

REPORT DOCUMENTATION PAGE

(2)

1a. REPORT SECURITY CLASSIFICATION		1b. RESTRICTIVE MARKINGS	
AD-A228 961		3. DISTRIBUTION / AVAILABILITY STATEMENT	
		Unlimited DTIC FILE COPY	
4. AUTHOR(OR ORGANIZATION) REPORT NUMBER		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
TR-90-1167			

6a. NAME OF PERFORMING ORGANIZATION	6b. OFFICE SYMBOL (if applicable)	7a. NAME OF MONITORING ORGANIZATION	
Cornell University		Office of Naval Research	
6c. ADDRESS (City, State, and ZIP Code)		7b. ADDRESS (City, State, and ZIP Code)	
Department of Computer Science Upson Hall, Cornell University Ithaca, NY 14853		800 North Quincy Street Arlington, VA 22217-5000	
8a. NAME OF FUNDING / SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
Office of Naval Research		N00014-86-K-0092	
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS	
800 North Quincy Street Arlington, VA 22217-5000		PROGRAM ELEMENT NO.	PROJECT NO.
		TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

Progress Measures for Verification Involving Nondeterminism

12. PERSONAL AUTHOR(S)

Nils Klarlund and Fred B. Schneider

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim	FROM _____ TO _____	1990 October 30	19

16. SUPPLEMENTARY NOTATION

Computer

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
			concurrent program verification, progress measures, safety properties, infinite-state automata

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

➤ Using the notion of progress measures, we give a complete verification method for proving that a program satisfies a property specified by an automaton having bounded nondeterminism. Such automata can express any safety property. Previous methods, which can be derived from the method presented here, either rely on transforming the program or are not complete.

**DISTRIBUTION STATEMENT A**

Approved for public release  
Distribution Unlimited

DTIC  
ELECTE  
NOV 20 1990  
S B D

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT		21. ABSTRACT SECURITY CLASSIFICATION	
<input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL	
Fred B. Schneider	(607) 255-9221		

# Progress Measures for Verification Involving Nondeterminism

Nils Klarlund<sup>1</sup> and Fred B. Schneider<sup>2</sup>

October 30, 1990

## Abstract

Using the notion of progress measures, we give a complete verification method for proving that a program satisfies a property specified by an automaton having bounded nondeterminism. Such automata can express any safety property. Previous methods, which can be derived from the method presented here, either rely on transforming the program or are not complete.

## 1 Introduction

Nondeterministic automata are a convenient mathematical abstraction for programs and specifications that define infinite sequences of events [Arn83, Par81, Sis89b, Var87]. A program is modelled as an automaton  $A_P$ , called the *program automaton*, which accepts a language  $L(A_P)$  of infinite behaviors (words); a specification is modelled as an automaton  $A_S$ , called a *specification automaton*, which accepts the language  $L(A_S)$ .  $A_P$  satisfies  $A_S$  if every behavior of  $A_P$  is allowed by  $A_S$ ; that is, if  $L(A_P) \subseteq L(A_S)$ .

In this article we describe a new method for verifying that  $L(A_P) \subseteq L(A_S)$ . Our approach is based on the notion of *progress measure*, introduced in [Kla90]. A progress measure  $\mu$  for establishing  $L(A_P) \subseteq L(A_S)$  quantifies how a behavior of  $A_P$  converges towards a behavior that would be accepted by  $A_S$ . This convergence is characterized by using a *progress relation*  $\triangleright_S$  (which depends only on  $A_S$ ) and is established by proving the verification condition:

---

<sup>1</sup>Supported by grants from the University of Aarhus, Denmark, the Danish Research Academy, and the Thanks to Scandinavia Foundation Inc.

Current address: IBM T.J. Watson Research Center, PO BOX 704, Yorktown Heights, NY 10598

<sup>2</sup>Supported in part by the Office of Naval Research under contract N00014-86-K-0092, the National Science Foundation under Grant No. CCR-8701103, and Digital Equipment Corporation. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not reflect the views of these agencies.

Current address: Department of Computer Science, Upson Hall, Cornell University, Ithaca, New York 14853

For any transition in  $A_P$  from state  $p$  to  $p'$  emitting symbol  $e$ ,  $\mu(p) \triangleright \mu(p')$  holds.

(In addition, some technical conditions relating the initial states of  $A_P$  and  $A_S$  must hold.)

A verification method based on progress measures for proving  $L(A_P) \subseteq L(A_S)$  is *sound* if the existence of a progress measure implies that  $L(A_P) \subseteq L(A_S)$  holds. In that case, whenever  $p_0, p_1, \dots$  is a run over a behavior  $e_0, e_1, \dots$  in  $L(A_P)$ ,<sup>3</sup> the  $\triangleright$ -related sequence  $\mu(p_0) \triangleright \mu(p_1) \triangleright \dots$  (whose existence is guaranteed by the verification condition above) gives rise to a run of  $A_S$  over  $e_0, e_1, \dots$ . A method is *complete* if such a  $\mu$  is guaranteed to exist whenever  $A_P$  satisfies  $A_S$ .

In this paper we describe some progress measures and corresponding verification methods. The progress measures that we call the *refinement measure*, the *prophecy measure*, and the *history measure* form the basis for the verification methods presented in [Mer89, Lam83, LT87, Par81, Sis89a, Sta88]. A new progress measure, called the *ND measure*, yields a new sound and complete verification method for specifications defined by *safety automata*, infinite-state automata with bounded non-determinism. Such automata can express any safety property,<sup>4</sup> but cannot specify liveness properties [AL88, Kla90]. Thus, this paper describes the use of progress measures to derive a new verification method as well as their use to better understand the power and limitations of existing methods.

The remainder of the paper is organized as follows. In Section 2 we describe some simple progress measures, consider how they might be used in a verification method, and explain why the resulting methods are incomplete. Section 3 discusses properties of refinement, prophecy, history, and ND progress measures. Then Section 4 explains how to reformulate verification methods from the literature for safety properties in terms of these progress measures. Section 5 relates our approach to recursion theory. Section 6 contains a summary.

## 2 Motivation

In this section we consider some candidate progress measures for showing that  $L(A_P) \subseteq L(A_S)$  holds. This leads to a proof that there can be no sound and

---

<sup>3</sup>A run of an automaton is a sequence of automaton states corresponding to a behavior accepted by that automaton.

<sup>4</sup>Informally, a safety property is one stating that some "bad thing" does not happen. Formally, a safety property is a closed set [AS85].

complete verification method based on a progress measure that maps states of  $A_P$  either to states of  $A_S$  or to sets of states of  $A_S$ .

## 2.1 Definitions

Let  $\Sigma$  be a fixed (at most) countable alphabet of symbols called *events* (representing actions, communications, or observable parts of states). A *behavior* is a sequence (infinite if not otherwise stated)  $e_0, e_1, \dots$  of events. Let  $V$  be a set of *states*. A *transition relation on  $V$*  is a relation  $\rightarrow \subseteq V \times \Sigma \times V$ , where a *transition*  $(v, e, v') \in \rightarrow$  is denoted  $v \xrightarrow{e} v'$ . An *automaton  $A = (\Sigma, V, \rightarrow, V^0)$*  consists of an alphabet  $\Sigma$ , a state space  $V$ , a transition relation  $\rightarrow$  on  $V$ , and a set of *initial states*  $V^0 \subseteq V$ . A *run of  $A$  over a behavior  $e_0, e_1, \dots$*  is an infinite  $\rightarrow$ -related sequence of states  $v_0 \xrightarrow{e_0} v_1 \xrightarrow{e_1} \dots$  with  $v_0 \in V^0$ . A behavior  $e_0, e_1, \dots$  is *accepted by  $A$* —or is a *behavior of  $A$* —if there is a run of  $A$  over  $e_0, e_1, \dots$ . The *language or property  $L(A)$  accepted by  $A$*  is the set of behaviors of  $A$ .

