The Inclusion Problem for Some Subclasses of Context-Free Languages

Peter R.J. Asveld and Anton Nijholt

Department of Computer Science, Twente University of Technology
P.O. Box 217, 7500 AE Enschede, The Netherlands

Abstract – By a reduction to Post's Correspondence Problem we provide a direct proof of the known fact that the inclusion problem for unambiguous context-free grammars is undecidable. The argument or some straightforward modification also applies to some other subclasses of context-free languages such as linear languages, sequential languages, and DSC-languages (i.e., languages generated by context-free grammars with disjunct syntactic categories). We also consider instances of the problem "Is $L(D_1) \subseteq L(D_2)$?" where D_1 and D_2 are taken from possibly different descriptor families of subclasses of context-free languages.

Keywords: inclusion problem, containment problem, unambiguous context-free grammar, linear grammar, sequential grammar, LL(k)-grammar, nonterminal separating or NTS-grammar, context-free grammar with disjunct syntactic categories (DSC-grammar), deterministic PDA (push-down automaton), real-time strict deterministic PDA, simple deterministic PDA, superdeterministic PDA.

When we change a context-free language by e.g. modifying its grammar G_1 into a new context-free grammar G_2, the obvious questions are: What is the relationship between $L(G_1)$ and $L(G_2)$? Are they equal? Is one language a (proper) subset of the other one? Are these two languages incomparable? In answering questions of this type the (decidability of the) inclusion problem plays a principal part.

Consider two descriptors $\mathbf{D}_1$ and $\mathbf{D}_2$ of subclasses of context-free languages; e.g., $\mathbf{D}_1$ and $\mathbf{D}_2$ are two particular kinds of context-free grammars or push-down automata. Then the *inclusion* or *containment problem* for $(\mathbf{D}_1, \mathbf{D}_2)$ is the question whether for arbitrary $D_1 \in \mathbf{D}_1$ and $D_2 \in \mathbf{D}_2$ the inclusion $L(D_1) \subseteq L(D_2)$ holds. In case $\mathbf{D}_1 = \mathbf{D}_2$ we refer to this problem as the *inclusion problem* for $\mathbf{D}_1$.

It is well known that the inclusion problem for regular languages is decidable, whereas it is undecidable for context-free languages; cf., e.g., [4, 6, 8]. In [2] it has been established that the inclusion problem for simple deterministic languages is undecidable. Clearly, this result implies the undecidability of the inclusion problem for unambiguous context-free languages.

In this note we provide an alternative, direct proof of this latter fact (Theorem 1) which consists of a reduction to Post's Correspondence Problem over two-letter alphabets. As a consequence of this proof we obtain the undecidability of the inclusion problem for linear and sequential languages (Corollary 2). A slight modification of the argument yields the undecidability of the inclusion problem for context-free grammars with disjunct syntactic categories (Theorem 3). This result also follows from the undecidability of the inclusion problem for NTS (or nonterminal separating) languages established in [9]. Finally, we consider some consequences for inclusion problems of the form $(\mathbf{D}_1, \mathbf{D}_2)$ with $\mathbf{D}_1 \neq \mathbf{D}_2$ (Theorem 5 and Table 1), and we survey the open problems in the area (Table 1).

The emphasis in this note is on the application of the proof technique used in establishing Theorem 1 and on surveying results with respect to the inclusion problem rather than deriving new results. Actually, only Corollary 2 and its consequences (see Table 1), a minor part of Theorem 5, and the proofs of Theorems 1 and 3 seem to be new.

Theorem 1. *Let G_1 and G_2 be unambiguous context-free grammars. Then the problem "Is $L(G_1) \subseteq L(G_2)$?" is undecidable.*

Proof: Let I be an instance of Post's Correspondence Problem (PCP) over a two-letter alphabet, i.e., $I = (\alpha_1, \cdots, \alpha_n; \beta_1, \cdots, \beta_n)$ with $\alpha_i, \beta_i \in \Delta^+$ $(1 \leq i \leq n)$ and $\Delta = \{a, b\}$. Let Θ be an alphabet of n new symbols, say $\Theta = \{a_1, \cdots, a_n\}$, and define the homomorphism $h : (\Theta \cup \Delta)^* \to \Delta^*$ by

$$\begin{aligned} h(a_i) &= \lambda \qquad \text{for all } i\ (1 \leq i \leq n) \\ h(a) &= a \\ h(b) &= b \end{aligned}$$

