for $i = 1, 2, 3, 12$.

For $m = 1, 2, 3$ we know that $m \in S_m$ by (4), (5) and (6).

Induction step: let us start with $m = 4^{n+1} \in A_1$. By induction assumption $2 \cdot 4^n \in S_2$ and hence $(2 \cdot 4^n) + (2 \cdot 4^n) = 4^{n+1} \in S_2S_2$. Also by induction

assumption $4^n \in S_1$ and $3 \cdot 4^n \in S_3$, hence $4^n + (3 \cdot 4^n) = 4^{n+1} \in S_1 S_3$, and so $4^{n+1} \in S_2 S_2 \cap S_1 S_3$ and by (4) we conclude that $4^{n+1} \in S_1$.

For $m = 6 \cdot 4^n$ by induction assumption $3 \cdot 4^n \in S_3$, $2 \cdot 4^n \in S_2$ and $1 \cdot 4^{n+1} = 4 \cdot 4^n \in S_1$. Hence $6 \cdot 4^n \in S_3 S_3 \cap S_1 S_2$ and by (7) we get $6 \cdot 4^n \in S_{12}$.

For $m = 2 \cdot 4^{n+1}$ note that $2 \cdot 4^n \in S_2$, $6 \cdot 4^n \in S_{12}$ and $1 \cdot 4^{n+1} \in S_1$ hence $2 \cdot 4^{n+1} \in S_1 S_1 \cap S_{12} S_2$ and by (5) $2 \cdot 4^{n+1} \in S_2$.

For $m = 3 \cdot 4^{n+1}$ notice that $2 \cdot 4^{n+1} \in S_2$, $6 \cdot 4^n \in S_{12}$ and $1 \cdot 4^{n+1} \in S_1$ hence $3 \cdot 4^{n+1} \in S_{12} S_{12} \cap S_1 S_2$ and by (6) $3 \cdot 4^{n+1} \in S_3$. This ends induction step. □

Lemma 3. *Sets* (A_1, A_2, A_3, A_{12}) *satisfy*

$$A_1 \supseteq (A_2 A_2 \cap A_1 A_3) \cup \{1\} , \tag{8}$$

$$A_2 \supseteq (A_{12} A_2 \cap A_1 A_1) \cup \{2\} , \tag{9}$$

$$A_3 \supseteq (A_{12} A_{12} \cap A_1 A_2) \cup \{3\} , \tag{10}$$

$$A_{12} \supseteq (A_3 A_3 \cap A_1 A_2) . \tag{11}$$

Proof. Consider first (8). Let m belong to the right-hand side of (8). If $m = 1$ then the thesis is obvious. So consider $m \in A_2 A_2 \cap A_1 A_3$. Hence there are numbers $k, l \in A_2$ and $m = k + l$. Then either $k = l$ and $\mathrm{Dig}(m) = \{1\}$, hence $m \in A_1$ or $k \neq l$ and so $\mathrm{Dig}(m) = \{2, 2\}$. On the other hand $m \in A_1 A_3$, so there are $k' \in A_1$, $l' \in A_3$ such that $m = l' + k'$. And so either $\mathrm{Dig}(m) = \{1\}$, if $l' = 3k'$ and hence $m \in A_1$, or $\mathrm{Dig}(m) = \{1, 3\}$. But this is a contradiction with a claim that $\mathrm{Dig}(m) = \{2, 2\}$.

Consider (9). Let m belong to the right-hand side of (9). If $m = 2$ then the thesis is obvious. So consider $m \in A_{12} A_2 \cap A_1 A_1$. Then $m \in A_1 A_1$. There are numbers $k, l \in A_1$ and $m = k + l$. Note, that $\mathrm{Dig}(k) + \mathrm{Dig}(l) = 2$. On the other hand $m \in A_{12} A_2$, so there are $k' \in A_{12}$, $l' \in A_2$ such that $m = l' + k'$. Here $\mathrm{Dig}(l') + \mathrm{Dig}(k') = 5$. As the equation $k + l = k' + l'$ holds, there is a carry in $k' + l'$. And this implies $k' = 3l'$ and so $k' + l' \in A_2$.

Consider (10). Let m belong to the right-hand side of (10). If $m = 3$ then the thesis is obvious. So consider $m \in A_{12} A_{12} \cap A_1 A_2$. Then $m \in A_{12} A_{12}$ and so there are $k, l \in A_{12}$ and $m = k + l$. Note that $\mathrm{Dig}(k) + \mathrm{Dig}(l) = 6$. On the other hand $m \in A_1 A_2$, so there are $k' \in A_1$, $l' \in A_2$ such that $m = l' + k'$. Here $\mathrm{Dig}(k') + \mathrm{Dig}(l') = 3$ and so since $k + l = k' + l'$ there is a carry in $k + l$, but this is possible only when $k = l$ and clearly $k + l \in A_3$.

Consider (11). Let m belong to the right-hand side of (11), that is $m \in A_3 A_3 \cap A_1 A_2$. Then $m \in A_3 A_3$. There are numbers $k, l \in A_3$ such that $m = k + l$. Hence $\mathrm{Dig}(k) + \mathrm{Dig}(l) = 6$. On the other hand $m \in A_1 A_2$, so there are $k' \in A_1$, $l' \in A_2$ such that $m = l' + k'$. Here $\mathrm{Dig}(k') + \mathrm{Dig}(l') = 3$ and so since $k + l = k' + l'$ there is a carry in $k + l$, but this is possible only when $k = l$ and clearly $k + l \in A_{12}$. □

Theorem 1. *Sets* A_1, A_2, A_3, A_{12} *are the least solution of* (4)–(7).

