Discrete Event Dynamic Systems: Theory and Applications, 5, 357–381 (1995)

# On Control of Systems Modelled as Deterministic Rabin Automata

J. G. THISTLE
*Département de génie électrique et de génie informatique, École Polytechnique de Montréal, C.P. 6079, succ. Centre-ville, Montreal (Quebec), Canada H3C 3A7*

*Received October 22, 1993; Revised November 28, 1994*

**Abstract.** Recent results on the control of infinite behaviour of finite automata are extended to allow Rabin acceptance conditions as modelling assumptions as well as specifications. The key result is a fixpoint characterization of the automaton's *controllability subset*—the set of states from which it can be controlled to the satisfaction of its associated specification. The fixpoint characterization allows for straightforward computation of the subset and for effective synthesis of controllers. The results have potential applications to supervisory control synthesis, the synthesis of reactive systems, and decision procedures for modal logics.

**Keywords:** $\omega$-automata, supervisory control, Church's problem, control synthesis

**Editor:** J. van Schuppen

## 1. Introduction

This article extends the main result of Thistle and Wonham (1994a), which has potential applications in supervisory control, in the synthesis of "reactive" systems, and in decision procedures for propositional modal logics.

Some of the basic results of supervisory control theory[1] have recently been extended to the setting of infinite-string formal languages (*$\omega$-languages*) and the associated finite automata (*$\omega$-automata*).[2] Such extension not only admits natural modelling of nonterminating processes (by allowing explicit representation of infinite event streams), but also increases the scope of the theory and establishes connections between control synthesis for discrete event systems and the verification and synthesis of computer systems. Indeed, $\omega$-languages and $\omega$-automata are standard tools for the analysis and design of computer systems (Manna and Pnueli 1992, Vardi 1991, Kurshan 1988, Thomas 1990) and have already found application in control (Varaiya 1993).

Discrete event systems (DES) are modelled in supervisory control theory as controlled generators of formal languages. Desired closed-loop behaviour is typically specified by requiring the language generated[3] under control to lie within some prespecified range, in the sense of set inclusion. In other words, the language generated by the controlled DES is required to be included in some "maximal legal language" and in turn to contain some "minimal acceptable language" (Ramadge and Wonham 1987). In the original finite-string case, such language inclusions allow the specification of so-called "safety" properties, which, roughly speaking, assert that some given (undesirable) condition must never obtain (Lamport 1977). On the other hand, "liveness" properties, which state roughly that some (desirable) condition must eventually obtain (Lamport 1977), cannot be expressed through finite-string language inclusions. Indeed, safety properties have been formally defined in Alpern and Schneider (1985) as representing restrictions on finite event streams, while (pure) liveness properties have been defined as placing no conditions on finite event streams but rather restricting the set of infinite event sequences. The infinite-string extension thus allows the expression of certain properties of asymptotic behaviour that the finite-string theory does not.

This increased expressiveness necessitates a strengthening of the controllability property of the original theory (Ramadge and Wonham 1987). An appropriate property—*$\omega$-controllability*—is defined in Thistle and Wonham (1994b).[4] As in the finite-string case, a key step in infinite-string supervisor synthesis is the computation of the supremal controllable sublanguage of the legal language. If the discrete event system (DES) to be controlled and the maximal legal language are both represented by finite automata on infinite strings ($\omega$-automata), then the supremal $\omega$-controllable sublanguage can be found by computing the *controllability subset* of a controlled automaton.

The controllability subset can be defined as the set of states from which the $\omega$-automaton can be controlled to generate only strings satisfying its acceptance condition. This acceptance condition is based on the set of states visited infinitely often in the course of the generation of an infinite string.[5] It is possible to restrict attention without loss of generality to a special form of acceptance condition: all formal languages accepted by $\omega$-automata—the so-called *$\omega$-regular* languages—are accepted by finite, deterministic automata equipped with *Rabin* acceptance conditions (see section 2 for a definition).

The computation of the controllability subset of a deterministic Rabin automaton is discussed in Thistle and Wonham (1994a). This problem is formally equivalent to the solution of Church's problem (Church 1963, Büchi and Landweber 1969), a well-known automaton synthesis problem that has recently been proposed as a paradigm for the synthesis of "reactive" systems (Pnueli and Rosner 1989a, Pnueli and Rosner 1989b). Another equivalent problem is that of deciding the emptiness of the set of infinite trees accepted by a Rabin tree automaton (Rabin 1972, Hossley and Rackoff 1972, Emerson and Jutla 1988). This last problem is central to deciding satisfiability of propositional modal logics and monadic second-order theories (Emerson 1990, Rabin 1969). Like those of Pnueli and Rosner (1989a) and Emerson and Jutla (1988), the solution of Thistle and Wonham (1994a) matches the best known upper bounds on computational complexity, and is in fact essentially optimal in this respect.

As shown in Thistle and Wonham (1994b), the computation of controllability subsets allows effective supervisor synthesis in the presence of liveness specifications. The present article extends the result of Thistle and Wonham (1994a) to allow the use of liveness properties not only in specification but also in modelling. Liveness properties—and in particular so-called "fairness properties" (Francez 1986)—play an important role in the modelling of concurrency. For example, in the absence of precise information on the relative "speeds" of a collection of asynchronous subsystems, it is often desirable to assume that events occur infinitely often in each subsystem; indeed, Ramadge (1989) introduced deterministic Büchi automata to capture such a fairness condition in the supervisory control of "product systems."

Systems modelled by deterministic Büchi automata have also been considered in Kumar et al. (1992) and Young et al. (1992). Such automata recognize only a proper subset of the $\omega$-regular languages, and do not allow the expression of certain useful fairness properties.[6] The present article allows the use of the more powerful deterministic Rabin automata.

In particular, this article deals with the computation of the controllability subset of a finite, deterministic automaton equipped with two Rabin conditions, one representing a specification (as in Thistle and Wonham 1994a) and the other representing a modelling assumption that might, for example, capture fairness properties. The controllability subset is here defined as the set of all states from which the automaton can be controlled so that any infinite event sequence that is consistent with the control action and satisfies the modelling assumption also satisfies the specification. To rule out trivial solutions, we require the "deadlock-freedom" condition that any finite event string generated by the controlled system extend to an infinite string that is also generated by the controlled system. This generalization of the problem of Thistle and Wonham (1994a) is similar to an extension of Church's problem that was proposed in Abadi et al. (1989) (for the synthesis of reactive systems under fairness assumptions) but not solved by direct construction. Another similar problem is considered in Wong-Toi and Dill (1991), but there the requirement of deadlock-freedom is dropped, effectively reducing the problem to that of Church.

The solution presented here extends that of Thistle and Wonham (1994a);[7] it features a fixpoint characterization of the controllability subset based on the fixpoint calculus approach of Emerson and Jutla (1988), and methods of induction on the structure of automata inspired by those of Rabin (1972). The problem is formally defined in the following section. The monotone operators employed in the fixpoint representation are introduced in section 3. Structural operations on automata that facilitate the induction are defined in section 4. The fixpoint representation of the controllability subset is established in section 5 and in section 6 the computational complexity of the method is analyzed. The definitions and results are illustrated by a simple example in section 7. Related work is discussed in section 8. The proof of the main result is provided in the appendix; proofs of intermediate results are available in the addendum (Thistle 1994a).

A preliminary version of the main result was outlined in Thistle (1992).

## 2. Control of Automata

We first introduce some standard notation for formal languages. If $\Sigma$ is a finite alphabet then $\Sigma^*$ represents the set of finite strings over $\Sigma$, plus the empty string, denoted by 1; $\Sigma^\omega$ represents the set of (countably) infinite words over $\Sigma$ and $\Sigma^\infty$ denotes $\Sigma^* \cup \Sigma^\omega$. A *language* is a subset of $\Sigma^\infty$—in particular, an *$\omega$-language* is a subset of $\Sigma^\omega$. A finite string $k \in \Sigma^*$ is a *prefix* of $v \in \Sigma^\infty$ if it is an initial substring of $v$; we write $k \leq v$, or $k < v$ if $k$ is a proper initial substring. let pre$(L)$ denote the set of all prefixes of strings belonging to a language $L \subseteq \Sigma^\infty$.

We consider automata of the form

$$\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, \{(R_q, I_q) : q \in Q\}, \mathbb{C})$$

where:

- $\Sigma$ is a finite alphabet of *event symbols*;
- $X$ is a finite *state set*;
- $\delta : \Sigma \times X \longrightarrow 2^X$ is a *transition function*;
- $x_0 \in X$ is an *initial state*;
- $\{(R_p, I_p) : p \in P\}, \{(R_q, I_q) : q \in Q\}$ are families of pairs of subsets of $X$, each family determining a *Rabin recognition condition*; and
- $\mathbb{C} \subseteq 2^\Sigma$ is a family of *control patterns*.

A *path* on $\mathcal{A}$ of a string $v \in \Sigma^\infty$ is a total map $\pi : \text{pre}(\{v\}) \longrightarrow X$ such that

$$\pi(1) = x_0 \;\&\; \forall k \in \text{pre}(\{v\}), \sigma \in \Sigma \;:\; k\sigma \in \text{pre}(\{v\}) \Longrightarrow \pi(k\sigma) \in \delta(\sigma, \pi(k))$$

Thus a path associates a state trajectory with a string in a manner consistent with the transition function.

Note that this state trajectory begins with the initial state; in order to discuss trajectories beginning at another state $x \in X$, let $\mathcal{A}_x$ denote the automaton obtained by replacing the initial state $x_0$ with $x$.

The *recurrence set* of a path $\pi$ on a string $s \in \Sigma^\omega$ is $\Omega_\pi := \{x \in X : |\pi^{-1}(x)| = \omega\}$; in other words, the recurrence set is the set of states that recur infinitely often along the corresponding state trajectory.

For either family of state subset pairs $\{(R_r, I_r) : r \in R\}$, we say that a path $\pi$ is *recognized* according to that family if there exists $r \in R$ s.t. $\Omega_\pi \cap R_r \neq \emptyset$ and $\Omega_\pi \subseteq I_r$. Thus a path is recognized if for some $r \in R$, the subset $R_r$ is visited infinitely often along the corresponding state trajectory, and the subset $I_r$ almost always. Restriction to this form of recognition condition entails no loss of generality in the sense that all $\omega$-languages that can be represented by finite automata (the so-called $\omega$-regular languages) can be represented by deterministic, finite *Rabin* automata employing conditions of this form (Thomas 1990).

This report is concerned exclusively with *deterministic* automata (for which $|\delta(\sigma, x)| \leq 1, \forall \sigma \in \Sigma, x \in X$); we shall therefore consider $\delta$ as a partial function $\delta : \Sigma \times X \longrightarrow X$, writing $\delta(\sigma, x)!$ to signify that the map $\delta$ is defined for the pair $(\sigma, x)$.[8] Furthermore, we shall extend all such transition functions to partial functions $\delta : \Sigma^* \times X \longrightarrow X$ in the usual manner:

$$(1, x) \overset{\delta}{\mapsto} x$$

$$(k\sigma, x) \overset{\delta}{\mapsto} \delta(\sigma, \delta(k, x)), \text{ for all } k \in \Sigma^*, \sigma \in \Sigma \text{ s.t. } \delta(k, x) \text{ and } \delta(\sigma, \delta(k, x)) \text{ are defined.}$$

Note that a string has at most one path on a deterministic automaton.

The condition corresponding to the first family of state subset pairs, $\{(R_p, I_p) : p \in P\}$, will be viewed (as in Thistle and Wonham 1994a) as a specification; hence any string $s \in \Sigma^\omega$ having a path that is recognized according to this family will be said to be *accepted* by $\mathcal{A}$.

The second family of state subsets (which is absent from earlier studies) will be used to represent modelling assumptions relating to the asymptotic behaviour of the uncontrolled automaton: any string $s \in \Sigma^\omega$ having a path that is *not* recognized according to this family will be said to be *generated* by $\mathcal{A}$.[9] Any finite string $k \in \Sigma^*$ that has a path on $\mathcal{A}$ is also said to be generated by $\mathcal{A}$.

