Information and Computation **142**, 182–206 (1998)
Article No. IC972695

One-Unambiguous Regular Languages*

Anne Brüggemann-Klein†

Institut für Informatik, Technische Universität München, Arcisstrasse 21, 80290 München, Germany

and

Derick Wood‡

Department of Computer Science, Hong Kong University of Science & Technology, Clear Water Bay, Kowloon, Hong Kong
E-mail: brueggem@informatik.tu-muenchen.de, dwood@cs.ust.hk

The ISO standard for the Standard Generalized Markup Language (SGML) provides a syntactic meta-language for the definition of textual markup systems. In the standard, the right-hand sides of productions are based on regular expressions, although only regular expressions that denote words unambiguously, in the sense of the ISO standard, are allowed. In general, a word that is denoted by a regular expression is witnessed by a sequence of occurrences of symbols in the regular expression that match the word. In an unambiguous regular expression as defined by Book *et al.* (1971, *IEEE Trans. Comput.* **C-20**(2), 149–153) each word has at most one witness. But the SGML standard also requires that a witness be computed incrementally from the word with a one-symbol lookahead; we call such regular expressions *1-unambiguous*. A regular language is a *1-unambiguous language* if it is denoted by some 1-unambiguous regular expression. We give a Kleene theorem for 1-unambiguous languages and characterize 1-unambiguous regular language in terms of structural properties of the minimal deterministic automata that recognize them. As a result we are able to prove the decidability of whether a given regular expression denotes a 1-unambiguous language; if it does, then we can construct an equivalent 1-unambiguous regular expression in worst-case optimal time. © 1998 Academic Press

* The work of the second author was supported under a Natural Sciences and Engineering Research Council of Canada Grant and under an Information Technology Research Centre of Ontario Grant.

† WWW: http://www11.informatik.tu-muenchen.de.

‡ WWW: http://www.cs.ust.hk/~dwood.

0890-5401/98 $25.00
Copyright © 1998 by Academic Press
All rights of reproduction in any form reserved.

1. INTRODUCTION

Document processing systems such as editors, formatters, and retrieval systems deal with many different types of documents, such as books, articles, memoranda, dictionaries, and letters. The Standard Generalized Markup Language (SGML) establishes a common platform for the syntactic specification of document types and conforming documents [ISO86, Gol90]. SGML is an ISO standard that has been endorsed by a number of publishing houses throughout North America and Europe, by the European Community, and by the U.S. Department of Defense.

Document types in SGML are defined, essentially, by bracketed, extended context-free grammars [GH67, Woo87]. The right-hand sides of productions, called *model groups*, are essentially regular expressions with two major differences. First, model groups allow three new operators ?, &, and $^+$. Second, the model groups must be *unambiguous* in the sense of Clause 11.2.4.3 of the standard. The intent of the standard is to make it easier for a human to write regular expressions that can be interpreted unambiguously. The notion of unambiguity for model groups used in the ISO standard differs from, but is related to, unambiguity as defined by Book *et al.* [BEGO71] for regular expressions. Eilenberg [Eil74] defined ambiguity for finite-state automata, rather than for regular expressions; we discuss the relationship of his notion to the one of Book *et al.* at the end of Section 2. Book *et al.* required that each word is witnessed by at most one sequence of positions of symbols in the regular expression that matches the word. For example, consider the regular expression $(a+b)^* aa^*$. If we mark different positions of the same symbol with subscripts, we get $(a_1+b_1)^* a_2 a_3^*$; now, there are three witnesses for the word *aaa*, namely $a_1 a_1 a_2$, $a_1 a_2 a_3$, and $a_2 a_3 a_3$. Thus, $(a+b)^* aa^*$ is ambiguous; however, there is an unambiguous regular expression that denotes the same language, namely $(a+b)^* a$.

Unambiguity as defined in SGML is a one-symbol-lookahead version of unambiguity as defined by Book *et al.* In Clause 11.2.4.3 of the SGML standard a model group (regular expression) is defined to be unambiguous if[1] "an element [a symbol] ... that occurs in the document instance [word] must be able to satisfy only one primitive content token [position of the symbol in the regular expression] without looking ahead in the document instance." In other words, the only valid regular expressions are those that permit us to determine uniquely which position of a symbol in a regular expression should match a symbol in an input word without looking beyond that symbol in the input word. We call such regular expressions 1-*unambiguous*.

Consider the regular expression $(a+b)^* a$ marked as $(a_1+b_1)^* a_2$. In the word *baa*, after we match symbol b with position b_1, we cannot decide whether we should match the subsequent a in the word with position a_1 or with position a_2 without looking ahead beyond the current symbol a in the word. Therefore, although $(a+b)^* a$ is unambiguous in the sense of Book *et al.*, it is not 1-unambiguous; however, $b^* a(b^* a)^*$ is a 1-unambiguous regular expression that denotes the same language as $(a+b)^* a$.

[1] The phrases in brackets are our interpretations.

For unambiguous regular expressions in the sense of Book *et al.* [BEGO71], the following results are known: Book *et al.* gave a construction that, for each regular expression E, gives a nondeterministic finite-state automaton (NFA) G_E that recognizes the language of E. They showed that E is unambiguous if and only if G_E is unambiguous. Berry and Sethi [BS86] showed that this NFA is the canonical representation of the corresponding regular expression, because it has a natural connection with the derivatives [Brz64] of the regular expression.

Regular expressions are built with the usual operators $+$, $\cdot$, and *. SGML, however, deals with model groups that may also contain the operators ?, &, and $^+$. (E? denotes $\mathsf{L}(E+\varepsilon)$, $F\&G$ denotes $\mathsf{L}(FG+GF)$, and E^+ denotes $\mathsf{L}(EE^*)$.) Whereas the transformations of E^+ into EE^* and of $F\&G$ into $FG+GF$ preserve languages, they do not preserve 1-unambiguity. For example, $(a^*+b)^+$ is 1-unambiguous, but $(a^*+b)(a^*+b)^*$ is not. Similarly, $a?\&b$ is 1-unambiguous, but $a?b+ba?$ is not. In fact, there are languages that can be denoted by a 1-unambiguous model group but not by any 1-unambiguous regular expression [BK93a]. Furthermore, 1-unambiguous model groups are exponentially more succinct than are 1-unambiguous regular expressions. For example, the smallest 1-unambiguous regular expression equivalent to the 1-unambiguous model group $a_1\&\cdots\&a_n$ has size exponential in n.

We establish the basic results for 1-unambiguous regular expressions and languages which are the basis for the results by Ahonen [AH97] for transforming an ambiguous model group into an unambiguous one by generalizing the language of the model group, the decidability results by Brüggemann-Klein [BK93a] for model groups, and the results for SGML exceptions by Kilpeläinen and Wood [KILW97]. In Section 2, after giving the basic definitions, we show that a regular expression E is 1-unambiguous if and only if G_E is a deterministic finite-state automaton (DFA). Thus, one can decide, in time linear in the size of a regular expression E, whether E is 1-unambiguous, and if it is, then one can also construct the deterministic automaton G_E in linear time [BK92a, BK93c].

We establish, in Section 3, that the family of 1-unambiguous languages is closed under derivatives and, in Section 4, we establish a Kleene characterization of the family of 1-unambiguous languages. An analogous Kleene characterization for model groups still eludes us.

In Section 5, we present the main result of the paper—a characterization of the 1-unambiguous languages in terms of their minimal deterministic finite-state automata. As one application of this result we prove that the family of 1-unambiguous languages forms a proper subfamily of the regular languages, in contrast to the result of Book *et al.* that each regular language is denoted by some unambiguous regular expression. The characterization yields a decision algorithm for 1-unambiguous languages. Moreover, if a regular language is 1-unambiguous, then we can construct an equivalent 1-unambiguous regular expression. The decision algorithm runs in time quadratic in the size of the minimal deterministic finite-state automaton for the given language. The 1-unambiguous regular expression that we construct from its minimal deterministic finite-state automaton M can have size exponential in the size of M, which is worst-case optimal.

2. UNAMBIGUOUS REGULAR EXPRESSIONS

Let Σ be an alphabet of symbols. Regular expressions over Σ are built from ε, $\varnothing$, and symbols in Σ using the binary operators $+$ and $\cdot$ and the unary operator *. The language specified by a regular expression E is denoted by $\mathsf{L}(E)$. The symbols that occur in a regular expression E are denoted by $sym(E)$.

To indicate different positions of the same symbol in a regular expression, we mark symbols with subscripts. For example, $(a_1 + b_1)^* a_2(a_3 b_2)^*$ and $(a_4 + b_2)^* a_1(a_5 b_1)^*$ are both *markings* of the regular expression $(a + b)^* a(ab)^*$. For each regular expression E over Σ, a marking of E is denoted by E'. If H is a subexpression of E, we assume that markings H' and E' are chosen in such a way that H' is a subexpression of E'. A *marked regular expression* E is a regular expression over Π, the alphabet of subscripted symbols, where each subscripted symbol occurs at most once in E.

The reverse of marking is the dropping of subscripts, indicated by $^\natural$ and defined as follows: If E is a regular expression over Π, then $E^\natural$ is the regular expression over Σ that is obtained from E by dropping all subscripts in E. Thus, a marked regular expression H is a *marking* of regular expression E if and only if $H^\natural = E$. Unmarking can also be extended to words and languages: For a word w over Π, let $w^\natural$ denote the word over Σ that is constructed from w by dropping all subscripts. For a language L over Π, let $L^\natural$ denote $\{w^\natural \mid w \in L\}$. Then, for each regular expression E over Π, $\mathsf{L}(E^\natural) = \mathsf{L}(E)^\natural$.

Uppercase letters from E through J denote regular expressions over Σ or over Π; the letters a, b, and c denote symbols in Σ; the letters x, y, and z denote subscripted symbols in Π; and the letters u, v, and w denote words over Σ or over Π.

We now give a concise definition of the SGML notion of unambiguity.

DEFINITION 2.1. A regular expression E is 1-*unambiguous* if and only if, for all words u, v, w over Π and all symbols x, y in Π, the conditions $uxv, uyw \in \mathsf{L}(E')$ and $x \neq y$ imply $x^\natural \neq y^\natural$. A regular language is 1-*unambiguous* if it is denoted by some 1-unambiguous regular expression.

In other words, for each word w denoted by a 1-unambiguous regular expression E, there is exactly one witness; that is, there is one marked word v in $\mathsf{L}(E')$ such that $v^\natural = w$. Furthermore, v can be constructed incrementally by examining the next symbol of w that matches the next position of v. It is not hard to see that this definition is independent of the marking E' chosen for E. We derive an alternative definition in terms of the pairs of positions that follow each other in a word of $\mathsf{L}(E')$.

