Toward a Theory of Superdense Time in Simulation Models

JAMES NUTARO, Oak Ridge National Laboratory, USA

We develop a theory of superdense time that encompasses existing uses of superdense time in discrete event simulations and points to new forms that have not previously been explored. A central feature of our development is a set of axioms for superdense time. The sufficiency of these axioms is demonstrated by using them to prove that a general model of a discrete event simulation procedure, expressed in terms of a mathematical system, constitutes a state transition function. Several forms of superdense time, both known and novel, are shown to satisfy the axioms.

CCS Concepts: • **Computing methodologies → Modeling and simulation**; **Discrete-event simulation**;

Additional Key Words and Phrases: Systems theory, discrete-event simulation, agent/discrete models, modeling methodologies

ACM Reference format:
James Nutaro. 2020. Toward a Theory of Superdense Time in Simulation Models. *ACM Trans. Model. Comput. Simul.* 30, 3, Article 16 (May 2020), 13 pages.
https://doi.org/10.1145/3379489

16

1 INTRODUCTION

Superdense time is an important tool for modeling simultaneity in discrete event simulations. A superdense time base is typically constructed by augmenting the real numbers with information for ordering events that would otherwise appear to be simultaneous. When time is restricted to physically meaningful quantities, such as seconds, then simultaneous and zero-time events in a discrete event model may assign several values to a variable at a single instant of time. One readily apparent consequence of this ambiguity is in parallel discrete event simulations when the ordering of events with identical timestamps changes from run to run; see, e.g., the discussion in Refs [4] and [24]. Superdense time addresses this problem with an augmented timestamp that enables repeatable, unambiguous simulation traces.

The first use of superdense time was by Maler, Manna, and Pnueli [14] to distinguish simultaneity from sequences of instantaneous actions in a study of hybrid automata. Somewhat later, superdense time was adopted as a solution to the same problem in optimistic and conservative parallel discrete event simulations; see, e.g., Refs [9], [19], [20], [24], and [25]. Most recently, work

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).
Authors' address: J. Nutaro, Oak Ridge National Laboratory, One Bethel Valley Road, Oak Ridge, TN, 37830; email: nutarojj@ornl.gov.

2020. 1049-3301/2020/05-ART16 $15.00
https://doi.org/10.1145/3379489

on co-simulation of cyber-physical systems encountered the simultaneous event problem and has used superdense time as a solution; see, e.g., Refs [1], [3], [11], [12], and [16].

Examples of superdense time, discussions of time's general properties in various modeling and simulation contexts, and related studies concerning the general structure of systems are many and scattered. Nonetheless, these disparate studies share common ideas about how time should appear in a simulation model. Our main objective is to weld key ideas from these studies into a concise description of the properties of superdense time.

This attempt begins with properties that time must possess. The necessity of a zero element is taken from Ref. [14], as is the notion of an interval length consistent with stepping through time. In Section 3, these appear as properties P1, P2, and P4. Of course, Maler, Manna, and Pnueli are not the only authors to require these properties; they are implicit in the works of systems' theorists like Zeigler et al. [21, 27], Mesarovic and Takahara [15], Wymore [26], and Klir [10]. In a similar fashion, Cremona et al. [3] argue for an associative + operator, which is P3 in our set of properties. The need for a successor of each time instant is inseparable from discrete time simulations, and superdense time, as proposed in Refs [12], [14], [19], [20], and elsewhere, bring this requirement to discrete event simulation; it appears as P5.

Our assemblage of properties is not all encompassing. For example, Maler, Manna, and Pnueli require that + commute, but we do not. Likewise, resolution, numerical rounding, and other practical issues are not considered. Our choice of properties is limited to those sufficient for proving that an abstract, but very general, model of a discrete event simulation procedure constitutes a state transition function. This selection criteria emerges from the concept of a system as described, for example, by Zeigler et al. [21, 27], Mesarovic and Takahara [15], Wymore [26], Klir [10], and others. With this approach, we hope to reveal a simple structure of time, which is sufficient to implement a simulation procedure that behaves in an intuitively appealing way.

After introducing the key properties of time and demonstrating their sufficiency, we present several types of superdense time, which include both known and novel forms, and show that they share the key properties. It is significant that the forms of superdense time suitable for simulation applications are prolific. The unifying axioms offered here make possible simulation tools in which the most useful form of time can be selected by the modeler for a given application. The many existing proposals for superdense time suggest that this is a desirable feature.

2 SYSTEMS

To go beyond specific instances of superdense time and arrive at unifying principles, it is necessary to distill the common elements of many simulation procedures as they relate to time. One approach is to use the concept of a mathematical system. For our purposes, it is enough to consider a closed system without input and output.

An abstract model of a closed system has a state $q(t)$ that is a function of time and evolves over an interval $[t, t')$ via a state transition function

$$q(t') = \Delta(q(t), [t, t')).$$

This function must satisfy the *composition property*, which stipulates that for any division of $[t, t')$ into sub-intervals $[t, t_m)$ and $[t_m, t')$,

$$\Delta(q(t), [t, t')) = \Delta(\Delta(q(t), [t, t_m)), [t_m, t')) \text{ and}$$
$$\Delta(q(t), [t, t)) = q(t).$$

In short, this is a requirement that, regardless of whether we examine intermediate states in an interval, the end point is always the same.

