TUCS

Tommi Lehtinen | Alexander Okhotin

Boolean grammars are closed under inverse homomorphisms

Turku Centre for Computer Science

TUCS Technical Report
No 846, October 2007

Boolean grammars are closed under inverse homomorphisms

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 846, October 2007

Abstract

It is proved that for every Boolean grammar G and for every homomorphism h, the set $h^{-1}(L(G))$ 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, homomorphism

TUCS Laboratory
Discrete Mathematics for Information Technology

1 Introduction

Boolean grammars [6] 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 [6, 7].

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, $\mathit{DTIME}(n^3) \cap \mathit{DSPACE}(n)$ [6].

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, 5]. The closure under non-erasing homomorphisms remains an open problem.

This paper considers *inverse homomorphism*, and it is established that the family of languages generated by Boolean grammars is closed under this operation. It is proved that for every Boolean grammar G over an alphabet Γ and for every homomorphism $h : \Sigma^* \to \Gamma^*$, the set $h^{-1}(L(G)) = \{w \in \Sigma^* \mid h(w) \in L(G)\}$ of pre-images of words generated by G is generated by a Boolean grammar. An effective construction of this grammar is given in two steps: first, it is done for the case of non-erasing homomorphism; then, for a projection, that is, for a homomorphism that maps every letter to itself or to ε. Since every homomorphism is a composition of such mappings, this construction applies to every given homomorphism. Furthermore, if the grammar G is unambiguous [8], then the constructed grammar for $h^{-1}(L(G))$ is unambiguous as well, that is, the family of unambiguous Boolean grammars is also closed under this operation.

2 Definition of Boolean grammars

Definition 1 ([6])**.** *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* [4], 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, 5].

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 $\{a^n b^n c^n \,|\, n \geqslant 0\}$:*

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

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 [3, 6], 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*

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

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 [6] and no better upper bound is known, unambiguous Boolean grammars allow square-time parsing [8]. 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 [8]. The rest of the classes in the figure are regular (*Reg*), linear context-free (*LinCF*), context-free (*CF*) and deterministic context-sensitive languages (*DetCS*).

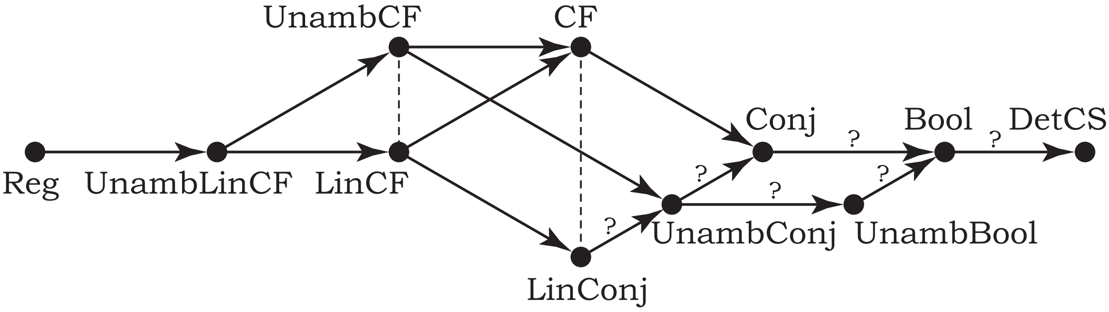

Figure 1: The hierarchy of language families.

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*

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

$$A \to a$$

$S \to \varepsilon$ (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 [3, 6].

Proposition 1 ([6, 8])**.** *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.*

3 Closure under inverse homomorphisms

Let Σ and Γ be finite alphabets and $h : \Sigma^* \to \Gamma^*$ a homomorphism. We shall prove:

Theorem 1. *For every Boolean grammar (conjunctive grammar) $G = (\Gamma, N, P, S)$ there exists a Boolean grammar (conjunctive grammar, respectively) $G' = (\Sigma, N', P', S')$ with $L(G') = h^{-1}(L(G))$. Furthermore if G is unambiguous, then G' is unambiguous as well.*

We will prove the statement in two steps using the representation $h = h_1 \circ h_0$, where h_1 is non-erasing and h_0 is a projection. First we divide the alphabet as $\Sigma = \Sigma_0 \cup \Sigma_1$, where $\Sigma_0 = \{a \in \Sigma \mid h(a) = \varepsilon\}$ and $\Sigma_1 = \{a \in \Sigma \mid h(a) \neq \varepsilon\}$. Then define $h_0 : \Sigma^* \to \Sigma_1^*$, where $h_0(a_0) = \varepsilon$ for all $a_0 \in \Sigma_0$ and $h_0(a) = a$ for all $a \in \Sigma_1$, and $h_1 : \Sigma_1^* \to \Gamma^*$, where $h_1(a) = h(a)$ for all $a \in \Sigma_1$. Now $h = h_1 \circ h_0$ is the requested representation.

Since $h^{-1}(K) = h_0^{-1}(h_1^{-1}(K))$, we can prove Theorem 1 by proving the statement separately for non-erasing homomorphisms and projections. We shall do this in Sections 3.1 and 3.2, respectively.

All constructions are done for Boolean grammars. However, it can be observed that if the original grammar has no negative conjuncts, then the resulting grammar has no negative conjuncts besides those of the form $X \to \neg\varepsilon$. The latter can be eliminated by expressing the language Σ^+. Thus the results apply to conjunctive grammars as well.

3.1 Non-erasing homomorphisms

Let $G = (\Gamma, N, P, S)$ be a Boolean grammar in the binary normal form and h a non-erasing homomorphism. We construct a grammar $G' = (\Sigma, N', P', S')$ for the language $h^{-1}(L(G))$ as follows.

