Parametric updates in parametric timed automata

Étienne André
Université Paris 13, LIPN, CNRS, UMR 7030, F-93430, Villetaneuse, France

Didier Lime
École Centrale de Nantes, LS2N, CNRS, UMR 6004, France

Mathias Ramparison
Université Paris 13, LIPN, CNRS, UMR 7030, F-93430, Villetaneuse, France

Abstract

We introduce a new class of Parametric Timed Automata (PTAs) where we allow clocks to be compared to parameters in guards, as in classic PTAs, but also to be updated to parameters. We focus here on the EF-emptiness problem: "is the set of parameter valuations for which some given location is reachable in the instantiated timed automaton empty?". This problem is well-known to be undecidable for PTAs, and so it is for our extension. Nonetheless, if we update all clocks each time we compare a clock with a parameter and each time we update a clock to a parameter, we obtain a syntactic subclass for which we can decide the EF-emptiness problem and even perform the exact synthesis of the set of rational valuations such that a given location is reachable. To the best of our knowledge, this is the first non-trivial subclass of PTAs, actually even extended with parametric updates, for which this is possible.

2012 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases parametric timed model checking, reachability emptiness, decidability

Digital Object Identifier $d_{0,i}$10.4230/LIPIcs...

Funding This work is partially supported by the ANR national research program "PACS" (ANR-14-CE28-0002).

Acknowledgements We would like to thank anonymous reviewers for constructive remarks on a preliminary version of this manuscript.

1 Introduction

Timed automata (TAs) are a powerful formalism to model and verify timed concurrent systems, both expressive enough to model many interesting systems and enjoying several decidability properties. In particular, the reachability of a discrete state is PSPACE-complete [AD94]. In TAs, clocks can be compared with constants in guards, and can be updated to 0 along edges.

Timed automata may turn insufficient to verify systems where the timing constants themselves are subject to some uncertainty, or when they are simply not known at the early design stage. Parametric timed automata (PTAs) [AHV93] address this drawback by allowing parameters (unknown constants) in the timing constraints; this high expressive power comes at the cost of the undecidability of most interesting problems. In particular, the basic problem of EF-emptiness ("is the set of valuations for which a given location is reachable in the instantiated timed automaton empty?") is "robustly" undecidable: even for a single rational-valued [Mil00] or integer-valued parameter [AHV93, BBLS15], or when only strict constraints are used [Doy07]. A famous syntactic subclass of PTAs that enjoys limited decidability is

L/U-PTAs [HRSV02], where the parameters set is partitioned into lower-bound and upper-bound parameters, *i. e.,* parameters that can only be compared to a clock as a lower-bond (resp. upper-bound). The EF-emptiness problem is decidable for L/U-PTAs [HRSV02, BL09]; however, most other problems are undecidable (*e. g.,* [BL09, JLR15, ALR16, AL17]).

Contributions. We investigate parametric updates, by showing that the EF-emptiness problem is decidable for PTAs augmented with parametric updates (*i. e.,* U2P-PTA), with the additional condition that whenever a clock is compared to a parameter in a guard, all clocks must be updated (possibly to parameters)—this gives R-U2P-PTA. This result holds when the parameters are bounded rationals in guards, and possibly unbounded rationals in updates. Non-trivial decidable subclasses of PTAs are a rarity (to the best of our knowledge, only L/U-PTAs [HRSV02] and IP-PTAs [ALR16]); this makes our positive result very welcome. In addition, not only the emptiness is decidable, but exact synthesis for bounded rational-valued parameters can be performed—which contrasts with L/U-PTAs and IP-PTAs as synthesis was shown intractable [JLR15, ALR16].

All proofs (or proof sketches) are available in the appendix. In addition, a full version of this paper with all detailed proofs is available at [ALR18a].

Related work. Allowing parameters in clock updates is inspired by the updatable TA formalism defined in [BDFP04] where clocks can be updated not only to 0 ("reset") but also to rational constants ("update"). In [ALR18b], we extended the result of [BDFP04] by allowing parametric updates (and no parameter elsewhere, *e. g.,* in guards): the EF-emptiness is undecidable even in the restricted setting of bounded rational-valued parameters, but becomes decidable when parameters are restricted to (unbounded) integers.

Synthesis is obviously harder than EF-emptiness: only three results have been proposed to synthesize the exact set of valuations for subclasses of PTAs, but they are all concerned with *integer*-valued parameters [BL09, JLR15, ALR18b]. In contrast, we deal here with (bounded) rational-valued parameters—which makes this result the first of its kind. The idea of updating all clocks when compared to parameters comes from our class of *reset-PTAs* briefly mentioned in [ALR16], but not thoroughly studied. Finally, updating clocks on each transition in which a parameter appears is reminiscent of the initialized rectangular hybrid automata [HKPV98], which remains one of the few decidable subclasses of hybrid automata.

2 Preliminaries

Throughout this paper, we assume a set $\mathbb{X} = \{x_1, \dots, x_H\}$ of *clocks*, *i. e.,* real-valued variables evolving at the same rate. A clock valuation is $w : \mathbb{X} \to \mathbb{R}_+$. We write $\vec{0}$ for the clock valuation that assigns 0 to all clocks. Given $d \in \mathbb{R}_+$, $w + d$ (resp. $w - d$) denotes the valuation such that $(w + d)(x) = w(x) + d$ (resp. $(w - d)(x) = w(x) - d$ if $w(x) - d > 0$, 0 otherwise), for all $x \in \mathbb{X}$. We assume a set $\mathbb{P} = \{p_1, \dots, p_M\}$ of *parameters*, *i. e.,* unknown constants. A parameter *valuation* v is a function $v : \mathbb{P} \to \mathbb{Q}_+$. We identify a valuation v with the point $(v(p_1), \dots, v(p_M))$ of $\mathbb{Q}_+^M$. Given $d \in \mathbb{N}$, $v + d$ (resp. $v - d$) denotes the valuation such that $(v + d)(p) = v(p) + d$ (resp. $(v - d)(p) = v(p) - d$ if $v(p) - d > 0$, 0 otherwise), for all $p \in \mathbb{P}$.

In the following, we assume $\triangleleft \in \{<, \leq\}$ and $\bowtie \in \{<, \leq, \geq, >\}$.

A *parametric guard* g is a constraint over $\mathbb{X} \cup \mathbb{P}$ defined as the conjunction of inequalities of the form $x \bowtie z$, where x is a clock and z is either a parameter or a constant in $\mathbb{Z}$. A *non-parametric guard* is a parametric guard without parameters (*i. e.,* over $\mathbb{X}$).

Given a parameter valuation v, $v(g)$ denotes the constraint over $\mathbb{X}$ obtained by replacing in g each parameter p with $v(p)$. We extend this notation to an *expression*: a sum or difference of parameters and constants. Likewise, given a clock valuation w, $w(v(g))$ denotes the expression obtained by replacing in $v(g)$ each clock x with $w(x)$. A clock valuation w *satisfies* constraint $v(g)$ (denoted by $w \models v(g)$) if $w(v(g))$ evaluates to true. We say that v *satisfies* g, denoted by $v \models g$, if the set of clock valuations satisfying $v(g)$ is nonempty. We say that g is *satisfiable* if $\exists w, v$ s.t. $w \models v(g)$.

A *parametric update* is a partial function $u : \mathbb{X} \rightharpoonup \mathbb{N} \cup \mathbb{P}$ which assigns to some of the clocks an integer constant or a parameter. For v a parameter valuation, we define a partial function $v(u) : \mathbb{X} \rightharpoonup \mathbb{Q}_+$ as follows: for each clock $x \in \mathbb{X}$, $v(u)(x) = k \in \mathbb{N}$ if $u(x) = k$ and $v(u)(x) = v(p) \in \mathbb{Q}_+$ if $u(x) = p$ a parameter. A non-parametric update is $u_{np} : \mathbb{X} \rightharpoonup \mathbb{N}$. For a clock valuation w and a parameter valuation v, we denote by $[w]_{v(u)}$ the clock valuation obtained after applying $v(u)$.We first define a new class of parametric timed automata in order to properly define further plain parametric timed automata and timed automata.

▶ **Definition 1.** An update-to-parameter PTA (U2P-PTA) $\mathcal{A}$ is a tuple $\mathcal{A} = (\Sigma, L, l_0, \mathbb{X}, \mathbb{P}, \zeta)$, where: *i)* Σ is a finite set of actions, *ii)* L is a finite set of locations, *iii)* $l_0 \in L$ is the initial location, *iv)* $\mathbb{X}$ is a finite set of clocks, *v)* $\mathbb{P}$ is a finite set of parameters, *vi)* ζ is a finite set of edges $e = \langle l, g, a, u, l' \rangle$ where $l, l' \in L$ are the source and target locations, g is a parametric guard, $a \in \Sigma$ and $u : \mathbb{X} \rightharpoonup \mathbb{N} \cup \mathbb{P}$ is a parametric update function.

Given a parameter valuation v, we denote by $v(\mathcal{A})$ the structure where all occurrences of a parameter p_i have been replaced by $v(p_i)$. If $v(\mathcal{A})$ is such that all constants in guards and updates are rationals, then $v(\mathcal{A})$ is a *updatable timed automaton* [BDFP04] but will be called *timed automaton* (TA) for the sake of simplicity in this paper.

A *bounded* U2P-PTA is a U2P-PTA with a bounded parameter domain that assigns to each parameter a minimum integer bound and a maximum integer bound. That is, each parameter p_i ranges in an interval $[a_i, b_i]$, with $a_i, b_i \in \mathbb{N}$. Hence, a bounded parameter domain is a hyperrectangle of dimension M.

A parametric timed automaton (PTA) [AHV93] is a U2P-PTA where, for any edge $e = \langle l, g, a, u, l' \rangle \in \zeta$, $u : \mathbb{X} \rightharpoonup \{0\}$.

▶ **Definition 2** (Concrete semantics of a TA)**.** Given a U2P-PTA $\mathcal{A} = (\Sigma, L, l_0, \mathbb{X}, \mathbb{P}, \zeta)$, and a parameter valuation v, the concrete semantics of $v(\mathcal{A})$ is given by the timed transition system $(S, s_0, \rightarrow)$, with

- $S = \{(l, w) \in L \times \mathbb{R}_+^H\}$, $s_0 = (l_0, \vec{0})$
- $\rightarrow$ consists of the discrete and (continuous) delay transition relations:
  - discrete transitions: $(l, w) \overset{e}{\mapsto} (l', w')$, if $(l, w), (l', w') \in S$, there exists $e = \langle l, g, a, u, l' \rangle \in \zeta$, $w' = [w]_{v(u)}$, and $w \models v(g)$.
  - delay transitions: $(l, w) \overset{d}{\mapsto} (l, w + d)$, with $d \in \mathbb{R}_+$.

Moreover we write $(l, w) \xrightarrow{e} (l', w')$ for a combination of a delay and discrete transitions where $((l, w), e, (l', w')) \in \rightarrow$ if $\exists d, w'' : (l, w) \overset{d}{\mapsto} (l, w'') \overset{e}{\mapsto} (l', w')$.

Given a TA $v(\mathcal{A})$ with concrete semantics $(S, s_0, \rightarrow)$, we refer to the states of S as the *concrete states* of $v(\mathcal{A})$. A (concrete) *run* of $v(\mathcal{A})$ is a possibly infinite alternating sequence of concrete states of $v(\mathcal{A})$ and edges starting from s_0 of the form $s_0 \xrightarrow{e_0} s_1 \xrightarrow{e_1} \cdots \xrightarrow{e_{m-1}} s_m \xrightarrow{e_m} \cdots$, such that for all $i = 0, 1, \ldots$, $e_i \in \zeta$, and $(s_i, e_i, s_{i+1}) \in \rightarrow$. Given a state $s = (l, w)$, we say that s is reachable (or that $v(\mathcal{A})$ reaches s) if s belongs to a run of $v(\mathcal{A})$. By extension, we say that l is reachable in $v(\mathcal{A})$, if there exists a state (l, w) that is reachable.

Throughout this paper, let K denote the largest constant in a given U2P-PTA, *i. e.,* the maximum between the largest constant compared to a clock in a guard and the largest bound of a parameter (if the U2P-PTA is bounded).

Let us recall the notion of clock region [AD94]. Given a clock x and a clock valuation w, recall that $\lfloor w(x) \rfloor$ denotes the integer part of $w(x)$ while $frac(w(x))$ denotes its fractional part. We define the same notation for parameter valuations.

▶ **Definition 3** (clock region)**.** For two clock valuations w and w', $\sim$ is an equivalence relation defined by: $w \sim w'$ iff

1. for all clock x, either $\lfloor w(x) \rfloor = \lfloor w'(x) \rfloor$ or $w(x), w'(x) > K$;
2. for all clocks x, y with $w(x), w(y) \leq K$, $frac(w(x)) \leq frac(w(y))$ iff $frac(w'(x)) \leq frac(w'(y))$;
3. for all clock x with $w(x) \leq K$, $frac(w(x)) = 0$ iff $frac(w'(x)) = 0$.

A *clock region* R_c is an equivalence class of $\sim$.

Two clock valuations in the same clock region reach the same regions by time elapsing, satisfy the same guards and can take thus the same transitions [AD94].

In this paper, we address the *EF-emptiness* problem: **given a U2P-PTA $\mathcal{A}$ and a location l, is the set of valuations v such that l is reachable in $v(\mathcal{A})$ empty?**

3 A decidable subclass of U2P-PTAs

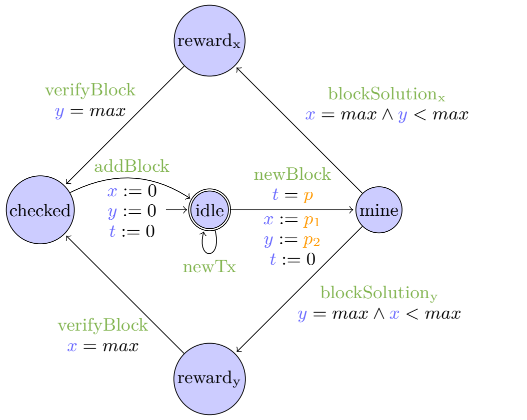
We now impose that, whenever a guard or an update along an edge contains parameters, then all clocks must be updated (to constants or parameters) and prove that it makes EF-emptiness decidable.

▶ **Definition 4.** An *R-U2P-PTA* is a U2P-PTA where for any edge $e = \langle l, g, a, u, l' \rangle$, u is a total function whenever:[1]

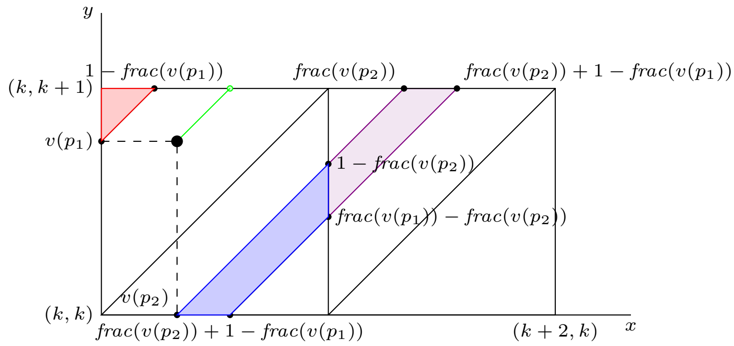
1. g is a parametric guard, or
2. $u(x) \in \mathbb{P}$ for some $x \in \mathbb{X}$,

▶ **Example 5.** Consider the R-U2P-PTA in Figure 1a with five locations, three parametric clocks (x, y, t), one constant (max) and two parameters (p_1, p_2). As a motivating toy example, consider the case of a network of peers exchanging transactions grouped by blocks, *e. g.,* a blockchain, using the Proof-of-Work as a mean to validate new blocks to add. In this simplified example, we consider a set of two peers (represented by x, y) which have different computation power (represented by p_1, p_2). We ask: *"what are the possible computation power configurations so that there is an execution s.t. x is eventually rewarded"* ($EF(\mathrm{reward}_{\mathrm{x}})$-synthesis). A peer can write a new transaction on the current block (newTx). If it is full ($t = p$), it will try to add a new block (newBlock) to write the transaction on it. We update x to p_1, y to p_2, and t to 0 as the peers have a different power computation, and they start "mining" the block (find a solution to a computation problem). Either x or y will eventually find the solution ($\mathrm{blockSolution}_{\mathrm{x}}$ if $x = max$ or $\mathrm{blockSolution}_{\mathrm{y}}$ if $y = max$) and be rewarded. Finally, the other peer checks for the solution (verifyBlock when it reaches max) and the block is added to the blockchain (addBlock).

[1] In the following we only consider either non-parametric, or (necessarily total) fully parametric update functions. A total update function which is not fully parametric (*i. e.,* an update of some clocks to parameters and all others to constants) can be encoded as a total fully parametric update immediately followed by a (partial) non-parametric update function.



(a) A peer modeled with a bounded R-U2P-PTA



(b) A point–p–PDBM (black dot) and several open–p–PDBMs resulting of time elapsing in green, update of y to k in red, update of x to k in blue and time elapsing in purple, with $k \in \mathbb{N}$.

Figure 1 A motivating example of bounded R-U2P-PTA, and a representation of p–PDBMs

The main idea for proving decidability is the following: as in [AD94], given an R-U2P-PTA $\mathcal{A}$ we want to construct a finite region automaton that bisimulates $\mathcal{A}$. For this purpose, we construct regions for clocks and parameters, of which we will show there is a finite number. Since parameters are allowed in guards, we need to construct parameter regions and more restricted clock regions. We will define a form of Parametric Difference Bound Matrices (viz., p–PDBMs for precise PDBMs, inspired by [HRSV02]) in which, once valuated by a parameter valuation, two clock valuations have the same discrete behavior and satisfy the same non-parametric guards. A key point is that in our p–PDBMs the parametric constraints used in the matrix will belong to a *finite* set of predefined expressions involving parameters and constants. We define this set as follows: $\mathcal{PLT} = \{frac(p_i), 1 - frac(p_i), frac(p_i) - frac(p_j), frac(p_j) + 1 - frac(p_i), 1, 0, frac(p_i) - 1 - frac(p_j), -frac(p_i), frac(p_i) - 1\}$, for all $1 \leq i, j \leq M$.

Let us now define an equivalence relation between parameter valuations.

▶ **Definition 6** (regions of parameters)**.** We write that $v \frown v'$ if

1. for all parameter p, $\lfloor v(p) \rfloor = \lfloor v'(p) \rfloor$;
2. for all $d_1, d_2, d_3 \in \mathcal{PLT}$, $v(d_1) \leq v(d_2) + v(d_3)$ iff $v'(d_1) \leq v'(d_2) + v'(d_3)$;

Parameter regions are defined as the equivalence classes of $\frown$, and we will use the notation R_p for parameter regions. The set of all *parameter regions* is denoted by $\mathcal{R}_p$. The definition is in a way similar to Definition 3 but also involves comparisons of sums of elements of $\mathcal{PLT}$. In fact, we will need this kind of comparisons to define our p–PDBMs. Nonetheless we do not need more complicated comparisons as in R-U2P-PTA whenever a parametric guard or update is met the update is a total function: this preserves us from the parameter accumulation, *e. g.,* obtaining expressions of the form $5frac(p_i) - 1 - 3frac(p_j)$ (that may occur in usual PTAs).

In the following, our p–PDBMs will contain pairs of the form $D = (d, \triangleleft)$, where $d \in \mathcal{PLT}$. We therefore need to define comparisons on these pairs.

We define an associative and commutative operator $\oplus$ as $\triangleleft_1 \oplus \triangleleft_2 = <$ if $\triangleleft_1 \neq \triangleleft_2$, or $\triangleleft_1$ if $\triangleleft_1 = \triangleleft_2$. We define $D_1 + D_2 = (d_1 + d_2, \triangleleft_1 \oplus \triangleleft_2)$. Following the idea of parameter regions, we define the *validity* of a comparison between pairs of the form $(d_i, \triangleleft_i)$ within a given parameter region, *i. e.,* whether the comparison is true for all v in the parameter region.

▶ **Definition 7** (validity of comparison)**.** Let R_p be a parameter region. Given any two linear terms d_1, d_2 over $\mathbb{P}$ (*i. e.,* of the form $\sum_i \alpha_i p_i + d$ with $\alpha_i, d \in \mathbb{Z}$) not necessarily in $\mathcal{PLT}$, the comparison $(d_1, \triangleleft_1) \triangleleft (d_2, \triangleleft_2)$ is *valid for* R_p if:

1. $\triangleleft = <$, and either
   a. for all $v \in R_p$, $v(d_1) < v(d_2)$ evaluates to true regardless of $\triangleleft_1, \triangleleft_2$, or
   b. for all $v \in R_p$, $v(d_1) \leq v(d_2)$ evaluates to true, $\triangleleft_1 = <$ and $\triangleleft_2 = \leq$;

2. $\triangleleft = \leq$, and either
   a. for all $v \in R_p$, $v(d_1) < v(d_2)$ evaluates to true regardless of $\triangleleft_1, \triangleleft_2$, or
   b. for all $v \in R_p$, $v(d_1) \leq v(d_2)$ evaluates to true, and $\triangleleft_1 = \triangleleft_2$, or $\triangleleft_1 = <$;

Transitivity is immediate from the definition: if $D_1 \triangleleft_1 D_2$ and $D_2 \triangleleft_2 D_3$ are valid for R_p, $D_1(\triangleleft_1 \oplus \triangleleft_2)D_3$ is valid for R_p.

We can now define our data structure, namely p–PDBMs (for *precise Parametric Difference Bound Matrices*), inspired by the PDBMs of [HRSV02]; PDBMs were themselves inspired by DBMs [Dil89]. However, our p–PDBM compare differences of *fractional parts* of clocks, instead of clocks as in classical DBMs; therefore, our p–PDBMs are closer to *clock regions* of [AD94] than to DBMs. A p–PDBM is a pair made of an integer vector (encoding the clocks integer part), and a matrix (encoding the parametric differences between any two clock fractional parts). Their interpretation also follows that of PDBMs and DBMs: for $i \neq 0$, the matrix cell $D_{i,0} = (d_{i,0}, \triangleleft_{i0})$ is interpreted as the constraint $\mathit{frac}(x_i) \triangleleft_{i0} d_{i,0}$, and $D_{0,i} = (d_{0,i}, \triangleleft_{0i})$ as the constraint $-d_{0,i} \triangleleft_{0i} \mathit{frac}(x_i)$. For $i \neq 0$ and $j \neq 0$, the matrix cell $D_{i,j} = (d_{i,j}, \triangleleft_{ij})$ is interpreted as $\mathit{frac}(x_i) - \mathit{frac}(x_j) \triangleleft_{ij} d_{i,j}$. Finally for all i, $D_{i,i} = (0, \leq)$.

Our p–PDBMs are partitioned into two types: open–p–PDBMs and point–p–PDBMs. A point–p–PDBM is a clock region defined by only parameters which contains only one clock valuation; that is, it corresponds to a set of inequalities of the form $x_i = p_j$. In contrast, an open–p–PDBM is a clock region which can contain several clock valuations satisfying some possibly parametric constraints, or contain at least one clock valuation satisfying non-parametric constraints (as the corner-point of [AD94]). In particular, the initial clock region $\{0^H\}$ and any clock region $\{E_i^H\}$ where E_i is an integer vector for all clock x_i, is an open–p–PDBM.

Basically, only the first p–PDBM after a (necessarily total) parametric clock update will be a point–p–PDBM; any following p–PDBM will be an open–p–PDBM until the next (total) parametric update.

▶ **Definition 8** (open–p–PDBM)**.** Let R_p be a parameter region. An open–p–PDBM for R_p is a pair (E, D) with $E = (E_1, \cdots, E_H)$ a vector of H integers (or ∞ when it exceeds a possible upper-bound) which is the integer part of each clock, and D is an $(H+1)^2$ matrix where each element $D_{i,j}$ is a pair $(d_{i,j}, \triangleleft_{ij})$ for all $0 \leq i, j \leq H$, where $d_{i,j} \in \mathcal{PLT}$. Moreover, for all $0 \leq i \leq H$, $D_{i,i} = (0, \leq)$. In addition:

1. For all i, $(-1, <) \leq D_{0,i} \leq (0, \leq)$ and $(0, \leq) \leq D_{i,0} \leq (1, <)$ are valid for R_p,
2. For all $i \neq 0, j \neq 0$, either $(0, \leq) \leq D_{i,j} \leq (1, <)$ is valid for R_p and $(-1, <) \leq D_{j,i} \leq (0, \leq)$ is valid for R_p or $(0, \leq) \leq D_{j,i} \leq (1, <)$ is valid for R_p and $(-1, <) \leq D_{i,j} \leq (0, \leq)$ is valid for R_p.
3. For all i, j, if $d_{i,j} = -d_{j,i}$ and is different from 1 then $\triangleleft_{ij} = \triangleleft_{ji} = \leq$, else $\triangleleft_{ij} = \triangleleft_{ji} = <$,
4. For all i, j, k, $D_{i,j} \leq D_{i,k} + D_{k,j}$ is valid for R_p (canonical form), and
5. **a.** *(Border* open–p*–PDBM)* there is at least one i s.t. $D_{i,0} = D_{0,i} = (0, \leq)$, or
   b. *(Center* open–p*–PDBM)* there is at least one i s.t. $D_{i,0} = (1, <)$ and for all j s.t. $D_{0,j} = (0, \triangleleft_{oj})$, then we have $\triangleleft_{oj} = <$.

An open–p–PDBM satisfying condition 5a can be seen as a subregion of an open line segment or a corner point region of [AD94, fig. 9 example 4.4] and one satisfying condition 5b can be seen as a subregion of an open region of [AD94, fig. 9 example 4.4]. Remark that sets of the form $\{\mathit{frac}(w(x)) \mid 0 \leq \mathit{frac}(w(x)) \leq 1\}$ are forbidden by Definition 8 (3), as in the regions of [AD94].

Let R_p be a parameter region. In the following, $p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ is the set of all possible open–p–PDBMs (E, D) for R_p.

The second type is the point–p–PDBM. It represents the unique clock valuation (for a given parameter valuation) obtained after a total parametric update in an U2P-PTA.

▶ **Definition 9** (point–p–PDBM). Let R_p be a parameter region. A point–p–PDBM for R_p is a pair (E, D) where D is an $(H+1)^2$ matrix where each element $D_{i,j}$ is a pair $(d_{i,j}, \leq)$ and for all $0 \leq i, j \leq H$, $d_{i,0} = \mathit{frac}(p_1) = -d_{0,i}$, and $d_{i,j} = \mathit{frac}(p_1) - \mathit{frac}(p_2) = -d_{j,i}$, for any $p_1, p_2 \in \mathbb{P}$. and for all $1 \leq i \leq H$, $E_i = \lfloor p_k \rfloor$ if $d_{i,0} = \mathit{frac}(p_k)$, for $1 \leq k \leq M$. In addition:

1. For all i, $(-1, <) \leq D_{0,i} \leq (0, \leq)$ and $(0, \leq) \leq D_{i,0} \leq (1, <)$ are valid for R_p,
2. For all i, j, k, $D_{i,j} \leq D_{i,k} + D_{k,j}$ is valid for R_p (canonical form).

The fact that D is antisymmetric *i. e.,* for all i, j, $D_{i,j} = -D_{j,i}$, means that each clock is valuated to a parameter and each difference of clocks is valuated to a difference of parameters.

The set of all point–p–PDBM for R_p is denoted by $p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, and the set of all p–PDBMs for R_p by $p\text{–}\mathcal{PDBM}(R_p)$ (hence $p\text{–}\mathcal{PDBM}(R_p) = p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p) \cup p\text{–}\mathcal{PDBM}_{\odot}(R_p)$).

Given a p–PDBM (E, D) and a parameter valuation v, we denote by $(E, v(D))$ the *valuated* p*–PDBM*, *i. e.,* the set of clock valuations defined by:

$$\bigwedge_{i,j \in [0,H]} \mathit{frac}(x_i) - \mathit{frac}(x_j) \triangleleft_{i,j} v(d_{i,j}) \wedge \bigwedge_{i \in [1,H]} \lfloor x_i \rfloor = E_i.$$

For a clock valuation w, we write $w \in (E, v(D))$ if it satisfies all constraints of $(E, v(D))$.

