Tommi Lehtinen | Alexander Okhotin

Boolean grammars are closed under inverse gsm mappings

Turku Centre *for* Computer Science

TUCS Technical Report
No 911, October 2008

Boolean grammars are closed under inverse gsm mappings

Tommi Lehtinen
Department of Mathematics, University of Turku
Turku FIN–20014, Finland
tojleht@utu.fi

Alexander Okhotin
Academy of Finland, *and*
Department of Mathematics, University of Turku, *and*
Turku Centre for Computer Science
Turku FIN–20014, Finland
alexander.okhotin@utu.fi

TUCS Technical Report
No 911, October 2008

Abstract

It is proved that for every Boolean grammar G and for every generalized sequential machine M, the set $M^{-1}(L(G))$ of pre-images of words generated by G is generated by a Boolean grammar, which can be effectively constructed. Furthermore, if G is unambiguous, the constructed grammar is unambiguous as well. These results extend to conjunctive grammars.

Keywords: Boolean grammars, conjunctive grammars, closure properties, gsm mappings

TUCS Laboratory
Discrete Mathematics for Information Technology

1 Introduction

Boolean grammars [11] are an extension of the context-free grammars, in which the rules may contain explicit Boolean operations. The extended expressive power and the intuitive clarity of the new operations make these grammars a much more powerful tool for specifying languages than the context-free grammars. Another important fact is that the main context-free parsing algorithms, such as the Cocke–Kasami–Younger, the recursive descent and the generalized LR, can be extended to Boolean grammars without increasing their computational complexity [11, 12].

Though the Boolean grammars easily inherit many good practical properties of context-free grammars, their theoretical properties present a greater challenge to a researcher. No methods of proving any limitations of Boolean grammars are known up to date, and the languages they generate still could not be separated from their complexity-theoretic upper bound, $\mathit{DTIME}(n^3) \cap \mathit{DSPACE}(n)$ [11].

Also quite little progress has been made on the closure properties of the languages generated by Boolean grammars. This is the question of whether applications of certain operations to these languages always yield languages generated by Boolean grammars. Boolean grammars are trivially closed under Boolean operations and concatenation, since all these operations are included in their formalism. The same can be said with respect to star, which can be expressed by iterating a single nonterminal, as in the context-free case. Unlike the context-free languages, the languages generated by Boolean grammars are not closed under homomorphisms: in fact, all recursively enumerable languages can be obtained as homomorphic images of languages generated by a subclass of Boolean grammars, the *linear conjunctive grammars* [2, 10]. The closure under non-erasing homomorphisms remains an open problem.

Among the standard operations on languages are inverse homomorphisms and the more general *inverse gsm mappings*. These are pre-images of languages under mappings $M : \Sigma^* \to \Gamma^*$ implemented by deterministic transducers (generalized sequential machines, gsm). The pre-image of a language $L \subseteq \Gamma^*$ is defined as $M^{-1}(L) = \{w \in \Sigma^* \mid M(w) \in L\}$. It is known from Ginsburg and Rose [3] that context-free languages are closed under inverse gsm mappings. This argument was adapted to unambiguous context-free languages by Ginsburg and Ullian [4]. More accessible proofs of these results based upon pushdown automata were given by Harrison [5]. An examination of this argument shows that it also applies to linear context-free languages, which are consequently closed under inverse gsm mappings. The aforementioned linear conjunctive languages are closed under this operation by an argument due to Ibarra and Kim [7] done in terms of trellis automata [1].

This paper investigates the closure of Boolean grammars under inverse gsm mappings. It is established that for every Boolean grammar G over an alphabet Γ and for every gsm mapping $M : \Sigma^* \to \Gamma^*$, the language $M^{-1}(L(G))$ of pre-images of words generated by G is generated by a Boolean grammar. Since no automaton representation for Boolean grammars is known, the grammar for $M^{-1}(L(G))$ is constructed directly from G. Furthermore, if the Boolean grammar G is *unambiguous* [13], then the constructed grammar for $M^{-1}(L(G))$ is unambiguous as well, and if G does not use negation (that is, it is a *conjunctive grammar* [9]), then negation can be eliminated in the constructed grammar.

Thus four language families are shown to be closed under inverse gsm mapping: these are languages generated by Boolean grammars, unambiguous Boolean grammars conjunctive grammars and unambiguous conjunctive grammars. As a direct corollary, these families are also seen to be closed under inverse homomorphism.

2 Definition of Boolean grammars

Definition 1 ([11])**.** *A Boolean grammar is a quadruple $G = (\Sigma, N, P, S)$, where Σ and N are disjoint finite nonempty sets of terminal and nonterminal symbols respectively; P is a finite set of* rules *of the form*

$$A \to \alpha_1 \& \ldots \& \alpha_m \& \neg\beta_1 \& \ldots \& \neg\beta_n, \tag{1}$$

where $m+n \geqslant 1$, $\alpha_i, \beta_i \in (\Sigma \cup N)^*$; $S \in N$ is the start symbol of the grammar.

For each rule (1), the terms α_i and $\neg\beta_j$ (for all i, j) are called *conjuncts*, *positive* and *negative* respectively. A conjunct with any sign is denoted $\pm\gamma$. Occasionally conjuncts will be written together with the left-hand sides of the rules from which they originate, as $A \to \alpha_i$, $A \to \neg\beta_j$ or $A \to \pm\gamma$. The entire right-hand side of a rule (1) will sometimes be denoted by φ, and the whole rule by $A \to \varphi$.

A Boolean grammar is called a *conjunctive grammar* [9], if negation is never used, that is, $n = 0$ for every rule (1). It is a *context-free grammar* if neither negation nor conjunction are allowed, that is, $m = 1$ and $n = 0$ for each rule. Another important particular case of Boolean grammars is formed by *linear conjunctive grammars*, in which every conjunct is of the form $A \to uBv$ or $A \to w$, with $u, v, w \in \Sigma^*$, $A \in N$. Linear conjunctive grammars are equal in power to *linear Boolean grammars* with conjuncts $A \to \pm uBv$ or $A \to w$, as well as to trellis automata, also known as one-way real-time cellular automata [1, 10].

Intuitively, a rule (1) of a Boolean grammar can be read as follows: every string w over Σ that satisfies each of the syntactical conditions represented by $\alpha_1, \ldots, \alpha_m$ and none of the syntactical conditions represented by $\beta_1, \ldots, \beta_m$ therefore satisfies the condition defined by A. Though this is not yet a formal definition, this understanding is sufficient to construct grammars.

Example 1. *The following grammar generates the language* $\{a^n b^n c^n | n \geqslant 0\}$*:*

$$\begin{array}{lcl} S & \rightarrow & AB\&DC \\ A & \rightarrow & aA \mid \varepsilon \\ B & \rightarrow & bBc \mid \varepsilon \\ C & \rightarrow & cC \mid \varepsilon \\ D & \rightarrow & aDb \mid \varepsilon \end{array}$$

This grammar, which is actually conjunctive, represents this language as an intersection of two context-free languages:

$$\underbrace{\{a^n b^n c^n \mid n \geqslant 0\}}_{L(S)} = \underbrace{\{a^i b^j c^k \mid j = k\}}_{L(AB)} \cap \underbrace{\{a^i b^j c^k \mid i = j\}}_{L(DC)}$$

A related non-context-free language can be specified by inverting the sign of one of the conjuncts in this grammar.

Example 2. *The following Boolean grammar generates the language* $\{a^m b^n c^n \mid m, n \geqslant 0, m \neq n\}$*:*

$$\begin{array}{lcl} S & \rightarrow & AB\&\neg DC \\ A & \rightarrow & aA \mid \varepsilon \\ B & \rightarrow & bBc \mid \varepsilon \\ C & \rightarrow & cC \mid \varepsilon \\ D & \rightarrow & aDb \mid \varepsilon \end{array}$$

This grammar is based upon the following representation.

$$\underbrace{\{a^n b^m c^m \mid m, n \geqslant 0, m \neq n\}}_{L(S)} = \{a^i b^j c^k \mid j = k \text{ and } i \neq j\} = L(AB) \cap \overline{L(DC)}$$

Example 3. *The following Boolean grammar generates the language* $\{ww \mid w \in \{a, b\}^*\}$*:*

$$\begin{array}{lcl} S & \rightarrow & \neg AB\&\neg BA\&C \\ A & \rightarrow & XAX \mid a \\ B & \rightarrow & XBX \mid b \\ C & \rightarrow & XXC \mid \varepsilon \\ X & \rightarrow & a \mid b \end{array}$$

According to the intuitive semantics of Boolean grammars described above, the nonterminals A, B, C and X generate context-free languages

$$L(A) = \{uav \mid u, v \in \{a, b\}^*, |u| = |v|\},$$
$$L(B) = \{ubv \mid u, v \in \{a, b\}^*, |u| = |v|\}.$$

Then

$$L(AB) = \{uavxby \mid u, v, x, y \in \{a, b\}^*, |u| = |x|, |v| = |y|\},$$

in other words, $L(AB)$ is the set of all strings of even length with a mismatch a on the left and b on the right (in any position). Similarly,

$$L(BA) = \{ubvxay \mid u, v, x, y \in \{a, b\}^*, |u| = |x|, |v| = |y|\}$$

specifies the mismatch formed by b on the left and a on the right. Then the rule for S specifies the set of strings of even length without such mismatches:

$$L(S) = \overline{L(AB)} \cap \overline{L(BA)} \cap \{aa, ab, ba, bb\}^* = \{ww \mid w \in \{a, b\}^*\}.$$

A formal definition of the language generated by a Boolean grammar. can be given in several different ways [8, 11], which ultimately yield the same class of languages. We shall use the most straightforward of these definitions, which begins with the interpretation of a grammar as a system of equations with formal languages as unknowns:

Definition 2. *Let $G = (\Sigma, N, P, S)$ be a Boolean grammar. The system of language equations associated with G is a resolved system of language equations over Σ in variables N, in which the equation for each variable $A \in N$ is*

$$A = \bigcup_{A \to \alpha_1 \& \ldots \& \alpha_m \& \neg\beta_1 \& \ldots \& \neg\beta_n \in P} \left[\bigcap_{i=1}^{m} \alpha_i \cap \bigcap_{j=1}^{n} \overline{\beta_j} \right] \qquad (2)$$

Each instance of a symbol $a \in \Sigma$ in such a system defines a constant language $\{a\}$, while each empty string denotes a constant language $\{\varepsilon\}$. 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 (2) into an equality.

