Programming and Computer Software, Vol. 28, No. 5, 2002, pp. 243–249. Translated from Programmirovanie, Vol. 28, No. 5, 2002.

Conjunctive Grammars and Systems of Language Equations

A. S. Okhotin

Department of Computing and Information Science, Queen's University, Kingston, Ontario, Canada K7L3N6
e-mail: okhotin@cs.queensu.ca

Received January 10, 2002

Abstract—This paper studies systems of language equations that are resolved with respect to variables and contain the operations of concatenation, union and intersection. Every system of this kind is proved to have a least fixed point, and the equivalence of these systems to conjunctive grammars is established. This allows us to obtain an algebraic characterization of the language family generated by conjunctive grammars.

1. INTRODUCTION

Conjunctive grammars, introduced in [1–3], are an extension of context-free grammars that permits the use of the explicit set-theoretic intersection operation in the body of any rule.

The family of languages generated by conjunctive grammars is known to strictly include the closure of the class of context-free languages under intersection [3]; on the other hand, it is contained in the class of context-sensitive languages [2, 3], and the latter inclusion is strict if **P** ≠ **PSPACE** [3].

Practical significance of conjunctive grammars follows from the facts that, on the one hand, their expressiveness is greater than that of context-free grammars and, on the other hand, there exist effective recognition and parsing algorithms for them, such as the algorithm for grammars in binary normal form [2, 3] of complexity $O(n^3)$; the top-down parsing algorithm and the recursive descent method [4], which require linear time for parsing certain languages; the tabular algorithm for arbitrary grammars [5] with cubic dependence on time; and the bottom-up parsing algorithm, which recognizes whether a string belongs to the language in time varying from $O(n)$ to $O(n^3)$ [6]. Note that, in the case of the latter algorithm, we have linear time for any language belonging to the closure of the class of deterministic context-free languages under all set-theoretic operations. Moreover, there exists a class of linear conjunctive grammars for which the upper bound for the complexity of recognition is quadratic in time and linear in memory [3, 5].

Certain problems for conjunctive grammars are known to be unsolvable. These are problems of emptiness, universality, finiteness, context-freeness, regularity, equivalence, and inclusion [1–3]. All these problems are unsolvable for linear conjunctive grammars as well.

Similar to the case of context-free grammars, the membership problem for conjunctive grammars is **P**-complete [5, 7]. The membership problem for linear conjunctive grammars is also **P**-complete [5], unlike that for linear context-free grammars, which is **NLOGSPACE**-complete [8].

In this paper, we consider characterization of conjunctive grammars by systems of language equations, which are resolved with respect to language variables and contain the union, intersection, and concatenation operations.

Similar characterization of context-free languages is well known to be as follows [9, 10]. Context-free grammars are put in one-to-one correspondence with systems of language equations in n unknown languages of the form $X_i = s_{11} \cdot \ldots \cdot s_{1l_1} + \ldots + s_{m1} \cdot \ldots \cdot s_{ml_m}$, where "+" and "·" denote the union and concatenation of languages, respectively. The least solution of systems of this kind is a vector of languages generated by the nonterminals of the original grammar. For this reason, in early works on the theory of formal languages, the context-free languages were referred to as *algebraic* languages, to emphasize their mathematical meaning. Note that, in the French literature, this term, *langages algébriques*, is used as the main name of this family.

In the context-free case, the motivation for the consideration of language equations was that the set of formal languages over a given finite nonempty alphabet Σ forms a semiring with respect to the union and concatenation operations (which play roles of the addition and multiplication), zero $\emptyset$, and unity $\{\epsilon\}$. Therefore, a powerful algebraic apparatus turned out applicable to formal languages making it possible to apply classical methods related to power series and systems of equations to certain problems of formal language theory. This, in turn, made it possible to build the theory of context-free languages with ultimate mathematical rigorousness [10].

The set of all languages with given independent concatenation, union, and intersection operations cannot, of course, be described by such a simple and attractive model as semiring. It turns out, however, that, although in this case there is no clear algebraic interpre-

0361-7688/02/2805-0243$27.00

tation as in the context-free case, the consideration of language equations with an additional operation of intersection is still possible. There exists a relationship between solutions of such systems and languages generated by conjunctive grammars, which, on the one hand, allows us to understand semantics of conjunctive grammars better and, on the other hand, to develop a new method for strictly proving statements on properties of conjunctive grammars and to solve problems of their analysis.

2. CONJUNCTIVE GRAMMARS

First, we give a definition of a conjunctive grammar and the language it generates and formulate some statements we will need in what follows. The discussion follows basically the work [3].

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

$$A \longrightarrow \alpha_1 \& \ldots \& \alpha_n \quad (A \in N, n \geq 1, \quad \alpha_i \in (\Sigma \cup N)^* \text{ for any } i) \tag{1}$$

where α_i are distinct strings, the order of which is not important; and $S \in N$ is a nonterminal designated as the *start symbol* of the grammar.

