# On Discretization of Delays in Timed Automata and Digital Circuits

Eugene Asarin[1], Oded Maler[2] and Amir Pnueli[3]

[1] Institute for Information Transmission Problems, 19 Bol. Karetnyi per., 101447 Moscow, Russia. **asarin@aha.ru**[†]

[2] VERIMAG, Centre Equation, 2, av. de Vignate, 38610 Gières, France, **maler@imag.fr**

[3] Dept. of Computer Science, Weizmann Inst. Rehovot 76100, Israel, **amir@wisdom.weizmann.ac.il**[‡]

**Abstract.** In this paper we solve the following problem: "given a digital circuit composed of gates whose real-valued delays are in an integer-bounded interval, is there a way to discretize time while preserving the qualitative behavior of the circuit?" This problem is described as open in [BS94]. When "preservation of qualitative behavior" is interpreted in a strict sense, as having all original sequences of events with their original ordering we obtain the following two results:
1) For acyclic (combinatorial) circuits whose inputs change only once, the answer is positive: there is a constant $\delta$, depending on the maximal number of possible events in the circuit, such that if we restrict all events to take place at multiples of $\delta$, we still preserve qualitative behaviors.
2) For cyclic circuits the answer is negative: a simple circuit with three gates can demonstrate a qualitative behavior which cannot be captured by any discretization.
Nevertheless we show that a weaker notion of preservation, similar to that of [HMP92], allows in many cases to verify discretized circuits with $\delta = 1$ such that the verification results are valid in dense time.

## 1 Introduction

The analysis of digital circuits[1] whose components exhibit uncertain delay parameters is a challenging task. A commonly-used model for specifying such systems is the *bi-bounded delay* model where the output of every gate passes through a delay element characterized by some interval $[l, u]$. Roughly speaking, changes at the input port of the delay element are propagated to its output port after some time $t$ taken from the interval $[l, u]$.

[†] The results were obtained while the author was a visiting professor at ENSIMAG, INPG, Grenoble

[‡] The results were obtained while the author was a visiting professor at UJF, Grenoble.

[1] In this paper, we treat digital circuits which we consider to be a well-behaving subset of timed automata. While many of the results can be extended to arbitrary timed automata, we prefer clarity of presentation over generality.

Adding quantitative timing information to a discrete transition system $\mathcal{A}$ amounts to connecting $\mathcal{A}$ to a special system called Time, which is viewed as a transition system with a special structure, namely, a linear order, such that all transitions go "to the right". The composition of $\mathcal{A}$ and Time consists of a system where transitions of $\mathcal{A}$ and time passage transitions are interleaved.

Consider the example in figure 1: Initially we have a two-state automaton which can decide at *any time* to take a *single* transition labeled by $a$, and a time structure annotated with $t$ transitions. Adding timing constraints to $\mathcal{A}$ consists in: 1) annotating the $a$ transition with a condition $T \in [2, 4]$ on the state of Time and 2) adding "idling" transitions to both in order to synchronize: each system takes its real transitions when the other is idling. The product of the two is a system which makes $a$ at some time in $[2, 4]$.

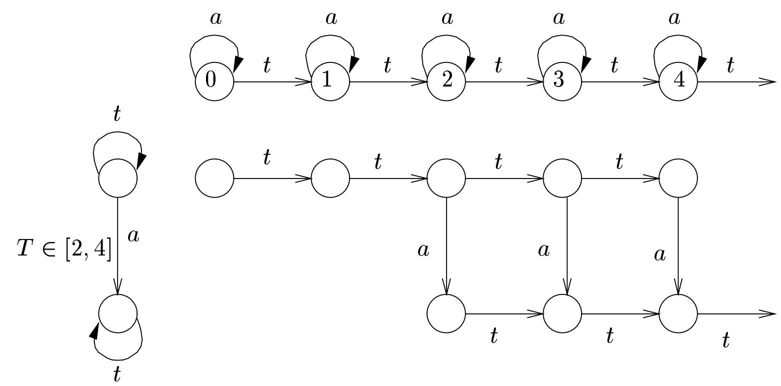



**Fig. 1.** An initialized product of a two-state one-transition automaton with Time.

**Remark**: This picture is intentionally over-simplified, mainly because we do not have two consecutive transitions and the reference time value is always 0. Otherwise we need to introduce an additional unbounded state variable of type Time, memorizing the time of the *last* transition since the beginning. If we had a product of several systems, we would have needed such a variable for each.

Note that we were not very specific about one important property of Time, whether its order is *dense* or *discrete*. One can imagine (if not draw) an analogue of figure 1 where the states of Time are labeled by all the real numbers. The structure of the interaction between Time and $\mathcal{A}$ remains the same. In fact, there is a slight misconception concerning the significance of timed models such as timed automata. Our view is that one should distinguish two aspects of timed models: one is the interaction with a special process such as Time, whose state-space admits order and metric, and the other is the use of continuous dense Time.[2] The latter is not necessarily implied by the former, and the goal of the

[2] We owe some of this insight to [RT97].

paper is to investigate what expressive power (in the sense of modeling) is lost if we refrain from using dense time models, and stay within the familiar (to computer scientists, that is) realm of discrete systems.

Consider again figure 1 with a discrete time interpretation where every $t$ indicates 1 time unit. What does it really mean to move to a coarser time scale of 2 time units? One interpretation is that odd Time states are removed and that $t$ represents 2 units. Alternatively, we can maintain the *same* intrinsic structure of Time but erase all the $a$ transitions from the odd time instants, restricting the product system to take untimed transitions only at even times. In this example the possibility of taking $a$ at $T = 3$ is lost. If we restrict transitions to occur at multiples of 5 we may miss the transition altogether. However, if the granularity of time is at least as fine as the scale of the timing constraints, we are sure not to miss any event in a single-clock (single variable) system. Suppose now that we have two such systems running in parallel, one can make $a$ in $[2, 3]$ and the other can make $b$ at $[3, 4]$ (figure 2). Here, the integer time-scale allows $a$ and $b$ to occur either simultaneously (at 3) or one after the other. By restricting transitions to occur either at odd or even time instants, only one of the above possibilities is allowed.

