JOURNAL OF COMPUTER AND SYSTEM SCIENCES 35, 59–71 (1987)

Complementing Deterministic Büchi Automata in Polynomial Time

R. P. KURSHAN

AT & T Bell Laboratories, Murray Hill, New Jersey 07974

Received December 16, 1985; revised July 22, 1986

For any Büchi automaton Γ with n states which accepts the (ω-regular) language $\mathscr{L}(\Gamma)$, an explicit construction is given for a Büchi automaton $\tilde{\Gamma}$ with $2n$ states which, when Γ is deterministic, accepts exactly the complementary language $\mathscr{L}(\Gamma)'$. It follows that the nonemptiness of complement problem for deterministic Büchi automata (*i.e.*, whether $\mathscr{L}(\Gamma)' = \varnothing$) is solvable in polynomial time. The best previously known construction for complementing a deterministic Büchi automaton with n states has $O(2^{4n^2})$ states; for nondeterministic Γ, determining whether $\mathscr{L}(\Gamma)' = \varnothing$, is known to be PSPACE-complete. Interest in deterministic Büchi automata arises from the suitability of deterministic automata in general to describe properties of physical systems; such properties have been found to be more naturally expressible by deterministic automata than by nondeterministic automata. However, if Γ is nondeterministic, then $\tilde{\Gamma}$ provides a "poor man's" approximate inverse to Γ in the following sense: $\mathscr{L}(\Gamma)' \subset \mathscr{L}(\tilde{\Gamma})$, and as nondeterministic branches of Γ are removed, the two languages become closer. Hence, for example, given two nondeterministic Büchi automata Λ and Γ, one may test for containment of their associated languages through use of the corollary that $\mathscr{L}(\Lambda * \tilde{\Gamma}) = \varnothing \Rightarrow \mathscr{L}(\Lambda) \subset \mathscr{L}(\Gamma)$ (where $\Lambda * \tilde{\Gamma}$ is one of the standard constructions satisfying $\mathscr{L}(\Lambda * \tilde{\Gamma}) = \mathscr{L}(\Lambda) \cap \mathscr{L}(\tilde{\Gamma})$). The "error term" $\mathscr{L} = \mathscr{L}(\tilde{\Gamma}) \backslash \mathscr{L}(\Gamma)'$ may be determined exactly, and whether $\mathscr{L} = \varnothing$ may be determined in time $O(e^2)$, where e is the number of edges of Γ.

1. INTRODUCTION

An ω-regular language $\mathscr{L}$ over a finite alphabet Σ is a subset $\mathscr{L} \subset \Sigma^\omega$ (Σ^ω is the set of (infinite) sequences in Σ) with the property that each element of $\mathscr{L}$ is "accepted" by some fixed finite state automaton, of which there are a number of varieties. There is an established theory surrounding this concept; a reasonable sample is provided by Choueka [CH74] and the references therein. There are many similarities and a few profound differences between the theory of ω-regular languages and the more classical theory of ordinary regular languages [RS59]; this is clearly brought out in [Ch74]. One such difference is that while all the different varieties of classical finite state automata (which accept finite strings) are easily transformable into one another, the same is not true in the ω-regular theory. Equivalence in the ω-regular theory, when it exists, is often very difficult to prove, and in one notable case, it is absent (deterministic Büchi automata do not generate

0022-0000/87 $3.00

all the ω-regular languages). Another difference is that while any regular language defined by a nondeterministic automaton with n states also may be defined by a deterministic automaton with $m = 2^n$ states via the "subset construction" [RS59, Definition 11], the analogous construction for ω-regular languages does not work (see Example (3.12)(2)); the best known constructions [Ch74] have $m > 2^{2^n}$.

Nonetheless, the ω-regular theory and its generalizations have some applications in areas unattainable to the finite theory, such a certain fields of logic [Bu62, Ra69, Ra69, Ra72], topology and game theory [Ra69], and modelling non-terminating physical systems such as computer communication protocols and their paradigms [CE82, MP81, AKS83, MW84, Ku85].

Apparently the first variety of finite state automaton used to define ω-regular languages was defined by Büchi [Bu62]. Büchi automata in general, and the problem of their complementation in particular, have been studied in various contexts [Bu62, McN66, Si70, BS73, Ch74, SVW86]. One attraction of Büchi automata over other varieties of automata used to define ω-regular languages is that Büchi automata are particularly easy to define and to relate to physical systems; in particular, temporal logic, as it is used to study physical systems, is most conveniently related to Büchi automata [SVW86].

Deterministic Büchi automata suffer from an already mentioned weakness (e.g., $(a+b)^*a^\omega$: the set of sequences in $\{a, b\}$ which are eventually constantly a, is not definable by any deterministic Büchi automaton, as is easily proved). Nonetheless, in the context of physical systems, deterministic Büchi automata have been found to provide a more natural medium than the more general nondeterministic Büchi automata, for defining system requirements. The reason is that physical system requirements, when defined in a practical framework, have branch points which virtually always are related to observable conditions [CE82, MP81, AKS83, MW84, Ku85], and hence these conditional branches are deterministic. (For example, a system requirement might be stated as "send a message; if it gets lost, resend it; otherwise, send the next message" [deterministic branch]. The system requirement would probably never be stated as "send a message; then [nondeterministically] either resend it or send the next message".) Consequently, it is more suitable to provide other deterministic structures [Ku86a] to compensate for the weakness of deterministic Büchi automata, rather than to force descriptions in less natural nondeterministic terms.

The problem of complementation of automata arises in the context of proving containment of languages. If $\mathscr{L}(\Gamma)$ and $\mathscr{L}(\Lambda)$ are the ω-regular languages defined by the automata Γ and Λ respectively, the usual technique used to determine whether

$$\mathscr{L}(\Lambda) \subset \mathscr{L}(\Gamma) \tag{1.1}$$

is to determine whether the equivalent condition

$$\mathscr{L}(\Lambda * \tilde{\Gamma}) = \varnothing \tag{1.2}$$

holds, where $\tilde{\Gamma}$ is an automaton which defines the complementary language, $\mathscr{L}(\tilde{\Gamma}) = \mathscr{L}(\Gamma)'$, and $*$ is a product operator with the property $\mathscr{L}(A * B) = \mathscr{L}(A) \cap \mathscr{L}(B)$. The problem of determining whether (1.1) holds arises, for example, when Λ defines a physical system and Γ defines what is acceptable behavior in Λ [CE82, MW84, SVW86, Ku86a, b].