Proof. By Lemma 1 it is enough to show that (A_1, A_2, A_3, A_{12}) are included in every solution and that $\varphi(A_1, A_2, A_3, A_{12}) \subseteq (A_1, A_2, A_3, A_{12})$. The former was shown in Lemma 2 and the latter in Lemma 3. □

4 Number of Nonterminals Required

The grammar described in the previous section uses four nonterminals. It is easily converted to Chomsky normal form—we introduce two new nonterminals for languages $\{1\}$ and $\{2\}$, respectively. Hence grammar for language $\{4^n : n \in \mathbb{N}\}$ in Chomsky normal form requires at most six nonterminals. It is an interesting question, which mechanisms of conjunctive grammars and how many of them are required to generate a non-regular language? How many nonterminals are required? How many of them must generate non-regular languages? How many intersections are needed? Putting this question in the other direction, are there any natural sufficient conditions for a conjunctive grammar to generate regular language?

We are able to reduce the number of nonterminals to three, but we sacrifice Chomsky normal form and introduce also concatenations of three nonterminals in productions. This can be seen as trade-off between number of nonterminals and length of concatenations. Consider language equations:

$$B_1 = (B_{2,12}B_{2,12} \cap B_1B_3) \cup \{1\} , \tag{12}$$

$$B_{2,12} = \Big((B_{2,12}B_{2,12} \cap B_1B_1) \cup \{2\}\Big) \cup \Big((B_3B_3 \cap B_{2,12}B_{2,12})\Big) , \tag{13}$$

$$B_3 = (B_{2,12}B_{2,12} \cap B_1B_1B_1) \cup \{3\} . \tag{14}$$

These are basically the same equations as (4)–(7), except that variables B_2 and B_{12} are identified (or merged) and B_2B_1 in (6) was changed to $B_1B_1B_1$.

Theorem 2. *The least solution of* (12)–(14) *is* $(A_1,\ A_2 \cup A_{12},\ A_3)$.

Proof. The proof is a slight modification of the proof of Theorem 1. The main idea is to think of nonterminal $B_{2,12}$ that corresponds to the set $A_2 \cup A_{12}$ as two nonterminals: B_2 and B_{12}, corresponding to sets A_2 and A_{12}, respectively.

Firstly it is easy to check that replacing B_2B_1 with $B_1B_1B_1$ in (6) requires only small changes of proofs of Lemma 2 and Lemma 3.

Let $(S_1, S_{2,12}, S_3)$ be the least solution of (12)–(14). We want to show, that

$$(A_1, A_2 \cup A_{12}, A_3) \subseteq (S_1, S_{2,12}, S_3) ,$$

in analogy to Lemma 2. It is enough to show, that $S_{2,12}$ is a superset of both A_2 and A_{12}. But this is obvious—in the equations we have replaced every occurrence of B_1 and B_{12} by $B_{2,12}$. In terms of grammar this is exactly adding new productions $B_2 \to B_{12}$ and $B_{12} \to B_2$. Clearly adding productions cannot decrease the generated language.

The more interesting part is showing, that:

$$A_1 \supseteq \Big((A_2 \cup A_{12})(A_2 \cup A_{12}) \cap A_1 A_3\Big) \cup \{1\} \,, \tag{15}$$

$$A_2 \cup A_{12} \supseteq \Big(\big((A_2 \cup A_{12})(A_2 \cup A_{12}) \cap A_1 A_1\big) \cup \{2\}\Big) \cup \tag{16}$$

$$\cup \Big(\big(A_3 A_3 \cap (A_2 \cup A_{12})(A_2 \cup A_{12})\big)\Big) \,, \tag{17}$$

$$A_3 \supseteq \Big((A_2 \cup A_{12})(A_2 \cup A_{12}) \cap A_1 A_1 A_1\Big) \cup \{3\} \,. \tag{18}$$

We show even stronger claim, that is

$$A_1 \supseteq \Big((A_2 \cup A_{12})(A_2 \cup A_{12}) \cap A_1 A_3\Big) \cup \{1\} \,, \tag{19}$$

$$A_2 \supseteq \Big((A_2 \cup A_{12})(A_2 \cup A_{12}) \cap A_1 A_1\Big) \cup \{2\} \,, \tag{20}$$

$$A_{12} \supseteq A_3 A_3 \cap (A_2 \cup A_{12})(A_2 \cup A_{12}) \,, \tag{21}$$

$$A_3 \supseteq \Big((A_2 \cup A_{12})(A_2 \cup A_{12}) \cap A_1 A_1 A_1\Big) \cup \{3\} \,. \tag{22}$$

These equations are similar to (8)–(11), apart that on the right-hand side each A_2 and A_{12} was replaced by $A_2 \cup A_{12}$. We show, that each $A_2 \cup A_{12}$ can be replaced back by exactly one A_2 or A_{12} and keep the value of the right-hand side constant. We use the fact, that if $n + m = n' + m'$ then $\mathrm{Dig}_4(n) + \mathrm{Dig}_4(m) \equiv_3 \mathrm{Dig}_4(n') + \mathrm{Dig}_4(m')$.

Consider (19):

$$\mathrm{Dig}(A_1) + \mathrm{Dig}(A_3) = \{4\} \quad \text{and} \quad \mathrm{Dig}(A_2 \cup A_{12}) + \mathrm{Dig}(A_2 \cup A_{12}) = \{4, 5, 6\} \,,$$

with 4 only for two choices of A_2. So we remove the A_{12} from the equations.

