Acta
Universitatis
Tamperensis

ser A vol 198

Erkki Mäkinen

On context-free derivations

University of Tampere

Tampere 1985

Acta
Universitatis
Tamperensis

ser A vol 198

Erkki Mäkinen

On context-free derivations

ACADEMIC DISSERTATION

To be presented, with the permission of the Faculty of Economics and Administration of the University of Tampere, for public discussion in Auditorium A1 of the University on November 30th, 1985, at 12 noon.

University of Tampere

Tampere 1985

Acta
Universitatis
Tamperensis

ser A vol 198

Erkki Mäkinen

On context-free derivations

University of Tampere

Tampere 1985

978-951-44-8123-9 (pdf)

ISBN 951-44-1837-9
ISSN 0496-7909

Tampereen Pikakopio Oy, Tampere 1985

Abstract. *The Szilard language of a context-free grammar is the set of all terminating derivations. By restricting the form of derivations we obtain the classes of left, depth-first and breadth-first Szilard languages of context-free grammars. This work surveys these classes of languages.*

In the case of (unrestricted) Szilard languages we mainly deal with different kinds of decidability problems and the relationship between Szilard languages and context-free languages. Two interesting topics concerning left Szilard languages are the properties of bounded left Szilard languages and the relationship between left Szilard languages and pure languages. Depth-first and breadth-first Szilard languages are not studied in the literature as much as Szilard and left Szilard languages. We shall present their basic properties only.

A great variety of grammatical similarity relations are introduced in the literature in order to control grammatical transformations. We shall show that left Szilard languages are useful when studying the properties of these similarity relations.

CR Categories and Subject Descriptors: F.4.2.[Mathematical Logic and Formal Languages]: Grammars and Other Rewriting Systems - Grammar types; F.4.3. [Mathematical Logic and Formal Languages]: Formal Languages - Classes defined by grammars or automata, Decision problems.

General Terms: Theory

Additional Keywords and Phrases: Szilard language, grammatical similarity.

ACKNOWLEDGEMENTS

I wish to thank Professors Pertti Järvinen and Reino Kurki-Suonio for their advice during the course of my work. I am especially indebted to Dr. Martti Penttonen for his kind interest and expert criticism at various stages of my work. For the revision of the English manuscript I am grateful to Mr. Walter Bacon, M.Sc.

This work was supported by the Academy of Finland.

CONTENTS

CHAPTER 1: INTRODUCTION AND PRELIMINARIES

1.1. Introduction

There are two major things of interest when studying context-free grammars as a mathematical model of programming languages: (1) the language generated by the grammar and (2) the derivational structures used. This study is devoted to the latter subject.

A traditional way of presenting the derivational structure of a context-free grammar is to use the forest of derivation trees. In this paper we present the derivational structure of a context-free grammar by a language in which we have a word for every string of productions that corresponds to a derivation from the start symbol to a terminal word. Such a language is called the Szilard language of a context-free grammar. (Szilard languages are called "associate languages" in [Mor] and "derivation languages" in [Pen74].) By setting conditions to the form of derivations, we can define leftmost, depth-first and breadth-first derivations and left, depth-first and breadth-first Szilard languages, respectively. Rightmost derivations are not studied, since the results would be analogous to those of leftmost derivations.

In chapter 2 we survey the material concerning Szilard languages associated with arbitrary context-free derivations and in chapter 3 the same is done for left Szilard languages. Although left Szilard languages are always deterministic context-free languages and Szilard languages are not necessarily even context-free, the two classes of languages have several similarities, e.g. their closure properties are alike.

In chapter 4 we are looking for ways to restrict arbitrary context-free derivations such that the resulting derivations are more general than leftmost derivations and the Szilard languages associated with them are context-free. The depth-first restriction is our best result to this direction. We also define breadth-first Szilard languages, but this class does not have the desired context-freeness property. However, there are context-free grammars which have a regular breadth-first Szilard language but a non-regular left Szilard language.

In chapter 5, our purpose is to apply the results of chapter 3 concerning left Szilard languages to the study of grammatical similarity relations. We show that the relations undercover [S-SW] and cover [GH] can be characterized by using the properties of left Szilard languages and the concept of homomorphism equivalence.

1.2. Preliminaries

We assume a familiarity with the basics of context-free grammars and languages, decidablity and other related topics as given in

[Har,HU79]. All unexplained concepts are as in these references.

1.2.1. Context-free grammars

We denote a *context-free grammar* by $G = (N,T,P,S)$ where N is the alphabet of *nonterminals*, T is the alphabet of *terminals*, P is the set of *productions*, and S is the *start symbol*.

Relations *"derives directly"* ($\Rightarrow$) and *"derives directly leftmost"* ($\Rightarrow_l$) and their transitive ($\Rightarrow^+$ and $\Rightarrow_l^+$) and reflexive, transitive ($\Rightarrow^*$ and $\Rightarrow_l^*$) closures are defined as usual. $L(G)$ stands for the language generated by G.

A derivation $S \Rightarrow^+ \alpha$ in $G = (N,T,P,S)$ is said to be a *terminal derivation*, if $\alpha \in T^*$. If not otherwise stated, we suppose that all context-free grammars are *reduced*, i.e. all nonterminals and terminals appear in some terminal derivation. A derivation $A \Rightarrow^+ \alpha$, $\alpha \in (N \cup T)^*$, is *recursive*, if $\alpha = \alpha_1 A \alpha_2$. Nonterminal A is then said to be a *recursive nonterminal*.

The empty word is denoted by λ, the length of a word α by $lg(\alpha)$, the empty set by Φ, the cardinality of a set A by $card(A)$, and the set of n-tuples of non-negative integers by $\mathbb{N}^n$.

The following subclasses of context-free languages (i.e. languages generated by context-free grammars) are assumed to be known: *simple languages* (*s-languages*), *linear languages* [Har,HU79], and *regular languages* [HU69]. We also consider *s-*, *linear* [Har, HU79] and *regular grammars* [HU69].

We shall need some normal forms for context-free grammars. A

context-free grammar $G = (N,T,P,S)$ is said to be in *Chomsky normal form* (in CNF, for short) if $P \subseteq N \times (N^2 \cup T)$. The production $S \to \lambda$ is also allowed if the start symbol S does not appear in the right-hand side of any production. G is said to be in *Greibach normal form* (in GNF, for short) if $P \subseteq N \times TN^*$. The production $S \to \lambda$ is allowed as above. G is in *semi-GNF*, if $P \subseteq N \times T^*N^*$. Appearances of the start symbol are not restricted in semi-GNF.

A context-free grammar G is *ambiguous* if there exists a word w in $L(G)$ such that w has at least two leftmost derivations from the start symbol. Otherwise, G is *unambiguous*. A context-free language for which every context-free grammar is ambiguous is said to be an *inherently ambiguous context-free language*.

Let $G = (N,T,P,S)$ be a context-free grammar and assume that the productions in P are uniquely labeled by the symbols of an alphabet C. The alphabet C is then called the *label alphabet* of G. If a production $A \to \alpha$ is associated with a label ρ, we write $\rho : A \to \alpha$. If a sequence $\rho_1 \ldots \rho_n = \rho$ of labeled productions is applied in a derivation (resp. leftmost derivation) $\beta \Rightarrow^* \gamma$ (resp. $\beta \Rightarrow_l^* \gamma$), where β and γ are in $(N \cup T)^*$, we can write $\beta \Rightarrow^\rho \gamma$ (resp. $\beta \Rightarrow_l^\rho \gamma$). The *Szilard language* $Sz(G)$ and the *left Szilard language* $Szl(G)$ of G are defined as

$$Sz(G) = \{ \pi \mid S \Rightarrow^\pi w,\ w \in T^* \}$$

and

$$Szl(G) = \{ \pi \mid S \Rightarrow_l^\pi w,\ w \in T^* \}.$$

Hence, both $Sz(G)$ and $Szl(G)$ are languages over label alphabet C.

We shall sometimes speak about derivation π, when we

actually mean derivation $\alpha \Rightarrow^{\pi} \beta$ (or $\alpha \Rightarrow_l^{\pi} \beta$).

The following notational conventions concerning symbols and strings are generally used: small Latin letters from the beginning of the alphabet a, b, c,... denote terminals, capital Latin letters from the beginning of the alphabet A, B, C,... denote nonterminals, small Latin letters from the end of the alphabet u, v, w,... denote terminal strings, and Greek letters from the beginning of the alphabet α, β, γ,... denote general strings, i.e. strings consisting of terminals, nonterminals or both. We need one more convention for the symbols and strings of label alphabets. We shall use Greek letters from the end of the alphabet π, ρ, σ,... for denoting both symbols of a label alphabet and strings of labels. This should not cause any confusion since it is possible to conclude from the context whether we mean labels of productions or strings of labeled productions.

Let $G = (N,T,P,S)$ be a context-free grammar. We often need the homomorphism $\eta:(N \cup T)\to(N \cup \{\lambda\})$ defined by $\eta(A) = A$, if $A \in N$, and $\eta(a) = \lambda$, if $a \in T$. (A homomorphism $h:\Delta\to\Sigma$, where Δ and Σ are alphabets, is naturally extended to be a function from Δ^* to Σ^* by the conditions $h(\lambda) = \lambda$ and $h(\pi\rho) = h(\pi)h(\rho)$ for all π in Δ^* and ρ in Δ.)

1.2.2. Pushdown automata

A *pushdown automaton* is a 7-tuple $A = (Q,\Sigma,\Gamma,\delta,q_0,Z_0,F)$, where Q is the set of *states*, Σ and Γ are the alphabets of *input* and *pushdown symbols*, respectively, $q_0 \in Q$ is the *initial state*, $Z_0 \in \Gamma$ is the *initial symbol* of the pushdown store, $F \subseteq Q$ is

the set of *final states* and δ is the *transition function* from $Q \times (\Sigma \cup \{\lambda\}) \times \Gamma$ to finite subsets of $Q \times \Gamma^*$ [Har].

We consider only pushdown automata, which accept their input "by final state and empty store" [Har, p. 139]. $L(A)$ stands for the language accepted by a pushdown automaton A.

An *instantaneous description* is an element of $Q \times \Sigma^* \times \Gamma^*$. A move is a change in instantaneous description according to the transition function δ. If (q,x,α) is an instantaneous description, then $lg(\alpha)$ is the *height* of the pushdown store.

A pushdown automaton $A = (Q,\Sigma,\Gamma,\delta,q_0,Z_0,F)$ is *deterministic* if for all $(q,a,z) \in Q \times (\Sigma \cup \{\lambda\}) \times \Gamma$, we have $card(\delta(q,a,Z)) \leq 1$ and if $\delta(q,\lambda,Z) \neq \Phi$ then $\delta(q,a,Z) = \Phi$ for each $a \in \Sigma$. A context-free language is *deterministic*, if some deterministic pushdown automaton accepts it.

A pushdown automaton $A = (Q,\Sigma,\Gamma,\delta,q_0,Z_0,F)$ is said to be a *one-counter automaton*, if the set of pushdown symbols contains only one symbol other than the initial symbol of the pushdown store. Since our pushdown automata accept by final state and empty store, the counter may count down to zero more than once. Such automata are called "iterated one-counter automata" in [Har, pp. 147-148]. By *"one-counter language"* we mean a language acceptable by a one-counter automaton, i.e. "iterated one-counter language" of [Har].

In the sequel, we assume that the concepts *"the delay of a pushdown automaton"* and *"a realtime automaton"* [Har, p. 142] are known.

1.2.3. Decision problems and complexity

When we speak about decidability and complexity of problems we suppose that the problems are encoded to language recognition problems as in [HU79, p. 178].

A problem whose language is recursive is said to be *decidable*. Otherwise, the problem is *undecidable*.

P, NP and PSPACE are classes of languages recognized by deterministic polynomially time bounded Turing machines, by nondeterministic polynomially time bounded Turing machines and by deterministic polynomially space bounded Turing machines [HU79, AHU], respectively.

A language L is *polynomially reducible* to a language L' if there exists a function f computable by some deterministic polynomially time bounded Turing machine such that x is in L if and only if f(x) is in L'. Languages L_1 and L_2 are *polynomially equivalent* if they are polynomially reducible to each other.

NP-complete and *PSPACE-complete* languages (and problems) are supposed to be defined as in [HU79].

CHAPTER 2: ON SZILARD LANGUAGES

This chapter deals with Szilard languages of context-free grammars. In section 2.1 we introduce some basic properties of Szilard languages. Section 2.2 shows the relationship between Szilard languages and counter automata. Since there are non-context-free Szilard languages, we need more than one counter when recognizing these languages. On the other hand, we can represent Szilard languages as intersections of one-counter languages. Section 2.3 is devoted to a study of decision problems for Szilard languages. It is shown that many problems undecidable for context-free grammars and languages are decidable for Szilard languages. In section 2.4 we study the resemblance between Szilard languages and context-free languages. Different characterizations of context-free languages are applied to Szilard languages in order to check whether these characterizations are strong enough to distinguish between Szilard languages and context-free languages.

2.1. Basic properties of Szilard languages

We shall first make some elementary remarks concerning Szilard languages.

Let $G = (N,T,P,S)$ be a context-free grammar with label alphabet C. A production $A \to \alpha$ is said to be *terminating*, if

$\alpha \in T^*$; otherwise it is *nonterminating*. If a word in $Sz(G)$ is of length two or more, it begins with the label of a nonterminating production which has the start symbol S in its left-hand side. Every word in $Sz(G)$ ends with the label of a terminating production. Based on this simple remark we could now find counterexamples which show that the class of Szilard languages is not closed under union, catenation, homomorphism, inverse homomorphism, and Kleene closure [Pen74]. Language $\{ \pi^*\rho \}$ shows that the class of Szilard languages is not closed under reversal.

The following example shows that the class of Szilard languages is not closed under intersection with regular languages. We need here the fact that every infinite Szilard language contains an infinite regular language [Pen74].

Example 2.1 [Pen74]. Let G be a context-free grammar with productions

$\pi: S \to ABS$

$\rho: A \to \lambda$

$\sigma: B \to \lambda$

$\tau: S \to \lambda$.

Furthermore, let $R = \pi^*\rho^*\sigma^*\tau$. Now we have $Sz(G) \cap R = \{ \pi^n\rho^n\sigma^n\tau \mid n \geq 0 \}$, which does not contain an infinite regular language. □

A few further comments concerning example 2.1 are in order. First, productions $\rho: A \to \lambda$, $\sigma: B \to \lambda$ and $\tau: S \to \lambda$ all have empty right-hand sides. In fact, it would not make any difference, if there were

terminal symbols in these right-hand sides. However, if we do not want to allow more than one label for a single production, we sometimes need terminal symbols to make productions distinguishable.

Secondly, example 2.1 shows that there are non-context-free Szilard languages.

Recall that a language L is *prefix-free*, if $x \in L$ and $xy \in L$ always imply $y = \lambda$. Let π be in Sz(G) for a context-free grammar G. It is not possible to continue derivation after production string π, because there are no nonterminals left. That is, Sz(G) is prefix-free.

For a language L, define init(L) to be the set $init(L) = \{ x \mid xy \in L \}$, i.e. init(L) is the set of prefixes for a language L. Every word in Sz(G) corresponds to a terminal derivation of a context-free grammar $G = (N,T,P,S)$. Similarly, every word ρ in init(Sz(G)) corresponds to a derivation $S \Rightarrow^{\rho} \alpha$, where $\alpha \in (N \cup T)^*$.

2.1.1. The rank of context-free grammars

We have already seen that not all Szilard languages are context-free. Following [Mor] we now give sufficient and necessary conditions for a Szilard language to be regular or context-free.

A context-free grammar is said to have *rank* k if there exists a natural number n such that in any sentential form at most

k nonterminals have more than n occurrences, and moreover, there does not exist any k', $k' < k$, having the same property.

If a context-free grammar has rank 0, it is called *nonterminal bounded* and if its rank is at most 1, then it is called *half-bounded*.

When speaking about symbols and strings the following notation is useful: $X(\alpha)$ denotes the number of occurrences of symbol X in string α.

A nonterminal A is said to be *bounded* in a context-free grammar G, if there is a natural number n such that every sentential form α of G has $A(\alpha) < n$. Otherwise, A is *unbounded*.

If a context-free grammar is nonterminal bounded, then it has a finite number of possible nonterminal combinations in its sentential forms. By replacing every different combination by a different nonterminal, we see that in this case the Szilard language can be generated by a regular grammar. On the other hand, it is obvious that if a context-free grammar is not nonterminal bounded, then its Szilard language cannot be regular. Hence, we obtain

Theorem 2.1 [Mor]. The Szilard language $Sz(G)$ of a context-free grammar G is regular if and only if G is nonterminal bounded.

Moriya [Mor] has proved a similar result for half-bounded grammars and context-free Szilard languages. We can state the theorem in somewhat stronger form. Consider first the if-part.

Lemma 2.1. Let $G = (N,T,P,S)$ be a half-bounded context-free grammar. Then $Sz(G)$ is a deterministic one-counter language.

Proof. The problem is that we might have more than one unbounded nonterminal. We must be able to decide which unbounded nonterminal has to be counted by the counter.

For each unbounded nonterminal A we must have a nonterminal C such that $C \Rightarrow^* \beta$, where $C(\beta) = 1$ and $A(\beta) > 0$ (or $A = C$ and $A(\beta) > 1$), is a derivation in G. Let N_A be the set of all nonterminals C fulfilling the condition above. If B is another unbounded nonterminal in G, we have $N_A \cap N_B = \Phi$. Moreover, if C is in N_A, then no nonterminal in N_B is reachable from C. The latter property means that if we ever decide to use the counter for counting the number of a certain nonterminal, there is no need to change the nonterminal counted in any further stage.

If the start symbol is unbounded in a half-bounded context-free grammar, then it clearly is the only nonterminal of the grammar. Hence, we can assume that the start symbol is bounded.

We can now construct a deterministic one-counter automaton F which accepts $Sz(G)$ as follows. Automaton F starts reading its input and remembering the numbers of nonterminals in the corresponding derivation of G in its state system. When F reads a label ρ such that $\rho: B \to \alpha$ is a production and α contains a nonterminal from some N_A, all A's kept in the state system will be moved to the counter. Henceforth, reading a label of a production containing A causes a change in the counter. So, for every unbounded nonterminal of G, F has a block of states in which it stays while reading the rest of its input after it has decided which nonterminal must be counted by the counter. □

Similarly, we could prove the following: for every half-bounded context-free grammar G we can find context-free grammars $G_1,\ldots,G_n$ such that each G_i, $i = 1,\ldots,n$, has only one unbounded nonterminal, $L(G) = \bigcup_{i=1}^{n} L(G_i)$ and each derivation of G is in G_i, for at least one value of i [Mäk85d].

Consider now the converse of lemma 2.1. Following [Mor] we prove it by supposing first that a given context-free grammar $G = (N,T,P,S)$ is not half-bounded. Hence, we have sentential forms with occurrences of two, say A and B, $A \neq B$, unbounded nonterminals.

