Simply exponential complexity for LR(1) grammars (?)

Theoretical Computer Science 88 (1991) 117–125
Elsevier

A graph-based regularity test for deterministic context-free languages

Priti Shankar

Department of Computer Science and Automation, Indian Institute of Science, Bangalore-560 012, India

B.S. Adiga

Systems Engineering Division, National Aeronautical Laboratory, Bangalore-560 017, India

Communicated by R. Siromoney
Received May 1987
Revised June 1989

Abstract

Priti, S. and B.S. Adiga, A graph-based regularity test for deterministic context-free languages, Theoretical Computer Science 88 (1991) 117–125.

It is shown that there exists a test of complexity $O((qt)^2 t^{qt})$ for testing the regularity of a deterministic context-free language, where q is the number of states and t, the stack alphabet size of the pushdown automaton derived from an LR(1) grammar for the language. The previously established upper bound for the test is t^{q^q}.

1. Introduction

A method of proof for the decidability of the question whether the language recognized by an arbitrary deterministic pushdown automaton (DPDA) is regular was suggested in [3] and involved a test whose complexity was bounded by $t^{q^{q^q}}$, where t is the stack alphabet size and q, the number of states of the DPDA. This bound was reduced to t^{q^q} in [4]. It is shown in this note that if the DPDA in question is represented by a finite graph, then it is possible to construct a test whose complexity is bounded by $(qt)^2 t^{qt}$.

2. Extended transition systems

Let $G=(N, T, P, S)$ be an LR(1) grammar for L. Let us refer to the deterministic finite automaton (DFA) for the canonical set of LR(1) items for G as the LR(1) DFA

0304-3975/91/$03.50 © 1991—Elsevier Science Publishers B.V.

for L. A DPDA can be constructed from the LR(1) parsing tables for L using the technique described below. This has states from the set

$$K=\{[q,\varepsilon]\}\cup\{[q_f,\$]\}\cup\{[q,a]: a\in T\}$$
$$\cup\{[q_\pi^l,a]: a\in T,\ \pi=A\rightarrow\alpha\in P,\ 0\leqslant l<\text{length}(\alpha)\}.$$

A state $[q,a]$ represents a state of the parser when the next lookahead is a and the next action is either a "shift" or a "reduce by π" action. States of the form $[q_\pi^l,a]$ are essentially handle-popping states; a state $[q_\pi^l,a]$ is reached while reducing by $\pi: A\rightarrow\alpha$ if l symbols of the handle remain to be popped. The state $[q_\pi^0,a]$ pushes on to the stack, a state of the LR(1) DFA corresponding to a transition on the left-hand side symbol in π; state $[q,\varepsilon]$ is the initial and $[q_f,\$]$, the final state of the DPDA. The stack symbol set Γ is the set of states $Q=\{q_1,q_2\ldots,q_n\}$ of the LR(1) DFA, where the start state q_1 is the start stack symbol of the DPDA. The DPDA has four types of moves. Let "action" and "goto" denote the standard LR parser actions.

(1) "*Accumulate initial lookahead*" *moves*:

$$\delta([q,\varepsilon],a,q_1)=([q,a],q_1)\quad\text{for all } a \text{ in } T.$$

(2) "*Shift terminal*" *moves*:

$$\delta([q,a],b,q_i)=([q,b],\text{goto }(q_i,a)),\quad\text{for all } a, b \text{ in } T,\ q_i \text{ in } Q.$$

(3) "*Reduce*" *moves*:

$$\delta([q,a],\varepsilon,q_i)=([q_\pi^l,a],\varepsilon)$$

if action $(q_i,a)=$reduce by π, $\pi=A\rightarrow\alpha$ and $l=\text{length}(\alpha)-1$.

$$\delta([q_\pi^l,a],\varepsilon,q_i)=([q_\pi^{l-1},a],\varepsilon)\quad\text{for all } q_i \text{ in } Q,\ l-1\geqslant 0.$$

$$\delta([q_\pi^0,a],\varepsilon,q_i)=([q,a],q_j)\quad\text{if goto }(q_i,A)=q_j.$$