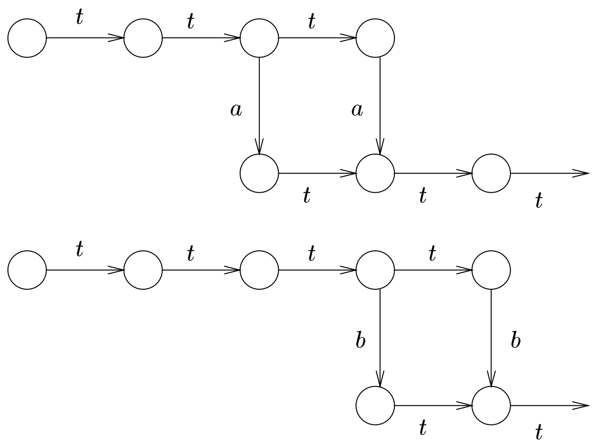



**Fig. 2.** Two one-transition timed automata in parallel.

The passage from dense to discrete time can be viewed in a similar spirit. We can assume a generic dense model of Time, isomorphic to $(\mathbb{R}_+, <)$, and regard every Time discretization as a restriction of the discrete transitions to occur at a certain discrete subset of Time instants. Most of this paper is dedicated to the investigation of the effects of such restrictions on the semantics of automaton and circuit models. More concretely, if $\mathcal{A}$ is the timed automaton associated with a circuit, $L_\mathcal{A}$ is its corresponding set of behaviors (Boolean-valued signals) and $[L_\mathcal{A}]$ is its set of qualitative behaviors (Boolean-valued sequences, obtained from $L_{\mathcal{A}}$ by suppressing the quantitative timing information), we ask under what conditions there exists a discretized semantics $L'_{\mathcal{A}}$ such that $[L'_{\mathcal{A}}] = [L_{\mathcal{A}}]$. Note that the inclusion $[L'_{\mathcal{A}}] \subseteq [L_{\mathcal{A}}]$ follows immediately from $L'_{\mathcal{A}} \subseteq L_{\mathcal{A}}$.

An important related question is under what conditions we have $[L'_{\mathcal{A}} \cap L_{\neg\varphi}] = \emptyset$ iff $[L_{\mathcal{A}} \cap L_{\neg\varphi}] = \emptyset$ where $L_{\neg\varphi}$ is the complement of the specification for a property we wish to establish for the automaton $\mathcal{A}$. When this holds, verification results on the discrete and dense semantics coincide. This is very significant because discrete time models can benefit from many techniques developed for untimed verification. For example, in [ABK$^+$97,BMPY97] we have presented an approach for discrete time verification based on viewing clocks as bounded integer variables, and representing sets of clock valuations using BDDs on the bits of these values. In [BM98] a claim of the form $[L'_{\mathcal{A}} \cap L_{\neg\varphi}] = \emptyset$ has been verified for a discretized system of up to 55 clocks. However, due to the strict inclusion between the semantics, it was not at all evident that the verification results are valid for the dense time model. The results of the current paper show that for the example treated in [BM98], this is indeed the case, i.e. $[L_{\mathcal{A}} \cap L_{\varphi}] = \emptyset$. Similar investigations were carried out in [HMP92] using a different model and a different technique.

The rest of the paper is organized as follows: In section 2 we describe the circuit and delay models that we use. In section 3 we show how the realizability of a qualitative behavior is related to the emptiness of certain polyhedra (possibly infinite-dimensional). These results are used to show that, essentially, acyclic circuits (and automata) admit a discretization, while cyclic circuits (and timed automata in general) do not. In section 4 we show that untimed properties can essentially be verified using discrete time models. Some short contemplations on the potential implications of the results conclude the paper.

## 2 Signals and Circuits

Let $T = \mathbb{R}_+$, $\mathbb{B} = \{0, 1\}$ and $K = \{1, \ldots, k\}$.

**Definition 1 (Boolean Signals).** *A Boolean signal is a left-continuous function $\alpha : T \to \mathbb{B}^k$ admitting a countable*[3] *increasing sequence (which is either finite or diverging) $\mathcal{J}(\alpha) = t_0, t_1, \ldots$ of transition points such that $t_0 = 0$ and $\alpha$ is constant at every interval $(t_j, t_{j+1}]$ and discontinuous at every $t_j$.*

A signal $\alpha$ is ultimately-constant if $\mathcal{J}(\alpha)$ is finite. We denote the set of all Boolean signals by $\mathcal{S}^k$. A *Boolean function* is a function $f : \mathbb{B}^k \to \mathbb{B}$ for some $k \geq 0$. For any such function we define its pointwise extension $f : \mathcal{S}^k \to \mathcal{S}$ in the obvious way, namely $\beta = f(\alpha)$ iff for every $t \in T$, $\beta[t] = f(\alpha[t])$. We call this an *instantaneous* signal function. At the level of modeling in which we are interested, a gate is usually viewed as a composition of an instantaneous function and a *delay* element which holds the output of the function for some time before transmitting it outside. There are several realistic properties of delays which must be accounted for in the model:

[3] And of order type $\leq \omega$ if you want to be pedantic.