DEFINITION 2.2. For each language L, we define the following four sets:

$$first(L) = \{b \mid \text{there is a word } w \text{ such that } bw \in L\}.$$

$$last(L) = \{b \mid \text{there is a word } w \text{ such that } wb \in L\}.$$

$$follow(L, a) = \{b \mid \text{there are words } v \text{ and } w \text{ such that } vabw \in L\}, \text{ for each symbol } a.$$

$$followlast(L) = \{b \mid \text{there are words } v \text{ and } w \text{ such that } v \in L,\ v \neq \varepsilon, \text{ and } vbw \in L\}.$$

Furthermore, we extend these sets to regular expressions E by defining $first(E) = first(\mathsf{L}(E))$ and similarly for the other sets.

LEMMA 2.1 [BE96]. *For each marked regular expression E, a word $x_1 \cdots x_n$ over Π, $n \geqslant 1$, belongs to $L(E)$ if and only if the following three conditions hold*:

1. $x_1 \in \text{first}(E)$.
2. $x_n \in \text{last}(E)$.
3. $x_{i+1} \in \text{follow}(E, x_i)$, *for all* i, $1 \leqslant i < n$.

The proof is a straightforward induction on E. It is essential, though, that E is marked; for the regular expression aa, the word aaa is not in $\mathsf{L}(aa)$, although $a \in first(aa) \cap last(aa) \cap follow(aa, a)$.

Lemma 2.1 shows that, for each marked regular expression E, the conditions $u_1 z v_1, u_2 z v_2 \in \mathsf{L}(E)$ imply $u_1 z v_2 \in \mathsf{L}(E)$. Therefore, we can give an alternative characterization of 1-unambiguous regular expressions.

LEMMA 2.2. *A regular expression E is* 1*-unambiguous if and only if the following two conditions hold*:

1. *For all x, y in* $\text{first}(E')$, $x \neq y$ *implies* $x^\natural \neq y^\natural$.
2. *For all z in* $\text{sym}(E')$ *and x, y in* $\text{follow}(E', z)$, $x \neq y$ *implies* $x^\natural \neq y^\natural$.

Glushkov [Glu61] and McNaughton and Yamada [MY60] were the first researchers to construct finite-state automata from marked regular expressions using the functions *first*, *last*, and *follow*; the automata are, however, deterministic. Motivated by the work of Glushkov, Book *et al.* [BEGO71] defined a nondeterministic automaton G_E, for each regular expression E, that we call the Glushkov automaton of E. Berstel and Pin [BE96] observed that an NFA for a regular expression E can be constructed from the *first*, *last*, and *follow* functions of E as opposed to a marking of E, provided that the language denoted by E is local. Berry and Sethi [BS86] showed that Glushkov automata are natural representations of regular expressions.

DEFINITION 2.3. We define the Glushkov automaton $G_E = (Q_E, \Sigma, \delta_E, q_I, F_E)$ of a regular expression E as follows:

1. $Q_E = sym(E') \cup \{q_I\}$; that is, the states of G_E are the positions of E' together with a new, initial state q_I.
2. For $a \in \Sigma$, $\delta_E(q_I, a) = \{x \mid x \in first(E'), x^\natural = a\}$.
3. For $x \in sym(E')$ and $a \in \Sigma$, $\delta_E(x, a) = \{y \mid y \in follow(E', x), y^\natural = a\}$.
4. $F_E = \begin{cases} last(E') \cup \{q_I\}, & \text{if } \varepsilon \in \mathsf{L}(E), \\ last(E'), & \text{otherwise.} \end{cases}$

PROPOSITION 2.3 [BEGO71]. $\mathsf{L}(G_E) = \mathsf{L}(E)$.

Since $\mathsf{L}(E) = \mathsf{L}(E')^\natural$, the proof is a direct consequence of Lemma 2.1. Some simple observations follow directly from the definition of the Glushkov automaton.

LEMMA 2.4. *Let E be a regular expression. Then,*

1. *The Glushkov automaton G_E has no transitions that lead to the initial state; that is, it is* nonreturning [Lei81].

2. *Any two transitions that lead to the same state in G_E have identical labels.*

Using Glushkov automata, we can give another characterization of 1-unambiguous regular expressions that is a direct consequence of Lemma 2.2.

LEMMA 2.5. *A regular expression E is* 1*-unambiguous if and only if G_E is deterministic; that is, if and only if G_E is a DFA.*

Figure 1a demonstrates that $(a+b)^* a+\varepsilon$ is not a 1-unambiguous regular expression. Nevertheless, the language denoted by $(a+b)^* a+\varepsilon$ is a 1-unambiguous language, since it is also denoted by $(b^* a)^*$, which is a 1-unambiguous regular expression; see Fig. 1b.

The Glushkov automaton G_E can be computed in time quadratic in the size of E, which is worst-case optimal [BK92a, BK93c, CP92, CP97]. We can also construct the Glushkov automaton G_E by induction on E. The inductive definition goes back to Mirkin [Mir66] and Leiss [Lei81]. Recently, Champarnaud [Cha97] has shown that both methods produce the same automaton; see also Watson's thesis [Wat95].

Book *et al.* [BEGO71] defined an NFA to be unambiguous if each word is denoted by at most one accepting path from the initial state to some final state. They also define unambiguity of regular expressions and prove that a regular expression is unambiguous if and only if its Glushkov automaton is unambiguous. Eilenberg [Eil74] introduced a notion of unambiguous finite-state automata and unambiguous regular languages that is different, yet related. He deals with *generalized finite-state automata* whose transitions have multiplicities; such a generalized finite-state automaton induces in a canonical way multiplicities on

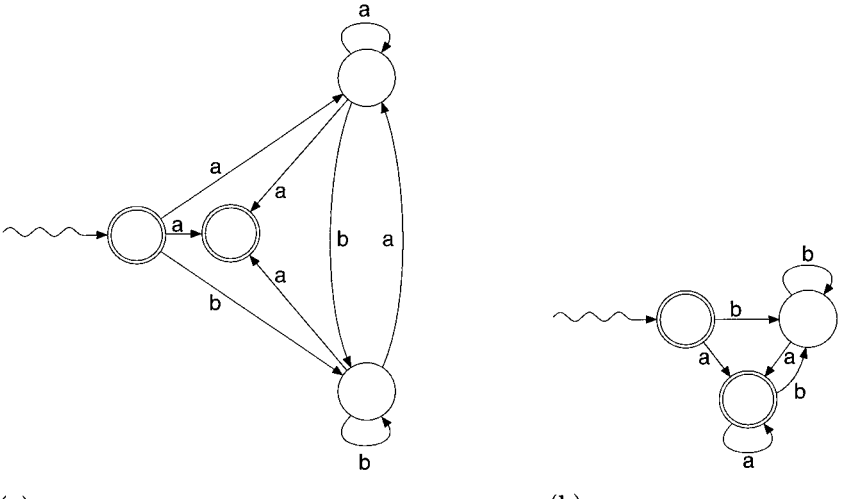

FIG. 1. The Glushkov automata corresponding to (a) $(a+b)^* a+\varepsilon$ and (b) $(b^* a)^*$.

words that depend on the number of accepting paths in the automaton and on the multiplicities of the transitions. Then, an NFA M is unambiguous in the sense of Book *et al.* if and only if the generalized finite-state automaton M', whose transitions all have the multiplicity 1, denotes only words of multiplicity 1; that is, if and only if its language is unambiguous in the sense of Eilenberg.

3. DERIVATIVES OF 1-UNAMBIGUOUS LANGUAGES

We now prove that the family of 1-unambiguous languages is closed under derivatives [Brz64]. This result is essential for characterizing the 1-unambiguous languages. The proof makes use of a linear-time algorithm to convert regular expressions into *star normal form* [BK92a, BK93c]. We use the same technique, in Section 4, to obtain a Kleene theorem for 1-unambiguous languages.

Derivatives of languages and regular expressions were introduced by Brzozowski [Brz64], who used them to construct minimal DFAs for regular languages represented by regular expressions. We show that, given a 1-unambiguous regular expression E, its derivatives are also 1-unambiguous provided that E is in star normal form. We also show that each 1-unambiguous language can be represented by a 1-unambiguous regular expression in star normal form. Thus, the family of 1-unambiguous languages is closed under derivatives.

DEFINITION 3.1. The derivative $w\backslash L$ of a language L with respect to a word w is the language $\{v \mid wv \in L\}$.

DEFINITION 3.2. The derivative $a\backslash E$ of a regular expression E with respect to a symbol a in Σ is defined inductively as follows:

$E = \varnothing$ or $E = \varepsilon$: $a\backslash E = \varnothing$.

$$E = b\text{: } a\backslash E = \begin{cases} \varepsilon, & \text{if } \ a = b, \\ \varnothing, & \text{otherwise.} \end{cases}$$

$E = F + G$: $a\backslash E = a\backslash F + a\backslash G$.

$$E = FG\text{: } a\backslash E = \begin{cases} (a\backslash F)\, G + a\backslash G, & \text{if } \ \varepsilon \in \mathsf{L}(F), \\ (a\backslash F)\, G, & \text{otherwise.} \end{cases}$$

$E = F^*$: $a\backslash E = (a\backslash F)\, F^*$.

PROPOSITION 3.1 [Brz64]. *The family of regular languages is closed under derivatives. In particular,* $\mathsf{L}(a\backslash E) = a\backslash \mathsf{L}(E)$, *for each regular expression* E.

EXAMPLE. Let $H = ((ab)^* + c)^*$ and $I = (ab)^* + c$; then, $H = I^*$. Hence, $a\backslash H = (a\backslash I)\, I^*$ and

$$\begin{aligned} a\backslash I &= a\backslash (ab)^* + a\backslash c \\ &= (a\backslash (ab))(ab)^* + \varnothing \\ &= \varepsilon b (ab)^* + \varnothing. \end{aligned}$$

Therefore,

$$a\backslash H = ((\varepsilon b)(ab)^* + \varnothing)((ab)^* + c)^*.$$

If we mark this regular expression as

$$((\varepsilon b_1)(a_2 b_3)^* + \varnothing)((a_4 b_5)^* + c_6)^*,$$

we see that b_3 can be followed by a_2 and by a_4. Thus, $a\backslash H$ is not 1-unambiguous, whereas H is 1-unambiguous.