(4) "*Accept*" *move*:

$$\delta([q_{\pi_0}^0,\$],\varepsilon,q_1)=([q_f,\$],q_1),$$

where π_0 is a production with the start symbol on the left-hand side and \$, the end of string marker. A finite graph called an extended transition system (ETS) can be derived from the DPDA. The node set X consists of nodes with labels encoding [DPDA state, top stack symbol] pairs. The arc set A consists of arcs with labels encoding (input symbol, change to stack pairs). A symbol Z in the second component of an arc label represents a push of Z on stack, a symbol $\bar{Z}$ represents a pop of Z and ε represents no change to the stack.

The start node x_s of the ETS is labelled by $([q,\varepsilon],q_1)$ and the final node x_f by $([q_f,\$],q_1)$. The ETS for the LR(1) grammar of Fig. 1, and the associated LR(1) DFA of Fig. 2 is displayed in Fig. 3. It can be shown [1] that an ETS can be constructed from

$$G = (\{Z, S\}, \{0, 1\}, P, Z)$$

$$\begin{aligned} P: &1.\ Z \to S \\ &2.\ S \to 0S1 \\ &3.\ S \to 01 \end{aligned}$$

Fig. 1. LR(1) grammar

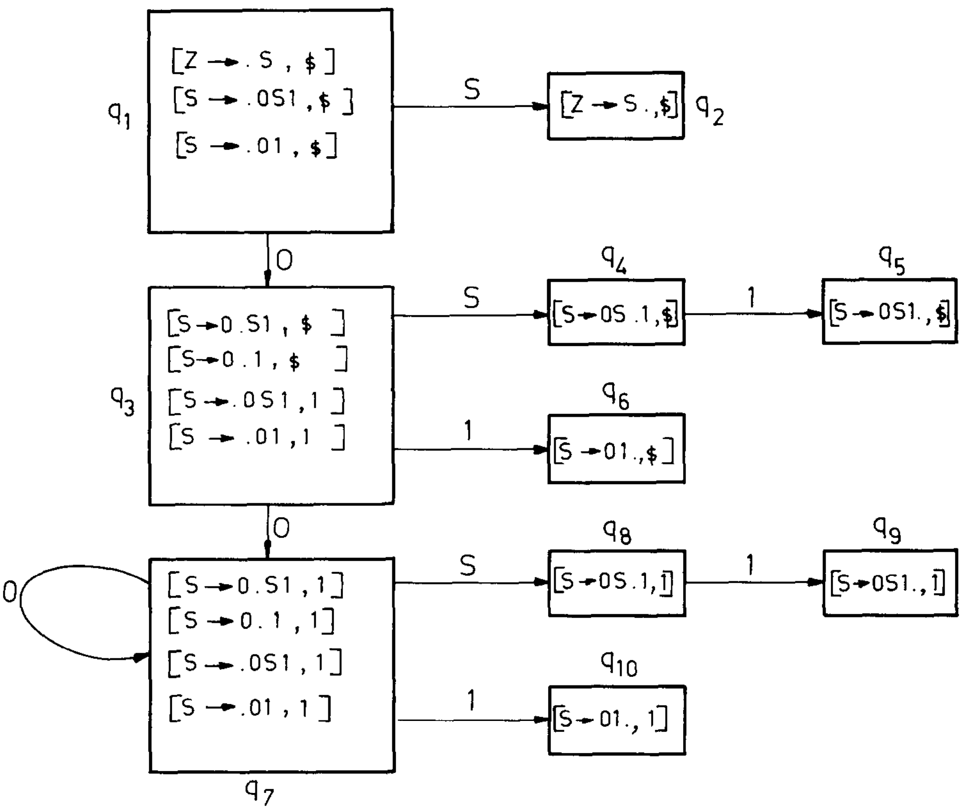

Fig. 2. LR(1)DFA for grammar of Fig. 1.

the LR(1) DPDA in time $O(n_1 + dn_2)$, where n_1 and n_2 are the number of push and pop moves of the DPDA and d is the maximum indegree of a node in the LR(1) DFA.