(λ denotes the empty word). Consider the context-free grammar $G_I = (V, \Sigma, P_I, S)$ with $\Sigma = \Theta \cup \Delta \cup \{c\}$, $V = \Sigma \cup \{S\}$ and P_I consists of the productions

$$\begin{aligned} S &\to a_i\, \alpha_i\, S\, \beta_i^R \qquad \text{for all } i\ (1 \leq i \leq n), \\ S &\to a_i\, \alpha_i\, c\, \beta_i^R \qquad \text{for all } i\ (1 \leq i \leq n), \end{aligned}$$

where R is the reversal or mirror operation. Then we have

$$L(G_I) = \{a_{i_1}\alpha_{i_1} \cdots a_{i_k}\alpha_{i_k} c\, (\beta_{i_1} \cdots \beta_{i_k})^R \mid k \geq 1,\ 1 \leq i_j \leq n,\ \text{for all } j \text{ with } 1 \leq j \leq k\}$$

and G_I is unambiguous. Next we define the context-free grammar $G = (V_0, \Sigma, P, S)$ with alphabet $V_0 = \Sigma \cup \{S, A, B, C, D\}$ and P consisting of the productions

$$\begin{aligned} &(1) \quad S \to Aa \mid Ab \\ &(2) \quad A \to Aa \mid Ab \mid D \\ &(3) \quad S \to \xi aB \mid \xi bB \\ &(4) \quad B \to \xi aB \mid \xi bB \mid D \\ &(5) \quad D \to \xi aDa \mid \xi bDb \mid \xi aDb \mid \xi bDa \mid c \\ &(6) \quad S \to C \\ &(7) \quad C \to \xi aCa \mid \xi bCb \mid \xi aDb \mid \xi bDa \end{aligned}$$

with $\xi \in \{\lambda\} \cup \Theta$. It is easy to see that

$$L(G) = \{ wcv \mid w \in ((\Theta \cup \{\lambda\})\Delta)^+,\ v \in \Delta^+,\ h(w) \neq v^R \},$$

as well as the following fact: for each $wcv \in L(G)$ we have

(a) $|h(w)| < |v|$ if and only if wcv has been derived using the rules (1), (2) and (5) only,

(b) $|h(w)| > |v|$ if and only if wcv has been derived using the rules (3), (4) and (5) only,

(c) $|h(w)| = |v|$ if and only if wcv has been derived using the rules (5), (6) and (7) only,

where as usual $|x|$ denotes the length of the string x. Using this observation it is straightforward to show that G is unambiguous.

Now suppose the instance I has a solution. Thus there exists a sequence $a_{i_1} a_{i_2} \cdots a_{i_k}$ such that

$$h(a_{i_1}\alpha_{i_1} \cdots a_{i_k}\alpha_{i_k}) = \beta_{i_1} \cdots \beta_{i_k}.$$

This means that $L(G_I) - L(G) \neq \emptyset$, and consequently $L(G_I)$ is not included in the language $L(G)$.

Conversely, suppose $L(G_I)$ is not included in $L(G)$. Then there exists a string wcv in $L(G_I)$ with $h(w) = v^R$. But then the sequence of symbols from Θ that occur from left to right in w determines a solution for I.

Summarizing, we have that the inclusion problem for unambiguous context-free grammars is reducible to PCP. Hence it is undecidable. □

Notice that both grammars constructed in the proof are linear and sequential. Remember that a context-free grammar $G = (V, \Sigma, P, S)$ is called *sequential* [4] if $V - \Sigma$ can be provided with a linear order $\leq$ such that for each rule $A \to w$, $A \leq B$ holds for all nonterminal symbols B that occur in w. (The linear order for the grammar G in our proof is: $S \leq A \leq B \leq C \leq D$). Therefore we have

Corollary 2. *Let G_1 and G_2 be unambiguous sequential linear context-free grammars. Then the problem "Is $L(G_1) \subseteq L(G_2)$?" is undecidable.* □