The last component of the automaton represents a control mechanism. Feedback is modelled by partial functions $f : \Sigma^* \longrightarrow \mathbb{C}$, interpreted as mapping the sequence of past events to a corresponding control action. Formally, we say that a string $v \in \Sigma^\infty$ is generated by $\mathcal{A}$ *under* $f : \Sigma^* \longrightarrow \mathbb{C}$ if there exists a path $\pi : \mathrm{pre}(\{v\}) \longrightarrow X$ such that for all prefixes $k\sigma$ of $v$, $\sigma \in f(k)$; if $v \in \Sigma^\omega$ we also require that $\pi$ not be recognized according to the second family of state subsets. In order for this definition to make physical sense we shall restrict attention to maps satisfying the following condition: $f : \Sigma^* \longrightarrow \mathbb{C}$ is said to be *complete* if for every $k \in \Sigma^*$ for which $f(k)$ is defined, and every $\sigma \in f(k)$, $f(k\sigma)$ is also defined.

The main result of the article provides a fixpoint representation of the set of states from which a deterministic automaton $\mathcal{A}$ can be controlled in deadlock-free fashion so that all infinite strings generated by the controlled automaton are also accepted by the automaton:

*Definition 2.1.* Let $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, \{(R_q, I_q) : q \in Q\}, \mathbb{C})$. Its *controllability subset* $F^{\mathcal{A}} \subseteq X$ is the set of all states $x \in X$ for which there exists a complete map $f : \Sigma^* \longrightarrow \mathbb{C}$ such that

i. every $s \in \Sigma^\omega$ generated by $\mathcal{A}_x$ under $f$ is accepted by $\mathcal{A}_x$; and

ii. for any $k \in \Sigma^*$ generated by $\mathcal{A}_x$ under $f$, there exists $t \in \Sigma^\omega$ such that $kt$ is accepted by $\mathcal{A}_x$ under $f$.

## 3. The Inverse Dynamics and Reachability Operators

We shall characterize $F^{\mathcal{A}} \subseteq X$ as a certain fixpoint of the following monotone operator:

*Definition 3.1.* Let $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, \{(R_q, I_q) : q \in Q\}, \mathbb{C})$ be a deterministic automaton. Its *inverse dynamics operator* is given by

$$\begin{aligned}\theta^{\mathcal{A}} &: 2^X \longrightarrow 2^X \\ &X' \mapsto \{x \in X : (\exists \Gamma \in \mathbb{C})[(\forall \sigma \in \Gamma)\delta(\sigma, x) \in X' \; \& \; (\exists \sigma \in \Gamma)\delta(\sigma, x)!]\}\end{aligned}$$

For any $X' \subseteq X$, $\theta^{\mathcal{A}}(X')$ is the set of all states in which the automaton can be controlled so that its next state belongs to $X'$.

The subset $F^{\mathcal{A}}$ is indeed one of the fixpoints of $\theta^{\mathcal{A}}$:

PROPOSITION 3.2 *Let $\mathcal{A}$ be a deterministic automaton. Then*

$$F^{\mathcal{A}} = \theta^{\mathcal{A}}(F^{\mathcal{A}})$$

We shall characterize $F^{\mathcal{A}}$ uniquely with the aid of the fixpoint calculus employed in Thistle and Wonham (1992, 1994a), whereby for any expression $\phi(X_1)$ containing the variable $X_1$, $\mu X_1.\ \phi(X_1)$ (resp. $\nu X_1.\ \phi(X_1)$) denotes the least (resp. greatest) $X_1 \subseteq X$ (in the sense of set inclusion) such that $X_1 = \phi(X_1)$. The existence of such fixpoints will follow from monotonicity properties of the expressions $\phi(\cdot)$ that we shall employ. (See the addendum (Thistle 1994a) for some preliminary results from fixpoint theory, and refer to Thistle and Wonham (1994a) for control interpretations of some simple fixpoint calculus formulas.)

For the time being we use the fixpoint notation to extend $\theta^{\mathcal{A}}$ to the following operator:

*Definition 3.3.* Let $\mathcal{A}$ be a deterministic automaton. The *reachability operator* of $\mathcal{A}$ is given by

$$\begin{aligned} \rho^{\mathcal{A}} :\ & 2^X \longrightarrow 2^X \\ & X_1 \mapsto \mu X_2.\ [X_1 \cup \theta^{\mathcal{A}}(X_2)] \end{aligned}$$

Thus $\rho^{\mathcal{A}}$ maps any subset $X_1$ to its reachability subset—the set of all states from which $\mathcal{A}$ can be controlled to reach $X_1 \subseteq X$ in zero or more transitions.

## 4. Automaton Structure

To facilitate structural induction we bring in operations that potentially reduce either the number of state subset pairs associated with an automaton or the number of "live" states as defined by Rabin (1972).

Let $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, \{(R_q, I_q) : q \in Q\})$. The set of *live* states of $\mathcal{A}$ is given by

$$L(\mathcal{A}) := \{x \in X : (\exists \sigma \in \Sigma)\ \delta(\sigma, x) \neq x\}$$

In other words, a state is live if other states can be reached from it. An approximate opposite to liveness is "degeneracy." A state $x \in X$ is *degenerate* if there are transitions leaving $x$ but all of them simply lead back to $x$; more precisely, $x \in X$ is degenerate if

$$\exists \sigma \in \Sigma : \ \delta(\sigma, x)!\ \&\ \forall \sigma \in \Sigma : \delta(\sigma, x) = x$$

The set of degenerate states of $\mathcal{A}$ is denoted by $D(\mathcal{A})$. The subsets $L(\mathcal{A})$ and $D(\mathcal{A})$ are of course disjoint but $L(\mathcal{A}) \cup D(\mathcal{A})$ may be a proper subset of $X$; indeed, $L(\mathcal{A}) \cup D(\mathcal{A}) = \{x \in X : (\exists \sigma \in \Sigma)\delta(\sigma, x)!\}$.

For any $x \in X$, $X' \subseteq X$ and $p \in P$, we generalize the operations employed in Thistle and Wonham (1992, 1994a) to automata $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, \{(R_q, I_q) : q \in Q\}, \mathbb{C})$; the resulting operations potentially reduce the complexity of $\mathcal{A}$ as measured by $|L(\mathcal{A})|$ and $|P \dot{\cup} Q|$:

**self-looping** of a subset: $\mathcal{A}(\hookrightarrow X') := (\Sigma, X, \delta', x_0, \{(R'_p, I'_p) : p \in P\}, \{(R_q, I_q) : q \in Q\}, \mathbb{C})$, where

$$\delta'(\sigma, x') = \begin{cases} x' & \text{if } x' \in X' \\ \delta(\sigma, x') & \text{otherwise} \end{cases}$$
$$\& \; R'_p = R_p \cup X' \; \& \; I'_p = I_p \cup X', \; \forall p \in P$$

**restriction** to a subset: $\mathcal{A}\upharpoonright X' := (\Sigma, X, \delta', x_0, \{(R'_p, I'_p) : p \in P\}, \{(R_q, I_q) : q \in Q\}, \mathbb{C})$, where

$$\delta'(\sigma, x') = \begin{cases} \delta(\sigma, x') & \text{if } x' \in X' \cup \{x_0\} \\ x' & \text{otherwise} \end{cases}$$
$$\& \; R'_p = R_p \cap X' \; \& \; I'_p = I_p \cap X', \; \forall p \in P$$

**exclusion** of a pair: For $r \in P\dot{\cup}Q$, $\mathcal{A}\rfloor r$ is obtained by restricting $\mathcal{A}$ to the subset $I_r \cup D(\mathcal{A})$ and deleting the pair $(R_r, I_r)$.

Self-looping of a subset $X' \subseteq X$ turns every $x \in X'$ into a degenerate state and ensures that the singleton $\{x\}$ satisfies the acceptance criterion. On the other hand, restriction to a subset $X' \subseteq X$ turns all other states into degenerate states that do *not* satisfy the acceptance condition. Finally, exclusion of a pair indexed by $p \in P$ restricts the automaton to the subset $I_p \cup D(\mathcal{A})$ and, provided $|P| > 1$, eliminates the pair $(R_p, I_p)$. All three of these operations potentially reduce the number of live states while the third potentially reduces the number of pairs in the acceptance condition.

Without loss of generality, we shall henceforth assume that for all $\sigma \in \Sigma$, $x, x' \in X$, $\delta(\sigma, x)! \; \& \; \delta(\sigma, x')! \implies x = x'$. (That is, distinct transitions carry distinct event symbols.) This allows us to bring in the operation $(\nrightarrow X')$ for $X' \subseteq X$, whereby the family $\mathbb{C}$ of control patterns is replaced by

$$\mathbb{C}[\mathcal{A}(\nrightarrow X')]$$
$$:= \{\Gamma' \subseteq \Sigma : (\exists \Gamma \in \mathbb{C})[\Gamma \setminus \{\sigma \in \Sigma : (\exists x \in X)[\delta(\sigma, x) \in \rho^{\mathcal{A}}(X')]\} \subseteq \Gamma' \subseteq \Gamma]\}$$

In other words, $\mathcal{A}(\nrightarrow X')$ is the automaton obtained from $\mathcal{A}$ by allowing the disablement of events that take $\mathcal{A}$ into states belonging to $X'$ or from which $\mathcal{A}$ can be controlled to reach $X'$.

We further assume that $\mathcal{A}$ has the special form

$$(\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in \{0\}\dot{\cup}P\}, \{(R_q, I_q) : q \in \{1\}\dot{\cup}Q\}, \mathbb{C})$$

where $0, 1 \notin P\dot{\cup}Q$ and where

$$R_0 \subseteq D(\mathcal{A}) \cap I_0 \; \& \; I_1 \supseteq L(\mathcal{A}) \cup R_1$$

We now define

$\mathcal{A}(+X')$ to be the automaton obtained from $\mathcal{A}$ by replacing $(R_1, I_1)$ with $(R_1 \cup X', I_1 \cup X')$.

Note that the special form of $\mathcal{A}$ is preserved by all of the above operations, with the exception of the exclusion of either $(R_0, I_0)$ or $(R_1, I_1)$.

Some of the effects of these operations on the controllability subset $F^{\mathcal{A}}$ are summarized in the following result:

PROPOSITION 4.1 *Let* $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in P\}, \{(R_q, I_q) : q \in Q\}, \mathbb{C})$ *and suppose* $x \in X$, $X' \subseteq X$ *and* $r \in P\dot{\cup}Q$. *Then*

$$\begin{array}{ll} \text{(a)} & F^{\mathcal{A}} \cap D(\mathcal{A}) = [\bigcup_{p\in P}(R_p \cap I_p) \cap D(\mathcal{A})] \cup R_0 \\ \text{(b)} & F^{\mathcal{A}} \cup X' \subseteq F^{\mathcal{A}(\hookrightarrow X')} \\ \text{(c)} & F^{\mathcal{A}\restriction X'} \subseteq F^{\mathcal{A}} \cap X' \\ \text{(d)} & F^{\mathcal{A}\downharpoonright r} \subseteq F^{\mathcal{A}} \cap (I_r \cup D(\mathcal{A})) \end{array}$$

Proposition 4.1 is similar to proposition 4.1 of Thistle and Wonham (1994a). Part (a) says that the degenerate states that belong to the controllability subset are exactly those that belong to $R_0$ or to $R_p \cap I_p$, for some $p \in P$. Part (b) states that self-looping of a state subset enlarges the controllability subset (by turning the self-looped states into degenerate states where the acceptance criterion is satisfied). Part (c) says that restriction to a state subset shrinks the controllability subset (by creating degenerate states that fail to satisfy the acceptance criterion). Finally, part (d) asserts that exclusion of a state-subset pair $(R_r, I_r)$ from a Rabin recognition condition shrinks the controllability subset (by strengthening the recognition condition and restricting the automaton to $I_r \cup D(\mathcal{A})$).