An examination of Fig. 3 indicates that there are two kinds of paths from x_s to x_f. In the first kind, the top-stack-symbol component of the node is consistent with the implied stack contents of the DPDA at every step. In the second, it is not. In the former case there is always a sequence of moves of the LR(1) DPDA corresponding to the path, which is therefore termed *feasible*. The path 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 in Fig. 1.3 is feasible, whereas 1, 2, 13, 14, 4, 5, 6, 7, 8, 9, 10, 11, 12 is not. We define, for a path p, the *path transmission* $T(p)$ as the concatenation of input symbols on the arcs

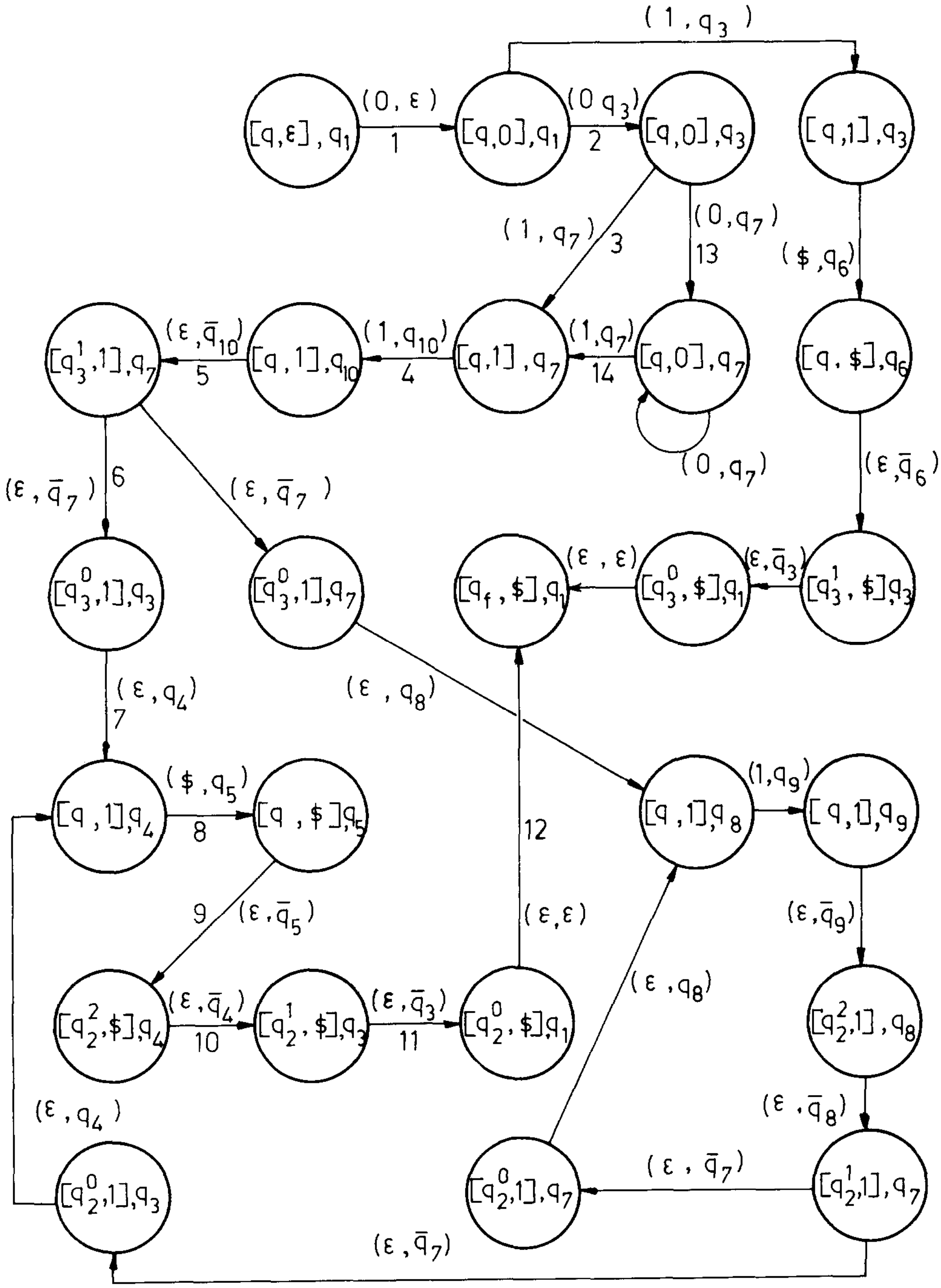

Fig. 3. ETS for the LR(1) DFA of Fig. 2.

along the path, and the *path accumulation* $C(p)$ as the concatenation of change to stack symbols on the arcs along the path. Given a string α representing a path accumulation, we define by $\mu(C(\alpha))$ the shortest string to which α can be reduced using the rule $Z\bar{Z}=\varepsilon$. $\mu(C(\alpha))$ at any point on a feasible path is nothing but the stack contents above q_1 on the parsing stack if p begins at x_s. Clearly, all feasible paths from x_s to x_f represent valid move sequences of the DPDA. Further, it can be shown by a simple inductive proof that for each $w \in L(P)$, where P is the DPDA, there exists a feasible path with transmission w and reduced accumulation ε from x_s to x_f in the ETS.

Let $\langle \alpha_1, \alpha_2 \rangle$ denote a path segment bounded by and including the arcs α_1, α_2. A *matching arc pair* (MAP) of the ETS is a pair of arcs (α_1, α_2), where α_2 corresponds to a DPDA transition that unstacks the symbol pushed by α_1. In other words, α_1, α_2 appear in a feasible path from x_s to x_f