First we define few notions we shall use in formulating G'. Define sets

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

For all $B, C \in N$, $x \in \mathrm{suff}(h(\Sigma))$ and $y \in \mathrm{pref}(h(\Sigma))$, let $\Phi(x, B, C, y)$ be the union of the following four sets:

$$\{(x', C, y) \mid x = x''x';\ x', x'' \in \Sigma^+;\ x'' \in L(B)\}, \tag{3a}$$

$$\{(x, B, \varepsilon)(\varepsilon, C, y)\}, \tag{3b}$$

$$\{(x, B, y')a(x', C, y) \mid a \in \Sigma;\ h(a) = y'x';\ x', y' \in \Sigma^+\}, \tag{3c}$$

$$\{(x, B, y') \mid y = y'y'';\ y', y'' \in \Sigma^+;\ y'' \in L(C)\}. \tag{3d}$$

The sets correspond to all possible types of factorizations of $xh(w)y$ as in the figure.

Now we are ready to construct the grammar G'.

As nonterminals we have the set

$$N' = \{(x, A, y) \mid A \in N,\ x \in \mathrm{suff}(h(\Sigma)),\ y \in \mathrm{pref}(h(\Sigma))\}.$$

For every rule $A \to B_1C_1\& \ldots \& B_mC_m \& \neg D_1E_1 \& \ldots \& \neg D_nE_n \& \neg\varepsilon$, in P and for every $x \in \mathrm{suff}(h(\Sigma))$ and $y \in \mathrm{pref}(h(\Sigma))$, define a corresponding set of rules $P'_{x,A\to B_1C_1\&\ldots\&B_mC_m\&\neg D_1E_1\&\ldots\&\neg D_nE_n\&\neg\varepsilon,y}$. For all $\alpha_i \in \Phi(x, B_i, C_i, y)$, $1 \leqslant i \leqslant m$, this set contains a rule

$$(x, A, y) \to \alpha_1 \& \ldots \& \alpha_n \& \Big(\mathop{\&}_{\beta \in \Phi(x, D_1, E_1, y)} \neg\beta \Big) \& \ldots \& \Big(\mathop{\&}_{\beta \in \Phi(x, D_n, E_n, y)} \neg\beta \Big) \& \neg\varepsilon. \tag{4}$$

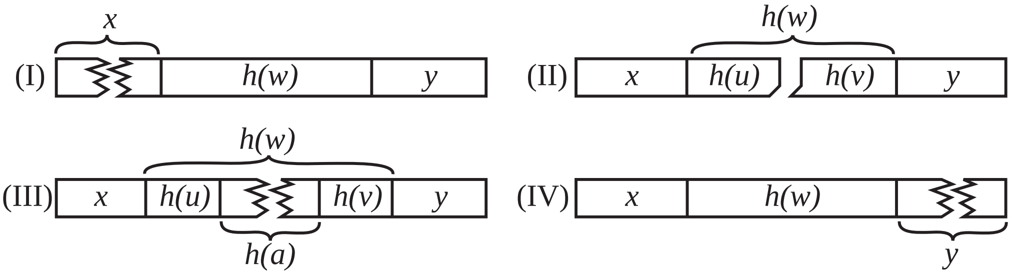

Figure 2: Factorizations of $xh(w)y$.

Next, for all $a \in \Sigma$, such that $h(a) \in \Gamma$, and P contains the rule $A \to h(a)$, define the corresponding set of new rules as $P'_{\varepsilon,A\to h(a),\varepsilon} = \big\{(\varepsilon, A, \varepsilon) \to a\big\}$.

Define a set P'_ε of additional rules generating ε as follows. If $x \in \text{suff}(h(\Sigma))$, $y \in \text{pref}(h(\Sigma))$ and $xy \in L_G(A)$, then P'_ε contains the rule $(x, A, y) \to \varepsilon$; note that in this case $|xy| > 0$, except maybe in the case $A = S$.

Finally, define the entire set of rules of G' as

$$P' = P'_\varepsilon \cup \bigcup_{A\to\varphi\in P} \bigcup_{\substack{x\in\text{suff}(h(\Sigma))\\ y\in\text{pref}(h(\Sigma))}} P'_{x,A\to\varphi,y}$$

Its start symbol is $S' = (\varepsilon, S, \varepsilon)$.

First we have to prove that the constructed grammar G' is compatible with the chosen semantics to be sure that it defines a language.

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

Proof. Let M be a finite subword-closed language, it has to be proved that the solution modulo M is unique. Induction on $|M|$.

Induction basis: $M_0 = \{\varepsilon\}$. The unique solution modulo M_0 is $(x, A, y) = \{\varepsilon\}$ if $xy \in L_G(A)$, $(x, A, y) = \varnothing$ otherwise.

Induction hypothesis: There is a unique solution modulo M'.

Induction step: Let $M = M' \cup \{w\}$, where $w \notin M'$, but all subwords of w are in M'. By the induction hypothesis, the solution modulo M' is unique.

Suppose it is not unique modulo M. Then there are solutions $(x, A, y) = L_{x,A,y}$ (L for the vector) and $(x, A, y) = L'_{x,A,y}$ (L' for the vector) that differ on w. Now there is (x, A, y), such that $w \in L_{x,A,y}$, but $w \notin L'_{x,A,y}$. Choose (x, A, y) with the least $|xy|$.