The best previously known construction for complementing a deterministic Büchi automaton is a construction [SVW86] which works generally for nondeterministic Büchi automata. Given a (nondeterministic) Büchi automaton Γ with n states, their construction provides a (nondeterministic) Büchi automaton $\tilde{\Gamma}$ with $O(2^{4n^2})$ states such that

$$\mathscr{L}(\tilde{\Gamma}) = \mathscr{L}(\Gamma)'. \tag{1.3}$$

Using this construction they prove that determining whether $\mathscr{L}(\Gamma)' = \varnothing$ is PSPACE-complete.

In this paper, a construction is given for a (nondeterministic) Büchi automaton $\tilde{\Gamma}$ with only $2n$ states, which has the property that when Γ is a deterministic Büchi automaton, (1.3) holds. Furthermore, when Γ is an arbitrary (nondeterministic) Büchi automaton, $\tilde{\Gamma}$ satisfies

$$\mathscr{L}(\Gamma)' \subset \mathscr{L}(\tilde{\Gamma}), \tag{1.4}$$

in which case (1.2) $\Rightarrow$ (1.1). A Büchi automaton Δ_Γ with $O(n^2)$ states is constructed which accepts the "error" in (1.4). That is, $\mathscr{L}(\Delta_\Gamma) = \mathscr{L}(\tilde{\Gamma}) \backslash \mathscr{L}(\Gamma)'$. If (1.2) fails and

$$\mathscr{L}(\Lambda * \Delta_\Gamma) = \varnothing \tag{1.5}$$

holds, then clearly (1.1) fails. Thus, when (1.2) holds or (1.2) fails and (1.5) holds, for $\tilde{\Gamma}$ as constructed here, a proof of (1.1) for arbitrary (nondeterministic) Λ and Γ, or a proof of its failure, is obtained which is "exponentially cheaper" than what would be afforded by the [SVW86] construction, since their construction may require an exponential number of states whether or not Λ or Γ is deterministic. Furthermore, if one removes one or more nondeterministic branches of Γ, $\mathscr{L}(\tilde{\Gamma})$ and $\mathscr{L}(\Delta_\Gamma)$ become smaller in the lattice of subsets of Σ^ω (and (1.3) holds when the last such branch is removed). One direct consequence of this construction is that the emptiness of the complement problem for deterministic Büchi automata is solvable in polynomial time. Although this in itself is not a deep result, it turns out to be very practical [Ku85; Ku86a, b].

A slight variation in the definition of Büchi automata is given here. A conventional Büchi automaton Γ, called here a *state-recurring* Büchi automaton, has associated with it a set $R(\Gamma)$ of states of Γ, the *recurring* states of Γ, which are used to define acceptance in Γ: a sequence in Σ^ω is *accepted* by Γ iff it follows a path in Γ which hits $R(\Gamma)$ infinitely often. The variation defined here is called an *edge-recurring* Büchi automaton. It is the same as the other but that $R(\Gamma)$ is a set of edges rather than states. The respective sets of languages defined by the two varieties of automata are the same, in both the nondeterministic and deterministic cases, respectively. However, the new variety is expressively more efficient: it requires fewer states, fewer by as much as half, to define the same language. It is also more convenient to use, in that it provides a finer control in the definition of acceptance.

The construction for $\tilde{\Gamma}$ given here is given in terms of an edge-recurring Γ, and has $2n$ states and $2e+r$ edges, where Γ has n states and e edges, and $r = \text{card } R(\Gamma)$. Starting from an edge-recurring automaton Γ with n states, the construction for $\tilde{\Gamma}$ given in [SVW86] would first construct an equivalent state-recurring automaton with $2n$ states. The resulting size of $\tilde{\Gamma}$ then would be $O(2^{16n^2})$ states.

2. Preliminaries

Conventionally, an automaton is viewed as a set of states and a successor relation which takes a "current" state and "current" input and returns a "next" state (for deterministic automata) or a set of "next" states, in general. I prefer to view an automaton as a directed graph whose vertices are the automaton states, and each edge of which is labelled with the set of inputs which enables that state transition. The labelled graph is defined in terms of its adjacency matrix.

Let Σ, V be nonempty sets, let $L = 2^{\Sigma}$ (the set of subsets of Σ), and let M be a map

$$M\colon V^2 \to L$$

(where $V^2 = V \times V$ is the cartesian product). Then M is said to be an *L-matrix* with *state space* $V(M) = V$. The elements of $V(M)$ are said to be *states* or *vertices* of M. The element $\varnothing \in L$ is denoted by 0 and $\Sigma \in L$ is denoted by 1. (Thus, always $\{0, 1\} \subset L$, and if card $\Sigma = 1$ then $L = \{0, 1\}$.)

An *edge* of an L-matrix M is an element $e \in V(M)^2$ for which $M(e) \neq 0$. ($M(e)$ is the "label" on the edge e.) The set of edges of M is denoted by $E(M)$. A *cycle* in M of *length* n is an n-tuple $\mathbf{v} = (v_1, ..., v_n) \in V(M)^n$ such that $M(v_i, v_{i+1}) \neq 0$ for all $0 \leqslant i < n$ and $M(v_n, v_1) \neq 0$. The cycle $\mathbf{v}$ *contains* the edge $(v, w) \in E(M)$ if for some i, $1 \leqslant i < n$, $v_i = v$, $v_{i+1} = w$, or if $v_n = v$ and $v_1 = w$.

A *graph* is a $\{0, 1\}$-matrix. The *graph* of the L-matrix M is the graph $\bar{M}$ with state space $V(\bar{M}) = V(M)$, defined by

$$\bar{M}(e) = \begin{cases} 1 & \text{if } M(e) \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Clearly, $\mathbf{v} \in V(M)^n$ is a cycle in M iff $\mathbf{v}$ is a cycle in $\bar{M}$.

An L-matrix M is *lockup-free* if for all $v \in V(M)$, $\bigcup_{w \in V(M)} M(v, w) = 1$. An L-matrix M is *deterministic* if for all $u, v, w \in V(M)$, if $v \neq w$ then $M(u, v) \cap M(u, w) = 0$.