and α_2 is the first arc following α_1 on the path such that $\mu(C(\langle \alpha_1, \alpha_2 \rangle))=\varepsilon$. Clearly, any two MAP instances on a path are either disjoint, or one is nested in the other. A MAP is said to be *self-nesting* if an instance of the MAP can nest another instance of itself.

We now define certain cycles that can appear on a feasible path. Let u_1, ξ, u_2 be a feasible path from x_s to x_f, where ξ is a cycle with $\mu(C(\xi))=\varepsilon$. Clearly $T(u_1, \xi^i, u_2)$ is in the language $L(E)$ defined by the ETS for all $i \geqslant 0$. ξ is called an *independent cycle*, as an arbitrary number of traversals of ξ preserves the feasibility of the path. Further, if two instances of an MAP are immediately nested within the same MAP, they define an independent cycle (IC). For example, let $u_1, \alpha, u_2, \alpha_1, u_3, \bar{\alpha}_1, u_4, \alpha_1, u_5, \bar{\alpha}_1, u_6, \bar{\alpha}, u_7$ be the path with both instances of the MAP $(\alpha_1, \bar{\alpha}_1)$ immediately nested within the instance of $(\alpha, \bar{\alpha})$ (i.e. with u_4, u_2 and u_6 having no unmatched arcs). Then, $\xi=\alpha_1, u_3, \bar{\alpha}_1, u_4$ is an independent cycle.

It is obvious that for an ETS derived from an LR(1) DPDA, the existence of an independent cycle implies the existence of a path with repeated arc instances at the same level of nesting within the same MAP.

We next define what are termed matching cycle pairs. Let $(\xi, \bar{\xi})$ be a pair of cycles that occur on a feasible path $u=u_0, \xi, u_1, \bar{\xi}, u_2$ such that

(1) $\mu(C(\xi))\in\Gamma^+, \mu(C(\bar{\xi}))\in\bar{\Gamma}^+ \quad (\bar{\Gamma}=\{\bar{Z}:Z\in\Gamma\})$,

(2) $\mu(C(\xi)C(\bar{\xi}))=\varepsilon$,

(3) $\mu(C(u_1))=\varepsilon$.

