JOURNAL OF COMPUTER AND SYSTEM SCIENCES 10, 340–350 (1975)

Deterministic One-Counter Automata

LESLIE G. VALIANT*

Centre for Computer Studies, University of Leeds, Leeds, LS2 9JT, United Kingdom

AND

MICHAEL S. PATERSON

Department of Computer Science, University of Warwick, Coventry, CV4 7AL, United Kingdom

Received January 15, 1974

The equivalence problem for deterministic one-counter automata is shown to be decidable. A corollary for schema theory is that equivalence is decidable for Ianov schemas with an auxiliary counter.

1. INTRODUCTION

We present an analysis of deterministic one-counter automata in order to show that the equivalence problem for them is decidable. All our arguments and results can be translated directly into schema theoretic terms. The corollary that then follows is that equivalence is decidable for Ianov schemas even when these are allowed an auxiliary counter.

A deterministic one-counter automaton (doca) is a deterministic pushdown automaton (dpda) [1] with a stack alphabet of just one symbol. It is known that this restriction of stacks to counters leaves the decidability properties of several related problems invariant. For example it can be derived from a result of Minsky [3] about two-register machines, that inclusion for deterministic one-counter, equivalence for nondeterministic one-counter, and emptiness for two-counter automata, are all undecidable, as are their counterparts with stacks. Our positive result for doca, together with positive solutions for other subfamilies [6], can therefore be interpreted as lending weight to the conjecture that equivalence is decidable for the class of all dpda.

In Section 3 we establish some preliminary technical results that are used in Section 4

* Supported during the research by a grant from the Science Research Council of the United Kingdom.

to show that the configurations of a doca obey certain periodic relationships. In Section 5 we describe how a decision procedure for equivalence can be derived as a consequence of this periodicity. The procedure also incorporates an extension of the technique of simulation first introduced by Rosenkrantz and Stearns [5]. To gain an overview of our strategy, this section may be read first. Section 6 establishes that the time complexity of the decision procedure is bounded above by an expression exponential in about the square root of the number of states. Section 7 gives some further applications of the analysis and results.

2. Definitions

A deterministic one-counter automaton M is described by a set Q of q states $\{s_i\}$, a distinguished starting state, a set of accepting states, a finite input alphabet, and, a set of transition rules. For each combination of state and emptiness condition of counter, the transitions specify *either* an *ϵ-move* (i.e., one to be executed without inputs), *or* a unique *reading move* for each input character. We shall assume that M is in a *normal form* where each move can change the stack height by at most one, where each input string can be read in a finite number of steps, and where acceptance can only occur in configurations which are about to read new input characters. This assumption is justified since there are well-known constructions [6] by which any dpda can be transformed into such a normal form, without changing the stack alphabet.

A *configuration* c of M is described by (s, n) where s is a state, and n is a nonnegative integer representing the contents of the counter. Then the *height* of c, denoted by $| c |$, is just n. We define the configuration $(s, n) + m$ to be $(s, n + m)$ provided that $n + m \geqslant 0$. A derivation $c \xrightarrow{\alpha} c'$ is a sequence of moves specified by the transition rules, that leads from c to c', and, in the process, reads the word α over the input alphabet. It is a *positive derivation*, written as $c \xrightarrow[+]{\alpha} c'$, if no intermediate configuration in the derivation has an empty counter.

A word α *distinguishes* the configurations c, c' iff derivations reading α can take one to a configuration with an accepting state, but not the other. The *length* $| \alpha |$ of α is the number of characters in α. The *rank* of a pair c, c', denoted by rank (c, c'), is the length of a shortest string distinguishing c and c', if one exists, and ∞ otherwise. Two configurations are equivalent, written as $c \equiv c'$, iff their rank is ∞. Two machines are equivalent iff $(s, 0) \equiv (s', 0)$ where s, s' are their starting states.

3. Preliminary Results

The dominating factor in the bounds we shall derive is the function $S(q)$ that is defined as follows.

DEFINITION. $S(q) = \max\{\text{l.c.m.}\{n_i\} \mid \sum n_i = q, n_i > 0\}$. Using number theoretic arguments the following well-known result which we shall not prove here can be derived [2, Section 61].