1. Positive lower-bound: there is a minimal amount of time that has to elapse between the change of the input and the change in the output.[4]
2. Uncertainty: the exact delay is usually unknown and can only be estimated to be within an interval.
3. Inertia: small fluctuations in the input are ignored by the delay element, and only changes that persist for a minimal duration are propagated to the output.

These considerations are reflected in the following definition:

**Definition 2 (Non-Deterministic Inertial Delay).** *Let $l$ and $u$ be two non-negative numbers such that $l \leq u$. The non-deterministic inertial delay associated with $l, u$ is a function $\Delta_{[l,u]} : \mathbb{B} \times \mathcal{S} \rightarrow 2^{\mathcal{S}}$ defined as: $\beta \in \Delta_{[l,u]}(b, \alpha)$ iff*

1. *$\beta[t] = b$ for every $t \in [0, l)$*
   (Initialization).
2. *For every $t \geq l$, $t \in \mathcal{J}(\beta) \Rightarrow \exists t' \in \mathcal{J}(\alpha) \cap [t-u, t-l]$ such that $\beta[t] = \alpha[t']$ and $(t', t) \cap \mathcal{J}(\alpha) = \emptyset$.*
   (Every change in $\beta$ must be preceded by a persistent change in $\alpha$ which happened at least $l$ time units before).
3. *For every $t \in \mathcal{J}(\alpha)$, $(t, t+u] \cap \mathcal{J}(\alpha) \neq \emptyset \vee [t+l, t+u] \cap \mathcal{J}(\beta) \neq \emptyset$.*
   (Every $u$-persistent change in $\alpha$ must be reflected in $\beta$).

Essentially this means that changes in $\alpha$ that persist less than $l$ are ignored (filtered), those that persist between $l$ and $u$ time can be either filtered or propagated to $\beta$, and those that persist for $u$ or more time *must* be propagated to $\beta$. The distance between a change in $\alpha$ and its corresponding change in $\beta$ must be the interval $[l, u]$. These notions are illustrated in figure 3.

**Remark**: This model is only one among possible alternative models for the delay phenomenon. One could assume, for example, that changes should persist for at least $l_1$ time units, but propagated after $l_2$, $l_2 > l_1$ time. On the other hand, the requirement that an input change persists until its propagation to the output may be relaxed. Incorporating such delay models can be done in the timed automaton framework by adding additional states to the basic automaton. The choice among models depends on the trade-off between model complexity and the faithfulness to the physical reality. Also, we use the closed interval $[l, u]$ in the discussion, but the results in the following sections treat intervals which can be open at one or two ends.

Non-deterministic delays pose problems for traditional simulation methods as the next "event" in the simulation can take place anywhere within an interval. In the sequel, in order not to drag with us too much notation, we will omit the reference to the initial value from the delay equations and use equations of the form $\beta = \Delta_{[l,u]}(\alpha)$.

**Definition 3 (Circuit).** *A $k$-variable digital circuit is a tuple $\mathcal{N} = (X, F, D)$ where $X = \{x_1, \ldots, x_k\}$ is a set of variables, $F = \{f_1, \ldots, f_k\}$ is a set of Boolean*

[4] Some models relax this condition and allow unboundedly small (but positive) delays.

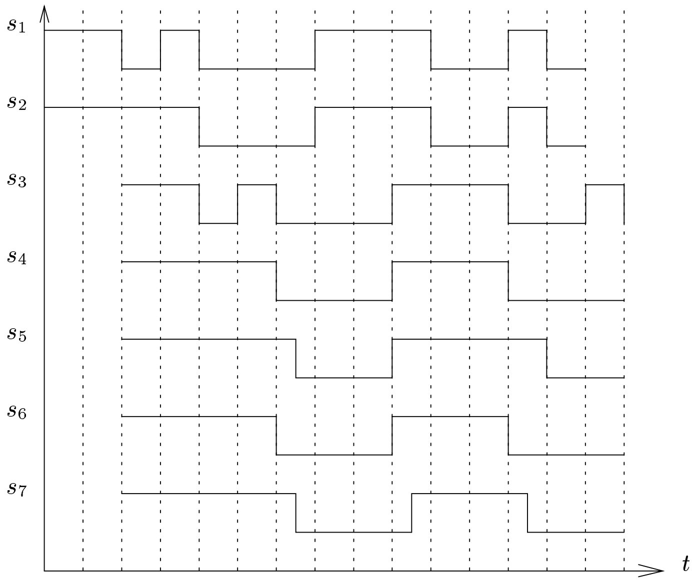



**Fig. 3.** The signal $s_2$ is a result of filtering away changes in $s_1$ which do not persist for 2 time units, $s_3$ is an ideal delay of $s_1$, shifted by 2, while $s_4$ is the inertial $[2, 2]$-delay of $s_1$. Finally $\{s_4, s_5, s_6, s_7\} \subseteq \Delta_{[2,3]}(s_1)$.

*functions of the form* $f_i : \mathbb{B}^k \to \mathbb{B}$ *and* $D = \{(l_1, u_1), \ldots, (l_k, u_k)\}$ *is a set of positive pairs of integers such that* $l_i \leq u_i$. *An* observable behavior *of the circuit is any* $\mathbb{B}^k$*-valued signal* $x = \langle x_1, \ldots, x_k \rangle$ *satisfying the system of simultaneous inclusions:*

$$\begin{array}{c} x_1 \in \Delta_{[l_1,u_1]}(f_1(x_1, \ldots, x_k)) \\ \cdots \\ x_k \in \Delta_{[l_k,u_k]}(f_k(x_1, \ldots, x_k)) \end{array} \tag{1}$$

A circuit appears in figure 4-(a). The correspondence between a circuit and the system of inclusions (1) is straightforward and we will refer to the latter as the description of the circuit. Needless to say, the system of inclusions (1) need not have a unique solution. The set of solutions is called the semantics of the circuit and is denoted by $L_{\mathcal{N}}$.