Automaton  $A$  is *complete* if  $V \neq \emptyset$  and every state  $v \in V$  appears in some run. Note that a complete automaton accepts a non-empty language. Moreover, from every automaton  $A = (\Sigma, V, \rightarrow, V^0)$  such that  $L(A) \neq \emptyset$ , it is possible to obtain a complete automaton  $A'$  such that  $L(A') = L(A)$  by deleting from  $V$  those states that do not appear in any run. This procedure, however, is not computable, since it requires deciding whether there is an infinite path from a node in a graph—something that is  $\Sigma_1^1$ -complete for countable recursive graphs [Rog67].

Denote by  $v \xrightarrow{u} v'$  that there exist  $v_0, \dots, v_n$  such that  $v = v_0 \xrightarrow{e_0} \dots \xrightarrow{e_n} v_{n+1} = v'$ , where  $u = e_0, \dots, e_n$  is a finite behavior. Similarly,  $v \xrightarrow{w}$  will mean that there exist  $v_0, v_1, \dots$  such that  $v = v_0 \xrightarrow{e_0} v_1 \xrightarrow{e_1} \dots$ , where  $w = e_0, e_1, \dots$  is an infinite behavior. A state  $v$  is *reachable over  $u$*  if there is some  $v^0 \in V^0$  such that  $v^0 \xrightarrow{u} v$ .

In what follows, we consider a program automaton  $A_P = (\Sigma, V_P, \rightarrow_P, V_P^0)$  and a specification automaton  $A_S = (\Sigma, V_S, \rightarrow_S, V_S^0)$ .

## 2.2 Incompleteness of Refinement Measures

To define a verification method, we first consider the use of a progress measure  $\mu$  that maps each state of  $A_P$  to a single state of  $A_S$ . This approach is plausible, for whenever there is a run of  $A_P$  over some behavior  $e_0, e_1, \dots$ , there must be a corresponding behavior of  $A_S$  over  $e_0, e_1, \dots$ . More precisely, the method consists of finding  $\mu$ , called a *refinement measure*, that satisfies two criteria:



For

<input checked="" type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>

Availability Codes  
 Available and/or  
 Special

A-1

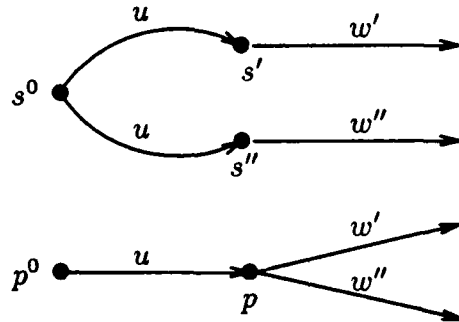
**Definition 1** A refinement measure  $\mu$  for  $(A_P, A_S)$  is a mapping  $\mu : V_P \rightarrow V_S$  such that:

$$(RE\mu 1) \quad p \in V_P^0 \Rightarrow \mu(p) \in V_S^0$$

$$(RE\mu 2) \quad p \xrightarrow{c}_P p' \Rightarrow \mu(p) \xrightarrow{c}_S \mu(p')$$

Verification condition (RE $\mu$ 1) states that  $\mu$  maps any initial state of  $A_P$  to an initial state of  $A_S$ , and (RE $\mu$ 2) states that for every transition  $p \xrightarrow{c}_P p'$  of  $A_P$ , there is a transition  $\mu(p) \xrightarrow{c}_S \mu(p')$  of  $A_S$ . It is not hard to see that together these verification conditions imply that any behavior of  $A_P$  is a behavior of  $A_S$ : let  $e_0, e_1, \dots$  be a behavior of  $A_P$ ; then there is a run  $p_0 \xrightarrow{e_0}_P p_1 \xrightarrow{e_1}_P \dots$  of  $A_P$ , and from (RE $\mu$ 1) and (RE $\mu$ 2) it follows that  $\mu(p_0) \xrightarrow{e_0}_S \mu(p_1) \xrightarrow{e_1}_S \dots$  is a run of  $A_S$ .

Unfortunately, refinement measures do not yield a complete method for non-deterministic automata. Even for finite-state automata  $A_P$  and  $A_S$  such that  $A_P$  satisfies  $A_S$ , a refinement measure may not exist. To see this, assume that  $A_P$  satisfies  $A_S$  and that this can be proved by some refinement measure  $\mu$ . Consider the situation:



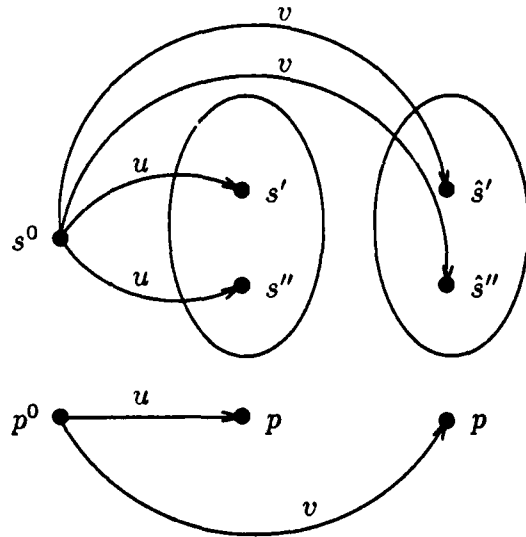
where states  $p$  of  $A_P$  and  $s', s''$  of  $A_S$  are all the states reachable over some finite  $u$ . Also assume that there exist  $w'$  and  $w''$  such that  $u \cdot w'$  and  $u \cdot w''$  are different behaviors. Suppose that  $p_0, \dots, p, p'_0, p'_1, \dots$  is a run of  $A_P$  over  $u \cdot w'$  and that  $p_0, \dots, p, p''_0, p''_1, \dots$  is a run over  $u \cdot w''$ . Thus  $\mu(p_0), \dots, \mu(p), \mu(p'_0), \mu(p'_1), \dots$  must be a run of  $A_S$  over  $u \cdot w'$  and  $\mu(p_0), \dots, \mu(p), \mu(p''_0), \mu(p''_1), \dots$  must be a run of  $A_S$  over  $u \cdot w''$ , because  $A_P$  satisfies  $A_S$ . However, this is impossible because for  $u \cdot w'$ , it must be the case that  $\mu(p) = s'$ , and for  $u \cdot w''$ , it must be the case that  $\mu(p) = s''$ .

### 2.3 Incompleteness of Measures Mapping to Sets of States

To avoid the incompleteness inherent in refinement measures, we might consider a progress measure that maps program states to sets of specification states. For the

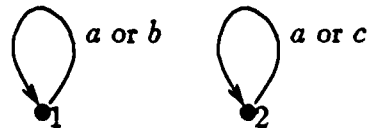
situation above, we would define  $\mu(p) = \{s', s''\}$ , where the set  $\{s', s''\}$  is called a *prophecy set*,<sup>5</sup> because it predicts that either  $s'$  or  $s''$  is the state of the specification automaton corresponding to  $p$ .