It is the rule $a\backslash E = (a\backslash F)\, F^*$ when $E = F^*$ that gives derivatives that are not necessarily 1-unambiguous. In I, we have a last position labeled b that can be followed by a first position labeled a. In $a\backslash I$, an a position can still follow a last position that is labeled b. Considering $a\backslash G = (a\backslash I)\, I^*$, however, the same b position in $a\backslash I$ can now also be followed by a first position in I^* that is labeled a; hence, $a\backslash G$ is not 1-unambiguous. Expressions for which this situation cannot arise are said to be in *star normal form* [BK92a, BK93c]. For each regular expression E in star normal form, 1-unambiguity of E can be characterized solely in terms of the languages of the subexpressions of E; see Lemma 3.2. Using this characterization, we can prove that derivatives of 1-unambiguous regular expressions in star normal form are again 1-unambiguous. As has already been mentioned, because each 1-unambiguous language can be denoted by a 1-unambiguous regular expression in star normal form, derivatives preserve 1-unambiguity of regular languages.

DEFINITION 3.3. A regular expression E is in *star normal form* if, for each starred subexpression H^* of E, $\mathit{followlast}(H') \cap \mathit{first}(H') = \varnothing$ and $\varepsilon \notin \mathsf{L}(H)$; in particular, no first position in H can follow a last position in H.

LEMMA 3.2. *Let E be in star normal form.*

$E = \varnothing$, $E = \varepsilon$, or $E = a$: E is 1-unambiguous.

$E = F + G$: E is 1-unambiguous if and only if F and G are 1-unambiguous and $\mathrm{first}(F) \cap \mathrm{first}(G) = \varnothing$.

$E = FG$: If $\mathsf{L}(E) = \varnothing$, then E is 1-unambiguous.

If $\mathsf{L}(E) \neq \varnothing$ and $\varepsilon \in \mathsf{L}(F)$, then E is 1-unambiguous if and only if F and G are 1-unambiguous, $\mathrm{first}(F) \cap \mathrm{first}(G) = \varnothing$, *and* $\mathrm{followlast}(F) \cap \mathrm{first}(G) = \varnothing$.

If $\mathsf{L}(E) \neq \varnothing$ and $\varepsilon \notin \mathsf{L}(F)$, then E is 1-unambiguous if and only if F and G are 1-unambiguous and $\mathrm{followlast}(F) \cap \mathrm{first}(G) = \varnothing$.

$E = F^*$: E is 1-unambiguous if and only if F is 1-unambiguous and $\mathrm{followlast}(F) \cap \mathrm{first}(F) = \varnothing$.

Proof. First note that we can prove a more general result by dropping the lemma-wide condition that E be in star normal form. If we do so, we should add an extra condition to the statement of the lemma ($\mathit{followlast}(F') \cap \mathit{first}(F') = \varnothing$) and change the proof of the last case slightly, as the we now have $\mathit{followlast}(F') \cap \mathit{first}(F') = \varnothing$ by assumption.

We prove the three most interesting cases. First, we assume that $E = FG$, $\mathsf{L}(E) \neq \emptyset$, and $\varepsilon \in \mathsf{L}(F)$. Let F and G be 1-unambiguous, $first(F) \cap first(G) = \emptyset$, and $followlast(F) \cap first(G) = \emptyset$. We show that E is 1-unambiguous. Let $E' = F'G'$ and let $uxv, uyw \in \mathsf{L}(E')$ such that $x^\natural = y^\natural$. If $x, y \in sym(F')$, then there are v_1 and w_1 such that $uxv_1, uyw_1 \in \mathsf{L}(F')$, because E' is marked; hence, since F is 1-unambiguous, $x = y$. The case when $x, y \in sym(G')$ is analogous. Finally, we show that $x \in sym(F')$ and $y \in sym(G')$ cannot occur. If $u = \varepsilon$, then $x \in sym(F')$ and $y \in sym(G')$ imply that $x \in first(F')$ and $y \in first(G')$; hence, $x^\natural \in first(F)$ and $y^\natural \in first(G)$, which contradicts the assumption that $first(F) \cap first(G) = \emptyset$. If $u \neq \varepsilon$, then $x \in sym(F')$ and $y \in sym(G')$ imply that $u \in \mathsf{L}(F')$, because E' is marked; hence, $x \in followlast(F')$ and $y \in first(G')$. Therefore, $x^\natural \in followlast(F)$ and $y^\natural \in first(G)$, which contradicts the assumption that $followlast(F) \cap first(G) = \emptyset$.

Second, let $E = F^*$, F be 1-unambiguous, and $followlast(F) \cap first(F) = \emptyset$. We show that E is 1-unambiguous. Let $E' = F'^*$. Lemma 2.1 implies that, for each word $w \in \mathsf{L}(E')$, there is exactly one canonical decomposition $w = w_1 \cdots w_n$, $n \geqslant 0$, such that, for $1 \leqslant i \leqslant n$,

1. $w_i \in \mathsf{L}(F') \backslash \{\varepsilon\}$; in particular, w_i starts with a symbol in $first(F')$ and ends with a symbol in $last(F')$.
2. w_i contains no subword xy such that $x \in last(F')$ and $y \in first(F')$.

Now let $uxv, uyw \in \mathsf{L}(E')$ such that $x^\natural = y^\natural$. If $u = \varepsilon$, then $x, y \in first(F')$; hence, $x = y$, because F is 1-unambiguous. So let us assume that $u = u_0 z$. If $z \in last(F')$ and $x \in first(F')$, then $y \in first(F')$ (otherwise $y \in followlast(F')$ and $y^\natural \in followlast(F) \cap first(F)$). Therefore, the canonical decompositions of uxv and uyw separate either both of zx and zy or none of zx and zy. Hence, uxv and uyw can be decomposed as $uxv = u_1 \cdots u_n x v_1 \cdots v_l$ and $uyw = u_1 \cdots u_n y w_1 \cdots w_k$ such that $n \geqslant 1$, $k, l \geqslant 0$, and $u_1, ..., u_{n-1}, u_n x v_1, u_n y w_1, v_2, ..., v_l, w_2, ..., w_k \in \mathsf{L}(F')$. Because F is 1-unambiguous, $x = y$.

Third, let $E = F^*$ and $a \in followlast(F) \cap first(F)$. We show that E is not 1-unambiguous. Once more let $E' = F'^*$. There are x, y, z, u, v, and w such that $xu, vz, vzyw \in \mathsf{L}(F')$ and $x^\natural = y^\natural = a$. Therefore, $x \in first(F')$ and $y \in followlast(F')$; however, since E is in star normal form, $first(F') \cap followlast(F') = \emptyset$, $x \neq y$ and E is not 1-unambiguous. ▌

THEOREM A. *Each 1-unambiguous language can be denoted by a 1-unambiguous regular expression in star normal form.*

Proof. Given a regular expression E, we can construct a regular expression $E^\bullet$ in star normal form such that $G_{E^\bullet} = G_E$ and $\mathsf{L}(E^\bullet) = \mathsf{L}(E)$ [BK92a, BK93c]. Hence, by Lemma 2.5, $E^\bullet$ is 1-unambiguous if and only if E is 1-unambiguous. ▌

At this point, this paper is not quite self-contained; we have to refer to other papers [BK92a, BK93c] for a full proof of Theorem A. However, the main idea, why each regular expression can be transformed into star normal form without any changes in the Glushkov automaton, is quite easy to grasp. Given a regular expression H, we can remove from its Glushkov automaton G_H all the transitions that lead from a state in $last(H')$ to a state in $first(H')$, and the resulting automaton is

the Glushkov automaton of another regular expression, say of H°. Then, $\mathit{followlast}(H^{\circ\prime}) \cap \mathit{first}(H^{\circ\prime}) = \varnothing$ and $G_{H^{\circ *}} = G_{H^*}$. Therefore, if we replace each subexpression H^* of E with $H^{\circ *}$, the resulting regular expression $E^\bullet$ is in star normal form and $G_{E^\bullet} = G_E$. For example, if $H = a^*b^*$, then $H^\circ = a + b$; hence, the star normal form of $E = (a^*b^*)^*$ is $E^\bullet = (a+b)^*$. Figure 2 demonstrates that the Glushkov automata of $(a^*b^*)^*$ and $(a+b)^*$ are identical.

The importance of star normal form for derivatives is captured in the following result.

THEOREM B. *If E is a 1-unambiguous regular expression in star normal form, then $a\backslash E$ is 1-unambiguous and in star normal form, for each a in Σ.*

Proof. A straightforward induction on E to show that $a\backslash E$ is in star normal form. Using Lemma 3.2, we show by induction on E that $a\backslash E$ is 1-unambiguous.

$E = \varnothing$, $E = \varepsilon$, or $E = b$: $a\backslash E$ is 1-unambiguous.

$E = F + G$: By the induction hypothesis, $a\backslash F$ and $a\backslash G$ are both 1-unambiguous. Furthermore, because $\mathit{first}(F) \cap \mathit{first}(G) = \varnothing$, we have $\mathsf{L}(a\backslash F) = \varnothing$ or $\mathsf{L}(a\backslash G) = \varnothing$; hence, $\mathit{first}(a\backslash F) \cap \mathit{first}(a\backslash G) = \varnothing$. Thus, $a\backslash E$ is 1-unambiguous.

$E = FG$: Without loss of generality, $\mathsf{L}(a\backslash F) \neq \varnothing$ and $\mathsf{L}(E) \neq \varnothing$. Since E is 1-unambiguous, $\mathsf{L}(a\backslash F) \neq \varnothing$ implies $\mathsf{L}(a\backslash G) = \varnothing$. Therefore, we have only to show that $(a\backslash F)\,G$ is 1-unambiguous. By the induction hypothesis, $a\backslash F$ is 1-unambiguous. Furthermore, $\mathit{followlast}(a\backslash F) \cap \mathit{first}(G)$ is a subset of $\mathit{followlast}(F) \cap \mathit{first}(G)$; hence, it is empty. Finally, if $\varepsilon \in \mathsf{L}(a\backslash F)$, then $\mathit{first}(a\backslash F) \cap \mathit{first}(G)$ is a subset of $\mathit{followlast}(F) \cap \mathit{first}(G)$; hence, it is empty. Therefore, $(a\backslash F)\,G$ is 1-unambiguous.

$E = F^*$: Since E is 1-unambiguous and is in star normal form, $\mathit{followlast}(F) \cap \mathit{first}(F) = \varnothing$; thus, $\mathit{followlast}(a\backslash F) \cap \mathit{first}(F^*) = \varnothing$. Furthermore, if $\varepsilon \in \mathsf{L}(a\backslash F)$, then $\mathit{first}(a\backslash F) \cap \mathit{first}(F^*) = \varnothing$, because it is a subset of $\mathit{followlast}(F) \cap \mathit{first}(F)$. Hence, by the induction hypothesis, $a\backslash E$ is 1-unambiguous. ∎

Because the derivative of a language L with respect to a word is the repeated derivative of L with respect to the word's symbols, we need prove only that the derivative of L with respect to a symbol is 1-unambiguous. Therefore, Theorems A and B immediately imply the following result.

