SOONER IS SAFER THAN LATER

19970523 129

STANFORD UNIV., CA

28 MAY 91

U.S. DEPARTMENT OF COMMERCE
National Technical Information Service
Sooner is Safer Than Later

by

Thomas A. Henzinger

Department of Computer Science
Stanford University
Stanford, California 94305
Abstract. It has been repeatedly observed that the standard safety-liveness classification of 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.
Sooner is Safer than Later*

Thomas A. Henzinger
Department of Computer Science
Stanford University
May 28, 1991

Abstract. It has been repeatedly observed that the standard safety-liveness classification of 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

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

over an alphabet $\Sigma$, which represents the states of the system. A property $\Pi$ is a subset of $\Sigma^\omega$, the set of all infinite strings over $\Sigma$; a reactive system has property $\Pi$ iff all of its possible behaviors are contained in $\Pi$.

It is useful to classify properties of reactive systems into two categories, because they require fundamentally different means for their specification and verification [Lam77]:

- 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:

\[ \Pi \subseteq \Sigma^\omega \text{ is a safety property iff for all } \sigma \in \Sigma^\omega, \text{ whenever every finite prefix of } \sigma \text{ can be extended to a string in } \Pi, \text{ then } \sigma \in \Pi \] [ADS86].

- A liveness property stipulates that "something good" will happen, eventually, during the execution of a system. If "nothing good" were to happen during the execution, an irremediable situation would have to be reached within a finite number of states. Thus we can formalize liveness as follows:

\[ \Pi \subseteq \Sigma^\omega \text{ is a liveness property iff every finite prefix of a string in } \Sigma^\omega \text{ can be extended to a string in } \Pi \text{ [AS85].} \]

There is a natural topology on \( \Sigma^\omega \) in which the safety properties are exactly the closed sets, and the liveness properties are exactly the dense sets. It follows immediately that only \( \Sigma^\omega \) itself is both a safety and a liveness property.

We say that a safety property \( \Pi_S \) and a liveness property \( \Pi_L \) specify the property \( \Pi = \Pi_S \cap \Pi_L \) congruously iff every finite prefix of a string in \( \Pi_S \) can be extended to a string in \( \Pi \). In other words, the safety part of a congruous specification is complete: the liveness part does not preclude any safe prefixes. A congruous pair \( (\Pi_S, \Pi_L) \) is called machine closed in [AL88], feasible in [AFK88], and \( \Pi_L \) is called live with respect to \( \Pi_S \) in [DW90].

In [AS85] it is shown that every property is the intersection of a safety property and a liveness property. It is well-known that the construction given there actually proves the following stronger result.

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

**Proof sketch of Theorem 1** 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 \( \Pi \). Given a property \( \Pi \), let \( \Pi_S \) be \( \overline{\Pi} \). For \( \Pi_L \) take the complement of \( \Pi_S - \Pi \). Then \((\Pi_S, \Pi_L)\) specifies \( \Pi \) 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 [AFK88]. Examples of congruous specifications are fair transition systems [Pnu86]; examples of formalisms that admit incongruous specifications are temporal logic [Pnu77] and finite automata [Tho90].
2 Relative Safety and Liveness

Instead of looking at all strings in $\Sigma^\omega$, 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 this notion safety and liveness relative to the property $\Psi$:

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

Thus unconditional safety and liveness are safety and liveness relative to $\Sigma^\omega$.

The natural topology on $\Sigma^\omega$ induces a topological subspace on $\Psi \subseteq \Sigma^\omega$, which is called the relativization of the $\Sigma^\omega$ topology to $\Psi$ [Kel55]. We show that the properties that are safe relative to $\Psi$ are exactly the closed sets of the relative topology, and the properties that are live relative to $\Psi$ are exactly the dense sets of the relative topology.

**Proposition 1 (Relative safety)** $\Pi \subseteq \Psi$ is a safety property relative to $\Psi \subseteq \Sigma^\omega$ iff $\Pi \cap \Psi \subseteq \Pi$.

**Proposition 2 (Relative liveness)** $\Pi \subseteq \Psi$ is a liveness property relative to $\Psi \subseteq \Sigma^\omega$ iff $\Psi \subseteq \Pi$.