If $w \in L_{x,A,y}$, then, by the equation for (x, A, y), there exists a rule (4), such that $w \in L_{\alpha_i}$ for $1 \leqslant i \leqslant m$ and $w \notin L_{\beta_j}$ for $1 \leqslant j \leqslant n$, or, in case $|w| = 1$, $x = y = \varepsilon$ and $A \to h(w) \in P$, a rule $(\varepsilon, A, \varepsilon) \to w$. The latter case is impossible, because then w would be in $L'_{x,A,y}$ by the same rule. In the former case, since $w \notin L'_{x,A,y}$, it follows that $w \notin L'_{\alpha_i}$ for some $1 \leqslant i \leqslant m$

or $w \in L'_{\beta_j}$ for some $1 \leqslant j \leqslant n$, that is, $w \in \gamma(L)$ and $w \notin \gamma(L')$ for some $B, C \in N \setminus \{S\}$ and $\gamma \in \Phi(x, B, C, y)$. By the definition of Φ, there are four cases to consider:

1. If $\gamma = (x', C, y)$, then $|x'y| < |xy|$ and, by the choice of (x, A, y), $L_{x',C,y} = L'_{x',C,y}$.

2. Consider the case of $\gamma = (x, B, \varepsilon)(\varepsilon, C, y)$, that is, $w \in L_{x,B,\varepsilon}L_{\varepsilon,C,y}$ and $w \notin L'_{x,B,\varepsilon}L'_{\varepsilon,C,y}$. Then there exists a factorization $w = uv$, with $u \in L_{x,B,\varepsilon}$ and $v \in L_{\varepsilon,C,y}$.

   If $u = \varepsilon$, then $x \neq \varepsilon$ and accordingly $w \in L_{\varepsilon,C,y}$ and $w \notin L'_{\varepsilon,C,y}$. Since $|\varepsilon \cdot y| < |xy|$, the solution differ with respect to w on a variable with narrower margins, which contradicts the choice of (x, A, y). The case of $v = \varepsilon$ is proved symmetrically.

   Suppose $u, v \neq \varepsilon$, then $u, v \in M'$ and we have by the induction hypothesis that $w = uv \in L'_{x,B,\varepsilon}L'_{\varepsilon,C,y}$.

3. $w \in (x, B, y')a(x', C, y)(L)$. Then there is a factorization $w = uav$, with $u \in L_{x,B,y'}$ and $v \in L_{x',C,y}$. Since $w \notin (x, B, y')a(x', C, y)(L)$, it follows that $u \notin L'_{x,B,y'}$ or $v \notin L'_{x',C,y}$, and the solutions differ on a string in M', which again contradicts the assumption.

4. If $\gamma = (x, B, y')$, then $|xy'| < |xy|$ and, by the choice of (x, A, y), $L_{x,B,y'} = L'_{x,B,y'}$. □

Now $L'_G((x, A, y))$ is well-defined, and the statement of correctness of the construction can be formulated. We will first prove the following correspondence between conjuncts BC and the conjuncts in $\Phi(x, B, C, y)$, and between rules $A \to \varphi$ and the rules in $P'_{x,A\to\varphi,y}$.

Lemma 2. *Let $\ell \geqslant 2$ and assume that for all $w' \in \Sigma^*$, $x' \in \operatorname{suff}(h(\Sigma))$, $y' \in \operatorname{pref}(h(\Sigma))$ and $A \in N$, with $|x'h(w')y'| < \ell$, it holds that $x'h(w')y' \in L_G(A)$ if and only if $w' \in L_{G'}((x', A, y'))$.*

Then for all $w \in \Sigma^*$, $x \in \operatorname{suff}(h(\Sigma))$ and $y \in \operatorname{pref}(h(\Sigma))$, with $|xh(w)y| = k$,

I. For every $B, C \in N$, $xh(w)y \in L_G(BC)$ if and only if $w \in L_{G'}(\alpha)$ for some $\alpha \in \Phi(x, B, C, y)$.

II. If additionally $w \neq \varepsilon$, then for every rule $A \to \varphi$ in P, $xh(w)y \in L_G(\varphi)$ if and only if $w \in L_{G'}(\psi)$ for some $(x, A, y) \to \psi \in P'_{x,A\to\varphi,y}$.

Proof. Let us begin with the proof of the first claim.

$\ominus\!\!\!\Rightarrow$ Suppose $xh(w)y \in L_G(BC)$. Then there is a factorization $xh(w)y = z_1z_2$, with $z_1 \in L_G(B)$ and $z_2 \in L_G(C)$. Note that $z_1, z_2 \neq \varepsilon$. There are four cases to consider:

1. $z_1 = x'$, $z_2 = x''h(w)y$, $x = x'x''$
2. $z_1 = xh(u)$, $z_2 = h(v)y$, $w = uv$
3. $z_1 = xh(u)y'$, $z_2 = x'h(v)y$, $w = uav$, $h(a) = y'x'$
4. $z_1 = xh(w)y'$, $z_2 = y''$

In each case one can construct a suitable α:

1. By definition, $(x'', C, y) \in \Phi(x, B, C, y)$, which implies $w \in L_{G'}((x'', C, y))$ by assumption, since $|x''h(w)y| < \ell$.
2. By definition, $(x, B, \varepsilon)(\varepsilon, C, y) \in \Phi(x, B, C, y)$. By assumption, $xh(u) \in L_G(B)$ implies $u \in L_{G'}((x, B, \varepsilon))$, while $h(v)y \in L_G(C)$ implies and $v \in L_{G'}((\varepsilon, C, y))$, so that $w \in L_{G'}((x, B, \varepsilon)(\varepsilon, C, y))$.
3. By definition, $(x, B, y')a(x', C, y) \in \Phi(x, B, C, y)$. Using the assumption, $xh(u)y' \in L_G(B)$ implies $u \in L_{G'}((x, B, y'))$ and $x'h(v)y \in L_G(C)$ implies $v \in L_{G'}((x', C, y))$. Therefore, $w \in L_{G'}((x, B, y')a(x', C, y))$.
4. By definition, $(x, B, y') \in \Phi(x, B, C, y)$. Then, as in the first case, $w \in L_{G'}((x, B, y'))$.