LEMMA 1.

$$S(q) = e^{(q \cdot \log_e q)^{1/2}} \cdot 1(q), \qquad \textit{where} \quad 1(q) \to 1 \qquad \textit{as} \quad q \to \infty.$$

Derivations of the following periodic form play a central role in our proofs.

DEFINITION. The input word β is a *standard sequence* for the configurations c, c' iff

(i) β is a shortest string such that $c \xrightarrow[+]{\beta} c'$;

(ii) $\beta = \beta_1\beta_2{}^r\beta_3$ where $|\beta_1\beta_3| < q^2$, $|\beta_2| \leqslant q$, and $r > 0$; and

(iii) for some state s_e and positive integers w and d, for $v = 0, \ldots, r$,

$$c \xrightarrow[+]{\beta_1\beta_2{}^v} (s_e, w - vd).$$

In the Appendix we prove, for a suitably defined number X, the following fundamental property of positive derivations.

LEMMA 2. *For each q-state doca there is an integer X, $0 < X \leqslant S(q)$, such that if $|c| - |c'| \geqslant q^2$ and $c \xrightarrow[+]{} c'$, then there is a standard sequence for c, c' in which the loop drop d divides X.*

Proof. See Appendix. ∎

X is defined in the Appendix to be the least common multiple of the net stack drops due to a certain set of disjoint loops in the state diagram. These certainly include all ϵ-loops, i.e., those which involve only ϵ-moves when the counter is not empty.

As we are concerned only with asymptotic bounds, and as $S(q)$ clearly dominates any fixed polynomial in q as q becomes large, it will be sufficient for our purposes to prove the existence of, rather than obtain specific expressions for, the various polynomials we derive. The proof of the following lemma introduces a useful technique.

LEMMA 3. *There is a polynomial p_3 such that for any configuration c with $|c| \geqslant p_3(q)$, and any positive multiple Y of X,*

(i) $\text{rank}(c, c + Y) + Y/q \leqslant \text{rank}(c + Y, c + 2Y) \leqslant \text{rank}(c, c + Y) + Yq$,

(ii) $c \equiv c + Y$ iff $c + Y \equiv c + 2Y$.

Proof. Assume c and $c + Y$ are distinguishable. Provided that p_3 is sufficiently large, there must be a minimal distinguishing sequence $\beta\delta$ where

$$c \xrightarrow[+]{\beta} (s, q^2)$$

for some s, and then Lemma 2 ensures that β may be taken to be in the form of a standard sequence $\beta_1\beta_2{}^r\beta_3$. Let the drop due to β_2 be d, where $d > 0$. Since $\beta\delta$ distinguishes c and $c + Y$, clearly $\beta_1\beta_2^{r+Y/d}\beta_3\delta$ distinguishes $c + Y$ and $c + 2Y$. Since $|\beta_2| \leqslant q$ and $d \geqslant 1$, the right-hand inequality is proved.

In a similar fashion we can choose $\beta_1\beta_2{}^r\beta_3\delta$ to be a minimal string distinguishing $c + Y$ and $c + 2Y$, where, if d is the stack drop due to β_2 then $0 < d \leqslant q$ and $r > Y/d$. Therefore $\beta_1\beta_2^{r-Y/d}\beta_3\delta$ also distinguishes c and $c + Y$. β_2 cannot be null, for then the drop due to its ϵ-loop would divide X, and thus also Y, and therefore c and $c + Y$ would not be distinguished. Thus the left-hand inequality is established.

Statement (ii) is an immediate consequence of (i) ∎

4. Propriety

We establish next some relationships that hold for periodic sets of configurations.

Definition. A configuration c is *improper* iff $c \equiv c + mX$ for all integers m (not necessarily positive) such that

$$|c| + mX \geqslant p_3(q).$$

The significance of this property is illuminated by the following lemma.