THEOREM C. *If L is a 1-unambiguous language, then so is $w\backslash L$, for all w in Σ^*.*

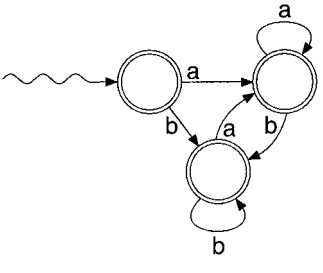

FIG. 2. The Glushkov automaton corresponding to $(a^*b^*)^*$ and its star normal form $(a+b)^*$.

4. A KLEENE THEOREM FOR 1-UNAMBIGUOUS LANGUAGES

We characterize the family of 1-unambiguous languages in terms of closure properties. So far, we have seen one operation under which this family is closed, namely, derivatives. The family of 1-unambiguous languages is not closed under any of the operations that constitute the Kleene characterization of the regular languages, namely union, concatenation, and star. We can, however, define restrictions under which these operations preserve 1-unambiguous languages; the restricted operations are still powerful enough to generate each 1-unambiguous language from the usual primitives. The restrictions involve the first and last symbols of the words in a language.

It is convenient, for a language L, to denote $L\setminus\{\varepsilon\}$ by L^-.

THEOREM D. *The family of* 1*-unambiguous languages is the smallest family* $\mathscr{D}$ *of languages that satisfies the following conditions*:

1. $\varnothing$, ε, *and* $\{a\}$ *are in* $\mathscr{D}$, *for each symbol* a.
2. *If* $A, B \in \mathscr{D}$ *and* $\mathrm{first}(A) \cap \mathrm{first}(B) = \varnothing$, *then* $A \cup B \in \mathscr{D}$.
3. *If* $A, B \in \mathscr{D}$, $\varepsilon \notin A$, *and* $\mathrm{followlast}(A) \cap \mathrm{first}(B) = \varnothing$, *then* $AB \in \mathscr{D}$.
4. *If* $A \in \mathscr{D}$, *then* $A^- \in \mathscr{D}$.
5. *If* $A \in \mathscr{D}$ *and* $\mathrm{followlast}(A) \cap \mathrm{first}(A) = \varnothing$, *then* $A^* \in \mathscr{D}$.

We prove Theorem D by first showing that L is 1-unambiguous if and only if L^- is 1-unambiguous. To this end, for each regular expression E, we construct a regular expression E^- that denotes $\mathsf{L}(E)^-$. We then show that E is 1-unambiguous if and only if E^- is 1-unambiguous, provided that E is in star normal form.

DEFINITION 4.1. For each regular expression E, define E^- inductively as follows:

$E = \varepsilon$ or $E = \varnothing$: $E^- = \varnothing$.

$E = a$: $E^- = a$.

$E = F + G$: $E^- = F^- + G^-$.

$$E = FG\text{: } E^- = \begin{cases} F^-G + G^-, & \text{if } \varepsilon \in \mathsf{L}(E), \\ FG, & \text{otherwise.} \end{cases}$$

$E = F^*$: $E^- = F^-F^*$.

LEMMA 4.1.

1. $\mathsf{L}(E^-) = \mathsf{L}(E)^-$.
2. *If* E *is in star normal form, then so is* E^-.

The proof is a straightforward induction on E.

LEMMA 4.2. *A regular expression E in star normal form is* 1*-unambiguous if and only if E^- is* 1*-unambiguous.*

Proof. The proof is by induction on E. Each subexpression of E and E^- is in star normal form. Thus, we can apply Lemma 3.2. We show only the case when $E = FG$ and $\varepsilon \in \mathsf{L}(E)$. In this case, $E^- = F^-G + G^-$. If $\mathsf{L}(F^-) = \varnothing$, then $\mathsf{L}(F) \subseteq \{\varepsilon\}$; hence, both F and F^- are 1-unambiguous in this case. Now, E is 1-unambiguous if and only if F and G are 1-unambiguous, $first(F) \cap first(G) = \varnothing$, and $followlast(F) \cap first(G) = \varnothing$. On the other hand, by the induction hypothesis, E^- is 1-unambiguous if and only if F^- and G^- are 1-unambiguous, $first(F^-G) \cap first(G) = \varnothing$, and $followlast(F^-) \cap first(G) = \varnothing$. This property holds even if $\mathsf{L}(F^-) = \varnothing$ and Lemma 3.2 cannot be applied to F^-G. Again by the induction hypothesis, the two conditions are equivalent. ▮

Proof of Theorem D. First, let $\mathscr{D}$ be a family of languages that satisfies conditions 1 through 5. We show that $\mathscr{D}$ contains all 1-unambiguous languages. For this purpose, we prove, by induction on E, that $\mathsf{L}(E) \in \mathscr{D}$ if E is in star normal form. It is a straightforward application of Lemma 3.2. We show only the case when $E = FG$, $\mathsf{L}(E) \neq \varnothing$, and $\varepsilon \in \mathsf{L}(F)$. By Lemma 3.2, F and G are 1-unambiguous, $first(F) \cap first(G) = \varnothing$, and $followlast(F) \cap first(G) = \varnothing$. Furthermore, $\mathsf{L}(E) = \mathsf{L}(F)^- \mathsf{L}(G) \cup \mathsf{L}(G)$. By the induction hypothesis, $\mathsf{L}(F)^- \in \mathscr{D}$ and $\mathsf{L}(G) \in \mathscr{D}$. Finally, $\varepsilon \notin \mathsf{L}(F)^-$, $followlast(\mathsf{L}(F)^-) \cap first(G) = followlast(F) \cap first(G) = \varnothing$, and $first(\mathsf{L}(F)^- \mathsf{L}(G)) \cap first(\mathsf{L}(G)) = first(F) \cap first(G) = \varnothing$; hence, $\mathsf{L}(E) \in \mathscr{D}$.

To complete the proof we prove that the family of 1-unambiguous languages fulfills conditions 1 through 5. We show only that it fulfills condition 5. Let A be a 1-unambiguous language such that $followlast(A) \cap first(A) = \varnothing$. Without loss of generality, we assume that $\varepsilon \notin A$. Let E be a 1-unambiguous regular expression in star normal form that denotes A. We show first that E^* is also in star normal form; that is, that $followlast(E') \cap first(E') = \varnothing$. It suffices to show that $(followlast(E') \cap first(E'))^\natural = \varnothing$. This equation holds because $(followlast(E') \cap first(E'))^\natural$ is a subset of $followlast(E) \cap first(E)$, or, equivalently, of $followlast(A) \cap first(A)$. By Lemma 3.2, E^* is 1-unambiguous. Thus, A^*, which is denoted by E^*, is a 1-unambiguous language. ▮

5. THE RECOGNITION OF 1-UNAMBIGUOUS LANGUAGES

It is well known that, for each regular language L, the minimum-state deterministic automaton $MS(L)$ of L is uniquely determined up to a renaming of states. We examine the structural properties of $MS(L)$ that characterize a 1-unambiguous language L.

Starting from a regular expression E that denotes L, we can construct the minimal DFA $MS(L)$ by applying the subset construction to the Glushkov automaton G_E and then minimizing the resulting DFA using the equivalence-class construction [ASU86]. If E is a 1-unambiguous regular expression, however, then its Glushkov automaton is already a DFA and we do not need the subset construction. Thus, we look for structural properties of Glushkov automata that are preserved under minimization, but are not necessarily preserved under the subset construction.

Automata that contain no useless states[2] are called *trim* [Eil74, Per90]. In this paper, we also consider $G_\varnothing$, the Glushkov automaton of the empty set, to be trim, although its single state is useless. In this section, we denote regular languages by *trim regular expressions*; that is, regular expressions that are either syntactically identical to $\varnothing$ or do not contain $\varnothing$ as a syntactic constituent. On the one hand, it is not hard to see that the Glushkov automaton of a trim regular expression is also trim. On the other hand, $\varnothing$ in a regular expression E can introduce useless states into G_E; consider, for example, $a(\varnothing b + c)$. As usual, we can transform a regular expression E into a trim regular expression E^{t} such that $\mathsf{L}(E) = \mathsf{L}(E^{\mathrm{t}})$ by applying the following rules:

$$E + \varnothing, \varnothing + E \to E; \qquad E\varnothing, \varnothing E \to \varnothing; \qquad \varnothing^* \to \varepsilon.$$

In the context of 1-unambiguous languages, this transformation is justified by the following lemma.

LEMMA 5.1. *Each* 1*-unambiguous language can be denoted by a trim*, 1*-unambiguous regular expression. In particular, a regular expression E is* 1*-unambiguous if and only if* E^{t} *is* 1*-unambiguous.*

Proof. It is not hard to see that E'^{t} is a marking of E^{t}. Since $\mathsf{L}(E') = \mathsf{L}(E'^{\mathrm{t}})$, the regular expression E is 1-unambiguous if and only if E^{t} is 1-unambiguous. ∎

Now we consider the cyclic structure of Glushkov automata that we describe in terms of *orbits* and *gates*. It turns out that the structure of orbits and gates is preserved under minimization; hence, orbits and gates are exactly the right tools for characterizing 1-unambiguous languages.

DEFINITION 5.1. For a state q of an NFA M, the *orbit* of q, denoted by $\mathcal{O}(q)$, is the strongly connected component of q; that is, it is the set of states of M that can be reached from q and from which q can be reached. We consider the orbit of q to be *trivial* if it consists of only the state q and there are no transitions from q to itself in M.

DEFINITION 5.2. A state q of an NFA M is a gate of its orbit $\mathcal{O}(q)$ if q is a final state or q has a transition to a state outside $\mathcal{O}(q)$. The NFA M has the *orbit property* if all the gates of each orbit have identical connections to the outside world. More precisely, if any pair q_1 and q_2 of gates in the same orbit satisfies the following two conditions:

1. q_1 is final if and only if q_2 is final.
2. For all states q outside the orbit of q_1 and q_2, there is a transition (q_1, a, q) in M if and only if there is a transition (q_2, a, q) in M.

The two automata in Fig. 3 have three orbits, namely $\{1\}$, $\{2, 3\}$, and $\{4\}$. The singleton orbits are trivial, and each state is a gate of its orbit. The orbit property holds for the automaton in Fig. 3a, whereas it does not hold for the one in Fig. 3b.

