Robust Model-Checking of Timed Automata via Pumping in Channel Machines

Patricia Bouyer, Nicolas Markey, Ocan Sankur

Research report LSV-11-19

Laboratoire Spécification & Vérification

École Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Robust Model-Checking of Timed Automata via Pumping in Channel Machines

Patricia Bouyer, Nicolas Markey, Ocan Sankur

LSV, CNRS & ENS Cachan, France

Abstract. Timed automata are governed by a mathematical semantics which assumes perfectly continuous and precise clocks. This requirement is not satisfied by digital hardware on which the models are implemented. In fact, it was shown that the presence of imprecisions, however small they may be, may yield extra behaviours. Therefore correctness proven on the formal model does not imply correctness of the real system.
The problem of robust model-checking was then defined to circumvent this inconsistency. It consists in computing a bound on the imprecision under which the system will be correct.
In this work, we show that robust model-checking against ω-regular properties for timed automata can be reduced to standard model-checking of timed automata, by computing an adequate bound on the imprecision. This yields a new algorithm for robust model-checking of ω-regular properties, which is both optimal and valid for general timed automata.

1 Introduction

Timed automata [1] are a well-established model in real-time system design. These are finite automata augmented with *clocks*, which are used to measure the time elapsed between events, and to constrain the runs of the automaton. Timed automata provide a powerful way of modelling and verifying real-time systems. However, timed automata make idealistic assumptions on the system, such as the perfect continuity of clocks and instantaneous reaction time, which are known not to be preserved in implementation even in digital hardware with arbitrarily small imprecisions. It was shown that even the smallest imprecisions on the clocks yield a different semantics than the *exact* one [13,7] (see Fig. 2 for an example). This suggests that even if the exact semantics is proven correct, the implementation on a physical machine is not guaranteed to respect the specification. In order to prove the correctness of implementations, a framework was proposed in [8], where a detailed model of the implementation of timed automata is given, as programs executed on a simple micro-processor. A simpler over-approximation, the so-called *enlarged semantics* was also studied, which models the imprecisions by *relaxing* all clock constraints of the automaton of the form $x \in [a, b]$ to $x \in [a - \delta, b + \delta]$ for some $\delta > 0$. The problem of *robust model-checking*, that is determining whether for *some* $\delta > 0$, the enlarged semantics satisfies a given property, was first solved for safety properties [13,7], then for linear temporal logic (LTL) [4] (both in PSPACE, which is the complexity of the problem in the exact semantics), and for a timed extension of LTL [5].

These robust model-checking algorithms are all valid for a particular class of timed automata, namely, those in which all cycles are *progress cycles*. Roughly, a progress cycle is a cycle of the timed automaton which resets all clocks that are below the maximal constant at least once. We argue that this can be restrictive for modeling. In fact, a timed automaton model of a system under this assumption cannot measure the time spent in a cycle. As an example, consider a simple system which waits for a special signal, while ignoring any other signal, and triggers a time-out action if the expected signal is not received after one second (Fig. 1). In order to ignore any number of signals during this time, we need a cycle in the automaton. But if all clocks are reset on this cycle, then we cannot measure the time spent in it in order to issue the time-out. One could model such a system using progress cycles by explicitly defining an upper bound m on the number of events that can be treated by the system in one time unit, and unfolding the cycle for m iterations. This would remove the cycle. However this requires the prior knowledge of m which may not be obvious in the design phase, and moreover, this may increase the size of the model and render model-checking infeasible.

```
bool timeout := false;
clock x;
...
x := 0;
while ( x <= 1 ){
   ...
   if ( signal() == A )
      break;
}
if ( x >= 1 )
   timeout := true;
```

Fig. 1. A program that waits for a signal A and issues a time-out if it is not received in one time unit. A timed automaton model of this program naturally contains a non-progress cycle.

Our contribution. We propose a new algorithm for robust model-checking timed automata against ω-regular properties, with optimal complexity (PSPACE). Our algorithm consists in reducing the problem to classical model-checking of timed automata and is valid for general timed automata: we do not assume progress cycles, nor any upper bound on the clocks (Assuming bounded clocks is not restrictive in terms of expressivenes but has a negative effect on the size of the models [2]). We prove that any timed automaton satisfies a given ω-regular property under enlargement by *some* value $\delta > 0$ if, and only if, it satisfies the formula under enlargement by δ_0, where δ_0 only depends on the size of the timed automaton. Then the algorithm simply consists in model-checking the automaton enlarged by δ_0, which can be done using well-known algorithms and tools for timed automata. An algorithm was given in [4] for this problem but only for timed automata with progress cycles and bounded clocks, and because it is based on a modification of the region automaton construction, one cannot use directly the existing model-checking tools. For safety properties, an algorithm similar to ours can be derived from [7], but the complexity would be exponentially higher due to the bound given for δ_0. For automata without nested cycles, [11] gives an algorithm to compute the greatest δ under which a safety property holds, but does not provide a bound on δ.

Although the worst-case complexity of our algorithm is not higher than classical model-checking, in practice, the timed automaton enlarged by δ_0 can yield a model with a state space that is much larger than that of the initial automaton (see Section 3 for the precise value). However, we observed that this does not always increase the time and space necessary for verification. In fact, we used Uppaal [12] to test our algorithm on some benchmarks given with safety specifications. [1] We were able to show the non-robustness of the Fischer protocol upto three agents, and Uppaal returned almost immediately. With more than three agents, the problem is not due to time or space resources but to the fact that Uppaal only allows 32-bit integers as constants in timed automata; when δ_0 requires more precision, the model does not compile. However, Uppaal found counter-examples in less than one minute, for the protocol upto thirty agents, when enlarged by 10^{-8}. We believe that extending Uppaal with arbitrary precision integers, one should be able to use our algorithm for larger models. We could assess the robustness of the CSMA/CD protocol described in [14], the Bang & Olufsen Collision Detection Protocol [10], and the Token Ring Protocol upto thirty agents in less than one minute. Note that the correctness of a model under an enlargement δ implies the correctness for all $0 \leq \delta' < \delta$, so we only verified the above robust models under enlargement that we chose arbitrarily as $\delta = 10^{-6}$ (see [8]). All verification queries returned almost as fast as for the non-enlarged models.

In order to establish our results, we develop proof techniques based on the encoding of the states of timed automata with *channel machines*, introduced in [3], and used in [5] in the context of robustness. In this encoding, a word represents the content of a FIFO channel, which roughly contains all clock symbols ordered by their fractional parts. Time delays are simulated by sequences of read and writes on this channel, whereas action transitions also use a special renaming operation. It turns out that the finitary representation by these words capture well the behaviour of timed automata under enlargement. This was used in [5] to design a robust model-checking algorithm for a timed extension of LTL. We further develop these techniques and prove a pumping lemma for those channel machines, which preserves ω-regular properties. This enables us to prove new properties on the runs of enlarged timed automata, to refine some previously known results and obtain our algorithm. The proof follows the ideas of [13,7] but the techniques are different, and moreover, our analysis is finer since it yields an exponentially better bound for δ_0, as we also noted above.

2 Preliminaries

2.1 Timed Automata

A *labelled timed transition system (LTTS)* is a tuple $(S, s_0, \Sigma, \rightarrow)$, where S is the set of *states*, $s_0 \in S$ the initial state, Σ a finite alphabet, and $\rightarrow \subseteq S \times (\Sigma \cup \mathbb{R}_{\geq 0}) \times S$ the *transitions*.

[1] See `http://www.uppaal.org/benchmarks/`.

Given a finite set of clocks $\mathcal{C}$, we call *valuations* the elements of $\mathbb{R}_{\geq 0}^{\mathcal{C}}$. For a subset $R \subseteq \mathcal{C}$, a real $\alpha \in \mathbb{R}_{\geq 0}$ and a valuation v, we write $v[R \leftarrow \alpha]$ for the valuation defined by $v[R \leftarrow \alpha](x) = v(x)$ for $x \in \mathcal{C} \setminus R$ and $v[R \leftarrow \alpha](x) = \alpha$ for $x \in R$. Given $d \in \mathbb{R}_{\geq 0}$, the valuation $v + d$ is defined by $(v + d)(x) = v(x) + d$ for all $x \in \mathcal{C}$. We extend these operations to sets of valuations in the obvious way. We write $\mathbf{0}$ for the valuation which assigns 0 to every clock.

Let $\mathbb{Q}_\infty = \mathbb{Q} \cup \{-\infty, \infty\}$. An *atomic clock formula* is a formula of the form $k \leq x \leq l$ where $x \in \mathcal{C}$ and $k, l \in \mathbb{Q}_\infty$. A *guard* is a conjunction of atomic clock formulas. We denote by $\Phi_{\mathcal{C}}$ the set of guards on the clock set $\mathcal{C}$. We define the *enlargement* of atomic clock constraints by $\delta \in \mathbb{Q}$ as follows: for $x, y \in \mathcal{C}$ and $k, l \in \mathbb{Q}_{>0}$, we let

$$\langle k \leq x \leq l \rangle_\delta \;=\; k - \delta \leq x \leq l + \delta.$$

The enlargement of a guard g, denoted by $\langle g \rangle_\delta$, is obtained by enlarging all its atomic clock constraints. A valuation v *satisfies* a guard g, denoted $v \models g$, if all constraints are satisfied when each $x \in \mathcal{C}$ is replaced by $v(x)$. We denote by $[\![g]\!]$ the set of valuations that satisfy g.

Definition 1. *A timed automaton $\mathcal{A}$ is a tuple $(\mathcal{L}, \mathcal{C}, \Sigma, l_0, E)$, consisting of finite sets $\mathcal{L}$ of* locations*, $\mathcal{C}$ of* clocks*, Σ of* labels*, $E \subseteq \mathcal{L} \times \Phi_{\mathcal{C}} \times \Sigma \times 2^{\mathcal{C}} \times \mathcal{L}$ of edges, and where $l_0 \in \mathcal{L}$ is the* initial location*. An edge $e = (l, g, \sigma, R, l')$ is also written as $l \xrightarrow{g,\sigma,R} l'$. Guard g is called the* guard *of e.*

A timed automaton is *integral* if all constants that appear in its guards are integers. For any $\delta \in \mathbb{Q}$, $\mathcal{A}_\delta$ denotes the timed automaton where all guards are enlarged by δ. In the sequel, we only consider integral timed automata as input, and only their enlarged counterparts might not be integral.

Definition 2. *The semantics of a timed automaton $\mathcal{A} = (\mathcal{L}, l_0, \mathcal{C}, \Sigma, E)$ is an LTTS over alphabet Σ, denoted $[\![\mathcal{A}]\!]$, whose state space is $\mathcal{L} \times \mathbb{R}_{\geq 0}^{\mathcal{C}}$. The initial state is $(l_0, \mathbf{0})$.* Delay transitions *are defined as $(l, v) \xrightarrow{\tau} (l, v + \tau)$ for any state (l, v) and $\tau \geq 0$.* Action transitions *are defined as $(l, v) \xrightarrow{\sigma} (l', v')$, for any edge $l \xrightarrow{g,\sigma,R} l'$ in $\mathcal{A}$ such that $v \models g$ and $v' = v[R \leftarrow 0]$.*

Consider any timed automaton $\mathcal{A} = (\mathcal{L}, \mathcal{C}, \Sigma, l_0, E)$ and let $[\![\mathcal{A}]\!] = (S, s_0, \Sigma, \rightarrow)$. A *run* of $[\![\mathcal{A}]\!]$ is a finite or infinite sequence $\rho = (s_i, \sigma_i, \tau_i)_{i \geq 0}$, where $s_i = (l_i, v_i) \in S$, $\sigma_i \in \Sigma$, $\tau_i \in \mathbb{R}_{\geq 0}$ and $s_i \xrightarrow{\tau_i, \sigma_i} s_{i+1}$ for all $i \geq 0$. The word $l_0 l_1 \ldots$ is the *trace* of the run ρ, denoted $\mathsf{trace}(\rho)$. The i-th state s_i of a run ρ is denoted by $(\rho)_i$.

We define the usual notion of *regions* [1]. Pick a timed automaton $\mathcal{A}$ with clock set $\mathcal{C}$, and let M be the largest constant that appears in its guards. For any $(l, u), (l', v) \in \mathcal{L} \times \mathbb{R}_{\geq 0}^{C}$, we let $(l, u) \simeq (l', v)$ if, and only if, $l = l'$ and for all $x, y \in \mathcal{C}$, the following conditions are satisfied:

- either $\lfloor u(x) \rfloor = \lfloor v(x) \rfloor$ or $u(x), v(x) > M$;
- if $u(x) \leq M$, $\mathsf{frac}(u(x)) = 0$ iff $\mathsf{frac}(v(x)) = 0$;
- if $u(x), u(y) \leq M$, $\mathsf{frac}(u(x)) < \mathsf{frac}(u(y))$ iff $\mathsf{frac}(v(x)) < \mathsf{frac}(v(y))$,

where $\mathsf{frac}(\cdot)$ denotes the fractional part. The equivalence class of a state (l, v) for the relation $\simeq$ is denoted by $\mathsf{reg}((l, v))$, and called a *region* of $\mathcal{A}$. The *region automaton* of $\mathcal{A}$ is a finite automaton $\mathcal{R}(\mathcal{A})$ defined as follows. The states of $\mathcal{R}(\mathcal{A})$ are regions r of $\mathcal{A}$. There is a transition from r to r' labelled by $\sigma \in \Sigma$ if there is an edge (l, g, σ, R, l') such that for some $(l, u) \in r$ and $d \geq 0$, $u + d \models g$, and $(l', u[R \leftarrow 0]) \in r'$. This automaton is known to be time-abstract bisimilar to $[\![\mathcal{A}]\!]$ [1]. The number W of regions is bounded by $|\mathcal{L}| \cdot (2M + 2)^{|\mathcal{C}|} \cdot |\mathcal{C}|! \cdot 2^{|\mathcal{C}|}$. A *progress cycle* in $\mathcal{A}$ is a cycle in $\mathcal{R}(\mathcal{A})$ along which each clock $x \in \mathcal{C}$ is either reset or remains larger than M.

2.2 Robust model-checking of timed automata

It has been remarked long ago that the semantics of timed automata is not realistic: while this was first exemplified by the so-called *Zeno runs*, the problem goes far beyond, and includes other convergence phenomena [6], or isolated traces [9].

Among the possible approaches to circumvent this problem, *robust model checking* was introduced in [13]: it consists in checking a given property on the extended version of the timed automaton under study; here, *extended* includes clock drifts (clocks may evolve at different rates between $1 - \epsilon$ and $1 + \epsilon$) and guard enlargement. Robust model checking consists in deciding the existence of positive values for ϵ and/or δ for which the property holds in the extended timed automaton. In this paper, we only focus on guard enlargement (*i.e.*, we assume $\epsilon = 0$, so that clocks won't drift); in that setting, robust model checking amounts to deciding the existence of a positive δ for which $\mathcal{A}_\delta$ satisfies a given property.

Take the timed automaton depicted on Fig. 2, and the property that the rightmost location ℓ_3 is never reached. While this property can be checked to hold under the classical semantics, any positive enlargement of the clock constraints will make location ℓ_3 reachable (see [7]); this timed automaton does not *robustly* fulfill the safety property.

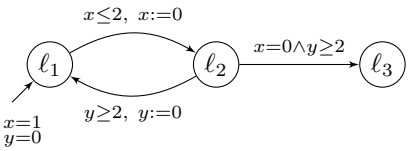

Fig. 2. A (non-robust) timed automaton

Robust model checking has been revisited recently in the setting of *implementability* [8]. Implementability also involves a new semantics for timed automata, the so-called *program semantics*, which simulates the execution of timed automata on a simplified hardware (with digital clock and finite-frequency CPU). This semantics can be over-approximated by the enlarged semantics, so that robust model checking provides an approximate technique to check implementability of timed automata [7].

Robust model checking was proved decidable for safety properties in [13], for timed automata in which all cycles are progress cycles. This was then extended to ω-regular properties [4], and then to timed properties [5].

3 Results

The following theorem is our main result.

Theorem 3. *Let $\mathcal{A}$ be a timed automaton and W be the number of regions of $\mathcal{A}$. Consider any $0 < \delta_0 < \left(8|\mathcal{C}|^2 \cdot (W+1)\right)^{-1}$ if $\mathcal{A}$ has only progress cycles, and $0 < \delta_0 < \left(5(W+1) \cdot |\mathcal{C}|^3 \cdot (2 \cdot |\mathcal{L}| \cdot |\mathcal{C}|! \cdot 4^{|\mathcal{C}|} + 4)^2\right)^{-1}$ otherwise. For any ω-regular property[2] ϕ, if $\mathcal{A}_\delta \models \phi$ for some positive δ, then $\mathcal{A}_{\delta_0} \models \phi$.*

Thus, one can decide robust satisfaction of any ω-regular property by checking whether the property holds for some fixed δ_0, which only depends on the size of the automaton. Now, using the usual model-checking algorithms, one can analyze $\mathcal{A}_{\delta_0}$ in polynomial space. In fact, the greatest constant in $\mathcal{A}$ is now multiplied by $\frac{1}{\delta_0}$ and the regions of $\mathcal{A}_{\delta_0}$ can still be encoded in polynomial space. The problem is PSPACE-hard since it is already for timed automata with progress cycles [5].

Corollary 4. *Robust model-checking of general timed automata against ω-regular properties is* PSPACE*-complete.*

The proof of Theorem 3 uses the encoding by channel machines proposed in [5]. The complex mechanism of the channel machine is not required for our purpose. We therefore hide it as much as possible and focus on the underlying transition system. The transition system and its relation to timed automata is presented in section 4. In section 5, we state our main technical results (namely, the pumping lemma and the cycling lemma), which we use to prove Theorem 3. The rest of the paper is then devoted to the proof of these lemmas.

Remark 1. The results of [7] can be lifted to the region-automaton construction, by adding extra transitions representing (progress) cycles [4]. Using our results, this can be further adapted by adding transitions corresponding to weak cycles, which can be detected on the transition system of the channel automaton.

4 Encoding by Channel Machines

In this section, we show how we encode the behaviour of $\mathcal{A}_\delta$ (where $\mathcal{A}$ is a timed automaton and $\delta > 0$) as the transition system of a channel machine. Channel machines are finite-state automata equipped with a FIFO channel. Intuitively, a state of $\mathcal{A}_\delta$ is encoded as follows: the location and the integer parts of the clocks are stored in a *discrete* location, while the channel contains the clock symbols, ordered according to their fractional parts. When a clock is popped out from the tail of the channel, it is (almost) immediately pushed back to the head of the

[2] With *ω-regular property*, we mean state-based properties whose truth value only depends on the set of locations that are visited infinitely often. By an adequate product, we could handle properties expressed by, say, deterministic Muller automata (hence including LTL properties); we omit these details to keep focus on the main objectives of this paper. For an ω-regular property ϕ, we write $\mathcal{A} \models \phi$ when all the runs of automaton $\mathcal{A}$ satisfy ϕ.

channel (hence it is assumed to have small fractional part). This corresponds to a delay transition along which that clock has changed integer value. Some additional symbols (Δ's) will appear on the channel, which serve for refining the region equivalence, and for approximating the values of the clocks. Our encoding is a slightly simplified version of [5], ignoring technicalities such as non-deterministic renaming and occurence testing operations. This is sufficient since the transition system will have access to the whole content of the channel, not only to the head and the queue (as this is the case for the standard mechanism of the channel machines).

We fix for the rest of this section a timed automaton $\mathcal{A} = (\mathcal{L}, \mathcal{C}, \Sigma, l_0, E)$, and a symbol $\Delta \notin \mathcal{C}$.

4.1 Channel Machine Associated to a Timed Automaton

For any word w over alphabet $2^{\mathcal{C}} \setminus \{\emptyset\} \cup \{\Delta\}$, $|w|_\Delta$ denotes the number of occurences of symbol Δ in w, and for any $x \in \mathcal{C}$, $|w|_x$ denotes the number of times x appears inside the symbols of $2^{\mathcal{C}}$ in w. For any integer $N > 0$, let Γ_N be the set of words w over alphabet $2^{\mathcal{C}} \setminus \{\emptyset\} \cup \{\Delta\}$ such that $|w|_\Delta = N$ and $|w|_x \leq 1$ for all $x \in \mathcal{C}$. For any $w \in \Gamma_N$, we define $\mathsf{right}^w_\Delta(x)$ as 0 if $|w|_x = 0$, and as the number of symbols Δ that appears on the right of the (unique) symbol containing x in w. We define $\mathsf{left}^w_\Delta(x)$ symmetrically.