$\textcircled{\Leftarrow}$ Conversely, let us prove that if $w \in L_{G'}(\alpha)$ for some $\alpha \in \Phi(x, B, C, y)$, then $xh(w)y \in L_G(BC)$:

1. If $\alpha = (x', C, y)$, then, by the definition of Φ, $x = x''x'$ and $x'' \in L_G(B)$. Since $w \in L_{G'}\big((x', C, y)\big)$ and $|x'h(w)y| < \ell$, by assumption, $x'h(w)y \in L_G(C)$. Therefore, $xh(w)y = x''x'h(w)y \in L_G(BC)$.
2. If $\alpha = (x, B, \varepsilon)(\varepsilon, C, y)$, then $w = uv$ for some $u \in L_{G'}\big((x, B, \varepsilon)\big)$ and $v \in L_{G'}\big((\varepsilon, C, y)\big)$. Since $|xh(u)|, |h(v)y| < \ell$, by the assumption, $xh(u) \in L_G(B)$ and $h(v)y \in L_G(C)$. Concatenating these statements, we obtain $xh(w)y = xh(u)h(v)y \in L_G(BC)$.
3. If $\alpha = (x, B, y')a(x', C, y)$, then $w = uav$ for some $u \in L_{G'}\big((x, B, y')\big)$ and $v \in L_{G'}\big((x', C, y)\big)$. Then, by the assumption, $xh(u)y' \in L_G(B)$ and $x'h(v)y \in L_G(C)$. Therefore, $xh(w)y = xh(u)h(a)h(v)y = xh(u)y'x'h(v)y \in L_G(BC)$.
4. If $\alpha = (x, B, y')$, then, by the definition of Φ, $y = y'y''$ and $y'' \in L_G(C)$. Then, as in the first case, $xh(w)y' \in L_G(B)$ by the assumption, so that $xh(w)y = xh(w)y'y'' \in L_G(BC)$.

This proves the first part of the lemma. Let us then prove the second statement.

Suppose $xh(w)y \in L_G(\varphi)$, where

$$\varphi = B_1C_1\& \ldots \&B_mC_m\&\neg D_1E_1\& \ldots \&\neg D_nE_n\&\neg\varepsilon.$$

Then $xh(w)y \in L_G(B_iC_i)$ for $1 \leqslant i \leqslant m$ and $xh(w)y \notin L_G(D_jE_j)$ for $1 \leqslant j \leqslant n$. By the first claim of this lemma, this is equivalent to the following statement: for every i there exists $\alpha_i \in \Phi(x, B_i, C_i, y)$, such that $w \in L_{G'}(\alpha_i)$, and for every j and for every $\beta_j \in \Phi(x, D_j, E_j, y)$, $w \notin L_{G'}(\beta_j)$. If this statement holds, then we have $\alpha_1, \ldots, \alpha_n$, such that

$$w \in L_{G'}\Big(\alpha_1 \& \ldots \& \alpha_m \& \big(\mathop{\&}_{\beta \in \Phi(x,D_1,E_1,y)} \neg\beta\big) \& \ldots \& \big(\mathop{\&}_{\beta \in \Phi(x,D_n,E_n,y)} \neg\beta\big)\Big), \quad (5)$$

And by definition,

$$A \to \alpha_1 \& \ldots \& \alpha_m \& \big(\mathop{\&}_{\beta \in \Phi(x,D_1,E_1,y)} \neg\beta\big) \& \ldots \& \big(\mathop{\&}_{\beta \in \Phi(x,D_n,E_n,y)} \neg\beta\big) \& \neg\varepsilon \in P'_{x,A\to\varphi,y}.$$

Conversely, let $w \in L_{G'}(\psi)$ for some $A \to \psi \in P'_{x,A\to\varphi,y}$. Now

$$\psi = \alpha_1 \& \ldots \& \alpha_n \& \big(\mathop{\&}_{\beta \in \Phi(x,D_1,E_1,y)} \neg\beta\big) \& \ldots \& \big(\mathop{\&}_{\beta \in \Phi(x,D_n,E_n,y)} \neg\beta\big) \& \neg\varepsilon,$$

where $\alpha_i \in \Phi(x, B_i, C_i, y)$ and

$$w \in L_{G'}(\alpha_1 \& \ldots \& \alpha_n \& \big(\mathop{\&}_{\beta \in \Phi(x,D_1,E_1,y)} \neg\beta\big) \& \ldots \& \big(\mathop{\&}_{\beta \in \Phi(x,D_n,E_n,y)} \neg\beta\big) \& \neg\varepsilon).$$

Now by construction

$$\varphi = B_1C_1 \& \ldots \& B_mC_m \& \neg D_1E_1 \& \ldots \& \neg D_nE_n \& \neg\varepsilon,$$

and by the first claim of this lemma, $xh(w)y \in L_G(B_iC_i)$ for $1 \leqslant i \leqslant m$ and $xh(w)y \notin L_G(D_jE_j)$ for $1 \leqslant j \leqslant n$. Therefore, $xh(w)y \in L_G(\varphi)$. □

And then we proceed to the actual proof for the correctness of the construction.

