One-nonterminal conjunctive grammars over a unary alphabet

Artur Jeż[1] * and Alexander Okhotin[2,3] **

[1] Institute of Computer Science, University of Wrocław, Poland aje@ii.uni.wroc.pl
[2] Academy of Finland
[3] Department of Mathematics, University of Turku, Finland
alexander.okhotin@utu.fi

Abstract. It is shown that equations $X = \varphi(X)$, in which the unknown X is a set of natural numbers and φ uses operations of union, intersection and addition $S + T = \{ m + n \mid m \in S, n \in T\}$, can simulate systems of equations $X_i = \varphi_i(X_1, \ldots, X_n)$ with $1 \leqslant i \leqslant n$, in the sense that the solution of a system is encoded in the solution of an equation. This implies undecidability of some properties of one-nonterminal conjunctive grammars over a unary alphabet.
Key-words: language equations, conjunctive grammars, decision problems.

1 Introduction

This paper is concerned with systems of equations, in which the unknowns are sets of natural numbers, while the left- and right-hand sides use Boolean operations, as well as element-wise addition of sets defined as $S + T = \{m + n \mid m \in S,\ n \in T\}$. On one hand, such equations can be regarded as a generalization of *integer expressions*, introduced in the seminal paper by Stockmeyer and Meyer [13] and later systematically studied by McKenzie and Wagner [8]. On the other hand, these equations are a particular case of *language equations* defined over a unary (one-letter) alphabet.

Language equations, which have formal languages as unknowns, have recently received much attention [7]. Their most well-known kind are systems of the form

$$\begin{cases} X_1 = \varphi_1(X_1, \ldots, X_n) \\ \quad \vdots \\ X_n = \varphi_n(X_1, \ldots, X_n) \end{cases} \tag{*}$$

in which the right-hand sides φ_i may contain union and concatenation of languages, as well as singleton constants. These equations, first proposed by Ginsburg and Rice [1], provide the most natural semantics for context-free grammars.

* Supported by MNiSW grants N206 024 31/3826 2006–2008 and N206 259035 2008–2010.
** Supported by the Academy of Finland under grant 118540.

If intersection is further allowed, then systems (*) represent *conjunctive grammars*, which are a natural extension of the context-free grammars introduced and studied by Okhotin [9,10].

The expressive power of conjunctive grammars over a unary alphabet has been realised only recently, once Jeż [3] constructed a grammar for the nonregular language $\{a^{4^n} \mid n \geqslant 0\}$. This grammar can be equally regarded as a system of four equations over sets of numbers, using union, intersection and addition, and one of the questions raised by Jeż [3] was how many variables are necessary to obtain any non-periodic solution. This question was answered by Okhotin and Rondogiannis [11], who constructed a single univariate equation $X = \varphi(X)$ with a non-periodic solution, as well as presented a class of sets of numbers that are not representable by any such equations.

This paper generalizes the construction of Okhotin and Rondogiannis [11]. It will be shown that for every unary conjunctive grammar, the languages generated by all of its nonterminal symbols can be encoded together in a single unary language generated by a one-nonterminal conjunctive grammar. This construction implies that some undecidability and complexity results for unary conjunctive grammars due to Jeż and Okhotin [4,5] hold already for one-variable grammars.

Then consequences for decision problems for one-nonterminal unary conjunctive grammars are studied. Several decidable and undecidable problems are identified. In general, the complexity of such problems may differ from the complexity in the case of unary conjunctive grammars with many non-terminals.

2 Conjunctive grammars and systems of equations

Conjunctive grammars generalize context-free grammars by allowing an explicit conjunction operations in the rules.

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*

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

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

The semantics of conjunctive grammars may be defined 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, and the ***.

Definition 2 ([10]). *Let $G = (\Sigma, N, P, S)$ be a conjunctive grammar. The associated system of language equations is the following system in variables N:*

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

Let $(\ldots, L_A, \ldots)$ *be its least solution and denote* $L_G(A) := L_A$ *for each* $A \in N$. *Define* $L(G) := L_G(S)$.

The existence of a least solution with respect to componentwise inclusion follows from the basic fixpoint theory. Moreover, it can be easily shown *** cite ???* that for a large class of systems of language equations there is a unique ε-free solution: a system of language equations is *strict* if there is no production into ε and no chain dependency. A strict system of language equations has a unique ε-free solution. *** cite? ***

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

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

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

The question of whether conjunctive grammars can generate any non-regular unary languages has been an open problem for some years [10], until recently solved by Jeż [3], who constructed a grammar for the language $\{a^{4^n} \mid n \geqslant 0\}$. Let us formulate this grammar as the following resolved system of four equations over sets of numbers:

Example 1 (Jeż [3]). The system

$$\begin{cases} X_1 = \big((X_2 + X_2) \cap (X_1 + X_3)\big) \cup \{1\} \\ X_2 = \big((X_6 + X_2) \cap (X_1 + X_1)\big) \cup \{2\} \\ X_3 = \big((X_6 + X_6) \cap (X_1 + X_2)\big) \cup \{3\} \\ X_6 = \big((X_3 + X_3) \cap (X_1 + X_2)\big) \end{cases}$$

has least solution $X_i = \{i \cdot 4^n \mid n \geqslant 0\}$, for $i = 1, 2, 3, 6$.

Sets of this kind can be conveniently specified by regular expressions for the corresponding sets of base-k notations of numbers, which in this case are $\mathtt{10}^*$, $\mathtt{20}^*$, $\mathtt{30}^*$ and $\mathtt{120}^*$, respectively. In the following, some parentheses in the right-hand sides of equations shall be omitted, and the following default precedence of operations shall be assumed: addition has the highest precedence, followed by intersection, and then by union with the least precedence.

The construction in Example 1 essentially uses all four variables, and there seems to be no apparent way to replicate it using a single variable. However, this was achieved in the following example:

Example 2 (Okhotin, Rondogiannis [11]). The univariate equation

$$X = \big[(X{+}X{+}11) \cap (X{+}X{+}22)\big] \cup \big[(X{+}X{+}1) \cap (X{+}X{+}9)\big] \cup \\ \cup \big[(X{+}X{+}7) \cap (X{+}X{+}12)\big] \cup \big[(X{+}X{+}13) \cap (X{+}X{+}14)\big] \cup \{56, 113, 181\}$$

has the unique solution

$$S = \{4^n - 8 \,|\, n \geqslant 3\} \cup \{2 \cdot 4^n - 15 \,|\, n \geqslant 3\} \cup \{3 \cdot 4^n - 11 \,|\, n \geqslant 3\} \cup \{6 \cdot 4^n - 9 \,|\, n \geqslant 3\}.$$

This equation is actually derived from Example 1, and its solution encodes the values of all four sets in Example 1. Each of the four components in S represents one of the variables in Example 1 with a certain *offset* (8, 15, 11 and 9).

Note that the set from Example 2 is exponentially growing. It is known that unary conjunctive grammars can generate a set that grows faster than any given recursive set:

Proposition 1 (Jeż, Okhotin [4]). *For every recursively enumerable set of natural numbers S there exists a system $X_i = \varphi_i(X_1, \ldots, X_n)$ over sets of natural numbers with the least solution $X_i = S_i$, such that the growth function of S_1 is greater than that of S at any point.*

On the contrary, for univariate equations it has been proved that if a set grows faster than exponentially (for example, $\{n! \,|\, n \geqslant 1\}$), then it is not representable:

Proposition 2 (Okhotin, Rondogiannis [11]). *Let $S = \{n_1, n_2, \ldots, n_i, \ldots\}$ with $0 \leqslant n_1 < n_2 < \ldots < n_i < \ldots$ be an infinite set of numbers, for which $\liminf_{i \to \infty} \frac{n_i}{n_{i+1}} = 0$. Then S is not the least solution of any equation $X = \varphi(X)$.*

However, even though one-nonterminal conjunctive grammars cannot generate *all* unary conjunctive languages, it will now be demonstrated that they can represent a certain encoding of any conjunctive language.