Consider (20):

$$\Sigma\mathrm{Dig}(A_1 A_1) = \{2\} \quad \text{and} \quad \mathrm{Dig}(A_2 \cup A_{12}) + \mathrm{Dig}(A_2 \cup A_{12}) = \{4, 5, 6\} \,,$$

with 5 only for choices of A_2 and A_{12}. So we replace $(A_2 \cup A_{12})(A_2 \cup A_{12})$ by $A_2 A_{12}$.

Consider (21):

$$\mathrm{Dig}(A_3) + \mathrm{Dig}(A_3) = \{6\} \quad \text{and} \quad \mathrm{Dig}(A_2 \cup A_{12}) + \mathrm{Dig}(A_2 \cup A_{12}) = \{4, 5, 6\} \,,$$

with 6 only for two choices of A_{12}. So we remove the A_2 from the equations.

Consider (22):

$$\Sigma\mathrm{Dig}(A_1 A_1 A_1) = \{3\} \quad \text{and} \quad \mathrm{Dig}(A_2 \cup A_{12}) + \mathrm{Dig}(A_2 \cup A_{12}) = \{4, 5, 6\} \,,$$

with 6 only for two choices of A_{12}. So we remove the A_2 from the equations.

Hence (19)–(22) follow from the Lemma 3. By Lemma 1 the theorem follows. □

5 Languages Regular in Base-k Notation

We give major generalization of the Theorem 1. Let $\Sigma_k = \{0, \ldots, k-1\}$. We deal with languages $\{a^n : n \in L\}$, where L is some regular subset of Σ_k^*. From the following on we consider regular languages over Σ_k for some k that do not have words with leading 0, since this is meaningless in case of numbers. Still, note that since for regular language R language obtained by removing the leading 0'es is regular as well, hence we loose nothing by this assumption.

Definition 4. *Let $w \in \Sigma_k^*$ be a word. We define its unary representation as*

$$f_k(w) = \{a^n : w \text{ read as base-}k \text{ number is } n\} .$$

We also use f_k applied to languages with an obvious meaning.

Fact 3. *For every $k = l^n$, $n > 0$ and every unary language L language $f_k^{-1}(L)$ is regular iff language $f_l^{-1}(L)$ is regular.*

In the following we use 'big enough' k, say $k \geq 100$. We claim, that for regular L language $f_k(L)$ is unary conjunctive.

As in this section we use positional notation extensively, it is convenient to think of number as string of digits. Hence we will write $n = ijw$, meaning that w begins with digit i, then digit j follows and then some string of digits w.

5.1 Languages with two leading digits fixed

Theorem 3. *For every natural number k, and every $i \in \{1, \ldots, k-1\}$ there is a conjunctive grammar over unary alphabet generating language*

$$\{i \cdot k^n : n \in \mathbb{N}\} ,$$

for every $i, j \in \{1, \ldots, k-1\}$ there is a conjunctive grammar over unary alphabet generating language

$$\{(ki + j) \cdot k^n : n \in \mathbb{N}\} .$$

Proof. For $k > 5$ we introduce variables $B_{i,j}$, where $i = 1, \ldots k-1$ and $j = 0, \ldots, k-1$, with intention that $B_{i,j}$ defines language of numbers beginning with digits i, j and then only zeroes in base-k positional system. We show that sets

$$L_{i,j} = \{(k \cdot i + j) \cdot k^n : n \in \mathbb{N}\} \quad \text{for } j \neq 0 \quad L_{i,0} = \{i \cdot k^n : n \in \mathbb{N}\} .$$

are the least solution of the system

$$B_{1,j} = \bigcap_{n=1}^{2} B_{k-n,0}B_{j+n,0} \cup \{1 : j = 0\} \qquad \text{for } j = 0, 1, 2 \tag{23}$$

$$B_{i,j} = \bigcap_{n=1}^{2} B_{i-1,k-n}B_{j+n,0} \cup \{i : j = 0\} \qquad \text{for } j = 0, 1, 2\ i > 1 \tag{24}$$

$$B_{i,j} = \bigcap_{n=1}^{2} B_{i,j-n}B_{n,0} \cap B_{i,0}B_{j,0} \qquad \text{for } j > 2 \tag{25}$$

For $k = 2, \ldots, 5$ we have to sum up some languages generated in cases of $k = 8, 9, 25$, respectively. The case of $k = 1$ is trivial.

The proof follows by Lemma 4 and Lemma 5. □

Lemma 4. *Every solution* $(\ldots, S_{i,j}, \ldots)$ *of equations* (23)–(25) *satisfies*

$$(\ldots, S_{i,j}, \ldots) \supseteq (\ldots, L_{i,j}, \ldots) .$$

Proof. We proceed on induction on n, proving that $n \in L_{i,j}$ implies $n \in S_{i,j}$. For $n < k$ the thesis is clear by the second summand in (23) and (24).

Let $n = ij0^m \geq k$. If $i = 1$ and $j < 3$ then we use (23). By induction assumption $(k-1)0^m \in S_{k-1,0}$ and $(j+1)0^m \in S_{j+1,0}$. Adding:

$$(k-1)0^m + (j+1)0^m = 1j0^m \in S_{k-1,0}S_{j+1,0} .$$

By induction assumption $(k-2)0^m \in S_{k-2,0}$ and $(j+2)0^m \in S_{j+2,0}$. Adding:

$$(k-2)0^m + (j+2)0^m = 1j0^m \in S_{k-2,0}S_{j+2,0} .$$

Hence $1j0^m \in S_{1,j}$, by (23).