For any rule of form (1) and any number i ($1 \leq i \leq n$), an object $A \longrightarrow \alpha_i$ is referred to as a *conjunct*.

In what follows, we will use three following special symbols: '(', '&', and ')' assuming that none of them is contained in $\Sigma \cup N$.

A conjunctive grammar generates strings by deriving them from the start symbol. Intermediate objects being transformed in the course of the derivation are formulas over the basis consisting of the concatenation and conjunction.

Definition 2. Let $G = (\Sigma, N, P, S)$ be a conjunctive grammar. The set of formulas $\mathcal{F} = \mathcal{F}(S \cup N) \subseteq (\Sigma \cup N \cup \{'(', '\&', ')'\})^*$ is specified by the following inductive definition:

(i) any terminal or nonterminal symbol is a formula;

(ii) the empty string ϵ is a formula;

(iii) if $\mathcal{A} \neq \epsilon$ and $\mathcal{B} \neq \epsilon$ are formulas, then $\mathcal{A}\mathcal{B}$ is a formula;

(iv) if $\mathcal{A}_1, \ldots, \mathcal{A}_n$ ($n \geq 1$) are formulas, then $(\mathcal{A}_1 \& \ldots \& \mathcal{A}_n)$ is a formula.

Definition 3. Let $G = (\Sigma, N, P, S)$ be a conjunctive grammar. The relation $\overset{G}{\Rightarrow}$ of derivability in one step on the set of formulas is defined as follows:

(i) for any strings $s', s'' \in (\Sigma \cup N \cup \{'(', '\&', ')'\})^*$ and a nonterminal $A \in N$, such that $s'As'' \in \mathcal{F}$, and for any rule $A \longrightarrow \alpha_1 \& \ldots \& \alpha_n \in P$,

$$s'As'' \Rightarrow s'(\alpha_1 \& \ldots \& \alpha_n)s''; \tag{2}$$

(ii) for any strings $s', s'' \in (\Sigma \cup N \cup \{'(', '\&', ')'\})^*$, a number $n \geq 1$, and a terminal string $w \in \Sigma^*$ such that $s'(\underbrace{w\& \ldots \&w}_{n})s'' \in \mathcal{F}$,

$$\underbrace{s'(w \& \ldots \& w)}_{n}s'' \overset{G}{\Rightarrow} s'ws''. \tag{3}$$

The reflexive transitive closure of $\overset{G}{\Rightarrow}$ is denoted as $\overset{G}{\Rightarrow}*$.

Definition 4. Let $G = (\Sigma, N, P, S)$ be a conjunctive grammar. The language generated by a certain formula $\mathcal{A}$ is the set of all strings over Σ that are derivable from this formula in a finite (greater than or equal to zero) number of steps: $L_G(\mathcal{A}) = \{w \in \Sigma^* | \mathcal{A} \overset{G}{\Rightarrow} w\}$. The language generated by a grammar is a language generated by its start symbol: $L(G) = L_G(S)$.

Definition 5. A conjunctive grammar $G = (\Sigma, N, P, S)$ is said to be linear if each rule from P is of the form

$$A \longrightarrow u_1 B_1 v_1 \& \ldots \& u_n B_n v_n \quad (u_i, v_i \in \Sigma^*; B_i \in N), \tag{4a}$$

$$A \longrightarrow w \quad (w \in \Sigma^*). \tag{4b}$$

A linear conjunctive grammar G is said to be in the linear normal form if each rule from P is of the form

$$A \longrightarrow bB_1 \& \ldots \& bB_m \& C_1 c \& \ldots \& C_n c \quad (b, c \in \Sigma; m, n \geq 0; m + n \geq 1; B_i, C_i \in N), \tag{5a}$$

$$A \longrightarrow a \quad (a \in \Sigma). \tag{5b}$$

$$S \longrightarrow \epsilon, \tag{5c}$$

and rule (5c) is admissible only if S does not occur on the right-hand sides of the rules.

For any linear conjunctive grammar, there exists a grammar in the linear normal form that generates the same language, and this transformation can be implemented algorithmically [3].

The following theorem [1–3] shows that the language generated by a formula inductively depends on its structure and that the operations on formulas correspond to operations on languages.