## 5. Fixpoint Characterization of $F^{\mathcal{A}}$

We can now write down a fixpoint characterization of the controllability subset:

*Definition 5.1.* Let $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in \{0\}\dot{\cup}P\}, \{(R_q, I_q) : q \in \{1\}\dot{\cup}Q\}, \mathbb{C})$ be an automaton of the special form described above. Then

$$C^{\mathcal{A}} := \nu X_0.\ \mu X_1.\ \left[\theta^{\mathcal{A}(\not\hookrightarrow X_1 \cup (X_0 \cap R_1))}(X_1 \cup R_0) \cup \bigcup_{r \in P\dot{\cup}Q} C_r^{\mathcal{A}(\not\hookrightarrow X_1 \cup (X_0 \cap R_1))}(X_1)\right]$$

where, for any such automaton $\mathcal{A}$ and any $p \in P$, $q \in Q$ and $X_1 \subseteq X$,

$$\begin{aligned} & C_p^{\mathcal{A}}(X_1) := \nu X_2.\ [\theta^{\mathcal{A}}(C^{\mathcal{A}(\hookrightarrow X_1 \cup (X_2 \cap R_p))\downharpoonright p}) \cap I_p] \\ \& \ & C_q^{\mathcal{A}}(X_1) := C^{\mathcal{A}(\hookrightarrow X_1)(+R_q \cap I_q)\downharpoonright q} \end{aligned}$$

(The existence of this fixpoint follows by induction on $|P\dot{\cup}Q|$ from Proposition 5.2 (a) & (c) below.)

This representation of the controllability subset generalizes that of Thistle and Wonham (1994a) by treating the liveness assumption represented by the second family $\{(R_q, I_q) : q \in \{1\}\dot{\cup}Q\}$ of state subset pairs as, in effect, affording greater control over the automaton. In particular, since $\mathcal{A}$ is assumed not to generate any strings that visit $R_1$ infinitely often, we must have $F^{\mathcal{A}} = \nu X_0.\ F^{\mathcal{A}(\not\to X_0 \cap R_1)}$—that is, the controllability subset must be the largest subset $X_0$ from which the automaton can be controlled to the satisfaction of its specification *under the assumption* that any transitions leading to $\rho^{\mathcal{A}}(X_0 \cap R_1)$ can be disabled. Indeed, by forcing the automaton to visit $X_0 \cap R_1$ every time an undesired transition to $\rho^{\mathcal{A}}(X_0 \cap R_1)$ occurs, one can ensure that such undesired transitions will occur only finitely often along any trajectory generated by the controlled automaton. Furthermore, in computing $F^{\mathcal{A}(\not\to X_0 \cap R_1)}$, one may neglect transitions that lead to states already known to belong to $F^{\mathcal{A}}$—thus $F^{\mathcal{A}} = \nu X_0.\ F^{\mathcal{A}(\not\to X_0 \cap R_1)} = \nu X_0.\ \mu X_1.\ F^{\mathcal{A}(\not\to X_1 \cup (X_0 \cap R_1))}$. These observations motivate the use of the operation $(\not\to X_1 \cup (X_0 \cap R_1))$ with $X_0$ quantified by $\nu$ and $X_1$ by $\mu$.

The rest of the expression for $C^{\mathcal{A}}$ has an interpretation similar to that of the fixpoint characterization of Thistle and Wonham (1994a). For $\mathcal{A}$ to be suitably controlled, it must be forced eventually to reach $R_0$, or eventually to reach $I_r$ for some $r \in P\dot{\cup}Q$, and remain within that subset, satisfying the acceptance condition. For any automaton $\mathcal{A}$, the term $\theta^{\mathcal{A}}(X_0 \cup R_0)$ represents the set of states from which $\mathcal{A}$ can be controlled to reach $X_1 \cup R_0$ in a single transition; for $r \in P\dot{\cup}Q$, the term $C_r^{\mathcal{A}}(X_1)$ represents the set of states $x \in I_r$ from which $\mathcal{A}$ can be controlled to remain within $I_r \cup D(\mathcal{A})$ and generate only strings accepted by $\mathcal{A}_x$, or eventually to reach the subset $X_1$. It follows that the expression for $C^{\mathcal{A}}$ indeed denotes the controllability subset. (Note that $C_p^{\mathcal{A}}$ and $C_q^{\mathcal{A}}$ are defined in terms of subsets $C^{\mathcal{A}'}$ only for automata $\mathcal{A}'$ with fewer state subset pairs than $\mathcal{A}$, so $C^{\mathcal{A}}$ is well defined.)

Before proving formally that $C^{\mathcal{A}} = F^{\mathcal{A}}$ (Theorem 5.3 below), we state some properties of $C^{\mathcal{A}}$.

PROPOSITION 5.2 *Let $\mathcal{A}$ be a deterministic automaton of the form assumed in section 4. Suppose $X' \subseteq X$. Then*

(a) *If $\mathcal{A}'$ is obtained from $\mathcal{A}$ by replacing $\mathbb{C}$ with $\mathbb{C}' \supseteq \mathbb{C}$, then $C^{\mathcal{A}'} \supseteq C^{\mathcal{A}}$.*

(b) $$C^{\mathcal{A}} \cap D(\mathcal{A}) = [\textstyle\bigcup_{p \in P}(R_p \cap I_p) \cap D(\mathcal{A})] \cup R_0$$

(c) $$C^{\mathcal{A}(\hookrightarrow X')} \supseteq C^{\mathcal{A}} \cup X'$$

(d) $$C^{\mathcal{A}(\hookrightarrow X')} = C^{\mathcal{A}} \iff X' \subseteq C^{\mathcal{A}}$$

(e) $$C^{\mathcal{A} \upharpoonright X'} \subseteq C^{\mathcal{A}} \cap X'$$

(f) $$\forall r \in P\dot{\cup}Q\ :\ C^{\mathcal{A}\downarrow r} \subseteq C^{\mathcal{A}} \cap [I_r \cup D(\mathcal{A})]$$

(g) $$X' \subseteq C^{\mathcal{A}} \implies C^{\mathcal{A}(\not\to X')} = C^{\mathcal{A}}$$

(h) $$C^{\mathcal{A}} = \nu X_0.\ C^{\mathcal{A}(\not\to X_0 \cap R_1)} = \nu X_0.\ \mu X_1.\ C^{\mathcal{A}(\not\to X_1 \cup (X_0 \cap R_1))}$$

$$\text{(i)} \qquad \forall q \in Q \;:\; C^{\mathcal{A}} \cap [I_q \cup D(\mathcal{A}) \supseteq C^{\mathcal{A}(+R_q \cap I_q)\downarrow q}$$

$$\text{(j)} \qquad C^{\mathcal{A}} = \theta^{\mathcal{A}}(C^{\mathcal{A}})$$

$$\begin{aligned} &\text{(k)}\ \forall p \in P \;: \\ &\quad L(\mathcal{A}) \subseteq I_p \;\;\&\;\; X' \supseteq C^{\mathcal{A}(\hookrightarrow X' \cap R_p \cap I_p)} \cap R_p \cap I_p \\ &\qquad\qquad \Longrightarrow\;\; C^{\mathcal{A}(\hookrightarrow X' \cap R_p \cap I_p)\downarrow p} = C^{\mathcal{A}(\hookrightarrow X' \cap R_p \cap I_p)} \end{aligned}$$

$$\begin{aligned} &\text{(l)}\ \forall p \in P \;: \\ &\quad L(\mathcal{A}) \subseteq I_p \;\;\Longrightarrow\;\; C^{\mathcal{A}} = \nu X_2.\, \theta^{\mathcal{A}}(C^{\mathcal{A}(\hookrightarrow X_2 \cap R_p \cap I_p)}) \end{aligned}$$

**Proof:** See the addendum (Thistle 1994a). ■

Proposition 5.2 generalizes proposition 6.2 of Thistle and Wonham (1994a), and shows that the fixpoint $C^{\mathcal{A}}$ has many properties that one would expect of the controllability subset $F^{\mathcal{A}}$, including those of proposition 4.1. Part (a) says that strengthening the controllability mechanism enlarges $C^{\mathcal{A}}$. Part (b) states that a degenerate state belongs to $C^{\mathcal{A}}$ if and only if looping infinitely through that state alone satisfies the acceptance condition of $\mathcal{A}$—this is the counterpart of proposition 4.1 (a).

Part (c) is the counterpart of proposition 4.1 (b); it says that self-looping enlarges $C^{\mathcal{A}}$. On the other hand, part (d) says that the self-looping of states already belonging to $C^{\mathcal{A}}$ does not enlarge the fixpoint $C^{\mathcal{A}}$.

Part (e) asserts that restriction shrinks $C^{\mathcal{A}}$, just as it does $F^{\mathcal{A}}$ (cf. proposition 4.1 (c)), while Part (f) says that exclusion of a state subset pair has a similar effect (see proposition 4.1 (d)).

Part (g) says that if execution of a transition would allow the system to be controlled into the fixpoint $C^{\mathcal{A}}$, then allowing the disablement of that transition does not enlarge the fixpoint. Part (h) captures the following property of the controllability subset: if from any element of a given state subset $X_0$, one can control the automaton either to satisfy its acceptance condition or to enter $X_0 \cap R_1$, then (by repeating this process as necessary) one can control the automaton from any initial state in $X_0$ to the satisfaction of its acceptance condition.

Part (i) reflects the evident fact that if—under the additional liveness assumption that $\mathcal{A}$ not visit $R_q \cap I_q$ infinitely often—one can control the automaton to remain within $I_q \cup D(\mathcal{A})$ and satisfy its acceptance condition, then one can suitably control the automaton under the unaltered liveness assumption.

Part (j) simply asserts that the fixpoint $C^{\mathcal{A}}$, like the controllability subset, is a fixpoint of $\theta^{\mathcal{A}}$.

Part (k) captures the fact that if all live states belong to some $I_p$, and if from all states of some subset $X_2$ one can control the automaton either to satisfy its acceptance condition or eventually to enter $X_2 \cap R_p \cap I_p$, then (by repeating as necessary) one can control the automaton from any initial state in $X_2$ to the satisfaction of its acceptance condition.

Given these properties of the fixpoint $C^{\mathcal{A}}$ we are now ready to prove that it equals the controllability subset $F^{\mathcal{A}}$:

THEOREM 5.3 *Let $\mathcal{A}$ be an automaton of the special form assumed in section 4. Then*

$$F^{\mathcal{A}} = C^{\mathcal{A}}$$

**Proof:** See the appendix. ■

The proof generalizes that of proposition 6.1 of Thistle and Wonham (1994a). The inclusion ($\supseteq$) is the more straightforward: on the basis of the definition of the fixpoint we construct a suitable feedback map. Here we exploit the fact that greatest fixpoints correspond roughly to control-invariant subsets and least fixpoints to "control-reachability" subsets, from which the automaton can be controlled eventually to enter some given state subset.

In contrast to that of Thistle and Wonham (1994a), this feedback map cannot be represented as a state feedback control for the automaton $\mathcal{A}$: an extra bit of information must first be added to the automaton state. Indeed the controller consists of two state feedback controls for $\mathcal{A}$. One of these is based on the assumption represented by the operation ($\not\rightarrowtail X_1 \cup (X_0 \cap R_1)$) that transitions leading to $\rho^{\mathcal{A}}(X_1 \cup (X_0 \cap R_1))$ can be disabled. This controller forces satisfaction of the acceptance condition as long as the assumption is violated only finitely often. Whenever this assumption is violated, the second state feedback map is applied, and forces the automaton into the subset $X_1 \cup (X_0 \cap R_1)$. In this way, the assumption can only be violated finitely often without violating the liveness assumption represented by the second Rabin condition. An extra bit of memory must be added to the state of the automaton to indicate which feedback map should apply at any given point in the system's evolution.