In the following subsections Sections 3.1.1 to 3.1.4, we are going to define operations on p–PDBMs (*i. e.,* update of clocks, time elapsing and guards satisfaction), and will show that the set of p–PDBMs is stable under them. But let us first clarify our needs graphically.

▶ **Example 10.** Let v be a parameter valuation. We assume $\lfloor v(p_2) \rfloor = \lfloor v(p_1) \rfloor = k$ and $\mathit{frac}(v(p_1)) > \mathit{frac}(v(p_2))$. In Figure 1b, the black lines represent clock regions as defined in [AD94]. The p–PDBM represented as a big dot is obtained after an update $u(x) = v(p_2)$ and $u(y) = v(p_1)$. It is a point–p–PDBM. The green p–PDBM is obtained by time elapsing from the black point–p–PDBM. Here the blue p–PDBM is obtained after an update $u(y) = k$ where $k \in \mathbb{N}$, *i. e.,* $\mathit{frac}(y) = 0$, while in the green p–PDBM, before it reaches its upper bound 1 and time elapsing. The purple one is obtained after letting time elapse from the blue one. The red p–PDBM is obtained after an update $u(x) = k$ where $k \in \mathbb{N}$, *i. e.,* $\mathit{frac}(x) = 0$ while in the green p–PDBM, with the same conditions, and time elapsing.

3.1 Operations on p–PDBMs

3.1.1 Non-parametric update

To apply a non-parametric update on a p–PDBM, following classical algorithms for DBMs [BY03], we define an update operator, given in Algorithm 1.

Algorithm 1: *update*(D, u_{np}): for all clock x_i where u_{np} is defined, update $\mathit{frac}(x_i) := 0$

```
foreach x_i where u_np(x_i) is defined do
    D_{i,0} := D_{0,i} = (0, ≤)
    for j from 1 to H do
        D_{i,j} = D_{0,j}
        D_{j,i} = D_{j,0}
    end
end
```

▶ **Definition 11** (update of a p–PDBM). Let u_{np} be a non-parametric update function. Given $(E, D) \in$

p–$\mathcal{PDBM}(R_p)$, we define the *update* of (E, D), denoted by $(E', D') = \mathit{update}((E, D), u_{np})$ as: D' is the result of Algorithm 1 and for each clock x if $u_{np}(x)$ is defined $E'_x := u_{np}(x)$, $E'_x := E_x$ otherwise.

▶ **Lemma 12** (stability under update). *Let R_p be a parameter region and $(E, D) \in p$–$\mathcal{PDBM}(R_p)$. Let u_{np} be a non-trivial non-parametric update. Then $\mathit{update}((E, D), u_{np}) \in p$–$\mathcal{PDBM}_{\blacksquare}(R_p)$.*

Proof idea. Intuitively, we update in (E, D) the lower and upper bounds of some clocks to $(0, \leq)$ and the difference between two clocks $D_{i,j}$ to $D_{0,j}$ if x_i is updated: that is, the new difference between two clocks if one has been updated is just the lower/upper bound of the one that is not updated. This allows us to conserve the canonical form as we only "moved" some cells in D that already verified the canonical form. See Appendix .1 for details. ◀

Applying a non-parametric *update* on any point–p–PDBM transforms it into an open–p–PDBM, and open–p–PDBMs are stable under *update*. It can seem a paradox that the (non-parametric) update of a point–p–PDBM becomes an open–p–PDBM; in fact, it remains geometrically speaking a point, *i. e.,* a singleton containing one clock valuation. Recall that our open–p–PDBMs include p–PDBMs geometrically corresponding to a point for each valuation. In contrast, point–p–PDBMs are also punctual (for each valuation), but are fully parametric.

The following lemma states that the update operator behaves as expected.

▶ **Lemma 13** (semantic of *update* on p–$\mathcal{PDBM}(R_p)$). *Let R_p be a parameter region and $(E, D) \in p$–$\mathcal{PDBM}(R_p)$. Let $v \in R_p$. Let u_{np} be a non-parametric update. For all w, $w \in \mathit{update}((E, v(D)), u_{np})$ iff $w' \in (E, v(D))$ for some w' s.t. $w = [w']_{u_{np}}$.*

Proof idea. The technical part is ($\Rightarrow$). The idea is to prove that, given $w' \in \mathit{update}((E, v(D)), u_{np})$ there is a non-empty set of clock valuations w s.t. $w' = [w]_{u_{np}}$ that is precisely defined by the constraints in $(E, v(D))$. See Appendix .2 for details. ◀

3.1.2 Time elapsing

Given R_p and $(E, D) \in p$–$\mathcal{PDBM}_{\blacksquare}(R_p)$, we show that the clocks with the (possibly parametric) largest fractional part *i. e.,* the clocks that have a larger fractional part than any other clock, can always be identified by their bounds in D.

▶ **Definition 14** (clocks with the largest fractional part in a p–PDBM). Let R_p be a parameter region and $(E, D) \in p$–$\mathcal{PDBM}(R_p)$. A *clock with the (possibly parametric) largest fractional part* is a clock x s.t. for all $0 \leq i \leq H$, $0 \leq D_{x,i}$ is valid for R_p.

There is at least one clock with the (possibly parametric) largest fractional part:

▶ **Lemma 15** (existence of a clock with the largest fractional part). *Let R_p be a parameter region and $(E, D) \in p$–$\mathcal{PDBM}(R_p)$. There is at least one clock x s.t. for all $0 \leq i \leq H$, $0 \leq D_{x,i}$ is valid for R_p.*

Note that several clocks may have the largest fractional parts (up to some syntactic replacements , in that case they satisfy the same constraints in (E, D)).

Algorithm 2: $TE_{<}((E, D))$: set upper bound of all $\mathit{frac}(x_i) \in \mathsf{LFP}_{R_p}(D)$ to 1

pick $x_i \in \mathsf{LFP}_{R_p}(D)$
for j *from* 1 *to* H **do**
  if $j \in \mathsf{LFP}_{R_p}(D)$ **then**
    $D_{j,0} := (1, <)$
  else
    $D_{j,0} := D_{j,i} + (1, <)$
  end
  $D_{0,j} := D_{0,j} + (0, <)$
end

For a p–PDBM (E, D), we define the set of clocks with the largest fractional part (LFP) as $\mathsf{LFP}_{R_p}(D) = \{x \in [1, H] \mid 0 \leq D_{x,i}$ is valid for R_p, for all $0 \leq i \leq H\}$.

As we are able, thanks to the parameter regions, to order our parameter valuations (*i. e.,* whether one is greater or less than another one), we can define LFP from the constraints defined in the point–p–PDBM. We will define and apply successively two time-elapsing algorithms: the first one starts from a point–p–PDBM or an open–p–PDBM respecting condition Definition 8 (5a). We will prove that we obtain an open–p–PDBM respecting condition Definition 8 (5b). The second one, starts from an open–p–PDBM respecting condition Definition 8 (5b) and will define the set of constraints defining the possible clocks valuations exactly when any clock of LFP has reached its upper bound 1. We will prove that we obtain an open–p–PDBM respecting condition Definition 8 (5a). As we will obtain at each iteration of the algorithm an open–p–PDBM respecting either condition Definition 8 (5a) or (5b), this will prove we have a stable set of open–p–PDBMs. Now we explain our algorithms more precisely.

Clocks belonging to LFP are the first to reach the upper bound 1 by letting time elapse. Since LFP can contain multiple clocks and they have the same fractional part, we can consider any $x \in \mathsf{LFP}$.

Let $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$ and $x_i \in \mathsf{LFP}_{R_p}(D)$. To formalize time elapsing until the largest fractional part $\mathit{frac}(x)$ reaches 1, we define two algorithms, $TE_<$ and $TE_=$.

Algorithm 3: $TE_=((E, D))$: set upper and lower bound of all $\mathit{frac}(x_i) \in \mathsf{LFP}_{R_p}(D)$ to 1

```
pick x_i ∈ LFP_{R_p}(D)
for j from 1 to H do
    if j ∈ LFP_{R_p}(D) then
        D_{j,0} := (0, ≤)
        D_{0,j} := (0, ≤)
        E_j := E_j + 1
    else
        D_{j,0} := D_{j,i} + (1, ≤)
        D_{0,j} := D_{i,j} + (−1, ≤)
    end
end
for j from 1 to H do
    D_{j,i} := D_{j,0}
    D_{i,j} := D_{0,j}
end
```

The first one $TE_<$ is applied to point–p–PDBMs and open–p–PDBMs respecting condition 5a; it sets $D_{x,0} := (1, <)$ and $D_{0,x} := D_{0,x} + (0, <)$ for all $x \in \mathsf{LFP}_{R_p}(D)$. Then, for all clocks $1 \leq j \leq H$ not in LFP sets $D_{j,0} := D_{j,i} + (1, <)$ and $D_{0,j} := D_{0,j} + (0, <)$. This gives the range of possible clock valuations before $\mathit{frac}(x_i)$ reaches 1. The obtained result is denoted by $TE_<((E, D))$, and it leaves E unchanged. The second one $TE_=$ is applied to open–p–PDBMs respecting condition 5b and sets $D_{x,0} := (0, \leq)$ and $D_{0,x} := (0, \leq)$ for all $x \in \mathsf{LFP}_{R_p}(D)$. Then, for all clocks $x_j \in H \setminus \mathsf{LFP}_{R_p}(D)$ sets $D_{0,j} := (-1, \leq) + D_{i,j}$ and $D_{j,0} := D_{j,i} + (1, \leq)$; it gives the range of clock valuations when $\mathit{frac}(x)$ reaches 1, and increments E_x, for $x \in \mathsf{LFP}_{R_p}(D)$. It then sets, regardless of whether $x_j \in \mathsf{LFP}_{R_p}(D)$ $D_{i,j} := D_{0,j}$ and $D_{j,i} := D_{j,0}$. The last step is to set E_x to ∞ if $E_x > K$ (the maximum between constants and parameter bounds in the R-U2P-PTA) or is ∞. The obtained result is denoted by $TE_=((E, D))$.

▶ **Definition 16** (time elapsing in a p–PDBM)**.** Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}_\odot(R_p) \cup p\text{–}\mathcal{PDBM}_\blacksquare(R_p)$. We define $(E', D') = TE((E, D))$ as applying either $TE_<$ if (E, D) respects condition 5a or $(E, D) \in p\text{–}\mathcal{PDBM}_\odot(R_p)$, or $TE_=$ if (E, D) respects condition 5b.

▶ **Lemma 17** (stability under time elapsing)**.** *Let R_p be a parameter region. Let $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$. Then $TE((E, D)) \in p\text{–}\mathcal{PDBM}(R_p)$.*

Proof idea. The idea is to apply the time elapsing to a p–PDBM (depending on its type), and to check that the result still matches our Definition 8. Although we perform some additions such as $D_{j,i} + (1, <)$, we do not create new expressions that are not in $\mathcal{PLT}$. In fact, this addition is performed on a negative term (*e. g.*, $\mathit{frac}(p) - 1$), as x_i is a clock with the largest fractional part and adding 1 transforms it into another term of $\mathcal{PLT}$. The intuition is similar when performing additions such as $D_{i,j} + (-1, \leq)$: as x_i is a clock with the largest fractional part, $d_{i,j}$ is a positive term. See Appendix .3 for details. ◀

Note that, from Lemma 17 (E', D') is *always* an open–p–PDBM. open–p–PDBMs are stable under $TE_<$ and $TE_=$, switching the condition they respect (5a, 5b). Applying $TE_<$ on a point–p–PDBM transforms it into an open–p–PDBM. The following proposition proves that time elapsing behaves as we expect.

▶ **Proposition 18** (semantic of p–PDBM under TE). *Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$. Let $v \in R_p$. There exists $w' \in TE((E, v(D)))$ iff there exist $w \in (E, v(D))$ and a delay δ s.t. $w' = w + \delta$.*

Proof idea. This proof is quite technical. Intuitively, we bound the difference of each upper bound $v(d_{i,0})$ and $w(x_i)$ and each lower bound $v(d_{0,i})$ and $w(x_i)$. This allows us to take a delay δ inside these bounds that allows us to reach the next p–PDBM. See Appendix .19 for details. ◀

3.1.3 Non-parametric guard

We claim that intersecting a non-parametric guard with a p–PDBM does not affect the p–PDBM. It is sufficient to test whether the solution set is empty or not, and it is non empty if and only if every clock valuation in the p–PDBM satisfies the guard.

Our idea is to define a clock region "larger" than our p–PDBM and show that, even for this (larger) clock region, either all clock valuations satisfy the guard—or none do.

▶ **Definition 19.** Let R_p be a parameter region, $v \in R_p$. Let (E, D) be a p–PDBM for R_p. We define the *clock region containing* $(E, v(D))$, denoted by $[(E, v(D))]_{R_c}$, as follows: for all $w \in [(E, v(D))]_{R_c}$, for all clocks x_i, x_j,

- if $E_{x_i} < K$, $\lfloor w(x_i) \rfloor = E_{x_i}$, else if $E_{x_i} = \infty$, $w(x_i) \geq K$
- if $(0, \leq) < D_{i,j}$ is valid for R_p and $E_{x_i} < K$, $\mathit{frac}(w(x_j)) < \mathit{frac}(w(x_i))$
- if $(0, \leq) = D_{i,j}$ is valid for R_p and $E_{x_i} < K$, $\mathit{frac}(w(x_j)) = \mathit{frac}(w(x_i))$
- if $D_{i,0} = D_{0,i} = (0, \leq)$ and $E_{x_i} < K$, $\mathit{frac}(w(x_i)) = 0$.
- if $D_{i,0} \neq (0, \leq), D_{0,i} \neq (0, \leq)$ and $E_{x_i} < K$, $\mathit{frac}(w(x_i)) \neq 0$.

▶ **Lemma 20.** *Let (E, D) be a p–PDBM for R_p and $v \in R_p$. We have $(E, v(D)) \subseteq [(E, v(D))]_{R_c}$.*

Proof. Clock regions of Definition 3 define constraints on clocks of the form $0 = \mathit{frac}(x)$, $0 < \mathit{frac}(x) < 1$, $0 = \mathit{frac}(x) - \mathit{frac}(y)$ and $0 < \mathit{frac}(x) - \mathit{frac}(y) < 1$ for some x, y, and $\lfloor x \rfloor = k$ for some integer k. Let (E, D) be a p–PDBM for R_p and $v \in R_p$. It defines a set of constraints

$$\bigwedge_{i,j \in [0,H]^2} \mathit{frac}(x_i) - \mathit{frac}(x_j) \lhd_{i,j} v(d_{i,j}) \quad \wedge \quad \bigwedge_{i \in [1,H]} \lfloor x_i \rfloor = E_i.$$

traints $\bigwedge_{i \in [1,H]} \lfloor x_i \rfloor = E_i$. Clearly, if $w \in (E, v(D))$ satisfies $\lfloor x_i \rfloor = E_i$ then it satisfies the same constraint defined in $[(E, v(D))]_{R_c}$.

Consider the constraints $\mathit{frac}(x_i) - \mathit{frac}(x_j) \lhd_{i,j} v(d_{i,j})$ and $\mathit{frac}(x_j) - \mathit{frac}(x_i) \lhd_{j,i} v(d_{j,i})$.

- If i, j are both different from 0. From Definition 8 (3) and Definition 9, either $d_{i,j} = d_{j,i}$ and then $\triangleleft_{i,j} = \leq = \triangleleft_{j,i}$, then if $d_{i,j} = d_{j,i} = 0$ it satisfies the same constraint defined in $[(E, v(D))]_{R_c}$, or $d_{i,j}$ and $d_{j,i}$ are different from 0, as they are elements of $\mathcal{PLT}$ which are strictly smaller than 1, it satisfies either $0 < \mathit{frac}(x_i) - \mathit{frac}(x_j) < 1$ or $0 = \mathit{frac}(x_i) - \mathit{frac}(x_j)$ in $[(E, v(D))]_{R_c}$. Finally if $d_{i,j} \neq d_{j,i}$, then $\triangleleft_{i,j} = < = \triangleleft_{j,i}$ and it satisfies $0 < \mathit{frac}(x_i) - \mathit{frac}(x_j) < 1$ in $[(E, v(D))]_{R_c}$.
- If i is different from 0 and $j = 0$. From Definition 8 (3) and Definition 9, either $d_{i,0} = d_{0,i}$ and then $\triangleleft_{i,0} = \leq = \triangleleft_{0,i}$, then if $d_{i,0} = d_{0,i} = 0$ it satisfies the same constraint defined in $[(E, v(D))]_{R_c}$, or $d_{i,0}$ and $d_{0,i}$ are different from 0, as they are elements of $\mathcal{PLT}$ which are strictly smaller than 1, it satisfies either $0 < \mathit{frac}(x_i) < 1$ or $0 = \mathit{frac}(x_i)$ in $[(E, v(D))]_{R_c}$. Finally if $d_{i,0} \neq d_{0,i}$, then $\triangleleft_{i,0} = < = \triangleleft_{0,i}$ and it satisfies $0 < \mathit{frac}(x_i) < 1$ in $[(E, v(D))]_{R_c}$.
- The case j is different from 0 and $i = 0$ is similar.
- If both i, j are 0, the constraint is not taken into account as we have not x_0 in $[(E, v(D))]_{R_c}$.

Finally, we have that if $w \in (E, v(D))$ then $w \in [(E, v(D))]_{R_c}$. ◀

From [AD94, Section 4.2] we have that either every clock valuation of a clock region satisfies a guard, or none of them does. As a p–PDBM for R_p is contained into its containing clock region from Lemma 20, we have that if $w \in (E, v(D))$ satisfies a non-parametric guard g, then for all $w' \in (E, v(D))$ we also have w' satisfies g.

Let $v \in R_p$. We define $v \in \mathit{guard}_\forall(g, E, D)$ iff for all $w \in (E, v(D))$, $w \models g$. As any two $v, v' \in R_p$ satisfy the same constraints, we state the following lemma:

▶ **Lemma 21.** *Let (E, D) be a p–PDBM for R_p and $v \in R_p$. Let g be a non-parametric guard. If $v \in \mathit{guard}_\forall(g, E, D)$, then for all $v' \in R_p$, $v' \in \mathit{guard}_\forall(g, E, D)$.*

Proof. Let (E, D) be a p–PDBM for R_p and $v \in R_p$. It defines a set of constraints

$$\bigwedge_{i,j \in [0,H]^2} \mathit{frac}(x_i) - \mathit{frac}(x_j) \triangleleft_{i,j} v(d_{i,j}) \quad \wedge \quad \bigwedge_{i \in [1,H]} \lfloor x_i \rfloor = E_i.$$

Moreover, let g be a non-parametric guard. It defines a set of constraints for a finite number of integer constants k_i with $i \in I \subseteq [1, H]$

$$\bigwedge_{i \in I} \mathit{frac}(x_i) \leq 0 \quad \wedge \quad \bigwedge_{i \in I} -\mathit{frac}(x_i) \leq 0 \quad \wedge \quad \bigwedge_{i \in I} \lfloor x_i \rfloor \bowtie k_i.$$

The intersection between the two is given by the conjunction of those constraints. We project this intersection on parameter variables (by elimination of clock variables) and we prove that the intersection does not create new constraints on parameters different from those we already have in $(E, v(D))$ (and therefore in R_p). For some set of clocks $I \subseteq [1, H]$ and $i \in I$, suppose we have the constraints $\mathit{frac}(x_i) \leq 0$ and $-\mathit{frac}(x_i) \leq 0$ in g. When eliminating x_i in any constraint of the form $\mathit{frac}(x_i) - \mathit{frac}(x_j) \triangleleft_{i,j} v(d_{i,j})$, it is clear that we proceed on $\mathcal{PLT}$ to the operation $(0, \leq) + (d_{i,j}, \triangleleft_{i,j}) = (0 + d_{i,j}, \leq \oplus \triangleleft_{i,j}) = (d_{i,j}, \triangleleft_{i,j})$. The same way on any constraint of the form $\mathit{frac}(x_i) \triangleleft_{i,0} v(d_{i,0})$, eliminating x_i gives the constraint $(0, \leq) + (d_{i,0}, \triangleleft_{i,0}) = (d_{i,0}, \triangleleft_{i,0})$. Hence it does not create new inequalities not belonging to R_p.

Now suppose $v \in \mathit{guard}_\forall(g, E, D)$. We have that all $w \in (E, v(D))$ satisfies g. As no new constraints not in PLT have been created, all $v' \in R_p$ respect the same constraints on their fractional part and integer part as v and therefore, $(E, v'(D))$ is contained in the same clock region as $(E, v(D))$ is, *i. e.*, $[(E, v(D))]_{R_c} = [(E, v'(D))]_{R_c}$. Finally, $v' \in \mathit{guard}_\forall(g, E, D)$. ◀

3.1.4 Parametric guard

We assume a parametric guard g composed of conjunctions of comparisons $x \bowtie p$ for some clocks x and parameters p and $x \bowtie k$ for some integers. We focus here on the conjunctions of comparisons $x \bowtie p$. Given $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$ and $v \in R_p$, we compute the intersection between the set of constraints defined by (E, D) and g. We again prove that it does not create new constraints on parameters different from those in R_p. This is a key point in the overall process of proving the decidability of our R-U2P-PTAs.

Claim: *Let $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$ and $v \in R_p$. Let g be a parametric guard. The projection onto the parameters of the intersection of $(E, v(D))$ and $v(g)$ is contained in R_p.*

As for the previous result, using a projection on parameters *i. e.,* eliminating clocks, does not create new constraints on parameters that are not already in a parameter region R_p. Indeed, a parametric guard g only adds new constraints of the form $x \bowtie p$ which gives, when eliminating clocks in both a p–PDBM (E, D) and a parametric guard, again a comparison between elements of $\mathcal{PLT}$. Therefore, these new constraints already belong to $\mathcal{PLT}$ and we can decide whether the set of clock valuations satisfying this set of constraints is non-empty *i. e.,* given $v \in R_p$, $v(g)$ is satisfied by some clock valuation $w \in (E, v(D))$. See Appendix A for a detailed argument.

Note that there will also be additional constraints involving clocks (with other clocks, constants or parameters), but they will not be relevant as we immediately update all clocks, therefore replacing these constraints with new constraints encoding the clock updates.

Let $v \in R_p$. We define $v \in p\text{-}guard_\exists(g, E, D)$ iff there is a $w \in (E, v(D))$ s.t. $w \models v(g)$.[2] Again, as any two $v, v' \in R_p$ satisfy the same constraints, we state the following lemma:

▶ **Lemma 22.** *Let (E, D) be a* p*–PDBM for R_p and $v \in R_p$. Let g be a parametric guard. If $v \in p\text{-}guard_\exists(g, E, D)$, then for all $v' \in R_p$, $v' \in p\text{-}guard_\exists(g, E, D)$.*

Proof. Let (E, D) be a p–PDBM for R_p and $v \in R_p$. Let g be a parametric guard and suppose $v \in p\text{-}guard_\exists(g, E, D)$. After applying a projection on parameters, we obtain a set of constraints on elements of $\mathcal{PLT}$. By hypothesis, all these constraints are satisfied by v. Suppose $v' \in R_p$. By definition of our parameter regions, and since v and v' both belong to R_p, v' satisfies the same set of constraints on elements of $\mathcal{PLT}$. Therefore, the same set of constraints is satisfied by v' and $v' \in p\text{-}guard_\exists(g, E, D)$. ◀

Now that we have defined useful operations on p–PDBMs, we are going, given a parameter region R_p, to construct a finite region automaton in which for any run, there is an equivalent concrete run in the R-U2P-PTA.

3.2 Parametric region automaton

▶ **Definition 23** (clock valuations in equivalent valuated p–PDBMs)**.** Let R_p be a parameter region. $\doteq$ is an equivalence relation defined by: $(w, v) \doteq (w', v')$ iff $v \in R_p, v' \frown v$, there is $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$ s.t. $w \in (E, v(D))$ and $w' \in (E, v'(D))$.

Let $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$, we say $(E', D') \in \mathsf{Succ}((E, D)) \Leftrightarrow \exists\, i \geq 0$ s.t. $(E', v(D')) = TE^i((E, D))$. In other words, (E', D') is obtained after applying $TE((E, D))$ a finite number of times. $\mathsf{Succ}((E, D))$ is also called a *time successor* of (E, D).

[2] Remark that here is why our construction works for EF-emptiness, but cannot be used for, *e. g.,*, AF-emptiness (“is there a parameter valuation such that all runs reach a goal location l”): unlike $guard_\forall(g, E, D)$, not all clock valuations in a p–PDBM $(E, v(D))$ can satisfy a parametric guard if $v \in p\text{-}guard_\exists(g, E, D)$.

Given $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$ we write $\overline{update}((E, D), u)$ to denote the update of (E, D) by u, when u is a total parametric update function, *i. e.,* updating the set of clocks exclusively to parameters. We therefore obtain a point–p–PDBM, containing the parametric set of constraints defining a unique clock valuation. Recall that a total update function which is not fully parametric (*i. e.,* an update of some clocks to parameters and some others to constants) can be encoded as a total parametric update immediately followed by a partial non-parametric update function.

In order to finitely bisimulate an R-U2P-PTA, we create a parametric region automaton.

▶ **Definition 24** (Parametric region automaton)**.** Let R_p be a parameter region. For an R-U2P-PTA $\mathcal{A} = (\Sigma, L, l_0, \mathbb{X}, \mathbb{P}, \zeta)$, given (E_0, D_0) the initial open–p–PDBM $\{\vec{0}\} \times R_p$, the *parametric region automaton* $\mathcal{R}(\mathcal{A})$ over R_p is the tuple $(L', \Sigma, L'_0, \zeta')$ where:

1. $L' = L \times p\text{–}\mathcal{PDBM}(R_p)$
2. $L'_0 = (l_0, (E_0, D_0))$
3. $\zeta' = \{\big((l, (E, D)), e, (l', (E', D')\big) \in L' \times \zeta \times L' \mid$ either $\exists e = \langle l, g, a, u_{np}, l' \rangle \in \zeta$, g is a non-parametric guard, $\exists (E'', D'') \in \mathsf{Succ}((E, D))$, $R_p \subseteq guard_{\forall}(g, (E'', D''))$ and $(E', D') = update(E'', D'', u_{np})$ is an open–p–PDBM, or $\exists e = \langle l, g, a, u, l' \rangle \in \zeta$, g is a parametric guard, $\exists (E'', D'') \in \mathsf{Succ}((E, D))$, $R_p \subseteq p\text{-}guard_{\exists}(g, (E'', D''))$ and $(E', D') = \overline{update}(E'', D'', u)$ is a point–p–PDBM.$\}$

Let R_p be a parameter region, $\mathcal{A}$ be an R-U2P-PTA and $\mathcal{R}(\mathcal{A}) = (L', \Sigma, L'_0, \zeta')$ its parametric region automaton over R_p. A run in $\mathcal{R}(\mathcal{A})$ is an untimed sequence $\sigma : (l_0, (E_0, D_0))e_0(l_1, (E_1, D_1))e_1 \cdots (l_i, (E_i, D_i))e_i(l_{i+1}, (E_{i+1}, D_{i+1}))e_{i+1} \cdots$ such that for all i we have $\big((l_i, (E_i, D_i)), e_i, (l_{i+1}, (E_{i+1}, D_{i+1}))\big) \in \zeta'$, which we also write $(l_i, (E_i, D_i)) \xrightarrow{e_i} (l_{i+1}, (E_{i+1}, D_{i+1}))$ where e_i. Note that we label our transitions with the edges of the R-U2P-PTA.

3.3 Decidability of EF-emptiness and synthesis

Using our construction of the parametric region automaton $\mathcal{R}(\mathcal{A})$ for a given R-U2P-PTA $\mathcal{A}$, we state in the next proposition that there is a bisimulation between $\mathcal{A}$ and $\mathcal{R}(\mathcal{A})$.