Unfortunately, even a method based on mapping program states to sets of specification states cannot be complete. For example, if we employ prophecy sets, a problem arises when a state  $p$  of  $A_P$  can be reached by different behaviors, each giving rise to a different set. Two such sets, corresponding to finite behaviors  $u$  and  $v$ , are depicted below:



It turns out that a complete method must distinguish between such prophecy sets.

To see more formally that no method based on progress measures that map to sets of specification states exists, consider an automaton  $A_S$  given by



where both states 1 and 2 are initial states. The behaviors defined by  $A_S$  are the sequences that consist of either  $a$ 's and  $b$ 's or  $a$ 's and  $c$ 's (i.e. the  $\omega$ -regular language  $(a + b)^\omega \cup (a + c)^\omega$ ). We first show that there can be no progress relation  $\triangleright_S$  on  $V_S = \{1, 2\}$  yielding a reasonable verification method. Such a method would satisfy two criteria:

- i. If  $C_0 \stackrel{e_0}{\triangleright}_S C_1 \stackrel{e_1}{\triangleright}_S C_2 \stackrel{e_2}{\triangleright}_S \dots$ , where  $C_0, C_1, \dots$  are sets of specification states,

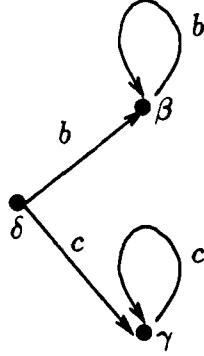
<sup>5</sup>The notions of *prophecy* and *history* are from [AL88].

then there are  $s_0 \in C_0, s_1 \in C_1, \dots$  such that  $s_0 \xrightarrow{e_1}_s s_1 \xrightarrow{e_2}_s \dots$  is a run of  $A_S$ .

- ii. If  $L(A_P) \subseteq L(A_S)$  then a progress measure  $\mu$  exists such that a state  $s$  need only be in  $\mu(p)$  if there is a  $u$  such that both  $p$  and  $s$  are reachable over  $u$  and for some infinite behavior  $w, u \cdot w$  is allowed by  $A_S$ .

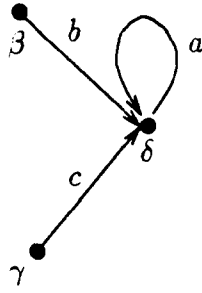
Criterion i must hold for the method to be sound; note that there need not be any condition on  $C_0$  with respect to initial states, because both states of  $A_S$  are initial. Criterion ii is an assumption that if  $L(A_P) \subseteq L(A_S)$ , then a progress measure  $\mu$  exists such that  $\mu(p)$  only contains states that actually occur in runs of  $A_S$  when  $p$  occurs in a corresponding run of  $A_P$ .

Our proof that there is no complete verification method based on a progress relation  $\triangleright_S$  satisfying the criteria involves two programs. The first is  $A_{P_1}$



where state  $\delta$  is the initial state. There are two infinite behaviors of  $A_{P_1}$ , namely  $b, b, b, \dots$  and  $c, c, c, \dots$ , i.e.  $A_{P_1}$  satisfies  $A_S$ . Thus since we assume that the hypothesized method is complete, there must exist a progress measure  $\mu$ . By Criterion i,  $\mu(\delta)$  must contain state 1, because  $\delta \xrightarrow{b}_{P_1} \beta \xrightarrow{b}_{P_1} \beta \xrightarrow{b}_{P_1} \dots$  is a run of  $A_{P_1}$  and the only corresponding run of  $A_S$  is  $1 \xrightarrow{b}_S 1 \xrightarrow{b}_S \dots$ . Similarly,  $\mu(\delta)$  must contain 2,  $\mu(\beta)$  must contain 1, and  $\mu(\gamma)$  must contain 2. By Criterion ii,  $\mu(\beta)$  does not contain 2, and  $\mu(\gamma)$  does not contain 1. Thus  $\mu(\delta) = \{1, 2\}$ ,  $\mu(\beta) = \{1\}$ , and  $\mu(\gamma) = \{2\}$ . Since  $\delta \xrightarrow{b}_{P_1} \beta \xrightarrow{b}_{P_1} \beta \xrightarrow{b}_{P_1} \dots$  is a run of  $A_{P_1}$ ,  $\mu(\delta) \stackrel{b}{\triangleright}_S \mu(\beta) \stackrel{b}{\triangleright}_S \mu(\beta) \stackrel{b}{\triangleright}_S \dots$  must hold. In particular,  $\{1, 2\} \stackrel{b}{\triangleright}_S \{1\}$  must hold. By an analogous argument,  $\{1, 2\} \stackrel{c}{\triangleright}_S \{2\}$  must hold.

The second program is  $A_{P_2}$



whose initial states are  $\beta$  and  $\gamma$ . The behaviors of this program are  $b, a, a, a, \dots$  and  $c, a, a, a, \dots$ . Thus  $A_{P_2}$  satisfies  $A_S$ . By arguments similar to those above,  $\mu(\beta) = \{1\}$ ,  $\mu(\gamma) = \{2\}$ , and  $\mu(\delta) = \{1, 2\}$  hold. Thus  $\{1\} \stackrel{b}{\triangleright}_S \{1, 2\}$  and  $\{2\} \stackrel{c}{\triangleright}_S \{1, 2\}$  must hold.

From  $A_{P_1}$  and  $A_{P_2}$ , we conclude that there is a sequence  $\{1, 2\} \stackrel{c}{\triangleright}_S \{2\} \stackrel{c}{\triangleright}_S \{1, 2\} \stackrel{b}{\triangleright}_S \{1\} \stackrel{b}{\triangleright}_S \{1, 2\} \stackrel{c}{\triangleright}_S \dots$ . However, this contradicts Criterion i because  $A_S$  does not allow the behavior  $c, c, b, b, c, c, b, b, \dots$ .

### 3 Measures for Nondeterministic Automata

We now develop verification methods for establishing  $L(A_P) \subseteq L(A_S)$  by means of progress measures. In particular, we introduce the *ND progress measure* and define the simpler *refinement measure*, *prophecy measure*, and *history measure* along the way. We also give necessary conditions for these measures to constitute complete verification methods.

The following definitions will be required. For an automaton  $A$ , the set of states reachable over  $e_0, \dots, e_n$  is denoted  $\mathcal{R}_A(e_0, \dots, e_n)$ . Note that  $\mathcal{R}_A() = V^0$ . A transition relation  $\rightarrow$  has *bounded nondeterminism* if for all  $e \in \Sigma$  and all  $v \in V$ , the set  $\{v' \mid v \xrightarrow{e} v'\}$  is finite. Automaton  $A = (\Sigma, V, \rightarrow, V^0)$  is a *safety automaton* if  $V^0$  is finite and  $\rightarrow$  has bounded nondeterminism. And, if  $V^0$  and all sets  $\{v' \mid v \xrightarrow{e} v'\}$  have at most one element, then  $A$  is *deterministic*. Observe that if  $A$  is deterministic, then for all  $e_0, \dots, e_n$ , the set  $\mathcal{R}_A(e_0, \dots, e_n)$  has at most one element. If for all  $v$  in  $V$  there is at most one finite behavior  $e_0, \dots, e_n$  such that  $v \in \mathcal{R}_A(e_0, \dots, e_n)$ , then  $A$  is *historical*; the intuition is that each state corresponds to at most one finite behavior or *history* leading up to that state.

#### 3.1 Refinement Measure

By imposing restrictions on  $A_P$  and  $A_S$ , a complete verification method based on refinement measures can be obtained:

**Proposition 1** Let  $A_P$  historical automaton (assumed complete according to the discussion in Section 2.1) and let  $A_S$  be a deterministic automaton. Then,  $L(A_P) \subseteq L(A_S)$  if and only if  $(A_P, A_S)$  has a refinement measure.

