Checking Timed Büchi Automata Emptiness Using the Local-Time Semantics

Frédéric Herbreteau
Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400, Talence, France

B. Srivathsan
Chennai Mathematical Institute, India
CNRS IRL 2000, ReLaX, Chennai, India

Igor Walukiewicz
Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400, Talence, France

Abstract

We study the Büchi non-emptiness problem for networks of timed automata. Standard solutions consider the network as a monolithic timed automaton obtained as a synchronized product and build its zone graph on-the-fly under the classical global-time semantics. In the global-time semantics, all processes are assumed to have a common global timeline.

Bengtsson et al. in 1998 have proposed a local-time semantics where each process in the network moves independently according to a local timeline, and processes synchronize their timelines when they do a common action. It has been shown that the local-time semantics is equivalent to the global-time semantics for finite runs, and hence can be used for checking reachability. The local-time semantics allows computation of a local zone graph which has good independence properties and is amenable to partial-order methods. Hence local zone graphs are able to better tackle the state-space explosion due to concurrency.

In this work, we extend the results to the Büchi setting. We propose a local zone graph computation that can be coupled with a partial-order method, to solve the Büchi non-emptiness problem in timed networks. In the process, we develop a theory of regions for the local-time semantics.

2012 ACM Subject Classification Theory of computation → Verification by model checking

Keywords and phrases Timed Büchi automata, local-time semantics, zones, abstraction, partial-order reduction

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2022.12

Funding *Frédéric Herbreteau*: ANR project Ticktak (ANR-18-CE40-0015)
Igor Walukiewicz: ANR project Ticktac (ANR-18-CE40-0015)

1 Introduction

Timed automata [2] are a popular model for real-time systems. Typically, systems are modeled as a *network of timed automata* that communicate with each other via synchronizing actions. We are interested in verifying Büchi properties of such models: does there exist a run of the network that executes transitions from a given set infinitely often? This is called the Büchi non-emptiness problem. Model checking LTL specifications can be reduced to the Büchi non-emptiness problem. Moreover, verifying Büchi properties can be useful in trouble-shooting the model under consideration, for example, a typo in the benchmark CSMA/CD protocol model was discovered through a Büchi property verification [18]. Recent works go even further and consider synthesis questions for Büchi timed automata [3, 8].

Existing algorithms for the Büchi non-emptiness problem view the network as a single timed automaton obtained by a synchronized product, and build the so-called zone graph of this product automaton on-the-fly [29, 28, 25, 23, 18]. The main challenge lies in guaranteeing

33rd International Conference on Concurrency Theory (CONCUR 2022).
Editors: Bartek Klin, Sławomir Lasota, and Anca Muscholl; Article No. 12; pp. 12:1–12:24
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the termination of the zone graph computation. This is achieved through a *finite abstraction* of the zone graph that preserves the Büchi property. The aim of course is to get as small a graph as possible. This has been the central subject of study in timed automata verification [11, 7, 4, 21, 20]. As a matter of fact, new abstraction methods are usually first studied in the context of reachability verification and then lifted to the Büchi setting. In this paper, we continue this trend by extending a recent work on abstractions based on the *local-time semantics* [15], to the Büchi non-emptiness problem.

The local-time semantics for timed automata networks was proposed by Bengtsson *et al.* [5] with the aim of applying partial-order reduction methods that can exploit the network representation of the model to build a smaller zone graph. In the local-time semantics, each process in the network moves independently according to its local timeline which contrasts with the standard semantics where time elapses synchronously in all the processes. When processes perform a shared action, they synchronize their local timelines. This semantics gives good independence properties: for instance, if a and b are actions performed by processes P_a and P_b, an execution $(a, 2)(b, 1)$ means a happens when the local time of P_a is 2 and b happens when local time of P_b is 1. There is no "happens-before" between $(a, 2)$ and $(b, 1)$. The local-time semantics leads to a local-zone graph computation in which performing ab or ba from a local-zone leads to the same local-zone. This diamond property is essential for applying partial-order reduction methods. In [15] we have proposed abstractions for the local zone graph that can be coupled with partial-order methods to solve the reachability problem. Extending these methods to the Büchi non-emptiness problem, poses certain technical questions and requires some adaptations of the setting. We settle these issues here.

1.1 Contributions

The first question is whether the local-time semantics is sound for Büchi runs: does existence of an infinite Büchi run in the local-time semantics ensure existence of an infinite Büchi run in the usual global-time semantics? Surprisingly, we answer in the affirmative without any extra assumption. This said, let us remark that this is not true if one allows invariants in states. The solution is significantly different from the soundness argument used for reachability where the last valuation of a local run needs to be synchronized.

The next question is whether the local-zone graph is sound for Büchi runs: does an infinite Büchi run in the local-zone graph ensure existence of an infinite Büchi run in the local-time semantics. For every finite prefix, we can get a finite run in the local-time semantics. But the question of whether these prefixes can be glued together to form an infinite run is non-trivial. The same question arises in the global-time semantics as well, and there the solution makes a crucial use of Alur-Dill regions [2]. For the local-time semantics, there is no known notion of a region equivalence. We have shown [15] that in general, there can be no finite time-abstract bisimulation for the local-time semantics and proposed to restrict attention to a class of networks called bounded spread networks. Every network can be converted to a bounded spread network at the cost of reducing concurrency. In this work, we develop a finite region equivalence over the local-time semantics for bounded spread networks.

Finally, we prove that the combination of the abstraction from [15] and partial-order reduction can be suitably applied on the local-zone graph to solve the Büchi non-emptiness problem. For the argument to work we need to assume that the network is deterministic. This is usually not a strong assumption, as a network can be made deterministic by renaming actions. The proof of correctness appropriately combines the guarantees known over finite runs and the region machinery developed above.

We remark that we do not propose here a concrete POR method. Instead, we consider a POR method as an oracle that assigns a subset of edges to be explored from each node of the local-zone graph. Our work can be seen as a theoretical development that allows to plug in any POR method which is correct for the Büchi problem on untimed networks, to timed networks. Most recent works in the POR literature consider a very special case where every process is acyclic and has at most one outgoing action in every state [1, 33, 9, 22]. As a further work we would like to have equally efficient methods for more general settings as considered in works on stubborn/ample/persistent sets [30, 27, 13, 32].

1.2 Related Work

The early abstraction methods studied for timed automata depended on the maximum constant appearing in the automaton [11, 7]. These abstractions were extended to Büchi runs by Tripakis *et al.* [29, 28]. Later, superior abstractions were proposed based on the maximum constant L occurring in lower bound guards ($x > c, x \geq c$) and the maximum constant U occurring in upper bound guards ($x < c, x \leq c$) [4]. Li [25] showed that these LU abstractions can be used to solve Büchi non-emptiness problem.

An abstraction method comes with an operator $\mathfrak{a}$ that can be applied on zones Z. For reachability, it is enough to check $\mathfrak{a}(Z) \subseteq \mathfrak{a}(Z')$ (called a subsumption) to discard further exploration from Z and continue from Z'. On the other hand, for Büchi non-emptiness, we need to check for equality $\mathfrak{a}(Z) = \mathfrak{a}(Z')$. Subsumptions are instrumental in reducing the size of the graph obtained. For the Büchi problem, a restricted usage of subsumption is possible [23, 18]. However, the gains due to this restricted subsumption are less pronounced in the Büchi setting as compared to the gains achieved for reachability. There is even a concrete argument to support this statement: deciding Büchi non-emptiness starting from a zone graph with subsumption is PSpace-hard [18]. The moral is that graphs computed for the Büchi problem are in general expected to be much larger than the reachability counterparts. In this situation it is even more interesting to use POR to reduce their size.

POR techniques have been applied for the reachability problem, but over the zone graph computed using the standard global-time semantics. There is much less independence in the global-time zone graph and the POR method needs to be restricted accordingly. Therefore some approaches limit the POR methods to parts where independent actions occur in zero time [26, 24, 6], and other approaches discover which actions remain independent either statically [10] or dynamically [17].

1.3 Outline of the Paper

In the next section we introduce networks of timed automata and their local-time semantics. Standard global-time semantics is a special case of the local-time semantics. We define the Büchi non-emptiness problem over the global-time semantics. In Section 3 we show that for the Büchi non-emptiness problem it is sound to use the local-time semantics instead of the global one. We also recall local-zones, local-zone graphs and their properties from [15]. In Section 4, we develop a theory of regions for the local-time semantics. We employ the concept of a bounded spread network from [15], and show that for such networks the number of regions is finite if we have a bound on the constants used in the guards of the transitions. Finally, in Section 5 we recall the abstraction operation $\mathfrak{a}^{D}_{\preccurlyeq_{LU}}$ from [15], introduce an abstract notion of a POR method based on a source function, and show how to obtain a finite abstract local-zone graph where we can use a POR method while retaining soundness and completeness for Büchi runs. Missing proofs are presented in appendices.

2 Preliminaries

We write $\mathbb{R}$, $\mathbb{R}_{\geq 0}$ and $\mathbb{N}$ for the set of reals, non-negative reals and natural numbers, respectively. We will write 2^S for the power set of a set S. Let X be a set of real valued variables. A constraint over X is described by the grammar: $\phi := x\#c \mid \phi \wedge \phi$, where $x \in X$, $c \in \mathbb{N}$ and $\# \in \{<, \leq, =, >, \geq\}$. Let $\Phi(X)$ denote the set of all constraints over X.

A *network of timed automata* $\mathcal{N}$ is a tuple $\langle A_1, A_2, \ldots, A_k\rangle$ of k timed automata, each A_i is called a *process* or a *component* of the network. Let $Proc = \{1, \ldots, k\}$ denote the set of process identifiers. Process A_i is given by $(Q_i, q_i^{init}, \Sigma_i, X_i, \Delta_i)$ consisting of a finite set Q_i of states, an initial state $q_i^{init} \in Q_i$, a finite alphabet of actions Σ_i, a finite set of clocks X_i, and a finite set of transitions $\Delta_i \subseteq Q_i \times \Sigma_i \times \Phi(X_i) \times 2^{X_i} \times Q_i$.

Transitions in A_i are of the form (p, a, g, R, q) where p and q are the source and target of the transition, $a \in \Sigma_i$ is the action, $g \in \Phi(X_i)$ is a *guard* over local clocks X_i, and $R \subseteq X_i$ is the set of local clocks of X_i that are *reset* along the transition. We assume that $Q_i \cap Q_j = \emptyset$ and $X_i \cap X_j = \emptyset$ for all distinct pairs $i, j \in \{1, \ldots, k\}$. We define $\Sigma := \bigcup_{i=1}^{i=k} \Sigma_i$, $X := \bigcup_{i=1}^{i=k} X_i$ and $Q := \prod_{i=1}^{i=k} Q_i$. For $a \in \Sigma$, we write $dom(a) := \{i \in \{1, \ldots, k\} \mid a \in \Sigma_i\}$. For $q = (q_1, \ldots, q_k)$ in Q, we write $q(i)$ for q_i.

We say that $\mathcal{N}$ is *deterministic* if for every component A_i and for every action a, there is at most one local transition (p, a, g, R, q) from every local state $p \in Q_i$. We will assume deterministic networks in Section 5.

There are two ways to describe the semantics of a network: one in which all the components share a common timeline (global-time semantics), and another where each of them work with a local timeline (local-time semantics). We define the local-time semantics and view global-time semantics as a special case.

2.1 Local-Time Semantics

Fix a network $\mathcal{N} = \langle A_1, \ldots, A_k\rangle$ for the rest of the section. We assume that each A_i has a special clock t_i called the *reference clock* of process A_i. Intuitively, it represents the local time of process A_i. Let $T = \{t_1, \ldots, t_k\}$ denote the set of all reference clocks. A valuation $v : X \cup T \to \mathbb{R}$ is a function that maps each variable in $X \cup T$ to a real number under the condition that $v(t_i) \geq v(x_i)$ for all $x_i \in X_i$. The value $v(x_i)$ represents the local time at process A_i when x_i was last reset. This explains why we require $v(t_i) \geq v(x_i)$. The value of clock x_i is then obtained as $v(t_i) - v(x_i)$. This semantics that keeps reset time points instead of clock ages has previously been introduced in [12] in the global-time setting. In the rest of the document, we use the notation $v(x - y)$ for $v(x) - v(y)$. The semantics relies on two operations.

The first one is a *local-time elapse.* Given a valuation v, a delay $\delta_i \in \mathbb{R}$ and a process i, the valuation $v +_i \delta_i$ describes a *local delay* of δ_i units at process i. It is given by $(v +_i \delta_i)(t_i) = v(t_i) + \delta_i$ and $(v +_i \delta_i)(x) = v(x)$ for all other variables x. Notice that $((v +_i \delta_i) +_j \delta_j)$ is the same as $((v +_j \delta_j) +_i \delta_i)$: the order in which we sum the local delays does not matter. We extend this notion to a tuple $\Delta := (\delta_1, \ldots, \delta_k) \in \mathbb{R}^k_{\geq 0}$ of delays, one for each process: $v + \Delta$ is the valuation $v +_1 \delta_1 +_2 \delta_2 \cdots +_k \delta_k$.