Theorem 1. Let $G = (\Sigma, N, P, S)$ be a conjunctive grammar, $\mathcal{A}_1, \ldots, \mathcal{A}_n, \mathcal{B}$ be formulas, $\mathcal{A} \in N$, and $a \in \Sigma$. Then,

$$L_G(\epsilon) = \{\epsilon\}, \tag{6a}$$

$$L_G(a) = \{a\}, \tag{6b}$$

$$L_G(A) = \bigcup_{A \to \alpha_1 \& \ldots \& \alpha_m \in P} L_G((\alpha_1 \& \ldots \& \alpha_m)), \tag{6c}$$

$$L_G(\mathcal{A}\mathcal{B}) = L_G(\mathcal{A})L_G(\mathcal{B}), \tag{6d}$$

$$L_G((\mathcal{A}_1 \& \ldots \& \mathcal{A}_n)) = \bigcap_{i=1}^{n} L_G(\mathcal{A}_i). \tag{6e}$$

Assertions (6a) and (6b) are trivial. To prove (6c), it is sufficient to note that any derivation from the formula A begins with the application of one of the rules for the nonterminal A. Assertions (6d) and (6e) are proved by induction on the derivation length.

Theorem 1 already suggests a certain interpretation of conjunctive grammars by systems of language equations, since (6a)–(6e) are, in fact, equations, such that the substitution of (6a), (6b), (6d), and (6e) into (6c) for all $A \in N$ results in a system of $|N|$ equations resolved with respect to nonterminal variables.

We will consider a mathematical approach to the treatment of the language equations of this kind, which will allow us to get a better characterization of conjunctive languages by systems of equations than that given in Theorem 1.

3. LANGUAGE EQUATIONS WITH THE INTERSECTION OPERATION

Definition 6 (Expression). Let Σ be a finite nonempty alphabet. Let $X = \{X_1, \ldots, X_n\}$ be a vector of variables. An expression over the alphabet Σ depending on the variables X is defined inductively as follows:

- ϵ is an expression;
- any symbol $a \in \Sigma$ is an expression;
- any variable $X_i \in X$ is an expression;
- if φ_1 and φ_2 are expressions, then $\varphi_1\varphi_2$, $(\varphi_1 \mid \varphi_2)$, and $(\varphi_1 \,\&\, \varphi_2)$ are also expressions.

We will use the notation $\varphi(X)$ for an expression.

To simplify the expression notation, we will omit parentheses when it does not result in any confusion. In this case, the greatest precedence is assigned to the concatenation operation, the next precedence is given to the conjunction "&", and the disjunction "|" has the least precedence.

Note that any formula in the sense of Definition 2 is an expression in the sense of Definition 6, with the nonterminal symbols of the formula playing role of variables of the expression. On the other hand, any expression without the disjunction operation is a formula.

Assuming that variables X_i take values of languages over Σ, we define a value of an expression for given values of variables.

Definition 7 (Value of expression). Let $L = (L_1, \ldots, L_n)$ $(L_i \subseteq \Sigma^*)$ be a vector of n languages over Σ, where $n \geq 1$. Let φ be an expression over Σ depending on variables $X_1, \ldots, X_n$. A value of the expression φ on the vector L is a language over the same alphabet Σ. It is denoted as $\varphi(L)$ and defined by induction on the expression structure:

- $\epsilon(L) = \{\epsilon\}$,
- $a(L) = \{a\}$ for all $a \in \Sigma$,
- $X_i(L) = L_i$ for all $X_i \in X$,
- $\varphi_1\varphi_2(L) = \varphi_1(L) \cdot \varphi_2(L)$, $(\varphi_1 \mid \varphi_2)(L) = \varphi_1(L) \cup \varphi_2(L)$ and $(\varphi_1 \,\&\, \varphi_2)(L) = \varphi_1(L) \cap \varphi_2(L)$ for any expressions φ_1 and φ_2.

Let us prove an important property of such expressions, which says that the membership of a string of length l in a language $\varphi(L_1, \ldots, L_n)$ is determined by whether strings of length l or less belong to the languages $L_1, \ldots, L_n$.

First, we introduce additional terminology.

Definition 8. Languages L' and L'' are said to be equal modulo L, which is written as $L' = L'' \pmod{L}$, if $L' \cap L = L'' \cap L$.

Vectors of languages $L' = (L'_1, \ldots, L'_n)$ and $L'' = (L''_1, \ldots, L''_n)$ are said to be equal modulo L if $L'_i = L''_i \pmod{L}$ for all i.

The set of all strings over the alphabet Σ of length l or less is denoted as $\Sigma^{\leq l}$.

Lemma 1. Let $n > 0$. Let $L' = (L'_1, \ldots, L'_n)$ and $L'' = (L''_1, \ldots, L''_n)$ be vectors of languages over Σ. Let φ be an expression depending on variables $X_1, \ldots, X_n$ over Σ. If $L' = L'' \pmod{\Sigma^{\leq l}}$, then $\varphi(L') = \varphi(L'') \pmod{\Sigma^{\leq l}}$.

Proof. The lemma is proved by induction on the structure of φ.