Let Σ be a nonempty set. A *state-recurring Büchi automaton* (*over the language* Σ) is a triple $\Gamma=(M_\Gamma, I(\Gamma), R(\Gamma))$, where M_Γ, the *transition matrix* of Γ, is a lockup-free 2^Σ-matrix, $\varnothing \neq I(\Gamma)\subset V(\Gamma)$ (the *initial* states of Γ) and $R(\Gamma)\subset V(\Gamma)$ (the *recurring* states of Γ). An *edge-recurring Büchi automaton* is defined in the same way, except that $R(\Gamma)\subset E(\Gamma)$ (the *recurring* edges of Γ). Set $V(\Gamma)=V(M_\Gamma)$, $E(\Gamma)=E(M_\Gamma)$, and $\Gamma(v, w)=M_\Gamma(v, w)$ for v, $w\in V(\Gamma)$.

A *chain* of a Büchi automaton Γ is a sequence $(v_0, v_1,...)\in V(\Gamma)^\omega$ such that $v_0\in I(\Gamma)$ and $(v_i, v_{i+1})\in E(\Gamma)$ for all $i\geqslant 0$. The set of all chains of Γ is denoted by $\mathscr{C}(\Gamma)$.

A chain $\mathbf{v}\in\mathscr{C}(\Gamma)$ is an *acceptance* chain of Γ if $\mathbf{v}$ "hits" $R(\Gamma)$ infinitely often (i.e., for infinitely many values of $i, v_i\in R(\Gamma)$ in the state-recurring case and $(v_i, v_{i+1})\in R(\Gamma)$ in the edge-recurring case). The set of all acceptance chains of Γ is denoted by $\mathscr{C}_{\text{acc}}(\Gamma)$.

Let Γ be a Büchi automaton over Σ. The sequence $(t_0, t_1,...)\in\Sigma^\omega$ (sometimes called an "input tape" to Γ) *follows* a chain $\mathbf{v}\in\mathscr{C}(\Gamma)$ provided $t_i\in\Gamma(v_i, v_{i+1})$ for all $i\geqslant 0$. The sequence $\mathbf{t}\in\Sigma^\omega$ is *accepted* by Γ if it follows an acceptance chain of Γ. Set $\mathscr{L}(\Gamma)=\{\mathbf{t}\in\Sigma^\omega \mid \mathbf{t}$ is accepted by $\Gamma\}$, the *language* defined by Γ. The *complementary* language $\mathscr{L}(\Gamma)'=\Sigma^\omega\backslash\mathscr{L}(\Gamma)$.

A Büchi automaton Γ is *deterministic* if M_Γ is. Following convention, an automaton is called *nondeterministic* in order to indicate that it may or may not be deterministic (so every automaton is nondeterministic).

It is now shown that state-recurring Büchi automata define the same set of languages as edge-recurring, while the latter are more economical in size of state space.

Let Γ be an edge-recurring Büchi automaton over Σ. For each $v\in V(\Gamma)$ with the property that there exist $u, w\in V(\Gamma)$ such that $(u, v)\in R(\Gamma)$, $(w, v)\in E(\Gamma)$ but $(w, v)\notin R(\Gamma)$, define a new symbol $\hat{v}\notin V(\Gamma)$, and let $\hat{V}(\Gamma)$ be the set of symbols thus defined. Set $V=V(\Gamma)\cup\hat{V}(\Gamma)$ and define the map

$$\phi\colon V\to V(\Gamma)$$

by $\phi(v)=v$ for $v\in V(\Gamma)$, while $\phi(\hat{v})=v$ for $\hat{v}\in\hat{V}(\Gamma)$. Extend ϕ to $\phi\colon V^2\to V(\Gamma)^2$ by defining $\phi(v, w)=(\phi(v), \phi(w))$. Define the 2^Σ-matrix M with state space $V(M)=V$, as follows: let $\hat{V}=\hat{V}(\Gamma)\cup\{v\in V(\Gamma)\mid \hat{v}$ is undefined$\}$ and set

$$M(v, w)=\begin{cases}\Gamma\phi(v, w) \text{ if } (\phi(v, w)\in R(\Gamma) \text{ and } w\in\hat{V}) \text{ or } (\phi(v, w)\notin R(\Gamma) \text{ and } w\notin\hat{V}(\Gamma)),\\ 0 \qquad \text{otherwise.}\end{cases}$$

Define $R=E(M)\cap\phi^{-1}R(\Gamma)$ and define the edge-recurring Büchi automaton $\hat{\Gamma}$ over Σ by

$$\hat{\Gamma}=(M, I(\Gamma), R).$$

(2.1) LEMMA. $\phi E(\hat{\Gamma})=E(\Gamma)$.

Proof. If $e \in E(\hat{\Gamma})$ then $\hat{\Gamma}(e) \neq 0$ so $0 \neq \hat{\Gamma}(e) = \Gamma\phi(e)$ and $\phi(e) \in E(\Gamma)$. If $(v, w) \in R(\Gamma)$ and $\hat{w}$ is defined, then $\hat{\Gamma}(v, \hat{w}) = \Gamma(v, w) \neq 0$ so $(v, \hat{w}) \in E(\hat{\Gamma})$ and $\phi(v, \hat{w}) = (v, w)$; if $(v, w) \in E(\Gamma)\backslash R(\Gamma)$ or $(v, w) \in R(\Gamma)$ and $\hat{w}$ is not defined, then $\hat{\Gamma}(v, w) = \Gamma(v, w) \neq 0$ so $(v, w) \in E(\hat{\Gamma})$ and $\phi(v, w) = (v, w)$.

(2.2) THEOREM. *Let Γ be an edge-recurring Büchi automaton. Then*

(a) $\hat{\hat{\Gamma}} = \hat{\Gamma}$;

(b) *$\hat{\Gamma}$ is deterministic iff Γ is*;

(c) $\mathscr{L}(\hat{\Gamma}) = \mathscr{L}(\Gamma)$.

Proof. (a) If $\hat{V}(\Gamma) = \varnothing$ then ϕ is the identity and $M = M_\Gamma$. Hence, it suffices to show that $\hat{V}(\hat{\Gamma}) = \varnothing$. If $(u, v) \in R(\hat{\Gamma})$ then $\phi(u, v) \in R(\Gamma)$ so by the definition of $M_{\hat{\Gamma}}$, either $v \in \hat{V}(\Gamma) \subset \hat{V}$ or $v \in V(\Gamma)$ and $\hat{v}$ is undefined. In either case, for any $w \in V(\hat{\Gamma})$ with $(w, v) \in E(\hat{\Gamma})$, by the definition of $M_{\hat{\Gamma}}$, if $v \in \hat{V}(\Gamma)$ then $\phi(w, v) \in R(\Gamma)$ whence $(w, v) \in R(\hat{\Gamma})$. Hence, $\hat{v}$ is undefined and it follows that $\hat{V}(\hat{\Gamma}) = \varnothing$.