Now the following restriction is imposed upon these equations, so that their solutions can be used to define the languages generated by grammars:

Definition 3. *Let $G = (\Sigma, N, P, S)$ be a Boolean grammar, let (2) be the associated system of language equations. Suppose that for every finite language $M \subset \Sigma^*$ (such that for every $w \in M$ all substrings of w are also in M) there exists a unique vector of languages $(\ldots, L_C, \ldots)_{C \in N}$ ($L_C \subseteq M$), such that a substitution of L_C for C, for each $C \in N$, turns every equation (2) into an equality modulo intersection with M.*

Then, for every $A \in N$, the language $L_G(A)$ is defined as L_A, while the language generated by the grammar is $L(G) = L_G(S) = L_S$.

There exists an unambiguous subclass of Boolean grammars, which generalizes unambiguous context-free grammars.

Definition 4. *A Boolean grammar $G = (\Sigma, N, P, S)$ is unambiguous if*

I. *Different rules for every single nonterminal A generate disjoint languages, that is, for every string w there exists at most one rule*

$$A \to \alpha_1 \& \ldots \& \alpha_m \& \neg\beta_1 \& \ldots \& \neg\beta_n,$$

such that $w \in L_G(\alpha_1) \cap \ldots \cap L_G(\alpha_m) \cap \overline{L_G(\beta_1)} \cap \ldots \cap \overline{L_G(\beta_n)}$.

II. *All concatenations are unambiguous, that is, for every conjunct* $A \to \pm s_1 \ldots s_\ell$ *and for every string w there exists at most one factorization* $w = u_1 \ldots u_\ell$, *such that* $u_i \in L_G(s_i)$ *for all i.*

While the languages generated by Boolean grammars can be recognized in cubic time [11] and no better upper bound is known, unambiguous Boolean grammars allow square-time parsing [13]. However, no proofs of inherent ambiguity of any languages generated by Boolean grammars are known. It is known that all linear conjunctive languages have unambiguous grammars.

The relation between the families of languages generated by Boolean grammars (*Bool*), conjunctive grammars (*Conj*) and linear conjunctive grammars (*LinConj*), their unambiguous variants (*UnambBool* and *UnambConj*), as well as other common families of formal languages, is shown in Figure 1 [13]. The rest of the classes in the figure are regular (*Reg*), linear context-free (*LinCF*), context-free (*CF*) and deterministic context-sensitive languages (*DetCS*).

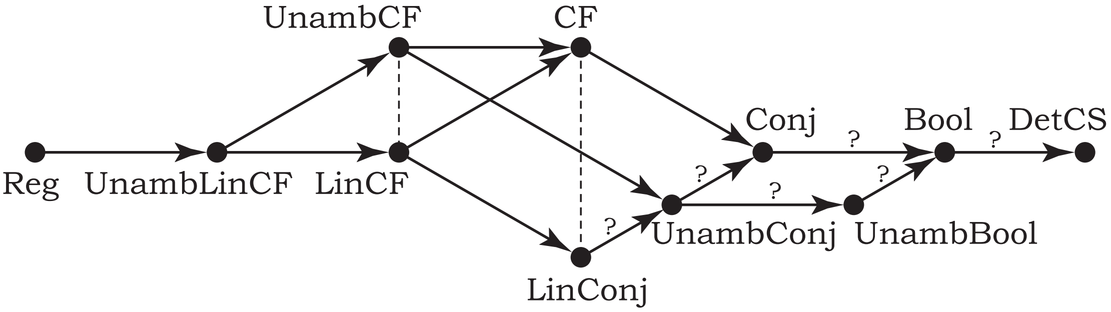

Figure 1: The hierarchy of language families.

The following normal form for Boolean grammars, which generalizes Chomsky normal form for the context-free grammars, is known.

Definition 5. *A Boolean grammar* $G = (\Sigma, N, P, S)$ *is in the binary normal form if every rule in P is of the form*

$A \to B_1C_1 \& \ldots \& B_mC_m \& \neg D_1E_1 \& \ldots \& \neg D_nE_n \& \neg\varepsilon \quad (m \geqslant 1, n \geqslant 0)$

$A \to a$

$S \to \varepsilon$ (only if S does not appear in right-hand sides of rules)

Every grammar of this form is well-defined in the sense of Definition 3, as well as according to other definitions of Boolean grammars [8, 11].

Proposition 1 ([11, 13]). *For every Boolean grammar there exists and can be effectively constructed a Boolean grammar in the binary normal form generating the same language. Furthermore, if the given grammar is unambiguous, then so is the constructed grammar.*

Definition 6 (Ginsburg and Rose [3]; Harrison [5]). *A (deterministic) generalized sequential machine (gsm) is a septuple $M = (\Sigma, \Gamma, Q, q^0, \delta, \lambda, F)$, where*

Σ *is a finite nonempty* input alphabet*,*

Γ *is a finite nonempty* output alphabet*,*

Q *is a finite nonempty set of* states*,*

$q^0 \in Q$ *is the* start state*,*

$\delta : Q \times \Sigma \to Q$ *is the* transition function*,*

$\lambda : Q \times \Sigma \to \Gamma^*$ *is the* output function*, and*

$F \subseteq Q$ *is the set of* final states*.*

The functions δ and λ are extended to $Q \times \Sigma^*$ in the usual way, as $\delta(q, \varepsilon) = q$, $\delta(q, aw) = \delta(\delta(q, a), w)$ and as $\lambda(q, \varepsilon) = \varepsilon$, $\lambda(q, aw) = \lambda(q, a)\lambda(\delta(q, a), w)$. A gsm M computes a partial function $M : \Sigma^* \to \Gamma^*$, where $M(w) = \lambda(q^0, w)$ and $\delta(q^0, w) \in F$.

We are interested in the inverse image under this partial mapping M, that is, for a language $L \subseteq \Gamma^*$ we have the inverse image $M^{-1}(L) = \{w \in \Sigma^* \mid M(w) \in L\}$.

3 Closure under inverse gsm mappings

Theorem 1. *For every generalized sequential machine M implementing a function $M : \Sigma^* \to \Gamma^*$ and for every Boolean (conjunctive, unambiguous Boolean, unambiguous conjunctive) grammar G over Γ there exists and can be effectively constructed a Boolean (conjunctive, unambiguous Boolean, unambiguous conjunctive) grammar over Σ generating the language $M^{-1}(L(G))$.*

3.1 The form of a gsm

Our construction requires separating the transitions that output ε, since these are handled differently from those that output something non-empty. In order to do this, we will assume that the state set of M is divided into two separate sets Q_ε and $Q_{\neg\varepsilon}$, such that

$$\delta(q, a) \in \begin{cases} Q_\varepsilon, & \text{if } \lambda(q, a) = \varepsilon \\ Q_{\neg\varepsilon}, & \text{if } \lambda(q, a) \neq \varepsilon. \end{cases}$$

This can be done by duplicating the original state set Q into two disjoint sets $Q_\varepsilon = \{q_\varepsilon \mid q \in Q\}$ and $Q_{\neg\varepsilon} = \{q_{\neg\varepsilon} \mid q \in Q\}$ and defining for all $q \in Q$

$$\lambda(q_\varepsilon, a) = \lambda(q_{\neg\varepsilon}, a) = \lambda(q, a)$$

and

$$\delta(q_\varepsilon, a) = \delta(q_{\neg\varepsilon}, a) = \begin{cases} \delta(q, a)_\varepsilon, & \text{if } \lambda(q, a) = \varepsilon \\ \delta(q, a)_{\neg\varepsilon}, & \text{if } \lambda(q, a) \neq \varepsilon. \end{cases}$$

Final states are the duplicates of states in F, denoted by F_ε and $F_{\neg\varepsilon}$, and as the start state we choose $q^0_{\neg\varepsilon}$. Now the gsm always remembers (in its current state) whether the previous symbol read had an empty or non-empty image. This property will be used in the below construction.

3.2 Construction of the grammar

Let Σ and Γ be finite alphabets, let $M = (\Sigma, \Gamma, Q_\varepsilon \cup Q_{\neg\varepsilon}, q^0, \delta, \lambda, F)$ be a generalized sequential machine as defined above. Assume that $\lambda(q, a) \neq \varepsilon$ for some $q \in Q$ and $a \in \Sigma$; otherwise $M^{-1}(L(G)) = \{w \in \Sigma^* \mid M(w) = \varepsilon\}$ if $\varepsilon \in L(G)$ and $M^{-1}(L(G)) = \varnothing$ if $\varepsilon \notin L(G)$, and either language is regular.

