ON THE EQUIVALENCE AND CONTAINMENT PROBLEMS FOR CONTEXT-FREE LANGUAGES

TR68-19

by John Hopcroft

On the Equivalence and Containment Problems for Context-Free Languages

by

J. E. HOPCROFT

Cornell University

ABSTRACT

Let G and G_0 be context-free grammars. Necessary and sufficient conditions on G_0 are obtained for the decidability of $L(G_0) \subseteq L(G)$. It is also shown that it is undecidable for which G_0, $L(G) \subseteq L(G_0)$ is decidable. Furthermore, given that $L(G) \subseteq L(G_0)$ is decidable for a fixed G_0, there is no effective procedure to determine the algorithm which decides $L(G) \subseteq L(G_0)$. If $L(G_0)$ is a regular set, $L(G) = L(G_0)$ is decidable if and only if $L(G_0)$ is bounded. However, there exist non-regular, unbounded $L(G_0)$ for which $L(G) = L(G_0)$ is decidable.

1. Introduction

The problem of deciding whether two context-free grammars generate the same language is referred to as the equivalence problem for context-free languages, and it is well known that this is a recursively undecidable problem. On the other hand, if one of the grammars is fixed, then the problem of deciding whether an arbitrary grammar generates the same language as the fixed grammar can be either decidable or undecidable depending on the fixed grammar. For example, the equivalence problem is undecidable if the fixed grammar generates the set Σ^* of all finite-length strings of terminal symbols, but it is decidable if the fixed grammar generates a finite set.

A similar situation exists if one considers decidability of containment of one context-free language within another. This suggests that an attempt to characterize the context-free languages with a decidable equivalence problem and the context-free languages with a decidable containment problem may lend more insight into the structure of context-free languages than the fact that the containment and equivalence problems in general are undecidable.

In this paper it is shown that for fixed G_0, $L(G_0) \subseteq L(G)$ is decidable if and only if $L(G_0)$ is a bounded language, and that it is undecidable for which G_0, $L(G) \subseteq L(G_0)$ is decidable. Furthermore, there is no partial algorithm to determine which algorithm decides $L(G) \subseteq L(G_0)$ for those G_0 for which it is decidable. That is, even if a "birdie" tells us that for a certain G_0, $L(G) \subseteq L(G_0)$ is decidable, there still is no effective procedure to determine which algorithm to use. This does not imply that there is no "nice" characterization of the class of

MATHEMATICAL SYSTEMS THEORY, Vol. 3, No. 2.
Published by Springer-Verlag New York Inc.

G_0 such that $L(G) \subseteq L(G_0)$ is decidable, and what such a characterization might be is an interesting open question. Whatever the class, it is undecidable whether an arbitrary context-free grammar generates a language in the class. Note that there are many classes with this latter property, e.g., the deterministic context-free languages.

We also consider the equivalence problem and obtain the following partial results. If $L(G_0)$ is regular, then $L(G_0) = L(G)$ is decidable if and only if $L(G_0)$ is bounded. However, there exist context-free languages which are neither regular nor bounded but for which equivalence is decidable.

2. Definitions and Notations

In this section we recall some of the basic definitions and notation used in discussing context-free grammars, finite automata and Turing machines.

A *context-free grammar* (cfg) G is a system (V_N, V_T, P, S), where V_N and V_T are finite sets (of *variables* and *terminals*), P is a finite set (of *productions*) of the form $A \to \alpha$, A is in V_N, α is in $(V_N \cup V_T)^*$, and S (the *start symbol*) is in V_N.

If $A \to \beta$ is in P, then for each α_1 and α_2 in $(V_N \cup V_T)^*$, we write $\alpha_1 A \alpha_2 \Rightarrow \alpha_1 \beta \alpha_2$. If $\alpha_1 \Rightarrow \alpha_2, \alpha_2 \Rightarrow \alpha_3, \cdots, \alpha_{n-1} \Rightarrow \alpha_n$, then we write $\alpha_1 \Rightarrow^* \alpha_n$. The language generated by G, denoted by $L(G)$, is $\{x|\ x \text{ in } V_T^*,\ S \Rightarrow^* x\}$.

A finite automaton (fa) M is a system $(K, \Sigma, \delta, q_0, F)$, where K and Σ are finite sets (of *states* and *inputs* respectively), $\delta\colon K \times \Sigma \to K$, q_0 is in K (the *start* state) and $F \subseteq K$ (the set of *final* states). We extend δ to $K \times \Sigma^*$ as follows. For each q in K, a in Σ and x in Σ^*, $\delta(q, \epsilon) = q$ and $\delta(q, xa) = \delta(\delta(q, x), a)$. The language accepted by M, denoted by $T(M)$, is the set $\{x|\ x \text{ in } \Sigma^*,\ \delta(q_0, x) \text{ in } F\}$. A set is *regular* if it is the set accepted by some finite automaton.