Lemma 4. *If $c \equiv c + mX$ for some $m > 0$ and $|c| \geqslant p_3(q)$ then c is improper.*

Proof. It is easy to see that for any set of configurations $\{c_1, \ldots, c_n\}$,

$$\operatorname{rank}(c_n, c_1) \geqslant \min_{1 \leqslant j < n} \{\operatorname{rank}(c_j, c_{j+1})\}.$$

Hence if $\operatorname{rank}(c, c + mX) = \infty$, then

$$\operatorname{rank}(c, c + X) \geqslant \min_{1 \leqslant i < m} \{\operatorname{rank}(c + iX, c + (i + 1)X)\}.$$

By Lemma 3(i) it follows that these ranks must all be ∞, and therefore by Lemma 3(ii), c is improper. ∎

Definition. A configuration is *proper* iff it is not improper.

Lemma 5. *There is a polynomial p_5 such that if $|c| > p_5(q) \cdot X$, $|c'| < q^2$ and $c \equiv c'$, then c is improper.*

Proof. Suppose that c is proper, and let $\beta_1\beta_2{}^r\beta_3\delta$ be a string distinguishing c and

$c + X$, constructed exactly as in the first part of the proof of Lemma 3, but for the case $Y = X$. We define c_n, c_n' for all $n \geqslant 0$, by

$$c \xrightarrow{\beta_1\beta_2^n} c_n \qquad \text{and} \qquad c' \xrightarrow{\beta_1\beta_2^n} c_n',$$

where, in the case of ϵ-moves, maximal derivations are taken. If $c \equiv c'$, then $c_n \equiv c_n'$ for all n. Also c_n is proper for all n, $n \leqslant r$, since $\beta_2^{r-n}\beta_3\delta$ distinguishes c_n and $c_n + X$. p_5 is chosen to ensure that r is sufficiently large for the following argument to work. Either in $c_0', \ldots, c_{(2q^4)}'$ some configuration repeats, or else some c_k' in this set has height at least $2q^3$. In the latter case it is easy to verify that for some i, j such that $i < j \leqslant 2q^4$,

$$c_i' \xrightarrow[+]{\beta_2^{j-i}} c_j' \qquad \text{and} \qquad c_j' = c_i' + w \qquad \text{for some} \quad w > 0.$$

In either case for some i, j such that $i < j \leqslant 2q^4$, we have, putting $l = (j - i)X$,

$$c'_{i+ml} = c_i' + mwX \qquad \text{for all} \quad m \geqslant 0 \text{ and some} \quad w \geqslant 0.$$

Trivially if $w = 0$, and by Lemma 3(i) if $w > 0$

$$\text{rank}(c'_{i+l}, c'_{i+2l}) \leqslant \text{rank}(c'_{i+2l}, c'_{i+3l}).$$

However, from the propriety of c_i, and Lemma 3(i), if p_5 is large enough

$$\text{rank}(c_{i+l}, c_{i+2l}) > \text{rank}(c_{i+2l}, c_{i+3l}).$$

This contradicts the assumption that $c_n \equiv c_n'$ for all n. ∎

We can now derive, as a consequence of this result, the property of equivalent configurations on which our decision procedure depends.

DEFINITION. Integers m, n are (x, y)-*rationally related* iff there exist integers a, b with $0 < a, b \leqslant x$, such that

$$| ma - nb | \leqslant y.$$

LEMMA 6. *There exist polynomials p_6, $\bar{p}_6$ such that if $c \equiv c'$, $| c | > \bar{p}_6(q) \cdot X$, and c is proper, then $| c |$, $| c' |$ are $(q^2, p_6(q) \cdot X)$-rationally related.*

Proof. Suppose that $c \equiv c'$, that c is proper, and that $| c | > \bar{p}_6(q) \cdot X$, where $\bar{p}_6$ is sufficiently large for the following argument to work. Construct $\beta_1\beta_2^r\beta_3\delta$ and define c_n, c_n' as in the proof of Lemma 5. $c_n \equiv c_n'$ for all n, and c_n is proper for $n \leqslant r$. Let l be the least n such that

$$\min(| c_n |, | c_n' |) < q^2.$$

l must exist, for otherwise $\{| c_n |\}$ would be an infinite strictly decreasing sequence. Suppose $| c_l | > p_5(q) \cdot X$, then clearly $l < r$, c_l is proper and we have a contradiction from Lemma 5.

