On Properties of the Inverse Method: Commutation of Instanciation and Full Covering of the Behavioral Cartography

Romain Soulat

December 2010

Research report LSV-10-22

Laboratoire Spécification & Vérification

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

On Properties of the Inverse Method: Commutation of Instantiation and Full Covering of the Behavioral Cartography

Romain Soulat

December 7, 2010

Abstract

When one performs an Inverse Method [5] on a Parametric Timed Automata [1] over an instance π_0, one can instantiate some parameters at the very beginning of the analysis or at the end, with a restriction of the constraint K_0 obtained in order to get a constraint over a subset of the parameters. In this report, we show that the results of either methods are the same. We present a theoretical proof and then an illustration of this property on the flip-flop example and the Root Contention protocol. We also present some results about the coverage of behavioral cartography and an illustration of the full covering on the SPSMALL memory.

1 Introduction

In this report, we present some results about the two part of the IMITATOR II tool, the Inverse Method algorithm and the Bahovioral Cartography. In a first part, we show that a subset of the parameters can be instantiate before or after the analysis without any difference on the results. We present an illustration of this property and some of the implication in two case studies, one over the flip-flop example and then on the Root Contention protocol. In a second part we present some results on the Bahvaioral Cartography algorithm. We recall the results of [4] for the case of an acyclic Parametric Timed Automata but with some extended proofs for the full coverage of the whole real-valued parametric space. We also present some results in a more general case, and we show that the whole real-valued parametric space will not be fully covered, but we show that teh Behavioral Cartography will cover alomost the whole real-valued V_0. We illustrate this last part on the SPSMALL memory.

First, we recall some of the definitions. These definitions are from [4].

The Inverse Method is an algorithm used to solve the Inverse problem that can be stated as follows :

The Inverse Problem
Given a PTA $\mathcal{A}$ and a reference valuation π_0, find a constraint K_0 on the parameters such that:

- $\pi_0 \models K_0$, and
- for all $\pi \models K_0$, the trace sets of $\mathcal{A}[\pi]$ and $\mathcal{A}[\pi_0]$ are the same.

We recall in the following the *inverse method* [5], which is a solution to the inverse problem stated above. The inverse method consists in generating runs starting from the initial state, and removing states incompatible with the reference values by appropriately refining the current constraint K_0 on the parameters. The generation procedure is then restarted until a new incompatible state is produced, and so on, iteratively until no incompatible state is generated.

We first informally describe the algorithm *IM* in the following. Starting with $K = \texttt{true}$, we iteratively compute a growing set of reachable states. When a π-*incompatible* state (q, C) is encountered (i.e., when $\pi \not\models C$), K is refined as follows: a π-incompatible inequality J (i.e., such that $\pi \not\models J$) is selected within the projection of C onto the parameters and $\neg J$ is added to K. The procedure is then started again with this new K, and so on, until no new state is computed. We finally return the intersection of the projection onto the parameters of all the constraints associated to the reachable states. See Figure 1 for the algorithm.

Algorithm 1: Inverse method algorithm $IM(\mathcal{A}, \pi)$

input : A PTA $\mathcal{A}$ of initial state $s_0 = (q_0, C_0)$
input : Valuation π of the parameters
output: Constraint K_0 on the parameters

$i \leftarrow 0$; $K \leftarrow \texttt{true}$; $S \leftarrow \{s_0\}$
while true **do**
  while *there are* π*-incompatible states in* S **do**
    Select a π-incompatible state (q, C) of S (i.e., s.t. $\pi \not\models C$) ;
    Select a π-incompatible J in $(\exists X : C)$ (i.e., s.t. $\pi \not\models J$) ;
    $K \leftarrow K \wedge \neg J$;
    $S \leftarrow \bigcup_{j=0}^{i} Post^{j}_{\mathcal{A}(K)}(\{s_0\})$;
  if $Post_{\mathcal{A}(K)}(S) \sqsubseteq S$ **then**
    return $\bigcap_{(q,C)\in S}(\exists X : C)$
  $i \leftarrow i + 1$;
  $S \leftarrow S \cup Post_{\mathcal{A}(K)}(S)$ $S = \bigcup_{j=0}^{i} Post^{j}_{\mathcal{A}(K)}(\{s_0\})$

Figure 1: The algorithm for the Inverse Method

Most of the notations are borrowed from [4, 5]. Afterwards, we assume, wothout loss of generality, that the partial instantiation of the parameters only concerned the subset $(p_1, ..., p_j)$. We also assume that the instantiation must be compatible with the π_0 used in the analysis.