The theoretical and practical importance of the composition property has been discussed elsewhere; see, e.g., Refs [15], [20], [21], and [27]. An example of one of its roles in simulation is to enable checkpoints. A simulation checkpoint has the form described above when t_m is the instant of the checkpoint. We expect that running the simulation to a checkpoint, stopping, and then restarting at that checkpoint will give the same result as running from beginning to end without a pause. Another example is parallel in time simulation techniques. These implicitly assume a calculation can be split into two or more time segments that, when stitched back together, give the same solution as running the calculation from beginning to end [5]. The widespread, but often implicit, importance of the composition property in theoretical and practical simulation problems justifies its use in our axiomatic approach to superdense time.

A trivial example of a mathematical system uses the integers for time and evolves its state according to

$$q(t+1) = \delta(q(t)). \tag{1}$$

Defining $\delta(q(t)) = \Delta(q(t), [t, t+1))$, the composition property follows by induction and so δ is an instance of a mathematical system. This may be generalized in a natural way to describe a discrete event simulation [20, 21]. To do so, we retain the integers for time and explicitly identify the model state s and time e for which that state has persisted. This makes the total state a pair $q = (s, e)$.

Each state s has a lifetime $ta(s)$, which is the amount of time that must elapse before the next event occurs. Hence, the state of the model changes when $e = ta(s)$. At this instant, an event handler $\rho(s) = s'$ operates on s and e becomes zero. Naturally, $e \leq ta(s)$ because s does not persist longer than its lifetime. A definition of Δ that corresponds to this behavior is

$$q(t') = \Delta((s,e),[t,t')) = \begin{cases} \Delta((\rho(s),0),[t+1,t')) & e = ta(s) \text{ and } t < t' \\ \Delta((s,ta(s)),[t+ta(s)-e,t')) & e < ta(s) \text{ and } t+ta(s)-e \leq t' \\ (s, e+t'-t) & \text{otherwise.} \end{cases} \tag{2}$$

When t is taken from the integers, it is straightforward to show that Equations (1) and (2) generate the same state trajectories; again, see Refs [20] and [21]. However, the latter acts as a discrete event simulation by taking advantage of the lifetime $ta(s)$ (or, less abstractly, the event scheduler) to skip over intervals in which nothing interesting happens; that is, when s does not change. In what follows, we assume ta and ρ are such that Equation (2) defines a legitimate system, free of Zeno behaviors, which cause the simulation clock to become stuck at a given instant of time; see, e.g., Ref. [21], and [28] for a technical treatment of Zeno behaviors.

A fundamental motivation for superdense time is apparent in the first and second lines of Equation (2). When the lifetime expires, the model must have a single state. In general, this state cannot be both s and $\rho(s)$. When simulating with Equation (2), the state at the moment of expiration is s, and in the next moment, it is $\rho(s)$. If the event handler schedules a new event such that $ta(s) = 0$, then the new state belongs to the next instant of time.

In practice, the integers are an unattractive model of time in discrete event simulations because the integers do not allow for a natural notion of instantaneous, or zero-time, events. More useful forms for time that support a natural model of instantaneous events can be used in Equation (2) if they have a structure similar to the integers in key aspects. For example, suppose we replace integer time in Equation (2) with a time pair (t, c). The element t is a real valued, physically meaningful unit of time within the context of the model. The integer c counts events that have been executed. Hence, the initial state of the model is $q((t_0, 0))$ and subsequent states are $q((t_1, 1))$, $q((t_2, 2))$, and so forth.

Consider how ta, addition, and subtraction could be defined for time pairs so that Equation (2) keeps the composition property. The lifetime $ta(s)$ is a pair $(h, 1)$, indicating that this single event will advance the real time by h units while adding one to the count of events. We define $+$, $-$, 1, and 0 in the narrow sense that they appear in Equation (2) by

$$\begin{aligned}(t,c)+(h,k)&=(t+h,c+k),\\(t,c)-(h,k)&=(t-h,c-k),\\0&=(0,0),\ \text{and}\\1&=(0,1).\end{aligned}$$

Close examination of $-$ as it is used in Equation (2) suggests that it is not subtraction in the usual sense, but has the much more limited role of measuring intervals. In particular, the elapsed time has real and integer parts less than or equal to those of $ta(s)$, and so $ta(s)-e$ is always non-negative.

Replacing the integers with this form of time, we find that the model still behaves as intended in the sense that there is a single state value at each time instant, and the composition property holds. Because of this, we can claim that the above scheme is a legitimate form of simulation time.

There are other forms of time that allow Equation (2) to behave in the expected way. One example also uses the pair (t,c), $0=(0,0)$, and $1=(0,1)$, but addition and subtraction are defined by

$$(t,c)+(h,k)=\begin{cases}(t+h,k) & h\neq 0\\(t,c+k) & h=0\end{cases}\ \text{and}$$
$$(t,c)-(h,k)=\begin{cases}(t-h,c) & t\neq h\\(0,c-k) & t=h.\end{cases}$$

This form of time arises in simulations of cyber-physical systems where the integer part counts "ticks" of a logical clock at each real instant.

3 TIME

If there are three acceptable forms of simulation time, we may ask how many more exist and what do they look like? One approach to this question examines what properties time must have for Equation (2) to have the composition property. The following five are sufficient.

- P1. There exists 0 such that, for all t, $t+0=0+t=t$.
- P2. If $h_1>h_2\geq 0$, then $t+h_1>t+h_2$.
- P3. If $h_1\geq 0$ and $h_2\geq 0$, then $(t+h_1)+h_2=t+(h_1+h_2)$.
- P4. If $t_2\geq t_1$, then there exists $h\geq 0$ such that $t_1+h=t_2$.
- P5. There is a successor function $S(t)$ such that the interval $[t,S(t))$ contains exactly t.