▶ **Proposition 25.** *Let R_p be a parameter region. Let $\mathcal{A}$ be an R-U2P-PTA and $\mathcal{R}(\mathcal{A})$ its parametric region automaton over R_p. There is a run $\sigma : (l_0, (E_0, D_0)) \xrightarrow{e_0} (l_1, (E_1, D_1)) \xrightarrow{e_1} \cdots (l_{f-1}, (E_{f-1}, D_{f-1})) \xrightarrow{e_{f-1}} (l_f, (E_f, D_f))$ in $\mathcal{R}(\mathcal{A})$ iff for all $v \in R_p$ there is a run $\rho : (l_0, w_0) \xrightarrow{e_0} (l_1, w_1) \xrightarrow{e_1} \cdots (l_{f-1}, w_{f-1}) \xrightarrow{e_{f-1}} (l_f, w_f)$ in $v(\mathcal{A})$ s.t. for all $0 \leq i \leq f$, $w_i \in (E_i, v(D_i))$.*

From Proposition 25, we deduce that if there is a run reaching a goal location in an instantiated R-U2P-PTA, then for another parameter valuation in the same parameter region there is a run in the instantiated R-U2P-PTA with the same locations and transitions (but possibly different delays), reaching the same location.

▶ **Theorem 26.** *Let $\mathcal{A}$ be an R-U2P-PTA. Let R_p be a parameter region and $v \in R_p$. If there is a run $\rho = (l_0, w_0) \xrightarrow{e_0} \cdots \xrightarrow{e_{i-1}} (l_i, w_i)$ in $v(\mathcal{A})$, then for all $v' \in R_p$ there is a run $\rho' = (l_0, w'_0) \xrightarrow{e_0} \cdots \xrightarrow{e_{i-1}} (l_i, w'_i)$ in $v'(\mathcal{A})$ such that for all i, $(w_i, v) \simeq (w'_i, v')$.*

Note that there is a finite number of p–PDBMs for each parameter region R_p. Let $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$ and consider $\mathcal{PLT}$: D is an $(H + 1)^2$ matrix made of pairs $(d, \triangleleft)$ where $d \in \mathcal{PLT}$ and $\triangleleft \in \{\leq, <\}$. Therefore the number of possible D is bounded by $(2 \times (2 + 3 \times \binom{M}{2}) +$

$4 \times M))^{(H+1)^2}$. Moreover the number of E is unbounded, but only a finite subset of all values needs to be explored, *i. e.,* those smaller than $K+1$: indeed, following classical works on timed automata [AD94, BDFP04], (integer) values exceeding the largest constant used in the guards or the parameter bounds are equivalent.

To test EF-emptiness given a bounded R-U2P-PTA $\mathcal{A}$ and a goal location l, we first enumerate all parameter regions (which are in finite number), and apply for each R_p the following process: we pick $v \in R_p$ (*e. g.,* using a linear programming algorithm [Kar84]). Then, we consider $v(\mathcal{A})$ which is an updatable timed automaton and test the reachability of l in $v(\mathcal{A})$ [BDFP04]. Then EF-emptiness is false if and only if there is v and a run in $v(\mathcal{A})$ reaching l.

▶ **Theorem 27.** *The EF-emptiness problem is PSPACE-complete for bounded R-U2P-PTAs.*

Given a goal location l and a bounded R-U2P-PTA $\mathcal{A}$, we can exactly synthesize the parameter valuations v s.t. there is a run in $v(\mathcal{A})$ reaching l by enumerating each parameter region (of which there is a finite number) and test if l is reachable for one of its parameter valuations. The result of the synthesis is the union of the parameter regions for which one valuation (and, from our results, all valuations in that region) indeed reaches the goal location in the instantiated TA.

▶ **Corollary 28.** *Given a bounded R-U2P-PTA $\mathcal{A}$ and a goal location l we can effectively compute the set of parameter valuations v s.t. there is a run in $v(\mathcal{A})$ reaching l.*

▶ Remark. By bounding parameter valuations in guards but not those used in updates, we still have a finite number of parameter regions. Indeed, an integer vector E with components E_x greater than $\lfloor K \rfloor + 1$ is equivalent to an integer vector E' with $E'_x = E_x$ if $E_x < \lfloor K \rfloor + 1$ and $E'_x = \lfloor K \rfloor + 1$ if $E_x \geq \lfloor K \rfloor + 1$. Moreover for all p, we have to replace each parameter valuation v used in an update by $v(p) = v'(p)$ if $v(p) \leq K$ and $v'(p) = K + 1$ if $v(p) > K$.

4 Conclusion and perspectives

Our class of bounded R-U2P-PTAs is one of the few subclasses of PTAs (actually even extended with parametric updates) to enjoy decidability of EF-emptiness. In addition, R-U2P-PTAs are the first “subclass” of PTAs to allow exact synthesis of bounded *rational*-valued parameters.

In terms of future works, beyond reachability emptiness, we aim at studying unavoidability-emptiness and language preservation emptiness, as well as their synthesis.

Finally, we would like to investigate whether our parametric updates can be applied to decidable hybrid extensions of TAs [HKPV98, BDG$^+$13].

References

AD94 Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, April 1994.

AHV93 Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. Parametric real-time reasoning. In *STOC*, pages 592–601. ACM, 1993.

AL17 Étienne André and Didier Lime. Liveness in L/U-parametric timed automata. In *ACSD*, pages 9–18. IEEE, 2017.

ALR16 Étienne André, Didier Lime, and Olivier H. Roux. Decision problems for parametric timed automata. In *ICFEM*, volume 10009 of *Lecture Notes in Computer Science*, pages 400–416. Springer, 2016.

ALR18a Étienne André, Didier Lime, and Mathias Ramparison. Parametric resets in parametric timed automata (full version), 2018. `https://lipn.univ-paris13.fr/~ramparison/articles/ResetPTAfull.pdf`.

ALR18b Étienne André, Didier Lime, and Mathias Ramparison. Timed automata with parametric resets. In *ACSD*, 2018. To appear, `https://lipn.univ-paris13.fr/~ramparison/articles/PresetTA.pdf`.

BBLS15 Nikola Beneš, Peter Bezděk, Kim G. Larsen, and Jiří Srba. Language emptiness of continuous-time parametric timed automata. In *ICALP, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 69–81. Springer, 2015.

BDFP04 Patricia Bouyer, Catherine Dufourd, Emmanuel Fleury, and Antoine Petit. Updatable timed automata. *Theoretical Computer Science*, 321(2-3):291–345, 2004.

BDG+13 Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. Time-bounded reachability for monotonic hybrid automata: Complexity and fixed points. In *ATVA*, volume 8172 of *Lecture Notes in Computer Science*, pages 55–70. Springer, 2013.

BL09 Laura Bozzelli and Salvatore La Torre. Decision problems for lower/upper bound parametric timed automata. *Formal Methods in System Design*, 35(2):121–151, 2009.

BY03 Johan Bengtsson and Wang Yi. Timed automata: Semantics, algorithms and tools. In *Lectures on Concurrency and Petri Nets, Advances in Petri Nets*, volume 3098 of *Lecture Notes in Computer Science*, pages 87–124. Springer, 2003.

Dil89 David Dill. Timing assumptions and verification of finite-state concurrent systems. In *CAV*, volume 407, pages 197–212, 1989.

Doy07 Laurent Doyen. Robust parametric reachability for timed automata. *Information Processing Letters*, 102(5):208–213, 2007.

HKPV98 Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences*, 57(1):94–124, 1998.

HRSV02 Thomas Hune, Judi Romijn, Mariëlle Stoelinga, and Frits W. Vaandrager. Linear parametric model checking of timed automata. *Journal of Logic and Algebraic Programming*, 52-53:183–220, 2002.

JLR15 Aleksandra Jovanović, Didier Lime, and Olivier H. Roux. Integer parameter synthesis for timed automata. *IEEE Transactions on Software Engineering*, 41(5):445–461, 2015.

Kar84 Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–396, 1984.

Mil00 Joseph S. Miller. Decidability and complexity results for timed automata and semi-linear hybrid automata. In *HSCC*, volume 1790 of *Lecture Notes in Computer Science*, pages 296–309. Springer, 2000.

Sav70 Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

The following three lemmas are direct from the definitions, and will be used in the subsequent results.

▶ **Lemma 29** (validity of addition). *Let $d_1, d_2, d_3, d_4 \in \mathcal{PLT}$. Let R_p be a parameter region. If $(d_1, \lhd_1) \leq (d_2, \lhd_2)$ and $(d_3, \lhd_3) \leq (d_4, \lhd_4)$ are valid for R_p then $(d_1, \lhd_1) + (d_3, \lhd_3) \leq (d_2, \lhd_2) + (d_4, \lhd_4)$ is valid for R_p.*

Proof. Four cases show up: for all $v \in R_p$,

- $v(d_1) < v(d_2)$ and $v(d_3) < v(d_4)$, then clearly $v(d_1) + v(d_3) < v(d_2) + v(d_4)$ and we have our result from Definition 7 (2a).
- $v(d_1) < v(d_2)$ and $v(d_3) \leq v(d_4)$, then $v(d_1) + v(d_3) < v(d_2) + v(d_4)$ and we have our result from Definition 7 (2a).
- $v(d_1) \leq v(d_2)$ and $v(d_3) < v(d_4)$, then $v(d_1) + v(d_3) < v(d_2) + v(d_4)$ and we have our result from Definition 7 (2a).
- $v(d_1) \leq v(d_2)$ and $v(d_3) \leq v(d_4)$, then $v(d_1) + v(d_3) \leq v(d_2) + v(d_4)$ and
  1. if $\lhd_1 = \lhd_2$ and $\lhd_3 = \lhd_4$ then $\lhd_1 \oplus \lhd_3 = \lhd_2 \oplus \lhd_4$ and we have our result from Definition 7 (2b).
  2. if $\lhd_1 = \lhd_2$ and $\lhd_3 = <$, $\lhd_4 = \leq$ then $\lhd_1 \oplus \lhd_3 = <$ and $\lhd_2 \oplus \lhd_4$ is either $<$ or $\leq$ and we have our result from Definition 7 (2b).
  3. if $\lhd_1 = <$, $\lhd_2 = \leq$ and $\lhd_3 = \lhd_4$ then $\lhd_1 \oplus \lhd_3 = <$ and $\lhd_2 \oplus \lhd_4$ is either $<$ or $\leq$ and we have our result from Definition 7 (2b).
  4. if $\lhd_1 = \lhd_3 =<$ and $\lhd_2 = \lhd_4 =\leq$ then $\lhd_1 \oplus \lhd_3 =<$ and $\lhd_2 \oplus \lhd_4 =\leq$ and we have our result from Definition 7 (2b).

From Definition 7 (2a, 2b) we have that $(d_1, \lhd_1) + (d_3, \lhd_3) \leq (d_2, \lhd_2) + (d_4, \lhd_4)$ is valid for R_p. ◀

▶ **Lemma 30** (positivity of reflexivity). *Let R_p be a parameter region and (E, D) be a p–PDBM for R_p. For all clocks i, j, $(0, \leq) \leq D_{i,j} + D_{j,i}$ is valid for R_p.*

Proof. By condition (4) in Definition 8 and Definition 9 (2), we have that $D_{i,i} \leq D_{i,j} + D_{j,i}$ is valid for R_p; the result follows from the fact that $D_{i,i} = (0, \leq)$ (again from Definition 8 and Definition 9). ◀

▶ **Lemma 31** (neutral element of the set of cells). *Let R_p be a parameter region and (E, D) be a p–PDBM for R_p. For all clocks i, j, $D_{i,j} \leq D_{i,j} + D_{j,j}$ and $D_{i,j} \leq D_{i,i} + D_{i,j}$ are valid for R_p.*

Proof. Let R_p be a parameter region and (E, D) be a p–PDBM for R_p. Let $D_{i,j} = (d_{i,j}, \lhd_{ij})$ with $d_{i,j} \in \mathcal{PLT}$. By Definition 8 and Definition 9 for all clock i, $D_{i,i} = (0, \leq)$. We have $D_{j,i} + D_{i,i} = (d_{j,i} + 0, \lhd_{ij} \oplus \leq) = D_{j,i}$. Moreover from Definition 7 (2b) $D_{i,j} \leq D_{i,j}$ is valid for R_p. Hence $D_{i,j} \leq D_{i,i} + D_{i,j}$ is valid for R_p. The same way we prove $D_{i,j} \leq D_{i,j} + D_{j,j}$ is valid for R_p. ◀

.1 Proof of Lemma 12

We split this proof in two parts: the first one treats the case of point–p–PDBMs and the second one of open–p–PDBMs.

The following lemma shows that applying a *update* on any point–p–PDBM transforms it into an open–p–PDBM.

▶ **Lemma 32** (p–$\mathcal{PDBM}_{\odot}(R_p)$ becomes p–$\mathcal{PDBM}_{\blacksquare}(R_p)$ after *update*)**.** *Let R_p be a parameter region and $(E, D) \in p$–$\mathcal{PDBM}_{\odot}(R_p)$. Let u_{np} be a non-parametric update. Then update$((E, D), u_{np}) \in p$–$\mathcal{PDBM}_{\blacksquare}(R_p)$.*

Proof. Let R_p be a parameter region and $(E, D) \in p$–$\mathcal{PDBM}_{\odot}(R_p)$. Consider $(E', D') =$ *update*$((E, D), u_{np})$. After applying Algorithm 1, for all clock x_i of (E, D) where u_{np} is defined, $E'_i = u_{np}(x_i)$; moreover for all clock j, $D'_{i,j} = D_{0,j}$ and $D'_{j,i} = D_{j,0}$. First note that if x_i, x_j have been updated, $D'_{i,j} = D'_{j,i} = D'_{0,j} = D'_{j,0} = D'_{0,i} = D'_{i,0} = (0, \leq) = D_{0,0}$. For all clocks i, j, k, the following inequalities are valid for R_p:

1. **a.** if x_i is updated: $D'_{i,0} = (0, \leq) = D'_{0,i}$ and therefore trivially it holds that $-1 \leq D'_{0,i} \leq 0$ and $0 \leq D'_{i,0} \leq 1$ are valid for R_p;
   b. if x_i is not updated: $D'_{i,0} = D_{i,0}$ and therefore $-1 \leq D'_{0,i} \leq 0$ and $0 \leq D'_{i,0} \leq 1$ are valid for R_p because these constraints were already satisfied in (E, D).
2. For all x_i, x_j, if neither x_i nor x_j is updated, $D_{i,j}$ and $D_{j,i}$ are not modified so condition Definition 8 (2) still holds. If either x_i is updated, as $D'_{i,j} = D_{0,j}$ and $D'_{j,i} = D_{j,0}$ condition Definition 8 (2) still holds as it holds for $D_{0,j}$ and $D_{j,0}$ and we apply the same reasoning if x_j is updated. If both x_i, x_j are updated, condition Definition 8 (2) trivially holds.
3. For all x_i, if it is updated then $D'_{0,i} = D'_{i,0} = (0, \leq)$, hence $d_{0,i} = -d_{i,0} = 0$ and $\triangleleft_{0i} = \triangleleft_{i0} = \leq$; condition Definition 8 (3) holds. For all x_i, x_j, if neither x_i nor x_j is updated, $D'_{i,j} = D_{i,j}$ and $D'_{j,i} = D_{j,i}$ so condition Definition 8 (3) holds as it holds for $D_{i,j}$ and $D_{j,i}$. If either x_i is updated, as $D'_{i,j} = D_{0,j}$ and $D'_{j,i} = D_{j,0}$, condition Definition 8 (3) holds as it holds for $D_{0,j}$ and $D_{j,0}$. We treat the case where x_j is updated similarly. If both x_i, x_j are updated, condition Definition 8 (3) trivially holds.
4. Canonical form is preserved:
   a. if x_i, x_j, x_k are not updated: since no clock is updated we have $D'_{i,j} = D_{i,j}$, $D'_{j,k} = D_{j,k}$ and $D'_{i,k} = D_{i,k}$ since $(E, D) \in p$–$\mathcal{PDBM}_{\odot}(R_p)$ from Definition 9 (2), we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; therefore it remains valid.
   b. if x_k is updated and x_i, x_j are not updated: $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,0}$, $D'_{i,k} = D_{i,0}$ because x_k is updated. Since $(E, D) \in p$–$\mathcal{PDBM}_{\odot}(R_p)$ from Definition 9 (2), we know that $D_{i,0} \leq D_{i,j} + D_{j,0}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
   c. if x_j is updated and x_i, x_k are not updated: then $D'_{i,k} = D_{i,k}$ because neither x_i nor x_k are updated; since x_k is updated we have $D'_{j,k} = D_{0,k}$ and $D'_{i,j} = D_{i,0}$; since $(E, D) \in p$–$\mathcal{PDBM}_{\odot}(R_p)$ from Definition 9 (2), we know that $D_{i,k} \leq D_{i,0} + D_{0,k}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
   d. if x_j, x_k are updated and x_i is not updated: then $D'_{i,k} = D_{i,0}$ because x_k is updated; since x_j is updated we have $D'_{i,j} = D_{i,0}$ and $D'_{j,k} = D_{0,0}$; since $(E, D) \in p$–$\mathcal{PDBM}_{\odot}(R_p)$ from Definition 9 (2) and Lemma 31, we know that $D_{i,0} \leq D_{i,0} + D_{0,0}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
   e. if x_i is updated and x_j, x_k are not updated: then $D'_{i,k} = D_{0,k}$, $D'_{i,j} = D_{0,j}$ because x_i is updated; since x_j, x_k are not updated, we have $D'_{j,k} = D_{j,k}$; since $(E, D) \in p$–$\mathcal{PDBM}_{\odot}(R_p)$ from Definition 9 (2), we know that $D_{0,k} \leq D_{0,j} + D_{j,k}$ is valid for R_p; therefore $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
   f. if x_i, x_k are updated and x_j is not updated: we have $D'_{i,k} = (0, \leq) = D_{0,0}$, $D'_{i,j} = D_{0,j}$ and $D'_{j,k} = D_{j,0}$ because x_i, x_k are updated. Since $(E, D) \in p$–$\mathcal{PDBM}_{\odot}(R_p)$ from Definition 9 (2), we know that $D_{0,0} \leq D_{0,j} + D_{j,0}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.

g. if x_i, x_j are updated and x_k is not updated: we have $D'_{i,k} = D_{0,k}$, $D'_{i,j} = (0 <\leq) = D_{0,0}$ and $D'_{j,k} = D_{0,k}$ because x_i, x_j are updated. Since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$ from Definition 9 (2) and Lemma 31, we know that $D_{0,k} \leq D_{0,0} + D_{0,k}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.

h. if x_i, x_j, x_k are updated: we have $D'_{i,k} = D_{0,0}$, $D'_{i,j} = D_{0,0}$ and $D'_{j,k} = D_{0,0}$ because x_i, x_j, x_k are updated. Since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$ from Definition 9 (2) and Lemma 31, we know that $D_{0,0} \leq D_{0,0}+D_{0,0}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j}+D'_{j,k}$ is valid for R_p.

5. there is at least one clock x s.t. $D'_{x,0} = D'_{0,x} = (0, \leq)$.

Therefore, $(E', D') \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$. ◀

The following lemma shows that applying a *update* on any open–p–PDBM transforms it into an open–p–PDBM respecting Definition 8 (2).

▶ **Lemma 33** (stability of $p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ under *update*)**.** *Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$. Let u_{np} be a non-parametric update. Then update$((E, D), u_{np}) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$.*

Proof. Most cases are similar to the proof of Lemma 32.

The remaining cases to treat are the cases of Definition 8 (2). If i, j are different from 0, and

1. if i, j are not updated then $D'_{i,j} = D_{i,j}$ and since it is the case in (E, D), condition Definition 8 (2) holds.
2. if j is updated and i is not updated then $D'_{i,j} = D_{i,0}$ and $D'_{j,i} = D_{0,i}$ and as condition Definition 9 (1) holds for $D_{i,0}$ and $D_{0,i}$ in (E, D), condition Definition 8 (2) holds in (E', D').
3. if i is updated and j is not updated then $D'_{i,j} = D_{0,j}$ and $D'_{j,i} = D_{j,0}$ and as condition Definition 9 (1) holds for $D_{j,0}$ and $D_{0,j}$ in (E, D), condition Definition 8 (2) holds in (E', D').
4. if i, j are updated then trivially $D'_{i,j} = D'_{j,i} = (0, \leq)$ and condition Definition 8 (2) holds.

◀

.2 Proof of Lemma 13

Theorem 13 (recalled). *Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$. Let $v \in R_p$. Let u_{np} be a non-parametric update. For all w, $w \in$ update$((E, v(D)), u_{np})$ iff $w' \in (E, v(D))$ for some w' s.t. $w = [w']_{u_{np}}$.*

Proof. We first treat the case of the $p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ (the case of the $p\text{–}\mathcal{PDBM}_{\odot}(R_p)$ will be handled similarly at the end). We also prove this lemma for a singleton update (only one clock, say x_i) since updating several clocks can be done by applying several singleton updates in a 0 delay.

.2.1 $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, $(\Rightarrow)$

Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$. Let $v \in R_p$. Let u_{np} be a non-parametric update which updates x_i to an integer n and lets the value of other clocks unchanged. Consider $(E', D') =$ *update*$((E, v(D)), u_{np})$ and suppose $w' \in (E', D')$. We want to construct a valuation $w \in (E, v(D))$ s.t. $w' = u_{np}(w)$.

Let w be a clock valuation s.t. for all clock x_j where $i \neq j$, $w(x_j) = w'(x_j)$. That means that for all $j \neq i$,

$$\mathit{frac}(w(x_j)) \triangleleft_{j0} v(d_{j,0}), \quad -\mathit{frac}(w(x_j)) \triangleleft_{0j} v(d_{0,j}) \quad \text{and} \quad \lfloor w(x_j) \rfloor = E_j$$

hold from Definition 11 since it is the case in (E', D') and these values are left untouched by the update. Moreover for all $j \neq i$, $k \neq i$,

$$\mathit{frac}(w(x_j)) - \mathit{frac}(w(x_k)) \triangleleft_{jk} v(d_{j,k}) \quad \text{and} \quad \mathit{frac}(w(x_k)) - \mathit{frac}(w(x_j)) \triangleleft_{kj} v(d_{k,j})$$

again hold from Definition 11 since it is the case in (E', D') and these values are left untouched by the update.

We want a valuation for $w(x_i)$ s.t.

$$\mathit{frac}(w(x_i)) \triangleleft_{i0} v(d_{i,0}) \quad - \mathit{frac}(w(x_i)) \triangleleft_{0i} v(d_{0,i}) \quad \text{and} \quad \lfloor w(x_i) \rfloor = E_i$$

hold, and for all $j \neq i$, $k \neq i$,

$$\mathit{frac}(w(x_i)) - \mathit{frac}(w(x_j)) \triangleleft_{ij} v(d_{i,j}) \quad \text{and} \quad \mathit{frac}(w(x_k)) - \mathit{frac}(w(x_i)) \triangleleft_{ki} v(d_{k,i}) \qquad (1)$$

hold. Let us prove that such a valuation w exists. We set $\lfloor w(x_i) \rfloor = E_i$.

The following lemma proves transitivity of constraints on clocks with respect to constraints in a p–PDBM.

▶ **Lemma 34.** *Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$. Let $v \in R_p$. Let $w \in (E, v(D))$. For all clocks i, j, k, $\mathit{frac}(w(x_j)) - \mathit{frac}(w(x_k))(\triangleleft_{ji} \oplus \triangleleft_{ik})v(d_{j,i}) + v(d_{i,k})$.*

Proof. Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$. Let $v \in R_p$. Let $w \in (E, v(D))$.

Since $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$, for all i, j, k we have from Definition 8 (4),

$$D_{j,k} \leq D_{j,i} + D_{i,k}$$

is valid for R_p hence since $v \in R_p$, we have $v(D_{j,k}) \leq v(D_{j,i}) + v(D_{i,k})$. Precisely that is $(v(d_{j,k}), \triangleleft_{jk}) \leq (v(d_{j,i}), \triangleleft_{ji}) + (v(d_{i,k}), \triangleleft_{ik})$ *i. e.*,

$$(v(d_{j,k}), \triangleleft_{jk}) \leq (v(d_{j,i}) + v(d_{i,k}), \triangleleft_{ji} \oplus \triangleleft_{ik}).$$

For all clocks j, k satisfying constraints of (E, D),

$$\mathit{frac}(w(x_j)) - \mathit{frac}(w(x_k)) \triangleleft_{jk} v(d_{j,k}).$$

Then for all i, j, k, either:

- from Definition 7 (2a): $v(d_{j,k}) < v(d_{j,i}) + v(d_{i,k})$ and then, regardless of $\triangleleft_{jk}$ and $\triangleleft_{ji} \oplus \triangleleft_{ik}$ we have $\mathit{frac}(w(x_j)) - \mathit{frac}(w(x_k))(\triangleleft_{ji} \oplus \triangleleft_{ik})v(d_{j,i}) + v(d_{i,k})$, or
- from Definition 7 (2b):
  - $v(d_{j,k}) \leq v(d_{j,i}) + v(d_{i,k})$ and $\triangleleft_{jk} = <$, $\triangleleft_{ji} \oplus \triangleleft_{ik} = \leq$ and then we have $\mathit{frac}(w(x_j)) - \mathit{frac}(w(x_k))(\triangleleft_{ji} \oplus \triangleleft_{ik})v(d_{j,i}) + v(d_{i,k})$, or
  - $v(d_{j,k}) \leq v(d_{j,i}) + v(d_{i,k})$ and $\triangleleft_{jk} = \triangleleft_{ji} \oplus \triangleleft_{ik}$ and then we have $\mathit{frac}(w(x_j)) - \mathit{frac}(w(x_k))(\triangleleft_{ji} \oplus \triangleleft_{ik})v(d_{j,i}) + v(d_{i,k})$ which completes the proof.

This completes the proof of Lemma 34. ◀

For all $j \neq i$ and $k \neq i$, since $v(D_{j,k}) \leq v(D_{j,i}) + v(D_{i,k})$ from Definition 8 (4), we have $frac(w(x_j)) - frac(w(x_k)) \lhd_{jk} v(d_{j,k})$ and

$$frac(w(x_j)) - frac(w(x_k))(\lhd_{ji} \oplus \lhd_{ik}) v(d_{j,i}) + v(d_{i,k})$$

holds from Lemma 34. Hence

$$frac(w(x_j)) - v(d_{j,i})(\lhd_{ji} \oplus \lhd_{ik}) frac(w(x_k)) + v(d_{i,k}) \tag{2}$$

holds. Note that $\lhd_{ji} \oplus \lhd_{ik}$ is either $\leq$ or $<$. Note the following trick is inspired by [HRSV02, Proof of Lemma 3.5] and [HRSV02, Proof of Lemma 3.13]. Hence

$$I = \{t \in \mathbb{R}_+ \mid frac(w(x_j)) - v(d_{j,i}) \leq t \leq frac(w(x_k)) + v(d_{i,k}) \text{ for all clocks } j, k\}$$

is a non empty set. That means that choosing a $frac(w(x_i))$ with respect to constraints (1), recall that they are

$$frac(w(x_j)) - frac(w(x_i)) \lhd_{ji} v(d_{j,i}) \quad \text{and} \quad frac(w(x_i)) - frac(w(x_k)) \lhd_{ik} v(d_{i,k})$$

is equivalent to choose a $frac(w(x_i))$ s.t.

$$frac(w(x_j)) - v(d_{j,i}) \lhd_{ji} frac(w(x_i)) \quad \text{and} \quad frac(w(x_i)) \lhd_{ik} frac(w(x_k)) + v(d_{i,k})$$

which is a nonempty set from formula (2). Finally we choose a $frac(w(x_i)) \in I$, then $w \in (E, v(D))$ and it completes the proof.

.2.2 $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, $(\Leftarrow)$

Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$. Let $v \in R_p$. Let u_{np} be a non-parametric update which updates x_i to an integer n and lets the value of other clocks unchanged. Consider $(E', D') = update((E, v(D)), u_{np})$. Now suppose $w \in (E, v(D))$ and let $w' = [w]_{u_{np}}$.