(b) For $u, v, w \in V(\hat{\Gamma})$ with $v \neq w$, if $\phi(v) = \phi(w)$ then either $\hat{\Gamma}(u, v) = 0$ (if $v \in \hat{V}(\Gamma)$ and $\phi(u, v) \notin R(\Gamma)$ or $v \notin \hat{V}(\Gamma)$ and $\phi(u, v) \in R(\Gamma)$) or $\hat{\Gamma}(u, w) = 0$ (if $\hat{\Gamma}(u, v) \neq 0$). Since $\hat{\Gamma}(u, v) \cap \hat{\Gamma}(u, w) \neq \varnothing \Rightarrow (\Gamma\phi(u, v)) \cap (\Gamma\phi(u, w)) \neq \varnothing$, if Γ is deterministic then $\hat{\Gamma}$ is as well. On the other hand, if $\Gamma(a, b) \cap \Gamma(a, c) \neq 0$ for some $a, b, c \in V(\Gamma)$ with $b \neq c$, then by Lemma (2.1) there exist $u, u', v, w \in V(\hat{\Gamma})$ such that $\phi(u) = \phi(u') = a$, $\phi(v) = b$, $\phi(w) = c$, $v \neq w$ and (u, v), $(u', w) \in E(\hat{\Gamma})$. Then by the definition of $M_{\hat{\Gamma}}$, $(u, w) \in E(\hat{\Gamma})$ and $\hat{\Gamma}(u, v) \cap \hat{\Gamma}(u, w) = \Gamma(a, b) \cap \Gamma(a, c) \neq 0$. Hence, if $\hat{\Gamma}$ is deterministic, so is Γ.

(c) By Lemma (2.1) and the definition of $\hat{\Gamma}$, if $\mathbf{v} \in \mathscr{C}_{\text{acc}}(\hat{\Gamma})$ then $\phi(\mathbf{v}) \in \mathscr{C}_{\text{acc}}(\Gamma)$ (where $\phi(v_0, v_1, \ldots) \equiv (\phi(v_0), \phi(v_1), \ldots)$) and $\mathbf{t} \in \Sigma^\omega$ follows $\mathbf{v}$ iff $\mathbf{t}$ follows $\phi(\mathbf{v})$. Thus, $\mathscr{L}(\hat{\Gamma}) \subset \mathscr{L}(\Gamma)$. Now, suppose $\mathbf{v} = (v_0, v_1, \ldots) \in \mathscr{C}_{\text{acc}}(\Gamma)$. We inductively construct a chain $\mathbf{w} \in \mathscr{C}_{\text{acc}}(\hat{\Gamma})$ such that $\phi(\mathbf{w}) = \mathbf{v}$. Define $w_0 = v_0$ and suppose $w_0, \ldots, w_n \in V(\hat{\Gamma})$ have been defined, with $\phi(w_i) = v_i$ for $1 \leqslant i \leqslant n$. If $(v_n, v_{n+1}) \in R(\Gamma)$ and $\hat{v}_{n+1}$ is defined, define $w_{n+1} = \hat{v}_{n+1}$; otherwise define $w_{n+1} = v_{n+1}$. It follows that $\mathbf{w} \in \mathscr{C}_{\text{acc}}(\hat{\Gamma})$ and by construction, $\mathbf{t} \in \Sigma^\omega$ follows $\mathbf{w}$ iff $\mathbf{t}$ follows $\mathbf{v}$. Hence $\mathscr{L}(\Gamma) \subset \mathscr{L}(\hat{\Gamma})$.

(2.3) COROLLARY. *For any [deterministic] state-recurring automaton Γ, there exists a [deterministic] edge-recurring automaton Γ' with $V(\Gamma') = V(\Gamma)$, $E(\Gamma') = E(\Gamma)$, and $\mathscr{L}(\Gamma') = \mathscr{L}(\Gamma)$. For any [deterministic] edge-recurring automaton Γ, there exists a [deterministic] state-recurring automaton Γ' such that* card $V(\Gamma') \leqslant 2$ card $V(\Gamma)$ *and $\mathscr{L}(\Gamma') = \mathscr{L}(\Gamma)$. Thus, for a given language $\mathscr{L}$, if Λ and Γ are the minimum-state [deterministic] edge-recurring and state-recurring automata, respectively, with $\mathscr{L}(\Lambda) = \mathscr{L}(\Gamma) = \mathscr{L}$, then* card $V(\Lambda) \leqslant$ card $V(\Gamma) \leqslant 2$ card $V(\Lambda)$.

Proof. If Γ is a state-recurring automaton then defining Γ' to be Γ with $R(T)$ replaced by $R(\Gamma') = \{(v, w) \in E(\Gamma) \mid w \in R(\Gamma)\}$, gives an edge-recurring automaton Γ' which accepts the same language. Conversely, given an edge-recurring automaton Γ, define Γ' to be the state-recurring automaton $\hat{\Gamma}$ with $R(\hat{\Gamma})$ replaced by $R(\Gamma') = \{w \in V(\hat{\Gamma}) \mid (v, w) \in R(\hat{\Gamma}) \text{ for some } v \in V(\hat{\Gamma})\}$. The inequalities follow from the construction of $\hat{\Gamma}$. The fact that $\mathscr{L}(\Gamma') = \mathscr{L}(\Gamma)$ follows from Theorem (2.2c).

(2.4) *Remark.* Experience suggests that edge-recurring automata provide a more natural medium for expressing certain types of properties than state-recurring automata [Ku86a, b]. It is not hard to find examples of languages for which the minimum-state edge-recurring automaton which defines the language has fewer states than the corresponding state-recurring automaton. For example, it is easy to show that $(a+b)(a^+b^+)^\omega$ requires a 3-state state-recurring automaton, but is defined by the 2-state edge-recurring automaton Γ in the following example.