3 One-nonterminal conjunctive grammars

The goal is to simulate an arbitrary conjunctive grammar over $\{a\}$ by a conjunctive grammar with a single nonterminal symbol. The construction formalizes and elaborates the intuitive idea of Example 2, making it provably work for any grammar.

The first step towards the construction is a small refinement of the known normal form for unary conjunctive grammars. It is known that every conjunctive language over every alphabet can be generated by a conjunctive grammar in the *binary normal form*, with all rules of the form $A \to B_1C_1 \& \ldots \& B_nC_n$ with $n \geqslant 1$ or $A \to a$. The following stronger form is required by the below construction.

Lemma 1. *For every conjunctive grammar $G = (\Sigma, N, P, S)$ there exists a conjunctive grammar $G' = (\Sigma, N', P', S')$ generating the same language, in which every rule is of the form $A \to a$ with $a \in \Sigma$ or*

$$A \to B_1C_1\& \ldots \& B_nC_n \quad (\text{with } n \geqslant 2),$$

in which the sets $\{B_1, C_1\}$, …, $\{B_n, C_n\}$ are pairwise disjoint.

Proof. If there is a rule with no intersection, that is, $A \to \alpha$ for some nonterminal A and $\alpha \in (N \cup \Sigma)^*$, it can be replaced by a trivial intersection $A \to \alpha\&\alpha$.

Let m be the greatest number of conjuncts in the rules in P. Define m copies of every nonterminal: $N' = N \times \{1, \ldots, m\}$. Replace every rule

$$A \to B_1C_1\& \ldots \& B_\ell C_\ell$$

with

$$(A, i) \to (B_1, 1)(C_1, 1)\& \ldots \&(B_\ell, \ell)(C_\ell, \ell)$$

For every rule $A \to a$ in the original grammar, define a new rule $(A, i) \to a$. Let $S' = (S, 1)$ be the new start symbol. The resulting grammar generates the same language. □

Theorem 1. *For every unary conjunctive grammar $G = (\{a\}, \{A_1, \ldots, A_m\}, P, A_1)$ of the form given in Lemma 1 there exist numbers $0 < d_1 < \ldots < d_m < p$ and an equation of the form*

$$X = F \cup \bigcup \bigcap (X + X + \{\ell\})$$

over a set of natural numbers X, with a unique solution $S = \bigcup_{i=1}^{m} S_i$, where $S_i = \{np - d_i \mid a^n \in L_G(A_i)\}$.

*(**to be done: improve the form of the equation, mention that F is finite, mention that each ℓ is positive**)*

Furthermore,

1. *The numbers p and $d_1, \ldots, d_m$ depend only on m.*
2. *The size of φ is polynomial in the size of G.*
3. *Each subexpression to which the union is applied generates a subset of some S_i.*

Let $p = 4^{m+2}$ and let $d_i = \frac{p}{4} + 4^i$ for every nonterminal A_i. For every number $t \in \{0, \ldots, p\}$, the set $\{np - t \mid n \geqslant 0\}$ is called *track number t*. The goal of the construction is to represent each set S_i in the track d_i. The rest of the tracks should be empty.

For every rule $A_i \to \alpha$, where $\alpha = A_{j_1}A_{k_1}\& \ldots \&A_{j_\ell}A_{k_\ell}$, consider the following expression over sets of numbers:

$$\varphi_{i,\alpha}(X) = \bigcap_{t=1}^{\ell} X + X + (d_{j_t} + d_{k_t} - d_i).$$

Define the following equation:

$$X = \bigcup_{A_i \to \alpha \in P} \varphi_{i,\alpha}(X) \cup \bigcup_{A_i \to a \in P} \{p - d_i\}$$

Now the task is to prove that the unique solution of this equation is $S = \bigcup_i S_i$, where $S_i = \{np - d_i \mid a^n \in L_G(A_i)\}$.

Each time X appears in the right-hand side of the equation, it is used in the context of an expression $\varphi_{i,\alpha}(X)$. The proof of the theorem is based upon the following property of these expressions.

Lemma 2. *Let $i, j, k, \ell \in \{1, \ldots, m\}$ with $\{i, j\} \cap \{k, \ell\} = \varnothing$. Then*

$$(S + S + d_i + d_j) \cap (S + S + d_k + d_\ell) = (S_i + S_j + d_i + d_j) \cap (S_k + S_\ell + d_k + d_\ell)\,.$$

Proof. As addition is distributive over union and union is distributive over intersection,

$$\begin{aligned}(S + S + d_i + d_j) \cap (S + S + d_k + d_\ell) = \\ = \bigcup_{i',j'} (S_{i'} + S_{j'} + d_i + d_j) \cap \bigcup_{k',j'} (S_{k'} + S_{\ell'} + d_k + d_\ell) = \\ = \bigcup_{i',j',k',j'} (S_{i'} + S_{j'} + d_i + d_j) \cap (S_{k'} + S_{\ell'} + d_k + d_\ell)\end{aligned}$$

It is sufficient to prove that if $\{i', j'\} \neq \{i, j\}$ or $\{k', \ell'\} \neq \{k, \ell\}$, then the intersection is empty. Consider any such intersection

$$\begin{aligned}(S_{i'} + S_{j'} + d_i + d_j) \cap (S_{k'} + S_{\ell'} + d_k + d_\ell) = \\ (\{np \mid a^n \in L(A_{i'})\} - d_{i'} + \{np \mid a^n \in L(A_{j'})\} - d_{j'} + d_i + d_j) \cap \\ (\{np \mid a^n \in L(A_{k'})\} - d_{k'} + \{np \mid a^n \in L(A_{\ell'})\} - d_{\ell'} + d_k + d_\ell)\,,\end{aligned}$$

and suppose it contains any number, which must consequently be equal to $d_i + d_j - d_{i'} - d_{j'}$ modulo p and to $d_k + d_\ell - d_{k'} - d_{\ell'}$ modulo p. As each d_t satisfies $\frac{p}{2} > d_t > \frac{p}{4}$, both offsets are between $-\frac{p}{2}$ and $\frac{p}{2}$, and therefore they must be equal to each other:

$$d_i + d_j - d_{i'} - d_{j'} = d_k + d_\ell - d_{k'} - d_{\ell'}.$$

Equivalently, $d_i + d_j + d_{k'} + d_{\ell'} = d_k + d_\ell + d_{i'} + d_{j'}$, and since each d_t is defined as $\frac{p}{4} + 4^t$, this holds if and only if

$$4^i + 4^j + 4^{k'} + 4^{\ell'} = 4^k + 4^\ell + 4^{i'} + 4^{j'}.$$

Consider the largest of these eight numbers, let its value be d. Without loss of generality, assume that it is on the left-hand side. Then the left-hand side is greater than d. On the other hand, if no number on the right-hand side is d, then the sum is at most $4 \cdot \frac{d}{4} = d$. Thus at least one number on the right-hand side

