Information Processing Letters 43 (1992) 135–141
North-Holland

14 September 1992

Sooner is safer than later

Thomas A. Henzinger

Computer Science Department, Cornell University, Ithaca, NY 14853, USA

Communicated by F.B. Schneider
Received 29 May 1991
Revised 11 June 1992

Abstract

Henzinger, T.A., Sooner is safer than later, Information Processing Letters (43) (1992) 135–141.

It has been observed repeatedly that the standard safety-liveness classification for properties of reactive systems does not fit for real-time properties. This is because the implicit "liveness" of time shifts the spectrum towards the safety side. While, for example, response—that "something good" will happen eventually—is a classical liveness property, bounded response—that "something good" will happen soon, within a certain amount of time—has many characteristics of safety. We account for this phenomenon formally by defining safety and liveness *relative* to a given condition, such as the progress of time.

Keywords: Safety, liveness, real time, topology, concurrency, semantics

1. Safety, liveness, and operationality

The behavior of a discrete reactive system can be described as an infinite string

$\sigma: \quad \sigma_0\ \sigma_1\ \sigma_2\ \sigma_3\ \sigma_4 \ldots$

over an alphabet Σ, which represents the states of the system. A *property* Π is a subset of Σ^ω, the set of all infinite strings over Σ; a reactive system has property Π iff all of its possible behaviors are contained in Π.

It is useful to classify properties of reactive systems into two categories, because they require qualitatively different means for their specification and verification [13]:

- A *safety* property stipulates that "nothing bad" will happen, ever, during the execution of a system. If "something bad" were to happen during the execution, it would have to happen within a finite number of states. Thus we can formalize safety as follows:

  $\Pi \subseteq \Sigma^\omega$ is a *safety* property iff for all $\sigma \in \Sigma^\omega$, whenever every finite prefix of σ can be extended to a string in Π, then $\sigma \in \Pi$ [3].

- A *liveness* property stipulates that "something good" will happen, eventually, during the execution of a system. Even if "nothing good" were to happen within a finite prefix of the execution, "something good" could still happen in a later state; only if an irremediably bad situation is reached within a finite number of states, "nothing good" will happen during the entire execution. Thus we can formalize liveness as follows:

  $\Pi \subseteq \Sigma^\omega$ is a *liveness* property iff every finite prefix of a string in Σ^ω can be extended to a string in Π [4].

There is a natural topology on Σ^ω – the Cantor topology – in which the safety properties are

Correspondence to: T.A. Henzinger, Department of Computer Science, 4130 Upson Hall, Cornell University, Ithaca, NY 14853-7501, USA.

exactly the closed sets, and the liveness properties are exactly the dense sets. It follows that (1) only Σ^ω itself is both a safety and a liveness property and (2) every property is the intersection of a safety property and a liveness property. Hence any correctness proof for a reactive system can be decomposed into a safety part and a liveness part.

Let us briefly sketch the standard topological construction for showing observation (2) [4], because we shall generalize it later. The construction is well-known to prove a strong formulation of the observation that is based on the following definition. We say that a safety property Π_S and a liveness property Π_L specify the property $\Pi = \Pi_S \cap \Pi_L$ *congruously* iff every finite prefix of a string in Π_S can be extended to a string in Π. In other words, the safety part of a congruous specification is complete: the liveness part does not preclude any safe prefixes. A congruous pair (Π_S, Π_L) is called *machine closed* in [1], *feasible* in [8], and Π_L is called *live with respect to* Π_S in [9].