We obviously need recursive derivations which increase the numbers of A's and B's. The recursive nonterminal in these derivations is A, B or some distinct nonterminal C. All possible forms of these derivations are listed below according to the number of nonterminals (other than A or B) needed.

(0) a. $A \Rightarrow^* \ldots A \ldots A \ldots$

$B \Rightarrow^* \ldots B \ldots B \ldots$

b. $A \Rightarrow^* \ldots B \ldots A \ldots A \ldots$

c. $B \Rightarrow^* \ldots A \ldots B \ldots B \ldots$

(1) d. $C \Rightarrow^* \ldots A \ldots B \ldots C \ldots$

e. $C \Rightarrow^* \ldots A \ldots C \ldots$

$B \Rightarrow^* \ldots B \ldots B \ldots$

f. $C \Rightarrow^* \ldots B \ldots C \ldots$

$A \Rightarrow^* \ldots A \ldots A \ldots$

(2) g. $C \Rightarrow^* \ldots A \ldots C \ldots$

$D \Rightarrow^* \ldots B \ldots D \ldots$

We consider case d. in detail; the other cases are similar. Name

first some derivations in G:

$C \Rightarrow^{\pi} u_1Au_2Bu_3Cu_4$, $u_i \in T^*$, $i = 1,\ldots,4$,

$A \Rightarrow^{\rho} v_1$, $B \Rightarrow^{\sigma} v_2$, $C \Rightarrow^{\tau} v_3$, $v_i \in T^*$, $i = 1,\ldots,3$.

Furthermore, G must have derivation

$S \Rightarrow^{\phi} w_1Cw_2$, $w_1, w_2 \in T^*$.

Now, we have $Sz(G) \cap \phi\pi^*\rho^*\sigma^*\tau = \{ \phi\pi^n\rho^n\sigma^n\tau \mid n \geq 0 \}$. Since $Sz(G)$ has a non-context-free intersection with a regular language, it cannot be even context-free.

We have proved the following

Theorem 2.2. Let G be a context-free grammar. Language $Sz(G)$ is a deterministic one-counter language if and only if G is half-bounded.

Notice, that the rank of a context-free grammar is always finite, because the set of nonterminals is supposed to be finite. There is an algorithm for determining the rank of a given context-free grammar [Mor].

2.1.2. On grammars generating Szilard languages

In this section we introduce two methods for generating Szilard languages. We shall also make some remarks concerning the generative capacity of permutative grammars which are introduced as the other method for generating Szilard languages.

Recall first that a (context-free) *matrix grammar* [Sal] consists of sequences of context-free productions, i.e. matrices, which

can be applied only in such a way that all productions in a matrix are applied in succession.

Let $G = (N,T,P,S)$ be a context-free grammar. Define a matrix grammar M, which has matrices $[S\to\$S]$, $[\$\to\lambda]$ and for every production $\rho:A\to\alpha$ in G a matrix $[\$\to\rho\$, A\to\eta(\alpha)]$. The language generated by M is $Sz(G)$ [Pen74].

A matrix of the form $[\$\to\rho\$, A\to\eta(\alpha)]$ makes it possible to arbitrarily pick a nonterminal A to be the nonterminal to which the next production is applied. We can do the same thing by using permutation productions of the form $AB\to BA$. In this case we actually transform the chosen nonterminal to the left-hand side end of the nonterminal portion in a sentential form.

A grammar is said to be *permutative* if it has permutation productions of the form $AB\to BA$ besides context-free ones [Sil].

Let $G = (N,T,P,S)$ be a context-free grammar with label alphabet C. Define a permutative grammar $H = (N,C,P_{per},S)$, where P_{per} contains

1) $A\to\rho\eta(\alpha)$ for every production $\rho:A\to\alpha$ in P

and

2) $AB\to BA$ for every pair (A,B), $A \neq B$, in $N \times N$.

We omit the straightforward proof of the fact that $L(H) = Sz(G)$.

By theorem 2.2, we obtain a permutative grammar generating a context-free language if and only if the original context-free grammar G is half-bounded. By relieving the one-to-one correspondence

between productions and terminal symbols in grammar H, we fall into the class of so-called label grammars [Höp, JO].

A permutative grammar $H = (N,T,P,S)$ is a *label grammar* if all context-free productions in P have the form $A \to a\alpha$, where $a \in (T \cup \{\lambda\})$ and $\alpha \in N^*$, and moreover, P contains a production $AB \to BA$ for every pair (A,B), $A \neq B$, in $N \times N$. A language L is a *label language* if there is a label grammar H such that H generates L.

Label languages are images of Szilard languages under decreasing homomorphisms (a homomorphism h is *decreasing* if $lg(h(a)) \leq 1$, for all a) [JO]. Hence, theorem 2.2 gives us a sufficient condition for a label grammar to generate a context-free language.

Theorem 2.3 [Mäk85d]. Let $H = (T,N,P,S)$ be a label grammar and let $G = (N,T,P',S)$ be the context-free grammar obtained from H by deleting all permutation productions. If G is half-bounded, then L(H) is context-free.

The following example demonstrates why half-boundedness is not necessary for a label grammar to generate a context-free language.

Example 2.2. Consider label grammars G_1, G_2 and G_3 with the following context-free productions

G_1:	G_2:	G_3:
S→cSAB	S→SAB	S→cSAB
A→a	A→a	A→a
B→b	B→b	B→a
S→d	S→d	S→d.

The corresponding context-free grammars all have rank 2. Since G_1 generates the Szilard language of a context-free grammar of rank 2, $L(G_1)$ is not context-free. On the other hand, $L(G_2)$ and $L(G_3)$ are both context-free. □

Label grammars have the right-hand sides of their context-free productions in $TN^* \cup N^*$. If we allow arbitrary context-free productions, then half-boundedness is not sufficient for context-freeness, as shown in the following example.

Example 2.3 [Mäk85d]. Consider a permutative grammar G with the following context-free productions

$S \to aAB$

$B \to Sc$

$A \to b$

$S \to d$

and with all possible permutation productions. Each word in $L(G)$ contains equal number of terminals a, b and c. By applying derivations of the form

$$S \Rightarrow aAB \Rightarrow aASc \Rightarrow aSAc \Rightarrow aaABAc \Rightarrow aaAABc \Rightarrow aaAAScc \Rightarrow aaASAcc \Rightarrow$$
$$aaSAAcc \Rightarrow aaaABAAcc \Rightarrow aaaAABAcc \Rightarrow aaaAAABcc \Rightarrow aaaAAAScccc \Rightarrow \ldots$$

words $a^n b^n d c^n$, $n > 0$, can be generated in G. Hence, $L(G) \cap a^*b^*dc^* = \{ a^n b^n d c^n \mid n > 0 \}$. This shows that $L(G)$ is not context-free. □

The situation becomes even more complicated when we do not require that a permutation production $AB \to BA$ exists for every nonterminal pair (A,B), $A \neq B$, but only for a subset of $N \times N$. We omit these considerations.

2.2 Counter automata and Szilard languages

A deterministic m-counter automaton consists of a finite state system and m counters (for details, see [FMR]). Such a device recognizes Szilard languages in realtime as describred in [Pen77]. When an m-counter automaton recognizes a word in a Szilard language, it simulates the corresponding derivation of the context-free grammar in question by counting the numbers of nonterminals. The contents of these counters are linearly bounded (by the length of the input string) and the counter automaton can be simulated by a deterministic Turing machine which works in space $\log n$ [Pen77]. This Turing machine construction is directly given in [Iga], where it is also shown that $\log n$ is optimal space bound for on-line deterministic Turing machines, i.e. for machines which have read-only input tape and are not allowed to move the input head to the left.

What is the number of counters needed in an m-counter automaton which recognizes a Szilard language? Since the numbers of occurrences of bounded nonterminals can be counted by the state system, we need a counter for each unbounded nonterminal only. Instead of proving this fact in detail, we consider the following theorem, which states a similar result and is given in [Jan] without a proof.

Theorem 2.4. If a context-free grammar $G = (N,T,P,S)$ has k unbounded nonterminals, then the Szilard language $Sz(G)$ is expressible as the intersection of k deterministic one-counter languages.

Proof. Notice first that when $k = 0$, we have a special case where $Sz(G)$ is a regular language and hence, the phrase "the

intersection of 0 deterministic one-counter languages" must be interpreted as "a regular language". When $k = 1$, we have the case of lemma 2.1.

We shall now turn to the case of arbitrary $k > 1$. Let A be a nonterminal in G. Consider those labels in a word $\rho = \rho_1 \ldots \rho_n$ of $Sz(G)$ whose productions contain at least one occurrence of A. If ρ_i has A in its left-hand side, we must have at least one A left in the sentential form obtained by applying productions $\rho_1 \ldots \rho_{i-1}$. On the other hand, after the last production "consuming" A (i.e. having A in the left-hand side but not in the right-hand side) all A's must be "consumed" and after that, new A's cannot be "produced".

Define L_A to be a language over label alphabet C such that L_A contains all balanced words with respect to labels of productions containing occurrences of A.

The deterministic one-counter automaton F_A accepting L_A scans its input and performs the following actions:

1) F_A checks that the first symbol of its input is an allowed one,
2) when F_A reads the label of a production with A as the left-hand side, F_A decrements its counter by one,
3) when F_A reads the label of a production with at least one occurrence of A in the right-hand side α, F_A increments its counter by one $A(\alpha)$ times.

Action 2) precedes action 3) if some production requires both of them. Since our one-counter automata accept with final state

and zero counter, F_A accepts its input if and only if the counter contains zero after reading its input. That is, F_A accepts L_A.

Consider now the intersection $\bigcap_{A \in N} L(A)$. For each nonterminal A we have a language L_A which forces the intersection to contain only words with right relative order of labels of productions having A in the left- or in the right-hand side. Since every L_A contains all words fulfilling this condition, the intersection must coincide with Sz(G).

If a nonterminal A is bounded then L_A is regular. Hence, all L_A's where A is bounded, can be replaced by one regular language. Moreover, deterministic one-counter languages are closed under intersection with regular languages. This completes the proof of the theorem. □

Corollary 2.1 [C-RM]. The complement $\overline{Sz(G)}$ of a Szilard language Sz(G) is context-free.

Is it possible to strengthen theorem 2.5 such that the phrase "k unbounded nonterminals" could be replaced by "rank k"? Lemma 2.1 shows that this is possible in the case k = 1. We leave the other cases open.

2.3. Decision problems for Szilard languages

Most reasonable decision problems are undecidable for context-free grammars and languages. However, it is decidable whether the language generated by a context-free grammar is empty, finite or

infinite. These problems are decidable for Szilard languages, too.

Theorem 2.5 [Pen74]. Let G be a context-free grammar. There are algorithms to determine if $Sz(G)$ is empty, finite or infinite.

Proof. $Sz(G)$ is empty if and only if $L(G)$ is empty and it is finite if and only if G does not have recursive nonterminals. Hence, we have algorithms for all three cases. □

Notice that $L(G)$ can be finite when $Sz(G)$ is infinite.

Consider now problems "$Sz(G) = R$?", "$Sz(G) \subseteq R$?" and "$R \subseteq Sz(G)$?", where G is a context-free grammar and R is a regular language.

Theorem 2.6. Let G be a context-free grammar and let R be a regular language. It is decidable whether or not $R \subseteq Sz(G)$ and $R = Sz(G)$.

Proof. We have $R \subseteq Sz(G)$ if and only if $R \cap \overline{Sz(G)} = \Phi$. By corollary 2.1, language $R \cap \overline{Sz(G)}$ is context-free and hence, it is decidable whether or not it is empty.

If G has rank higher than 0, then $Sz(G) = R$ is impossible, because $Sz(G)$ is not regular. Otherwise $Sz(G) = R$ is decidable, since the both languages are regular. □

Recall that a language L is *bounded*, if there exist finite words $w_1, \ldots, w_n$ such that $L \subseteq w_1^* \ldots w_n^*$. Otherwise, L is *unbounded*.

The following theorem is a simple remark concerning the problem "$Sz(G) \subseteq R$?". We shall reconsider this problem at the end of this section.

Theorem 2.7. Let G be a context-free grammar and let R be a bounded regular language. Then it is decidable whether or not $Sz(G) \subseteq R$.

Proof. If $G = (N,T,P,S)$ has rank 0 or rank 1, it is decidable whether $Sz(G) \subseteq R$, since $Sz(G)$ is deterministic context-free [Har].

Suppose now that the rank of G is at least 2. Let A and B be unbounded nonterminals in G such that the number of their simultaneous occurrences is not limited. Since we consider only reduced grammars, there are derivations $A \Rightarrow^{\rho} u$ and $B \Rightarrow^{\pi} v$ such that u and v are in T^*. Derivations ρ and π must begin with different productions. Now, all strings in $\{ \rho,\pi \}^*$ are subwords of some words in $Sz(G)$. Hence, $Sz(G)$ is not bounded and $Sz(G) \subseteq R$, where R is a bounded regular language, is impossible. □

2.3.1. The equivalence problem

In this subsection we consider the equivalence problem for Szilard languages. We need some new definitions. Let α and β be strings of nonterminals. We write $\alpha \equiv \beta$ if and only if for all nonterminals A we have $A(\alpha) \equiv A(\beta)$.

The following definition of grammar isomorphism differs slightly from that in [Pen74].

Definition 2.1. Let $G_1 = (N_1,T_1,P_1,S_1)$ and $G_2 = (N_2,T_2,P_2,S_2)$ be context-free grammars with a common label alphabet C. G_1 and

G_2 are said to be *isomorphic*, if their productions can be labeled so that one can define a homomorphism $h:N_1 \to N_2$ which has the following properties

1) $h:N_1 \to N_2$ is a bijection

and

2) for every pair $\rho:A \to \alpha$ in P_1 and $\rho:B \to \beta$ in P_2 we have $h(A) = B$ and $h(\eta(\alpha_1)) \equiv \eta(\alpha_2)$.

The following theorem is essential.

Theorem 2.8 [Pen74]. Let $G_1 = (N_1,T_1,P_1,S_1)$ and $G_2 = (N_2,T_2,P_2,S_2)$ be context-free grammars. Then the productions in P_1 and P_2 can be labeled so that $Sz(G_1) = S(G_2)$ if and only if G_1 and G_2 are isomorphic.

Proof. It follows directly from definition 2.1 that if G_1 and G_2 are isomorphic and if their productions are labeled as required in definition 2.1, then $Sz(G_1) = Sz(G_2)$.

Suppose now that $Sz(G_1) = Sz(G_2)$. Our first task is to show that relation $h:N_1 \to N_2$, defined by $h(A) = B$ if productions $A \to \alpha$ in P_1 and $B \to \beta$ in P_2 have a common label, is a bijection. We prove two claims, which will later be used as lemmas.

Claim 1. If $\rho:A \to \alpha$ is in P_1 and $\rho:B \to \beta$ is in P_2, then there exists a string π in C^* (C is the common label alphabet of G_1 and G_2) such that $A \Rightarrow^{\pi} v$ and $B \Rightarrow^{\pi} w$, where $v \in T_1^*$ and $w \in T_2^*$.

We write $\tau \leq \nu$ if string τ is obtained from string ν by erasing symbols of ν or if $\tau = \nu$.

We have a derivation $S_1 \Rightarrow^{\sigma} v_1Av_2 \Rightarrow^{\rho} v_1\alpha v_2 \Rightarrow^{\pi 0} v_1v_2v_3$, where $v_1, v_2, v_3 \in T_1^*$, in G_1. Since $\sigma\rho\pi_0$ is in $Sz(G_1)$ and $Sz(G_1) = Sz(G_2)$, we also have $S_2 \Rightarrow^{\sigma} \beta_1 \Rightarrow^{\rho\pi 0} w_1$, $w_1 \in T_2^*$, for some $\beta_1 \in (N_2 \cup T_2)^*$. Since production $\rho:B\to\beta$ is applicable to β_1, we have $B(\beta_1) > 0$. Hence, there is $\pi_1 \in C^*$, such that $B \Rightarrow^{\rho\pi 1} w_2$, $w_2 \in T_2^*$, and $\pi_1 \leq \pi_0$.

We can change the roles of G_1 and G_2 and find $\pi_2 \in C^*$ such that $A \Rightarrow^{\rho\pi 2} v_2$, $v_2 \in T_1^*$, and $\pi_2 \leq \pi_1$. We can repeat this until $\pi_{k+1} = \pi_k$. This completes the proof of claim 1.

Claim 2. If $S_1 \Rightarrow^{\pi} \gamma$ and $S_2 \Rightarrow^{\pi} \delta$, then $h(\eta(\gamma)) \equiv \eta(\delta)$.

Let A be any nonterminal in γ. By claim 1 we have σ such that $A \Rightarrow^{\sigma} v$, $v \in T_1^*$ and $h(A) \Rightarrow^{\sigma} w$, $w \in T_2^*$. Let k be the largest natural number such that $\pi\rho^k$ is in $init(Sz(G_1))$. Since $Sz(G_1) = Sz(G_2)$, $\pi\sigma^k$ is also in $init(Sz(G_2))$. Clearly k is the number of occurrences of A in γ and it must coincide with the number of occurrences of $h(A)$ in δ. This proves claim 2.

We now return to the proof of the claim about the bijectivity of h. It is sufficient to show that $h:N_1\to N_2$ and analogously defined $h':N_2\to N_1$ are both functions. By symmetry, we consider h only.

Let $\rho_1:A\to\alpha_1$ and $\rho_2:A\to\alpha_2$ be in P_1 such that $\rho_1:B\to\beta$ and $\rho_2:C\to\gamma$ are in P_2. Suppose $B \neq C$ and consider derivation $S_1 \Rightarrow^{\tau} u'Au''$, where $u', u'' \in T_1^*$. Both $\tau\rho_1$ and $\tau\rho_2$ are in $init(Sz(G_1))$ and hence, also in $init(Sz(G_2))$. We must have

$S_2 \Rightarrow^{\rho} \delta$ such that δ contains both B and C. This contradicts claim 2 and we must have $B = C$.

Since every production $\rho:A\to\alpha$ in P_1 has a corresponding production $\rho:B\to\beta$ in P_2, h maps every nonterminal A in N_1 to some B in N_2. The single-valueness of h is proved above. Hence, h is a function. By symmetry, this holds for h', too. That is, h is a bijection.