The second case, for $i > 1$ and $j < 3$ is similar—we use (24). By induction assumption $(i-1)(k-1)0^m \in S_{i-1,k-1}$ and $(j+1)0^m \in S_{j+1,0}$. Adding:

$$(i-1)(k-1)0^m + (j+1)0^m = ij0^m \in S_{i-1,k-1}S_{j+1,0} .$$

By induction assumption $(i-1)(k-2)0^m \in S_{i-1,k-2}$ and $(j+2)0^m \in S_{j+2,0}$. Adding:

$$(i-1)(k-2)0^m + (j+2)0^m = ij0^m \in S_{i-1,k-2}S_{j+2,0} .$$

Hence $ij0^m \in S_{i,j}$, by (24).

The last case, for $j > 2$—we use (25). By induction assumption $i(j-1)0^m \in S_{i,j-1}$ and $10^m \in S_{1,0}$. Adding:

$$i(j-1)0^m + 10^m = ij0^m \in S_{i,j-1}S_{1,0} .$$

By induction assumption $i(j-2)0^m \in S_{i,j-2}$ and $20^m \in S_{2,0}$. Adding

$$i(j-2)0^m + 20^m = ij0^m \in S_{i,j-2}S_{2,0} \ .$$

By induction assumption $i00^m \in S_{i,0}$ and $j0^m \in S_{j,0}$. Adding:

$$i00^m + j0^m = ij0^m \in S_{i,0}S_{j,0} \ .$$

Hence $ij0^m \in S_{i,j}$, by (25). □

Lemma 5. *Languages* $(\ldots, L_{i,j}, \ldots)$ *satisfy* $\varphi(\ldots, L_{i,j}, \ldots) \subseteq (\ldots, L_{i,j}, \ldots)$.

Proof. Consider first (23). Let $n \in L_{k-1,0}$, $n' \in L_{j+1,0}$ and $m \in L_{k-2,0}$, $m' \in L_{j+2,0}$ such that $n+n' = m+m'$. If there is a carry in one of the sums $n+n$ or $m+m'$ then the result belongs to the $L_{1,j}$ and we are done. So we consider the case, when ther is no carry in both sums. Hence $\mathrm{Dig}(n+n') = \{j+1, k-1\}$ and $\mathrm{Dig}(m+m') = \{j+2, k-2\}$. Since $j+1 < j+2 \leqslant k-2 < k-1$ we conclude that $n+n' \neq m+m'$.

Consider now (24). Let $n \in L_{i-1,k-1}$, $n' \in L_{j+1,0}$ and $m \in L_{i-1,k-2}$, $m' \in L_{j+2,0}$ such that $n+n' = m+m'$. Since

$$\Sigma\mathrm{Dig}(n) + \Sigma\mathrm{Dig}(n') = i+j+k-1 = \Sigma\mathrm{Dig}(m) + \Sigma\mathrm{Dig}(m')$$

there is either no carry of digits in both sums or there is a carry in both sums (in both cases it is not possible to have two carries). If there is no carry and we have three non-zero digits in both sums then $\mathrm{Dig}(n+n') = \{i-1, k-1, j+1\}$ and $\mathrm{Dig}(m+m') = \{i-1, k-2, j+2\}$. They must equal and so $\{k-1, j+1\} = \{k-2, j+2\}$, which is not possible, since $j+1 < j+2 \leq k-2 < k-1$. If there is no carry and there are two non-zero digits then in $n+n'$ the leading digit is $i+j$ and in $m+m'$ it is $i+j+1$, contradiction. If a carry in one of the sums $n+n$ or $m+m'$ in the 'desired position', that is in $n+n'$ digit $(j+1)$ adds up with $(k-1)$ or in $m+m'$ digit $(j+2)$ adds up with $(k-2)$ then the result is as desired. And so we have to deal with the case, when both carries are in different position. But this is possible only when $(j+1)$ adds up with $(i-1)$ in $n+n'$ and $(j+2)$ adds up with $(i-1)$ in $m+m'$. Then in $n+n'$ the second digit is $i+j-k$ and n $m+m'$ the digit is $i+j+1-k$, contradiction.

Consider (25). Let $n \in L_{i,j-1}$, $n' \in L_{1,0}$ and $m \in L_{i,j-2}$, $m' \in L_{2,0}$ and $p \in L_{i,0}$, $p' \in L_{j,0}$ such that $n+n' = m+m' = p+p'$. Again

$$\Sigma\mathrm{Dig}(n+n') = \Sigma\mathrm{Dig}(m+m') = \Sigma\mathrm{Dig}(p+p')$$

and so either there is no carry in all of the sums or exactly one carry in each sum (clearly $p+p'$ cannot have two carries). Since digits $i, j-1, 1$ from $n+n'$ are all non-zero, then $n+n'$ has at least two non-zero digits. Since $p+p'$ has at most two non-zero digits, then there are exactly two non-zero digits in the result. Suppose that there was a carry and we ended up with

two non-zero digits. Then $p+p'$ has 1 as its first digit and $(i+j-k)$ as the second an then only 0'es. On the other hand $n+n'$ has 1 as the first digit, 0 as the second and $j-1>0$ as the third, contradiction. And so there is no carry. If in at least one sum the adding is as desired, then we end up with a good result. If no adding is as desired, then $\mathrm{Dig}(n+n')=\{i+1,j-1\}$, $\mathrm{Dig}(m+m')=\{i+2,j-2\}$ and $\mathrm{Dig}(p+p')=\{i,j\}$. Contradiction. $\square$