The length ℓ of an interval $[t_1,t_2)$ is the number h in P4. Two necessary properties of ℓ follow from this definition.

- P4.1. $\ell[t_1,t_2)\geq 0$.
- P4.2. $t_1+\ell[t_1,t_2)=t_2$.

The general model of a discrete event simulation can be rewritten in terms of S and ℓ as

$$q(t')=\Delta((s,e),[t,t'))=\begin{cases}\Delta((\rho(s),0),[S(t),t')) & e=ta(s)\text{ and }t<t'\\\Delta((s,ta(s)),[t+\ell[e,ta(s)),t')) & e<ta(s)\text{ and }t+\ell[e,ta(s))\leq t'\\(s,e+\ell[t,t')) & \text{otherwise.}\end{cases}\tag{3}$$

Several assumptions concerning the simulation application are implicit in P1–P5. Two central assumptions are that the initial time of the simulation can be anywhere on the number line and that our modeling objectives only concern moving forward in time. The latter implies that $ta(s)\geq 0$.

If time is restricted to numbers not less than 0, then the caveats ≥ 0 can be removed from P2 and P3. Perhaps this is reasonable, but we chose not to do so here.

Backward motion through time requires an analog to subtraction as it usually appears in algebraic structures, rather than in the limited sense of measuring interval lengths. Introducing this would significantly restrict the allowable forms of superdense time and, perhaps most importantly, appears to eliminate some popular choices. The latter observation strongly suggests forward motion in time is the most important case, and this is where we focus our attention.

The successor function required by P5 implies that the real numbers cannot be used as a model for time in discrete event simulations. The reason for including P5 is to avoid circumstances where the model state at time t is s and then, if $ta(s) = 0$, we assign a new state s' at time $t + ta(s) = t$. In this case, the state trajectory ceases to be a function and, therefore, Δ is not a state transition function. In particular, we cannot replace $[S(t), t')$ in Equation (3) with $[t, t')$ and still ensure a single value for q is assigned to each time t. It is conceivable that some weaker form of P5 could replace $S(t)$ with some other quantity not equal t, but it is not clear what that alternative might be.

If time is the integers, then P1–P5 are satisfied by the usual notion of addition and $\ell[t_1, t_2) = t_2 - t_1$. However, the integers have more structure than is needed. For example, the second form of time discussed briefly in Section 2 has a + operator that does not commute, P2–P4 hold only when their precondition applies, and ℓ is not associative. Indeed, if we equate subtraction with ℓ, then

$$\ell[(1,3), \ell[(2,2),(4,3))) = ((4,3) - (2,2)) - (1,3) = (2,3) - (1,3) = (1,3), \text{ but}$$
$$\ell[\ell[(1,3),(2,2)),(4,3)) = (4,3) - ((2,2) - (1,3)) = (4,3) - (1,2) = (3,3).$$

Nonetheless, P1–P5 impose a structure on time that is fundamental to simulation applications. The most important aspects of this structure feature prominently in our proof that Equation (3) has the composition property. These structural features are summarized by Propositions 1–7 below. Proofs of these propositions are given in the appendices.

Proposition 1. *The number 0 is unique.*

Proposition 2. *The successor $S(t)$ of t is unique.*

Proposition 3. $\ell[t_0, t_f) = 0$ *if, and only if,* $t_0 = t_f$.

Proposition 4. *If* $b \geq 0$, $c \geq 0$, *and* $a + b = a + c$, *then* $b = c$.

Proposition 5. $\ell[t_0, t_f) = \ell[t_0, t_m) + \ell[t_m, t_f)$.

Proposition 6. $\ell[t, t + h) = h$.

Consider an interval $[t, t')$. Let t_a in this interval be the location of the first event. By definition (the second line of Equation (3)), $t_a = t + \ell[e, ta(s))$. It is straightforward to show (see the appendices) that stepping from t to t_a is the same as advancing e to $ta(s)$. Hence, we have

Proposition 7. $e + \ell[t, t_a) = ta(s)$ *if, and only if,* $t + \ell[e, ta(s)) = t_a$.

With these Propositions, we can show that Equation (3) has the composition property. Each step of the proof that relies on one of the above Propositions is labeled as such. Otherwise, the step depends directly on Equation (3) or P1–P5. By examining the proofs in Appendix A, it can be seen that P1–P5 are used, either directly or indirectly, in the proof of Theorem 1.

Theorem 1. *Equation (3) has the composition property.*

Proof. Consider an interval $[t, t')$ and t_a as defined above. If $t = t'$, then the interval is empty and

$$\begin{aligned}\Delta((s,e),[t,t)) &= (s, e + \ell[t,t)) \\ &= (s, e+0) && \text{Proposition 3} \\ &= (s,e).\end{aligned}$$

Let $t < t'$, and pick a point t_m in the interval. There are four possibilities.