Let $G = (\Gamma, N, P, S)$ be a Boolean grammar in the binary normal form without a rule $S \to \varepsilon$. The case of the empty word can be handled using the equality $M^{-1}(L(G) \cup \{\varepsilon\}) = M^{-1}(L(G)) \cup \{w \in \Sigma^* \mid M(w) = \varepsilon\}$. Now a grammar $G' = (\Sigma, N', P', S')$ for the language $M^{-1}(L(G))$ is constructed as follows.

The set of nonterminals of G' is $N' = N'_1 \cup N'_2 \cup T' \cup \{S'\}$, where S' is a distinguished start symbol. The subset T' is defined as

$$T' = \{T_{q,q'} \mid q \in Q,\ q' \in Q\} \cup \{\widetilde{T}_q \mid q \in Q\},$$

These nonterminals should generate the languages $L_{G'}(T_{q,q'}) = \{w \in \Sigma^* \mid \delta(q, w) = q',\ \lambda(q, w) = \varepsilon\}$ and $L_{G'}(\widetilde{T}_q) = \{a \in \Sigma \mid \lambda(q, a) \neq \varepsilon\}\Sigma^*$. These languages are regular, so there obviously exists an unambiguous linear context-free grammar generating them. We omit the explicit construction of the corresponding set of rules.

Turning to the sets of nonterminals N'_1 and N'_2, let us first define two sets of *margins:*

$$\begin{aligned} \text{suff} &= \{x \mid x \text{ a proper suffix of some } \lambda(q, a),\ q \in Q,\ a \in \Sigma\} \quad \text{and} \\ \text{pref} &= \{y \mid y \text{ a proper prefix of some } \lambda(q, a),\ q \in Q,\ a \in \Sigma\}. \end{aligned}$$

Now the set of nonterminals contains quintuples

$$N'_1 = \{(x, q, A, q', y) \mid x \in \text{suff},\ q \in Q_\varepsilon \cup Q_{\neg\varepsilon},\ A \in N,\ q' \in Q_{\neg\varepsilon},\ y \in \text{pref}\}$$

and sextuples

$$N'_2 = \{(x, q, B, C, q', y) \mid x \in \text{suff},\ q \in Q_\varepsilon \cup Q_{\neg\varepsilon},\ B, C \in N,\ q' \in Q_{\neg\varepsilon},\ y \in \text{pref}\},$$

where BC or $\neg BC$ appears as a conjunct in P. Each of these nonterminals should generate a word w if and only if $\delta(q, w) = q'$ and $x\lambda(q, w)y \in L_G(A)$ in the first case and $x\lambda(q, w)y \in L_G(BC)$ in the second case. Such words will be generated by different types of rules, depending on q, q' and w.

The rules

$$P'_\varepsilon = \{(x, q, A, q, y) \to \varepsilon \mid (x, q, A, q, y) \in N'_1;\ xy \in L_G(A)\}$$

are the only rules for nonterminals in N'_1 that generate ε, which will be stated in Lemma 3. Note that $xy \neq \varepsilon$ in any such rule because $\varepsilon \notin L_G(A)$ for every $A \in N$.

The rules

$$P'_1 = \{(\varepsilon, q, A, \delta(q, a), \varepsilon) \to a \mid \lambda(q, a) \in \Gamma;\ A \to \lambda(q, a) \in P\}$$

correspond to rules of the form $A \to a$ in P.

The rules

$$P'_0 = \{(x, q, A, q', y) \to a(x, \delta(q, a), A, q', y) \mid (x, q, A, q', y) \in N'_1;\ \lambda(q, a) = \varepsilon\}$$

generate the words that begin with a symbol with an empty image when the mapping starts from q.

For every nonterminal $(x, q, B, C, q', y) \in N'_2$ the set P' contains a rule $(x, q, B, C, q', y) \to \alpha \& \neg\varepsilon$ for each α belonging to any of the following four sets:

$$\{(x', q, C, q', y) \mid x = x''x' :\ x'' \in L(B),\ x' \in \Sigma^+\}, \tag{3a}$$
$$\{(x, q, B, q'', \varepsilon)(\varepsilon, q'', C, q', y) \mid q'' \in Q_{\neg\varepsilon}\}, \tag{3b}$$
$$\{(x, q, B, q'', y')T_{q'',q'''}a(x', \delta(q''', a), C, q', y) \mid x', y' \in \Gamma^+;\ a \in \Sigma;\ \lambda(q''', a) = y'x'\}, \tag{3c}$$
$$\{(x, q, B, q', y') \mid y = y'y'' :\ y' \in \Sigma^+,\ y'' \in L(C)\}. \tag{3d}$$

In the case of $q \in Q_\varepsilon$ and $x \in L_G(B)$, the set (3b) also contains a conjunct

$$(\varepsilon, q, C, q', y), \tag{3b$'$}$$

Similarly, if $q \in Q_\varepsilon$, then the set (3c) also has the conjuncts

$$\{T_{q,q'''}a(x', \delta(q''', a), C, q', y) \mid x', y' \in \Gamma^+;\ a \in \Sigma;\ \lambda(q''', a) = y'x';\ xy' \in L_G(B)\}. \tag{3c$'$}$$

These sets of conjuncts correspond to different factorizations of $x\lambda(q, w)y$ into $L_G(B) \cdot L_G(C)$, as illustrated in Figure 2.

For all $\varphi = B_1C_1 \& \ldots \& B_mC_m \& \neg D_1E_1 \& \ldots \& \neg D_nE_n \& \neg\varepsilon$ appearing as the right-hand side of a rule in P we write

$$(x, q, \varphi, q', y) = \mathop{\&}_{1 \leqslant i \leqslant m} (x, q, B_i, C_i, q', y) \& \mathop{\&}_{1 \leqslant j \leqslant n} \neg(x, q, D_j, E_j, q', y) \& \widetilde{T}_q$$

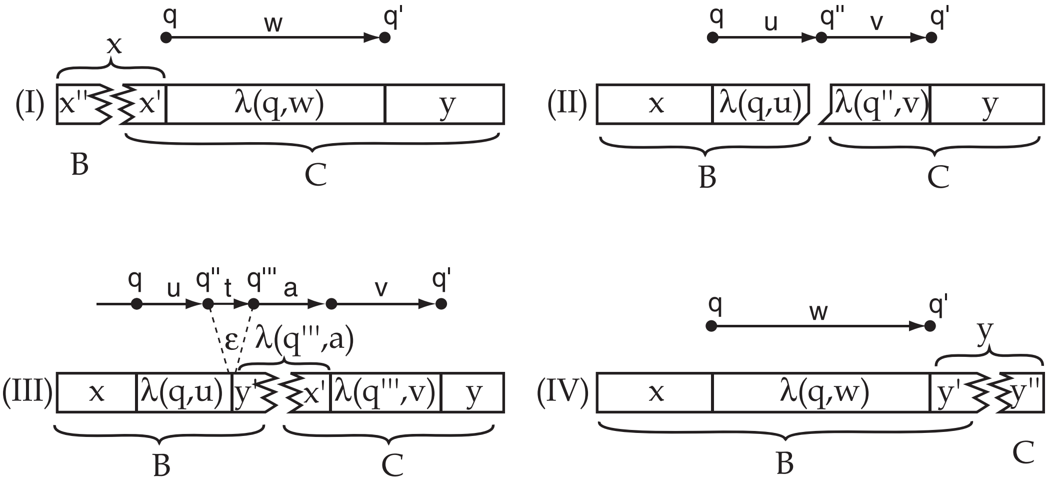

Figure 2: Factorizations of $x\lambda(q,w)y$.

Now we can define

$$P'_{\text{long}} = \{(x, q, A, q', y) \to (x, q, \varphi, q', y) \mid A \to \varphi \in P\}.$$

$$P'_S = \{S' \to (\varepsilon, q^0_{\neg\varepsilon}, S, q, \varepsilon)T_{q,f} \mid q \in Q_{\neg\varepsilon};\ f \in F\}$$

$$P' = P'_0 \cup P'_1 \cup P'_\varepsilon \cup P'_{\text{long}} \cup P'_S.$$

It should be noted that these sets of rules are disjoint. Furthermore, if the original grammar G is conjunctive, then no new negations are added, except for the conjuncts $\neg\varepsilon$. The latter can be replaced by positive conjuncts C, where C generates the regular language Σ^+. So, in this case the constructed grammar G' is conjunctive as well.

3.3 Correctness of the construction

First we have to prove that the grammar G' defines a language under the chosen semantics of strongly unique solution.

Lemma 1. *The system of equations corresponding to G' has a strongly unique solution.*

Proof. Let K be a subword-closed language, it has to be proved that the solution modulo K is unique. The proof is induction on $|K|$. The nonterminals $T_{q,q'}$ and T_q generate regular languages, and thus can be assumed to have a unique solution modulo every language. The uniqueness of the solutions for S' follow from the uniqueness of solutions for nonterminals in N'_1 and T', since S' doesn't appear in the right side of any rule.

Induction basis: $K = \{\varepsilon\}$. For nonterminals in N_1', the unique solution modulo K has $(x, q, A, q'', y) = \{\varepsilon\}$ if $q = q'$ and $xy \in L_G(A)$ (according to a rule from P_ε') and $(x, q, A, q'', y) = \varnothing$ otherwise (since the rules from P_1' and P_{long}' cannot generate ε, the latter due to a conjunction with $\widetilde{T}_q$). For all $(x, q, B, C, q', y) \in N_2'$ the unique solution is $(x, q, B, C, q', y) = \varnothing$, because all rules for these nonterminals contain a conjunction with $\neg\varepsilon$.

Induction hypothesis: The solution modulo K' is unique.

