Timed Basic Parallel Processes

Lorenzo Clemente
University of Warsaw, Poland

Piotr Hofman
University of Warsaw, Poland

Patrick Totzke
University of Liverpool, UK

Abstract

Timed basic parallel processes (TBPP) extend communication-free Petri nets (aka. BPP or commutative context-free grammars) by a global notion of time. TBPP can be seen as an extension of timed automata (TA) with context-free branching rules, and as such may be used to model networks of independent timed automata with process creation. We show that the coverability and reachability problems (with unary encoded target multiplicities) are PSPACE-complete and EXPTIME-complete, respectively. For the special case of 1-clock TBPP, both are NP-complete and hence not more complex than for untimed BPP. This contrasts with known super-Ackermannian-completeness and undecidability results for general timed Petri nets. As a result of independent interest, and basis for our NP upper bounds, we show that the reachability relation of 1-clock TA can be expressed by a formula of polynomial size in the existential fragment of linear arithmetic, which improves on recent results from the literature.

2012 ACM Subject Classification Theory of computation → Timed and hybrid models

Keywords and phrases Timed Automata, Petri Nets

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2019.15

Related Version An accompanying technical report is available at `https://arxiv.org/abs/1907.01240` [17].

Funding *Lorenzo Clemente*: Partially supported by Polish NCN grant 2017/26/D/ST6/00201.
Piotr Hofman: Partially supported by Polish NCN grant 2016/21/D/ST6/01368.

Acknowledgements Many thanks to Rasmus Ibsen-Jensen for helpful discussions and pointing us towards [29].

1 Introduction

We study safety properties of unbounded networks of timed processes, where time is global and elapses at the same rate for every process. Each process is a *timed automaton* (TA) [7] controlling its own set of private clocks, not accessible to the other processes. A process can dynamically create new sub-processes, which are thereafter independent from each other and their parent, and can also terminate its execution and disappear from the network.

While such systems can be conveniently modelled in *timed Petri nets* (TdPN), verification problems for this model are either undecidable or prohibitively complex: The reachability problem is undecidable even when individual processes carry only one clock [41] and the coverability problem is undecidable for two or more clocks. In the one-clock case coverability remains decidable but its complexity is hyper-Ackermannian [6, 28].

These hardness results however require unrestricted synchronization between processes, which motivates us to study of the communication-free fragment of TdPN, called *timed basic parallel processes* (TBPP) in this paper. This model subsumes both TA and communication-

free Petri nets (a.k.a. BPP [14, 24]). The general picture that we obtain is that extending communication-free Petri nets by a global notion of time comes at no extra cost in the complexity of safety checking, and it improves on the prohibitive complexities of TdPN.

Our contributions. We show that the TBPP coverability problem is PSPACE-complete, matching same complexity for TA [7, 25], and that the more general TBPP reachability problem is EXPTIME-complete, thus improving on the undecidability of TdPN. The lower bounds already hold for TBPP with two clocks if constants are encoded in binary; EXPTIME-hardness for reachability with no restriction on the number of clocks holds for constants in $\{0, 1\}$. The upper bounds are obtained by reduction to TA reachability and reachability games [30], and assume that process multiplicities in target configurations are given in unary.

In the single-clock case, we show that both TBPP coverability and reachability are NP-complete, matching the same complexity for (untimed) BPP [24]. This paves the way for the automatic verification of unbounded networks of 1-clock timed processes, which is currently lacking in mainstream verification tools such as UPPAAL [34] and KRONOS [46]. The NP lower bound already holds when the target configuration has size 2; when it has size one, 1-clock TBPP coverability becomes NL-complete, again matching the same complexity for 1-clock TA [33] (and we conjecture that 1-clock reachability is in PTIME under the same restriction).

As a contribution of independent interest, we show that the ternary reachability relation of 1-clock TA can be expressed by a formula of existential linear arithmetic ($\exists$LA) of polynomial size. By *ternary reachability relation* we mean the family of relations $\{\rightarrow_{pq}\}$ s.t. $\mu \xrightarrow{\delta}_{pq} \nu$ holds if from control location p and clock valuation $\mu \in \mathbb{R}^k_{\geq 0}$ it is possible to reach control location q and clock valuation $\nu \in \mathbb{R}^k_{\geq 0}$ in exactly $\delta \in \mathbb{R}_{\geq 0}$ time. This should be contrasted with analogous results (cf. [27]) which construct formulas of exponential size, even in the case of 1-clock TA. Since the satisfiability problem for $\exists$LA is decidable in NP, we obtain a NP upper bound to decide ternary reachability $\rightarrow_{pq}$. We show that the logical approach is optimal by providing a matching NP lower bound for the same problem. Our NP upper bounds for the 1-clock TBPP coverability and reachability problems are obtained as an application of our logical expressibility result above, and the fact that $\exists$LA is in NP; as a further technical ingredient we use polynomial bounds on the piecewise-linear description of value functions in 1-clock priced timed games [29].

Related research. Starting from the seminal PSPACE-completeness result of the nonemptiness problem for TA [7] (cf. also [25]), a rich literature has emerged considered more challenging verification problems, including the symbolic description of the reachability relation [20, 22, 31, 23, 27]. There are many natural generalizations of TA to add extra modelling capabilities, including *time Petri Nets* [36, 38] (which associate timing constraints to transitions) the already mentioned timed Petri nets (TdPN) [41, 6, 28] (where tokens carry clocks which are tested by transitions), *networks of timed processes* [5], several variants of *timed pushdown automata* [12, 21, 8, 43, 2, 40, 9, 19, 18], *timed communicating automata* [32, 16, 4, 15], and their lossy variant [1], and timed process calculi based on Milners CCS (e.g. [10]). While decision problems for TdPN have prohibitive complexity/are undecidable, it has recently been shown that structural safety properties are PSPACE-complete using forward accelerations [3].

Outline. In Section 2 we define TBPP and their reachability and coverability decision problems. In Section 3 we show that the reachability relation for 1-clock timed automata can be expressed in polynomial time in an existential formula of linear arithmetic, and that

the latter logic is in NP. We apply this result in Sec. 4 to show that the reachability and coverability problems for 1-clock TBPP is NP-complete. Finally, in Section 5 we study the case of TBPP with $k \geq 2$ clocks, and in Section 6 we draw conclusions. Full proofs can be found in the technical report [17].

2 Preliminaries

Notations. We use $\mathbb{N}$ and $\mathbb{R}_{\geq 0}$ to denote the sets of nonnegative integers and reals, respectively. For $c \in \mathbb{R}_{\geq 0}$ we write $\mathit{int}(c) \in \mathbb{N}$ for its integer part and $\mathit{frac}(c) \stackrel{def}{=} c - \mathit{int}(c)$ for its fractional part. For a set $\mathcal{X}$, we use $\mathcal{X}^*$ to denote the set of finite sequences over $\mathcal{X}$ and $\mathcal{X}^{\oplus}$ to denote the set of finite multisets over $\mathcal{X}$, i.e., functions $\mathbb{N}^{\mathcal{X}}$. We denote the empty multiset by $\emptyset$, we denote the union of two multisets $\alpha, \beta \in \mathbb{N}^{\mathcal{X}}$ by $\alpha + \beta$, which is defined point-wise, and by $\alpha \leq \beta$ we denote the natural partial order on multisets, also defined point-wise. The *size* of a multiset $\alpha \in \mathbb{N}^{\mathcal{X}}$ is $|\alpha| = \sum_{X \in \mathcal{X}} \alpha(X)$. We overload notation and we let $X \in \mathcal{X}$ denote the singleton multiset of size 1 containing element X. For example, if $X, Y \in \mathcal{X}$, then the multiset consisting of 1 occurrence of X and 2 of Y will be denoted by $\alpha = X + Y + Y$; it has size $|\alpha| = 3$.