Let M denote the largest constant that appears in $\mathcal{A}$. We assume that clocks are indexed by $\{1, \ldots, n\}$ for some $n > 0$, and we write $\mathcal{C} = \{x_1, \ldots, x_n\}$. We define *the channel machine associated with* $\mathcal{A}$ as the transition system $\mathcal{C}_\mathcal{A}(\Delta^N)$, parameterized by an integer $N \geq 0$, as follows. The states of $\mathcal{C}_\mathcal{A}(\Delta^N)$ are elements of $(\mathcal{L} \times 2^{\mathcal{C}} \times \{0, \ldots, M, \infty\}^{\mathcal{C}}) \times \Gamma_N$. The first component of a state q is the *discrete state*, made of a location, denoted by $\mathsf{loc}(q)$, the set of clocks that have integer values, and a mapping from clocks to their integer parts which is denoted by $\mathsf{int}(q)$ (we write ∞ if it is larger than M); the second component is the *channel content* where clocks are ordered according to their fractional parts. For a state $q = (d, w)$, we extend $\mathsf{right}_\Delta(\cdot)$ as $\mathsf{right}^q_\Delta(x) = \mathsf{right}^w_\Delta(x)$, and similarly for $\mathsf{left}^q_\Delta(x)$. The initial state of $\mathcal{C}_\mathcal{A}(\Delta^N)$ is $((l_0, \mathcal{C}, \mathbf{0}), \Delta^N)$, where l_0 is the initial location of $\mathcal{A}$. Forgetting Δ's, each state q of $\mathcal{C}_\mathcal{A}(\Delta^N)$ naturally encodes a region of $\mathcal{A}$, which we denote by $\mathsf{reg}(q)$: if $\mathcal{C} = \{x, y, z\}$, the state $((l, \{y\}, \binom{\lfloor x \rfloor = \lfloor y \rfloor = 2}{\lfloor z \rfloor = 1}), \Delta^2\{x\}\Delta\{z\}\Delta^4)$ encodes the region where $y = 2$, $\lfloor x \rfloor = 2$, $\lfloor z \rfloor = 1$ and $0 < \mathsf{frac}(x) < \mathsf{frac}(z)$. We will explain the role of the Δ's later.

Transitions of $\mathcal{C}_\mathcal{A}(\Delta^N)$ are labelled by $\Sigma \cup \{\tau\}$, where $\tau \notin \Sigma$. *Elementary delay transitions* are defined as follows, for any state $((l, Z, \iota), w)$:

$$\textit{(i)}\ ((l, Z, \iota), w) \xrightarrow{\tau} ((l, \emptyset, \iota), Z \cdot w) \qquad \textit{(ii)}\ ((l, \emptyset, \iota), w \cdot \Delta) \xrightarrow{\tau} ((l, \emptyset, \iota), \Delta \cdot w)$$
$$\textit{(iii)}\ ((l, \emptyset, \iota), w \cdot X) \xrightarrow{\tau} ((l, X', \iota'), w), \text{ where } \iota'(x) = \iota(x) + 1 \text{ for } x \in X,$$
$$\text{and } \iota'(y) = \iota(y) \text{ for } y \notin X, \text{ and } X' = X \cap {\iota'}^{-1}([0, M]),$$

where we write $M + 1 = \infty$ (all clocks whose integral part reaches $M + 1$ are abstracted to ∞ and they do not appear anymore in the word of Γ_N—this is the role of ${\iota'}^{-1}([0, M])$). *Delay transitions* are defined as the reflexive and transitive

closure of $\xrightarrow{\tau}$, and we also write $\xrightarrow{\tau}$ whenever $\xrightarrow{\tau}^*$. Viewing w as the content of a channel (the head being the first letter and the tail being the last letter), the delay transitions correspond to sequences of reads and writes at the channel, while the discrete state is changed to keep track of the integer parts, whenever a clock subset symbol is read. We say that a clock y *disappears* during a delay transition whenever the rule (iii) is applied, with $y \in X$ and $y \notin X'$. Obviously, delay transitions in $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ correspond to time elapsing in $\mathcal{A}$.

We now define when a state of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ satisfies a guard. A clock formula $x \leq k$ is *exactly satisfied* by a state $q = ((l, Z, \iota), w)$ if either $\iota(x) \leq k-1$, or $\iota(x) = k$ and $x \in Z$ (this is equivalent to say that $\mathsf{reg}(q)$ satisfies $x \leq k$). The formula is *satisfied* if either it is exactly satisfied or $\iota(x) = k$ and $\mathsf{left}^w_\Delta(x) \leq 1$. Intuitively, the value of x is then a bit larger than k (this will be made clearer when explaining the role of the Δ's). A formula $x \geq k$ is exactly satisfied if $\iota(x) \geq k$, and satisfied if it is exactly satisfied or if $\iota(x) = k-1$ and $\mathsf{right}^w_\Delta(x) \leq 1$. *Action transitions* are defined as follows. For any edge $l \xrightarrow{g,\sigma,R} l'$ of $\mathcal{A}$, we let $((l, Z, \iota), w) \xrightarrow{\sigma} ((l, Z \cup R, \iota'), w')$ if $((l, Z, \iota), w)$ satisfies g, $\iota'(x) = 0$ if $x \in R$ and $\iota'(x) = \iota(x)$ if $x \notin R$, and w' is obtained from w by removing the occurences of all clocks in R. This rule is not a valid operation in a channel machine, since some symbols may be removed from w, and checking guards requires reading the tail of w. However this can be simulated using rewriting and occurrence testing, see [5]. Action transitions in $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ where guards are exactly satisfied correspond to action transitions in $\mathcal{A}$. Non-exact satisfaction of guards represents enlarged timing constraints. We will see the precise correspondence later.

A *path* of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ is a sequence $\pi = (q_i, \sigma_i)_{i \geq 1}$ where q_i's are states of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$, and $\sigma_i \in \Sigma \cup \{\tau\}$, and there is a transition labelled by σ_i from q_i to q_{i+1}. We only consider w.l.o.g paths that are alternations of delay and action transitions. The *length* of π, denoted $|\pi|$, is the length of the sequence π. We denote by $\mathsf{trace}(\pi)$ the sequence of locations $\mathsf{loc}(q_0)\mathsf{loc}(q_2)\ldots$ visited by π, and by $\pi_{i\ldots j}$ the path defined by the subsequence between indices i and j. A path is *exact* if all guards in its transitions are satisfied exactly. The i-th state of a path π is denoted by $(\pi)_i$.

Representation of the states of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$. In the sequel, to help manipulate the transition system of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$, we use a *flat representation* of the states. We say that $((l, Z, \iota), \widetilde{w})$ is a *flat representation* of state $(d, w) = ((l, Z, \iota), w)$ whenever $\widetilde{w} \in \left(\mathcal{C} \cup \Delta \cup \Delta^{-1}\right)^*$ can be written as

$$\widetilde{w} = \Delta^{n_0} x_{i_1} \Delta^{n_1} \ldots x_{i_m} \Delta^{n_m}, \tag{1}$$

where $\{x_{i_j} \mid 1 \leq j \leq m\} = \{x \in \mathcal{C} \mid \iota(x) \leq M\}$ is the set of clocks whose integral part is no more than M in (d, w) (some appear in w, some, whose values is integral, do not appear in w), $n_0, \ldots, n_m \geq -1$, and:

- if we remove the maximal prefix of the form $\Delta^{-1} y_1 \Delta^{-1} \ldots \Delta^{-1} y_p$,
- if we remove the maximal suffix of the form $y_p \Delta^{-1} \ldots \Delta^{-1} y_1 \Delta^{-1}$, and
- if we replace all maximal factors of the form $y_1 \Delta^{-1} \ldots \Delta^{-1} y_p$ by $\{y_1, \ldots, y_p\}$,

then we obtain w. Such a flat representation $(d, \widetilde{w})$ contains exactly the same information as w (though with some redundancy) but will be easier to manipulate.

Note that there can be several flat representations for a given state (since both $x\Delta^{-1}y$ and $y\Delta^{-1}x$ can be used to represent $\{x,y\}$). Two clocks x_{i_j} and $x_{i_{j+1}}$ which are separated by Δ^{-1} in $\widetilde{w}$ will belong to the same set in w, hence the corresponding clocks will have the same fractional part in $\mathsf{reg}((d,w))$. If $n_0 = -1$, then x_{i_1} has an integer value in (d,c), and similarly for x_{i_m} if $n_m = -1$. Notice that all clocks whose values are (strictly) less than $M+1$ are present in this word, even those having an integral value. When clock indices $i_1,\ldots,i_m$ are clear from the context, or implicit, we also represent the channel content (1) by its block sizes $(n_0,n_1,\ldots,n_m)$. The channel content in (1) defines $m+1$ *blocks*, which are words of $\Delta^* \cup \{\Delta^{-1}\}$ separated by the clock symbols. We enumerate these from 0 to m, and say, for example, that block i has *size* n_i. In the rest, we only use flat representations, for which we easily infer the transition relation.

Example 1. The following is a path in $\mathcal{C}_{\mathcal{A}}(\Delta^{14})$, for the timed automaton $\mathcal{A}$ depicted on Fig. 2. This path simulates the run of the automaton that enters location ℓ_1 with $x=1$ and $y=0$; delays in ℓ_1 for $1+\delta$ time units, and then moves to ℓ_2, resetting x along that transition. It then waits for $1-2\delta$ time units, until $y = 2-\delta$, and goes back to ℓ_1, and so on.

$$\big((\ell_1,\{x,y\},\binom{\lfloor x\rfloor=1}{\lfloor y\rfloor=0}),\Delta^{-1}x\Delta^{-1}y\Delta^{14}\big) \xrightarrow{\tau} \big((\ell_1,\emptyset,\binom{\lfloor x\rfloor=2}{\lfloor y\rfloor=1}),\Delta x\Delta^{-1}y\Delta^{13}\big) \xrightarrow[x:=0]{x\le 2}$$
$$\big((\ell_2,\{x\},\binom{\lfloor x\rfloor=0}{\lfloor y\rfloor=1}),\Delta^{-1}x\Delta y\Delta^{13}\big) \xrightarrow{\tau} \big((\ell_2,\emptyset,\binom{\lfloor x\rfloor=0}{\lfloor y\rfloor=1}),\Delta^{12}x\Delta y\Delta\big) \xrightarrow[y:=0]{y\ge 2}$$
$$\big((\ell_1,\{y\},\binom{\lfloor x\rfloor=0}{\lfloor y\rfloor=0}),\Delta^{-1}y\Delta^{12}x\Delta^2\big) \xrightarrow{\tau} \big((\ell_1,\emptyset,\binom{\lfloor x\rfloor=1}{\lfloor y\rfloor=2}),\Delta x\Delta^2 y\Delta^{11}\big)\cdots$$

4.2 Relation with Timed Automata

We now define the relation between $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ and $\mathcal{A}$, through a relation on their associated time-abstract transition systems. Next, we write $s \overset{\sigma}{\Longrightarrow} s'$ if $s \xrightarrow{\tau} s'' \xrightarrow{\sigma} s'$ for some state s'' (where $\xrightarrow{\tau}$ is a delay transition).

Definition 5. *Let $\mathcal{S} = (S, s_0, \Sigma, \to)$ be an LTTS. A relation $R \subseteq S \times S$ is a* two-way simulation *if for all $(s_1,s_2) \in R$, if $s_1 \overset{\sigma}{\Longrightarrow} s_1'$ for some $\sigma \in \Sigma$ then $s_2 \overset{\sigma}{\Longrightarrow} s_2'$ for some s_2' with $(s_1',s_2') \in R$, and if $s_1' \overset{\sigma}{\Longrightarrow} s_1$ for some $\sigma \in \Sigma$, then $s_2' \overset{\sigma}{\Longrightarrow} s_2$ for some s_2' with $(s_1',s_2') \in R$. A state s_2* simulates *a state s_1 whenever there exists a two-way simulation R such that $(s_1,s_2) \in R$. In that case we write $s_1 \sqsubseteq s_2$.*

For any state (d,w) of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ with $w \in \Gamma_N$, we define $\mathsf{concrete}\big((d,w)\big)$ as a subset of $\mathsf{reg}((d,w))$ as follows. It contains a state $(l,v) \in \mathsf{reg}((d,w))$, if, and only if, $l = \mathsf{loc}((d,w))$ and there exists $\delta \in \mathbb{R}^N_{\ge 0}$ that satisfies $0 < \delta(1) < \delta(2) < \ldots < \delta(N) < 1$, $\delta(i+1) - \delta(i) = \frac{1}{N}$ for all $1 \le i \le N-1$, and $\delta(i) \neq \mathsf{frac}(v(x))$ for all i and $x \in \mathcal{C}$ that appears in w, and, assuming that $\delta(i)$ is the value of the i-th Δ-symbol in w, the extended valuation $v \cup \{\delta(i)\}_{1\le i\le N}$ is ordered according to w.

For example, consider the state $(d,w) = \big((l,\{x\},(\binom{\lfloor x\rfloor=\lfloor y\rfloor=0}{\lfloor z\rfloor=1})),\Delta^3 z\Delta^3 y\Delta^4\big)$, and valuation v defined by $v(x)=0$, $v(y)=0.6$ and $v(z)=1.3$. We have

$v \in \mathsf{concrete}\big((d,c)\big)$ since for $\delta \in \mathbb{R}^{10}_{\geq 0}$ with $\delta(i) = 0.05 + \frac{i-1}{10}$, the ordering of the fractional parts of $v(x), v(y), v(z), \delta(1), \ldots, \delta(10)$ agree with that given in (d,w).

Lemma 6. *For any timed automaton $\mathcal{A}$ and $N \geq 1$, $[\![\mathcal{A}_{\frac{1}{N}}]\!] \sqsubseteq \mathcal{C}_{\mathcal{A}}(\Delta^N) \sqsubseteq [\![\mathcal{A}_{\frac{2}{N}}]\!]$.*

A weaker version of this lemma was proven in [5]. The two-way simulations in the above lemma are given by relations R defined between states of $[\![\mathcal{A}_\delta]\!]$ and $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ by $(l,v)R(d,w)$ iff $v \in \mathsf{concrete}\big((d,w)\big)$ and $l = \mathsf{loc}(d)$.

4.3 Δ-Distance in $\mathcal{C}_{\mathcal{A}}(\Delta^N)$

If q is a state of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$, we will denote by $[q]$ the topological closure of the region encoded by q. The following lemma characterizes the inclusion of region closures using flat representations and follows from definitions.

Lemma 7. *Let q and q' be two states of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$, and let (d,w) be a flat representation of q, with $w = \Delta^{n_0} x_{i_1} \Delta^{n_1} \ldots x_{i_m} \Delta^{n_m}$. Then $[q] \subseteq [q']$ iff w' has a flat representation of the form (d',w') where $w' = \Delta^{n'_0} x_{i_1} \Delta^{n_1} \ldots x_{i_m} \Delta^{n'_m}$, s.t.*

- $\mathsf{loc}(d) = \mathsf{loc}(d')$,
- *for every $0 \leq i \leq m$, $n'_i = -1$ implies $n_i = -1$, and*
- *there exists $1 \leq r \leq m$ s.t. $n_{r+1} = n_{r+2} = \cdots = n_m = -1$, and:*
  - *for every $1 \leq j \leq r$, $\mathsf{int}(d)(x_{i_j}) = \mathsf{int}(d')(x_{i_j})$,*
  - *for every $r < j \leq m$, $\mathsf{int}(d)(x_{i_j}) = \mathsf{int}(d')(x_{i_j}) + 1$.*

We now define an edit-distance between the states of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$, called the *$\Delta$-distance*. We define the Δ-distance between any pair of states q and q' as infinite unless $[q] \subseteq [q']$ or $[q'] \subseteq [q]$. Fix two flat representations (d,w) and (d',w') that satisfy the conditions in Lemma 7 with block sizes $\boldsymbol{n}$ and $\boldsymbol{n'}$. We define $d_\Delta\big(q,q'\big) = \sum_i (\max({n'_i}^+ - {n_i}^+, 0))$, and notice that this is independent of the choice of the flat representations. This function can be seen to be symmetric (by the fact that both words have the same total number of Δ symbols), and to satisfy the triangular inequality. However, when the function equals 0, this does not imply the equality between states due to the -1-sized blocks. This pseudo-distance has the following important property.

Lemma 8. *For any timed automaton $\mathcal{A}$ and $N \geq |\mathcal{C}| + 2$, for any states q, q' of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$, $d_\Delta(q,q') \leq \frac{N}{|\mathcal{C}|} - 2$ implies that $[q] \cap [q'] \neq \emptyset$.*

5 Proof

5.1 Proof of the Main Theorem

The main theorem is a consequence of the following lemma, using Lemma 6.

Lemma 9. *Let $\mathcal{A}$ be any timed automaton and let W denote its number of regions. Let $K_0 = 2|\mathcal{C}|! \cdot |\mathcal{L}| \cdot 4^{|\mathcal{C}|} + 4$, and $N_1 \geq 8|\mathcal{C}|^2 \cdot (W+1)$ if $\mathcal{A}$ has only progress cycles, and $N_1 \geq 5|\mathcal{C}|^3 \cdot K_0^2 \cdot (W+1)$ otherwise. For any ω-regular property ϕ, if there exists $N > 0$ such that $\mathcal{C}_{\mathcal{A}}(\Delta^N) \models \phi$, then $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1}) \models \phi$.*

We show that if $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1}) \not\models \phi$, then $\mathcal{C}_{\mathcal{A}}(\Delta^{N}) \not\models \phi$ for all $N > 0$. The case where $N < N_1$ is easy, and is implied in the following lemma.

Lemma 10. *For any timed automaton $\mathcal{A}$ and any $N > 0$, $\mathcal{C}_{\mathcal{A}}(\Delta^{N}) \sqsubseteq \mathcal{C}_{\mathcal{A}}(\Delta^{N-1})$.*

The idea is that any path of the channel machine can be carried out when a Δ symbol is removed from the channel. In fact, all guards satisfied in the former system are also satisfied when a Δ symbol is removed. This implies that $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1}) \sqsubseteq \mathcal{C}_{\mathcal{A}}(\Delta^{N})$, hence if the property is violated by a path of $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1})$, then $\mathcal{C}_{\mathcal{A}}(\Delta^{N})$ also has a path violating the property.

In the case where $N_1 < N$, we do not have a simulation between $\mathcal{C}_{\mathcal{A}}(\Delta^{N})$ and $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1})$, but assuming that $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1})$ has a path violating the desired property, we transform it into a path of $\mathcal{C}_{\mathcal{A}}(\Delta^{N})$ also violating the property. This transformation may modify the trace but it does not affect the satisfaction of ϕ (locations that appear infinitely often remain the same). This is stated in the following pumping lemma.

Lemma 11 (Pumping Lemma). *Consider a timed automaton $\mathcal{A}$, and let W denote its number of regions. Let $K_0 = 2|\mathcal{C}|!\cdot|\mathcal{L}|\cdot 4^{|\mathcal{C}|}+4$, and $N_1 \geq 8|\mathcal{C}|^2 \cdot (W+1)$ if $\mathcal{A}$ has only progress cycles, and $N_1 \geq 5|\mathcal{C}|^3 \cdot K_0^2 \cdot (W+1)$ otherwise. Then, for any path π of $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1})$, for any $L \geq 0$, there exists a path π' of $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1+L})$, such that the same set of locations appear infinitely often in π and π'.*

The rest of the paper is devoted to the proof of the pumping lemma. The path of $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1+L})$ is obtained by repeating some factors of the path of $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1})$, then repeating some factors of the resulting word, and repeating this a finite number of times. This operation preserves ω-regular properties.

Overview of the proof. We start by studying, in Subsection 5.2, how the sizes of the blocks evolve along a path. We characterize the blocks whose sizes do not become small along a path; these are called the blocks that *stay united*. We prove a pumping lemma for these blocks (Lemma 12). Then, we study, in Subsection 5.3, exact paths of $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1})$ and show that for any path of bounded length, there is an exact path that follows the same trace and that is close in terms of Δ-distance (Lemma 14). In Subsection 5.4, we apply the above results to bounded paths to prove the pumping lemma for unbounded paths.