(2.5) EXAMPLE. Let Γ be the deterministic edge-recurring automaton over the alphabet $\Sigma = \{\alpha, \beta\}$ where, denoting the singleton sets $a = \{\alpha\}$, $b = \{\beta\}$, Γ is defined by $V(\Gamma) = \{1, 2\}$, $I(\Gamma) = \{1\}$, $R(\Gamma) = \{(1, 2)\}$ and

$$M_\Gamma = \begin{pmatrix} a & b \\ a & b \end{pmatrix}.$$

Then $V(\hat{\Gamma}) = \{1, 2, \hat{2}\}$, $I(\hat{\Gamma}) = \{1\}$, $R(\hat{\Gamma}) = \{(1, \hat{2})\}$ and

$$M_{\hat{\Gamma}} = \begin{pmatrix} a & 0 & b \\ a & b & 0 \\ a & b & 0 \end{pmatrix},$$

where the rows and columns are in the order $1, 2, \hat{2}$. It is easily seen that $\mathscr{L}(\Gamma)$ $(= \mathscr{L}(\hat{\Gamma})) = (a+b)(a^+b^+)^\omega$.

3. CONSTRUCTION OF $\tilde{\Gamma}$

Let Γ be an edge-recurring Büchi automaton with n states, over an alphabet Σ. A (nondeterministic) Büchi automaton $\tilde{\Gamma}$ over Σ will be constructed, with $2n$ states, having the property

$$\mathscr{L}(\tilde{\Gamma}) \supset \mathscr{L}(\Gamma)',$$

for which equality holds when Γ is deterministic.

Define the 2^Σ-matrix M as follows: $V(M)$ is a "copy" of $V(\Gamma)$, say $\phi\colon V(M) \to V(\Gamma)$ is a bijection (with $V(M) \cap V(\Gamma) = \varnothing$), and for $\phi\colon V(M)^2 \to V(\Gamma)^2$ by $\phi(v, w) = (\phi(v), \phi(w))$, let

$$M(e) = \begin{cases} M_\Gamma \phi(e) & \text{if } \phi(e) \notin R(\Gamma) \\ 0 & \text{otherwise.} \end{cases}$$

Find (in time linear in card $E(M)$) a maximal spanning forest F of the graph $\bar{M}$ [AHU74]. Relative to F, let R be the set of back-edges of $\bar{M}$.

(3.1) LEMMA. *Every cycle in M contains an edge in R.*

Proof. Define the matrix M' by $V(M') = V(M)$, $M'(e) = M(e)$ if $e \notin R$, $M'(e) = 0$ otherwise. It is enough to show that there are no cycles in M'. Indeed, by the definition of R, F is also a maximal spanning forest of $\bar{M}'$. Thus, as is well known [AHU74], if M' contains a cycle, then it contains a cycle with a back-edge of $\bar{M}'$, relative to F, of which there are none.

The matrix M will be "spliced" to M_Γ, forming a matrix N, as follows. Set

$$V' = V(\Gamma) \cup V(M),$$

let \$ be a symbol distinct from the elements of V' and set $V = V' \cup \{\$\}$. Define the 2^Σ-matrix N with $V(N) = V$ and

$$N(v, w) = \begin{cases} M(v, w) & \text{if } (v, w) \in E(M); \\ \Gamma(v, w) & \text{if } (v, w) \in E(\Gamma); \\ \Gamma(v, \phi(w)) & \text{if } v \in V(\Gamma), \text{ and } w \in V(M) \text{ and } (v, \phi(w)) \in R(\Gamma); \\ \Sigma \mid \bigcup\limits_{u \in V(M)} M(v, u) & \text{if } v \in (M) \text{ and } w = \$; \\ \Sigma & \text{if } v = w = \$; \\ 0 & \text{otherwise.} \end{cases}$$

(3.2) LEMMA. *N is lockup-free.*

Proof. Let $v \in V(N)$. If $v \in V(\Gamma)$ then $\bigcup_{w \in V(N)} N(v, w) \supset \bigcup_{w \in V(\Gamma)} N(v, w) = 1$. Given $v \in V(M)$ it follows that $\bigcup_{w \in V(N)} N(v, w) \supset (\bigcup_{w \in V(M)} M(v, w)) \cup N(v, \$) = 1$. If $v = \$$ then $\bigcup_{w \in V(N)} N(v, w) \supset N(\$, \$) = 1$.

Define $I = I(\Gamma) \cup \phi^{-1} I(\Gamma)$ and let $\tilde{\Gamma}$ be the edge-recurring Büchi automaton over Σ defined by

$$\tilde{\Gamma} = (N, I, R).$$

Note. $\tilde{\Gamma}$ is not a unique function of Γ, as R is dependent upon the choice of F. At the expense of increasing the size of R we could define $R = E(M)$, with the result that $\tilde{\Gamma}$ would then be uniquely determined by Γ.

(3.3) THEOREM. *$\mathscr{L}(\tilde{\Gamma}) \supset \mathscr{L}(\Gamma)'$, while $\mathscr{L}(\tilde{\Gamma}) = \mathscr{L}(\Gamma)'$ when Γ is deterministic.*

Proof. Let $\mathbf{t} \in \Sigma^\omega$. If $\mathbf{t} \in \mathscr{L}(\Gamma)$ then $\mathbf{t}$ follows a chain of Γ which hits some element of $R(\Gamma)$ infinitely often. If Γ is deterministic, then $M_{\tilde{\Gamma}}$ is deterministic except for transitions from M_Γ to M; since $R(\Gamma)$ is missing from $E(M)$, any chain of

$\tilde{\Gamma}$ which **t** follows must stay in M_Γ. By construction, no such chain can reach $R(\tilde{\Gamma})$, so $\mathbf{t} \notin \mathscr{L}(\tilde{\Gamma})$. Now suppose $\mathbf{t} \in \mathscr{L}(\Gamma)'$. Then any chain of Γ which **t** follows must eventually never again hit any element of $R(\Gamma)$. Let $\mathbf{v} \in \mathscr{C}(\Gamma)$ be such a chain (there always exists at least 1), and let

$$n = \sup\{i > 0 \mid (v_{i-1}, v_i) \in R(\Gamma)\}.$$

Then $-\infty \leqslant n < \infty$. If $n = -\infty$, let $\mathbf{v}' \in \mathscr{C}(\tilde{\Gamma})$ be the copy of **v** in M. (Such a $\mathbf{v}'$ exists because in this case **v** avoids $R(\Gamma)$ altogether.) Since $V(\tilde{\Gamma})$ is finite, **v** must follow a cycle in M infinitely often, and by Lemma (3.1), it follows that $\mathbf{t} \in \mathscr{C}(\tilde{\Gamma})$. If $-\infty < n$ then define $\mathbf{v}' \in \mathscr{C}(\tilde{\Gamma})$ to be that chain which stays in M_Γ until v_{n-1}, at which point it crosses over into M (and continues there). The previous argument then applies and in this case too, $\mathbf{t} \in \mathscr{L}(\tilde{\Gamma})$.