Clocks. Let $\mathcal{C}$ be a finite set of *clocks*. A *clock valuation* is a function $\mu \in \mathbb{R}_{\geq 0}^{\mathcal{C}}$ assigning a nonnegative real to every clock. For $t \in \mathbb{R}_{\geq 0}$, we write $\mu + t$ for the valuation that maps clock $x \in \mathcal{C}$ to $\mu(x) + t$. For a clock $x \in \mathcal{C}$ and a clock or constant $e \in \mathcal{C} \cup \mathbb{N}$ let $\mu[x := e]$ be the valuation ν s.t. $\nu(x) = \mu(e)$ and $\nu(z) = \mu(z)$ for every other clock $z \neq x$ (where we assume $\mu(k) = k$ for a constant $k \in \mathbb{N}$); for a sequence of assignments $R = (x_1 := e_1; \cdots; x_n := e_n)$ let $\mu[R] = \mu[x_1 := e_1] \cdots [x_n := e_n]$. A *clock constraint* is a conjunction of linear inequalities of the form $c \bowtie k$, where $c \in \mathcal{C}$, $k \in \mathbb{N}$, and $\bowtie \in \{<, \leq, =, \geq, >\}$; we also allow **true** for the trivial constraint which is always satisfied. We write $\mu \models \varphi$ to denote that the valuation μ satisfies the constraint φ.

Timed basic parallel processes. A *timed basic parallel process* (TBPP) consists of finite sets $\mathcal{C}$, $\mathcal{X}$, and $\mathcal{R}$ of clocks, nonterminal symbols, and rules. Each rule is of the form

$$X \stackrel{\varphi;R}{\Longrightarrow} \alpha$$

where $X \in \mathcal{X}$ is a nonterminal, φ is a clock constraint, R is a sequence of assignments of the form $x := e$, where e is either a constant in $\mathbb{N}$ or a clock in $\mathcal{C}$, and $\alpha \in \mathcal{X}^{\oplus}$ is a finite multiset of successor nonterminals[1]. Whenever the test $\varphi \equiv$ **true** is trivial, or R is the empty sequence, we just omit the corresponding component and just write $X \stackrel{\varphi}{\Longrightarrow} \alpha$, $X \stackrel{R}{\Longrightarrow} \alpha$, or $X \Longrightarrow \alpha$. Finally, we say that we *reset* the clock x_i if we assign it to 0.

Henceforth, we assume w.l.o.g. that the size $|\alpha|$ is at most 2. A rule with $\alpha = \emptyset$ is called a *vanishing* rule, and a rule with $|\alpha| = 2$ is called a *branching* rule. We will write k-TBPP to denote the class of TBPP with k clocks.

A *process* is a pair $(X, \mu) \in \mathcal{X} \times \mathbb{R}_{\geq 0}^{\mathcal{C}}$ comprised of a nonterminal X and a clock valuation μ, and a *configuration* α is a multiset of processes, i.e., $\alpha \in (\mathcal{X} \times \mathbb{R}_{\geq 0}^{\mathcal{C}})^{\oplus}$. For a process $P = (X, \mu)$ and $t \in \mathbb{R}_{\geq 0}$, we denote by $P + t$ the process $(X, \mu + t)$, and for a

[1] We note that clock updates $x := k$ with $k \in \mathbb{N}$ can be encoded with only a polynomial blow-up by replacing them with $x := 0$, while recording in the finite control the last update k, and replacing a test $x \bowtie h$ with $x \bowtie h - k$. We use them as a syntactic sugar to simplify the presentation of some constructions.

configuration $\alpha = P_1 + \cdots + P_n$, we denote by $\alpha + t$ the configuration $Q_1 + \cdots + Q_n$, where $Q_1 = P_1 + t, \ldots, Q_n = P_n + t$. The semantics of a TBPP $(\mathcal{C}, \mathcal{X}, \mathcal{R})$ is given by an infinite timed transition system $(C, \to)$, where $C = (\mathcal{X} \times \mathbb{R}_{\geq 0}^{\mathcal{C}})^{\oplus}$ is the set of configurations, and $\to \subseteq C \times \mathbb{R}_{\geq 0} \times C$ is the transition relation between configurations. There are two kinds of transitions:

Time elapse: For every configuration $\alpha \in C$ and $t \in \mathbb{R}_{\geq 0}$, there is a transition $\alpha \xrightarrow{t} \alpha + t$ in which all clocks in all processes are simultaneously increased by t. In particular, the empty configuration stutters: $\emptyset \xrightarrow{t} \emptyset$, for every $t \in \mathbb{R}_{\geq 0}$.

Discrete transitions: For every configuration $\gamma = \alpha + (X, \mu) + \beta \in C$ and rule $X \xRightarrow{\varphi; R} Y + Z$ s.t. $\mu \models \varphi$ there is a transition $\gamma \xrightarrow{0} \alpha + (Y, \nu) + (Z, \nu) + \beta$, where $\nu = \mu[R]$. Analogously, rules $X \xRightarrow{\varphi; R} Y$ and $X \xRightarrow{\varphi; R} \emptyset$ induce transitions $\gamma \xrightarrow{0} \alpha + (Y, \nu) + \beta$ and $\gamma \xrightarrow{0} \alpha + \beta$.

A *run* starting in α and ending in β is a sequence of transitions $\alpha = \alpha_0 \xrightarrow{t_1} \alpha_1 \cdots \xrightarrow{t_n} \alpha_n = \beta$. We write $\alpha \xrightarrow{t} \beta$ whenever there is a run as above where the sum of delays is $t = t_1 + \cdots + t_n$, and we write $\alpha \xrightarrow{*} \beta$ whenever $\alpha \xrightarrow{t} \beta$ for some $t \in \mathbb{R}_{\geq 0}$.

TBPP generalise several known models: A *timed automaton* (TA) [7] is a TBPP without branching rules; in the context of TA, we will sometimes call nonterminals with the more standard name of *control locations*. Untimed *basic parallel processes* (BPP) [14, 24] are TBPP over the empty set of clocks $\mathcal{X} = \emptyset$. TBPP can also be seen as a structural restriction of *timed Petri nets* [6, 28] where each transition consumes only one token at a time.

TBPP are related to *alternating timed automata* (ATA) [37, 35]: Branching in TBPP rules corresponds to universal transitions in ATA. However, ATA offer additional means of synchronisation between the different branches of a run tree: While in a TBPP synchronisation is possible only through the elapse of time, in an ATA all branches must read the same timed input word.

Decision problems. We are interested in checking safety properties of TBPP in the form of the following decision problems. The *reachability problem* asks whether a target configuration is reachable from a source configuration.

Input: A TBPP $(\mathcal{C}, \mathcal{X}, \mathcal{R})$, an initial $X \in \mathcal{X}$ and target nonterminals $T_1, \ldots, T_n \in \mathcal{X}$.
Question: Does $(X, \vec{0}) \xrightarrow{*} (T_1, \vec{0}) + \cdots + (T_n, \vec{0})$ hold?

It is crucial that we reach all processes in the target configurations *at the same time*, which provides an external form of global synchronisation between processes.

Motivated both by complexity considerations and applications for safety checking, we study the *coverability problem*, where it suffices to reach some configuration larger than the given target in the multiset order. For configurations $\alpha, \beta \in (\mathcal{X} \times \mathbb{R}_{\geq 0}^{\mathcal{C}})^{\oplus}$, let $\alpha \xrightarrow{*} \cdot \geq \beta$ whenever there exists $\gamma \in (\mathcal{X} \times \mathbb{R}_{\geq 0}^{\mathcal{C}})^{\oplus}$ s.t. $\alpha \xrightarrow{*} \gamma \geq \beta$.

Input: A TBPP $(\mathcal{C}, \mathcal{X}, \mathcal{R})$, an initial $X \in \mathcal{X}$ and target nonterminals $T_1, \ldots, T_n \in \mathcal{X}$.
Question: Does $(X, \vec{0}) \xrightarrow{*} \cdot \geq (T_1, \vec{0}) + \cdots + (T_n, \vec{0})$?

The *simple* reachability/coverability problems are as above but with the restriction that the target configuration is of size 1, i.e., a single process. Notice that this is a proper restriction, since reachability and coverability do not reduce in general to their simple variant. Finally, the *non-emptiness problem* is the special case of the reachability problem where the target configuration α is the empty multiset $\emptyset$.