Case 1. If t_a is not in the interval, then

$$\begin{aligned}\Delta((s,e),[t,t')) &= (s, e + \ell[t,t')) \text{ and} \\ \Delta(\Delta((s,e),[t,t_m)),[t_m,t')) &= \Delta((s, e+\ell[t,t_m)),[t_m,t')) \\ &= (s, e + \ell[t,t_m) + \ell[t_m,t')) \\ &= (s, e+\ell[t,t')). && \text{Proposition 5}\end{aligned}$$

Case 2. If $t_m < t_a \le t'$, then

$$\begin{aligned}\Delta((s,e),[t,t')) &= \Delta((s,ta(s)),[t+\ell[e,ta(s)),t')) \\ &= \Delta((s,ta(s)),[t_a,t')), \text{ and} \\ \Delta(\Delta((s,e),[t,t_m)),[t_m,t')) &= \Delta((s,e+\ell[t,t_m)),[t_m,t')) \\ &= \Delta((s,ta(s)),[t_m+\ell[e+\ell[t,t_m),ta(s)),t')).\end{aligned}$$

For equality, we need $t_m + \ell[e+\ell[t,t_m),ta(s)) = t_a$. Using Propositions 5–7,

$$\begin{aligned}e + \ell[t,t_m) + \ell[t_m,t_a) &= e + \ell[t,t_a) && \text{Proposition 5} \\ &= ta(s), \text{ and so} && \text{Proposition 7} \\ \ell[e+\ell[t,t_m),ta(s)) &= \ell[e+\ell[t,t_m), e+\ell[t,t_m)+\ell[t_m,t_a)) \\ &= \ell[t_m,t_a). && \text{Proposition 6}\end{aligned}$$

It follows that

$$t_m + \ell[e+\ell[t,t_m),ta(s)) = t_m + \ell[t_m,t_a) = t_a$$

as desired.

Case 3. If $t_m = t_a$, then

$$\begin{aligned}\Delta((s,e),[t,t')) &= \Delta((s,ta(s)),[t_m,t')) = \Delta((\rho(s),0),[S(t_m),t')), \text{ and} \\ \Delta(\Delta((s,e),[t,t_m)),[t_m,t')) &= \Delta((s,e+\ell[t,t_m)),[t_m,t')) \\ &= \Delta((s,ta(s)),[t_m,t')) \\ &= \Delta((\rho(s),0),[S(t_m),t')).\end{aligned}$$

Case 4. Otherwise, $t_a < t_m < t'$. Proceeding to the first event at t_a, we have

$$\begin{aligned}\Delta((s,e),[t,t')) &= \Delta((s,ta(s)),[t_a,t')) \\ &= \Delta((\rho(s),0),[S(t_a),t')).\end{aligned}$$

There is a first event at time t_{aa} in the interval $[S(t_a), t')$ or no event exists in this interval. If $t_m > t_{aa}$, then we proceed to t_{aa} in the same manner as before. Advancing in this way, t_m must eventually be found in some interval $[t_{a\ldots a}, t')$. If there is no event in this interval, then Case 1 completes the proof. Otherwise, Case 2 or 3 completes the proof. □

4 EXAMPLES OF SUPERDENSE TIME

Several prominent forms of superdense time can be expressed in terms of a set with the $+$, ℓ, and S operators, and when expressed in this form, can be shown to satisfy P1–P5. The models of time presented here are cited frequently, but no attempt has been made to create an exhaustive catalog. Indeed, the two novel forms demonstrate that the well–known models of time are not the only possible models.

4.1 Rönngren and Liljenstam; Lee and Zheng

Rönngren and Liljenstam introduced a timestamping algorithm that can be reinterpreted as a form of superdense time. Their objective was to resolve the problem of variability in otherwise identical runs of a parallel discrete event simulation algorithm when events with the same timestamp are present. Their solution was to provide the same support for event ordering as is found in a sequentially executing simulation. A sequentially executing simulation is deterministic in the sense that two executions of a single simulation model with identical input will produce identical output; the results are reproducible.

This is in contrast to a parallel discrete event simulation where the result of a simulation experiment is not necessarily reproducible. The root cause is that in a parallel discrete event simulation, there is no fixed order in which events are generated. Moreover, the local causality constraint, which is the common criteria for correct execution [8], does not constrain the execution order of events with identical timestamps. Consequently, two events with identical timestamps may be executed in a different order in two otherwise identical simulation runs.

Rönngren and Liljenstam resolve this problem by appending a counter to the event time forming the pair (t, c), and these are sorted in lexicographic order. Each logical process maintains its own counter c and model time t. If t does not change while processing events with time stamps (t, k), c is updated by assigning $c \leftarrow \max\{c, k\} + 1$. When t advances, this updating scheme can continue, or c may be reset to some smaller value. The latter is proposed in Ref. [24] to prevent the counter from rolling over, and a reset to zero is made possible by adding a third field to the timestamp.

As proposed, the only purpose of this timestamp scheme is to create a causally consistent ordering of events and consideration of time's structure is restricted to order. The $\max\{c, k\}$ rule is a natural choice in that context, but formulating a superdense time using this rule is difficult, if not impossible, without violating P2. However, maximization is not the only update compatible with their proposed scheme. The general requirement is that when h is not zero, the update to c be greater than or equal to c and k. The rule $c \leftarrow c + k$ satisfies this requirement. Hence, we may keep essentially the same scheme from Ref. [24] without the additional timestamp field by altering the update rule and reset value.

If at time (t, c) an event is processed with timestamp $(t + h, k)$, $h > 0$, the modified scheme sets the current simulation time to $(t + h, k)$. If $h = 0$, the order counter is updated by setting the simulation clock to $(t, c + k)$. Addition by one in the original scheme is accomplished by Equation (3) with the successor function.