Theorem 1. (Existence of congruous specifications.) *Every property has a congruous specification.*

Proof (sketch). Since safety properties are closed under intersection, we can define the *closure* $\overline{\Pi}$ of $\Pi \subseteq \Sigma^\omega$ as the smallest safety property containing Π. Given a property Π, let Π_S be $\overline{\Pi}$. For Π_L take the complement of $\Pi_S - \Pi$. Then (Π_S, Π_L) specifies Π congruously. □

Congruous specifications are *operational*: a machine that incrementally generates safe execution sequences will never reach an irremedial situation from which the liveness conditions cannot be satisfied. On the other hand, a machine trying to execute an incongruous specification without look-ahead may "paint itself into a corner" from which no legal continuation is possible [8]. Examples of congruous specifications are fair transition systems; examples of formalisms that admit incongruous specifications are temporal logic and finite automata (see [17] and [19] for surveys of these formalisms).

2. Relative safety and liveness

Instead of looking at all strings in Σ^ω, it is often useful to have a concept of safety and liveness under the assumption that, a priori, only a certain subset $\Psi \subseteq \Sigma^\omega$ of strings are possible behaviors of a system. We call these notions safety and liveness *relative* to the property Ψ:

- $\Pi \subseteq \Psi$ is a *safety* property *relative to* $\Psi \subseteq \Sigma^\omega$ iff for all $\sigma \in \Psi$, whenever every finite prefix of σ can be extended to a string in Π, then $\sigma \in \Pi$.
- $\Pi \subseteq \Psi$ is a *liveness* property *relative to* $\Psi \subseteq \Sigma^\omega$ iff every finite prefix of a string in Ψ can be extended to a string in Π.

Thus unconditional safety and liveness are safety and liveness relative to Σ^ω.

The Cantor topology on Σ^ω induces a topological subspace on $\Psi \subseteq \Sigma^\omega$, which is called the *relativization* of the Σ^ω-topology to Ψ [11]. We show that the properties that are safe relative to Ψ are exactly the closed sets of the relative topology, and the properties that are live relative to Ψ are exactly the dense sets of the relative topology.

Proposition 2. (Relative safety.) *$\Pi \subseteq \Psi$ is a safety property relative to $\Psi \subseteq \Sigma^\omega$ iff $\overline{\Pi} \cap \Psi \subseteq \Pi$.*

Proposition 3. (Relative liveness.) *$\Pi \subseteq \Psi$ is a liveness property relative to $\Psi \subseteq \Sigma^\omega$ iff $\Psi \subseteq \overline{\Pi}$.*

Proof of Propositions 2 and 3. First observe that a string $\sigma \in \Sigma^\omega$ is in the closure of a property $\Pi \subseteq \Sigma^\omega$ (that is, $\sigma \in \overline{\Pi}$) iff every finite prefix of σ can be extended to a string in Π. Then apply this observation to the definitions of relative safety and relative liveness. □

It follows that Π is safe relative to Ψ iff $\Pi = \Pi_S \cap \Psi$ for some unconditional safety property Π_S. In particular, if the property $\Pi = \Pi_S \cap \Pi_L$ is specified by a safety property Π_S and a liveness property Π_L, then Π is safe relative to Π_L. Furthermore, if the specification (Π_S, Π_L) is congruous, then Π is live relative to Π_S.

It is convenient to extend the notions of safety and liveness relative to a property Ψ to properties that are not necessarily subsets of Ψ: we say that $\Pi \subseteq \Sigma^{\omega}$ is a safety (liveness) property relative to $\Psi \subseteq \Sigma^{\omega}$ iff $\Pi \cap \Psi$ is safe (live) relative to Ψ. Clearly, unconditional safety properties are, in this sense, safe relative to any property Ψ. More generally:

Proposition 4. (Downward preservation of safety.) *Suppose that $\Psi_1 \subseteq \Psi_2$. If Π is a safety property relative to Ψ_2, then it is also a safety property relative to Ψ_1.*

Proof. Let $\Psi_1 \subseteq \Psi_2$. First observe that the closure operator is monotonic; that is, $\Pi \subseteq \Psi$ implies $\overline{\Pi} \subseteq \overline{\Psi}$ for all $\Pi, \Psi \in \Sigma^{\omega}$. In particular, we have $\overline{\Pi \cap \Psi_1} \subseteq \overline{\Pi \cap \Psi_2}$.