In all decision problems above the restriction to zero-valued clocks in the initial process is mere convenience, since we could introduce a new initial nonterminal Y and a transition $Y \xrightarrow{x_1:=\mu(x_1);\ldots;x_n:=\mu(x_n)} X$ initialising the clocks to the initial values provided by the (rational) clock valuation $\mu \in \mathbb{Q}^{\mathcal{C}}_{\geq 0}$. Similarly, if we wanted to reach the final configuration $\alpha = (X_1, \mu_1) + (X_2, \mu_2)$ with $\mu_1, \mu_2 \in \mathbb{Q}^{\mathcal{C}}_{\geq 0}$, then we could add two nonterminals Y_1, Y_2 and two new rules $X_1 \xrightarrow{x_1=\mu_1(x_1)\wedge\cdots\wedge x_n=\mu_1(x_n);x_1:=0;\ldots;x_n:=0} Y_1$ and $X_2 \xrightarrow{x_1=\mu_2(x_1)\wedge\cdots\wedge x_n=\mu_2(x_n);x_1:=0;\ldots;x_n:=0} Y_2$ and check whether $X \xrightarrow{*} (Y_1, \vec{0}) + (Y_2, \vec{0})$ holds. (It is standard to transform TBPP with constraints of the form $x_i = k$ with $k \in \mathbb{Q}_{\geq 0}$ in the form $x_i = k$ with $k \in \mathbb{N}$.) Similarly, the restriction of having just one initial nonterminal process is also w.l.o.g., since if we wanted to check reachability from $(X_1, \vec{0}) + (X_2, \vec{0})$ we could just add a new initial nonterminal X and a branching rule $X \Longrightarrow X_1 + X_2$.

For complexity considerations we will assume that all constants appearing in clock constraints are given in binary encoding, and that the multiplicities of target processes are in unary.

3 Reachability Relations of One-Clock Timed Automata

In this section we show that the reachability relation of 1-clock TA is expressible as an existential formula of linear arithmetic of polynomial size. Since the latter fragment is in NP, this gives an NP algorithm to check whether a family of TA can reach the respective final locations *at the same time*. This result will be applied in Sec. 4 to show that coverability and reachability of 1-TBPP are in NP. We first show that existential linear arithmetic is in NP (which is an observation of independent interest), and then how to express the reachability relation of 1-TA in existential linear arithmetic in polynomial time.

The set of *terms* t is generated by the following abstract grammar

$$s, t \ ::= \ x \mid k \mid \lfloor t \rfloor \mid \mathit{frac}(t) \mid -t \mid s + t \mid k \cdot t,$$

where x is a rational variable, $k \in \mathbb{Z}$ is an integer constant encoded in binary, $\lfloor t \rfloor$ represents the integral part of t, and $\mathit{frac}(t)$ its fractional part. *Linear arithmetic* (LA) is the first order language with atomic proposition of the form $s \leq t$ [45], we denote by $\exists$LA its existential fragment, and by qf-LA its quantifier-free fragment. Linear arithmetic generalises both *Presburger arithmetic* (PA) and *rational arithmetic* (RA), whose existential fragments are known to be in NP [39, 26]. This can be generalised to $\exists$LA. (The same result can be derived from the analysis of [11, Theorem 3.1]).

▶ **Theorem 1.** *The existential fragment $\exists$LA of LA is in* NP.

Let $\mathcal{A} = (\mathcal{C}, \mathcal{X}, \mathcal{R})$ be a k-TA. The *ternary reachability relation* of $\mathcal{A}$ is the family of relations $\{\rightarrow_{XY}\}_{X,Y\in\mathcal{X}}$, where each $\rightarrow_{XY} \subseteq \mathbb{R}^{\mathcal{C}}_{\geq 0} \times \mathbb{R}_{\geq 0} \times \mathbb{R}^{\mathcal{C}}_{\geq 0}$ is defined as: $\mu \xrightarrow{\delta}_{XY} \nu$ iff $(X, \mu) \xrightarrow{\delta} (Y, \nu)$. We say that the reachability relation is *expressed* by a family of LA formulas $\{\varphi_{XY}\}_{X,Y\in\mathcal{X}}$ if

$$\mu \xrightarrow{\delta}_{XY} \nu \quad \text{iff} \quad (\mu, \delta, \nu) \models \varphi_{XY}(\vec{x}, t, \vec{y}), \text{ for every } X, Y \in \mathcal{X}, \mu, \nu \in \mathbb{R}^{\mathcal{C}}_{\geq 0}, \delta \in \mathbb{R}_{\geq 0}.$$

In the formula $\varphi_{XY}(\vec{x}, t, \vec{y})$, $\vec{x}$ are k variables representing the clock values in location X at the beginning of the run, $\vec{y}$ are k variables representing the clock values in location Y at the end of the run, and t is a single variable representing the total time elapsed during the run. In the rest of this section, we assume that the TA has only one clock $\mathcal{X} = \{x\}$.

The main result of this section is that 1-TA reachability relations are expressible by ∃LA formulas constructible in polynomial time.

▶ **Theorem 2.** *Let $\mathcal{A}$ be a 1-TA. The reachability relation $\{\rightarrow_{XY}\}_{X,Y\in\mathcal{X}}$ is expressible as a family of formulas $\{\varphi_{XY}\}_{X,Y\in\mathcal{X}}$ of existential linear arithmetic ∃LA in polynomial time.*

In the rest of the section we prove the theorem above. We begin with some preliminaries.

Interval abstraction. We replace the integer value of the clock x by its interval [33]. Let $0 = k_0 < k_1 < \cdots < k_n < k_{n+1} = \infty$ be all integer constants appearing in constraints of $\mathcal{A}$, and let the set of intervals be the following totally ordered set:

$$\Lambda = \{\{k_0\} < (k_0, k_1) < \{k_1\} < \cdots < (k_{n-1}, k_n) < \{k_n\} < (k_n, k_{n+1})\}\,.$$

Clearly, we can resolve any constraint of $\mathcal{A}$ by looking at the interval $\lambda \in \Lambda$. We write $\lambda \models \varphi$ whenever $v \models \varphi$ for some $v \in \lambda$ (whose choice does not matter by the definition of λ).

The construction. Let $\mathcal{A} = (\{x\}, \mathcal{X}, \mathcal{R})$ be a TA. In order to simplify the presentation below, we assume w.l.o.g. that the only clock updates are resets $x := 0$ (cf. footnote 1). We build an NFA $\mathcal{B} = (\Sigma, Q, \rightarrow)$ where Σ contains symbols (r, ε) and $(r, \checkmark_\lambda)$ for every transition $r \in \mathcal{R}$ of $\mathcal{A}$ and interval $\lambda \in \Lambda$, and an additional symbol τ representing time elapse, and $Q = \mathcal{X} \times \Lambda$ is a set of states of the form (X, λ), where $X \in \mathcal{X}$ is a control location of $\mathcal{A}$ and $\lambda \in \Lambda$ is an interval. Transitions $\rightarrow \subseteq Q \times \Sigma \times Q$ are defined as follows. A rule $r = X \xrightarrow{\varphi;R} Y \in \mathcal{R}$ of $\mathcal{A}$ generates one or more transitions in $\mathcal{B}$ of the form

$$(X, \lambda) \xrightarrow{(r,a)} (Y, \mu)$$

whenever $\lambda \models \varphi$ and any of the following two conditions is satisfied:

- the clock is not reset i.e R is equal $x := x$, and $\mu = \lambda, a = \varepsilon$, or
- the clock is reset $x := 0$, $\mu = \{0\}$, and the automaton emits a tick $a = \checkmark_\lambda$.