The next operation is *clock reset.* In the local-time interpretation, resetting a clock $x_i \in X_i$ amounts to updating its value to the local-time of i given by t_i. Given valuation v, and a set of clocks $R \subseteq X$, we write $v[R]$ to be the valuation obtained as $(v[R])(x) = v(t_i)$ when $x \in R \cap X_i$ for some $i \in \{1, \ldots, k\}$ and $v[R](x) = v(x)$ otherwise. Valuation v is said to satisfy a constraint $x\#c$ for $x \in X_i$ if $v(t_i - x)\#c$. We write $v \models (x\#c)$ in this case. A valuation satisfies a conjunction of constraints $\phi_1 \wedge \phi_2$ if $v \models \phi_1$ and $v \models \phi_2$.

A *configuration* of a network is a pair (q, v) where $q \in Q$ and v is a valuation. Recall that we have defined Q to be the product of the local states Q_i. A valuation v is said to be *initial* if $v(x) = v(y)$ for all $x, y \in X \cup T$: all timelines are synchronized and the constraint $x = 0$ holds for every clock. A configuration (q, v) is initial if $q = (q_1^{init}, \dots, q_k^{init})$ and v is an initial valuation.

There are two kinds of transitions between configurations: local delays and action transitions. From a configuration (q, v) there is a local delay transition $(q, v) \xrightarrow{\Delta} (q, v + \Delta)$ for each $\Delta \in \mathbb{R}^k_{\geq 0}$. For each $b \in \Sigma$, a b-transition is a tuple $\{(q_i, b, g_i, R_i, q_i')\}_{i \in dom(b)}$ of local transitions one from each process in its domain. From (q, v), we have an action transition $(q, v) \xrightarrow{b} (q', v')$ if there exists a b-transition $\{(q_i, b, g_i, R_i, q_i')\}_{i \in dom(b)}$ such that:

- source states match: $q(i) = q_i$ for all $i \in dom(b)$,
- valuation satisfies guard: $v \models g_i$ for all $i \in dom(b)$,
- all processes in $dom(b)$ are synchronized: $v(t_i) = v(t_j)$ for all $i, j \in dom(b)$,
- resets are performed: $v' = v[\bigcup_{i \in dom(b)} R_i]$,
- target states are reached: $q'(i) = q_i'$ if $i \in dom(b)$ and $q'(i) = q(i)$ otherwise.

The important point is that when a common action is performed, the local times of all the processes in its domain are the same.

We will write $(q, v) \xrightarrow{\Delta, b} (q', v')$ for $(q, v) \xrightarrow{\Delta}\xrightarrow{b} (q', v')$, a local delay Δ transition followed by an action b. Observe that Δ is determined by v and v' because $\delta_p = v'(t_p) - v(t_p)$ where t_p is the reference clock of process p. A *run* is an infinite sequence $(q_0, v_0) \xrightarrow{\Delta_0, b_0} (q_1, v_1) \xrightarrow{\Delta_1, b_1} \cdots$ of transitions starting from an initial configuration (q_0, v_0). A finite run is defined similarly: $(q_0, v_0) \xrightarrow{\Delta_0, b_0} (q_1, v_1) \xrightarrow{\Delta_1, b_1} \cdots (q_n, v_n) \xrightarrow{\Delta_n} (q_n', v_n')$. Observe that finite runs have a final delay. We write $(q, v) \overset{\sigma}{\dashrightarrow} (q', v')$ to say that there is a finite run on a sequence of actions σ.

2.2 The Büchi Non-Emptiness Problem

Global-time semantics is a local-time semantics restricted to *synchronized valuations*. A valuation v is said to be *synchronized* if $v(t_p) = v(t_q)$ for all processes p, q. This implies that in every delay $(\delta_1, \dots, \delta_k)$ that is part of the global-time semantics, we have $\delta_1 = \delta_2 = \cdots = \delta_k$. The global-time semantics obtained this way is the usual semantics that is used in tools and studies that involve networks of timed automata. To make a distinction, we refer to runs in the local-time semantics as local runs and runs in the global-time semantics as global runs. Clearly, a global run is also a local run.

▶ **Definition 1** (Büchi non-emptiness problem). *Given a network $\mathcal{N}$ and a set of actions $F \subseteq \Sigma$, decide if $\mathcal{N}$ has a global run with infinitely many occurrences of actions from F.*

In the case of finite runs, the correspondence between local and global semantics is well-known. Indeed there is a local run to a configuration (q, v) with a synchronized valuation v if and only if there exists a global run to (q, v) that follows the same transitions, although in a different order. This is captured by the notion of independent actions and traces.

▶ **Definition 2** (Independence). *Two actions $a, b \in \Sigma$ are said to be independent if $dom(a) \cap dom(b) = \emptyset$. Two sequences of actions $u, w \in \Sigma^*$ are* trace equivalent, *$u \sim w$, if one of them can be obtained from the other by permuting adjacent independent actions.*

▶ **Lemma 3** ([16]). *Let v, v' be synchronized valuations, and let $(q, v) \overset{u}{\dashrightarrow} (q', v')$ be a local run. Then there exists a global run $(q, v) \overset{w}{\dashrightarrow} (q', v')$ such that $u \sim w$.*

▶ Remark 4. Putting Büchi conditions on transitions makes them trace invariant which is important if we want to do partial-order methods: if u has infinitely many actions from F, then so does every $w \sim u$, which may not be true with Büchi conditions on states.

Moreover, the local-time semantics enjoys a very nice property: any two independent actions commute as stated by the following lemma.

▶ **Lemma 5** (Diamond property). *[16] Let $a, b \in \Sigma$ with $dom(a) \cap dom(b) = \emptyset$. If $(q, v) \overset{ab}{\dashrightarrow} (q', v')$ then $(q, v) \overset{ba}{\dashrightarrow} (q', v')$.*

Observe that this commutation property does not hold in the global-time semantics since delays are synchronous in all processes. Our motivation for using local time over global time comes from the fact that local time allows to reorder the actions in a run, hence using partial-order reduction techniques becomes possible.

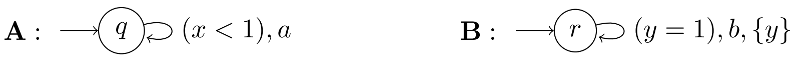
3 Büchi Runs in the Local-Time Semantics

In this section, we show that using the local-time semantics is sound for the Büchi emptiness problem. Hence the local-time semantics looks appealing as it enables to use partial-order reduction techniques. However, verification algorithms cannot work directly from the state-space of the network as it is uncountable. We will then introduce the local zone graph as a symbolic representation of the state-space of timed networks in the local time semantics.

3.1 The Local-Time Semantics is Sound

When state-reachability is considered the soundness of the local-time semantics follows from Lemma 3. However, it is based on the notion of independence, Definition 2, that is not adequate for infinite runs. The point is that for infinite runs we may need an infinite number of permutations to get w from u. The more general definition refers to partial orders defined by runs. These partial orders are often called traces. The other obstacle in repeating Lemma 3 is that the lemma refers to runs ending in synchronized valuations, while for infinite runs we do not have final valuations. For these two reasons the soundness argument is more involved than that of reachability.

We start with an intuition and an illustrating example. To get a global run from a local run, the natural idea would be to re-order the events based on the time-stamps. For example, if we have a finite sequence $(a, 2)(b, 1)(c, 3)(a, 2.5)(b, 1.5)$ where the number represents the local time when the action occurred, then we get a reordered sequence $(b, 1)(b, 1.5)(a, 2)(a, 2.5)(c, 3)$. It can then be argued that this sequence has a global run where each clock can be delayed up to the next action to be read. For the case of infinite runs, consider the following example that consists of two processes A and B.



Consider an infinite local run:

$$(a, \frac{1}{2})\ (b, 1)\ (a, \frac{1}{2} + \frac{1}{2^2})\ (b, 2)\ \cdots\ (a, \frac{1}{2} + \cdots + \frac{1}{2^i})\ (b, i)\ \cdots$$

Notice that all the a actions happen before global time 1, and b actions start from 1. Therefore, a re-ordering of the events based on the time-stamps gives an infinite sequence of a's "followed by" an infinite sequence of b's. This does not correspond to a global run. However, if a was accepting, we have a global run $(a, \frac{1}{2})(a, \frac{1}{2} + \frac{1}{2^2}) \cdots$ that is accepting. Similarly, if b was accepting, we have a global run $(b,1)(b,2)\cdots$. In each of these runs, the other process plays no role except for elapsing time. Observe that the absence of state invariants is crucial here as the infinite sequence of b's would not be feasible in the global time semantics if state q of process A had invariant $(x<1)$.

In general, when we have a local run, we can reorder it based on timestamps to get a sequence which may have a block of infinite events (say between 0 and 1), followed by another block of events (say between 1 and 2), and so on. However, the processes that participate infinitely often in some block do not participate in the future blocks. This means there can be only finitely many such blocks. Moreover, if the original run is Büchi accepting, then there is some block (along with some events in blocks before it) that gives a global run which is Büchi accepting. We formalize this intuition below.

For an infinite sequence $w\in\Sigma^\omega$ we define a *trace of* w as a labelled partial order $T_\sigma=\langle\mathbb{N},\lambda_\sigma,\trianglelefteq_\sigma\rangle$ where $\lambda_\sigma(i)=w_i$, and $\trianglelefteq_\sigma$ the smallest transitive relation such that $i\trianglelefteq_\sigma j$ if $dom(w_i)\cap dom(w_j)\neq\emptyset$ and $i\leq j$. Observe that $\trianglelefteq_\sigma$ is reflexive.

▶ **Definition 6.** *Two infinite runs $u,w\in\Sigma^\omega$ are* trace equivalent*, $u\sim w$, if the traces T_u and T_w are isomorphic*

We extend this notion to runs. Consider a local run

$$\sigma=(q_0,v_0)\xrightarrow{\Delta_0}(q_0,v_0')\xrightarrow{b_0}(q_1,v_1)\xrightarrow{\Delta_1}(q_1,v_1')\xrightarrow{b_1}(q_2,v_2)\xrightarrow{\Delta_2}\cdots$$

A *trace of* σ is a labelled partial order $T_\sigma=\langle\mathbb{N},\lambda_\sigma,\trianglelefteq_\sigma\rangle$ where $\lambda_\sigma(i)=(q_i,v_i',b_i)$, and as before $\trianglelefteq_\sigma$ the smallest transitive relation such that $i\trianglelefteq_\sigma j$ if $dom(b_i)\cap dom(b_j)\neq\emptyset$ and $i\leq j$. Observe that we take valuation v_i' in the label as it ensures that b_i is enabled in (q_i,v_i').

To connect local and global time semantics we will rearrange actions depending on their local time. We use $\theta_\sigma(i)$ for the local time of execution of action b_i in the trace σ; it is given by $\theta_\sigma(i)=v_i'(t_p)$ where $p\in dom(b_i)$. Recall that by definition of an action transition, $v_i'(p)=v_i'(q)$ for all $p,q,\in dom(b_i)$.

▶ **Lemma 7.** *If $i\trianglelefteq_\sigma j$ then $\theta_\sigma(i)\leq\theta_\sigma(j)$.*

A *trace prefix* T' of a trace T_σ is a restriction of T_σ to some $\trianglelefteq_\sigma$-downward closed subset of $\mathbb{N}$. A *linearization* of a trace T_σ is a bijection $f:\mathbb{N}\to\mathbb{N}$ such that if $f(i)\trianglelefteq_\sigma f(j)$ then $i\leq j$. This means that the order of elements in a linearization should respect the order in T_σ. Observe that the sequence given by f should use all elements of T_σ, because f is bijective. The next lemma says that every linearization yields a local run.

▶ **Lemma 8.** *For every T' a prefix of T_σ, every linearization f of T' gives a local run.*

The next result states the desired soundness property of the local-time semantics with respect to the global semantics. As every global run is a local run, one direction of the proposition is easy. The other side consists in finding a prefix of T_σ from which we can build a global run as explained in the example above.

▶ **Proposition 9.** *Consider a network of timed automata $\mathcal{N}$. There is a Büchi local run of $\mathcal{N}$ iff there is a Büchi global run of $\mathcal{N}$.*

3.2 Local-Zone Graphs

Thanks to Proposition 9, we know that we can safely use the local-time semantics to detect Büchi runs. However, we cannot solve the Büchi non-emptiness problem by an exploration of the state-space of timed automata as it is uncountable. Algorithms for timed automata work with sets of configurations sharing the same discrete state. We copy this approach to our setting. To start off, we lift operations on individual valuations to sets of valuations. For a set of local valuations W, define:

- local-elapse$(W) := \{v + \Delta \mid v \in W,\ \Delta \in \mathbb{R}^k_{\geq 0}\}$,
- $W[R] := \{v[R] \mid v \in W\}$, for a set of clocks $R \subseteq X$.
- $W \cap g := \{v \mid v \vDash g\}$ for a guard g.

Next, the transition relation on configurations can be lifted to sets. We write $(q, W) \xRightarrow{b} (q', W')$ when there exists a b-transition $\{(q_i, b, g_i, R_i, q_i')\}_{i \in dom(b)}$ such that:

- source states match: $q(i) = q_i$ for all $i \in dom(b)$,
- target states are reached: $q'(i) = q_i'$ for all $i \in dom(b)$ and $q'(i) = q(i)$ otherwise,
- $W' =$ local-elapse(W_2) where $W_2 = W_1[\bigcup_{p \in dom(b)} R_p]$ with $W_1 = W \cap (\bigwedge_{p \in dom(b)} g_p \wedge \bigwedge\{t_p = t_q \mid p, q \in dom(b)\})$, and W' is not empty.