- If $\varphi = \epsilon$, then $\varphi(L') = \varphi(L'') = \{\epsilon\}$.
- If $\varphi = a$ $(a \in \Sigma)$, then $\varphi(L') = \varphi(L'') = \{a\}$.
- Let $\varphi = X_i$, and let $u \in \Sigma^{\leq l}$ be a string. Then, $u \in \varphi(L')$ is equivalent to $u \in L'_i$, $u \in L''_i$, and $u \in \varphi(L'')$.
- Let $\varphi = \varphi_1\varphi_2$ and $w \in \Sigma^{\leq l}$. In this case, $w \in \varphi(L')$ if and only if there exists a factorization $w = uv$ $(u, v \in \Sigma^{\leq l})$ such that $u \in \varphi_1(L')$ and $v \in \varphi_2(L')$. By the induction hypothesis, this is equivalent to $u \in \varphi_1(L'')$ and $v \in \varphi_2(L'')$, which, in turn, holds only if $uv \in \varphi(L'')$.
- Let $\varphi = \varphi_1 \mid \varphi_2$. Consider an arbitrary string $u \in \Sigma^{\leq l}$. If $u \in \varphi(L')$, then $u \in \varphi_1(L')$ or $u \in \varphi_2(L')$, which, by the induction hypothesis implies that $u \in \varphi_1(L'')$ or $u \in \varphi_2(L'')$, i.e., $u \in \varphi(L'')$.

The case of $\varphi = \varphi_1 \& \varphi_2$ is considered similar to the previous one.

Now, we generalize Definition 7 to the case of vectors of expressions.

Definition 9 (Value of a vector of expressions). Let $L = (L_1, \ldots, L_n)$ $(L_i \subseteq \Sigma^*)$ be a vector of languages over Σ. Let $\varphi_1, \ldots, \varphi_m$ be expressions over the alphabet Σ depending on variables $X_1, \ldots, X_n$. A value of the vector of expressions $P = (\varphi_1, \ldots, \varphi_m)$ on the vector of languages L is the vector of m languages

$$P(L) = (\varphi_1(L), \ldots, \varphi_m(L)). \tag{7}$$

Let us define a partial order with respect to the inclusion "$\leqslant$" on the set of languages and extend it to the set of vectors of languages of length n as follows:

$(L_1', \ldots, L_n') \preccurlyeq (L_1'', \ldots, L_n'')$ if and only if $L_i' \subseteq L_i''$ for all i $(1 \le i \le n)$.

Definition 10. A system of equations over an alphabet Σ in unknowns $X = \{X_1, \ldots, X_n\}$ is

$$X = P(X), \tag{8}$$

where $P = (\varphi_1, \ldots, \varphi_n)$ is a vector of expressions over the alphabet Σ depending on X.

A vector of languages $L = (L_1, \ldots, L_n)$ is a solution of the system of equations (8) if

$$L = P(L). \tag{9}$$

A vector of languages L is said to be the least solution of system (8) if **(i)** it is a solution of the system and **(ii)** $L \preccurlyeq L'$ for any other solution L' of (8).

Note that P is an operator on the set $2^\Sigma \times \ldots \times 2^\Sigma$, a solution of system (8) is a fixed point of P, and the least solution is the least fixed point of the operator.

Let us establish some properties of the operator P, which follow from Definition 10.

Lemma 2. For any system (8) in unknowns $X_1, \ldots, X_n$, the operator $P = (\varphi_1, \ldots, \varphi_n)$ is monotonous with respect to "$\preccurlyeq$"; i.e., for any vectors of languages $L' = (L_1', \ldots, L_n')$ and $L'' = (L_1'', \ldots, L_n'')$, such that $L' \preccurlyeq L''$, we have $P(L') \preccurlyeq P(L'')$.

Proof. Let $L' \preccurlyeq L''$. Consider an arbitrary expression ψ and an arbitrary string $w \in \Sigma^*$ such that $w \in \psi(L')$. By applying induction on the structure of ψ, it is not difficult to show that $w \in \psi(L'')$, since, in view of Definition 7, all operations occurring in the expressions are interpreted by monotonous functions.

Applying this auxiliary statement to the components $\varphi_1, \ldots, \varphi_n$ of the operator P, we get the assertion of the lemma.

Lemma 3. For any system (8) in unknowns $X_1, \ldots, X_n$, the operator $P = (\varphi_1, \ldots, \varphi_n)$ is $\cup$-continuous with respect to "$\preccurlyeq$"; i.e., for any monotonically increasing sequence of vectors of languages $\{L^{(i)} = (L_1^{(i)}, \ldots, L_n^{(i)})\}_{i=1}^{\infty}$ converging to a limit $L = (L_1, \ldots, L_n)$, the sequence $\{P(L^{(i)})\}_{i=1}^{\infty}$ converges to the limit $P(L)$.