A time elapse transition is simulated in $\mathcal{B}$ by transitions of the form

$$(X, \lambda) \xrightarrow{\tau} (X, \mu), \qquad \lambda \leq \mu \text{ (the total ordering on intervals).}$$

Reachability relation of $\mathcal{A}$. For a set of finite words $L \subseteq \Sigma^*$, let $\psi_L(\vec{y})$ be a formula of existential Presburger arithmetic with a free integral variable y_λ for every interval $\lambda \in \Lambda$ counting the number of symbols of the form $(r, \checkmark_\lambda)$, for some $r \in \mathcal{R}$. The formula ψ_L can be computed from the Parikh image of L: By [44, Theorem 4], a formula $\tilde{\psi}_L(\vec{z})$ of existential Presburger arithmetic can be computed in linear time from an NFA (or even a context-free grammar) recognising L, and then one just defines $\psi_L(\vec{y}) \equiv \exists \vec{z} \cdot \tilde{\psi}_L(\vec{z}) \wedge \bigwedge_{\lambda\in\Lambda} y_\lambda = \sum_{r\in\mathcal{R}} z_{r,\lambda}$. Let L_{cd} be the regular language recognised by $\mathcal{B}$ by making c initial and d final, and let $\Lambda = \{\lambda_0, \ldots, \lambda_{2n+1}\}$ contain $2n + 2$ intervals. Let $\psi_{cd}(x, t, x')$ be a formula of existential Presburger arithmetic computing the total elapsed time t, given the initial x and final x' values of the unique clock:

$$\begin{aligned}
\psi_{cd}(x,t,x') \;\equiv\; & \exists y_0, \ldots, y_{2n+1} \cdot \psi_{L_{cd}}(\lfloor y_0 \rfloor, \ldots, \lfloor y_{2n+1} \rfloor) \;\wedge \\
& \exists z_0, \ldots, z_{2n+1} \cdot \bigwedge_{\lambda_i \in \Lambda} (z_i \in y_i \cdot \lambda_i) \;\wedge\; t = x' - x + \sum_{\lambda_i \in \Lambda} z_i, \text{ where} \\
z \in y \cdot \lambda \;\equiv\; & \begin{cases} a \cdot y < z < b \cdot y & \text{if } \lambda = (a, b), \\ z = a \cdot y & \text{if } \lambda = \{a\}\,. \end{cases}
\end{aligned}$$

Intuitively, y_i represents the total number of times the clock is reset while in interval λ_i, and z_i represents the sum of the values of the clock when it is reset in interval λ_i. For control locations X, Y of $\mathcal{A}$, let

$$\varphi_{XY}(x,t,x') \equiv \bigvee_{\lambda,\mu\in\Lambda} \{x \in \lambda \wedge x' \in \mu \wedge \psi_{cd}(x,t,x') \mid c = (X,\lambda), d = (Y,\mu)\}.$$

The correctness of the construction is stated below.

▶ **Lemma 3.** *For every configurations (X,u) and (Y,v) of $\mathcal{A}$ and total time elapse $\delta \geq 0$,*

$$u \xrightarrow{\delta}_{XY} v \quad \textit{iff} \quad (u,\delta,v) \models \varphi_{XY}(x,t,x').$$

We conclude this section by applying Theorem 2 to solve the 1-TA ternary reachability problem. The *ternary reachability problem* takes as input a TA $\mathcal{A}$ as above, with two distinguished control locations $X, Y \in \mathcal{X}$, and a total duration $\delta \in \mathbb{Q}$ (encoded in binary), and asks whether $(\vec{0}) \xrightarrow{\delta}_{XY} (\vec{0})$. The result below shows that computing 1-TA reachability relations is optimal in order to solve the ternary reachability problem.

▶ **Theorem 4.** *The ternary reachability problem for 1-TA is* NP*-complete.*

Proof. For the upper bound, apply Theorem 2 to construct in polynomial time a formula of $\exists$LA expressing the reachability relation and check satisfiability in NP thanks to Theorem 1.

The lower bound can be seen by reduction from SubsetSum. Let $\mathcal{S} = \{a_1, \ldots, a_k\} \subseteq \mathbb{N}$ and $a \in \mathbb{N}$ be the input to the subset sum problem, whereby we look for a subset $\mathcal{S}' \subseteq \mathcal{S}$ s.t. $a = \sum_{b\in\mathcal{S}'} b$. We construct a TA with a single clock x and locations $\mathcal{X} = \{X_0, \ldots, X_k\}$, where X_0 is the initial location and X_k the target. A path through the system describes a subset by spending exactly 0 or a_i time in location X_i (see Figure 1). In the constructed automaton, $(X_1, 0) \xrightarrow{a} (X_k, 0)$ iff the subset sum instance was positive. ◀

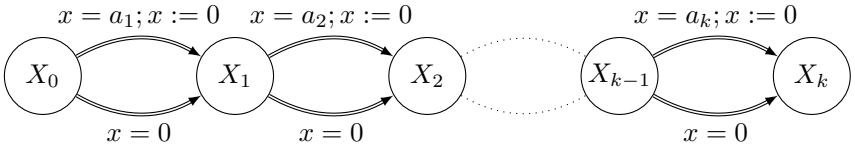

Figure 1 Reduction from subset sum to 1-TA (ternary) reachability. We have $(X_0, 0) \xrightarrow{t} (X_k, 0)$ iff $t = \sum_{b\in\mathcal{S}'} b$ for some subset $\mathcal{S}' \subseteq \{a_1, \ldots, a_k\}$.

4 One-Clock TBPP

As a warm-up we note that the simple coverability problem for 1-TBPP, where the target has size one, is inter-reducible with the reachability problem for 1-clock TA and hence NL-complete [33].

▶ **Theorem 5.** *The simple coverability problem for 1-clock TBPP is* NL*-complete.*

Proof. The lower bound is trivial since 1-TBPP generalize 1-TA. For the other direction we can transform a given TBPP into a TA by replacing branching rules of the form $X \xRightarrow{\varphi,R} Y + Z$ with two rules $X \xRightarrow{\varphi,R} Y$ and $X \xRightarrow{\varphi,R} Z$. In the constructed TA we have $(X,\mu) \xrightarrow{*} (Y,\nu)$ if, and only if, $(X,\mu) \xrightarrow{*} (Y,\nu) + \gamma$ for some γ in the original TBPP. ◀

The construction above works because the target is a single process and so there are no constraints on the other processes in γ, which were produced as side-effects by the branching rules. The (non-simple) 1-TBPP coverability problem is in fact NP-complete. Indeed, even for untimed BPP, coverability is NP-hard, and this holds already when target sets are encoded in unary (which is the setting we are considering here) [24]. We show that for 1-TBPP this lower bound already holds if the target has fixed size 2.

▶ **Lemma 6.** *Coverability is* NP*-hard for 1-TBPP already for target sets of size* ≥ 2*.*

Proof. We proceed by reduction from subset sum as in Theorem 4. The only difference will be that here, we use one extra process to keep track of the total elapsed time. Let $\mathcal{S} = \{a_1, \ldots, a_k\} \subseteq \mathbb{N}$ and $t \in \mathbb{N}$ be the input to the subset sum problem. We construct a 1-TBPP with nonterminals $\mathcal{X} = \{S, X_0, \ldots, X_k, Y\}$, where S is the initial nonterminal and Y will be used to keep track of the total time elapsed. The rules are as in the proof of Theorem 4, and additionally we have an initial branching rule $S \overset{x=0}{\Longrightarrow} X_0 + Y$. We have $(S,0) \xrightarrow{*} \cdot \geq (X_k, 0) + (Y, t)$ if, and only if, $t = \sum_{b \in \mathbb{B}} b$ for some subset $\mathcal{S}' \subseteq \mathcal{S}$. ◀

In the remainder of this section we will argue (Theorem 11) that a matching NP upper bound even holds for the *reachability* problem for 1-TBPP. Let us first motivate the key idea behind the construction. Consider the following TBPP coverability query:

$$(S,0) \xrightarrow{*} \cdot \geq (A,0) + (B,0). \tag{$\dagger$}$$

If ($\dagger$) holds, then there is a derivation tree witnessing that $(S,0) \xrightarrow{*} (A,0) + (B,0) + \gamma$ for some configuration γ. The least common ancestor of leaves $(A,0)$ and $(B,0)$ is some process $(C,c) \in (\mathcal{X} \times \mathbb{R}_{\geq 0})$. Consider the TA $\mathcal{A}$ obtained from the TBPP by replacing branching rules $X \overset{\varphi;R}{\Longrightarrow} X_i + X_j$ with linear rules $X \overset{\varphi;R}{\Longrightarrow} X_i$ and $X \overset{\varphi;R}{\Longrightarrow} X_j$, and let the reachability relation of $\mathcal{A}$ be expressed by ∃LA formulas $\{\varphi_{XY}\}_{X,Y \in \mathcal{X}}$, which are of polynomial size by Theorem 2. Then our original coverability query ($\dagger$) is equivalent to satisfiability the following ∃LA formula:

$$\psi \;\equiv\; \exists t_0, t_1, c \in \mathbb{R} \cdot (\varphi_{SC}(0, t_0, c) \;\wedge \varphi_{CA}(c, t_1, 0) \wedge \varphi_{CB}(c, t_1, 0))\,.$$

More generally, for any coverability query $(S,0) \xrightarrow{*} \cdot \geq \alpha$ the number of common ancestors is linear in $|\alpha|$, and thus we obtain a ∃LA formula ψ of polynomial size, whose satisfiability we can check in NP thanks to Theorem 1.