For certain purposes it is useful to introduce an auxiliary set of variables $Y = \{y_1, \ldots, y_k\}$ and consider the signal $y = \langle y_1, \ldots, y_k \rangle$ such that for every $i \in K$,

$$y_i = f_i(x_1, \ldots, x_k).$$

Every $y_i$ represents the "hidden" value of $x_i$, that is, the value that $x_i$ is about to obtain given that $f_i(x_1, \ldots, x_k)$ remains stable for a sufficiently long period. The signal $y$ is called the *hidden behavior* associated with $x$.

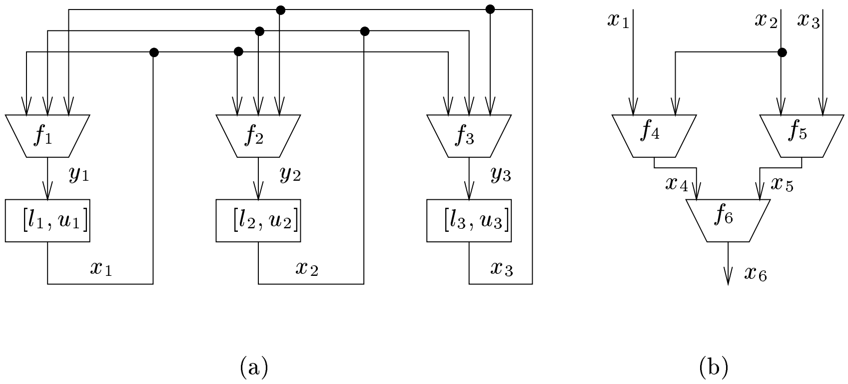


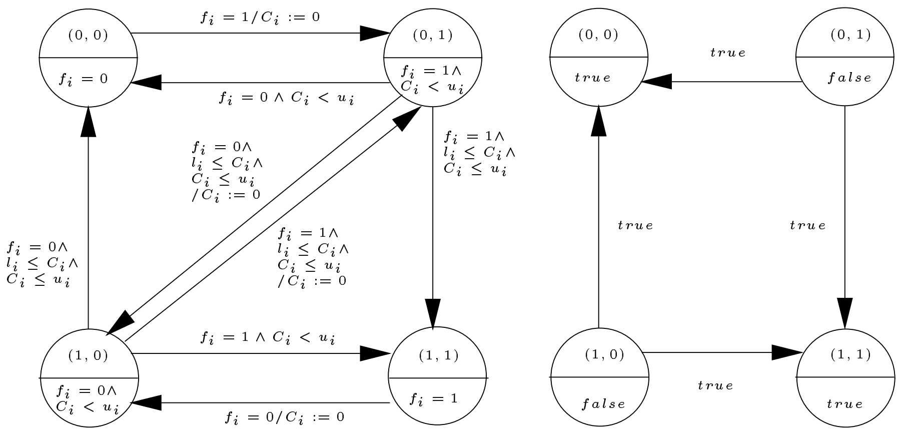

**Fig. 4.** (a) A 3-variable circuit. (b) An acyclic circuit (delays omitted) with 3 primary inputs.

In [MP95] it has been shown how to translate every equation of the form $x_i \in \Delta_{[l_i,u_i]}(f_i(x_1,\ldots,x_k))$ into a timed automaton with two Boolean variables (four states) and one clock (see figure 5). The composition of these $k$ automata yields an automaton $\mathcal{A}$ with $2^k$ states[5] and $k$ clocks, whose semantics $L_{\mathcal{A}}$ is exactly $L_{\mathcal{N}}$. This translation has been used for applying timed automata verification techniques [D89,AD94,HNSY94,ACD93] and tools [DOTY96] to various circuits, e.g. [MY96,BMPY97,TB97,BM98].

The model captured by the system of inclusions (1) is very general in the sense that it assumes that all the Boolean functions are $k$-ary, and, in principle, every change in one variable can trigger a change in any other variable. In practice, gates have a limited fan-in and each $f_i$ refers only to a small subset of the variables. Moreover there is some order in which information flows which can be captured by the wiring topology of the circuit (or the syntactic structure of $F$). For example, if the only equation in which $x_i$ appears on the left-hand side is of the form $x_i \in \Delta_{[d,\infty]}(\neg x_i)$, $x_i$ is an *input* signal whose rising and falling are separated by at least $d$ time units. Similarly $x_i$ is an *unconstrained* input signal if it does not appear in the left-hand side of *any* equation.

In the analysis of synchronous circuits with a central clock, it is often assumed that the circuit is *acyclic*, i.e. there is no cycle in the circuit layout. Such a circuit appears in figure 4-(b). The signals entering at the top are called the *primary inputs* of the circuit. A primary input which may change at most once at the beginning of the execution can be modeled by a timed automaton of the type appearing in figure 5-(b). We leave it to the reader to verify that a product of such input automata with the automata corresponding to the equations of an acyclic circuit is an acyclic automaton (no cycles in the transition graph), and hence the number of transitions in any run is finite and bounded.

[5] After composition, the values of the $y$-components are uniquely determined by the $x$-components and hence only $2^k$ out of the $4^k$ global states are possible.




(a) (b)

**Fig. 5.** (a) The automaton for the equation $x_i \in \Delta_{[l_i, u_i]}(f_i(x_1, \ldots, x_k))$. The states of the automaton correspond to the values of $(x_i, y_i)$. (b) An acyclic automaton for a primary input.

## 3 Qualitative Behaviors and their Realizability

In this section we introduce the notion of a qualitative behavior, a result of stripping away the quantitative properties of a signal and considering only the ordering relation among events.