5.2 Any regular language

We now define the resolved language equations for fixed regular language $L\subseteq\Sigma_k^*\backslash 0\Sigma_k^*$. Let $M=\langle\Sigma_k,Q,\delta,F,q_0\rangle$ be the (non-deterministic) automaton recognizing L^r. The set of variables is

$$N=\{A_{i,j,q},A_{i,j}:1\le i<k,\ 0\le j<k,\ q\in Q\}\cup\{S\}\ .$$

The intended solution is

$$L(A_{i,j})=\{n\ :\ f_k^{-1}(n)=ij0^\ell \text{ for some natural } \ell\}\ , \tag{26}$$
$$L(A_{i,j,q})=\{n:f_k^{-1}(n)=ijw,\ \delta(q_0,w^r,q)\}\ , \tag{27}$$
$$L(S)=f_k(L)\ . \tag{28}$$

We denote sets defined in (27) by $L_{i,j,q}$ and sets defined by (26) by $L_{i,j}$.

By Theorem 3 sets $L_{i,j}$ can be defined by resolved language equations, and so we focus on equations for $A_{i,j,q}$. In following equations triples (x,q,q') satisfy $\delta(q',x,q)$.

$$A_{i,j,q}=\bigcap_{n=0}^{3}A_{i,n}A_{j-n,x,q'}\cup\{ij:q_0=q\}\qquad\text{for } j>3,\ i\neq 0 \tag{29}$$
$$A_{i,j,q}=\bigcap_{n=1}^{4}A_{i-1,j+n}A_{k-n,x,q'}\cup\{ij:q_0=q\}\qquad\text{for } j<4,\ i\neq 0,1 \tag{30}$$
$$A_{1,j,q}=\bigcap_{n=1}^{4}A_{k-n,0}A_{j+n,x,q'}\cup\{1j:q_0=q\}\qquad\text{for } j<4 \tag{31}$$
$$S=(L\cap\Sigma_k)\cup\bigcup_{q,i,j:\ \delta(q,ji)\cap F\neq\emptyset}A_{i,j,q} \tag{32}$$

We prove that $(\ldots,L_{i,j,q},\ldots)$ is the least solution of (29)–(31). The case of $L(S)$ in (32) then follows.

Lemma 6. *Every solution* $(\ldots,X_{i,j,q},\ldots)$ *of* (29)–(31) *satisfies*

$$(\ldots,L_{i,j,q},\ldots)\subseteq(\ldots,X_{i,j,q},\ldots)\ .$$

Proof. We prove by induction that for every $n > 1$ that $n \in L_{i,j,q}$ implies $n \in X_{i,j,q}$. When $n = ij$ then this is obvious by the last summand in (29)–(30).

Induction step. Let $n = ijw$ and $w = xw'$. Let p be a state such that $\delta(q_0, w'^r, p)$ and $\delta(p, x, q)$.

Suppose $j > 3$, by induction assumption:

$$jxw' \in X_{j,x,p}\ , \qquad (j-1)xw' \in X_{j-1,x,p}\ ,$$
$$(j-2)xw' \in X_{j-2,x,p}\ , \qquad (j-3)xw' \in X_{j-3,x,p}\ .$$

Adding $i00^{|w'|+1}$, $i10^{|w'|+1}$, $i20^{|w'|+1}$, $i30^{|w'|+1}$, respectively, gives ijw in all cases, and so by (29) $ijw \in X_{i,j,q}$.

Suppose $j < 4$ and $i > 1$, by induction assumption:

$$(k-1)xw' \in X_{k-1,x,p}\ , \qquad (k-2)xw' \in X_{k-2,x,p}\ ,$$
$$(k-3)xw' \in X_{k-3,x,p}\ , \qquad (k-4)xw' \in X_{k-4,x,p}\ .$$

Adding $(i-1)(j+1)0^{|w'|+1}$, $(i-1)(j+2)0^{|w'|+1}$, $(i-1)(j+3)0^{|w'|+1}$, $(i-1)(j+4)0^{|w'|+1}$, respectively, gives ijw in all cases, and so by (30) $ijw \in X_{i,j,q}$.

Suppose $j < 4$ and $i = 1$, by induction assumption:

$$(j+1)xw' \in X_{j+1,x,p}\ , \qquad (j+2)xw' \in X_{j+2,x,p}\ ,$$
$$(j+3)xw' \in X_{j+3,x,p}\ , \qquad (j+4)xw' \in X_{j+4,x,p}\ .$$

Adding $(k-1)0^{|w'|+1}$, $(k-2)0^{|w'|+1}$, $(k-3)0^{|w'|+1}$, $(k-4)0^{|w'|+1}$, respectively, gives $1jw$ in all cases, and so by (31) $1jw \in X_{1,j,q}$. □

Lemma 7. *Languages $L_{i,j,q}$ satisfy $\varphi(\ldots, L_{i,j,q}, \ldots) \subseteq (\ldots, L_{i,j,q}, \ldots)$.*

Proof. We proceed by induction on number of digits in n. We first proof, that it in fact has the desired two first digits. Then we deal with the q index.