- for x_i, since u_{np} is defined, $w'(x_i) = u_{np}(x_i) = E'_{x_i}$ (*i. e.*, $frac(w'(x_i)) = 0$) by applying *update* as defined in Definition 11. By applying *update* as defined in Definition 11, $D'_{i,0} = D'_{0,i} = (0, \leq)$, hence

  $$-frac(w'(x_i)) \lhd_{0i} v(d'_{0,i}) \quad \text{and} \quad frac(w'(x_i)) \lhd_{i0} v(d'_{i,0})$$

  hold from Definition 11 and Lemma 32. Moreover we know that for all $j \neq i$

  $$-v(D'_{i,j}) = -v(D'_{0,j}) \quad \text{and} \quad v(D'_{j,i}) = v(D'_{j,0}) \tag{3}$$

  holds from Definition 11, and we also know that

  $$frac(w'(x_j)) - frac(w'(x_i)) = frac(w'(x_j)) \tag{4}$$

  since $frac(w'(x_i)) = 0$. Hence, combining (3) and (4), clearly since

  $$-frac(w'(x_j)) \lhd_{0j} v(d'_{0,j}) \quad \text{and} \quad frac(w'(x_j)) \lhd_{j0} v(d'_{j,0})$$

  hold in (E', D'),

  $$frac(w'(x_j)) - frac(w'(x_i)) \lhd_{ji} v(d'_{j,i}) \quad \text{and} \quad frac(w'(x_i)) - frac(w'(x_j)) \lhd_{ij} v(d'_{i,j})$$

  hold.

- for any two clocks x_j, x_k where u_{np} is not defined, $w(x_j) = w'(x_j)$ and $w(x_k) = w'(x_k)$. Hence

$$-v(D'_{0,j}) \triangleleft_{0j} \mathit{frac}(w'(x_j)) \triangleleft_{j0} v(D'_{j,0})$$

and

$$-v(D'_{k,j}) \triangleleft_{kj} \mathit{frac}(w'(x_j)) - \mathit{frac}(w'(x_k)) \triangleleft_{jk} v(D'_{j,k})$$

hold from Definition 11 and Lemma 32 since bounds remain unchanged.

Then $w' \in \mathit{update}((E, v(D)), u_{np})$.

This concludes the case $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$.

Let us now treat the case $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$.

.2.3 $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, $(\Rightarrow)$

Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$. Let $v \in R_p$. Let u_{np} be a non-parametric update which updates x_i to an integer n and lets the value of other clocks unchanged. Consider $(E', D') = \mathit{update}((E, v(D)), u_{np})$ and suppose $w' \in (E', D')$. We want to construct a valuation

$$w \in (E, v(D)) \quad \text{s.t.} \quad w' = u_{np}(w)$$

Let w be a clock valuation s.t. for all clock x_j where $j \neq i$, $w(x_j) = w'(x_j)$. That means for all $j \neq i$,

$$\mathit{frac}(w(x_j)) \triangleleft_{j0} v(d_{j,0}), \quad -\mathit{frac}(w(x_j)) \triangleleft_{0j} v(d_{0,j}) \quad \text{and} \quad \lfloor w(x_j) \rfloor = E_j$$

hold from Definition 11 since it is the case in (E', D') and bounds remain unchanged *i. e.*, $D_{0,j} = D'_{0,j}$ and $D_{j,0} = D'_{j,0}$. Moreover for all $k \neq i$ and $k \neq j$,

$$\mathit{frac}(w(x_j)) - \mathit{frac}(w(x_k)) \triangleleft_{jk} v(d_{j,k}) \quad \text{and} \quad \mathit{frac}(w(x_k)) - \mathit{frac}(w(x_j)) \triangleleft_{kj} v(d_{k,j})$$

also hold from Definition 11 since it is the case in (E', D') and bounds remain unchanged *i. e.*, $D_{k,j} = D'_{k,j}$ and $D_{j,k} = D'_{j,k}$.

Recall that (E, D) contains only one clock valuation for each parameter valuation $v \in R_p$.

Let $\mathit{frac}(w(x_i)) = v(d_{i,0})$ (or equivalently $\mathit{frac}(w(x_i)) = -v(d_{0,i})$ since by Definition 9 we have $(d_{i,0}, \triangleleft_{i0}) = (-d_{0,i}, \triangleleft_{0i})$). Then, as it is the case in (E, D),

$$\mathit{frac}(w(x_i)) \triangleleft_{i0} v(d_{i,0}), \quad -\mathit{frac}(w(x_i)) \triangleleft_{0i} v(d_{0,i}) \quad \text{and} \quad \lfloor w(x_i) \rfloor = E_i$$

hold, and for all $j \neq i$, $k \neq i$,

$$\mathit{frac}(w(x_i)) - \mathit{frac}(w(x_j)) \triangleleft_{ij} v(d_{i,j}) \quad \text{and} \quad \mathit{frac}(w(x_k)) - \mathit{frac}(w(x_i)) \triangleleft_{ki} v(d_{k,i})$$

hold, which completes the proof, as $w \in (E, v(D))$ and $w' = u_{np}(w)$.

.2.4 $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, $(\Leftarrow)$

This case is straightforward and similar to the case $(\Leftarrow)$ above of open–p–PDBMs.

◀

.3 Proof of Lemma 17

We prove our lemma for the two types of open–p–PDBMs and for point–p–PDBMs, and split this proof in three lemmas.

.3.1 Definition 8 type (5a) to (5b)

▶ **Lemma 35** (modification of an open–p–PDBM respecting condition 5a under $TE_<$)**.** *Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}_\blacksquare(R_p)$ respecting condition 5a, then $TE_<((E, D)) \in p\text{–}\mathcal{PDBM}_\blacksquare(R_p)$ respecting condition 5b.*

Proof of Lemma 35. Suppose $(E, D) \in p\text{–}\mathcal{PDBM}_\blacksquare(R_p)$ respects condition (5a) of Definition 8, *i. e.,* we have at least an x s.t. $D_{x,0} = D_{0,x} = (0, \leq)$. Since, in R_p, we know which parameters have the largest fractional part, we can determine $\mathsf{LFP}_{R_p}(D)$ from Lemma 15. If more than one clock belong to $\mathsf{LFP}_{R_p}(D)$ then their valuations have the same fractional part. Indeed, from Definition 14 if $x_i, x_j \in \mathsf{LFP}_{R_p}(D)$ then both $(0, \leq) \leq D_{i,j}$ and $(0, \leq) \leq D_{j,i}$ are valid for R_p, and from Definition 8 (2) we must have $D_{i,j} = D_{j,i} = (0, \leq)(\star)$.

Let $v \in R_p$. Assume $x_i \in \mathsf{LFP}_{R_p}(D)$ and $w \in (E, v(D))$, by letting time elapse, $\mathit{frac}(w(x_i))$ is the first that might reach 1. Moreover, for all $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $\mathit{frac}(w(x_j))$ cannot reach 1 before $\mathit{frac}(w(x_i))$. We are going to construct a new $(E', D') = TE_<((E, D))$, which will be an open–p–PDBM respecting condition 5b of Definition 8. While detailing the procedure of $TE_<$, we are going to prove that Definition 8 (1) and (2) hold for (E', D'). Further we will prove that (4) and (5b) also hold.

.4 proof that Definition 8 (1) holds

According to the definition of $TE_<$ (Algorithm 9), the first step is to set a new upper bound

$$D'_{i,0} = (1, <) \quad \text{for all } x_i \in \mathsf{LFP}_{R_p}(D)$$

and obviously $(0, \leq) \leq D'_{i,0} \leq (1, \leq)$ is valid for R_p. Then we set new upper bounds for all other clock $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$ by setting

$$D'_{j,0} = D_{j,i} + (1, <).$$

Indeed, $D_{j,i}$ is the constraint on the lower bound of $\mathit{frac}(w(x_j)) - \mathit{frac}(w(x_i))$ and since the upper bound of x_i has increased, this gives the new upper bound of x_j. Note that since $x_i \in \mathsf{LFP}_{R_p}(D)$, from Definition 14 and Definition 8 (2) we have that $-1 \leq D_{j,i} \leq 0$ is valid for R_p for all clock x_j. Precisely, $d_{j,i} \in \{0, -p_1, p_2 - p_1, p_1 - 1 - p_2, p_1 - 1\}$ for some $p_1, p_2 \in \mathbb{P}$ where $p_2 \leq p_1$ is valid for R_p. Hence as $d_{j,i} + 1 \in \{1, 1 - p_1, p_2 + 1 - p_1, p_1 - p_2, p_1\}$, we have that $d'_{j,0} \in \mathcal{PLT}$, $\triangleleft_{ji'} = \triangleleft_{ji} \oplus < = <$ so $(0, \leq) \leq D'_{j,0} \leq (1, <)$ is valid for R_p.

Note that we cannot have $(d_{j,i}, \triangleleft_{ji}) = (-1, <)$ because even if $(d_{i,j}, \triangleleft_{ij}) = (1, <)$, since $(E, D) \in p\text{–}\mathcal{PDBM}_\blacksquare(R_p)$ we do not have have $0 \leq D_{j,i} + D_{i,j}$ is valid for R_p from Definition 8 (4) and Lemma 30.

Secondary we set for all clock x regardless of whether they are in $\mathsf{LFP}_{R_p}(D)$

$$D'_{0,x} = D_{0,x} + (0, <).$$

Since some time elapsed, lower bounds of all clocks are increased. Moreover, as $(-1, <) \leq D_{0,x} \leq (0, \leq)$ is valid for R_p from Definition 8 (1), $(-1, \leq) \leq D'_{0,x} \leq (0, \leq)$ is also valid for R_p.

Therefore, Definition 8 (1) holds.

.5 proof that Definition 8 (2) holds

Third we set for all clocks x, y regardless of whether they are in $\mathsf{LFP}_{R_p}(D)$

$$D'_{x,y} = D_{x,y}$$

so as Definition 8 (2) holds in (E, D), it still does. More intuitively since no fractional part has reached 1, constraints on differences of clocks and integer parts remain unchanged.

.6 proof that Definition 8 (3) holds

For all x_i:

- if $x_i \in \mathsf{LFP}_{R_p}(D)$, $D'_{i,0} = (1, <)$, $D'_{0,i} = D_{0,i} + (0, <)$ hence $d'_{i,0} \neq d'_{0,i}$ and $\triangleleft_{i0'}\triangleleft_{0i'} = <$, condition Definition 8 (3) holds;
- if $x_i \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $x \in \mathsf{LFP}_{R_p}(D)$, $D'_{i,0} = D_{i,x} + (1, <)$, $D'_{0,i} = D_{0,i} + (0, <)$ hence as $(0, \leq) \leq D'_{i,0}$ is valid for R_p and $D'_{0,i} \leq (0, \leq)$ is valid for R_p, we have $d'_{i,0} \neq d'_{0,i}$ and $\triangleleft_{i0'}\triangleleft_{0i'} = <$ and condition Definition 8 (3) holds.

For all x_i, x_j:

- if $x_i, x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $D'_{i,j} = D_{i,j}$ and $D'_{j,i} = D_{j,i}$, condition Definition 8 (3) holds as it holds for $D_{i,j}$ and $D_{j,i}$.
- if $x_i \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $x_j \in \mathsf{LFP}_{R_p}(D)$, $D'_{i,0} = D_{i,j} + (1, <)$, $D'_{0,i} = D_{0,i} + (0, <)$ hence as $(0, \leq) \leq D'_{i,0}$ is valid for R_p and $D'_{0,i} \leq (0, \leq)$ is valid for R_p, we have $d'_{i,0} \neq d'_{0,i}$ and $\triangleleft_{i0'}\triangleleft_{0i'} = <$, condition Definition 8 (3) holds. The case $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $x_i \in \mathsf{LFP}_{R_p}(D)$ is treated similarly.
- if $x_i, x_j \in \mathsf{LFP}_{R_p}(D)$, $D'_{i,j} = D'_{j,i} = (0, \leq)$, hence $d'_{i,j} = -d'_{j,i} = 0$ and $\triangleleft_{ij'}\triangleleft_{ji'} =\leq$ and condition Definition 8 (3) holds.

.7 proof that Definition 8 (4) holds

Now we prove that Definition 8 (4) holds, *i. e.*, for all clocks x_i, x_j, x_k, valid conditions such as $D'_{i,j} \leq D'_{i,k} + D'_{k,j}$ remain valid in R_p. Indeed, when time elapses, all clocks have the same behavior, hence the difference between two clocks does not change without an update. Precisely, for all clocks x_i, x_j, x_k, are valid for R_p:

1. if $x_i, x_j, x_k \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: let $x \in \mathsf{LFP}_{R_p}(D)$ and
   - if i, j, k are different from 0, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ from Definition 8 (4), we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
   - if i, j are different from 0, $k = 0$, we have $D'_{i,0} = D_{i,x} + (1, <)$, $D'_{i,j} = D_{i,j}$ and $D'_{j,0} = D_{j,x} + (1, <)$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ from Definition 8 (4), we know that $D_{i,x} \leq D_{i,j} + D_{j,x}$ is valid for R_p; then $D_{i,x} + (1, <) \leq D_{i,j} + D_{j,x} + (1, <)$ is valid for R_p from Lemma 29 and therefore, $D'_{i,0} \leq D'_{i,j} + D'_{j,0}$ is valid for R_p.
   - if i, k are different from 0, $j = 0$, we have $D'_{i,k} = D_{i,k}$, $D'_{i,0} = D_{i,x} + (1, <)$ and $D'_{0,k} = D_{0,k} + (0, <)$; we claim that

     $$D_{i,k} \leq D_{i,x} + (1, <) + D_{0,k} + (0, <) \tag{5}$$

     is valid for R_p, which is equivalent to $D'_{i,k} \leq D'_{i,0} + D'_{0,k}$ is valid for R_p. Since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ from Definition 8 (1), we know that

     $$D_{x,0} \leq (1, <); \tag{6}$$

moreover we have

$$(1, <) + (0, <) = (1 + 0, < \oplus <) = (1, <) \tag{7}$$

Since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ from Definition 8 (4), we know that $D_{x,k} \leq D_{x,0} + D_{0,k}$ is valid for R_p; combining with (6) and (7) we obtain

$$D_{x,k} \leq (1, <) + D_{0,k} + (0, <). \tag{8}$$

Now, since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ from Definition 8 (4), we know that $D_{i,k} \leq D_{i,x} + D_{x,k}$ is valid for R_p and combining with (8) we obtain (5) and therefore our result.

- if i is different from 0, $j = k = 0$, we have $D'_{i,0} = D_{i,x} + (1, <)$; from Definition 7 (2b) we have that

$$D_{i,x} + (1, <) \leq D_{i,x} + (1, <)$$

is valid for R_p. Hence from Lemma 31

$$D'_{i,0} \leq D'_{i,0} + D'_{0,0}$$

is valid for R_p.

- if j, k are different from 0, $i = 0$, we have $D'_{0,k} = D_{0,k} + (0, <)$, $D'_{0,j} = D_{0,j} + (0, <)$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{0,k} \leq D_{0,j} + D_{j,k}$ is valid for R_p. Moreover we have that

$$D_{0,k} + (0, <) = (d_{0,k}, <) \quad \text{and} \quad D_{0,j} + (0, <) + D_{j,k} = (d_{0,j} + d_{j,k}, <)$$

so we have from Definition 7 (2b)

$$D_{0,k} + (0, <) \leq D_{0,j} + (0, <) + D_{j,k}$$

is valid for R_p. Hence $D'_{0,k} \leq D'_{0,j} + D'_{j,k}$ is valid for R_p.

- if j is different from 0, $i = k = 0$, we have $D'_{0,0} = (0, \leq)$, $D'_{0,j} = D_{0,j} + (0, <)$ and $D'_{j,0} = D_{j,x} + (1, <)$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{0,x} \leq D_{0,j} + D_{j,x}$ is valid for R_p; moreover, from Definition 7 (2b) and Lemma 29,

$$D_{0,x} + (0, <) \leq D_{0,j} + (0, <) + D_{j,x}$$

is valid for R_p. Recall that from Lemma 30 $(0, \leq) \leq D_{0,x} + D_{x,0}$ is valid for R_p and since $D_{x,0} \leq (1, <)$ from Definition 8 (1), we have

$$(0, \leq) \leq D_{0,x} + (1, <)$$

is valid for R_p. As we have $(1, <) + (0, <) = (1 + 0, < \oplus <) = (1, <)$, we obtain that

$$D_{0,x} + (1, <) \leq D_{0,j} + D_{j,x} + (1, <)$$

is valid for R_p and therefore $D'_{0,0} \leq D'_{0,j} + D'_{j,0}$ is valid for R_p.

- if k is different from 0, $i = j = 0$, we have $D'_{0,k} = D_{0,k} + (0, <)$; From Definition 7 (2b) and Lemma 29 we have that

$$D_{0,k} + (0, <) \leq D_{0,k} + (0, <)$$

is valid for R_p. Hence from Lemma 31

$$D'_{0,k} \leq D'_{0,0} + D'_{0,k}$$

is valid for R_p.

- if $i = j = k = 0$, from Definition 8 (4) and Lemma 31 we trivially have

$$D'_{0,0} \le D'_{0,0} + D'_{0,0}$$

is valid for R_p.

2. if $x_k \in \mathsf{LFP}_{R_p}(D)$ and $x_i, x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $k \neq 0$ and

- if i, j are different from 0, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ from Definition 8 (4), we know that $D_{i,k} \le D_{i,j} + D_{j,k}$; therefore, $D'_{i,k} \le D'_{i,j} + D'_{j,k}$.
- if $i \neq 0$, $j = 0$, we have $D'_{i,k} = D_{i,k}$, $D'_{i,0} = D_{i,k} + (1, <)$ and $D'_{0,k} = D_{0,k} + (0, <)$; we claim that $D_{i,k} \le D_{i,k} + (1, <) + D_{0,k} + (0, <)$ is valid for R_p, *i. e.*,

$$(0, \le) \le (1, <) + D_{0,k} + (0, <) \tag{9}$$

is valid for R_p, which is equivalent to $D'_{i,k} \le D'_{i,0} + D'_{0,k}$ is valid for R_p. We have

$$(1, <) + (0, <) = (1 + 0, < \oplus <) = (1, <). \tag{10}$$

Since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $(0, \le) \le D_{0,k} + D_{k,0}$ is valid for R_p and from Definition 8 (1) that $D_{k,0} \le (1, <)$ is valid for R_p; combining with (9) and (10) we obtain our result.
- if $i = 0$, $j \neq 0$, we have $D'_{0,k} = D_{0,k} + (0, <)$, $D'_{0,j} = D_{0,j} + (0, <)$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{0,k} \le D_{0,j} + D_{j,k}$. Moreover we have that

$$D_{0,k} + (0, <) = (d_{0,k}, <) \quad \text{and} \quad D_{0,j} + (0, <) + D_{j,k} = (d_{0,j} + d_{j,k}, <)$$

so we have from Definition 7 (2b)

$$D_{0,k} + (0, <) \le D_{0,j} + (0, <) + D_{j,k}$$

is valid for R_p. Hence $D'_{0,k} \le D'_{0,j} + D'_{j,k}$ is valid for R_p.
- if $i = j = 0$, from Definition 8 (4) and Lemma 31 we trivially have

$$D'_{0,k} \le D'_{0,0} + D'_{0,k}$$

is valid for R_p.

3. if $x_j \in \mathsf{LFP}_{R_p}(D)$ and $x_i, x_k \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $j \neq 0$ and

- if i, k are different from 0, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{i,k} \le D_{i,j} + D_{j,k}$ is valid for R_p; therefore, $D'_{i,k} \le D'_{i,j} + D'_{j,k}$ is valid for R_p.
- if $i \neq 0$, $k = 0$, we have $D'_{i,0} = D_{i,j} + (1, <)$, $D'_{i,j} = D_{i,j}$ and $D'_{j,0} = (1, <)$; From Definition 7 (2b) we trivially have that $D_{i,j} + (1, <) \le D_{i,j} + (1, <)$ is valid for R_p and therefore, $D'_{i,0} \le D'_{i,j} + D'_{j,0}$ is valid for R_p.
- if $i = 0$, $k \neq 0$, we have $D'_{0,k} = D_{0,k} + (0, <)$, $D'_{0,j} = D_{0,j} + (0, <)$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{0,k} \le D_{0,j} + D_{j,k}$ is valid for R_p. Moreover we have that

$$D_{0,k} + (0, <) = (d_{0,k}, <) \quad \text{and} \quad D_{0,j} + (0, <) + D_{j,k} = (d_{0,j} + d_{j,k}, <)$$

so we have from Definition 7 (2b) and Lemma 29

$$D_{0,k} + (0, <) \le D_{0,j} + (0, <) + D_{j,k}$$

is valid for R_p. Hence $D'_{0,k} \le D'_{0,j} + D'_{j,k}$ is valid for R_p.

- if $i = k = 0$, we have $D'_{0,0} = (0, \leq)$, $D'_{0,j} = D_{0,j} + (0,<)$ and $D'_{j,0} = (1,<)$; since $(E,D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Lemma 30 we know that $(0,\leq) \leq D_{0,j} + D_{j,0}$ is valid for R_p, and since from Definition 8 (1) $D_{j,0} \leq (1,\leq)$ is valid for R_p, that means $(0,\leq) \leq D_{0,j} + (1,<)$ is valid for R_p. As we have

$$(1,<) + (0,<) = (1+0, < \oplus <) = (1,<)$$

we obtain that

$$(0,\leq) \leq D_{0,j} + (0,<) + (1,<)$$

is valid for R_p and therefore $D'_{0,0} \leq D'_{0,j} + D'_{j,0}$ is valid for R_p.

4. if $x_j, x_k \in \mathsf{LFP}_{R_p}(D)$ and $x_i \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $j \neq 0, k \neq 0$ and

- if i is different from 0, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E,D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$.
- if $i = 0$, we have $D'_{0,k} = D_{0,k} + (0,<)$, $D'_{0,j} = D_{0,j} + (0,<)$ and $D'_{j,k} = D_{j,k}$; since $(E,D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{0,k} \leq D_{0,j} + D_{j,k}$. Moreover we have that

$$D_{0,k} + (0,<) = (d_{0,k}, <) \quad \text{and} \quad D_{0,j} + (0,<) + D_{j,k} = (d_{0,j} + d_{j,k}, <)$$

so we have from Definition 7 (2b) and Lemma 29

$$D_{0,k} + (0,<) \leq D_{0,j} + (0,<) + D_{j,k}$$

is valid for R_p. Hence $D'_{0,k} \leq D'_{0,j} + D'_{j,k}$ is valid for R_p.

5. if $x_i \in \mathsf{LFP}_{R_p}(D)$ and $x_j, x_k \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $i \neq 0$ and

- if j, k are different from 0, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E,D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$.
- if $j \neq 0$, $k = 0$, we have $D'_{i,0} = (1,<)$, $D'_{i,j} = D_{i,j}$ and $D'_{j,0} = D_{j,i} + (1,<)$; from Definition 8 (4) and Lemma 30 we know that $(0,\leq) \leq D_{i,j} + D_{j,i}$ is valid for R_p. Since, from Definition 7 (2b) $(1,<) \leq (1,<)$ is valid for R_p, then from Lemma 29

$$(1,<) \leq D_{i,j} + D_{j,i} + (1,<)$$

is valid for R_p and therefore, $D'_{i,0} \leq D'_{i,j} + D'_{j,0}$ is valid for R_p.
- if $j = 0$, $k \neq 0$, we have $D'_{i,k} = D_{i,k}$, $D'_{i,0} = (1,<)$ and $D'_{0,k} = D_{0,k} + (0,<)$; we claim that

$$D_{i,k} \leq (1,<) + D_{0,k} + (0,<)$$

is valid for R_p, which is equivalent to $D'_{i,k} \leq D'_{i,0} + D'_{0,k}$ is valid for R_p. Since $(E,D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ from Definition 8 (4), we know that $D_{i,k} \leq D_{i,0} + D_{0,k}$ is valid for R_p; moreover, from Definition 8 (1), we know that $D_{i,0} \leq (1,<)$ is valid for R_p. We have

$$(1,<) + (0,<) = (1+0, < \oplus <) = (1,<)$$

so we obtain that

$$D_{i,k} \leq D_{i,0} + D_{0,k} \leq (1,<) + D_{0,k} = (1,<) + D_{0,k} + (0,<)$$

is valid for R_p and therefore our result.

- if i is different from 0, $j = k = 0$, we have $D'_{i,0} = (1, <)$, $D'_{0,0} = (0, \leq)$; from Definition 7 (2b) we have that

$$(1, <) \leq (1, <)$$

is valid for R_p. Hence from Lemma 31

$$D'_{i,0} \leq D'_{i,0} + D'_{0,0}$$

is valid for R_p.

6. if $x_i, x_k \in \mathsf{LFP}_{R_p}(D)$ and $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $i \neq 0, k \neq 0$ and

- if $j \neq 0$, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
- if $j = 0$, we have $D'_{i,k} = D_{i,k}$, $D'_{i,0} = (1, <)$ and $D'_{0,k} = D_{0,k} + (0, <)$; we claim that

$$D_{i,k} \leq (1, <) + D_{0,k} + (0, <)$$

is valid for R_p, which is equivalent to $D'_{i,k} \leq D'_{i,0} + D'_{0,k}$ is valid for R_p. Since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ from Definition 8 (4), we know that $D_{i,k} \leq D_{i,0} + D_{0,k}$ is valid for R_p; moreover, from Definition 8 (1), we know that $D_{i,0} \leq (1, <)$ is valid for R_p. We have

$$(1, <) + (0, <) = (1 + 0, < \oplus <) = (1, <)$$

so we obtain that

$$D_{i,k} \leq D_{i,0} + D_{0,k} \leq (1, <) + D_{0,k} = (1, <) + D_{0,k} + (0, <)$$

is valid for R_p and therefore our result.

7. if $x_i, x_j \in \mathsf{LFP}_{R_p}(D)$ and $x_k \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $i \neq 0, j \neq 0$ and

- if $k \neq 0$, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
- if $k = 0$, we have $D'_{i,0} = (1, <)$, $D'_{i,j} = D_{i,j} = (0, \leq)$ since both $x_i, x_j \in \mathsf{LFP}_{R_p}(D)$ (cf.($\star$)) and $D'_{j,0} = (1, <)$; then $(1, <) \leq (0, \leq) + (1, <)$ is valid for R_p and therefore, $D'_{i,0} \leq D'_{i,j} + D'_{j,0}$ is valid for R_p.

8. if $x_i, x_j, x_k \in \mathsf{LFP}_{R_p}(D)$: i, j, k are different from 0, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.

.8 proof that Definition 8 (5b) holds

Finally, for $x_i \in \mathsf{LFP}_{R_p}(D)$, $D'_{i,0} = (1, <)$ and for all clock j s.t. $D'_{0,j} = (0, \lhd_{0j'})$, then we have $\lhd_{0j'} = <$. Condition Definition 8 (5b) is satisfied.

We denote by (E, D') the obtained p–PDBM and $(E, D') \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$.

◀

.8.1 Definition 8 type 5b to (5a)

▶ **Lemma 36.** *Let* $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$*; let* $x_i \in \mathsf{LFP}_{R_p}(D)$*,* $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$*. If* $(d_{i,j}, \lhd_{ij}) = (0, \lhd)$*, then* $\lhd = <$

Proof. Let $x_i \in \mathsf{LFP}_{R_p}(D)$, $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$. Suppose $(d_{i,j}, \lhd_{ij}) = (0, \leq)$. From Definition 8 (2) we should have that $(d_{j,i}, \lhd_{ji}) = (0, \leq)$ so Lemma 30 is satisfied, and then $x_j \in \mathsf{LFP}_{R_p}(D)$. ◀

▶ **Lemma 37** (modification of an open–p–PDBM respecting condition 5b under $TE_=$)**.** *Let* R_p *be a parameter region and* $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ *respecting condition 5b, then* $TE_=(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ *respecting condition 5a.*

Proof. Suppose $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ respects condition (5a) of Definition 8 *i. e.,* we have at least an x s.t. $D_{x,0} = (1, <)$ and for all other j s.t. $D_{0,j} = (0, \lhd_{0j})$, $\lhd_{0j} = <$. First we can determine $\mathsf{LFP}_{R_p}(D)$. Let $x \in \mathsf{LFP}_{R_p}(D)$. If more than one clock belong to $\mathsf{LFP}_{R_p}(D)$ then their valuations have the same fractional part. Indeed, from Definition 14 if $x_i, x_j \in \mathsf{LFP}_{R_p}(D)$ then both $(0, \leq) \leq D_{i,j}$ and $(0, \leq) \leq D_{j,i}$ are valid for R_p, and from Definition 8 (2) we must have $D_{i,j} = D_{j,i} = (0, \leq)$.