Consider productions $\rho:A\to\alpha$ in P_1 and $\rho:B\to\beta$ in P_2 and derivations

$$S_1 \Rightarrow^{\pi} \alpha_1 A \alpha_2 \Rightarrow^{\rho} \alpha_1\alpha\alpha_2$$
$$S_2 \Rightarrow^{\pi} \beta_1 B \beta_2 \Rightarrow^{\rho} \beta_1\beta\beta_2.$$

By claim 2, we have

$$h(\eta(\alpha_1 A \alpha_2)) \equiv \eta(\beta_1 B \beta_2)$$

and

$$h(\eta(\alpha_1\alpha\alpha_2)) \equiv \eta(\beta_1\beta\beta_2).$$

Hence, we have $h(\eta(\alpha)) \equiv \eta(\beta)$. This completes the proof. □

There is a deterministic polynomial time algorithm for deciding whether or not $Sz(G_1) = Sz(G_2)$ holds for arbitrary context-free grammars G_1 and G_2, whose productions are labeled. This algorithm first checks that the left-hand sides of productions with a common label define a bijection h. And secondly, h must map the right-hand sides of G_1's productions such that the images are congruent (modulo $\equiv$) with the right-hand sides of corresponding productions in G_2.

Next, we shall study a more complicated problem. Let G_1 and G_2

be context-free grammars. Is it possible to label the productions of G_1 and G_2 so that $Sz(G_1) = Sz(G_2)$? We shall show that this problem is polynomially equivalent to the digraph isomorphism problem. The digraph isomorphism problem and problems polynomially equivalent to it form an interesting class of problems, since it is open whether or not these problems are NP-complete (consult [GJ] for further details).

Digraphs $H_1 = (V_1,E_1)$ and $H_2 = (V_2,E_2)$ [HU79] are said to be *isomorphic* if there is a bijection $f:V_1 \to V_2$ such that (u,v) is in E_1 if and only if $(f(u),f(v))$ is in E_2. The *digraph isomorphism problem* is that of determining, for a pair of digraphs, whether they are isomorphic.

Theorem 2.9. The digraph isomorphism problem is polynomially equivalent to the problem of deciding whether the productions of two context-free grammars G_1 and G_2 can be labeled so that $Sz(G_1) = Sz(G_2)$.

Proof. We shall first show that the digraph isomorphism problem is polynomially equivalent to the problem of deciding whether two regular grammars are isomorphic.

Let G_1 and G_2 be isomorphic regular grammars. From G_i, $i = 1, 2$, construct a digraph $H_i = (V_i,E_i)$ as follows. H_i has vertex for each nonterminal in G_i and one additional vertex W. If $A \to aB$ is a production in G_i, take (A,B) to E_i, and if $A \to a$ is a production, then take (A,W) to E_i. It is obvious that digraphs H_1 and H_2 are isomorphic.

Suppose now that H_1 and H_2 are isomorphic digraphs. The regular

grammar $G_i = (N_i,T_i,P_i,S_i)$, $i = 1, 2$, corresponding to digraph $H_i = (V_i,E_i)$ has $N_i = V_i \cup \{S_i\}$ and $T_i = \{a_i\}$. For each arc (A,B) in E_i, grammar G_i has a production $A \to a_i B$. Moreover, for each vertex A in V_i, grammar G_i has productions $S_i \to a_i A$ and $A \to a_i$. Grammars G_1 and G_2 are isomorphic.

On the other hand, grammar isomorphism implies the equivalence of Szilard languages, if the productions are labeled according to the homomorphism required in definition 2.1. Hence, it suffices to show that the problem of deciding whether two context-free grammars are isomorphic (where productions are not beforehand labeled) is polynomially reducible to the regular grammar version of the same problem.

Let $G = (N,T,P,S)$ be a context-free grammar. Suppose G has k productions and let $m = \max_{B \in N} \{ n \mid A \to \alpha \in P \text{ and } B(\alpha) = n \}$. The case $m = 0$ is trivial, and we suppose that $m > 0$. Define a regular grammar $R = (N',T',P',S')$ by

$$N' = N \cup \{ A_o \mid A \in N \} \cup \{ B_i \mid i = 1,\ldots,k \} \cup \{ S' \},$$

$$T' = \{ a, b, d, e \} \cup \{ c_j \mid j = 1,\ldots,m \}$$

and

$$\begin{aligned} P' = \{ & S' \to aB_i \mid i = 1,\ldots,k \} \cup \\ & \{ B_i \to bA_o \mid A \text{ is the left-hand side of production } i \} \cup \\ & \{ B_i \to c_j A \mid A \text{ is the } j\text{-th occurrence of the nonterminal} \\ & \quad \text{in question in the right-hand side of production } i \} \cup \\ & \{ A \to d \mid A \in N \} \cup \{ A_o \to e \mid A \in N \}. \end{aligned}$$

Let G_1 and G_2 be context-free grammars and let R_1 and R_2 be regular grammars obtained by the above method from G_1 and G_2, respectively. It is clear that G_1 and G_2 are isomorphic if and only if R_1 and R_2 are isomorphic. □

As an example of isomorphic context-free grammars consider now a syntax-directed translation schema [AU]. A syntax-directed translation schema is a 5-tuple $Y = (N,T_{in},T_{out},P,S)$, where N is the alphabet of nonterminals, T_{in} is the alphabet of input terminals, T_{out} is the alphabet of output terminals, P is the set of productions of the form $A \to \alpha,\beta$, where $\alpha \in (N \cup T_{in})^*$ and $\beta \in (N \cup T_{out})^*$ and the nonterminals in β are a permutation of the nonterminals in α, and finally, S is the start symbol. The translation $T(Y)$ defined by Y is the set of pairs (x,y), where $x \in T_{in}^*$ and $y \in T_{out}^*$, obtained by starting from the pair (S,S) and consequently applying the production in P (for further details, see [AU]).

By the definition above, it is clear that context-free grammars $G_{in} = (N,T_{in},P_{in},S)$ and $G_{out} = (N,T_{out},P_{out},S)$, where $P_{in} = \{ A \to \alpha \mid A \to \alpha,\beta \in P \}$ and $P_{out} = \{ A \to \beta \mid A \to \alpha,\beta \in P \}$, are isomorphic. If for every production $A \to \alpha,\beta$ in P we label $A \to \alpha$ in P_{in} and $A \to \alpha$ in P_{out} with the same label, we have $Sz(G_{in}) = Sz(G_{out})$.

The inclusion problem is undecidable for context-free languages. The inclusion problem for Szilard languages is studied in [KM], where it is shown that the problem is decidable. In [KO], it is studied whether or not it is decidable if $Szl(G_1) \subseteq Sz(G_2)$ holds for arbitrary context-free grammars G_1 and G_2. The answer is affirmative.

The use of these inclusion properties in connection with syntax-directed translation schemata is discussed in [KM,KO].

2.3.2. The emptiness of intersection problem

It is well known that the emptiness problem for the intersection of two context-free languages is undecidable, while the emptiness problem for a language $L \cap R$, where L is context-free and R is regular, is decidable.

Next, we consider similar questions for Szilard languages. The following theorem is proved in [C-RM] by showing that both emptiness problems are polynomially equivalent to the reachability problem for Petri nets [C-RM]. We reduce the emptiness problems directly to each others.

Theorem 2.10 [C-RM]. The emptiness problem for the intersection $Sz(G_1) \cap Sz(G_2)$, where G_1 and G_2 are context-free grammars, is polynomially equivalent to the emptiness problem for the intersection $Sz(G) \cap R$, where G is a context-free grammar and R is a regular language.

Proof. Let $G_1 = (N_1,T_1,P_1,S_1)$ and $G_2 = (N_2,T_2,P_2,S_2)$ be context-free grammars with label alphabets C_1 and C_2, respectively. Define sets P_1', $\tilde{N}_2$, $\tilde{C}_2$ and $\tilde{P}_2$ as follows:

$$P_1' = \{ A \to \rho A_1 \ldots A_n \mid \rho : A \to \alpha \text{ is in } P_1 \text{ and } \eta(\alpha) = A_1 \ldots A_n \},$$

$$\tilde{N}_2 = \{ \tilde{A} \mid A \in N_2 \},$$

$$\tilde{C}_2 = \{ \tilde{\rho} \mid \rho \in C_2 \},$$

$$\tilde{P}_2 = \{ \tilde{A} \to \tilde{\rho}\tilde{A}_1 \ldots \tilde{A}_n \mid \rho : A \to \alpha \text{ is in } P_2 \text{ such that } \eta(\alpha) = A_1 \ldots A_n \}.$$

Furthermore, define a permutative grammar $H = (N,T,P,S)$ by

$N = N_1 \cup \tilde{N}_2 \cup \{ S \}$ (S is a new symbol),

$T = C_1 \cup \tilde{C}_2 \cup \{ \pounds \}$ ($\pounds$ is a new symbol) and

$P = \{ S \to \varepsilon S_1 \tilde{S}_2 \} \cup P_1' \cup \tilde{P}_2 \cup P_{per}$,

where $P_{per} = \{ AB \to BA \mid A, B \ (A \neq B)$ are in $N_1 \cup \tilde{N}_2 \}$.

Since H has the form introduced in subsection 2.1.2, it generates the Szilard language of some context-free grammar G. If $Sz(G_1) \cap Sz(G_2)$ is non-empty, then $L(H)$ contains at least one word common with $R = \{ \varepsilon(a\tilde{b})^+ \mid a \in C_1, \tilde{b} \in \tilde{C}_2, a = b \}$. That is, $Sz(G_1) \cap Sz(G_2) = \Phi$ if and only if $Sz(G) \cap R = \Phi$. This proves that the emptiness problem for $Sz(G_1) \cap Sz(G_2)$ is polynomially reducible to the emptiness problem for $Sz(G) \cap R$.

Let $G = (N,T,P,S)$ be a context-free grammar with label alphabet C and let $G_r = (N_r,T_r,P_r,S_r)$ be a regular grammar such that $L(G_r) = R$. Our problem is that of finding two context-free grammars G_1 and G_2 such that $Sz(G_1) \cap Sz(G_2) = \Phi$ if and only if $Sz(G) \cap R = \Phi$.

Let $T_r = \{ a_1,\ldots, a_n \}$ and for each a_i in T_r let $P_{a_i} = \{ A \to a_iB \mid A \to a_iB$ is in P_r, $B \in N_r \cup \{\lambda\} \}$. If every P_{a_i} has at most one production, then R is the Szilard language of some context-free grammar (having rank 0) and hence, our proof is complete. Otherwise do the following. Suppose P_{a_i}, $i = 1,\ldots,n$, has j_i productions. Let $T_r' = \{ a_{i,k} \mid 1 \leq i \leq n, 1 \leq k \leq j_i \}$ and let P_r' be a new set of productions such that it contains the productions of P_r with their right-hand sides uniquely indexed by the symbols of T_r'. Regular language $G_r' = (N_r,T_r',P_r',S_r)$ generates the Szilard language of some context-free grammar G_1.

Suppose $G_{sz} = (N,C,P'UP",S)$ is the permutative grammar defined

as in subsection 2.1.2 such that $L(G_{sz}) = Sz(G)$ and P' is the set of context-free productions and P'' is the set of permutation productions. Let $P^{\#} = \{ A \to a_{i,k}\alpha \mid 1 \le i \le n, A \to a_i\alpha$ is in $P', 1 \le k \le j_i \}$. Grammar $G'_{sz} = (N, T'_r, P''UP^{\#}, S)$ generates the Szilard language of some context-free grammar G_2. We have $Sz(G_1) \cap Sz(G_2) = \Phi$ if and only if $Sz(G) \cap R = \Phi$. This completes the proof. □

As mentioned above, the both emptiness problems discussed are polynomially equivalent to the reachability problem for Petri nets [C-RM]. This reachability problem is known to be decidable [Kos]. Hence, the emptiness problems are decidable, too.

In theorem 2.7 we proved that the problem "$Sz(G) \subseteq R$?" is decidable in the special case where R is bounded. Since the emptiness problem for $Sz(G) \cap R$ is decidable and $Sz(G) \subseteq \overline{R}$ if and only if $Sz(G) \cap R = \Phi$, "$Sz(G) \subseteq R$?" must also be decidable.

2.4. Context-free languages vs. Szilard languages

Context-free languages are characterized in many different ways in the literature. The most typical characterizations are the (classical) pumping lemma, some of its modifications [Ogd, BM], and Parikh's theorem [Har]. The sufficiency of these characterizations is widely discussed in the literature (see e.g. [Wis]).

Theorem 2.2 gives the necessary and sufficient conditions for a Szilard language to be context-free. However, it is quite natural to expect that also non-context-free Szilard languages share some

properties characteristic to context-free languages. The purpose of this section is to study to what extent (insufficient) characterizations for context-free languages are applicable also to (non-context-free) Szilard languages.

The material of this section is from [Mäk84a].

2.4.1. Pumping

We start with the definition of a well-known property of context-free languages.

Definition 2.2 (Classical pumping property). A language is said to have the *classical pumping property*, if there are natural numbers p and q such that every word z in L, which satisfies lg(z) > p, can be written as z = uvwxy where

1) lg(vwx) ≤ q

2) lg(vx) > 0

and

3) $uv^iwx^iy \in L$, for each $i \geq 0$.

The class of non-context-free languages satisfying the classical pumping property is studied e.g. in [Hor]. The next theorem shows that there are no Szilard languages in that class.

Theorem 2.11. There are no non-context-free Szilard languages satisfying the classical pumping property.

Proof. We consider here only Szilard languages of context-free grammars with rank 2. It is obvious that if the theorem holds in this case, it must hold for higher ranks, too.

Let $G = (N,T,P,S)$ be a context-free grammar with label alphabet C and suppose that G has rank 2. By definition, for each $k \geq 0$ and some nonterminals A and B, G has sentential forms ϕ such that $A(\phi) > k$ and $B(\phi) > k$. For producing such sentential forms, grammar G must have either

(i) a derivation $D \Rightarrow^{\rho} u_1Au_2Bu_3Du_4$, $u_i \in T^*$, $i = 1,\ldots, 4$,

or

(ii) derivations $D \Rightarrow^{\rho_1} v_1Av_2Dv_3$ and $E \Rightarrow^{\rho_2} w_1Bw_2Ew_3$, where $v_j, w_j \in T^*$, $j = 1, 2, 3$.

Suppose now that $Sz(G)$ has the classical pumping property and that the case (i) holds. G must have derivations $S \Rightarrow^{\phi} x_1Dx_2$, $A \Rightarrow^{\sigma} y_1$, $B \Rightarrow^{\tau} y_2$ and $D \Rightarrow^{\nu} z$, where $x_1, x_2, y_1, y_2, z \in T^*$ and ϕ, σ, τ and ν do not contain any recursive subderivations. The Szilard language $Sz(G)$ contains arbitrarily long words without any other recursive subderivations than ρ. We next show that some of these words must contradict the classical pumping property.

Language $Sz(G)$ contains words $\chi_k = \phi\rho^k\sigma^k\tau^k\nu$, $k \geq 0$. We can choose k such that $lg(\chi_k) > p$ where p is the natural number presupposed by definition 2.2. In order to apply the classical pumping property, we must find subwords of χ_k to be repeated. Since ρ is the only subword of χ_k corresponding to a recursive derivation of G, we must repeat ρ. This increases the number of nonterminals A and B, and so we must also repeat subwords σ and τ. Since k can get arbitrarily large values and σ is non-empty, we have arbitrarily long strings between the subwords to be repeated. Hence, there cannot exist the natural number q presupposed by definition 2.2.

Suppose then that the case (ii) holds. G now has derivation $S \Rightarrow^{\phi} x_1Dx_2Ex_3$, where $x_1, x_2, x_3 \in T^*$, or derivations $S \Rightarrow^{\phi'} x_1Dx_2$ and $D \Rightarrow^{\phi''} x_1'Ex_2'$, where $x_1, x_2, x_1', x_2' \in T^*$. In both cases G also has derivation $E \Rightarrow^{\psi} z'$, $z' \in T^*$. Sz(G) contains words $\chi_{m,n} = \phi\rho_1^m\rho_2^n\sigma^m\tau^n\nu\psi$ or $\chi'_{m,n} = \phi'\rho_1^m\phi''\rho_2^n\sigma^m\tau^n\nu\psi$, where $m, n > 0$. We can choose m and n such that $lg(\chi_{m,n}) > p$ and $lg(\chi'_{m,n}) > p$ where p is as above. We must repeat subwords ρ_1 and σ or subwords ρ_2 and τ, but we have strings ρ_2^n (resp. $\phi''\rho_2^n$) or σ^m between the subwords to be repeated. These strings can be arbitrarily long. Hence, as above, we cannot choose the natural number q of definition 2.2. □

The result of theorem 2.11 makes it unnecessary to study the stronger versions of context-free pumping, i.e. Ogden's lemma [Ogd] (Iteration theorem of [Har]) and its generalization [BM]. Instead, we must go to the opposite direction and relieve the conditions of definition 2.2. We use the "generalized pumping" of [Klø].

Definition 2.3 (Generalized pumping property). A language L is said to have the *generalized pumping property of degree* k, $k \geq 1$, if there is a natural number p such that every word z in L, which satisfies $lg(z) \geq p$, can be written as

$z = u_1v_1u_2v_2\ldots u_kv_ku_{k+1}$, where

1) $v_1v_2\ldots v_k \neq \lambda$

and

2) $u_1v_1^iu_2v_2^i\ldots u_kv_k^iu_{k+1} \in L$, for each $i \geq 0$.

Hence, a language has the classical pumping property, if it has the generalized pumping property of degree 2 and condition 1) of

definition 2.2 holds. The proof of theorem 2.11 was based on the fact that it is impossible to fulfil this condition in non-context-free Szilard languages.

Example 2.4. Let G_1 be a context-free grammar with productions

$\pi: S \to ABS$

$\rho: A \to \lambda$

$\sigma: B \to \lambda$

$\tau: S \to \lambda$.

G_1 has rank 2. The Szilard language $Sz(G_1)$ is not context-free, but it has the generalized pumping property of degree 2. This can seen as follows. Let ϕ be in $Sz(G_1)$ such that $lg(\phi) > 4$. Word ϕ begins with a string π^n, $n \geq 1$. We have three different cases depending on the first symbol of ϕ different from π. Suppose first, that this symbol is ρ (the case with symbol σ is analogous). Hence, we can write $\phi = \pi^n \rho \phi_1$, where $\phi_1 = \phi_1' \sigma \phi_1''$, $\phi_1', \phi_1'' \in \{ \pi, \rho, \sigma, \tau \}^*$ (respectively $\phi = \pi^n \sigma \phi_2$, where $\phi_2 = \phi_2' \rho \phi_2''$ $\phi_1', \phi_2'' \in \{ \pi, \rho, \sigma, \tau \}^*$). We can choose $u_1 = \pi^{n-1}$, $v_1 = \pi\rho$, $u_2 = \phi_1'$, $v_2 = \sigma$ and $u_3 = \phi_1''$ (resp. $v_1 = \pi\sigma$, $u_2 = \phi_2'$, $v_2 = \rho$ and $u_3 = \phi_2''$.) $Sz(G_1)$ contains all words $\pi^{n-1}(\pi\rho)^i \phi_1'(\sigma)^i \phi_1''$ (resp. $\pi^{n-1}(\pi\sigma)^i \phi_2'(\rho)^i \phi_2''$), $i \geq 0$.