We begin with (29). Suppose w belongs to the right-hand side. We show that it also belongs to the left-hand side. Consider the possible positions of the two first digits of each summand. Notice, that if j is on the position one to the right of i, then the two first digits are ij. And so we may exclude this case from our consideration. The Table 1 summarizes the results, it has some drawbacks:

- some digits sum up to k or more and influence another digit by a carry,
- in the second column i may be or may be not on the same position as x, but we deal with those two cases together,
- in the second column there may be an add up to k somewhere to the right, and hence we can add 1 to x.

	i and j are together	j is leading	i is leading
$A_{i,0}A_{j,xq'}$	$(i+j), x$	$j, x\langle+i\rangle$	$i, 0$
$A_{i,1}A_{j-1,xq'}$	$(i+j-1), (x+1)$	$(j-1), x\langle+i\rangle$	$i, 1$
$A_{i,2}A_{j-2,xq'}$	$(i+j-2), (x+2)$	$(j-2), x\langle+i\rangle$	$i, 2$
$A_{i,3}A_{j-3,xq'}$	$(i+j-3), (x+3)$	$(j-3), x\langle+i\rangle$	$i, 3$

Table 1: The possible leading digits of numbers resulting from adding in (29)

This possibilities in the second point were marked in the table by writing $\langle+i\rangle$.

If we want the intersection to be non-empty we have to choose four items from the Table 1, each in a different row. We show, that this is not possible. We say that some choices *fit*, if the digits included in the table are the same in those choices.

First of all, no two elements in the third column fit. They have fixed leading digits and they clearly are different.

Suppose that we choose two elements from the first column. We show that if in one of them $(i+j-z)$ sums up to k (or more) then the same thing happens in the second choice. If $i+j-z \geq k$ (perhaps by additional 1 carried from the previous position) then the first digit is 1. In the second element the first digit can be 1 (if there is a carry of 1) or at least $i+j-z'$, but the latter is not possible, since $i+j-z' > 1$. In both cases it is not possible to fit the digits on the column with x: they all are on the same position and are different, since there can be no carry from previous position.

It is not possible to choose three elements from the second column: as previously we may argue, that either in all of them in the leading digit (that is with $j-z$ for some z) there is an adding to k and a carry to the (newly created) position or in all of them there is no adding to k in the leading position.

Suppose there is a carry in a leading position. Since $j < k$ then this is possible only for the first row. In particular it is not possible to have two choices when there is a carry in the leading position. Suppose they do not add up. Since there are three fitting choices, in one of them we must increase the value of the second digit by at least 2. But the maximal value carried from the previous position is 1. Contradiction. As a consequence if there are two fitting choices then the first digit is in range $(j-1, j)$.

And so if there are four fitting choices, then exactly one of them is in the first column, one in the third column and two in the middle column. The third column always begins with i. In the first column the leading digit is at least $i+j-3 > i$ or it is 1. Hence $i = 1$. And so the choices in the second column begin with 1 as well. Hence $j < 3$, which is a contradiction.

We move to (30). The Table 2 summarizes the possible first two digits: as before we may argue, that if there are some fitting entries in some column

	i and $k-z$ are together	$k-z$ is leading	$i-1$ is leading
$A_{i-1,j+1}A_{k-1,x,q'}$	$(k+i-2),(1+j+x)$	$(k-1),x\langle+i\rangle$	$i-1,j+1$
$A_{i-1,j+2}A_{k-2,x,q'}$	$(k+i-3),(2+j+x)$	$(k-2),x\langle+i\rangle$	$i-1,j+2$
$A_{i-1,j+3}A_{k-3,x,q'}$	$(k+i-4),(3+j+x)$	$(k-3),x\langle+i\rangle$	$i-1,j+3$
$A_{i-1,j+4}A_{k-4,x,q'}$	$(k+i-5),(4+j+x)$	$(k-4),x\langle+i\rangle$	$i-1,j+4$

Table 2: The possible leading digits of numbers resulting from adding in (30)

	$k-z$ is leading	$j+z$ is leading
$A_{k-1,0}A_{j+1,x,q'}$	$(k-1),0\langle+j+1\rangle$	$(j+1),x\langle+k-1\rangle$
$A_{k-2,0}A_{j+2,x,q'}$	$(k-2),0\langle+j+2\rangle$	$(j+2),x\langle+k-2\rangle$
$A_{k-3,0}A_{j+3,x,q'}$	$(k-3),0\langle+j+3\rangle$	$(j+3),x\langle+k-3\rangle$
$A_{k-4,0}A_{j+4,x,q'}$	$(k-4),0\langle+j+4\rangle$	$(j+4),x\langle+k-4\rangle$

Table 3: The possible leading digits of numbers resulting from adding in (31)

then on their leading position digits sum up to k in all choices or in all choices they do not sum up to k.

We cannot have two choices from the third column (the second digits do not match). We can have at most two from the second column (to obtain three we would have to carry at least 2 to the first digit in one of them and this is not possible).

For the same reason there can be at most two choices from the first column. But if there are two choices from the first column then we cannot match the positions with x. Hence there is at the most one choice from the first column.

And so we have one choice from the first column, one from the third and two from the second. Since the third and the second column match, then $i \geqslant k-3$. But in such a case in the first column we have at least $k+k-3-5>k$ and so the leading digit is 1. Contradiction.

Consider the last possibility, the (31).The Table 3 summarizes the possible first two digits: in the first column there are no two fitting choices—since in the second digit in this column is at most $j+4<k$ and there is no carry to the first digit. And clearly the first digits are different. So we have to choose at least one element from the second column. Its leading digit is at most $j+4+2$—carry of more than 2 from the previous position is not possible. But $j+6<k-4$ and hence the first digit do not match. Contradiction.