By Proposition 2, we may assume that

$$\overline{(\Pi \cap \Psi_2)} \cap \Psi_2 \subseteq \Pi \cap \Psi_2$$

and need to show that, then,

$$\overline{(\Pi \cap \Psi_1)} \cap \Psi_1 \subseteq \Pi \cap \Psi_1.$$

The derivation is simple. □

The converse of Proposition 4 holds only in a very restricted case:

Proposition 5. (Upward preservation of safety.) *Suppose that $\Pi \subseteq \Psi_1 \subseteq \Psi_2$. If Π is a safety property relative to Ψ_1 and Ψ_1 is a safety property relative to Ψ_2, then Π is a safety property relative to Ψ_2.*

Proof. Again, use Proposition 2 and the monotonicity of the closure operator. □

In general, properties become "safer" when they are viewed relative to stronger (i.e., more restrictive) properties: a property that is not an unconditional safety property may be safe relative to another property.

Indeed, there are natural properties relative to which all properties are safety properties. Let $z \in \Sigma$ be a symbol that signals the termination of a reactive system Let $\Psi_{fin} \subseteq \Sigma^{\omega}$ contain all infinite strings that are of the form that a finite prefix over the alphabet $\Sigma - \{z\}$ is followed by an infinite suffix over the alphabet $\{z\}$; that is, the property Ψ_{fin} of a reactive system asserts that "the system terminates." It is not difficult to see that every property Π is safe relative to Ψ_{fin} (which itself is neither a safety property nor a liveness property). For suppose that every finite prefix of a string $\sigma \in \Psi_{fin}$ can be extended to a string in $\Pi \cap \Psi_{fin}$. Then we can choose a sufficiently long prefix of σ that contains a z; any extension of this prefix must be σ itself, which implies that $\sigma \in \Pi \cap \Psi_{fin}$.

In other words, under the assumption that all systems under consideration terminate, every property of a reactive system is a safety property. In the final section, we will present a less stringent assumption about reactive systems that, nonetheless, shifts interesting properties "towards safety".

3. Operationality and verification of relative specifications

We say that a pair (Π_S, Π_L) specifies the property $\Pi \subseteq \Psi$ *congruously relative to* $\Psi \subseteq \Sigma^{\omega}$ iff

(1) $\Pi = \Pi_S \cap \Pi_L \cap \Psi$,

(2) Π_S is safe relatively to Ψ and Π_L is live relative to Ψ, and

(3) every finite string that is both a prefix of a string in Π_S and a prefix of a string in Ψ can be extended to a string in Π.

Thus a specification is unconditionally congruous iff it is congruous relative to Σ^{ω}. The following theorem generalizes the main result about the unconditional safety-liveness classification (Theorem 1).

Theorem 6. (Existence of relatively congruous specifications.) *For all $\Psi \subseteq \Sigma^{\omega}$, every property $\Pi \subseteq \Psi$ has a specification that is congruous relative to Ψ.*

Proof. Let $\Pi_S = \overline{\Pi}$ and $\Pi_L = \neg((\Pi_S \cap \Psi) - \Pi)$; then Π_S is unconditionally safe. Alternatively, let $\Pi_S = \overline{\Pi} \cap \Psi$ and $\Pi_L = \neg(\Pi_S - \Pi)$; then $\Pi_S \subseteq \Psi$.

We show that (Π_S, Π_L) specifies Π congruously relative to Ψ in either case.

(1) It is not hard to check that $\Pi = \Pi_S \cap \Pi_L \cap \Psi$.