Suppose now that $\phi = \pi^n \tau \phi'$, where $n \geq 2$ and $\phi' \in \{ \rho, \sigma \}^{2n}$. We can write $\phi' = \phi_1 \phi_2 \phi_3$, where $\phi_2 = \rho\sigma$ or $\phi_2 = \sigma\rho$ and $\phi_1, \phi_3 \in \{ \rho, \sigma \}^*$, and further, $u_1 = \pi^{n-1}$, $v_1 = \pi$, $u_2 = \tau\phi_1$, $v_2 = \phi_2$, and $u_3 = \phi_3$. $Sz(G_1)$ contains all words $\pi^{n-i}(\pi)^i \tau \phi_1 (\phi_2)^i \phi_3$, $i \geq 0$. □

Example 2.5. Let G_2 be a context-free grammar with productions

$\pi: S \to ABS$

$\rho: A \to \lambda$

$\sigma: B \to \lambda$

$\tau: S \to C$

$\nu: C \to \lambda$.

G_2 has rank 2, but its Szilard language $Sz(G_2)$ does not have the generalized pumping property of degree 2. $Sz(G_2)$ contains words $\pi^n\tau\rho^n\nu\sigma^n$, for each $n \geq 1$. These words cannot be written in the form $u_1v_1u_2v_2u_3$. □

Example 2.5 shows that for each k, $k \geq 1$, there are Szilard languages of context-free grammars with rank k such that they do not have the generalized pumping property of degree k.

2.4.2. Sokolowski's criterion

The following property is necessary for a language to be context-free.

Definition 2.4 (Sokolowski's criterion) [Sok]. A language L ($\subseteq \Sigma^*$) is said to satisfy *Sokolowski's criterion*, if for every subset Σ' of Σ containing at least two distinct symbols, and for all words $u, v, w \in \Sigma^*$, if $\{ uxvxw \mid x \in \Sigma'^+ \} \subseteq L$ then there exist two different words x' and x'' such that $ux'vx''w$ is in L.

The sufficiency of this criterion for context-freeness is discussed in [Nij82]. It is easy to see that every Szilard language

satisfies Sokolowski's criterion. Moreover, we can strengthen the criterion in the case of Szilard languages as follows.

Theorem 2.12. Let $Sz(G)$ $(\subseteq C^+)$ be the Szilard language of a context-free grammar $G = (N,T,P,S)$. For every non-empty subset C' of C and for all strings $\rho, \sigma, \tau \in C^*$, if $\{ \rho\pi\sigma\pi\tau \mid \pi \in C'^+ \} \subseteq Sz(G)$ and strings π' and π'' are in $K(\rho,\sigma,\tau) = \{ \pi \in C'^+ \mid \rho\pi\sigma\pi\tau \in Sz(G) \}$, then strings $\rho\pi'\sigma\pi''\tau$ and $\rho\pi''\sigma\pi'\tau$ are in $Sz(G)$.

Proof. Let $\rho\pi\sigma\pi\tau$ be in $Sz(G)$. The prefix ρ of $\rho\pi\sigma\pi\tau$ corresponds to a derivation $S \Rightarrow^{\rho} \alpha$, $\alpha \in (N \cup T)^+$, the suffix τ corresponds to a derivation $\beta \Rightarrow^{\tau} w$, where $S \Rightarrow^{\rho\pi\sigma\pi} \beta$ and $w \in T^*$, and the infix σ corresponds to a derivation $\alpha' \Rightarrow^{\sigma} \beta'$, where $S \Rightarrow^{\rho\pi} \alpha'$ and $S \Rightarrow^{\rho\pi\sigma} \beta'$. Hence, all π's in $K(\rho,\sigma,\tau)$ must correspond to derivations $\alpha \Rightarrow^{\pi} \alpha'$ and $\beta' \Rightarrow^{\pi} \beta$. Since π can be any string from C'^+, each symbol in C' must be associated with a production of P having the form $A \to uAv$, where $u, v \in T^*$. Hence, both π's in $\rho\pi\sigma\pi\tau$ can be replaced by any strings from $K(\rho,\sigma,\tau)$. □

Let $L = \{ ba^nca^nc \mid n \geq 0 \}$. L is a typical example of the languages which do not satisfy the property of theorem 2.12. By replacing equal numbers of a's by arbitrary strings of a's, we get the language $\{ ba^*ca^*c \}$, which satisfies the property and is a Szilard language. Naturally, the property of theorem 2.12 is not sufficient for a language to be a Szilard language.

A stronger form of Sokolowski's criterion is given in [Gra]. This extension does not hold for Szilard languages.

2.4.3. Parikh mapping

Let $\Sigma = \{ a_1, a_2, \ldots, a_n \}$ and let L $(\subseteq \Sigma^*)$ be a language. *Parikh mapping* ψ is a function from Σ^* to $\mathbb{N}^n$ defined by $\psi(w) = (a_1(w), a_2(w), \ldots, a_n(w))$. Let $\psi(L) = \{ \psi(w) \mid w \in L \}$. Languages L_1, L_2 $(\subseteq \Sigma^*)$ are said to be *Parikh equivalent* if $\psi(L_1) = \psi(L_2)$. A set of the form $\{ \alpha_0 + n_1\alpha_1 + \ldots + n_m\alpha_m \mid n_j \geq 0, j = 1, \ldots, n \}$, where $\alpha_0, \alpha_1, \ldots, \alpha_m$ are elements of $\mathbb{N}^n$, is said to be a *linear subset* of $\mathbb{N}^n$.

A *semilinear set* is a finite union of linear sets. A language L is *semilinear* if $\psi(L)$ is a semilinear set. Context-free languages are semilinear [Har]. A language L is semilinear if and only if L is Parikh-equivalent to a regular language [Har].

Lemma 2.2. Let G be a context-free grammar. The languages $Sz(G)$ are $Szl(G)$ are Parikh-equivalent.

Proof. The only difference between leftmost and arbitrary derivations of a context-free grammar is in the order in which productions are applied.□

It is now easy to prove the following

Theorem 2.13 [Höp]. Szilard languages are semilinear.

Proof. Let G be a context-free grammar. $Szl(G)$ is context-free and hence, there is a regular language R such that $\psi(Szl(G)) = \psi(R)$. By lemma 2.2, $\psi(Szl(G)) = \psi(Sz(G))$. Hence, $Sz(G)$ is Parikh-equivalent to a regular language, and equivalently, $Sz(G)$ is semilinear. □

A language L is *Parikh-bounded* if it contains a bounded language B such that $\psi(L) = \psi(B)$ [BL]. Context-free languages are Parikh-bounded [BL]. We can show that Szilard languages have the same property.

Theorem 2.14. Szilard languages are Parikh-bounded.

Proof. Let G be a context-free grammar. Since Szl(G) is a context-free language, it is Parikh-bounded and contains a bounded language B such that $\psi(Szl(G)) = \psi(B)$. Again, by lemma 2.2, $\psi(Szl(G)) = \psi(Sz(G))$ and hence, $\psi(Sz(G)) = \psi(B)$. Moreover $B \subseteq Szl(G) \subseteq Sz(G)$.□

Hence, semilinearity and Parikh boundedness are not strong enough to distinguish between context-free languages and non-context-free Szilard languages.

CHAPTER 3: ON LEFT SZILARD LANGUAGES

In this chapter we restrict ourselves to the leftmost derivations of context-free grammars. While a Szilard language can be non-context-free, the left Szilard language of a context-free grammar always belongs to a subclass of the class of s-languages. Hence, when we study grammars generating left Szilard languages (section 3.1), automata recognizing them (section 3.2) and decision problems for this class of languages (section 3.3), we are dealing with quite restricted special cases of the same questions for deterministic context-free grammars and languages.

By definition, we have $Szl(G) \subseteq Sz(G)$ for all context-free grammars G. Hence, left Szilard languages are prefix-free.

Left Szilard languages are closed under none of the operations mentioned in chapter 2 in connection with Szilard languages. We consider intersection with regular languages as an example.

Example 3.1 [Mäk83a]. Let G_1 and G_2 be context-free grammars with the following productions

G_1:	G_2:
$\pi: S \to AAA$	$\pi: S \to A$
$\rho: A \to A$	$\rho: A \to AA$
$\sigma: A \to \lambda$	$\sigma: A \to \lambda$.

We have $Szl(G_1) = \{ \pi\rho^i\sigma\rho^j\sigma\rho^k\sigma \mid i, j, k \geq 0 \}$, which is a

regular language. The intersection $L = Szl(G_1) \cap Szl(G_2)$ consists of those words in $Szl(G_1)$ which have exactly three occurrences of symbol σ. This means $L = \{ \pi\rho\sigma\rho\sigma\sigma, \pi\rho\rho\sigma\sigma\sigma \}$, which is not a left Szilard language. □

Example 3.1 shows that left Szilard languages are closed neither under intersection with regular languages nor under intersection.

3.1. Ss-grammars

A context-free grammar $G = (N,T,P,S)$ is said to be an *ss-grammar* if for each production $A \to \alpha$ in P, α is in TN^*, and for each pair $A \to a\alpha$ and $B \to b\beta$ in P, we have $a \neq b$. Hence, each production has a terminal symbol, which uniquely identifies the production. Recall that in an s-grammar, it is possible to have $A \to a\alpha$ and $B \to a\beta$, where $A \neq B$. There are trivial examples of s-languages which cannot be generated by any ss-grammar (f.ex. { aa }).

The following theorem shows the well-known connection between ss-grammars and left Szilard languages.

Theorem 3.1. The left Szilard language of each context-free grammar can be generated by an ss-grammar. Every ss-grammar generates the left Szilard language of some context-free grammar.

Proof. Let $G = (N,T,P,S)$ be a context-free grammar with label alphabet C. Define an ss-grammar $G' = (N',T',P',S')$ as follows. Let $N' = N$, $T' = C$ and $S' = S$ and for every $\rho: A \to \alpha$ in P, take $A \to \rho\eta(\alpha)$ to P',

Let $S \Rightarrow_l^{\pi} u\alpha$, where $u \in T^*$ and $\alpha \in (N \cup T)^*$, be a leftmost derivation in G. By induction on the length of π, it is straightforward to show that grammar G' has derivation $S \Rightarrow_l^* \pi\eta(\alpha)$. Especially, when $S \Rightarrow_l^{\pi} w$, $w \in T^*$, is in G we have $S' \Rightarrow_l^* \pi$ in G'.

Similarly, we can show that for every $S' \Rightarrow_l \pi$, $\pi \in C^*$, in G' we have $S \Rightarrow_l^{\pi} v$ in G, where v is in T^*. Hence, $L(G') = Szl(G)$.

Consider an ss-grammar $G = (N,T,P,S)$. Let T be the label alphabet of G. Label every production $A \to a\alpha$ in P by a. Now $L(G) = Szl(G)$ holds. □

The proof of theorem 3.1 contains a method to define an ss-grammar which generates the left Szilard language of a given context-free grammar. This method is originally presented in [AB]. In the sequel, it is called the method of theorem 3.1.

Next, we shall study a situation similar to theorem 2.1.

Definition 3.1 [Wal]. A context-free grammar $G = (N,T,P,S)$ is said to be *left-derivation bounded* if there is a natural number k such that for every leftmost derivation $S \Rightarrow_l \alpha$, where $\alpha \in (N \cup T)^*$, we have $lg(\eta(\alpha)) \leq k$.

Let G be left-derivation bounded and define the ss-grammar G' that generates Szl(G) as in the proof of theorem 3.1. Then G' is left-derivation bounded, too. We can "simulate" the leftmost derivations of G' by using a regular grammar. Hence, if G is left-derivation bounded, then Szl(G) is regular. Conversely,

if $Szl(G)$ is regular, then G must be left-derivation bounded.

A necessary and sufficient condition for a context-free grammar $G = (N,T,P,S)$ to be left-derivation bounded is given in [Wal]. This condition requires that every recursive derivation $A \Rightarrow_l^* \alpha_1 A \alpha_2$ must have α_2 in T^*.

As an example, consider grammars G_1 and G_2 with the following productions

G_1:	G_2:
$\pi: S \to AS$	$\pi: S \to SA$
$\rho: A \to \lambda$	$\rho: A \to \lambda$
$\sigma: S \to \lambda$	$\sigma: S \to \lambda$.

$Szl(G_1)$ can be denoted by $(\pi\rho)^*\sigma$, while $Szl(G_2) = \{ \pi^k \sigma \rho^k \mid k > 0 \}$ which is a non-regular language.

However, there are regular left Szilard languages which cannot be generated by any regular ss-grammar.

Theorem 3.2 [Mäk83a]. All regular left Szilard languages cannot be generated by regular ss-grammars.

Proof. Consider regular left Szilard language $L = \{ abcb \}$. Every regular grammar generating L must have productions of the form $A \to bB$ and $C \to b$. □

The following example is related to language L in the proof above.

Example 3.2 [Mäk83a]. Let G be a left-derivation bounded ss-grammar such that k is the fixed upper bound required in

theorem 3.1. Then G cannot generate language $L_k = \{ ab^kcb \}$. Similarly, if G is left-derivation bounded then it cannot generate language $\{ ab^*cb \}$. □

A variant of ss-grammars is introduced in [Yaf], where a context-free grammar $G = (N,T,P,S)$ is said to be *left-structural* with respect to a subalphabet Σ ($\subseteq T$) if

(i) each production in P has the form $A \to a\alpha$, where a is in Σ and α in $(N \cup (T \setminus \Sigma))^*$

and

(ii) for any two distinct productions $A \to a\alpha$ and $B \to b\beta$, we have $a \neq b$.

Every ss-grammar is left-structural with respect to its terminal alphabet. On the other hand, in any production of a left-structural grammar G, we can replace each terminal c of $(T \setminus \Sigma)$ with a new nonterminal A_c, and add $A_c \to c$ to the set of productions. This does not change the language L(G). Hence, left-structural grammars generate the class of left Szilard languages.

3.2. Recognition of left Szilard languages

A deterministic pushdown automaton is said to be a *simple machine*, if it operates in realtime and has only one state. The class of languages recognizable by simple machines and the class of s-languages coincide [Har]. A simple machine that recognizes a left Szilard language has the following additional property.

Theorem 3.3 [Pen74]. A language L is a left Szilard language if and only if L can be recognized by a simple machine

$A = (\{q\},\Sigma,\Gamma,\delta,q,Z_0,\{q\})$ in which conditions

$(q,a,X) = (q,\alpha)$ and $(q,a,Y) = (q,\beta)$

always implies $X = Y$ and $\alpha = \beta$.

Proof. Suppose first that L is the left Szilard language of a context-free grammar G' and let $G = (N,T,P,S)$ be an ss-grammar such that $L(G) = L$.

Let $\Sigma = T$, $\Gamma = N$ and $Z_0 = S$ and for every production $A \to a\alpha$ in P define $\delta(q,a,A) = (q,\alpha)$. Simple machine $A = (\{q\},\Sigma,\Gamma,\delta,q,Z_0,\{q\})$ has the required property, and it clearly recognizes L.

It is now obvious how an ss-grammar can be defined when a simple machine has the additional property. □

Hence, every left Szilard language can be recognized by a simple machine in which input symbols always uniquely determine the topmost element of the pushdown store and the change made to the store contents.

Next, we shall define the concept of a superdeterministic pushdown automaton. In [GF] this was done for an arbitrary deterministic pushdown automaton. For our purposes it is sufficient to define superdeterminism for simple machines only.

Definition 3.2 [GF]. A simple machine is *superdeterministic* if reading an input string w always causes the same change in the height of the pushdown store (i.e. the change does not depend on the actual store contents.) A language L is *superdeterministic* if there is a superdeterministic pushdown automaton which accepts L.

The simple automaton introduced in the proof of theorem 3.3 clearly is superdeterministic. Hence, we obtain

Theorem 3.4. Every left Szilard language is superdeterministic.

It is observed in [GF] that each language generated by a left-structural grammar is superdeterministic. By our earlier remark concerning left-structural grammars and left Szilard languages, this is equivalent to theorem 3.4.

The class of left Szilard languages can also be recognized by many other types of pushdown automata which are studied in connection with the equivalence problem for deterministic context-free languages. For example, it is shown in [Lin] that every left Szilard language is "nonsingular".

Like Szilard languages, left Szilard languages are recognizable in space $\log n$ by a deterministic Turing machine [Pen77]. However, the corresponding counter machine construction is more complicated for left Szilard languages than for Szilard languages. For example, the counter automaton introduced in [Pen77] works in time n^3.

3.3. Decision problems for left Szilard languages

In this section we give without proofs some known decision results for certain subclasses of context-free grammars and languages. As special cases we obtain corresponding results for left Szilard languages.

The equivalence problem for s-languages is decidable [Har]. Hence, it is decidable for left Szilard languages, too. It is shown in [HRS76b] that there exists a deterministic polynomial time algorithm for the case where the languages are generated by linear s-grammars. We have not been able to find a deterministic polynomial time algorithm for ss-grammars.

The following lemma will be used in chapter 5.

Lemma 3.1 [GF]. Let G be a context-free grammar and let L be a superdeterministic language. Then there is an algorithm to determine whether or not $L(G) \subseteq L$ holds.

Lemma 3.1 also implies that the inclusion problem is decidable for left Szilard languages; this is also proved in [Lin].

We cannot replace "$\subseteq$" by "=" in lemma 3.1. Namely, "$L_0 = L(G)?$" is decidable for an arbitrary context-free grammar G and a fixed context-free language L_0 if and only if L_0 is bounded [HR]. The simplest example of unbounded left Szilard languages is the language $\{ \pi,\rho \}^*\sigma$.

However, "$L(G) = L?$" is decidable if G is a deterministic context-free grammar and L is a left Szilard language [GF]. (This follows also from [TK], since left Szilard languages are "nonsingular" [Lin]).

3.4. On boundedness of left Szilard languages

In this section we study situations where the language $L(G)$ generated by a context-free grammar or the left Szilard language $Szl(G)$ is

bounded. We are able to prove that if G is unambiguous, then L(G) is bounded if and only if Szl(G) is bounded. Based on this result, we give in subsection 3.4.2 a deterministic polynomial time algorithm which determines whether or not an unambiguous context-free grammar generates a bounded language.

3.4.1 On bounded L(G)'s and Szl(G)'s

We start with some definitions and an important lemma.

Let $G = (N,T,P,S)$ be a context-free grammar. A leftmost derivation $A \Rightarrow_l \alpha_1 \Rightarrow_l \ldots \Rightarrow_l \alpha_n \Rightarrow_l uA\alpha$, where u is in T^* and α is in $(N \cup T)^*$, is *minimal recursive* if A is not the leftmost nonterminal in any word α_i, $i = 1,\ldots, n$.

A language L is *commutative*, if $uv = vu$ holds for all u and v in L.