The inclusion ($\subseteq$) is proved by structural induction—in particular, by induction on the number of live states. The general form of the argument is based on that of Rabin (1972).

## 6. Complexity Analysis

The fixpoint characterization allows straightforward computation of the controllability subset (and of a suitable controller, if $C^{\mathcal{A}}$ is nonempty) by iteration of the appropriate monotone operators (see Thistle and Wonham 1994a, Thistle 1994a). In this section, we show that this algorithm is essentially optimal in computational complexity: namely, though the problem of deciding membership in the controllability subset is NP-complete, our method of computation is polynomial in the number of states of the automaton, and exponential only in the total number of state subset pairs. The proofs are similar to those of the corresponding results of Thistle and Wonham (1994a), and are omitted.

THEOREM 6.1 *The problem of deciding membership in the controllability subset $F^{\mathcal{A}}$ of a deterministic automaton $\mathcal{A}$ is NP-complete.*

THEOREM 6.2 *The controllability subset of an automaton of the special form assumed above can be computed in time $\bigcirc(kl(mn)^{3m})$, where k is the size of the alphabet, l is the number* *of control patterns, m is the total number of state subset pairs used to define the acceptance conditions and n is the number of states.*

## 7. Example

Consider a version of the example of Thistle and Wonham (1994a) that includes a second Rabin recognition condition representing a liveness assumption.

The automaton $\mathcal{A} = (\Sigma, X, \delta, \{(R_p, I_p) : p \in \{0\}\dot{\cup}P\}, \{(R_q, I_q) : q \in \{1\}\dot{\cup}Q\}, \mathbb{C})$ is pictured in figure 1. The index set $P$ is $\{\alpha, \beta\}$, and $Q$ is $\{\gamma\}$. The subset pairs $(R_\alpha, I_\alpha) = (\{4\}, \{1, 2, 3, 4\})$ and $(R_\beta, I_\beta) = (\{-4\}, \{-1, -2, -3, -4\})$ are represented by the pairs of dotted and dashed boxes; the pair $(R_\gamma, I_\gamma) = (\{-1, 0, 1\}, \{-1, 0, 1\})$ is represented by the solid box. The pairs $(R_0, I_0)$ and $(R_1, I_1)$ both equal $(\emptyset, X)$. We have omitted event symbols from the diagram for clarity—according to our assumptions, each transition carries a distinct event symbol. The family of control patterns is the collection of all subsets that contain all event symbols corresponding to arcs without slashes across them. Arcs with slashes thus represent "controllable" events; those without represent "uncontrollable events."

The controllability subset is computed by nested iteration according to the fixpoint characterization. We begin by setting $X_0 = X$ and $X_1 = \emptyset$ and computing

$$\theta^{\mathcal{A}(\nrightarrow\emptyset)}(\emptyset \cup R_0) \cup \bigcup_{r\in P\dot{\cup}Q} C_r^{\mathcal{A}(\nrightarrow\emptyset)}(\emptyset)$$

$$= \bigcup_{p\in P} \nu X_2.\, [\theta^{\mathcal{A}}(C^{\mathcal{A}(\hookrightarrow X_2\cap R_p)\downharpoonleft p}) \cap I_p] \cup C^{\mathcal{A}(+R_\gamma\cap I_\gamma))\downharpoonleft\gamma}$$

The automaton $\mathcal{A}(\hookrightarrow R_\alpha)\downharpoonleft\alpha$ is pictured in figure 2. By Proposition 5.2 (b), the state $-1$ does not belong to the controllability subset, and therefore, by Proposition 5.2 (j), neither does state 1; hence the controllability subset of $\mathcal{A}(\hookrightarrow R_\alpha)\downharpoonleft\alpha$ is $\{2, 3, 4\}$. Since this subset contains $R_\alpha$, it is in fact the greatest fixpoint $\nu X_2.\, [\theta^{\mathcal{A}}(C^{\mathcal{A}(\hookrightarrow X_2\cap R_\alpha)\downharpoonleft\alpha}) \cap I_\alpha]$. The automaton $\mathcal{A}(+R_\gamma \cap I_\gamma)\downharpoonleft\gamma$ has the empty set as its first family of accepting pairs. It is easy to see that its controllability subset is therefore empty.

Iterating on $X_1$, we next compute

$$\theta^{\mathcal{A}(\nrightarrow X_1)}(X_1) \cup \textstyle\bigcup_{p\in P} \nu X_2.\, [\theta^{\mathcal{A}(\nrightarrow X_1)(\hookrightarrow X_1\cup(X_2\cap R_p))\downharpoonleft p}) \cap I_p] \cup C^{\mathcal{A}(\nrightarrow X_1)(\hookrightarrow X_1)(+R_\gamma\cap I_\gamma)\downharpoonleft\gamma}$$

with $X_1 = \{-4, -3, -2, 2, 3, 4\}$ (by symmetry).

The automaton $\mathcal{A}(\nrightarrow X_1)(\hookrightarrow X_1 \cup R_\alpha)\downharpoonleft\alpha$ is shown in figure 3. Again, by Proposition 5.2 (b) & (j), states $-1$ and 1 do not belong to the controllability subset. Thus the controllability subset is $X_1$ itself, by Proposition 5.2 (b). As this subset contains $R_\alpha$, $X_1$ is the largest fixpoint $\nu X_2.\, [\theta^{\mathcal{A}(\nrightarrow X_1)}(C^{\mathcal{A}(\nrightarrow X_1)(\hookrightarrow X_1\cup(X_2\cap R_\alpha))\downharpoonleft\alpha}) \cap I_\alpha]$.

The automaton $\mathcal{A}(\nrightarrow X_1)(\hookrightarrow X_1)(+R_\gamma \cap I_\gamma)\downharpoonleft\gamma$ is shown in figure 4. In this case the states $-1, 0, 1$ are added to the controllability subset, owing to the first term in the fixpoint characterization (since $R_1 = \{-1, 0, 1\}$ for this automaton).

In the next step of the iteration on $X_1$, $X_1$ is set equal to $\{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$, yielding the fixpoint $X \setminus \{-6, 6\}$. Since $R_1 = \emptyset$, setting $X_0$ to $X \setminus \{-6, 6\}$ does not

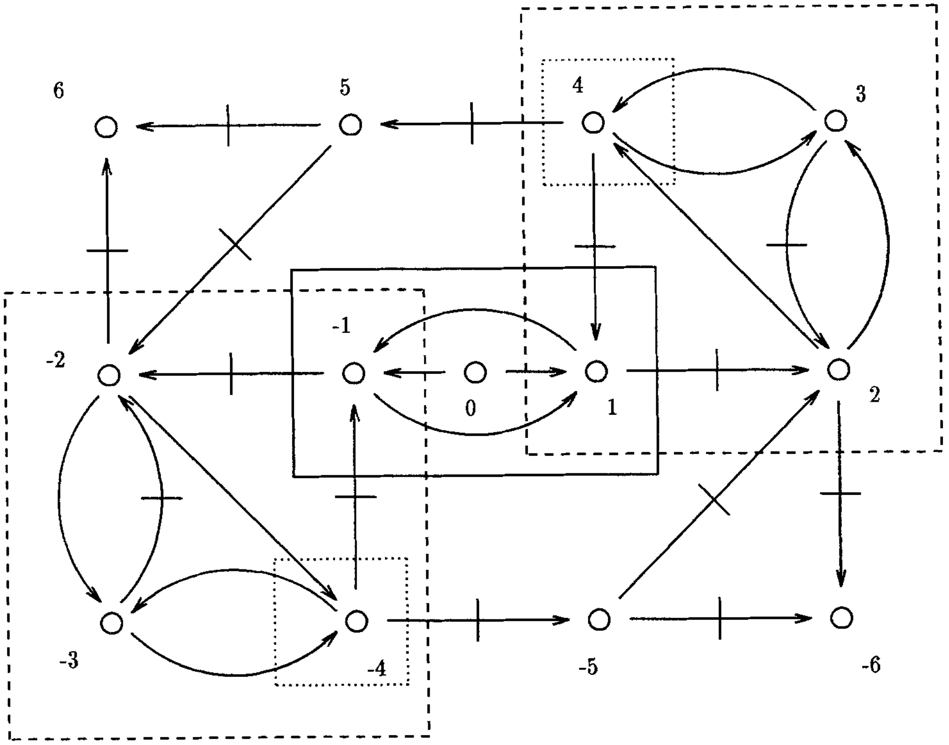

*Figure 1.* Automaton $\mathcal{A}$.

affect the result, so $X \setminus \{-6, 6\}$ in fact represents the controllability subset. As expected, this controllability subset differs from that of the example of Thistle and Wonham (1994a), owing to the liveness assumption represented by $(R_\gamma, I_\gamma)$, without which the states $-1, 0, 1$ do not belong to $C^{\mathcal{A}}$.

## 8. Conclusion

We have extended the methods of Thistle and Wonham (1992, 1994a) and Thistle (1991), to provide a fixpoint characterization of the controllability subset of an automaton whose specifications and liveness assumptions are both represented by Rabin recognition conditions.[10] This result allows for straightforward computation of the controllability subset and for the effective synthesis of suitable supervisors.

The approach of Thistle and Wonham (1992, 1994a) draws heavily on earlier solutions to Church's automaton synthesis problem and the emptiness problem for automata on infinite trees.[11] In particular, it synthesizes some of the methods of Rabin (1972) and Emerson and Jutla (1988). The problem solved in the present report represents an extended version of

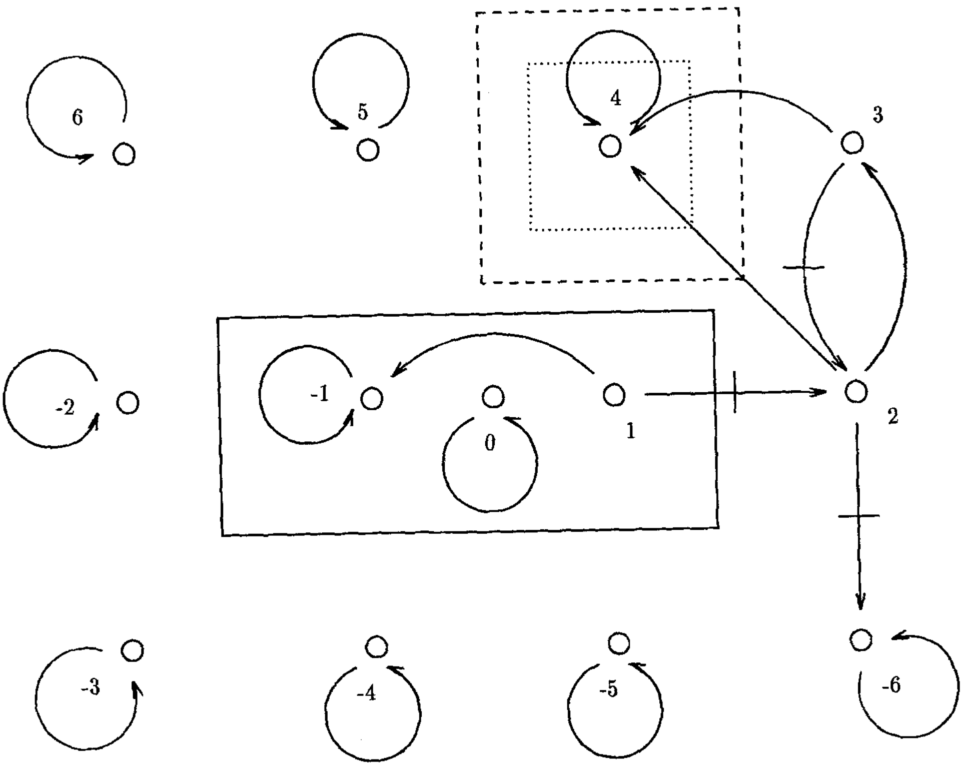

*Figure 2.* Simplified automaton $\mathcal{A}(\hookrightarrow R_\alpha)\downarrow\alpha$

Church's problem that allows for liveness assumptions concerning the "exosystem" with which the system under synthesis is to interact. Such an extension was proposed but not constructively solved in Abadi et al. (1989). The recent solutions to Church's problem given in Emerson and Jutla (1988) and Pnueli and Rosner (1989a) do not appear readily to admit such an extension. A synthesis problem for concurrent systems under $\omega$-regular specifications and modelling assumptions was presented in Wong-Toi and Dill (1991) but this formulation lacks the notion of deadlock freedom implicit in our definition of the controllability subset, and as a consequence reduces to Church's problem. A version of the tree automaton emptiness problem that incorporates fairness assumptions was solved in Courcoubetis et al. (1986) and applied to the satisfiability of branching-time temporal logic formulas in transition structures equipped with Rabin fairness conditions. The application of this emptiness problem to control synthesis is a topic for research.