Lemma 3. *Let $w \in \Sigma^*$, $x \in \mathrm{suff}(h(\Sigma))$ and $y \in \mathrm{pref}(h(\Sigma))$. Then, for every $A \in N$, $xh(w)y \in L_G(A)$ if and only if $w \in L_{G'}((x, A, y))$.*

Proof. The proof is an induction on $|xh(w)y|$.

Induction basis: First consider the case $|xh(w)y| = 0$. By the construction of the grammar G', $\varepsilon \in L_{G'}((x, A, y))$ if and only if $xy \in L_G(A)$. Since $xy = xh(w)y = \varepsilon$ and $w = \varepsilon$, the claim is proved.

In the case $|xh(w)y| = 1$ there are two possibilities.

- If $|xy| = 1$ and thus $w = \varepsilon$, then, by the construction of G', $xh(w)y = xy \in L_G(A)$ if and only if there is a rule $(x, A, y) \to \varepsilon$ in P'.
- Suppose $|h(w)| = 1$ and thus $x = y = \varepsilon$ and $w = a$ for some $a \in \Sigma$. If $h(a) \in L_G(A)$, then there is a rule $(\varepsilon, A, \varepsilon) \to a$ in P', hence $a \in L_{G'}((\varepsilon, A, \varepsilon))$.

Conversely, if $a \in L_{G'}((\varepsilon, A, \varepsilon))$, then there is is rule for $(\varepsilon, A, \varepsilon)$ in P' which generates a. If this is a rule of the form $(\varepsilon, A, \varepsilon) \to a$, then there is a rule $A \to h(a)$ in P.

Let us show that no long rule of the form (4) for $(\varepsilon, A, \varepsilon)$ can generate a with $|h(a)| = 1$. If there were such a rule, there would be $a \in L_{G'}(\alpha)$ for some $\alpha \in \Phi(\varepsilon, B, C, \varepsilon)$, where BC is a positive conjunct in some rule for A in P. Note that $B, C \neq S$ because of the normal form. Consider the four possible cases for α:

1. Case $\alpha = (x', C, \varepsilon)$. This is impossible, since x' would have to be a proper suffix of ε.
2. Case $\alpha = (\varepsilon, B, \varepsilon)(\varepsilon, C, \varepsilon)$. There would have to be $\varepsilon \in L_{G'}((\varepsilon, B, \varepsilon))$ or $\varepsilon \in L_{G'}((\varepsilon, C, \varepsilon))$, which is impossible, since this would hold only for $B = S$ or $C = S$.
3. Case $\alpha = (\varepsilon, B, x')a(y', C, \varepsilon)$. In this case one of x' or y' would have to be a nonempty proper substring of $h(a)$, which is impossible by $|h(a)| = 1$.
4. Case $\alpha = (\varepsilon, B, y')$. This is impossible, since y' would have to be a proper suffix of ε.

Induction hypothesis: Suppose the claim holds for strings shorter than $xh(w)y$, where $|xh(w)y| \geqslant 2$.

Induction step: Let $xh(w)y \in L_G(A)$. Now if $w = \varepsilon$, then there is a rule $(x, A, y) \to \varepsilon$ in P'. If $w \neq \varepsilon$, then $xh(w)y \in L_G(\varphi)$ for some $A \to \varphi \in P$ and by Lemma 2(part II) there is a rule $(x, A, y) \to \psi \in P'_{x,A\to\varphi,y}$ for which $w \in L_{G'}(\psi)$. This means $w \in L_{G'}\big((x, A, y)\big)$.

Conversely, let $w \in L_{G'}\big((x, A, y)\big)$. If $w = \varepsilon$, then $xy \in L_G(A)$ by construction. If $w \neq \varepsilon$, then there is a rule $(x, A, y) \to \psi$ with $w \in L_{G'}(\psi)$. Now $(x, A, y) \to \psi \in P'_{x,A\to\varphi,y}$ for some $A \to \varphi \in P$ and by Lemma 2(part II) $xh(w)y \in L_G(\varphi)$. Hence $xh(w)y \in L_G(A)$. □

In particular, $w \in L_{G'}((\varepsilon, S, \varepsilon)) = L(G')$ if and only if $xh(w)y \in L_G(S) = L(G)$. In other words $L(G') = h^{-1}(L(G))$. We will then show that the construction gives an unambiguous grammar if the original grammar is unambiguous.

Lemma 4. *If G is unambiguous, then G' is unambiguous as well.*

Proof. Let us first prove that the factorizations in conjuncts of G' are unique. Consider each conjunct in each rule. Only conjuncts of the form (3b) and (3c) that come from φ have to be considered, since no other conjuncts in G' have multiple nonterminals.

(3b) Suppose $w \in L_{G'}\big((x, B, \varepsilon)\big)L_{G'}\big((\varepsilon, C, y)\big)$ admits multiple factorizations. Let $w_1, w_3 \in L_{G'}\big((x, B, \varepsilon)\big)$ and $w_2, w_4 \in L_{G'}\big((\varepsilon, C, y)\big)$, with

$w = w_1 w_2 = w_3 w_4$. Then, by Lemma 3 four times, $xh(w_1), xh(w_3) \in L_G(B)$ and $h(w_2)y, h(w_4)y \in L_G(C)$. Concatenating these strings, we obtain $xh(w_1)h(w_2)y, xh(w_3)h(w_4)y \in L_G(B)L_G(C)$. Since $xh(w)y = xh(w_1)h(w_2)y = xh(w_3)h(w_4)y$, these are two factorizations of the same string, and as a factorization of a string in $L_G(B) \cdot L_G(C)$ is unique by assumption, $xh(w_1)$ must be equal to $xh(w_3)$. Because one of w_1, w_3 is a prefix of the other and h is nonerasing, it follows that w_1 and w_3 are equal.

