states before the triviality of the automorphism group is realized. The utility of this result is enhanced by the fact that the length of a state is easily computed [2].

2. Preliminaries

The notation, definitions, and results in this section are collected mostly from [1]. For a nonempty set Σ, we denote by Σ^* the *free monoid* over Σ, i.e. the set of all strings of finite length of members of Σ including the empty string ϵ.

An *automaton* is a triple $A = (S, \Sigma, \delta)$, where S is a set (*of states*), Σ is a nonempty set (*the input alphabet*), and $\delta: S \times \Sigma^* \rightarrow S$ is the *transition function* satisfying: $\forall s \in S$ and $\forall x,y \in \Sigma^*$, $\delta(s, xy) = \delta[\delta(s, x), y]$; and $\delta(s, \epsilon) = s$, $\forall s \in S$.

An automaton $B = (T, \Sigma, \delta')$ is a *subautomaton* of $A = (S, \Sigma, \delta)$, written $B \ll A$, if and only if $T \subseteq S$ and δ' is the restriction of δ to $T \times \Sigma^*$. We use δ for δ', as no ambiguity arises. S_B denotes the set of states of an automaton B.

The set of *successors* of $s \in S$ is $\delta(s) = \{\delta(s, x) : x \in \Sigma^*\}$. The *automaton generated* by $s \in S$ is $\langle s\rangle = (\delta(s), \Sigma, \delta)$; i.e. the subautomaton whose set of states is the set of successors of s. An automaton $A = (S, \Sigma, \delta)$ is *singly generated* if and only if $\exists s \in S$ such that $A = \langle s\rangle$ and in that event s is a generator of $\langle s\rangle$. The *set of generators* of $\langle s\rangle$ is gen $\langle s\rangle = \{r \in S_{\langle s\rangle} : \langle r\rangle = \langle s\rangle\}$.

An automaton is *finite* if and only if its set of states is finite. The cardinality of a set S is denoted by $|S|$.

An automorphism of the automaton $A = (S, \Sigma, \delta)$ is a monic mapping f of S onto S (and the identity mapping on Σ^*) satisfying $f[\delta(s, x)] = \delta[f(s), x]$, $\forall s \in S$, $\forall x \in \Sigma^*$. The *set* (group) *of automorphisms* of an automaton A is denoted by $G(A)$. Where H is a subgroup of $G(A)$ and s is a state of $A = (S, \Sigma, \delta)$, the *H-orbit* of s is $O_H(s) = \{h(s) : h \in H\}$.

For each $u \in \Sigma^*$, where $u = x_1 \cdots x_k$ and $x_i \in \Sigma$, $i \in \{1, \cdots, k\}$, the length of u is $/u/ = /x_1 \cdots x_k/ = k$. The *length* of a state s of A is

$$/s/ = \max_{r \in S_{\langle s\rangle}} \{\min_{u \in \Sigma^*} \{/u/ : \delta(s, u) = r\}\};$$

i.e. the length of the shortest route to the state farthest from s.

3. A Divisibility Bound on $G(\langle s\rangle)$

The following three results are proved by the author in [1].

LEMMA 1. *An automorphism of $\langle s\rangle$ is completely determined by its value on s.*

LEMMA 2. *Where f is an automorphism of an automaton A and s is a state of A, $\langle f(s)\rangle = f(\langle s\rangle)$.*

LEMMA 3. *Let $A = (S, \Sigma, \delta)$, let p, $q \in S$, and let H be a subgroup of $G(A)$. Then $O_H(p)$ and $O_H(q)$ are either identical or disjoint.*

With the aid of the three lemmas we now have:

THEOREM. *Let $\langle s\rangle = (S, \Sigma, \delta)$ be a finite automaton, let $M = \{m \in$ gen $\langle s\rangle : /m/ \leqq /s/, \forall s \in S\}$, and let H be a subgroup of $G(\langle s\rangle)$. Then $|H|$ divides $|M|$.*

PROOF. Let $r \in$ gen $\langle s\rangle$ and let $f \in G(\langle s\rangle)$. Then $f(r) \in$ gen $\langle s\rangle$, by Lemma 2. Thus, since gen $\langle s\rangle$ is finite, $f($gen $\langle s\rangle) =$ gen $\langle s\rangle$, i.e. automorphisms preserve generators. For any $t \in S$ and any $x,y \in \Sigma^*$, $f[\delta(t, x)] = f[\delta(t, y]$ if and only if $\delta(t, x) = \delta(t, y)$ and hence $/f(t)/ = /t/$, i.e. automorphisms preserve length. Therefore, $f(M) = M$.

By Lemma 1, distinct automorphisms have distinct images on members of gen $\langle s\rangle$ and thus $|O_H(t)| = |H|$, $\forall t \in$ gen $\langle s\rangle$. Thus, by Lemma 3, H partitions M into disjoint subsets of the form $O_H(t)$, and hence $|H|$ divides $|M|$. ∎

REFERENCES

1. BAVEL, Z. Structure and transition-preserving functions of finite automata. *J. ACM 15*, 1 (Jan. 1968), 135–158.
2. ——. Algorithms in the structure and transition-preserving functions of finite automata. Submitted to a technical journal.
3. WEEG, G. P. The structure of an automaton and its operation preserving transformation group. *J. ACM 9*, 3 (July 1962), 345–349.

ALGORITHMS

Remarks on Algorithms with Numerical Constants

C. B. DUNHAM

University of Western Ontario, London, Canada*

Keywords and Phrases: numerical algorithm, numerical constants

CR Categories: 5.10

Algorithms continue to be published in which undefined mathematical constants appear as a finite number of decimal digits. Such constants even appear in algorithms which explicitly claim to be of arbitrary precision; for example, Algorithm 349 [*Comm. ACM 12* (Apr. 1969), 213–214] has an undefined constant *piq* given to 48 decimal digits. Such algorithms are not useful in high precision unless the author defines all constants and tells how they can be obtained. It should be required of all published algorithms that all constants be defined or that working precision be explicitly stated.

[EDITOR'S NOTE. I agree completely with the suggested requirement and will try to enforce it in the future.—*L.D.F.*]

* Computer Science Department