This timestamping scheme constitutes a superdense time. The interval between (t, c) and $(t + h, k)$ has a length

$$\ell[(t, c), (t + h, k)) = \begin{cases} (0, k - c) & h = 0 \\ (h, k) & h > 0. \end{cases}$$

Forward motion in time occurs via

$$(t, c) + (h, k) = \begin{cases} (t, c + k) & h = 0 \\ (t + h, k) & h > 0. \end{cases}$$

Now, if $h > 0$, then advancing the clock from (t, c) to $(t + h, k)$ is the same as calculating

$$(t, c) + \ell[(t, c), (t + h, k)) = (t, c) + (h, k) = (t + h, k).$$

If $h = 0$ and we are advancing from (t, c) to (t, k), then $c \leq k$ due to lexicographic ordering, and this is the same as calculating

$$(t, c) + \ell[(t, c), (t, k)) = (t, c) + (0, k - c) = (t, k).$$

We may take $0 = (0, 0)$ and $S((t, c)) = (t, c + 1)$. The latter ensures that $S(t)$ in Equation (3) has the same effect as adding one to the logical process's order counter. Proofs that this satisfies P1–P5 are in the appendix.

This form of superdense time encompasses a family of schemes used in simulations of cyber-physical systems. If the time advance operator is such that $k = 1$ when $h = 0$, and $k = 0$ when $h > 0$, then this is a typical realization of the superdense timestamping scheme proposed by Lee and Zheng in Ref. [12]. If $k = 0$ when $h > 0$, but $k \geq 0$ otherwise, then this is the timestamping scheme proposed in Ref. [20] (Chapter 4).

4.2 Benveniste et al.; Mosterman et al.; Barros

Replacing reals with hyper-reals in Section 4.1 gives a form of superdense time first proposed by Iwasaki et al. [7], later expanded upon by Benvensite et al. [2], and then further extended by Mosterman et al. [16] to create hyper-dense time, which distinguishes physical time, discontinuities that approximate fast physical dynamics, and sequences of logical events. A similar use of the hyper-reals is proposed by Barros [1]. These time sets that are derived from the hyper-reals have a richer structure than superdense time requires, which may make them particularly attractive for models that have continuous dynamics. It is likely that some new simulation capabilities are inherent in the hyper-reals, which are not properly captured by reducing them to a minimalist structure satisfying P1–P5.

Nonetheless, for many computations, we can treat these hyper-real time bases as being encoded in the set $\mathbb{R} \times \mathbb{Z} \times \mathbb{Z}$. The time (t, r, k) in this set maps to the hyper-real $t + r\epsilon$ and integer k. Ordering of triples is lexicographic. Forward motion in time is modeled by

$$(t, r, k) + (\Delta t, \Delta r, \Delta k) = \begin{cases} (t + \Delta t, r + \Delta r, \Delta k) & \Delta t \neq 0 \text{ or } \Delta r \neq 0 \\ (t, r, k + \Delta k) & \text{otherwise,} \end{cases}$$

and the length of an interval is

$$\ell[(t_1, r_1, k_1), (t_2, r_2, k_2)) = \begin{cases} (t_2 - t_1, r_2 - r_1, k_2) & (t_2, r_2) \neq (t_1, r_1) \\ (0, 0, k_2 - k_1) & (t_2, r_2) = (t_1, r_1). \end{cases}$$

Proofs that these satisfy P1–P5 with $(0, 0, 0)$ as zero and $S((t, r, k)) = (t, r, k + 1)$ are essentially identical to those for the superdense time in Section 4.1.

4.3 A Novel Form of Time

The first example of superdense time given in Section 2 can be restated in terms of $+$, ℓ, and S. Time pairs are from the set $\mathbb{R} \times \mathbb{Z}$. Advancement through time and the lengths of intervals are defined by

$$(t_1, k_1) + (t_2, k_2) = (t_1 + t_2, k_1 + k_2)$$
$$\ell[(t_1, k_1), (t_2, k_2)) = (t_2 - t_1, k_2 - k_1).$$

Zero for this time base is $(0, 0)$ and $S((t, c)) = (t, c + 1)$. Properties P1–P5 follow immediately from their satisfaction by $\mathbb{R}$ and $\mathbb{Z}$, respectively. This model of time also satisfies the requirements given by Maler, Manna, and Pnueli [14].

4.4 Another Novel Form of Time

Time points are drawn from the set of tuples $\bigcup_{k=0}^{\infty} \mathbb{R}^k \times \mathbb{Z}$. Each time point has the form $(t_1, t_2, \ldots, t_k, n)$, and these are ordered first by the number of elements k and then in lexicographic order. So, for example, $(\pi, 1/2, 3) > (\pi/2, 1/2, 3) > (\pi, 2) > (1, 3) > (4) > (1)$. Addition is done by adding elements that exist in both tuples and then appending new elements. Specifically,

$$(t_1, \ldots, t_n, a) + (h_1, \ldots, h_m, b) = \begin{cases} (t_1 + h_1, \ldots, t_m + h_m, t_{m+1}, \ldots, t_n, a + b) & m < n \\ (t_1 + h_1, \ldots, t_n + h_n, h_{n+1}, \ldots, h_m, a + b) & m \geq n. \end{cases}$$

The length of an interval is

$$\ell[(t_1, \ldots, t_n, a), (h_1, \ldots, h_m, b)) = (h_1 - t_1, \ldots, h_n - t_n, h_{n+1}, \ldots, h_m, b - a),$$

which is well-defined because the interval exists only if $n \leq m$. The zero element is (0) and $S((t_1, \ldots, t_k, a)) = (t_1, \ldots, t_k, a + 1)$. Properties P1–P3 and P5 are trivially true, as is P4.1. To confirm P4.2,

$$\begin{aligned} (t_1, \ldots, t_n, a) + \ell[(t_1, \ldots, t_n, a),(h_1, \ldots, h_m, b)) \\ &= (t_1, \ldots, t_n, a) + (h_1 - t_1, \ldots, h_n - t_n, h_{n+1}, \ldots, h_n, b - a) \\ &= (h_1, \ldots, h_n, h_{n+1}, \ldots, h_n, b). \end{aligned}$$

This form of superdense time or something similar may be useful to encode multi-scale timestamps, such as proposed by Goldy et al. [6]. In such a scheme, each element of the tuple would represent a distinct time scale (e.g., seconds, minutes, hours, days). How to do so may be an interesting question for future research.