(2) The unconditional safety property $\Pi_S = \overline{\Pi}$ is safe relative to Ψ, and so is $\Pi_S = \overline{\Pi} \cap \Psi$. To see that Π_L is live relative to Ψ, by Proposition 3 it suffices to show that

$$\Psi \subseteq \overline{\neg((\overline{\Pi} \cap \Psi) - \Pi) \cap \Psi}.$$

Since $\Pi \subseteq \Psi$, this condition is equivalent to

$$\Psi \subseteq \overline{\Pi \cup (\Psi - \overline{\Pi})}.$$

We can derive both

$$\overline{\Pi} \cap \Psi \subseteq \overline{\Pi \cup (\Psi - \overline{\Pi})}$$

and

$$\neg\,\overline{\Pi} \cap \Psi \subseteq \overline{\Pi \cup (\Psi - \overline{\Pi})},$$

using the monotonicity of the closure operator.

(3) Since $\Pi_S \subseteq \overline{\Pi}$, every finite prefix of a string in Π_S can be extended to a string in Π. □

Our definition of relative congruity ensures again operationality: a machine that incrementally generates prefixes in Π_S that are also prefixes of Ψ will never reach an irremedial situation from which the liveness conditions of $\Pi_L \cap \Psi$ cannot be satisfied. Next we shall see that the relative congruity of system descriptions is desirable also from a verification point of view.

The notion of relative safety has ramifications for both the specification and the verification of reactive systems. Suppose that a property Π is safe relative to an assumption Ψ. We can take advantage of this fact in two ways:

1. The property Π can be *specified* by an unconditional safety property, namely, $\overline{\Pi \cap \Psi}$. This is because $\overline{(\Pi \cap \Psi)} \cap \Psi = \Pi \cap \Psi$ by Proposition 2.
2. The property Π can be *verified* by safety reasoning. Suppose that the possible behaviors of a reactive system $\hat{\Pi}$ are given by the congruous pair $(\hat{\Pi}_S, \hat{\Pi}_L)$. In order to verify that the system $\hat{\Pi}$ has the property Π, it suffices to show that the safety component $\hat{\Pi}_S$ of the system $\hat{\Pi}$ satisfies the safety property $\overline{\Pi \cap \Psi}$.

This verification strategy is justified by the following theorem; the strategy is complete, provided that (1) we may also use the safety component Ψ_S of the assumption Ψ in the verification process, and (2) the system specification $(\hat{\Pi}_S, \hat{\Pi}_L)$ is congruous relative to the assumption Ψ.

Theorem 7. (Verification of relative safety properties.) *Let (Ψ_S, Ψ_L) be a congruous specification of $\Psi \subseteq \Sigma^\omega$, let $(\hat{\Pi}_S, \hat{\Pi}_L)$ be a specification of $\hat{\Pi} \subseteq \Psi$ that is congruous relative to Ψ, and let $\Pi \subseteq \Sigma^\omega$ be safe relative to Ψ. Then $\hat{\Pi} \subseteq \Pi$ iff $\hat{\Pi}_S \cap \Psi_S \subseteq \overline{\Pi \cap \Psi}$.*

Proof. First, assume that $\hat{\Pi}_S \cap \Psi_S \subseteq \overline{\Pi \cap \Psi}$. Then

$$\hat{\Pi} \subseteq \overline{(\Pi \cap \Psi)} \cap \Psi$$

and, since Π is safe relative to Ψ, we have

$$\overline{(\Pi \cap \Psi)} \cap \Psi \subseteq \Pi$$

by Proposition 2. By transitivity, $\hat{\Pi} \subseteq \Pi$ follows.

Second, assume that $\hat{\Pi} \subseteq \Pi$. Since the pair (Ψ_S, Ψ_L) is congruous, $\hat{\Pi}_S \cap \Psi_S \subseteq \hat{\Pi}_S \cap \overline{\Psi}$. As the specification $(\hat{\Pi}_S, \hat{\Pi}_L)$ is congruous relative to Ψ, we have

$$\hat{\Pi}_S \cap \overline{\Psi} \subseteq \overline{\hat{\Pi}}.$$

By our assumption, $\hat{\Pi} \subseteq \Pi \cap \Psi$ and, by the monotonicity of the closure operator,

$$\overline{\hat{\Pi}} \subseteq \overline{\Pi \cap \Psi}.$$

By transitivity, $\hat{\Pi}_S \cap \Psi_S \subseteq \overline{\Pi \cap \Psi}$ as desired. □