(3c) Suppose $w \in L_{G'}\big((x, B, y')\big) a L_{G'}\big((x', C, y)\big)$ admits multiple factorizations. Let $w_1, w_3 \in L_{G'}\big((x, B, y')\big)$ and $w_2, w_4 \in L_{G'}\big((x', C, y)\big)$, with $w = w_1 a w_2 = w_3 a w_4$. Then, by Lemma 3 four times, $xh(w_1)y', xh(w_3)y' \in L_G(B)$ and $x'h(w_2)y, x'h(w_4)y \in L_G(C)$. Concatenating these strings, we obtain $xh(w)y = xh(w_1)y'x'h(w_2)y = xh(w_3)y'x'h(w_4)y \in L_G(B)L_G(C)$. Again, the factorization of every string into $L_G(B) \cdot L_G(C)$ is unique, hence $xh(w_1)y'$ must be equal to $xh(w_3)y'$, and, as in the previous case, $w_1 = w_3$.

To prove that different rules for (x, A, y) generate disjoint languages, consider a word w in $L_{G'}\big((x, A, y)\big)$ and suppose there are two different rules of the form (4) that generate w. Since the original grammar is unambiguous, the corresponding word $xh(w)y \in L_G(A)$ is generated by a unique rule $A \to \varphi$ in G. By Lemma 2(part II), all rules in G' generating w are in $P'_{x,A\to\varphi,y}$. The negative conjuncts of all rules in this set are identical, so the two rules generating w differ on a pair of positive conjuncts $\alpha_1, \alpha_2 \in \Phi(x, B, C, y)$, where BC is one of the positive conjuncts in $A \to \varphi$.

Thus we have $w \in L_{G'}(\alpha_1)$ and $w \in L_{G'}(\alpha_2)$. There is a unique factorization of $xh(w)y$ into $z_1 z_2$, with $z_1 \in L_G(B)$ and $z_2 \in L_G(C)$.

We have 9 possible cases of distinct α_1 and α_2.

1. Case $\alpha_1 = (x'_1, C, y)$ and $\alpha_2 = (x'_2, C, y)$, with $x'_1 \neq x'_2$. Now the definition (3a) and Lemma 3 would give $z_1 = x \cdot (x'_1)^{-1} = x \cdot (x'_2)^{-1}$ and $z_2 = x'_1 h(w)y = x'_2 h(w)y$, which is a contradiction since $x \cdot (x'_1)^{-1} \neq x \cdot (x'_2)^{-1}$.

2. Case $\alpha_1 = (x'_1, C, y)$ and $\alpha_2 = (x, B, \varepsilon)(\varepsilon, C, y)$. Now the definitions (3a) and (3b), and Lemma 3 would give $z_1 = x \cdot (x'_1)^{-1} = xh(u)$ and $z_2 = x'_1 h(w)y = h(v)y$, where u and v are subwords of w generated by (x, B, ε) and (ε, C, y) such that $uv = w$. Since $|x \cdot (x'_1)^{-1}| < |xh(u)|$, this is a contradiction.

3. Case $\alpha_1 = (x'_1, C, y)$ and $\alpha_2 = (x, B, y'_2) a_2 (x'_2, C, y)$. Now the definitions (3a) and (3c), and Lemma 3 would give $z_1 = x \cdot (x'_1)^{-1} = xh(u_2)y'_2$ and $z_2 = x'_1 h(w)y = x'_2 h(v_2)y$, where u_2 and v_2 are subwords of w generated by (x, B, y'_2) and (x'_2, C, y) such that $u_2 a_2 v_2 = w$. This is a contradiction, because $|x \cdot (x'_1)^{-1}| < |xh(u_2)y'_2|$.

4. Case $\alpha_1 = (x'_1, C, y)$ and $\alpha_2 = (x, B, y'_2)$. Now the definitions (3a) and (3d), and Lemma 3 would give $z_1 = x \cdot (x'_1)^{-1} = xh(w)y'_2$ and $z_2 = x'_1h(w)y = (y'_2)^{-1} \cdot y$, which is a contradiction since $|x \cdot (x'_1)^{-1}| < |xh(w)y'_2|$.

5. Case $\alpha_1 = (x, B, \varepsilon)(\varepsilon, C, y)$ and $\alpha_2 = (x, B, y'_2)a_2(x'_2, C, y)$. Now the definitions (3b) and (3c), and Lemma 3 would give $z_1 = xh(u_1) = xh(u_2)y'_2$ and $z_2 = h(v_1)y = x'_2h(v_2)y$, where u_1 and v_1 are subwords of w generated by (x, B, ε) and (ε, C, y) such that $u_1v_1 = w$, and u_2 and v_2 are subwords of w generated by (x, B, y'_2) and (x'_2, C, y) such that $u_2a_2v_2 = w$. Now one of u_1 and u_2 is a prefix of the other. If $|u_1| \leqslant |u_2|$, then $|xh(u_1)| < |xh(u_2)y'_2|$ and if $|u_2| < |u_1|$, then $|u_2a_2| \leqslant |u_1|$ and thus $|xh(u_1)| > |xh(u_2)y'_2|$. We have a contradiction.