On the other hand suppose $| c_l' | > p_5(q) \cdot X$, then c_l' is improper by Lemma 5. The sequence of the states of c_0', c_1',..., is ultimately periodic and there exist k, e with $0 < k \leqslant q$ and $| e | \leqslant q^3$ such that

$$c'_{l-mk} = c_l' + em$$

for all positive m such that $l - mk \geqslant q$. If $\bar{p}_6$ is large enough we can find $i \geqslant 0$ such that $i \equiv l \bmod k$, and $i + kX \leqslant r$. Then,

$$c_i' = c'_{i+kX} + eX \equiv c'_{i+kX} \text{ since } c_i' \text{ is improper.}$$

Therefore,

$$c_i \equiv c_{i+kX} = c_i - dkX,$$

which contradicts the propriety of c_i.

Thus $\max\{| c_l |, | c_l' |\} \leqslant p_5(q) \cdot X$ and so for a suitable choice of p_6' we have, in both cases,

$$| | c | - dl | < p_6'(q) \cdot X \qquad \text{and} \qquad | | c' | - el/k | < p_6'(q) \cdot X.$$

Since $0 < d \leqslant q$ and $0 < e/k \leqslant q^2$, it follows for some p_6 that

$$| | c | \cdot e/k - | c' | \cdot d | < p_6(q) \cdot X. \quad \blacksquare$$

5. Simulating Machine and Decision Procedure

Using the result of Lemma 6, we can construct a *nondeterministic* one-counter automaton which is able, in a certain sense, to simulate the computations of a pair of equivalent doca. By taking the disjoint union of the states and transitions of the two machines, we can regard the simulation as maintaining a representation of pairs of equivalent configurations, c and c', of the combined machine. We ensure that at each instant $a \cdot | c | - b \cdot | c' | = f$ for some integers a, b, f such that $0 < a$, $b \leqslant q^2$ and $| f | < p_6(q) \cdot X$. Then the simulating machine can represent c and c' by holding $a | c |$ in its counter, and remembering f, a, b, and the states of c and c', in its finite-state control.

The action of the simulating machine is as follows. Let p_0 be some polynomial such that whenever $| c |$ or $| c' |$ is greater than $p_0(q) \cdot X$, and they are rationally related, then they are related with respect to only one admissible rational ratio a/b. Then whenever $| c |$, $| c' |$ are both less than $p_0(q) \cdot X$, their values are recorded in the

finite-state control. When a simulation step is about to exceed this bound, the finite-state control determines the coefficients a, b, if any, and sets up the counter for the appropriate representation. When a simulation step would reach a pair of configurations not rationally related, say c is just too large for c', then by Lemma 6, if $c \equiv c'$ then c must be improper. Instead of continuing the simulation with (c, c'), a nondeterministic step is made *either* to the simulation of $(c - X, c')$ *or* to the simulation of $(c, c - X)$. Then if $c \equiv c'$, and so also $c \equiv c - X$, the simulation continues to be one for equivalent configurations in both cases.

If the original doca is in normal form as assumed, it is easy to arrange for the simulating machine to accept an input string if and only if one but not both of the simulated configurations results in acceptance. Thus if the starting configurations are indeed equivalent, then our discussion shows that no string is accepted. On the other hand, if they are inequivalent, we can show that some string must be accepted by the simulating machine. For if some α distinguishes the starting configurations, then either both derivations will be simulated directly to their different conclusions, or else the rational relationship will fail. In the latter case if (c, c') is reached where $c \not\equiv c'$ then the remainder of α distinguishes one of the new pairs created. The normal form we have assumed for the machines guarantees that any long ϵ-derivations cause net drops in stack height. This in turn ensures that further progress along α can always be made in a finite number of steps, and therefore that α will eventually be accepted.

The construction and testing for emptiness of the simulating machine described therefore constitutes a decision procedure for equivalence.

6. Complexity of Decision Procedure

It is a well-known result that the emptiness problem for nondeterministic pushdown automata is solvable, and there is a decision procedure which takes time that is bounded by a polynomial in the length of description of the tested machine [6]. For our one-counter simulating machine the number of states is at most $p(q) \cdot X^2$ for some polynomial p, where q is the total number of states of the two tested machines, and X is bounded above by $S(q) \doteq e^{(q \cdot \log q)^{1/2}}$. Assuming a fixed input alphabet, the description of this machine will be of length no more than a polynomial in this expression. Thus we can conclude the following.