must be equal to d as well. Removing those two numbers and giving the same argument for the sum of 3, 2 and 1 summands yields that

$$\{d_i, d_j, d_{k'}, d_{\ell'}\} = \{d_k, d_\ell, d_{i'}, d_{j'}\}.$$

Then, by the assumption that $\{i, j\} \cap \{k, \ell\} = \varnothing$,

$$\{d_i, d_j\} = \{d_{i'}, d_{j'}\} \quad \text{and} \quad \{d_{k'}, d_{\ell'}\} = \{d_k, d_\ell\},$$

and since the addition is commutative,

$$i = i', \quad j = j', \quad k = k' \quad \text{and} \quad \ell = \ell'.$$

Therefore,

$$\begin{aligned}(S + S + d_i + d_j) \cap (S + S + d_k + d_\ell) = \\ \bigcup_{i',j',k',j'} (S_{i'} + S_{j'} + d_i + d_j) \cap (S_{k'} + S_{\ell'} + d_k + d_\ell) = \\ (S_i + S_j + d_i + d_j) \cap (S_k + S_\ell + d_k + d_\ell)\,,\end{aligned}$$

which completes the proof.

Proof (Proof of Theorem 1). Let $P = P_1 \cup P_0$, where P_0 contains rules of the form $A_i \to a$, while P_1 consists of multiple-conjunct rules. The equation is strict and thus has a unique solution in the set of positive natural numbers, so it is enough to show that S is a solution, that is,

$$S = \bigcup_{A_i \to \alpha \in P_1} \varphi_{i,\alpha}(S) \cup \bigcup_{A_i \to a \in P_0} \{p - d_i\}$$

Consider each rule $A_i \to \alpha \in P_1$ with $\alpha = A_{j_1} A_{k_1} \& \ldots \& A_{j_t} A_{k_t}$. Then

$$\varphi_{i,\alpha}(S) = \bigcap_{t=1}^{\ell} (d_{j_t} + d_{k_t} - d_i) + S + S = \bigcap_{t=1}^{\ell} (d_{j_t} + d_{k_t} - d_i) + S_{j_t} + S_{k_t}$$

by Lemma 2, and it is easy to calculate that

$$\bigcap_{t=1}^{\ell} (d_{j_t} + d_{k_t} - d_i) + S_{j_t} + S_{k_t} = \{np - d_i \mid a^n \in L(\alpha)\}.$$

Calculating further,

$$\bigcap_{t=1}^{\ell}(d_{j_t}+d_{k_t}-d_i)+S_{j_t}+S_{k_t}=$$

$$\bigcap_{t=1}^{\ell}(d_{j_t}+d_{k_t}-d_i)+\{pn_j-d_{j_t}\mid a^{n_j}\in L(A_{j_t})\}+\{pn_k-d_{k_t}\mid a^{n_k}\in L(A_{k_t})\}=$$

$$\bigcap_{t=1}^{\ell}\{p(n_j+n_k)-d_i\mid a^{n_j}\in L(A_{j_t}),a^{n_k}\in L(A_{k_t})\}=$$

$$\bigcap_{t=1}^{\ell}\{np-d_i\mid a^n\in L(A_{j_t})\cdot L(A_{k_t})\}=$$

$$\{np-d_i\mid a^n\in L(\alpha)\}.$$

Similarly for $A_i\to a\in P_0$,

$$\{p-d_i\}=\{np-d_i\mid a^n\in L(\{a\})\}.$$

Altogether,

$$\bigcup_{A_i\to\alpha\in P_1}\varphi_{i,\alpha}(S)\cup\bigcup_{A_i\to a\in P_0}\{p-d_i\}=$$

$$\bigcup_i\Big(\bigcup_{A_i\to\alpha\in P_1}\varphi_{i,\alpha}(S)\cup\bigcup_{A_i\to a\in P_0}\{p-d_i\}\Big)=$$

$$\bigcup_i\Big(\bigcup_{A_i\to\alpha\in P_1}\{np-d_i\mid a^n\in L(\alpha)\}\cup\{np-d_i\mid a^n\in L(a)\}\Big)=$$

$$\bigcup_i\bigcup_{A_i\to\beta\in P}\{np-d_i\mid a^n\in L(\beta)\}.$$

Since $(\ldots,L(A_i),\ldots)$ is the solution of the associated system of language equations, $L(A_i)=\bigcup_{A_i\to\beta\in P}L(\beta)$, and hence the latter expression equals

$$\bigcup_i\{np-d_i\mid a^n\in L(A_i)\}=\bigcup_i S_i=S,$$

which completes the proof.

Corollary 1. *For every unary conjunctive language $L\subseteq a^+$ there exist numbers $p\geqslant d\geqslant 1$ and a conjunctive grammar $G=(\{a\},\{S\},P,S)$, such that $L(G)\cap(a^p)^*a^{p-d}=\{a^{np-d}\mid a^n\in L\}$.*

Example of this transformation:

Example 3. Consider the four-nonterminal grammar from Example 1. It satisfies the condition in Lemma1, but it is not precisely in the binary normal form, as it contains rules $A_2\to aa$ and $A_3\to aaa$.

However, these rules do not affect the general construction, and one can extend the transformation of Theorem 1 to this grammar. The constants are $p = 4^{m+2} = 4096$, $d_1 = 516$, $d_2 = 520$, $d_3 = 528$ and $d_4 = 544$, and the transformation yields an equation

$$X = \big[(X+X+524) \cap (X+X+528)\big] \cup \big[(X+X+544) \cap (X+X+512)\big] \cup \\ \cup \big[(X+X+560) \cap (X+X+508)\big] \cup \big[(X+X+512) \cap (X+X+492)\big] \cup \\ \cup \{3580, 7672, 11760\}$$

with a unique solution ***.

Note that this equation can be transcribed as a conjunctive grammar

$$S \to a^{524}SS \& a^{528}SS \mid a^{544}SS \& a^{512}SS \mid a^{560}SS \& a^{508}SS \mid a^{512}SS \& a^{492}SS \mid a^{3580} \mid a^{7672} \mid a^{11760}$$

generating the language ***.

4 Complexity of the membership problem

The most fundamental problem for every grammar, in particualr for a conjunctive gramamar is the membership problem, for the grammar G and a word w an answer, whether $w \in L(G)$ is expected. The complexity of this problem may depend on the way word and grammar are encodeed, especially when w is a unary word. We say that unary $w = a^n$ is compressed if it is encoed as a binary string representing n. Also a conjunctive grammar (context-free grammar) with a single non-terminal over a unary alphabet is in *compressed form* if conjuncts in its production are encoded in binary, i.e. $A^k a^\ell$ is encoded as a pair of binary numbers (k, ℓ). Compressed membership problem is a membership problem in which the input word w is compressed and membership problem for grammar in compressed form is a memebership problem in which the gramamar is in compresed form. In the follwoing it is shown that the compression of the word has impact on the complexity of the membership problem in case of the conjunctive grammars with one non-terminal even if the grammr is not compressed. On the contrary, the complexity of membership problem for context-free grammar with a single non-terminal remains in P for a compressed word or for a compressed grammar. On the other hand, this problem is NP-complete when compression of both input word and the grammar is allowed.

The following is known:

Proposition 3 (Jeż, Okhotin [5]). *There exists a EXPTIME-complete set of numbers $S \subseteq \mathbb{N}$, such that the language $L = \{a^n \mid n \in S\}$ of unary notations of numbers from S is generated by a conjunctive grammar.*

The problem stated as "Given a unary conjunctive grammar G and a number n in binary, determine whether $a^n \in L(G)$" is EXPTIME-complete.