Proof. In view of monotonous convergence of $\{L^{(i)}\}$, for any length $l \ge 0$ of strings, there exists a number $n_0 > 0$ such that

$$\begin{aligned} L &= L^{(n_0)} = L^{(n_0+1)} = \ldots \\ &= L^{(n_0+k)} = \ldots \pmod{\Sigma^{\le l}}. \end{aligned} \tag{10}$$

By virtue of Lemma 1, it follows from (10) that

$$\begin{aligned} P(L) &= P(L^{(n_0)}) = P(L^{(n_0+1)}) = \ldots \\ &= P(L^{(n_0+k)}) = \ldots \pmod{\Sigma^{\le l}}, \end{aligned} \tag{11}$$

which, in view of arbitrariness of l, implies that the sequence $\{P(L^{(i)})\}$ converges to the limit $P(L)$.

Now, the existence of a least solution of the system easily follows from the classical theory of fixed point.

Theorem 2. For any system (8) in unknowns $X_1, \ldots, X_n$, the operator $P = (\varphi_1, \ldots, \varphi_n)$ has a least fixed point given by

$$L = (L_1, \ldots, L_n) = \lim_{i \to \infty} P^i(\underbrace{\emptyset, \ldots, \emptyset}_{n}). \tag{12}$$

Note that the property of $\cap$-continuity of the operator P can be established in a similar way. The latter property implies that there exists a greatest solution of the system of equations $X = P(X)$.

Consider now the problem of solution uniqueness. It is easy to see that some systems have only one solution. For example, the only solution of the system consisting of one equation $X_1 = aX_1b \mid \epsilon$ is the language $\{a^n b^n \mid n \ge 0\}$. On the other hand, some systems may have several solutions. For example, a solution of the system $(X_1 = X_1 \mid X_2,\ X_2 = X_2)$ is any pair (L_1, L_2) such that $L_2 \subseteq L_1$.

We give a simple sufficient condition of solution uniqueness by defining a sufficiently wide class of systems that have exactly one solution.

Definition 11. Let $P = (\varphi_1, \ldots, \varphi_n)$ be a vector of expressions depending on a vector of variables $X = (X_1, \ldots, X_n)$ over an alphabet Σ. A system $X = P(X)$ is said to be *strict* if each expression φ_i from the vector P can be constructed as follows:

- the expression ϵ is admissible in a strict system;
- if ψ_1 and ψ_2 are arbitrary expressions and $a \in \Sigma$, then $\psi_1 a \psi_2$ is an expression admissible in a strict system;
- if ψ_1 and ψ_2 are admissible expressions in a strict system, then expressions $\psi_1\psi_2$, $(\psi_1 \& \psi_2)$, and $(\psi_1 \mid \psi_2)$ are also admissible in a strict system.

Let us show first that, for expressions that are admissible in a strict system, Lemma 1 can be strengthened.

Lemma 4. For any expression φ that is admissible in a strict system, for any vectors of languages L' and L'', and for any number $l \ge 0$, if $L' = L''$ (mod $\Sigma^{\le l}$), then $\varphi(L') = \varphi(L'')$ (mod $\Sigma^{\le l+1}$).

Proof. We consider an arbitrary number $l \ge 0$ and an arbitrary string $w \in \Sigma^{l+1}$. Applying induction on the structure of φ, we will show that $w \in \varphi(L')$ if and only if $w \in \varphi(L'')$.

- $\varphi = \epsilon$. Then, $w \notin \varphi(L')$ and $w \notin \varphi(L'')$.
- $\varphi = \psi_1 a \psi_2$. If $w \in \varphi(L')$, then there exists a factorization $w = uav$ $(u, v \in \Sigma^{\le l}, a \in \Sigma)$, where $u \in \psi_1(L')$ and $v \in \psi_2(L')$. Since $L' = L''$ (mod $\Sigma^{\le l}$), it follows that $u \in \psi_1(L'')$ and $v \in \psi_2(L'')$ and, hence, $uav \in \varphi(L'')$.
- $\varphi = \psi_1\psi_2$, with both expressions ψ_i being admissible in a strict system. Then, if $w \in \varphi(L')$, there exists a factorization $w = u_1u_2$, where $u_i \in \psi_i(L')$. By the induction hypothesis, $u_i \in \psi_i(L'')$ and, hence, $w \in \varphi(L'')$.

• The conjunction and disjunction cases are proved similar to the case of the concatenation. □

Theorem 3. Any strict system has only one solution.

Proof. Let $X = P(X)$ be a strict system, where $X = (X_1, \ldots, X_n)$ and $P = (\varphi_1, \ldots, \varphi_n)$. Let vectors L' and L'' be arbitrary solutions of the system. Applying induction on l, we will show that $L' = L''$ (mod $\Sigma^{\le l}$) for any number $l \ge 0$.