Let $v \in R_p$. Let $x_i \in \mathsf{LFP}_{R_p}(D)$ and $w \in (E, v(D))$. By letting time elapse, $\mathit{frac}(w(x))$ is the first to actually reach 1. Moreover, for all $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $\mathit{frac}(w(x_j))$ cannot reach 1 before $\mathit{frac}(w(x_i))$. We are going to construct a new $(E', D') = TE_=((E, D))$ which is an open–p–PDBM respecting condition 5b. While detailing the procedure of $TE_=$, we are going to prove that Definition 8 (1) and (2) hold for (E', D'). Further we will prove that (4) and (5a) also hold.

.9 proof that Definition 8 (1) holds

According to the definition of $TE_=$ (Algorithm 15), the first step is to fix the value of $\mathit{frac}(x_i)$ to 0 by setting

$$D'_{i,0} = (0, \leq) \quad \text{and} \quad D'_{0,i} = (0, \leq) \quad \text{for all } x_i \in \mathsf{LFP}_{R_p}(D).$$

Indeed, when $\mathit{frac}(x_i)$ reaches 1, in the constraints expressed by $(E, v(D))$ we have to increase the integer part by 1 and set the new constraints on the fractional part to 0.

Secondary we set new upper and lower bound for all other clock $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$

$$D'_{0,j} = D_{i,j} + (-1, \leq) \quad \text{and} \quad D'_{j,0} = D_{j,i} + (1, \leq).$$

We have to force now upper and lower bounds for other clocks since we know the interval of time that elapsed when x_i reached 1.

Note that since $x_i \in \mathsf{LFP}_{R_p}(D)$, $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$ from Definition 14 we have that $(0, \leq) \leq D_{i,j} \leq (1, <)$ is valid for R_p for all clock x_j. Nonetheless, since $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, we even have $D_{i,j} \neq (0, \leq)$: suppose $(d_{i,j}, \lhd_{ij}) = (0, \leq)$: from Definition 8 (2) we should have that $(d_{j,i}, \lhd_{ji}) = (0, \leq)$ so Lemma 30 is satisfied, and then $x_j \in \mathsf{LFP}_{R_p}(D)$. The same reasoning leads to $D_{j,i} \neq (0, \leq)$.

Obviously, we have $D_{i,j} \neq (0, <)$: suppose $D_{i,j} = (0, <)$, since $x_i \in \mathsf{LFP}_{R_p}(D)$ then from Definition 14 $(0, \leq) \leq D_{i,j}$ should be valid for R_p, which is not from Definition 7 (2b).

Precisely, $d_{i,j} \in \{1, 1 - p_1, p_2 + 1 - p_1, p_1 - p_2, p_1\}$ for any two $p_1, p_2 \in \mathbb{P}$ where $p_2 \leq p_1$ is valid for R_p. Hence as $-1 + d_{i,j} \in \{0, -p_1, p_2 - p_1, p_1 - 1 - p_2, p_1 - 1\}$, we have that $D'_{0,j} \in \mathcal{PLT}$ and $(-1, <) \leq D'_{0,j} \leq (0, \leq)$ is valid for $R_p$from Lemma 36.

Also note that since $x_i \in \mathsf{LFP}_{R_p}(D)$, from Definition 14 and Definition 8 (2) we have that $(-1,<) \leq D_{j,i} \leq (0,\leq)$ is valid for R_p for all clock x_j. Precisely, $d_{j,i} \in \{0, -p_1, p_2 - p_1, p_1 - 1 - p_2, p_1 - 1\}$ for some $p_1, p_2 \in \mathbb{P}$ where $p_2 \leq p_1$ is valid for R_p. Hence as $d_{j,i}+1 \in \{1, 1-p_1, p_2+1-p_1, p_1-p_2, p_1\}$, we have that $d'_{j,0} \in \mathcal{PLT}$ and $(0,\leq) \leq D'_{j,0} \leq (1,<)$ is valid for R_p.

Clearly Definition 8 (1) holds.

.10 proof that Definition 8 (2) holds

Third we set for all two clocks i, j where $x_i \in \mathsf{LFP}_{R_p}(D)$, $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$

$$D'_{i,j} = D'_{0,j} \quad \text{and} \quad D'_{j,i} = D'_{j,0},$$

for all two clocks $x_j, x_k \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$

$$D'_{j,k} = D_{j,k}$$

and for all two clocks $x, y \in \mathsf{LFP}_{R_p}(D)$

$$D'_{x,y} = D'_{y,x} = (0,\leq).$$

Here as we have already proven above that $(-1,<) \leq D'_{0,j} \leq (0,\leq)$ and $(0,\leq) \leq D'_{0,j} \leq (1,<)$ are valid for R_p, Definition 8 (2) holds.

.11 proof that Definition 8 (3) holds

For all x_i:

- if $x_i \in \mathsf{LFP}_{R_p}(D)$, $D'_{i,0} = (0,\leq)$, $D'_{0,i} = (0,\leq)$ hence $d'_{i,0} = -d'_{0,i}$ and $\lhd_{i0'} \lhd_{0i'} = \leq$, condition Definition 8 (3) holds;
- if $x_i \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $x \in \mathsf{LFP}_{R_p}(D)$, $D'_{i,0} = D_{i,x} + (1,\leq)$, $D'_{0,i} = D_{x,i} + (-1,\leq)$ as condition Definition 8 (3) holds for $D_{i,x}$ and $D_{x,i}$ and $\lhd_{ij} \oplus \leq = \lhd_{ij}$, $\lhd_{ji} \oplus \leq = \lhd_{ji}$, condition Definition 8 (3) holds for $D'_{i,0}$ and $D'_{0,i}$.

For all x_i, x_j:

- if $x_i, x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $D'_{i,j} = D_{i,j}$ and $D'_{j,i} = D_{j,i}$, condition Definition 8 (3) holds as it holds for $D_{i,j}$ and $D_{j,i}$.
- if $x_i \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $x_j \in \mathsf{LFP}_{R_p}(D)$, $D'_{i,j} = D_{i,j} + (1,\leq)$, $D'_{j,i} = D_{j,i} + (-1,\leq)$ condition Definition 8 (3) holds for $D_{i,j}$ and $D_{j,i}$ and $\lhd_{ij} \oplus \leq = \lhd_{ij}$, $\lhd_{ji} \oplus \leq = \lhd_{ji}$, condition Definition 8 (3) holds for $D'_{i,j}$ and $D'_{j,i}$. The case $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $x_i \in \mathsf{LFP}_{R_p}(D)$ is treated similarly.
- if $x_i, x_j \in \mathsf{LFP}_{R_p}(D)$, $D'_{i,j} = D'_{j,i} = (0,\leq)$, hence $d'_{i,j} = -d'_{j,i} = 0$ and $\lhd_{ij'} \lhd_{ji'} = \leq$ and condition Definition 8 (3) holds.

.12 proof that Definition 8 (4) holds

Now we prove that Definition 8 (4) holds, *i. e.*, for all clocks x_i, x_j, x_k, valid conditions such as $D'_{i,j} \leq D'_{i,k} + D'_{k,j}$ remain valid in R_p. This is not trivial since, in this construction some clocks have been updated. Precisely, for all clocks x_i, x_j, x_k, are valid for R_p:

1. if $x_i, x_j, x_k \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: let $x \in \mathsf{LFP}_{R_p}(D)$ and

- if i, j, k are different from 0, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
- if i, j are different from 0, $k = 0$, we have $D'_{i,0} = D_{i,x} + (1, \leq)$, $D'_{i,j} = D_{i,j}$ and $D'_{j,0} = D_{j,x} + (1, \leq)$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{i,x} \leq D_{i,j} + D_{j,x}$ is valid for R_p; then from Lemma 29 $D_{i,x} + (1, \leq) \leq D_{i,j} + D_{j,x} + (1, \leq)$ is valid for R_p and therefore, $D'_{i,0} \leq D'_{i,j} + D'_{j,0}$ is valid for R_p .
- if i, k are different from 0, $j = 0$, we have $D'_{i,k} = D_{i,k}$, $D'_{i,0} = D_{i,x} + (1, \leq)$ and $D'_{0,k} = D_{x,k} + (-1, \leq)$; we claim that

$$D_{i,k} \leq D_{i,x} + (1, \leq) + D_{x,k} + (-1, \leq) \tag{11}$$

is valid for R_p, which is equivalent to $D'_{i,k} \leq D'_{i,0} + D'_{0,k}$ is valid for R_p. We have

$$(1, \leq) + (-1, \leq) = (1 + -1, \leq \oplus \leq) = (0, \leq) \tag{12}$$

Since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ from Definition 8 (4), we know that $D_{i,k} \leq D_{i,x} + D_{x,k}$ is valid for R_p; combining with (12) and since $D_{x,k} + (0, \leq) = D_{x,k}$, we obtain (11) and therefore our result.
- if i is different from 0, $j = k = 0$, we have $D'_{i,0} = D_{i,x} + (1, \leq)$, $D'_{j,k} = D'_{0,0} = (0, \leq)$; we have from Definition 7 (2b) that

$$D_{i,x} + (1, \leq) \leq D_{i,x} + (1, \leq)$$

is valid for R_p. Hence Lemma 31 gives that

$$D'_{i,0} \leq D'_{i,0} + D'_{0,0}$$

is valid for R_p.
- if j, k are different from 0, $i = 0$, we have $D'_{0,k} = D_{x,k} + (-1, \leq)$, $D'_{0,j} = D_{x,j} + (-1, \leq)$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{x,k} \leq D_{x,j} + D_{j,k}$ is valid for R_p. Moreover we have that

$$(-1, \leq) \leq (-1, \leq)$$

is valid for R_p so we have from Definition 7 (2b) and Lemma 29

$$D_{x,k} + (-1, \leq) \leq D_{x,j} + (-1, \leq) + D_{j,k}$$

is valid for R_p. Hence $D'_{0,k} \leq D'_{0,j} + D'_{j,k}$ is valid for R_p.
- if j is different from 0, $i = k = 0$, we have $D'_{0,j} = D_{x,j} + (-1, \leq)$ and $D'_{j,0} = D_{j,x} + (1, \leq)$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Lemma 30 we know that $(0, \leq) \leq D_{x,j} + D_{j,x}$ is valid for R_p; moreover, we have that

$$(1, \leq) + (-1, \leq) = (1 + -1, \leq \oplus \leq) = (0, \leq)$$

and $D_{j,x} + (0, \leq) = D_{j,x}$. Then we have from Lemma 29

$$(0, \leq) \leq D_{x,j} + (-1, \leq) + D_{j,x} + (1, \leq)$$

is valid for R_p and therefore $D'_{0,0} \leq D'_{0,j} + D'_{j,0}$ is valid for R_p.

- if k is different from 0, $i = j = 0$, we have $D'_{0,k} = D_{x,k} + (-1, \leq)$, $D'_{i,j} = D'_{0,0} = (0, \leq)$; we have from Definition 7 (2b) that

$$D_{x,k} + (-1, \leq) \leq D_{x,k} + (-1, \leq)$$

is valid for R_p. Hence, as $D_{x,k} + (-1, \leq) + (0, \leq) = D_{x,k} + (-1, \leq)$ we have

$$D'_{0,k} \leq D'_{0,0} + D'_{0,k}$$

is valid for R_p.

- if $i = j = k = 0$, we trivially have from Definition 8 (4) and Lemma 31

$$D'_{0,0} \leq D'_{0,0} + D'_{0,0}$$

is valid for R_p.

2. if $x_k \in \mathsf{LFP}_{R_p}(D)$ and $x_i, x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $k \neq 0$ and

- if i, j are different from 0, we have $D'_{i,k} = D'_{i,0} = D_{i,k} + (1, \leq)$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D'_{j,0} = D_{j,k} + (1, \leq)$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; moreover, since we have $(1, \leq) \leq (1, \leq)$ is valid for R_p then from Lemma 29

$$D_{i,k} + (1, \leq) \leq D_{i,j} + D_{j,k} + (1, \leq)$$

is valid for R_p, therefore we have $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.

- if $i \neq 0$, $j = 0$, we have $D'_{i,k} = D'_{i,0} = D_{i,k} + (1, \leq)$, $D'_{i,0} = D_{i,k} + (1, \leq)$ and $D'_{0,k} = (0, \leq)$; clearly

$$(1, \leq) \leq (1, \leq) + (0, \leq)$$

and

$$D_{i,k} \leq D_{i,k}$$

are valid for R_p, then from Lemma 29 we obtain $D'_{i,k} \leq D'_{i,0} + D'_{0,k}$ is valid for R_p.

- if $i = 0$, $j \neq 0$, we have $D'_{0,k} = (0, \leq)$, $D'_{0,j} = D_{k,j} + (-1, \leq)$ and $D'_{j,k} = D'_{j,0} = D_{j,k} + (1, \leq)$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Lemma 30 we know that $(0, \leq) \leq D_{k,j} + D_{j,k}$ is valid for R_p. Moreover we have that

$$(1, \leq) + (-1, \leq) = (1 + -1, \leq \oplus \leq) = (0, \leq)$$

so we have from Lemma 29

$$(0, \leq) \leq D_{k,j} + D_{j,k} + (0, \leq)$$

is valid for R_p. Hence $D'_{0,k} \leq D'_{0,j} + D'_{j,k}$ is valid for R_p.

- if $i = j = 0$, we trivially have from Definition 8 (4) and Lemma 31

$$D'_{0,k} \leq D'_{0,0} + D'_{0,k}$$

is valid for R_p.

3. if $x_j \in \mathsf{LFP}_{R_p}(D)$ and $x_i, x_k \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $j \neq 0$ and

- if i, k are different from 0, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D'_{i,0} = D_{i,j} + (1, \leq)$ and $D'_{j,k} = D'_{0,k} = D_{j,k} + (-1, \leq)$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; moreover, since we have

$$(1, \leq) + (-1, \leq) = (1 + (-1), \leq \oplus \leq) = (0, \leq)$$

then as $D_{i,j} + D_{j,k} + (0, \leq) = D_{i,j} + D_{j,k}$, clearly $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
- if $i \neq 0$, $k = 0$, we have $D'_{i,0} = D_{i,j} + (1, \leq)$, $D'_{i,j} = D'_{i,0} = D_{i,j} + (1, \leq)$ and $D'_{j,0} = (0, \leq)$; From Definition 7 (2b) we trivially have that $D_{i,j} + (1, \leq) \leq D_{i,j} + (1, \leq)$ is valid for R_p and therefore, $D'_{i,0} \leq D'_{i,j} + D'_{j,0}$ is valid for R_p.
- if $i = 0$, $k \neq 0$, we have $D'_{0,k} = D_{j,k} + (-1, \leq)$, $D'_{0,j} = (0, \leq)$ and $D'_{j,k} = D'_{0,k} = D_{j,k} + (-1, \leq)$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{0,k} \leq D_{0,j} + D_{j,k}$ is valid for R_p. From Definition 7 (2b) we trivially have that $D_{j,k} + (-1, \leq) \leq D_{j,k} + (-1, \leq)$ is valid for R_p. As $(-1, \leq) + (0, \leq) = (-1, \leq)$, we have $D'_{0,k} \leq D'_{0,j} + D'_{j,k}$ is valid for R_p.
- if $i = k = 0$, we have $D'_{0,j} = (0, \leq)$ and $D'_{j,0} = (0, \leq)$; As we have

$$(0, \leq) + (0, \leq) = (0, \leq)$$

we clearly have that $D'_{0,0} \leq D'_{0,j} + D'_{j,0}$ is valid for R_p.

4. if $x_j, x_k \in \mathsf{LFP}_{R_p}(D)$ and $x_i \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $j \neq 0, k \neq 0$ and

- if i is different from 0, we have $D'_{i,k} = D'_{i,0} = D_{i,k} + (-1, \leq)$, $D'_{i,j} = D'_{i,0} = D_{i,k} + (-1, \leq)$ and $D'_{j,k} = (0, \leq)$; we have that $(-1, \leq) + (0, \leq) = (-1, \leq)$ and

$$D_{i,k} + (-1, \leq) \leq D_{i,k} + (-1, \leq)$$

holds from Definition 7 (2b). Therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$.
- if $i = 0$, we have $D'_{0,k} = (0, \leq)$, $D'_{0,j} = (0, \leq)$ and $D'_{j,k} = (0, \leq)$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ from Definition 8 (4), we know that $D_{0,k} \leq D_{0,j} + D_{j,k}$. As we have

$$(0, \leq) + (0, \leq) = (0, \leq)$$

we clearly have that $D'_{0,k} \leq D'_{0,j} + D'_{j,k}$ is valid for R_p.

5. if $x_i \in \mathsf{LFP}_{R_p}(D)$ and $x_j, x_k \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $i \neq 0$ and

- if j, k are different from 0, we have $D'_{i,k} = D'_{0,k} = D_{i,k} + (-1, \leq)$, $D'_{i,j} = D'_{0,j} = D_{i,j} + (-1, \leq)$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Definition 8 (4) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; moreover, since we have

$$(-1, \leq) \leq (-1, \leq)$$

is valid for R_p then from Lemma 29 we have $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
- if $j \neq 0$, $k = 0$, we have $D'_{i,0} = (0, \leq)$, $D'_{i,j} = D'_{0,j} = D_{i,j} + (-1, \leq)$ and $D'_{j,0} = D_{j,i} + (1, \leq)$; from Lemma 30 we know that $(0, \leq) \leq D_{i,j} + D_{j,i}$ is valid for R_p. Moreover, we have

$$(1, \leq) + (-1, \leq) = (1 + (-1), \leq \oplus \leq) = (0, \leq)$$

then

$$(0, \leq) \leq D_{i,j} + D_{j,i} + (0, \leq)$$

is valid for R_p and therefore, $D'_{i,0} \leq D'_{i,j} + D'_{j,0}$ is valid for R_p.

- if $j = 0$, $k \neq 0$, we have $D'_{i,k} = D_{i,k}$, $D'_{i,0} = (1, \leq)$ and $D'_{0,k} = D_{i,k} + (-1, \leq)$; we have that

$$(1, \leq) + (-1, \leq) = (1 + (-1), \leq \oplus \leq) = (0, \leq)$$

and from Definition 7 (2b) that

$$D_{i,k} \leq D_{i,k} + (0, \leq)$$

is valid for R_p, which gives us our result.

- if i is different from 0, $j = k = 0$, we have $D'_{i,0} = (0, \leq)$, $D'_{j,k} = D'_{0,0} = (0, \leq)$; we have from Definition 7 (2b) that

$$(0, \leq) \leq (0, \leq)$$

is valid for R_p. Hence

$$D'_{i,0} \leq D'_{i,0} + D'_{0,0}$$

is valid for R_p.

6. if $x_i, x_k \in \mathsf{LFP}_{R_p}(D)$ and $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $i \neq 0, k \neq 0$ and

- if $j \neq 0$, we have $D'_{i,k} = (0, \leq)$, $D'_{i,j} = D'_{0,j} = D_{i,j} + (-1, \leq)$ and $D'_{j,k} = D'_{j,0} = D_{j,i} + (1, \leq)$; since $(E, D) \in p\text{-}\mathcal{PDBM}_{\blacksquare}(R_p)$, from Lemma 30 we know that $(0, \leq) \leq D_{i,j} + D_{j,i}$ is valid for R_p; we have

$$(1, \leq) + (-1, \leq) = (1 + (-1), \leq \oplus \leq) = (0, \leq)$$

and therefore from Lemma 29, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.

- if $j = 0$, we have $D'_{i,k} = (0, \leq)$, $D'_{i,0} = (0, \leq)$ and $D'_{0,k} = (0, \leq)$; we have that $(0, \leq) + (0, \leq) = (0, \leq)$ and from Definition 7 (2b)

$$(0, \leq) \leq (0, \leq)$$

is valid for R_p. Therefore we obtain our result.

7. if $x_i, x_j \in \mathsf{LFP}_{R_p}(D)$ and $x_k \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $i \neq 0, j \neq 0$ and

- if $k \neq 0$, we have $D'_{i,k} = D'_{0,k} = D_{i,k} + (-1, \leq)$, $D'_{i,j} = (0, \leq)$ and $D'_{j,k} = D'_{0,k} = D_{i,k} + (-1, \leq)$; we have that

$$D_{i,k} \leq D_{i,k}$$

is valid for R_p and from Lemma 29

$$(-1, \leq) \leq (-1, \leq)$$

is valid for R_p. Therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.

- if $k = 0$, we have $D'_{i,0} = (0, \leq)$, $D'_{i,j} = (0, \leq)$ and $D'_{j,0} = (0, \leq)$; we have that $(0, \leq) + (0, \leq) = (0, \leq)$ and from Definition 7 (2b)

$$(0, \leq) \leq (0, \leq)$$

is valid for R_p: therefore $D'_{i,0} \leq D'_{i,j} + D'_{j,0}$ is valid for R_p.

8. if $x_i, x_j, x_k \in \mathsf{LFP}_{R_p}(D)$: i, j, k are different from 0, we have $D'_{i,k} = (0, \leq)$, $D'_{i,j} = (0, \leq)$ and $D'_{j,k} = (0, \leq)$; we have that $(0, \leq) + (0, \leq) = (0, \leq)$ and from Definition 7 (2b)

$$(0, \leq) \leq (0, \leq)$$

is valid for R_p: therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.

.13 proof that Definition 8 (5a) holds

Finally, there is at least one clock $x_i \in \mathsf{LFP}_{R_p}(D)$ s.t. $D_{0,i} = D_{i,0} = (0, \leq)$. Hence condition Definition 8 (5a) holds.

Finally, we set $E'_i = E_i + 1$ if $x_i \in \mathsf{LFP}_{R_p}(D)$ and $E'_j = E_j$ if $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$ We denote by (E, D') the obtained p–PDBM and $(E', D') \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$.

◀

.13.1 Definition 9 to Definition 8 type (5a)

▶ **Lemma 38** ($p\text{–}\mathcal{PDBM}_{\odot}(R_p)$ becomes $p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ after $TE_<$)**.** *Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, then $TE_<\big((E, D)\big) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ respecting condition 5b.*

Proof. Suppose $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$. Since, in R_p, we know which parameters have the largest fractional part, we can determine $\mathsf{LFP}_{R_p}(D)$ from Lemma 15. If more than one clock belong to $\mathsf{LFP}_{R_p}(D)$ then their valuations have the same fractional part.

Indeed, from Definition 14 if $x_i, x_j \in \mathsf{LFP}_{R_p}(D)$ then both $(0, \leq) \leq D_{i,j}$ and $(0, \leq) \leq D_{j,i}$ are valid for R_p, and from Definition 8 (2) we must have $D_{i,j} = D_{j,i} = (0, \leq)$.

Let $v \in R_p$. Let $x_i \in \mathsf{LFP}_{R_p}(D)$ and $w \in (E, v(D))$. By letting time elapse, $\mathit{frac}(w(x_i))$ is the first that might reach 1. Moreover, for all $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $\mathit{frac}(w(x_j))$ cannot reach 1 before $\mathit{frac}(w(x_i))$. We are going to construct a new $(E', D') = TE_<(E, D))$ which is an open–p–PDBM respecting condition 5b. While detailing the procedure of $TE_<$, we are going to prove that Definition 8 (1) and (2) hold for (E', D'). Further we will prove that (4) and (5b) also hold.

.14 proof that Definition 8 (1) holds

According to the definition of $TE_<$ (Algorithm 9), the first step is to set a new upper bound

$$D'_{i,0} = (1, <) \quad \text{for all } x_i \in \mathsf{LFP}_{R_p}(D)$$

and obviously $(0, \leq) \leq D'_{i,0} \leq (1, <)$ is valid for R_p. Then we set new upper bounds for all other clock $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$ by setting

$$D'_{j,0} = D_{j,i} + (1, <).$$

Indeed, $D_{j,i}$ is the constraint on the lower bound of $w(x_j) - w(x_i)$ and since the upper bound of x_i has increased, this gives the new upper bound of x_j. Note that since $x_i \in \mathsf{LFP}_{R_p}(D)$, from Definition 14 we have for all clock x_j that $(-1, <) \leq D_{j,i} \leq (0, \leq)$ is valid for R_p. Precisely, $d_{j,i} \in \{0, -p_1, p_2 - p_1, p_1 - 1 - p_2, p_1 - 1\}$ for some $p_1, p_2 \in \mathbb{P}$ where $p_2 \leq p_1$ is valid for R_p. Hence as $d_{j,i}+1 \in \{1, 1-p_1, p_2+1-p_1, p_1-p_2, p_1\}$, we have that $d'_{j,0} \in \mathcal{PLT}$, $\triangleleft_{j0'} = \triangleleft_{j0'} \oplus < = <$ and $(0, \leq) \leq D'_{j,0} \leq (1, <)$ is valid for R_p. Note that we cannot have $(d_{j,i}, \triangleleft_{ji}) = (-1, <)$ because even if $(d_{i,j}, \triangleleft_{ij}) = (1, <)$, since $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ we do not have have $0 \leq D_{j,i} + D_{i,j}$ is valid for R_p from Definition 8 (4) and Lemma 30.

Secondary we set for all clock x regardless of whether they are in $\mathsf{LFP}_{R_p}(D)$

$$D'_{0,x} = D_{0,x} + (0, <).$$

Since some time elapsed, lower bounds of all clocks are increased. Moreover, from Definition 9 (1) as $(-1, <) \leq D_{0,x} \leq (0, \leq)$ is valid for R_p, $(-1, <) \leq D'_{0,x} \leq (0, \leq)$ is also valid for R_p.

.15 proof that Definition 8 (2) holds

Third we set for all clocks x, y regardless of whether they are in $\mathsf{LFP}_{R_p}(D)$

$$D'_{x,y} = D_{x,y}$$

since no fractional part has reached 1, constraints on differences of clocks and integer parts remain unchanged. As it is the case in (E, D), Definition 8 (2) holds.

.16 proof that Definition 8 (3) holds

For all x_i:

- if $x_i \in \mathsf{LFP}_{R_p}(D)$, $D'_{i,0} = (1, <)$, $D'_{0,i} = D_{0,i} + (0, <)$ hence $d'_{i,0} \neq d'_{0,i}$ and $\triangleleft_{i0'} \triangleleft_{0i'} = <$, condition Definition 8 (3) holds;
- if $x_i \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $x \in \mathsf{LFP}_{R_p}(D)$, $D'_{i,0} = D_{i,x} + (1, <)$, $D'_{0,i} = D_{0,i} + (0, <)$ hence as $(0, \leq) \leq D'_{i,0}$ is valid for R_p and $D'_{0,i} \leq (0, \leq)$ is valid for R_p, we have $d'_{i,0} \neq d'_{0,i}$ and $\triangleleft_{i0'} \triangleleft_{0i'} = <$ and condition Definition 8 (3) holds.

For all x_i, x_j:

- if $x_i, x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $D'_{i,j} = D_{i,j}$ and $D'_{j,i} = D_{j,i}$, condition Definition 8 (3) holds as it holds for $D_{i,j}$ and $D_{j,i}$.
- if $x_i \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $x_j \in \mathsf{LFP}_{R_p}(D)$, $D'_{i,0} = D_{i,j} + (1, <)$, $D'_{0,i} = D_{0,i} + (0, <)$ hence as $(0, \leq) \leq D'_{i,0}$ is valid for R_p and $D'_{0,i} \leq (0, \leq)$ is valid for R_p, we have $d'_{i,0} \neq d'_{0,i}$ and $\triangleleft_{i0'} \triangleleft_{0i'} = <$, condition Definition 8 (3) holds. The case $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $x_i \in \mathsf{LFP}_{R_p}(D)$ is treated similarly.
- if $x_i, x_j \in \mathsf{LFP}_{R_p}(D)$, $D'_{i,j} = D'_{j,i} = (0, \leq)$, hence $d'_{i,j} = -d'_{j,i} = 0$ and $\triangleleft_{ij'} \triangleleft_{ji'} = \leq$ and condition Definition 8 (3) holds.