To show that both results still hold for one-nonterminal unary conjunctive grammars, it is sufficient to take the grammar generating L and to transform it according to Theorem 1.

Theorem 2. *There exists a EXPTIME-complete set of numbers $S \subseteq \mathbb{N}$, such that the language $L = \{a^n \mid n \in S\}$ is generated by a one-nonterminal conjunctive grammar. The general membership problem for one-nonterminal unary conjunctive grammars with input encoded in binary is EXPTIME-complete.*

Proof. This problem clearly belongs to this complexity class, as it is decidable in exponential time in more general case of systems of such equations [5, Th. 4.1].

Hardness follows from Theorem 1 as follows. It is known [5, Th. 4.1] that there exists a system of equations $X_i = \varphi_i(X_1, \dots X_k)$ with a least solution $(S_1, \dots S_k)$, in which S_1 is an EXPTIME-hard set.

By Theorem 1 one can efficiently construct an equation $Y = \psi(Y)$ with a least solution S and numbers $p, d \geqslant 1$ such that $n \in S_1$ if and only if $pn - d \in S$. Hence the general membership problem for a system of equations polynomially reduces to the general membership problem for a single equation. □

It is natural to ask, what is the complexity of the compressed membership problem for unary context free grammars with one non-terminal — it is known, that in case of many non-terminals it is NP-complete. Does this hold also for one non-terminal? In the following it is shown that the answer to the above question is yes, if we allow the productions of the grammar to be compressed (as well as the input string) and no, if the grammar cannot be compressed.

Theorem 3. *The fully compressed membership problem for one-nonterminal unary compressed CFG is NP-hard.*

Proof. It is known that the compressed membership problem for compressed unary context-free grammars is in NP [2].

The NP-hardness is proved by reduction from the NP-complete Knapsack problem. In its original Karp's formulation the problem is stated as follows: "Given integers $b_1, \dots, b_n$ and z in binary notation, determine whether there exist $c_1, \dots, c_n \in \{0, 1\}$ with $\Sigma_{i=1}^n b_i c_i = z$".

Assume, without loss of generality, that $z > \max_i b_i$

Based on the instance of the Knapsack problem, a grammar G and a string a^ℓ are constructed, such that $a^\ell \in L(G)$ if and only if the given instance of the Knapsack problem is positive.

Let m be the least power of two with $m \geqslant \max\{z, 2^n\} + 2$. Let $d_{i,0} = m^2 + 2^{i-1}$ and $d_{i,1} = m^2 + mb_i + 2^{i-1}$. Construct a context-free grammar G with a set of rules $\{S \to a^{d_{i,0}}, S \to a^{d_{i,1}} \mid i \in \{1, \dots, n\}\} \cup \{S \to S^n\}$ and a string $a^{nm^2+zm+(2^n-1)}$. Clearly the size of G and $a^{nm^2+zm+(2^n-1)}$ is polynomial in the size of the given Knapsack problem.

It is claimed that the numbers $c_1, \dots, c_n \in \{0, 1\}$ with $\Sigma_{i=1}^n b_i c_i = z$ exist if and only if $a^{nm^2+zm+(2^n-1)} \in L(G)$. This will prove the NP-hardness and establish the theorem.

⟸ Suppose that there exist such $c_1, \dots c_n$. Then the string $a^{nm^2+zm+(2^n-1)}$ can be derived as follows:

$$S \Longrightarrow S^n \Longrightarrow a^{d_{1,c_1}} S^{n-1} \Longrightarrow a^{d_{1,c_1}} a^{d_{2,c_2}} S^{n-2} \Longrightarrow \dots \Longrightarrow a^{d_{1,c_1}} \dots a^{d_{n,c_n}},$$

where

$$\sum_{i=1}^{n} d_{i,c_i} = \sum_{i=1}^{n}(m^2 + c_i b_i m + 2^{i-1}) = nm^2 + zm + 2^n - 1 = |w|.$$

$\Leftarrow$ Assume now that the string $a^{nm^2+zm+(2^n-1)}$ can be derived by G. It is claimed that the derivation of $a^{nm^2+zm+(2^n-1)}$ must be of the same general form as in the previous part of the proof. This allows recreating the values c_i out of the choices between $d_{i,0}$ and $d_{i,1}$.

Claim 1. In the derivation of $a^{nm^2+zm+(2^n-1)}$, the production $S \to S^n$ is used exactly once.

Proof. Suppose that the rule $S \to S^n$ was used at least twice. Then there were at least $2n-1$ productions of the form into terminal symbols, each generating string of length greater than m^2. Then the produced string is of length greater than $(2n-1)m^2 = nm^2 + (n-1)m^2 \geqslant nm^2 + zm + 2^n - 1 = |w|$. At the same time, $a^{nm^2+zm+(2^n-1)}$ cannot be derived without using the rule $S \to S^n$, as each $d_{i,0}$ and $d_{i,1}$ is at most $m^2 + (\max_i b_i)m + 2^{n-1} < nm^2 + zm + 2^n - 1$.

Claim 2. In derivation of $a^{nm^2+zm+(2^n-1)}$ after the production $A \to A^n$ for each i exactly one of the productions $S \to a^{d_{i,0}}$ or $S \to a^{d_{i,1}}$ was used.

Proof. By Claim 1 there are exactly n productions into terminal symbols, i.e. $a^{x_1}, \ldots, a^{x_n}$, where for each j holds $x_j \in \{d_{i,0}, d_{i,1} \mid i = 1, \ldots, n\}$. As the string $a^{nm^2+zm+(2^n-1)}$ is derived, we obtain that

$$\sum_{j=1}^{n} x_j = nm^2 + mz + 2^n - 1.$$

Let us look at the numbers $d_{i,0}, d_{i,1}$ and $nm^2 + mz + 2^n - 1$ written in binary. Then on the last n bits $nm^2 + mz + 2^n - 1$ has only 1's and for each i numbers $d_{i,0}$, $d_{i,1}$ have exactly i^{th} bit set to 1 and each other to 0. So each x_j has exactly one non-zero bit among the last n bits. Thus it cannot hold that any two x_j and $x_{j'}$ have the same considered non-zero bit. Hence for each i exactly one of $d_{i,0}$, $d_{i,1}$ is among $\{x_j \mid j = 1, \ldots, n\}$. □

Using Claim 2 c_i's can be defined: $c_i = 0$ if the production $S \to a^{d_{i,0}}$ was used and $c_i = 1$ if the production $S \to a^{d_{i,1}}$. Then $a^{m^2 n + m\Sigma_{i=1}^{n} c_i b_i + 2^n - 1} = a^{mn+zm+2^n-1}$ and hence $\Sigma_{i=1}^{n} c_i b_i = z$. Hence the answer for the instance of the Knapsack problem is YES. □

This result can be compared with complexity of membership problem, when the grammar or the string is not compressed.

Lemma 3. *Let $G = (\{a\}, \{S\}, P, S)$ be a one-nonterminal context-free grammar with m rules and with the longest right-hand side of a rule of length k. Then $L(G)$ is periodic starting from $2k^2m + k$ with a period at most k^2.*

Proof. Let $P = \{S \to a^{\ell_i} S^{k_i}\}_{i=1}^{m}$ with $\ell_i, k_i \geqslant 0$.