We will call $\xRightarrow{b}$ a symbolic transition. We write $(q, W) \xRightarrow{b_1 \dots b_n} (q_n, W_n)$ if there is a sequence of symbolic transitions $(q, W) \xRightarrow{b_1} (q_1, W_1) \cdots \xRightarrow{b_n} (q_n, W_n)$.

Local-zones are special sets of valuations that occur naturally while computing the reachable configurations. A local-zone is a set of valuations described by a conjunction of constraints of the form $x - y \# c$ where $x, y \in X \cup T$, $c \in \mathbb{Z}$ and $\# \in \{<, \leq\}$. Local-zones can be efficiently represented using Difference Bound Matrices (DBMs). It can be shown that for a local-zone Z, the sets local-elapse(Z), $Z[R]$ and $Z \cap g$ (intersection with guard) are local-zones [5, 16]. This leads to the definition of a local-zone graph that captures the reachable configurations of a network.

▶ **Definition 10** (Local-zone graph $\mathrm{LZG}(\mathcal{N})$). *The local-zone graph* $\mathrm{LZG}(\mathcal{N})$ *of a network* $\mathcal{N}$ *is a transition system whose nodes are of the form* (q, Z) *where* q *is a state of the network, and* Z *is a local-zone. The initial node is* (q_0, Z_0) *with* $Z_0 =$ local-elapse(V_0) *where* V_0 *is the set of initial valuations (given by the local-zone* $\wedge_{x,y \in X \cup T} x - y = 0$*) and* $q_0 = (q_1^{init}, \dots, q_k^{init})$*. The transitions are given by the symbolic transition relation* $(q, Z) \xRightarrow{b} (q', Z')$*.*

The next lemma relates transitions over valuations and zones. Proof follows from the definition of symbolic transitions. If $Z =$ local-elapse(Z) then Z is *time-elapsed.*

▶ **Lemma 11.** *For every network of timed automata and every action b:*

pre-property: *If* $(q, v) \overset{b}{\dashrightarrow} (q', v')$ *and* $v \in Z$ *for some time-elapsed local-zone* Z *then* $(q, Z) \xRightarrow{b} (q', Z')$ *and* $v' \in Z'$ *for some local-zone* Z'*.*

post-property: *If* $(q, Z) \xRightarrow{b} (q', Z')$ *and* $v' \in Z'$ *for local-zones* Z, Z'*, then* $(q, v) \overset{b}{\dashrightarrow} (q', v')$ *for some* $v \in Z$*.*

The initial zone is time-elapsed. By definition of the symbolic transition, every zone reachable by $\Longrightarrow$ transitions is also time-elapsed. Using this observation along with the pre- and post-properties of Lemma 11, we get the following theorem.

▶ **Theorem 12** ([16, 5]). *For a given network* $\mathcal{N}$*, there is a run of* $\mathcal{N}$ *reaching a state* q *iff there is a path in* $\mathrm{LZG}(\mathcal{N})$ *from the initial node to a node* (q, Z)*.*

This shows soundness and completeness of the local-zone graph with respect to state reachability. In particular, soundness follows from the post-property and completeness follows from the pre-property. Interestingly, these two properties are not sufficient to show soundness for infinite runs. In other words, does an infinite run in the local-zone graph imply existence of an infinite run in the local-time semantics? From the post-property, each prefix of the infinite run in the local-zone graph leads to corresponding finite local-time run in the network. However the problem of forming an infinite run from these prefixes is non-trivial.

In the global-time settings, the proof of soundness crucially relies on Alur&Dill's finite region bisimulation. However, there is no finite abstraction that is sound, complete and that preserves all runs on the local zone graph [15]. This last property is crucial to apply partial-order reduction techniques. In the next section, we consider a subclass of timed networks for which we can define a finite region abstraction in the local-time settings.

4 A Region Equivalence for Bounded Spread Valuations

In this section we recall the notion of bounded spread networks of timed automata [15]. We then define a region equivalence $\equiv_M^D$ for such networks.

4.1 Bounded Spread Networks

We consider networks of timed automata where every feasible sequence of actions can be done with bounded desynchronisation between the processes.

▶ **Definition 13.** *Let $D \in \mathbb{N}$. A valuation v is said to have spread D if $|v(t_p - t_q)| \leq D$ for all processes p, q. A run $(q_0, v_0) \xrightarrow{\Delta_0, b_0} (q_1, v_1) \xrightarrow{\Delta_1, b_1} \cdots$ has spread D if v_i and $v_i + \Delta_i$ have spread D for all $i \geq 0$. A network $\mathcal{N}$ has spread D if for every run $(q_0, v_0) \xrightarrow{\Delta_0, b_0} (q_1, v_1) \xrightarrow{\Delta_1, b_1} \cdots$, there exists a D-spread run $(q_0, v_0) \xrightarrow{\Delta'_0, b_0} (q_1, v_1) \xrightarrow{\Delta'_1, b_1} \cdots$ over the same sequence of actions, but with possibly different delays.*

It has been shown that every network can be converted into a D spread network for any arbitrary $D \geq 1$, by adding extra synchronizations [15]. Moreover, for D-spread networks it is possible to get a finite abstraction of the local zone graph, by making use of simulations.

4.2 A Finite Region Bisimulation

A *time-abstract simulation relation* on the local semantics is a reflexive and transitive relation between configurations having the same control state. Two conditions need to be satisfied when $(q, v) \preccurlyeq (q, v')$: (1) for every local delay Δ there exists Δ' such that $(q, v + \Delta) \preccurlyeq (q, v' + \Delta')$, and (2) for every transition $(q, v) \xrightarrow{b} (q_1, v_1)$ there exists a transition $(q, v') \xrightarrow{b} (q_1, v'_1)$ such that $(q_1, v_1) \preccurlyeq (q_1, v'_1)$. When $\Delta = \Delta'$, we simply call $\preccurlyeq$ a simulation relation. For technical convenience, we will use (time-abstract) simulation relations that do not rely on the control state and depend only on the valuations. Hence we will write it as $v \preccurlyeq v'$, a relation over valuations and use its straightforward extension to configurations: $(q, v) \preccurlyeq (q, v')$ if $v \preccurlyeq v'$. The relation $\preccurlyeq$ is a (time-abstract) bisimulation relation if both $\preccurlyeq$ and $\preccurlyeq^{-1}$ are (time-abstract) simulation relations.

The region equivalence of Alur and Dill [2] is a fundamental concept in the global-time semantics that leads to a finite region automaton recognizing the untimed behaviour of the system. This has been cornerstone of several decidability results for timed automata. In the global-time semantics two valuations are made region equivalent with respect to a constant M if for every clock, the values given by the two valuations lie in one of the intervals $[0], (0, 1), [1], \ldots, [M], (M, \infty)$ and the ordering of fractional parts of clocks less than M is the same in both valuations. In the local-time setting, there is an additional challenge. While in the global-time semantics, when a clock goes beyond M, its actual value is irrelevant, it is not the case in the local-time semantics. Indeed, if the difference $t_p - t_q > M$ we cannot forget the actual value, since t_q can elapse some local time and bring the difference $t_p - t_q$ to something lesser than M. This is a fundamental difference between the two semantics, and this is the basic reason that leads to the result of [15] that there is no finite simulation for the local-time semantics. This is why we need to restrict to D-spread valuations v where $|v(t_p - t_q)| \leq D$. We show that with this restriction there is an adequate notion of a region equivalence. We proceed in two steps: first we define when two local valuations are "close-enough", next we factor in the maximum constant M and bounded spread D to get a finite time abstract bisimulation. Some of these results appear in [14].

For $x \in \mathbb{R}$, we write $\lfloor x \rfloor$ for the greatest integer that is lesser than or equal to x. We will write $\{x\}$ for $x - \lfloor x \rfloor$, the fractional part of x starting from $\lfloor x \rfloor$. For example, $\lfloor 4.2 \rfloor = 4, \{4.2\} = 0.2$ and $\lfloor -4.2 \rfloor = -5, \{-4.2\} = 0.8$. Notice that $0 \leq \{x\} < 1$ for all $x \in \mathbb{R}$.

▶ **Definition 14.** *For local valuations v and v', we define $v \approx^\star v'$ if for all pairs of clocks $x, y \in X \cup T$ (including reference clocks), we have $\lfloor v(x-y) \rfloor = \lfloor v'(x-y) \rfloor$.*

Notice that the above definition does not explicitly make use of $\{v(x-y)\}$. Some relation between fractional values gets derived through the definition. For example, suppose $v(x-y) = 1$, then $v(y-x)$ is -1. This will ensure $v'(x-y) = 1$ and $v'(y-x) = -1$. But if $v(x-y) = 1.5$, then $v(y-x) = -1.5$, and in particular $\lfloor v(y-x) \rfloor = \lfloor v'(y-x) \rfloor = -2$. This will say that $1 < v'(x-y) < 2$ and $-2 < v'(y-x) < -1$. We will precisely derive some useful properties later. Before that, we modify the definition to account for the maximum constant.

▶ **Definition 15** ((M, D)-equivalence). *Let $M : X \to \mathbb{N} \cup \{-\infty\}$ be a* bounds function *mapping each process clock to a non-negative constant or $-\infty$ when the value of the clock is irrelevant. Let $D \in \mathbb{N}$ denote a spread. For a local valuation v, let* $\text{Bounded}(v) = \bigcup_{p \in Proc}\{t_p\} \cup \{x \in X_p \mid v(t_p - x) \leq M(x)\}$. *Notice that* $\text{Bounded}(v)$ *contains all the clocks that have a value below bound M in v as well as all reference clocks.*

Two D-spread local valuations v, v' are (M, D)-equivalent*, denoted as $v \equiv_M^D v'$, if*

- $\text{Bounded}(v) = \text{Bounded}(v')$
- $v_{|B} \approx^\star v'_{|B}$ *where $v_{|B}$ and $v'_{|B}$ denote the valuations v and v' restricted to clocks in* $B = \text{Bounded}(v)$.

For a D-spread valuation v, we write $[v]_M^D$ to denote the equivalence class of v under $\equiv_M^D$ and refer to it as the (M, D)-region *of v.*

Intuitively, the above definition says that $v \equiv_M^D v'$ if the bounded part of v and v' are close enough. This is in the same spirit as in the global-time semantics. Now the goal is to show that $\equiv_M^D$ is a time-abstract bisimulation. The most difficult part is to prove that for all local delays Δ, there exist local delays Δ' such that $v + \Delta \equiv_M^D v' + \Delta'$. This is shown in the next lemma. We denote by $\xrightarrow{\delta}_p$ a delay of δ time units in process A_p.

▶ **Lemma 16.** *Let $v \approx^\star v'$. For every local delay $v \xrightarrow{\delta}_p u$, there exists a δ' such that $v' \xrightarrow{\delta'}_p u'$ where $u \approx^\star u'$.*

The next task is to show that if M is appropriately chosen, the $\equiv_M^D$ equivalence also preserves actions. We say that a network $\mathcal{N}$ conforms to bounds function M if every constraint $x \sim c$ in $\mathcal{N}$ satisfies $c \leq M(x)$.

▶ **Lemma 17.** *Let v, v' be D-spread valuations such that $v \equiv_M^D v'$. Let $\mathcal{N}$ be a network that conforms to M. For every action transition $(q, v) \xrightarrow{b} (q_1, v_1)$ we have $(q, v') \xrightarrow{b} (q_1, v'_1)$ such that $v_1 \equiv_M^D v'_1$.*

▶ **Corollary 18.** *Let $\mathcal{N}$ be a network that conforms to M and let $v \equiv_M^D v'$ be D-spread valuations. For every $(q, v) \xrightarrow{\Delta}\xrightarrow{b} (q_1, v_1)$ such that $v + \Delta$ is D-spread, there exists a Δ' such that $v' + \Delta'$ is D-spread and $(q, v') \xrightarrow{\Delta'}\xrightarrow{b} (q_1, v_1')$ with $v_1 \equiv_M^D v_1'$.*

The corollary shows that $\equiv_M^D$ is a time-abstract bisimulation on the local semantics restricted to D-spread configurations. This also motivates the following definition of a region graph obtained as a quotient of the $\equiv_M^D$ equivalence. Recall that the initial valuations are given by $\{v \mid v(x) = v(y) \text{ for all clocks } x, y\}$. Hence by definition of $\equiv_M^D$ equivalence all of them fall in one equivalence class.