Now let us illustrate the application of this result with the termination assumption Ψ_{fin}. Consider the liveness property $\Pi_{\diamond p}$ that contains all infinite strings with at least one occurrence of the symbol p. Since every property is safe relative to Ψ_{fin}, so is in particular $\Pi_{\diamond p}$. Thus Theorem 7 tells us that, over terminating systems, $\Pi_{\diamond p}$ can be specified and verified as the safety property $\overline{\Pi_{\diamond p} \cap \Psi_{fin}}$. This property consists of all infinite strings such that (1) each occurrence of z is followed by a z and (2) there is an occurrence of p before the first occurrence of z (including all strings that contain neither a p nor a z). Note that, indeed, if all runs of a system satisfy the safety property $\overline{\Pi_{\diamond p} \cap \Psi_{fin}}$, then all terminating runs of the system satisfy the desired property $\Pi_{\diamond p}$.

4. Real-time safety and liveness

The behavior of a discrete real-time system can be described by an infinite sequence of pairs

$$\rho: \quad (\sigma_0, \tau_0) \to (\sigma_1, \tau_1) \to (\sigma_2, \tau_2) \to \cdots$$

of states $\sigma_i \in \Sigma$, for $i \geqslant 0$, and corresponding times $\tau_i \in \mathscr{T}$. While we do not commit to any particular time domain $\mathscr{T}$, we assume that there is a real-valued distance function d on $\mathscr{T}^2$ with $d(x, x) = 0$ for all $x \in \mathscr{T}$. The sequence $\rho = (\sigma, \tau)$ is called a *timed state sequence*.

A *real-time property* Π is a subset of Ψ_{all}, the set of all timed state sequences. It is straightforward to extend the definitions of unconditional and relative safety and liveness to real-time properties. All results of the previous sections carry over. In particular, any trivial one-element time domain yields a model that is isomorphic to the original untimed setup.

Different models of time and computation put vastly different requirements on the time component τ of legal behaviors $\rho = (\sigma, \tau)$ of a real-time system. For instance:

- *Interval* models of time associate with every state its duration over time, while *clock* models stamp observations of the system state with time instants. Invervals of the real line are a suitable time domain for the former model, points for the latter.
- *Analog-clock* models of time record the exact time of every state, while *digital-clock* models measure the time of a state only with finite precision. The reals are a suitable time domain for the former model, the integers for the latter.
- In *synchronous* models of computation, all concurrent activity happens in lock-step, while *asynchronous* (*interleaving*) models sequentialize simultaneous actions nondeterministically. Strictly monotonic time is appropriate for the former model, while instantaneous actions are required by the latter.

(See [7] for a survey of various models of time that have been proposed for the verification of real-time systems.)

Given a particular choice of model, we consider, by definition, only a subset $\Psi \subseteq \Psi_{all}$ of timed state sequences as possible behaviors of a real-time system; that is, the specification of property Π really defines $\Pi \cap \Psi$. Thus we can specify Π by describing any property Π' with $\Pi' \cap \Psi = \Pi \cap \Psi$, possibly even using a safety property Π' to specify a liveness property Π. Precisely this phenomenon is captured formally by the concept of safety and liveness relative to the *timing assumption* Ψ.