6. Case $\alpha_1 = (x, B, \varepsilon)(\varepsilon, C, y)$ and $\alpha_2 = (x, B, y'_2)$: symmetric to case 2.

7. Case $\alpha_1 = (x, B, y'_1)a_1(x'_1, C, y)$ and $\alpha_2 = (x, B, y'_2)a_2(x'_2, C, y)$, where $a_1 \neq a_2$, $x'_1 \neq x'_2$ or $y'_1 \neq y'_2$. Now the definition (3c), and Lemma 3 would give $z_1 = xh(u_1)y'_1 = xh(u_2)y'_2$ and $z_2 = x'_1h(v_1)y = x'_2h(v_2)y$, where, for $i \in \{1, 2\}$, u_i and v_i are subwords of w generated by (x, B, y'_i) and (x'_i, C, y), such that $u_ia_iv_i = w$. If $a_1 \neq a_2$, then one of u_1 and u_2 is a proper prefix of the other, say $|u_1| < |u_2|$, and thus $xh(u_1)y'_1 < xh(u_2)y'_2$, which forms a contradiction. On the other hand, if $a_1 = a_2$ and $u_1 = u_2$, then $|y'_1| \neq |y'_2|$ and $|xh(u_1)y'_1| \neq |xh(u_2)y'_2|$, which is again a contradiction.

8. Case $\alpha_1 = (x, B, y'_1)a_1(x'_1, C, y)$ and $\alpha_2 = (x, B, y'_2)$. Now the definitions (3b) and (3d), and Lemma 3 would give $z_1 = xh(u_1)y'_1 = xh(w)y'_2$ and $z_2 = x'_1h(v_1)y = (y'_2)^{-1}y$, where u_1 and v_1 are subwords of w generated by (x, B, y'_1) and (x'_1, C, y) such that $u_1a_1v_1 = w$, which is a contradiction since $|x'_1h(v_1)y| > |(y'_2)^{-1}y|$.

9. The case of $\alpha_1 = (x, B, y'_1)$ and $\alpha_2 = (x, C, y'_2)$, with $y'_1 \neq y'_2$, is symmetric to case 1.

Thus every word in $L_{G'}((x, A, y))$ is generated by exactly one rule. □

3.2 Projections

Let $\Sigma = \Sigma_0 \cup \Sigma_1$ be an alphabet and $h_0 : \Sigma^* \to \Sigma_1^*$ a projection, that is, $h_0(a_0) = \varepsilon$ for all $a_0 \in \Sigma_0$ and $h_0(a_1) = a_1$ for all $a_1 \in \Sigma_1$.

Let $G = (\Sigma_1, N, P, S)$ be a Boolean grammar in binary normal form. We will construct a grammar $G' = (\Sigma, N', P_0, S')$ for the language $h_0^{-1}(L(G))$.

Let $N' = N \cup \{S', T\}$ be the set of nonterminals. Then P_0 contains rules

$$S' \to TST \tag{6a}$$

$$T \to a_0 T \mid \varepsilon \quad (\text{for all } a_0 \in \Sigma_0) \tag{6b}$$

$$\begin{aligned} &A \to B_1 T C_1 \& \ldots \& B_m T C_m \& \neg D_1 T E_1 \& \ldots \& \neg D_n T E_n \& \neg \varepsilon \\ &\quad (\text{for all } A \to B_1 C_1 \& \ldots \& B_m C_m \& \neg D_1 E_1 \& \ldots \& \neg D_n E_n \& \neg \varepsilon \in P) \end{aligned} \tag{6c}$$

$$A \to a \quad (\text{for all } A \to a \in P) \tag{6d}$$

It is easy to see that the constructed grammar is well-defined:

Lemma 5. *The system of equations corresponding to G' has a strongly unique solution.*

Sketch of a proof. For every finite subword-closed language M it has to be proved that the solution modulo M is unique. Induction on $|M|$.

Induction basis. The unique solution modulo $\{\varepsilon\}$ is has $L_A = \varnothing$ for all $A \in N \setminus \{S\}$, $L_T = \{\varepsilon\}$ and $L_S = \{\varepsilon \mid S \to \varepsilon \in P\}$.

Induction step. Let $M = M' \cup \{w\}$, with $w \notin M'$ and with all subwords of w in M'. By the induction hypothesis, the solution modulo M' is unique.

Let $(L_{S'}, L_S, \ldots, L_A, \ldots)$ be any solution modulo M. Then the membership of w in L_A depends upon its membership in concatenations of the form $L_B \cdot L_T \cdot L_C$, with $B, C \in N$. Since $\varepsilon \notin L_B, L_C$, this depends upon the membership of words shorter than w in these languages, which is uniquely defined by assumption. □

In proving the construction correct, we first prove a correspondence of nonterminals in G and G'.

Lemma 6. *For every $A \in N$, $w \in L_{G'}(A)$ if and only if $h_0(w) \in L_G(A)$ and $w \in \Sigma_1 \cup \Sigma_1(\Sigma_0 \cup \Sigma_1)^*\Sigma_1$.*

Proof. Induction on $|w|$.

Basis: $|w| = 1$. If $w = a \in \Sigma_1$. Now $a \in L_{G'}(A)$ if and only if there is a rule $A \to a$ in P', which, by (6d) exists if and only if $h_0(a) = a \in L_G(A)$.

On the other hand no $a_0 \in \Sigma_0$ can be in $L_{G'}(A)$, since no rule of the form (6d) generate them, and all words generated by rules of the form (6c) are of length at least 2.

Induction hypothesis: $w' \in L_{G'}(A)$ if and only if $h_0(w') \in L_G(A)$ and $w' \in \Sigma_1 \cup \Sigma_1(\Sigma_0 \cup \Sigma_1)^*\Sigma_1$ for $|w'| < |w|$.