▶ **Definition 19** ((M, D)-region graph)**.** *Let $\mathcal{N}$ be a network. A node of an (M, D)-region graph of $\mathcal{N}$ is of the form $(q, [v]_M^D)$ where q is a state of $\mathcal{N}$ and v is a D-spread valuation. There is a transition $(q, [v]_M^D) \xrightarrow{b} (q_1, [v_1]_M^D)$ if $(q, v) \xrightarrow{\Delta}\xrightarrow{b} (q_1, v_1)$ for some local delay Δ such that $v + \Delta$ is D-spread. The initial node is $(q_0, [v_0]_M^D)$ where v_0 is any initial valuation and q_0 is the tuple of initial states.*

▶ **Theorem 20.** *Let $\mathcal{N}$ be a network that conforms to bounds function M. Then:*

- *For every D-spread local run $(q_0, v_0) \xrightarrow{\Delta_0, b_0} (q_1, v_1) \cdots$, there exists a run $(q_0, [v_0]_M^D) \xrightarrow{b} (q_1, [v_1]_M^D) \cdots$ in the (M, D)-region graph.*
- *For every run $(q_0, [v_0]_M^D) \xrightarrow{b_0} (q_1, [v_1]_M^D) \cdots$ in the (M, D)-region graph, there exists a D-spread local run $(q_0, v_0') \xrightarrow{\Delta_0, b_0} (q_1, v_1') \cdots$ such that $v_i' \in [v_i]_M^D$ for all $i \geq 0$.*
- *The (M, D)-region graph is finite.*

5 Abstraction and Partial-Order Reduction for Büchi Runs

The goal of this section is to make use of the local zone graph (Definition 10) for solving the Büchi non-emptines problem. We will start by showing that the local zone graph $\mathrm{LZG}(\mathcal{N})$ is sound and complete for infinite runs of bounded spread networks (Proposition 21). This is only the beginning because: (i) the local zone graph is still potentially infinite, and (ii) the statement does not talk about partial-order reduction. In the next step we introduce a quasi-abstraction $\mathfrak{a}^D_{\preccurlyeq LU}$, and a partial-order approach. Then, we show that their combination maintains correctness for Büchi runs.

▶ **Proposition 21.** *Let $\mathcal{N}$ be a D-spread network. There is an infinite D-spread local run $(q_0, v_0) \xrightarrow{\Delta_0, b_0} (q_1, v_1) \xrightarrow{\Delta_1, b_1} \cdots$ in $\mathcal{N}$ iff there is an infinite path $(q_0, Z_0) \xRightarrow{b_0} (q_1, Z_1) \xRightarrow{b_1} \cdots$ in $\mathrm{LZG}(\mathcal{N})$.*

Proof. Left-to-right direction follows from the pre-property of local zone graph, Lemma 11. We focus on the right-to-left direction.

Let S_i be the set of all D-spread valuations $u_i \in Z_i$ such that there is a D-spread run as below leading to u_i:

$$(q_0, u_0) \xrightarrow{\Delta_0'}\xrightarrow{b_0} (q_1, u_1) \xrightarrow{\Delta_1'}\xrightarrow{b_1} \cdots \xrightarrow{\Delta_{i-1}'}\xrightarrow{b_{i-1}} (q_i, u_i)$$

with u_0 an initial local valuation. The set S_i need not contain all D-spread valuations of Z_i. Consider some D-spread valuation v of Z_i. Due to the post-property, it has some run leading to it, not necessarily D-spread. As the network is D-spread, there is a corresponding D-spread run over the same sequence of actions. However this run may not end up in the same valuation v.

Let us come back to the D-spread run given above. Due to the pre-property of local zones, we have $u_k \in Z_k$ for all $0 \leq k \leq i$. As $\mathcal{N}$ is D-spread, S_i is indeed non-empty, for all $i \geq 0$. In fact, each u_k in the above run belongs to S_k as the prefix with actions $b_0 \cdots b_{k-1}$ is a D-spread run leading to (q_k, u_k). Therefore, for every $u_{i+1} \in S_{i+1}$, there exists a $u_i \in S_i$ such that $(q_i, u_i) \xrightarrow{\Delta}\xrightarrow{b_i} (q_{i+1}, u_{i+1})$ for some local delay Δ'_i.

Construct a graph with nodes $(i, q_i, [u_i]^D_M)$ for each $u_i \in S_i$. Add an edge $(i, q_i, [u_i]^D_M) \to (i+1, q_{i+1}, [u_{i+1}]^D_M)$ if $(q_i, u_i) \xrightarrow{\Delta}\xrightarrow{b_i} (q_{i+1}, u_{i+1})$. Due to the discussion in the previous paragraph, every node has a predecessor. Moreover, by Theorem 20, there are finitely many (M, D)-regions, so this graph is finitely branching. Hence there is an infinite path in this graph. This path corresponds to an infinite path in the (M, D)-region graph. Thanks to Theorem 20, this can be instantiated into an infinite D-spread local run ◀

5.1 Abstractions and Partial-Order Methods

We recall some notions from [15]. A quasi-abstraction $\mathfrak{a}$ is a function that maps each zone Z to a set of valuations $\mathfrak{a}(Z)$ such that $\mathfrak{a}(\mathfrak{a}(Z)) = \mathfrak{a}(Z)$. A finite quasi-abstraction function $\mathfrak{a}^D_{\preccurlyeq_{LU}}$ has been studied in the context of reachability [15]. It is based on a preorder relation between local valuations.

▶ **Definition 22** (The $\preccurlyeq^\star_{LU}$-preorder). *Let $L : X \to \{-\infty\} \cup \mathbb{N}$ and $U : X \to \{-\infty\} \cup \mathbb{N}$ be two functions. For two valuations v and v', we say $v \preccurlyeq^\star_{LU} v'$ if:*

- *$v(t_p - t_q) = v'(t_p - t_q)$ for all processes p, q*
- *for all processes p and all $x \in X_p$,*
  - *$v(t_p - x) \leq U(x)$ implies $v'(t_p - x) \leq v(t_p - x)$,*
  - *$v(t_p - x) \leq L(x)$ implies $v'(t_p - x) \geq v(t_p - x)$,*
  - *$v(t_p - x) > L(x)$ implies $v'(t_p - x) > L(x)$*

Notice that if v is a D-spread valuation and if $v \preccurlyeq^\star_{LU} v'$, valuation v' is also D-spread. Here is a known result about $\preccurlyeq^\star_{LU}$. We say that a network $\mathcal{N}$ conforms to bound functions L and U if for every process clock x we have $L(x) \geq c$ for every lower bound guard $x \geq c, x > c$ occurring in $\mathcal{N}$, and $U(x) \geq c$ for every upper bound guard $x \leq c, x < c$ in $\mathcal{N}$.

▶ **Lemma 23** ([15]). *Let $\mathcal{N}$ be a D-spread network that conforms to given LU bounds. The $\preccurlyeq^\star_{LU}$ pre-order is a simulation on the local semantics of $\mathcal{N}$: if $v \preccurlyeq^\star_{LU} v'$ and $(q, v) \xrightarrow{\Delta, b} (q_1, v_1)$ then $(q, v') \xrightarrow{\Delta, b} (q_1, v'_1)$ and $v_1 \preccurlyeq^\star_{LU} v'_1$.*

The $\preccurlyeq^\star_{LU}$ relation is now lifted to zones, but restricted to D-spread valuations.

▶ **Definition 24** ($\mathfrak{a}^D_{\preccurlyeq_{LU}}$-quasi-abstraction). *For a zone Z, we define* $\mathsf{spread}_D(Z) = \{v \in Z \mid$ *v has spread $D\}$. We define $\mathfrak{a}^D_{\preccurlyeq_{LU}}(Z) = \{v \mid \exists v' \in$* $\mathsf{spread}_D(Z)$ *such that $v \preccurlyeq^\star_{LU} v'\}$.*

The $\mathfrak{a}^D_{\preccurlyeq_{LU}}$ operator can be used to give a finite abstraction of the local zone graph LZG($\mathcal{N}$), by truncating exploration of Z if $\mathfrak{a}^D_{\preccurlyeq_{LU}}(Z) \subseteq \mathfrak{a}^D_{\preccurlyeq_{LU}}(Z')$ and continuing the exploration from Z'. The operation $\mathfrak{a}^D_{\preccurlyeq_{LU}}(Z) \subseteq \mathfrak{a}^D_{\preccurlyeq_{LU}}(Z')$ is known as subsumption in the literature [23, 18]. For Büchi non-emptiness, subsumptions cannot be used directly since we need to find cycles [23, 18]. We will define an abstraction of the local zone graph that makes use of equality with respect to $\mathfrak{a}^D_{\preccurlyeq_{LU}}$. Notice that the equality $\mathfrak{a}^D_{\preccurlyeq_{LU}}(Z) = \mathfrak{a}^D_{\preccurlyeq_{LU}}(Z')$ can be checked efficiently in time $\mathcal{O}((|X| + |T|)^2)$ [15]. We will first present our view of partial-order methods and then combine this with the $\mathfrak{a}^D_{\preccurlyeq_{LU}}$ operator in our new abstraction of the local zone graph.

We describe a generic approach to partial-order methods on local zone graphs. Then our main result will say that once we have a method that works on $\mathrm{LZG}(\mathcal{N})$, we can use the same method on a finite abstraction of $\mathrm{LZG}(\mathcal{N})$ obtained using $\mathfrak{a}^{D}_{\preccurlyeq LU}$.

We formalize what it means to have a partial-order method on $\mathrm{LZG}(\mathcal{N})$ using a notion of a source function. Let $\mathrm{enabled}(q, Z)$ denote the set of actions $b \in \Sigma$ that are enabled from the node (q, Z), i.e. such that there exists an edge $(q, Z) \xRightarrow{b} (q', Z')$ for some (q', Z').

▶ **Definition 25.** *A* source function *for a timed network $\mathcal{N}$ is a function $src : Q \times \mathcal{P}(\Sigma) \to \mathcal{P}(\Sigma)$. An action b is source enabled in (q, Z) if $b \in src(q, \mathrm{enabled}(q, Z))$. A source path is a path taking only source enabled actions. A source function is* trace faithful *if for every node (q, Z) of $\mathrm{LZG}(\mathcal{N})$, and an infinite path u from (q, Z) in $\mathrm{LZG}(\mathcal{N})$ there is a source path $w \sim u$ from (q, Z) in $\mathrm{LZG}(\mathcal{N})$.*

Obeserve that this definition of a source function allows to store some information in the state (like which process moved just before, etc.). Indeed such information is important for certain partial-order reduction approaches.

5.2 Local Zone Graph with Abstraction and Partial-Order

We are now in a position to define a local zone graph for the Büchi non-emptiness problem. This zone graph will use $\mathfrak{a}^{D}_{\preccurlyeq LU}$ for finiteness and an arbitrary source function for partial-order reduction.

▶ **Definition 26** ($\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$)**.** *Let $\mathcal{N}$ be a D-spread network conforming to a given LU-bounds. Let $src : Q \times \mathcal{P}(\Sigma) \to \mathcal{P}(\Sigma)$ be a trace faithful source function. The graph $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$ is a subset of nodes and edges of $\mathrm{LZG}(\mathcal{N})$ together with some new edges called equality edges. Each node is labeled either* covered *or* uncovered*. The graph must satisfy the following conditions:*

- *The initial node of $\mathrm{LZG}(\mathcal{N})$ belongs to the graph.*
- *For every uncovered node (q, Z) of $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$ and for every $b \in src(q, \mathrm{enabled}(Z))$ the transition $(q, Z) \xRightarrow{b} (q', Z')$ present in $\mathrm{LZG}(\mathcal{N})$ should be in $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$.*
- *For every covered node (q, Z') there exists an uncovered node (q, Z) with $\mathfrak{a}^{D}_{\preccurlyeq LU}(Z) = \mathfrak{a}^{D}_{\preccurlyeq LU}(Z')$; moreover there is an explicit* equality edge *$(q, Z) \to_e (q, Z')$ in $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$.*
- *Every node of the graph is reachable from the initial node by a path of $\Rightarrow$ edges.*

We write $\overset{b}{\leadsto}$ to mean a, possibly empty, sequence of equality edges followed by $\xRightarrow{b}$ edge. Similarly $\overset{\sigma}{\leadsto}$ stands for a sequence of $\xRightarrow{\sigma}$ edges possibly with equality edges in between.

Let M be defined as $M(x) = \max(L(x), U(x))$ for every process clock x. The next lemma gives a useful property of the $\mathfrak{a}^{D}_{\preccurlyeq LU}$ abstraction which entails that the above local zone graph is a finite object.

▶ **Lemma 27.** *For every zone Z, the abstraction $\mathfrak{a}^{D}_{\preccurlyeq LU}(Z)$ is a union of (M, D)-regions. The graph $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$ is finite for D-spread networks $\mathcal{N}$.*

Our goal is to show soundness and completeness of $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$ for Büchi non-emptiness. We have already seen in Proposition 21 that $\mathrm{LZG}(\mathcal{N})$ is sound and complete. Therefore we will now relate paths in $\mathrm{LZG}(\mathcal{N})$ with paths in $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$.

5.2.1 Soundness

For soundness, we want to show that every infinite path in $\mathrm{eLZG}_{LU}^{D,src}(\mathcal{N})$ corresponds to an infinite path in $\mathrm{LZG}(\mathcal{N})$. The main difficulty in the argument comes from equality edges. Consider two reachable nodes (q_1, Z_1) and (q_1, Z_1') such that $\mathfrak{a}^D_{\preccurlyeq_{LU}}(Z_1) = \mathfrak{a}^D_{\preccurlyeq_{LU}}(Z_1')$. Let us say there is an equality edge $(q_1, Z_1) \rightarrow_e (q_1, Z_1')$. Hence, $\mathrm{eLZG}_{LU}^{D,src}(\mathcal{N})$ does not contain paths from (q_1, Z_1). We must thus show that for every path from (q_1, Z_1) in $\mathrm{LZG}(\mathcal{N})$ there is a similar path from (q_1, Z_1'). One may guess that as $\preccurlyeq^\star_{LU}$ is a simulation, for every $(q_1, Z_1) \overset{b}{\Rightarrow} (q_2, Z_2)$, we have $(q_1, Z_1') \overset{b}{\Rightarrow} (q_2, Z_2')$ with $\mathfrak{a}^D_{\preccurlyeq_{LU}}(Z_1') = \mathfrak{a}^D_{\preccurlyeq_{LU}}(Z_2')$. But this is not true. Notice that the $\mathfrak{a}^D_{\preccurlyeq_{LU}}$ operator restricts to D-spread valuations. So it can say nothing about the other valuations of Z_1 and Z_1' and these non D-spread valuations may lead to D-spread valuations in Z_2 and Z_2'. We have no control on such valuations just by using the fact that $\preccurlyeq^\star_{LU}$ is a simulation. Nevertheless, we are able to show soundness of $\mathrm{eLZG}_{LU}^{D}(\mathcal{N})$, albeit with a more involved reasoning that additionally uses the fact that $\mathcal{N}$ is a D-spread system and the network $\mathcal{N}$ is deterministic.