[2] A state is *useless* if it does not lie on any path from the initial state to some final state.

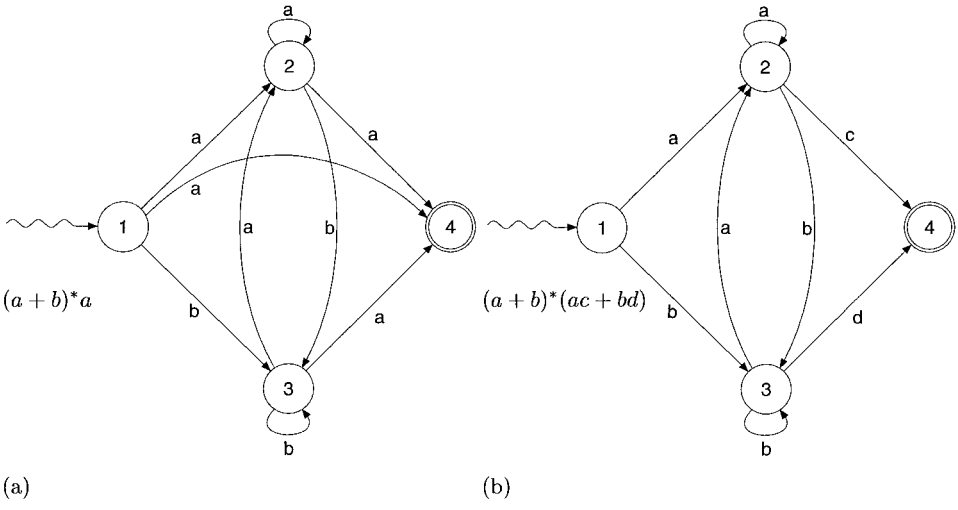

FIG. 3. Two NFAs: (a) fulfills the orbit property, (b) does not.

DEFINITION 5.3. For a state q of an NFA M, let the *orbit automaton* M_q of q be the automaton obtained by restricting the state set of M to $\mathcal{O}(q)$ with initial state q and with the gates of $\mathcal{O}(q)$ as the final states of M_q. The language of M_q is called the *orbit language* of q. The languages $\mathsf{L}(M_q)$ are also called the *orbit languages* of M. We also consider a larger subautomaton of M related to q: The NFA M^q is M with its state set restricted to the states reachable from q, with q as its initial state, and its transitions are similarly restricted.

We now establish the following characterization of 1-unambiguous languages.

THEOREM E. *Let M be a minimal DFA. Then, $L(M)$ is* 1*-unambiguous if and only if M has the orbit property and all orbit languages of M are* 1*-unambiguous. If $\mathsf{L}(M)$ is* 1*-unambiguous, then a* 1*-unambiguous regular expression denoting $\mathsf{L}(M)$ can be constructed from* 1*-unambiguous regular expressions for the orbit languages.*

We begin the proof of Theorem E by demonstrating that it holds in the left-to-right direction. First, we show that the Glushkov automaton of a 1-unambiguous regular expression has the orbit property and that all its orbit languages are 1-unambiguous. Second, we show that these two properties are preserved under minimization. Thus, we first investigate the orbit structure of a Glushkov automaton. Then, we demonstrate how the orbit structure of a DFA is related to the orbit structure of the DFA's equivalent minimal automaton.

Let us recall that, for each regular expression E, the state set of the Glushkov automaton G_E consists of $sym(E')$, the true states of G_E, plus the initial state. Let H be a subexpression of E. Then, $sym(H')$ is a subset of $sym(E')$. Given x, y in $sym(H')$ such that $y \in follow(H', x)$, we also have $y \in follow(E', x)$, provided that E is trim. Therefore, all transitions between true states of G_H are also transitions in G_E. In this sense, G_H is embedded in G_E. It is essential, though, that E is trim, because otherwise H can be blocked by the empty set in E; for example, when $E = aa\varnothing$ and $H = aa$, G_H has two a transitions, but G_E has none.

LEMMA 5.2. *Let E be a trim regular expression and $x \in \mathrm{sym}(E')$. Then, the orbits and gates in G_E can be described as follows*:

1. *If there is no starred subexpression H^* of E such that $x \in \mathrm{sym}(H')$, then $\mathcal{O}(x) = \{x\}$ and $\mathcal{O}(x)$ is trivial. On the other hand, if H^* is the maximal starred subexpression of E such that $x \in \mathrm{sym}(H')$, then $\mathcal{O}(x) = \mathrm{sym}(H')$ and $\mathcal{O}(x)$ is not trivial. Finally, the orbit of the initial state is trivial.*

2. *If H^* is a maximal starred subexpression of E such that $\mathrm{sym}(H') \neq \emptyset$, then the last positions of H' are the gates of the orbit given by $\mathrm{sym}(H')$. Furthermore, each transition of G_E between two states of $\mathrm{sym}(H')$ is already present in G_{H^*}.*

The proof is by a straightforward induction on E.

LEMMA 5.3. *For each trim regular expression E, the Glushkov automaton G_E satisfies the orbit property.*

Proof. First, we observe that, for each subexpression H of E, $last(H') \subseteq last(E')$ or $last(H') \cap last(E') = \emptyset$. Then, the claim follows from the previous lemma by induction on E. We show only the case when $E = FG$ and we consider only nontrivial orbits that are contained in $sym(F')$. Thus, let H^* be a maximal starred subexpression of F such that $sym(H^*) \neq \emptyset$. By the induction hypothesis, the orbit $sym(H')$ of G_F satisfies the orbit property. If $last(H') \cap last(E') = \emptyset$, then no gate of $sym(H')$ is final in G_E and the transitions of G_E that leave the orbit $sym(H')$ are the same as the transitions of G_F that leave $sym(H')$. On the other hand, if $last(H') \subseteq last(E')$, then all the gates of $sym(H')$ in G_E are final states and, for each x in $first(G')$, an $x^\natural$ transition to x is added to each gate of $sym(H')$ in G_E in addition to the transitions that are already present in G_F. Therefore, in both cases, the orbit given by $sym(H')$ in G_E satisfies the orbit property. ∎

LEMMA 5.4. *Let E be a trim, 1-unambiguous regular expression. Then, each orbit language of G_E is 1-unambiguous. In particular, if the orbit of state x in G_E is nontrivial, then there is a maximal starred subexpression H^* of E such that the orbit language of x is a derivative of $\mathsf{L}(H^*)$.*

Proof. We consider only nontrivial orbits. Thus, let H^* be a maximal starred subexpression of E and $x \in sym(H')$. Since $\mathsf{L}(H^*)$ is 1-unambiguous and the derivatives of 1-unambiguous languages are 1-unambiguous, it suffices to show that the orbit language $\mathsf{L}((G_E)_x)$ of x in G_E is a derivative of $\mathsf{L}(H^*)$. All transitions of G_E among the states of $sym(H')$ are already transitions in G_{H^*}. Furthermore, if we remove the initial state and its transitions from G_{H^*} and make x the new initial state, the resulting automaton is $(G_E)_x$. Now let w be a word that takes the initial state to x in G_{H^*}. Because G_{H^*} is nonreturning and deterministic, $\mathsf{L}((G_E)_x)$ is the derivative of $\mathsf{L}(G_{H^*})$ with respect to w. ∎

Now we consider the relationship between the orbit structure of a DFA and its minimization.

DEFINITION 5.4. Let M be a DFA and $\bar{M}$ be its equivalent minimal DFA. For a state q of M let $[q]$ denote the set of states of M that are equivalent to q. In

particular, $\{[q] \mid q \text{ state of } M\}$ is the set of states of $\bar{M}$. Then, an orbit C of M is a *lift* of an orbit K of $\bar{M}$ (or C *lifts* K) if $K = \{[q] \mid q \in C\}$.

LEMMA 5.5. *Let M be a DFA and let $\bar{M}$ be its equivalent minimal DFA. Then, for each orbit K of $\bar{M}$ there is an orbit C of M such that*

1. *C is a lift of K and*
2. *For each q in C, the orbit languages of q in M and $[q]$ in $\bar{M}$ are identical.*

Proof. First we show that, given an orbit K of $\bar{M}$ and a state p of M such that $[p] \in K$, there is an orbit C' of M that is reachable from p and lifts K. Indeed, there are states $q_0, q_1, ..., q_n$ of M, $p = q_0$, and symbols $a_0, ..., a_n$ such that

$$[q_0] \xrightarrow{a_0} [q_1] \longrightarrow \cdots \longrightarrow [q_n] \xrightarrow{a_n} [q_0]$$

and

$$K = \{[q_i] \mid 0 \leqslant i \leqslant n\}.$$

Unfortunately, the states of M reachable from q_0 with $a_0 \cdots a_i$, $0 \leqslant i \leqslant n$, do not necessarily belong to a single orbit of M. By repeatedly applying the same sequence $a_0 \cdots a_n$ to q_0 in M, however, we finally get into a cycle whose states all belong to a single orbit C' of M. By construction, $K \subseteq \{[q] \mid q \in C'\}$. Furthermore, if q and r belong to one orbit of M, then $[q]$ and $[r]$ belong to one orbit in $\bar{M}$. Thus, C' lifts K.

The orbits of M are partially ordered with respect to reachability; therefore, we can choose C as a maximal (with respect to reachability) orbit C' that lifts K. It suffices to prove the following two properties:

1. For each q in C, q is a gate of C if and only if $[q]$ is a gate of K.
2. For all q, r in C and a in Σ, if $[q] \xrightarrow{a} [r]$, then there is an s in C equivalent to r such that $q \xrightarrow{a} s$.

The first claim follows from the second. So let $q, r \in C$, $[q] \xrightarrow{a} [r]$. Then, there is a state s of M equivalent to r such that $q \xrightarrow{a} s$. We have to show that $s \in C$. Since $[s] = [r] \in K$, we can reach an orbit C' of M from s that lifts K. The maximality of C implies $C = C'$; hence, $s \in C$. ∎

LEMMA 5.6. *Let M be a DFA and $\bar{M}$ be its equivalent minimal DFA.*

1. *If M satisfies the orbit property, then so does $\bar{M}$.*
2. *If all orbit languages of M are 1-unambiguous, then so are all orbit languages of $\bar{M}$.*

Proof. Let K be an orbit of $\bar{M}$ and C be a lift of K according to Lemma 5.5. Furthermore, we assume that M has an a transition from q in C to r outside C. We show that $[r] \notin K$.