5 CONCLUSIONS

The proposed structure for superdense time is a starting point for understanding how time is used in simulation models, and the value of an axiomatic approach is displayed in Section 4 where superdense schemes proposed in dissimilar contexts and, in some cases, decades apart, nonetheless share an overarching form. Future work in this direction will almost certainly refine or replace our selection criteria, which is that the structure of time be sufficient for defining a state transition function. It is plausible, even likely, that P1–P5 are stronger than needed, and so weaker criteria can be found that are necessary and sufficient. Otherwise, a proof that P1–P5 are necessary remains to be discovered.

Models that have continuous dynamics were not considered in our analysis. A possible method for including them could be to treat the state s as a location in a hybrid automaton; see, e.g., Ref. [23]. The lifetime function would give the first moment that a transition condition is satisfied; the function ρ then effects the transition to a new location. An exploration of this problem is a topic for future research. Similarly, we have not attempted to account for computational techniques that exploit the possibility of obtaining a correct result while executing events out of timestamp order, e.g., as described by Quaglia and Baldoni [22].

The form of time discussed in Section 4.1 has a potentially useful interpretation in the context of agent-based models. The integer part of time becomes the order in which agents are updated at a given real time t. Viewed in terms of Equation (3), the lifetime of an agent's state is the real step size h and the position k of that agent in the update order. By encoding the update order for agents into the time base, it may be possible to realize agent-based models, as expressed with tools like MASON and Repast [13, 18], within a discrete event simulation. Very sophisticated parallel discrete event simulation tools could enable gigantic MASON or Repast models. The encoding of agent order in time also offers an appealing foundation for positioning agent-based models within general systems theory; see, e.g., the proposal by Müller [17].

APPENDIX

A PROOFS OF PROPOSITIONS IN SECTION 3

PROPOSITION 1. *The number 0 is unique.*

PROOF. Suppose there exists $h \neq 0$ such that $h + t = t + h = t$. Then, $h + 0 = 0 + h = 0$ and $h + 0 = 0 + h = h$, contradicting our assertion that $h \neq 0$. □

PROPOSITION 2. *The successor $S(t)$ of t is unique.*

PROOF. Let $a > t$, $a \neq S(t)$ be such that $[t, a)$ contains only t. Then, $\ell[t, a) = \ell[t, S(t))$ and $t + \ell[t, a) = t + \ell[t, S(t))$. Using P4.2, we conclude $a = S(t)$, contradicting our assertion that $a \neq S(t)$. □

PROPOSITION 3. $\ell[t_0, t_f) = 0$ *if, and only if,* $t_0 = t_f$.

PROOF. Suppose $t_0 = t_f = t$. From P4.2, $t + \ell[t, t) = t$. Hence, $\ell[t, t) = 0$ because 0 is unique. Now suppose $\ell[t_0, t_f) = 0$, but $t_0 < t_f$. Again, from P4.2, $t_0 + \ell[t_0, t_f) = t_f \neq t_0$ contradicting the assertion $\ell[t_0, t_f) = 0$. □

PROPOSITION 4. *If $b \geq 0$, $c \geq 0$, and $a + b = a + c$, then $b = c$.*

PROOF. Suppose $b \neq c$. Time is a totally ordered set and so, without loss of generality, let $b > c$. P2 requires that $a + b > a + c$, and so $a + b \neq a + c$. □

PROPOSITION 5. $\ell[t_0, t_f) = \ell[t_0, t_m) + \ell[t_m, t_f)$.

PROOF. The following are true.

$$
\begin{aligned}
&t_0 + \ell[t_0, t_m) = t_m, \\
&t_0 + \ell[t_0, t_f) = t_f, \\
&t_m + \ell[t_m, t_f) = (t_0 + \ell[t_0, t_m)) + \ell[t_m, t_f) = t_f, \\
&(t_0 + \ell[t_0, t_m)) + \ell[t_m, t_f) = t_0 + (\ell[t_0, t_m) + \ell[t_m, t_f)) = t_f, \text{ and so} \\
&t_0 + (\ell[t_0, t_m) + \ell[t_m, t_f)) = t_0 + \ell[t_0, t_f).
\end{aligned}
$$

The proposition follows from P4.1 and Proposition 4. □

PROPOSITION 6. $\ell[t, t + h) = h$.

PROOF. Let $t + \ell[t, t + h) = t'$. Using P4.2, $t + \ell[t, t + h) = t + h$; hence, $t + h = t'$, and the proposition follows immediately. □

PROPOSITION 7. $e + \ell[t, t_a) = ta(s)$ *if and only if* $t + \ell[e, ta(s)) = t_a$.