**Proof** “ $\Leftarrow$ ” (Soundness) Argument was given in Section 2.2.

“ $\Rightarrow$ ” (Completeness) Let  $p \in V_P$ . Note that:

- By the assumption that  $A_P$  is complete, there is a finite behavior  $e_0, \dots, e_n$  such that  $p \in \mathcal{R}_{A_P}(e_0, \dots, e_n)$ , and by assumption that  $A_P$  is historical, this behavior is unique.
- Since  $A_P$  is complete, there are  $e_{n+1}, e_{n+2}, \dots$  such that  $e_0, e_1, \dots \in L(A_P)$ , whence, by the assumption that  $L(A_P) \subseteq L(A_S)$ ,  $\mathcal{R}_{A_S}(e_0, \dots, e_n) \neq \emptyset$ .
- Since  $A_S$  is deterministic,  $\mathcal{R}_{A_S}(e_0, \dots, e_n)$  has at most one element.

Thus we can define  $\mu(p) = s$ , where  $\{s\} = \mathcal{R}_{A_S}(e_0, \dots, e_n)$  and  $e_0, \dots, e_n$  is the unique finite behavior such that  $p \in \mathcal{R}_{A_P}(e_0, \dots, e_n)$ .

Since  $A_P$  is historical, there is a single initial state  $p^0$ , and by the definition of  $\mu$ ,  $\mu(p^0) = s^0$ , where  $s^0$  is the single initial state of  $A_S$ ; thus (RE $\mu$ 1) is satisfied.

To see that (RE $\mu$ 2) holds, let  $p$ ,  $e$ , and  $p'$  be such that  $p \xrightarrow{e}_P p'$ . Then  $p$  is reachable over a unique finite behavior  $e_0, \dots, e_n$ . Thus  $p'$  is reachable over the finite behavior  $e_0, \dots, e_n, e$ . It follows that  $\mathcal{R}_{A_S}(e_0, \dots, e_n) = \{s\}$  and  $\mathcal{R}_{A_S}(e_0, \dots, e_n, e) = \{s'\}$  with  $s \xrightarrow{e}_S s'$ . Thus  $s = \mu(p) \xrightarrow{e}_S \mu(p') = s'$ .  $\square$

## 3.2 Prophecy Relation and Measure

By imposing restrictions on  $A_P$  and  $A_S$ , a complete method based on mapping program states to prophecy sets can be obtained. For this method, if  $p$  is a program state reachable by  $e_0, \dots, e_n$ ,  $\mu(p)$  is a set of specification states reachable by  $e_0, \dots, e_n$ . Hence on a transition  $p \xrightarrow{e}_P p'$ , the value of the progress measure  $\mu$  should change so that every  $s' \in \mu(p')$  is reachable. This can be assured by requiring that to every state  $s'$  in  $\mu(p')$  there corresponds a state  $s$  in  $\mu(p)$  such that  $s \xrightarrow{e}_S s'$ . Thus we define:

**Definition 2** The *prophecy relation*  $\triangleright_{PR}$  of a transition relation  $\rightarrow$  on  $V$  is the transition relation on  $\mathcal{P}V$  given as:<sup>6</sup>

$$(\triangleright_{PR}) \quad S \stackrel{e}{\triangleright_{PR}} S' \text{ if } \forall s' \in S' : \exists s \in S : s \xrightarrow{e} s'$$

An infinite  $\triangleright_{PR}$ -related sequence of non-empty finite sets gives rise to an infinite  $\rightarrow$ -related sequence of states:

**Lemma 1** (Prophecy Relation Lemma) If  $S_0 \stackrel{e_0}{\triangleright_{PR}} S_1 \stackrel{e_1}{\triangleright_{PR}} \dots$  and  $S_i \neq \emptyset$  is finite for all  $i$ , then there exists a sequence  $s_0 \xrightarrow{e_0} s_1 \xrightarrow{e_1} \dots$  with  $s_i \in S_i$  for  $i \geq 0$ .

**Proof** Construct a forest as follows. Each node is of the form  $s_0 \xrightarrow{e_0} \dots \xrightarrow{e_{n-1}} s_n$  such that  $s_i \in S_i$  for  $i \leq n$  and  $s_i \xrightarrow{e_i} s_{i+1}$  for  $i < n$ ; in particular, the roots are elements of  $S_0$ . The edges are of the form

$$(s_0 \xrightarrow{e_0} \dots \xrightarrow{e_{n-1}} s_n, s_0 \xrightarrow{e_0} \dots \xrightarrow{e_n} s_{n+1}).$$

Since  $S_i$  is finite, the forest is a finite collection of finitely branching trees. The forest is infinite, because for all  $n$ , it follows from  $S_0 \stackrel{e_0}{\triangleright_{PR}} S_1 \stackrel{e_1}{\triangleright_{PR}} \dots$  and  $S_i \neq \emptyset$  that there are some  $s_0, \dots, s_n$  such that  $s_0 \xrightarrow{e_0} \dots \xrightarrow{e_{n-1}} s_n$  is a node. Hence by König's Lemma, there is an infinite path through one of the trees. This path defines  $s_0 \xrightarrow{e_0} s_1 \xrightarrow{e_1} \dots$ .  $\square$

A prophecy progress measure maps each program state to a finite set of specification states:

**Definition 3** A *prophecy measure*  $\mu$  for  $(A_P, A_S)$  is a mapping  $\mu : V_P \rightarrow \mathcal{F}V_S$  such that:<sup>7</sup>

- (PR $\mu$ 1)  $p \in V_P^0 \Rightarrow \mu(p) \subseteq V_S^0$
- (PR $\mu$ 2)  $p \xrightarrow{e_P} p' \Rightarrow \mu(p) \stackrel{e}{\triangleright_{PR}} \mu(p')$
- (PR $\mu$ 3)  $\mu(p) \neq \emptyset$

where  $\triangleright_{PR}$  is the prophecy relation of  $\rightarrow_S$ .

Prophecy measures give a sound and complete verification method for historical program automata and safety specification automata:

<sup>6</sup> $\mathcal{P}V$  denotes the set of all subsets of  $V$

<sup>7</sup> $\mathcal{F}V$  denotes the set of all finite subsets of  $V$

**Proposition 2** Let  $A_P$  be a historical automaton (assumed complete) and let  $A_S$  be a safety automaton. Then,  $L(A_P) \subseteq L(A_S)$  if and only if  $(A_P, A_S)$  has a prophecy measure.

“ $\Leftarrow$ ” Assume that  $(A_P, A_S)$  has a prophecy measure  $\mu$ . Let  $p_0 \xrightarrow{e_0}_P p_1 \xrightarrow{e_1}_P \dots$  be a run of  $A_P$ . By (PR $\mu$ 2),  $\mu(p_0) \stackrel{e_0}{\underset{\text{PR}}{\supseteq}} \mu(p_1) \stackrel{e_1}{\underset{\text{PR}}{\supseteq}} \dots$ , and by (PR $\mu$ 3),  $\mu(p_i) \neq \emptyset$  for  $i \geq 0$ . We can use the Prophecy Relation Lemma to obtain a sequence  $s_0 \xrightarrow{e_0}_S s_1 \xrightarrow{e_1}_S \dots$  where  $s_0 \in \mu(p_0)$ . By (PR $\mu$ 1),  $s_0 \in \mu(p_0) \subseteq V_S^0$ , whence  $s_0 \xrightarrow{e_0}_S s_1 \xrightarrow{e_1}_S \dots$  is a run of  $A_S$ .