Because M is deterministic, M_q has no a transition from q; hence, $a \backslash \mathsf{L}(M_q) = \varnothing$. Since the orbit languages of q and $[q]$ are identical, $a \backslash \mathsf{L}(\bar{M}_{[q]}) = \varnothing$; hence, $\bar{M}_{[q]}$

has no a transition from $[q]$. Yet, $\bar{M}$ has an a transition from $[q]$ to $[r]$; thus, $[r] \notin K$.

The proof of the lemma is completed by applying Lemma 5.5. ▮

Proof of Theorem E. Lemmas 5.3, 5.4, and 5.6 establish the left-to-right direction. We show the implication from right to left by induction on the number of orbits of M. We assume that M has more than one orbit and consider the orbit $\mathcal{O}(q_I)$ of the initial state q_I. Let $a_1, ..., a_n$ be the distinct symbols of the transitions that leave $\mathcal{O}(q_I)$. Since M satisfies the orbit property, there are states $q_1, ..., q_n$ outside $\mathcal{O}(q_I)$ such that all gates of $\mathcal{O}(q_I)$ have an a_i transition to q_i, and there are no other outgoing transitions from $\mathcal{O}(q_I)$ to the outside. Since M is deterministic, M_{q_I} has no a_i transition from a final state.

We consider M^{q_i}, the subautomaton of M whose states are the states of M that are reachable from q_i as the initial state. Because M^{q_i} is a minimal DFA that has fewer orbits than M, the language of M^{q_i} is 1-unambiguous by the induction hypothesis. Furthermore, either

$$\mathsf{L}(M) = \mathsf{L}(M_{q_I})(a_1 \mathsf{L}(M^{q_1}) \cup \cdots \cup a_n \mathsf{L}(M^{q_n}))$$

or

$$\mathsf{L}(M) = \mathsf{L}(M_{q_I})(a_1 \mathsf{L}(M^{q_1}) \cup \cdots \cup a_n \mathsf{L}(M^{q_n}) \cup \{\varepsilon\}).$$

By Lemma 3.2, a 1-unambiguous regular expression for M can be constructed from 1-unambiguous regular expressions for M_{q_I} and $M^{q_1}, ..., M^{q_n}$, which completes the proof. ▮

Theorem E suggests an inductive algorithm to determine, given a minimal DFA M, whether $\mathsf{L}(M)$ is 1-unambiguous; in fact, if M satisfies the orbit property, then all orbit automata of M are also minimal. Yet we still have to cover the case when M consists of a single, nontrivial orbit. In this case, Lemmas 5.4 and 5.5 imply that there is a 1-unambiguous regular expression H^* such that $\mathsf{L}(M)$ is a derivative of $\mathsf{L}(H^*)$. The question is, how can we construct G_H from M? Obviously, we have to cut from M the "feedback" transitions that distinguish G_{H^*} from G_H. It turns out that, if we cut from M the maximum number of "feedback" transitions, then we arrive at an automaton N that consists of several disconnected subautomata. Each of these subautomata recognizes a 1-unambiguous language, and a 1-unambiguous regular expression for $\mathsf{L}(M)$ can be constructed from the 1-unambiguous regular expressions for the languages of the subautomata. Then, Theorem E can be applied to the subautomata of N recursively.

DEFINITION 5.5. For a DFA M, a symbol a in Σ is M *consistent* if there is a state $f(a)$ in M such that all final states of M have an a transition to $f(a)$. A set S of symbols is M *consistent* if each symbol in S is M consistent.

DEFINITION 5.6. Let M be an NFA and S be a set of symbols. The S cut M_S of M is constructed from M by removing, for each $a \in S$, all a transitions that leave a final state of M.

Figure 4a gives a DFA for which $\{a, b\}$ is consistent and Fig. 4b gives its $\{a, b\}$ cut.

The following theorem is a generalization of Theorem E. It also deals with minimal automata that consist of a single, nontrivial orbit.

THEOREM F. *Let M be a minimal DFA and S be an M-consistent set of symbols. Then, $\mathsf{L}(M)$ is 1-unambiguous if and only if*

1. *M_S satisfies the orbit property and*
2. *All orbit languages of M_S are 1-unambiguous.*

Furthermore, if M consists of a single, nontrivial orbit and $\mathsf{L}(M)$ is 1-unambiguous, M has at least one M-consistent symbol.

We illustrate Theorem F with two examples. The language recognized by the automaton in Fig. 4a is 1-unambiguous, because its $\{a, b\}$ cut has only trivial orbits; a 1-unambiguous regular expression for the whole language is $c(a + b(\varepsilon + cc))^*$.

Figure 5 shows the minimal DFA that recognizes $(0 + 1)^* 0(0 + 1)$. It consists of a single orbit with two gates, 00 and 01, but neither 0 nor 1 is consistent. Thus, $(0 + 1)^* 0(0 + 1)$ does not denote a 1-unambiguous language. Similarly, the regular expressions $(0 + 1)^* 0(0 + 1)^n$, for each $n \geqslant 1$, do not denote 1-unambiguous languages.

We need some preparation before we can prove Theorem F. The relationship between a DFA and its cuts is captured in the following result.

LEMMA 5.7. *Let M be a trim DFA and S be an M-consistent set of symbols. Then, $\mathsf{L}(M)$ is 1-unambiguous if and only if, for each state q, the language $\mathsf{L}(M_S^q)$ is 1-unambiguous. If $\mathsf{L}(M)$ is 1-unambiguous, a 1-unambiguous regular expression denoting $\mathsf{L}(M)$ can be constructed from 1-unambiguous regular expressions for the languages $\mathsf{L}(M_S^q)$.*

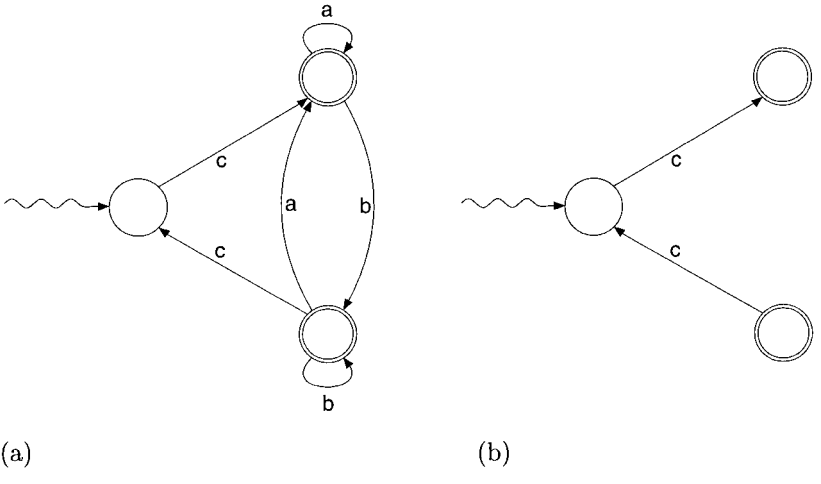

FIG. 4. DFAs and cuts: (a) A DFA for which $\{a, b\}$ is consistent, (b) its $\{a, b\}$ cut.

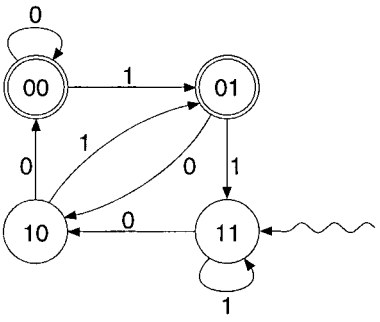

FIG. 5. The minimal DFA for $(0+1)^*0(0+1)$.

Proof. First, we show that

$$\mathsf{L}(M) = \mathsf{L}(M_S)\left(\bigcup_{a\in S} a\mathsf{L}(M_S^{f(a)})\right)^*.$$

Let $w = a_1\cdots a_n \in \mathsf{L}(M)$, $n \geqslant 1$, let

$$s = \{i \mid 1 \leqslant i \leqslant n, a_i \in S, a_1\cdots a_{i-1} \in \mathsf{L}(M)\} \cup \{n+1\},$$

and let $i_1, ..., i_k$ be the elements of s in increasing order. Then,

1. $a_1\cdots a_{i_1-1} \in \mathsf{L}(M_S)$ and
2. $a_{i_j}\cdots a_{i_{j+1}-1} \in a_{i_j}\mathsf{L}(M_S^{f(a_{i_j})})$,

for $1 \leqslant j < k$, because M is deterministic. Thus,

$$w \in \mathsf{L}(M_S)\left(\bigcup_{a\in S} a\mathsf{L}(M_S^{f(a)})\right)^*.$$

On the other hand, if $w \in \mathsf{L}(M_S)$, $a_i \in S$, $w_i \in \mathsf{L}(M_S^{f(a_i)})$, for $1 \leqslant i \leqslant n$, $n \geqslant 0$, then w takes the initial state to a final state in M, a_i takes the final state to $f(a_i)$ in M, and w_i takes $f(a_i)$ to a final state in M, for $1 \leqslant i \leqslant n$. Thus, $wa_1w_1\cdots a_nw_n \in \mathsf{L}(M)$.

Next, $followlast(\mathsf{L}(M_S)) \subseteq \Sigma\setminus S$ and $followlast(a\mathsf{L}(M_S^{f(a)}) \subseteq \Sigma\setminus S$. Furthermore, if $\varepsilon \in \mathsf{L}(M_S)$, then $\varepsilon \in \mathsf{L}(M)$; hence, $first(\mathsf{L}(M_S)) \subseteq \Sigma\setminus S$. By Lemma 3.2, if $\mathsf{L}(M_S^q)$ is 1-unambiguous, for each state q of M, then $\mathsf{L}(M)$ is 1-unambiguous.

Finally, we assume that $\mathsf{L}(M)$ is 1-unambiguous; that is, there is a trim, 1-unambiguous regular expression E in star normal form that denotes $\mathsf{L}(M)$. Since M is trim, there is, for each state q, a word w that takes the initial state to q in M. Let p be the state to which w takes the initial state in G_E. Since M and G_E are both deterministic, $\mathsf{L}(M_S^q) = \mathsf{L}(G_{E_S^p})$. By Lemma 5.8, which we prove later, $\mathsf{L}(M_S^q)$ is 1-unambiguous. ∎

In the proof of Lemma 5.7, we have used the fact that $\mathsf{L}(G_{E_S^p})$ is 1-unambiguous, for each trim, 1-unambiguous regular expression E in star normal form. To prove this fact, we define a regular expression E_S^q that denotes $\mathsf{L}(G_{E_S^q})$ and show that E_S^q

is 1-unambiguous and in star normal form if E is. For $I = \{i_1, ..., i_n\}$, $n \geqslant 0$, we use $\bigoplus_{i \in I} E_i$ as shorthand for the regular expression $E_{i_1} + \cdots + E_{i_n}$.

