Review: [untitled]
Author(s): D. Terence Langendoen
Reviewed work(s):
Ambiguity in Context Free Languages. by Seymour Ginsburg ; Joseph Ullian
Preservation of Unambiguity and Inherent Ambiguity in Context-Free by Seymour Ginsburg ; Joseph Ullian
The Independence of Inherent Ambiguity from Complementednes Among Context-Free Languages. by Thomas N. Hibbard ; Joseph Ullian
Source: *The Journal of Symbolic Logic*, Vol. 33, No. 2 (Jun., 1968), pp. 301-302
Published by: Association for Symbolic Logic
Stable URL: http://www.jstor.org/stable/2269906
Accessed: 12/05/2009 12:21

structure (p.s.) grammar is, strictly speaking, a semi-Thue system with the supplementary condition that if a pair of words (x, y) is a production in an alphabet V, then there exists a single symbol A in the alphabet and words z, w, and v, where v is not an empty word, such that $x = zAw$ and $y = zvw$. Moreover, existence of a terminal alphabet is assumed, which results in an extension of the class of languages definable by p.s. grammars. Context-free grammars form a particular case of p.s. grammars obtained by assuming that if (x, y) is a production, then x is a single symbol. It has been proved in 1960 that the class of all context-free (c.f.) languages, i.e., languages which can be described by c.f. grammars, is identical with the class of the so-called ALGOL-like languages, i.e., languages which can be defined by the same metalanguage (Backus normal form) which is used to describe ALGOL. This result, however arguable today as to its meaning, has provoked an intensive development of the theory of context-free languages. Many interesting results have been obtained, both for application in the theory of algorithmic languages and more theoretical results.

The book of Ginsburg under review is the first publication to present these results in a systematic way. It is divided into six chapters which we shall describe in order.

Chapter one, *Context-free and ALGOL-like languages*, contains formal definitions of phrase-structure grammar, context-free grammar, and ALGOL-like language. It is proved that the class of all c.f. languages is equal to the class of all ALGOL-like languages. Then the notions of unambiguous grammar and ε-free grammar are introduced and it is proved that for every c.f. grammar there exists an equivalent ε-free grammar.

Chapter two, *Acceptors and languages*, concerns different types of abstract automata which can be used to describe languages. Thus, finite-state automata and regular languages are briefly discussed and the notion of a pushdown automaton is introduced in a formal way. It is proved that the class of all c.f. languages is identical with the class of all languages definable by pushdown automata. The notion of deterministic language is also introduced and discussed.

Chapter three, *Operations*, is devoted to the study of several types of operations on languages. The two-argument operations, intersection and difference, are discussed, as well as the one-argument operations which are mappings realized by certain types of transducers (sequential transducers, generalized sequential machines, and pushdown transducers). Several theorems are proved concerning properties of languages which are preserved in transductions. Next, a homomorphism on languages is defined and a very important theorem concerning the homomorphic characterization of c.f. languages is proved.

Chapter four, *Decidability*, contains theorems about the decidability and undecidability of the most important properties of c.f. languages.

Chapter five, *Bounded languages*, is devoted to the discussion of those c.f. languages L for which there exist words $x_1, \cdots, x_n$ such that $L \subseteq \{x_1\}^*\{x_2\}^* \cdots \{x_n\}^*$. As it is shown, many properties undecidable for c.f. languages become decidable for bounded c.f. languages. An interesting algebraic discussion of bounded languages is given.

Chapter six, *Inherent ambiguity*, contains a study of c.f. languages for which there does not exist an unambiguous grammar. The notions of ambiguity and unambiguity are of a special interest for the theory of algorithmic languages. Unfortunately, not many results are known in this field.

Each of the six chapters described is complemented with a large list of exercises, open problems, and research problems. Full historical references are also given, so that every result discussed in the book can easily be found in its original version. ANDRZEJ BLIKLE

SEYMOUR GINSBURG and JOSEPH ULLIAN. *Ambiguity in context free languages.* ***Journal of the Association for Computing Machinery,*** vol. 13 (1966), pp. 62–89.

SEYMOUR GINSBURG and JOSEPH ULLIAN. *Preservation of unambiguity and inherent ambiguity in context-free languages.* Ibid., pp. 364–368.

THOMAS N. HIBBARD and JOSEPH ULLIAN. *The independence of inherent ambiguity from complementedness among context-free languages.* Ibid., pp. 588–593.

These three papers considerably extend our understanding of inherent ambiguity (= ambiguity) in context-free languages (= languages). All of the results of these papers are now presented in or given as exercises in Chapter 6 of Ginsburg's book reviewed above.