5.2 Pumping Lemma: Bounded case

We fix a timed automaton $\mathcal{A} = (\mathcal{L}, \mathcal{C}, \Sigma, l_0, E)$ and $N > 0$. Let W denote the number of regions of $\mathcal{A}$. For any channel content $w \in \Gamma_N$ with clocks $x_{i_1}, \ldots, x_{i_m}$ and block sizes $(n_0, \ldots, n_m)$, and $L \geq 0$, we define $w[x_{i_j} \leftarrow x_{i_j}\Delta^L]$ as the word of Γ_{N+L} obtained from w by replacing the j-th block Δ^{n_j} with $\Delta^{n_j^+ + L}$. We assume that i_0 is an index fixed to 0. We extend the above definition to the 0-th block, by writing $w[x_{i_0} \leftarrow x_{i_0}\Delta^L] = w[x_{i_1} \leftarrow \Delta^L x_{i_1}]$, obtained by inserting L new Δ symbols in the 0-th block.

We fix the constant $N_0 = 2W + 2$. A block is *small* if it has size $\{-1, 0, 1\}$, *medium* if it has size $\{2, \ldots, N_0 - 1\}$, and *large* otherwise. An important observation about $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ is that, if a guard is satisfied at some state (d, w), then, when an arbitrary number of Δ symbols are inserted in medium/large blocks, the same guard is still satisfied. However, if we insert additional Δ symbols inside small blocks, then formulas which are satisfied but not exactly satisfied may not be satisfied anymore. We define a notion of *staying united* along a path for blocks. Intuitively, such blocks are those that are either always at least medium, or are cut into at least one medium/large block along the path. We then show that one can insert any number of Δ symbols inside a block that stays united, and adapt the original path of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ to a path in $\mathcal{C}_{\mathcal{A}}(\Delta^{N+L})$.

We define a relation on pairs of states and clock indices as follows. Let $q = (d, w)$ and $q' = (d', w')$ denote two flat representations of states of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ such that $q \xrightarrow{\tau} q'$ is an elementary delay transition. We let $(q, i) \prec (q', j)$ whenever for every integer $L > 0$, there is a delay transition $q[x_i \leftarrow x_i\Delta^L] \xrightarrow{\tau} q'[x_j \leftarrow x_j\Delta^L]$. This relation can be characterized rather easily by analysing all possible cases for elementary delays $q \xrightarrow{\tau} q'$, see Appendix C.1. This relation is extended to the transitive closure of delay transitions. Similarly, assuming $q \xrightarrow{\sigma} q'$ is an action transition, we write $(q, i) \prec (q', j)$ whenever for every integer L, there is an action transition $q[x_i \leftarrow x_i\Delta^L] \xrightarrow{\sigma} q'[x_j \leftarrow x_j\Delta^L]$. This can be characterized easily as well, see the appendix.

For any finite path π in $\mathcal{C}_{\mathcal{A}}(\Delta^N)$, and any block i_1 in $(\pi)_1$, we say that *block* i_1 *stays united* in π, if block i_1 is large in $(\pi)_1$, and if there exists clock indices $i_2, \ldots, i_m$ such that $((\pi)_1, i_1) \prec ((\pi)_2, i_2) \prec \ldots \prec ((\pi)_n, i_n)$, with $n = |\pi|$.

By definition, the paths along which a block stays united are not sensitive to the precise size of those blocks. This is formalized in the following lemma, which is a pumping lemma for particular finite paths.

Lemma 12. *Let π be a path of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ such that $((\pi)_1, i_1) \prec ((\pi)_2, i_2) \prec \ldots \prec ((\pi)_n, i_n)$, for some $n > 0$. Then for any $L > 0$, $\mathcal{C}_{\mathcal{A}}(\Delta^{N+L})$ has a path π' with $(\pi')_j = (\pi)_j[x_{i_j} \leftarrow x_{i_j}\Delta^L]$ for any $1 \leq j \leq n$ and $\mathsf{trace}(\pi) = \mathsf{trace}(\pi')$.*

We now state a lower bound on the length of a path along which a block does not stay united. The idea is that if a block does not stay united, then whenever it is cut in two parts during a transition, either one of the resulting blocks is small, or none of them stays united in the rest of the path.

Lemma 13. *Let π be a path of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ with $|\pi| = p$. Then all blocks in $(\pi)_1$ that have size at least $p + 1$ stay united along π.*

5.3 Making Exact Paths

In this section, we show how to transform an arbitrary path of bounded length of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ into an exact one. By definition, if a path is not exact, then there are states with small blocks. The idea of our transformation is to replace all small blocks and the blocks that do not stay united by -1-sized blocks, while preserving

the ordering of the clocks. Notice that by Lemma 7, the states obtained by this operation define closed subregions of those defined by the original states. Clearly, if all small blocks have size -1, then any guard that is satisfied by a state is satisfied exactly, so the new path is exact.

Take $N \geq (|\mathcal{C}|+1) \cdot N_0$, and fix a path π of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ of length $n \leq N_0 - 1$. For each $1 \leq i < i' \leq n$, we associate with $(\pi)_i$ a state $H(\pi, i, i')$ where any small or medium block that does not stay united along $\pi_{i \ldots i'}$ is replaced by blocks of size -1. Formally, let $j_1, \ldots, j_k$ denote the indices of the blocks of $(\pi)_i$ that do not stay united along the path $\pi_{i \ldots i'}$. Let us write $(\pi)_i = (d, \boldsymbol{n})$. We define $H(\pi, i, i') = (d', \boldsymbol{n}')$ as follows. We let $n'_{j_1} = \ldots = n'_{j_k} = -1$, and $n'_{j_0} = n_{j_0} + n^+_{j_1} \ldots + n^+_{j_k}$, for the large block with the minimal index j_0, which exists by the choice of N. (Notice that the closed region $[H(\pi, i, i')]$ is independent of j_0). We have, by Lemma 7, $[H(\pi, i, i')] \subseteq [(\pi)_i]$. The same lemma implies that any state q with $[q] \subseteq [H(\pi, i, i')]$, has only blocks that stay united along $\pi_{i \ldots i'}$. Observe that because $(\pi)_i$ has at least one large block, by Lemma 13, $H(\pi, i, i')$ is well-defined. Last, we have $d_{\Delta}((\pi)_i, H(\pi, i, i')) \leq |\mathcal{C}| \cdot N_0$ by construction.

We construct exact paths that are "close" to the original ones, as follows.

Lemma 14. *Let $\mathcal{A}$ be any timed automaton having only progress cycles, and $N \geq (|\mathcal{C}|+1) \cdot N_0$. Let π a path of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ of length at most N_0. Then, there exists an exact path π' of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ over trace $\mathsf{trace}(\pi_{1 \ldots N_0})$, with $(\pi')_1 = H(\pi, 1, N_0)$ and $[(\pi')_i] \subseteq [H(\pi, i, N_0)]$ and $d_{\Delta}(\pi, \pi') \leq (|\mathcal{C}|+1)N_0$ for all $1 \leq i \leq N_0$.*

A result similar to Lemma 14 was given in [7, Th. 44] for runs of timed automata and distance d_∞ over valuations. The proof there involved approximation of the width of parametric DBMs. Our approach is in some sense closer to the input timed automaton, which may explain why we get an exponentially better distance to the original run.

As one might expect, exact paths satisfy the following property. The idea is that exact paths of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ are not sensitive to the sizes of the blocks.

Lemma 15. *Let $\mathcal{A}$ be any timed automaton, $N \geq 1$ and π an exact path of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$. Then, for any $N' \geq N$, and any state q of $\mathcal{C}_{\mathcal{A}}(\Delta^{N'})$ with $[q] \subseteq [\mathsf{first}(\pi)]$, there exists an exact path π' over the same trace as π, with $\mathsf{first}(\pi') = q$ and $[(\pi')_i] \subseteq [(\pi)_i]$ for all $1 \leq i \leq |\pi|$. The same property holds backwards: for any $q \in [\mathsf{last}(\pi)]$, there exists an exact path π' over the trace of π in $\mathcal{C}_{\mathcal{A}}(\Delta^{N'})$ with $\mathsf{last}(\pi') = q$ and $[(\pi')_i] \subseteq [(\pi)_i]$ for $1 \leq i \leq |\pi|$.*

5.4 Pumping Lemma with Progress Cycles: Unbounded case

The previous sections dealt with the properties of the bounded paths of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$. We now use these to prove the pumping lemma for infinite paths. Let us first define a transformation on the traces of the runs. For any finite trace $w \in \mathcal{L}^*$, we let $\widetilde{w} = \{u_1^+ u_2^+ \ldots u_n^+ \mid u_1 u_2 \ldots u_n = w\}$.

We first need the following lemma, which is an adaptation of Lemma 29 of [7] to channel machines. The proof is given in Appendix F for a more general case (for a generalization of the notion of progress cycles).

Lemma 16 (Cycling Lemma). *Let $\mathcal{A}$ be any timed automaton, $N \geq 1$ and π an exact progress cycle in $\mathcal{C}_{\mathcal{A}}(\Delta^N)$. Then, for all states q with $[q] \subseteq [\mathsf{last}(\pi)]$, there exists a path π' in $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ with $\mathsf{first}(\pi') = \mathsf{first}(\pi)$ and $[\mathsf{last}(\pi')] \subseteq [q]$, and $\mathsf{trace}(\pi') \in \widetilde{\mathsf{trace}(\pi)}$.*

We are now ready to prove the pumping lemma, for timed automata with progress cycles. Figure 3 illustrates a step of the proof.

Proof (of Lemma 11). We prove the result for $1 \leq L \leq |\mathcal{C}|N_0 - 2$. For larger L, one can repeat this construction. Let $N \geq 4|\mathcal{C}|^2 N_0$, and consider an infinite path π of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ where $\mathsf{first}(\pi)$ is the initial state of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$.

Let $n = N_0 - 1$. Let $G^L(\pi, i, i+n)$ denote the state of $\mathcal{C}_{\mathcal{A}}(\Delta^{N+L})$ obtained from $(\pi)_i$ by inserting Δ^L in the block with minimal index that stays united along $\pi_{i \dots i+n}$ (such a block exists by Lemma 13). We also define $H^L(\pi, i, i+n)$ by inserting Δ^L to the same block in $H(\pi, i, i+n)$ ($H(\cdot)$ is defined right before Lemma 14). Then, by construction, $d_\Delta\big(H^L(\pi, i, i+n), G^L(\pi, i, i+n)\big) \leq |\mathcal{C}|N_0$.

We now define a path π' of $\mathcal{C}_{\mathcal{A}}(\Delta^{N+L})$ over trace $\widetilde{\mathsf{trace}(\pi)}$. At each step $i \geq 1$, we construct $\pi'_{\beta_i \dots \beta_{i+1}}$, where $(\beta_i)_{i \geq 1}$ is an increasing sequence. Our construction satisfies $\mathsf{trace}(\pi'_{\beta_i \dots \beta_{i+1}}) \in \widetilde{\mathsf{trace}(\pi_{\alpha_i \dots \alpha_{i+1}})}$, and $[(\pi')_{\beta_i}] \subseteq [(\pi)_{\alpha_i}]$, for some possibly different increasing sequence $(\alpha_i)_{i \geq 1}$.

We define $(\pi')_1$ as the initial state of $\mathcal{C}_{\mathcal{A}}(\Delta^{N+L})$, which satisfies $[(\pi')_1] \subseteq [(\pi)_1]$. Suppose now that $\pi'_{1 \dots \beta_i}$ has been constructed for some $\beta_i \geq 1$. By Lemma 12, there is a path g of $\mathcal{C}_{\mathcal{A}}(\Delta^{N+L})$ from $G(\pi, \alpha_i, \alpha_i + n)$ over $\mathsf{trace}(\pi_{\alpha_i \dots \alpha_i + n})$, such that $(g)_j = (\pi)_{\alpha_i + j}[z_j \leftarrow z_j \Delta^L]$ for some clocks z_j. We then apply Lemma 14 to g to get an exact path h with $(h)_1 = H^L(\pi, \alpha_i, \alpha_i + n)$ over the trace of g, with $d_\Delta\big((h)_j, (g)_j\big) \leq (|\mathcal{C}| + 1)N_0$, and $[(h)_j] \subseteq [H^L(\pi, \alpha_i + j, \alpha_i + n)]$ for all $1 \leq j \leq n$. Now, h contains at least $n/2 \geq W$ action transitions, so there exist $1 \leq l_0 < l_1 \leq n$ such that $\mathsf{reg}((h)_{l_0}), \mathsf{reg}((h)_{l_0+1}), \dots, \mathsf{reg}((h)_{l_1})$ is a progress cycle of the region automaton of $\mathcal{A}$. We have $d_\Delta\big((h)_{l_1}, H^L(\pi, \alpha_i + l_1, \alpha_i + l_1 + n)\big) \leq$

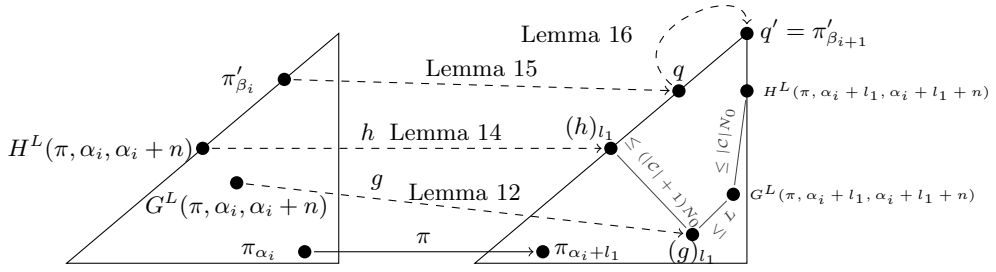

Fig. 3. An induction step of the proof of Lemma 11. Two triangles represent two closed regions. Their sides, interiors and corners are subregions. The proof constructs the dashed paths bottom-up.

$(3|\mathcal{C}|+1)N_0 - 2$ by combining the following inequalities:

$$d_\Delta((h)_{l_1}, (g)_{l_1}) \leq (|\mathcal{C}|+1)N_0,$$
$$d_\Delta\big((g)_{l_1}, G^L(\pi, \alpha_i + l_1, \alpha_i + l_1 + n)\big) \leq L \leq |\mathcal{C}|N_0 - 2,$$
$$d_\Delta\big(G^L(\pi, \alpha_i + l_1, \alpha_i + l_1 + n), H^L(\pi, \alpha_i + l_1, \alpha_i + l_1 + n)\big) \leq |\mathcal{C}|N_0.$$

By Lemma 15, $\mathcal{C}_\mathcal{A}(\Delta^{N+L})$ has a path over $\mathsf{trace}(h)$ from $(\pi')_{\beta_i}$ to some state q with $[q] \subseteq [(h)_{l_1}]$. By Lemma 8, we get $[(h)_{l_1}] \cap [H^L(\pi, \alpha_i + l_1, \alpha_i + l_1 + n)] \neq \emptyset$. Lemma 16 provides a path of $\mathcal{C}_\mathcal{A}(\Delta^{N+L})$ from q to some state q' with $[q'] \subseteq [(h)_{l_1}] \cap [H^L(\pi, \alpha_i + l_1, \alpha_i + l_1 + n)]$, over a trace in $\mathsf{trace}(h_{l_0 \ldots l_1})^+$. The concatenation of these two paths define $\pi'_{\beta_i .. \beta_{i+1}}$. This concludes a step of the induction. □

5.5 Pumping Lemma with Non-Progress Cycles

In this subsection, we explain the generalization of the proof of the pumping lemma to the case of timed automata with non-progress cycles. Let us call *weak cycle*, a path π of $\mathcal{C}_\mathcal{A}(\Delta^N)$ with $[\mathsf{last}(\pi)] \subseteq [\mathsf{first}(\pi)]$ that is not a progress cycle. Thus, π is a cycle along which at least one clock that is present on the channel is not reset. We show that all weak cycles can be transformed into *weak quasi-exact cycles* (defined below). We then define *quasi-exact paths* of $\mathcal{C}_\mathcal{A}(\Delta^N)$, which are paths that are exact except along weak quasi-exact cycles. Quasi-exact paths behave very much like exact paths: We adapt each of the previous lemmas involving exact paths for quasi-exact paths, and the proof in the presence of weak cycles is very similar to the proof which assumes progress cycles.

Example 2. The following path of $\mathcal{C}_\mathcal{A}(\Delta^{10})$ is an example of weak cycle, in which clock z is not reset.

$$(\ell_1, \{x\}, \cdot), \Delta^3 y \Delta^4 z \Delta^3 \xrightarrow{\tau} (\ell_1, \emptyset, \cdot), \Delta^2 x \Delta^3 y \Delta^4 z \Delta \xrightarrow{\sigma}$$
$$(\ell_2, \{y\}, \cdot), \Delta^2 x \Delta^7 z \Delta \xrightarrow{\tau} (\ell_2, \emptyset, \cdot), y\Delta^2 x \Delta^7 z \Delta \xrightarrow{\sigma'} (\ell_1, \{x\}, \cdot), y\Delta^9 z \Delta$$

It can be seen on this example how clock z prevents from accessing the Δ's that accumulate immediately on its left.

Given a weak cycle, clocks that are reset along the cycle are called *active*, and others *inactive*. Consider a weak cycle π. Suppose that $A = \{i_1, \ldots, i_r\}$ are the indices of the active clocks in π, given in the order of their appearance in $\mathsf{last}(\pi)$. Then, π can be factorized as,

$$\pi = \pi_{i_r} \pi_{i_{r-1}} \ldots \pi_{i_1} \pi', \tag{2}$$

where π_{i_j} ends with the last reset of the clock x_{i_j} in π, and no clock is reset in π'. An important observation that we use is that the size of the block i_j in $\mathsf{last}(\pi_{i_j})$ is determined by the number of Δ's read inside π_{i_j}, for any $1 \leq j \leq r-1$. The block i_r is particular, since its size at the last state is the sum of all the blocks $i_1, \ldots, i_r$ in $\mathsf{first}(\pi)$, plus the Δ's read during π_{i_r}. Among the blocks A of $\mathsf{last}(\pi)$, some will be called *pumpable*. Formally, for $K_0 = |\mathcal{L}| \cdot |\mathcal{C}|! \cdot 4^{|\mathcal{C}|+1} + 4$, we let

$$\mathsf{Pumpable}(\pi) = \{i_j \in A \mid \exists \tau \in \pi_{i_j}, \mathsf{time}_\Delta(\tau) \geq 2 \text{ or } \mathsf{time}_\Delta(\pi_{i_j}) \geq K_0\},$$

where $\mathsf{time}_\Delta(\tau)$ denotes the number of Δ's read from the channel in the delays of a given path τ, and with the abusive notation $\exists \tau \in \pi_{i_j}$ meaning "*for some delay transition in* π_{i_j}". We prove a pumping lemma for pumpable blocks inside weak cycles. In fact, if π_{i_j} has a delay transition which reads at least two Δ symbols, then block i_0 becomes medium or large during this delay, so it can be extended to read more Δ symbols. But if all delay transitions are too short then this trick cannot be used; in that case when a large number of transitions occur inside π_{i_j}, we show that some factor can be repeated, while additional delays that read Δ's are inserted (this is why we need a large constant K_0).

We then define *quasi-exact paths*, which are paths that are made of delays, exact transitions and weak cycles in which any block that is active is either pumpable, or it ends with size -1. We show that these paths behave very much like exact paths, and we follow step-by-step the same lemmas to construct the proof of the pumping lemma in the general case. Details are given in Appendix E.

References

1. R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
2. P. Bouyer and F. Chevalier. On conciseness of extensions of timed automata. *Journal of Automata, Languages and Combinatorics*, 10(4):393–405, 2005.
3. P. Bouyer, N. Markey, J. Ouaknine, and J. Worrell. The cost of punctuality. In LICS'07, p. 109–118, Wrocław, Poland, 2007. IEEE Computer Society Press.
4. P. Bouyer, N. Markey, and P.-A. Reynier. Robust model-checking of linear-time properties in timed automata. In LATIN'06, LNCS 3887, p. 238–249. Springer, 2006.
5. P. Bouyer, N. Markey, and P.-A. Reynier. Robust analysis of timed automata via channel machines. In FoSSaCS'08, LNCS 4962, p. 157–171. Springer, 2008.
6. F. Cassez, T. A. Henzinger, and J.-F. Raskin. A comparison of control problems for timed and hybrid systems. In HSCC'02, LNCS 2289, p. 134–148. Springer, 2002.
7. M. De Wulf, L. Doyen, N. Markey, and J.-F. Raskin. Robust safety of timed automata. *Formal Methods in System Design*, 33(1-3):45–84, 2008.
8. M. De Wulf, L. Doyen, and J.-F. Raskin. Almost ASAP semantics: From timed models to timed implementations. *Formal Aspects of Comput.*, 17(3):319–341, 2005.
9. V. Gupta, T. A. Henzinger, and R. Jagadeesan. Robust timed automata. In HART'97, LNCS 1201, p. 331–345. Springer, 1997.
10. K. Havelund, A. Skou, K. G. Larsen, and K. Lund. Formal modeling and analysis of an audio/video protocol: an industrial case study using uppaal. In RTSS'97. IEEE Computer Society, 1997.
11. R. Jaubert and P.-A. Reynier. Quantitative robustness analysis of flat timed automata. In FoSSaCS'11, LNCS 6604, p. 229–244. Springer, 2011.
12. K. G. Larsen, P. Pettersson, and W. Yi. Uppaal in a nutshell. *Int. Journal on Software Tools for Technology Transfer*, 1:134–152, 1997.
13. A. Puri. Dynamical properties of timed systems. *Discrete Event Dynamic Systems*, 10(1-2):87–113, 2000.
14. S. Yovine. Kronos: A verification tool for real-time systems. *International Journal on Software Tools for Technology Transfer*, 1:123–133, 1997.