Main Theorem. *The equivalence problem for doca is solvable, and there is a decision procedure which, for q state machines, has a running time bounded above by*

$$2^{k \cdot (q \cdot \log q)^{1/2}}$$

for some constant k.

7. Applications

Regularity

If a doca is to accept a regular set then it must have only a finite number of pairwise distinguishable configurations reachable from the starting configuration. This means certainly that no infinite set of proper configurations c, $c + X$,..., $c + iX$,..., can be visited. It follows easily that any configuration of height greater than q that can be visited, must be improper. From Lemma 4 and the fact that there are just q distinct configurations of any one height, we conclude that the number of pairwise distinguishable configurations that the machine can use is no more than $q(p_3(q) + X)$, which is asymptotically $qS(q)$. This is clearly also the upper bound for the number of states required in any finite-state automaton equivalent to a doca with q states.

This bound, and the period X used in the definition of improper configurations, can be almost achieved, as is shown by the following example. Let $\{n_i\}$ be a partition of $q - 1$ with l.c.m. $S(q - 1)$, and let M be a q-state doca with an input alphabet of $\{a\} \cup \{a_i\}$. M accepts the language

$$\{a^r a_i \mid n_i \text{ divides } r\}.$$

The transitions of M are such that for any $m \geqslant 0$,

$$(s', m) \xrightarrow{a} (s', m + 1) \qquad \text{and} \qquad (s', m) \xrightarrow{a_i} (s_i, m),$$

where s' is the starting state, and each s_i is in a distinct ϵ-loop with n_i states and a stack drop of n_i. If the accepting configurations are $\{(s_i, 0)\}$ then clearly the required language will be recognized.

All the configurations of this machine are improper. Moreover, the smallest positive integer i such that $c + i \equiv c$, for all c, is $X = S(q - 1)$. Also, for any $c = (s', m)$ and any j that is not a multiple of X, $c + j \not\equiv c$. Thus the bounds claimed for both the period and the regularity can be achieved to within a factor of q or q^2.

Semicanonical Machine

Since we can test arbitrary configurations of a doca for equivalence, we can test an arbitrary c for propriety. By first deciding for each s and each $0 < n \leqslant X$ whether (s, n) is improper, we can transform any machine into a form in which any improper configuration is immediately reduced via an ϵ-derivation to the smallest one equivalent to it. Thus we can obtain a semicanonical machine in which no "redundant" use of the counter will be made.

Schemas

The correspondence between doca and Ianov schemas with an auxiliary counter, is analogous to that between finite-state automata and ordinary Ianov schemas, or

between certain restricted dpda and monadic functional schemas. Details of these relationships can be found elsewhere (e.g. [4]). Using this correspondence, the decidability of the equivalence problem for Ianov schemas with a counter, can be deduced from our main theorem.

8. Conclusion

The class of deterministic one-counter automata is a natural extension of the class of finite-state machines. We have shown that in contrast with the inclusion and nullity of intersection problems, which become undecidable under this generalization the equivalence problem remains decidable.

We have established that these automata have certain periodic structural properties, and have derived an upper bound for the associated period that is asymptotically achievable. The resulting bound on the time complexity of the derived decision procedure is exponential in about the square root of the number of states of the tested machines. Whether a polynomial time test exists, remains open.

Appendix

For simplicity of exposition we prove Lemma 2 by first proving a slightly relaxed form of it, and then showing how the same argument can be strengthened.

Lemma 2′ *If* $| c | - | c' | \geqslant q^2$, $| c' | \geqslant q^2$ *and* $c \to c'$ *then there is a standard sequence for* c, c'.

Proof. We define the *efficiency* of state s to be the maximum value (possibly infinite) of $d/|\gamma|$, where for sufficiently large n, the derivation $(s, n + d) \xrightarrow{\gamma}_{+} (s, n)$ exists, but repeats no state except s at the beginning and end. We call such a derivation an efficient simple loop of s. Clearly $d, |\gamma| \leqslant q$.