DEFINITION 5.7. Let E be a trim regular expression, S be a set of symbols, and q be a state of G_E; that is, $q = q_I$ or $q \in sym(E')$. We define a regular expression E_S^q that denotes $\mathsf{L}(G_{E_S^q})$ by induction on E as follows:

$E = \varnothing$ or $E = \varepsilon$: $E_S^q = E$.

$$E = a\colon E_S^q = \begin{cases} a, & \text{if } q = q_I, \\ \varepsilon, & \text{if } q \in sym(E'). \end{cases}$$

$$E = F + G\colon E_S^q = \begin{cases} \varepsilon + \bigoplus_{a \notin S} (aa\backslash F_S^{q_I}) + \bigoplus_{a \notin S} (aa\backslash G_S^{q_I}), & \text{if } q = q_I, \varepsilon \in \mathsf{L}(E), \\ F_S^q + G_S^q, & \text{if } q = q_I, \varepsilon \notin \mathsf{L}(E), \\ F_S^q, & \text{if } q \in sym(F'), \\ G_S^q, & \text{if } q \in sym(G'). \end{cases}$$

$$E = FG\colon E_S^q = \begin{cases} F_S^q(\varepsilon + \bigoplus_{a \notin S} (aa\backslash G_S^{q_I})), & \text{if } q = q_I \text{ or } q \in sym(F'), \varepsilon \in \mathsf{L}(G), \\ F_\varnothing^q G_S^{q_I}, & \text{if } q = q_I \text{ or } q \in sym(F'), \varepsilon \notin \mathsf{L}(G), \\ G_S^q, & \text{if } q \in sym(G'). \end{cases}$$

$E = F^*\colon E_S^q = F_S^q(\bigoplus_{a \notin S} (aa\backslash F_S^{q_I}))^*$.

It is a straightforward, though laborious, induction to show that $\mathsf{L}(G_{E_S^q}) = \mathsf{L}(E_S^q)$, for each trim regular expression E.

LEMMA 5.8. *Let E be a trim regular expression, S be a set of symbols, and q be a state of G_E. If E is 1-unambiguous and in star normal form, then so is E_S^q.*

Proof. The proof is by induction on E. We apply the inductive definition of 1-unambiguity given in Lemma 3.2. Furthermore, we make use of the fact that $first(aa\backslash H) \subseteq first(H)$, because, if $a \notin first(H)$, then $\mathsf{L}(a\backslash H) = \varnothing$ and, thus, $first(aa\backslash H) = \varnothing$.

We establish the three most interesting cases. First, let $E = F + G$, $q = q_I$, and $\varepsilon \in \mathsf{L}(E)$. By the induction hypothesis, $F_S^{q_I}$ and $G_S^{q_I}$ are 1-unambiguous and in star normal form; hence, by Theorem B, so are $a\backslash F_S^{q_I}$ and $a\backslash G_S^{q_I}$. Furthermore, $first(aa\backslash F_S^{q_I}) \cap first(aa\backslash G_S^{q_I})$ is a subset of $first(F_S^{q_I}) \cap first(G_S^{q_I})$ and, thus, of $first(F) \cap first(G)$, which is empty. Therefore, E_S^q is also 1-unambiguous and in star normal.

Second, let $E = FG$, $q = q_I$ or $q \in sym(F')$, and $\varepsilon \notin \mathsf{L}(G)$. Then, $followlast(F_\varnothing^q) \subseteq followlast(F)$; hence, $followlast(F_\varnothing^q) \cap first(G_S^{q_I}) \subset followlast(F) \cap first(G) = \varnothing$. Furthermore, if $\varepsilon \in \mathsf{L}(F_\varnothing^q)$ and $q = q_I$, then $\varepsilon \in \mathsf{L}(F)$ and $first(F_\varnothing^q) = first(F)$; hence, $first(F_\varnothing^q) \cap first(G_S^{q_I}) \subseteq first(F) \cap first(G) = \varnothing$. On the other hand, if $\varepsilon \in \mathsf{L}(F_\varnothing^q)$ and $q \in sym(F')$, then $q \in last(F')$, because F is trim; hence, $first(F_\varnothing^q) \cap first(G_S^{q_I}) \subseteq followlast(F) \cap first(G) = \varnothing$. Finally, by the induction hypothesis, $F_\varnothing^q$ and $G_S^{q_I}$ are 1-unambiguous and in star normal form. Thus, E_S^q is also 1-unambiguous and in star normal form.

Third, let $E = F^*$. Then, $followlast(F_S^q) \cap first(aa\backslash F_S^{q_I}) \subseteq followlast(F) \cap first(F) = \varnothing$, since E is in star normal form. Next, if $\varepsilon \in \mathsf{L}(F_S^q)$, then $q \neq q_I$, because E is in star normal form implies that $\varepsilon \notin \mathsf{L}(F)$; hence, as in the previous case, $first(F_S^q) \cap first(aa\backslash F_S^{q_I}) \subseteq followlast(F) \cap first(F) = \varnothing$. Finally, $followlast(aa\backslash F_S^{q_I}) \subseteq followlast(F)$; hence, for

$H = \bigoplus_{a \notin S} (aa \backslash F_S^{q_I})$, $followlast(H) \cap first(H) \subseteq followlast(F) \cap first(F) = \varnothing$. Therefore, $followlast(H') \cap first(H') = \varnothing$. By the induction hypothesis and Theorem B, E_S^q is 1-unambiguous and in star normal form. ▮

To prove Theorem F, we need to apply Lemma 5.7 and Theorem E. We now prove that the preconditions of Theorem E are satisfied.

LEMMA 5.9. *Let M be a minimal DFA and S be an M-consistent set of symbols. Then, M_S^q is minimal, for each state q of M.*

Proof. We show that no two states of M_S are equivalent in M_S. Let p and q be two distinct states of M and w be a word of minimal length that distinguishes between p and q; that is, that takes one, but not both, of p and q to a final state in M. If $w = uav$ and $a \in S$, then the minimality of w implies that u takes neither p nor q to a final state in M, or u takes both p and q to a final state in M. However, if u takes p and q to a final state in M, then ua takes p and q to the same state $f(a)$, because a is M consistent; thus, in this case, w cannot distinguish between p and q. Therefore, u takes neither p nor q to a final state in M; hence, w distinguishes between p and q in M_S. ▮

Proof of Theorem F. Let M be a minimal DFA and S be an M-consistent set of symbols. By Lemma 5.9, M_S^q is minimal, for each state q of M. By Lemma 5.7 and Theorem E, $\mathsf{L}(M)$ is 1-unambiguous if and only if each state q of M satisfies the following two conditions:

1. M_S^q satisfies the orbit property and is trim.
2. Each orbit language of M_S^q is 1-unambiguous.

However, the orbit automaton of q in M_S^q is identical to the orbit automaton of q in M_S and the transitions that leave the orbit of q in M_S^q are the same as the transitions that leave the orbit of q in M_S. Therefore, $\mathsf{L}(M)$ is 1-unambiguous if and only if the two conditions in the theorem hold.

Now let M consist of a single, nontrivial orbit and $\mathsf{L}(M)$ be 1-unambiguous. Lemmas 5.4 and 5.5 imply that there is a 1-unambiguous regular expression E^* such that $\mathsf{L}(M)$ is a derivative of $\mathsf{L}(E^*)$; that is, $\mathsf{L}(M) = \mathsf{L}(G_{E^*}^q)$, for some state q. Furthermore, $first(E)$ is $G_{E^*}^q$ consistent. Since M is the minimization of $G_{E^*}^q$, $first(E)$ is also M consistent. Because M is a nontrivial orbit, $first(E) \neq \varnothing$, which concludes the proof. ▮

Theorem F gives rise to the following decidability result for 1-unambiguous regular languages, which we prove after establishing one preliminary result.

THEOREM G. *It is decidable whether the language of a DFA M is* 1*-unambiguous and the decision algorithm runs in time quadratic in the size of M. If the language of the DFA M is* 1*-unambiguous, then an equivalent* 1*-unambiguous regular expression can be constructed in time exponential in the size of M.*

Starting from a minimal DFA M and a set S of M-consistent symbols, we apply Theorem F recursively to the orbit automata of M_S. For this reason we first ensure that the orbit automata are also minimal.

LEMMA 5.10. *Let M be a minimal DFA that satisfies the orbit property. Then, M_q is minimal, for each state q of M.*

Proof. Let w be a word in Σ^* that distinguishes between two states p and q in M that belong to the same orbit of M. Then, w takes one, but not both of p and q to a final state in M.

Let u be the longest subword of w such that both computations with u starting from p and from q stay within M_q. If $u = w$, then w distinguishes between p and q in M_q. Otherwise, consider the symbol a in Σ such that ua is a subword of w. We assume without loss of generality, that the computation of ua starting from p does not stay within M_q. There can be two reasons for this occurrence. The first possibility is that u takes p to a final state in M_q and there is an a transition from the final state out of the orbit of q. Then, u does not take q to a final state in M_q, because otherwise ua would take p and q to the same state in M (by the orbit property). Hence, w cannot distinguish between p and q in M. Therefore, u distinguishes between p and q in M_q. The second possibility is that u takes p to a state in M_q that has no a transition in M. Then, w takes q to a final state in M. Thus, there is a subword v of w that is at least as long as u and takes q to a final state in M_q. This word v distinguishes p from q in M_q. ∎

Proof of Theorem G. Since the minimal DFA for each DFA M can be computed in time quadratic in the size of M [HoU79], we can assume without loss of generality that M is minimal. Fig. 6 defines a function 1-*unambiguous*(M) that returns true if and only if $\mathsf{L}(M)$ is 1-unambiguous.

After statement 5 of Figure 6, each orbit automaton of M_S is smaller than M and is minimal, since M_S has the orbit property. Therefore, the algorithm halts.

If x and y belong to the same orbit of M_S, then $\mathsf{L}((M_S)_x)$ is a derivative of $\mathsf{L}((M_S)_y)$ and vice versa; hence, $\mathsf{L}((M_S)_x)$ is 1-unambiguous if and only if $\mathsf{L}((M_S)_y)$ is 1-unambiguous. Therefore, by Theorem F, the algorithm is correct.

Because the strongly connected components of a directed graph can be computed in linear time [AHU74], Steps 1 through 5 of the algorithm take time linear in the size of M. If we sum the sizes of all automata $(M_S)_x$ that are considered in Step 6, we get at most the size of M. Thus, the total run time of the algorithm is quadratic in the size of M.