.17 proof that Definition 8 (4) holds

Now we prove that Definition 8 (4) holds, *i. e.*, for all clocks x_i, x_j, x_k valid conditions such as $D'_{i,j} \leq D'_{i,k} + D'_{k,j}$ remain valid in R_p. Indeed, when time elapses, all clocks have the same behavior, hence the difference between two clocks does not change without an update. Precisely, for all clocks x_i, x_j, x_k, are valid for R_p:

1. if $x_i, x_j, x_k \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: let $x \in \mathsf{LFP}_{R_p}(D)$ and
   - if i, j, k are different from 0, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$ from Definition 9 (2), we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
   - if i, j are different from 0, $k = 0$, we have $D'_{i,0} = D_{i,x} + (1, <)$, $D'_{i,j} = D_{i,j}$ and $D'_{j,0} = D_{j,x} + (1, <)$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $D_{i,x} \leq D_{i,j} + D_{j,x}$ is valid for R_p; then from Lemma 29 $D_{i,x} + (1, <) \leq D_{i,j} + D_{j,x} + (1, <)$ is valid for R_p and therefore, $D'_{i,0} \leq D'_{i,j} + D'_{j,0}$ is valid for R_p.
   - if i, k are different from 0, $j = 0$, we have $D'_{i,k} = D_{i,k}$, $D'_{i,0} = D_{i,x} + (1, <)$ and $D'_{0,k} = D_{0,k} + (0, <)$; we claim that

     $$D_{i,k} \leq D_{i,x} + (1, <) + D_{0,k} + (0, <) \tag{13}$$

     is valid for R_p, which is equivalent to $D'_{i,k} \leq D'_{i,0} + D'_{0,k}$ is valid for R_p. Since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (1) we know that

     $$D_{x,0} \leq (1, <) \tag{14}$$

is valid for R_p; moreover we have

$$(1,<) + (0,<) = (1+0, < \oplus <) = (1,<). \tag{15}$$

Since $(E,D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $D_{x,k} \leq D_{x,0} + D_{0,k}$ is valid for R_p; combining with (14) and (15) we obtain $D_{x,k} \leq (1,<) + D_{0,k} + (0,<)$ is valid for R_p. As $D_{i,x} \leq D_{i,x}$ is valid for R_p, using Lemma 29 we obtain

$$D_{i,x} + D_{x,k} \leq D_{i,x} + (1,<) + D_{0,k} + (0,<) \tag{16}$$

is valid for R_p. Now, since $(E,D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $D_{i,k} \leq D_{i,x} + D_{x,k}$ is valid for R_p and combining with (16) we obtain (13) and therefore our result.

- if i is different from 0, $j = k = 0$, we have $D'_{i,0} = D_{i,x} + (1,<)$, $D'_{j,k} = D'_{0,0} = (0,\leq)$; we have from Definition 7 (2b) that

$$D_{i,x} + (1,<) \leq D_{i,x} + (1,<)$$

is valid for R_p. Hence

$$D'_{i,0} \leq D'_{i,0} + D'_{0,0}$$

is valid for R_p.

- if j,k are different from 0, $i = 0$, we have $D'_{0,k} = D_{0,k} + (0,<)$, $D'_{0,j} = D_{0,j} + (0,<)$ and $D'_{j,k} = D_{j,k}$; since $(E,D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $D_{0,k} \leq D_{0,j} + D_{j,k}$ is valid for R_p. Moreover we have that

$$D_{0,k} + (0,<) = (d_{0,k},<) \quad \text{and} \quad D_{0,j} + (0,<) + D_{j,k} = (d_{0,j} + d_{j,k},<)$$

so we have from Lemma 29

$$D_{0,k} + (0,<) \leq D_{0,j} + (0,<) + D_{j,k}$$

is valid for R_p. Hence $D'_{0,k} \leq D'_{0,j} + D'_{j,k}$ is valid for R_p.

- if j is different from 0, $i = k = 0$, we have $D'_{i,k} = D'_{0,0} = (0,\leq)$, $D'_{0,j} = D_{0,j} + (0,<)$ and $D'_{j,0} = D_{j,x} + (1,<)$; since $(E,D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $D_{0,x} \leq D_{0,j} + D_{j,x}$ is valid for R_p; moreover from Lemma 29,

$$D_{0,x} + (0,<) \leq D_{0,j} + (0,<) + D_{j,x}$$

is valid for R_p. As we have

$$(1,<) + (0,<) = (1+0, < \oplus <) = (1,<)$$

we obtain from Lemma 29 that

$$D_{0,x} + (1,<) \leq D_{0,j} + D_{j,x} + (1,<)$$

is valid for R_p. Recall that from Lemma 30 $(0,\leq) \leq D_{0,x} + D_{x,0}$ is valid for R_p. Since from Definition 9 (1) $D_{x,0} \leq (1,<)$ is valid for R_p, we have $(0,\leq) \leq D_{0,x} + (1,<)$ is valid for R_p. Therefore $D'_{0,0} \leq D'_{0,j} + D'_{j,0}$ is valid for R_p.

- if k is different from 0, $i = j = 0$, we have $D'_{i,k} = D'_{j,k} = D'_{0,k} = D_{0,k} + (0,<)$, $D'_{i,j} = D'_{0,0} = (0, \leq)$; we have from Definition 7 (2b) that

  $$D_{0,k} + (0,<) \leq D_{0,k} + (0,<)$$

  is valid for R_p. Hence from Lemma 31

  $$D'_{0,k} \leq D'_{0,0} + D'_{0,k}$$

  is valid for R_p.
- if $i = j = k = 0$, we trivially have

  $$D'_{0,0} \leq D'_{0,0} + D'_{0,0}$$

  is valid for R_p.

2. if $x_k \in \mathsf{LFP}_{R_p}(D)$ and $x_i, x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $k \neq 0$ and

- if i, j are different from 0, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
- if $i \neq 0$, $j = 0$, we have $D'_{i,k} = D_{i,k}$, $D'_{i,0} = D_{i,k} + (1,<)$ and $D'_{0,k} = D_{0,k} + (0,<)$; we claim that $D_{i,k} \leq D_{i,k} + (1,<) + D_{0,k} + (0,<)$, *i. e.*,

  $$0 \leq (1,<) + D_{0,k} + (0,<) \tag{17}$$

  is valid for R_p, which is from Lemma 29 equivalent to $D'_{i,k} \leq D'_{i,0} + D'_{0,k}$ is valid for R_p. We have

  $$(1,<) + (0,<) = (1+0, < \oplus <) = (1,<). \tag{18}$$

  Since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $0 \leq D_{0,k} + D_{k,0}$ is valid for R_p and from Definition 9 (1) that $D_{k,0} \leq (1,<)$ is valid for R_p; combining with (18) we obtain (17) and therefore our result.
- if $i = 0$, $j \neq 0$, we have $D'_{0,k} = D_{0,k} + (0,<)$, $D'_{0,j} = D_{0,j} + (0,<)$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $D_{0,k} \leq D_{0,j} + D_{j,k}$ is valid for R_p. Moreover we have that $(0,<) \leq (0,<)$ is valid for R_p so we have from Lemma 29

  $$D_{0,k} + (0,<) \leq D_{0,j} + (0,<) + D_{j,k}$$

  is valid for R_p. Hence $D'_{0,k} \leq D'_{0,j} + D'_{j,k}$ is valid for R_p.
- if $i = j = 0$, from Definition 9 (2) we trivially have

  $$D'_{0,k} \leq D'_{0,0} + D'_{0,k}$$

  is valid for R_p.

3. if $x_j \in \mathsf{LFP}_{R_p}(D)$ and $x_i, x_k \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $j \neq 0$ and

- if i, k are different from 0, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
- if $i \neq 0$, $k = 0$, we have $D'_{i,0} = D_{i,j} + (1,<)$, $D'_{i,j} = D_{i,j}$ and $D'_{j,0} = (1,<)$; from Definition 7 (2b) we trivially have that $D_{i,j} + (1,<) \leq D_{i,j} + (1,<)$ is valid for R_p and therefore, $D'_{i,0} \leq D'_{i,j} + D'_{j,0}$ is valid for R_p.

- if $i = 0$, $k \neq 0$, we have $D'_{0,k} = D_{0,k} + (0,<)$, $D'_{0,j} = D_{0,j} + (0,<)$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $D_{0,k} \leq D_{0,j} + D_{j,k}$ is valid for R_p. Moreover we have that $(0,<) \leq (0,<)$ is valid for R_p so we have

$$D_{0,k} + (0,<) \leq D_{0,j} + (0,<) + D_{j,k}$$

holds from Definition 7 (2b). Hence $D'_{0,k} \leq D'_{0,j} + D'_{j,k}$ is valid for R_p.
- if $i = k = 0$, we have $D'_{0,j} = D_{0,j} + (0,<)$ and $D'_{j,0} = (1,<)$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Lemma 30 we know that $0 \leq D_{0,j} + D_{j,0}$ is valid for R_p, from Definition 9 (1) we know that $D_{j,0} \leq 1$ is valid for R_p which means $0 \leq D_{0,j} + (1,<)$ is valid for R_p. As we have

$$(1,<) + (0,<) = (1 + 0, < \oplus <) = (1,<)$$

we obtain that

$$(0,\leq) \leq D_{0,j} + (0,<) + (1,<)$$

is valid for R_p and therefore $D'_{0,0} \leq D'_{0,j} + D'_{j,0}$ is valid for R_p.

4. if $x_j, x_k \in \mathsf{LFP}_{R_p}(D)$ and $x_i \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $j \neq 0, k \neq 0$ and
- if i is different from 0, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
- if $i = 0$, we have $D'_{0,k} = D_{0,k} + (0,<)$, $D'_{0,j} = D_{0,j} + (0,<)$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $D_{0,k} \leq D_{0,j} + D_{j,k}$ is valid for R_p. Moreover we have that $(0,<) \leq (0,<)$ is valid for R_p so we have from Lemma 29

$$D_{0,k} + (0,<) \leq D_{0,j} + (0,<) + D_{j,k}$$

is valid for R_p. Hence $D'_{0,k} \leq D'_{0,j} + D'_{j,k}$ is valid for R_p.

5. if $x_i \in \mathsf{LFP}_{R_p}(D)$ and $x_j, x_k \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $i \neq 0$ and
- if j, k are different from 0, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
- if $j \neq 0$, $k = 0$, we have $D'_{i,0} = (1,<)$, $D'_{i,j} = D_{i,j}$ and $D'_{j,0} = D_{j,i} + (1,<)$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Lemma 30 we know that $0 \leq D_{i,j} + D_{j,i}$. Then from Lemma 29

$$(1,<) \leq D_{i,j} + D_{j,i} + (1,<)$$

is valid for R_p and therefore, $D'_{i,0} \leq D'_{i,j} + D'_{j,0}$ is valid for R_p.
- if $j = 0$, $k \neq 0$, we have $D'_{i,k} = D_{i,k}$, $D'_{i,0} = (1,<)$ and $D'_{0,k} = D_{0,k} + (0,<)$; we claim that

$$D_{i,k} \leq (1,<) + D_{0,k} + (0,<)$$

is valid for R_p, which is equivalent to $D'_{i,k} \leq D'_{i,0} + D'_{0,k}$ is valid for R_p. Since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$ from Definition 9 (2), we know that $D_{i,k} \leq D_{i,0} + D_{0,k}$ is valid for R_p; moreover, from Definition 9 (1), we know that $D_{i,0} \leq (1,<)$ is valid for R_p. We have

$$(1,<) + (0,<) = (1 + 0, < \oplus <) = (1,<)$$

We obtain that

$$D_{i,k} \leq D_{i,0} + D_{0,k} \leq (1,<) + D_{0,k} = (1,<) + D_{0,k} + (0,<)$$

is valid for R_p and therefore our result.

- if i is different from 0, $j = k = 0$, we have $D'_{i,0} = (1,<)$, $D'_{j,k} = D'_{0,0} = (0,\leq)$; from Definition 7 (2b) we have that

$$(1,<) \leq (1,<)$$

is valid for R_p. Hence

$$D'_{i,0} \leq D'_{i,0} + D'_{0,0}$$

is valid for R_p.

6. if $x_i, x_k \in \mathsf{LFP}_{R_p}(D)$ and $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $i \neq 0, k \neq 0$ and
   - if $j \neq 0$, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
   - if $j = 0$, we have $D'_{i,k} = D_{i,k}$, $D'_{i,0} = (1,<)$ and $D'_{0,k} = D_{0,k} + (0,<)$; we claim that

$$D_{i,k} \leq (1,<) + D_{0,k} + (0,<)$$

is valid for R_p, which is equivalent to $D'_{i,k} \leq D'_{i,0} + D'_{0,k}$ is valid for R_p. Since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$ from Definition 9 (2), we know that $D_{i,k} \leq D_{i,0} + D_{0,k}$ is valid for R_p; moreover, from Definition 9 (1), we know that $D_{i,0} \leq (1,<)$ is valid for R_p. We have

$$(1,<) + (0,<) = (1+0, < \oplus <) = (1,<)$$

We obtain that

$$D_{i,k} \leq D_{i,0} + D_{0,k} \leq (1,<) + D_{0,k} = (1,<) + D_{0,k} + (0,<)$$

is valid for R_p and therefore our result.

7. if $x_i, x_j \in \mathsf{LFP}_{R_p}(D)$ and $x_k \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: $i \neq 0, j \neq 0$ and
   - if $k \neq 0$, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2), we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.
   - if $k = 0$, since both $x_i, x_j \in \mathsf{LFP}_{R_p}(D)$ we have $D'_{i,j} = D_{i,j} = (0,\leq)$, $D'_{i,0} = (1,<)$ and $D'_{j,0} = (1,<)$; trivially $(1,<) \leq (0,\leq) + (1,<)$ is valid for R_p and therefore, $D'_{i,0} \leq D'_{i,j} + D'_{j,0}$ is valid for R_p.
8. if $x_i, x_j, x_k \in \mathsf{LFP}_{R_p}(D)$: i, j, k are different from 0, we have $D'_{i,k} = D_{i,k}$, $D'_{i,j} = D_{i,j}$ and $D'_{j,k} = D_{j,k}$; since $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$, from Definition 9 (2) we know that $D_{i,k} \leq D_{i,j} + D_{j,k}$ is valid for R_p; therefore, $D'_{i,k} \leq D'_{i,j} + D'_{j,k}$ is valid for R_p.

.18 proof that Definition 8 (5b) holds

Finally, for $x_i \in \mathsf{LFP}_{R_p}(D)$, $D'_{i,0} = (1,<)$ and for all clock j s.t. $D'_{0,j} = (0,\triangleleft)$, then we have $\triangleleft = <$. Condition Definition 8 (5b) is satisfied.

We set $E' = E$ and denote by (E, D') the obtained p–PDBM, which is $(E, D') \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$.

◀

.19 Proof of Proposition 18

Proposition 18 (recalled). *Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$. Let $v \in R_p$. There exists $w' \in TE((E, v(D)))$ iff there exist $w \in (E, v(D))$ and a delay δ s.t. $w' = w + \delta$.*

Proof. Note that this proof is inspired by [HRSV02, Proof of Lemma 3.13]

▶ **Lemma 39.** *Let $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$. If (E, D) satisfies condition Definition 8 (5b) it has been obtained after applying Algorithm 9 on another* open*–*p*–PDBM satisfying condition Definition 8 (5a) or a* point*–*p*–PDBM.*

Let $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$. If (E, D) satisfies condition Definition 8 (5a) it has been obtained after applying Algorithm 15 on another open*–*p*–PDBM satisfying condition Definition 8 (5b) or after a non-parametric update applied on another* open*–*p*–PDBM or a* point*–*p*–PDBM.*

Proof. Let $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ and suppose (E, D) satisfies condition Definition 8 (5b). Since for all y, if $d_{0,y} = 0$ we have $\lhd_{0y} = <$, from Lemma 32 and Lemma 33 it cannot be the result of a non-parametric update where there is at least a clock x update and $D_{x,0} = D_{0,x} = (0, \leq)$. From Lemma 37 it cannot be the result of Algorithm 15, as there must be at least a clock x s.t. $D_{x,0} = D_{0,x} = (0, \leq)$. Then it is the result either from Lemma 35 of Algorithm 9 applied on an open–p–PDBM satisfying condition Definition 8 (5a), or from Lemma 38 of Algorithm 9 applied on a point–p–PDBM.

Let $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ and suppose (E, D) satisfies condition Definition 8 (5a). Since there is at least a clock y s.t. $D_{y,0} = D_{0,y} = (0, \leq)$, from Lemma 35 and Lemma 38 it cannot be the result of Algorithm 9, as for all x, if $d_{0,x} = 0$ we must have $\lhd_{ox} = <$. Then it is the result of either from Lemma 37 of Algorithm 15 applied on an open–p–PDBM satisfying condition Definition 8 (5b) or from Lemma 32 and Lemma 33 of Algorithm 1 applied on an open–p–PDBM or a point–p–PDBM. ◀

Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$. We have to consider two different cases: $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$ and $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$.

▶ **Lemma 40.** *Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$. Let $v \in R_p$. There is $w' \in TE((E, v(D)))$ iff there is $w \in (E, v(D))$ and a delay δ s.t. $w' = w + \delta$.*

Proof. Let R_p a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}_{\blacksquare}(R_p)$. Let $v \in R_p$.

$\Longrightarrow$ .19.0.1 open–p–PDBM respecting Definition 8 (5a)

Let $v \in R_p$. Consider $(E', D') = TE((E, D))$ respecting condition Definition 8 (5a), *i. e.*, suppose there is x_i s.t. $D'_{i,0} = -D'_{0,i} = (0, \leq)$. Let $w' \in (E', v(D'))$, for this x_i we have $w'(x_i) = 0$. We need to find a value δ s.t. $w' - \delta \in (E, v(D))$ which is equivalent to prove for all x_i, x_j

$$frac(w'(x_j)) - \delta - frac(w'(x_i)) + \delta \lhd_{ji} v(d_{j,i})$$

and

$$frac(w'(x_i)) - \delta - frac(w'(x_j)) + \delta \lhd_{ij} v(d_{i,j})$$

and

$$-frac(w'(x_j)) + \delta \lhd_{0j} v(d_{0,j}) \quad \text{and} \quad frac(w'(x_j)) - \delta \lhd_{j0} v(d_{j,0}).$$

In this proof we are going to define a δ which is different from 0, and give it an upper bound in order to show that constraints in (E, D) are satisfied while going backward of δ units of time from w'.
First we will prove that for all clock j, its constraints of lower bound $D_{0,j}$ and upper bound $D_{j,0}$ are satisfied. Second we will prove that for all i, bounds on their difference $D_{i,j}$ and $D_{j,i}$ are also satisfied.
We want to show that we have to go a little backward in time from w' to ensure the upper bounds $D_{j,0}$ of (E, D) hold. For this purpose, we are going to prove that for all x_j

$$D_{j,0} \leq D'_{j,0}$$

is valid for R_p. Intuitively this means upper bounds of clocks in (E', D') are greater than in (E, D), which is consistent as time is elapsing.
As (E', D') respects Definition 8 (5a) and precisely $(E', D') = \mathit{TE}_{=}((E, D))$, we know (E, D) is respecting condition Definition 8 (5b) from Lemma 35. As $\mathit{frac}(w'(x_i)) = 0$ it was in (E, D) a clock with the largest fractional part, *i. e.*, $x_i \in \mathsf{LFP}_{R_p}(D)$ and $D_{i,0} = (1, <)$.
By definition of $\mathit{TE}_{<}$ (cf. Algorithm 9), in (E, D) which is the open–p–PDBM obtained after the application of $\mathit{TE}_{<}$ on another p–PDBM (see Lemma 39), for each $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $D_{j,0} = D_{j,i} + (1, <)$ and for all $x_j \in \mathbb{X}$, we have $D_{j,0}$ is of the form $(d_{j,0}, <)$ for some $d_{j,0}$.
By definition of $\mathit{TE}_{=}$ applied to (E, D) (cf. Algorithm 15), in (E', D'), for each $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $D'_{j,0} = D_{j,i} + (1, \leq)$, *i. e.*, $d_{j,0} = d'_{j,0}$. Hence by Definition 7 (2b) and as $\triangleleft_{j0'}$ is either $\leq$ or $<$, we have

$$(d_{j,0}, <) = D_{j,0} \leq D'_{j,0} = (d_{j,0}, \triangleleft_{j0'})$$

is valid for R_p. Next we define the largest amount of time so that all upper bounds of (E, D) are satisfied.
We claim that for all x_j, $\mathit{frac}(w'(x_j)) - v(d_{j,0}) \leq 0$. Indeed, remark that by applying Algorithm 9 then Algorithm 15, constraints on upper bounds of clocks in (E, D) and (E', D') differ only by their $\triangleleft$. As for $i \in \mathsf{LFP}_{R_p}(D)$ and $j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$ it we have $D_{j,0} = D_{j,i} + (1, <)$ in (E, D) and $D'_{j,0} = D_{j,i} + (1, \leq)$ in (E', D'), so $d_{j,0} = d'_{j,0}$. Since for any x, its fractional part is less or equal to its upper bound in D and therefore in D', any difference between a fractional part and its upper bound is either negative or null. For all x, since $\mathit{frac}(w'(x)) \triangleleft_{x0'} v(d'_{x,0})$ we have $\mathit{frac}(w'(x)) - v(d'_{x,0}) \triangleleft_{x0'} 0$. Since $v(d'_{x,0}) = v(d_{x,0})$, $\mathit{frac}(w'(x)) - v(d_{x,0}) \triangleleft_{x0'} 0$, therefore we have our result.
Now we claim that we have to go at least an $\epsilon > 0$ backward in time to ensure all bounds of (E, D) are met. Let $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$. As

$$\mathit{frac}(w'(x_j)) \triangleleft_{j0'} v(d_{j,0})$$

we have

- either $\triangleleft_{j0'} = <$ and we already have $\mathit{frac}(w'(x_j)) < v(d_{j,0})$,
- or $\triangleleft_{j0'} = \leq$ and for any $\epsilon > 0$ we have $\mathit{frac}(w'(x_j)) - \epsilon < v(d_{j,0})$.

It is also true for each $x_i \in \mathsf{LFP}_{R_p}(D)$: after applying $\mathit{TE}_{<}$ recall that we have $D_{i,0} = (1, <)$. We can take $\epsilon > 0$ and define $\mathit{frac}(w(x_i)) = 1 - \epsilon$, so we have $\mathit{frac}(w(x_i)) < v(d_{i,0})$.
Now that we know we have to go a little backward in time (at least an $\epsilon > 0$) so upper bounds of (E, D) are satisfied, we are going to give an upper bound to ϵ so that all lower bounds $D_{0,j}$ of (E, D) are also satisfied.

Let

$$t_1 = \min_{x \in \mathbb{X}} \{frac(w'(x)) + v(d_{0,x})\}$$

We want to prove that $t_1 > 0$.
Let us prove that for all x_j, $D'_{0,j} \leq D_{0,j}$ is valid for R_p. Recall that for $x_i \in \mathsf{LFP}_{R_p}(D)$, we have that $D_{i,0} = (1, <)$. Moreover, from Definition 8 (4) $D_{i,j} \leq D_{i,0} + D_{0,j}$ is valid for R_p, then we have

$$D_{i,j} \leq (1, <) + D_{0,j}$$

is valid for R_p. Recall that after applying Algorithm 15, $D'_{0,j} = D_{i,j} + (-1, \leq)$. By Definition 7 (2b) we have $(-1, \leq) \leq (-1, \leq)$. We invoke Lemma 29 which gives

$$D_{i,j} + (-1, \leq) \leq (1, <) + D_{0,j} + (-1, \leq) = D_{0,j} + (0, <) \text{ is valid for } R_p. \tag{19}$$

As, from Definition 7 (2b) we have $D_{0,j} + (0, <) \leq D_{0,j}$ is valid for R_p, we infer (19) and it gives

$$D'_{0,j} \leq D_{0,j} \text{ is valid for } R_p.$$

Since $w' \in (E', v(D'))$ we have $-frac(w'(x_j)) \lhd_{0j'} v(d'_{0,j})$,

$$0 \lhd_{0j'} frac(w'(x_j)) + v(d'_{0,j}).$$

Then we have that

$$0 \lhd_{0j'} frac(w'(x_j)) + v(d'_{0,j}) \leq frac(w'(x_j)) + v(d_{0,j})$$

where ,

- either from Definition 7 (2a) $d'_{0,j} < d_{0,j}$;
- or from Definition 7 (2b), $d'_{0,j} \leq d_{0,j}$ and then $\lhd_{0j'} = \lhd_{0j} = <$. Indeed as $D'_{0,j} \leq D_{0,j}$ is valid for R_p, and since (E, D) is the open–p–PDBM obtained after the application of $TE_<$ (cf. Algorithm 9) on another p–PDBM (see Lemma 39), we have $\lhd_{0j} = <$.

To conclude we have that for all x_j either

$$0 \lhd_{0j'} frac(w'(x_j)) + v(d'_{0,j}) < frac(w'(x_j)) + v(d_{0,j})$$

or

$$0 < frac(w'(x_j)) + v(d'_{0,j}) \leq frac(w'(x_j)) + v(d_{0,j}).$$

As t_1 is by definition the minimum value of an expression $frac(w'(x_j)) + v(d_{0,j})$ for a given x_j, which as we just proved are all strictly positive, we have that for all x_j

$$0 < t_1 \leq frac(w'(x_j)) + v(d_{0,j}).$$

We proved that $t_1 > 0$, so we can set $\delta = \frac{t_1}{2}$ (therefore $\delta > 0$).
More intuitively δ is the value right in the middle of the least and the largest amount of time s.t. we can go backward in time from w' and respect all constraints defined in $(E, v(D))$.

Now we are going to prove that for any clock x_j, its constraints on lower and upper bounds are satisfied, *i. e.,*

$$-v(d_{0,j}) \lhd_{0j} \mathit{frac}(w'(x_j)) - \delta \lhd_{j0} v(d_{j,0}).$$

First as $\delta < t_1$, we have

$$-\mathit{frac}(w'(x_j))+\delta < -\mathit{frac}(w'(x_j))+t_1 \leq -\mathit{frac}(w'(x_j))+\mathit{frac}(w'(x_j))+v(d_{0,j}) = v(d_{0,j})$$

which is $-v(d_{0,j}) < \mathit{frac}(w'(x_j)) - \delta$. Since (E, D) is the open–p–PDBM obtained after the application of $TE_<$ (cf. Algorithm 15) on another p–PDBM (see Lemma 39), we have $\lhd_{0j} = <$ so $-v(d_{0,j}) \lhd_{0j} \mathit{frac}(w'(x_j)) - \delta$. Secondary as $0 < \delta$, we have

$$\mathit{frac}(w'(x_j)) - \delta < \mathit{frac}(w)'(x_j) - 0 \leq \mathit{frac}(w'(x_j)) - \mathit{frac}(w'(x_j)) + v(d_{j,0}) = v(d_{j,0})$$

which is $\mathit{frac}(w'(x_j)) - \delta < v(d_{j,0})$. Since (E, D) is the open–p–PDBM obtained after the application of $TE_<$ (cf. Algorithm 15) on another p–PDBM (see Lemma 39), we have $\lhd_{j0} = <$ so $\mathit{frac}(w'(x_j)) - \delta \lhd_{j0} v(d_{j,0})$

Now we prove that constraints defined in (E, D) on differences of clocks are also satisfied by going back of δ units of time from w'.

Recall that in (E', D') we have for all clock x_j,

$$D'_{j,i} = D'_{j,0} = D_{j,i} + 1 \quad \text{and} \quad D'_{i,j} = D'_{0,j} = -1 + D_{i,j}.$$

In addition by definition of $TE_=$, for $x_i \in \mathsf{LFP}_{R_p}(D)$, $E_{x_i} = E'_{x_i} - 1$ and for $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$, $E_{x_j} = E'_{x_j}$.

We already treated the case whether i or j are 0, now suppose i, j are both different from 0.