Lemma 3.2 [Gin]. Let $G = (N,T,P,S)$ be a context-free grammar. The language $L(G)$ is bounded if and only if for each nonterminal A in N the languages

$$Lf(A) = \{ u \mid A \Rightarrow^* uA\alpha,\ u \in T^* \}$$

and

$$Rg(A) = \{ v \mid A \Rightarrow^* \beta Av,\ v \in T^* \}$$

are commutative.

The following theorem is a direct application of lemma 3.2.

Theorem 3.5 [Mäk83a]. Let $G = (N,T,P,S)$ be an ss-grammar. The language $L(G)$ is bounded if and only if the following conditions

hold

1) for every A in N there is at most one minimal recursive derivation $A \Rightarrow_l^* uA\alpha$

and

2) if the word α above is not equal to λ, there is exactly one derivation $\alpha \Rightarrow_l^* v$, where v is in T^*.

Proof. Suppose first that $L(G)$ is bounded. If there are two minimal recursive derivations $A \Rightarrow_l^* u_1 A \alpha_1$ and $A \Rightarrow_l^* u_2 A \alpha_2$, we must have $u_1 \neq u_2$. But, by lemma 3.2, $u_1 u_2 = u_2 u_1$. Hence, u_1 is a prefix of u_2, or vice versa. In both cases, we have a contradiction with the left Szilard property.

If the condition 2) does not hold, we have v_1 and v_2 with $v_1 \neq v_2$ and $v_1 v_2 = v_2 v_1$, and we can derive a contradiction as above.

Conversely, if the conditions 1) and 2) hold for some context-free grammar G, then $L(G)$ clearly is bounded. □

The following remark deals with a class of context-free grammars in which each grammar has an unbounded left Szilard language.

Remark 3.1 [Mäk83a]. A context-free grammar is said to be *expansive* (with respect to leftmost derivations) if there is a nonterminal A such that $A \Rightarrow_l^+ \alpha$, where α contains at least two occurrences of A.

If a context-free grammar G is expansive, then the ss-grammar G' obtained from G by the method of theorem 3.1 is expansive, too. Hence, G' has a minimal recursive derivation of the form $A \Rightarrow_l^* uA\alpha$,

where u is a terminal string and α contains at least one occurrence of A. Now we have infinite number of leftmost derivations (instead of one) from α to terminal strings. □

Theorem 3.6 [Mäk83a]. If an unambiguous context-free grammar G generates a bounded language, then the left Szilard language $Szl(G)$ is bounded, too.

Proof. Let G' be the ss-grammar obtained from $G = (N,T,P,S)$ by the method of theorem 3.1.

Every minimal recursive derivation in G has a corresponding minimal recursive derivation in G', and vice versa. Hence, for the condition 1) of theorem 3.5, it is sufficient to show that G has at most one minimal recursive derivation for any nonterminal A. To derive a contradiction we suppose we have derivations $A \Rightarrow_l^{\pi} uA\alpha$ and $A \Rightarrow_l^{\rho} vA\beta$. We must also have derivations $\alpha \Rightarrow_l^{\sigma} w$, $\beta \Rightarrow_l^{\tau} y$ and $A \Rightarrow_l^{\nu} x$, where w, y, and x are in T^*. Since $L(G)$ is bounded, we have $uv = vu$ and $wy = yw$. Grammar G has derivations

$$A \Rightarrow_l^{\pi} uA\alpha \Rightarrow_l^{\rho} uvA\beta\alpha \Rightarrow_l^{\nu} uvx\beta\alpha \Rightarrow_l^{\tau} uvxy\alpha \Rightarrow_l^{\sigma} uvxyw$$

and

$$A \Rightarrow_l^{\rho} vA\beta \Rightarrow_l^{\pi} vuA\alpha\beta \Rightarrow_l^{\nu} vux\alpha\beta \Rightarrow_l^{\sigma} vuxw\beta \Rightarrow_l^{\tau} uvxwy.$$

$uvxyw = vuxwy$ holds, but $\pi\rho\nu\tau\sigma \neq \rho\pi\nu\sigma\tau$, which contradicts the unambiguous property.

In the same way, we can derive a contradiction by supposing two terminating derivations from α, when $A \Rightarrow_l^{*} uA\alpha$ is in G. Again, the same holds in grammar G'. This completes the proof. □

We cannot strengthen theorem 3.6 to cover all context-free grammars. A counterexample is the ambiguous context-free grammar G with productions S→aS, S→aaS and S→b. In this case L(G) is bounded, but Szl(G) is unbounded.

Similarly, if a context-free grammar is expansive (with respect to leftmost derivations) and generates a bounded language, it can serve as a counterexample. The simplest one of such grammars has productions S→SS and S→a.

By remark 3.1 and theorem 3.6, we also obtain the following little theorem.

Theorem 3.7. Let a context-free grammar G be expansive (with respect to leftmost derivations). If G generates a bounded language, then G is ambiguous.

The converse of theorem 3.6 holds for all context-free grammars.

Theorem 3.8 [Mäk83a]. Let G be a context-free grammar. If Szl(G) is bounded, then L(G) is bounded, too.
Proof. Since Szl(G) is bounded, the ss-grammar G' obtained from G by the method of theorem 3.1 must fulfil conditions 1) and 2) of theorem 3.5. This implies that G must also fulfil these conditions. A context-free grammar fulfilling these conditions generates a bounded language. □

3.4.2. Boundedness testing for unambiguous context-free grammars

It is decidable whether a given context-free language is commuta-

tive [Gin]. Hence, by lemma 3.2, we have an algorithm to determine whether a given context-free grammar generates a bounded language. Since this algorithm is of exponential time complexity, it was posed open in [HRS76b] whether the test can be done in polynomial time. In [HRS76b] the given problem was also positively solved in the case of linear grammars. As shown in [Mäk83b], there is a deterministic polynomial time algorithm for unambiguous grammars. We shall now present a version of this algorithm.

By theorems 3.6 and 3.8, an unambiguous context-free grammar G generates a bounded language if and only if $Szl(G)$ is bounded. We shall test whether $Szl(G)$ fulfils the conditions 1) and 2) of theorem 3.5.

Condition 1) is tested by drawing a digraph, which represents all leftmost derivations starting from a certain nonterminal A. Each circuit corresponds to a minimal recursive derivation. If there is more than one circuit, we know that an unbounded language is generated. Condition 2) is tested by first constructing a set M which contains all nonterminals having unique leftmost derivation to a terminal string. Suppose that for a nonterminal A we have one minimal recursive derivation $A \Rightarrow_l^+ uA\alpha$, where u contains terminals and α contains nonterminals. It is now sufficient to test whether or not all nonterminals of α are in M.

Algorithm

Input: An unambiguous context-free grammar $G = (N,T,P,S)$.

Output: "Yes", if $L(G)$ is bounded, otherwise "No".

Method:

1. Use the method of theorem 3.1 to construct an ss-grammar $G' = (N,C,P',S)$ such that $L(G') = Szl(G)$.
2. Construct a subset M of N as follows.
   2.1. Set initially $i = 0$ and $M_0 = \{ A \mid A \to \rho \in P', \rho \in C,$ and $A \to \rho$ is the only production in P' with the left-hand side $A \}$.
   2.2. Set $i = i + 1$ and $M_i = \{ A \mid A \to a\alpha \in P', \alpha \in M_{i-1}^*,$ and $A \to a\alpha$ is the only production in P' with left-hand side $A \} \cup M_{i-1}$.
   2.3. If $M_i = M_{i-1}$, then set $M = M_i$. Otherwise goto step 2.2.
3. Draw a digraph D with a vertex for each nonterminal in N and an edge from A to B if there is in P' a production with A as the left-hand side and B in the right-hand side.
4. Set all nonterminals in N unmarked.
5. If all nonterminals are marked, then output "Yes" and halt. Otherwise, select an arbitrary unmarked A from N.
6. Mark A and do the following.
   6.1. If vertex A is not contained in any circuit of D, then goto step 5.
   6.2. If in D there is more than one circuit containing vertex A then output "No" and halt.
   6.3. If in D there is exactly one circuit containing vertex A, then find the corresponding minimal recursive derivation $A \Rightarrow_l^* uA\alpha$, where $u \in C^+$ and $\alpha \in N^*$. If each nonterminal B having $B(\alpha) > 0$ is in M, then goto step 5. Otherwise output "No" and halt.

It is clear that the algorithm always eventually halts and checks conditions 1) and 2) of theorem 3.5.

It is evidently possible to construct set M in polynomial time. The same holds for constructing digraph D and finding circuits.

Suppose that digraph D has exactly one circuit containing a vertex A. According to the algorithm, our next task is to find minimal recursive derivation $A \Rightarrow_l^* uA\alpha$. The length of α is bounded by $\mathrm{card}(N)^2 \cdot m$, where $m = \max \{ \lg(\eta(\beta)) \mid A \to \beta \in P' \}$.

Hence, the time needed is a polynomial function of the size $|G|$, $|G| = \sum_{(A\to\alpha)\in P} \lg(A\alpha)$, of the input grammar $G = (N,T,P,S)$.

3.5. An undecidable problem for context-free grammars

Let $G = (N,T,P,S)$ be a context-free grammar in semi-GNF with label alphabet C. Define a homomorphism $g: C \to T^*$ by setting $g(\rho) = w$, if $\rho: A \to w\alpha$, where $w \in T^*$ and $\alpha \in N^*$, is in P. We obviously have $g(Szl(G)) = L(G)$. This property is needed in this section; it will be used in chapter 5, too.

Consider the following problem: Let $G = (N,T,P,S)$ be a context-free grammar and let $h: T \to T_1^*$ be a homomorphism. Are there distinct words v and w in $L(G)$ such that $h(v) = h(w)$? We shall show that this problem is undecidable for arbitrary context-free grammars. Moreover, it is undecidable even for ss-grammars.

A similar (but more general) problem is studied in connection with the homomorphism equivalence problem in [Cul79, Cul80]. It is studied there, among other things, whether a translation defined by a finite or pushdown transducer [AU] is one-to-one or functional on a context-free language. (A translation τ is said to be *functional* if $(x,y) \in \tau$ and $(x,z) \in \tau$ implies $y = z$ [Cul79].) Note that if G and h are as above, homomorphism h defines a translation $T = \{ (w,h(w)) \mid w \in \mathcal{L}(G) \}$. So our problem is that of deciding whether translations such as T are one-to-one.

The following lemma gives a characterization of the ambiguity of a context-free grammar in semi-GNF.

Lemma 3.3 [Mäk85b]. Let $G = (N,T,P,S)$ be a context-free grammar in semi-GNF and let homomorphism g be as above. Then G is ambiguous if and only if the homomorphism g has the property that for at least one pair of distinct words π and ρ in $Szl(G)$, $g(\pi) = g(\rho)$ holds.

Proof. Recall that $g(Szl(G)) = L(G)$ holds. Words π and ρ represent derivations $S \Rightarrow_l^{\pi} v$ and $S \Rightarrow_l^{\rho} w$, where $g(\pi) = v$ and $g(\rho) = w$. If we have $g(\pi) = g(\rho)$, then G is ambiguous. Conversely, if G is ambiguous, we must have words π and ρ (i.e. two leftmost derivations) such that $g(\pi) = g(\rho)$ (i.e. these derivations produce equal terminal words). □

Actually, lemma 3.3 is applicable to all context-free grammars in the following sense. Let G be a context-free grammar not in semi-GNF. Replace every terminal a that offends the semi-GNF property by a new nonterminal A_a and add $A_a \to a$ to the set of productions. The original context-free grammar is unambiguous

if and only if the new grammar is unambiguous.

Theorem 3.9 [Mäk85b]. Let $G = (N,T,P,S)$ be an ss-grammar and let $g:T\to T_1^*$ be a homomorphism. Then it is undecidable whether $L(G)$ contains distinct words v and w such that $g(v) = g(w)$.
Proof. If our problem were decidable then, by lemma 3.3, it would be decidable whether or not a given context-free grammar is ambiguous. Since we know [Har] that this is not the case for the latter problem, our proof is complete. □

A similar result is proved in [Iba]. It is shown there that deciding whether a homomorphism is one-to-one on a deterministic one-counter language is undecidable. Notice that the classes of left Szilard languages and deterministic one-counter languages are incomparable.

Theorem 3.10. Let $G = (N,T,P,S)$ be an ss-grammar generating a bounded language and let $g:T\to T_1^*$ be a homomorphism. Then it is decidable whether or not g is one-to-one on $L(G)$.
Proof. Grammar G and homomorphism g uniquely define a context-free grammar $G_1 = (N,T_1,P_1,S)$, where P_1 contains a production $A\to g(\rho)\alpha$ for each $A\to\rho\alpha$ in P. G_1 is in semi-GNF and its left Szilard language is $Szl(G)$. By theorem 3.8, $L(G_1)$ is bounded. From [Gin] we know that it is decidable whether a context-free grammar generating a bounded language is ambiguous. The theorem now follows by lemma 3.3. □

The existence of inherently ambiguous context-free languages proves the following theorem.

Theorem 3.11. There are context-free languages L ($\subseteq \Sigma^*$) for for which the following conditions

1) $g(L(G)) = L$

2) $G = (N,T,P,S)$ is an ss-grammar

and

3) $g:T\to\Sigma^*$ is a homomorphism

always imply that g is not one-to-one on $L(G)$.

On the other hand, we know some conditions which guarantee that a context-free grammar is unambiguous, i.e. homomorphism g is one-to-one on the left Szilard language. The LL(k) property [AU] is one such condition. It can be given in the following form by using homomorphism g. (In the definition, notation $k:\alpha$, where k is a natural number and α is a string of symbols, means the prefix of α of length k, if $lg(\alpha) \geq k$, and string α itself, if $lg(\alpha) < k$.)

Definition 3.3. Let $G = (N,T,P,S)$ be a context-free grammar in semi-GNF with label alphabet C and let $g:C\to T^*$ be defined as above. G is an LL(k) grammar if for all words $\pi_1 = \rho\sigma_1\tau_1$ and $\pi_2 = \rho\sigma_2\tau_2$, where $\rho, \tau_1, \tau_2 \in C^*$ and $\sigma_1, \sigma_2 \in C$, in $Szl(G)$ the condition $k:g(\sigma_1\tau_1) = k:g(\sigma_2\tau_2)$ always implies $\sigma_1 = \sigma_2$.

As well known, it is decidable whether a given context-free grammar is an LL(k) grammar for a fixed k, but undecidable whether there exists some value of k such that G is an LL(k) grammar [AU].

3.6. On the length of context-free derivations

Let $G = (N,T,P,S)$ be a context-free grammar. Given a word w in $L(G)$, what can be said about the length of derivation π for which we have $S \Rightarrow_l^{\pi} w$? In this section we answer the question in some special cases.

Notice first that by permuting the symbols of π we usually obtain several general (i.e. non-leftmost) derivations for w. Naturally, these derivations are exactly as long as π. Hence, the order in which productions are applied does not make any difference when the length of the derivation is concerned and we could have represented this material already in chapter 2.

We start with simple remarks concerning context-free grammars in certain normal forms. Let $G = (N,T,P,S)$ be a context-free grammar in GNF. If $S \Rightarrow_l^{\pi} w$, $w \in T^+$, then clearly $lg(\pi) = lg(w)$. Similarly, if G is in CNF, we have $lg(\pi) = 2lg(w) - 1$.

For most classes of context-free grammars, only an upper bound for the length of a derivation can be given. As an example, we consider context-free grammars which do not have null productions, i.e. λ-free grammars. In order to avoid difficulties caused by chain derivations of the form $A \Rightarrow_l^+ A$, we consider "chain-free" context-free grammars, only.

Theorem 3.12 [Sip]. Let $G = (N,T,P,S)$ be a λ-free context-free grammar such that $m = card(N)$ and $A \Rightarrow_l^+ A$ is impossible for each nonterminal A in N. If $S \Rightarrow_l^{\pi} w$, $w \in T^*$, then $lg(\pi) \leq 2m \cdot lg(w) - m$. This bound is minimal.

Proof. The proof is an induction on $lg(w)$. If $lg(w) = 1$, then only unit productions of the form $A \to B$ and terminating productions are possible in π. The longest possible sequence of unit productions has length $m - 1$, since each sequence of length m has at least one chain of the form $A \Rightarrow_l^+ A$. Hence, we have $lg(\pi) \leq m = 2m \cdot lg(w) - m$.

Suppose now that $lg(w) > 1$, and as an induction hypothesis that if $lg(w') < lg(w)$, then $S \Rightarrow_l^{\pi'} w'$ implies $lg(\pi') \leq 2m \cdot lg(w') - m$.

Consider derivation $S \Rightarrow_l^{\pi} w$. Let ρ be the possible sequence of unit productions in the beginning of π. As above, $lg(\rho) \leq m - 1$. Let $\sigma: A \to X_1 \ldots X_n$, $n > 1$, be the first non-unit production. We now have strings of productions $\tau_1, \ldots, \tau_n$ and terminal strings $w_1, \ldots, w_n$ such that $X_i \Rightarrow_l^{\tau_i} w_i$, $i = 1, \ldots, n$, and $w_1 \ldots w_n = w$. (Notice, that $\tau_i = \lambda$ if X_i is a terminal symbol.) We can apply the induction hypothesis to w_i's and hence, we have

$$lg(\pi) = lg(\rho\sigma\tau_1 \ldots \tau_n) = lg(\rho) + 1 + \sum_{i=1}^{n} lg(\tau_i) \leq$$
$$(m - 1) + 1 + \sum_{i=1}^{n} (2m \cdot lg(w_i) - m) = 2m \cdot lg(w) + (1 - n) \cdot m \leq$$
$$2m \cdot lg(w) - m.$$

The minimality of the given bound can be seen by studying a λ-free grammar with the following productions

$$S \to A_1$$
$$A_1 \to A_2$$
$$\vdots$$
$$A_{m-1} \to a$$
$$A_{m-1} \to SS.$$

In this grammar $S \Rightarrow_l^{\pi} a^k$, $k \geq 1$, implies $lg(\pi) = 2mk - m$. □

Several results similar to theorem 3.11 are presented in [Sip].

3.7. Left Szilard languages are pure

In a pure grammar no distinction is made between terminals and nonterminals. In this section we study whether left Szilard languages are pure context-free languages. It will be shown that there are left Szilard languages which are not pure context-free languages, but on the other hand, they all belong to a wider class of pure languages, namely to the class of pure length-increasing languages. The material of this section is from [Mäk85e].

3.7.1. Pure languages

A *pure grammar* is a system $H = (\Sigma,P,\sigma)$ where Σ is a finite alphabet, σ is a finite subset of Σ^+ and P is a finite set of productions of the form $\alpha\rightarrow\beta$, where α and β are words over Σ. Relation $\Rightarrow$ (*yields directly*) and its reflexive transitive closure $\Rightarrow^*$ are defined in Σ^* as usual. The language generated by a system $H = (\Sigma,P,\sigma)$ is defined as $L(H) = \{ w \mid s \Rightarrow^* w, s \in \sigma \}$. Languages generated by pure grammars are referred to as *pure languages* [MSW].