For $L \subseteq \Sigma^*$, L is *bounded* if there exist $w_1, w_2, \cdots, w_n$ in Σ^* such that $L \subseteq w_1^* w_2^* \cdots w_n^*$.

A Turing machine is a system $\{K, \Sigma, \delta, q_0\}$, where K is a finite set (of *states*), Σ is a finite set (of *tape* symbols) which always contains the blank symbol B, $\delta\colon K \times \Sigma \to K \times (\Sigma - \{B\}) \times \{L, R\}$ and q_0 is in K (the *start* state). If $\delta(q, a) = (p, b, L)$, then for each x_1 and x_2 in Σ^* and each c in Σ, we write $x_1 cqax_2 \vdash x_1 pcbx_2$. If $\delta(q, a) = (p, b, R)$, then for each x_1 and x_2 in Σ^* we write $x_1 qax_2 \vdash x_1 bpx_2$. Furthermore, we write $x_1 q \vdash x_1 qB$. If $\alpha_1 \vdash \alpha_2, \alpha_2 \vdash \alpha_3, \cdots, \alpha_{n-1} \vdash \alpha_n$, we write $\alpha_1 \vdash^* \alpha_n$. The language accepted by M, denoted by $T(M)$, is the set $\{x|\ x \text{ in } (\Sigma - \{B\})^*,\ q_0 x \vdash^* \alpha, \text{ and for no } \alpha' \text{ does } \alpha \vdash \alpha'\}$. If $q_0 x \vdash \alpha_1$, $\alpha_1 \vdash \alpha_2, \cdots, \alpha_{n-1} \vdash \alpha_n$ for x in $(\Sigma - \{B\})^*$ and if for no α does $\alpha_n \vdash \alpha$, then the string $q_0 x \# \alpha_1 \# \alpha_2 \# \cdots \# \alpha_{n-1} \# \alpha_n$ is said to be a *valid computation* of M. The set of *invalid computations* of M is the complement with respect to $(\Sigma \cup \{\#\})^*$ of the set of valid computations of M.

3. Results

The first result is concerned with the class of grammars G_0 such that $L(G) \subseteq L(G_0)$ is decidable.

THEOREM 3.1. *It is undecidable for an arbitrary context-free grammar G_0 whether the containment for fixed G_0, $L(G) \subseteq L(G_0)$, is decidable for arbitrary G.*

Proof. Let M be a Turing machine which accepts a nonrecursive set. Given an arbitrary Turing machine M_i', we can effectively construct M_i such that $T(M_i) = T(M)$ if M_i' halts on ϵ input, and $T(M_i) = \varnothing$ if M_i' does not halt on ϵ input. Let L_{M_i} be the set of all invalid computations of M_i. It is known that L_{M_i} is a *cfℓ* for each M_i. For arbitrary cfg G, if $T(M_i) = \varnothing$ and $L_{M_i} = \Sigma^*$, then $L(G) \subseteq L_{M_i}$ is decidable. If $T(M_i) \neq \varnothing$, then $T(M_i)$ is a nonrecursive set. Now, if $L(G) \subseteq L_{M_i}$ is decidable, then $\overline{T(M_i)}$ could be enumerated as follows. Let $x_1, x_2, \cdots$ be an enumeration of Σ^*. For each x_1 we can effectively construct a cfg G_i generating $q_0 x_i \# \Sigma^*$. Now x_i is in $\overline{T(M_i)}$ if and only if $q_0 x_i \# \Sigma^* \subseteq L_{M_i}$. Thus if $T(M_i) \neq \varnothing$, then $L(G) \subseteq L_{M_i}$ must be undecidable for arbitrary cfg G. Since $T(M_i) = \varnothing$ if and only if M_i' halts on ϵ input, and since M_i' halting on ϵ input is undecidable, the theorem follows.

Theorem 3.1 suggests the following question. Is there a simple property of those context-free languages L such that $L(G) \subseteq L$ is decidable?

We note in passing that the result in Theorem 3.1 could also be obtained from the following theorem of Greibach [1]. Let P be a nontrivial property on the context-free languages preserved by inverse gsm, union with $\{\epsilon\}$ and intersection with regular sets. If P is true for all regular sets, then P is undecidable. However, it appears to be as difficult to show that the property P defined by "$P(L) = 1$ if and only if $L(G) \subseteq L$ is decidable" is nontrivial as it is to establish the result directly.