Then, $(\xi, \bar{\xi})$ is called a *matching cycle pair* (MCP). We observe that a matching number of traversals of ξ and $\bar{\xi}$ preserves the feasibility of the path, as $\mu(C(u_0, \xi^i, u_1, \bar{\xi}^i, u_2))=\varepsilon$ for all $i \geqslant 0$. Every self-nested instance of an MAP defines an MCP. For, let $u=u_0, \alpha, u_1, \alpha, u_2, \bar{\alpha}, u_3, \bar{\alpha}, u_4$ be the feasible path from x_s to x_f on which there is a self-nested instance of the MAP $(\alpha, \bar{\alpha})$. Then $\xi=u_1, \alpha$ and $\bar{\xi}=\bar{\alpha}, u_3$. Also the existence of a matching cycle pair implies the existence of a path with a self-nesting MAP.

Finally, an independent cycle (matching cycle pair) with no subcycles that are independent cycles or matching cycle pairs is called a minimal independent cycle (MIC) (minimal matching cycle pair (MMCP)).

Let u be any feasible path from x_s to x_f. We define a special subseqence. If we remove from u *all* instances of MICs and MMCPs, the modified path remains feasible and is called a *cycle free feasible path* (CFFP). Note that a cycle-free feasible path may not be cycle-free in the graph-theoretic sense, as it may contain cycles that are matched against straight line segments on a feasible path.

Lemma 2.1. *Let $M(E)$ be the number of MAPs of an ETS E and let u be a feasible path from x_s to x_f with length exceeding $K=M(E)!$. Then, u must contain an MIC or an MMCP.*

Proof. We note that except for the first and the last arc on the path, every other arc is an element of an MAP. Every MAP can immediately nest instances of a subset of the MAPs of E. Thus, if the path length exceeds K, there is either a repeated instance of an MAP, both instances immediately nested within the same MAP, or there is a self-nested instance of an MAP. From the earlier discussion we conclude that the path contains an MMCP or an MIC. □

Corollary. *CFFPs, MICs and MMCPs have bounded lengths.*

This leads to the following lemma.

Lemma 2.2. *Every feasible path from x_s to x_f can be partitioned into subsequences consisting of a CFFP, MICs and MMCPs, the underlying CFFP, and the MICs and MMCPs all belonging to finite sets.*

3. The regularity test

The original paper by Stearns [3] which established the decidability of the regularity of a DCFL contains a necessary and sufficient condition for a DCFL to be a nonregular set. This appears as statement (b) of the following theorem.

Theorem 3.1. *The following statements are equivalent for L, a DCFL over the alphabet T:*

(a) *If E is the ETS for L, then there exists an MMCP $(\xi, \bar{\xi})$ of E, with $T(\bar{\xi})\neq\varepsilon$.*

(b) *Define an equivalence relation $\simeq$ on T^* such that for α_1, α_2 in T^*, $\alpha_1 \simeq \alpha_2$ if α_1, α_2 are either both in L or both not in L. Then there exist strings $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$ in T^* such that*

(i) *for all i, j, $k \geqslant 0$*

$$\alpha_1\alpha_2^i\alpha_3\alpha_4^j\alpha_5 \simeq \alpha_1\alpha_2^{i+k}\alpha_3\alpha_4^{j+k}\alpha_5,$$

(ii) *there exists an l such that for all $i \geqslant l$*

$$\alpha_1\alpha_2^i\alpha_3\alpha_5 \not\equiv \alpha_1\alpha_3\alpha_5.$$

(c) *L is nonregular.*

Before we prove Theorem 3.1, we state, without proof, two lemmas. The proofs follow from the definition of an MMCP and from the fact that E is constructed from a deterministic device.