PROOF. Suppose $e + \ell[t, t_a) = ta(s)$. Using Proposition 6

$$
\begin{aligned}
t + \ell[e, ta(s)) &= t + \ell[e, e + \ell[t, t_a)) \\
&= t + \ell[t, t_a) \\
&= t_a.
\end{aligned}
$$

In the other direction,

$$
\begin{aligned}
e + \ell[t, t_a) &= e + \ell[t, t + \ell[e, ta(s))) \\
&= e + \ell[e, ta(s)) \\
&= ta(s).
\end{aligned}
$$

□

B PROOFS FOR SECTION 4.1

Proposition 8 (P1). $(t,k) + (0,0) = (0,0) + (t,k) = (t,k)$

Proof. $(t,k) + (0,0) = (t,k+0) = (t,k)$. If $t \neq 0$, then $(0,0) + (t,k) = (0+t,k) = (t,k)$. If $t = 0$, then $(0,0) + (0,k) = (0,0+k) = (0,k)$. □

Proposition 9 (P2). *If* $(t_2,k_2) > (t_3,k_3) \geq 0$, *then* $(t_1,k_1) + (t_2,k_2) > (t_1,k_1) + (t_3,k_3)$.

Proof. Either $t_2 > t_3$ or $t_2 = t_3 = t$, and $k_2 > k_3$. If $t_2 = t_3$, then $t_1 + t_2 > t_1 + t_3$, and the proposition follows. Otherwise, $(t_1,k_1) + (t_2,k_2) = (t_1 + t_2, k_2)$, $(t_1,k_1) + (t_2,k_3) = (t_1 + t_2, k_3)$, and because $k_2 > k_3$, the proposition follows. □

Proposition 10 (P3). *If* $t_2 \geq 0$ *and* $t_3 \geq 0$, *then* $((t_1,k_1) + (t_2,k_2)) + (t_3,k_3) = (t_1,k_1) + ((t_2,k_2) + (t_3,k_3))$

Proof. *Case 1:* $t_2 = t_3 = 0$.

$$((t_1,k_1) + (0,k_2)) + (0,k_3) = (t_1,k_1+k_2) + (0,k_3) = (t_1,k_1+k_2+k_3).$$
$$(t_1,k_1) + ((0,k_2) + (0,k_3)) = (t_1,k_1) + (0,k_2+k_3) = (t_1,k_1+k_2+k_3).$$

Case 2: $t_2 = 0$ and $t_3 > 0$.

$$((t_1,k_1) + (0,k_2)) + (t_3,k_3) = (t_1,k_1+k_2) + (t_3,k_3) = (t_1+t_3,k_3).$$
$$(t_1,k_1) + ((0,k_2) + (t_3,k_3)) = (t_1,k_1) + (t_3,k_3) = (t_1+t_3,k_3).$$

Case 3: $t_2 > 0$ and $t_3 = 0$.

$$((t_1,k_1) + (t_2,k_2)) + (0,k_3) = (t_1+t_2,k_2) + (0,k_3) = (t_1+t_2,k_2+k_3).$$
$$(t_1,k_1) + ((t_2,k_2) + (0,k_3)) = (t_1,k_1) + (t_2,k_2+k_3) = (t_1+t_2,k_2+k_3).$$

Case 4: $t_2 > 0$ and $t_3 > 0$.

$$((t_1,k_1) + (t_2,k_2)) + (t_3,k_3) = (t_1+t_2,k_2) + (t_3,k_3) = (t_1+t_2+t_3,k_3).$$
$$(t_1,k_1) + ((t_2,k_2) + (t_3,k_3)) = (t_1,k_1) + (t_2+t_3,k_3) = (t_1+t_2+t_3,k_3).$$ □

The above proof does not work if t_2 or t_3 are less than zero, illustrating that + only advances forward in time. For example, let $t_2 = -t_3$; then,

$$((t_1,k_1) + (-t_3,k_2)) + (t_3,k_3) = (t_1-t_3,k_2) + (t_3,k_3) = (t_1,k_3), \text{ and}$$
$$(t_1,k_1) + ((-t_3,k_2) + (t_3,k_3)) = (t_1,k_1) + (0,k_3) = (t_1,k_1+k_3).$$

Proposition 11 (P4). *Let* $(t_1,k_1) > (t_2,k_2)$. *There exists* (h,k) *such that* $(t_1,k_1) + (h,k) = (t_2,k_2)$.

Proof. If $t_1 = t_2$, then $(h,k) = (0,k_2-k_1)$ satisfies the equality. Otherwise, $t_1 > t_2$ and $(h,k) = (t_2-t_1,k_2)$ satisfies the equality. □

Proposition 12 (P5). *The interval* $[(t,c), S((t,c)))$ *contains exactly* (t,c).

Proof. $S((t,c)) = (t, c+1)$. Let $(t', k) \in [(t,c),(t,c+1))$. Then, $t' = t$ and $k \in [c, c+1)$. Because k, c are integers, we must have $k = c$, and so $(t', k) = (t, c)$. □

REFERENCES

[1] F. J. Barros. 2016. On the representation of time in modeling simulation. In *Proceedings of the 2016 Winter Simulation Conference (WSC)*. IEEE, Piscataway, NJ, 1571–1582.

[2] Albert Benveniste, Timothy Bourke, Benoît Caillaud, and Marc Pouzet. 2012. Non-standard semantics of hybrid systems modelers. *Journal of Computer and System Sciences* 78, 3 (2012), 877–910.