▶ **Lemma 28.** *Let $\mathcal{N}$ be a deterministic D-spread network conforming to LU-bounds. Let (q_1, Z_1) be a node reachable from (q_0, Z_0), namely $(q_0, Z_0) \xRightarrow{\sigma_1} (q_1, Z_1)$ for some $\sigma_1 \in \Sigma^*$. Let (q_1, Z_1') be a reachable node of* $\mathrm{LZG}(\mathcal{N})$ *that satisfies* $\mathfrak{a}^D_{\preccurlyeq_{LU}}(Z_1) = \mathfrak{a}^D_{\preccurlyeq_{LU}}(Z_1')$.

For every finite or infinite sequence of transitions σ_2: if $(q_1, Z_1) \xRightarrow{\sigma_2}$ in $\mathrm{LZG}(\mathcal{N})$*, then $(q_1, Z_1') \xRightarrow{\sigma_2}$ in* $\mathrm{LZG}(\mathcal{N})$*. Moreover, if σ_2 is finite then* $\mathrm{enabled}(q_2, Z_2) = \mathrm{enabled}(q_2, Z_2')$*, where $(q_1, Z_1) \xRightarrow{\sigma_2} (q_2, Z_2)$ and $(q_1, Z_1') \xRightarrow{\sigma_2} (q_2, Z_2')$.*

Proof. Suppose σ_2 is finite. Consider the sequence $(q_0, Z_0) \xRightarrow{\sigma_1} (q_1, Z_1) \xRightarrow{\sigma_2} (q_2, Z_2)$. By post-property, there exists a local run $(q_0, v_0) \overset{\sigma_1}{\dashrightarrow} (q_1, v_1) \overset{\sigma_2}{\dashrightarrow} (q_2, v_2)$ with $v_0 \in Z_0, v_1 \in Z_1$ and $v_2 \in Z_2$. As $\mathcal{N}$ is D-spread, we can assume this run to be D-spread. Thus, $v_1 \in \mathsf{spread}_D(Z_1)$.

As $\mathfrak{a}^D_{\preccurlyeq_{LU}}(Z_1) = \mathfrak{a}^D_{\preccurlyeq_{LU}}(Z_1')$, there exists $v_1' \in \mathsf{spread}_D(Z_1')$ such that $v_1 \preccurlyeq^\star_{LU} v_1'$. Hence there exists a run $(q_1, v_1') \overset{\sigma_2}{\dashrightarrow} (q_2, v_2')$. By pre-property, there exists a sequence of symbolic transitions $(q_1, Z_1') \xRightarrow{\sigma_2} (q_2, Z_2')$.

If σ_2 is infinite, then the above argument says that for every finite prefix σ_3 of σ_2 there is a sequence $(q_2, Z_2') \xRightarrow{\sigma_3}$. Since $\mathcal{N}$ is deterministic, the local zone graph $\mathrm{LZG}(\mathcal{N})$ is deterministic. Hence we get the presence of the infinite path σ_2 from (q_2, Z_2'), that is, $(q_2, Z_2') \xRightarrow{\sigma_2}$.

For the last statement consider $(q_1, Z_1) \xRightarrow{\sigma_2} (q_2, Z_2)$ and $(q_1, Z_1') \xRightarrow{\sigma_2} (q_2, Z_2')$. Say b is enabled from (q_2, Z_2). Using the first statement of the lemma $\sigma_2 b$ is possible from (q_1, Z_1'), thus b is possible from (q_2, Z_2'), as the transition system is deterministic. The case of b from (q_2, Z_2') is the same by exchanging the roles of Z_2 and Z_2'. ◀

▶ **Lemma 29.** *Let $\mathcal{N}$ be a deterministic D-spread network that conforms to a bounds LU-bounds.*

Let $(q_0, Z_0) \overset{\sigma_p}{\rightsquigarrow} (q, Z) \overset{\sigma_c}{\rightsquigarrow} (q, Z)$ be a path in $\mathrm{eLZG}_{LU}^{D,src}(\mathcal{N})$ *which could potentially contain equality edges. Then, there exists an infinite D-spread local run over the sequence $\sigma_p(\sigma_c)^\omega$.*

5.2.2 Completeness

We now move on to showing that the graph that is computed is complete for Büchi non-emptiness. Recall that covered nodes and successors via actions that are outside the *src* are not explored in $\mathrm{eLZG}_{LU}^{D,src}$. We will now make use of Lemma 28 to show that source paths in $\mathrm{LZG}(\mathcal{N})$ are preserved in $\mathrm{eLZG}_{LU}^{D,src}(\mathcal{N})$. Later, for the final result, we will use the fact that *src* is trace faithful.

▶ **Lemma 30.** *Let $\mathcal{N}$ be a D-spread network that conforms to a bounds function LU. Let $(q_0, Z_0) \xRightarrow{b_0} (q_1, Z_1) \xRightarrow{b_1}$ be an infinite source path in* $\mathrm{LZG}(\mathcal{N})$*. Then there is an infinite path $(q_0, Z_0) \overset{b_0}{\rightsquigarrow} (q_1, Z_1) \overset{b_1}{\rightsquigarrow} \cdots$ in* $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$.

▶ **Theorem 31.** *Let $\mathcal{N}$ be a deterministic D-spread network that conforms to LU-bounds and let F be a set of accepting actions. Then, there is an infinite global run visiting F infinitely often, iff there is a reachable cycle in* $\mathrm{eLZG}^{D}_{LU}(\mathcal{N})$ *containing an edge over an action in F.*

Proof. By Proposition 9, there is an infinite global run visiting F infinitely often iff there is an infinite local run visiting F infinitely often. Since $\mathcal{N}$ is D-spread, there is an infinite D-spread run with the same sequence of actions. Therefore it remains to prove that there is a D-spread local run visiting F infinitely often iff there is a reachable cycle in $\mathrm{eLZG}^{D}_{LU}(\mathcal{N})$ with an action in F.

Suppose there is such a local run. Proposition 21 says that there is an infinite Büchi path in $\mathrm{LZG}(\mathcal{N})$ from (q_0, Z_0). Since the source function is assumed to be trace faithful, there is a source path that visits F infinitely often. Lemma 30, gives us a Büchi source path in $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$. The $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$ graph is finite (Lemma 27). Hence the infinite path leads to a cycle. As the infinite path contains F infinitely often, the cycle contains an action in F.

For the other direction, suppose there is a reachable cycle containing F in $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$. Lemma 29 gives a D-spread local run with the same sequence of actions and control states. Hence, this local run visits F infinitely often. ◀

6 Conclusions

We have developed a setting allowing to use partial-order methods for solving the Büchi non-emptiness problem for timed systems. Partial-order methods exploit commutation of independent actions. This is why we use local-time semantics for networks of timed automata. For a given network $\mathcal{N}$ we define a finite local-zone graph $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$ such that there is a Büchi run in $\mathcal{N}$ if there is a Büchi path in $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$. Moreover, if we have a partial order method that works on the, potentially infinite, local-zone graph $\mathrm{LZG}(\mathcal{N})$, this method can be used for exploring $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$. We find this a satisfying formulation since in $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$ independent actions do not necessarily commute, due to equality edges.

We did not present here a concrete partial-order method that can be used in our setting. In principle, any ample/persistent/stubborn set method can be used to calculate what we call here source sets. These methods become quite complicated when dealing with infinitary conditions, and these complications limit the efficiency of partial-order reductions. As a first step, it would be reasonable to assume some structural properties, like may-termination [31], but we do not have a satisfying solution at this point.

We did not address the question of Zeno runs. Often one is not just interested in existence of a Büchi run but also wants it to be non-Zeno, that is, a run where time diverges. While there is no consensus on what kind of infinite runs can be considered realistic, it is rather clear that Zeno runs are not realistic. It is always possible to convert a network to a strongly non-Zeno network [29] and encode the non-Zeno requirement in a Büchi condition. Sometimes this construction can produce a blow-up than can be alleviated with more complicated approach [19]. It remains to be seen if this construction can be adapted to the local-time semantics.

References

1 Parosh Aziz Abdulla, Stavros Aronis, Bengt Jonsson, and Konstantinos Sagonas. Source sets: A foundation for optimal dynamic partial order reduction. *J. ACM*, 64(4):25:1–25:49, 2017. doi:10.1145/3073408.

2 Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994. doi:10.1016/0304-3975(94)90010-8.

3 Étienne André, Jaime Arias, Laure Petrucci, and Jaco van de Pol. Iterative bounded synthesis for efficient cycle detection in parametric timed automata. In Jan Friso Groote and Kim Guldstrand Larsen, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 27th International Conference, TACAS 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 - April 1, 2021, Proceedings, Part I*, volume 12651 of *Lecture Notes in Computer Science*, pages 311–329. Springer, 2021. doi:10.1007/978-3-030-72016-2_17.

4 Gerd Behrmann, Patricia Bouyer, Kim Guldstrand Larsen, and Radek Pelánek. Lower and upper bounds in zone-based abstractions of timed automata. *Int. J. Softw. Tools Technol. Transf.*, 8(3):204–215, 2006.

5 Johan Bengtsson, Bengt Jonsson, Johan Lilius, and Wang Yi. Partial order reductions for timed systems. In *CONCUR*, volume 1466 of *Lecture Notes in Computer Science*, pages 485–500, 1998.

6 Frederik Meyer Bønneland, Peter Gjøl Jensen, Kim Guldstrand Larsen, Marco Muñiz, and Jirí Srba. Stubborn set reduction for two-player reachability games. *Log. Methods Comput. Sci.*, 17(1), 2021. URL: https://lmcs.episciences.org/7278.

7 Patricia Bouyer. Forward analysis of updatable timed automata. *Formal Methods Syst. Des.*, 24(3):281–320, 2004.

8 Damien Busatto-Gaston, Benjamin Monmege, Pierre-Alain Reynier, and Ocan Sankur. Robust controller synthesis in timed büchi automata: A symbolic approach. In Isil Dillig and Serdar Tasiran, editors, *Computer Aided Verification - 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I*, volume 11561 of *Lecture Notes in Computer Science*, pages 572–590. Springer, 2019. doi:10.1007/978-3-030-25540-4_33.

9 Krishnendu Chatterjee, Andreas Pavlogiannis, and Viktor Toman. Value-centric dynamic partial order reduction. *Proc. ACM Program. Lang.*, 3(OOPSLA):124:1–124:29, 2019. doi:10.1145/3360550.

10 Dennis Dams, Rob Gerth, Bart Knaack, and Ruurd Kuiper. Partial-order reduction techniques for real-time model checking. *Formal Aspects Comput.*, 10(5-6):469–482, 1998. doi:10.1007/s001650050028.

11 Conrado Daws and Stavros Tripakis. Model checking of real-time reachability properties using abstractions. In *TACAS*, volume 1384 of *Lecture Notes in Computer Science*, pages 313–329. Springer, 1998.

12 Laurent Fribourg. A closed-form evaluation for extended timed automata. Technical report, CNRS and École Normale Supérieure de Cachan, 1998.

13 Patrice Godefroid and Pierre Wolper. A partial approach to model checking. *Inf. Comput.*, 110(2):305–326, 1994. doi:10.1006/inco.1994.1035.

14 R. Govind. *Partial-order reduction for timed systems.* PhD thesis, Université de Bordeaux, France and Chennai Mathematical Institute, India (cotutelle), 2021.

15 R. Govind, Frédéric Herbreteau, B. Srivathsan, and Igor Walukiewicz. Abstractions for the Local-time Semantics of Timed Automata: a Foundation for Partial-order Methods. To appear at 37th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2022. URL: https://hal.archives-ouvertes.fr/hal-03644039.

16 R. Govind, Frédéric Herbreteau, B. Srivathsan, and Igor Walukiewicz. Revisiting local time semantics for networks of timed automata. In *CONCUR*, volume 140 of *LIPIcs*, pages 16:1–16:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

17 Henri Hansen, Shang-Wei Lin, Yang Liu, Truong Khanh Nguyen, and Jun Sun. Diamonds are a girl's best friend: Partial order reduction for timed automata with abstractions. In *CAV*, volume 8559 of *Lecture Notes in Computer Science*, pages 391–406. Springer, 2014.

18 Frédéric Herbreteau, B. Srivathsan, Thanh-Tung Tran, and Igor Walukiewicz. Why liveness for timed automata is hard, and what we can do about it. *ACM Trans. Comput. Log.*, 21(3):17:1–17:28, 2020.

19 Frédéric Herbreteau, B. Srivathsan, and Igor Walukiewicz. Efficient emptiness check for timed büchi automata. *Formal Methods Syst. Des.*, 40(2):122–146, 2012.