Claim 3. For any productions $S \to S^{k_i} a^{\ell_i}$ for $k_i > 0$and $S \to a^{\ell_j}$. Then $p = (k_i - 1)\ell_j + \ell_i \leq 2k^2$ is a period of $L(G)$.

Proof. Given a derivation of a string a^n, one can derive a string of length $n + (k_i - 1)\ell_j + \ell_i$ by first using the production $S \to S^{k_i} a^{\ell_i}$, then substituting each of $k_i - 1$ copies of S by a^{ℓ_j}, thus obtaining $Sa^{(k_i-1)\ell_j+\ell_i}$ and deriving a^n from S.

Hence, for each $0 \leqslant i < p - 1$ it is enough to find the smallest n_i such that $a^{n_i p+i} \in L(G)$ or find out that no such n_i exists. We show that $n_i < \widehat{n}$, where $\widehat{n}$ is polynomial in the size of the grammar G.

We say that a production $S \to S^{k_i} a^{\ell_i}$ is of arity k_i.

Claim 4. Let $\{q_1, \ldots, q_m\}$ be a set of natural numbers. Then there exists a derivation of G in which every i^{th} production $S \to a^{\ell_i} S^{k_i}$ is used exactly q_i times if and only if

$$1 + \sum_i q_i(k_i - 1) = 0$$

Proof. $\Rightarrow$ Rule of arity $k \geqslant 1$ creates $k - 1$ new copies of S, each application of rule of arity 0 destroys one copy S. Also there is one additioanl non-terminal — the starting one. Since after the derivation there is no S left in the string, it holds that

$$1 + \sum_{i:k_i \geqslant 1} q_i(k_i - 1) - \sum_{i:k_i=0} q_i = 0$$

and hence the claim follows.

$\Leftarrow$ Assume the equality holds. Without loosing generality we may assume that the productions of arity 0 are the last one with respect to the enumeration. Then a derivation is constructed as follows:

$$S \Longrightarrow^{q_1} a^{q_1\ell_1} S^{1+q_1(k_1-1)} \Longrightarrow^{q_2} a^{q_1\ell_1+q_2\ell_2} S^{1+q_1(k_1-1)+q_2(k_2-1)} \Longrightarrow$$
$$\Longrightarrow \ldots \Longrightarrow a^{\sum_{i=1}^{m} q_i\ell_i} S^{1+\sum_{i=1}^{m} q_i(k_i-1)} = a^{\sum_{i=1}^{m} q_i\ell_i}$$

with the last equality following from the assumption. Since rules of arity greater than 0 do not decrese the number of copie sof S in the string, then a rule with arity greater than 0 can always be applied. Rules of arity 0 can be applied, as in the end no copy of S remains.

□

Claim 5. Let $a^{np+i} \in L(G)$, fix its derivation. If there is a production $S \to S^k a^\ell$ used at least p times in the derivation and a production $S \to a^{\ell'}$ used at least $(k-1)p$ in the derivation then a string $a^{p(n-(k-1)\ell'-\ell)+i}$ shorter than a^{np+i} can also be derived.

Proof. This follows from the Claim 4 — we can produce a string with p less productions $S \to S^k a^\ell$ and $(k-1)p$ less productions $S \to a^{\ell'}$. □

Let k be the length of the largest concatenation of symbols (terminals and non-terminals) in productions. Consider the $a^{n_i p+i} \in L(G)$ such that n_i is as small as possible. We show that the derivation of $a^{n_i p+i}$ cannot use more than $2kmp + m(p-1)$ productions, and thus the length of $a^{n_i p+i}$ is at most $(2kmp + m(p-1))k$. Suppose it uses more. Then one of the following holds:

1. it uses more than $m(p-1)$ productions of arity 1
2. it uses more than kmp productions of arity at least 2
3. it uses more than kmp productions of arity 0

We show that in each case it cannot be the shortest such string.

If it uses more than $m(p-1)$ productions of arity 1 then one of them is used at least p times and all those p productions can be removed and shorter string derived, by Claim 5.

If it uses more than kmp production of arity 2 or more then at least one production of arity 2 or more is used kp times. By Claim 4 applied to this derivation, there must be at least $kmp + 1$ productions of arity 0, hence one of them appears at least $kp + 1$ times. Then, again, Claim 5 is applicable, and it yields a shorter string, which is a contradiction.

If there are kmp productions of arity 0 then by Claim 4 there are at least mp productions of arity greater than 0, thus one of them occurred at least p times, and, as in the previous case, a contradiction is obtained via Claim 5.

Hence $\widehat{n} \leqslant \frac{(2kmp+m(p-1))k}{p} \leqslant 2k^2m + k$, which is polynomial in the size of the input grammar. □

Theorem 4. *The compressed membership problem for one-nonterminal unary CFGs is in NLOGSPACE.*

Proof. Firstly an algorithm for uncompressed membership problem is given. :

Note that we may assume that there are no ε prodcuctions in the grammar. If there were then we may omit them and construct grammar G' with productions $A \to A^{k'_i} a^{\ell_i}$ whenever $k'_i \geqslant k_i$ and $A \to A^{k_i} a^{\ell_i}$ is a production of the grammar G. Clearly $L(G) = L(G') \cup \{\varepsilon\}$ and this construction can be simulated in LOGSPACE — if we want to check that $A \to A^{k'_i} a^{\ell_i}$ is a production of G' it is enough to find a production in G $A \to A^{k_i} a^{\ell_i}$ for $k_i \geqslant k'_i$. The non-determinsitic algorithm deriving a^n from G' needs only to remember the current number of terminals and non-termianls in the derived string. Moreover, as soon as there are more than n of them we may reject, as clearly the derived word is longer.

For (2), note that while the grammar is compressed, it cannot have productions into very long concatenations of non-terminals or terminals — if the string a^n then concatenations of more than n symbols cannot derive this string. Hence we can remove such long productions and then decompress the description of the grammar — each of the productions is of length n at most, thus the growth of size of the grammar is quadratic at most. Thus the problem was reduced to (1). This reduction can be done in LOGSPACE, so also in this case the problem is in NLOGSPACE.

Case (3) follows from Lemma3 — the following simple algorithm is guaranteed to run in polynomial time:

If there is no termianting rule then reject, as such grammar cannot produce any word. If there are no non-terminating rules then check, whether a^n is one of the word appearing on the right-handside of the produciton. This ends trivial cases.

Take any terminating rule $S \to a^{\ell_j}$ and non-terminating rule $S \to S^{k_i}a^{\ell_i}$. Then $p = (k_i - 1)\ell_j + \ell_i$ is a period and $\widehat{n} \leftarrow 2k^2m + k$ is a periodicity bolund. If $n < p(\widehat{n} + 1)$ then a^n is short and we can check whether $a^n \in L_(G)$ by an algorithm running in time polynomial in n. If $n > p(\widehat{n})$ then let $n = n'p + i$. Since n is larger than the periodicity bound then $a^n \in L(G)$ if and only if $a^{p\widehat{n}+i} \in L(G)$. Thus we can check whether $a^{p\widehat{n}+i} \in L(G)$ by an algorithm running in time polynomial in $\widehat{n}$.