The next result shows that even for those G_0 for which we know $L(G) \subseteq L(G_0)$ is decidable, we still may not be able to decide which algorithm to use.

THEOREM 3.2. *Given that $L(G) \subseteq L(G_0)$ is decidable for some fixed G_0, there is no effective procedure to determine the appropriate algorithm.*

Proof. Given an arbitrary Turing machine M_i', we can effectively construct M_i such that $T(M_i) = \{\epsilon\}$ if M_i' halts on ϵ input, and $T(M_i) = \varnothing$ if M_i' does not halt on ϵ input. Let L_{M_i} be the set of invalid computations of M_i. Now L_{M_i} is either Σ^* or Σ^* with one sentence deleted. In either case $L(G) \subseteq L_{M_i}$ is decidable. However, if we could determine which algorithm to use, we could determine whether M_i' halts on ϵ input.

It is known [2] that $L(G_0) \subseteq L(G)$ is decidable for arbitrary cfg G if $L(G_0)$ is a bounded cfℓ. The next result shows that $L(G_0)$ bounded is both necessary and sufficient for the decidability of $L(G_0) \subseteq L(G)$ for arbitrary cfg G.

First we prove the following technical lemma.

LEMMA 3.1. *Let $M = (M, \Sigma, \delta, q_0, F)$ be a finite automaton. Then either (1) $T(M)$ is bounded, or (2) there exist a and b in Σ, $a \neq b$, x_1, x_2, x_3 and x_4 in Σ^* and p in K such that $\delta(q_0, x_1) = p$, $\delta(p, ax_2) = p$, $\delta(p, bx_3) = p$ and $\delta(p, x_4)$ is in F.*

Proof. Assume that the lemma is true for all finite automata of k states or fewer. Let $M = (K, \Sigma, \delta, q_0, F)$ be a finite automaton with $k+1$ states which accepts a nonempty set. Assume that Condition 2 of the lemma is not satisfied. Then for at most one a in Σ does there exist an x in Σ^* such that $\delta(q_0, ax) = q_0$. Furthermore, ax is unique if we require that $\delta(q_0, y) = q_0$ for no initial segment y. For each a_i in Σ, let M_{a_i} be the k state finite automaton obtained from M by deleting the state q_0 and all transitions involving q_0, and using $\delta(q_0, a_i)$ as the start state. Now, $T(M) = (ax)^* (\bigcup_{a_i \text{ in } \Sigma} a_i T(M_{a_i}))$ if q_0 is not in F, and $(ax)^* (\bigcup_{a_i \text{ in } \Sigma} a_i T(M_{a_i})) \cup (ax)^*$ if q_0 is in F. By the induction hypothesis, $T(M_{a_i})$ is a bounded language for each a_i in Σ. But a finite number of unions and products of bounded languages is a bounded language.

THEOREM 3.3. *For a fixed cfℓ L and an arbitrary cfg $G, L \subseteq L(G)$ is decidable if and only if L is bounded.*

Proof. The "if" portion has already been established [2]. Thus we need only consider the "only if." Assume that L is not bounded. Let $G' = (V_N, V_T, P, S)$ be a cfg in Chomsky normal form generating L. Without loss of generality, assume that there exists an A in V_N such that $A \Rightarrow^* x_1 A x_3$ and $A \Rightarrow^* x_2 A x_4$, where $\{x_1, x_2\}^*$ is not bounded. (To see this, assume for all grammars in Chomsky normal form with k or fewer nonterminals that either the language generated is bounded, or there exists a nonterminal A such that $A \Rightarrow^* x_1 A x_3$ and $A \Rightarrow^* x_2 A x_4$, where either $\{x_1, x_2\}^*$ or $\{x_3, x_4\}^*$ is not bounded. The assumption is trivially true for $k = 1$. Let $G = (V_N, V_T, P, S)$ be a cfg in Chomsky normal form with $k+1$ nonterminals. For each A in V_N and x_i in Σ^*, $1 \le i \le 4$, such that $A \Rightarrow^* x_1 A x_3$ and $A \Rightarrow^* x_2 A x_4$, assume that $\{x_1, x_2\}^*$ and $\{x_3, x_4\}^*$ are bounded. Each derivation is of the form $S \Rightarrow^* x_1 S x_4 \Rightarrow x_1 A B x_4 \Rightarrow^* x_1 x_2 x_3 x_4$ or $S \Rightarrow^* x_1 S x_3 \Rightarrow x_1 x_2 x_3$, where $A \Rightarrow^* x_2$, $B \Rightarrow^* x_3$ and S does not appear in the derivations of x_2 and x_3 from A and B. Let $L_1 = \{x_1 \mid S \Rightarrow^* x_1 S x_3\}$; L_1 is regular and $L_1 = L_1^*$. By an argument similar to that used in Lemma 3.1, either L_1 is bounded or there exist y_1 and y_2 in $\{x_1 \mid S \Rightarrow^* x_1 S x_3\}$ such that $\{y_1, y_2\}^*$ is not bounded; similarly for $L_2 = \{x_3 \mid S \Rightarrow^* x_1 S x_3\}$. By the inductive hypothesis, $L_A = \{x_2 \mid A \Rightarrow^* x_2\}$ and $L_B = \{x_3 \mid B \Rightarrow^* x_3\}$ are bounded. Let $L_{AB} = L_1 L_A L_B L_2$. Now $L(G)$ is contained in the union of $L_1 \{x_2 \mid S \to x_2\} L_2$ with the union of L_{AB} over all A and B in $V_N - \{S\}$. Thus $L(G)$ is bounded.)