Induction step: Let us first prove that under the induction hypothesis

$$w \in L_{G'}(BTC) \text{ if and only if } h_0(w) \in L_G(BC). \tag{7}$$

If $w \in L_{G'}(BTC)$, there is a factorization $w = uxv$, where $u \in L_{G'}(B)$, $x \in L_{G'}(T)$ and $v \in L_{G'}(C)$. Then $u, v \neq \varepsilon$, and hence $|u|, |v| < |w|$. By the induction hypothesis for u and v, $h_0(u) \in L_G(B)$ and $h_0(v) \in L_G(C)$. So $h_0(uxv) = h_0(u)h_0(x)h_0(v) = h_0(u)h_0(v) \in L_G(BC)$.

Conversely if $h_0(w) \in L_G(BC)$, there is a factorization $w = u'v'$, such that $h_0(u') \in L_G(B)$ and $h_0(v') \in L_G(C)$. This implies $u', v' \neq \varepsilon$ and thus $|u'|, |v'| < |w|$. Let $x \in \Sigma_0^*$ be the longest suffix of u' comprised of symbols from Σ_0, that is, $u' = ux$ with $u \in \Sigma_1 \cup \Sigma_1(\Sigma_0 \cup \Sigma_1)^*\Sigma_1$. Then, by the induction hypothesis, $u \in L_{G'}(B)$. Similarly, let $y \in \Sigma_0^*$ be the longest prefix of v' containing only symbols from Σ_0: we have $v' = yv$ with $v \in \Sigma_1 \cup \Sigma_1(\Sigma_0 \cup \Sigma_1)^*\Sigma_1$, and the induction hypothesis gives $v \in L_{G'}(C)$. Combining these, we obtain $w = uxyv \in L_{G'}(BTC)$, which completes the proof of (7).

To prove the induction step, first consider that $w \in L_{G'}(A)$ is equivalent to the existence of a rule (6c) in P', such that $w \in L_{G'}(B_iTC_i)$ for all applicable i and $w \notin L_{G'}(D_jTE_j)$ for all applicable j. Such a rule exists if and only if P contains a rule

$$A \to B_1C_1 \& \ldots \& B_mC_m \& \neg D_1E_1 \& \ldots \& \neg D_nE_n \& \neg\varepsilon. \tag{8}$$

On the other hand, $h_0(w) \in L_G(A)$ holds if and only if there exists a rule (8) with $h_0(w) \in L_G(B_iC_i)$ and $h_0(w) \notin L_G(D_jE_j)$. Now, by (7), $w \in L_{G'}(B_iTC_i)$ holds if and only if $h_0(w) \in L_G(B_iC_i)$, and $w \notin L_{G'}(D_jTE_j)$ holds if and only if $h_0(w) \notin L_G(D_jE_j)$. Therefore, $w \in L_{G'}(A)$ is equivalent to $h_0(w) \in L_G(A)$. □

Now we are ready to prove the construction correct.

Lemma 7. *For the constructed grammar G' it holds that $L(G') = h_0^{-1}(L(G))$.*

Proof. Let $w \in L_{G'}(S')$. By Lemma 6, this is equivalent with $w = xw'y$, where $x, y \in \Sigma_0^*$, $w' \in \Sigma_1 \cup \Sigma_1(\Sigma_0 \cup \Sigma_1)^*\Sigma_1$ and $h_0(w') \in L_G(S)$. Furthermore, this is equivalent with $h_0(w) \in L_G(S)$. □

We will complete the proof of Theorem 1 by showing that also the construction in this section preserves unambiguity.

Lemma 8. *If G is unambiguous, then G' is also unambiguous.*

Proof. If $w \in L_{G'}(TST)$, then $w = xw'y$, where $x, y \in \Sigma_0^*$ and $w' \in \Sigma_1 \cup \Sigma_1(\Sigma_0 \cup \Sigma_1)^*\Sigma_1$, is the unique factorization of w with respect to $L_{G'}(T)$ and $L_{G'}(S)$.

All the other conjuncts that have multiple nonterminals are of the form BTC. So, let $w \in L_{G'}(BTC)$. Suppose $w = u_1x_1v_1 = u_2x_2v_2$, with $u_1, u_2 \in L_{G'}(B)$, $x_1, x_2 \in L_{G'}(T)$ and $v_1, v_2 \in L_{G'}(C)$. Then by Lemma 6, $h_0(w) = h_0(u_1v_1) = h_0(u_2v_2) \in L_G(BC)$. In addition, $u_1, u_2 \in \Sigma_1 \cup \Sigma_1(\Sigma_0 \cup \Sigma_1)^*\Sigma_1$. It follows that if $|u_1| < |u_2|$, then also $|h_0(u_1)| < |h_0(u_2)|$. This means there would be two different factorizations of $h_0(w)$ with respect to $L_G(B)$ and $L_G(C)$, which contradicts the unambiguity of G. This proves that conjuncts of G' yield unique factorizations.

Since the rules (6c) of G' are in one to one correspondence with long rules of G, the languages generated by these are disjoint. □

4 Conclusion

All known closure properties of Boolean grammars and their subfamilies are given in Table 1. The bottom half of the last column 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}
Reg	+	+	+	+	+	+	+	+	+
LinCF	+	−	−	−	−	+	+	+	+
CF	+	−	−	+	+	+	+	+	+
LinConj	+	+	+ [5]	−	− [5]	+	−	−	+ [2]
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] 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.

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

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

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

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

[8] A. Okhotin, “Unambiguous Boolean grammars”, TUCS Technical Report No 802, Turku Centre for Computer Science, Turku, Finland, January 2007; also in *Proceedings of LATA 2007* (Tarragona, Spain).

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 978–952–12–1965–8
ISSN 1239-1891