Next we turn to context-free grammars that possess disjunct syntactic categories or that satisfy the NTS (nonterminal separating) property. A *DSC-grammar* or a *context-free grammar with disjunct syntactic categories* is a context-free grammar $G = (V, \Sigma, P, X)$ with $X \subseteq V - \Sigma$, such that for all $A, B \in V - \Sigma$, $A \neq B$ implies $L(G, A) \cap L(G, B) = \emptyset$, where for each A, $L(G, A) = \{w \in V^* \mid A \Rightarrow^* w\}$. The language generated by a DSC-grammar $G = (V, \Sigma, P, X)$ is defined by $L(G) = \{w \in \Sigma^* \mid A \Rightarrow^* w \text{ for some } A \in X\}$.

A context-free grammar $G = (V, \Sigma, P, X)$ with $X \subseteq V - \Sigma$ is an *NTS-grammar* (or satisfies the nonterminal separating property [1, 9]) if for all $A \in V - \Sigma$, and for all $w \in V^*$, $A \Rightarrow^* w$ holds if, and only, if $A \Leftrightarrow^* w$, where $\Leftrightarrow^*$ is the reflexive and transitive closure of the union of $\Rightarrow$ and its converse relation $\Leftarrow$. So, roughly spoken, $A \Leftrightarrow^* w$ means that w may be obtained from A by using the productions of P in both directions. The language generated by an NTS-grammar $G = (V, \Sigma, P, X)$ is defined

by $L(G) = \{w \in \Sigma^* \mid A \Rightarrow^* w \text{ for some } A \in X\}$.

Each NTS-grammar has disjunct syntactic categories [1]; but the converse does not hold.† For instance, the language $\{a^n b^n \mid n \geq 1\} \cup \{a^n b^{2n} \mid n \geq 1\}$ is not an NTS-language [1], but it is easy to show that this language can be generated by a DSC-grammar. The inclusion problem for NTS-grammars is undecidable [9], which also implies the undecidability of the inclusion problem for DSC-grammars. Here we provide a direct proof of this latter statement.

Theorem 3. *Let G_1 and G_2 be context-free grammars with disjunct syntactic categories. Then the problem "Is $L(G_1) \subseteq L(G_2)$?" is undecidable.*

Proof: We slightly change the proof of Theorem 1. First, we observe that G_I is trivially a DSC-grammar. Secondly, we replace the grammar G in that proof by $G_0 = (V_0, \Sigma, P_0, X_0)$ with $\Sigma = \Delta \cup \Theta \cup \{c\}$, $\Delta = \{a, b\}$, $V_0 = \Sigma \cup \{S, T, C, D, E\}$, $X_0 = \{S,T,C,D\}$ and P_0 consists of the productions

$$
\begin{aligned}
S &\rightarrow \xi aS \mid \xi bS \mid \xi aC \mid \xi bC \mid \xi aD \mid \xi bD \mid \xi aE \mid \xi bE \mid \xi ac \mid \xi bc \\
T &\rightarrow Ta \mid Tb \mid Ca \mid Cb \mid Da \mid Db \mid Ea \mid Eb \mid ca \mid cb \\
C &\rightarrow \xi aCa \mid \xi bCb \mid \xi aDa \mid \xi bDb \\
D &\rightarrow \xi aDb \mid \xi bDa \mid \xi aCb \mid \xi bCa \mid \xi aEb \mid \xi bEa \\
E &\rightarrow \xi aEa \mid \xi bEb \mid c
\end{aligned}
$$