A central result connected with Church's problem and the emptiness problem is a so-called "small model theorem" (Emerson 1985) that states that if solutions exist then there exists a solution represented by a finite graph embedded in the transition structure of the given automaton. Under the control formulation, this result means that from any state

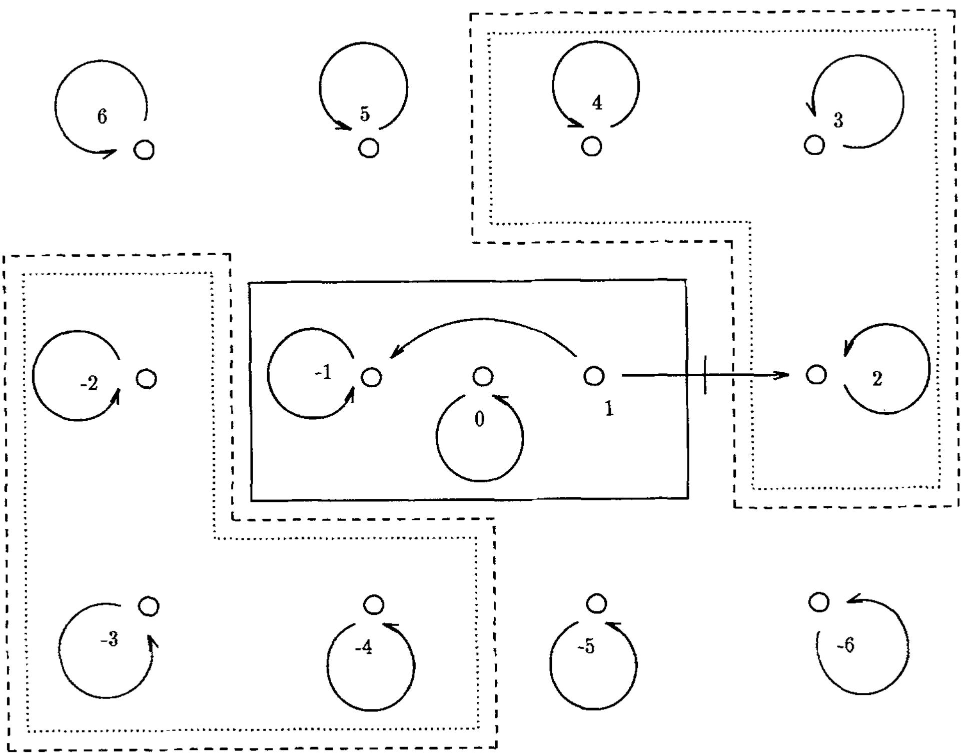

*Figure 3.* Simplified automaton $\mathcal{A}(\not\to X_1)(\hookrightarrow X_1 \cup R_\alpha)\downharpoonleft\alpha$, for $X_1 = \{-4, -3, -2, 2, 3, 4\}$. (The two pairs of dotted and dashed boxes represent a single state subset pair.)

in the controllability subset, the automaton can be controlled to the satisfaction of its acceptance condition by state feedback alone. The theorem is proved by direct construction in Thistle and Wonham (1994a). In the extended problem formulated in the present report, this particular small model theorem fails; however, the proof of Theorem 5.3 shows that, beyond the information contained in the state of the automaton, only one additional bit of information is needed for control (specifically, to establish priorities between the two Rabin conditions).

Other synthesis methods proposed in the control literature allow for less general classes of models and specifications. Ramadge and Golaszewski (Ramadge 1989, Golaszewski and Ramadge 1988) consider only safety specifications. The work of Kumar et al. (1992) is similar in this respect. Young, Spanjol and Garg consider systems modelled by deterministic Büchi automata and subject to the language property of *finite stabilizability* (Young et al. 1992).

The study of different classes of models and specifications of infinite behaviour, with regard to the tradeoff between generality and computational complexity, is a current topic of research. Indeed, in Thistle and Malhamé (1994), automata are equipped with liveness

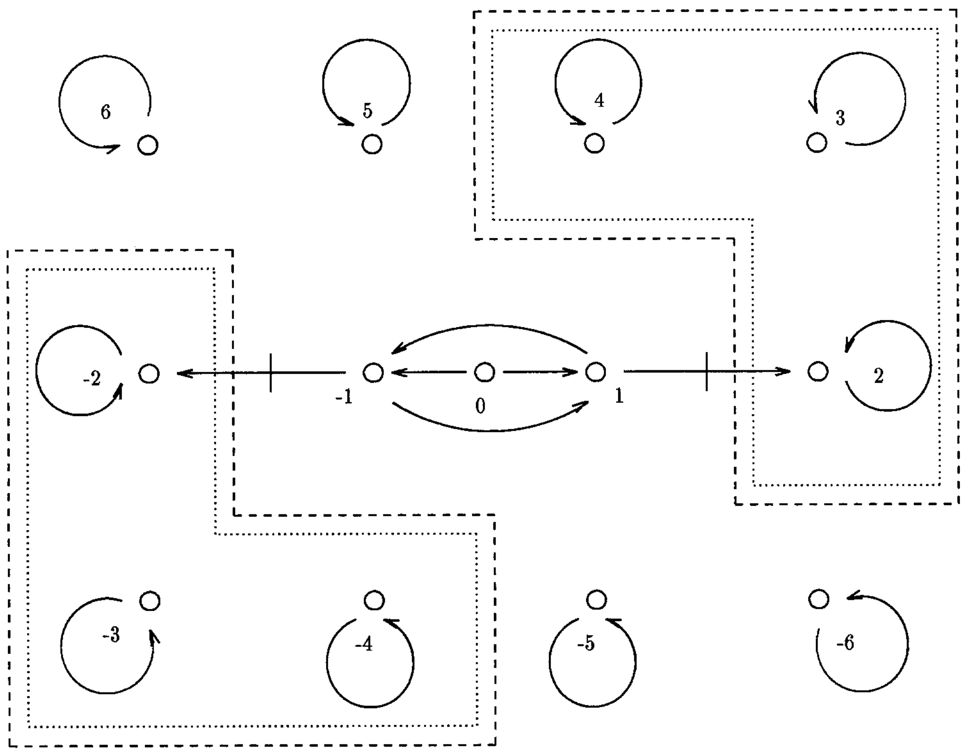

*Figure 4.* Simplified automaton $\mathcal{A}(\not\rightarrow X_1)(\hookrightarrow X_1)(+R_\gamma \cap I_\gamma)\downharpoonleft\gamma$, for $X_1 = \{-4, -3, -2, 2, 3, 4\}$ ($R_1 = \{-1, 0, 1\}$).

assumptions in the form of *state fairness* conditions (Courcoubetis et al. 1986), which state that any state transition that is infinitely often enabled (not only according to the automaton's transition structure, but also according to the action of the controller) must eventually occur. This decouples the control problem, allowing each disjunct in the Rabin acceptance condition to be considered separately; as a consequence, the controllability subset becomes polynomial-time computable. Yet this form of liveness assumption seems a natural one, especially as it is contingent on the action of the controller.

## Appendix

### A1. Proof of Theorem 5.3

We first establish $F^{\mathcal{A}} \subseteq C^{\mathcal{A}}$, proceeding by induction on the number of live states of $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in \{0\}\dot{\cup}P\}, \{(R_q, I_q) : q \in \{1\}\dot{\cup}Q\}, \mathbb{C})$ (Rabin 1972, Thistle and Wonham 1994a), and on the size of the index set $Q$.

Note that

$$\begin{aligned} & F^{\mathcal{A}} \cap D(\mathcal{A}) \\ = \ & [\bigcup_{p \in P}(R_p \cap I_p) \cap D(\mathcal{A})] \cup R_0 \quad \text{(Prop. 4.1 (a))} \\ = \ & C^{\mathcal{A}} \cap D(\mathcal{A}) \qquad\qquad\qquad\qquad \text{(Prop. 5.2 (b))} \end{aligned}$$

It thus suffices to show that $F^{\mathcal{A}} \cap L(\mathcal{A}) \subseteq C^{\mathcal{A}}$.

If $\mathcal{A}$ contains no live states then the result holds vacuously. For the induction step, suppose $x \in F^{\mathcal{A}} \cap L(\mathcal{A})$ and assume that the result holds for all automata with fewer live states than $\mathcal{A}$ or with a smaller index set $Q$. We shall prove that $x \in C^{\mathcal{A}}$. By assumption, there exists some map $f : \Sigma^* \longrightarrow \mathbb{C}$ satisfying both clauses of definition 2.1. Consider the following comprehensively exhaustive set of cases:

(a) there exists a live state $x' \in X$ such that, for all $k' \in \Sigma^*$ generated by $\mathcal{A}_x$ under $f$, $\delta(k', x) \neq x'$;

(b) for some pair of live states $x', x'' \in X$, there exists $k' \in \Sigma^*$ generated by $\mathcal{A}_x$ under $f$ such that $\delta(k', x) = x'$ and for all $k'' \in \Sigma^*$ generated by $\mathcal{A}_x$ under $f$ such that $k' < k''$, $\delta(k'', x) \neq x''$;

(c) for all pairs of live states $x', x'' \in X$, and every $k' \in \Sigma^*$ generated by $\mathcal{A}_x$ under $f$ such that $\delta(k', x) = x'$, there exists $k'' \in \Sigma^*$ generated by $\mathcal{A}_x$ under $f$ such that $k' < k''$ and $\delta(k'', x) = x''$.

In case (a) we have

$$\begin{aligned} x \ & \in \ F^{\mathcal{A}\restriction(X\setminus\{x'\})} \\ & \subseteq \ C^{\mathcal{A}\restriction(X\setminus\{x'\})} \qquad \text{(ind. hyp.)} \\ & \subseteq \ C^{\mathcal{A}} \qquad\qquad \text{(Prop. 5.2 (e))} \end{aligned}$$

Similarly, for case (b) we have

$$x' \in C^{\mathcal{A}}$$

Thus,

$$\begin{aligned} x \ & \in \ F^{\mathcal{A}} \\ & \subseteq \ F^{\mathcal{A}(\hookrightarrow x')} \quad \text{(Prop. 4.1 (b))} \\ & \subseteq \ C^{\mathcal{A}(\hookrightarrow x')} \qquad \text{(ind. hyp.)} \\ & = \ C^{\mathcal{A}} \qquad \text{(Prop. 5.2 (d))} \end{aligned}$$

In case (c), there exists a string $s \in \Sigma^{\omega}$ having a path $\pi$ on $\mathcal{A}_x$ under $f$ such that $\Omega_{\pi} = L(\mathcal{A})$. It follows that one of the following three cases holds:

i. for some $p \in P$, $L(\mathcal{A}) \subseteq I_p \, \& L(\mathcal{A}) \cap R_p \neq \emptyset$;

ii. for some $q \in Q$, $L(\mathcal{A}) \subseteq I_q \, \& L(\mathcal{A}) \cap R_q \neq \emptyset$;

iii. $L(\mathcal{A}) \cap R_1 \neq \emptyset$.