Suppose α is a shortest string such that $c \xrightarrow{\alpha}_{+} c'$. We mark an occurrence of one of the states with greatest efficiency occurring in this derivation. Let this state be s_e and let its efficient simple loop be generated by γ and cause a drop of d, where $d > 0$. We first assume that $d < q$. Now excise from this derivation a set of disjoint state loops (with state repetitions within each one allowed) of maximal total length such that the total drop is a multiple of d and the marked occurrence of s_e is preserved. We can show that the length, m, of the remaining derivation is no more than $(d + 1)\, q - 2$, by first observing that at least $k = [(m - 2)/q - 2]$ disjoint simple loops, not containing the marked s_e in their interior, must occur in it. If $k \geqslant d$ then some nonnull subset of these loops accounts for a total drop which is a multiple of d.

This subset could therefore have been removed in the original excision, contrary to the maximality condition. Hence $(m + 2)/q - 2 \leqslant d - 1$, and so $m \leqslant (d + 1)q - 2 < q^2$.

Let β_1, β_3 be the input strings for the parts of this remaining derivation before and after the chosen occurrence of s_e, respectively. Then clearly for some integer r,

$$c \xrightarrow{\beta_1\beta_3} c' + rd.$$

But $| c | - | c' | \geqslant q^2 > m$ implies that $r > 0$. Thus,

$$c \xrightarrow{\beta_1\gamma^r\beta_3} c',$$

since during the derivation the counter cannot fall below $| c' | - m - \frac{1}{2}(q - d)$, which is positive. Since we have replaced arbitrary loops by ones of at least the same efficiency, the string $\beta_1\gamma^r\beta_3$ must still be of minimal length. We note that in our arguments we have not excluded the case of γ being null.

If $d = q$ then all states occur in the maximal efficient loop and any state can serve as s_e. In this case an easy argument establishes that the bounds of the lemma are sufficient.

To obtain the required strengthening of this result we investigate the set of possible values of d in the above construction. For each state s we select, if possible, a maximally efficient simple loop through s, and denote the set of states in this loop by Loop(s). Clearly if $s' \in$ Loop(s) then the efficiency of s' is greater than or equal to the efficiency of s. Also, any standard sequence whose principal loop is based on s could be replaced by one based on s', by applying the construction in the proof of the Lemma to the derivation of the old sequence, in which s' must occur, since, by definition, $r > 0$.

Let $s' \geqslant s$ be the transitive closure of the relation defined by $s' \in$ Loop(s). Defining s, s' to be equivalent iff $s' \geqslant s$ and $s \geqslant s'$, the relation $\geqslant$ becomes a partial ordering on the equivalence classes. Let $s_1, \ldots, s_k$ be a selection of representatives, one for each class that is maximal in the ordering. It is easily verified that the corresponding loops must be disjoint, and that standard sequences can always be based on some such loop. Then the drops due to all these must add up to no more than q, and also each one must divide X where $X = \text{lcm}\{d_i \mid d_i$ is the drop due to Loop(s_i), $1 \leqslant i \leqslant k\}$. Thus $X \leqslant S(q)$. This completes the proof of Lemma 2. ∎

References

1. J. E. Hopcroft and J. D. Ullman, "Formal Languages and Their Relation to Automata," Addison–Wesley, Reading, MA, 1969.
2. E. Landau, "Handbuch der Lehre von der Verteilung der Primzahlen," Teubner, Leipzig und Berlin, 1909. (Reprinted 1953, Chelsea).

3. M. L. MINSKY, "Computation: Finite and Infinite Machines," Prentice–Hall, Englewood Cliffs, NJ, 1967.
4. M. S. PATERSON, Decision problems in computational models, *in* "Proc. ACM Conference on Proving Assertions about Programs," Las Cruces, New Mexico, 1972.
5. D. J. ROSENKRANTZ AND R. E. STEARNS, Properties of deterministic top–down grammars, *Information and Control* **17** (1970), 226–256.
6. L. G. VALIANT, Decision procedures for families of deterministic pushdown automata, Ph.D. Thesis., Report No. 7 (1973), University of Warwick Computer Centre.