TEST-MEMBERSHIP(G, a^n)

if there is no rule $S \to a^{\ell_i}$
  then return NO
if there is no rule $S \to S^{k_i}a^{\ell_i}$ with $k_i > 0$
  then return, whether $n = \ell_i$ for some i

let $S \to S^{k_i}a^{\ell_i}$ for $k_i > 0$ and $S \to a^{\ell_j}$ be rules of G
let $p \leftarrow (k_i - 1)\ell_j + \ell_i$ be a period
let $\widehat{n} \leftarrow 2k^2m + k$ be a periodicity bolund

if $n < (p + 1)\widehat{n}$
  then return TEST-MEMBERSHIP-SIMPLE(G, a^n), i.e. a^n is short.
if $n \geq (p + 1)\widehat{n}$
  then let $n = n'p + i$ for $0 \leq i < p$
return TEST-MEMBERSHIP-SIMPLE$(G, a^{p\widehat{n}+i})$,
as a^n is longer than periodicity bound.

By Claim 3 p is a period. By Lemma 3 if $n > (p + 1)\widehat{n}$ then it is in the periodic part of the language and hence $a^{n'p+i} \in L(G)$ if and only if $a^{\widehat{n}p+i} \in L(G)$. □

5 Decision problems

(demagogy TBW: the membership problem considered, now some other)

Let us now consider the decidability of basic properties of one-nonterminal unary conjunctive grammars. In the case of multiple nonterminals, most basic problems are undecidable:

Proposition 4 (Jeż, Okhotin [4]). *For every fixed unary conjunctive language $L_0 \subseteq a^*$, the problem of whether a given conjunctive grammar over $\{a\}$ generates the language L_0 is Π_1-complete.*

	uncompressed	compressed	fully compressed
Context-free			
general case	P-complete	PSPACE-complete [12]	n/a
$\Sigma = \{a\}$, any N	P-complete	NP-complete [2]	NP-complete [2]
$\Sigma = \{a\}$, $N = \{S\}$	**in NLOGSPACE**	**in NLOGSPACE**	**NP-complete**
Conjunctive			
general case	P-complete	EXPTIME-complete [5]	n/a
$\Sigma = \{a\}$, any N	P-complete	EXPTIME-complete [5]	EXPTIME-complete [5]
$\Sigma = \{a\}$, $N = \{S\}$	in P	**EXPTIME-complete**	EXPTIME-complete [5]

Table 1. Complexity of general membership problems.

equality to any fixed ultimately periodic set is clearly decidable: substitute, check. but even more:

Theorem 5. *There exists an algorithm, which, given a one-nonterminal conjunctive grammar $G = (\{a\}, \{S\}, P, S)$ over a unary alphabet and a finite automaton M over an alphabet $\Sigma_k = \{0, 1, \ldots, k-1\}$, determines whether $L(G) = \{a^n \mid$ the k-ary notation of n is in $L(M)\}$.*

*** the notation is not satisfactory! *** *** introduce $\boxplus$ here? ***

Lemma 4. *Let A and B be NFAs over an alphabet $\Sigma_k = \{0, 1, \ldots, k-1\}$, with $L(M_1) \cap 0\Sigma_k^* = L(M_2) \cap 0\Sigma_k^* = \varnothing$, let A and B have m and n states, respectively. Then there exists and can be effectively constructed a $(2mn + 2m + 2n + 1)$-state NFA over Σ_k, which recognizes the language $\{$the k-ary notation of $n_1 + n_2 \mid$ the k-ary notation of n_i is in $L(M_i)\}$.*

The same method can be elaborated to characterize equality to any given finite or co-finite language. By this characterization, both problems are clearly decidable. However, the more general problem of equivalence of two grammars is undecidable.

Theorem 6. *The equivalence problem for one-nonterminal unary conjunctive grammars is Π_1-complete.*

Proof. The proof is by reduction from the equivalence problem for unary conjunctive grammars with multiple nonterminals. Two grammars are combined into one, the construction of Theorem 1 is applied, and then the start symbols of the two grammars are exchanged and the construction is applied again. The two resulting one-nonterminal grammars are equivalent if and only if the original grammars generate the same language.

Before approaching the equivalence problem for one-nonterminal conjunctive grammars, let us establish the undecidability of the following technical problem:

Claim 6. The problem of testing whether for a given conjunctive grammar $G = (\{a\}, N, P, S)$ with two designated nonterminals S and S', $L_G(S) = L_G(S')$, is undecidable.

It is known that the problem of whether two unary conjunctive grammars generate the same language is undecidable. Let $G_1 = (\{a\}, P_1, N_1, S_1)$ and $G_2 = (\{a\}, P_2, N_2, S_2)$ be any two conjunctive grammars over $\{a\}$. Assume, without loss of generality, that $N_1 \cap N_2 = \varnothing$. Construct a new conjunctive grammar $G = (\{a\}, P_1 \cup P_2, N_1 \cup N_2, S_1)$. Then $L_G(S_1) = L(G_1)$ and $L_G(S_2) = L(G_2)$, and therefore testing the equality of $L_G(S_1)$ and $L_G(S_2)$ solves the equivalence problem for G_1 and G_2.

Now this technical problem may be easily reduced to the equivalence problem for one-nonterminal conjunctive grammars over $\{a\}$. Let a grammar $G = (\{a\}, \{A_1, A_2, \ldots, A_m\}, P, A_1)$ be given, and assume without loss of generality that it is of the form required in Lemma 1; it is asked whether $L_G(A_1) = L_G(A_2)$. Construct a one-nonterminal unary conjunctive grammar G' that encodes G according to Theorem 1, with

$$L(G') = \{a^{np-d_1} \mid a^n \in L_G(A_1)\} \cup \{a^{np-d_2} \mid a^n \in L_G(A_2)\} \cup \bigcup_{i \geqslant 3} \{a^{np-d_i} \mid a^n \in L_G(A_i)\}.$$

Next, the same transformation is applied to the grammar $G = (\{a\}, \{A_2, A_1, A_3, \ldots, A_m\}, P, A_2)$, with nonterminals A_1 and A_2 exchanged. The values of $p, d_1, \ldots, d_m$ are the same, as they depend only on m, so the generated language is

$$L(G'') = \{a^{np-d_2} \mid a^n \in L_G(A_1)\} \cup \{a^{np-d_1} \mid a^n \in L_G(A_2)\} \cup \bigcup_{i \geqslant 3} \{a^{np-d_i} \mid a^n \in L_G(A_i)\}.$$

Clearly, the two languages are the same if and only if $L_G(A_1) = L_G(A_2)$. □

Theorem 7. *The co-finiteness problem for one-nonterminal unary conjunctive grammars is Σ_1-complete.*

Proof. The Σ_1-hardness of the co-finiteness problem is established by a reduction from the emptiness problem for unary conjunctive grammars with unrestricted number of nonterminals, which is Σ_1-hard by Proposition 4. As before, to shorten the notation we focus on equations over sets of numbers.

(***handle the small difficulty with ε in $L(G_0)$***)

Let G_0 be a unary conjunctive grammar with starting symbol A_1. Construct a grammar G by introducing an additional non-terminal A_2 with the same set of productions as A_1. It is easy to see that $L_G(A_1) = L_G(A_2) = L_{G_0}(A_1)$. Then turn G into equation over sets of natural numbers $X = \varphi(X)$ by Theorem 1 and let $S_i = \{np - d_i \mid n \in L_G(A_i)\}$, as promised by Theorem 1. Now turn φ into φ' by introducing another term to φ':

$$\varphi'(X) = \varphi(X) \cup \bigcup_{i=0}^{p-1} (X + d_1 + i \cap X + d_2 + i).$$

Claim 7. If $L(G_0) = \varnothing$ then the unique solution S' of φ' is the same as the unique solution S of φ and it is not co-finite.