(3.4) Corollary. *Let Γ be a Büchi automaton. There is an algorithm whose running time is polynomial in the number of states of Γ which can sometimes prove that $\mathscr{L}(\Gamma)' = \varnothing$. In particular, when Γ is deterministic, the algorithm always determines whether or not $\mathscr{L}(\Gamma)' = \varnothing$.*

Proof. Convert Γ to an edge-recurring automaton, using (2.2). Use depth-first search to determine whether there is a strongly connected component of the graph $\bar{M}_{\tilde{\Gamma}}$ which contains both endpoints of an edge in $R(\tilde{\Gamma})$ and is reachable from $I(\tilde{\Gamma})$. Clearly $\mathscr{L}(\tilde{\Gamma}) = \varnothing$ iff no such strongly connected component exists. Using Tarjan's algorithm [AHU74], the strongly connected components of G may be found in time linear in the number m of edges of $M_{\tilde{\Gamma}}$. By construction, $m = O(n^2)$, where $n = \text{card } V(\Gamma)$. By Theorem (3.3), $\mathscr{L}(\tilde{\Gamma}) = \varnothing \Rightarrow \mathscr{L}(\Gamma)' = \varnothing$ and when Γ is deterministic, $\mathscr{L}(\tilde{\Gamma}) = \varnothing \Leftrightarrow \mathscr{L}(\Gamma)' = \varnothing$.

Let us now determine the "error term" $\mathscr{L}(\tilde{\Gamma}) \backslash \mathscr{L}(\Gamma)'$. For any Büchi automaton Γ, say that two chains, $\mathbf{v}, \mathbf{w} \in \mathscr{C}(\Gamma)$ are *related*, denoted $\mathbf{v} \approx \mathbf{w}$, if for all $i \geqslant 0$,

$$\Gamma(v_i, v_{i+1}) \cap \Gamma(w_i, w_{i+1}) \neq 0. \tag{3.5}$$

(3.6) Proposition. *Let Γ be any edge-recurring Büchi automaton. Then $\mathscr{L}(\tilde{\Gamma}) = \mathscr{L}(\Gamma)'$ iff $\mathbf{v} \in \mathscr{C}(\Gamma)$, $\mathbf{w} \in \mathscr{C}_{\text{acc}}(\Gamma)$, $\mathbf{v} \approx \mathbf{w} \Rightarrow \mathbf{v} \in \mathscr{C}_{\text{acc}}(\Gamma)$.*

Proof. Extend ϕ: $V(M) \to V(\Gamma)$ to ϕ: $\mathscr{C}(\tilde{\Gamma}) \to \mathscr{C}(\Gamma)$ by defining $\phi(v_0, v_1, \ldots) = (\phi(v_0), \phi(v_1), \ldots)$, where $\phi(v) = v$ for $v \in V(\Gamma)$. The key to the proof is the observation that by construction of $\tilde{\Gamma}$, $\phi\mathscr{C}_{\text{acc}}(\hat{\Gamma}) = \mathscr{C}(\Gamma) \backslash \mathscr{C}_{\text{acc}}(\Gamma)$. Thus, $\mathscr{L}(\tilde{\Gamma}) \neq \mathscr{L}(\Gamma)' \Leftrightarrow \mathscr{L}(\Gamma)' \nsubseteq \mathscr{L}(\tilde{\Gamma}) \Leftrightarrow$ there exist $\mathbf{t} \in \Sigma^\omega$ and chains $\mathbf{v} \in \mathscr{C}_{\text{acc}}(\Gamma)$, $\mathbf{w} \in \mathscr{C}(\Gamma) \backslash \mathscr{C}_{\text{acc}}(\Gamma)$ such that **t** follows both **v** and $\mathbf{w} \Leftrightarrow$ by Definition (3.5), there exist chains $\mathbf{v} \in \mathscr{C}_{\text{acc}}(\Gamma)$, $\mathbf{w} \in \mathscr{C}(\Gamma) \backslash \mathscr{C}_{\text{acc}}(\Gamma)$ such that $\mathbf{v} \approx \mathbf{w}$.

The criterion given by (3.6), while sometimes directly useful (e.g., Example (3.12)(2)), is nonconstructive. A polynomial time algorithm is next derived from (3.6) to determine whether or not the criterion holds. In the course of doing this, an edge-recurring automaton Δ_Γ is constructed with $\text{card } V(\Delta_\Gamma) = O(n^2)$, for

$n = \text{card}\, V(\Gamma)$, which satisfies $\mathscr{L}(\Delta_\Gamma) = \mathscr{L}(\tilde{\Gamma}) \backslash \mathscr{L}(\Gamma)'$. Whether or not $\mathscr{L}(\Delta_\Gamma) = \varnothing$ may be determined in time $O(n^4)$.

Define the 2^Σ-matrix M as follows: $V(M) = V(\Gamma) \cup \{\$\}$ ($\$ \notin V(\Gamma)$),

$$M(v, w) = \begin{cases} \Gamma(v, w) & \text{if } (v, w) \in E(\Gamma) \backslash R(\Gamma), \\ \bigcup\limits_{(v,u) \in R(\Gamma)} \Gamma(v, u) & \text{if } v \in V(\Gamma),\ w = \$, \\ \Sigma & \text{if } v = w = \$, \\ 0 & \text{otherwise.} \end{cases}$$

Define the "tensor product" $M_\Gamma \otimes M$ to be the 2^Σ-matrix with $V(M_\Gamma \otimes M) = V(M_\Gamma) \times V(M)$ defined by

$$(M_\Gamma \otimes M)((v, w), (v', w')) = M_\Gamma(v, v') \cap M(w, w').$$

(3.7) LEMMA. *$M_\Gamma \otimes M$ is lockup-free.*

Proof. By definition M_Γ is lockup-free, while by construction M is lockup-free. More generally, the tensor product of lockup-free matrices is lockup-free. Specifically, for any $v \in V(M_\Gamma)$, $w \in V(M)$, we have

$$\bigcup_{v'} M_\Gamma(v, v') = 1$$

and

$$\bigcup_{w'} M(w, w') = 1$$

so

$$\begin{aligned} 1 &= \Big(\bigcup M_\Gamma(v, v')\Big) \cap \Big(\bigcup M(w, w')\Big) = \bigcup_{v', w'} (M_\Gamma(v, v') \cap M(w, w')) \\ &= \bigcup_{v', w'} (M_\Gamma \otimes M)((v, w), (v', w')). \end{aligned}$$

(3.8) LEMMA. *If M_Γ is deterministic, so is $M_\Gamma \otimes M$.*

Proof. If M_Γ is deterministic, then so is M by construction. More generally, the tensor product of deterministic matrices is deterministic, as is easily seen from the definitions.