20 Frédéric Herbreteau, B. Srivathsan, and Igor Walukiewicz. Lazy abstractions for timed automata. In *CAV*, volume 8044 of *Lecture Notes in Computer Science*, pages 990–1005. Springer, 2013.

21 Frédéric Herbreteau, B. Srivathsan, and Igor Walukiewicz. Better abstractions for timed automata. *Inf. Comput.*, 251:67–90, 2016. doi:10.1016/j.ic.2016.07.004.

22 Michalis Kokologiannakis, Iason Marmanis, Vladimir Gladstein, and Viktor Vafeiadis. Truly stateless, optimal dynamic partial order reduction. *Proc. ACM Program. Lang.*, 6(POPL), January 2022. doi:10.1145/3498711.

23 Alfons Laarman, Mads Chr. Olesen, Andreas Engelbredt Dalsgaard, Kim Guldstrand Larsen, and Jaco van de Pol. Multi-core emptiness checking of timed büchi automata using inclusion abstraction. In *CAV*, volume 8044 of *Lecture Notes in Computer Science*, pages 968–983. Springer, 2013.

24 Kim G. Larsen, Marius Mikucionis, Marco Muñiz, and Jirí Srba. Urgent partial order reduction for extended timed automata. In Dang Van Hung and Oleg Sokolsky, editors, *Automated Technology for Verification and Analysis - 18th International Symposium, ATVA 2020, Hanoi, Vietnam, October 19-23, 2020, Proceedings*, volume 12302 of *Lecture Notes in Computer Science*, pages 179–195. Springer, 2020. doi:10.1007/978-3-030-59152-6_10.

25 Guangyuan Li. Checking timed büchi automata emptiness using lu-abstractions. In *FORMATS*, volume 5813 of *Lecture Notes in Computer Science*, pages 228–242. Springer, 2009.

26 Jesper B. Møller, Jakob Lichtenberg, Henrik Reif Andersen, and Henrik Hulgaard. Fully symbolic model checking of timed systems using difference decision diagrams. *Electron. Notes Theor. Comput. Sci.*, 23(2):88–107, 1999. doi:10.1016/S1571-0661(04)80671-6.

27 Doron A. Peled. All from one, one for all: on model checking using representatives. In Costas Courcoubetis, editor, *Computer Aided Verification, 5th International Conference, CAV '93, Elounda, Greece, June 28 - July 1, 1993, Proceedings*, volume 697 of *Lecture Notes in Computer Science*, pages 409–423. Springer, 1993. doi:10.1007/3-540-56922-7_34.

28 Stavros Tripakis. Checking timed büchi automata emptiness on simulation graphs. *ACM Trans. Comput. Log.*, 10(3):15:1–15:19, 2009.

29 Stavros Tripakis, Sergio Yovine, and Ahmed Bouajjani. Checking timed Büchi automata emptiness efficiently. *Form. Methods Syst. Des.*, 26(3):267–292, 2005.

30 Antti Valmari. Stubborn sets for reduced state space generation. In Grzegorz Rozenberg, editor, *Advances in Petri Nets 1990 [10th International Conference on Applications and Theory of Petri Nets, Bonn, Germany, June 1989, Proceedings]*, volume 483 of *Lecture Notes in Computer Science*, pages 491–515. Springer, 1989. doi:10.1007/3-540-53863-1_36.

31 Antti Valmari. Stop it, and be stubborn! *ACM Trans. Embed. Comput. Syst.*, 16(2):46:1–46:26, 2017. doi:10.1145/3012279.

32 Antti Valmari and Henri Hansen. Stubborn set intuition explained. *Trans. Petri Nets Other Model. Concurr.*, 12:140–165, 2017. doi:10.1007/978-3-662-55862-1_7.

33 Naling Zhang, Markus Kusano, and Chao Wang. Dynamic partial order reduction for relaxed memory models. In David Grove and Stephen M. Blackburn, editors, *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation, Portland, OR, USA, June 15-17, 2015*, pages 250–259. ACM, 2015. doi:10.1145/2737924.2737956.

A Appendix for Section 3

▶ **Lemma 8.** *For every T' a prefix of T_σ, every linearization f of T' gives a local run.*

Proof. We will suppose that T' is infinite. For finite T' the argument is essentially the same. Consider a linearization of a prefix of T_σ given by a 1-1 function $f : \mathbb{N} \to \mathbb{N}$. This means that $\mathrm{rng}(f)$ is $\trianglelefteq_\sigma$-downwards closed: if $j \in \mathrm{rng}(f)$ and $i \trianglelefteq_\sigma j$ then $i \in \mathrm{rng}(f)$. We construct a local run:

$$(\bar{q}_{f(0)}, \bar{v}_{f(0)}) \xrightarrow{\Delta_{f(0)}, b_{f(0)}} (\bar{q}_{f(1)}, \bar{v}_{f(1)}) \xrightarrow{\Delta_{f(1)}, b_{f(1)}} \cdots$$

In order to state the invariant we define a function $\mathit{flast}(i,p)$ giving the last action before i with an action of process p: $\mathit{flast}(i,p) = f(l)$ where l the largest number strictly smaller than i with $b_{f(l)}$ an action of p; we keep $\mathit{flast}(i,p)$ undefined if there is no such l. Recall that a state $q = (q^1, \dots, q^k)$ is a tuple of states of component processes. We use q^p for the p-th component of q. The invariants are:

- for every process p, $\bar{q}^p_{f(i)} = q^p_{\mathit{flast}(i,p)+1}$ or $\bar{q}^p_{f(i)} = q^p_0$ if $\mathit{flast}(i,p)$ undefined.
- for every process p, $\bar{v}_{f(i)}(t_p) = \theta(f(i))$ if $p \in \mathit{dom}(b_{f(i)})$; otherwise $\bar{v}_{f(i)}(t_p) = \theta(\mathit{flast}(i,p))$ or $\bar{v}_{f(i)}(t_p) = v_0(t_p)$ if $\mathit{flast}(i,p)$ is not defined.
- for every $x \neq t_p$ a clock of process p: if $\mathit{flast}(i,p)$ is defined then $\bar{v}_{f(i)}(x) = v(x)$ where $v(x) = v_{\mathit{flast}(i,p)}[R]$ and R are resets of $b_{\mathit{flast}(i,p)}$; if $\mathit{flast}(i,p)$ not defined then $\bar{v}_{f(i)}(x) = v_0(x)$.

We set $\bar{q}_{f(0)} = q_{f(0)}$ and $\bar{v}_{f(0)}(t_p) = \theta(f(0))$ for $p \in \mathit{dom}(b_{f(0)})$ and $\bar{v}_{f(0)}(t_p) = v_0(t_p)$ otherwise. Finally, we set $\bar{v}_{f(0)} = v_0(x)$ for all other clocks. This satisfies the invariant as $b_{f(0)}$ is $\trianglelefteq$-minimal action.

Suppose that we have constructed a run up to $(\bar{q}_{f(i)}, \bar{v}_{f(i)})$. The invariants guarantee that for every process $p \in \mathit{dom}(b_{f(i)})$ we have:

- $\bar{q}^p_{f(i)} = q^p_{f(i)}$,
- $\bar{v}_{f(i)}(t_p) = v_{f(i)}(t_p)$ and $\bar{v}_{f(i)}(x) = v_{f(i)}(x)$ for every clock x of p.

Since $b_{f(i)}$ is enabled from $(q_{f(i)}, v_{f(i)})$ this shows that it is also enabled from $(\bar{q}_{f(i)}, \bar{v}_{f(i)})$. Consider $(\bar{q}_{f(i)}, \bar{v}_{f(i)}) \xrightarrow{b_{f(i)}} (q_b, v_b)$. We need to show that $q_b = \bar{q}_{f(i+1)}$ and that there is Δ giving $v_b + \Delta = \bar{v}(f(i+1))$.

We start with q_b. Take a process $p \in \mathit{dom}(b_{f(i)})$ then $\mathit{flast}(i+1,p) = f(i)$ and we get $q^p_b = q^p_{\mathit{flast}(i+1,p)+1}$. For $p \notin \mathit{dom}(b_{f(i)})$, we have $q^p_b = \bar{q}^p_{f(i)} = q^p_{\mathit{flast}(i,p)+1} = q^p_{\mathit{flast}(i+1,p)+1}$. The last equality is because $\mathit{flast}(i,p) = \mathit{flast}(i+1,p)$ when $p \notin \mathit{dom}(b_{f(i)})$.

Now we look at reference clocks t_p. If $p \in \mathit{dom}(b_{f(i)}) \setminus \mathit{dom}(b_{f(i+1)})$ then we have $v_b(t_p) = \theta(f(i)) = \theta(\mathit{flast}(i,p))$ as required. If $p \notin \mathit{dom}(b_{f(i)}) \cup \mathit{dom}(b_{f(i+1)})$ then $v_b(t_p) = \theta(\mathit{flast}(i,p)) = \theta(\mathit{flast}(i+1,p))$. If $p \in \mathit{dom}(b_{f(i+1)})$ then $v_b(t_p) = \theta(\mathit{flast}(i+1,p)) \leq \theta(f(i+1))$ so we take $\delta_p = \theta(f(i+1)) - v_b(t_p)$ to reestablish the invariant.

Finally, to check the third invariant take a clock $x \neq t_p$ of a process p. Recall that $v_b = v_{f(i)}[R_b]$ where R_b are resets of action $b_{f(i)}$. If $p \in \mathit{dom}(b_{f(i)})$ then v in the invariant is $v_{f(i)}[R]$ because $\mathit{flast}(i+1,p) = f(i)$. This is as required. If $p \notin \mathit{dom}(b_{f(i)})$ then $\mathit{flast}(i+1,p) = \mathit{flast}(i,p)$ and $v_b(x) = \bar{v}_{f(i)}$, so we are done in this case too.

Hence, we have prolonged the run to $(\bar{q}_{f(i+1)}, \bar{v}_{f(i+1)})$ and all invariants are satisfied. By induction we obtain the desired local run corresponding to a linearization f. ◀

▶ **Proposition 9.** *Consider a network of timed automata $\mathcal{N}$. There is a Büchi local run of $\mathcal{N}$ iff there is a Büchi global run of $\mathcal{N}$.*

Proof. Since a global run is also a local run, one direction is easy.

For the other direction, let us take a local run with infinitely many occurrences of actions from F, and construct a global run. Recall that $\theta(i)$ is the time of the execution of action b_i, namely $v'_i(t_p)$ for $p \in \mathit{dom}(b_i)$. Consider an order $i \lessdot j$ when $\theta(i) < \theta(j)$ or $\theta(i) = \theta(j)$ and $i < j$.

This order is a linear order $\trianglelefteq_\sigma$ and moreover if $i \lessdot j$ then it is not the case that $j \trianglelefteq i$ (Lemma 7). Yet the linear order $\lessdot$ may not have type ω meaning that it can be a transfinite sequence. We find a prefix T of T_σ such that $\lessdot$ is a linearization of T of type ω and T has infinitely many occurrences of actions from F.

Before doing this let us see how this gives us a desired global run. The $\lessdot$-linearization of some prefix T of T_σ gives us a local run (Lemma 8):

$$\sigma = (q_0, v_0) \xrightarrow{\Delta_0} (q_0, v'_0) \xrightarrow{b_0} (q_1, v_1) \xrightarrow{\Delta'_1} \cdots$$

with $\theta(0) \leq \theta(1) \leq \cdots$. We define $\bar{v}'_i(t_p) = \theta(i)$ for all process p and $\bar{v}'_i(x) = v_i(x)$ for all other clocks x. Clearly all $\bar{v}'_i$ are synchronized valuations. We claim that

$$\sigma = (q_0, \bar{v}_0) \xrightarrow{\delta_0} (q_0, \bar{v}'_0) \xrightarrow{b_0} (q_1, \bar{v}_1) \xrightarrow{\delta_1} (q_1, \bar{v}'_1) \xrightarrow{b_1} \cdots$$

is a global run (where $\delta_i = \theta(i) - \theta(i-1)$, and $\bar{v}_{i+1}$ is determined by $\bar{v}'_i$ and resets of b_i). For this we verify that every transition $(q_i, \bar{v}'_i) \xrightarrow{b_i} (q_{i+1}, \bar{v}_{i+1}) \xrightarrow{\delta_{i+1}} (q_{i+1}, \bar{v}'_{i+1})$ exists. We know $(q_i, v'_i) \xrightarrow{b_i} (q_{i+1}, v_{i+1})$. We also have by assumption that $\bar{v}_i(t_p) = \theta(i) = v'_i(t_p)$ for all $p \in \mathit{dom}(b_i)$, as well as $\bar{v}'_i(x) = v'_i(x)$ for all clocks x that are not reference clocks. Hence $(q_i, \bar{v}'_i) \xrightarrow{b_i} (q_{i+1}, v_b)$ exists. We have $v_b(x) = v_{i+1}(x)$ for all clocks that are non-reference clocks. Additionally $v_b(t_p) = v'_i(t_p) = \theta(i)$ for every process p. Hence, if we take $\delta_{i+1} = \theta(i+1) - \theta(i)$ we get $\bar{v}'_{i+1} = v_b + \delta_{i+1}$ as required. Repeating this reasoning we construct a desired run.