Let $x$ be an observable behavior of a given circuit and let $y$ be the corresponding hidden behavior. We define three function $\mathcal{E}_X$, $\mathcal{E}_Y$ and $\mathcal{E} : \mathcal{J}(x) \to 2^K$ as follows:

$$\begin{aligned} \mathcal{E}_X(j) &= \{i : x_i[t_j] \neq x_i[t_{j-1}]\} \\ \mathcal{E}_Y(j) &= \{i : y_i[t_j] \neq y_i[t_{j-1}]\} \\ \mathcal{E}(j) &= \mathcal{E}_X(j) \cup \mathcal{E}_Y(j) \end{aligned}$$

In other words, $\mathcal{E}_X(j)$ is the set of all indices of the $x$-variables that change at time $t_j$. If $i \in \mathcal{E}_X(j)$ (resp. $i \in \mathcal{E}_Y(j)$) we say that $t_j$ is an $x_i$-event (resp. a $y_i$-event). If $i \in \mathcal{E}(j)$ we say that $t_j$ is an $i$-event. Note that $\mathcal{E}_Y(j) \neq \emptyset$ only if $\mathcal{E}_X(j) \neq \emptyset$.

Two behaviors $x$ and $x'$ are equivalent, denoted by $x \sim x'$, if their corresponding functions $\mathcal{E}_X$ and $\mathcal{E}'_X$ are identical. A *qualitative behavior* is an equivalence class of $\sim$, denoted by $[x]$, and it can be viewed as a string (without repetition) taken from $(\mathbb{B}^k)^* \cup (\mathbb{B}^k)^\omega$, which records the values of $x$ at $\mathcal{J}(x)$. We extend this notion to sets of signals, i.e. $[L] = \{[x] : x \in L\}$. The *number of events* in a signal $x$ is defined as:

$$Z(x) = \sum_{j \in \mathcal{J}(x)} |\mathcal{E}_X(j)|.$$

Let $\mathcal{N} = (X, F, D)$ be a circuit. A signal can be generated by $\mathcal{N}$ if it satisfies two types of constraints. The first type is *logical* and does not depend on the delay parameters:

1. For every $i$, $y_i = f_i(x_1, \ldots, x_k)$, where $f_i \in F$.
2. Every $y_i$-event is followed by an $i$-event. This means that every triggering of a variable is either aborted or concluded successfully.
3. Every $x_i$-event is preceded by a $y_i$ event (without any $x_i$-event between them): observable changes must be triggered first.

On the basis of these conditions we can rule out qualitative behaviors which are not realizable regardless of quantitative timing. For the rest of signals we define a partial function $\mathcal{F} : K \times \mathcal{J}(x) \to \mathcal{J}(x)$, which associates with every $i \in \{1, \ldots, k\}$ and $j$, such that $t_j$ is a $y_i$-event, a number $m > j$ such that $t_m$ is the time of the *next* $i$-event. Formally:

$$\mathcal{F}(i,j) = m \;\text{ iff }\; i \in \mathcal{E}_Y(j) \;\wedge\; i \in \mathcal{E}(m) \;\wedge \\ \forall m' \in [j+1, m-1] \;\; i \notin \mathcal{E}(m').$$

Note that $\mathcal{F}$ is a qualitative characteristics of a signal and is identical for every $x' \in [x]$. Moreover, the size of $\mathcal{F}$ (viewed as a relation) is at most $Z(x)$. The temporal distance between $t_m$ and $t_j$ must satisfy the timing constraints associated with $x_i$.

**Claim 1 (Characteristic Inequalities).** *A signal $x$ can be generated by a circuit $\mathcal{N} = (X, F, D)$ iff it satisfies the logical constraints as well as the following set of inequalities over $\mathcal{J}(x)$ (where $(l_i, u_i) \in D$):*

- ***Ordering Constraints:***
  *for every $j < |\mathcal{J}(x)|$*
  $$0 < t_{j+1} - t_j \tag{2}$$
- ***Timing Constraints:***
  *for every $m = \mathcal{F}(i,j)$ such that $i \notin \mathcal{E}_X(m)$ (abortion)*
  $$t_m - t_j \leq u_i \tag{3}$$
  *for every $m = \mathcal{F}(i,j)$ such that $i \in \mathcal{E}_X(m)$ (completion)*
  $$l_i \leq t_m - t_j \leq u_i \tag{4}$$

We denote the set of solutions of the system of inequalities (2), (3), and (4) associated with $[x]$ by $\mathcal{P}_{\mathcal{N}}([x])$. We use the term *t-polyhedra* to denote subsets of $T^n = \mathbb{R}^n_+$ which can be written as intersections of half-spaces of the form $t_m - t_j \prec c$ where $c$ is an integer and $\prec$ is either $<$ or $\leq$. By definition, $t$-polyhedra are convex.

**Corollary 1.** *A qualitative behavior $[x]$ is realizable by a circuit $\mathcal{N}$ iff its associated $t$-polyhedron $\mathcal{P}_{\mathcal{N}}([x])$ is non-empty.*

Let $T_\delta$ denote the set $\{m\delta : m \in \mathbb{N}\}$. An $n$-dimensional non-empty polyhedron $\mathcal{P}$ is $\delta$-discretizable if it has a non-empty intersection with the $\delta$-grid $T_\delta^n$. The problem of behavior-preserving discretization is reduced to a linear-algebraic problem:

**Corollary 2.** *A qualitative behavior $[x]$ realizable by a circuit $\mathcal{N}$ is preserved by a $\delta$-discretization of Time iff $\mathcal{P}_\mathcal{N}([x])$ is $\delta$-discretizable.*

We distinguish three types of $t$-polyhedra: *open* (all inequalities are strict), *closed* (all inequalities are non-strict), and *mixed.* Note that a non-empty open $t$-polyhedron is full-dimensional while a closed or mixed one might have degeneracies.