Proof. It is claimed that in this case each term $S' + d_1 + i \cap S' + d_2 + i$ is empty.

Suppose the contrary. Consider the smallest number $n \in \bigcup_{t=0}^{p-1}(S' + d_1 + t \cap S' + d_2 + t)$. Then on all smaller numbers S and S' coincide. In particular, if $n \in S' + d_1 + t \cap S' + d_2 + t$ then also $n \in S + d_1 + t \cap S + d_2 + t$, as the numbers used to produce n on the right-hand side of the equation are smaller than n. Now, by calculations similar to those in Lemma 2, it will be proved that if $n \in S + d_1 + t \cap S + d_2 + t$, then $n \in S_1 + d_1 + t \cap S_2 + d_2 + t$.

By distributivity,

$$S+d_1 \ \cap \ S+d_2 = \bigcup_i (S_i + d_1) \cap \bigcup_j (S_j + d_2) = \bigcup_{i,j} (S_i + d_1) \cap (S_j + d_2),$$

and the value of each subexpression is

$$(S_i+d_1)\cap(S_j+d_2) = \big(\{np | n \in L_G(A_i)\}-d_i+d_1\big)\cap\big(\{np | n \in L_G(A_j)\}-d_j+d_2\big).$$

Since $d_1, d_2, d_i, d_j \in \{1, \ldots, \frac{p}{4} - 1\}$, the differences $-d_i + d_1, -d_j + d_2$ are in $\{-\frac{p}{4}+1, -\frac{p}{4}+2, \ldots, \frac{p}{4}-1\}$, and thus any number that belongs to this intersection is equal modulo p both to $d_1 - d_i$ and to $d_2 - d_j$. Accordingly,

$$4^1 - 4^i = 4^2 - 4^j,$$

which is true only for $i = 1$ and $j = 2$. Therefore, both S_1 and S_2 are nonempty, which yields a contradiction, as $S_1 = S_2 = \varnothing$ by the assumption.

The contradiction obtained proves that all the terms $S' + d_1 + t \cap S' + d_2 + t$ are empty, and thus the unique solution S' of $X = \varphi'(X)$ satisfies the equation $X = \varphi(X)$, and hence must be equal to S. For the definition of S according to Theorem 1, it is easy to see that it is never co-finite. □

Claim 8. If $a^n \in L(G_0)$ for $n \geqslant 1$, then every number greater or equal to pn is in S', and thus S' is co-finite.

Proof. By Theorem 1, $pn - d_1, pn - d_2 \in S$, and accordingly $pn - d_1, pn - d_2 \in S'$, since $S \subseteq S'$.

Let $m = pn' + i$ for some $0 \leqslant i < p$ and $n' \geqslant n$. By an induction on n' it will be proved that $m \in S'$. If $n' = n$ then, as stated above, $pn' - d_1, pn' - d_2 \in S'$, and if $n' > n$, then $pn' - d_0, pn' - d_1 \in S$ by the induction assumption. In each case m is produced by the subexpression $X+d_1+i \ \cap \ X+d_2+i$ as follows: $pn' + i = pn' - d_1 + d_1 + i \in S' + d_1 + i$, $pn' + i = pn' - d_2 + d_2 + i \in S' + d_2 + i$ and thus $m \in S' + d_1 + i \cap S' + d_2 + i \subseteq \varphi'(S') = S'$. □

It follows from Claim 7 and Claim 8 that $L(G_0)$ is non-empty if and only if the solution of $X = \varphi(X)$ is co-finite, which shows the correctness of the reduction. □

Theorem 8. *The finiteness problem for one-nonterminal unary conjunctive grammars is Σ_1-complete.*

Proof. To see that the problem is in Σ_1 consider the following nondeterministic Turing machine that tests whether a given conjunctive grammar $G = (\{a\}, \{S\}, P, S)$ generates a finite language. The machine starts with guessing a finite language $F \subset a^*$ and then uses the method of Theorem 5 to check whether $L(G) = F$.

The Σ_1-hardness is shown by reduction from the problem of whether a given unary conjunctive grammar $G = (\{a\}, \{A_1, \ldots, A_n\}, P, A_1)$ generates a language *other than* a^+. This problem is Σ_1-complete, because testing whether G generates a^+ is known to be a Π_1-complete problem [4, Thm.4].

Assume without loss of generality that G contains a nonterminal that generates an infinite language and that G is of the form given in Lemma 1. By Theorem 1, there exist numbers $1 \leqslant d_1 < \ldots < d_n < p$, and a one-nonterminal grammar $G_1 = (\{a\}, \{S\}, P_1, S)$ generating the language $\{a^{np-d_i} \mid a^n \in L_G(A_i)\}$ can be constructed. Accordingly, $\{a^{np-d_1} \mid n \geqslant 1\} \subseteq L(G_1)$ if and only if $L(G) = a^+$.

Note that, according to the theorem, for each rule

$$S \to a^{\ell_1} SS \& \ldots \& a^{\ell_k} SS \tag{1}$$

of this grammar there exists a number i with $L(a^{\ell_1} SS \& \ldots \& a^{\ell_k} SS) \subseteq \{a^{np-d_i} \mid n \geqslant 1\}$. Let such a rule be called an i-rule. Also note that $L(G_1)$ is always infinite, because of a nonterminal generating an infinite language.

Now construct a new grammar $G_2 = (\{a\}, \{S\}, P_2, S)$, where P_2 contains all "short" rules of G_1, as well as a "long" rule

$$S \to a^{\ell_1} SS \& \ldots \& a^{\ell_k} SS \& a^{p+d_1-d_i} S \tag{2}$$

for each i-rule (1).

Clearly, $L(G_2) \subseteq L(G_1)$, as every rule in P_2 is a more restrictive version of some rule from P_1 containing an extra conjunct, and thus every derivation in G_2 can be simplified down to a derivation of the same string in G_1. The goal of the additional conjunct is to make the membership of a^{np-d_1} in $L(G_2)$ a necessary condition for generating the number $a^{(n+1)p-d_i}$. In this way, if any number in track d_1 is missing, then no larger numbers will be generated, and the language will be finite.

Note that the grammar G_2 inherits the property of G_1 that a rule (2) generates a subset of $\{a^{np-d_i} \mid n \geqslant 1\}$, for d_i given in the last conjunct. ***check*** ***put this to a better place***

Formally, it is claimed that G_2 generates an infinite language if and only if $\{a^{np-d_1} \mid n \geqslant 1\} \subseteq L(G_1)$.

$\Leftarrow$ Assume that $\{a^{np-d_1} \mid n \geqslant 1\} \subseteq L(G_1)$, that is, every string a^{np-d_1} with $n \geqslant 1$ is in $L(G_1)$. It is claimed that $L(G_1) \subseteq L(G_2)$ (as the converse inclusion is known, this would show the equality of these languages).

Suppose the contrary, that $L(G_2) \setminus L(G_1) \neq \varnothing$, and let a^{np-d_i} with $n \geqslant 1$ and $1 \leqslant i \leqslant m$ be the shortest string in $L(G_1)$ that is not in $L(G_2)$. This

string must be produced by a long rule of G_1. because all the short rules of G_1 are in G_2 as well, and therefore $n \geqslant 2$. Consider the i-rule (1) by which a^{np-d_i} is generated. Then $a^{np-d_i} \in a^{\ell_j} L(G_1)^2$ and hence $a^{np-d_i} \in a^{\ell_j} L(G_2)^2$, as $\ell_j \geqslant 1$ and $L(G_1)$ and $L(G_2)$ do not differ on strings shorter than a^{np-d_i}. The number $(n-1)p-d_1$ is positive, as $n \geqslant 2$, and thus the string $a^{(n-1)p-d_1}$ is well-defined and for the same reason, $a^{(n-1)p-d_1} \in L(G_1), L(G_2)$, and, accordingly, $a^{np-d_i} \in a^{p+d_1-d_i} L(G_1)$. Therefore, a^{np-d_i} is generated in G_2 by the rule (2) corresponding to (1), which contradicts the assumption.