We now take the indices denoting states of the automaton into our consideration. Consider equation (29) and some w belonging to the right-hand side. We have already proved that $w=ijw'$. If $w'=\epsilon$ then it clearly belongs to the right hand side by the last summand. Else consider $A_{i,0}A_{j,x,q'}$ and $jxw'' \in L_{j,x,q'}$ that was used in derivation of w. Note that $w'=xw''$. By definition of $L_{j,x,q'}$ we obtain $\delta(q_0,w''^r,q')$ and by definition of the (29) we

obtain $\delta(q', x, q)$, and so $\delta(q_0, w''^r x, q)$. But $w = ijxw''$, therefore it belongs to the left-hand side.

Consider equation (30) and some w belonging to the right-hand side. We have already proved that $w = ijw'$. If $w' = \epsilon$ then it clearly belongs to the right hand side by the last summand. Else consider $(k-1)xw'' \in L_{k-1,x,q'}$ that was used in derivation of w. Note that $w' = xw''$. By definition of $L_{k-1,x,q'}$ we obtain $\delta(q_0, w''^r, q')$ and by definition of the (30) we obtain $\delta(q', x, q)$, and so $\delta(q_0, w''^r x, q)$. But $w = ijxw''$, therefore it belongs to the left-hand side.

Consider equation (31) and some w belonging to the right-hand side. We have already proved that $w = 1jw'$. If $w' = \epsilon$ then it clearly belongs to the right hand side by the last summand. Else consider $A_{k-1,0}A_{j+1,x,q'}$. Consider $(j+1)xw'' \in L_{j+1,x,q'}$ that was used in derivation of w. Note that $w' = xw''$. By definition of $L_{j+1,x,q'}$ we obtain $\delta(q_0, w''^r, q')$ and by definition of the (31) we obtain $\delta(q', x, q)$, and so $\delta(q_0, w''^r x, q)$. But $w = 1jxw''$, therefore it belongs to the left-hand side. □

Finally we conclude with.

Theorem 4. *For every natural $k > 1$ and every regular $L \subseteq \Sigma_k^*$ language $f_k(L)$ is a conjunctive unary language.*

Proof. First note that language $L' = L \setminus 0\Sigma_k^*$ is regular as well and $f_k(L) = f_k(L')$, hence without losing generality we may assume that $L \subset \Sigma_k^* \setminus 0\Sigma_k^*$.

By Lemma 3 it is enough to consider 'big' k, say $k > 100$.

Lemma 6 and Lemma 7 assume that $k > 100$ and they guarantee that the sets $(\ldots, L_{i,j,q_i}, \ldots)$ fulfill the assumptions of Lemma 1. Now the only thing left is to see that (32) defines S properly. If $w \in L$ then either $|w| = 1$ and hence $w \in S \cap \Sigma_k$ or $|w| \geq 2$ and hence $w = ijw'$. Let q be such that $\delta(q_0, w'^r, q) \quad \text{and} \quad \delta(q, ji) \in F$. Clearly such state exists, since $w \in L$ and so automata recognizing L^r has some intermediate state q. But this means that $w \in A_{i,j,q}$ and $S \supset A_{i,j,q}$. □

6 Unambiguity of grammars

We say that conjunctive grammar $\langle \Sigma, N, P, S \rangle$ is *unambiguous* if:

1. For every nonterminal A and word w there is at most one production for A that can generate w.

2. For every conjunct $\alpha_i = \beta_1 \ldots \beta_n$ in production $A \to \alpha_1 \& \alpha_2 \& \ldots \& \alpha_k$ there is at most one factorization of word $w = w_1 w_2 \ldots w_n$ such that β_j generates w_j for $j = 1, \ldots, n$.

We will prove, that the construction of the languages $ij0^*$ can in fact be done by an unambiguous grammar. To do this we should slightly complicate the construction of (23)–(25).

Theorem 5. *For every $k > 1$ there is an unambiguous conjunctive grammar over the unary alphabet generating language*

$$\{i \cdot k^n : n \in \mathbb{N}\} ,$$

for every $i, j \in \{1, \ldots, k-1\}$ there is an unambiguous conjunctive grammar over unary alphabet generating language

$$\{(ki + j) \cdot k^n : n \in \mathbb{N}\} .$$

Proof. First of all we assume that k is substantially large, that is $k \geqslant 9$. For such k we define the rules.

$$B_{1,j} = \bigcap_{n=1}^{2} B_{k-1,0} B_{j+n,0} \cup \{1 : j = 0\} \qquad \text{for } 0 \leqslant j \leqslant 4 \tag{33}$$

$$B_{i,j} = \bigcap_{n=1}^{2} B_{i-1,k-n} B_{j+n,0} \cup \{i : j = 0\} \qquad \text{for } 0 \leqslant j \leqslant 4,\ i > 1 \tag{34}$$

$$B_{i,j} = \bigcap_{n=1}^{2} B_{i,j-n} B_{n,0} \qquad \text{for } j > 4 \tag{35}$$

For smaller k one simply has to sum up the languages generated by systems of language equations for big enough power of k, say k^4. Such summing clearly preserves unambiguity.

Checking that the presented grammar generates the desired languages is done in the same way as in Theorem 3 and related Lemmata and therefore is omitted. What we want to show that this grammar is in fact unambiguous. First of all for different rules the set of produced words is clearly different: only for B_{i0} there are two rules: one of them produces i, in the other the result is at least $i0$, and so they are disjoint. So the other condition to check is whether for any concatenation of nonterminals XY and any word w produced by this concatenation there is exactly one factorization of w into parts produced by X and Y.