Induction step: Let $K = \{w\} \cup K'$, where $w \notin K'$, but all subwords of w are in K'. Suppose that the solution modulo K is not unique. Then there are two different solutions L and L', such that $w \in L_X$ and $w \notin L_X'$ for some $X \in N_1' \cup N_2'$. Choose a nonterminal with minimal margins $|xy|$. First let $X = (x, q, A, q', y) \in N_1'$. Now if w is generated by a rule in P_1', then it would be in $L'_{(x,q,A,q',y)}$ by the same rule. If it were generated by a rule $(x, q, A, q', y) \to a(x, \delta(q, a), A, q', y)$ from P_0', then $w = aw'$, with $w' \in L_{(x,\delta(q,a),A,q',y)}$. Since $w' \in K'$, it follows that $w' \in L'_{(x,\delta(q,a),A,q',y)}$ and $w \in L'_{(x,q,A,q',y)}$. If w is generated by a rule in P_{long}', then the solutions differ on a nonterminal $(x, q, B, C, q', y) \in N_2'$. There are four possible types of rules $(x, q, B, C, q', y) \to \alpha \& \neg\varepsilon$:

(3a) If $\alpha = (x', q, C, q', y)$, then $|x'y| < |xy|$ and $L_{(x',q,C,q',y)} = L'_{(x',q,C,q',y)}$ by the choice of X.

(3b) In the case $\alpha = (x, q, B, q'', \varepsilon)(\varepsilon, q'', C, q', y)$ we have that $w \in L_{(x,q,B,q'',\varepsilon)} L_{(\varepsilon,q'',C,q',y)}$ and $w \notin L'_{(x,q,B,q'',\varepsilon)} L'_{(\varepsilon,q'',C,q',y)}$. So $w = uv$, with $u \in L_{(x,q,B,q'',\varepsilon)}$ and $v \in L_{(\varepsilon,q'',C,q',y)}$.

If $u = \varepsilon$, then, by the induction hypothesis, $\varepsilon \in L'_{(x,q,B,q'',\varepsilon)}$. Then ε must be generated by a rule from P_ε', and therefore $x \neq \varepsilon$ by the construction of these rules. In addition, $v = w \in L_{(\varepsilon,q'',C,q',y)}$, and therefore $w \notin L'_{(\varepsilon,q'',C,q',y)}$ (because otherwise $w \in L_X'$). Since $x \neq \varepsilon$ and $|\varepsilon \cdot y| < |xy|$, $Y = (\varepsilon, q'', C, q', y)$ is a nonterminal with smaller margins with $w \in L_Y \setminus L_Y'$, which contradicts the choice of X. The case of $v = \varepsilon$ is symmetrical.

If $u, v \neq \varepsilon$, then $u, v \in K'$ and by the induction hypothesis $w = uv \in L'_{(x,q,B,q'',\varepsilon)} L'_{(\varepsilon,q'',C,q',y)}$.

If $\alpha = (\varepsilon, q, C, q', y)$, then $x \in L_G(B)$ by the construction, and thus $x \neq \varepsilon$. Then $L_{(\varepsilon,q,C,q',y)} = L'_{(\varepsilon,q,C,q',y)}$ by the induction hypothesis.

(3c) If $\alpha = (x, q, B, q'', y') T_{q'',q'''} a(x', \delta(q''', a), C, q', y)$, then $w = utav$, with $u \in L_{(x,q,B,q'',y')}$, $t \in L_{T_{q'',q'''}}$ and $v \in L_{(x',\delta(q''',a),C,q',y)}$. Since $u, t, v \in K'$, by the induction hypothesis $w \in L'_{(x,q,B,q'',y')} L'_{T_{q'',q'''}} a L'_{(x',\delta(q''',a),C,q',y)}$.

In the case $\alpha = T_{q,q'''} a(x', \delta(q''', a), C, q', y)$ we have $w = tav$, with $t \in L_{T_{q,q'''}}$ and $v \in L_{(x',\delta(q''',a),C,q',y)}$, where $t, v \in K'$. By the induction hypothesis, $w \in L'_{T_{q,q'''}} a L'_{(x',\delta(q''',a),C,q',y)}$.

(3d) Finally if $\alpha = (x, q, B, q', y')$, then $|xy'| < |xy|$ and $L_{(x,q,B,q',y')} = L'_{(x,q,B,q',y')}$ as in (3a). □

Lemma 2. *If $w \in L_{G'}\big((x, q, A, q', y)\big)$ or $w \in L_{G'}\big((x, q, B, C, q', y)\big)$, then $\delta(q, w) = q'$.*

Proof. Induction on lexicographically ordered pairs $(|w|, |xy|)$.

Basis: $w = \varepsilon$. Then w is generated by a rule from P'_ε, and hence $w = \varepsilon$, $q = q'$ and $\delta(q, \varepsilon) = q$.

Induction step: Let w be generated by a rule from P'_1. Then $w = a$ and $q' = \delta(q, w)$ by definition.

If w is generated by a rule from P'_0, then $w = aw'$, with $w' \in L_{G'}\big((x, \delta(q, a), A, q', y)\big)$. Then, by the induction hypothesis, $\delta(\delta(q, a), w') = q'$, and thus $\delta(q, w) = q'$.

Finally we have the possibility that w is generated by a rule in P'_{long}. The right-hand side has a positive conjunct (x, q, B, C, q', y) and thus $w \in L_{G'}(\alpha)$ for some $(x, q, B, C, q', y) \to \alpha \& \neg\varepsilon$. There are four possible types of conjuncts α.

In the cases (3a) and (3d), α consists of a single nonterminal with shorter margins and the same states. Then, by the induction hypothesis, $\delta(q, w) = q'$.

In the case (3b) $\alpha = (x, q, B, q'', \varepsilon)(\varepsilon, q'', C, q', y)$ or $\alpha = (\varepsilon, q, C, q', y)$. In the first case $w = uv$ with $u \in L_{G'}\big((x, q, B, q'', \varepsilon)\big)$ and $v \in L_{G'}\big((\varepsilon, q'', C, q', y)\big)$. If both u and v are non-empty, then, by the induction hypothesis, $\delta(q, u) = q''$ and $\delta(q'', v) = q'$, so $\delta(q, w) = q'$. Otherwise suppose $u = \varepsilon$, then it is generated by a rule from P'_ε, so $x \neq \varepsilon$ by the construction of P'_ε. Then $w \in L_{G'}\big((\varepsilon, q, C, q', y)\big)$ with $|\varepsilon y| < |xy|$, and so $\delta(q, w) = q'$ by the induction hypothesis. This is the case with $\alpha = (\varepsilon, q, C, q', y)$ also.

In the case (3c) $w = utav$ or $w = tav$, with $u \in L_{G'}\big((x, q, B, q'', y')\big)$, $t \in L_{G'}(T_{q'',q'''})$ and $v \in L_{G'}\big((x', \delta(q''', a), B, q', y)\big)$. Since $|u|, |v| < |w|$, the induction hypothesis asserts that $\delta(q, u) = q''$ and $\delta(\delta(q''', a), v) = q'$. By definition of $T_{q'',q'''}$ also $\delta(q'', t) = q'''$. Combining these we get $\delta(q, w) = q'$. This completes the proof of the last case. □

Next we will prove that $w \in L_{G'}\big((x, q, A, \delta(q, w), y)\big)$ if and only if $x\lambda(q, w)y \in L_G(A)$. Let us first prove results for words generated by rules in P'_ε, P'_0 and P'_1.

Lemma 3. *$\varepsilon \in L_{G'}\big((x, q, A, q', y)\big)$ if and only if $q = q'$ and $xy \in L_G(A)$. In this case ε can only be generated by a rule in P'_ε.*

Proof. The rules in P'_0, P'_1 or in any P'_{long} can't generate ε, so that $\varepsilon \in L_{G'}\big((x, q, A, q', y)\big)$ if and only if it is generated by a rule in P'_ε. Existence of this rule is equivalent to $q = q'$ and $xy \in L_G(A)$. □

Lemma 4. *If $w \in L_{G'}\big((x, q, A, q', y)\big)$, then $x\lambda(q, w)y \neq \varepsilon$.*

Proof. If $w = \varepsilon$, then, by Lemma 3, $xy \in L_G(A)$. Since G is in the normal form and $L_G(A) \subseteq \Sigma^+$, $xy \neq \varepsilon$ and the claim follows.

Assume $w \neq \varepsilon$. By Lemma 2, $\delta(q, w) = q'$, and $q' \in Q_{\neg\varepsilon}$ by the construction of N_1'. Now $\lambda(q, w) \neq \varepsilon$. □

Lemma 5. *If $a \in \Sigma$, $\lambda(q, a) = \varepsilon$ and $w \in \Sigma^*$, then $aw \in L_{G'}\big((x, q, A, q', y)\big)$ if and only if $w \in L_{G'}\big((x, \delta(q, a), A, q', y)\big)$. In this case aw can only be generated by a rule in P_0'.*

Proof. Suppose aw is generated by a rule for (x, q, A, q', y). The rules in P_ε' and P_1' clearly don't generate such words, and in the rules of P_{long}' there is a conjunct $\widetilde{T}_q$, with $aw \notin \widetilde{T}_q$. Therefore, w is generated by a rule $(x, q, A, q', y) \to a(x, \delta(q, a), A, q', y)$ in P_0', with $w \in L_{G'}\big((x, \delta(q, a), A, q', y)\big)$.