“ $\Rightarrow$ ” Assume  $L(A_P) \subseteq L(A_S)$ . Define  $\mu(p) = \mathcal{R}_{A_S}(e_0, \dots, e_n)$ , where  $e_0, \dots, e_n$  is the unique finite behavior such that  $p \in \mathcal{R}_{A_P}(e_0, \dots, e_n)$ ; by the assumption that  $A_P$  is complete, there is a sequence  $e_0, \dots, e_n$ , and by the assumption that  $A_P$  is historical, this sequence is unique. By the assumption that  $A_S$  is a safety automaton,  $\mu(p)$  is finite.

Since  $A_P$  is complete, there are  $e_{n+1}, e_{n+2}, \dots$  such that  $e_0, e_1, \dots \in L(A_P)$ . Since  $L(A_P) \subseteq L(A_S)$ ,  $\mathcal{R}_{A_S}(e_0, \dots, e_n) \neq \emptyset$ . Hence for all  $p$ ,  $\mu(p)$  is nonempty, i.e. (PR $\mu$ 3) holds.

By the assumption that  $A_S$  is historical, there is a single initial state  $p^0$ , and by the definition of  $\mu$ ,  $\mu(p^0) = V_S^0$ ; thus (PR $\mu$ 1) is satisfied.

Finally to prove that (PR $\mu$ 2) holds, let  $p, e$ , and  $p'$  be such that  $p \xrightarrow{e}_P p'$ . There is a unique finite word  $e_0, \dots, e_n$  such that  $p \in \mathcal{R}_{A_P}(e_0, \dots, e_n)$ . Define

$$S' = \{s' \mid \exists s \in \mu(p) : s \xrightarrow{e}_S s'\}.$$

It can be seen that  $S' = \mathcal{R}_{A_S}(e_0, \dots, e_n, e) = \mu(p')$ . By definition of the prophecy relation,  $\mu(p) \stackrel{e}{\underset{\text{PR}}{\supseteq}} \mu(p')$ , whence (PR $\mu$ 2) holds.  $\square$

Note that the proof of the “ $\Rightarrow$ ” direction does not depend on any assumptions about  $A_P$  or  $A_S$ .

It follows from the discussion in Section 2.2 that the method of prophecy measures is not complete if the restriction that  $A_P$  be historical is removed. In the next section, we overcome this limitation by instead imposing a further restriction on the specification automaton.

### 3.3 History Relation and Measure

Assume now that  $L(A_P) \subseteq L(A_S)$  and that specification automaton  $A_S$  is deterministic. Consider a program state  $p$ . It can be reached by different finite behaviors. Let the progress measure  $\mu(p)$  be the *history set*—the set of specification states that are reached by these finite behaviors (there is one such state per

behavior because  $A_S$  is deterministic). On a transition  $p \xrightarrow{e}_P p'$  and for each state  $s \in \mu(p)$ , there must be a state  $s' \in \mu(p')$  such that  $s \xrightarrow{e}_S s'$ ; this ensures that every partial run (history) of  $A_S$  can be extended. Thus we define:

**Definition 4** The *history relation*  $\triangleright_{HI}$  of a transition relation  $\rightarrow$  on  $V$  is the transition relation on  $\mathcal{P}V$  given as:

$$(\triangleright_{HI}) \quad C \stackrel{e}{\triangleright}_{HI} C' \text{ if } \forall s \in C : \exists s' \in C' : s \xrightarrow{e} s'$$

The history relation of  $\rightarrow$  has the following property:

**Lemma 2** (History Relation Lemma) If  $C_0 \stackrel{e_0}{\triangleright}_{HI} C_1 \stackrel{e_1}{\triangleright}_{HI} \dots$ , then for all  $s_0 \in C_0$ , there exists a sequence such that  $s_0 \xrightarrow{e_0} s_1 \xrightarrow{e_1} \dots$  with  $s_i \in C_i$  for all  $i$ .

**Proof** Let  $s_0$  be any state in  $C_0$ . Then by definition of  $\triangleright_{HI}$  there is a state  $s_1$  in  $C_1$  such that  $s_0 \xrightarrow{e_0} s_1$ . By iterating this argument, we obtain  $s_0 \xrightarrow{e_0} s_1 \xrightarrow{e_1} \dots$  such that for all  $i$ ,  $s_i \in C_i$ .  $\square$

A history measure maps a program state to a possibly infinite set of specification states:

**Definition 5** A *history measure*  $\mu$  for  $(A_P, A_S)$  is a mapping  $\mu : V_P \rightarrow \mathcal{P}V_S$  such that

$$(HI\mu 1) \quad p \in V_P^0 \Rightarrow \exists s \in \mu(p) : s \in V_S^0$$

$$(HI\mu 2) \quad p \xrightarrow{e}_P p' \Rightarrow \mu(p) \stackrel{e}{\triangleright}_{HI} \mu(p')$$

where  $\triangleright_{HI}$  is the history relation of  $\rightarrow_S$ .

History measures give a complete verification method for deterministic specification automata:

**Proposition 3** Let  $A_P$  be an automaton (assumed complete) and let  $A_S$  be a deterministic automaton. Then,  $L(A_P) \subseteq L(A_S)$  iff  $(A_P, A_S)$  has a history measure.

**Proof** " $\Leftarrow$ " Let  $p_0 \xrightarrow{e_0}_P p_1 \xrightarrow{e_1}_P \dots$  be a run of  $A_P$ . By (HI $\mu$ 1) there is an  $s_0 \in \mu(p_0)$  such that  $s_0 \in V_S^0$ . By (HI $\mu$ 2),  $\mu(p_0) \stackrel{e_0}{\triangleright}_{HI} \mu(p_1) \stackrel{e_1}{\triangleright}_{HI} \dots$ . Thus by the History Relation Lemma, there is a run  $s_0 \xrightarrow{e_0}_S s_1 \xrightarrow{e_1}_S \dots$  of  $A_S$ .

" $\Rightarrow$ " Assume  $L(A_P) \subseteq L(A_S)$ . Define  $\mu(p) = \bigcup_{p \in \mathcal{R}_{A_P}(e_0, \dots, e_n)} \mathcal{R}_{A_S}(e_0, \dots, e_n)$ , where the union is over all finite behaviors  $e_0, \dots, e_n$  such that  $p \in \mathcal{R}_{A_P}(e_0, \dots, e_n)$ .

To prove (HI $\mu$ 1), let  $p \in V_P^0$ . Note that  $\mathcal{R}_{A_S}()$  contains the initial state  $s^0$  of  $A_S$  because  $A_P$  is complete. Since  $p \in \mathcal{R}_{A_P}()$ , it follows that  $\mathcal{R}_{A_S}() \subseteq \mu(p)$ ; thus  $s^0 \in \mu(p^0)$ .

To see that (HI $\mu$ 2) holds, let  $p$ ,  $e$ ,  $p'$ , and  $s$  be such that  $p \xrightarrow{e}_P p'$  and  $s \in \mu(p)$ . Thus there is a behavior  $e_0, \dots, e_n$  such that  $s \in \mathcal{R}_{A_S}(e_0, \dots, e_n)$  and  $p \in \mathcal{R}_{A_P}(e_0, \dots, e_n)$ . Since  $A_S$  is deterministic,  $\mathcal{R}_{A_S}(u)$  is the singleton  $\{s\}$ . It follows that  $\mathcal{R}_{A_S}(e_0, \dots, e_n, e)$  is a set  $\{s'\}$  such that  $s \xrightarrow{e}_S s'$  and that  $s' \in \mu(p')$ , because  $p' \in \mathcal{R}_{A_P}(e_0, \dots, e_n, e)$ . Thus (HI $\mu$ 2) holds.  $\square$

According to the results of Section 2.2, history measures do not constitute a complete verification method for nondeterministic  $A_S$ .