Lemma 3.2. *Let E be an ETS constructed from a DPDA. Then, if $(\xi, \bar{\xi})$ is an MMCP of E, $T(\xi) \neq \varepsilon$.*

Lemma 3.3. *Let E be an ETS constructed from a DPDA. Then, for an MMCP $(\xi, \bar{\xi})$, if $T(\bar{\xi}) = \varepsilon$, neither MICs nor first elements of MMCPs can originate on any intermediate node of $\bar{\xi}$ in a feasible path from x_s to x_f.*

Proof of Theorem 3.1. (i) We first prove that (a)$\Rightarrow$(b). Assume that there exists an MMCP $(\xi, \bar{\xi})$ with $T(\bar{\xi}) \neq \varepsilon$. Let $u = u_0, \xi, u_1, \bar{\xi}, u_2$ be a feasible path from x_s to x_f. Let $\alpha_1 = T(u_0)$, $\alpha_2 = T(\xi)$, $\alpha_3 = T(u_1)$, $\alpha_4 = T(\bar{\xi})$, $\alpha_5 = T(u_2)$. We observe that condition (i) of (b) holds as for all $i = j \geqslant 0$ and $k \geqslant 0$ $\alpha_1\alpha_2^i\alpha_3\alpha_4^j\alpha_5$ and $\alpha_1\alpha_2^{i+k}\alpha_3\alpha_4^{j+k}\alpha_5$ are both in L, and because of Lemma 3.2, for all $i \neq j \geqslant 0$ and $k \geqslant 0$ they are both not in L. Also, as a consequence of Lemma 3.2, if we choose $l = 0$, then for all $i \geqslant l$, $\alpha_1\alpha_2^i\alpha_3\alpha_5 \notin L$ but $\alpha_1\alpha_3\alpha_5 \in L$; hence, condition (ii) of (b) holds.

(ii) (b)$\Rightarrow$(c) is proved in [3].

(iii) (c)$\Rightarrow$(a).

We will prove the equivalent condition not(a) $\Rightarrow$ not(c), i.e. if every MMCP $(\xi, \bar{\xi})$ of E has $T(\bar{\xi}) = \varepsilon$, then L is regular. We first need a few definitions. Define a minimal initial feasible segment (MIFS) as follows. Let u be a feasible path from x_s to a node x_i such that it has a feasible continuation u_c to x_f. A subsequence $\underline{u}$ of u obtained by removing from u, u_c all MICs and MMCPs subject to the condition that u still passes through x_i, is called an MIFS for u with respect to the continuation u_c. For a feasible path u terminating on x_i, the associated set of MIFSs is finite. This follows from the fact that the length of an MIFS is bounded. We now prove that not(a)$\Rightarrow$not(c).

We show that $L(E)$ is the union of a finite number of equivalence classes of a right-invariant equivalence relation of finite index and, hence, by the Myhill Nerode theorem $L(E)$ is regular. Define a relation R_E on T^* as follows. For $x, y \in T^*$, xR_Ey iff the feasible paths u_x and u_y from node x_s, having transmissions x and y, respectively, lead to the same node x_i and have the same set of MIFSs. Clearly, R_E is an equivalence relation with finite index. We will prove that R_E is right-invariant, i.e. if xR_Ey, then xzR_Eyz for all z in T^*. We will show that given any z in T^* and a feasible continuation of one of the paths with transmission z, we can always find a feasible continuation of the other path on input z terminating on the same node and having the same set of MIFSs.