- if $x_i, x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: let $x \in \mathsf{LFP}_{R_p}(D)$ and recall that after applying Algorithm 15, $D'_{i,j} = D_{i,j}$, $D'_{j,i} = D_{j,i}$; we have that $\mathit{frac}(w'(x_j)) - \mathit{frac}(w'(x_i)) \lhd_{ij'} d'_{j,i} = d_{j,i}$, and therefore $\mathit{frac}(w'(x_j)) - \delta - \mathit{frac}(w'(x_i)) + \delta \lhd_{ji} d_{j,i}$.
  We also have that $\mathit{frac}(w'(x_i)) - \mathit{frac}(w'(x_j)) \lhd_{ij'} d'_{i,j} = d_{i,j}$, therefore $\mathit{frac}(w'(x_i)) - \delta - \mathit{frac}(w'(x_j)) + \delta \lhd_{ij} d_{i,j}$;
- if $x_i \in \mathsf{LFP}_{R_p}(D)$ and $x_j \in \mathbb{X} \setminus \mathsf{LFP}_{R_p}(D)$: recall that after applying Algorithm 15, $D'_{j,0} = D_{j,i} + (1, \leq)$, and $D'_{0,j} = D_{i,j} + (-1, \leq)$. Observe that as we added $\leq$ which is the neutral element of the addition $\oplus$ between two operators $\lhd$, we have $\lhd_{j0'} = \lhd_{ji}$ and $\lhd_{0j'} = \lhd_{ij}$. Note that as $x_i \in \mathsf{LFP}_{R_p}(D)$, in (E', D') we have $D'_{0,i} = (0, \leq) = D'_{i,0}$ which means $\mathit{frac}(w'(x_i)) = 0$. Going backward in time of δ units of time from $w'(x_i)$ means that $\mathit{frac}(w(x_i)) = 1 - \delta$.
  We have that

  $$\mathit{frac}(w'(x_j)) \lhd_{j0'} v(d'_{j,0}) = v(d_{j,i}) + 1$$

  hence $\mathit{frac}(w'(x_j)) - 1 \lhd_{ji} v(d_{j,i})$ which is equivalent to

  $$\mathit{frac}(w'(x_j)) - \delta - 1 + \delta \lhd_{ji} v(d_{j,i}).$$

  The same way we have

  $$-\mathit{frac}(w'(x_j)) \lhd_{0j'} v(d'_{0,j}) = v(d_{i,j}) - 1$$

  hence $1 - \mathit{frac}(w'(x_j)) \lhd_{ij} v(d_{i,j})$ which is equivalent to

  $$1 - \delta - \mathit{frac}(w'(x_j)) + \delta \lhd_{ij} v(d_{i,j}).$$

To conclude, we define for all x_j s.t. $D'_{0,j} \neq (0, \leq)$ and $D'_{j,0} \neq (0, \leq)$

$$w(x_j) = w'(x_j) - \delta$$

and for all x_i s.t. $D'_{0,i} = (0, \leq) = D'_{i,0}$

$$w(x_i) = (w'(x_i) - 1) + 1 - \delta$$

and clearly, $w \in (E, v(D))$.

$\Longrightarrow$ .19.0.2 open–p–PDBM respecting Definition 8 (5b)

Let $v \in R_p$. Consider $(E', D') = TE((E, D))$ respecting condition Definition 8 (5b), *i. e.,* suppose there is at least an x_i s.t. $D'_{i,0} = (1, <)$ and for all j s.t. $D_{0,j} = (0, \triangleleft_{0j})$, then we have $\triangleleft_{0j} = <$. Let $w' \in (E', v(D'))$.
We need to find a value δ s.t. $w' - \delta \in (E, v(D))$ which is equivalent to prove for all x_i, x_j

$$\mathit{frac}(w'(x_j)) - \delta - \mathit{frac}(w'(x_i)) + \delta \triangleleft_{ji} v(d_{j,i})$$

and

$$\mathit{frac}(w'(x_i)) - \delta - \mathit{frac}(w'(x_j)) + \delta \triangleleft_{ij} v(d_{i,j})$$

and

$$-\mathit{frac}(w'(x_j)) + \delta \triangleleft_{0j} v(d_{0,j}) \quad \text{and} \quad \mathit{frac}(w'(x_j)) - \delta \triangleleft_{j0} v(d_{j,0}).$$

As done previously we are going to define a δ which is different from 0 so we satisfy condition Definition 8 (5a), and show that constraints in (E, D) are satisfied while going backward of δ units of time from w'.
We define the largest and the least amount of time so that all upper bounds of (E, D) are satisfied. Let

$$t_0 = \max_{x \in \mathbb{X}} \{0, \mathit{frac}(w'(x)) - v(d_{x,0})\}$$

and

$$t_1 = \min_{x \in \mathbb{X}} \{\mathit{frac}(w'(x)) + v(d_{0,x})\}.$$

We want to prove that $t_0 = t_1 > 0$. For this purpose, let us first show that for all i, j we have $\mathit{frac}(w'(x_j)) - v(d'_{j,0}) \leq \mathit{frac}(w'(x_i)) + v(d'_{0,i})$, which is $t_0 \leq t_1$.
First note that for all i, j

$$\mathit{frac}(w'(x_j)) - \mathit{frac}(w'(x_i)) \triangleleft_{ji'} v(d'_{j,i}).$$

By applying $TE_<$ (Algorithm 9) to (E, D), we have that $D'_{j,i} = D_{j,i}$, *i. e.,* $(d_{i,j}, \triangleleft_{ij}) = (d'_{i,j}, \triangleleft_{ij'})$, and from Definition 8 (4) we have that $D_{j,i} \leq D_{j,0} + D_{0,i}$ is valid for R_p.
Hence, we have from Definition 7 (2b) that either $v(d_{j,i}) < v(d_{j,0}) + v(d_{0,i})$ or $v(d_{j,i}) \leq v(d_{j,0}) + v(d_{0,i})$ and $\triangleleft_{ji} = \triangleleft_{j0} \oplus \triangleleft_{0i}$ or $\triangleleft_{ji} = <$ and $\triangleleft_{j0} \oplus \triangleleft_{0i} = \leq$.
We can then write that

$$\mathit{frac}(w'(x_j)) - \mathit{frac}(w'(x_i))(\triangleleft_{j0} \oplus \triangleleft_{0i}) v(d_{j,0}) + v(d_{0,i})$$

which is equivalent to

$$frac(w'(x_j)) - v(d_{j,0})(\lhd_{j0} \oplus \lhd_{0i})frac(w'(x_i)) + v(d_{0,i})$$

so we obtain our result, as $(\lhd_{j0} \oplus \lhd_{0i})$ is either $\leq$ or $<$.
Now, recall that (E, D) respects condition Definition 8 (5a) so we have at least an x s.t. $D_{x,0} = D_{0,x} = (0, \leq)$.
For this clock x we have that $frac(w'(x)) = frac(w'(x)) - v(d_{x,0}) \leq t_0$ and that $t_1 \leq frac(w'(x)) + v(d_{0,x}) = frac(w'(x))$.
Hence $t_0 = t_1 = frac(w'(x))$.
As $\lhd_{x0} = \leq$, we have $(\lhd_{x0} \oplus \lhd_{0i}) = \lhd_{0i}$ and $(\lhd_{j0} \oplus \lhd_{0x}) = \lhd_{j0}$, which gives

$$frac(w'(x)) = frac(w'(x)) - v(d_{x,0}) \lhd_{0i} frac(w'(x_i)) + v(d_{0,i})$$

and

$$frac(w'(x_j)) - v(d_{j,0}) \lhd_{j0} frac(w'(x)) + v(d_{0,x}) = frac(w'(x)).$$

Moreover in (E', D') we have that $frac(w'(x)) \lhd_{0x'} v(d'_{0,x})$. Since (E', D') respects condition Definition 8 (5b), if $D'_{0,x} = (0, \lhd_{0x'})$ then $\lhd_{0x'} = <$. Hence $0 < frac(w'(x))$ and

$$0 < t_0 = t_1.$$

Let $\delta = t_0 = t_1$. More intuitively δ is the value right in the middle of the least and the largest amount of time s.t. we can go backward in time from w' and respect all constraints defined in $(E, v(D))$.
First we have

$$-frac(w'(x_j)) + \delta \leq -frac(w'(x_j)) + t_1 \lhd_{j0} - frac(w'(x_j)) + frac(w'(x_j)) + v(d_{0,j}) = v(d_{0,j})$$

which is $-v(d_{0,j}) \lhd_{j0} frac(w'(x_j)) - \delta$.
Secondary we have

$$frac(w'(x_j)) - \delta \leq frac(w)'(x_j) - t_0 \lhd_{0j} frac(w'(x_j)) - frac(w'(x_j)) + v(d_{j,0}) = v(d_{j,0})$$

which is $frac(w'(x_j)) - \delta \lhd_{0j} v(d_{j,0})$.
Now we prove that constraints defined in (E, D) on differences of clocks are also satisfied by going back of δ units of time from w'
Recall that in (E', D') from the definition of Algorithm 9 we have for all clocks x_i, x_j,

$$D'_{j,i} = D_{j,i} \quad \text{and} \quad D'_{i,j} = D_{i,j}.$$

Since we already treated the case whether i or j are 0, now suppose i, j are both different from 0. We have that $frac(w'(x_j)) - frac(w'(x_i)) \lhd_{ji'} v(d'_{j,i}) = v(d_{j,i})$, and therefore as $\lhd_{ji'} = \lhd_{ji}$,

$$frac(w'(x_j)) - \delta - frac(w'(x_i)) + \delta \lhd_{ji} v(d_{j,i}).$$

We also have that $frac(w'(x_i)) - frac(w'(x_j)) \lhd_{ij'} v(d'_{i,j}) = v(d_{i,j})$, therefore as $\lhd_{ij'} = \lhd_{ij}$,

$$frac(w'(x_i)) - \delta - frac(w'(x_j)) + \delta \lhd_{ij} v(d_{i,j}).$$

To conclude, we define for all x_j

$$w(x_j) = w'(x_j) - \delta$$

and clearly, $w \in (E, v(D))$.

Conversely, let $w \in (E, v(D))$,

$\Longleftarrow$ .19.0.3 open–p–PDBM respecting Definition 8 (5b)

Suppose in (E, D) there is at least an x_i s.t. $D_{i,0} = (1, <)$ and for all j s.t. $D_{0,j} = (0, \lhd)$, we have $\lhd = <$. Let x_i be such a clock and $v \in R_p$.
Now consider $(E', D') = TE((E, D))$. We need to find a value δ s.t. $w + \delta \in (E', v(D'))$. which is equivalent to prove for all x_i, x_j

$$frac(w(x_j)) + \delta - frac(w(x_i)) - \delta \lhd_{ji'} v(d'_{j,i})$$

and

$$frac(w(x_i)) + \delta - frac(w(x_j)) - \delta \lhd_{ij'} v(d'_{i,j})$$

and

$$-frac(w(x_j)) - \delta \lhd_{0j'} v(d'_{0,j}) \quad \text{and} \quad frac(w(x_j)) + \delta \lhd_{j0'} v(d'_{j,0}).$$

As done previously we are going to define a δ which is different from 0, and show that constraints in (E, D) are satisfied while going forward of δ units of time from w.
Recall that $x_i \in \mathsf{LFP}_{R_p}(D)$ and let $\delta = 1 - frac(w(x_i)$ which we will prove is the exact amount of time so that all upper bounds of (E', D') are satisfied. Let

$$t_0 = \max_{x \in \mathbb{X}} \{-frac(w(x)) - frac(v(d'_{0,x}))\}$$

and

$$t_1 = \min_{x \in \mathbb{X}} \{frac(v(d'_{x,0})) - frac(w(x))\}.$$

Recall that since (E, D) respects condition Definition 8 (5b), for all j s.t. $D_{0,j} = (0, \lhd_{0j})$, we have $\lhd_{0j} = <$. Hence as $-frac(w(x_i) < v(d_{0,j})$, $frac(w(x_i)) \neq 0$. Using the same reasoning as before, we are going to prove that $t_0 \leq \delta \leq t_1$.
First we will prove that $t_0 \leq \delta$. Consider $x_i \in \mathsf{LFP}_{R_p}(D)$. For all clock x_j, since $w \in (E, v(D))$ we have $frac(w(x_i)) - frac(w(x_j)) \lhd_{ij} frac(v(d_{i,j}))$.
From Algorithm 15 applied to (E, D) and since $x_i \in \mathsf{LFP}_{R_p}(D)$ we obtain in (E', D') that $D'_{0,j} = D_{i,j} + (-1, \leq)$. Clearly we have $\lhd_{0j'} = \lhd_{ij} \oplus \leq = \lhd_{ij}$. It gives that

$$frac(w(x_i)) - frac(w(x_j)) - 1(\lhd_{ij} \oplus \leq) frac(v(d_{i,j})) - 1$$

which is equivalent to $frac(w(x_i)) - frac(w(x_j)) - 1 \lhd_{0j'} frac(v(d'_{0,j}))$ which is equivalent to

$$frac(w(x_i)) - 1 \lhd_{0j'} frac(v(d'_{0,j})) + frac(w(x_j)).$$

This gives us our first result.
Second we will prove that $\delta \leq t_1$. Consider $x_i \in \mathsf{LFP}_{R_p}(D)$. For all clock x_j, from Definition 8 (4) we have $frac(w(x_j)) - frac(w(x_i)) \lhd_{ji} frac(v(d_{j,i}))$. We have

$$frac(w(x_j)) - frac(w(x_i)) + 1 \lhd_{ji} frac(v(d_{j,i})) + 1.$$

From Algorithm 15 applied to (E, D) and since $x_i \in \mathsf{LFP}_{R_p}(D)$ we obtain in (E', D') that $D'_{j,0} = D_{j,i} + (1, \leq)$. Clearly we have $\lhd_{j0'} = \lhd_{ji} \oplus \leq = \lhd_{ji}$. Then we can write that $frac(w(x_j)) - frac(w(x_i)) + 1 \lhd_{j0'} frac(v(d'_{j,0}))$ which is equivalent to

$$1 - frac(w(x_i)) \lhd_{j0'} frac(v(d'_{j,0})) - frac(w(x_j)).$$

This gives us our second result.

Now for all clock x_j, we obtain two results. First we have

$$-frac(w(x_j))-\delta \lhd_{0j'} -frac(w(x_j))-t_1 \leq -frac(w(x_j))+frac(w(x_j))+v(d'_{0,j}) = v(d'_{0,j})$$

which is $-v(d'_{0,j}) \lhd_{0j'} frac(w(x_j)) + \delta$.

Secondary we have

$$frac(w(x_j)) + \delta \lhd_{j0'} frac(w(x_j)) + t_0 \leq frac(w(x_j)) - frac(w(x_j)) + v(d'_{j,0}) = v(d'_{j,0})$$

which is $frac(w(x_j)) + \delta \lhd_{j0'} v(d'_{j,0})$.

Since we already treated the case whether i or j are 0, now suppose i, j are both different from 0.

Note that if both $x_i, x_j \in \mathsf{LFP}_{R_p}(D)$, as $frac(w(x_i)) = frac(w(x_j))$, $D_{i,j} = D'_{i,j} = (0, \leq)$ and $D_{j,i} = D'_{j,i} = (0, \leq)$ from Definition 14. Hence $frac(w(x_i)) + \delta - frac(w(x_j)) - \delta \lhd_{ij'} frac(v(d'_{i,j}))$ and $frac(w(x_j)) + \delta - frac(w(x_j)) - \delta \lhd_{ji'} frac(v(d'_{j,i}))$.

The same way, if both $x_i, x_j \notin \mathsf{LFP}_{R_p}(D)$ we have $D_{i,j} = D'_{i,j}$ and $D_{j,i} = D'_{j,i}$ and again our result. If either x_i or x_j is in $\mathsf{LFP}_{R_p}(D)$, the case is similar to $D'_{0,j}$ or $D'_{i,0}$.

Finally, we define $w' = w + \delta$ and $w' \in (E', v(D'))$.

$\Longleftarrow$ .19.0.4 open–p–PDBM respecting Definition 8 (5a)

Suppose in (E, D) there is at least an x_j s.t. $D_{j,0} = D_{0,j} = (0, \leq)$ Let $v \in R_p$, and $x_i \in \mathsf{LFP}_{R_p}(D)$.

Now consider $(E', D') = TE((E, D))$. We need to find a value δ s.t. $w + \delta \in (E', v(D'))$. which is equivalent to prove for all x_i, x_j

$$frac(w(x_j)) + \delta - frac(w(x_i)) - \delta \lhd_{ji'} v(d'_{j,i})$$

and

$$frac(w(x_i)) + \delta - frac(w(x_j)) - \delta \lhd_{ij'} v(d'_{i,j})$$

and

$$-frac(w(x_j)) - \delta \lhd_{0j'} v(d'_{0,j}) \quad \text{and} \quad frac(w(x_j)) + \delta \lhd_{j0'} v(d'_{j,0}).$$

As done previously we are going to define a δ which is different from 0, and show that constraints in (E, D) are satisfied while going forward of δ units of time from w.

Let

$$t_0 = \max_{x \in \mathbb{X}} \{0, -frac(w(x)) - frac(v(d'_{0,x}))\}$$

and

$$t_1 = \min_{x \in \mathbb{X}} \{frac(v(d'_{x,0})) - frac(w(x))\}.$$

We want to prove that $t_0 \leq t_1$. For this purpose, we are going to prove for all clocks i, j that $-frac(w(x_j)) - v(d'_{j,0}) \leq v(d'_{0,i}) - frac(w(x_i))$.

First note that

$$frac(w(x_j)) - frac(w(x_i)) \lhd_{ji} v(d_{j,i})$$

By definition of $TE_<$ applied to (E, D), we have that $D'_{j,i} = D_{j,i}$, and from Definition 8 (4) we have that $D'_{j,i} \leq D'_{j,0} + D'_{0,i}$.
Hence, we have from Definition 7 (2b) that either $d'_{j,i} < d'_{j,0} + d'_{0,i}$ or $d'_{j,i} = d'_{j,0} + d'_{0,i}$ and $\lhd_{ji'} = \lhd_{j0'} \oplus \lhd_{0i'}$ or $\lhd_{ji'} = <$ and $\lhd_{j0'} \oplus \lhd_{0i'} = \leq$.
We can then write that

$$\mathit{frac}(w(x_j)) - \mathit{frac}(w(x_i))(\lhd_{j0'} \oplus \lhd_{0i'})v(d'_{j,0}) + v(d'_{0,i})$$

which is equivalent to

$$-\mathit{frac}(w(x_i)) - v(d'_{0,i})(\lhd_{j0'} \oplus \lhd_{0i'})v(d'_{j,0}) - \mathit{frac}(w(x_j))$$

Now we prove that $t_0 = 0$. Clearly from Definition 8 for any clock i we have that $-\mathit{frac}(w(x_i)) \lhd_{0i} v(d_{0,i})$ which is equivalent to $-\mathit{frac}(w(x_i)) - v(d_{0,i}) \lhd_{0i} 0$.
Hence if as (E, D) there is at least an x_j s.t. $D_{j,0} = D_{0,j} = (0, \leq)$, for this clock j we have $-\mathit{frac}(w(x_j)) - v(d_{0,j}) = 0$.
By definition of $TE_<$ applied to (E, D), we have that $D'_{0,i} = D_{0,i} + (0, <)$. In order to respect the constraint $-\mathit{frac}(w(x_i)) - \delta \lhd_{0i'} v(d'_{0,i})$ which is, as $\lhd_{0i'} =<$, $-\mathit{frac}(w(x_i)) - \delta < v(d'_{0,i})$ and especially for j where $v(d'_{0,j}) = 0$ we have to find a $\delta > 0$.
In order to find an upper bound for δ, we are going to prove that $t_1 > 0$. From Definition 8 (4) we have in (E, D) that for any clocks i, j $D_{j,0} \leq D_{j,i} + D_{i,0}$. Let $x_i \in \mathsf{LFP}_{R_p}(D)$. From Definition 8 (1), we have that $D_{i,0} \leq (1, <)$. This gives that $D_{j,i} + D_{i,0} \leq D_{j,i} + (1, <)$.
By definition of $TE_<$ applied to (E, D), we have that $D'_{j,0} = D_{j,i} + (1, <)$. Hence we have $D_{j,0} \leq D'_{j,0}$.
Now as $\mathit{frac}(w(x_i)) \lhd_{i0} v(d_{i,0})$ we can write $\mathit{frac}(w(x_i)) \lhd_{i0'} v(d'_{i,0})$ and then $0 \lhd_{i0'} v(d'_{i,0}) - \mathit{frac}(w(x_i))$ where $\lhd_{i0'} =<$, which prove our result.
We define $\delta = \frac{t_1}{2}$, therefore $t_0 < \delta < t_1$. Now for all clock x_j, we obtain two results.
First we have

$$-\mathit{frac}(w(x_j)) - \delta < -\mathit{frac}(w(x_j)) - t_1 \lhd_{0j'} -\mathit{frac}(w(x_j)) + \mathit{frac}(w(x_j)) + v(d'_{0,j}) = v(d'_{0,j})$$

which is $-v(d'_{0,j}) \lhd_{0j} \mathit{frac}(w(x_j)) + \delta$ as $\lhd_{0j'} = <$.
Secondary we have

$$\mathit{frac}(w(x_j)) + \delta < \mathit{frac}(w(x_j)) + t_0 \lhd_{j0'} \mathit{frac}(w(x_j)) - \mathit{frac}(w(x_j)) + v(d'_{j,0}) = v(d'_{j,0})$$

which is $\mathit{frac}(w(x_j)) + \delta \lhd_{j0} v(d'_{j,0})$ as $\lhd_{0j'} = <$.
Now we prove that constraints defined in (E', D') on differences of clocks are also satisfied by going forward of δ units of time from w
Recall that in (E', D') from the definition of Algorithm 9 we have for all clock x_j,

$$D'_{j,i} = D_{j,i} \quad \text{and} \quad D'_{i,j} = D_{i,j}.$$

Since we already treated the case whether i or j are 0, now suppose i, j are both different from 0. We have that $\mathit{frac}(w(x_j)) - \mathit{frac}(w(x_i)) \lhd_{ji} v(d_{j,i}) = v(d'_{j,i})$, and therefore as $\lhd_{ji'} = \lhd_{ji}$,

$$\mathit{frac}(w(x_j)) + \delta - \mathit{frac}(w(x_i)) - \delta \lhd_{ji'} v(d'_{j,i}).$$

We also have that $\mathit{frac}(w(x_i)) - \mathit{frac}(w(x_j)) \lhd_{ij} v(d_{i,j}) = v(d'_{i,j})$, therefore as $\lhd_{ij'} = \lhd_{ij}$,

$$\mathit{frac}(w(x_i)) + \delta - \mathit{frac}(w(x_j)) - \delta \lhd_{ij'} v(d'_{i,j}).$$

Finally, we define $w' = w + \delta$ and $w' \in (E', v(D'))$.

◀

▶ **Lemma 41.** *Let R_p be a parameter region and $(E, D) \in p\text{–}\mathcal{PDBM}_{\odot}(R_p)$. Let $v \in R_p$. There is $w' \in TE((E, v(D)))$ iff there is $w \in (E, v(D))$ and a delay δ s.t. $w' = w + \delta$.*

Proof. $\Longleftarrow$ **.19.0.5** $p\text{–}\mathcal{PDBM}_{\odot}(R_p)$

Let $v \in R_p$. Consider $(E', D') = TE((E, D))$ respecting condition Definition 8 (5b), *i. e.*, suppose there is at least an x_i s.t. $D'_{i,0} = (1, <)$ and for all j s.t. $D_{0,j} = (0, \triangleleft_{0j})$, then we have $\triangleleft_{0j} = <$. Let $w' \in (E', v(D'))$.

We need to find a value δ s.t. $w' - \delta \in (E, v(D))$ which is equivalent to prove for all x_i, x_j

$$frac(w'(x_j)) - \delta - frac(w'(x_i)) + \delta \triangleleft_{ji} v(d_{j,i})$$

and

$$frac(w'(x_i)) - \delta - frac(w'(x_j)) + \delta \triangleleft_{ij} v(d_{i,j})$$

and

$$-frac(w'(x_j)) + \delta \triangleleft_{0j} v(d_{0,j}) \quad \text{and} \quad frac(w'(x_j)) - \delta \triangleleft_{j0} v(d_{j,0}).$$

As done previously we are going to define a δ which is different from 0, and show that constraints in (E, D) are satisfied while going backward of δ units of time from w'.

We define the largest and the least amount of time so that all upper bounds of (E, D) are satisfied. Let

$$t_0 = \max_{x \in \mathbb{X}} \{0, frac(w'(x)) - v(d_{x,0})\}$$

and

$$t_1 = \min_{x \in \mathbb{X}} \{frac(w'(x)) + v(d_{0,x})\}.$$

We want to prove that $t_0 = t_1 > 0$. For this purpose, let us first show that for all i, j we have $frac(w'(x_j)) - v(d'_{j,0}) \leq frac(w'(x_i)) + v(d'_{0,i})$, which is $t_0 \leq t_1$.

First note that for all i, j

$$frac(w'(x_j)) - frac(w'(x_i)) \triangleleft_{ji'} v(d'_{j,i}).$$

By applying $TE_<$ (Algorithm 9) to (E, D), we have that $D'_{j,i} = D_{j,i}$, *i. e.*, $(d_{i,j}, \triangleleft_{ij}) = (d'_{i,j}, \triangleleft_{ij'})$, and from Definition 9 (2) we have that $D_{j,i} \leq D_{j,0} + D_{0,i}$ is valid for R_p.

Hence, we have from Definition 7 (2b) that either $v(d_{j,i}) < v(d_{j,0}) + v(d_{0,i})$ or $v(d_{j,i}) \leq v(d_{j,0}) + v(d_{0,i})$ and $\triangleleft_{ji} = \triangleleft_{j0} \oplus \triangleleft_{0i}$ or $\triangleleft_{ji} = <$ and $\triangleleft_{j0} \oplus \triangleleft_{0i} = \leq$.

We can then write that

$$frac(w'(x_j)) - frac(w'(x_i))(\triangleleft_{j0} \oplus \triangleleft_{0i}) v(d_{j,0}) + v(d_{0,i})$$

which is equivalent to

$$frac(w'(x_j)) - v(d_{j,0})(\triangleleft_{j0} \oplus \triangleleft_{0i}) frac(w'(x_i)) + v(d_{0,i})$$

so we obtain our result, as $(\triangleleft_{j0} \oplus \triangleleft_{0i})$ is either $\leq$ or $<$.

Now, recall that in (E, D) for all x we have $d_{0,x} = -d_{x,0}$ and $\triangleleft_{0x} = \triangleleft_{x0}$.

For any clock x we have that $\mathit{frac}(w'(x)) - v(d_{x,0}) \leq t_0$ and that $t_1 \leq \mathit{frac}(w'(x)) + v(d_{0,x}) = \mathit{frac}(w'(x)) - v(d_{x,0})$.
Hence $t_0 = t_1$.
As for all x, $\triangleleft_{x0} = \leq$, we have for all i, j that $(\triangleleft_{x0} \oplus \triangleleft_{0i}) = \triangleleft_{0i}$ and $(\triangleleft_{j0} \oplus \triangleleft_{0x}) = \triangleleft_{j0}$, which gives

$$t_1 \triangleleft_{0i} \mathit{frac}(w'(x_i)) + v(d_{0,i})$$

and

$$\mathit{frac}(w'(x_j)) - v(d_{j,0}) \triangleleft_{j0} t_0.$$

Moreover in (E', D') we have that $\mathit{frac}(w'(x)) \triangleleft_{0x'} v(d'_{0,x})$. From Lemma 39, (E', D') is obtained after applying Algorithm 9 and therefore $\triangleleft_{0x'} = <$. Hence $0 < \mathit{frac}(w'(x))$ and

$$0 < t_0 = t_1.$$

Let $\delta = t_0 = t_1$. More intuitively δ is the value right in the middle of the least and the largest amount of time s.t. we can go backward in time from w' and respect all constraints defined in $(E, v(D))$.
First we have

$$-\mathit{frac}(w'(x_j)) + \delta \leq -\mathit{frac}(w'(x_j)) + t_1 \triangleleft_{j0} - \mathit{frac}(w'(x_j)) + \mathit{frac}(w'(x_j)) + v(d_{0,j}) = v(d_{0,j})$$

which is $-v(d_{0,j}) \triangleleft_{j0} \mathit{frac}(w'(x_j)) - \delta$.
Secondary we have

$$\mathit{frac}(w'(x_j)) - \delta \leq \mathit{frac}(w)'(x_j) - t_0 \triangleleft_{0j} \mathit{frac}(w'(x_j)) - \mathit{frac}(w'(x_j)) + v(d_{j,0}) = v(d_{j,0})$$

which is $\mathit{frac}(w'(x_j)) - \delta \triangleleft_{0j} v(d_{j,0})$.
Now we prove that constraints defined in (E, D) on differences of clocks are also satisfied by going back of δ units of time from w'
Recall that in (E', D') from the definition of Algorithm 9 we have for all clocks x_i, x_j,

$$D'_{j,i} = D_{j,i} \quad \text{and} \quad D'_{i,j} = D_{i,j}.$$

Since we already treated the case whether i or j are 0, now suppose i, j are both different from 0. We have that $\mathit{frac}(w'(x_j)) - \mathit{frac}(w'(x_i)) \triangleleft_{ji'} v(d'_{j,i}) = v(d_{j,i})$, and therefore as $\triangleleft_{ji'} = \triangleleft_{ji}$,

$$\mathit{frac}(w'(x_j)) - \delta - \mathit{frac}(w'(x_i)) + \delta \triangleleft_{ji} v(d_{j,i}).$$

We also have that $\mathit{frac}(w'(x_i)) - \mathit{frac}(w'(x_j)) \triangleleft_{ij'} v(d'_{i,j}) = v(d_{i,j})$, therefore as $\triangleleft_{ij'} = \triangleleft_{ij}$,

$$\mathit{frac}(w'(x_i)) - \delta - \mathit{frac}(w'(x_j)) + \delta \triangleleft_{ij} v(d_{i,j}).$$

To conclude, we define for all x_j

$$w(x_j) = w'(x_j) - \delta$$

and clearly, $w \in (E, v(D))$.