▶ **Theorem 7.** *The coverability problem for 1-clock TBPP is* NP*-complete.*

In order to witness reachability instances we need to refine the argument above to restrict the TA in such a way that they do not accidentally produce processes that cannot be removed in time. To illustrate this point, consider a 1-TBPP with rules

$$X \overset{x=0}{\Longrightarrow} Y + Z \qquad \text{and} \qquad Z \overset{x>0}{\Longrightarrow} \emptyset.$$

Clearly $(X,0) \xrightarrow{*} (Y,0)$ holds in the TA with rules $X \overset{x=0}{\Longrightarrow} Y$ and $X \overset{x=0}{\Longrightarrow} Z$ instead of the branching rule above. In the TBPP however, $(X,0)$ cannot reach $(Y,0)$ because the branching rule produces a process $(Z,0)$, which needs a positive amount of time to be rewritten to $\emptyset$.

▶ **Definition 8.** *For a nonterminal X let $Vanish_X \subseteq \mathbb{R}^2$ be the binary predicate such that*

$$Vanish_X(x,t) \quad \text{if} \quad (X,x) \xrightarrow{t} \emptyset$$

Intuitively, $Vanish_X(x,t)$ holds if the configuration (X,x) can vanish in time at most t.

The time it takes to remove a processes (Z,z) can be computed as the value of a one-clock priced timed game [13, 29]. These are two-player games played on 1-clock TA where players aim to minimize/maximize the cost of a play leading up to a designated target state. Nonnegative costs may be incurred either by taking transitions, or by letting time elapse. In the latter case, the incurred cost is a linear function of time, determined by the current control-state. Bouyer et al. [13] prove that such games admit ε-optimal strategies for both players, so have well-defined cost value functions determining the best cost as a function of control-states and clock valuation. They prove that these value functions are in fact piecewise-linear. Hansen et al. [29] later show that the piecewise-linear description has only polynomially many line segments and can be computed in polynomial time[2]. We derive the following lemma.

▶ **Lemma 9.** *A qf-LA formula expressing $Vanish_X$ is effectively computable in polynomial time. More precisely, there is a set $\mathcal{I}$ of polynomially many consecutive intervals $\{a_0\}(a_0,a_1)\{a_1\}(a_1,a_2)\{a_2\},\dots(a_k,\infty)$ so that*

$$Vanish_X(x,t) \;\equiv \bigvee_{0\le i\le k} (x = a_i \wedge t \geq_i c_i) \;\vee\; (a_i < x < a_{i+1} \wedge t \geq_i c_i - b_i x),$$

where the $a_i, c_i \in \mathbb{R}$ can be represented using polynomially many bits, $\geq_i \in \{\geq, >\}$ and $b_i \in \{0,1\}$, for all $0 \le i < k$.

Proof (Sketch). One can construct a one-clock priced timed game in which minimizer's strategies correspond to derivation trees. To do this, let unary rules $X \overset{\varphi;R}{\Longrightarrow} Y$ carry over as transitions between (minimizer) states X, Y; vanishing rules $X \overset{\varphi}{\Longrightarrow} \emptyset$ are replaced by transitions leading to a new target state $\bot$, which has a clock-resetting self-loop. Branching rules $X \overset{\varphi;R}{\Longrightarrow} Y + Z$ can be implemented by rules $X \overset{\varphi}{\Longrightarrow} [Y,Z,\varphi]$, $[Y,Z,\varphi] \overset{\varphi;R}{\Longrightarrow} Y$ and $[Y,Z,\varphi] \overset{\varphi;R}{\Longrightarrow} Z$, where X, Y, Z are minimizer states and $[Y,Z,\varphi]$ is a maximizer state. The cost of staying in a state is 1, transitions carry no costs. Moreover, we need to prevent maximizer from elapsing time from the states she controls. For this reason, we consider an extension of price timed games where maximizer cannot elapse time. In the constructed game, minimizer has a strategy to reach $(\bot, 0)$ from (X,x) at cost t iff $(X,x) \xrightarrow{t} \emptyset$. The result now follows from [29, Theorem 4.11] (with minor adaptations in order to consider the more restrictive case where maximizer cannot elapse time) that computes value functions for the cost of reachability in priced timed games. These are piecewise-linear with only polynomially many line segments of slopes 0 or 1 which allows to present $Vanish_X$ in qf-LA as stated. ◀

Lemma 9 allows us to compute a polynomial number of intervals $\mathcal{I}$ sufficient to describe the $Vanish_X$ predicates. We will call a pair $(X,I) \in \mathcal{X} \times \mathcal{I}$ a *region* here. A crucial ingredient for our construction will be timed automata that are restricted in which regions they are allowed to produce as side-effects. To simplify notations let us assume w.l.o.g. that the given

[2] This observation was already made, without proof, in [42, Sec. 7.2.2].

TBPP has no resets along branching rules. Let $I(r) \in \mathcal{I}$ denote the unique interval containing $r \in \mathbb{R}_{\geq 0}$, and for a subset $S \subseteq \mathcal{X} \times \mathcal{I}$ of regions write "$Z \in S$" for the clock constraint expressing that $(Z, I(x)) \in S$. More precisely,

$$Z \in S \;\equiv \bigvee_{(Z,(a_i,a_{i+1}))\in S} a_i < x < a_{i+1} \;\vee \bigvee_{(Z,\{a_i\}))\in S} a_i = x.$$

▶ **Definition 10.** *Let $S \subseteq \mathcal{X} \times \mathcal{I}$ be a set of regions for the 1-TBPP $(\{x\}, \mathcal{X}, \mathcal{R})$. We define a timed automaton $TA_S = (\{x\}, \mathcal{X}, \mathcal{R}_S)$ so that $\mathcal{R}_S$ contains all of the rules in $\mathcal{R}$ with rhs of size 1 and none of the vanishing rules. Moreover, every branching rule $X \xRightarrow{\varphi} Y + Z$ in $\mathcal{R}$ introduces*

- *a rule $X \xRightarrow{\varphi'} Y$ guarded by $\varphi' \stackrel{def}{=} \varphi \wedge Z \in S$, and*
- *a rule $X \xRightarrow{\varphi'} Z$ guarded by $\varphi' \stackrel{def}{=} \varphi \wedge Y \in S$.*

▶ **Theorem 11.** *The reachability problem for 1-clock TBPP is in* NP.

Proof. Suppose that there is a derivation tree witnessing a positive instance of the reachability problem and so that all branches leading to targets have duration τ. We can represent a node by a triple $(A, a, \hat{a}) \in (\mathcal{X} \times \mathbb{R} \times \mathbb{R})$, where (A, a) is a TBPP process and the third component $\hat{a}$ is the total time elapsed so far. Call a node $(A, a, \hat{a})$ *productive* if it lies on a branch from root to some target node. Naturally, every node $(A, a, \hat{a})$ has a unique region $(A, I(a))$ associated with it. For a productive node let us write

$$S(A, a, \hat{a})$$

for the set of regions of nodes which are descendants of $(A, a, \hat{a})$ which are non-productive but have a productive parent. See Figure 2 (left) for an illustration. Observe that

1. The sets $S(A, a, \hat{a})$ can only decrease along a branch from root to a target.
2. If $(A, a, \hat{a})$ has a productive descendant $(C, c, \hat{c})$ such that $S(A, a, \hat{a}) = S(C, c, \hat{c})$, then $(A, a) \xrightarrow{\hat{c}-\hat{a}} (C, c)$ in the timed automaton TA_S.
3. Suppose $(A, a, \hat{a})$ has only one productive child $(C, c, \hat{c})$ and that $S(A, a, \hat{a}) \supset S(C, c, \hat{c})$. Then it must also have another child $(B, b, \hat{b})$ s.t. $Vanish_B(b, \tau - \hat{b})$ holds.