Assume that u_{xc} is a feasible continuation of u_x, with $T(u_{xc})=z$. If u_{xc} is also a feasible continuation of u_y, then the result follows; so, let us assume that it is not, and let α_x be the first arc of u_{xc} at which the feasible continuations of u_x and u_y on input z diverge, and α_y, the corresponding arc for u_y. (Clearly, both α_x and α_y are pop arcs). The paths from x_s upto and including the arcs that diverge may be written as u_x, u'_{xc}, α_x and u_y, u'_{xc}, α_y. Since u_x, u'_{xc} and u_y, u'_{xc} have the same set of MIFSs, at least one of α_y or α_x must be matched against the first arc of a cycle ξ in u_y or u_x, respectively, and is the first arc of a cycle $\bar{\xi}$, where $(\xi, \bar{\xi})$ is an MMCP. By assumption $T(\bar{\xi})=\varepsilon$ and, hence, from Lemma 3.3 we conclude that no cycles with nonnull trasmissions can originate on $\bar{\xi}$. Consequently, any feasible continuation consisting of a cycle θ beginning with α_x or α_y has null transmission. Assuming without loss of generality that α_y begins such a cycle, but α_x does not, we see that $v_y=u'_{xc}$, θ, α_x and $v_x=u'_{xc}$, α_x are feasible continuations of u_y and u_x, respectively, terminating on the same node and having identical transmissions, with u_x, v_x and u_y, v_y having the same set of MIFS. Thus, $T(u_x, v_x)R_E T(u_y, v_y)$. Repeated use of such an argument until we reach the end of the path u_{xc} gets us the final result. Finally, $L(E)$ is the union of all the equivalence classes associated with x_f.

We next simplify the test for regularity by assuming the LR(1) grammar is in reverse Greibach normal form [2].

Lemma 3.4. *Let E be the ETS for an LR(1) grammar in reverse Greibach normal form. Then, L is regular iff E has no self-nesting MAPS.*

Proof. We first observe that E can have no MMCPs $(\xi, \bar{\xi})$ with $T(\bar{\xi})=\varepsilon$. This follows from the fact that the quantity $\mu(C(\bar{\xi}))$ represents the net change to the parsing stack after traversing $\bar{\xi}$, and $T(\bar{\xi})=\varepsilon$ implies that there exists a rightmost derivation sequence of the form

$$S \underset{\text{rm}}{\overset{*}{\Rightarrow}} \alpha A w \underset{\text{rm}}{\overset{*}{\Rightarrow}} \alpha' A w,$$

which is not possible as G is not right-recursive. Thus, $L(E)$ is nonregular iff there exists an MMCP or equivalently iff there exists a self-nesting MAP. □

Lemma 3.5. *Let E be an ETS derived from an LR(1) grammar. Then E has an MMCP iff it has an elementary cycle ω with $\mu(C(\omega))\in Q^+$.*

Proof. (if) Since every node is reachable by a feasible path, let ρ be such a path from the start node to the initial node of ω. Clearly, $\rho\omega^i$ is a feasible path for all $i\geqslant 0$. Since the number of MAPs is finite, there must be a value of i for which the feasible continuation of $\rho\omega^i$ gives rise to a selfnested MAP instance, implying the existence of an MMCP.

(only if) This follows from the construction of the ETS and the definition of a MMCP. □

Theorem 3.6. *The complexity of the regularity test is bounded by* $(qt)^2 t^{qt}$.

Proof. From Lemma 3.5, we conclude that the test reduces to checking whether E has an elementary cycle whose net effect is to push a nonnull string on the stack. Each such elementary cycle can have length at most the number of nodes of the ETS. The number of such elementary cycles originating at each node is at most t^{qt} as t is a bound on the out degree of any node, and qt is the number of nodes. Since we have to check for elementary cycles originating at each node, and each check takes an amount of computation proportional to the length of the cycle, the result follows. □

References

[1] B.S. Adiga, Applications of finite graph models of context-free languages, Ph.D. Thesis, Indian Institute of Science, 1989.
[2] M.A. Harrison, *Introduction to Formal Language Theory* (Addison-Wesley, Reading, MA, 1978).
[3] R.E. Stearns, A regularity test for pushdown machines, *Inform. and Control* **11**(3) (1967) 323–340.
[4] L.G. Valiant, Regularity and related problems for deterministic pushdown automata, *J. ACM* **22**(1) 1975.