There are two particularly important model-independent timing assumptions:

1. All "reasonable" models of time require that time must not decrease. A timed state sequence (σ, τ) is called *monotonic* iff time increases (weakly) monotonically:

   $$d(\tau_i, \tau_j) \leqslant d(\tau_i, \tau_k) \quad \text{for all } 0 \leqslant i \leqslant j \leqslant k.$$

   The set $\Psi_{mon} \subseteq \Psi_{all}$ of all monotonic timed state sequences is clearly a safety property.
2. The behavior of a continuous system that may change its state infinitely often between any two points in time cannot be modeled adequately by an ω-sequence of states. Thus, given our choice of a timed state sequence semantics, we may "reasonably" demand that time diverges. A timed state sequence (σ, τ) is called *divergent* iff time eventually proceeds to any point:

   $$\text{for all } i \geqslant 0 \text{ and } x \in \mathscr{T}, \text{ there is some } j \geqslant i \text{ such that } d(\tau_i, \tau_j) \geqslant d(\tau_i, x).$$

   It can be checked that the set $\Psi_{div} \subseteq \Psi_{all}$ of all divergent timed state sequences is a liveness property.

It follows that typical timing assumptions are subsets of $\Psi_{time} = \Psi_{mon} \cap \Psi_{div}$.

Therefore we are especially interested in safety, liveness and operationality *relative to* *monotonic divergence* (i.e., relative to Ψ_{time}). The class of properties that are safe relative to monotonic divergence includes many important real-time properties that are unconditional liveness properties; that is, all the liveness they stipulate is subsumed by the divergence of time.

Bounded response is the standard example of a real-time property that is unconditionally live and becomes safe under strong enough timing assumptions [10,14,15,18]. Let $p,q \in \Sigma$ and let δ be a nonnegative real. The *bounded-response* property $\Pi^{\delta}_{p \mapsto q}$ contains a timed state sequence (σ, τ) iff for all $i \geqslant 0$, whenever $\sigma_i = p$, then $\sigma_j = q$ and $d(\tau_i, \tau_j) \leqslant \delta$ for some $j \geqslant i$; that is, every p-state is followed by a q-state within time δ. Since any finite prefix of a timed state sequence containing (p, x) can be extended with the pair (q, x), the property $\Pi^{\delta}_{p \mapsto q}$ is an unconditional liveness property.

Now let us consider $\Pi^{\delta}_{p \mapsto q}$ relative to monotonicity, and then relative to monotonic divergence. Provided that p and q are different states, $\Pi^{\delta}_{p \mapsto q}$ is not safe relative to Ψ_{mon}, because it contains all monotonic timed state sequences of the form

$$(p, x) \to \cdots \to (p, x) \to (q, x) \to \cdots,$$

without containing the monotonic sequence

$$(p, x) \to (p, x) \to (p, x) \to \cdots.$$

Provided that there are two times $x,y \in \mathscr{T}$ with $d(x, y) > \delta$, the property $\Pi^{\delta}_{p \mapsto q}$ is not live relative to Ψ_{mon} either, because the finite prefix

$$(p, x) \to (p, y)$$

cannot be extended to a monotonic sequence in $\Pi^{\delta}_{p \mapsto q}$. Finally, suppose that for all $x \in \mathscr{T}$ there is some $y \in \mathscr{T}$ such that $d(x, y) > \delta$. Then it is not hard to check that the bounded-response property $\Pi^{\delta}_{p \mapsto q}$ is a safety property relative to monotonic divergence; the "bad thing" that is not supposed to happen is that, after a p-state, δ time units pass without a q-state occurring.

Specifications that are congruous relative to monotonic divergence are called *nonZeno* [2], because they cannot define Zeno machines that force time to converge. Real-time transition systems [10] and extended state machines [16] are examples of specifications that are nonZeno, and thus operational descriptions of real-time systems. So are the timed automata of [15], which specify only properties that are safe relative to monotonic divergence. On the other hand, real-time temporal logics such as [6,12,16] and the timed automata of [5] permit, relative to monotonic divergence, incongruous specifications of real-time systems. A machine trying to execute such a specification without look-ahead may find itself in a situation from which time cannot diverge without violating the specification.