Basis $l = 0$. Let us show that, for any expression φ that is admissible in a strict system, $\varphi(L) \equiv const$ (mod $\{\epsilon\}$). Applying induction on the structure of φ, we have the following.

• If $\varphi = \epsilon$, then $\varphi(L) \equiv \{\epsilon\}$ for any vector of languages L.

• $\varphi = \psi_1 a \psi_2$. Then, $\epsilon \notin \varphi(L)$ for all L and, therefore, $\varphi(L) \equiv \emptyset$ (mod $\{\epsilon\}$).

• $\varphi = \psi_1\psi_2$, with both expressions ψ_1 and ψ_2 being admissible in a strict system. Then, by the induction hypothesis, $\psi_i \equiv C_i$ (mod $\{\epsilon\}$) and, hence, $\varphi \equiv C_1 \cdot C_2 = const$ (mod $\{\epsilon\}$).

• The conjunction and disjunction cases are proved similar to the case of the concatenation.

Thus, we proved that, for all j ($1 \le j \le n$), the equality $\varphi_j(L) \equiv const$ (mod $\{\epsilon\}$) holds. Hence, taking into account that L' and L'' are solutions, we have $L' = P(L') = P(L'') = L''$(mod$\{\epsilon\}$).

Induction step $l \longrightarrow l + 1$. Let $L' = L''$ (mod $\Sigma^{\le l}$). Then, by Lemma 4, $P(L') = P(L'')$ (mod $\Sigma^{\le l+1}$). Since L' and L'' are solutions, i.e., $L' = P(L')$ and $L'' = P(L'')$, it follows that $L' = L''$(mod$\Sigma^{\le l+1}$).

It follows from the above proof that the languages L' and L'' are equal, which implies that there exists only one solution of the system $X = P(X)$.

In light of Theorem 3, it is natural to set the question of existence of a *necessary* and *sufficient* condition of solution uniqueness. In Section 4, we will prove that the problem of whether a solution of a given system of language equations is unique is algorithmically undecidable.

4. CHARACTERIZATION OF CONJUNCTIVE GRAMMARS

Definition 12. Let $G = (\Sigma, N, P, S)$ be a conjunctive grammar, where $N = \{X_1, \ldots, X_n\}$ and $S = X_1$. For every nonterminal X_i, we consider an expression φ_i either of the form

$$\varphi_i = \alpha_{11}\&\ldots\&\alpha_{1m_1} \mid \ldots \mid \alpha_{l1}\&\ldots\&\alpha_{lm_l}, \qquad (13a)$$

if the grammar has rules $X_i \longrightarrow \alpha_{l1}\&\ldots\&\alpha_{1m_1}$, …, $X_i \longrightarrow \alpha_{l1}\&\ldots\&\alpha_{lm_l}$, for the nonterminal X_i or of the form

$$\varphi_i = X_i \qquad (13b)$$

if the grammar has no rules for X_i.

In view of the evident correspondence between the set of rules P and the vector of expressions $(\varphi_1, \ldots, \varphi_n)$, we will not distinguish between them, denoting $P(L) = (\varphi_1, \ldots, \varphi_n)(L)$ for any vector of languages L.

A system of equations corresponding to the grammar G is the system

$$X = P(X) \qquad (14)$$

over the alphabet Σ in unknowns $X_1, \ldots, X_n$.

Let us prove the following auxiliary statement, which establishes relationship between semantics of conjunctive grammars (derivability in a finite number of steps) and semantics of solutions of the system of equations.

Lemma 5. Let G be a grammar, $X = P(X)$ be the corresponding system of equations, and $L = (L_1, \ldots, L_n)$ be a solution of this system. Let $\mathcal{A}$ and $\mathcal{B}$ be formulas such that $\mathcal{A} \Rightarrow^* \mathcal{B}$. Then, $\mathcal{B}(L) \subseteq \mathcal{A}(L)$.

Proof. It is sufficient to prove the assertion of the lemma for a one-step derivation from $\mathcal{A} \Rightarrow \mathcal{B}$. Let $\mathcal{A} \Rightarrow \mathcal{B}$, and let $w \in \mathcal{B}(L)$, where w is some string. By applying induction on the structure of $\mathcal{A}$ and $\mathcal{B}$ (which differ in that one rule was applied to $\mathcal{B}$ compared to $\mathcal{A}$), we will show that $w \in \mathcal{A}(L)$.

• $\mathcal{A} = \mathcal{C}\mathcal{D}$ and $\mathcal{B} = \mathcal{C}'\mathcal{D}'$. Since $\mathcal{A}$ and $\mathcal{B}$ differ in only one component, either $\mathcal{C} = \mathcal{C}'$ and $\mathcal{D} \Rightarrow \mathcal{D}'$ or $\mathcal{C} \Rightarrow \mathcal{C}'$ and $\mathcal{D} = \mathcal{D}'$. Without loss of generality, we assume the latter.