Define $I = I(\Gamma)^2 \subset V(M_\Gamma \otimes M)$ and set

$$R = \{((v, w), (v', w')) \in E(M_\Gamma \otimes M) \mid (v', v') \in R(\Gamma),\ w' \neq \$\}.$$

Define the edge-recurring automaton $\Delta_\Gamma = (M_\Gamma \otimes M, I, R)$. Clearly, $\text{card}\, V(\Delta_\Gamma) = n(n+1)$ ($n = \text{card}\, V(\Gamma)$), and this may be reduced by discarding states not reachable from $I(\Delta_\Gamma)$.

(3.9) THEOREM. *Let Γ be an edge-recurring Büchi automaton. Then $\mathscr{L}(\tilde{\Gamma})\backslash\mathscr{L}(\Gamma)' = \mathscr{L}(\Delta_\Gamma)$.*

Proof. Let $\mathbf{t}\in\mathscr{L}(\Delta_\Gamma)$. Then there exists a chain $\mathbf{u}\in\mathscr{C}_{\text{acc}}(\Delta_\Gamma)$ such that $\mathbf{t}$ follows $\mathbf{u}$. Say $\mathbf{u}=((v_0, w_0), (v_1, w_1),...)$, where for $i\geqslant 0$, $(v_i, w_i)\in V(\Gamma)\times V(M)=V(\Delta_\Gamma)$. Then for $i\geqslant 0$, $u_i\in\Gamma(v_i, v_{i+1})\cap M(w_i, w_{i+1})$ so surely $\mathbf{t}$ follows $\mathbf{w}=(w_0, w_1,...)\in\mathscr{C}(\Gamma)$. But, by construction of M, $\mathbf{w}\notin\mathscr{C}_{\text{acc}}(\Gamma)$ and thus by construction of $\tilde{\Gamma}$, $\mathbf{t}\in\mathscr{L}(\tilde{\Gamma})$. On other hand, $\mathbf{t}$ also follows $\mathbf{v}=(v_0, v_1,...)\in\mathscr{C}_{\text{acc}}(\Gamma)$, and thus $\mathbf{t}\notin\mathscr{L}(\Gamma)'$. Thus $\mathscr{L}(\Delta_\Gamma)\subset\mathscr{L}(\tilde{\Gamma})\backslash\mathscr{L}(\Gamma)'$. Now suppose $\mathbf{t}\in\mathscr{L}(\tilde{\Gamma})\backslash\mathscr{L}(\Gamma)'$. By (3.6) there exist chains $\mathbf{v}\in\mathscr{C}_{\text{acc}}(\Gamma)$, $\mathbf{w}\in\mathscr{C}(\Gamma)\backslash\mathscr{C}_{\text{acc}}(\Gamma)$ such that $t_i\in\Gamma(v_i, v_{i+1})\cap\Gamma(w_i, w_{i+1})$ for $i\geqslant 0$. It follows that $\mathbf{u}=((v_0, w_0), (v_1, w_1),...)\in\mathscr{C}_{\text{acc}}(\Delta_\Gamma)$ and $\mathbf{t}$ follows $\mathbf{u}$. Thus, $\mathbf{t}\in\mathscr{L}(\Delta_\Gamma)$, and the reverse containment is obtained, completing the proof.

(3.10) COROLLARY. *There is an algorithm which, given an edge-recurring Büchi automaton Γ with n edges, determines in time $O(n^2)$ whether or not $\mathscr{L}(\Gamma)'=\mathscr{L}(\tilde{\Gamma})$.*

Proof. Find the strongly connected components of Δ_Γ which are reachable from $I(\Delta_\Gamma)$, using Tarjan's algorithm [AHU74]. By (3.9), $\mathscr{L}(\Gamma)'=\mathscr{L}(\tilde{\Gamma})$ iff $\mathscr{L}(\Delta_\Gamma)=\varnothing$ iff no such strongly connected component contains both vertices of an edge of $R(\Gamma_\Gamma)$. Since vard $E(\Delta_\Gamma)\leqslant n^2$, and Tarjan's algorithm is linear in $E(\Delta_\Gamma)$, the time bound follows.

Note that if Γ is deterministic then (3.8) shows that Δ_Γ is deterministic. But then $\mathscr{C}_{\text{acc}}(\Delta_\Gamma)=\varnothing$ and (3.9) gives another proof that when Γ is deterministic, $\mathscr{L}(\tilde{\Gamma})=\mathscr{L}(\Gamma)'$.

Let Γ be an edge-recurring Büchi automaton, and let Γ_{D} be the deterministic Büchi automaton derived from Γ through the "subset construction" [RS59, Definition 11], in which $R(\Gamma_{\text{D}})=\{(A, B)\in E(\Gamma_{\text{D}}) \mid \text{for some } (v, w)\in R(\Gamma),\ v\in A,\ w\in B\}$. It is easily shown that $\mathscr{L}(\Gamma)\subset\mathscr{L}(\Gamma_{\text{D}})$ (and the inclusion may be proper, as illustrated by (3.12)(2) below). Since Γ_{D} is deterministic, $\mathscr{L}(\tilde{\Gamma}_{\text{D}})=\mathscr{L}(\Gamma_{\text{D}})'$. Thus

$$\mathscr{L}(\tilde{\Gamma}_{\text{D}})\subset\mathscr{L}(\Gamma)'\subset\mathscr{L}(\tilde{\Gamma}) \tag{3.11}$$

and when Γ is deterministic, they are all equal.