The first two conditions are immediate from the definitions of S and TA_S. To see the third, note that the $S(A, a, \hat{a}) \supset S(C, c, \hat{c})$ implies that $(A, a, \hat{a})$ has some non-productive descendant $(B, b, \hat{b})$ whose region $(B, I(b))$ is not in $S(C, c, \hat{c})$. Since $(C, c, \hat{c})$ is the only productive child, that descendant must already be a child of $(A, a, \hat{a})$. Finally, observe that every non-productive node $(B, b, \hat{b})$ satisfies $Vanish_B(b, \tau - \hat{b})$, as otherwise one of its descendants is present at time τ, and thus must be a target node, contradicting the non-productivity assumption.

The conditions above allow us to use labelled trees of polynomial size as reachability witnesses: These witnesses are labelled trees as above where only as polynomial number of checkpoints along branches from root to target are kept: A checkpoint is either the least common ancestor of two target nodes (in which case a corresponding branching rule must exist), or otherwise it is a triple of nodes as described by condition (3), where a region $(B, I(b))$ is produced for the last time. The remaining paths between checkpoints are positive reachability instances of timed automata TA_S, as in condition (2), where the bottom-most automata TA_S satisfy that if $(U, I) \in S$ then $(z, 0) \in Vanish_U$ for all $z \in I$. Cf. Figure 2 (right). Notice that the existence of a witness of this form is expressible as a polynomially large ∃LA formula thanks to Lemma 9 and Theorem 2.

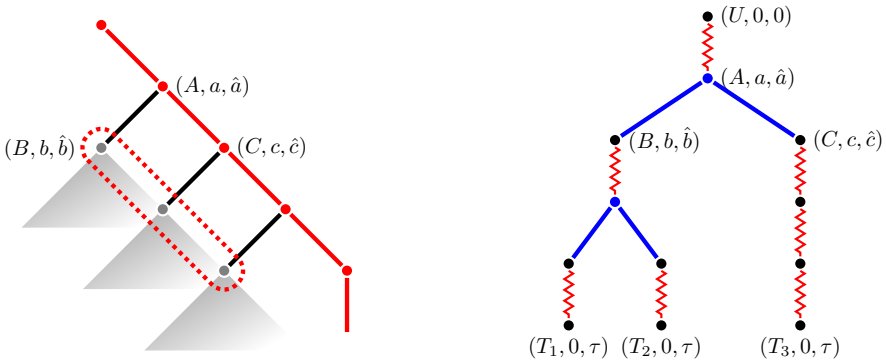

Figure 2 Left: Nodes on the red branch are productive, grey sub-trees are non-productive. $S(A, a, \hat{a})$ contains the regions of nodes in the dotted region. It holds that $S(A, a, \hat{a}) \supseteq S(C, c, \hat{c})$ and the inequality is strict iff $(B, I(b)) \in S(C, c, \hat{c})$. **Right:** small reachability witnesses contain checkpoint where two productive branches split (in blue) or where the allowed side-effects S strictly decrease (red). The intermediate paths are runs of S-restricted TA.

Clearly, every full derivation tree gives rise to a witness of this form. Conversely, assume a witness tree as above exists. One can build a partial derivation tree by unfolding all intermediate TA paths between consecutive checkpoints. It remains to show that whenever some TA_S uses a rule $A \overset{\varphi}{\Longrightarrow} C$ originating from a TBPP rule $A \overset{\varphi}{\Longrightarrow} B + C$ to produce a productive node $(C, c, \hat{c})$ then the node $(B, b, \hat{b})$ produced as side-effect can vanish in time $\tau - \hat{b}$, i.e., we have to show that then $\mathit{Vanish}_B(b, \tau - \hat{b})$.

W.l.o.g. let $\mathit{Vanish}_B(x, t) \equiv t \geq d - xf$ (the case with $>$ is analogous) for some $d \in \mathbb{R}_{\geq 0}$ and $f \in \{0, 1\}$. Observe that the region $(B, I(b))$ is in S by definition of TA_S and that the witness contains a later node $(B, b', \hat{b}')$ with $\mathit{Vanish}_B(b', \tau - \hat{b}')$, and thus

$$\tau - \hat{b}' \geq d - b'f.$$

Notice also that $b' \geq b + \hat{b}' - \hat{b}$ as in the worst-case no reset appears on the path between the parent of $(B, b, \hat{b})$ and $(B, b', \hat{b}')$. Together with the inequality above we derive that $\tau - \hat{b} \geq d - bf$, meaning that indeed $\mathit{Vanish}_B(b, \tau - \hat{b})$ holds, as required. ◀

5 Multi-Clock TBPP

In this section we consider the complexities of coverability and reachability problems for TBPP with multiple clocks. For the upper bounds we will reduce to the reachability problem for TA [7] and to solving reachability games for TA [30].

▶ **Theorem 12.** *The coverability problem for k-TBPP with $k \geq 2$ clocks is* PSPACE*-complete.*

Proof. The lower bound already holds for the reachability problem of 2-clock TA [25] and hence for the *simple* TBPP coverability. For the upper bound, consider an instance where $\mathcal{A} = (\mathcal{C}, \mathcal{X}, \mathcal{R})$ is a k-TBPP and $T_1, \ldots, T_m$ are the target nonterminals. We reduce to the reachability problem for TA $\mathcal{B} = (\mathcal{C}', \mathcal{X}', \mathcal{R}')$ with exponentially many control states $\mathcal{X}'$, but only $|\mathcal{C}'| = O(k \cdot |\mathcal{X}| \cdot m)$ many clocks. The result then follows by the classical region construction of [7], which requires space logarithmic in the number of nonterminals and

polynomial in the number of clocks. The main idea of this construction is to introduce (exponentially many) new nonterminals and rules to simulate the original behaviour on bounded configurations only.

Let $n = m + 2$. We have a clock $x_{X,i} \in \mathcal{C}'$ for every original clock $x \in \mathcal{C}$, nonterminal $X \in \mathcal{X}$, and index $1 \leq i \leq n$, and a nonterminal of the form $[\alpha] \in \mathcal{R}'$ for every multiset $\alpha \in \mathcal{X}^{\oplus}$ of size at most $|\alpha| \leq n$. Since we are solving the coverability problem, we do not need to address vanishing rules $X \stackrel{\varphi;R}{\Longrightarrow} \emptyset$ in $\mathcal{R}$, which are ignored. We will use clock assignments $S_{X,i} \equiv \bigwedge_{i \leq j \leq n-1} x_{X,j} := x_{X,j+1}$ shifting by one position the clocks corresponding to occurrences $j = i, i+1, \ldots, n-1$ of X. We have three families of rules:

1. **(unary rules).** For each rule $X \stackrel{\varphi;R}{\Longrightarrow} Y$ in $\mathcal{R}$ and multiset $\beta \in \mathcal{X}^{\oplus}$ of the form $\beta = \gamma + X + \delta$ of size $|\beta| \leq n$, for some $\gamma, \delta \in \mathcal{X}^{\oplus}$, we have a corresponding rule in $\mathcal{R}'$

$$[\beta] \stackrel{\varphi|_{X,i};\, R|_{X,i;Y,j};\, S_{X,i}}{\longrightarrow} [\gamma + Y + \delta]$$

   for every occurrence $1 \leq i \leq \beta(X)$ of X in β and for $j = \beta(Y) + 1$, where $\varphi|_{X,i}$ is obtained from φ by replacing each clock x with $x_{X,i}$, and $R|_{X,i;Y,j}$ is obtained from R by replacing every assignment $x := y$ by $x_{Y,j} := x_{X,i}$, and $x := 0$ by $x_{Y,j} := 0$.
2. **(branching rules).** Let $X \Longrightarrow Y + Z$ in $\mathcal{R}$ be a branching rule. We assume w.l.o.g. that it has no tests and no assignments, and that X, Y, Z are pairwise distinct. We add rules in $\mathcal{R}'$

$$[\alpha + X] \stackrel{R;S_{X,i}}{\longrightarrow} [\beta], \quad \text{with } \beta = \alpha + Y + Z \text{ and } |\beta| \leq n,$$

   for all $1 \leq i \leq \alpha(X)$ and $\alpha \in \mathcal{X}^{\oplus}$, where $R \equiv \bigwedge_{x \in \mathcal{C}} x_{Y,\beta(Y)} := x_{X,i} \wedge x_{Z,\beta(Z)} := x_{X,i}$ copies each clock $x_{X,i}$ into $x_{Y,\beta(Y)}$ and $x_{Z,\beta(Z)}$, and $S_{X,i}$ was defined earlier.
3. **(shrinking rules).** We also add rules that remove unnecessary nonterminals: For every $\beta = \alpha + X \in \mathcal{X}^{\oplus}$ with $|\beta| \leq n$ and index $1 \leq i \leq \beta(X)$ denoting which occurrence of X in β we want to remove, we have a rule $[\alpha + X] \stackrel{S_i}{\Longrightarrow} [\alpha]$ in $\mathcal{R}'$.

It remains to argue that $(X, \vec{0}) \stackrel{*}{\longrightarrow} (T_1, \vec{0}) + \cdots + (T_n, \vec{0})$ in $\mathcal{A}$ if, and only if, $([X], \vec{0}) \stackrel{*}{\longrightarrow} ([T_1 + \cdots + T_n], \vec{0})$ in $\mathcal{B}$. This can be proven via induction on the depth of the derivation tree, where the induction hypothesis is that every configuration α of size at most n can be covered in with a derivation tree of depth d in $\mathcal{A}$ if, and only if, in the timed automaton $\mathcal{B}$ the configuration $[\alpha]$ can be reached via a path of length at most d. ◀

▶ **Theorem 13.** *The reachability problem for TBPP is* EXPTIME*-complete. Moreover,* EXPTIME*-hardness already holds for k-TBPP emptiness, if 1) $k \geq 2$ is any fixed number of clocks, or 2) k is part of the input but only* 0 *or* 1 *appear as constants in clock constraints.*

In the remainder of this section contains a proof of this result, in three steps: In the first step (Lemma 14) we show an EXPTIME upper bound for the special case of *simple reachability*, i.e., when the target configuration has size 1. As a second step (Lemma 15) we reduce general case to simple reachability and thereby prove the upper bound claimed in Theorem 13. As a third step (Lemma 16), we prove the corresponding lower bound.

▶ **Lemma 14** (Simple reachability)**.** *The simple reachability problem for TBPP is in* EXPTIME*. More precisely, the complexity is exponential in the number of clocks and the maximal clock constant, and polynomial in the number of nonterminals.*

Proof (Sketch). We reduce to TA reachability games, where two players (Min and Max) alternatingly determine a path of a TA, by letting the player who owns the current nonterminal pick time elapse and a valid successor configuration. Min and Max aim to minimize/maximize the time until the play first visits a target nonterminal T. TA reachability games can be solved in EXPTIME, with the precise time complexity claimed above [30, Theorem 5]. The idea of the construction is to let Min produce a derivation tree along the branch that leads to (unique) target process. Whenever she proposes to go along branching rule, Max gets to claim that the other sibling, not on the main branch, cannot be removed until the main branch ends. This can be faithfully implemented by storing only the current configuration on the main branch plus one more configuration (of Max's choosing) that takes the longest time to vanish. Min can develop both independently but must apply time delays to both simultaneously. Min wins the game if she can reach the target nonterminal and before that moment all the other branches have vanished. ◀

▶ **Lemma 15.** *The reachability problem for TBPP is in* EXPTIME.

Proof. First notice that the special case of reachability of the empty target set trivially reduces to the simple reachability problem by adding a dummy nonterminal, which is created once at the beginning and has to be the only one left at the end. Suppose we have an instance of the k-TBPP reachability problem with target nonterminals $T_1, T_2, \ldots, T_m$. We will create an instance of simple reachability where the number of nonterminals increases exponentially but the number of clocks is $O(k \cdot |\mathcal{X}| \cdot m)$. In both cases, the claim follows from Lemma 14.

We introduce a nonterminal $[\beta]$ for every multiset $\beta \in \mathcal{X}^{\oplus}$ of size $|\beta| \leq n := m + 2$, and we have the same three family of rules as in proof of Theorem 12, where the last family 3. is replaced by the family below:

3'. We add extra branching rules in order to maintain nonterminals $[\beta]$ corresponding to small multisets $|\beta| \leq n$. Let $\beta \in \mathcal{X}^{\oplus}$ of size $|\beta| \leq n$ and consider a partitioning $\beta = \beta_1 + \beta_2$, for some $\beta_1, \beta_2 \in \mathcal{X}^{\oplus}$. We identify β with the set $\beta = \{(X, i) \mid X \in \mathcal{X}, 1 \leq i \leq \beta(X)\}$ of pairs (X, i), where i denotes the i-th occurrence of X in β (if any), and similarly for β_1, β_2. We add a branching rule

$$[\beta] \Longrightarrow (\beta, f, \beta_1) + (\beta, f, \beta_2),$$

where (β, f, β_i) are intermediate locations, for every bijection $f : \beta \to \beta_1 \cup \beta_2$ assigning an occurrence of X in β to an occurrence of X either in β_1 or β_2. We then have clock reassigning (non-branching) rules

$$(\beta, f, \beta_1) \overset{S_1}{\Longrightarrow} [\beta_1] \quad \text{and} \quad (\beta, f, \beta_2) \overset{S_2}{\Longrightarrow} [\beta_2],$$

where $S_1 \equiv \bigwedge_{x \in \mathcal{C}} \bigwedge_{X \in \mathcal{X}} \bigwedge_{1 \leq i \leq \beta_1(X)} x_{X,i} := x_{f^{-1}(X,i)}$ and similarly for S_2. ◀

▶ **Lemma 16.** *The non-emptiness problem for TBPP is* EXPTIME*-hard already in discrete time, for 1) TBPP with constants in* $\{0, 1\}$ *(where the number of clocks is part of the input), and 2) for* k*-TBPP for every fixed number of clocks* $k \geq 2$.

6 Conclusion

We introduced basic parallel processes extended with global time and studied the complexities of several natural decision problems, including variants of the coverability and reachability problems. Table 1 summarizes our findings.

The exact complexity status of the simple reachability problem for 1-TBPP is left open. An NP upper bound holds from the (general) reachability problem (by Theorem 11) and PTIME-hardness comes from the emptiness problem for context-free grammars. We conjecture that a matching polynomial-time upper bound holds.

Also left open for future work are *succinct* versions the coverability and reachability problems, where the target size is given in binary. A reduction from subset-sum games [25] shows that the succinct coverability problem for 1-TBPP is PSPACE-hard. This implies that our technique showing the NP-membership for the non-succinct version of the coverability problem (cf. Theorem 7) does not extend to the succinct variant, and new ideas are needed.

Table 1 Results on TBPP and 1-clock TBPP. The decision problems are complete for the stated complexity class. Simple Coverability/Reachability refer to the variants where the target has size 1.

	Emptiness	Simple Coverability	Coverability	Simple Reachability	Reachability
TBPP	EXPTIME [Lem 16], [30]	PSPACE [Thm 12]	PSPACE [Thm 12]	EXPTIME [Thm 14]	EXPTIME [Thm 13]
1-TBPP	PTIME [33]	NL [33]	NP [Thm 7]	PTIME / NP	NP [Thm 11]

References

1 P. A. Abdulla, M. F. Atig, and J. Cederberg. Timed Lossy Channel Systems. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 2012. doi:10.4230/LIPIcs.FSTTCS.2012.374.

2 P. A. Abdulla, M. F. Atig, and J. Stenman. Dense-Timed Pushdown Automata. In *ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2012. doi:10.1109/LICS.2012.15.