A pure grammar $H = (\Sigma,P,\sigma)$ is a *pure context-free grammar* (a PCF grammar for short) if in each production $\alpha\rightarrow\beta$ we have α in Σ. A language is a *pure context-free language* (PCF language) if it can be generated by a PCF grammar [MSW].

If each production $\alpha\rightarrow\beta$ in a pure grammar H has the property that α is not longer than β, then H is a *PLI grammar* (where PLI stands for "pure length-increasing"). A language is a *PLI language* if it can be generated by a PLI grammar [MSW].

Define a context-free grammar $G = (N,T,P,S)$ to be in *pure form* if each production in P has the form $A \to \alpha$ where $\alpha \in N^*$ or $\alpha \in T$, and for each nonterminal occurring in the right-hand side of some production in P there exists exactly one production $A \to a$, $a \in T$. Moreover, $A \to a$ and $B \to b$, where $A \neq B$ and $a, b \in T$, imply $a \neq b$. (Since we consider reduced context-free grammars only, the condition "occurring in the right-hand side of some production" is for the start symbol.)

If a is a terminal in a context-free grammar in pure form then the unique nonterminal A having $A \to a$ is denoted by $\omega(a)$.

Theorem 3.13. A context-free language L is a PCF language if and only if it can be generated by a context-free grammar in pure form.

Proof. Suppose $H = (\Sigma,P,\sigma)$ is a PCF grammar generating L. Define a context-free grammar $G = (N,T,P',S)$ by setting $T = \Sigma$ and $N = \{ \bar{a} \mid a \in \Sigma \} \cup \{S\}$ (S is a new symbol), and to P' take productions

i) $S \to \bar{a}_1 \ldots \bar{a}_n$, for each $s = a_1 \ldots a_n$ in σ,

ii) $\bar{a} \to a$, for each a in Σ

and

iii) $\bar{a} \to \bar{a}_1 \ldots \bar{a}_n$, for each $a \to a_1 \ldots a_n$ in P.

For each word $\alpha = a_1 \ldots a_m$ in $L(H)$ we have a derivation $S \Rightarrow^* \bar{a}_1 \ldots \bar{a}_m$ in G. Hence, α is in $L(G)$. Conversely, for each $a_1 \ldots a_m$ in $L(G)$, we have $S \Rightarrow^* \bar{a}_1 \ldots \bar{a}_m$ and so, $s \Rightarrow^* a_1 \ldots a_m$ holds for some s in σ.

Given a context-free grammar $G = (N,T,P',S)$ in pure form, we can define a PCF grammar $H = (\Sigma,P,\sigma)$ generating $L(G)$ by

setting $\Sigma = T$, $\sigma = \{ a_1 \ldots a_n \mid S \to \omega(a_1) \ldots \omega(a_n)$ is in $P' \}$
and $P = \{ a \to a_1 \ldots a_n \mid \omega(a) \to \omega(a_1) \ldots \omega(a_n)$ is in $P' \}$. □

3.7.2. On pure and left Szilard languages

The generative capacity of PCF grammars is much weaker than the generative capacity of context-free grammars. It is shown in [MSW] that there are regular languages over one-letter alphabets which are not PCF languages. Similarly, there are simple languages which are not pure [Gab]. In contrast to the latter result mentioned we first show that all languages in a subclass of left Szilard languages are PCF languages.

Theorem 3.14. Let $G = (N,T,P,S)$ be a context-free grammar such that each recursive nonterminal A in N has at least one production of the form $A \to \alpha$, $\alpha \in T^*$. Then $Szl(G)$ is a PCF language.
Proof. Let C be the label alphabet of G and let $G' = (N,C,P',S)$ be the ss-grammar obtained from G by the method of theorem 3.1.

We shall define two transformations for grammar G'. The purpose of the transformations is to put the grammar into a form where all nonterminals have exactly one production of the form $A \to \rho$, $\rho \in C$. These transformations will be defined so that they do not change the language generated by G'.

Suppose a non-recursive nonterminal A occurring in the right-hand side of some production in P' does not have any production $A \to \rho$, $\rho \in C$. Let $\{ A \to \rho_1\alpha_1, \ldots, A \to \rho_n\alpha_n \}$ be the set of all productions in P' having A in the left-hand side. Since we

consider reduced context-free grammars only, this set has at least one element. Replace every production $B\to\rho\beta_1 A\beta_2\ldots\beta_k A\beta_{k+1}$, where $\beta_1,\ldots,\beta_{k+1}\in(N\setminus\{A\})^*$, in P by productions $B\to\rho\beta_1 A_{j_1}\alpha_{j_1}\beta_2\ldots\beta_k A_{j_k}\alpha_{j_k}\beta_{k+1}$, where $1\le j_i\le n$, $i=1,\ldots,k$ and A_j, $j=1,\ldots,n$ are new nonterminals. Moreover, replace each production $A\to\rho_j\alpha_j$, $j=1,\ldots,n$, by a production $A_j\to\rho_j$. Hence, we can hereafter suppose that each nonterminal occurring in the right-hand side of some production in P' has at least one production $A\to\rho$, $\rho\in C$.

Suppose now a nonterminal A has productions $A\to\pi$ and $A\to\rho$, $\pi,\rho\in C$, in P'. Let A_π and A_ρ be new nonterminals. Replace $A\to\pi$ by $A_\pi\to\pi$ and $A\to\rho$ by $A_\rho\to\rho$. Moreover, replace each production which contains i, $i>0$, occurrences of A by those 2^i productions which are obtained by replacing A's by all possible combinations of A_π's and A_ρ's. (If A is the start symbol, then add $A\to A_\pi$ and $A\to A_\rho$ to the set of productions.) This transformation reduces the problem to the case where each A (possibly excluding the start symbol) has exactly one production $A\to\rho$, $\rho\in C$.

If $A\to\pi$ and $B\to\rho\alpha$, where $\pi,\rho\in C$ and α is non-empty, are productions in P' then $\pi\neq\rho$. In each production $B\to\rho\alpha$ replace ρ by a new nonterminal A_ρ and add a production $A_\rho\to\rho$ to P'. This does not change the language generated. We now have the resulting context-free grammar G'' in pure form. Hence, by theorem 3.13, $Szl(G)=L(G'')$ is a PCF language. □

The following example demonstrates the difference between PCF and non-PCF left Szilard languages.

Example 3.3. Let G_1 be a context-free grammar with the following productions

$\pi: S \to A$

$\rho: A \to AC$

$\tau: A \to B$

$\upsilon: B \to a$

$\phi: C \to a$.

We have $Szl(G_1) = \{ \pi\rho^n\tau\upsilon\phi^n \mid n \geq 0 \}$. It is easy to see that $Szl(G_1)$ is not a PCF language (because no context-free production can simultaneously increase the numbers of ρ and ϕ). Consider now a context-free grammar G_2 which has all productions of G_1 and moreover, a production $\chi: A \to a$. We have $Szl(G_2) = Szl(G_1) \cup \{ \pi\rho^n\chi\phi^n \mid n \geq 0 \}$. Language $Szl(G_2)$ can be generated by a PCF grammar $(\{\pi,\rho,\tau,\upsilon,\phi,\chi\},\{\chi\to\rho\chi\phi,\chi\to\tau\upsilon\},\{\pi\chi\})$. □

All regular languages are pure languages [MSW]. We shall now show that a similar result holds for left Szilard languages, too.

Theorem 3.15. Let G be a context-free grammar. Then $Szl(G)$ is a PLI language.

Proof. Let $G = (N,T,P,S)$ be a context-free grammar with label alphabet C. We shall define a PLI grammar $H = (\Sigma,P',\sigma)$ such that $L(H) = Szl(G)$.

Set first $\Sigma = C$. Set Σ consists of all strings of productions corresponding to terminal leftmost derivations in G containing no recursive subderivations; i.e. if $S \Rightarrow_l^{\rho} w$, $w \in T^*$, is a derivation containing no recursive subderivations, then ρ is in σ. Let A be a recursive nonterminal in G and let $A \Rightarrow_l^{\tau} w$, $w \in T^*$, be a derivation containing no recursive subderivations. Moreover,

let $A \Rightarrow_l^{\upsilon} uA\alpha$ be a minimal recursive derivation in G such that u is produced without recursive subderivations. Now, for each derivation $\alpha \Rightarrow_l^{\phi} v$, $v \in T^*$, containing no recursive subderivations, take a production $\tau \to \upsilon\tau\phi$ to P'.

Set P' contains only length-increasing productions and PLI grammar H clearly generates the language $Szl(G)$. This completes the proof. □

Example 3.4. Consider again grammar G_1 of example 3.3. G_1 has a non-recursive terminal derivation $S \Rightarrow_l^{\pi\tau\upsilon} a$. Hence, we have $\sigma = \{\pi\tau\upsilon\}$. The only minimal recursive derivation is $A \Rightarrow_l^{\rho} AC$ and the only terminal derivation for A is $A \Rightarrow_l^{\tau\upsilon} a$. Hence, we have a length-increasing production $\tau\upsilon \to \rho\tau\upsilon\phi$. $Szl(G_1)$ can be generated by a PLI grammar $(\{\pi,\rho,\tau,\upsilon,\phi\},\{\tau\upsilon \to \rho\tau\upsilon\phi\},\{\pi\tau\upsilon\})$. □

CHAPTER FOUR: ON DEPTH-FIRST AND BREADTH-FIRST DERIVATIONS

By theorem 2.2, we know that the Szilard language $Sz(G)$ of a context-free grammar G is context-free if and only if G is half-bounded. Hence, many context-free grammars with rather simple derivational structure have non-context-free Szilard languages. This raises the question whether there are reasonable defined sets of derivations which are more general than leftmost derivations and for which Szilard languages are always context-free.

Depth-first derivations of context-free grammars were introduced in [Luk] as an extension of leftmost derivations. In this chapter we study properties of Szilard languages associated with depth-first derivations of context-free grammars and show that this class is a proper subclass of the class of s-languages. Moreover, we define the class of breadth-first Szilard languages and show that this class and the class of context-free languages are incomparable. The material of this chapter is from [Mäk85c].

4.1. Depth-first derivations

Let $G = (N,T,P,S)$ be a context-free grammar with label alphabet C and let $d:S = \alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots \Rightarrow \alpha_n$, $\alpha_n \in T^*$, be a derivation in G. If $\alpha_{i-1} = \beta A \gamma$ and $\alpha_i = \beta\delta\gamma$, $1 \leq i \leq n$, then let $f(B) = i$ for

every particular occurrence of a nonterminal B in δ. Derivation d is said to be *depth-first* if $f(A) \geq f(C)$ holds in $\alpha_{i-1} = \beta A \gamma$, $1 < i \leq n$, for all nonterminals C in $\beta\gamma$ [Luk]. If a derivation $S \Rightarrow^{\rho} w$ is depth-first, we write $S \Rightarrow^{\rho}_{df} w$. The *depth-first Szilard language* Szdf(G) of G can now be defined as $Szdf(G) = \{ \pi \mid S \Rightarrow^{\pi}_{df} w, w \in T^* \}$.

By definition and the fact that leftmost derivations are depth-first, we have $Szl(G) \subseteq Szdf(G) \subseteq Sz(G)$ for every context-free grammar G. The next theorem characterizes the cases where Szl(G) and Szdf(G) or Szdf(G) and Sz(G) coincide.

Theorem 4.1. Let G be a context-free grammar. Then

a. Szl(G) = Szdf(G) holds if and only if each production in G has occurrences of at most one nonterminal in its right-hand side,

and

b. Szdf(G) = Sz(G) holds if and only if G does not have sentential forms with nonterminals A and B such that $A \neq B$ and $f(A) \neq f(B)$.

Proof. omitted.

The following theorem shows that depth-first Szilard languages are not very much different from left Szilard languages.

Theorem 4.2. Let G be a context-free grammar. Then the depth-first Szilard language Szdf(G) is an s-language.

Proof. Let G be a context-free grammar with label alphabet C. Define an s-grammar $G' = (N', T', P', S')$ as follows. First set $T' = C$ and $S' = [S]$. For each production $A \to \alpha_0 A_1 \alpha_1 \cdots \alpha_{n-1} A_n \alpha_n$,

where $A_i \in N$ and $\alpha_j \in T^*$ for all $i = 1,\ldots, n$ and $j = 0, 1,\ldots, n$, in P add $[A_1\ldots A_n]$ to N'. Nonterminals $[A_1\ldots A_n]$ and $[B_1\ldots B_n]$ are considered equal if $B_1\ldots B_n$ is a permutation of $A_1\ldots A_n$. (Hence, the order of G's nonterminals within brackets does not make any difference.) Furthermore, for each $[A_1\ldots A_i\ldots A_n]$, where $n > 1$, in N' add $[A_1\ldots A_{i-1}A_{i+1}\ldots A_n]$ to N' (if it is not already there) and repeat this as long as new nonterminals can be found.

For each production $\rho: B_m \to \alpha_0 A_1 \alpha_1 \ldots \alpha_{n-1} A_n \alpha_n$, where α_j's and A_i's are as above, in P, and for each $[B_1\ldots B_m\ldots B_k]$, where $B_i \neq B_m$ for all $i = 1,\ldots,m-1$, in N' add to P' a production $[B_1\ldots B_m\ldots B_k] \to \rho\alpha\beta$, where

$$\alpha = \begin{cases} [A_1\ldots A_n], & \text{if } n > 0 \\ \lambda, & \text{otherwise} \end{cases}$$

and

$$\beta = \begin{cases} [B_1\ldots B_{m-1}B_{m+1}\ldots B_k], & \text{if } k > 1 \\ \lambda, & \text{otherwise.} \end{cases}$$

(It might be necessary to reorder G's nonterminals in α or β in order to obtain a nonterminal in N'.)

It is straightforward to show by induction that s-grammar G' generates the depth-first Szilard language $Szdf(G)$ of G. □

An s-grammar with productions $S \to aA$ and $A \to a$ shows that the class of depth-first Szilard languages is a proper subclass of the class of s-languages.

4.2. Breadth-first derivations

Another natural restriction of context-free derivations is to

require that nonterminals are replaced in a breadth-first order. Depth-first derivations have $f(A) \geq f(C)$ for nonterminal A to be replaced next and for all other nonterminals C in the sentential form in question. A derivation is said to be *breadth-first* if its sentential forms fulfil the condition $f(A) \leq f(C)$, where A and C are as in the definition of depth-first derivations. The *breadth-first Szilard language* associated with the breadth-first derivations of a context-free grammar G is denoted by $Szbf(G)$.

Notice that not all leftmost derivations are breadth-first.

Theorem 4.3. There are non-context-free breadth-first Szilard languages.

Proof. Consider a context-free grammar G with the following productions

$\rho: S \to SS$

$\sigma: S \to A$

$\tau: A \to \lambda$.

Let $L = Szbf(G) \cap \rho^+\sigma^*\tau^*$. Since each word in L starts with ρ, we have $L = \{ \rho^{n-1}\sigma^n\tau^n \mid n \geq 2 \}$. Since $Szbf(G)$ has a non-context-free intersection with a regular language, it cannot be context-free. □

There are also context-free grammars with a non-context-free Szilard language and a context-free breadth-first Szilard language. Consider a context-free grammar G_1 with the following productions

$\rho: S \to ABS$

$\sigma: A \to \lambda$

$\tau: B \to \lambda$

$\nu: S \to \lambda$.

$Sz(G_1)$ is non-context-free, but $Szbf(G_1)$ is even regular. Indeed, $Szbf(G_1)$ can be denoted by the regular expression

$$\nu + \rho(\rho\sigma\tau + \rho\tau\sigma + \sigma\rho\tau + \sigma\tau\rho + \tau\rho\sigma + \tau\sigma\rho)^*(\nu\sigma\tau + \nu\tau\sigma + \sigma\nu\tau + \sigma\tau\nu + \tau\nu\sigma + \tau\sigma\nu).$$

This can be seen as follows. After an application of $\rho:S\to ABS$ in a breadth-first derivation, we must apply productions $\sigma:A\to\lambda$, $\tau:B\to\lambda$ and $\rho:S\to ABS$ (or $\nu:S\to\lambda$) in some order. As long as we do not apply $\nu:S\to\lambda$, we can repeat all possible subderivations containing one occurrence of $\rho:S\to ABS$, $\sigma:A\to\lambda$ and $\tau:B\to\lambda$. When we replace $\rho:S\to ABS$ by $\nu:S\to\lambda$ in a subderivation, we cannot continue the derivation any longer.

Language $Szl(G_1)$ is also regular, while $Szdf(G_1)$ is non-regular.

Lastly, consider a context-free grammar G_2 obtained from G_1 by replacing production $\rho:S\to ABS$ by $\rho:S\to SAB$. We have $Szdf(G_1) = Szdf(G_2)$ and $Szbf(G_1) = Szbf(G_2)$, but the left Szilard language $Szl(G_2)$ is non-regular. Hence, G_2 is an example of context-free grammars which have regular breadth-first Szilard language and non-regular left Szilard language.

CHAPTER FIVE: ON GRAMMATICAL SIMILARITY

This chapter is devoted to a study of grammatical similarity relations. A concise survey of these relations is given in [Woo, section I.2]. We shall concentrate our study on some "cover-like" relations. Different versions of grammatical coverings are studied and surveyed in [GH, Nij80]. We give a special emphasis to an attempt to interpret properties of cover-like grammatical relations as properties of the left Szilard languages of the context-free grammars in question.

Following [Nij80, S-SW, RH] we now recall the motivation which is usually given for studying grammatical covering relations.

The convenience of expressing the semantics and the efficiency of parsing are often difficult to achieve simultaneously. For example, a parsing method may require that the grammar to be parsed is in some normal form or belongs to a given class of grammars (such as LL(k) grammars). Hence, it is sometimes better to use one grammar for expressing the semantics and a different grammar for parsing. The grammars are called the semantic grammar and parsing grammar [S-SW, RH], respectively. The grammars must be related to each others in such a way that the ability to effectively parse the parsing grammar allows one to systematically produce a parse in the semantic grammar. In covering relations

this is done by a homomorphism between the production sets of the grammars. Several formalizations of the homomorphism are introduced in the literature. We shall focus our study on relations undercover [S-SW] and cover [GH].

In section 5.1 we study the concept of derivation preservation, which is common to several covering relations. Indeed, many of them have the property that the homomorphic image of a terminal derivation in the parsing grammar is always a terminal derivation in the semantic grammar.

In section 5.2 we deal with undercover relation and in section 5.3 with cover relation. Our main results in these sections are homomorphic characterizations of undercover and cover relations.

5.2. Derivation preservation

A system consisting of two context-free grammars and a homomorphism between the production sets of the grammars is of interest in the theory of parsing. A great variety of grammatical relations is introduced for controlling these systems. A common basis for several such relations is that the homomorphism between production sets "preserve derivations", i.e. the homomorphism maps derivations to derivations and terminal derivations to terminal derivations. The concept of derivation preservation is introduced in [HRS76a]. The material of this section is from [Mäk84b].