**Proof of Propositions 1 and 2** First observe that a string $\sigma \in \Sigma^\omega$ is in the closure of a property $\Pi \subseteq \Sigma^\omega$ (that is, $\sigma \in \Pi$) iff every finite prefix of $\sigma$ can be extended to a string in $\Pi$. Then apply this observation to the definitions of relative safety and relative liveness. ■

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

It is convenient to extend the notions of safety and liveness relative to a property $\Psi$ to properties that are not necessarily subsets of $\Psi$: 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 $\Psi$. Clearly, unconditional safety properties are, in this sense, safe relative to any property $\Psi$. More generally:

**Proposition 3 (Downward preservation of safety)** Suppose that $\Psi_1 \subseteq \Psi_2$. If $\Pi$ is a safety property relative to $\Psi_2$, then it is also a safety property relative to $\Psi_1$. 

3
Proof of Proposition 3 Let \( \Psi_1 \subseteq \Psi_2 \). First observe that the closure operator is monotonic; that is, \( \Pi \subseteq \Psi \) implies \( \Pi \subseteq \overline{\Psi} \) for all \( \Pi, \Psi \in \Sigma^\omega \). In particular, we have \( \Pi \cap \overline{\Psi_1} \subseteq \Pi \cap \overline{\Psi_2} \).

By Proposition 1, we may assume that
\[
(\Pi \cap \overline{\Psi_2}) \cap \Psi_2 \subseteq \Pi \cap \Psi_2
\]
and need to show that, then,
\[
(\Pi \cap \overline{\Psi_1}) \cap \Psi_1 \subseteq \Pi \cap \Psi_1.
\]
The derivation is simple. ■

The converse of Proposition 3 holds only in a very restricted case:

Proposition 4 (Upward preservation of safety) Suppose that \( \Pi \subseteq \Psi_1 \subseteq \Psi_2 \). If \( \Pi \) is a safety property relative to \( \Psi_1 \) and \( \Psi_1 \) is a safety property relative to \( \Psi_2 \), then \( \Pi \) is a safety property relative to \( \Psi_2 \).

Proof of Proposition 4 Again, use Proposition 1 and the monotonicity of the closure operator. ■

In general, properties become "safer" if 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. In the next section, we will give interesting examples of such properties that are shifted "towards safety."

We say that a pair \( (\Pi_S, \Pi_L) \) specifies the property \( \Pi \subseteq \Psi \) 

congruously relative to \( \Psi \subseteq \Sigma^\omega \) if \( \Pi = \Pi_S \cap \Pi_L \cap \Psi \), and \( \Pi_S \) is safe relative to \( \Psi \) and \( \Pi_L \) is live relative to \( \Psi \), and every finite prefix of a string in \( \Pi_S \cap \Psi \) can be extended to a string in \( \Pi \). Thus a specification is unconditionally congruous if it is congruous relative to \( \Sigma^\omega \). The following theorem generalizes the main result about the unconditional safety-liveness classification (Theorem 1).

Theorem 2 (Existence of relatively congruous specifications) For all \( \Psi \subseteq \Sigma^\omega \), every property \( \Pi \subseteq \Psi \) has a specification that is congruous relative to \( \Psi \).

Proof of Theorem 2 Let \( \Pi_S = \Pi \) and \( \Pi_L = \neg((\Pi_S \cap \Psi) - \Pi) \); then \( \Pi_S \) is unconditionally safe. Alternatively, let \( \Pi_S = \Pi \cap \Psi \) and \( \Pi_L = \neg((\Pi_S \cap \Psi) - \Pi) \); then \( \Pi_S \subseteq \Psi \). We show that \( (\Pi_S, \Pi_L) \) specifies \( \Pi \) congruously relative to \( \Psi \) in either case.

It is not hard to see that \( \Pi = \Pi_S \cap \Pi_L \cap \Psi \) and that \( \Pi_S \cap \Psi \subseteq \Pi \) — that is, every finite prefix of a string in \( \Pi_S \cap \Psi \) can be extended to a string in \( \Pi \). Proposition 3 implies that \( \Pi_S = \Pi \), and thus also \( \Pi_S = \overline{\Pi \cap \Psi} \), is safe relative to \( \Psi \).