For case i, we have for any $x''' \in L(\mathcal{A})$,

$$\begin{aligned} x''' &\in F^{\mathcal{A}} \\ &= \theta^{\mathcal{A}}(F^{\mathcal{A}}) && \text{(Prop. 3.3.2)} \\ &\subseteq \theta^{\mathcal{A}}(F^{\mathcal{A}(\hookrightarrow L(\mathcal{A})\cap R_p)}) && \text{(Prop. 4.1 (b))} \\ &\subseteq \theta^{\mathcal{A}}(C^{\mathcal{A}(\hookrightarrow L(\mathcal{A})\cap R_p)}) && \text{(ind. hyp.)} \end{aligned}$$

Thus

$$\begin{aligned} L(\mathcal{A}) &\subseteq \nu X_2.\, \theta^{\mathcal{A}}(C^{\mathcal{A}(\hookrightarrow X_2\cap R_p\cap I_p)}) \\ &= C^{\mathcal{A}} && \text{(Prop. 5.2 (l))} \end{aligned}$$

For case ii we have

$$\begin{aligned} x &\in F^{\mathcal{A}(+R_q\cap I_q)\downharpoonleft q} \\ &\subseteq C^{\mathcal{A}(+R_q\cap I_q)\downharpoonleft q} && \text{(ind. hyp.)} \\ &\subseteq C^{\mathcal{A}} && \text{(Prop. 5.2 (i))} \end{aligned}$$

Finally, consider case iii. By definition of $F^{\mathcal{A}}$ there exists a string $s' \in \Sigma^{\omega}$ and a path $\pi'$ of $s'$ on $\mathcal{A}_x$ under $f$ such that $\Omega_{\pi'} \cap R_p \neq \emptyset$ and $\Omega_{\pi'} \subseteq I_p$, for some $p \in P$. It follows that there exists a feedback map $f'$ for $\mathcal{A}(\nrightarrow L(\mathcal{A}) \cap R_1)_x$ satisfying both clauses of the definition of $F^{\mathcal{A}(\nrightarrow L(\mathcal{A})\cap R_1)}$ such that no finite strings generated by $\mathcal{A}(\nrightarrow L(\mathcal{A}) \cap R_1)_x$ under $f'$ but not belonging to pre$(s')$ visit states that belong to $L(\mathcal{A}) \cap R_1$. If we assume without loss of generality that case (i) does not hold then we also have either $\Omega_{\pi'} \cap L(\mathcal{A}) = \emptyset$ or $\Omega_{\pi'} \subsetneq L(\mathcal{A})$. It follows that case (a) or case (b) must hold for the feedback map $f'$ and the automaton $\mathcal{A}(\nrightarrow L(\mathcal{A}) \cap R_1)_x$. Thus $x \in C^{\mathcal{A}(\nrightarrow L(\mathcal{A})\cap R_1)}$. Since this argument holds for arbitrary $x \in L(\mathcal{A}) \subseteq F^{\mathcal{A}}$, we have

$$\begin{aligned} L(\mathcal{A}) &\subseteq \nu X_0.\, C^{\mathcal{A}(\nrightarrow X_0\cap R_1)} \\ &= C^{\mathcal{A}} && \text{(Prop. 5.2 (h))} \end{aligned}$$

This completes the induction, and establishes the containment $F^{\mathcal{A}} \subseteq C^{\mathcal{A}}$.

For the reverse inclusion, let

$$\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in \{0\}\dot{\cup} P\}, \{(R_q, I_q) : q \in \{1\}\dot{\cup} Q\}, \mathbb{C})$$

be an automaton of the special form assumed above. For any $x \in C^{\mathcal{A}}$ we shall construct, by induction on $|P\dot{\cup}Q|$, a feedback map $f$ satisfying definition 2.1. We first define some state feedback maps on $C^{\mathcal{A}}$.

By definition, we have

$$\begin{aligned} C^{\mathcal{A}} \;:=\; \mu X_1.\; & [\theta^{\mathcal{A}(\nrightarrow X_1\cup(C^{\mathcal{A}}\cap R_1))}(X_1 \cup R_0) \\ & \cup \textstyle\bigcup_{p\in P}[\nu X_2.\, [\theta^{\mathcal{A}(\nrightarrow X_1\cup(C^{\mathcal{A}}\cap R_1))}(C^{\mathcal{A}(\nrightarrow X_1\cup(C^{\mathcal{A}}\cap R_1))(\hookrightarrow X_1\cup(X_2\cap R_p))\downharpoonleft p}) \cap I_p]] \\ & \cup \textstyle\bigcup_{q\in Q} C^{\mathcal{A}(\nrightarrow X_1\cup(C^{\mathcal{A}}\cap R_1))(\hookrightarrow X_1)(+R_q\cap I_q)\downharpoonleft q}] \end{aligned}$$

This fixpoint is the least upper bound of the nondecreasing sequence $C_0^{\mathcal{A}} \subseteq C_1^{\mathcal{A}} \subseteq C_2^{\mathcal{A}} \subseteq \cdots$ defined by

$$\begin{aligned}
C_0^{\mathcal{A}} &:= \emptyset \\
C_{i+1}^{\mathcal{A}} &:= \theta^{\mathcal{A}(\not\to C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))}(C_i^{\mathcal{A}} \cup R_0) \\
&\quad \cup \textstyle\bigcup_{p \in P} [\nu X_2.\ [\theta^{\mathcal{A}(\not\to C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))}(C^{\mathcal{A}(\not\to C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))(\hookrightarrow C_i^{\mathcal{A}} \cup (X_2 \cap R_p))\rfloor p}) \cap I_p]] \\
&\quad \cup \textstyle\bigcup_{q \in Q} C^{\mathcal{A}(\not\to C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))(\hookrightarrow C_i^{\mathcal{A}})(+R_q \cap I_q)\rfloor q} \\
&= C_{i+1.0}^{\mathcal{A}} \cup \textstyle\bigcup_{p \in P} C_{i+1.p}^{\mathcal{A}} \cup \bigcup_{q \in Q} C_{i+1.q}^{\mathcal{A}}
\end{aligned}$$

where,

$$C_{i+1.0}^{\mathcal{A}} := \theta^{\mathcal{A}(\not\to C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))}(C_i^{\mathcal{A}} \cup R_0)$$

and for any $p \in P$,

$$\begin{aligned}
C_{i+1,p}^{\mathcal{A}} &:= \nu X_2.\ [\theta^{\mathcal{A}(\not\to C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))}(C^{\mathcal{A}(\not\to C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))(\hookrightarrow C_i^{\mathcal{A}} \cup (X_2 \cap R_p))\rfloor p}) \cap I_p] \\
&= \theta^{\mathcal{A}(\not\to C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))}(C^{\mathcal{A}(\not\to C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))(\hookrightarrow C_i^{\mathcal{A}} \cup (C_{i+1,p}^{\mathcal{A}} \cap R_p))\rfloor p}) \cap I_p
\end{aligned}$$

and for any $q \in Q$,

$$C_{i+1.q}^{\mathcal{A}} := C^{\mathcal{A}(\not\to C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))(\hookrightarrow C_i^{\mathcal{A}})(+R_q \cap I_q)\rfloor q}$$

For the sake of brevity we shall let $\mathcal{A}[i, p]$ stand for $\mathcal{A}(\not\to C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))(\hookrightarrow C_i^{\mathcal{A}} \cup (C_{i+1,p}^{\mathcal{A}} \cap R_p))\rfloor p$ and $\mathcal{A}[i, q]$ for $\mathcal{A}(\not\to C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))(\hookrightarrow C_i^{\mathcal{A}})(+R_q \cap I_q)\rfloor q$.

It is convenient similarly to represent $\rho^{\mathcal{A}}((C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))$ as the least upper bound of a nondecreasing sequence:

$$\begin{aligned}
&\rho^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1)) \\
:= \ &\mu X_2.\ [C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1) \cup \theta^{\mathcal{A}}(X_2)] \\
= \ &\textstyle\bigcup_{j=0}^{\infty} \rho_{i+1,j}^{\mathcal{A}}
\end{aligned}$$

where

$$\begin{aligned}
\rho_{i+1,0}^{\mathcal{A}} &:= C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1) \\
\rho_{i+1,j+1}^{\mathcal{A}} &:= C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1) \cup \theta^{\mathcal{A}}(\rho_{i+1,j}^{\mathcal{A}})
\end{aligned}$$

For any automaton $\mathcal{A} = (\Sigma, X, \delta, x_0, \{(R_p, I_p) : p \in \{0\} \dot\cup P\}, \{(R_q, I_q) : q \in \{1\} \dot\cup Q\}, \mathbb{C})$, we define a state feedback map $\psi_{i+1,0}^{\mathcal{A}} : C_{i+1,0}^{\mathcal{A}} \to \mathbb{C}[\mathcal{A}(\not\to C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))]$, so that for any $x \in C_{i+1,0}^{\mathcal{A}}$, there exists $\sigma \in \psi_{i+1,0}^{\mathcal{A}}(x)$ such that $\delta(\sigma, x)$ is defined, and for all such $\sigma$, $\delta(\sigma, x) \in C_i^{\mathcal{A}} \cup R_0$. Similarly, define a map $\psi_{i+1,p}^{\mathcal{A}} : C_{i+1,p}^{\mathcal{A}} \to \mathbb{C}[\mathcal{A}(\not\to C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))]$, so that for any $x \in C_{i+1,p}^{\mathcal{A}}$, there exists $\sigma \in \psi_{i+1,p}^{\mathcal{A}}(x)$ such that $\delta(\sigma, x)$ is defined, and for all such $\sigma$, $\delta(\sigma, x) \in C^{\mathcal{A}[i,p]}$. Finally, define a map $\chi_{i+1,j+1}^{\mathcal{A}} : \rho_{i+1,j+1} \longrightarrow \mathbb{C}$ such that for any $x \in \rho_{i+1,j+1}$, there exists $\sigma \in \chi^{\mathcal{A}}_{i+1,j+1}(x)$ such that $\delta(\sigma, x)$ is defined, and for all such $\sigma$, $\delta(\sigma, x) \in \rho_{i+1,j}$.

We now use these state feedback maps to construct the feedback map $f$, which will be represented in terms of a transition structure

$$(\Sigma, C^{\mathcal{A}} \times \{0, 1\}, \xi, (x_0, 0))$$

and an accompanying state feedback map

$$\varphi^{\mathcal{A}} : C^{\mathcal{A}} \times \{0, 1\} \longrightarrow \mathbb{C}$$

(Roughly, $\varphi^{\mathcal{A}}$ applies the maps $\psi^{\mathcal{A}}_{i+1,r}$ to ensure acceptance under the assumption that undesired transitions of $\mathcal{A}$ leading from states $x \in C^{\mathcal{A}}_{i+1}$ to $\rho^{\mathcal{A}}(C^{\mathcal{A}}_i \cup (C^{\mathcal{A}} \cap R_1))$ can be prevented. Occurrences of such undesired transitions are recorded by switching the second component of the state from 0 to 1; this causes the state feedback map to apply instead the maps $\chi^{\mathcal{A}}_{i+1,j+1}$, driving $\mathcal{A}$ toward $C^{\mathcal{A}}_i \cup (C^{\mathcal{A}} \cap R_1)$; once this subset (or $R_0$) is reached, the second state component is switched back to 0. The end result is that along a given trajectory that satisfies the liveness assumption, undesired transitions occur only finitely often, so acceptance is ensured.) We leave it to the reader to verify that there does not in general exist a static controller (i.e. one that employs only feedback of the state of $\mathcal{A}$ itself) that applies suitable control.