Let $G_1 = (N_1,T_1,P_1,S_1)$ and $G_2 = (N_2,T_2,P_2,S_2)$ be two context-free grammars with label alphabets C_1 and C_2, respectively, and let h be a homomorphism $h:C_1 \to C_2 \cup \{\lambda\}$. It is sometimes more convenient to consider h as a homomorphism $h:P_1 \to P_2 \cup \{\lambda\}$. Since the labeling of productions is assumed to be bijective, we can always choose between $h:C_1 \to C_2 \cup \{\lambda\}$ and $h:P_1 \to P_2 \cup \{\lambda\}$.

If G_1, G_2 and h are as above, they are said to form a system (G_1,G_2,h). When we later speak about systems (G_1,G_2,h), we implicitly assume the notations given above. Following [HRS76a] we can now give

Definition 5.1. Homomorphism h *preserves derivations* in a system (G_1,G_2,h) if the following conditions hold

1) $\alpha \in (N_1 \cup T_1)^*$ and $S_1 \Rightarrow_l^{\pi} \alpha$ implies $S_2 \Rightarrow_l^{h(\pi)} \beta$ for some $\beta \in (N_2 \cup T_2)^*$

and

2) $v \in T_1^*$ and $S_1 \Rightarrow_l^{\pi} v$ implies $S_2 \Rightarrow_l^{h(\pi)} w$ for some $w \in T_2^*$.

We can equivalently define derivation preservation by using left Szilard languages as follows.

Definition 5.1'. Let (G_1,G_2,h) be a system and let $Szl(G_1)$ and $Szl(G_2)$ be the left Szilard languages of G_1 and G_2, respectively. Homomorphism h preserves derivations in (G_1,G_2,h) if the following conditions hold

1') $h(init(Szl(G_1))) \subseteq init(Szl(G_2))$

and

2') $h(Szl(G_1)) \subseteq Szl(G_2)$.

We now show that under the assumption that only reduced grammars are considered condition 2' implies condition 1'. For all words x in $init(Szl(G_1))$ we must have a string y such that xy is in $Szl(G_1)$. Now, by the properties of homomorphism, if $h(xy)$ is in $Szl(G_2)$, then $h(x)$ is in $init(Szl(G_2))$. We have proved the following

Theorem 5.1. Let (G_1,G_2,h) be a system. Homomorphism h preserves derivations if and only if $h(Szl(G_1)) \subseteq Szl(G_2)$.

By lemma 3.1 and theorem 5.1 we obtain

Theorem 5.2. It is decidable whether or not homomorphism h preserves derivations in a system (G_1,G_2,h).

Next we shall define some grammatical relations which guarantee derivation preservation. Notice that for undercover (item a.) and cover (item b.) relations we can define several variations of the main idea by using leftmost or rightmost derivations in both grammars or leftmost derivations in one grammar and rightmost derivations in the other one. Since in chapter 3 we studied left Szilard languages only, we now omit all other versions but the ones where leftmost derivations are used in both grammars. For obvious reasons we use names "undercover" and "cover" instead of the more complete names "left undercover" and "left cover" used in the literature.

Definition 5.2. Let $G_1 = (N_1,T_1,P_1,S_1)$ and $G_2 = (N_2,T_2,P_2,S_2)$ be context-free grammars. In system (G_1,G_2,h)

a. G_2 is said to *undercover* G_1 with respect to h if

$T_1 = T_2 = T$, $h(A \to \alpha) \neq \lambda$ for all $A \to \alpha$ in P_1, and for all w in T^* $S_1 \Rightarrow_l^{\pi} w$ in G_1 always implies $S_2 \Rightarrow_l^{h(\pi)} w$ in G_2 [S-SW]

b. G_1 is said to *cover* G_2 with respect to h if $T_1 = T_2 = T$ and for all w in T^* the following conditions hold
   1) whenever $S_1 \Rightarrow_l^{\pi} w$ is in G_1 then $S_2 \Rightarrow_l^{h(\pi)} w$ is in G_2
   2) whenever $S_2 \Rightarrow_l^{\pi} w$ is in G_2 then there exists a π' such that $h(\pi') = \pi$ and $S_1 \Rightarrow_l^{\pi'} w$ in G_1 [GH]

c. G_2 is said to *SI-cover* G_1 if there exists a homomorphism $f:(N_1 \cup T_1) \to (N_2 \cup T_2)$ such that $f(S_1) = S_2$, $f(N_1) \subseteq N_2$, $f(T_1) \subseteq T_2^*$ and $h(A \to \alpha) = f(A) \to f(\alpha)$ for all $A \to \alpha$ in P_1 [RH]

d. G_2 is said to *Reynolds cover* G_1 if G_2 SI-covers G_1, and moreover, $f(a) = a$ for all a in T_1 [GH, RH].

Notice, that in cover relation h maps the productions of the parsing grammar to the productions of the semantic grammar, while in other relations it maps productions of the semantic grammar to the productions of the parsing grammar. Hence, in all four cases the parsing grammar "covers" the semantic grammar.

Contrary to SI-cover, relations cover, undercover and Reynolds cover require that the corresponding derivations generate the same terminal word in both grammars.

Theorem 5.3. Let (G_1, G_2, h) be a system. If

a. G_2 undercovers G_1

b. G_1 covers G_2

c. G_2 SI-covers G_1

or

d. G_2 Reynolds covers G_1,

then h preserves derivations in (G_1,G_2,h).

Proof. Cases a. and b. are obvious.

Consider cases c. and d. By definition, we have $f(S_1) = S_2$ and hence, $h(S_1 \to \alpha) = S_2 \to \beta$, $\beta \in (N_2 \cup T_2)^*$, for every production $S_1 \to \alpha$ in G_1 with left-hand side S_1. Moreover, for every production $A \to \alpha$ in G_1, we have equal numbers of nonterminals in the right-hand sides of $A \to \alpha$ and $h(A \to \alpha)$. Hence, for every leftmost derivation π from S_1 in G_1, we have a corresponding leftmost derivation $h(\pi)$ from S_2 in G_2 such that π reaches a terminal word if and only if $h(\pi)$ reaches a terminal word. □

5.1.1. On the nonterminal relation induced by h

In this subsection we study the properties of a nonterminal relation induced by derivation preserving homomorphism h. This relation connects a nonterminal A in G_1 to a nonterminal B in G_2, if we have $h(A \to \alpha) = B \to \beta$ for some productions $A \to \alpha$ in G_1 and $B \to \beta$ in G_2.

Our first theorem concerning this nonterminal relation says that it is a partial function.

Theorem 5.4. Let (G_1,G_2,h) be a system such that h preserves derivations. If $h(A \to \alpha_1) = B \to \beta$ and $h(A \to \alpha_2) = C \to \gamma$, then $B = C$.

Proof. Suppose that h preserves derivations in (G_1,G_2,h) and $h(A \to \alpha_1) = B \to \beta$ and $h(A \to \alpha_2) = C \to \gamma$ but $B \neq C$.

We have $S_1 \Rightarrow_l^{\pi} uA\delta$ in G_1 for some u in T_1^* and δ in $(N_1 \cup T_1)^*$. Both $\rho: A \to \alpha_1$ and $\sigma: A \to \alpha_2$ can be applied to sentential form $uA\delta$. Hence, both $\pi\rho$ and $\pi\sigma$ are in $init(Szl(G_1))$. So, we have $h(\pi\rho)$ and $h(\pi\sigma)$ in $init(Szl(G_2))$. However, in G_2 we have $S_2 \Rightarrow_l^{h(\pi)} \zeta$ for some ζ in $(N_2 \cup T_2)^*$ and only one of $B \to \beta$ and $C \to \gamma$ can be applied to the leftmost nonterminal in ζ. This is a contradiction. □

Following [S-SW] the partial nonterminal function induced by h is denoted by $\bar{h}$. Hence, if $h(A \to \alpha) = B \to \beta$, then $\bar{h}(A) = B$. If it is not allowed to have $h(A \to \alpha) = \lambda$ for any production $A \to \alpha$ in G_1, we can prove the following theorem, which has been proved in [S-SW] in the case of undercover relation. Hence, the following is not essentially a property of undercover, but a property of any system in which derivations are preserved.

Theorem 5.5. Let (G_1, G_2, h) be a system such that h preserves derivations and $h(A \to \alpha) \neq \lambda$ for all $A \to \alpha$ in G_1. Then the nonterminal relation $\bar{h}$ induced by h is a function.

Proof. Since $h(A \to \alpha) = \lambda$ is impossible, $\bar{h}$ maps every nonterminal A in G_1 to some nonterminal in G_2. By theorem 5.4, $\bar{h}$ is single-valued. □

Note, that theorem 5.5 is applicable also to systems (G_1, G_2, h) where G_2 Reynolds covers or SI-covers G_1.

If we require a closer resemblance between a production $A \to \alpha$ in G_1 and its image $h(A \to \alpha)$, we come to the following theorem. The number of nonterminals in the right-hand side of a production labeled with ρ is denoted by $n(\rho)$.

Theorem 5.6. Let (G_1,G_2,h) be a system such that h preserves derivations and for all $\rho:A\to\alpha$ in G_1, $n(\rho) = n(h(\rho))$ holds. Then homomorphism h induces a function $\bar{h}:N_1\to N_2$ such that if $A\to\alpha$ in G_1 has nonterminals $A_1,\ldots,A_n$ in its right-hand side (in this order) then

$$h(A\to\alpha) = \bar{h}(A)\to\beta_1\bar{h}(A_1)\beta_2\ldots\beta_m\bar{h}(A_m)\beta_{m+1}$$

for some β_i in T_2^*, $i = 1,\ldots, m+1$.

Proof. The claim concerning left-hand sides holds by theorem 5.5. In order to prove the claim concerning right-hand sides, suppose it does not hold for some $\rho:A\to\alpha$ in G_1. By definition 5.1 and theorem 5.4, we have for every derivation

$$S_1 \Rightarrow_l^{\pi} uA\gamma \Rightarrow_l^{\rho} u\alpha\gamma$$

in G_1 a derivation

$$S_2 \Rightarrow_l^{h(\pi)} v\bar{h}(A)\delta \Rightarrow_l^{h(\rho)} v\beta\delta$$

in G_2.

Let i be the smallest index such that $\bar{h}(A_i)$ differs from the i-th nonterminal B_i in β. Since $n(\rho) = n(h(\rho))$, we have equally long derivations to terminal strings from $A_1,\ldots,A_{i-1}$ in G_1 and from $\bar{h}(A_1),\ldots,\bar{h}(A_{i-1})$ in G_2. Hence, we have

$$S_1 \Rightarrow_l^{\pi} uA\gamma \Rightarrow_l^{\pi'} u'A_i\gamma'$$

in G_1 and

$$S_2 \Rightarrow_l^{h(\pi)} v\bar{h}(A)\delta \Rightarrow_l^{h(\pi')} v'B_i\delta', \text{ where } \bar{h}(A_i) \neq B_i,$$

in G_2. This contradicts theorem 5.4. □

If a system (G_1,G_2,h) is as in theorem 5.6 we can define a homomorphism f such that it maps nonterminals as required in the definition of SI-cover. However, the situation of theorem 5.6 does not imply anything about terminal symbols, and hence, a SI-cover is not necessarily obtained.

5.2. Undercover

We start with some simple remarks concerning systems (G_1,G_2,h) where G_2 undercovers G_1 with respect to h.

Since a terminal derivation $S_1 \Rightarrow_l^{\pi} v$ in G_1 always implies that $S_2 \Rightarrow_l^{h(\pi)} v$ is in G_2, we have $L(G_1) \subseteq L(G_2)$. Because of this property we need an easy method for detecting if a given string is in $L(G_2)$ but not in $L(G_1)$.

If G_2 undercovers G_1 and G_2 is unambiguous, then G_1 is unambiguous, too.

In definition 5.2.a we required that $h(A\to\alpha) \neq \lambda$ holds for every production $A\to\alpha$ in G_1. This is a natural restriction, since if $A\to\alpha$ were not mapped to any production in parsing grammar, then the semantics associated with it would be lost.

It follows that the corresponding derivations in G_1 and G_2 must be equally long. Hence, we can conclude that if G_2 is in CNF or in GNF, then also G_1 must be in the same normal form.

Another consequence of the requirement $h(A\to\alpha) \neq \lambda$ is that theorem 5.5 is applicable to all systems (G_1,G_2,h) where G_1 undercovers G_2. If we further suppose that a condition similar to that in theorem 5.6 holds, we obtain the following theorem which gives a relationship between undercover and Reynolds cover.

Following [S-SW] we first give a definition concerning homomorphism

h in a system (G_1,G_2,h). A homomorphism h is said to be *type preserving* if $h(A\to X_1X_2\ldots X_m) = B\to Y_1Y_2\ldots Y_n$ implies $m = n$ and X_i, $i = 1,\ldots,n$, $n \geq 0$, is nonterminal if and only if Y_i is nonterminal.

Theorem 5.7 [S-SW]. Let (G_1,G_2,h) be a system where G_2 undercovers G_1 with respect to h. Then G_2 Reynolds covers G_1 if and only if h is type preserving.

Proof. If G_2 Reynolds covers G_1 then, by the definition of Reynolds cover, h must be type preserving.

Assume then that h is type preserving. By theorem 5.3, h preserves derivations. We can apply theorem 5.6, and hence, the nonterminal relation $\overline{h}$ induced by h fulfils the condition for nonterminals in the definition of Reynolds cover.

Suppose now that for some production $\sigma:A\to\alpha$ in P_1 we have $h(A\to\alpha) = B\to\beta$ such that $\alpha = \alpha_1a\alpha_2$ and $\beta = \beta_1b\beta_2$, where $lg(\alpha_1) = lg(\beta_1)$, $a \in T_1$, $b \in T_2$ and $a \neq b$.

Let $S_1 \Rightarrow_l^{\pi} uA\delta_1$, where $u \in T_1^*$, $\delta_1 \in (N_1 \cup T_1)^*$, be a derivation in G_1. Since G_2 undercovers G_1, we must have $S_2 \Rightarrow_l^{h(\pi)} vB\delta_2$ for some $v \in T_2^*$ and $\delta_2 \in (N_2 \cup T_2)^*$. Since h is type preserving, we have $lg(u) = lg(v)$ and $lg(\delta_1) = lg(\delta_2)$. For the same reason derivations $\alpha_1 \Rightarrow_l^{\rho} u'$, where $u' \in T_1^*$, in G_1 and $\beta_1 \Rightarrow^{h(\rho)} v'$ where $v' \in T_2^*$, have $lg(u') = lg(v')$. Derivations $S_1 \Rightarrow_l^{\pi\rho\sigma} uu'a\delta_1$ and $S_2 \Rightarrow_l^{h(\pi\rho\sigma)} vv'b\delta_2$, where $lg(uu') = lg(vv')$ and $a \neq b$, contradicts the undercover property. Hence, (G_1,G_2,h) also fulfils the condition for terminals in the definition of Reynolds cover. □

Let G_1 and G_2 be regular grammars. Then homomorphism h must be type preserving, if G_2 undercovers G_1 in (G_1,G_2,h). So we have the following corollary.

Corollary 5.1 [S-SW]. If G_1 and G_2 are regular grammars, then G_2 undercovers G_1 if and only if G_2 Reynolds covers G_1.

The following example shows that in theorem 5.7 we cannot replace condition "h is type preserving" by the condition of theorem 5.6 which says merely that the corresponding productions have equal numbers of nonterminals.

Example 5.1. Let G_1 and G_2 context-free grammars with productions

G_1:	G_2:
S→aA	S→abA
A→bc	A→c.

Then G_2 undercovers G_1 and vice versa, but neither G_2 Reynolds covers G_1 nor vice versa. □

Does there exist an algorithm for deciding whether or not G_2 undercovers G_1 with respect to h in a given system (G_1,G_2,h)? We leave the question open in the case of two arbitrary context-free grammars, but we shall show that there is an algorithm if G_1 and G_2 are in semi-GNF.

We need some concepts from the theory of homomorphism equivalence. Let L $(\subseteq \Sigma^*)$ be a language and let h_1 and h_2 be homomorphisms defined on Σ^*. We say that h_1 and h_2 are equivalent on L if $h_1(w) = h_2(w)$ holds for all words w in L. The *problem of homomorphism equivalence* for a class of languages is that of de-

ciding whether or not two arbitrary homomorphisms are equivalent on a language of this class [CS].

Lemma 5.1 [ACK,CS]. The problem of homomorphism equivalence is decidable for the class of context-free languages.

Next, we give a characterization of undercover relation in the case where both the semantic grammar and the parsing grammar are in semi-GNF. Recall first our notations. Let $G_1 = (N_1,T_1,P_1,S_1)$ and $G_2 = (N_2,T_2,P_2,S_2)$ be context-free grammars in semi-GNF, where $T_1 = T_2 = T$, with left Szilard languages $Szl(G_1)$ $(\subseteq C_1^*)$ and $Szl(G_2)$ $(\subseteq C_2^*)$, respectively, and a homomorphism $h:C_1 \to C_2$. Furthermore, define homomorphisms $h_i:C_i \to T^*$, $i = 1, 2$, by setting $h_i(\rho) = w$, if $\rho:A \to w\alpha$, where $w \in T^*$ and $\alpha \in N_i^*$, is in P_i.

We have the situation described in figure 5.1.

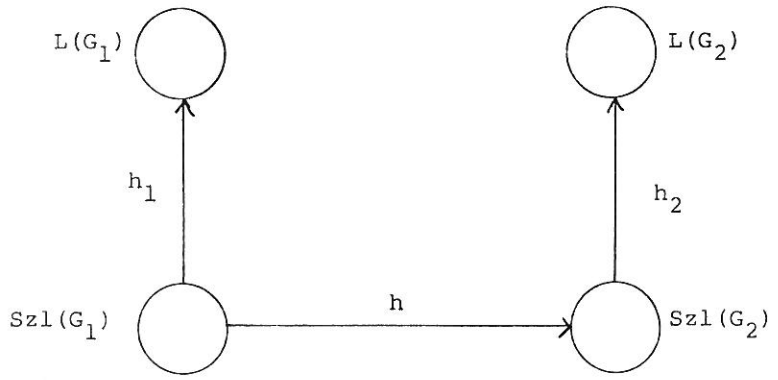

Figure 5.1.

In the next theorem, the composition of homomorphisms h and h_2 is denoted by $h_2 \circ h$.

Theorem 5.8 [Mäk85a]. Let G_i, $Szl(G_i)$, h, C_i and h_i, $i = 1, 2$, be as above. Then G_2 undercovers G_1 with respect to h if and only if the following two conditions hold

1) $h(Szl(G_1)) \subseteq Szl(G_2)$

and

2) homomorphisms h_1 and $h_2 \circ h$ are equivalent on $Szl(G_1)$.