Conversely, if $w \in L_{G'}\big((x, \delta(q, a), A, q', y)\big)$, then $aw \in L_{G'}\big((x, q, A, q', y)\big)$ by a rule $(x, q, A, q', y) \to a(x, \delta(q, a), A, q', y)$. □

Lemma 6. *If $a \in \Sigma$, $\lambda(q, a) \in \Gamma$, then $a \in L_{G'}\big((\varepsilon, q, A, q', \varepsilon)\big)$ if and only if $\lambda(q, a) \in L_G(A)$. In this case a can only be generated by a rule in P_1'.*

Proof. Suppose a is generated by $(\varepsilon, q, A, q', \varepsilon)$. Rules in P_ε' only generate ε and any word generated by a rule in P_0' begins with a symbol with an empty image. If it were generated by a rule in P_{long}', then it would be in $L_{G'}\big((\varepsilon, q, B, C, q', \varepsilon)\big)$ for some B, C, such that BC is a conjunct in some long rule for A in G. There are no rules of the form (3a) or (3d) for $(\varepsilon, q, B, C, q', \varepsilon)$, and the rules of the form (3c) can't generate a, since $|\lambda(q, a)| = 1$, while the definition of (3c) requires the image of a to be of length at least 2. Also a cannot be generated by a rule of the form (3b$'$), because there is no such rule for $x = \varepsilon$ (since $\varepsilon \notin L_G(B)$). So a would have to be in $L_{G'}\big((\varepsilon, q, B, \delta(q, u), \varepsilon)(\varepsilon, \delta(q, u), C, q', \varepsilon)\big)$, and one of $(\varepsilon, q, B, \delta(q, u), \varepsilon)$ and $(\varepsilon, \delta(q, u), C, q', \varepsilon)$ would generate ε, which is impossible by Lemma 3. Thus a is generated by a rule in P_1'. Then $\lambda(q, a) \in L_G(A)$ by the construction of P_1'.

Conversely, if $\lambda(q, a) \in L_G(A)$, then there is a rule $(\varepsilon, q, A, q', \varepsilon) \to a$ in P_1' and thus $a \in L_{G'}\big((\varepsilon, q, A, q', \varepsilon)\big)$. □

Let us then continue proving the main result. We will do this by induction on the length of $x\lambda(q, w)y \in \Gamma^*$, since the rules in P' correspond to rules in P in a natural way. We will start with the basis:

Lemma 7. *Let $q \in Q$, $q' \in Q_{\neg\varepsilon}$, $x \in \text{pref}$, $y \in \text{suff}$ and $w \in \Sigma^*$ with $|x\lambda(q, w)y| \leqslant 1$. Then $w \in L_{G'}\big((x, q, A, q', y)\big)$ if and only if $x\lambda(q, w)y \in L_G(A)$ and $\delta(q, w) = q'$.*

Proof. First consider the case of $\delta(q, w) \in Q_\varepsilon$. Then $\delta(q, w) \neq q'$. By Lemma 2, this implies $w \notin L_{G'}\big((x, q, A, q', y)\big)$. Thus both sides of the equivalence are false. So it can be assumed that $\delta(q, w) \in Q_{\neg\varepsilon}$.

Consider the case of $\lambda(q,w) = \varepsilon$. If $w \neq \varepsilon$, then $\delta(q,w) \in Q_\varepsilon$ by the construction of the gsm, which contradicts the above assumption. Then $w = \varepsilon$ and the statement holds by Lemma 3.

In the remaining case of $x = y = \varepsilon$ and $|\lambda(q,w)| = 1$, the proof is an induction on $|w|$.

Basis: $w = a \in \Sigma$. Then the statement holds by Lemma 6.

Induction step: Let $w = bw'a$, with $b, a \in \Sigma$. Then $\lambda(\delta(q,bw'),a) \neq \varepsilon$ because $\delta(q,bw'a) \in Q_{\neg\varepsilon}$. It follows that the whole output on w comes from a and, in particular, $\lambda(q,b) = \varepsilon$. Therefore, by Lemma 5, $bw'a \in L_{G'}\big((\varepsilon,q,A,q',\varepsilon)\big)$ if and only if $w'a \in L_{G'}\big((\varepsilon,\delta(q,b),A,q',\varepsilon)\big)$, which, by the induction hypothesis, is equivalent to $\lambda(\delta(q,b),w'a) \in L_G(A)$ and $\delta(\delta(q,b),w'a) = q'$. This is in turn equivalent to $\lambda(q,w) \in L_G(A)$ and $\delta(q,w) = q'$. □

Lemma 8. *Let $k \geqslant 2$ be a natural number. Assume that for all $\tilde{x} \in \mathrm{pref}$, $\tilde{y} \in \mathrm{suff}$, $\tilde{q} \in Q$ and $\tilde{w} \in \Sigma^*$ with $|\tilde{x}\lambda(\tilde{q},\tilde{w})\tilde{y}| < k$ it holds that $\tilde{w} \in L_{G'}\big((\tilde{x},\tilde{q},\tilde{A},\tilde{q}',\tilde{y})\big)$ if and only if $\tilde{x}\lambda(\tilde{q},\tilde{w})\tilde{y} \in L_G(\tilde{A})$ and $\delta(\tilde{q},\tilde{w}) = \tilde{q}'$.*

Let $w \neq \varepsilon$ and $|x\lambda(q,w)y| = k$. Then:

1. *Let $(x,q,B,C,q',y) \in N_2'$ and consider the rules for this nonterminal.*

   (a) *w is generated by a rule $(x,q,B,C,q',y) \to (x',q,C,q',y)\&\neg\varepsilon$ if and only if $\delta(q,w) = q'$ and there exists a factorization $x\lambda(q,w)y = z_1z_2$ with $z_1 \in L_G(B)$, $z_2 \in L_G(C)$ and $|z_1| < |x|$.*

   (b) *w is generated by a rule $(x,q,B,C,q',y) \to (x,q,B,q'',\varepsilon)(\varepsilon,q'',C,q',y)\&\neg\varepsilon$ (or in the case $q \in Q_\varepsilon$ and $x \in L_G(B)$ possibly by a rule $(x,q,B,C,q',y) \to (\varepsilon,q,C,q',y)\&\neg\varepsilon$) if and only if $\delta(q,w) = q'$ and there exists a factorization $x\lambda(q,w)y = z_1z_2$ with $z_1 \in L_G(B)$, $z_2 \in L_G(C)$ and $|z_1| = |x\lambda(q,u)|$ for some prefix u of w.*

   (c) *w is generated by a rule $(x,q,B,C,q',y) \to (x,q,B,q'',y')T_{q'',q'''}a(x',\delta(q''',a),C,q',y)\&\neg\varepsilon$ (or in the case $q \in Q_\varepsilon$ and $xy' \in L_G(B)$ possibly by a rule $(x,q,B,C,q',y) \to T_{q,q'''}a(x',\delta(q''',a),C,q',y)\&\neg\varepsilon$) if and only if $\delta(q,w) = q'$ and there exists a factorization $x\lambda(q,w)y = z_1z_2$ with $z_1 \in L_G(B)$, $z_2 \in L_G(C)$ and $|x\lambda(q,u)| < |z_1| < |x\lambda(q,ua)|$, for some $u \in \Sigma^*$ and $a \in \Sigma$, such that ua is a prefix of w.*

   (d) *w is generated by a rule $(x,q,B,C,q',y) \to (x,q,B,q',y')\&\neg\varepsilon$ if and only if $\delta(q,w) = q'$ and there exists a factorization $x\lambda(q,w)y = z_1z_2$ with $z_1 \in L_G(B)$, $z_2 \in L_G(C)$ and $|x\lambda(q,w)| < |z_1|$.*

2. *$w \in L_{G'}\big((x,q,B,C,q',y)\big)$ if and only if $x\lambda(q,w)y \in L_G(BC)$ and $\delta(q,w) = q'$.*

3. *Let $A \to \varphi$ be a rule in P. Then $w \in L_{G'}\big((x,q,\varphi,q',y)\big)$ if and only if $x\lambda(q,w)y \in L_G(\varphi)$ and $\delta(q,w) = q'$ and the first letter of w has non-empty image. If in this case $w \in L_{G'}\big((x,q,A,q',y)\big)$, then it can be generated only by a rule in P'_{long}.*

Proof. 1. (a) Let $w \in L_{G'}\big((x',q,C,q',y)\big)$. Then $x = x''x'$, with $x'' \in L_G(B)$. Now $|x'| < |x|$ and thus $|x'\lambda(q,w)y| < k$. Then, by assumption, $x'\lambda(q,w)y \in L_G(C)$. So $z_1 = x''$ and $z_2 = x'\lambda(q,w)y$ is a correct factorization. In addition, $\delta(q,w) = q'$ by Lemma 2.

Conversely, if there is such a factorization, then $z_1 \in L_G(B)$, $x = z_1x'$ and $x'\lambda(q,w)y = z_2 \in L_G(C)$. Now $|x'\lambda(q,w)y| < k$ and by the assumption $w \in L_{G'}\big((x',q,C,q',y)\big)$.

(b) If $w \in L_{G'}\big((x,q,B,q'',\varepsilon)(\varepsilon,q'',C,q',y)\big)$, then it can be factorized as $w = uv$, with $u \in L_{G'}\big((x,q,B,q'',\varepsilon)\big)$ and $v \in L_{G'}\big((\varepsilon,q'',C,q',y)\big)$. By Lemma 4, this is only possible if $x\lambda(q,u) \neq \varepsilon$ and $\lambda(q'',v)y \neq \varepsilon$. Then $|x\lambda(q,u)|, |\lambda(q'',v)y| < k$, so, by the assumption, $z_1 = x\lambda(q,u) \in L_G(B)$ and $z_2 = \lambda(q'',v)y \in L_G(C)$. By Lemma 2 twice, $\delta(q,u) = q''$ and $\delta(q'',v) = q'$, which implies $\delta(q,w) = q'$. Then $\lambda(q,w) = \lambda(q,u)\lambda(\delta(q,u),v) = \lambda(q,u)\lambda(q'',v)$, and therefore $z_1z_2 = x\lambda(q,w)y$ is the requested factorization.