with $\xi \in \{\lambda\} \cup \Theta$. Then it is easy to see that

$$
\begin{aligned}
L(G_0,S) &= \{wcv \mid w \in ((\Theta \cup \{\lambda\})\Delta)^*, v \in \Delta^*, |h(w)| > |v|\}, \\
L(G_0,T) &= \{wcv \mid w \in ((\Theta \cup \{\lambda\})\Delta)^*, v \in \Delta^*, |h(w)| < |v|\}, \\
L(G_0,C) &= \{wcv \mid w \in ((\Theta \cup \{\lambda\})\Delta)^*, v \in \Delta^*, |h(w)| = |v|, h(w) \neq v^R, \ 1{:}h(w) = 1{:}v^R\}, \\
L(G_0,D) &= \{wcv \mid w \in ((\Theta \cup \{\lambda\})\Delta)^*, \ v \in \Delta^*, |h(w)| = |v|, \ 1{:}h(w) \neq 1{:}v^R\}, \\
L(G_0,E) &= \{wcv \mid w \in ((\Theta \cup \{\lambda\})\Delta)^*, v \in \Delta^*, \ h(w) = v^R\},
\end{aligned}
$$

where $1{:}x$ denotes the first symbol of the string x. Hence G_0 is a DSC-grammar. Moreover, it is straightforward to prove that $L(G_0) = L(G)$. □

Although G_I is an NTS-grammar, it is unlikely that this proof can be modified in order to provide an alternative way of establishing the undecidability of the inclusion problem for NTS-grammars [9]. More concretely, $L(G_0)$ is probably not an NTS-language.

We assume the reader to be familiar with the notion of deterministic PDA (push-down automaton) and restricted variants such as simple deterministic PDA and real-time strict deterministic PDA; cf. [6] for an excellent survey. However, we will recall the definition of the somewhat less well-known concept of super-deterministic PDA [3, 5].

Definition. Let $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a deterministic push-down automaton. For each rule (q, a, A, p, y) in δ, the pair (q, A) is called the *mode* of the rule

† This latter observation and the following example are due to Jan Anne Hogendorp.

with input a; if $a = \lambda$, this is a λ-rule. If no rule is defined for (q, A) in $Q \times \Gamma$, it is a *blocking mode*; (q, λ) is also called a blocking mode. A pair (q, yA) with $q \in Q$, $y \in \Gamma^*$, and $A \in \Gamma$ is a *configuration of M with mode* (q, A). A configuration (q, yA) is in *reading mode*, if no λ-rule is defined for mode (q, A), and (q, A) is not a blocking mode.

M is *super-deterministic* if it is finite delay and for all accessible configurations in reading mode (q, s_1), (q, s_2), (q_1, t_1), and (q_2, t_2) in $Q \times \Gamma^*$ and $w \in \Sigma^*$, if $(q, s_1) \vdash^w (q_1, t_1)$ and $(q, s_2) \vdash^w (q_2, t_2)$, then $q_1 = q_2$ and $|s_1| - |t_1| = |s_2| - |t_2|$. The language $T(M)$ accepted by M by final state (accept mode) is

$$T(M) = \{w \in \Sigma^* \mid (q_0, Z_0) \vdash^w (q, s) \text{ for some } q \in F \text{ and some } s \in \Gamma^*\},$$

and the language $L(M)$ accepted by M by final state (accept mode) and empty store is

$$L(M) = \{w \in \Sigma^* \mid (q_0, Z_0) \vdash^w (q, sZ_0) \text{ for some } q \in F\}.$$

A language L_0 over Σ_0 is *super-deterministic* if there is a super-deterministic push-down automaton M such that either $L_0 = T(M)$ or $L_0\$ = T(M)$ for some symbol $\$$ not in Σ_0. □

The inclusion problem for super-deterministic PDA's highly depends on the way in which a language is accepted; viz.

Theorem 4. ([5] and [3], respectively). *The inclusion problem is decidable for languages accepted by super-deterministic PDAs by final state and empty store. In case of acceptance by final state only, the inclusion problem is undecidable.* □

We conclude this note with a few consequences for inclusion problems of the form $(\mathbf{D}_1, \mathbf{D}_2)$ in which $\mathbf{D}_1$ may differ from $\mathbf{D}_2$.

Theorem 5. *Let $\mathbf{D}_1$ and $\mathbf{D}_2$ be equal to one of the following descriptors:*

- *linear context-free grammar,*
- *sequential context-free grammar,*
- *unambiguous context-free grammar,*
- *deterministic push-down automaton,*
- *context-free grammar with disjunct syntactic categories,*
- *context-free grammar.*

Then the inclusion problem for $(\mathbf{D}_1, \mathbf{D}_2)$ is undecidable. The same conclusion holds if $\mathbf{D}_1$ is taken equal to "NTS-grammar".