We come back to the problem of finding a desired linearization. If $\lessdot$ gives a linearization of T_σ of type ω then we are done. Otherwise, consider the set $I = \{i : i \text{ has finitely many } \lessdot -\text{smaller elements}\}$. We have that $\lessdot$ is an order of type ω on I. Moreover I is a $\trianglelefteq_\sigma$-downward closed as it is impossible to have $j \trianglelefteq_\sigma i$ and $i \lessdot j$ at the same time. If I contains infinitely many occurrences of actions from F then I defines a prefix we are looking for. Otherwise $\mathbb{N} \setminus I$ is infinite and there are infinitely many actions from F in $\mathbb{N} \setminus I$. Consider the set of processes P_ω such that there is an action $b \in \mathbb{N} \setminus I$ with p in its domain. For every $p \in P$ find the $\lessdot$-smallest $j_p \in \mathbb{N} \setminus I$ such that $p \in \mathit{dom}(b_{j_p})$. We claim that there are finitely many $i \in I$ with $i \trianglelefteq_\sigma j_p$. Indeed $i \trianglelefteq_\sigma j_p$ implies $i \leq j$ in the standard order on natural numbers. Let I_0 contain all such i for all $p \in P$. We claim that for $i \in I \setminus I_0$ we have $i \ntrianglelefteq_\sigma j$ for every $j \in \mathbb{N} \setminus I$. Hence $I_0 \cup (\mathbb{N} \setminus I)$ is a prefix of T_σ. Moreover, it contains infinitely many occurrences of actions from F since I contains only finitely many of those and $\mathbb{N}$ has infinitely many. If $\lessdot$-linearization of $I_0 \cup (\mathbb{N} \setminus I)$ has type ω then we are done. If not then we repeat the argument. Observe that this time we have fewer processes such that there are infinitely many actions involving the process (none of the processes involved in actions from $I \setminus I_0$ are there). Thus the argument must terminate giving us the desired prefix. ◀

B Appendix for Section 4

▶ **Lemma 32.** *For all $x \in \mathbb{R} \setminus \mathbb{Z}$, we have $\{-x\} = 1 - \{x\}$.*

Proof. We have $x = \lfloor x \rfloor + \{x\}$ and $-x = \lfloor -x \rfloor + \{-x\}$. Therefore $-(\lfloor x \rfloor + \{x\}) = \lfloor -x \rfloor + \{-x\}$. Secondly, $\lfloor -x \rfloor = -\lfloor x \rfloor - 1$ for all $x \in \mathbb{R} \setminus \mathbb{Z}$. Plugging this into the previous equation gives the required conclusion. ◀

▶ **Lemma 33.** *For $x, y, z \in \mathbb{R}$ such that $z - x \in \mathbb{R} \setminus \mathbb{Z}$, we have $\{z - x\} \leq \{z - y\}$ iff $\lfloor x - y \rfloor = \lfloor x - z \rfloor + \lfloor z - y \rfloor + 1$.*

Proof.

$$\begin{aligned} x - y &= x - z + z - y \\ &= \lfloor x - z \rfloor + \lfloor z - y \rfloor + \{x - z\} + \{z - y\} \\ &= \lfloor x - z \rfloor + \lfloor z - y \rfloor + 1 - \{z - x\} + \{z - y\} \end{aligned}$$

This gives the statement of the lemma. ◀

▶ **Lemma 34.** *Let $x, y, z \in \mathbb{R}$, such that $\{x - z\} > 0$ and $\{y - z\} > 0$. Then, $\{x - z\} \leq \{y - z\}$ iff $\{z - x\} \geq \{z - y\}$.*

Proof. Follows by using $\{x - z\} = 1 - \{z - x\}$ and $\{y - z\} = 1 - \{z - y\}$ (Lemma 32). ◀

▶ **Lemma 35.** *Let $v \approx^\star v'$. Then, for variables $x, y, z \in X \cup X^t$, we have $\{v(z - x)\} \leq \{v(z - y)\}$ iff $\{v'(z - x)\} \leq \{v'(z - y)\}$.*

Proof. Follows from Lemma 33 and Definition 14. ◀

▶ **Lemma 16.** *Let $v \approx^\star v'$. For every local delay $v \xrightarrow{\delta}_p u$, there exists a δ' such that $v' \xrightarrow{\delta'}_p u'$ where $u \approx^\star u'$.*

Proof. We assume that $0 < \delta < 1$. If $\delta \geq 1$ then we can decompose it into its integral part and fractional part and repeat the reasoning.

We divide the variable differences into three sets:

$$\begin{aligned} C^+ &= \{t_p - z \mid z \in X \setminus \{t_p\}\} \\ C^- &= \{z - t_p \mid z \in X \setminus \{t_p\}\} \\ C^0 &= \{x - y \mid x, y \in X \setminus \{t_p\}\} \end{aligned}$$

A local delay of δ increases the value of differences in C^+, decreases the ones in C^- and leaves the C^0 differences unaltered. Consider an element $\phi \in C^+$. Based on the relation between δ and $1 - \{v(\phi)\}$, its value either stays in the same integer interval, or moves to the next integer point, or to the next integer interval. A symmetric change happens in C^-. We now make this idea more precise.

$$\begin{aligned} u(t_p - z) &= v(t_p - z) + \delta \\ &= \lfloor v(t_p - z) \rfloor + \{v(t_p - z)\} + \delta \\ &= \lfloor v(t_p - z) \rfloor + 1 - \{v(z - t_p)\} + \delta \qquad \text{when } v(t_p - z) \neq 0 \\ u(z - t_p) &= v(z - t_p) - \delta \\ &= \lfloor v(z - t_p) \rfloor + \{v(z - t_p)\} - \delta \end{aligned}$$

From the above calculations, we observe some properties:

- When $\{v(t_p - z)\} \neq 0$:

$$\lfloor u(t_p - z)\rfloor = \lfloor v(t_p - z)\rfloor + 1 \text{ iff } \delta \geq \{v(z - t_p)\} \tag{1}$$
$$\lfloor u(z - t_p)\rfloor = \lfloor v(z - t_p)\rfloor - 1 \text{ iff } \delta > \{v(z - t_p)\} \tag{2}$$

- When $\{v(t_p - z)\} = 0$, as $\delta < 1$ we have:

$$\lfloor u(t_p - z)\rfloor = \lfloor v(t_p - z)\rfloor \tag{3}$$
$$\lfloor u(z - t_p)\rfloor = \lfloor v(z - t_p)\rfloor - 1 \tag{4}$$

Note that the difference in the inequalities ($\geq$ in (1) and $>$ in (2)) is expected, since for any $x \in \mathbb{R}$ we have $\lfloor -x\rfloor = -\lfloor x\rfloor$ if $\{x\} = 0$, and $\lfloor -x\rfloor = -\lfloor x\rfloor - 1$ otherwise. Among the ordering of fractional parts of differences in C^- for v, consider $(z_1 - t_p), (z_2 - t_p)$ that are consecutive in this ordering such that $\{v(z_1 - t_p)\} \leq \delta < \{v(z_2 - t_p)\}$. Replace $\{v(z_1 - t_p)\}$ with 0 if no such z_1 exists, and replace $\{v(z_2 - t_p)\}$ with 1 if no such z_2 exists.

We now propose a δ' as required. From Lemmas 34 and 35, we know that the fractional parts of differences in C^- are ordered in the same way in v and v'. We take any δ' with $\{v'(z_1 - t_p)\} \leq \delta' < \{v'(z_2 - t_p)\}$, such that in addition $\delta' = \{v'(z_1 - t_p)\}$ if $\delta = \{v(z_1 - t_p)\}$. Let $u' = v' + \delta'$. Since we started with $v \approx^\star v'$, from (1) to (4) we get $u \approx^\star u'$. ◀

The following lemma shows that $\equiv^D_M$ equivalence is preserved by choosing appropriate local delays.

▶ **Lemma 36.** *Let v, v' be D-spread valuations such that $v \equiv^D_M v'$. For every local delay Δ such that $v + \Delta$ is D-spread, there exists a local delay Δ' such that $v' + \Delta'$ is D-spread and $v + \Delta \equiv^D_M v' + \Delta'$.*

Proof. Let $\Delta = \{\delta_p\}_{p \in Proc}$. We can break the local delay Δ into a sequence of local delays $\xrightarrow{\delta_{p_1}}\xrightarrow{\delta_{p_2}}\cdots$ happening one process at a time. Therefore it is sufficient to prove the lemma for a local delay of one process, say δ_p at process p.

Consider the given valuations v, v' which satisfy $v \equiv^D_M v'$. By definition, we have $\text{Bounded}(v) = \text{Bounded}(v')$ and $v_{\mid B} \approx^\star v'\mid_B$, where $B = \text{Bounded}(v)$. From Lemma 16, for every local delay δ_p, there exists a delay δ'_p such that $(v +_p \delta_p)|_B \approx^\star (v' +_p \delta'_p)|_B$. Clocks outside B are unbounded both in $v +_p \delta_p$ and $v' +_p \delta'_p$. Finally, we are interested only in delays δ_p such that $v +_p \delta_p$ is D-spread. Since all the reference clocks are present in B, we have $\lfloor (v +_p \delta_p)(t_r - t_s)\rfloor = \lfloor (v' +_p \delta'_p)(t_r - t_s)\rfloor$. This shows that $v' +_p \delta'_p$ is D-spread. All these observations lead to $v +_p \delta_p \equiv^D_M v' +_p \delta'_p$. ◀

▶ **Lemma 17.** *Let v, v' be D-spread valuations such that $v \equiv^D_M v'$. Let $\mathcal{N}$ be a network that conforms to M. For every action transition $(q, v) \xrightarrow{b} (q_1, v_1)$ we have $(q, v') \xrightarrow{b} (q_1, v'_1)$ such that $v_1 \equiv^D_M v'_1$.*

Proof. As $(q, v) \xrightarrow{b} (q_1, v_1)$, we have $v(t_p - t_q) = 0$ for all $p, q \in dom(b)$. Since $v \equiv^D_M v'$, we also have $v'(t_p - t_q) = 0$ for $p, q \in dom(b)$. Secondly, v satisfies the guard g present in the b-transition. As $\mathcal{N}$ conforms to M, every constraint in g is of the form $x < c, x \leq c$ or $x > c, x \geq c$ with $0 \leq c \leq M(x)$. Hence, by definition of $v \equiv^D_M v'$, valuation v' satisfies g too. This shows that b is enabled at (q, v'). Finally, resetting R from v sets differences $t_p - x$ with $x \in X_p \cap R$ to 0. It does not change the values of differences between reference clocks. Hence both $[R]v$ and $[R]v'$ are D-spread. Moreover, $\text{Bounded}([R](v)) = \text{Bounded}(v) \cup R$, which equals $\text{Bounded}(v') \cup R$ and hence $\text{Bounded}([R]v')$. We need to show that $[R](v)|_{B_1} \approx^\star [R](v')|_{B_1}$ where $B_1 = \text{Bounded}([R]v)$. So, we need to show that $\lfloor [R](v)(x - y)\rfloor = \lfloor [R](v')(x - y)\rfloor$.

This is direct when both $x, y \in R$, or when both $x, y \notin R$. Suppose $x \in R, y \notin R$. We have $[R]v(x-y) = v(t_p - y)$ when $x \in X_p$. Since t_p and y are already in Bounded(v), we have $\lfloor v(t_p - y) \rfloor = \lfloor v'(t_p - y) \rfloor$ and hence $\lfloor [R](v)(x-y) \rfloor = \lfloor [R](v')(x-y) \rfloor$. Symmetric reasoning works when $x \notin R, y \in R$. ◀

▶ **Theorem 20.** *Let $\mathcal{N}$ be a network that conforms to bounds function M. Then:*

- *For every D-spread local run $(q_0, v_0) \xrightarrow{\Delta_0, b_0} (q_1, v_1) \cdots$, there exists a run $(q_0, [v_0]^D_M) \xrightarrow{b} (q_1, [v_1]^D_M) \cdots$ in the (M, D)-region graph.*
- *For every run $(q_0, [v_0]^D_M) \xrightarrow{b_0} (q_1, [v_1]^D_M) \cdots$ in the (M, D)-region graph, there exists a D-spread local run $(q_0, v'_0) \xrightarrow{\Delta_0, b_0} (q_1, v'_1) \cdots$ such that $v'_i \in [v_i]^D_M$ for all $i \geq 0$.*
- *The (M, D)-region graph is finite.*

Proof.

- Follows from definition of region graph.
- Given a transition $(q_i, [v_i]^D_M) \xRightarrow{b_i} (q_{i+1}, [v_{i+1}]^D_M)$, for *every* valuation $u_i \in [v_i]^D_M$), there is a transition $(q_i, u_i) \xrightarrow{\Delta_i, b_i} (q_{i+1}, u_{i+1})$ with $u_{i+1} \in [v_{i+1}]^D_M$. This holds due to Corollary 18 and is sufficient to extract a run starting from some arbitrary initial configuration.
- We claim that an (M, D) region is specified by the following information:
  - a subset $B \subseteq \bigcup_{p \in Proc} X_p$ of bounded clocks,
  - for every process clock $x \in X_p$ that is bounded, whether $t_p - x = c$ or $e - 1 < t_p - x < e$ for $c \in \{0, \dots M(x)\}$ and $e \in \{1, \dots, M(x)\}$,
  - for every pair of reference clocks t_p, t_q, whether $t_p - t_q = c$ or $e - 1 < t_p - t_q < e$ for $c \in \{-D, \dots, D\}$ and $e \in \{-D+1, \dots, D\}$
  - for a pair of bounded process clocks $x, y \in B$, whether $x - y = c$ or $e - 1 < x - y < e$ for $c \in \{-M(x) - D, \dots, M(y) + D\}$ and $e \in \{-M(x) - D + 1, \dots M(y) + D\}$.