Since $\{x_1, x_2\}^*$ is not bounded, by Lemma 3.1 we can write $\{x_1, x_2\}^* = z_4 \{a z_1, b z_2\}^* z_3$, where $a \neq b$. Let z_5, z_6 and z_7 be such that $S \Rightarrow^* z_5 A z_7$ and $A \Rightarrow^* z_6$. Let $R = z_5 z_4 \{a z_1 a z_1, a z_1 b z_2\}^* b z_2 a z_2 z_3 z_6 \{x_3, x_4\}^* z_7$. Let h be the homomorphism of $\{0, 1\}^*$ into Σ^* defined by $h(0) = a z_1 a z_1$ and $h(1) = a z_1 b z_2$. Now given a cfg G_1, there is an effective procedure for constructing a cfg G_2 generating $z_5 z_4 h(L(G_1)) b z_2 a z_2 z_3 z_6 \{x_3, x_4\}^* z_7 \cup \bar{R}$. Now if $L(G_1) = \{0, 1\}^*$, then $L(G_2) = \Sigma^*$ and $L \subseteq L(G_2)$. If $L(G_1) \neq \{0, 1\}^*$, then $L(G_2)$ does not contain R and thus L is not contained in $L(G_2)$. Since $L(G_1) = \{0, 1\}^*$ is undecidable and since $L \subseteq L(G_2)$ if and only if $L(G_1) = \{0, 1\}^*$, $L \subseteq L(G_2)$ is undecidable.

We now consider the equivalence problem. Again, it is known [2] that $L(G_0) = L(G)$ is decidable for arbitrary cfg G if $L(G_0)$ is bounded. We shall now show that if $L(G_0)$ is regular, then $L(G_0) = L(G)$ is decidable if and only if $L(G_0)$ is bounded. Finally we shall show that there exists a non-bounded, non-regular cfℓ $L(G_0)$ such that $L(G_0) = L(G)$ is decidable for arbitrary cfg G.

LEMMA 3.2. *Let R be a regular set which is not bounded. Then for an arbitrary context-free grammar G, $L(G) = R$ is undecidable.*

Proof. Let $M = (K, \Sigma, \delta, q_0, F)$ be the minimum state finite automaton accepting R. By Lemma 3.1 there exist a and b in Σ with $a \neq b$, x and y in Σ^* and p in K such that $\delta(p, ax) = p$ and $\delta(p, by) = p$. Furthermore, there exist w

and z in Σ^* such that $\delta(q_0, w) = p$ and $\delta(p, z)$ in F. Let $R_1 = \{w\}\ \{ax, by\}^*\{z\}$; now $R_1 \subseteq R$. Thus $R = R_1 \cup (R \cap \bar{R}_1)$. Let $G = (V_N, \{0, 1\}, P, S)$ be an arbitrary context-free grammar with terminal symbols 0 and 1 and let h be the homomorphism of $\{0, 1\}^*$ into Σ^* defined by $h(0) = ax$ and $h(1) = by$. From G we can effectively construct a context-free grammar G' generating $\{w\}h(L(G))\{z\} \cup (R \cap \bar{R}_1)$. Now $L(G') = R$ if and only if $L(G) = \{0, 1\}^*$. Thus if $L(G') = R$ is decidable, then $L(G) = \{0, 1\}^*$ is decidable. But $L(G) = \{0, 1\}^*$ is undecidable and thus $L(G') = R$ is undecidable.