We begin with $B_{k-1,0}B_{j+1,0}$ from (33). Let n be the produced number and let if be represented as $n_1 + n_2$. Asking for unambiguity is the question of uniqueness of this representation for given n. If $\Sigma\mathrm{Dig}(n) = j + 1$ then there was a carry and this is possible only when n_1 and n_2 have the same amount of digits. And so we know the length of n_1 and n_2 and since $n_1 \neq n_2$ the representation is unique. If the $\Sigma\mathrm{Dig}(n) = j + k - 1$ then there was no carry. This is possible only when n_1 and n_2 have different lengths. Since $j + 1 < k - 1$ then we know the unique decomposition, as there are exactly two non-zero digits in n. The same analysis can be carried for $B_{k-2,0}B_{j+2,0}$ from (33).

Consider now $B_{i-1,k-1}B_{j+1,0}$ from (34), and again the representation of $n = n_1 + n_2$. If $\Sigma\mathrm{Dig}(n) = i + j + k - 1$ and n has three non-zero digits

then there was no adding of digits and no carry. And there is exactly one position on which digit $k-1$ stands, as $k-1 > i-1, j+1$. The digit to the left is $i-1$. The only remaining comes from $j+1$. And so we can read the unique representation from n. If $\Sigma\mathrm{Dig}(n) = i+j+k-1$ and n has exactly two non-zero digits, then $i-1, j+1$ have added (and there was no carry). Hence there are only two non-zero digit, the one to the right is $k-1$ and the one to the left is $i+j$. And so we again obtained a unique representation. If $\Sigma\mathrm{Dig}(n) = i+j$ then there was a carry. There are two possible carries: either we obtain leading digits ij (when $j+1$ was added to the $k-1$) or we obtain $1(i+j-k)(k-1)$. But these are clearly different, as $i \neq 1$. The sum of digits $i+j-k+1$ is not possible, as this would require two carries. The same analysis can be carried for $B_{i-1,k-2}B_{j+2,0}$ from (34).

Consider now $B_{i,j-1}B_{1,0}$ from (35) and $n = n_1 + n_2$. If $\Sigma\mathrm{Dig}(n) = i+j-k+1$ then the only possibility was 1 added to i. And hence the last non-zero digit is $j-1 > 0$. And so we know the positions of digits and know how to decompose (since $j-1 > 0$ then clearly $n_1 \neq n_2$). If $\Sigma\mathrm{Dig}(n) = i+j$ then there was no carry. If there are just two non-zero digits then they are leading digits and are either $(i+1)(j-1)$, when 1 adds to i, or ij when 1 adds to $j-1$. And those are different, hence the representation is unique. If there are three digits then they are $1, i, j-1$. Either the rightmost is 1, in which case it comes from n_2 or it is greater than 1, in which case it comes from n_1. In this case we also know how to decompose. The same argument applies to $B_{i,j-2}B_{2,0}$ from (35). □

7 Conclusions and open problems

The main result of this paper is an example of a conjunctive grammar over unary alphabet generating non-regular language. This grammar has six nonterminal symbols in Chomsky normal form. Number of nonterminals could be reduced to three if we consider a grammar that is not in a Chomsky normal form. It remains an open question, how many nonterminals, intersection *etc.* are required to generate a non-regular language. In particular, can we give natural sufficient conditions for a conjunctive grammar to generate a regular language? Also, no non-trivial algorithm for recognizing conjunctive languages over unary alphabet is known. An obvious modification of the CYK algorithm requires quadratic time and linear space. Can those bounds be lowered? Closure under complementation of conjunctive languages (both in general and and in case of unary alphabet) remains unknown, with conjectured negative answer.

The second important result is a generalization of the previous one: for every regular language $R \subseteq \{0, ..., k-1\}^*$ treated as set of k-ary numbers language $\{a^n : \exists w \in R \ w \text{ read as a number is } n\}$ is a conjunctive unary language. Also for some restricted class, that is languages of the form $\{a^n :$

$\exists w \in ij0^*$ w read as a number is $n\}$ we have constructed an unambiguous conjunctive unary grammar, therefore showing that unambiguity is not an obstacle for non-regularity of a conjunctive unary grammar. Still we do not know whether some larger class of languages can be defined by unambiguous unary conjunctive grammars.

Acknowledgments.

The author would like to thank Tomasz Jurdziński and Krzysztof Loryś for introducing to the subject and helpful discussion, Sebastian Bala and Marcin Bieńkowski for helpful discussion and Alexander Okhotin, who suggested the study of generalization to unary representations of all regular languages (Theorem 4) and the unambiguity questions.

Part of this research was done during the visit of the author in Turku in June 2007, supported by short visit grant from the European Science Foundation under project AutoMathA, reference number 1763.

Research supported by MNiSW grant number N206 024 31/3826, 2006-2008.

References

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

[2] Okhotin, A. Conjunctive grammars and systems of language equations. *Programming and Computer Software.*, 28:243–249, 2002.

[3] Okhotin, A. An overview of conjunctive grammars. *Formal Language Theory Column. Bulletin of the EATCS*, 79:145–163, 2003.

[4] Okhotin, A. Boolean grammars. *Information and Computation*, 194:1:19–48, 2004.

[5] Okhotin, A. Nine open problems on conjunctive and boolean grammars. *TUCS Technical Report*, 794, 2006.

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 978-952-12-1961-0
ISSN 1239-1891