**Lemma 1.** *Every non-empty $t$-polyhedron $\mathcal{P} \subseteq \mathbb{R}_+^n$ contains:*

1. *a point of $\mathbb{N}^n$, when $\mathcal{P}$ is closed;*
2. *a point $(t_1, \ldots, t_n)$ with all fractional parts of coordinates $\langle t_i \rangle$ different from $0$, when $\mathcal{P}$ is open;*
3. *a point, when $\mathcal{P}$ is mixed.*

**Proof**:

1. First notice that if $l \leq x - y \leq u$ then $l \leq \lfloor x \rfloor - \lfloor y \rfloor \leq u$ when $l, u \in \mathbb{N}$. Suppose $(t_1, \ldots, t_n) \in \mathcal{P}$. It is immediate that $(\lfloor t_1 \rfloor, \ldots, \lfloor t_n \rfloor) \in \mathcal{P} \cap \mathbb{N}^n$.
2. An open $t$-polyhedron $\mathcal{P}$ is full-dimensional and convex. If we remove from $P$ all the hyper-planes $t_i = c$ for $i = 1, \ldots, n$ and $c \in \mathbb{N}$, the resulting set is still an open non-empty set. Let $(t_1, \ldots, t_n)$ be a point in this set. By construction it satisfies the statement of the lemma.
3. By definition of non-emptiness. □

Now we define an equivalence relation on $\mathbb{R}_+^n$, which is commonly-used in the theory of timed automata [D89,AD94]. Two points $(t_1, \ldots, t_n)$ and $(s_1, \ldots, s_n)$ are equivalent if and only if the integer parts of their coordinates coincide (i.e. $\lfloor t_i \rfloor = \lfloor s_i \rfloor$) and the order between the fractional parts of their coordinates is the same (i.e. $\langle t_i \rangle < \langle t_j \rangle$ iff $\langle s_i \rangle < \langle s_j \rangle$). The main property of this relation is that equivalent points satisfy exactly the same set of inequalities, and hence, a $t$-polyhedron containing a point should contain all its equivalence class.

**Lemma 2.** *In $\mathbb{R}_+^n$*

1. *Any point with all fractional parts of coordinates $\langle t_j \rangle$ different from $0$ has an equivalent point on any $\delta$-grid with $\delta < 1/n$.*
2. *Any point has an equivalent point on any $\delta$-grid with $\delta = 1/M < 1/n$, $M \in \mathbb{N}$.*

**Proof**: Let $(t_1, \ldots, t_n)$ be a point and let

$$b_j = \max\{m\delta : m\delta \leq \lfloor t_j \rfloor\}.$$

Without loss of generality suppose that $\langle t_1 \rangle \leq \cdots \leq \langle t_n \rangle$. Let

$$p_j = |\{\langle t_i \rangle : 0 < \langle t_i \rangle \leq \langle t_j \rangle\}|,$$

that is, for each $j \in \{1, \ldots, n\}$, $p_j$ counts the number of *different* $\langle t_i \rangle$'s such that $0 < \langle t_i \rangle \leq \langle t_j \rangle$. Note, in particular, that $p_1 = 0$ if $t_1 = 0$ and $p_1 = 1$ otherwise. Also observe that the ordering among the $p_j$'s is the same as among the $\langle t_j \rangle$'s and that every $p_j$ is smaller than $n$. Then, by letting

$$s_j = b_j + p_j \delta$$

we obtain $(s_1, \ldots, s_n)$ which is a point on the $\delta$-grid equivalent to $(t_1, \ldots, t_n)$.

$\square$

**Remark**: The proof is similar to that of [GPV94] where the authors prove that every timed automaton is discretizable. Their sense of discretization, however, distorts the passage of time.

**Corollary 3 (Discretization of Finite-dimensional $t$-Polyhedra).** *Every $t$-polyhedron $\mathcal{P} \subseteq \mathbb{R}^n$ is $\delta$-discretizable where*

1. *$\delta$ is of the form $1/M$ where $M \in \mathbb{N}$ (when $\mathcal{P}$ is closed). In particular $\mathcal{P}$ is 1-discretizable.*
2. *$\delta < 1/n$ (when $\mathcal{P}$ is open).*
3. *$\delta < 1/n$ and is of the form $1/M$, $M \in \mathbb{N}$ (when $\mathcal{P}$ is mixed).*

*These estimates are exact.*

It is a straightforward exercise to demonstrate $t$-polyhedra which are not $\delta$-discretizable for $\delta$ not satisfying the above conditions.

**Claim 2 (Discretization of Infinite-dimensional $t$-Polyhedra).**
*For infinite-dimensional $t$-polyhedra the following holds:*

1. *There exist open and mixed $t$-polyhedra which are not $\delta$-discretizable for any $\delta > 0$.*
2. *A closed $t$-polyhedron is $\delta$-discretizable if $\delta$ is of the form $1/M$, where $M \in \mathbb{N}$. In particular it is 1-discretizable.*

**Proof**:

1. (We give the proof for mixed polyhedra). Consider the infinite-dimensional $t$-polyhedron $\mathcal{P}$ defined by the following system of equations:

$$\begin{array}{rcl} 1 \leq & t_1 & \leq 2 \\ 2 \leq & s_1 & \leq 3 \\ 2 \leq & r_1 & \leq 3 \\ 1 \leq & t_{j+1} - t_j & \leq 2 \\ 2 \leq & s_{j+1} - s_j & \leq 3 \\ 2 \leq & r_{j+1} - r_j & \leq 3 \end{array} \tag{5}$$

and

$$t_{2j-1} < s_{2j-1} < r_{2j-1} < t_{2j} < r_{2j} < s_{2j} < t_{2j+1}$$

for $j \in \mathbb{N}$. This polyhedron is non-empty and it contains, for example, the point

$$\begin{aligned} t_j &= 2j \\ s_j &= 2j + 2 - 2^{-j} + (-5)^{-j} \\ r_j &= 2j + 2 - 2^{-j} - (-5)^{-j}. \end{aligned}$$