$\Leftarrow$ Conversely, if $\{a^{np-d_1} \mid n \geqslant 1\} \not\subseteq L(G_1)$, then there is a number $n \geqslant 1$ with $a^{np-d_1} \notin L(G_1)$, and hence with $a^{np-d_1} \notin L(G_2)$ (as $L(G_2) \subseteq L(G_1)$). Now the claim is that no string longer than a^{np-d_1} is in $L(G_2)$.

Let $a^{n'p-d_i} \in L(G_2)$ for some $n' > n$ and $1 \leqslant i \leqslant m$ be the shortest string of length greater than $np-d_1$ generated by G_2, and let $a^{n'p-d_i}$ be generated by an i-rule (2). According to the last conjunct of this rule, $a^{n'p-d_i} \in a^{p+d_1-d_i} L(G_2)$ and hence $a^{(n'-1)p-d_1} \in L(G_2)$. Now if $n'-1 = n$, then this does not hold by assumption, and if $n'-1 > n$, then $a^{(n'-1)p-d_1}$ is a string shorter than $a^{n'p-d_i}$ satisfying the assumptions, and in both cases a contradiction is obtained.

The above claims imply that $L(G_2)$ is finite if and only if $L(G) \neq a^+$, which completes the reduction.

reference to general CF: perhaps Cudia.

	equiv. to reg. L_0	equivalence	finiteness	co-finiteness
Context-free				
general case	undecidable	undecidable	decidable	undecidable(?)
$\Sigma = \{a\}$, any N	decidable	decidable	decidable	decidable
$\Sigma = \{a\}$, $N = \{S\}$	decidable	decidable	decidable	decidable
Conjunctive				
general case	undecidable	undecidable	undecidable	undecidable
$\Sigma = \{a\}$, any N	Π_1-complete [4]	Π_1-complete [4]	undecidable [4]	**Σ_1-complete**
$\Sigma = \{a\}$, $N = \{S\}$	**decidable**	**Π_1-complete**	**Σ_1-complete**	**Σ_1-complete**

Table 2. Decision problems for grammars over $\{a\}$.

References

1. S. Ginsburg, H. G. Rice, "Two families of languages related to ALGOL", *Journal of the ACM*, 9 (1962), 350–371.
2. D. T. Huynh, "Commutative grammars: the complexity of uniform word problems", *Information and Control*, 57:1 (1983), 21–39.
3. A. Jeż, "Conjunctive grammars can generate non-regular unary languages", *International Journal of Foundations of Computer Science*, 19:3 (2008), 597–615.
4. A. Jeż, A. Okhotin, "Conjunctive grammars over a unary alphabet: undecidability and unbounded growth", *Theory of Computing Systems*, to appear.

5. A. Jeż, A. Okhotin, "Complexity of equations over sets of natural numbers", *25th Annual Symposium on Theoretical Aspects of Computer Science* (STACS 2008, Bordeaux, France, 21–23 February, 2008), 373–383.
6. 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.
7. 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.
8. P. McKenzie, K. W. Wagner, "The complexity of membership problems for circuits over sets of natural numbers", *Computational Complexity*, 16 (2007), 211–244.
9. A. Okhotin, "Conjunctive grammars", *Journal of Automata, Languages and Combinatorics*, 6:4 (2001), 519–535.
10. A. Okhotin, "Nine open problems for conjunctive and Boolean grammars", *Bulletin of the EATCS*, 91 (2007), 96–119.
11. A. Okhotin, P. Rondogiannis, "On the expressive power of univariate equations over sets of natural numbers", *IFIP Intl. Conf. on Theoretical Computer Science* (TCS 2008, Milan, Italy, 8–10 September, 2008), IFIP vol. 273, 215–227.
12. W. Plandowski, W. Rytter, "Complexity of language recognition problems for compressed words", in: J. Karhumäki, H. A. Maurer, G. Păun, G. Rozenberg (Eds.), *Jewels are Forever*, Springer, 1999, 262–272.
13. L. J. Stockmeyer, A. R. Meyer, "Word problems requiring exponential time", *STOC 1973*, 1–9.

A State complexity of symbolic addition

Lemma 4. Let A and B be NFAs over an alphabet $\Sigma_k = \{0, 1, \ldots, k-1\}$, with $L(M_1) \cap 0\Sigma_k^* = L(M_2) \cap 0\Sigma_k^* = \varnothing$, let A and B have m and n states, respectively. Then there exists and can be effectively constructed a $(2mn + 2m + 2n + 1)$-state NFA over Σ_k, which recognizes the language $\{$the k-ary notation of $n_1 + n_2 \mid$ the k-ary notation of n_i is in $L(M_i)\}$.

Proof (sketch). The new NFA has four types of states defined as follows:

(I) Each state q_{ijc}^{AB} corresponds to A in state i, B in state j and carry c (where c is 0 or 1). The initial state is q_{000}^{AB}. The following diagram illustrates this case:

```
              c
        <-A(i)-- x x x x x
+
        <-B(j)-- y y y y y
--------------------------
                 z z z z z
```

The string $zzzzz$ has been read, and the NFA has guessed its representation as $xxxxx \boxplus yyyyy$, where A goes to i by $xxxxx$ and B goes to j by $yyyyy$. If $c = 1$, then $xxxxx \boxplus yyyyy = 1zzzzz$.

For all digits $x, y \in \{0, ..., k-1\}$, such that A may go from i to i' by x and B may go from j to j' by y, the new automaton has a transition from q_{ijc}^{AB} to $q_{i'j'c'}^{AB}$ by $x + y + c \mod k$, where $c' = (x + y + c)/k$ rounded down.

(II) If the automaton B is no longer running (that is, the second number is over), while A still produces some digits, this case is implemented in states q_{ic}^{A}, where i is a state of A and c is a carry:

```
              c
        <-A(i)-- x x x x x
+
                    y y y
--------------------------
                 z z z z z
```

For every state q_{ijc}^{AB}, such that j is an accepting state of B, and for every digit $x \in \{0, ..., k-1\}$ there is a transition from q_{ijc}^{AB} to $q_{i'c'}^{A}$ by $x + c \mod k$, where $c' = (x + c)/k$ rounded down (this is the case when the second number has just finished).

For every q_{ic}^{A} and for every x, there is a transition from $q_{i'c'}^{A}$ by $x + c \mod k$, where $c' = (x + c)/k$ rounded down.

(III) Symmetrically, there is a group of states q_{jc}^{B}, which correspondings to the case when the first number has ended.

(IV) q_{Acc} is a dedicated accepting state with no outgoing transitions.

Other accepting states are the following: for every i accepting in A and j accepting in B, q_{ij0}^{AB}, q_{i0}^{A} and q_{j0}^{B} are accepting in the new automaton, while q_{ij1}^{AB}, q_{i1}^{A} and q_{j1}^{B} have transitions by 1 to q_{Acc}.

This completes the construction. □