(3.12) EXAMPLES. (1) Let Γ be the deterministic automaton defined in Example (2.5)(1). By construction, $\tilde{\Gamma}$ is unique. Let $V(\tilde{\Gamma})=\{1, 2, \phi^{-1}(1), \phi^{-1}(2), \$\}$, relative to which order on the rows and columns of $M_{\tilde{\Gamma}}$,

$$M_{\tilde{\Gamma}}=\left(\begin{array}{cc|ccc} a & b & 0 & b & 0 \\ a & b & 0 & 0 & 0 \\ \hline & & a & 0 & b \\ & 0 & a & b & 0 \\ & & 0 & 0 & 1 \end{array}\right)$$

while $I(\tilde{\Gamma})=\{1, \phi^{-1}(1)\}$ and $R(\tilde{\Gamma})=\{(\phi^{-1}(1), \phi^{-1}(1)), (\phi^{-1}(2), \phi^{-1}(2)\}$. It is easily checked that $\mathscr{L}(\tilde{\Gamma})=(a+b)^{+}(a^{\omega}+b^{\omega})$, and one may verify that $\mathscr{L}(\tilde{\Gamma})=\mathscr{L}(\Gamma)'$.

(2) Let Γ be the edge-recurring automaton defined as follows: $V(\Gamma)=\{1, 2, 3\}$, $I(\Gamma)=\{1\}$, $R(\Gamma)=\{(2, 3)\}$ and

$$M_{\Gamma}=\begin{pmatrix} a & b & b \\ 0 & a & b \\ 0 & b & a \end{pmatrix}.$$

Then the sequence which constantly β, $\beta^{\omega} \in \mathscr{L}(\Gamma)$ while the sequence which is constantly α, prefixed by β, $\beta\alpha^{\omega} \notin \mathscr{L}(\Gamma)$. The "subset construction" [RS59] applied to Γ gives the deterministic 2^{Σ}-matrix M with $V(M)=\{\{1\}, \{2, 3\}\}$, defined by

$$M=\begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix}$$

(where the first row and column correspond to $\{1\}$ and the second to $\{2, 3\}$). Any "chain" in M which is followed by β^{ω} is also followed by $\beta\alpha^{\omega}$ and thus there is no chain-dependent definition of acceptance with respect to which M is the transition matrix of an automaton over Σ which defines $\mathscr{L}(\Gamma)$. Using the criterion (3.6), it is easily seen that $\mathscr{L}(\Gamma)=(a^{*}b)^{\omega}$. By (3.6), $\mathscr{L}(\tilde{\Gamma})=\mathscr{L}(\Gamma)'$ and thus $\mathscr{L}(\Delta_{\Gamma})=\varnothing$.

(3) Add (2, 2) to $R(\Gamma)$ in Example (2.5). (Of course, Γ remains deterministic.) Then $\mathscr{L}(\Gamma)=(a+b)^{*}(a^{*}b)^{\omega}$, $\mathscr{L}(\Gamma)'=(a+b)^{*}a^{\omega}$ and by (3.3), $\mathscr{L}(\Gamma)'=\mathscr{L}(\tilde{\Gamma})$. There is no deterministic Büchi automaton which accepts exactly $(a+b)^{*}a^{\omega}$; thus, there can be no general construction for $\tilde{\Gamma}$ which is deterministic, even when Γ is assumed to be deterministic.

Acknowledgment

I thank Pierre Wolper for several useful conversations, and in particuar, for pointing out an oversight which unnecessarily complicated my original construction.

References

[RS59] M. O. Rabin and D. Scott, "Finite Automata and their Decisions Problems," *IBM J. Res. Develop.* **3** (1959), 114–125, reprinted in "Sequential Machines," Addison–Wesley, Reading, MA, 1964.

[Bu62] J. R. Büchi, On a diecision method in restricted second-order arithmatic, *in* "Proc. Internat. Cong. on Logic, Methodol. and Philos. of Sci., 1960, pp. 1–11, Stanford Univ. Press, Stanford, CA, 1962.

[Mo64] E. F. Moore (Ed.), "Sequential Machines," Addition–Wesley, Reading, MA, 1964.

[McN66] R. McNaughton, Testing and generating infinite sequences by a finite automaton, *Inform. and Control* **9** (1966), 521–530.

[Ra69] M. O. RABIN, Decidability of second-order theories and automata on infinite trees, *Trans. Amer. Math. Soc.* **141** (1969), 1–35.

[Si70] D. SIEFKES, "Büchi's Monadic Second-Order Successor Arithmetics," Lecture Notes in Math., Vol. **120**, Springer-Verlag, Berlin, 1970.

[Ra72] M. O. RABIN, "Automata on Infinite Objects and Church's Problem," Amer. Math. Soc., Providence, RI, 1972.

[BS73] J. R. BÜCHI AND D. SIEFKES, "The Monadic Second Order Theory of ω," Lecture Notes in Math., Vol. **328**, Springer-Verlag, Berlin, 1973.

[Ch74] Y. CHOUEKA, Theories of automata on ω-tapes: A simplified approach, *J. Comput. System Sci.* **8** (1974), 117–141.

[AHU74] A. V. AHO, J. E. HOPCROFT, AND J. D. ULLMAN, "The Design and Analysis of Computer Algorithms," Addition–Wesley, Reading, MA, 1974; "Data Structures and Algorithms," Addison–Wesley, Reading, MA, 1983.

[MP81] Z. MANNA AND A. PNUELI, "Verification of Concurrent Programs: The Temporal Framework," Stanford University, Techn. Report No. CS-81-826.

[CE82] E. M. CLARKE AND E. A. EMERSON, "Synthesis of Synchronization Skeletons from Branching Time Temporal Logic," Proc. Logic of Programs Workshop, 1981, Lecture Notes in Comput. Sci., Vol. **131**, pp. 52–71, Springer-Verlag, Berlin, 1982.

[AKS83] S. AGGARWAL, R. P. KURSHAN, AND K. K. SABNANI, A calculus for protocol specification and validation, *in* "Protocol Specification, Testing and Verification, III," pp. 19–34, North-Holland, Amsterdam, 1983.

[MW84] Z. MANNA AND P. WOLPER, Synthesis of communicating processes from temporal logic specifications, *ACM Trans. Program. Languages Systems* **6** (1984), 68–93.

[Ku85] R. P. KURSHAN, "Modelling Concurrent Processes," *Proc. Sympos. Appl. Math.* **3** (1985), 45–57.

[Ku86a] R. P. KURSHAN, "Testing Containment of ω-Regular Languages," unpublished report.

[Ku86b] R. P. KURSHAN, "Analysis of Coordination," unpublished report.

[SVW86] A. P. SISTLA, M. Y. VARDI, P. WOLPER, The complementation problem for Büchi automata, with applications to temporal logic, *in* "Proc. 12th Internat. Coll. on Automata, Languages and Programming, 1985," Lecture Notes in Comput. Sci., Springer-Verlag, Berlin, in press.