Finally, let k be the size of the alphabet Σ and $\mathsf{L}(M)$ be 1-unambiguous. If $S \neq \varnothing$, then $\mathsf{L}(M_S)$ and $\mathsf{L}(M_S^{f(a)})$, $a \in \Sigma$, are 1-unambiguous and the cuts of M

```
function 1-unambiguous(M : minimal DFA) : Boolean;
1   compute S := {a in Σ | a is M consistent};
2   if M has a single, trivial orbit then return true
3   if M has a single, nontrivial orbit and S = ∅ then return false;
4   compute the orbits of M_S;
5   if not OrbitProperty(M_S) then return false;
6   for each orbit K of M_S do
        choose x in K;
        if not 1-unambiguous((M_S)_x) then return false;
    end;
7   return true;
```

FIG. 6. The decision algorithm for 1-unambiguity of languages denoted by minimal DFAs.

are smaller than M. Hence, if E and E_a are 1-unambiguous regular expressions denoting $\mathsf{L}(M_S)$ and $\mathsf{L}(M_S^{f(a)})$, $a \in \Sigma$, then, by the proof of Lemma 5.7, the 1-unambiguous regular expression $E(\bigoplus_{a \in S} aE_a)^*$ denotes $\mathsf{L}(M)$. On the other hand, if $S = \varnothing$, then M has more than one orbit. If $a_1, ..., a_n$ are the distinct symbols of the transitions that leave the orbit of the initial state q_I of M and $q_1, ..., q_n$ are the states they lead to, then M_{q_I} and M^{q_i}, $1 \leqslant i \leqslant n$, are minimal automata that are smaller than M. If E and E_i are 1-unambiguous regular expressions that denote $\mathsf{L}(M_{q_I})$ and $\mathsf{L}(M^{q_i})$, $1 \leqslant i \leqslant n$, then, by the proof of Theorem E, the 1-unambiguous regular expression $E \oplus_{1 \leqslant i \leqslant n} a_i E_i$ denotes $\mathsf{L}(M)$. Hence, a 1-unambiguous regular expression for $\mathsf{L}(M)$ can be built from at most $k+1$ regular expressions for smaller automata, in run time that is exponential in the size of M. ▌

It is well known that, for each word w, the minimal DFA for the language $\Sigma^* w$ has size linear in the length of w [KMP77]. Using the algorithm in Figure 6, it is not hard to see that the language $\Sigma^* w$ is 1-unambiguous. However, we can show that the smallest 1-unambiguous regular expression that denotes this language has size exponential in the length of w. Therefore, DFAs are exponentially more succinct for 1-unambiguous languages than are 1-unambiguous regular expressions and the exponential run time for constructing a 1-unambiguous regular expression from a minimal DFA is worst-case optimal. Furthermore, regular expressions are exponentially more succinct for 1-unambiguous languages than are 1-unambiguous regular expressions. In particular, the finite language $(0+1)^{\leqslant n}1(0+1)^n$ can be denoted by a regular expression that has size linear in n, but the smallest DFA and, hence, the smallest 1-unambiguous regular expression for it, has size exponential in n [KW80].

6. CONCLUSIONS

The definition of 1-unambiguity in the SGML standard gives rise to at least two questions, both of theoretical interest and of relevance for systems that support SGML.

1. Is it decidable, given a model group E, whether E is unambiguous? And, if it is, then what is its time complexity?

2. Is it decidable, given a regular language L (represented, for example, by a finite-state automaton), whether L can be represented by an unambiguous model group? And, if it is, then what is its time complexity?

We have demonstrated in Section 2 the decidability of the first question for standard regular expressions; the implicit algorithm has a time complexity that is quadratic in the size of the regular expression. Brüggemann-Klein [BK92a, BK93c] has designed a linear-time algorithm for standard regular expressions, Ponty *et al.* [PZD97] have shown how to represent the Glushkov automaton in linear space, and Hromkovič *et al.* [HSW97] have modified the Glushkov construction to obtain an automaton that has size subquadratic in the size of the regular expression. For model groups, Brüggemann-Klein [BK93a, BK93b] has demonstrated decidability.

We have presented a solution for the second question for standard regular expressions. We leave, as an open problem, the characterization of the family of languages that can be denoted by 1-unambiguous model groups. A further research topic is to investigate k-unambiguous regular languages and regular expressions, when the lookahead is a constant $k \geqslant 1$.

Received December 31, 1993; final manuscript received September 30, 1997

REFERENCES

[AHU74] Aho, A. V., Hopcroft, J. E., and Ullman, J. D. (1974), "The Design and Analysis of Computer Algorithms," Addison-Wesley Series in Computer Science and Information Processing, Addison-Wesley, Reading, MA.

[ASU86] Aho, A. V., Sethi, R., and Ullman, J. D. (1986), "Compilers: Principles, Techniques, and Tools," Addison-Wesley Series in Computer Science, Addison-Wesley, Reading, MA.

[AH97] Ahonen, H. (1997), Disambiguation of SGML content models, *in* "PODP 96" (C. Nicholas and D. Wood, Eds.), Lecture Notes in Computer Science, Vol. 1293, pp. 27–37, Springer-Verlag, Berlin.

[BE96] Berstel, J., and Pin, J.-E. (1996), Local languages and the Berry–Sethi algorithm, *Theoret. Comput. Sci.* **155**(2), 439–446.

[BEGO71] Book, R., Even, S., Greibach, S., and Ott, G. (1971), Ambiguity in graphs and expressions, *IEEE Trans. Comput.* **C-20**(2), 149–153.

[BK92a] Brüggemann-Klein, A. (1992), Regular expressions into finite automata, *in* "Latin '92" (I. Simon, Ed.), Lecture Notes in Computer Science, Vol. 583, pp. 87–98, Springer-Verlag, Berlin.

[BK93a] Brüggemann-Klein, A. (1993), Formal models in document processing. Habilitationsschrift. Faculty of Mathematics at the University of Freiburg.

[BK93b] Brüggemann-Klein, A. (1993), Unambiguity of extended regular expressions in SGML document grammars, *in* "Algorithms-ESA '93" (Th. Lengauer, Ed.), pp. 73–84, Springer-Verlag, Berlin.

[BK93c] Brüggemann-Klein, A. (1993), Regular expressions into finite automata, 1992, *Theoret. Comput. Sci.* **120**(2), 197–213.

[BKW92] Brüggemann-Klein, A., and Wood, D. (1992), Deterministic regular languages, *in* "STACS 92" (A. Finkel and M. Jantzen, Eds.), Lectures Notes in Computer Science, Vol. 577, pp. 173–184, Springer-Verlag, Berlin.

[Brz64] Brzozowski, J. A. (1964), Derivatives of regular expressions, *J. Assoc. Comput. Mach.* **11**(4), 481–494.

[BS86] Berry, G., and Sethi, R. (1986), From regular expressions to deterministic automata, *Theoret. Comput. Sci.* **48**, 117–126.

[Cha97] Champarnaud, J.-M. (1997), From a regular expression to an automaton. Rapport Technique LIR97.08, Laboratoire d'Informatique de Rouen, Université de Rouen.

[CP92] Chang, C.-H., and Paige, R. (1992), New theoretical and computational results for regular languages. Technical report 587, Courant Institute, New York University. Proceedings of the Third Symposium on Combinatorial Pattern Matching.

[CP97] Chang, C.-H., and Paige, R. (1997), From regular expressions to DFAs using compressed NFAs, *Theoret. Comput. Sci.* **178**(1–2), 1–36.

[Eil74] Eilenberg, S. (1974), "Automata, Languages, and Machines," Vol. A, Academic Press, New York.

[GH67] Ginsburg, S., and Harrison, M. A. (1967), Bracketed context-free languages, *J. Comput. System Sci.* **1**(1), 1–23.

[Glu61] Glushkov, V. M. (1961), The abstract theory of automata, *Russian Math. Surveys* **16**, 1–53.

[Gol90] Goldfarb, C. F. (1990), "The SGML Handbook," Clarendon Press, Oxford.

[HoU79] Hopcroft, J. E., and Ullman, J. D. (1979), "Introduction to Automata Theory, Languages and Computation," Addison-Wesley Series in Computer Science. Addison-Wesley, Reading, MA.

[HSW97] Hromkovič, J., Seibert, S., and Wilke, Th. (1997), Translating regular expressions into small ε-free nondeterministic finite automata, *in* "STACS 97" (R. Reischuk and M. Morvan, Eds.), Lecture Notes in Computer Science, Vol. 1200, pp. 55–66, Springer-Verlag, Berlin.

[ISO86] International Organization for Standardization (1986), ISO 8879: Information processing—Text and office systems—Standard Generalized Markup Language (SGML).

[KILW97] Kilpeläinen, P., and Wood, D. (1997), SGML and exceptions, *in* "PODP 96" (C. Nicholas and D. Wood, Eds.), Lecture Notes in Computer Science, Vol. 1293, pp. 39–49, Springer-Verlag, Berlin.

[KMP77] Knuth, D. E., Morris, J., and Pratt, V. (1977), Fast pattern matching in strings, *SIAM J. Comput.* **6**(2), 323–350.

[KW80] Kintala, C. M. R., and Wotschke, D. (1980), Amounts of nondeterminism in finite automata, *Acta Inform.* **13**, 199–204.

[Lei81] Leiss, E. (1981), The complexity of restricted regular expressions and the synthesis problem of finite automata, *J. Comput. System Sci.* **23**(3), 348–354.

[Mir66] Mirkin, B. G. (1966), An algorithm for constructing a base in a language of regular expressions, *Engrg. Cybernet.* **5**, 110–116.

[MY60] McNaughton, R., and Yamada, H. (1960), Regular expressions and state graphs for automata, *IRE Trans. Electronic. Comput.* **EC-9**(1), 39–47.

[Per90] Perrin, D. (1990), Finite automata, *in* "Handbook of Theoretical Computer Science," (J. van Leeuwen, Ed.), Vol. B, pp. 1–57, Elsevier and MIT Press, Amsterdam and Cambridge, MA.

[PZD97] Ponty, J.-L., Zaidi, D., and Champarnaud, J.-M. (1997), A new quadratic algorithm to convert a regular expression into an automaton, *in* "WIA 96" (D. R. Raymond, D. Wood, and S. Yu, Eds.), Lecture Notes in Computer Science, Vol. 1260, pp. 109–119, Springer-Verlag, Berlin.

[Wat95] Watson, B. W. (1995), "Taxonomies and Toolkits of Regular Language Algorithms," Ph.D. thesis, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands.

[Woo87] Wood, D. (1987), "Theory of Computation," Wiley, New York.