However it is not $\delta$-discretizable for any $\delta$. Suppose the contrary. It follows from the inequalities (5) that the distance between $t_j$ and $s_j$ (or $r_j$) never decreases:

$$s_{j+1} - t_{j+1} \geq s_j - t_j; \qquad r_{j+1} - t_{j+1} \geq r_j - t_j$$

An induction proves that in any $\delta$-realization this distance, in fact, increases linearly:

$$\begin{aligned} s_{2j-1} - t_{2j-1} &\geq (2j-1)\delta \\ r_{2j-1} - t_{2j-1} &\geq 2j\delta \\ r_{2j} - t_{2j} &\geq 2j\delta \\ s_{2j} - t_{2j} &\geq (2j+1)\delta \end{aligned}$$

which contradicts the ordering inequality $s_{2j} < t_{2j+1} \leq t_{2j} + 2$ when $j$ is large enough (namely when $(2j+1)\delta \geq 2$).

2. Similarly to the finite-dimensional case. Suppose $(t_1, \ldots, t_j, \ldots) \in \mathcal{P}$. It is immediate that $(\lfloor t_1 \rfloor, \ldots, \lfloor t_j \rfloor, \ldots) \in \mathcal{P} \cap \mathbb{N}^\infty$. Hence $\mathcal{P}$ is 1-discretizable and consequently $1/M$-discretizable. □

The results concerning closed $t$-polyhedra might tempt one to think that by "closing" all timing constraints it is possible to 1-discretize all circuits (i.e. that for these circuits the dense-time and discrete-time semantics coincide). Unfortunately this is not the case: the characteristic $t$-polyhedron of a qualitative behavior is defined by two sets of inequalities. While the timing constraints can be made closed by an (infinitesimal) modification of the circuit model, the *ordering* constraints $t_0 < t_1 < t_2, \ldots$ are open by nature, the resulting polyhedron is mixed and a discretization of $\delta = 1/M < 1/n$ is necessary for the acyclic case. For cyclic circuits, the negative result of claim 2 applies.

By relaxing the ordering constraints into $t_0 \leq t_1 \leq t_2 \ldots$ we obtain a *weaker* notion of behavior preservation. For every qualitative behavior $[x]$, realizable by a dense time circuit, there is a qualitative behavior $[x']$, realizable in discrete time, such that some events that occur *at different time instants* in $x$, take place *at the same time instant* in $x'$. This is the notion of preservation used in [HMP92] who employ a "timed trace" model where $(a, t_1)(b, t_2) \sim (a, t_1)(b, t_1)$ but $(a, t_1)(b, t_1) \not\sim (b, t_1), (a, t_1)$. To demonstrate the weak preservation phenomenon consider the circuit described by

$$x_1 \in \Delta_{[1,2]}(\neg x_0) \quad x_2 \in \Delta_{[1,2]}(\neg x_0) \quad x_3 \in \Delta_{[1,2]}(\neg x_0)$$

The qualitative behavior

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

can be realized by $t_1$, $t_2$ and $t_3$ satisfying

$$1 \leq t_1 < t_2 < t_3 \leq 2.$$

Clearly, this $t$-polyhedron does not contain an integer point. Only by relaxing the ordering relation between the events into

$$1 \leq t_1 \leq t_2 \leq t_3 \leq 2$$

we can 1-discretize and obtain a behavior such as $\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$.

**Theorem 1 (Main Result).**

1. *Every acyclic circuit can be $\delta$-discretized with $\delta = 1/M < 1/n$, where $n$ is the maximum of $Z(x)$ over all qualitative behaviors which are logically realizable by the circuit.*
2. *There are cyclic circuits which are not discretizable at all.*
3. *All circuits with closed delay intervals can be 1-discretized with weak preservation of behaviors.*

**Proof**:
1. An immediate consequence of corollary 3.
2. Consider the circuit described by

$$x_1 \in \varDelta_{[1,2]}(\neg x_1) \quad x_2 \in \varDelta_{[2,3]}(\neg x_2) \quad x_3 \in \varDelta_{[2,3]}(\neg x_3)$$

   and the qualitative behavior

$$\left( \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right)^{\omega}.$$

   The characteristic polyhedron of this behavior is exactly the one defined by the inequalities (5), if we take $t_j$, $r_j$ and $s_j$ to denote the $j^{th}$ transition times of $x_1$, $x_2$ and $x_3$ respectively. The result follows from claim 2-1.
3. This is essentially the result of [HMP92] and it follows from claim 2-2. □

## 4 Preservation of Properties

In this section we use rather informally the term *closed* for speaking of circuits or timed automata whose timing conditions are closed, and for the languages of signals generated by such automata. For a non-closed automaton $\mathcal{A}$ we use $\bar{\mathcal{A}}$ to denote its closure, i.e. the automaton obtained by replacing all open inequalities by closed ones. Similarly we denote the closure of a sets of signals $L$ by $\bar{L}$ with the obvious property $L \subseteq \bar{L}$. From claim 2 we can conclude:

**Corollary 4 (Emptiness of Closed Circuits and Automata).** *Let $\mathcal{A}$ be a closed automaton, and let $\mathcal{A}'$ be the* 1*-discretization of* $\mathcal{A}$*. Then $L'_{\mathcal{A}} = \emptyset$ iff $L_{\mathcal{A}} = \emptyset$.*