We define the state feedback map $\varphi^{\mathcal{A}}$ and the transition function $\xi : \sigma \times (C^{\mathcal{A}} \times \{0, 1\}) \longrightarrow C^{\mathcal{A}} \times \{0, 1\}$ by induction on $|P \dot\cup Q|$. Choose an arbitrary total ordering of $\{0\} \dot\cup P \dot\cup Q$. Recalling the first of our state feedback maps, define the *$\psi$-rank* of any state $x \in C^{\mathcal{A}}$ to be the least pair $(i, r)$ in the lexicographic ordering of $\mathbb{N} \times \{0\} \dot\cup P \dot\cup Q$ such that $x \in C^{\mathcal{A}}_{i+1,r}$; define the *$\chi$-rank* of any $x \in C^{\mathcal{A}}$ to be the least pair $(i, j)$ in the lexicographic ordering of $\mathbb{N} \times \mathbb{N}$ such that $x \in \rho^{\mathcal{A}}_{i+1,j+1}$.

We first define $\varphi^{\mathcal{A}}$ and another map,

$$\hat{\varphi}^{\mathcal{A}}_i : C^{\mathcal{A}}_{i+1} \times \{0, 1\} \longrightarrow \mathbb{C}[\mathcal{A}(\not\rightarrow C^{\mathcal{A}}_i \cup (C^{\mathcal{A}} \cap R_1))]$$

by simultaneous induction. Let $x \in C^{\mathcal{A}}_{i+1}$ and $n \in \{0, 1\}$; let $r$ be the least element of $\{0\} \dot\cup P \dot\cup Q$ such that $x \in C^{\mathcal{A}}_{i+1,r}$. Then

$$\hat{\varphi}^{\mathcal{A}}_i(x, n) = \begin{cases} \varphi^{\mathcal{A}[i,r]}(x, n) & \text{if } r \in P \;\&\; x \notin R_r \text{ or if } r \in Q\,; \\ \psi^{\mathcal{A}}_{i+1,r}(x) & \text{otherwise;} \end{cases}$$

Now define $\varphi^{\mathcal{A}} : C^{\mathcal{A}} \times \{0, 1\} \longrightarrow \mathbb{C}$ so that if the $\psi$-rank of $x \in C^{\mathcal{A}}$ is $(i, r)$ and its $\chi$-rank is $(\imath', \jmath')$, then

- whenever $n = 1 \;\&\; \imath' \leq i$, $\varphi^{\mathcal{A}}(x, n) = \chi^{\mathcal{A}}_{\imath'+1,\jmath'+1}(x, n)$;
- and otherwise, $\varphi^{\mathcal{A}}(x, n) \supseteq \hat{\varphi}^{\mathcal{A}}_i(x, n)$ and for all $\sigma \in \varphi^{\mathcal{A}}(x, n) \setminus \hat{\varphi}^{\mathcal{A}}_i(x, n)$, $\delta(\sigma, x) \in \rho^{\mathcal{A}}(C^{\mathcal{A}}_i \cup (C^{\mathcal{A}} \cap R_1))$.

Following the definition of $\varphi^{\mathcal{A}}$, let the *$\varphi$-rank* of any state $(x, n) \in C^{\mathcal{A}} \times \{0, 1\}$ be the least $i \in \mathbb{N}$ s.t. $x \in C^{\mathcal{A}}_{i+1}$, if $n = 0$, and the least $i \in \mathbb{N}$ s.t. $x \in C^{\mathcal{A}}_{i+1} \cup \rho^{\mathcal{A}}(C^{\mathcal{A}}_i \cup (C^{\mathcal{A}} \cap R_1))$ if $n = 1$.

Next define the transition function

$$\begin{aligned} \xi : \ & \Sigma \times (C^{\mathcal{A}} \times \{0,1\}) \longrightarrow C^{\mathcal{A}} \times \{0,1\} \quad (\text{pfn}) \\ & (\sigma, (x, n)) \mapsto (\delta(\sigma, x), \eta^{\mathcal{A}}(\sigma, (x, n))) \quad (\forall \sigma \in \Sigma, x \in C^{\mathcal{A}}, n \in \{0, 1\} \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{s.t. } \delta(\sigma, x) \in C^{\mathcal{A}}) \end{aligned}$$

where the map $\eta^{\mathcal{A}} : \Sigma \times (C^{\mathcal{A}} \times \{0, 1\}) \longrightarrow \{0, 1\}$ is defined inductively as follows: let $i$ be the $\varphi$-rank of $(x, n)$; then

$$\eta^{\mathcal{A}}(\sigma, (x, n)) = \begin{cases} 0 & \text{if } \delta(\sigma, x) \in C^{\mathcal{A}}_i \cup (C^{\mathcal{A}} \cap R_1) \cup R_0, \\ 1 & \text{if } \delta(\sigma, x) \notin C^{\mathcal{A}}_i \cup (C^{\mathcal{A}} \cap R_1) \cup R_0, \\ & \text{and either } \sigma \notin \hat{\varphi}^{\mathcal{A}}_i(x, n) \\ & \text{or both} \\ & \ n = 1 \ \& \ x \in \rho^{\mathcal{A}}(C^{\mathcal{A}}_i \cup (C^{\mathcal{A}} \cap R_1)); \end{cases}$$

If neither of the above conditions holds, the $\psi$-rank of $x$ is $(i, r)$, for some $r \in \{0\} \dot\cup P \dot\cup Q$; then

$$\eta^{\mathcal{A}}(\sigma, (x, n)) = \begin{cases} \eta^{\mathcal{A}[i.r]}(\sigma, (x, n)) & \text{if } r \in P \dot\cup Q; \\ 0 & \text{if } r = 0\,. \end{cases}$$

We must show that for any $x \in X$ the map

$$\begin{aligned} f : \ & \Sigma^* \longrightarrow \mathbb{C} \\ & s \mapsto \varphi^{\mathcal{A}}(\xi(s, (x, 0))) \end{aligned}$$

satisfies the requirements of definition 2.1. Note that by proposition 5.2 (j) we have, for any $p \in P$,

$$\begin{aligned} & (C^{\mathcal{A}[i.p]} \setminus (C^{\mathcal{A}}_i \cup (C^{\mathcal{A}}_{i+1.p} \cap R_p))) \cap I_p \subseteq C^{\mathcal{A}}_{i+1.p} \\ \Longleftrightarrow \quad & C^{\mathcal{A}[i.p]} \cap I_p \subseteq C^{\mathcal{A}}_i \cup C^{\mathcal{A}}_{i+1.p} \end{aligned}$$

It follows by induction on $|P \dot\cup Q|$ that for all $(x, n) \in C^{\mathcal{A}} \times \{0, 1\}$, if $\sigma \in \varphi^{\mathcal{A}}(x, n)$ then $\delta(\sigma, x) \in C^{\mathcal{A}}$—thus $f$ is complete. We also have that if $\sigma \in \varphi^{\mathcal{A}}(x, n)$ and $\delta(\sigma, x) \notin R_0 \dot\cup R_1$ then the $\varphi$-rank of $\xi(\sigma, (x, n))$ is no greater than that of $(x, n)$ itself. Furthermore, if $x \in C^{\mathcal{A}}$ has $\psi$-rank $(i, r)$ and $\sigma \in \hat{\varphi}^{\mathcal{A}}_i(x, n)$ then the $\psi$-rank of $\delta(\sigma, x)$ is no greater than $(i, r)$; and if in addition $r = 0$ and $\delta(\sigma, x) \notin R_0$ then the $\psi$-rank of $\delta(\sigma, x)$ is strictly less than $(i, r)$. For $\chi$-rank we have the following: if the $\chi$-rank of $x$ is $(i, j)$ and $\sigma \in \chi^{\mathcal{A}}_{i+1.j+1}(x)$, then $\delta(\sigma, x)$ is of strictly lower $\chi$-rank than $x$.

Let $s \in \Sigma^\omega$ be any string generated by $\mathcal{A}_x$ under the feedback map $f$. We must show that $s$ is accepted by $\mathcal{A}_x$. It follows from the above observation on $\varphi$-rank that for all sufficiently long prefixes $k$ of $s$, the states $\xi(k, (x, 0))$ all have the same $\varphi$-rank, say $i$. By the observation on $\chi$-rank, we therefore have that for every sufficiently long prefix $k$ of $s$, either the first component of $\xi(k(x, 0))$ is not in $\rho^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))$, or the second component of $\xi(k(x, 0))$ is 0. It follows that $\sigma \in \hat{\varphi}_i^{\mathcal{A}}(\xi(k, (x, 0))$ for all sufficiently long prefixes $k\sigma$ of $s$ (where $\sigma \in \Sigma$). But the result on $\psi$-rank then implies that for all sufficiently long prefixes $k$ of $s$ the states $\delta(k, x)$ all have the same $\psi$-rank, say $(i, r)$. Suppose that $r = 0$: then the observation on $\psi$-rank shows that $\delta(k, x) \in R_0$ for sufficiently long $k$, so the result holds. If instead we have $r \in Q$ then the result holds by inductive assumption (by the definitions of $\hat{\varphi}^{\mathcal{A}}$ and $\eta^{\mathcal{A}}$). Finally, if $r \in P$, then $\delta(k, x) \in I_r$ for sufficiently long $k$ and either $\delta(k, x) \in R_r$ for infinitely many $k$ or, for all sufficiently long prefixes $k\sigma$ of $s$, $\sigma \in \varphi^{\mathcal{A}[i,r]}(\xi(k, (x, 0)))$. The result follows by inductive assumption.

This proves that the map $f$ satisfies clause i of definition 2.1. For clause ii we first show that for any $(x, n) \in C^{\mathcal{A}} \times \{0, 1\}$ and any $t \in \Sigma^{\omega}$ such that $t$ has a path on $\mathcal{A}_x$, if $\sigma \in \varphi^{\mathcal{A}}(\xi(k, (x, n)))$ for all prefixes $k\sigma$ of $t$, then there exist infinitely many prefixes $k$ of $t$ such that the second component of $\xi(k, (x, n))$ is 0. (In other words, we show that along any path through the state set $C^{\mathcal{A}} \times \{0, 1\}$ that is consistent with the feedback map $\varphi^{\mathcal{A}}$ and the transition function $\xi$, the second component of the state is infinitely often 0.)

The proof is similar to that used to establish clause i. Suppose that the result fails for some $t \in \Sigma^{\omega}$. Then for all sufficiently long prefixes $k$, the states $\xi(k, (x, n))$ all have the same $\varphi$-rank, say $i$ (by the above observation on $\varphi$-rank). Furthermore, there must exist only finitely many prefixes $k$ of $t$ such that $\delta(k, x) \in \rho^{\mathcal{A}}(C_i^{\mathcal{A}} \cup (C^{\mathcal{A}} \cap R_1))$ (this by the observation on $\chi$-rank and the definitions of $\chi_{i+1,j+1}^{\mathcal{A}}$ and $\eta^{\mathcal{A}}$). We therefore have that for all sufficiently long prefixes $k\sigma$, $\sigma \in \hat{\varphi}^{\mathcal{A}}(\xi(k, (x, n)))$. It follows by the observation on $\psi$-rank that for all sufficiently long prefixes $k$, the respective states $\delta(k, x)$ have the same $\psi$-rank, say $(i, r)$. If $r = 0$ then by the definition of $\hat{\varphi}_i^{\mathcal{A}}$ the second component of $\xi(k, (x, n))$ is 0 for all sufficiently large $k$, a contradiction. If $r \in Q$ then the result holds by inductive assumption. Finally if $r \in P$ then by the definition of $\eta^{\mathcal{A}}$ we must have $\delta(k, x) \in (C_{i+1}^{\mathcal{A}} \cap R_r) \cup R_0$ for only finitely many prefixes $k$; the result thus follows by inductive assumption.