In the case $w \in L_{G'}\big((\varepsilon,q,C,q',y)\big)$, we have the factorization $z_1 = x = x\lambda(q,\varepsilon)$ and $z_2 = \lambda(q,w)y$.

Conversely let $z_1 = x\lambda(q,u)$ for some prefix u of w. Now if it is the case that $q \in Q_\varepsilon$ and $z_1 = x$, then w is generated by the rule $(x,q,B,C,q',y) \to (\varepsilon,q,C,q',y)\&\neg\varepsilon$. Otherwise let $q'' = \delta(q,u)$, we can assume $q'' \in Q_{\neg\varepsilon}$. Then $z_2 = \lambda(q'',v)y$, where $w = uv$. Now the assumption applies, and $u \in L_{G'}\big((x,q,B,q'',\varepsilon)\big)$ and $v \in L_{G'}\big((\varepsilon,q'',C,q',y)\big)$. Hence, $w \in L_{G'}\big((x,q,B,q'',\varepsilon)(\varepsilon,q'',C,q',y)\big)$.

(c) If $w \in L_{G'}\big((x,q,B,q'',y')T_{q'',q'''}a(x',\delta(q''',a),C,q',y)\big)$, then $w = u'tav$, where $u' \in L_{G'}\big((x,q,B,q'',y')\big)$, $t \in L_{G'}(T_{q'',q'''})$ and $v \in L_{G'}\big((x,\delta(q''',a),C,q',y)\big)$. We have $x\lambda(q,u')y' \in L_G(B)$ and $x'\lambda(\delta(q''',a),v)y \in L_G(C)$ by the assumption, and $\lambda(q'',t) = \varepsilon$ and $\lambda(q''',a) = y'x'$ by definition. By Lemma 2, $q'' = \delta(q,u)$, and therefore $q''' = \delta(q'',t) = \delta(q,u't)$ by the definition of $T_{q'',q'''}$. Combining these we obtain $z_1 = x\lambda(q,u't)y'$, $z_2 = x'\lambda(\delta(q''',a),v)y'$ and $|x\lambda(q,u't)| < |z_1| < |x\lambda(q,u'ta)|$. Setting $u = u't$, this satisfies the condition.

In the case $w \in L_{G'}\big(T_{q,q'''}a(x',\delta(q''',a),C,q',y)\big)$ we get $w = tav$, with $t \in L_{G'}(T_{q,q'''})$ and $v \in L_{G'}\big((x,\delta(q''',a),C,q',y)\big)$. We obtain $z_1 = x\lambda(q,t)y'$, $z_2 = x'\lambda(\delta(q''',a),v)y'$ and $|x\lambda(q,t)| < |z_1| < |x\lambda(q,ta)|$. Then $u = t$ and the condition of the lemma is met.

Conversely, let $|x\lambda(q,u)| < |z_1| < |x\lambda(q,ua)|$, for some $u \in \Sigma^*$ and $a \in \Sigma$, such that $w = uav$.

If it is the case that $q \in Q_\varepsilon$ and $\lambda(q,u) = \varepsilon$, then $u \in L_{G'}(T_{q,q'''})$, where $q''' = \delta(q,u)$, by definition. Now $z_1 = xy'$ for some proper prefix y' of $\lambda(q''',a) = y'x'$ and $z_2 = x'\lambda(\delta(q''',a),v)y$. By the assumption, $v \in L_{G'}\big((x,\delta(q''',a),C,q',y)\big)$, and so w is generated by the rule $(x,q,B,C,q',y) \to T_{q,q'''}a(x',\delta(q''',a),C,q',y)\&\neg\varepsilon$.

Otherwise let u' be the longest prefix of u, such that $\delta(q,u') \in Q_{\neg\varepsilon}$. Now $u = u't$, with $\lambda(\delta(q,u'),t) = \varepsilon$ and $w = u'tav$. Then $z_1 = x\lambda(q,u')y'$ and $z_2 = x'\lambda(\delta(q,uta),v)y$, where $\lambda(\delta(q,ut),a) = y'x'$ is the image of a which is split between z_1 and z_2. Now, by the assumption, $u' \in L_{G'}\big((x,q,B,\delta(q,u),y')\big)$ and $v \in L_{G'}\big((x',\delta(q,uta),C,\delta(q,w),y)\big)$, and by definition $t \in L_{G'}(T_{\delta(q,u),\delta(q,ut)})$.

(d) Now $|z_2| < |y|$ and we have a symmetric situation to the first case of (3a).

2. Let $w \in L_{G'}\big((x,q,B,C,q',y)\big)$. Then it is generated by a rule $(x,q,B,C,q',y) \to \alpha$, where α is one of the conjuncts (3a)-(3d). In all of these cases we get, by the first part, a factorization $x\lambda(q,w)y = z_1z_2 \in L_G(B)L_G(C)$ and $\delta(q,w) = q'$ by Lemma 2. If conversely $x\lambda(q,w)y \in L_G(BC)$, then it has a factorization $x\lambda(q,w)y = z_1z_2 \in L_G(B)L_G(C)$, and it is of one of the types in the first part, and thus w is generated by one of the rules.

3. The fact that the first symbol of w has a non-empty image is equivalent to $w \in \widetilde{T}_q$, and since (x,q,φ,q',y) contains a conjunction with $\widetilde{T}_q$, this can be assumed. By definition, $x\lambda(q,w)y \in L_G(\varphi)$ is equivalent to $x\lambda(q,w)y \in L_G(B_iC_i)$ for all i and $x\lambda(q,w)y \notin L_G(D_jE_j)$ for all j. By the second part of the Lemma, this is again equivalent to $w \in L_{G'}\big((x,q,B_i,C_i,q',y)\big)$ for all i and $w \notin L_{G'}\big((x,q,D_j,E_j,q',y)\big)$ for all j, which, with the assumption $w \in \widetilde{T}_q$, is equivalent to $w \in L_{G'}\big((x,q,\varphi,q',y)\big)$.

   Suppose then that $w \in L_{G'}\big((x,q,A,q',y)\big)$. First $w \neq \varepsilon$, so it can't be generated by a rule in P'_ε. It can't be generated by a rule from P'_0, since it begins with a symbol that has a non-empty image. And it can't be generated by a rule in P'_1, because otherwise $x = y = \varepsilon$, $w = a \in \Sigma$ and $|x\lambda(q,w)y| = 1$ against the assumption. Thus w is generated by a rule in P'_{long}. □

Now we are ready to prove the statement on the correspondence between words over Σ^* generated by nonterminals in N'_1 and word over Γ^* generated by the original grammar.

Lemma 9. *Let $x \in \text{suff}$, $y \in \text{pref}$, $w \in \Sigma^*$, $q \in Q$ and $q' \in Q_{\neg\varepsilon}$. Then $w \in L_{G'}\big((x, q, A, q', y)\big)$ if and only if $x\lambda(q, w)y \in L_G(A)$ and $\delta(q, w) = q'$.*

Proof. Induction on lexicographically ordered pairs $(|x\lambda(q, w)y|, |w|)$.

Basis: If $|x\lambda(q, w)y| \leqslant 1$, then the statement holds by Lemma 7 and if $w = \varepsilon$, then the statement holds by Lemma 3.

Induction step: If the first symbol of $w \neq \varepsilon$ has an empty image, then $w = aw'$, with $\lambda(q, a) = \varepsilon$ and $\lambda(q, w) = \lambda(\delta(q, a), w')$. By Lemma 5, $aw' \in L_{G'}\big((x, q, A, q', y)\big)$ is equivalent to $w' \in L_{G'}\big((x, \delta(q, a), A, q', y)\big)$. Now, by the induction hypothesis, $w' \in L_{G'}\big((x, \delta(q, a), A, q', y)\big)$ if and only if $x\lambda(\delta(q, a), w')y \in L_G(A)$ and $\delta(\delta(q, a), w') = q'$. The statement holds, since $x\lambda(q, w)y = x\lambda(\delta(q, a), w')y$.

In the remaining case $w \neq \varepsilon$, $|x\lambda(q, w)y| > 1$ and the first symbol of w has a non-empty image. Since $|x\lambda(q, w)y| > 1$, the statement $x\lambda(q, w)y \in L_G(A)$ is equivalent to the existence of a long rule $A \to \varphi$ in P, such that $x\lambda(q, w)y \in L_G(\varphi)$. By the induction hypothesis and by Lemma 8, this is equivalent to $w \in L_{G'}\big((x, q, \varphi, \delta(q, w), y)\big)$ for some long rule $A \to \varphi \in P$.

If the latter holds, then $w \in L_{G'}\big((x, q, A, q', y)\big)$, which proves the lemma in one direction. Conversely, assume $w \in L_{G'}\big((x, q, A, q', y)\big)$ and consider the possible rule by which it can be generated. It cannot be a rule from P'_ε because $w \neq \varepsilon$. No rule from P'_1 is applicable since they require $|x\lambda(q, w)y| = 1$, and neither are the rules from P'_0 which generate words with empty image of the first symbol. Therefore, w is generated by a rule in P'_{long}. Let this be a rule $(x, q, A, q', y) \to (x, q, \varphi, q', y)$, where $A \to \varphi \in P$. By Lemma 2, $q' = \delta(q, w)$, and hence $w \in L_{G'}\big((x, q, \varphi, \delta(q, w), y)\big)$, which completes the proof. □

Finally we can complete the proof for the correctness of the constructed grammar.