### 3.4 ND Relation and Measure

We have discussed progress relations that give complete methods for two special cases above: prophecy relations when  $A_P$  is historical and history relations when  $A_S$  is deterministic. Our solution to the general case consists of combining these relations: the ND progress relation is the history relation of the prophecy relation.

**Definition 6** The ND relation  $\triangleright_{ND}$  on  $\mathcal{P}\mathcal{F}V$  of a transition relation  $\rightarrow$  on  $V$  is defined as:

$$(\triangleright_{ND}) \quad C \stackrel{e}{\triangleright}_{ND} C' \quad \text{if} \quad \forall S \in C : \exists S' \in C' : \\ \forall s' \in S' : \exists s \in S : s \xrightarrow{e} s'$$

An immediate consequence of the two preceding lemmas is:

**Lemma 3** (ND Relation Lemma) If  $C_0 \stackrel{e_0}{\triangleright}_{ND} C_1 \stackrel{e_1}{\triangleright}_{ND} \dots$ ,  $S_0 \in C_0$ , and  $\emptyset \notin C_i$  for all  $i$ , then there is a sequence  $s_0 \xrightarrow{e_0} s_1 \xrightarrow{e_1} \dots$  with  $s_0 \in S_0$ .

**Proof** As  $S_0 \in C_0$  and as  $C_0 \stackrel{e_0}{\triangleright}_{HI} C_1 \stackrel{e_1}{\triangleright}_{HI} \dots$ , there is by the History Relation Lemma a sequence  $S_0 \stackrel{e_0}{\triangleright}_{PR} S_1 \stackrel{e_1}{\triangleright}_{PR} \dots$  with  $S_i \in C_i$ .

Moreover since  $\emptyset \notin C_i$ , i.e.  $S_i \neq \emptyset$ , and since  $S_i$  is finite for all  $i$ , it follows by the Prophecy Relation Lemma that there is a sequence  $s_0 \xrightarrow{e_0} s_1 \xrightarrow{e_1} \dots$  such that  $s_0 \in S_0$ .  $\square$

An ND measure  $\mu$  associates with each program state a (history) set of (prophecy) sets of specification states:

**Definition 7** An ND measure  $\mu$  for  $(A_P, A_S)$  is a mapping  $\mu : V_P \rightarrow \mathcal{P}\mathcal{F}V_S$  such that

- (ND $\mu$ 1)  $p \in V_P^0 \Rightarrow \exists S \in \mu(p) : S \subseteq V_S^0$
- (ND $\mu$ 2)  $p \xrightarrow{e}_P p' \Rightarrow \mu(p) \stackrel{e}{\triangleright}_{ND} \mu(p')$
- (ND $\mu$ 3)  $\emptyset \notin \mu(p)$

Our main result is:

**Theorem 1** Let  $A_P$  be an automaton (assumed complete) and  $A_S$  a safety automaton. Then,  $L(A_P) \subseteq L(A_S)$  if and only if  $(A_P, A_S)$  has an ND measure.

**Proof “ $\Leftarrow$ ”** Let  $p_0 \xrightarrow{e_0}_P p_1 \xrightarrow{e_1}_P \dots$  be a run of  $A_P$ . By (ND $\mu$ 2),  $\mu(p_0) \stackrel{e_0}{\underset{ND}{\triangleright}} \mu(p_1) \stackrel{e_1}{\underset{ND}{\triangleright}} \dots$ , and by (ND $\mu$ 1), there is a set  $S_0 \in \mu(p_0)$  such that  $S_0 \subseteq V_S^0$ . Thus by (ND $\mu$ 3) and the ND Relation Lemma, there is  $s_0 \xrightarrow{e_0}_S s_1 \xrightarrow{e_1}_S \dots$  such that  $s_0 \in S_0 \subseteq V_S^0$ . Therefore  $s_0 \xrightarrow{e_0}_S s_1 \xrightarrow{e_1}_S \dots$  is a run of  $A_S$ .

**“ $\Rightarrow$ ”** Assume  $L(A_P) \subseteq L(A_S)$ . Define  $\mu$  such that  $S \in \mu(p)$  if and only if there is a finite behavior  $e_0, \dots, e_n$  such that  $p \in \mathcal{R}_{A_P}(e_0, \dots, e_n)$  and  $S = \mathcal{R}_{A_S}(e_0, \dots, e_n)$ . By the assumption that  $A_P$  is complete, there is for any such behavior  $e_0, \dots, e_n$ , a sequence  $e_{n+1}, e_{n+2}, \dots$  such that  $e_0, e_1, \dots \in L(A_P)$ . Thus since  $L(A_P) \subseteq L(A_S)$ , it follows that  $\mathcal{R}_{A_S}(e_0, \dots, e_n) \neq \emptyset$ . Hence for all  $p$ ,  $\mu(p)$  is a set of nonempty sets, i.e. (ND $\mu$ 3) holds.

To prove that  $\mu$  satisfies (ND $\mu$ 1), assume that  $p \in V_P^0$ . By the definition of  $\mu$ , for all  $p^0 \in V_P^0$ , it holds that  $V_S^0 \in \mu(p^0)$ , whence (ND $\mu$ 1) is satisfied.

To prove that (ND $\mu$ 2) holds, let  $p$ ,  $e$ , and  $p'$  be such that  $p \xrightarrow{e}_P p'$  and let  $S \in \mu(p)$ . Thus there is a finite behavior  $e_0, \dots, e_n$  such that  $p \in \mathcal{R}_{A_P}(e_0, \dots, e_n)$  and  $S = \mathcal{R}_{A_S}(e_0, \dots, e_n)$ . Define

$$S' = \{s' \mid \exists s \in S : s \xrightarrow{e}_S s'\}.$$

It can be seen that  $S' = \mathcal{R}_{A_S}(e_0, \dots, e_n, e) \in \mu(p')$ . By definition of the prophecy relation,  $S \stackrel{e}{\underset{PR}{\triangleright}} S'$ , whence (ND $\mu$ 2) holds.  $\square$

## 4 Derivation of Previous Methods

In this section we derive from our ND measures Abadi and Lamport’s method [AL88] as applied to safety properties. We also show how to obtain the verification method of Merritt [Mer89] and Sistla [Sis89a].

Formulated in our terminology, the goal of [AL88] is to show  $L(A_P) \subseteq L(A_S)$  by means of a refinement measure. This is done by adding *history* and *prophecy information* to the program automaton before the refinement mapping is constructed. This information is such that one can verify locally that the language  $L(A_P)$  accepted does not shrink when it is added. The main result of [AL88] is that  $L(A_P) \subseteq L(A_S)$  if and only if there is an automaton  $A_{P''}$ —obtained by adding first history, then prophecy information to  $A_P$ —and there is a refinement measure of  $(A_{P''}, A_S)$ . The work in [Mer89, Sis89a] represents what can be regarded as an intermediate approach between ours and that of [AL88]. Both [Mer89]

and [Sis89a] rely on modifying the program automaton and using a prophecy measure:  $L(A_P) \subseteq L(A_S)$  if and only if there is an automaton  $A_{P'}$ —obtained by adding history information to  $A_P$ —and there is a prophecy measure of  $(A_{P'}, A_S)$ .