It now suffices to show that for any $x' \in C^{\mathcal{A}}$ there exists a string $t \in \Sigma^{\omega}$ such that $t$ is accepted by $\mathcal{A}_{x'}$ and for every prefix $k\sigma$ of $t$, $\sigma \in \varphi^{\mathcal{A}}(\xi(k, (x', 0)))$. Strengthening the control action of $\hat{\varphi}_i^{\mathcal{A}}$, define

$$\overline{\varphi}^{\mathcal{A}} : C^{\mathcal{A}} \longrightarrow 2^{\Sigma}$$

$$x \mapsto \begin{cases} \overline{\varphi}^{\mathcal{A}[i,r]} & \text{if } r \in P \;\&\; x \notin R_r \text{ or if } r \in Q; \\ \psi_{i+1,r}^{\mathcal{A}}(x) & \text{otherwise} \end{cases}$$

where $(i, r)$ is the $\psi$-rank of $x$. By induction on $|P \cup Q|$, we have for all $x \in C^{\mathcal{A}}$ and $\sigma \in \overline{\varphi}^{\mathcal{A}}(x)$ that $\eta^{\mathcal{A}}(\sigma, (x, 0)) = 0$; and furthermore, $\emptyset \neq \overline{\varphi}^{\mathcal{A}}(x) \subseteq \hat{\varphi}_i^{\mathcal{A}}(x)$, where $i$ is the least natural number such that $x \in C_{i+1}^{\mathcal{A}}$. The result follows by the definition of $\varphi^{\mathcal{A}}$ and by an induction similar to those performed above.

## Acknowledgements

For the opportunity of discussing this work, the author wishes to thank the organizers and participants of the May 1993 Workshop on Discrete-Event Systems, Manufacturing Systems and Communication Networks held at the Institute for Mathematics and its Applications at the University of Minnesota, and the Atelier Bordeaux-Montréal: Analyse et synthèse de systèmes de transitions, held at the Université du Québec à Montréal in June 1993.

This work was supported in part by a grant offered by the Fonds pour la formation de chercheurs et l'aide à la recherche of the province of Quebec, Canada, Bell-Northern Research Ltd., and the Natural Sciences and Engineering Research Council of Canada under the Action concertée sur les méthodes mathématiques pour la synthèse de systèmes informatiques, and in part by the Natural Sciences and Engineering Research Council of Canada under a Postdoctoral Fellowship and grant no. OGP0155594.

## Notes

1. See Ramadge and Wonham (1989) and Thistle (1994b) for surveys of this language-based theory of the control of discrete event systems.
2. See for example Ramadge (1989), Kumar et al. (1992), Young et al. (1992), Thistle and Wonham (1994b).
3. Or alternatively, the language *marked*.
4. This controllability property is not to be confused with that of Ramadge (1989), which Kumar et al. (1992) also call $\omega$-controllability. The latter property characterizes achievable closed-loop behaviour, much as in the finite-string theory, but is not in general preserved under arbitrary unions, so that a language need not contain a supremal "$\omega$-controllable" sublanguage; the former is a weaker property that is preserved under arbitrary unions and yet, when conjoined with a suitable closure property, still serves to characterize achievable closed-loop languages.
5. See Thomas (1990) for a survey of the theory of automata on infinite objects, which has recently experienced a resurgence of research interest owing to its application to concurrent and reactive systems.
6. See for example Courcoubetis et al. (1986). Any $\omega$-regular language is accepted by some *nondeterministic* Büchi automaton; known algorithms for the "determinization" of such an automaton yield deterministic Rabin automata (McNaughton 1966, Safra 1988).
7. It is worth noting that the approaches of Emerson and Jutla (1988), Pnueli and Rosner (1989a), and Hossley and Rackoff (1972) do not appear readily to admit extension to the present problem.
8. As an anonymous reviewer has pointed out, the use of partial transition functions is technically unnecessary, but it is retained here for physical verisimilitude: intuitively, if $\delta(\sigma, x)$ is undefined, then the event $\sigma$ is "physically impossible" in state $x$.
9. The reader may wish to think of this second family as defining a *Streett* recognition condition, obtained by negating the Rabin condition. The author is grateful to an anonymous reviewer for suggesting this remark.
10. The special case where the liveness assumptions are represented by a Büchi recognition condition was considered in Thistle and Wonham (1992) and Thistle (1991), where the fixpoint characterization of Thistle and Wonham (1994a) was extended through a generalization of the inverse dynamics operator.
11. See Church (1963), Rabin (1972), Emerson and Jutla (1988), Pnueli and Rosner (1989a), Pnueli and Rosner (1989b).

## References

Abadi, M., Lamport, L., and Wolper, P. 1989. Realizable and unrealizable specifications of reactive systems. In *Automata, Languages and Programming*, 16th International Colloquium, Stresa, Italy, July 1989, Proceedings (Lecture Notes in Computer Science no. 372), pp. 1–17. Springer-Verlag.

Alpern, B., and Schneider, F. B. 1985. Defining liveness. *Information Processing Letters* 21: 181–185.

Büchi, J. R., and Landweber, L. H. 1969. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society* 138: 295–311.

Church, A. 1963. Logic, arithmetic and automata. In *Proceedings of the International Congress of Mathematicians*, 15–22 August, 1962, pp. 23–35, Djursholm, Sweden. Institut Mittag-Leffler.

Courcoubetis, C., Vardi, M. Y., and Wolper, P. 1986. Reasoning about fair concurrent programs (extended abstract). In *Symposium on the Theory of Computing*, pp. 283–294. ACM.

Emerson, E. A. 1985. Automata, tableaux and temporal logics. In R. Parikh (Ed.), *Logics of Programs* (Lecture Notes in Computer Science, vol. 193), pp. 79–87. Springer-Verlag.

Emerson, E. A. 1990. Temporal and modal logic. In J. van Leeuwen (Ed.), *Handbook of Theoretical Computer Science*, vol. B: *Formal Models and Semantics*, pp. 995–1072. Elsevier, The MIT Press.

Emerson, E. A., and Jutla, C. S. 1988. The complexity of tree automata and logics of programs (extended abstract). In *29th Annual Symposium on Foundations of Computer Science*, pp. 328–337.

Francez, N. 1986. *Fairness*. Texts and Monographs in Computer Science. New York: Springer-Verlag.

Golaszewski, C. H., and Ramadge, P. J. 1988. Mutual exclusion problems for discrete event systems with shared events. In *Proc. 27th IEEE Conference on Decision and Control*, pp. 234–239.

Hossley, R., and Rackoff, C. 1972. The emptiness problem for automata on infinite trees. In *Switching and Automata Theory Symposium*, pp. 121–124. IEEE.

Kumar, R., Garg, V., and Marcus, S. I. 1992. On supervisory control of sequential behaviors. *IEEE Trans. Automatic Control*, 37(12): 1978–1985.

Kurshan, R. P. 1988. Reducibility in analysis of coordination. In P. Varaiya and A. B. Kurzhanski (Eds.), *Discrete Event Systems: Models and Applications*, IIASA Conference, Sopron, Hungary, Aug. 3–7, 1987 (Lecture Notes in Control and Information Sciences, vol. 103), pp. 19–39. New York: Springer-Verlag.

Lamport, L. 1977. Proving the correctness of multiprocess programs. *ACM Transactions on Software Engineering* SE-3(2): 125–143.

Manna, Z., and Pnueli, A. 1992. *The Temporal Logic of Reactive and Concurrent Systems*, volume 1: *Specification*. New York: Springer-Verlag.

McNaughton, R. 1966. Testing and generating infinite sequences by a finite automaton. *Information and Control* 9: 521–530.

Pnueli, A., and Rosner, R. 1989a. On the synthesis of a reactive module. In *Sixteenth Annual Symposium on Principles of Programming Languages*, pp. 179–190. Association for Computing Machinery.

Pnueli, A., and Rosner, R. 1989b. On the synthesis of an asynchronous reactive module. In *Automata, Languages and Programming*, 16th International Colloquium, Stresa, Italy, July 1989, Proceedings (Lecture Notes in Computer Science no. 372), pp. 652–671. Association for Computing Machinery. Springer-Verlag.

Rabin, M. O. 1969. Decidability of second-order theories and automata on infinite trees. *American Mathematical Society Transactions* 141: 1–35.

Rabin, M. O. 1972. *Automata on Infinite Objects and Church's Problem*. Conference Board of the Mathematical Sciences Regional Conference Series in Mathematics No. 13. Providence, RI: American Mathematical Society. Lectures from the CBMS Regional Conference held at Morehouse College, Atlanta, Georgia, September 8–12, 1969.

Ramadge, P. J., and Wonham, W. M. 1987. Supervisory control of a class of discrete event processes. *SIAM J. Control and Optimization* 25(1): 206–230.

Ramadge, P. J., and Wonham, W. M. 1989. The control of discrete event systems. *Proceedings of the IEEE* 77(1): 81–98.

Ramadge, P. J. G. 1989. Some tractable supervisory control problems for discrete-event systems modeled by Büchi automata. *IEEE Trans. Automatic Control* 34(1): 10–19.

Safra, S. 1988. On the complexity of $\omega$-automata. In *29th Annual Symposium on the Foundations of Computer Science*, pp. 319–327.

Thistle, J. G. 1991. *Control of Infinite Behaviour of Discrete-Event Systems*. Ph.D. thesis, University of Toronto, Toronto, Canada. Available as Systems Control Group Report No. 9012, Systems Control Group, Dept. of

Electl. Engrg., Univ. of Toronto, January 1991.

Thistle, J. G. 1992. Controllability subsets of live Rabin automata. In *31st IEEE Conference on Decision and Control*, pp. 3746–3747.

Thistle, J. G. 1994a. Addendum to 'On control of systems modelled as deterministic Rabin automata'. Contains proofs of intermediate results of article.

Thistle, J. G. 1994b. Logical aspects of control of discrete event systems: A survey of tools and techniques. In G. Cohen and J.-P. Quadrat (Eds.), *11th Int'l Conf. on Analysis and Optimization of Systems, Discrete Event Systems, Sophia-Antipolis*, June 15–16–17, 1994, pp. 3–15. INRIA, École des Mines de Paris. Springer-Verlag. Lecture Notes in Control and Information Sciences 199.

Thistle, J. G., and Malhamé, R. P. 1994. Control of discrete-event systems under state fairness assumptions. In S. Brlek (Ed.), *BMW 94–Méthodes mathématiques pour la synthèse des systèmes informatiques*, pp. 57–66. Laboratoire de combinatoire et d'informatique théorique, Département de mathématiques et d'informatique, Université du Québec à Montréal.

Thistle, J. G., and Wonham, W. M. 1992. Control of $\omega$-automata, Church's problem, and the emptiness problem for tree $\omega$-automata. In E. Börger, G. Jäger, H. K. Büning, and M. M. Richter (Eds.), *Computer Science Logic*: 5th Workshop, CSL '91, Berne, Switzerland, October 1991, Proceedings (Lecture Notes in Computer Science vol. 626), pp. 367–381. Berlin: Springer-Verlag.

Thistle, J. G., and Wonham, W. M. 1994a. Control of infinite behaviour of finite automata. *SIAM J. Control and Optimization* 32(4): 1075–1097.

Thistle, J. G., and Wonham, W. M. 1994b. Supervision of infinite behaviour of discrete-event systems. *SIAM J. Control and Optimization* 32(4): 1098–1113.

Thomas, W. 1990. Automata on infinite objects. In J. van Leeuwen (Ed.), *Handbook of Theoretical Computer Science, vol. B: Formal Models and Semantics*, pp. 134–191. Elsevier, The MIT Press.

Varaiya, P. 1993. Smart cars on smart roads: Problems of control. *IEEE Trans. on Automatic Control* 38(2): 195–207.

Vardi, M. Y. 1991. Verification of concurrent programs: The automata-theoretic framework. *Annals of Pure and Applied Logic* 51: 79–98.

Wong-Toi, H., and Dill, D. L. 1991. Synthesizing processes and schedulers from temporal specifications. In *Computer-Aided Verification* (Proceedings of the CAV 90 Workshop) DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 3, pp. 272–281. American Mathematical Society.

Young, S., Spanjol, D., and Garg, V. K. 1992. Control of discrete event systems modeled with determnistic Büchi automata. In *Proceedings of 1992 American Control Conference*, pp. 2809–2813.