Lemma 10. $L_{G'}(S') = \{w \in \Sigma^* \mid M(w) \in L_G(S)\}$.

Proof. Assume $M(w) = \lambda(q^0_{\neg\varepsilon}, w) \in L_G(S)$. Then $\delta(q^0_{\neg\varepsilon}, w) \in F$. Let $w = w't$, where w' is the longest prefix of w with $\delta(q^0_{\neg\varepsilon}, w') = q' \in Q_{\neg\varepsilon}$; such a w' exists since $\delta(q^0_{\neg\varepsilon}, \varepsilon) \in Q_{\neg\varepsilon}$. Now $\lambda(q^0_{\neg\varepsilon}, w') = \lambda(q^0_{\neg\varepsilon}, w)$ and applying Lemma 9 to this, one obtains $w' \in L_{G'}\big((\varepsilon, q^0_{\neg\varepsilon}, S, q', \varepsilon)\big)$. At the same time, $\lambda(q', t) = \varepsilon$, and so $t \in L_{G'}(T_{q',\delta(q',t)})$. Then $w \in L_{G'}(S')$ by a rule $S' \to (\varepsilon, q^0_{\neg\varepsilon}, S, q', \varepsilon)T_{q',\delta(q',t)}$.

Conversely, let $w \in L_{G'}(S')$. Now w is generated by a rule $S' \to (\varepsilon, q^0_{\neg\varepsilon}, S, q, \varepsilon)T_{q,f}$, that is, $w = w't$ for some $w' \in L_{G'}\big((\varepsilon, q^0_{\neg\varepsilon}, S, q, \varepsilon)\big)$ and $t \in L_{G'}(T_{q,f})$. Then, by the definition of $T_{q,f}$, $\lambda(q, t) = \varepsilon$ and hence $M(w) = \lambda(q^0_{\neg\varepsilon}, w')$. Finally, $\lambda(q^0_{\neg\varepsilon}, w') \in L_G(S)$ by Lemma 9, which shows that $M(w) \in L_G(S)$. □

This proves Theorem 1 for Boolean grammars of the general form, as well as for conjunctive grammars.

3.4 Unambiguousness

The next step is to show that in fact the given construction preserves unambiguity. So through this section we assume that the grammar G is unambiguous and will prove that also G' is unambiguous. Let us first prove that the factorizations are unique.

Lemma 11. *Let factorizations of words produced by conjuncts in G be unique. Then the factorizations of words produced by conjuncts in G' are unique as well.*

Proof. The only conjuncts that have multiple nonterminals are $(\varepsilon, q^0_{\neg\varepsilon}, S, q, \varepsilon)T_{q,f}$ and those in (3b) and (3c). If $w \in L_{G'}\big((\varepsilon, q^0_{\neg\varepsilon}, S, q, \varepsilon)T_{q,f}\big)$, then $w = w't$ with $\delta(f, w') = q \in Q_{\neg\varepsilon}$ and $\lambda(\delta(q, w'a), t) = \varepsilon$ is the unique factorization. Any different factorization would have either t with a nonempty image (which would contradict the definition of $T_{q,f}$) or w' with a last symbol that has an empty image (hence q would be in Q_ε, which is not possible by the definition of N'_1).

If $w \in L_{G'}\big((x, q, B, q'', \varepsilon)(\varepsilon, q'', C, q', y)\big)$ would yield two factorizations $w = u_1v_1 = u_2v_2$, then by Lemma 9 $x\lambda(q, u_1), x\lambda(q, u_2) \in L_G(B)$ and $x\lambda(q, u_1) = x\lambda(q, u_2)$ by the unambiguity of G. We can assume by symmetry that u_1 is a prefix of u_2. If u_1 were a proper prefix of u_2, then $\lambda(q, u_1)$ would be a proper prefix of $\lambda(q, u_2)$, since $\delta(q, u_2) \in Q_{\neg\varepsilon}$. Thus $u_1 = u_2$.

Let $u_1t_1av_1$ and $u_2t_2av_2$ be two factorizations of $w \in L_{G'}\big((x, q, B, q'', y')T_{q'',q'''}a(x', \delta(q''', a), C, q', y)\big)$, with $u_1, u_2 \in L_{G'}\big((x, q, B, q'', y')\big)$, $t_1, t_2 \in L_{G'}(T_{q'',q'''})$ and $v_1, v_2 \in L_{G'}\big((x, \delta(q''', a), C, q', y)\big)$. By Lemma 9, $x\lambda(q, u_1)y', x\lambda(q, u_2)y' \in L_G(B)$ and $x'\lambda(\delta(q''', a), u_1)y, x'\lambda(\delta(q''', a), v_2)y \in L_G(C)$. Since G is unambiguous, $z_1 = x\lambda(q, u_1)y' = x\lambda(q, u_2)y'$. Again we can assume that u_1 is a prefix of u_2. We conclude that $u_1 = u_2$ by the same argument as above. In both of the factorizations a is the first symbol after u_1 and u_2 with a non-empty image, so $u_1t_1a = u_2t_2a$ and the factorizations are the same.

The case of $w \in L_{G'}\big(T_{q,q'''}a(x', \delta(q''', a), C, q', y)\big)$ is handled in the same way without u_i. □

Let us then prove that different rules generate disjoint languages. Let us start from the nonterminals in N'_1.

Lemma 12. *Let $w \in L_{G'}\big((x, q, A, q', y)\big)$. There is only one rule that generates w.*

Proof. If $w = \varepsilon$, then it is generated by a rule in P'_ε by Lemma 3. If w begins with a symbol that has an empty image, then it is generated by a rule in P'_0 by Lemma 5. If $w = a \in \Sigma$, $x = y = \varepsilon$ and the image $\lambda(q, a) \in \Gamma$, then it is generated by a rule in P'_1 by Lemma 6.

In the remaining case $|x\lambda(q, w)y| \geqslant 2$ and by Lemma 8(part 3) it is generated by a rule in P'_{long}, say $(x, q, A, q', y) \to (x, q, \varphi, q', y)$. If it was generated also by a rule $(x, q, A, q', y) \to (x, q, \psi, q', y)$ with $\varphi \neq \psi$, then, by Lemma 8(part 3), $x\lambda(q, w)y \in L_G(\varphi)$ and $x\lambda(q, w)y \in L_G(\psi)$, where $A \to \varphi$ and $A \to \psi$ are rules in P, contradicting the unambiguity of G. □

Lemma 13. *The rules for (x, q, B, C, q', y) generate disjoint languages.*

Proof. Let $w \in L_{G'}\big((x, q, B, C, q', y)\big)$. Then $x\lambda(q, w)y \in L_G(BC)$ by the second claim of Lemma 8, and it has a factorization $x\lambda(q, w)y = z_1z_2$ with $z_1 \in L_G(B)$ and $z_2 \in L_G(C)$. Since G is unambiguous, this factorization is unique.

Consider the cutting point of this factorization, which can be either inside of x or y, or inside the image of w; in the latter case it can be between images of symbols or in the middle of an image of a symbol. Lemma 8(part 1) lists these four cases, and each of them corresponds to a different type (3a)-(3d) of rules for (x, q, B, C, q', y). It remains to be proven that no two different rules of one type can generate w.

(3a) Suppose w is generated by a rule $(x, q, B, C, q', y) \to (x', q, C, q', y)\&\neg\varepsilon$ of the type (3a). Then $x = x''x'$ and $x'' \in L_G(B)$. Two different conjuncts of this type have different x'' and thus yield different factorizations.

(3b) Let w be generated by a rule $(x, q, B, C, q', y) \to (x, q, B, q'', \varepsilon)(\varepsilon, q'', C, q', y)\&\neg\varepsilon$ for some q'', that is, it can be factorized as $w = uv$, with $u \in L_{G'}\big((x, q, B, q'', \varepsilon)\big)$, $v \in L_{G'}\big((\varepsilon, q'', C, q', y)\big)$ and $q'' = \delta(q, u)$. If w can be generated by another rule of type (3b), $(x, q, B, C, q', y) \to (x, q, B, \tilde{q}'', \varepsilon)(\varepsilon, \tilde{q}'', C, q', y)\&\neg\varepsilon$, then at the same time $w = \tilde{u}\tilde{v}$, with $\tilde{u} \in L_{G'}\big((x, q, B, \tilde{q}'', \varepsilon)\big)$, $\tilde{v} \in L_{G'}\big((\varepsilon, \tilde{q}'', C, q', y)\big)$ and $\tilde{q}'' = \delta(q, \tilde{u})$.

If $u = \tilde{u}$, then $q'' = \tilde{q}''$ and the rule is actually the same. Assume u is shorter than $\tilde{u}$. Since $\tilde{q}'' \in Q_{\neg\varepsilon}$ by the definition of N_1'. Then the last symbol of $\tilde{u}$ has a non-empty image, and therefore $x\lambda(q, u)$ is shorter than $x\lambda(q, \tilde{u})$. So two different factorizations of $x\lambda(q, w)y$ into $L_G(B) \cdot L_G(C)$ are obtained, which cannot be the case because G is unambiguous.