  The claim gives a finite bound on the number of regions. It remains to prove the claim. Consider an (M, D)-region $[v]^D_M$. We have Bounded(v) = Bounded(v') for every valuation $v' \in [v]^D_M$. Hence the set B in the first item above is given by Bounded(v). The next three items follow from $v_{|B} \approx^\star v'|_B$ and noticing the bounds on the differences: we have $0 \leq v(t_p - x) \leq M(x)$ for $x \in B \cap X_p$ by definition of bounded clocks; we have $-D \leq v(t_p - t_q) \leq D$ since v has spread D; finally for $x, y \in B$, we have $v(x - y) \leq v(x - t_p) + v(t_p - t_q) + v(t_q - y)$ assuming $x \in X_p, y \in X_q$. Now, we use the inequalities: $-M(x) \leq v(x - t_p) \leq 0$, $-D \leq v(t_p - t_q) \leq D$ and $0 \leq v(t_q - y) \leq M(y)$ to get $-M(x) - D \leq v(x - y) \leq M(y) + D$. ◀

C Appendix for Section 5

▶ **Lemma 37.** *Let $\mathcal{N}$ be a deterministic D-spread network conforming to LU-bounds. For every path of the form $(q_0, Z_0) \overset{\sigma_1}{\rightsquigarrow} (q_1, Z'_1) \rightarrow_e (q_1, Z_1) \xRightarrow{\sigma_2} (q_2, Z_2)$ in $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$ there exists a path $(q_1, Z'_1) \xRightarrow{\sigma_2} (q_2, Z'_2)$ in $\mathrm{LZG}(\mathcal{N})$.*

Proof. Since (q_1, Z_1) is a node of $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$, it is reachable from (q_0, Z_0), and so is (q_1, Z'_1). Since $\mathfrak{a}^D_{\preccurlyeq LU}(Z_1) = \mathfrak{a}^D_{\preccurlyeq LU}(Z'_1)$, Lemma 28 gives us a path $(q_1, Z'_1) \xRightarrow{\sigma_2} (q_2, Z'_2)$ in $\mathrm{LZG}(\mathcal{N})$. ◀

▶ **Lemma 29.** *Let $\mathcal{N}$ be a deterministic D-spread network that conforms to a bounds LU-bounds.*

Let $(q_0, Z_0) \overset{\sigma_p}{\rightsquigarrow} (q, Z) \overset{\sigma_c}{\rightsquigarrow} (q, Z)$ be a path in $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$ which could potentially contain equality edges. Then, there exists an infinite D-spread local run over the sequence $\sigma_p(\sigma_c)^\omega$.

Proof. Let M be defined as $M(x) = \max(L(x), U(x))$ for every process clock x. Let $k \in \mathbb{N}$ be larger than the number of (M, D)-regions. Consider the finite path in $\mathrm{eLZG}^D_{LU}(\mathcal{N})$ obtained by the sequence $\sigma_p(\sigma_c)^k$. By repeated use of Lemma 37, there is a path $\sigma_p(\sigma_c)^k$ in $\mathrm{LZG}(\mathcal{N})$. By post-property of $\mathrm{LZG}(\mathcal{N})$, there is a local run $(q_0, v_0) \overset{\sigma_p}{\dashrightarrow} (q, v_1) \overset{\sigma_c}{\dashrightarrow} (q, v_2) \overset{\sigma_c}{\dashrightarrow} \cdots \overset{\sigma_c}{\dashrightarrow} (q, v_{k+1})$. As $\mathcal{N}$ is D-spread, this run can be assumed to be D-spread. Due to Theorem 20, there is a path $\sigma_p(\sigma_c)^k$ in the (M, D)-region graph: $(q_0, [v_0]^D_M) \xrightarrow{\sigma_p} (q, [v_1]^D_M) \xrightarrow{\sigma_c} \cdots \xrightarrow{\sigma_c} (q, [v_{k+1}]^D_M)$. As k is larger than the number of regions, there exist i, j such that $[v_i]^D_M = [v_j]^D_M$. This gives a path $\sigma_p\sigma_c^{i-1}\sigma_c^{j-i}\sigma_c^{k+1-j}$ in the region graph where the part σ_c^{j-i} is a cycle, which can be iterated infinitely often. Hence there is a path $\sigma_p(\sigma_c)^\omega$ in the region graph. By Theorem 20, there is an infinite local run over the sequence $\sigma_p(\sigma_c)^\omega$, whose intermediate valuations are all D-spread. ◀

▶ **Lemma 30.** *Let $\mathcal{N}$ be a D-spread network that conforms to a bounds function LU. Let $(q_0, Z_0) \xRightarrow{b_0} (q_1, Z_1) \xRightarrow{b_1}$ be an infinite source path in $\mathrm{LZG}(\mathcal{N})$. Then there is an infinite path $(q_0, Z_0) \overset{b_0}{\rightsquigarrow} (q_1, Z_1) \overset{b_1}{\rightsquigarrow} \cdots$ in $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$.*

Proof. Let $w_i = b_i b_{i+1} \ldots$. By induction on i we construct a path in $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$

$$(q_0, Z_0) \overset{b_0}{\rightsquigarrow} (q_1, Z'_1) \cdots \overset{b_i}{\rightsquigarrow} (q_i, Z'_i)$$

such that in $\mathrm{LZG}(\mathcal{N})$ there is a path

$$(q_i, Z'_i) \xRightarrow{b_{i+1}} (q_{i+1}, Z^i_{i+1}) \xRightarrow{b_{i+2}} (q_{i+2}, Z^i_{i+2}) \ldots \qquad \text{and } \mathrm{enabled}(q_j, Z^i_j) = \mathrm{enabled}(q_j, Z_j).$$

The second item implies that the later path is a source path in $\mathrm{LZG}(\mathcal{N})$.

If (q_i, Z'_i) is not covered in $\mathrm{eLZG}^{D,src}_{LU}$ then the induction step is direct.

If (q_i, Z'_i) is covered in $\mathrm{eLZG}^{D,src}_{LU}$ then there exists $(q_i, Z'_i) \rightarrow_e (q_i, Z''_i)$ with (q_i, Z''_i) uncovered and $\mathfrak{a}^D_{\preccurlyeq LU}(Z'_i) = \mathfrak{a}^D_{\preccurlyeq LU}(Z''_i)$. As (q_i, Z''_i) is reachable from (q_0, Z_0) in $\mathrm{LZG}(\mathcal{N})$ by the definition of $\mathrm{eLZG}^{D,src}_{LU}$, we can use Lemma 28. The lemma gives us a path $(q_i, Z''_i) \xRightarrow{b_{i+1}} (q_{i+1}, Z^{i+1}_{i+1}) \xRightarrow{b_{i+2}} (q_{i+2}, Z^{i+1}_{i+2}) \ldots$ such that $\mathrm{enabled}(q_j, Z^i_j) = \mathrm{enabled}(q_j, Z^{i+1}_j)$. Thus we also have $\mathrm{enabled}(q_j, Z_j) = \mathrm{enabled}(q_j, Z^{i+1}_j)$. We can prolong the finite prefix $(q_0, Z_0) \overset{b_0}{\rightsquigarrow} (q_1, Z'_1) \ldots \overset{b_i}{\rightsquigarrow} (q_i, Z'_i)$ by $(q_i, Z'_i) \rightarrow_e (q_i, Z''_i) \xRightarrow{b_{i+1}} (q_{i+1}, Z'_{i+1})$. ◀

The local zone graph $\mathrm{eLZG}^{D,src}_{LU}(\mathcal{N})$ is finite. We will prove that for every zone Z, the abstraction $\mathfrak{a}^D_{\preccurlyeq LU}(Z)$ is a union of (M, D)-regions. To prove this statement, we will need to reason about *canonical* representations of zones. Zones are typically represented using Difference-Bound-Matrices (DBMs) or distance graphs [21]. We will use the distance graph representation for our analysis. A constraint $x - y \lessdot c$ of the zone is represented as an edge $y \xrightarrow{\lessdot c} x$. An arithmetic over weights of $(\lessdot, c)$ can be suitably defined (see [21], [15]) for more details. A canonical graph is one where the shortest path from y to x is given by the direct edge $y \rightarrow x$. We will write Z_{yx} to denote the weight of the $y \rightarrow x$ edge in the canonical distance graph representing Z.

For convenience of presentation, we define two sets of clocks for a given local valuation v:

$$L\text{-bounded}(v) \;:=\; T \;\cup \bigcup_{p \in Proc} \{x \in X_p \mid v(t_p - x) \leq L_x\}$$

$$U\text{-bounded}(v) \;:=\; T \;\cup \bigcup_{p \in Proc} \{x \in X_p \mid v(t_p - x) \leq U_x\}$$

Notice that the reference clocks T are present in both L-bounded(v) and U-bounded(v).

Define $\langle v \rangle^\star := \{v' \mid v \preccurlyeq^\star_{LU} v'\}$. We will now recall the distance graph representation of $\langle v \rangle^\star$ and an important property of the intersection $\langle v \rangle^\star \cap Z$ for some arbitrary zone Z'.

▶ **Definition 38** (Distance graph H^v [15]). *Let $x, y \in X \cup X^t$ be two clocks, possibly reference clocks. Assume that $y \neq x$ and $y \in X_q \cup \{t_q\}$ for some process q. The weight of the edge $x \to y$ in the distance graph H^v is given by:*

$$\begin{cases} (\leq, v(y - x)) & \textit{if } x \in U\text{-bounded}(v), \\ & \quad y \in L\text{-bounded}(v) \\ (\leq, v(t_q - x)) + (<, -L_y) & \textit{if } x \in U\text{-bounded}(v), \\ & \quad y \notin L\text{-bounded}(v), L_y \neq -\infty \\ (\leq, v(t_q - x)) & \textit{if } x \in U\text{-bounded}(v), \\ & \quad y \notin L\text{-bounded}(v), L_y = -\infty \\ (<, \infty) & \textit{otherwise} \end{cases}$$

▶ **Proposition 39.** *[15] The intersection $\langle v \rangle^\star \cap Z$ is empty iff there are two variables $x, y \in X \cup T$ s.t. $x \in U$-bounded(v), $L_y \neq -\infty$ when y is a process clock, and $H^v_{xy} + Z_{yx} < (\leq, 0)$.*

The above proposition gives a simple characterization for when the upward closure of a valuation v wrt to the $\preccurlyeq^\star_{LU}$ simulation does not intersect zone Z. Using this, we can show that when two valuations belong to the same (M, D)-region, then one of them satisfies this characterization iff the other does so. Here, we will make use of the fact that our atomic constraints involve integer constants, and hence all zones that appear in the local zone graph computation will only involve integer constants.

▶ **Lemma 40.** *Let L, U and M be bound functions such that for every process clock x, we have $M(x) \geq L(x)$ and $M(x) \geq U(x)$. For every zone Z, the set $\mathfrak{a}^\star_{\preccurlyeq LU}(\mathsf{spread}_D(Z))$ is a finite union of (M, D)-regions.*

Proof. We first remark that every valuation in $\mathfrak{a}^\star_{\preccurlyeq LU}(\mathsf{spread}_D(Z))$ is D-spread. Let v be a D-spread valuation. We have $v \in \mathfrak{a}^\star_{\preccurlyeq LU}(\mathsf{spread}_D(Z))$ iff $\langle v \rangle^\star \cap Z$ is non-empty. Let $v \equiv^D_M v'$. We will show that $\langle v \rangle^\star \cap Z$ is empty iff $\langle v' \rangle^\star \cap Z$ is empty. This will prove the lemma. Let H^v and $H^{v'}$ be the canonical distance graphs representing $\langle v \rangle^\star$ and $\langle v' \rangle^\star$ respectively.

From Proposition 39, $\langle v \rangle^\star \cap Z$ is empty iff there exist two variables x, y such that $x \in U$-bounded(v), and $L_y \neq -\infty$ when y is a process clock, such that $H^v_{xy} + Z_{yx} < (\leq, 0)$. As $x \in U$-bounded(v), we also have $x \in$ Bounded(v), as $M(x) = \max(L(x), U(x))$. Since $v \equiv^D_M v'$, we have $x \in U$-bounded(v) iff $x \in U$-bounded(v'). When $y \in$ Bounded(v), we have $\lfloor H^v_{xy} \rfloor = \lfloor H^{v'}_{xy} \rfloor$. Since Z_{yx} is of the form $(\lessdot, c)$ with c an integer, we have $H^v_{xy} + Z_{yx} < (\leq, 0)$ iff $H^{v'}_{xy} + Z_{yx} < (\leq, 0)$. When $y \notin$ Bounded(v), then in particular, y is a process clock, $y \notin L$-bounded(v) and we have $H^v_{xy} = (\leq, v(t_q - x)) + (<, -L_y)$ where q is the process containing y. But, t_q belongs to both Bounded(v) and Bounded(v'). Hence $\lfloor v(t_q - x) \rfloor = \lfloor v'(t_q - x) \rfloor$ and the lemma follows for this case using the previous argument. ◀