Proof: These statements directly follow from the proofs of the previous results and the fact that $L(G_I)$ is an NTS-language. □

It remains an open problem whether "super-deterministic push-down automaton (acceptance by final state)" can be added to the list in Theorem 5.

In Table 1 we summarize known results with respect to the inclusion problem; it also includes the cases considered in the present paper to which we refer by [0]. Of course, Table 1 may be considered as an extension of the appropriate row from Figure 14.2 on page 230 in [8]. A table similar to Table 1 surveying the equivalence

$\subseteq$	regular	linear	unambiguous	sequential	deterministic	simple deterministic	real-time strict deterministic	super-deterministic	LL(k)	NTS	DSC	context-free
regular	D[8]	?	?	?	D[8]	D[8]	D[8]	D[8]	D[8]	?	?	U[8]
linear	D[8]	U[0]	U[0]	U[0]	U[0]	?	?	U[3]	?	?	U[0]	U[8]
unambiguous	D[8]	U[0]	U[0]	U[0]	U[0]	U[2]	U[2]	U[3]	U[2]	U[9]	U[0]	U[8]
sequential	D[8]	U[0]	U[0]	U[0]	U[0]	?	?	U[3]	?	?	U[0]	U[8]
deterministic	D[8]	U[0]	U[0]	U[0]	U[8]	U[2]	U[2]	U[3]	U[2]	U[9]	U[0]	U[8]
simple deterministic	D[8]	?	U[2]	?	U[2]	U[2]	U[2]	U[3]	U[2]	?	?	U[8]
real-time strict deterministic	D[8]	?	U[2]	?	U[2]	U[2]	U[2]	U[3]	U[2]	?	?	U[8]
super-deterministic	D[8]	?	U[3]	?	U[3]	U[3]	U[3]	U[3]	U[3]	?	?	U[8]
LL(k)	D[8]	?	U[2]	?	U[2]	U[2]	U[2]	U[3]	U[2]	?	?	U[8]
NTS	D[8]	U[0]	U[9]	U[0]	U[9]	?	?	?	?	U[9]	U[9]	U[9]
DSC	D[8]	U[0]	U[9]	U[0]	U[9]	?	?	?	?	U[9]	U[9]	U[9]
context-free	D[8]	U[0]	U[0]	U[0]	U[6]	U[2]	U[2]	U[3]	U[2]	U[9]	U[9]	U[8]

D = decidable, U = undecidable, ? = open problem

Table 1.

problem for some subclasses of context-free languages can be found in [7].

Acknowledgement. We are indebted to Rieks op den Akker and Jan Anne Hogendorp for some remarks on an earlier version of this note.

References

1. L. Boasson & G. Sénizergues: NTS languages are deterministic and congruential, *J. Comput. System Sci.* **31** (1985) 332-342.
2. E.P. Friedman: The inclusion problem for simple languages, *Theor. Comput. Sci.* **1** (1976) 297-316.
3. E.P. Friedman & S.A. Greibach: Superdeterministic DPDAs: the method of accepting does affect decision problems, *J. Comput. System Sci.* **19** (1979) 79-117.

4. S. Ginsburg & H.G. Rice: Two families of languages related to ALGOL, *J. Assoc. Comput. Mach.* **9** (1962) 350-371.

5. S.A. Greibach & E.P. Friedman: Superdeterministic PDAs: a subcase with a decidable inclusion problem, *J. Assoc. Comput. Mach.* **27** (1980) 675-700.

6. M.A. Harrison: *Introduction to Formal Language Theory* (1978), Addison-Wesley, Reading, Mass.

7. M.A. Harrison, I.M. Havel & A. Yehudai: On equivalence of grammars through transformation trees, *Theor. Comput. Sci.* **9** (1979) 173-205.

8. J.E. Hopcroft & J.D. Ullman: *Formal Languages and Their Relation to Automata* (1969), Chapter 14, Addison-Wesley, Reading, Mass.

9. G. Sénizergues: The equivalence and inclusion problems for NTS languages, *J. Comput. System Sci.* **31** (1985) 303-331.