This positive result is perhaps more significant from a practical point of view of verification than the negative result of theorem 1. Suppose that a desired property of an automaton $\mathcal{A}$ is specified by a formula $\varphi$ denoting a language $L_\varphi$ whose negation is $L_{\neg\varphi}$. If both $L_{\mathcal{A}}$ and $L_{\neg\varphi}$ are closed, one can do verification on their 1-discretization-s $L'_{\mathcal{A}}$ and $L'_{\neg\varphi}$ because $L'_{\mathcal{A}} \cap L'_{\neg\varphi} = \emptyset$ iff $L_{\mathcal{A}} \cap L_{\neg\varphi} = \emptyset$. In the case that $L_{\mathcal{A}}$ and $L_{\neg\varphi}$ are not closed, one can discretize their closures $\bar{L}_{\mathcal{A}}$ and $\bar{L}_{\neg\varphi}$ into $\bar{L}'_{\mathcal{A}}$ and $\bar{L}'_{\neg\varphi}$ and perform verification on those. The results are valid since $\bar{L}'_{\mathcal{A}} \cap \bar{L}'_{\neg\varphi} = \emptyset$ implies $L_{\mathcal{A}} \cap L_\varphi = \emptyset$.

Note that we have not treated the question of transforming $L_\varphi$ into $L_{\neg\varphi}$ due to the problematics of complementation for timed automata. However, in the special case where $L_\varphi$ is untimed (for every $[x]$, either $[x] \subseteq L_\varphi$ or $[x] \cap L_\varphi = \emptyset$), $L_{\neg\varphi}$ is untimed as well and the characteristic polyhedron of every qualitative behavior is either empty or universal and can be 1-discretized.

**Corollary 5 (Untimed Properties of Automata).** *Untimed properties of closed circuits/automata can be verified using the discrete time semantics. Untimed properties of non-closed automata can be verified using the discrete semantics with the risk of creating false negatives.*

In [BM98] a low-level asynchronous realization of a FIFO buffer was verified using a discrete time model. Since the specification of the desired behavior is the untimed language of compatible `read`s and `write`s from the buffer, the verification results carry over to dense time. We are currently investigating which other classes of properties can be verified safely using discrete time. Some suggestions appeared already in [HMP92].

## 5 Discussion

The main contribution of this paper is in shedding some more light on the relation between discrete and dense time models, and in solving an open problem concerning the discretization of circuits. We believe that the circuit model and the geometric analysis techniques introduced in this paper will be useful both for hardware timing verification and for advancing the theory of timed automata. In particular it currently seems that for most reasonable practical purposes, discrete time verification will do the job.

## References


[ACD93] R. Alur, C. Courcoubetis, and D.L. Dill, Model Checking in Dense Real Time, *Information and Computation* 104, 2–34, 1993.

[AD94] R. Alur and D.L. Dill. A theory of timed automata, *Theoretical Computer Science*, 126, 183–235, 1994.

[ABK+97] E. Asarin, M. Bozga, A. Kerbrat, O. Maler, A. Pnueli and A. Rasse, Data-Structures for the Verification of Timed Automata, in O. Maler (Ed.), *Proc. HART'97*, LNCS 1201, 346-360, Springer, 1997.

[BM98] M. Bozga and O. Maler, Modeling and Verification of the STARI Chip using Timed Automata, submitted, 1998.

[BMPY97] M. Bozga, O. Maler, A. Pnueli and S. Yovine, Some Progress in the Symbolic Verification of Timed Automata, in O. Grumberg (Ed.) *Proc. CAV'97*, 179-190, LNCS 1254, Springer, 1997.

[BS94] J.A. Brzozowski and C-J.H. Seger, *Asynchronous Circuits*, Springer, 1994.

[DOTY96] C. Daws, A. Olivero, S. Tripakis, and S. Yovine, The Tool KRONOS, in R. Alur, T.A. Henzinger and E. Sontag (Eds.), *Hybrid Systems III*, LNCS 1066, 208-219, Springer, 1996.

[D89] D.L. Dill, Timing Assumptions and Verification of Finite-State Concurrent Systems, in J. Sifakis (Ed.), *Automatic Verification Methods for Finite State Systems*, LNCS 407, 197-212, Springer, 1989.

[HNSY94] T. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine, Symbolic Model-checking for Real-time Systems, *Information and Computation* 111, 193–244, 1994.

[L90] H.R. Lewis, A logic of concrete time intevrals, *Proc. LICS'90*, IEEE, 1990.

[GPV94] A. Göllü, A. Puri and P. Varaiya, Discretization of Timed Automata, *Proc. 33rd CDC*, 1994.

[HMP92] T. Henzinger, Z. Manna, and A. Pnueli. What Good are Digital Clocks?, in W. Kuich (Ed.), *Proc. ICALP'92*, LNCS 623, 545-558, Springer, 1992.

[MP95] O. Maler and A. Pnueli, Timing Analysis of Asynchronous Circuits using Timed Automata, in P.E. Camurati, H. Eveking (Eds.), *Proc. CHARME'95*, LNCS 987, 189-205, Springer, 1995.

[MY96] O. Maler and S. Yovine, Hardware Timing Verification using KRONOS, In *Proc. 7th Israeli Conference on Computer Systems and Software Engineering*, Herzliya, Israel, June 1996.

[RT97] A. Rabinovich and B.A. Trakhtenbrot, From finite automata toward hybrid systems, *Proc. FCT'97*, 1997.

[TB97] S. Tasiran and R.K. Brayton, STARI: A Case Study in Compositional and Hierarchical Timing Verification, in O. Grumberg (Ed.) *Proc. CAV'97*, 191-201, LNCS 1254, Springer, 1997.