Suppose w is generated by the rule $(x, q, B, C, q', y) \to (\varepsilon, q, C, q', y)\&\neg\varepsilon$ from (3b′), which is only possible if $q \in Q_\varepsilon$ and $x \in L_G(B)$. Then $\lambda(q, w)y \in L_G(C)$ by Lemma 9. Suppose some rule of the form $(x, q, B, C, q', y) \to (x, q, B, q'', \varepsilon)(\varepsilon, q'', C, q', y)\&\neg\varepsilon$ generates w as well. Then $w = uv$, with $u \in L_{G'}\big((x, q, B, q'', \varepsilon)\big)$ and $v \in L_{G'}\big((\varepsilon, q'', C, q', y)\big)$. By Lemma 9 twice, $x\lambda(q, u) \in L_G(B)$ and $\lambda(q'', v)y \in L_G(C)$. The image of u, $\lambda(q, u)$, is non-empty because $q \in Q_\varepsilon$ and $\delta(q, u) = q'' \in Q_{\neg\varepsilon}$, and thus $|x| < |x\lambda(q, u)|$. Now $x \cdot \lambda(q, w)y$ and $x\lambda(q, u) \cdot \lambda(q'', v)y$ are two different factorizations of the same word into $L_G(B) \cdot L_G(C)$, contradicting the assumption that G is unambiguous.

(3c) Suppose w is generated by two distinct rules of the form (3c), namely by $(x, q, B, C, q', y) \to (x, q, B, q'', y')T_{q'',q'''}a(x', \delta(q''', a), C, q', y)\neg\varepsilon$ and by $(x, q, B, C, q', y) \to (x, q, B, \tilde{q}'', \tilde{y}')T_{\tilde{q}'',\tilde{q}'''}\tilde{a}(\tilde{x}', \delta(\tilde{q}''', \tilde{a}), C, q', y)\neg\varepsilon$ Then, on one hand, $w = utav$, with $u \in L_{G'}\big((x, q, B, q'', y')\big)$, $t \in L_{G'}(T_{q'',q'''})$ and $v \in L_{G'}\big((x', \delta(q''', a), C, q', y)\big)$, and on the other hand, $w = \tilde{u}\tilde{t}\tilde{a}\tilde{v}$, where $\tilde{u} \in L_{G'}\big((x, q, B, q'', \tilde{y}')\big)$, $\tilde{t} \in L_{G'}(T_{\tilde{q}'',\tilde{q}'''})$ and $\tilde{v} \in L_{G'}\big((\tilde{x}', \delta(\tilde{q}''', \tilde{a}), C, q', y)\big)$.

Now $\tilde{u} \in L_{G'}\big((x, q, B, \tilde{q}'', \tilde{y}')\big)$ and, by Lemma 9, $x\lambda(q, u)y', x\lambda(q, \tilde{u})\tilde{y}' \in L_G(B)$, and thus $z_1 = x\lambda(q, u)y' = x\lambda(q, \tilde{u})\tilde{y}'$. If $u \neq \tilde{u}$, then one of them is a proper prefix of the other, say $|u| < |\tilde{u}|$. In this case $|uta| \leqslant |\tilde{u}|$, since a is the first symbol after u with a non-empty image and $\delta(q, \tilde{u}) \in Q_{\neg\varepsilon}$, and hence $|x\lambda(q, uta)| \leqslant |x\lambda(q, \tilde{u})\tilde{y}'|$. At the same time, $|x\lambda(q, u)y'| < |x\lambda(q, uta)|$, because y' is a proper prefix of the image of a. Combining these, $x\lambda(q, u)y'$ is strictly shorter than $x\lambda(q, \tilde{u})\tilde{y}'$, which is a contradiction. It follows that $u = \tilde{u}$. Furthermore, since a and $\tilde{a}$ are the first symbols after u and $\tilde{u}$ with non-empty images, also $t = \tilde{t}$, $a = \tilde{a}$ and $v = \tilde{v}$. Then the two rules are the same.

Consider the possibility of w being generated at the same time by a rule from (3c′) and by another rule from (3c). Then $q \in Q_\varepsilon$ by the construction of (3c′). Let $(x, q, B, C, q', y) \to T_{q,q'''}a(x', \delta(q''', a), C, q', y)\neg\varepsilon$ and $(x, q, B, C, q', y) \to (x, q, B, \tilde{q}'', \tilde{y}')T_{\tilde{q}'',\tilde{q}'''}\tilde{a}(\tilde{x}', \delta(\tilde{q}''', \tilde{a}), C, q', y)\neg\varepsilon$ be two such rules, that us, $w = tav = \tilde{u}\tilde{t}\tilde{a}\tilde{v}$ with $tav \in L_{G'}\big(T_{q,q'''}a(x', \delta(q''', a), C, q', y)\big)$ and $\tilde{u}\tilde{t}\tilde{a}\tilde{v} \in L_{G'}\big((x, q, B, \tilde{q}'', \tilde{y}')T_{\tilde{q}'',\tilde{q}'''}\tilde{a}(\tilde{x}', \delta(\tilde{q}''', \tilde{a}), C, q', y)\big)$. Since $q \in Q_\varepsilon$ and $\delta(q, \tilde{u}) = \tilde{q}'' \in Q_{\neg\varepsilon}$, $\tilde{u}$ is non-empty and ends with a symbol with a non-empty image. The first such symbol in the factorization tav is a, and hence ta is a prefix of $\tilde{u}$. From this it follows that $|xy'| < |x\lambda(q, \tilde{u})\tilde{y}'|$. These are the first components of the factorizations of $x\lambda(q, w)y$ into $L_G(B) \cdot L_G(C)$ corresponding to these two rules, and their distinctness contradicts the unambiguity of G.

The remaining case is when w is generated by two rules of the form (3c′), $(x, q, B, C, q', y) \to T_{q,q'''}a(x', \delta(q''', a), C, q', y)\neg\varepsilon$ and $(x, q, B, C, q', y) \to T_{q,\tilde{q}'''}\tilde{a}(\tilde{x}', \delta(\tilde{q}''', \tilde{a}), C, q', y)\neg\varepsilon$. Then $w = tav = \tilde{t}\tilde{a}\tilde{v}$, where both a is known to be the first symbol of w with a non-empty image, and the same is known with respect to $\tilde{a}$. Therefore, $t = \tilde{t}$, which implies $q''' = \tilde{q}'''$ and $a = \tilde{a}$, and hence the two rules must be the same.

(3d) Symmetric to the first case. □

Lemma 14. *If G is unambiguous, then so is G'.*

Proof. In addition to the above lemmata, it remains to consider the start symbol S' and two rules $S' \to (\varepsilon, q^0_{\neg\varepsilon}, S, q, \varepsilon)T_{q,f}$ and $S' \to (\varepsilon, \widetilde{q^0_{\neg\varepsilon}}, S, \tilde{q}, \varepsilon)T_{\tilde{q},\tilde{f}}$

generating w. There are two corresponding factorizations $w = w't = \widetilde{w'}\tilde{t}$. In both of these the states $\delta(q, w') = q$ and $\delta(q, \widetilde{w'}) = \tilde{q}$ are the last ones in $Q_{\neg\varepsilon}$, so $q = \tilde{q}$ and the rules are the same. And the proof of unambiguity of G' is finished. □

This completes the proof of Theorem 1.

4 Conclusion

All known closure properties of Boolean grammars and their subfamilies are given in Table 1. The bottom right corner of the table has been established in this paper. The closure properties of the unambiguous families remain to be studied. In addition, it remains unknown whether conjunctive languages are closed under complementation.

	$\cup$	$\cap$	$\sim$	$\cdot$	$*$	R	h	$h_{\varepsilon\text{-free}}$	h^{-1}	gsm^{-1}
Reg	+	+	+	+	+	+	+	+	+	+
UnambCF	− [4]	−	− [6]	− [4]	?	+	+ [4]	+ [4]	+ [4]	+ [4]
LinCF	+	−	−	−	−	+	+	+	+	+ [5]
CF	+	−	−	+	+	+	+	+	+	+ [3]
LinConj	+	+	+ [10]	−	− [10]	+	−	−	+ [2]	+ [7]
UnambConj	?	+	?	?	?	+	−	?	+	+
UnambBool	+	+	+	?	?	+	−	?	+	+
Conj	+	+	?	+	+	+	−	?	+	+
Bool	+	+	+	+	+	+	−	?	+	+

Table 1: Closure properties of Boolean grammars, compared to other classes.

Acknowledgements

Research supported by the Academy of Finland under grant 118540.

References

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

[2] K. Culik II, J. Gruska, A. Salomaa, "Systolic trellis automata: stability, decidability and complexity", *Information and Control*, 71 (1986) 218–230.

[3] S. Ginsburg, G. Rose, , “Operations which preserve definability in languages”, *Journal of the ACM*, 10:2 (1963), 175–195.

[4] S. Ginsburg, J. Ullian, “Preservation of unambiguity and inherent ambiguity in context-free languages”, *Journal of the ACM*, 13:3 (1966), 364–368.

[5] M. A. Harrison, *Introduction to formal language theory*, Addison-Wesley, 1978.

[6] T. N. Hibbard, J. Ullian, “The independence of inherent ambiguity from complementedness among context-free languages”, *Journal of the ACM*, 13:4 (1966), 588–593.

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

[8] V. Kountouriotis, Ch. Nomikos, P. Rondogiannis, “Well-founded semantics for Boolean grammars”, *Developments in Language Theory* (DLT 2006, Santa Barbara, USA, June 26–29, 2006), LNCS 4036, 203–214.

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

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

[11] A. Okhotin, “Boolean grammars”, *Information and Computation*, 194:1 (2004), 19–48.

[12] A. Okhotin, “Nine open problems for conjunctive and Boolean grammars”, *Bulletin of the EATCS*, 91 (2007), 96–119.

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

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 978–952–12–2147–7
ISSN 1239-1891