Proof. Suppose first that G_2 undercovers G_1. By definition, we have a leftmost terminal derivation $h(\pi)$ in G_2 for every leftmost terminal derivation π in G_1. Hence, condition 1) holds. In order to prove that 2) holds, suppose there is a word π in $Szl(G_1)$ such that $h_1(\pi) \neq h_2(h(\pi))$. This means that we have derivations $S_1 \Rightarrow_l^{\pi} w$ and $S_2 \Rightarrow_l^{h(\pi)} v$ such that $w \neq v$, a contradiction.

Conversely, suppose that 1) and 2) hold. Condition 1) implies that for each terminal derivation π in G_1 we have a terminal derivation $h(\pi)$ in G_2. Moreover, condition 2) implies that these two derivations generate the same terminal word. Hence, G_2 undercovers G_1 with respect to h. □

By using this characterization, we can prove the following

Theorem 5.9 [Mäk85a]. Let G_1 and G_2 be context-free grammars in semi-GNF. It is decidable whether or not G_2 undercovers G_1 with respect to a homomorphism h.

Proof. Let $Szl(G_i)$, C_i and h_i, $i = 1, 2$, be as above. In condition 1) of theorem 5.8, $h(Szl(G_1))$ is a context-free lan-

guage and $Szl(G_2)$ is a left Szilard language. Hence, by lemma 3.1, we have an algorithm for checking condition 1). By lemma 5.1, we also have an algorithm for checking the equivalence of h_1 and $h' = h_2 \circ h$ on $Szl(G_1)$. Hence, we conclude the theorem. □

5.3. Cover

Formally, a system (G_1,G_2,h) where G_1 covers G_2 resembles closely a system in which G_2 undercovers G_1. We have only relinguished condition $h(A \to \alpha) \neq \lambda$ and on the other hand, we now forbid such derivations in G_2 which are not homomorphic images of any derivations in G_1. However, cover and undercover relations have the fundamental difference that the roles of semantic and parsing grammars are opposite.

The possibility to have "empty production" as the image of a production under h, releases us from the requirement that corresponding derivations are equally long. This makes it possible to find covering grammars in several normal forms. A survey of such covering grammars can be found in [Nij80].

On the other hand, if G_1 covers G_2 then $L(G_1) = L(G_2)$. Also the following simple theorem holds.

Theorem 5.10. Let $G_1 = (N_1,T_1,P_1,S_1)$ and $G_2 = (N_2,T_2,P_2,S_2)$ be context-free grammars such that G_1 covers G_2 in (G_1,G_2,h). Then $card(P_1) \geq card(P_2)$ and $card(N_1) \geq card(N_2)$.

The conclusion of theorem 5.10 naturally holds also when G_2 undercovers or Reynolds covers G_1.

In the previous section we left open the question if it is decidable whether G_2 undercovers G_1 in (G_1,G_2,h), where G_1 and G_2 are arbitrary context-free grammars. Consider now the corresponding problem for cover relation. We can restrict ourselves to the case where both G_1 and G_2 have only two, say 0 and 1, terminal symbols.

Let $G = (N,T,P,S)$ be a context-free grammar over $\{ 0,1 \}$ such that the right-hand sides of G's productions are in $0N^* \cup 1N^* \cup N^*$. It is undecidable whether $L(G) = \{ 0,1 \}^*$. We shall construct context-free grammars G_1 and G_2 such that $L(G) = \{ 0,1 \}^*$ if and only if G_1 covers G_2. Define $G_1 = (N \cup \{ S_1,\$ \}, \{ 0,1 \}, P \cup \{ S_1 \rightarrow S\$, \$ \rightarrow 1 \}, S_1)$, where $N \cap \{ S_1,\$ \} = \Phi$, and $G_2 = (\{ S_2 \}, \{ 0,1 \}, \{ S_2 \rightarrow 0S_2, S_2 \rightarrow 1S_2, S_2 \rightarrow 1 \}, S_2)$. Homomorphism h is defined by

$h(A \rightarrow 0\alpha) = S_2 \rightarrow 0S_2$, for each $A \rightarrow 0\alpha$, $\alpha \in N^*$, in P,

$h(A \rightarrow 1\alpha) = S_2 \rightarrow 1S_2$, for each $A \rightarrow 1\alpha$, $\alpha \in N^*$, in P,

$h(\$ \rightarrow 1) = S_2 \rightarrow 1$,

$h(A \rightarrow \alpha) = \lambda$, for each $A \rightarrow \alpha$, $\alpha \in N^*$, in P,

and

$h(S_1 \rightarrow S\$) = \lambda$.

It is straightforward to show that $L(G) = \{ 0,1 \}^*$ if and only if G_1 covers G_2 with respect to h (for further details, see [HRS76b]). Hence, we obtain the following

Theorem 5.11 [HRS76b]. Let G_1 and G_2 be context-free grammars over $\{ 0,1 \}$. Then it is undecidable whether G_1 covers G_2 in a system (G_1,G_2,h).

A characterization similar to theorem 5.8 is a useful tool when studying the undecidability of cover relation.

Theorem 5.12. Let G_i, $Szl(G_i)$, C_i and h_i, $i = 1, 2$, be as in theorem 5.8 and let h be a homomorphism $h:C_1 \to C_2 \cup \{ \lambda \}$. Then G_1 covers G_2 with respect to h if and only if the following three conditions hold

1) $h(Szl(G_1)) \subseteq Szl(G_2)$,

2) homomorphism h_1 and $h_2 \circ h$ are equivalent on $Szl(G_1)$, and

3) $Szl(G_2) \subseteq h(Szl(G_1))$.

Proof. As in the proof of theorem 5.8, we can show that condition 1) of definition 5.2.b holds if and only if 1) and 2) hold. (Indeed, in this respect the possibility to have $h(A \to \alpha) = \lambda$ for a production $A \to \alpha$ in G_1 does not make any difference.)

Similarly, condition 3) corresponds to condition 2) of definition 5.2.b. □

Since there are algorithms to check conditions 1) and 2) of theorem 5.12, the decidability of the cover problem depends on the inclusion $Szl(G_2) \subseteq h(Szl(G_1))$.

It is shown in [HRS76a] that the cover problem is decidable in (G_1, G_2, h), where G_1 and G_2 are linear grammars. (In fact this problem is shown to be PSPACE-complete in [HRS76a].) In this case both $Szl(G_2)$ and $h(Szl(G_1))$ are regular.

Theorem 5.13. Let G_1 and G_2 be context-free grammars in semi-GNF. There is an algorithm for deciding whether G_1 covers G_2 with respect to a given homomorphism h if

1) G_1 and G_2 are left-derivation bounded

or

2) G_1 (or G_2) is unambiguous and it generates a bounded language.

Proof. If G_1 and G_2 are left-derivation bounded, then $Szl(G_1)$, $Szl(G_2)$ and $h(Szl(G_2))$ are all regular. The inclusion problem is decidable for regular languages.

If a context-free grammar is unambiguous and generates a bounded language, then by theorem 3.6, its left Szilard language is bounded. Bounded languages are closed under homomorphism [Gin]. It is decidable whether $L_1 \subseteq L_2$ holds for context-free languages L_1 and L_2 if one of them is bounded [Gin]. □

If G_1 covers G_2, then by conditions 1) and 3) of theorem 5.12, $h(Szl(G_1)) = Szl(G_2)$. As mentioned in section 3.3, "$L_1 = L_2$?" is decidable if L_1 is a deterministic context-free language and L_2 is superdeterministic. A simple special case where $h(Szl(G_1))$ is deterministic is that where $h(A\to\alpha) \neq \lambda$ for all productions $A\to\alpha$ in G_1 and $h(A\to\alpha_1) \neq h(A\to\alpha_2)$ for all productions $A\to\alpha_1$ and $A\to\alpha_2$ with common left-hand side. In this case $h(Szl(G_1))$ is an s-language.

We also have the following

Theorem 5.14. Let (G_1,G_2,h) be a system where G_1 covers G_2 with respect to h. If $Szl(G_1)$ is regular (resp. bounded), then $Szl(G_2)$ is regular (resp. bounded), too. Moreover, if G_1 is unambiguous, then $Szl(G_1)$ is bounded if and only if $Szl(G_2)$ is bounded.

Proof. The first claim is obvious, since $h(Szl(G_1)) = Szl(G_2)$ and regular and bounded languages are closed under homomorphism.

Suppose now that G_1 is unambiguous. If $Szl(G_2)$ is bounded, then by theorem 3.7, $L(G_2)$ is bounded, too. We have $L(G_1) = L(G_2)$ and by theorem 3.6, $Szl(G_1)$ is bounded. □

CHAPTER SIX: CONCLUSION

We have studied context-free derivations by representing the set of derivations in a context-free grammar as a language, termed Szilard language, in which we have a word for every terminating derivation. Although there are non-context-free Szilard languages, several decision problems undecidable for context-free languages are decidable for Szilard languages. The relationship between context-free and Szilard languages was further studied in section 2.4.

By restricting the form of context-free derivations we obtain subsets of Szilard languages. The most natural restrictions are leftmost and rightmost derivations. In section 3 we studied left Szilard languages associated with leftmost derivations. Interesting results were obtained when we studied situations where the language generated by a context-free grammar or the left Szilard language of a context-free grammar is bounded.

If a context-free grammar is in semi-GNF, then we can define a homomorphism h such that $L(G)$ is the homomorphic image of $Szl(G)$ under h. In section 3.5 the ambiguity of a context-free grammar was characterized by the properties of homomorphism h.

Other restrictions of context-free derivations were studied in chapter 4. The main result of this chapter was that each depth-first Szilard language associated with depth-first derivations is an s-language.

Lastly, in chapter 5, we applied the results of chapter 3 to the

theory of grammatical similarity relations, especially to certain cover-like relations. Our main results in this chapter were homomorphic characterizations of undercover and cover relations for context-free grammar in semi-GNF.

References

(Note: References [Bor, Fle] are not citied in text.)

[ACK] Albert, J., Culik II, K., and Karhumäki, J., Test sets for context-free languages and algebraic systems of equations over a free monoid, Inform. Control 52 (1982), 172-186.

[AB] Altman, E., and Banerji, R., Some problems of finite representability, Inform. Control 8 (1965), 251-263.

[AHU] Aho, A.V., Hopcroft, J.E., and Ullman, J.D., The Design and Analysis of Computer Algorithms. Addison-Wesley, 1974.

[AU] Aho, A.V, and Ullman, J.D., The Theory of Parsing, Translation, and Compiling. Vols 1 and 2. Prentice-Hall, 1972 and 1973.

[BM] Bader, C., and Moura, A., A generalization of Ogden's lemma, J. ACM 29 (1982), 404-407.

[BL] Blattner, M., and Latteux, M., Parikh-bounded languages, in: Lecture Notes in Computer Science 115, Springer-Verlag, 1982, 316-323.

[Bor] Borgida, A.T., Some formal results about stratificational grammars and their relevance to linguistics, Math. Systems Theory 16 (1983), 29-56.

[C-RM] Crespi-Reghizzi, S., and Mandrioli, D., Petri nets and Szilard languages, Inform. Control 33 (1977), 177-192.

[Cul79] Culik II, K., Some decidability results about regular and pushdown translations, Inform. Process. Lett. 8 (1979), 5-8.

[Cul80] Culik II, K., Homomorphisms: decidability, equality, and test sets, in: Book, R., (ed.), Formal Language Theory, Perspectives and Open Problems. Academic Press, 1980.

[CS] Culik II, K., and Salomaa, A., On the decidability of homomorphism equivalence for languages, J. Comput. System Sci. 17 (1978), 163-175.

[FMR] Fischer, P., Meyer, A., and Rosenberg, A., Counter machines and counter languages, Math. Syst. Theory 2 (1968), 265-283.

[Fle] Fleck, A.C., An analysis of grammars by their derivation sets, Inform. Control 24 (1974), 389-398.

[Gab] Gabrielian, A., Pure grammars and pure languages, Intern. J. Comput. Math. 9 (1981), 3-16.

[GJ] Garey, M.R., and Johnson, D.S., Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, 1979.

[Gin] Ginsburg, S., The Mathematical Theory of Context-Free Languages. McGraw-Hill, 1966.

[Gra] Grant, P.W., Extensions of Sokolowski's theorem to prove languages are not context free or not regular, Intern J. Comput. Math. 11 (1982), 187-196.

[GH] Gray, J.N., and Harrison, M.A., On the covering and reduction problems for context-free grammars, J. ACM 19 (1972), 675-698.

[GF] Greibach, S.A., and Friedman, E.P., Superdeterministic PDAs: a subcase with a decidable inclusion problem, J. ACM 27 (1980), 675-700.

[Har] Harrison, M.A., Introduction to Formal Language Theory. Addison-Wesley, 1978.

[HU69] Hopcroft, J.E., and Ullman, J.D., Formal Languages and Their Relation to Automata. Addison-Wesley, 1969.

[HU79] Hopcroft, J.E., and Ullman, J.D., Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, 1979.

[Höp] Höpner, M., Über den Zusammenhang von Szilardsprachen und Matrixgrammatiken, Betricht Nr. 12, Institut für Informatik, Universität Hamburg, 1974.

[Hor] Horvath, S., The family of languages satisfying Bar-Hillel's lemma, RAIRO Inf. Theor. 12 (1978), 193-199.

[HR] Hunt III, H.B., and Rosenkrantz, D.J., On equivalence and containment problems for formal languages, J. ACM 24 (1977), 387-396.

[HRS76a] Hunt III, H.B., Rosenkrantz, D.J., and Szymanski, T.G., The covering problem for linear context-free grammars. Theoret. Comput. Sci. 2 (1976), 361-382.

[HRS76b] Hunt III, H.B., Rosenkrantz, D.J., and Szymanski, T.G., On the equivalence, containment, and covering problems for the regular and context-free languages, J. Comput. System Sci. 12 (1976), 222-268.

[Iba] Ibarra, O.H., On some decision questions concerning push-down machines, Theoret. Comput. Sci. 24 (1983), 313-322.

[Iga] Igarashi, Y., The tape complexity of some classes of Szilard languages, SIAM J. Comput. 6 (1976), 460-466.

[Jan] Jantzen, M., On the hierarchy of Petri net languages, RAIRO Inf. Theor. 13 (1979), 19-30.

[JO] Jantzen, M., and Opp, M., A normal form theorem for label grammars, Math. Syst. Theory 14 (1981), 289-303.

[Klø] Kløve, T., Pumping languages, Intern. J. Comput. Math. 6 (1977), 115-125.

[Kos] Kosaraju, S.R., Decidability of reachability in vector addition systems, Proceedings of the 14th Annual ACM Symposium on Theory of Computing, 1982, 267-281.

[KM] Kriegel, H.P., and Maurer, H.A., Formal translations and Szilard languages, Inform. Control 30 (1976), 187-198.

[KO] Kriegel, H.P., and Ottmann, Th., Left-fitting translations, in: Lecture Notes in Computer Science 52, Springer- Verlag, 1977, 309-322.

[Lin] Linna, M., Two decidability results for deterministic pushdown automata, J. Comput. System Sci. 18 (1979), 92-107.

[Luk] Luker, M., Control sets on grammars using depth-first derivations, Math. Syst. Theory 13 (1980), 349-359.

[Mäk83a] Mäkinen, E., On certain properties of left Szilard languages, EIK 19 (1983), 497-501.

[Mäk83b] Mäkinen, E., Boundedness testing for unambiguous context-free grammars, Inform. Process. Lett. 17 (1983), 181-183.

[Mäk84a] Mäkinen, E., On context-free and Szilard languages, BIT 24 (1984), 164-170.

[Mäk84b] Mäkinen, E., On derivation preservation, Inform. Process. Lett. 19 (1984), 225-228.

[Mäk85a] Mäkinen, E., A note on undercover relation, Inform. Process. Lett. 20 (1985), 19-21.

[Mäk85b] Mäkinen, E., An undecidable problem for context-free grammars, Inform. Process. Lett. 20 (1985), 141-142.

[Mäk85c] Mäkinen, E., A note on depth-first derivations, BIT 25 (1985), 293-296.

[Mäk85d] Mäkinen, E., On permutative grammars generating context-free languages, to appear in BIT.

[Mäk85e] Mäkinen, E., A note on pure grammars, submitted for publication to Inform. Process. Lett.

[MSW] Maurer, H.A., Salomaa, A., and Wood, D., Pure grammars, Inform. Control 44 (1980), 47-72.

[Mor] Moriya, E., Associative languages and derivational complexity of formal grammars and languages, Inform. Control 22 (1973), 117-130

[Nij80] Nijholt, A., Context-Free Grammars: Covers, Normal Forms and Parsing. Lecture Notes in Computer Science 93, Springer-Verlag, 1980.

[Nij82] Nijholt, A., A note on the sufficiency of Sokolowski's criterion for context-free languages, Inform. Process. Lett. 14 (1982), 207.

[Ogd] Ogden, W., A helpful result for proving inherent ambiguity, Math. Syst. Theory 2 (1968), 191-194.

[Pen74] Penttonen, M., On derivation languages corresponding to context-free grammars, Acta Inform. 3 (1974), 285-291.

[Pen77] Penttonen, M., Szilard languages are log n tape recognizable, EIK 13 (1977), 595-602.

[RH] Rosenkrantz, D.J., and Hunt III, H.B., Efficient algorithms for automatic construction and compactification of parsing grammars, Technical Report 82-12, Dept. of Computer Science, SUNY at Albany, 1982.

[Sal] Salomaa, A., Formal Languages. Academic Press, 1973.

[Sil] Sillars, W.A., Formal properties of certain classes of essentially context dependent languages, Ph. D. Thesis, Pennsylvania State University, 1968.

[Sip] Sippu, S., Derivational complexity of context-free grammars, Inform. Control 53 (1982), 52-65.

[S-SW] Soisalon-Soininen, E., and Wood. D., On a covering relation for context-free grammars, Acta Inform. 17 (1982), 435-449.

[Sok] Sokolowski, S., A method for proving programming languages non context-free, Inform. Process. Lett. 7 (1978), 151-153.

[TK] Taniguchi, K., and Kasami, T., A result on the equivalence problem for deterministic pushdown automata, J. Comput. System Sci. 13 (1976), 38-50.

[Wal] Walljasper, S.J., Left-derivation bounded languages, J. Comput. System Sci. 8 (1974), 1-7.

[Woo] Wood, D., Grammar and L Forms: An Introduction. Lecture Notes in Computer Science 91, Springer-Verlag, 1980.

[Yaf] Yaffe, V.A., Two classes of CF-languages with a solvable equivalence problem, Kibernetika 2 (1974), 89-93 [English translation].

Myyntijakelu: Tampereen yliopisto/julkaisujen myynti
PL 617, 33101 Tampere

Distributor: University of Tampere/Sales Office
P.O. Box 617, SF-33101 Tampere, Finland

ISBN 951-44-1837-9
ISSN 0496-7909