In [2] it was shown that for any bounded context-free language L and arbitrary context-free grammar G, $L(G) \subseteq L$ and $L \subseteq L(G)$ are decidable and therefore $L(G) = L$ is decidable. We state the following theorem.

THEOREM 3.4. *For a fixed regular set R and arbitrary context-free grammar G, $L(G) = R$ is decidable if and only if R is bounded.*

From the above theorem, one might suspect that for a fixed context-free grammar G_0 and arbitrary context-free grammar G, $L(G_0) = L(G)$ is decidable if and only if $L(G_0)$ is bounded. However, the following theorem shows that this is not true.

THEOREM 3.5. *For an arbitrary context-free grammar G, $L(G) = \{w \# w^R \mid w$ in $\{0, 1\}^*\}$ is decidable.*

Proof. Let $G = (V_N, V_T, P, S)$ be a context-free grammar. Without loss of generality, assume that for each A in V_N, $A \neq S$, there exist x_1, x_2, x_3, x_4 and x_5 in V_T^*, x_4 and x_5 not both ϵ, such that

$$\text{(i) } S \Rightarrow^* x_1 A x_5,$$
$$\text{(ii) } A \Rightarrow^* x_2 A x_4,$$
$$\text{(iii) } A \Rightarrow^* x_3.$$

Now for each A in V_N and x_3 in V_T^* such that $A \Rightarrow^* x_3$, x_3 is in $(V_T - \{\#\})^*\{\#\}(V_T - \{\#\})^*$, for otherwise a sentence not in $L(G)$ could be generated. (Clearly if x_3 contains two or more $\#$'s, then a sentence not in $L(G)$ could be generated. If x_3 is in $(\Sigma - \{\#\})^*$, then $S \Rightarrow^* x_1 A x_5 \Rightarrow^* x_1 x_2^n x_3 x_4^n x_5$, $n \geq 0$, where either x_1, x_2, x_4 or x_5 contains $\#$. But then $x_1 x_2^n x_3 x_4^n x_5$ can be in $L(G)$ for at most one value of n.) Thus at most one nonterminal can appear in any line of a derivation, which implies that each production in P must be of the form $A \to t$ or $A \to t_1 B t_2$, with A and B in V_N, t in $\{0, 1\}^* \# \{0, 1\}^*$ and t_1 and t_2 in $\{0, 1\}^*$.

For each A in V_N, find an x (call it x_A), such that $A \Rightarrow^* x_A$. Now x_A must be of the form $x_1 \# x_1^R x_2^R$ or $x_2 x_1 \# x_1^R$, where x_1 and x_2 are in $\{0, 1\}^*$. (Note that if $x_A = x_1 \# x_1^R x_2^R$, then A can appear only in sentential forms of the format $x_3 x_2 A x_3^R$, for otherwise a sentence not in L could be generated.) Now $L(G) \subseteq L$ if and only if

(i) for each production $A \to t$, $t = x_3 \# x_3^R x_2^R$ if $x_A = x_1 \# x_1^R x_2^R$ and $t = x_2 x_3 \# x_3^R$ if $x_A = x_2 x_1 \# x_1^R$ for some x_3 in $\{0, 1\}^*$;
(ii) for each production $A \to t_1 B t_2$, $t_1 x_B t_2 = x_3 \# x_3^R x_2^R$ if $x_A = x_1 \# x_1^R x_2^R$, and $t_1 x_B t_2 = x_2 x_3 \# x_3^R$ if $x_A = x_2 x_1 \# x_1^R$ for some x_3 in $\{0, 1\}^*$.

To determine whether $L \subseteq L(G)$, consider the grammar $G' = (V_N, V_T, P', S)$, where $A \to yB$ is in P' if $A \to yBy'$ is in P, and $A \to y$ is in P' if $A \to y \# y'$ is in

P. Given that $L(G) \subseteq L$, $L \subseteq L(G)$ if and only if $L(G') = \{0, 1\}^*$. But there exists an effective procedure for finding a finite automaton which accepts $L(G')$, and thus for determining whether $L(G') = \{0, 1\}^*$. Hence $L = L(G)$ is decidable.

REFERENCES

[1] S. GREIBACH, A note on undecidable properties of formal languages. SDC Document TM-738/038/00, August 1967.

[2] S. GINSBURG and E. H. SPANIER, Bounded ALGOL-like languages, *Trans. Amer. Math. Soc.* **113** (1964), 333–368.

(*Received 21 April 1968*)