3 Parosh Aziz Abdulla, Mohamed Faouzi Atig, Radu Ciobanu, Richard Mayr, and Patrick Totzke. Universal Safety for Timed Petri Nets is PSPACE-complete. In *International Conference on Concurrency Theory (CONCUR)*, 2018. doi:10.4230/LIPIcs.CONCUR.2018.6.

4 Parosh Aziz Abdulla, Mohamed Faouzi Atig, and Shankara Narayanan Krishna. Perfect Timed Communication Is Hard. In *International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, 2018.

5 Parosh Aziz Abdulla and Bengt Jonsson. Model checking of systems with many identical timed processes. *Theoretical Computer Science*, 2003. doi:10.1016/S0304-3975(01)00330-9.

6 Parosh Aziz Abdulla and Aletta Nylén. Timed Petri Nets and BQOs. In *Applications and Theory of Petri Nets and Concurrency (PETRI NETS)*, 2001.

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

8 M. Benerecetti, S. Minopoli, and A. Peron. Analysis of Timed Recursive State Machines. In *International Symposium on Temporal Representation and Reasoning (TIME)*, 2010. doi:10.1109/TIME.2010.10.

9 Massimo Benerecetti and Adriano Peron. Timed recursive state machines: Expressiveness and complexity. *Theoretical Computer Science*, 2016. doi:10.1016/j.tcs.2016.02.021.

10 Beatrice Bérard, Anne Labroue, and Philippe Schnoebelen. Verifying Performance Equivalence for Timed Basic Parallel Processes. In *International Conference on Foundations of Software Science and Computational Structures (FoSSaCS)*, 2000.

11 Bernard Boigelot, Sébastien Jodogne, and Pierre Wolper. An Effective Decision Procedure for Linear Arithmetic over the Integers and Reals. *ACM Trans. Comput. Logic*, 2005. doi:10.1145/1071596.1071601.

12 Ahmed Bouajjani, Rachid Echahed, and Riadh Robbana. On the Automatic Verification of Systems with Continuous Variables and Unbounded Discrete Data Structures. In *Hybrid Systems (HS)*, 1994.

13 Patricia Bouyer, Kim G. Larsen, Nicolas Markey, and Jacob Illum Rasmussen. Almost Optimal Strategies in One Clock Priced Timed Games. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 2006.

14 Søren Christensen. *Decidability and Decomposition in Process Algebras*. PhD thesis, School of Informatics, University of Edinburgh, 1993.

15 Lorenzo Clemente. Decidability of Timed Communicating Automata. *arXiv e-prints*, 2018. arXiv:1804.07815.

16 Lorenzo Clemente, Frédéric Herbreteau, Amelie Stainer, and Grégoire Sutre. Reachability of Communicating Timed Processes. In *International Conference on Foundations of Software Science and Computational Structures (FoSSaCS)*, 2013. doi:10.1007/978-3-642-37075-5_6.

17 Lorenzo Clemente, Piotr Hofman, and Patrick Totzke. Timed Basic Parallel Processes. *arXiv e-prints*, July 2019. arXiv:1907.01240.

18 Lorenzo Clemente and Sławomir Lasota. Binary reachability of timed pushdown automata via quantifier elimination. In *International Colloquium on Automata, Languages and Programming (ICALP)*, 2018. doi:10.4230/LIPIcs.ICALP.2018.118.

19 Lorenzo Clemente, Sławomir Lasota, Ranko Lazić, and Filip Mazowiecki. Timed pushdown automata and branching vector addition systems. In *ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2017.

20 Hubert Comon and Yan Jurski. Timed Automata and the Theory of Real Numbers. In *International Conference on Concurrency Theory (CONCUR)*, 1999.

21 Zhe Dang. Binary Reachability Analysis of Pushdown Timed Automata with Dense Clocks. In *Computer Aided Verification (CAV)*, 2001.

22 C. Dima. Computing reachability relations in timed automata. In *ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2002. doi:10.1109/LICS.2002.1029827.

23 Catalin Dima. A Class of Automata for Computing Reachability Relations in Timed Systems. In *Verification of Infinite State Systems with Applications to Security (VISSAS)*, 2005.

24 Javier Esparza. Petri Nets, Commutative Context-Free Grammars, and Basic Parallel Processes. *Fundamenta Informaticae*, 1997.

25 John Fearnley and Marcin Jurdziński. Reachability in two-clock timed automata is PSPACE-complete. *Information and Computation*, 2015. doi:10.1016/j.ic.2014.12.004.

26 Jeanne Ferrante and Charles Rackoff. A Decision Procedure for the First Order Theory of Real Addition with Order. *SIAM Journal on Computing*, 1975.

27 Martin Fränzle, Karin Quaas, Mahsa Shirmohammadi, and James Worrell. Effective Definability of the Reachability Relation in Timed Automata. *arXiv e-prints*, 2019. arXiv:1903.09773.

28 Serge Haddad, Sylvain Schmitz, and Philippe Schnoebelen. The Ordinal Recursive Complexity of Timed-Arc Petri Nets, Data Nets, and Other Enriched Nets. In *ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2012. doi:10.1109/LICS.2012.46.

29 Thomas Dueholm Hansen, Rasmus Ibsen-Jensen, and Peter Bro Miltersen. A Faster Algorithm for Solving One-Clock Priced Timed Games. In *International Conference on Concurrency Theory (CONCUR)*, 2013.

30 Marcin Jurdziński and Ashutosh Trivedi. Reachability-time Games on Timed Automata. In *International Colloquium on Automata, Languages and Programming (ICALP)*, 2007.

31 Pavel Krčál and Radek Pelánek. On Sampled Semantics of Timed Systems. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 2005.

32 Pavel Krcal and Wang Yi. Communicating timed automata: the more synchronous, the more difficult to verify. In *Computer Aided Verification (CAV)*, 2006. doi:10.1007/11817963_24.

33 F. Laroussinie, N. Markey, and Ph. Schnoebelen. Model Checking Timed Automata with One or Two Clocks. In *International Conference on Concurrency Theory (CONCUR)*, 2004.

34 Kim G. Larsen, Paul Pettersson, and Wang Yi. Uppaal in a Nutshell. *International Journal on Software Tools for Technology Transfer (STTT)*, 1997. doi:10.1007/s100090050010.

35 Slawomir Lasota and Igor Walukiewicz. Alternating timed automata. *ACM Trans. Comput. Logic*, 2008. doi:10.1145/1342991.1342994.

36 Philip Meir Merlin. *A Study of the Recoverability of Computing Systems*. PhD thesis, University of California, Irvine, 1974.

37 J. Ouaknine and J. Worrell. On the decidability of metric temporal logic. In *ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2005. doi:10.1109/LICS.2005.33.

38 Louchka Popova. On Time Petri Nets. *Journal of Information Processing and Cybernetics*, 1991.

39 Mojżesz Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. *Comptes Rendus Premier Congrès des Mathématicienes des Pays Slaves*, 1930.

40 Karin Quaas. Verification for Timed Automata extended with Unbounded Discrete Data Structures. *Logical Methods in Computer Science*, 2015. doi:10.2168/LMCS-11(3:20)2015.

41 Valentin Valero Ruiz, Fernando Cuartero Gomez, and David de Frutos Escrig. On Non-Decidability of Reachability for Timed-Arc Petri Nets. In *International Workshop on Petri Nets and Performance Models (PNPM)*, 1999.

42 Ashutosh Trivedi. *Competative Optimisation on Timed Automata*. PhD thesis, University of Warwick, 2009.

43 Ashutosh Trivedi and Dominik Wojtczak. Recursive Timed Automata. In *International Symposium on Automated Technology for Verification and Analysis (ATVA)*, 2010.

44 Kumar Neeraj Verma, Helmut Seidl, and Thomas Schwentick. On the Complexity of Equational Horn Clauses. In *International Conference on Automated Deduction (CADE)*, 2005. doi:10.1007/11532231_25.

45 Volker Weispfenning. Mixed Real-integer Linear Quantifier Elimination. In *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, 1999. doi:10.1145/309831.309888.

46 Sergio Yovine. KRONOS: a verification tool for real-time systems. *International Journal on Software Tools for Technology Transfer*, 1997. doi:10.1007/s100090050009.