[3] Fabio Cremona, Marten Lohstroh, David Broman, Edward A. Lee, Michael Masin, and Stavros Tripakis. 2019. Hybrid co-simulation: it's about time. *Software & Systems Modeling* 18, 3 (1 Jun 2019), 1655–1679.

[4] Richard M. Fujimoto. 1999. *Parallel and Distribution Simulation Systems* (1st ed.). John Wiley & Sons, Inc., New York, NY.

[5] Martin J. Gander. 2015. 50 years of time parallel time integration. In *Multiple Shooting and Time Domain Decomposition Methods*, Thomas Carraro, Michael Geiger, Stefan Körkel, and Rolf Rannacher (Eds.). Springer International Publishing, Cham, 69–113.

[6] Rhys Goldstein, Azam Khan, Olivier Dalle, and Gabriel Wainer. 2018. Multiscale representation of simulated time. *Simulation* 94, 6 (2018), 519–558.

[7] Yumi Iwasaki, Adam Farquhar, Vijay Saraswat, Daniel Bobrow, and Vineet. 1995. Modeling time in hybrid systems: How fast is instantaneous? In *International Workshop on Qualitative Reasoning*. Morgan Kaufmann, 1773–1780.

[8] David Jefferson. 1985. Virtual time. *ACM Transactions on Programming Languages and Systems* 7, 3 (July 1985), 404–425.

[9] Ki Hyung Kim, Yeong Rak Seong, Tag Gon Kim, and Kyu Ho Park. 1997. Ordering of simultaneous events in distributed DEVS simulation. *Simulation Practice and Theory* 5, 3 (1997), 253–268.

[10] George J. Klir. 1969. *An Approach to General Systems Theory*. Van Nostrand Reinhold Co., New York.

[11] Edward A. Lee. 2014. Constructive models of discrete and continuous physical phenomena. *IEEE Access* 2 (2014), 797–821.

[12] Edward A. Lee and Haiyang Zheng. 2005. Operational semantics of hybrid systems. In *Proceedings of the 8th International Workshop on Hybrid Systems: Computation and Control (HSCC'05)*. Springer Berlin, 25–53.

[13] Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Balan. 2005. MASON: A multi-agent simulation environment. *Simulation: Transactions of the Society for Modeling and Simulation International* 82, 7 (2005), 517–527.

[14] Oded Maler, Zohar Manna, and Amir Pnueli. 1992. From timed to hybrid systems. In *Proceedings of the Real-Time: Theory in Practice, REX Workshop*. Springer-Verlag, London, 447–484.

[15] Mihailo D. Mesarovic and Yasuhiko Takahara. 1989. *Abstract Systems Theory*. Springer, Berlin.

[16] Pieter J. Mosterman, Gabor Simko, Justyna Zander, and Zhi Han. 2014. A hyperdense semantic domain for hybrid dynamic systems to model different classes of discontinuities. In *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control (HSCC'14)*. ACM, New York, 83–92.

[17] Jean-Pierre Müller. 2009. Towards a formal semantics of event-based multi-agent simulations. In *Multi-Agent-Based Simulation IX: International Workshop, MABS 2008, Revised Selected Papers*. Springer Berlin, 110–126.

[18] M. J. North, N. T. Collier, J. Ozik, E. Tatara, M. Altaweel, C. M. Macal, M. Bragen, and P. Sydelko. 2013. Complex adaptive systems modeling with repast simphony. *Complex Adaptive Systems Modeling* 1, 3 (2013), 26.

[19] James Nutaro and Hessam Sarjoughian. 2004. Design of distributed simulation environments: A unified system-theoretic and logical processes approach. *Simulation* 80, 11 (Nov. 2004), 577–589.

[20] James J. Nutaro. 2011. *Building Software for Simulation: Theory and Algorithms with Applications in C++*. Wiley, Hoboken, NJ.

[21] Bernard P. Zeigler Herbert Praehofer and Tag Gon Kim. 2000. *Theory of Modeling and Simulation, 2nd Edition*. Academic Press, San Diego, CA.

[22] Francesco Quaglia and Roberto Baldoni. 1999. Exploiting intra-object dependencies in parallel simulation. *Information Processing Letters* 70, 3 (1999), 119–125.

[23] Jean-François Raskin. 2005. An introduction to hybrid automata. In *Handbook of Networked and Embedded Control Systems*, Dimitrios Hristu-Varsakelis and William S. Levine (Eds.). Birkhäuser Boston, Boston, MA, 491–517.

[24] R. Ronngren and M. Liljenstam. 1999. On event ordering in parallel discrete event simulation. In *Proceedings of the 13th Workshop on Parallel and Distributed Simulation (PADS'99)*. IEEE Computer Society, Washington, D.C., 38–45.

[25] Hessam S. Sarjoughian and Savitha Sundaramoorthi. 2015. Superdense time trajectories for DEVS simulation models. In *Proceedings of the Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium (DEVS'15)*. Society for Computer Simulation International, San Diego, CA, 249–256.

[26] A. Wayne Wymore. 1967. *A Mathematical Theory of Systems Engineering: The Elements*. Krieger, New York.
[27] Bernard P. Zeigler. 1976. *Theory of Modelling and Simulation*. Wiley, New York.
[28] Haiyang Zheng, Edward A. Lee, and Aaron D. Ames. 2006. Beyond zeno: Get on with it! In *Hybrid Systems: Computation and Control*, João P. Hespanha and Ashish Tiwari (Eds.). Springer Berlin, 568–582.

Received August 2018; revised April 2019; accepted January 2020