Since $w \in \mathcal{B}(L) = \mathcal{C}'\mathcal{D}'(L) = \mathcal{C}'(L) \cdot \mathcal{D}'(L)$, there exists a factorization $w = uv$ such that $u \in \mathcal{C}'(L)$ and $v \in \mathcal{D}'(L)$. By the induction hypothesis, $u \in \mathcal{C}(L)$. Since $\mathcal{D} = \mathcal{D}'$, $v \in \mathcal{D}(L)$. Hence, $w \in \mathcal{C}(L)\mathcal{D}(L) = \mathcal{A}(L)$.

• $\mathcal{A} = (\mathcal{C}_1\&\ldots\&\mathcal{C}_m)$ and $\mathcal{B} = (\mathcal{C}'_1\&\ldots\&\mathcal{C}'_m)$. This case is proved similar to the previous one.

• $\mathcal{A} = X_i$, $\mathcal{B} = (\alpha_1\&\ldots\&\alpha_m)$, and $X_i \longrightarrow \alpha_1\&\ldots\&\alpha_m \in P$. Then, $w \in \mathcal{B}(L) = \bigcap_{j=1}^{m} \alpha_j(L) \subseteq \varphi_i(L)$ by the definition of the system corresponding to the grammar, and $\varphi_i(L) = L_i$ since L is a solution. On the other hand, $\mathcal{A}(L) = L_i$ and, hence, $w \in \mathcal{A}(L)$.

• $\mathcal{A} = (u\&\ldots\&u)$ and $\mathcal{B} = u$. In this case, $\mathcal{B}(L) = \{u\}$, $u = w$, and, hence, $\mathcal{A}(L) = \{w\} \ni w$.

Corollary 1. For any fixed vector of languages that is a solution of the system of equations corresponding to the grammar and for any derivation, the sequence of values of the formulas forming the derivation monotonically decreases.

According to Theorem 2, for any grammar G, the corresponding system of equations has a least solution. It turns out that this solution can naturally be represented in terms of the original grammar.

Theorem 4. For any conjunctive grammar $G = (\Sigma, \{X_1, \ldots, X_n\}, P, X_1)$, the vector of languages

$$(L_G(X_1), \ldots, L_G(X_n)) \qquad (15)$$

is the least solution of the system of equations corresponding to the grammar G.

Proof. The fact that (15) is a solution of the system follows from Theorem 1. Let us prove that this is the least solution.

Consider an arbitrary solution $L' = (L'_1, \ldots, L'_n)$ of the system $X = P(X)$. Let $X_i \Rightarrow^* w$ for some $w \in \Sigma^*$ and $X_i \in N$. In accordance with Lemma 5, it follows that $w(L') \subseteq X_i(L')$ and, hence, $w \in L'_n$.

Thus, we proved that $(L_G(X_1), \ldots, L_G(X_n)) \preccurlyeq L'$ and, therefore, solution (15) is the least solution of the system. □

The characterization of conjunctive grammars given by Theorem 4 makes it possible to prove that the solution uniqueness problem for a given system is algorithmically unsolvable.

Theorem 5. The set of systems of language equations that have only one solution is not recursively enumerable.

Proof. Let us assume that there exists a Turing machine M that stops when it gets on its input a system of language equations that has only one solution and does not terminate on any other inputs.

Consider another Turing machine M'. The machine M' gets on its input a description of a linear conjunctive grammar $G = (\Sigma, N, P, S)$; transforms it to the linear normal form, obtaining a grammar $G_1 = (\Sigma, N_1, P_1, S_1)$; constructs the system of equations $X = P(X)$ corresponding to G_1, where $X = (X_1, \ldots, X_n)$, $P = (\varphi_1, \ldots, \varphi_n)$, and the variable X_1 corresponds to the nonterminal S_1; and, by this system, constructs a new system $X' = P'(X')$, in which $X' = (X_0, X_1, \ldots, X_n)$, $P' = (\varphi_0, \varphi_1, \ldots, \varphi_n)$, and $\varphi_0 = X_0 \,\&\, X_1$.

The system $X = P(X)$ is strict, since it is obtained from a grammar in the linear normal form. By Theorem 3, this system has only one solution. Denote it as $L = (L_1, \ldots, L_n)$. Then, any vector of the form $(L', L_1, \ldots, L_n)$ such that $L' \subseteq L_1$ is a solution of the system $X' = P'(X')$; hence, the system $X' = P'(X')$ has only one solution if and only if $L_1 = \emptyset$.