A Additional Definitions

Consider a path π of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$. The *duration* of π, denoted $\mathsf{time}(\pi)$ is the number of times an elementary delay is applied in π. The Δ-*duration* of π, denoted $\mathsf{time}_\Delta(\pi)$ is defined as follows: if $\mathsf{time}(\pi) > 0$, then it is the number of times the rule (ii) is applied in the delay transitions of π (this is the number of Δ symbols read and rewritten on the channel), and otherwise it is -1. The *transition trace* of π, written as $\mathsf{trans}(\pi)$, is the sequence $(\sigma_i)_{i\geq 1}$ of transitions. When the path π is implicit, we abusively write $\mathsf{time}(\sigma_i)$ and $\mathsf{time}_\Delta(\sigma_i)$ for the (Δ-)duration of a particular transition in π.

B Proofs of Section 4

Lemma 6. *For any timed automaton $\mathcal{A}$ and $N \geq 1$, $[\![\mathcal{A}_{\frac{1}{N}}]\!] \sqsubseteq \mathcal{C}_{\mathcal{A}}(\Delta^N) \sqsubseteq [\![\mathcal{A}_{\frac{2}{N}}]\!]$.*

Proof. – $[\![\mathcal{A}_{\frac{1}{N}}]\!] \sqsubseteq [\![\mathcal{C}_{\mathcal{A}}(\Delta^N)]\!]$: We show that relation R defined by $(l,v)R(d,w)$, iff $(l,v) \in \mathsf{concrete}((d,w))$ is a two-way simulation. Consider $(l,v)R(d,w)$. There exists $0 < \delta(1) < \ldots < \delta(N) < 1$ such that (d,w) describes the region of the valuation $v \cup \delta(i)_i$ (for $|\mathcal{C}| + N$ clocks).
For delay transitions, we only need to consider delay $t > 0$ such that $\mathsf{reg}((l, v+t))$ is the immediate time-successor region of $\mathsf{reg}((l,v))$. Assume this is the case, and that $v(x) \in \mathbb{N}$ for some clock $x \in \mathcal{C}$. Consider the valuation $v \cup \delta + t$. The idea is to read and rewrite to the channel all Δ symbols corresponding to $\delta(i)$'s such that $\delta(i) + t \geq 1$. If $\delta(N) + t < 1$, then we only apply one elementary delay transition from (d,w), and get $(l, v+t) \in \mathsf{concrete}((d',w'))$ (clock x has integer value in (d,w)). Otherwise, there exists j such that $\delta(1) + t, \ldots, \delta(j) + t < 1$ and $1 \leq \delta(j+1) + t < \ldots < \delta(N) + t$. In this case, we apply a delay transition of Δ-duration $N - j$. The resulting state (d',w') satisfies $(l, v+t) \in \mathsf{concrete}((d',w'))$. Consider now the case where no clock has an integer value in (d,w). Then the immediate time successor is obtained by delaying until a clock, say x, becomes integer. There exists j such that $\delta(1), \ldots, \delta(j-1) < \mathsf{frac}(v(x)) < \delta(j), \ldots, \delta(N)$. We apply a delay of Δ-duration $N - j + 1$ and obtain $(l, v+t) \in \mathsf{concrete}((d',w'))$.
Consider an action transition. If (l,v) satisfies a guard then (d,w) does too. In fact, any clock constraint "$x \leq k$" or "$x \geq k$" with $k \in \mathbb{N}$ satisfied by (l,v) is also satisfied by (d,w) since $(l,v) \in \mathsf{reg}((d,w))$. If (l,v) satisfies "$x \leq k + \frac{1}{N}$", then $\mathsf{frac}(v(x)) \leq \frac{1}{N}$ so necessarily $\mathsf{frac}(v(x)) < \delta(2)$ and $\mathsf{left}_\Delta^{(d,w)}(x) \leq 1$. The case where "$x \geq k - \frac{1}{N}$" is similar. Finally, when some subset of clocks is reset, say $R \subseteq \mathcal{C}$, we have $(l, v[R \leftarrow 0]) \in \mathsf{concrete}((d',w'))$, where (d',w') is the target state of this transition.
Same arguments are valid backwards in both cases.

– $[\![\mathcal{C}_{\mathcal{A}}(\Delta^{2N})]\!] \sqsubseteq [\![\mathcal{A}_{\frac{1}{N}}]\!]$: We show that relation S defined by $(d,w)S(l,v)$, iff $(l,v) \in \mathsf{concrete}((d,w))$ is a two-way simulation. Consider $(d,w)S(l,v)$. There exists $0 < \delta(1) < \ldots < \delta(2N) < 1$ such that (d,w) describes the region of the valuation $v \cup \delta(i)_i$ (for $|\mathcal{C}| + 2N$ clocks). Consider an elementary delay

transition from (d, w) to some state (d', w'). Suppose that a clock x has an integer value in (d, w). By hypothesis, we have $v(x) \in \mathbb{N}$. Let ϵ denote the greatest fractional part of $\{v(y)\}_{y \in \mathcal{C}} \cup \{\delta(i)\}$. Then we apply a delay of duration $\frac{1}{2}(1-\epsilon)$. Notice that all clocks and $\delta(i)$'s have the same integer parts in the target state (l, v'), but x does not have an integer value anymore. Hence $(l, v') \in \mathsf{concrete}((d', w'))$. Suppose now that a clock has an integer value in (d, w). By hypothesis this is also the case for (l, v). Then, we apply a delay of duration ϵ (as defined above), and get $(l, v') \in \mathsf{concrete}((d', w'))$. We now consider an action transition from (d, w) to a state (d', w'). Any guard satisfied exactly in (d, w) is also satisfied in (l, v) as in the previous step. If (d, w) satisfies non-exactly a guard of the form "$x \leq k$" then we have $\mathsf{left}^w_\Delta(x) \leq 1$. In this case, $\mathsf{frac}(x) < \delta(2) < 2\frac{1}{2N} = \frac{1}{N}$ so it is satisfied by $[\![\mathcal{A}_{\frac{1}{N}}]\!]$. The guards of the form "$x \geq k$" are treated similarly. Finally, relation S is maintained when a subset of clocks is reset.

□

We need the following lemma from [7].

Lemma 17 ([7, Lemma 16]). *For any regions $r, r' \subseteq \mathbb{R}^{\mathcal{C}}_{\geq 0}$, $d_\infty(r, r') < \frac{1}{|\mathcal{C}|}$ implies $\mathsf{clos}(r) \cap \mathsf{clos}(r') \neq \emptyset$.*

Lemma 8 is a corollary of the previous and the following lemmas.

Lemma 18. *Let q, q' be states of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ where $N \geq |\mathcal{C}|$ and $[q] \subseteq [q']$. If $d_\Delta(q, q') \leq \frac{N}{|\mathcal{C}|} - 2$, then for any valuations $v \in \mathsf{concrete}(q)$ and $v' \in \mathsf{concrete}(q')$, $d_\infty(v, v') < \frac{1}{|\mathcal{C}|}$.*

Proof. Consider q, q' with $d_\Delta(q, q') \leq \frac{N}{|\mathcal{C}|} - 2$, and let $\boldsymbol{n}, \boldsymbol{n'}$ denote the block sizes, and $i_1, \ldots, i_m$ the clock indices as they appear in q and q'. Consider valuations $v \in \mathsf{concrete}(q)$ and $v' \in \mathsf{concrete}(q')$, and any $\delta, \delta' \in \mathbb{R}^N_{\geq 0}$, with $0 < \delta(1) < \ldots, \delta(N) < 1$ and $\delta(i+1) - \delta(i) = \frac{1}{N}$ and $\delta(i) \neq v(x)$ for all i and $x \in \mathcal{C}$ (and similarly for δ' and v'). Observe first that either $[q] \subseteq [q']$ or $[q'] \subseteq [q]$, so v and v' agree on the integer parts of the clocks in the sense of Lemma 7. Because $\delta(i)$'s are regularly spaced in the interval $(0, 1)$, the choice of $\delta(1)$ determines all $\delta(i)$ by $\delta(i) = \delta(1) + \frac{i-1}{N}$. Thus, there exists $0 \leq \epsilon < \frac{1}{N}$ such that $\delta' = \delta + \epsilon$ or $\delta' = \delta - \epsilon$. Assume the former. We will now show that $|\mathsf{frac}(v(x)) - \mathsf{frac}(v'(x))| < (\frac{N}{|\mathcal{C}|} - 2)\frac{1}{N}$. Let us write $\delta(0) = \delta'(0) = 0$. Consider any $x \in \mathcal{C}$ and let i, i' be maximal such that $\delta(i) \leq \mathsf{frac}(v(x))$ and $\delta'(i') \leq \mathsf{frac}(v'(x))$. The Δ-distance between q and q' implies that for any clock x_{i_j}, $({n'_{i_1}}^+ + \ldots + {n'_{i_j}}^+) - (n^+_{i_1} + \ldots + n^+_{i_j}) \leq \frac{N}{|\mathcal{C}|} - 2$, which means here $|i - i'| \leq \frac{N}{|\mathcal{C}|} - 2$. Hence, $|\delta(i) - \delta'(i')| \leq \frac{|i-i'|}{N} + \epsilon \leq \frac{1}{|\mathcal{C}|} - \frac{2}{N} + \epsilon$. We have $\mathsf{frac}(v(x)) - \delta(i) < \frac{1}{N}$ and $\mathsf{frac}(v'(x)) - \delta'(i') < \frac{1}{N}$ which implies

$$|(\mathsf{frac}(v(x)) - \mathsf{frac}(v'(x))) - (\delta(i) - \delta'(i'))| < \frac{1}{N}.$$

Triangular inequality yields $|\mathsf{frac}(v(x)) - \mathsf{frac}(v'(x))| < \frac{1}{|\mathcal{C}|} - \frac{1}{N} + \epsilon < \frac{1}{|\mathcal{C}|}$. □

C Additional material for Subsection 5.2

C.1 Formal definition of the $\prec$ relation

We assume $q = (d, w) \xrightarrow{\tau} q' = (d', w')$ is an elementary delay transition. We note $w = \Delta^{n_0} x_{i_1} \Delta^{n_1} x_{i_2} \Delta^{n_2} \dots x_{i_m} \Delta^{n_m}$ as a flat representation. We will define relation $(q, i_j) \prec (q', i'_k)$ only when $n_j \geq 2$. We distinguish between the different possibilities for the elementary delay transition (see page 7):

- Case (i): either $n_0 = -1$ or $n_m = -1$.
  We distinsguish between two cases:
  - if $n_m \geq 0$ (and therefore $n_0 = -1$), then $w' = x_{i_1} \Delta^{n_1} \dots x_{i_m} \Delta^{n_m}$. And in this case, for every $1 \leq j \leq m$ such that $n_j \geq 2$, $(q, i_j) \prec (q', i_j)$.
  - if $n_m = -1$, let k be such that $n_k \geq 0$ and for every $h > k$, $n_h = -1$. Then, we have $w' = x_{i_{k+1}} \Delta^{-1} \dots \Delta^{-1} x_{i_m} \Delta^{n_0} x_{i_1} \Delta^{n_1} \dots x_{i_k} \Delta^{n_k}$. Writing $w' = \Delta^{n'_0} x_{i'_1} \Delta^{n'_1} \dots$, we have that $x_{i'_1} = x_{i_{k+1}}$, $x_{i'_2} = x_{i_{k+1}}$, $\dots x_{i'_{m-k}} = x_{i_m}$, $x_{i'_{m-k+1}} = x_{i_1}$, $\dots x_{i'_m} = x_{i_k}$. Therefore, for every $0 \leq j \leq k$ such that $n_j \geq 2$, $(q, i_j) \prec (q', i'_{m-k+j})$.
- Case (ii): $n_0 \geq 0$ and $n_m \geq 1$.
  In this case, $w' = \Delta^{n_0+1} x_{i_1} \Delta^{n_1} \dots x_{i_m} \Delta^{n_m - 1}$. We thus get that for every $0 \leq j \leq m$ such that $n_j \geq 2$, $(q, i_j) \prec (q', i_j)$, and if $n_m \geq 2$ then $(q, i_m) \prec (q', i_0)$.
- Case (iii): $n_0 \geq 0$ and $n_m = 0$.
  We assume $k < m$, $n_k \geq 0$, and for all $k < j < m$, $n_j = -1$. Then, $w' = \Delta^{-1} x_{i_{k+1}} \Delta^{-1} \dots \Delta^{-1} x_{i_m} \Delta^{n_0} x_{i_1} \Delta^{n_1} \dots \Delta^{n_k}$, where some of the clocks $x_{i_{k+1}}, \dots, x_{i_m}$ might have been removed (because they have reached value $M + 1$). Assume the number of such clocks is r. We then have for every $0 \leq j \leq k$ such that $n_j \geq 2$, $(q, i_j) \prec (q', i'_{m-k-r+j})$.

Note that with this definition, if $(q, i) \prec (q', j)$, then block j in q' has size at least 2.

We extend $\prec$ to pair of states separated by an action transition. Suppose that $q \xrightarrow{\sigma} q'$ over some action transition. In these transitions, blocks can be merged or renamed due to clock resets. Let $i_1, \dots, i_m$ denote the clock indices in q. Let $R(\sigma)$ denote the set of clocks that are reset in this transition, and define for any clock index i_j, $\mathsf{NR}_{q,\sigma}(i_j) = i_1$ if $i_1, \dots, i_j \in R(\sigma)$, and otherwise $\mathsf{NR}_{q,\sigma}(i_j) = i_{\max\{1 \leq s \leq j : i_s \notin R(\sigma)\}}$. The intuition behind this definition is that the block i_j is merged with the blocks $i_s, i_{s+1}, \dots, i_{j-1}$ after transition σ, where $i_s = \mathsf{NR}_{q,\sigma}(i_j)$. We extend this definition to block i_0 by $\mathsf{NR}_{q,\sigma}(i_0) = i_0$ if $R(\sigma) = \emptyset$, and $\mathsf{NR}_{q,\sigma}(i_0) = i_{\min\{j : i_j \in R(\sigma)\}}$. Now, we let $(q, i_j) \prec (q', \mathsf{NR}_{q_2,\sigma}(i_j))$ for any pair of states with $q \xrightarrow{\sigma} q'$.

C.2 Proof of Lemma 13

Lemma 13. *Let π be a path of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ with $|\pi| = p$. Then all blocks in $(\pi)_1$ that have size at least $p + 1$ stay united along π.*

Proof. Remember that by definition of finite paths, π has $\frac{p-1}{2}$ action transitions, and $\frac{p-1}{2}$ delay transitions.

Let us consider a path π and a block l of $(\pi)_1$. We define a directed graph $G_{\prec}(\pi, l)$ where nodes are pairs of states and clock indices. We start by placing the node $((\pi)_1, l)$. Then, for each node $((\pi)_i, l')$ created in the previous step, we create a new node $((\pi)_{i+1}, l'')$ for all indices l'' such that $((\pi)_i, l') \prec ((\pi)_{i+1}, l'')$ and add an arc from the former to the latter. If the same node $((\pi)_{i+1}, l'')$ is a successor of two nodes, we do not duplicate it. Intuitively, the successors of a node are precisely those blocks in the successor state in π, where one can pump additional Δ's. Now the idea of the proof is to show that if a block does not stay united, then in this tree-like structure all branches have length less than p. See Fig. 4 for an example.

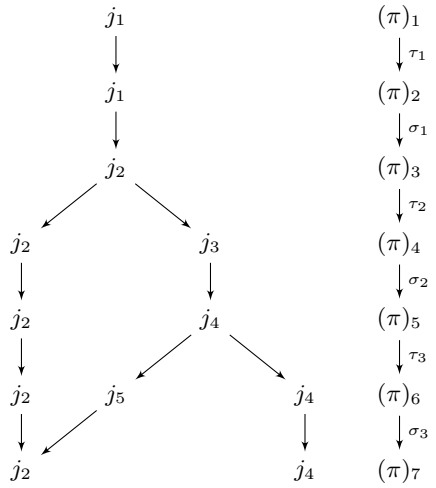

Fig. 4. An example of the graph $G_{\prec}(\pi, j_1)$ constructed in the proof of Lemma 13. A fictitious path π is shown at the right. Each level $i+1$ of the graph contains the blocks of $(\pi)_{i+1}$ which are $\prec$-successors of some block of $(\pi)_i$ that appear at level i. Several scenarios are possible: during a delay transition, a block can stay unchanged (for example, block j_1); or it can be split into two (for example, blocks j_2 and j_4). The latter case only occurs when the block is the rightmost block. Blocks can also be merged during action transitions (blocks j_2 and j_5).

We define an integer labelling $n(\cdot)$ on the nodes of $G_{\prec}(\pi, l)$, that gives lower bounds on the sizes of the corresponding blocks. We let $n((\pi)_1, l) = p + 1$. Assume we defined n upto some level k. If the $k+1$-th state is obtained by an action transition σ, then any node $((\pi)_k, i_j)$ has one successor which is $((\pi)_{k+1}, \mathsf{NR}_\sigma(i_j))$. Let us write $i_s = \mathsf{NR}_\sigma(i_j)$. We then define $n((\pi)_{k+1}, i_s) = n((\pi)_k, i_s) + n((\pi)_k, i_{s+1}) + \ldots + n((\pi)_k, i_j)$ (we assume that $n((\pi)_k, j) = 0$ if block j is not in $G_{\prec}(\pi, l)$). Observe that the sum of the labels of level k equals

that of level $k+1$. Suppose now that we have a delay transition. If some clock has an integer value in the target state, then blocks i_0 and i_m of $(\pi)_k$ are merged into block i_m, and block i_0 has size -1 in $(\pi)_{k+1}$ by hypothesis. We define $((\pi)_{k+1}, i_m) = n((\pi)_k, i_0) + n((\pi)_k, i_m)$. The sizes of all other blocks are preserved, and so are their labels. The sum of the labels of level $k+1$ equals that of level k in this case. If no clock has an integer value in the target state of this delay transition, then block i_0 has positive size in $(\pi)_{k+1}$. Let i_m denote the rightmost block in $(\pi)_{k+1}$. Notice that the sizes of all blocks other than i_0 and i_m are preserved. If there is an arc from some node of level k to $((\pi)_{k+1}, i_0)$, then block i_0 is medium or large. The same remark holds for $((\pi)_{k+1}, i_m)$. First, assume that block i_m is not the rightmost one in $(\pi)_k$. In this case, we define the labellings as maximal such that $n((\pi)_{k+1}, i_0) + n((\pi)_{k+1}, i_m) \leq n((\pi)_k, i_m)$, and respective labels are lower bounds to the sizes of blocks i_0 and i_m. Notice that we have equality here if none of the blocks is small (that is, both are in the graph $G_{\prec}(\pi, l)$). However, if one of the blocks is small, say i_0, then we may have $n((\pi)_{k+1}, i_m) = n((\pi)_k, i_m) - 1$. Hence, either the sum of the labels are preserved from one level to the next one, or it is decremented by at most 2 (the case where both i_0 and i_m have size less than 2 in the target state). One can continue this construction as long as there is at least one node at the current level, or equivalently, the sum of the labels are greater than 1. If we start with a block of size $p+1$, after the $p-1$-th transition, there is at least one node with a lower bound of at least 2. □

D Proofs of Subsection 5.3

Lemma 14. *Let $\mathcal{A}$ be any timed automaton having only progress cycles, and $N \geq (|\mathcal{C}|+1) \cdot N_0$. Let π a path of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ of length at most N_0. Then, there exists an exact path π' of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ over trace $\mathsf{trace}(\pi_{1\ldots N_0})$, with $(\pi')_1 = H(\pi, 1, N_0)$ and $[(\pi')_i] \subseteq [H(\pi, i, N_0)]$ and $d_\Delta(\pi, \pi') \leq (|\mathcal{C}|+1)N_0$ for all $1 \leq i \leq N_0$.*

Proof. We construct $(\pi')_i$ by induction on $i \geq 1$, which will satisfy the following properties.