For nonZeno specifications we can apply Theorem 7. If a system is given congruously relative to monotonic divergence, then the bounded-response property $\Pi^{\delta}_{p \mapsto q}$ can be verified as the safety property

$$\overline{\Pi^{\delta}_{p \mapsto q} \cap \Psi_{time}}$$

[10]. This property states that (1) time does not decrease and (2) whenever $\sigma_i = p$, then either $\sigma_j = q$ and $d(\tau_i, \tau_j) \leqslant \delta$ for some $j \geqslant i$ or $d(\tau_i, \tau_j) \leqslant \delta$ for all $j \geqslant i$.

Acknowledgment

The author thanks Martín Abadi, Rajeev Alur, David Dill, Leslie Lamport, Zohar Manna, Amir Pnueli, Fred Schneider, and two anonymous referees for many valuable suggestions and improvements.

References

[1] M. Abadi and L. Lamport, The existence of refinement mappings, In: *Proc. Third Ann. Symp. on Logic in Computer Science* (IEEE Computer Society Press, Silver Spring, MD, 1988) 165–175.

[2] M. Abadi and L. Lamport, An old-fashioned recipe for real time, in: *Proc. REX Workshop*, Real-time: Theory in Practice, Lecture Notes in Computer Science **600** (Springer, Berlin, 1992).

[3] B. Alpern, A.J. Demers and F.B. Schneider, Safety without stuttering, *Inform. Process. Lett.* **23** (1986) 177–180.

[4] B. Alpern and F.B. Schneider, Defining liveness, *Inform. Process. Lett.* **21** (1985) 181–185.

[5] R. Alur and D.L. Dill, Automata for modeling real-time systems, in: *Proc. 17th Internat. Coll. on Automata, Languages, and Programming*, Lecture Notes in Computer Science **443** (Springer, Berlin, 1990) 322–335.

[6] R. Alur and T.A. Henzinger, A really temporal logic, in: *Proc. 30th Ann. IEEE Symp. on Foundations of Computer Science* (1989) 164–169.

[7] R. Alur and T.A. Henzinger, Logics and models of real time: a survey, In: *Proceedings of the REX Workshop*, Real-time: Theory in Practice, Lecture Notes in Computer Science **600** (Springer, Berlin, 192).

[8] K.R. Apt, N. Francez and S. Katz, Appraising fairness in languages for distributed programming, *Distributed Comput.* **2** (1988) 226–241.

[9] F. Dederichs and R. Weber, Safety and liveness from a methodological point of view, *Inform. Process. Lett.* **36** (1) (1990) 25–30.

[10] T.A. Henzinger, Z. Manna and A. Pnueli, Temporal proof methodologies for real-time systems, in: *Proc. 18th Ann. ACM Symp. on Principles of Programming Languages* (ACM Press, New York, 1991) 353–366.

[11] J.L. Kelley, *General Topology* (Springer, Berlin, 1955).

[12] R. Koymans, Specifying real-time properties with metric temporal logic, *Real-time Systems* **2** (1990) 255–299.

[13] L. Lamport, Proving the correctness of multiprocess programs, *IEEE Trans. Software Engineering* **3** (1977) 125–143.

[14] L. Lamport, The temporal logic of actions, Tech. Rept., DEC Systems Research Center, February 1991.

[15] N.A. Lynch and H. Attiya, Using mappings to prove timing properties, in: *Proc. Ninth Ann. ACM Symp. on Principles of Distributed Computing* (ACM Press, New York, 1990) 265–280.

[16] J.S. Ostroff, *Temporal Logic of Real-time Systems* (Research Studies Press, 1990).

[17] A. Pnueli, Applications of temporal logic to the specification and verification of reactive systems: a survey of current trends, in: *Current Trends in Concurrency*, Lecture Notes in Computer Science **224** (Springer, Berlin, 1986) 510–584.

[18] F.B. Schneider, Private communication, February 1991.

[19] W. Thomas, Automata on infinite objects, in: Jan van Leeuwen, ed., *Handbook of Theoretical Computer Science, Vol. B* (Elsevier, Amsterdam and MIT Press, Cambridge, MA, 1990) 133–191.