## 4.1 Adding History Information

Using ND measures, we can derive the method of [Mer89, Sis89a] as follows.

**Definition 8** We say that  $A_{P'} = (\Sigma, V_{P'}, \rightarrow_{P'}, V_{P'}^0)$  is obtained from  $A_P$  by adding history information if  $V_{P'} \subseteq V_P \times I$ —with  $I$  countable—and

$$(HI1) \quad p \in V_P^0 \Rightarrow \exists i : (p, i) \in V_{P'}^0$$

$$(HI2) \quad p \rightarrow_P p' \wedge (p, i) \in V_{P'} \Rightarrow \exists i' : (p, i) \rightarrow_{P'} (p', i')$$

Note that (HI1) and (HI2) are equivalent to saying that  $\mu$  defined by  $\mu(p) = \{(p, i) \in V_{P'}\}$  is a history measure for  $(A_P, A_{P'})$ . Also observe that  $A_{P'}$  is not necessarily a complete automaton or a safety automaton, even if  $A_P$  is.

**Proposition 4** If  $A_{P'}$  is obtained from  $A_P$  by adding history information, then  $L(A_P) \subseteq L(A_{P'})$ .

**Proof** As noted above,  $\mu = \{(p, i) \in V_{P'}\}$  is a history measure for  $(A_P, A_{P'})$ . Thus by Proposition 3,  $L(A_P) \subseteq L(A_{P'})$  holds.  $\square$

The method of [Mer89, Sis89a] now follows from Theorem 1:

**Corollary 1** (of Theorem 1) Let  $A_P$  be a complete automaton and let  $A_S$  be a safety automaton. Then,  $L(A_P) \subseteq L(A_S)$  if and only if there is an automaton  $A_{P'}$ —obtained by adding history information to  $A_P$ —and there is a prophecy measure for  $(A_{P'}, A_S)$ .

**Proof “ $\Leftarrow$ ”** By Proposition 4,  $L(A_P) \subseteq L(A_{P'})$ , and by Proposition 2,  $L(A_{P'}) \subseteq L(A_S)$ .

**“ $\Rightarrow$ ”** By Theorem 1 there is an ND measure  $\mu_{ND}$  for  $(A_P, A_S)$ . Let  $I = \mathcal{F}V_S$ ,  $V_{P'} = \{(p, S) \mid S \in \mu_{ND}(p)\}$ ,  $V_{P'}^0 = \{(p, S) \in V_{P'} \mid p \in V_P^0, S \subseteq V_S^0\}$ , and  $(p, S) \xrightarrow{\varepsilon_{P'}} (p', S')$  if  $p \xrightarrow{\varepsilon_P} p'$  and  $S \stackrel{\varepsilon}{\vdash}_{PR} S'$ .

$A_{P'}$  is obtained from  $A_P$  by adding history information. In fact, (HI1) is satisfied, because if  $p \in V_P^0$ , then by (ND $\mu$ 1) there is a  $S \in \mu_{ND}(p)$  such that  $S \subseteq V_S^0$ , thus  $(p, S) \in V_{P'}^0$ . Similarly, (HI2) follows from (ND $\mu$ 2).

To finish the proof, we define the prophecy measure for  $(A_{P'}, A_S)$  as  $\mu(p, S) = S$ . Then (PR $\mu$ 1), (PR $\mu$ 2), and (PR $\mu$ 3) can be shown to hold.  $\square$

The completeness proofs of the methods in [AL88, Mer89, Sis89a] rely on changing  $A_P$  to an infinite-state automaton by adding information that records the past history of states. In contrast, the analysis above shows that if  $A_P$  and  $A_S$  are finite-state, then  $A_{P'}$  can be chosen to be finite-state; for in the proof of Corollary 1, the number of different history sets is finite when  $V_P$  is finite. In light of this observation, the concepts of history measure and history information are a bit misleading. Distinguishing among histories of the program automaton is not a cardinal point—what matters is to distinguish among prophecy sets of the specification automaton.

## 4.2 Adding Prophecy Information

To obtain the verification method of [AL88], we define:

**Definition 9**  $A_{P'} = (\Sigma, V_{P'}, V_{P'}^0, \rightarrow_{P'})$  is obtained from  $A_P$  by *adding* prophecy information if  $V_{P'} \subseteq V_P \times I$ —with  $I$  countable—and

$$(PR1) \quad p \in V_P^0 \wedge (p, i) \in V_{P'} \Rightarrow (p, i) \in V_{P'}^0$$

$$(PR2) \quad p \xrightarrow{c_P} p' \wedge (p', i') \in V_{P'} \Rightarrow \exists i : (p, i) \xrightarrow{c_{P'}} (p', i')$$

$$(PR3) \quad \emptyset \neq \{i \mid (p, i) \in V_{P'}\} \text{ is finite}$$

Requiring (PR1), (PR2), and (PR3) is equivalent to stating that  $\mu$  defined as  $\mu(p) = \{(p, i) \in V_{P'}\}$  is a prophecy measure of  $(A_P, A_{P'})$ . Also observe that  $A_{P'}$  is not necessarily a safety automaton nor is it necessarily complete, even if  $A_P$  has these properties.

**Proposition 5** If  $A_{P'}$  is a safety automaton obtained from  $A_P$  by adding prophecy information, then  $L(A_P) \subseteq L(A_{P'})$ .

**Proof** Follows from Proposition 2 and from the observation above that  $\mu(p) = \{(p, i) \in V_{P'}\}$  is a prophecy measure of  $(A_P, A_{P'})$ .  $\square$

A version of Theorem 2 of [AL88] follows from Theorem 1:

**Corollary 2** (of Theorem 1) Let  $A_P$  be an automaton (assumed complete) and let  $A_S$  be a safety automaton. Then  $L(A_P) \subseteq L(A_S)$  if there is a safety automaton  $A_{P''}$ —obtained by adding first history, then prophecy information to  $A_P$ —and there is a refinement measure for  $(A_{P''}, A_S)$ .

**Proof “ $\Leftarrow$ ”** By Proposition 4 and Proposition 5,  $L(A_P) \subseteq L(A_{P''})$ . Moreover, it is easy to see that every run of  $A_{P''}$  induces a run of  $A_S$ ; thus  $L(A_P) \subseteq L(A_{P''}) \subseteq L(A_S)$ .

" $\Rightarrow$ " Assume  $L(A_P) \subseteq L(A_S)$ . By Theorem 1 there is an ND measure  $\mu$  of  $(A_P, A_S)$ . Let  $A_{P''} = (\Sigma, V_{P''}, \rightarrow_{P''}, V_{P''}^0)$ , where  $V_{P''}^0, V_{P''} \subseteq V_P \times (V_S \times \mathcal{F}V_S)$  are given by:

$$\begin{aligned} V_{P''} &= \{(p, s, S) \mid s \in S \in \mu(p)\} \\ V_{P''}^0 &= \{(p, s, S) \mid s \in S \in \mu(p) \wedge p \in V_P^0 \wedge S \subseteq V_S^0\} \end{aligned}$$

and  $(p, s, S) \xrightarrow{e}_{P''} (p', s', S')$  if  $p \xrightarrow{e}_P p'$ ,  $s \xrightarrow{e}_S s'$ , and  $S \stackrel{e}{\underset{PR}{\triangleright}} S'$ .