- $[(\pi')_i] \subseteq [(\pi)_i]$,
- any block that does not stay united along $\pi_{i\ldots|\pi|}$ has size -1 in $(\pi')_i$.
- $d_\Delta((\pi)_i, (\pi')_i) \leq (|\mathcal{C}|+1)N_0$.

For each i, we will define a vector $\boldsymbol{k^i}$ such that $n'_j \leq n_j + k^i_j$, where $\boldsymbol{n}$ and $\boldsymbol{n'}$ denotes the block sizes of $(\pi)_i$ and $(\pi')_i$ respectively. The vector $\boldsymbol{k^i}$ is called the *excess vector* of the blocks of $(\pi')_i$ with respect to those of $(\pi)_i$. Notice that $d_\Delta((\pi)_i, (\pi')_i) \leq \|\boldsymbol{k^i}\|_1 = \sum_j k^i_j$. We will bound $\|\boldsymbol{k^i}\|_1$ at the end of the proof to show the third condition.

We let $(\pi')_1 = H(\pi, 1, n)$ which satisfies all conditions. We have $d_\Delta((\pi)_1, (\pi')_1) \leq |\mathcal{C}|N_0$ (see the remark preceding the lemma).

Suppose now that $\pi'_{1\ldots i}$ has been constructed, and let us write $(\pi)_i = (d, \boldsymbol{n})$ and $(\pi')_i = (d', \boldsymbol{n'})$. We distinguish two cases according to the nature of the transition from $(\pi)_i$ to $(\pi)_{i+1}$.

▶ Suppose $(\pi)_i \xrightarrow{\tau} (\pi)_{i+1}$. If a clock has integer value in $(\pi)_{i+1}$ (that is if $n_0 = -1$ or $n_m = -1$) then there is a sequence of delay transitions one can realize from $(\pi')_i$ which ends in a state where the same clock has integer value, and this is our choice for $(\pi')_{i+1}$ (this is because the clocks have the same ordering in $(\pi)_i$ and $(\pi')_i$). All conditions are satisfied in this case, and one can define $\boldsymbol{k^{i+1}}$ such that $\|\boldsymbol{k^{i+1}}\|_1 = \|\boldsymbol{k^i}\|_1$. If no clock has an integer value in $(\pi)_{i+1}$, then we we split the delay $(\pi)_i \xrightarrow{\tau} (\pi)_{i+1}$ into two: we first consider the maximal delay that ends with an integer clock and treat it as in the previous case (this delay is possibly empty), and then treat the last part of the delay where no clock becomes integer (*i.e.* the rule (iii) of the delay transitions is never applied). In this second part, the clocks have the same ordering in the source and target states of the delay. Thus, assume w.l.o.g. that this is the case in $(\pi)_i \xrightarrow{\tau} (\pi)_{i+1}$. This delay then has the following form.

$$(d, \Delta^{n_0} x_{i_1} \dots x_{i_m} \Delta^{n_m}) \xrightarrow{\tau} (d, \Delta^{n_0^+ + \alpha} x_{i_1} \dots x_{i_m} \Delta^{n_m - \alpha}).$$

We distinguish four cases.

1. If the block i_0 and i_m both stay united along $\pi_{i+1\dots n}$, then we can realize a delay transition of Δ-duration $\alpha' = \min(\alpha, n'_m)$ from $(\pi')_i$ to some state $(\pi')_{i+1}$ with $[(\pi')_{i+1}] \subseteq [(\pi)_{i+1}]$, ensuring that $\boldsymbol{k^i}$ is still an excess vector for $(\pi')_{i+1}$ w.r.t. $(\pi)_{i+1}$. In fact, we have $n'_0 \leq n_0 + k^i_0$ and $n'_m \leq n_m + k^i_m$, so we α' satisfies $n'_0 + \alpha' \leq n_0 + \alpha + k^i_0$ and $n'_m - \alpha' \leq n_m - \alpha + k^i_m$.
2. If block i_0 does not stay united along $\pi_{i+1\dots n}$ but block i_m does, then by induction hypothesis, we must have $n'_0 = -1$. In this case, we let $(\pi')_{i+1} = (\pi')_i$ (this is a trivial delay). Let p denote the difference between the sizes of block 0 between $(\pi)_i$ and $(\pi)_{i+1}$. Then the excess vector must be updated by $k^{i+1}_0 = 0$, $k^{i+1}_m = k^i_m + p$, and the other components are unchanged. We have $\|\boldsymbol{k^{i+1}}\|_1 \leq \|\boldsymbol{k^i}\|_1 + p$.
3. If block i_m does not stay united along $\pi_{i+1\dots n}$, but i_0 does, then block i_m has size -1 in $(\pi')_i$ by induction hypothesis. In this case we also choose $(\pi')_{i+1} = (\pi')_i$, and the excess vector is preserved.
4. If neither of the block stay united, then both have size -1, and again, we choose $(\pi')_{i+1} = (\pi')_i$ and the excess vector does not change.

▶ Suppose now that $(\pi)_i \xrightarrow{\sigma} (\pi)_{i+1}$, an action transition along an edge (l, σ, g, R, l'). Since $[(\pi')_i] \subseteq [(\pi)_i]$ and all small blocks have size -1, guard g is satisfied exactly by $(\pi')_i$. Let $(\pi')_{i+1}$ denote the target state when applying this action transition. Clearly, $[(\pi')_{i+1}] \subseteq [(\pi)_{i+1}]$. The excess vector needs to be updated since some blocks may have been merged due to clock resets. If $x_{i_j} \in R$, then all blocks $i_s, i_{s+1}, \dots, i_j$ get merged in the target state, where $i_s = \mathsf{NR}_\sigma((\pi)_i) = \mathsf{NR}_\sigma((\pi')_i)$. We let $k^{i+1}_s = k^i_s + \dots + k^i_j$ for all clock indices i_j which are maximal such that $i_s, i_{s+1}, \dots, i_j \in R$, and the other components of $\boldsymbol{k^{i+1}}$ are equal to those of $\boldsymbol{k^i}$. Then $\boldsymbol{k^{i+1}}$ defines indeed an excess vector for $(\pi')_{i+1}$ w.r.t. $(\pi)_{i+1}$ with $\|\boldsymbol{k^{i+1}}\|_1 = \|\boldsymbol{k^i}\|_1$.

We have $\|\boldsymbol{k^1}\|_1 \leq |\mathcal{C}|N_0$ as noted above. The norm of the excess vector only increases by p in case 2 of delay transitions and stay unchanged otherwise. This

happens only when a block of size p does not stay united along the rest of the path. Thus, either $p = 1$, or $p \geq 2$ and in that case, by Lemma 13, it takes more than $p + 1$ transitions to split this block to small parts (so that it does not stay united). In other terms, action transitions and the case 4 of the delay transitions above must happen at least $p+1$ times in the rest of the path, in such a way that at the end of the delay transitions, a block that does not stay united is split into blocks i_0 and i_m. But in these steps, the excess vector does not change (both blocks have size -1 by construction). Hence, $\|\boldsymbol{k^n}\| \leq \|\boldsymbol{k^1}\| + N_0 \leq (|\mathcal{C}| + 1)N_0$. □

Lemma 15. *Let $\mathcal{A}$ be any timed automaton, $N \geq 1$ and π an exact path of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$. Then, for any $N' \geq N$, and any state q of $\mathcal{C}_{\mathcal{A}}(\Delta^{N'})$ with $[q] \subseteq [\mathsf{first}(\pi)]$, there exists an exact path π' over the same trace as π, with $\mathsf{first}(\pi') = q$ and $[(\pi')_i] \subseteq [(\pi)_i]$ for all $1 \leq i \leq |\pi|$. The same property holds backwards: for any $q \in [\mathsf{last}(\pi)]$, there exists an exact path π' over the trace of π in $\mathcal{C}_{\mathcal{A}}(\Delta^{N'})$ with $\mathsf{last}(\pi') = q$ and $[(\pi')_i] \subseteq [(\pi)_i]$ for $1 \leq i \leq |\pi|$.*

Proof. From any state $[q] \subseteq [(\pi)_1]$, one can realize delay transitions so that at each step the clock have the same ordering as in $(\pi)_i$, that is $[(\pi')_i] \subseteq [(\pi)_i]$. Then, since all guards are satisfied exactly, they are also satisfied in $(\pi')_i$. Similar ideas were used in the proof of Lemma 14. □

E Pumping Lemma with Non-progress Cycles

A *weak path* of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ is a path along which at least one clock that is below the maximal constant at the target state is not reset, and such that no clock symbol is written on the channel (in other terms, the integer parts of the clocks do not change, except for the subset of clock that are rightmost in the channel, whose integer parts can be incremented but they are not rewritten on the channel). The clocks that are reset along a given weak path are called *active* and the others *inactive*. The path in Example 1 is a weak path, where x, y are active clocks and z is inactive.

Consider a weak path π, and let $x_{i_1}, \ldots, x_{i_m}$ be the clocks ordered as they appear in state $\mathsf{last}(\pi)$, and let $x_{i_1}, \ldots, x_{i_r}$ denote the active clocks. Then, π can be factorized as in (2). In the rest, clocks $x_{i_1}, \ldots, x_{i_r}$ will denote the active clocks, as they appear in $\mathsf{last}(\pi)$.

A *weak cycle* is a weak path π of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ such that $\mathsf{loc}((\pi)_1) = \mathsf{loc}((\pi)_p)$ and $[(\pi)_p] \subseteq [(\pi)_1]$. Note that a weak path is not a weak cycle in general, since a clock that is not in the channel content in $(\pi)_1$ may be reset along the path.

In the rest, when we consider a weak path, the clocks that appear in the last state will be denoted $i_1, \ldots, i_m$, and $i_1, \ldots, i_r$ will denote the set of active clocks.

E.1 Weak Paths

Without loss of generality, we consider weak paths and cycles for which $\mathsf{time}(\pi') = 0$ in (2). Given (2), state $\mathsf{last}(\pi)$ is determined, as shown by the following lemma.

Lemma 19. *Let $\pi = \pi_{i_r} \dots \pi_{i_1} \pi'$ be a weak path of length p of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ written as in (2). Let us write $(\pi)_1 = (d, \boldsymbol{n})$ and $(\pi)_p = (d', \boldsymbol{n'})$. Then, $n'_j = \mathsf{time}_\Delta(\pi_{i_j})$ for all $1 \leq j \leq r-1$ and $n'_r = n_r + \mathit{left}_\Delta^{(\pi)_1}(x_{i_r}) + \mathsf{time}_\Delta(\pi_{i_r})^+$.*

A *restricted weak path* is a weak path π with active clocks $x_{i_1}, \dots, x_{i_r}$, such that $\mathsf{left}_\Delta^{(\pi)_1}(x_{i_{r+1}}) \geq 2$ and $\mathsf{right}_\Delta^{(\pi)_k}(x_{i_j}) \geq 2$ for all $1 \leq k \leq |\pi|$ and $1 \leq j \leq m$. This restriction allows us to concentrate on the leftmost part of the channel only. In fact, in a restricted weak path, the only guards that are satisfied but not exactly are in the form $x = \alpha$ or $x \leq \alpha$ (where $x \in \mathcal{C}$ and $\alpha \in \mathbb{N}$. This property yields the following lemma shich states that if we replace all delay transitions by shorter ones, then all guards are still satisfied.

Lemma 20. *Let π be a weak path of length p with $\mathit{right}_\Delta^{(\pi)_i}(x_{i_j}) \geq 2$ for all clock indices i_j and $1 \leq i \leq p$, with $t = \mathsf{trans}(\pi)$. Let t' be a transition trace such that $t'_i = t_i$ for all i such that $t_i \in \Sigma$, and $t'_i \leq t_i$ otherwise. Then there exists a weak path π' over transition trace t' which satisfies $[(\pi')_i] \subseteq [(\pi)_i]$ for all $1 \leq i \leq p$.*

An immediate application of the previous property is that we can render any block *trivial*, *i.e.* of size -1.

Lemma 21. *Let π be a weak path of length p with $\mathit{right}_\Delta^{(\pi)_i}(x_{i_j}) \geq 2$ for all clock indices i_j and $1 \leq i \leq p$. Let us write $(\pi)_p = (d, \boldsymbol{n})$. Then, for all $j \in \{i_1, \dots, i_{r-1}\}$, there exists a weak restricted path π' over $\mathsf{trace}(\pi)$ such that $[(\pi')_i] \subseteq [(\pi)_i]$ for all $1 \leq i \leq |\pi|$, and if we write $(\pi')_p = (d', \boldsymbol{n'})$, then $n'_j = -1$, $n'_m = n_m + n_j^+$ et $n'_k = n_k$ for all $k \notin \{j, m\}$.*

The idea is that, given a weak path $\pi = \pi_{i_r} \dots \pi_{i_1}$, one can replace π_{i_j} by π'_{i_j}, obtained by removing all delay transitions. All guards in the resulting path are still satisfied thanks to the hypothesis, by Lemma 20. Note that the previous lemma can be applied successively for several blocks since the resulting path satisfies the hypotheses. The case of the block i_r is different since one can remove the delay transitions inside π_{i_r}, which will reduce the size of the block, but one cannot always reduce it to size -1 (see Lemma 19).

We now show that one can repeat a weak cycle by removing all delay transitions, and still reach the same target state.

Lemma 22. *Let π be a weak cycle of length p with $\mathit{right}_\Delta^{(\pi)_k}(x_{i_j}) \geq 2$ for all $1 \leq k \leq p$ and clock x_{i_j}, and let $t = \mathsf{trans}(\pi)$. Let t' be obtained from t by replacing all delay transitions by trivial ones, so that $\mathsf{time}(t') = 0$. Then, there exists a weak cycle π' over $t' \cdot t$ such that $(\pi')_1 = (\pi)_1$ and $(\pi')_{2p} = (\pi)_p$.*

Proof. By Lemma 21, and the remark that follows, t' can be realized from state $(\pi)_1$. Let $(\pi')_p$ denote the state reached by this path. All inactive clocks have the same position in $(\pi)_1$ and in $(\pi')_p$, while the active clocks are closer to the left end of the channel in $(\pi')_p$ than in $(\pi)_1$ (since they have been reset). Then one can realize t from here since $[(\pi')_p] \subseteq [(\pi)_1]$ and all guards that are satisfied non-exactly are of the form $x = \alpha$ or $x \leq \alpha$. Lemma 19 ensures that the blocks of $(\pi')_{2p}$ have the desired sizes. □

Now a pumping lemma for restricted weak cycles.

Lemma 23. *Let π be a restricted weak cycle with $\mathsf{last}(\pi) = (d, \boldsymbol{n})$. Then, there exists a weak cycle π' with $\mathsf{trace}(\pi') = \mathsf{trace}(\pi)^3$ such that if we write $\mathsf{last}(\pi') = (d', \boldsymbol{n'})$, we have $n'_r = n_r + 1$, $n'_m = n_m - 1$ and $n'_j = n_j$ for $j \neq r, m$.*

Proof. Let $t = \mathsf{trans}(\pi)$, and t' be the transition trace obtained by removing the delay transitions. By Lemma 22 there is a weak path over $t' \cdot t' \cdot t$ form $\mathsf{first}(\pi)$ to $\mathsf{last}(\pi)$. But one can insert a delay transition with a Δ-duration of 1 before the second repetition of t'. Then the remaining transition trace $t' \cdot t$ is still realizable since $\mathsf{left}_\Delta^{\mathsf{first}(\pi)}(x_{i_r}) \geq 2$ (in fact, the insertion of an additional Δ symbol preserves the satisfaction of the guards). □

E.2 Pumping Lemma: Bounded case (Bis)

We will define a new representation of the paths of $\mathcal{C}_\mathcal{A}(\Delta^N)$ where weak cycles are seen as atomic transitions. Formally, any path of $\mathcal{C}_\mathcal{A}(\Delta^N)$ is now a sequence $\pi = (q_i, \sigma_i)_{i \geq 1}$ where each q_i is a state of $\mathcal{C}_\mathcal{A}(\Delta^N)$, and $\sigma_i = \tau$ if i is odd, and σ_i is either an action transition or a trace of a weak cycle if i is even. Without loss of generality, we assume that all weak cycles start and end in a state where at least a clock has an integer value. In this representation of paths, we assume that weak cycles are chosen as maximal. Formally, if $\pi = (q_i, \sigma_i)_{i \geq 1}$ is a path, for any weak cycle σ_i, from q_i to q_{i+1}, for all even $j \geq i + 2$, $\sigma_i \cdot \sigma_{i+1} \cdot \ldots \cdot \sigma_j$ is not a weak cycle, and for all even $1 \leq j \leq i - 2$, $\sigma_j \cdot \sigma_{j+1} \cdot \ldots \cdot \sigma_i$ is not a weak cycle. Moreover, there is no weak cycle along action or delay transitions of π (all such weak cycles are necessarily marked explicitly as weak cycles).

Duration and Δ-duration are defined summing the (Δ-)durations of all delay transitions, including the ones inside weak cycles. For a transition trace t of a path as defined above, we define $\widetilde{t} = \widetilde{t'}$ where t' is the transition trace of π where weak cycles are "unfolded".

We adapt the notion of *staying united along a path* for the new representation of paths. The preorder $\prec$ is the same between states separated by a delay or action transition. Consider a weak cycle π from state q to q' with $q' = (d', \boldsymbol{n'})$. Let $i_1, \ldots, i_m$ denote the clock indices present in q' where $A = \{i_1, \ldots, i_r\}$ are the active clocks along this cycle. We extend $\prec$ by

$$\begin{array}{ll} (q, i_j) \prec (q', i_j) & \text{if } i_j \neq i_m \text{ and } x_{i_j} \notin A, \\ (q, i_j) \prec (q', i_r) & \text{if } i_j \in A, \\ (q, i_m) \prec (q', i_m) & \text{if } n'_m \geq 2, \\ (q, i_m) \prec (q', i_j) & \text{if } i_j \in \mathsf{Pumpable}(\pi). \end{array}$$

From this point on, a block that stays united along a path is defined according to the new definition of the paths and of $\prec$.

The following lemma justifies the calling "pumpable", by showing that one can pump Δ's into pumpable blocks.

Lemma 24. *Let π be a weak cycle with $\mathsf{last}(\pi) = (d, \boldsymbol{n})$. If $n_m \geq 1$, then for all $j \in \mathsf{Pumpable}(\pi)$, there exista a weak cycle π' with either $\mathsf{trace}(\pi') = \mathsf{trace}(\pi)$ or*

$\mathsf{trace}(\pi') = w_1 w_2^3 w_3$ *for some factorization* $\mathsf{trace}(\pi) = w_1 w_2 w_3$*, and such that if we write* $\mathsf{last}(\pi') = (d', \boldsymbol{n'})$*, then* $n'_m = n_m - 1$*,* $n'_j = n_j + 1$ *and* $n'_i = n_i$ *for all* $i \neq j, m$*.*

Proof. Consider $\pi = \pi_{i_r} \pi_{i_{r-1}} \dots \pi_{i_1}$ as in (2) and let t_{i_j} denote the transition trace of π_{i_j}.

If t_{i_j} contains a transition with a Δ-duration greater than 1, then we can extend this delay so that one more Δ is read and rewritten to the channel. Since block 0 is already medium or large right after this delay, extending it will preserve the satisfaction of all guards, and one can continue the rest of the transitions exactly the same as in π.