It remains to be shown that \( \Pi_L \) is live relative to \( \Psi \) or, by Proposition 2, that
\[
\Psi \subseteq \neg((\Pi \cap \Psi) - \Pi) \cap \Psi.
\]
Since \( \Pi \subseteq \Psi \), this condition is equivalent to
\[
\Psi \subseteq \overline{\Pi \cup (\Psi - \Pi)}.
\]
We can derive both
\[
\overline{\Pi \cap \Psi} \subseteq \overline{\Pi \cup (\Psi - \Pi)}
\]
and
\[
\overline{\neg \Pi \cap \Psi} \subseteq \overline{\Pi \cup (\Psi - \Pi)},
\]
using the monotonicity of the closure operator. ■

Note that our definition of relative congruity ensures again operationality:
a machine that incrementally generates prefixes in \( \Pi_s \cap \Psi \) will never reach an
irremediable situation from which the liveness conditions of \( \Pi_L \cap \Psi \) cannot be satisfied.

3 Real-time Safety and Liveness

The behavior of a discrete real-time system can be described by an infinite
sequence of pairs
\[
\rho : \ (\sigma_0, \tau_0) \rightarrow (\sigma_1, \tau_1) \rightarrow (\sigma_2, \tau_2) \rightarrow (\sigma_3, \tau_3) \rightarrow \cdots
\]
of states \( \sigma_i \in \Sigma, \ i \geq 0 \), and corresponding times \( \tau_i \in \mathcal{T} \). While we do not
commit to any particular time domain \( \mathcal{T} \), we assume that there is a real-valued
metric \( d \) on \( \mathcal{T} \). The sequence \( \rho = (\sigma, \tau) \) is called a timed state sequence.

A real-time property \( \Pi \) is a subset of \( \Psi_{all} \), 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 \( \tau \) 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 in-
stants. Intervals 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 [HMP90].

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 a property $\Pi$ really defines $\Pi \cap \Psi$. Thus we can specify $\Pi$ by describing any property $\Pi'$ with $\Pi' \cap \Psi = \Pi \cap \Psi$, possibly even using a safety property $\Pi'$ to specify a liveness property $\Pi \cap \Psi$. Precisely this phenomenon has been captured formally by the concept of safety and liveness relative to the timing assumption $\Psi$.

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 $(\sigma, \tau)$ is called monotonic iff time increases (weakly) monotonically:

$$d(\tau_0, \tau_i) \leq d(\tau_0, \tau_{i+1}) \text{ for all } i \geq 0.$$  

The set $\Psi_{mon} \subseteq \Psi_{all}$ of all monotonic timed state sequences is 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 $\omega$-sequence of states. Thus, given our choice of a timed state sequence semantics, we may "reasonably" demand that time diverges. A timed state sequence $(\sigma, \tau)$ is called divergent iff time eventually progresses beyond any point:

for every $\delta$ in the range of $d$, there is some $i \geq 0$ such that $d(\tau_0, \tau_i) > \delta$.

The set $\Psi_{div} \subseteq \Psi_{all}$ of all divergent timed state sequences is a liveness property.

It follows that most 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 $\Psi_{time}$). 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.
The bounded-response property $\Pi_{p\rightarrow q}^\delta$ contains a timed state sequence $(\sigma, \tau)$ iff for all $i \geq 0$, whenever $\sigma_i = p$, then $\sigma_j = q$ and $d(\tau_i, \tau_j) \leq \delta$ for some $j \geq i$; that is, every $p$ state is followed by a $q$ state within time $\delta$. Clearly, $\Pi_{p\rightarrow q}^\delta$ is an unconditional liveness property.

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

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

without containing the monotonic sequence

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

Provided that there are times $x$ and $y$ with $d(x, y) > \delta$, the property $\Pi_{p\rightarrow q}^\delta$ is not live relative to $\Psi_{\text{mon}}$ either, because the finite prefix

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

cannot be extended to a monotonic sequence in $\Pi_{p\rightarrow q}^\delta$. The bounded-response property $\Pi_{p\rightarrow q}^\delta$ is, however, a safety property relative to monotonic divergence; the "bad thing" that is not supposed to happen is that, after a $p$ state, $\delta$ time units pass without a $q$ state occurring.

Real-time transition systems [HMP91] and extended state machines [Ost90] are examples of specifications that are congruous relative to monotonic divergence, and thus operational descriptions of real-time systems. So are the timed automata of [LA90], which specify only properties that are safe relative to monotonic divergence. On the other hand, real-time temporal logics such as [AH89, Koy90, Ost90] and the timed automata of [AD90] 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 advance without violating the specification.

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

References