$\implies$ .19.0.6 p–$\mathcal{PDBM}_\odot(R_p)$

Assume in $(E, D) \in p$–$\mathcal{PDBM}_\odot(R_p)$. Let $v \in R_p$, and $x_i \in \mathsf{LFP}_{R_p}(D)$.
Now consider $(E', D') = TE((E, D))$. We need to find a value δ s.t. $w + \delta \in (E', v(D'))$.
which is equivalent to prove for all x_i, x_j

$$frac(w(x_j)) + \delta - frac(w(x_i)) - \delta \lhd_{ji'} v(d'_{j,i})$$

and

$$frac(w(x_i)) + \delta - frac(w(x_j)) - \delta \lhd_{ij'} v(d'_{i,j})$$

and

$$-frac(w(x_j)) - \delta \lhd_{0j'} v(d'_{0,j}) \quad \text{and} \quad frac(w(x_j)) + \delta \lhd_{j0'} v(d'_{j,0}).$$

As done previously we are going to define a δ which is different from 0, and show that constraints in (E, D) are satisfied while going forward of δ units of time from w.
Let

$$t_0 = \max_{x \in \mathbb{X}}\{0, -frac(w(x)) - frac(v(d'_{0,x}))\}$$

and

$$t_1 = \min_{x \in \mathbb{X}}\{frac(v(d'_{x,0})) - frac(w(x))\}.$$

We prove that $t_1 \le t_0$. for any clock i we have that $D_{i,0} = (frac(p), \le)$ and $D_{i,0} = (-frac(p), \le)$ *i. e.*, $d_{0,i} = -d_{i,0}$ for some p, hence $-frac(w(x_i)) - v(d_{0,i}) = -frac(w(x_i)) + v(d_{i,0})$.
By definition of $TE_<$ applied to (E, D), we have that $D'_{0,i} = D_{0,i} + (0, <)$. In order to respect the constraint $-frac(w(x_i)) - \delta \lhd_{0i'} v(d'_{0,i})$ which is, as $\lhd_{0i'} =<$, $-frac(w(x_i)) - \delta < v(d'_{0,i})$, we have to find a $\delta > 0$.
In order to find an upper bound for δ, we are going to prove that $t_1 > 0$. From Definition 9 (2) we have in (E, D) that for any clocks i, j $D_{j,0} \le D_{j,i} + D_{i,0}$. Let $x_i \in \mathsf{LFP}_{R_p}(D)$. From Definition 9 (1), we have that $D_{i,0} \le (1, <)$. This gives that $D_{j,i} + D_{i,0} \le D_{j,i} + (1, <)$.
By definition of $TE_<$ applied to (E, D), we have that $D'_{j,0} = D_{j,i} + (1, <)$. Hence we have $D_{j,0} \le D'_{j,0}$.
Now as $frac(w(x_i)) \lhd_{i0} v(d_{i,0})$ we can write $frac(w(x_i)) \lhd_{i0'} v(d'_{i,0})$ and then $0 \lhd_{i0'} v(d'_{i,0}) - frac(w(x_i))$ where $\lhd_{i0'} =<$, which prove our result.
We define $\delta = \frac{t_1}{2}$, therefore $t_0 < \delta < t_1$. Now for all clock x_j, we obtain two results. First we have

$$-frac(w(x_j)) - \delta < -frac(w(x_j)) - t_1 \lhd_{0j'} -frac(w(x_j)) + frac(w(x_j)) + v(d'_{0,j}) = v(d'_{0,j})$$

which is $-v(d'_{0,j}) \lhd_{0j} frac(w(x_j)) + \delta$ as $\lhd_{0j'} = <$.
Secondary we have

$$frac(w(x_j)) + \delta < frac(w(x_j)) + t_0 \lhd_{j0'} frac(w(x_j)) - frac(w(x_j)) + v(d'_{j,0}) = v(d'_{j,0})$$

which is $frac(w(x_j)) + \delta \lhd_{j0} v(d'_{j,0})$ as $\lhd_{0j'} = <$.
Now we prove that constraints defined in (E', D') on differences of clocks are also satisfied by going forward of δ units of time from w

Recall that in (E', D') from the definition of Algorithm 9 we have for all clock x_j,

$$D'_{j,i} = D_{j,i} \quad \text{and} \quad D'_{i,j} = D_{i,j}.$$

Since we already treated the case whether i or j are 0, now suppose i, j are both different from 0. We have that $frac(w(x_j)) - frac(w(x_i)) \lhd_{ji} v(d_{j,i}) = v(d'_{j,i})$, and therefore as $\lhd_{ji'} = \lhd_{ji}$,

$$frac(w(x_j)) + \delta - frac(w(x_i)) - \delta \lhd_{ji'} v(d'_{j,i}).$$

We also have that $frac(w(x_i)) - frac(w(x_j)) \lhd_{ij} v(d_{i,j}) = v(d'_{i,j})$, therefore as $\lhd_{ij'} = \lhd_{ij}$,

$$frac(w(x_i)) + \delta - frac(w(x_j)) - \delta \lhd_{ij'} v(d'_{i,j}).$$

Finally, we define $w' = w + \delta$ and $w' \in (E', v(D'))$.

◀

◀

A Argument of the claim on parametric guards

Argument. As before we use a projection on parameters and eliminate clocks variables. For some set of clocks $I \subseteq \mathbb{X}$ and $i \in I$, suppose we have the constraints $frac(x_i) \lhd_k frac(p_k)$ and $-frac(x_i) \lhd_l -frac(p_l)$ in g.

When eliminating x_i in any constraint of the form $frac(x_i) - frac(x_j) \lhd_{i,j} v(d_{i,j})$, it is clear that we proceed on $\mathcal{PLT}$ to the operation $(-frac(p_l), \lhd_l) + (d_{i,j}, \lhd_{i,j}) = (-frac(p_l) + d_{i,j}, \lhd_l \oplus \lhd_{i,j})$. In any constraint of the form $frac(x_j) - frac(x_i) \lhd_{j,i} v(d_{j,i})$, we proceed on $\mathcal{PLT}$ to the operation $(frac(p_k), \lhd_k) + (d_{j,i}, \lhd_{j,i}) = (frac(p_k) + d_{j,i}, \lhd_k \oplus \lhd_{j,i})$. We get the following constraints: $-frac(x_j)(\lhd_l \oplus \lhd_{i,j})v(d_{i,j}) - frac(v(p_l))$ and $frac(x_j)(\lhd_k \oplus \lhd_{j,i})frac(v(p_k)) + v(d_{j,i})$.

Moreover, in $(E, v(D))$ we have the constraints $frac(x_j) \lhd_{j,0} v(d_{j,0})$ and $-frac(x_j) \lhd_{0,j} v(d_{0,j})$. From Definition 6 we can perform comparisons between elements of $\mathcal{PLT}$, therefore in R_p we can define $min = \min(frac(v(p_k)) + v(d_{j,i}), v(d_{j,0}))$ and $max = \max(v(d_{i,j}) - frac(v(p_l)), v(d_{0,j}))$.

In order to eliminate x_j, we have to decide whether $min \leq max$. Four cases show up which are comparisons between elements of $\mathcal{PLT}$, as $frac(p_l), -frac(p_k), frac(p_l) - frac(p_k) \in \mathcal{PLT}$:

- $min = frac(v(p_k)) + v(d_{j,i})$ and $max = v(d_{i,j}) - frac(v(p_l))$. Then we obtain the constraint $0((\lhd_l \oplus \lhd_{i,j}) \oplus (\lhd_k \oplus \lhd_{j,i}))v(d_{i,j}) - frac(v(p_l)) + frac(v(p_k)) + v(d_{j,i})$. This is equivalent to $frac(v(p_l)) - frac(v(p_k))((\lhd_l \oplus \lhd_{i,j}) \oplus (\lhd_k \oplus \lhd_{j,i}))v(d_{i,j}) + v(d_{j,i})$, which is a constraint already belonging to R_p.
- $min = frac(v(p_k)) + v(d_{j,i})$ and $max = v(d_{0,j})$. Then we obtain the constraint $0(\lhd_{0,j} \oplus (\lhd_k \oplus \lhd_{j,i}))v(d_{0,j}) + frac(v(p_k)) + v(d_{j,i})$. This is equivalent to $-frac(v(p_k))(\lhd_{0,j} \oplus (\lhd_k \oplus \lhd_{j,i}))v(d_{0,j}) + v(d_{j,i})$, which is a constraint already belonging to R_p.
- $min = v(d_{j,0})$ and $max = v(d_{i,j}) - frac(v(p_l))$. Then we obtain the constraint $0(\lhd_{j,0} \oplus (\lhd_{i,j} \oplus \lhd_l))v(d_{j,0}) + v(d_{i,j}) - frac(v(p_l))$. This is equivalent to $frac(v(p_l))(\lhd_{j,0} \oplus (\lhd_{i,j} \oplus \lhd_l))v(d_{j,0}) + v(d_{i,j})$, which is a constraint already belonging to R_p.
- $min = v(d_{j,0})$ and $max = v(d_{0,j})$. Then we obtain the constraint $0(\lhd_{j,0} \oplus \lhd_{0,j})v(d_{j,0}) + v(d_{0,j})$ which is a constraint of R_p.

In the case where x_j is also in I, suppose we have the constraints $frac(x_j) \lhd_m frac(p_m)$ and $-frac(x_j) \lhd_n -frac(p_n)$ in g. Then we can eliminate x_j in the three last cases above with these constraints, and we obtain comparisons between elements of $\mathcal{PLT}$, as $frac(p_m) - frac(p_n), frac(p_m) - frac(p_l), frac(p_k) - frac(p_n) \in \mathcal{PLT}$:

- $min = frac(v(p_k)) + v(d_{j,i})$ and $max = v(-frac(p_n))$. Then we obtain the constraint $0(\triangleleft_n \oplus (\triangleleft_k \oplus \triangleleft_{j,i}))frac(v(p_k)) - frac(v(p_n)) + v(d_{j,i})$, which is a constraint already belonging to R_p.
- $min = frac(v(p_m))$ and $max = v(d_{i,j}) - frac(v(p_l))$. Then we obtain the constraint $0(\triangleleft_m \oplus (\triangleleft_{i,j} \oplus \triangleleft_l))frac(v(p_m)) + v(d_{i,j}) - frac(v(p_l))$, which is a constraint already belonging to R_p.
- $min = frac(v(p_m))$ and $max = v(-frac(p_n))$. Then we obtain the constraint $0(\triangleleft_m \oplus \triangleleft_n)frac(v(p_m)) - frac(v(p_n))$, which is a constraint of R_p.

Hence it does not create new inequalities not belonging to R_p. ◀

A.1 Proof of Proposition 25

Proposition 25 (recalled). *Let R_p be a parameter region. Let $\mathcal{A}$ be an R-U2P-PTA and $\mathcal{R}(\mathcal{A})$ its parametric region automaton over R_p. There is a run $\sigma : (l_0, (E_0, D_0)) \xrightarrow{e_0} (l_1, (E_1, D_1)) \xrightarrow{e_1} \cdots (l_{f-1}, (E_{f-1}, D_{f-1})) \xrightarrow{e_{f-1}} (l_f, (E_f, D_f))$ in $\mathcal{R}(\mathcal{A})$ iff for all $v \in R_p$ there is a run $\rho : (l_0, w_0) \xrightarrow{e_0} (l_1, w_1) \xrightarrow{e_1} \cdots (l_{f-1}, w_{f-1}) \xrightarrow{e_{f-1}} (l_f, w_f)$ in $v(\mathcal{A})$ s.t. for all $0 \leq i \leq f$, $w_i \in (E_i, v(D_i))$.*

Proof. $\Leftarrow$ By induction on the length of the run.

Let $v \in R_p$. As the basis for the induction, in the initial location $(l_0, \{0\}^H)$ the only valuation is reachable by an empty run of $v(\mathcal{A})$. Moreover $\{0\}^H \in (E_0, v(D_0))$ the initial p–PDBM containing only 0. Therefore the initial location $(l_0, (E_0, v(D_0)))$ is reachable by an empty run of $\mathcal{R}(\mathcal{A})$.

For the induction step, suppose for all v, there is run in $v(\mathcal{A})$ of length $f - 1$ we have our result.

Let $v \in R_p$ and $\rho = (l_0, w_0) \xrightarrow{e_0} \cdots \xrightarrow{e_{f-2}} (l_{f-1}, w_{f-1}) \xrightarrow{e_{f-1}} (l_f, w_f)$ be a run of $v(\mathcal{A})$ of length f. By induction hypothesis, there is a run $\sigma = (l_0, (E_0, D_0)) \xrightarrow{e_0} \cdots \xrightarrow{e_{f-2}} (l_{f-1}, (E_{f-1}, D_{f-1}))$ in $\mathcal{R}(\mathcal{A})$ and for all $0 \leq i \leq f-1$, $w_i \in (E_i, v(D_i))$.

Consider e_{f-1}. By Definition 24 of the parametric region automaton, it is also in its set of edges ζ'. Three cases show up:

- If $e_{f-1} = \langle l_{f-1}, a, g, u_{np}, l_f \rangle$ contains no parametric guard nor parametric update. Using Definition 2 there is a delay δ (possibly 0) s.t. $(l_{f-1}, w_{f-1}) \overset{\delta}{\mapsto} (l_{f-1} w'_{f-1}) \overset{e_{f-1}}{\mapsto} (l_f, w_f)$ where $w'_{f-1} \models g$ and $w_f = [w'_{f-1}]_{u_{np}}$. As $w_{f-1} \in (E_{f-1}, v(D_{f-1}))$ there is $(E'_{f-1}, D'_{f-1}) \in \mathsf{Succ}((E_{f-1}, D_{f-1}))$ s.t. from Proposition 18 we have $w'_{f-1} \in (E'_{f-1}, v(D'_{f-1}))$. As $w'_{f-1} \models g$ by construction of our p–PDBMs (see Section 3.1.3) any other clock valuation belonging to $(E'_{f-1}, v(D'_{f-1}))$ satisfies g. Therefore $v \in guard_{\forall}(g, E'_{f-1}, D'_{f-1})$ and from Lemma 21, $R_p \subseteq guard_{\forall}(g, E'_{f-1}, D'_{f-1})$. Now, as $w_f = [w'_{f-1}]_{u_{np}}$ consider the open–p–PDBM $(E_f, D_f) = update((E'_{f-1}, D'_{f-1}), u_{np})$; from Lemma 13 we have $w_f \in (E_f, v(D_f))$. Finally there is an edge $(l_{f-1}, (E_{f-1}, D_{f-1})) \xrightarrow{e_{f-1}} (l_f, (E_f, D_f))$.
- If $e_{f-1} = \langle l_{f-1}, a, g, u, l_f \rangle$ contains a parametric guard and a parametric update. Using Definition 2 there is a delay δ (possibly 0) s.t. $(l_{f-1}, w_{f-1}) \overset{\delta}{\mapsto} (l_{f-1}, w'_{f-1}) \overset{e_{f-1}}{\mapsto} (l_f, w_f)$ where $w'_{f-1} \models v(g)$ and $w_f = [w'_{f-1}]_{v(u)}$. As $w_{f-1} \in (E_{f-1}, v(D_{f-1}))$ there is $(E'_{f-1}, D'_{f-1}) \in \mathsf{Succ}((E_{f-1}, D_{f-1}))$ s.t. from Proposition 18 we have $w'_{f-1} \in (E'_{f-1}, v(D'_{f-1}))$. As $w'_{f-1} \models v(g)$, $v \in p\text{-}guard_{\exists}(g, E'_{f-1}, D'_{f-1})$ and from Lemma 22, $R_p \subseteq p\text{-}guard_{\exists}(g, E'_{f-1}, D'_{f-1})$. Now, as $w_f = [w'_{f-1}]_{v(u)}$ consider the point–p–PDBM $(E_f, D_f) = \overline{update}((E'_{f-1}, D'_{f-1}), u)$; $(E_f, v(D_f))$ contains only one clock valuation,

precisely defined by the fully parametric update $v(u)$ so we have $w_f \in (E_f, v(D_f))$. Finally there is an edge $(l_{f-1}, (E_{f-1}, D_{f-1})) \xrightarrow{e_{f-1}} (l_f, (E_f, D_f))$.

- The case where e_{f-1} contains a non parametric guard and a parametric update is similar to the previous one.

Finally, there is a run $\sigma' = \sigma \xrightarrow{e_{f-1}} (l_f, (E_f, D_f))$ of length f in $\mathcal{R}(\mathcal{A})$ s.t. for all $0 \le i \le f$, $w_i \in (E_i, v(D_i))$.

$\Rightarrow$ By induction on the length of the run.

Let $v \in R_p$. As the basis for the induction, the initial location $(l_0, (E_0, v(D_0)))$ is reachable by an empty run of $\mathcal{R}(\mathcal{A})$. Moreover, as $\{0\}^H \in (E_0, v(D_0))$, the initial location $(l_0, \{0\}^H)$ is reachable by an empty run of $v(\mathcal{A})$.

For the induction step, suppose it is true for all run in $\mathcal{R}(\mathcal{A})$ of length $f-1$.

Let $v \in R_p$ and $\sigma = (l_0, (E_0, D_0)) \xrightarrow{e_0} \cdots \xrightarrow{e_{f-2}} (l_{f-1}, (E_{f-1}, D_{f-1})) \xrightarrow{e_{f-1}} (l_f, (E_f, D_f))$ be a run of $\mathcal{R}(\mathcal{A})$ of length f. Consider e_{f-1}. By Definition 24 of the parametric region automaton, it is also in the set of edges ζ of $\mathcal{A}$. Two cases show up:

- If $e_{f-1} = \langle l_{f-1}, a, g, u_{np}, l_f \rangle$ contains no parametric guard nor parametric update. By induction hypothesis, there is a run $\rho = (l_0, w_0) \xrightarrow{e_0} \cdots \xrightarrow{e_{f-2}} (l_{f-1}, w_{f-1})$ of $v(\mathcal{A})$ of length $f-1$ s.t. for all $0 \le i \le f-1$, $w_i \in (E_i, v(D_i))$. Using Definition 24 there is $(E'_{f-1}, D'_{f-1}) \in \mathsf{Succ}((E_{f-1}, D_{f-1}))$, $R_p \subseteq guard_\forall(g, E'_{f-1}, D'_{f-1})$ and $(E_f, D_f) = update((E'_{f-1}, D'_{f-1}), u_{np})$. From Proposition 18 we have $w'_{f-1} \in (E'_{f-1}, v(D'_{f-1}))$ and a delay δ s.t. $w'_{f-1} = w_{f-1} + \delta$. As $R_p \subseteq guard_\forall(g, E'_{f-1}, D'_{f-1})$ from Lemma 21 we have $v \in guard_\forall(g, E'_{f-1}, D'_{f-1})$ and $w'_{f-1} \models g$. Moreover, since $(E_f, D_f) = update((E'_{f-1}, D'_{f-1}), u_{np})$, we define $w_f = [w'_{f-1}]_{u_{np}}$ and therefore from Lemma 13, $w_f \in (E_f, v(D_f))$. Finally there is an edge $(l_{f-1}, w_{f-1}) \xrightarrow{e_{f-1}} (l_f, w_f)$ and a run $\rho' = \rho \xrightarrow{e_{f-1}} (l_f, w_f)$ in $v(\mathcal{A})$ of length f s.t. for all $0 \le i \le f$, $w_i \in (E_i, v(D_i))$.
- If $e_{f-1} = \langle l_{f-1}, a, g, u, l_f \rangle$ contains a parametric guard and a parametric update. Using Definition 24 there is $(E'_{f-1}, D'_{f-1}) \in \mathsf{Succ}((E_{f-1}, D_{f-1}))$, $R_p \subseteq p\text{-}guard_\exists(g, E'_{f-1}, D'_{f-1})$ and $(E_f, D_f) = \overline{update}((E'_{f-1}, D'_{f-1}), u)$. From Lemma 22 we can take $w'_{f-1} \in (E'_{f-1}, v(D'_{f-1}))$ s.t. $w'_{f-1} \models v(g)$. Let $w_f = [w'_{f-1}]_{v(u)}$. Clearly, $(E_f, D_f) = \overline{update}((E'_{f-1}, D'_{f-1}), u)$ is a point–p–PDBM; as $(E_f, v(D_f))$ contains only one clock valuation precisely defined by the fully parametric update $v(u)$, we have $w_f \in (E_f, v(D_f))$. From Proposition 18 as $w'_{f-1} \in (E'_{f-1}, v(D'_{f-1}))$ there is a delay δ and a $w_{f-1} \in (E_{f-1}, v(D_{f-1}))$ s.t. $w'_{f-1} = w_{f-1} + \delta$. Using the induction hypothesis, there is a run $\rho = (l_0, w_0) \xrightarrow{e_0} \cdots \xrightarrow{e_{f-2}} (l_{f-1}, w_{f-1})$ of $v(\mathcal{A})$ of length $f-1$ s.t. for all $0 \le i \le f-1$, $w_i \in (E_i, v(D_i))$. Finally there is an edge $(l_{f-1}, w_{f-1}) \xrightarrow{e_{f-1}} (l_f, w_f)$ and a run $\rho' = \rho \xrightarrow{e_{f-1}} (l_f, w_f)$ in $v(\mathcal{A})$ of length f s.t. for all $0 \le i \le f$, $w_i \in (E_i, v(D_i))$.
- The case where e_{f-1} contains a non parametric guard and a parametric update is similar to the previous one.

◀

A.2 Proof of Theorem 26

Theorem 26 (recalled). *Let $\mathcal{A}$ be an R-U2P-PTA. Let R_p be a parameter region and $v \in R_p$. If there is a run $\rho = (l_0, w_0) \xrightarrow{e_0} \cdots \xrightarrow{e_{i-1}} (l_i, w_i)$ in $v(\mathcal{A})$, then for all $v' \in R_p$ there is a run $\rho' = (l_0, w'_0) \xrightarrow{e_0} \cdots \xrightarrow{e_{i-1}} (l_i, w'_i)$ in $v'(\mathcal{A})$ such that for all i, $(w_i, v) \simeq (w'_i, v')$.*

Proof. Let $v \in R_p$ and ρ a run of $v(\mathcal{A})$ reaching (l_i, w_i). From Proposition 25, there is a run σ in $\mathcal{R}(\mathcal{A})$ s.t. each clock valuation at a location in ρ is in the p–PDBM at the same location in σ. Still from Proposition 25, for all $v' \in R_p$ there is a run ρ' in $v'(\mathcal{A})$ reaching (l_i, w_i') s.t. each clock valuation at a location in ρ' is in the p–PDBM at the same location in σ (note that possibly $v = v'$). Therefore, we have for all $0 \leq j \leq i$ that $(w_i, v) \simeq (w_i', v')$ and the expected result.

◀

A.3 Proof of Theorem 27

Theorem 27 (recalled). *The EF-emptiness problem is PSPACE-complete for bounded R-U2P-PTAs.*

▶ **Lemma 42.** *There is a finite number of* p*–PDBMs.*

Proof. Let $(E, D) \in p\text{–}\mathcal{PDBM}(R_p)$ and consider $\mathcal{PLT}$: D is a $(H+1)^2$ matrix composed of pairs $(d, \triangleleft)$ where $d \in \mathcal{PLT}$ and $\triangleleft \in \{\leq, <\}$. Hence the number of possible D is bounded by $(2 \times (2 + M(3\frac{M-1}{2} + 4)))^{(H+1)^2}$. Moreover the number of E is bounded since clocks have a maximal value: it is a finite set of integer vectors of $\mathbb{N}^H$. ◀

▶ **Lemma 43.** *There is a finite number of Precise Parametric regions.*

Proof. Consider $\mathcal{PLT} \setminus \{0, 1\}$. Each constraint is a comparison $(\{\leq, <\})$ of plt_1 and $plt_2 \in \mathcal{PLT}$. Hence the number of possible constraints is bounded by $2 \times (2 + M(3\frac{M-1}{2} + 4))^3$. Moreover, the number of $\mathbb{P}$-region $\mathcal{R}_p$ is bounded since they have a maximal value: indeed, since $\mathbb{P}$-region are constructed as clocks regions of [AD94], it is bounded by $M! \times 2^M \times \prod_{p \in \mathbb{P}}(2M + 2)$ ◀

Proof. Since a TA is a special case of R-U2P-PTA we have the PSPACE-hardness [AD94]. Now, let G be a set of goal locations of $\mathcal{A}$. We build a non-deterministic Turing machine that:

1. takes $\mathcal{A}$, G and K as input
2. non-deterministically "guesses" a parameter region R_p
3. takes $v \in R_p$ and writes it to the tape
4. overwrite on the tape each parameter p by $v(p)$, giving the updatable TA $v(\mathcal{A})$
5. solves reachability in $v(\mathcal{A})$ for G
6. accepts iff the result of the previous step is "yes".

The machine accepts iff there is an integer valuation v bounded by K and a run in $v(\mathcal{A})$ reaching a location $l \in G$.

The size of the input is $|\mathcal{A}| + |G| + |K|$, using $|.|$ to denote the size in bits of the different objects. Moreover, the number of parameter regions is bounded (M is the number of parameters in $\mathcal{A}$) by $\big(M! \times 2^M \times \prod_{p \in \mathbb{P}}(2M+2)\big) \times \big(2 \times (2 + M(3\frac{M-1}{2} + 4))^3\big)$ since they are constructed as the clock regions of [AD94], the second part being the maximal number of constraints in a parameter region. Picking v at step *iii)* uses a PSPACE linear programming algorithm (*e. g.,* [Kar84]). Storing the valuation at step *iv)* uses at most $M \times |K|$ additional bits, which is polynomial w.r.t. the size of the input. Step *v)* also needs polynomial space from [BDFP04]. So globally this non-deterministic machine runs in polynomial space. Finally, by Savitch's theorem we have PSPACE = NPSPACE [Sav70], and the expected result.

◀

A.4 Proof of Corollary 28

The procedure to obtain synthesis is as follows. We assume an R-U2P-PTA $\mathcal{A}$ and a goal location l.

1. enumerate all parameter regions (of which there is a finite number)
2. for each R_p, pick a parameter valuation we pick $v \in R_p$ (*e. g.*, using a linear programming algorithm [Kar84])
3. test the reachability of l in the updatable timed automaton $v(\mathcal{A})$, which is decidable [BDFP04]
4. if l is reachable in $v(\mathcal{A})$, add R_p to the list of synthesized regions

We finally return the union of all regions R_p that reach l.

The correctness immediately comes from Theorems 26 and 27.