Then it is not hard to see that  $A_{P''}$  is obtained by adding prophecy information to  $A_P$  from the proof of Corollary 1. Also, it can be seen that  $\mu_{re}$  defined by  $\mu_{re}(p, s, S) = s$  is a refinement measure.  $\square$

## 5 Discussion

Our verification methods hinge on two restrictions: that the specification automaton has only bounded nondeterminism and that the program automaton is complete. The restriction to bounded nondeterminism is also imposed in previous methods. As discussed in [Sis89b], there are recursion-theoretic arguments showing that there does not exist any reasonable verification method for automata having unbounded nondeterminism.

The restriction to complete program automata is also rooted in the laws of recursion theory. Just to determine if an effectively presented nondeterministic automaton defines the empty set (i.e. that it has no infinite runs) is  $\Pi_1^1$ -complete, because there is a reduction from the  $\Pi_1^1$ -complete problem of determining whether an effectively represented tree has only finite paths. On the other hand, all the methods described here involve a second order existential quantification; i.e. each method is of the form:  $L(A_P) \subseteq L(A_S)$  if and only if there is a relation  $R$  such that some first-order conditions hold.<sup>8</sup> Thus the methods are essentially  $\Sigma_1^1$  and therefore cannot possibly be used for the general problem  $L(A_P) \subseteq L(A_S)$ , where  $A_P$  is nondeterministic and  $A_S$  is a safety automaton.

One can lower the computational complexity by reformulating the verification problem. We say that  $A_P$  *simulates*  $A_S$  if each finite and infinite behavior of  $A_P$  is a behavior of  $A_S$ . Paradoxically, the problem of determining whether nondeterministic  $A_P$  simulates safety automaton  $A_S$ —something that looks stronger than

<sup>8</sup>An ND measure  $\mu$  can be defined by  $S \in \mu(p)$  if and only if  $R(p, \#S)$ , where  $\#S$  is a number encoding the finite set  $S$ .

$L(A) \subseteq L(S)$ —is computationally much easier. In fact it can be shown that this problem is  $\Pi_1^0$ -complete.

Thus in order to avoid dealing with the rather strange concept of complete automata, it is not surprising that earlier papers [AL88, Mer89, Sis89a] are concerned with methods for showing that  $A_P$  simulates  $A_S$ . Whether one considers only infinite behaviors or both finite and infinite behaviors, there is no substantial difference in how automata are related—except for the treatment of reachable program states that are not parts of any run. In the first case they have to be excluded from consideration, in the second case they matter.

All our results are applicable for showing that  $A_P$  simulates  $A_S$ ; the only change is that a measure becomes a partial function, because there may be unreachable states of  $A_P$ . For example, Theorem 1 becomes:

**Theorem 1'** Let  $A_P$  be a nondeterministic automaton and  $A_S$  a safety automaton (with  $V_S^0 \neq \emptyset$ ). Then  $A_P$  simulates  $A_S$  if and only if  $(A_P, A_S)$  has a partial ND measure.

In the statement of Theorem 1' we define

**Definition 7'** A *partial ND measure*  $\mu$  for  $(A_P, A_S)$  is a partial mapping  $\mu : V_P \rightrightarrows \mathcal{P}\mathcal{F}V_S$  such that  $V_P^0 \subseteq \text{dom}\mu$  and for all  $p, p' \in \text{dom}\mu$ :

$$(ND\mu 1) \quad p \in V_P^0 \Rightarrow \exists S \in \mu(p) : S \subseteq V_S^0$$

$$(ND\mu 2) \quad p \xrightarrow{e}_P p' \Rightarrow \mu(p) \stackrel{e}{\dot{\subseteq}}_{ND} \mu(p')$$

$$(ND\mu 3) \quad \emptyset \notin \mu(p)$$

Here the reachable states are defined by  $\text{dom}\mu$ , which can be identified using traditional assertional techniques.

The approach of [AL88] is more general than ours in two respects. First, they show how safety and liveness issues can be separated by using automata that are equipped with auxiliary liveness properties. Second, stuttering automata are used. A *stuttering automaton* is one in which repetition of events is considered a single event. Stuttering is important when multiple steps of the program automaton correspond to a single step of the specification automaton. For simplicity we have not considered this issue here. In [AL89], translations between the method of [AL88] and our method (originally described in [KS89]) were first outlined.

## 6 Summary

We have described a verification method based on our ND progress measure for nondeterministic automata. Unlike previous complete methods, ours is direct in the sense that it requires modifying neither the program nor the specification. Progress measures also have allowed us to classify the applicability of previous methods that do not depend on program transformations. According to whether  $A_P$  is historical or not, or whether  $A_S$  is deterministic or safety, the progress measure indicated below constitutes a sound and complete verification method for showing  $L(A_P) \subseteq L(A_S)$ :

	$A_S$	
$A_P$	<i>deterministic</i>	<i>safety</i>
<i>historical</i>	refinement	prophecy
<i>nondeterministic</i>	history	ND

Unfortunately, the most powerful progress measure, the ND measure, is rather complex since it maps program states to sets of sets of specification states. This complexity is inherent in the verification problem. No method based on just mapping program states to sets of specification states can be both sound and complete for nondeterministic automata.

## Acknowledgments

We would like to thank M. Abadi, B. Alpern, D. Kozen, L. Lamport, and A. Zwarico for their very helpful comments on earlier versions of this article.

## References

- [AL88] M. Abadi and L. Lamport. The existence of refinement mappings. In *Proc. 2. Symp. on Logic in Computer Science*. IEEE, 1988. To appear in *Theoretical Computer Science*.
- [AL89] M. Abadi and L. Lamport. Private communication, November 1989.
- [Arn83] A. Arnold. Topological characterizations of infinite behaviors of transition systems. In *Proc. 10th Col. Automata, Languages and Programming*, pages 490–510. LNCS, Vol. 154, Springer-Verlag, 1983.
- [AS85] B. Alpern and F.B. Schneider. Defining liveness. *Information Processing Letters*, 21:181–185, Oct. 1985.

- [Kla90] Nils Klarlund. *Progress Measures and Finite Arguments for Infinite Computations*. PhD thesis, TR-1153, Cornell University, August 1990.
- [KS89] N. Klarlund and F.B. Schneider. Verifying safety properties using infinite-state automata. Technical Report TR-1036, Cornell University, 1989.
- [Lam83] L. Lamport. Specifying concurrent program modules. *ACM Transactions on Programming Languages and Systems*, 5(2):190-222, 1983.
- [LT87] N. Lynch and M. Tuttle. Hierarchical correctness proof for distributed algorithms. In *Proc. Sixth Symp. on the Principles of Distributed Computing*, pages 137-151. ACM, 1987.
- [Mer89] M. Merritt. Completeness theorems for automata. Technical report, AT&T Bell Laboratories, 1989.
- [Par81] D. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proc. 5th GI conference*, pages 167-183, 1981. In *Lecture Notes in Computer Science* 104.
- [Rog67] Hartley Rogers, Jr. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill Book Company, 1967.
- [Sis89a] A.P. Sistla. A complete proof system for proving correctness of nondeterministic safety specifications. Technical report, Computer and Intelligent Systems Laboratory, GTE Laboratories Inc., 1989.
- [Sis89b] A.P. Sistla. On verifying that a concurrent program satisfies a nondeterministic specification. *Information Processing Letters*, 32(1):17-24, July 1989.
- [Sta88] E. Stark. Proving entailment between conceptual state specifications. *Theoretical Computer Science*, 56:135-154, 1988.
- [Var87] M. Vardi. Verification of concurrent programs: The automata-theoretic framework. In *Proc. Symp. on Logic in Computer Science*. IEEE, 1987.