Otherwise, all delay transitions in t_{i_j} has Δ-duration at most 1. Since we have $\mathsf{time}_\Delta(t_{i_j}) \geq K_0$, there are at least K_0 delay transitions of Δ-duration exactly 1 in t_{i_j} separated by action transitions. By the choice of K_0, π_{i_j} contains a sub-path ρ_{i_j} over some transition trace t'_{i_j} between two states q and q' such that $[q'] \subseteq [q]$ and for all $x \in \mathcal{C}$, $\mathsf{left}^q_\Delta(x) \geq 2$ iff $\mathsf{left}^{q'}_\Delta(x) \geq 2$. Moreover, we can choose $t_{i_j} = u_1 t'_{i_j} u_2$ so that $\mathsf{time}_\Delta(u_1), \mathsf{time}_\Delta(t'_{i_j}), \mathsf{time}_\Delta(u_2) \geq 2$. (In fact, $K_0 \geq 2|\mathcal{L}||\mathcal{C}|!|\{-1, 0, 1, 2\}|^{\mathcal{C}} + 4$). Our choice ensures that ρ_{i_j} is a restricted weak cycle. If no clock is active in ρ_{i_j}, then one can add an additional delay transition of Δ-duration 1 since there are at least two Δ's between the left end of the channel and any clock symbol, at some point during this cycle. Otherwise, we apply Lemma 23, which yields a new weak cycle ρ''_{i_j} over transition trace t''_{i_j}. Then $u_1 t''_{i_j} u_2$ is a valid sequence of transitions from $\mathsf{first}(\pi_{i_j})$ since $\mathsf{last}(\rho''_{i_j})$ only differs from $\mathsf{last}(\rho_{i_j})$ by the size of its blocks j and m, but the former is already medium or large in $\mathsf{last}(\rho_{i_j})$ and is no more active in the rest of the path, and the size of the latter is decremented by one, which does not affect the satisfaction of the guards. For the same reasons, $\pi_{i_{j-1}} \dots \pi_{i_1}$ can now be adapted starting from the state obtained by applying $u_1 t''_{i_j} u_2$. Lemma 19 yields the sizes of the blocks in the last state as desired. □

In the construction of the previous lemma, for all pumpable blocks i_j of a weak cycle π, we showed that either π_{i_j} contains a delay transition with Δ-duration at least 2, or it can be factorized as $\pi_{i_j} = u_1 u_2 u_3$ where u_2 is a restricted weak cycle, in which case the trace of u_2 can be repeated while Δ's are pumped from block i_m to block i_j. Factor u_2 is called a *repeatable factor*. This notion will be used later in the proofs.

The following lemma is the pumping lemma for blocks that stay united, which is the adaptation of Lemma 12.

Lemma 25. *Let* π *be a path of* $\mathcal{C}_\mathcal{A}(\Delta^N)$ *such that* $((\pi)_1, j_1) \prec \dots \prec ((\pi)_p, j_p)$ *for some clock indices* $j_1, \dots, j_p$*. Then, for all* $L \geq 1$*, there exists a path* π' *of* $\mathcal{C}_\mathcal{A}(\Delta^{N+L})$ *such that* $(\pi')_i = (\pi)_i[x_{j_i} \leftarrow x_{j_i} \Delta^L]$*, and* $\mathsf{trace}(\pi') \in \widetilde{\mathsf{trace}(\pi)}$*.*

Proof. This follows from Lemmas 12 and 24. □

We now give a bound on the length of paths where a block does not stay united. This is an adaptation of Lemma 13.

Lemma 26. *Let π be a path of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ with $|\pi| = p$. Then all blocks of $\mathsf{first}(\pi)$ of size at least $p|\mathcal{C}|K_0$ stay united along π.*

Proof. The proof is similar to that of Lemma 13. Consider graph $G_{\prec}(\pi, l_1)$ constructed similarly over action and delay transitions. For any weak cycle from $(\pi)_i$ to $(\pi)_{i+1}$ a node $((\pi)_i, l)$ has a successor $((\pi)_{i+1}, l')$ if, and only if $((\pi)_i, l) \prec ((\pi)_{i+1}, l')$. We define the labelling $n(\cdot)$ as before, and update it similarly over action and delay transitions. Consider a weak cycle with active clock indices $A = \{i_1, \ldots, i_r\}$. If $l = l' \in A$, then this block is unchanged during the weak cycle so we let $n((\pi)_i, l) = n((\pi)_{i+1}, l')$. If $l' = i_r$, we let $n((\pi)_{i+1}, i_r) = \sum_k n((\pi)_i, k)$, where k ranges over the blocks in A such that $((\pi)_i, k) \prec ((\pi)_{i+1}, l')$ (in fact, all these blocks are merged with block i_r at the end of the weak cycle). If $l = l' = i_m$, then we choose a lower bound $n((\pi)_{i+1}, i_m) \leq n((\pi)_i, i_m)$ (the label is decremented by the number of Δ's that are removed from bock i_m during the weak cycle). Last, if $l = i_m$ and $l' = i_j$ for some $i_j \in A$, then $n((\pi)_{i+1}, i_j)$ is chosen as a lower bound on the size of block i_j in $(\pi)_{i+1}$. A maximal lower bound can be choosed for all pumpable blocks, in such a way that the sum of the labels of successor nodes of $((\pi)_i, i_m)$ do not exceed $n((\pi)_{i+1}, i_m)$.

In this construction, the sum of the labels from level i to level $i+1$ is decremented by at most $(K_0 - 1)(|\mathcal{C}| - 1)$. In fact, block i_m may yield at most $K_0 - 1$ Δ symbols to each non-pumpable block by definition, and the sum of the labels are preserved in other cases. This yields the result. □

E.3 Making Exact Paths (Bis)

In this subsection, we show how arbitrary paths of bounded length given with the new representation can be transformed into *quasi-exact* paths (defined below).

We redefine N_0 as $N_0 = |\mathcal{C}|K_0^2(W+1)$. From now on, *medium blocks* are those of size $\{2, \ldots, N_0 - 1\}$, and *large blocks* have size at least N_0. For any path π of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$, if $(\pi)_i$ is a state of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$, we define $H(\pi, i, n)$ as before using the new N_0 and the new definition of staying united.

A path π is *quasi-exact* if all weak cycles are quasi-exact and all other transitions are exact. We prove, in two lemmas, that any weak cycle can be transformed into one that is quasi-exact, provided that there is at least one pumpable block, or that block m is not small at the end. The first lemma treats the case where block m is not small, and the other one the case where it is small.

Lemma 27. *Let $\pi = \pi_{i_r} \ldots \pi_{i_1}$ be a weak cycle written as in (2) such that $\mathit{right}_{\Delta}^{\mathsf{last}(\pi)}(x_{i_m}) \geq 2$. Let $A = \{i_1, \ldots, i_r\}$ denote the set of active clocks. Then, for any subset of clock indices R such that $A \setminus \mathit{Pumpable}(\pi) \subseteq R \subseteq A$, there exists a weak quasi-exact cycle π' with $\mathsf{trace}(\pi) = \mathsf{trace}(\pi')$ and $[\mathsf{last}(\pi')] \subseteq [\mathsf{last}(\pi)]$, which satisfies the following conditions. Let us write $\mathsf{last}(\pi) = (d, \boldsymbol{n})$ and $\mathsf{last}(\pi') = (d', \boldsymbol{n'})$. Then,*

- $n'_j = -1$ *for any* $i_j \in R \setminus \{i_r\}$,

– $n'_r = n_r$ *if* $i_r \notin R$, *and* $n'_r = n_r - \mathsf{time}_\Delta(\pi_{i_r})^+$ *otherwise,*
– $n'_m = n_m + \sum_{i_j \in R} n_j^+ + (\mathsf{time}_\Delta(\pi_{i_r})^+$ *if* $i_r \in R)$,
– $n'_i = n_i$ *for any other* i.

Proof. The hypothesis we made on π implies that $\mathsf{right}_\Delta^{(\pi)_i}(x_{i_j}) \geq 2$ for all i and any clock x_{i_j}. Then π' is given by Lemma 21 applied to blocks of R. □

The construction is slightly different if block m becomes small during the weak cycle, and the trace is not always preserved in this case:

Lemma 28. *Let* $\pi = \pi_{i_r} \dots \pi_{i_1}$ *be a weak cycle written as in* (2). *Let* $A = \{i_1, \dots, i_r\}$ *denote the set of active clocks and assume that* $\mathsf{Pumpable}(\pi) \neq \emptyset$. *Then, for any subset of clock indices* R *such that* $A \setminus \mathsf{Pumpable}(\pi) \subseteq R \subsetneq A$, *there exists a weak quasi-exact cycle* π' *with* $\mathsf{trace}(\pi') \in \widetilde{\mathsf{trace}(\pi)}$ *and* $[\mathsf{last}(\pi')] \subseteq [\mathsf{last}(\pi)]$, *which satisfies the following conditions. Let us write* $\mathsf{last}(\pi) = (d, \boldsymbol{n})$ *and* $\mathsf{last}(\pi') = (d', \boldsymbol{n'})$. *Then,*

– $n'_j = -1$ *for any* $i_j \in R \setminus \{i_r\}$,
– $n'_r = n_r$ *if* $i_r \notin R$, *and* $n'_r = n_r - \mathsf{time}_\Delta(\pi_{i_r})^+$ *otherwise,*
– $n'_m = -1$,
– *there exists* $j_0 \in \mathsf{Pumpable}(\pi) \setminus R$, *such that* $n'_{j_0} = n_{j_0} + \sum_{i_j \in R} n_j^+ + (\mathsf{time}_\Delta(\pi_{i_r})^+$ *if* $i_r \in R)$,
– $n'_i = n_i$ *for any other* i.

Proof. Consider the factorization $\pi = \pi_{i_r} \dots \pi_{i_1}$. Let $1 \leq j_0 \leq r$ be minimal such that $i_{j_0} \in \mathsf{Pumpable}(\pi) \setminus R$. Since we have $\mathsf{time}_\Delta(\pi_{i_{j_0}}) \geq 2$ (this is a pumpable block), along the path $\pi_{i_r}\pi_{i_{r-1}} \dots \pi_{i_{j_0}}$, there is at least two Δ's between any clock and the right end of the channel. So we can apply Lemma 21, which yields paths π'_{i_k} for each $i_k \in R$ for $j_0 < k \leq r$, such that $\mathsf{time}(\pi'_{i_k}) = 0$. We then transform π_{j_0} using Lemma 24, which yields a path π'_{j_0} such that $\mathsf{time}_\Delta(\pi'_{j_0}) = \mathsf{time}_\Delta(\pi_{j_0}) + n_m + \sum_{i_k \in R \setminus \{i_r\}} n_k^+ + \big(\mathsf{time}_\Delta(\pi_{i_r})^+$ if $i_r \in R\big)$. Notice that block m has size -1 in $\mathsf{last}(\pi'_{j_0})$. Now, we define $\pi'_{i_{j_1-1}} \dots \pi'_{i_1}$ from $\pi_{i_{j_1-1}} \dots \pi_{i_1}$ by removing all delay transitions. Note that this path does not satisfy the hypothesis of Lemma 21, however, block m is smaller in $\mathsf{last}(\pi'_{j_0})$ than in $\mathsf{last}(\pi_{j_0})$ and only those blocks that are medium or large in $\mathsf{last}(\pi_{j_0})$ are increased in $\mathsf{last}(\pi'_{j_0})$, so all guards are still satisfied. We conclude with Lemma 19. □

The following lemma shows that any path of $\mathcal{C}_\mathcal{A}(\Delta^N)$ of bounded length can be transformed into a quasi-exact path. This is an adaptation of Lemma 14. Notice that by definition of N_0, along any path of length less than $K_0(W+1)$, all large blocks stay united, by Lemma 26.

Lemma 29. *Let* π *be a path of* $\mathcal{C}_\mathcal{A}(\Delta^N)$ *with* $|\pi| \leq K_0(W+1)$, *where* $N \geq |\mathcal{C}|N_0$. *Then, there exists a quasi-exact path* π' *such that* $\mathsf{first}(\pi') = H(\pi, 1, K_0(W+1))$, *and* $d_\Delta((\pi)_i, (\pi')_i) \leq (|\mathcal{C}| + 2)N_0$.

Proof. The proof is similar to the proof of Lemma 14. We construct $(\pi')_i$ by induction on $i \geq 1$, such that

- $[(\pi')_i] \subseteq [(\pi)_i]$,
- any block that does not stay united along $\pi_{i\ldots|\pi|}$ has size -1 in $(\pi')_i$,
- $d_\Delta\big((\pi)_i, (\pi')_i\big) \leq (|\mathcal{C}| + 2)N_0$.

We will define, for each $i \geq 1$, an excess vector $\boldsymbol{k}$ such that for any block j, $n'_i \leq n_i + k^i_j$, where $(\pi)_i = (d, \boldsymbol{k})$ and $(\pi')_i = (d', \boldsymbol{n'})$.

The first state is $(\pi')_1 = H(\pi, 1, n)$. The construction is the same as in Lemma 14 over delay and action transitions. Suppose that there is a weak cycle from $(\pi)_i$ to $(\pi)_{i+1}$.

If block m stays united along $\pi_{i+1\ldots|\pi|}$, then we apply Lemma 27 choosing R as the set of non-pumpable blocks and those pumpable blocks that do not stay united in the rest of the path. This gives a weak cycle that ends in some state $(\pi')_{i+1}$ such that $[(\pi')_{i+1}] \subseteq [(\pi)_{i+1}]$. Let us show that blocks that do not stay united along $\pi_{i+1\ldots|\pi|}$ have size -1 in $(\pi')_{i+1}$. This is the case for the blocks of the inactive clocks, since their sizes are unchanged along a weak cycle (except for block m, but it stays united by hypothesis) and by construction, any block among $\{i_1, \ldots, i_{r-1}\}$ that do not stay united has size -1 in $(\pi')_{i+1}$. Consider now block r. If $i_r \in R$, then block r has size $\mathsf{left}_\Delta^{(\pi')_i}(x_{i_{r+1}})$ in $(\pi')_{i+1}$ by Lemma 27. By induction hypothesis, this is positive if, and only if one of the blocks $i_1, \ldots, i_r$ stays united along $\pi_{i\ldots|\pi|}$. Moreover, by definition, block r stays united along $\pi_{i+1\ldots|\pi|}$ if, and only if one of these blocks do. So block r has indeed size -1 in $(\pi)_{i+1}$ when it does not stay united.

We now bound the Δ-distance between the states of π and π'. We define an excess vector $\boldsymbol{k^i}$ for all $i \geq 1$. We have $\|\boldsymbol{k^1}\| \leq |\mathcal{C}|N_0$. We define $\boldsymbol{k^i}$ over delay and action transitions as in Lemma 14. For a weak cycle from $(\pi)_i$ to $(\pi)_{i+1}$, vector the norm of $\boldsymbol{k^{i+1}}$ is incremented by at most $K_0 - 1$ for each non-pumpable block. It may be incremented by any $p \geq 1$ for pumpable blocks that do not stay united in the rest of the path, but then by Lemma 26, the excess vector is unchanged during at least $\lfloor \frac{p}{|\mathcal{C}|K_0} \rfloor$ transitions. In fact this is the number of transitions (either delay, action or weak cycles) necessary to split the blocks that do not stay united, during which such a block is the rightmost one (block i_m) and it has size -1 in $(\pi')_i$ by construction. So, the transition is carried out from $(\pi')_i$ with null length delays (for instance, a weak cycle will be followed from $(\pi')_i$ by removing all delay transitions). Hence the excess of the blocks in π' does not increase during such transitions. Thus, $\|\boldsymbol{k}\|_1$ increases at most by $|\mathcal{C}|K_0$ for each transition, and it can increaase, in total, by $p = |\mathcal{C}|K_0^2(W+1)$ because of the pumpable blocks (in fact, $\frac{p}{|\mathcal{C}|K_0} = K_0(W+1)$, which is the length of the run). We get that $\|\boldsymbol{k^n}\|_1 \leq |\mathcal{C}|N_0 + 2(W+1)|\mathcal{C}|K_0^2 \leq (|\mathcal{C}| + 2)N_0$. □

Last, we adapt Lemma 15 to quasi-exact paths.

Lemma 30. *Let $\mathcal{A}$ be any timed automaton, $N \geq 1$ and π a quasi-exact path of $\mathcal{C}_\mathcal{A}(\Delta^N)$. Then, for any $N' \geq N$, and any state q of $\mathcal{C}_\mathcal{A}(\Delta^{N'})$ with $[q] \subseteq$*

$[\mathsf{first}(\pi)]$, *there exists a quasi-exact path* π' *with* $\mathsf{trace}(\pi') \in \widetilde{\mathsf{trace}(\pi)}$, *with* $\mathsf{first}(\pi') = q$ *and* $[(\pi')_i] \subseteq [(\pi)_i]$ *for all* $1 \leq i \leq n$. *The same property holds backwards: for any* $q \in [\mathsf{last}(\pi)]$, *there exists a quasi-exact path* π' *with* $\mathsf{trace}(\pi') \in \widetilde{\mathsf{trace}(\pi)}$ *in* $\mathcal{C}_{\mathcal{A}}(\Delta^{N'})$ *with* $\mathsf{last}(\pi') = q$ *and* $[(\pi')_i] \subseteq [(\pi)_i]$ *for* $1 \leq i \leq |\pi|$.

Proof. This follows from Lemma 15 for delay and action transitions, and from Lemma 24 for weak cycles along π. □

A quasi-exact path π is a *quasi-exact cycle* if $\mathsf{reg}(\mathsf{first}(\pi)) = \mathsf{reg}(\mathsf{last}(\pi))$. We show that a long enough quasi-exact path contains a quasi-exact cycle.

Lemma 31. *Let* π *be a quasi-exact path with* $|\pi| \geq (2|\mathcal{L}||\mathcal{C}|!2^{|\mathcal{C}|} + 1)(W + 1)$. *Then* π *contains a quasi-exact cycle.*

Proof. Observe that any cycle along which the integer part of some clock changes is a progress cycle. In fact such a clock is necessarily reset later, and there must be delay transitions after this reset of duration at least one time unit. Since these delays also increase the value of any other clock, all clocks must be reset in order to go back to the intial region.

Let $n = 2|\mathcal{L}||\mathcal{C}|!2^{|\mathcal{C}|} + 1$. We argue that along any path π, for all $1 \leq i \leq |\pi|$, either $\pi_{i\ldots i+n}$ contains a progress cycles or there is some clock whose integer part changes at least once. Consider any $i \geq 1$ and assume that $\pi_{i\ldots i+n}$ does not contain a progress cycle and that the integer parts do not change in $\pi_{i\ldots i+n}$. The number of regions visited by this path between action transitions is at most $|\mathcal{L}||\mathcal{C}|!2^{|\mathcal{C}|}$ different regions ($|\mathcal{C}|!$ for the ordering of the clocks, and $2^{|\mathcal{C}|}$ to choose for each block a size, which is either -1 or larger). In particular, for any weak cycle γ along this path, $\mathsf{reg}(\gamma)$ is not visited before, and is not visited again, since this would contradict the maximality of the weak cycles we assumed (beginning of Section E.2). But then, there can be at most $2|\mathcal{L}||\mathcal{C}|!2^{|\mathcal{C}|}$ states in this path, and n is larger, which is a contradiction.

Hence, either there is some $i \geq 1$ such that $\pi_{i\ldots i+n}$ contains a progress cycle, or the value of some clock changes in all $\pi_{i\ldots i+n}$. If there is no progress cycles in $\pi_{i\ldots i+n}$ for all i, then consider the paths $\pi_{1\ldots n}, \pi_{n\ldots 2n}, \ldots, \pi_{nW\ldots n(W+1)}$. There exists $i < j$ such that $\mathsf{reg}((\pi)_{in}) = \mathsf{reg}((\pi)_{jn})$, and $\pi_{in\ldots jn}$ is a progress cycle by the remark above. □

E.4 Proof of the Pumping Lemma in General Case

We now have everything necessary to prove the pumping lemma in presence of weak cycles, except for the following "cycling lemma", whose proof is postponed to next section. This is the generalization of Lemma 16 that we stated in the core of the paper.

Lemma 32. *Let* $\mathcal{A}$ *be any timed automaton,* $N \geq 1$ *and* π *a quasi-exact progress cycle in* $\mathcal{C}_{\mathcal{A}}(\Delta^N)$. *Then, for all* $q \in \mathsf{last}(\pi)$, *there exists a path* π' *in* $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ *with* $\mathsf{first}(\pi') = \mathsf{first}(\pi)$ *and* $[\mathsf{last}(\pi')] \subseteq [q]$, *and* $\mathsf{trace}(\pi') = \widetilde{\mathsf{trace}(\pi)}$.

The proof of the pumping lemma follows the same steps as in the progress-cycle case, but we use the new constants, the new definitions and the new lemmas we developed in previous subsections.