The major results of the first paper are that the problem of determining whether an arbitrary language is ambiguous is recursively unsolvable, that however there is a decision procedure for determining whether arbitrary bounded grammars are ambiguous, and a statement of the algebraic condition for which languages in $w_1^* \cdots w_n^*$ are ambiguous. From the latter it is shown that no language in $w_1^*w_2^*$ is ambiguous. The class of bounded languages is thus the first (and so far only) interesting subfamily of languages for which a solution to the problem of ambiguity has been found.

In the second paper, a variety of results regarding the preservation of ambiguity and unambiguity under transduction are given. Unambiguity turns out to be easily preserved, but that ambiguity is in general only preserved by a transducer which is one-one on the universal language over the letters of the ambiguous language. The product of a language and a word preserves both unambiguity and ambiguity, but neither property is preserved under the language-preserving operations of product of a two-element set, transduction by a one-state complete sequential machine, or extraction of initial strings or subwords.

In the third paper, an ambiguous language whose complement is a language is exhibited, along with an unambiguous language whose complement is not a language. The construction of these languages is highly ingenious.

D. TERENCE LANGENDOEN

SEYMOUR GINSBURG and SHEILA GREIBACH. *Deterministic context free languages.* ***Information and control,*** vol. 9 (1966), pp. 620–648.

The large number of results concerning deterministic context-free languages contained in this paper increase our understanding of these languages to a stage commensurate in many respects with our understanding of the much more widely studied context-free (CF) languages. Recall that CF languages may be characterized as the languages accepted by pushdown store automata (Chomsky, XXXIII 299(2)); then deterministic CF languages are the languages accepted by pushdown automata (pda) which have no choice of moves when accepting an input tape. In I the authors show that regular languages and Dyck languages are deterministic (CF languages). In II they demonstrate via a fairly involved series of lemmas that the deterministic languages are closed under complementation but not under union or intersection, and then go on to show in another long proof that deterministic languages are unambiguous (where ambiguity is a property defined in terms of the grammars that generate the languages, not in terms of the pda that accept them).

The bulk of the results of the paper are contained in III, where various operations are discussed which preserve CF languages or deterministic languages or both. (An operation 0 preserves CF languages if $0(L)$ is CF when L is, and similarly for preservation of deterministic languages.) To summarize, let Σ be a finite alphabet, L a CF language over Σ, and R a regular language over Σ; Σ^* stands for the set of words over Σ. Consider the following operations: $S^{-1}(L) = \{w/\ S(w) \in L\}$, where S is a generalized sequential machine (Ginsburg, XXXIII 300); $f_c(L) = \{u \in (\Sigma - \{c\})^*/\ ucv \in L$ for some v in $\Sigma^*\}$; $L/R = \{u/uy \in L$ for some $y \in R\}$; $\text{Init}(L) = \{u/\ uv \in L$ for some v in $\Sigma^*\}$; $g(L) = \{w/\ \text{Init}(w) \in L\}$; $\text{Div}(R, L) = \{u/\ uR \subseteq L\}$; $\text{Min}(L) = \{y \in L/\ x \notin L$ if $x < y\}$; $\text{Max}(L) = \{y \in L/\ x \notin L$ if $y < x\}$ (where $x < y$ if $y = xz$ for some non-empty string z); $L^r = \{u_t \cdots u_1/\ u_1 \cdots u_t \in L\}$; and $L^* = \{x_1 \cdots x_n/\ x_i \in L\}$. Then, for any regular language R, the following operations preserve both CF languages and deterministic languages: $L \cup R$, $L \cap R$, $L - R$, $R - L$, $S^{-1}(L)$, LR, $f_c(L)$, L/R, and $\text{Init}(L)$. On the other hand, $\text{Div}(L, R)$, $\text{Min}(L)$, $\text{Max}(L)$, and $g(L)$ preserve deterministic languages but not CF languages in general, while RL, L^r, and L^* preserve CF languages but not deterministic languages. Some simple languages shown not to be deterministic in III and IV are $\{ww^r/\ w \in \Sigma^*$ and Σ contains at least two elements$\}$, $\{a^ib^ia^j\} \cup \{a^ib^ja^j\}$, and $\{a^nb^n\} \cup \{a^nb^{2n}\}$. Finally, in V, dealing with questions of solvability, it is shown that it is recursively solvable to determine for an arbitrary deterministic language L and an arbitrary regular set R whether $L = R$, though for an arbitrary language L (even over the set $\{a, b\}$) it is recursively unsolvable to determine whether L is deterministic.

RICHARD STANLEY

SHEILA A. GREIBACH. *A note on pushdown store automata and regular systems.* ***Proceedings of the American Mathematical Society,*** vol. 18 (1967), pp. 263–268.

The author shows that the set of words produced by a regular canonical system (Büchi XXXI 265) is regular, and, as a corollary, that the set of words accepted by a regular canonical