By the construction of the system $X = P(X)$, $L_1 = L_{G_1}(S_1) = L(G_1)$. By the construction of the grammar G_1, $L(G_1) = L(G)$. Therefore, the machine M' stops if and only if $L(G) = \emptyset$, which implies that the set of linear conjunctive grammars generating an empty language is recursively enumerable. However, it is known that this set is not recursively enumerable [3]. The contradiction obtained proves the theorem. □

In conclusion, we give a practical example of using systems of language equations for the analysis of conjunctive grammars.

Example 1. We prove that the conjunctive grammar

$$S \longrightarrow aCa\&aA\&aD \mid aCb\&aA\&aD \mid bCa\&bB\&bD \mid bCb\&bB\&bD \mid c$$

$$C \longrightarrow aCa \mid aCb \mid bCa \mid bCb \mid c$$

$$D \longrightarrow aA\&aD \mid bB\&bD \mid cE$$

$$A \longrightarrow aAa \mid aAb \mid bAa \mid bAb \mid cEa$$

$$B \longrightarrow aBa \mid aBb \mid bBa \mid bBb \mid cEb$$

$$E \longrightarrow aE \mid bE \mid \epsilon$$

generates the language $\{wcw | w \in \{a, b\}^*\}$. For this, we consider the system of equations (which is written in exactly the same form as the grammar with the substitution of the equality sign for the arrow) corresponding to the grammar and prove that the vector of languages

$$\begin{aligned} L = (&\{wcw | w \in \{a, b\}^*\}, \\ &\{ucv | u, v \in \{a, b\}^*; |u| = |v|\}, \\ &\{uczu | u, z \in \{a, b\}^*\}, \\ &\{xcvay | x, v, y \in \{a, b\}^*; |x| = |y|\}, \\ &\{xcvby | x, v, y \in \{a, b\}^*; |x| = |y|\}, \\ &\{a, b\}^*) \end{aligned} \tag{16}$$

is the least solution of this system. The desired result then immediately follows.

The fact that vector (16) is a solution of the system is easily checked by simple substitution of languages into all equations. For example, for the equation in the variable D, we have

$$\begin{aligned} &(aA\&aD \mid bB\&bD \mid cE)(L) \\ &= (a\{xcvay\} \cap a\{uczu\}) \cup \\ &\cup (b\{xcvby\} \cap b\{uczu\}) \cup c\{a, b\}^* \\ &= \underbrace{a(\{xcvay\} \cap \{uczu\})}_{\{uczu \mid u = au'\}} \cup \\ &\cup \underbrace{b(\{xcvby\} \cap \{uczu\})}_{\{uczu \mid u = bu'\}} \cup \underbrace{c\{a, b\}^*}_{\{uczu \mid u = \epsilon\}} \\ &= \{uczu \mid u, z \in \{a, b\}^*\}. \end{aligned}$$

Note now that the considered system of equations is strict. Therefore, on the strength of Theorem 3, (16) is the only and, hence, the least solution of this system.

REFERENCES

1. Okhotin, A.S., On Augmenting the Formalism of Context-free Grammars with an Intersection Operation, *Proc. of the IV Int. Conf. "Discrete Models in the Control System Theory,"* 2000, pp. 106–109.
2. Okhotin, A., Conjunctive Grammars, *Pre-proceedings of DCAGRS 2000*; *Techn. report of Dept. of Computer Sci-*

ence, University of Western Ontario, London, Ontario, Canada, 2000, no. 555.

3. Okhotin, A., Conjunctive Grammars, *J. Automata, Languages and Combinatorics*, 2001, vol. 6, no. 4, pp. 519–535.
4. Okhotin, A., Top-Down Parsing of Conjunctive Languages, *Grammars*, 2002, vol. 5, no. 1, pp. 21–40.
5. Okhotin, A., A Recognition and Parsing Algorithm for Arbitrary Conjunctive Grammars, *Theor. Comput. Sci. C*, in press.
6. Okhotin, A., LR Parsing for Conjunctive Grammars, *Grammars*, 2002, vol. 5, no. 2, pp. 81–125.
7. Okhotin, A.S., On *P*-Completeness of the Membership Problem for Conjunctive Grammars, *Proc. of the Int. Workshop on Discrete Mathematics and Mathematical Cybernetics*, Ratmino, 2001.
8. Sudborough, I.H., A Note on Tape-bounded Complexity Classes and Linear Context-free Languages, *J. ACM*, 1975, vol. 22, no. 4, pp. 499–500.
9. Autebert, J., Berstel, J., and Boasson, L., Context-Free Languages and Pushdown Automata, in *Handbook of Formal Languages*, Berlin: Springer, 1997, vol. 1, pp. 111–174.
10. Kuich, W., Semirings and Formal Power Series: Their Relevance to Formal Language and Automata, in *Handbook of Formal Languages*, Berlin: Springer, 1997, vol. 1, pp. 609–677.