Lemma 11 (Pumping Lemma). *Consider a timed automaton $\mathcal{A}$, and let W denote its number of regions. Let $K_0 = 2|\mathcal{C}|!\cdot|\mathcal{L}|\cdot 4^{|\mathcal{C}|}+4$, and $N_1 \geq 8|\mathcal{C}|^2 \cdot (W+1)$ if $\mathcal{A}$ has only progress cycles, and $N_1 \geq 5|\mathcal{C}|^3 \cdot K_0^2 \cdot (W+1)$ otherwise. Then, for any path π of $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1})$, for any $L \geq 0$, there exists a path π' of $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1+L})$, such that the same set of locations appear infinitely often in π and π'.*

Proof. We prove the result for $1 \leq L \leq |\mathcal{C}|N_0 - 2$. For larger L, one can repeat this construction. Let $N \geq 5|\mathcal{C}|^2 N_0$, and consider an infinite path π of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ over some trace $(l_i)_{i\geq 1}$, where $(\pi)_1$ is the initial state of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$.

Let $n = K_0(W+1)$. Let $G^L(\pi, i, n)$ denote the state of $\mathcal{C}_{\mathcal{A}}(\Delta^{N+L})$ obtained from $(\pi)_i$ by inserting Δ^L in the block with minimal index that stay united along $\pi_{i\dots i+n}$ (such a block always exists by the choice of N, and by Lemma 26; remember that $N_0 = |\mathcal{C}|K_0^2(W+1)$). We also define $H^L(\pi, i, n)$ by inserting Δ^L to the same block in $H(\pi, i, n)$. We have, by construction, $d_\Delta\big(H^L(\pi, i, n), G^L(\pi, i, n)\big) \leq |\mathcal{C}|N_0$.

We now define a path π' of $\mathcal{C}_{\mathcal{A}}(\Delta^{N+L})$ over trace $\widetilde{\mathsf{trace}(\pi)}$. At each step $i \geq 1$, we will construct $\pi'_{\beta_i\dots\beta_{i+1}}$, where $(\beta_i)_{i\geq 1}$ is an increasing sequence. Our construction will satisfy $\mathsf{trace}(\pi'_{\beta_i\dots\beta_{i+1}}) \in \widetilde{\mathsf{trace}(\pi_{\alpha_i\dots\alpha_{i+1}})}$, and $[(\pi')_{\beta_i}] \subseteq [(\pi)_{\alpha_i}]$, for some possibly different increasing sequence $(\alpha_i)_{i\geq 1}$.

We define $(\pi')_1$ as the initial state of $\mathcal{C}_{\mathcal{A}}(\Delta^{N+L})$, which satisfies $[(\pi')_1] \subseteq [(\pi)_1]$. Suppose now that $\pi'_{1\dots\beta_i}$ has been constructed for some $\beta_i \geq 1$. By Lemma 25, there is a path g of $\mathcal{C}_{\mathcal{A}}(\Delta^{N+L})$ from $G(\pi, \alpha_i, n)$ over $\mathsf{trace}(\pi_{\alpha_i\dots\alpha_i+n})$, such that $(g)_j = (\pi)_{\alpha_i+j}[z_j \leftarrow z_j\Delta^L]$ for some clocks z_j. We then apply Lemma 29 to g to get an exact path h with $(h)_1 = H^L(\pi, \alpha_i, n)$ over the trace of g, with $d_\Delta\big((h)_j, (g)_j\big) \leq (|\mathcal{C}|+2)N_0$, and $[(h)_j] \subseteq [H^L(\pi, \alpha_i + j, n)$ for all $1 \leq j \leq n$. We have $K_0 \geq (2|\mathcal{L}||\mathcal{C}|!2^{|\mathcal{C}|}+1)$, so by Lemma 31, there exists $1 \leq l_0 < l_1 \leq n$ such that $\mathsf{reg}((h)_{l_0}), \mathsf{reg}((h)_{l_0+1}), \dots, \mathsf{reg}((h)_{l_1})$ is a quasi-exact progress cycle. We have $d_\Delta\big((h)_{l_1}, H^L(\pi, \alpha_i + l_1, \alpha_i + l_1 + n)\big) \leq (3|\mathcal{C}|+2)N_0 - 2$ by combining following inequalities.

$$
\begin{aligned}
&d_\Delta((h)_{l_1}, (g)_{l_1}) \leq (|\mathcal{C}|+2)N_0,\\
&d_\Delta\big((g)_{l_1}, G^L(\pi, \alpha_i + l_1, \alpha_i + l_1 + n)\big) \leq L \leq |\mathcal{C}|N_0 - 2,\\
&d_\Delta\big(G^L(\pi, \alpha_i + l_1, \alpha_i + l_1 + n), H^L(\pi, \alpha_i + l_1, \alpha_i + l_1 + n)\big) \leq |\mathcal{C}|N_0.
\end{aligned}
$$

By Lemma 30, $\mathcal{C}_{\mathcal{A}}(\Delta^{N+L})$ has a path over $\mathsf{trace}(h)$ from $(\pi')_{\beta_i}$ to some state q with $[q] \subseteq [(h)_{l_1}]$. By Lemma 8, we get that $[(h)_{l_1}] \cap [H^L(\pi, \alpha_i + l_1, \alpha_i + l_1 + n)] \neq \emptyset$. Lemma 32 provides a path of $\mathcal{C}_{\mathcal{A}}(\Delta^{N+L})$ from q to some state q' such that $[q'] \subseteq [(h)_{l_1}] \cap [H^L(\pi, \alpha_i + l_1, \alpha_i + l_1 + n)]$, over a trace in $\mathsf{trace}(h_{l_0\dots l_1})^+$. □

F Cycling Lemma

In this section, we prove Lemma 32. Let us fix a quasi-exact progress cycle π. In the rest of the proof, for each weak cycle of π over a transition trace γ, where

$\gamma = \gamma_{i_r} \dots \gamma_{i_1}$ is the factorization as in (2), we fix one repeatable factor, say the leftmost one, in each γ_{i_j}. Let us write $\gamma_{i_j} = \gamma'_{i_j}\gamma''_{i_j}\gamma'''_{i_j}$ for each weak cycle γ and i_j, where γ''_{i_j} is this repeatable factor. We define

$$\mathsf{rep}(\gamma) = \gamma'_{i_1}(\gamma''_{i_1})^+\gamma'''_{i_1} \dots \gamma'_{i_r}(\gamma''_{i_r})^+\gamma'''_{i_r}.$$

We extend this to $\mathsf{rep}(\mathsf{trans}(\pi))$, where the transition trace γ of each weak cycle is replaced with $\mathsf{rep}(\gamma)$. We also define $\mathsf{rep}(\mathsf{trace}(\pi))$ similarly (by repeating the trace of the corresponding factors). In the proof of Lemma 32, we will construct the paths π' with $\mathsf{trace}(\pi') \in \mathsf{rep}(\mathsf{trace}(\pi))$. Clearly, $\mathsf{rep}(\mathsf{trace}(\pi)) \subseteq \widetilde{\mathsf{trace}(\pi)}$.

Lemma 32 is similar to Lemma 29 of [7] but we generalize it and the intermediate results to general timed automata and to channel machines. The proof follows the ideas of [7]. An important notion in this proof is that of *limit points* for a given path. We define this notion both for $[\![\mathcal{A}]\!]$ and $\mathcal{C}_{\mathcal{A}}(\Delta^N)$.

Definition 33. *The set of limit points of* $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ *for path* π*, denoted by* $L_\pi(\mathcal{C}_{\mathcal{A}}(\Delta^N))$ *contains any state* q*, if, and only if there is a quasi-exact path* π' *with* $\mathsf{first}(\pi') = \mathsf{last}(\pi') = q$*, and* $[(\pi')_i] \subseteq [(\pi)_i]$ *for all* $1 \leq i \leq |\pi|$*, and* $\mathsf{trans}(\pi') \in \mathsf{rep}(\mathsf{trans}(\pi))^+$*. Path* π' *is called a* witness path *for* q*.*

In other terms, this is the set of states of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ for which there exists a path which at each step stays inside the closed region defined by the states of π^k, for some $k \geq 1$, and follow the transition trace of π except that it can cycle in its repeatable factors.

We will define a similar set of limit points for $[\![\mathcal{A}_{\frac{1}{N}}]\!]$ but we first need to extend the definition of *quasi-exact progress cycles* to $\mathcal{A}$.

A *weak cycle* of $[\![\mathcal{A}_{\frac{1}{N}}]\!]$ is a run ρ such that $\mathsf{clos}(\mathsf{reg}(\mathsf{last}(\rho))) \subseteq \mathsf{clos}(\mathsf{reg}(\mathsf{first}(\rho)))$ along which not all clocks are reset. In this section, we redefine the runs of $[\![\mathcal{A}_{\frac{1}{N}}]\!]$ as sequences of states which are related either by action or delay transitions, or by weak cycles. We say that a run ρ of $[\![\mathcal{A}_{\frac{1}{N}}]\!]$ *follows* a path π of $\mathcal{C}_{\mathcal{A}}(\Delta^{2N})$, if $(\rho)_i \in \mathsf{clos}(\mathsf{concrete}((\pi)_i))$ for all $1 \leq i \leq |\pi|$, and the transitions satisfy the following conditions.

- If $\mathsf{trans}(\pi)_i \in \Sigma$, then $\mathsf{trans}(\rho)_i = \mathsf{trans}(\pi)_i$,
- If $\mathsf{trans}(\pi)_i = \tau$, then $\mathsf{trans}(\rho)_i \in \mathbb{R}_{\geq 0}$,
- If $\mathsf{trans}(\pi)_i$ is the trace of a weak cycle, then $\mathsf{trans}(\rho)_i$ is a weak cycle of $[\![\mathcal{A}_{\frac{1}{N}}]\!]$ that follows the weak cycle of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ defined by $\mathsf{trans}(\pi)_i$.

Using Lemma 6, one can apply the construction of Lemma 22 and iterate the repeatable factors in weak cycles of $[\![\mathcal{A}_{\frac{1}{N}}]\!]$ which follow weak cycles of $\mathcal{C}_{\mathcal{A}}(\Delta^{2N})$. Note also that when one iterates these repeatable factors with zero-length delays, this transformation preserves the last state of the run (this is also the case for $\mathcal{C}_{\mathcal{A}}(\Delta^N)$, in Lemma 22). A weak cycle of $[\![\mathcal{A}_{\frac{1}{N}}]\!]$ is said to be *quasi-exact* if it follows a weak quasi-exact cycle of $\mathcal{C}_{\mathcal{A}}(\Delta^{2N})$. For a weak cycle ρ that follows a weak cycle $\pi = \pi_{i_r} \dots \pi_{i_1}$ of $\mathcal{C}_{\mathcal{A}}(\Delta^{2N})$ written as in (2), we will also write $\rho = \rho_{i_r} \dots \rho_{i_1}$, where each ρ_{i_j} follows π_{i_j}. Observe that in quasi-exact weak cycles, we have $\mathsf{time}(\rho_{i_j}) = 0$ for all non-pumpable blocks. A *quasi-exact progress cycle*

of $[\![\mathcal{A}_{\frac{1}{N}}]\!]$ is a run that follows a quasi-exact progress cycle of $\mathcal{C}_{\mathcal{A}}(\Delta^{2N})$. Notice that all transitions are exact in quasi-exact progress cycles, except for those inside weak cycles.

Definition 34. *The limit points of $[\![\mathcal{A}_{\frac{1}{N}}]\!]$ for quasi-exact progress cycle π of $\mathcal{C}_{\mathcal{A}}(\Delta^{2N})$, denoted by $L_\pi([\![\mathcal{A}_{\frac{1}{N}}]\!])$, is the set of states (l, v) of $\mathcal{A}$ such that there exists a quasi-exact progress cycle ρ of $[\![\mathcal{A}_{\frac{1}{N}}]\!]$ such that $\mathsf{first}(\rho) = \mathsf{last}(\rho) = (l, v)$, which follows a witness path π' of a state of $L_\pi(\mathcal{C}_{\mathcal{A}}(\Delta^{2N}))$.*

It is easy to see that $L_\pi(\mathcal{C}_{\mathcal{A}}(\Delta^N))$ is forward and backward reachable from any state q such that $[q] \subseteq [\mathsf{first}(\pi)]$, using Lemma 30 (since $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ is a finite-state system, for fixed N). To prove Lemma 32, it then suffices to show that $L_\pi(\mathcal{C}_{\mathcal{A}}(\Delta^N))$ is connected by arcs following traces of $\mathsf{rep}(\mathsf{trace}(\pi))^+$. To obtain this, we first study $L_\pi([\![\mathcal{A}_{\frac{1}{N}}]\!])$ in Subsection F.1: we show that $L_\pi([\![\mathcal{A}_{\frac{1}{N}}]\!])$ is convex (Lemma 35), and connected by arcs (Lemma 36). This implies that any pair of points of $L_\pi([\![\mathcal{A}_{\frac{1}{N}}]\!])$ can be connected by a run following $\mathsf{rep}(\mathsf{trace}(\pi))^+$. We then prove some results on $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ in Subsection F.2 and finally prove the lemma in Subsection F.3.

F.1 Study of $L_\pi([\![\mathcal{A}_\Delta]\!])$

The following lemma generalizes Lemma 24 of [7] to quasi-exact paths.

Lemma 35. *$L_\pi([\![\mathcal{A}_{\frac{1}{N}}]\!])$ is convex.*

Proof. Let q and q' be two states of $L_\pi([\![\mathcal{A}_{\frac{1}{N}}]\!])$ and consider witness runs ρ and ρ'. These runs do not necessarily follow the same trace, but we have $\mathsf{trace}(\rho), \mathsf{trace}(\rho') \in \mathsf{rep}(\mathsf{trace}(\pi))$. Therefore, using the construction of Lemma 22, one can iterate the repeatable factors (denoted γ''_{i_j} above) with zero-length delays in every weak cycle in ρ and ρ'. Thus, let us assume that $\mathsf{trace}(\rho) = \mathsf{trace}(\rho')$. Then, for any λ, ρ'' defined by $(\rho'')_i = \lambda(\rho)_i + (1-\lambda)(\rho')_i$ for all $i \geq 1$, is a run of $[\![\mathcal{A}_{\frac{1}{N}}]\!]$ by the convexity of the guards. □

The following lemma generalizes Lemma 29 of [7] to quasi-exact progress cycles. The proof uses the same reasoning; the only difficulty is the treatment of weak quasi-exact cycles.

Lemma 36. *Let $(l, v) \in L_\pi([\![\mathcal{A}_{\frac{1}{N}}]\!])$ and ρ a witness run. Then, for all states $(l, v') \in \mathit{clos}(\mathit{reg}(q))$ and $d_\infty(v, v') \leq \frac{1}{2N}$, there exists a run ρ' such that $\mathsf{first}(\rho') = (l, v)$, $\mathsf{last}(\rho') = (l, v')$ and $\mathsf{trace}(\rho') \in \mathsf{rep}(\mathsf{trace}(\pi))^+$.*

Proof. The proof is adapted from Lemma 29 of [7]. Without loss of generality we assume that all clocks $x_1, \ldots, x_m$ are reset at least once in ρ (otherwise, some clocks are always above the maximal constant along ρ) and that they are ordered in such a way that $v(x_i) \geq v(x_{i+1})$ for all $1 \leq i \leq m-1$.

We let $\delta_i = v'(x_i) - v(x_i)$ for all $1 \leq i \leq m$, and $\delta_0 = \delta_{m+1} = 0$. Let $t = t_1 t_2 \ldots t_n$ denote the transition trace of ρ. There exists $\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_m$

such that t_{α_i} is the last action transition or weak cycle in t which resets clock x_i, and we define $\alpha_0 = 0$. If t_{α_i} is an action transition, then we let $t^-_{\alpha_i} = t_{\alpha_i}$ and $t^+_{\alpha_i} = \epsilon$ (the "empty" transition). If t_{α_i} is the transition trace of a weak cycle, then $t^-_{\alpha_i}$ is the prefix of t_{α_i} which stops at the last reset of the clock x_i, and we define $t^+_{\alpha_i}$ such that $t_{\alpha_i} = t^-_{\alpha_i} t^+_{\alpha_i}$. We have, for all clocks x_i,

$$v(x_i) = \mathsf{time}(t^+_{\alpha_i}) + \sum_{j=\alpha_i+1}^{n} \mathsf{time}(t_j). \tag{3}$$

Since $(l, v') \in \mathsf{clos}(\mathsf{reg}(v))$, all clock constraints satisfied in $\mathsf{clos}(\mathsf{reg}((l, v)))$ are also satisfied by v' and the fractional parts of the clock values have the same ordering in v and v', so we have $v'(x_i) \geq v'(x_{i+1})$ for $1 \leq i \leq m-1$. Then, we get,

$$\delta_{i+1} - \delta_i \leq v(x_i) - v(x_{i+1}) = \mathsf{time}(t^+_{\alpha_i}) + \sum_{j=\alpha_i+1}^{\alpha_{i+1}-1} \mathsf{time}(t_j) + \mathsf{time}(t^-_{\alpha_{i+1}}).$$

Let us define $f_i = t^+_{\alpha_{i-1}} t_{\alpha_i+1} t_{\alpha_i+2} \dots t^-_{\alpha_{i+1}}$ for all $0 \leq i \leq m$, (we let $t^-_{\alpha_0} = \epsilon$). We will transform each transition trace f_i into f'_i such that $\mathsf{time}(f'_i) = \mathsf{time}(f_i) + \delta_i - \delta_{i+1}$ and show that this new trace is realizable starting from (l, v). The idea is to extend the delays of f_i by a total duration of $\delta_i - \delta_{i+1}$. Note that we have $\mathsf{time}(f_i) + \delta_i - \delta_{i+1} \geq 0$, which follows from the above equation. This will define a new run ρ' whose last state is (l, v'), by (3). The construction is illustrated below.

$$\delta = \overbrace{[t_1 \dots t^-_{\alpha_1}]}^{f_0} \cdot \overbrace{[t^+_{\alpha_1} t_{\alpha_1+1} \dots t^-_{\alpha_2}]}^{f_1} \cdot \overbrace{[t^+_{\alpha_2} t_{\alpha_2+1} \dots t^-_{\alpha_3}]}^{f_2} \cdot \dots \cdot \overbrace{[t^+_{\alpha_3} t_{\alpha_m} \dots t_n]}^{f_m}.$$

$$\begin{array}{cccccc} & +\delta_0 = 0 & +(\delta_1 - \delta_2) & +(\delta_2 - \delta_3) & \dots & +\delta_m \\ \hline \delta' = & f'_0 & f'_1 & f'_2 & \dots & f'_m \end{array}$$

We will iteratively define runs ρ'_i of $[\![\mathcal{A}_{\frac{1}{N}}]\!]$ that follow f'_i's, so that $\rho' = \rho'_1 \dots \rho'_n$. For all $1 \leq i \leq n$, if $(l_k, v_k) = (rho_i)_k$ and $(l_k, v'_k) = (\rho'_i)_k$, these will satisfy, for any clock x, one of the following cases.

$$\begin{array}{l} v'_k(x) = v_k(x) + p_{k,x}(\delta_i - \delta_{i+1}) \\ v'_k(x) = v_k(x) + p_{k,x}(\delta_{i-1} - \delta_i) + q_{k,x}(\delta_i - \delta_{i+1}) \\ v'_k(x) = v_k(x) + p_{k,x}(\delta_j - \delta_{j+1}) + \delta_{j+1} - \delta_i + q_{k,x}(\delta_i - \delta_{i+1}), \end{array} \tag{4}$$

for some $p_{k,x}, q_{k,x} \in (0, 1)$, with the following meaning. The first case applies when the latest reset of x is inside f_i and in this case $p_{k,x}$ is the fraction of the quantity $\delta_i - \delta_{i+1}$ that is added to the delay transitions of f'_i upto state (l_k, v_k). The second case applies when the latest reset of x is in f_{i-1}, and the third case when it is in some f_j for $j < i-1$, where $p_{k,x}$ and $q_{k,x}$ are the fractions of the respective durations that are added since the latest reset of x. In all cases, we have $|v'_k(x) - v_k(x)| \leq \frac{1}{N}$ for all k and clock x, by triangular inequality, since $|\delta_i| \leq \frac{1}{2N}$ for all i. So when v satisfies a guard exactly, v' also satisfies it, but not necessarily exactly.

We let $f'_0 = f_0$, which defines $\rho'_0 = \rho_0$. Then $\mathsf{last}(\rho'_0)$ satisfies the first case of (4) with $p_{k,x} = 1$ (since $\delta_0 = 0$).

Now, assume that $f'_1 \dots f'_i$ have been defined, which leads to state $\mathsf{last}(\rho'_0 \dots \rho'_i)$ that satisfies (4). To define f'_{i+1}, we will add $\delta_{i+1} - \delta_{i+2}$ to the delays of f_{i+1}. If $\delta_{i+1} - \delta_{i+2} \geq 0$, we obtain f'_{i+1} from f_{i+1} by adding this to the first delay. The state reached in this manner satisfies (4). Let us show that the rest of f'_{i+1} defines a valid run from there; we will show that (4) is preserved along this run. Equation (4) is clearly preserved during delay transitions (these have the same duration in f_{i+1} and f'_{i+1}). Along an action transition, if some clock x is reset, then the first case of (4) is satisfied in the target state for $p_{k,x} = 0$, and other clock values are unchanged. The guards of the action transitions are satisfied as noted above. Weak cycles need a special treatment. Consider a weak cycle inside f_{i+1} from some state (l_k, v_k) to (l_{k+1}, v_{k+1}) following a transition trace γ. Because all guards may not be satisfied in a weak cycle, the transition trace γ may not be realizable from (l_k, v'_k). But we will slightly modify γ and adapt it. Let us write $\gamma = \gamma_{i_r} \dots \gamma_{i_1}$. Let j be the maximal index such that block i_j is pumpable, and $\gamma_{i_j} = \gamma'_{i_j} \gamma''_{i_j} \gamma'''_{i_j}$ such that γ''_{i_j} is the repeatable factor. Let $\hat{\gamma}'_{i_j}$ and $\hat{\gamma}''_{i_j}$ denote the transition traces obtained by removing all delay transitions from respective traces. Since γ is quasi-exact and j is maximal, we have $\mathsf{time}(\gamma_{i_r} \dots \gamma_{i_{j+1}}) = 0$. Again, since it is quasi-exact, inside $\gamma_{i_r} \dots \gamma_{i_{j+1}}$, all guards are satisfied exactly (since all small blocks have size -1 in the corresponding weak cycle of the channel machines). Therefore, $\gamma_{i_r} \dots \gamma_{i_{j+1}}$ is realizable from (l_k, v'_k). We can similarly extend this run applying $\hat{\gamma}'_{i_j} \hat{\gamma}''_{i_j}$. (This run is valid by the same arguments we used for the channel machines. In fact, ρ follows a weak quasi-exact cycle of $\mathcal{C}_{\mathcal{A}}(\Delta^{2N})$). We can then repeat the factor $\hat{\gamma}''_{i_j}$ any number of times and while inserting delays of duration at most $\frac{1}{N}$ between the repetitions, whose sum equals $\mathsf{time}(\gamma'_{i_j})$, and finally apply γ'''_{i_j}. The resulting run is valid (in fact we apply Lemma 24 to the weak cycle of $\mathcal{C}_{\mathcal{A}}(\Delta^{2N})$ which ρ follows). Moreover, the duration of γ_{i_j} is conserved. From here, we can apply $\gamma_{i_{j-1}} \dots \gamma_{i_1}$, again using the corresponding weak cycle in the channel machine. This defines the target state (l_{k+1}, v'_{k+1}). In v'_{k+1}, all inactive clocks satisfy (4) with the same j's as for v'_k, since these have not been reset during the weak cycle. For active clocks x, we have $v'_{k+1}(x) = v_{k+1}(x)$ by construction (delays have the same total duration between last resets).

When $\delta_{i+1} - \delta_{i+2} < 0$, the construction is similar. If the first delay is not long enough, then we distribute this quantity to other delays. Weak cycles are applied similarly as in the previous case. The delays inside weak cycles can be shortened if necessary (the only delays with positive durations are those in the pumpable blocks). □

F.2 Study of $L_\pi(\mathcal{C}_{\mathcal{A}}(\Delta^N))$

For any state q of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ define $\overline{q}$, the state of $\mathcal{C}_{\mathcal{A}}(\Delta^{2N})$, obtained from q by duplicating each Δ symbol.

Lemma 37. *For all $q \in L_\pi(\mathcal{C}_{\mathcal{A}}(\Delta^N))$, we have $\overline{q} \in L_\pi(\mathcal{C}_{\mathcal{A}}(\Delta^{2N}))$.*

Proof. Consider a witness path π' for q. We show that a witness path π'' can be constructed for $\overline{q}$ such that $(\pi'')_i = (\overline{\pi'})_i$ for all $1 \leq i \leq |\pi|$. The delay and action transitions can be adapted to $\mathcal{C}_{\mathcal{A}}(\Delta^{2N})$ using Lemma 15 since all action transitions are exact. Weak cycles are quasi-exact, so they can also be adapted using Lemma 24. In fact, non-pumpable blocks have size -1 at the end of the weak cycles, and we can pump into pumpable blocks. □

Lemma 38. *Let q be any state of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$, and (l, v) a state of $\mathcal{A}$ such that $(l, v) \in \mathsf{clos}(\mathsf{concrete}(\overline{q}))$. Then, $(l, v) \in \mathsf{concrete}(q)$.*

Proof. Since $(l, v) \in \mathsf{clos}(\mathsf{concrete}(\overline{q}))$, there exists Let $0 \leq \delta(1) \leq \ldots < \delta(2N) \leq 1$ with $\delta_{i+1} - \delta_i = \frac{1}{2N}$, which defines q. We let $\delta'(i) = \frac{\delta(2i-1)+\delta(2i)}{2}$. This satisfies $\delta'(1) < \ldots < \delta'(N)$ and $\delta'(i+1) - \delta'(i) = \frac{1}{N}$. Moreover, we have $\delta'(i) \neq v(x0)$ for all i and clock x. Since each block contains an even number of Δ's in $\overline{q}$, for all i, $\delta(2i-1)$ and $\delta(2i)$ correspond to Δ symbols in a same block. Hence $v \cup \delta'$ clock values whose fractional parts ordered according to q. □

F.3 Proof of the Cycling Lemma

Lemma 32. *Let $\mathcal{A}$ be any timed automaton, $N \geq 1$ and π a quasi-exact progress cycle in $\mathcal{C}_{\mathcal{A}}(\Delta^N)$. Then, for all $q \in \mathsf{last}(\pi)$, there exists a path π' in $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ with $\mathsf{first}(\pi') = \mathsf{first}(\pi)$ and $[\mathsf{last}(\pi')] \subseteq [q]$, and $\mathsf{trace}(\pi') = \widetilde{\mathsf{trace}(\pi)}$.*

Proof. Let us write $q_1 = \mathsf{first}(\pi)$ and $q_2 = \mathsf{last}(\pi)$. As we also noted above, $L_\pi(\mathcal{C}_{\mathcal{A}}(\Delta^N))$ is forward and backward reachable from any state of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ on a trace $\mathsf{rep}(\mathsf{trace}(\pi))^+$ so let $p_1, p_2 \in L_\pi(\mathcal{C}_{\mathcal{A}}(\Delta^N))$ such that there is a path in $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ from q_1 to p_1 and from p_2 to q_2 on a trace in $\mathsf{rep}(\mathsf{trace}(\pi))^+$. Let us show that one can choose (l, v_1) and (l, v_2), states of $[\![\mathcal{A}_{\frac{1}{N}}]\!]$ such that $(l, v_1) \in \mathsf{concrete}(\overline{p_1})$, $(l, v_2) \in \mathsf{concrete}(\overline{p_2})$ and $(l, v_1), (l, v_2) \in L_\pi([\![\mathcal{A}_{\frac{1}{N}}]\!])$. Consider any (l, u_1) with $(l, u_1) \in \mathsf{concrete}(\overline{p_1})$ and let π' denote the witness run for $\overline{p_1}$. By Lemma 6, for any $k \geq 1$, there is a run ρ_k in $[\![\mathcal{A}_{\frac{1}{N}}]\!]$ from (l, u_1) to some $(l, u_k) \in \mathsf{concrete}(p_1)$, following ${\pi'}^k$. We can assume that all delays have duration at most $M + 1$ in ρ_k, since any longer delay makes all clocks go above the maximum constant M. Moreover, all clocks that are present in $\mathsf{first}(\pi')$ are bounded (in fact, $\mathsf{reg}(\mathsf{first}(\pi'))$ has finite lower and upper bounds on these clocks). Thus, the sequence of runs ρ_k has a limit point ρ, which is a valid run since all clock constraints are closed. Moreover, $\mathsf{first}(\rho) = \mathsf{last}(\rho) \in \mathsf{clos}(\mathsf{concrete}(\overline{p_1}))$. Hence $\mathsf{first}(\rho) = L_\pi([\![\mathcal{A}]\!]_{\frac{1}{N}})$. We let $(l, v_1) = \mathsf{first}(\rho)$ and define (l, v_2) similarly for $\overline{p_2}$. By Lemma 38, we have $(l, v_1) \in \mathsf{concrete}(p_1)$ and $(l, v_2) \in \mathsf{concrete}(p_2)$. By Lemmas 35 and 36, there is a run from (l, v_1) to (l, v_2) on a trace in $\mathsf{rep}(\mathsf{trace}(\pi))$. By Lemma 6, there is a path from p_1 to some state p_2' such that $(l, v_2) \in \mathsf{concrete}(p_2')$, and also $(l, v_2) \in \mathsf{concrete}(p_2)$ (using Lemma 38). This implies $\mathsf{reg}(p_2) = \mathsf{reg}(p_2')$. Thus, by Lemma 15, there is a path in $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ from p_2' to some state q_2' such that $[q_2'] \subseteq [q_2]$. □

G Extended Region Automaton

In this section, we show how one can extend the usual region automaton by adding new transitions, and obtain a finite automaton that can be used to decide the robust satisfaction of co-Büchi properties. This yields a second algorithm for robust model-checking general timed automata. A similar construction was given in [4] for automata with only progress cycles. The present algorithm can be seen as a generalization to timed automata possibly having progress cycles. Our proofs are different however, since they are based on the results developed in this paper.

For any $L \subseteq \mathcal{L}$, an *L-chain* is a finite automaton given as a chain of size $|L|$, where locations are labeled by the elements of L in an arbitrary order.

Let $\mathcal{A}$ be a timed automaton and let N_1 be the constant defined in Lemma 11. We define the extended region automaton as follows. We start with $\mathcal{R}(\mathcal{A})$, the usual region automaton. We proceed in two steps.

- We define $\mathcal{R}(\mathcal{A})_w$ from $\mathcal{R}(\mathcal{A})$ as follows. For any quasi-exact weak cycle π of $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1})$ of size at most $|\mathcal{L}|^2|\mathcal{C}|K_0$, visiting locations $L \subseteq \mathcal{L}$ and where the active clocks are $A \subseteq \mathcal{C}$, we add an L-chain from $\mathsf{reg}(\mathsf{first}(\pi))$ to $\mathsf{reg}(\mathsf{last}(\pi))$. This chain will be seen as a single transition and written as $\mathsf{reg}(\mathsf{first}(\pi)) \xrightarrow[A,L]{}{}^{w} \mathsf{reg}(\mathsf{last}(\pi))$.
- A *progress cycle* of $\mathcal{R}(\mathcal{A})_w$ is a cycle along which all clocks are reset, either along usual transitions or along weak cycles. $\mathcal{R}(\mathcal{A})^\gamma_w$ is defined from $\mathcal{R}(\mathcal{A})_w$ by adding for any progress cycle π of $\mathcal{R}(\mathcal{A})_w$, a transition from $r \to^\gamma r'$ for all regions r, r' such that $r, r' \subseteq \mathsf{clos}(\mathsf{last}(\pi))$.

Notice that the first step of our construction is based on the encoding of the automaton by a channel machine. The second step is the same as in [4], with a slightly different notion of progress cycles including weak cycles.

We assume that any state of $\mathcal{R}(\mathcal{A})^\gamma_w$ is labelled with the location defined by its region (and the intermediate states in the L-chains are labelled as defined above). The notions of paths and satisfaction of co-Büchi properties are defined as usual. Our result is stated as follows.

Theorem 39. *For any co-Büchi property ϕ, $\mathcal{R}(\mathcal{A})^\gamma_w$ satisfies ϕ if, and only if there exists $\delta > 0$ such that $[\![\mathcal{A}_\delta]\!]$ satisfies ϕ. Moreover, the satisfaction of ϕ by $\mathcal{R}(\mathcal{A})^\gamma_w$ can be checked in polynomial space.*

In the rest, we prove the above theorem.

Lemma 40. *Let π be a quasi-exact path of $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1})$, for N_1 as defined in Lemma 11. Then, for any $\delta > 0$,*

- *for any $(\ell, v) \in \mathsf{clos}(\mathsf{reg}(\mathsf{first}(\pi)))$, there exists a run ρ of $[\![\mathcal{A}_\delta]\!]$ on a trace of $\widetilde{\mathsf{trace}(\pi)}$, with $\mathsf{first}(\rho) = (\ell, v)$ and $\mathsf{last}(\rho) \in \mathsf{clos}(\mathsf{reg}(\mathsf{last}(\pi)))$,*
- *for any $(\ell', v') \in \mathsf{clos}(\mathsf{reg}(\mathsf{last}(\pi)))$, there exists a run ρ of $[\![\mathcal{A}_\delta]\!]$ on a trace of $\widetilde{\mathsf{trace}(\pi)}$ with $\mathsf{first}(\rho) \in \mathsf{clos}(\mathsf{reg}(\mathsf{first}(\pi)))$ and $\mathsf{last}(\rho) = (\ell', v')$.*

Proof. Let $N \geq \max(N_1, 2\lceil 1/\delta \rceil)$ and a state $(\ell, v) \in \mathsf{clos}(\mathsf{reg}(\mathsf{first}(\pi)))$. Let q be the state of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ such that $(\ell, v) \in \mathsf{concrete}(q)$. By Lemma 30, there exists a path π' of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ from q to some state q' with $\mathsf{clos}(\mathsf{reg}(q')) \subseteq \mathsf{clos}(\mathsf{reg}(\mathsf{last}(\pi')))$. The run ρ is then given by Lemma 6.

The proof of the second statement is similar. □

The correctness of $\mathcal{R}(\mathcal{A})^{\gamma}_{w}$ is established by the following two lemmas. We denote by $\mathrm{Inf}(\cdot)$ the subset of $\mathcal{L}$ that is visited infinitely often in a given run.

Lemma 41. *Let $\Pi = (R_i)_{i \geq 0}$ be an infinite path in $\mathcal{R}(\mathcal{A})^{\gamma}_{w}$. Then for all $\delta > 0$, there exists a run ρ of $[\![\mathcal{A}_\delta]\!]$ with $\mathit{Inf}(\rho) = \mathit{Inf}(\Pi)$.*

Proof. We can construct ρ by induction on the length of Π, with $(\rho)_i \in \mathsf{clos}((\Pi)_i)$ for all $i \geq 0$. In fact, any regular transition in Π can be mimicked by $[\![\mathcal{A}_\delta]\!]$ by the well-known properties of regions; for transitions corresponding to weak cycles, one can apply Lemma 40 to the witnessing path of $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1})$; and for transitions corresponding to progress cycles, one can apply Lemma 16. □

Lemma 42. *Let $0 < \delta < \frac{1}{N_1}$, where N_1 is given by Lemma 11, and let ρ be an infinite run of $[\![\mathcal{A}_\delta]\!]$. Then, there exists an infinite path Π of $\mathcal{R}(\mathcal{A})^{\gamma}_{w}$ such that $\mathit{Inf}(\Pi) = \mathit{Inf}(\rho)$.*

Proof. Let $N \geq \max(N_1, \lceil 1/\delta \rceil)$. Let π denote the path of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ given by Lemma 6, and consider π' obtained from π using Lemma 11 with $L = 0$. By construction, π' is an alternation of quasi-exact paths and repeated progress cycles. We show that π' can be followed in $\mathcal{R}(\mathcal{A})^{\gamma}_{w}$, by induction on the length of π'. We construct Π such that $\mathsf{reg}((\pi')_i) = (\Pi)_i$ for each $i \geq 0$. In fact, any regular exact transition from a state $(\pi')_i$ to $(\pi')_{i+1}$ exists in $\mathcal{R}(\mathcal{A})^{\gamma}_{w}$, since it subsumes the region automaton. When the transition between these states is a weak cycle that visits locations $L \subseteq \mathcal{L}$ and where clocks $A \subseteq \mathcal{C}$ are active, then, Lemma 43 ensures that $\mathcal{R}(\mathcal{A})^{\gamma}_{w}$ has the transition $\mathsf{reg}((\pi')_i) \xrightarrow[A,L]{}^{w} \mathsf{reg}((\pi')_{i+1})$. In case of a progress cycle that starts in state $\mathsf{reg}((\pi')_i)$, let $j > i$ denote the greatest index such $\mathsf{reg}((\pi')_i) \dots \mathsf{reg}((\pi')_j)$ is obtained by repeating this progress cycle. We have by construction $\mathsf{clos}(\mathsf{reg}((\pi')_j)) \subseteq \mathsf{clos}(\mathsf{reg}((\pi')_i))$, thus $\mathcal{R}(\mathcal{A})^{\gamma}_{w}$ contains the transition $\mathsf{reg}((\pi')_i) \to^{\gamma} \mathsf{reg}((\pi')_j)$. □

The following lemma completes the proof by justifying the restriction to quasi-exact cycles of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$ of bounded length used in the construction of $\mathcal{R}(\mathcal{A})^{\gamma}_{w}$.

Lemma 43. *For any quasi-exact weak cycle of $\mathcal{C}_{\mathcal{A}}(\Delta^N)$, there exists one that has length at most $|\mathcal{C}|K_0|\mathcal{L}|^2$, with the same source and target regions, the same subset of active clocks, and visiting the same subset of locations.*

Proof. Let π be a quasi-exact weak cycle, and let us write $\pi = \pi_{i_r} \dots \pi_{i_1}$ as in (2). We construct π' from π as follows. First, if block i_m has size $n_m > |\mathcal{C}|K_0$ in $\mathsf{first}(\pi)$, then we can replace it with $|\mathcal{C}|K_0$ and move the $n_m - |\mathcal{C}|K_0$ symbols

of Δ to another block. In fact, since π is a weak cycle clock x_{i_m} is (and possibly other clocks are) not active, so we must have $\mathsf{left}_{\Delta}^{x_{i_m}}(\mathsf{first}(\pi)) \geq 2$. Therefore, there must exist a medium or large block other than i_m, and we can add there these $n - |\mathcal{C}|K_0$ symbols. Then we apply Lemma 20, to shorten some delays so that the resulting path π' satisfies $\mathsf{time}_{\Delta}(\pi') \leq |\mathcal{C}|K_0$. When doing so, we may ensure that if π_j contains a delay τ with $\mathsf{time}_{\Delta}(\tau) \geq 2$, then so does π'_j, and if $\mathsf{time}_{\Delta}(\pi_j) \geq K_0$, then also $\mathsf{time}_{\Delta}(\pi'_j) \geq K_0$. Moreover, we may assume that there is no delays in π' with $\mathsf{time}(\tau) > 0$ and $\mathsf{time}_{\Delta}(\tau) = 0$, since we may either remove these or merge with neighboring delays. We get that $\mathsf{Pumpable}(\pi) = \mathsf{Pumpable}(\pi')$. Now, π' contains at most $|\mathcal{C}|K_0$ non-trivial delay transitions. Let us show that we can also bound the number of action transitions. Consider maximal sequences of action transitions along trivial (null-length) delay transitions. Any sequence of such transitions longer than $|\mathcal{L}|$ contains a cycle. However, we cannot always remove such cycles since the visited locations are important. We will rather remove a cycle if all its locations have already been visited. Thus, we leave at most one such cycle in π', of length at most $|\mathcal{L}|$, per each location in $\mathcal{L}$, and we remove all other cycles. This leaves us a path π' of length $(|\mathcal{L}| - 1)|\mathcal{C}|K_0 + |\mathcal{L}|^2$, which we bound by $|\mathcal{C}|K_0|\mathcal{L}|^2$. □

Now, to obtain the polynomial-space algorithm, notice that $\mathcal{R}(\mathcal{A})^{\gamma}_{w}$ can be explored in non-deterministic polynomial space. In fact, the usual transitions in the region automaton can be taken in polynomial space. At any time, one may guess that there is a weak cycle from the current region to some neighboring one, and guess such a quasi-exact weak cycle in $\mathcal{C}_{\mathcal{A}}(\Delta^{N_1})$. The length of quasi-exact weak cycles can be assumed to be bounded by an exponential. Similarly, at any time, one may guess that there is a progress cycle including the current state. This can be verified in polynomial space, by continuing the exploration and waiting until the exploration comes back to current state by resetting all clocks. If this is the case, then one may proceed to any neighboring regions, following the definition of the γ-transitions.