Let $\mathcal{A} = (\Sigma, Q, q_0, K, I, \rightarrow)$ be a PTA, $\sigma = \{\pi | \pi(p_1) = \alpha_1, ..., \pi(p_j) = \alpha_j\}$ an instantiation of the parameters $(p_1, ..., p_j)$ and $\mathcal{A}_{/\sigma} = (\Sigma, Q, q_0, K \wedge Inst_{\alpha_1,...,\alpha_j}, I, \rightarrow)$ the instantiated version of $\mathcal{A}$. Let G be the reachability graph of $\mathcal{A}$, and G' the one for $\mathcal{A}_{/\sigma}$, possibly with a false constraint attached to some states.

Notation 1. *Let X be a set of clocks and C a constraint over X and a set of parameters P. We denote by $\exists X : C$ the constraint obtained by eliminating the clocks variables in C. More formally, $\exists X : C = \{\pi$ a parameter valuation $| \exists \omega : C[\omega][\pi] = \text{True}\}$*

Notation 2. *We denote by* $\mathrm{IM}(\mathcal{A}, \pi_0)$ *the result of the Inverse Method on automaton* $\mathcal{A}$ *and the parameter valuation* π_0

Lemma 1. *Let* Q *be a constraint over the parameters only,* Q *is identified with the set* $\{\pi | Q[\pi] = \mathrm{True}\}$.

$$\left(\bigwedge_{(q,C)\in S} (\exists X : C) \right) \wedge Q = \bigwedge_{(q,C)\in S} (\exists X : (C \wedge q))$$

Proof :

$$\exists X : (C \wedge Q) = \{\pi | \exists \omega : (C \wedge Q)[\omega][\pi] = \mathrm{True}\}$$

And Q is a constraint over the parameters only, therefore $Q[\omega][\pi] = P[\pi]$.
Therefore,

$$\exists X : (C \wedge Q) = \{\pi | \exists \omega : C[\omega][\pi] = \mathrm{True} \wedge Q[\pi] = \mathrm{True}\}$$

Therefore,

$$\exists X : (C \wedge Q) = \{\pi | \exists \omega : C[\omega][\pi] = \mathrm{True}\} \cap \{\pi | Q[\pi] = \mathrm{True}\}$$

Finally,

$$\exists X : (C \wedge Q) = (\exists X : C) \wedge Q$$

□

Lemma 2. *Let* (q, C) *be a state of* G *and* (q', C') *be one of its successor in* G. *If* $(q, C \wedge \sigma)$ *is a state of* G' *then* $(q', C' \wedge \sigma)$ *is one of its successor in* G'.

Proof : We know that (q', C') is a successor of (q, C) in G. Therefore, there exists an action a, a guard g, a reset ρ such that $(q, g, a, \rho, q') \in \rightarrow$. The same relation holds for $\mathcal{A}'$.
Therefore, there exists K such that (q', K) is the successor of $(q, C \wedge \sigma)$, with

$$K(X') = (\exists X, d : (C \wedge \sigma)(X) \wedge g(X) \wedge (X' = \rho(X)) \wedge I_{q'}(X') \wedge I_{q'}(X' + d))$$

And, because σ is a constraint over the parameters only,

$$\sigma(X) = \sigma$$

Therefore,

$$K(X') = (\exists X, d : C(X) \wedge g(X) \wedge (X' = \rho(X)) \wedge I_{q'}(X') \wedge I_{q'}(X' + d)) \wedge \sigma$$

And, by definition,
$C' = (\exists X, d : C(X) \wedge g(X) \wedge (X' = \rho(X)) \wedge I_{q'}(X') \wedge I_{q'}(X' + d))$ then,

$$K(X') = C' \wedge \sigma$$

□

We know that (q_0, C) is in G is the initial state in G and $(q_0, C \wedge \sigma)$ is the initial state in G', therefore we have the following lemma.

Lemma 3. *If* $s = (q, C)$ *is a state of* G, *then* $s' = (q, C \wedge \sigma)$ *is state of* G', *possibly with* $C \wedge \sigma = \mathrm{False}$.

Proof : We know that (q_0, C) is in G is the initial state in G and $(q_0, C \wedge \sigma)$ is the initial state in G', by induction and the lemma 2, the proof is straightforward. □

We know that every possible reachable state in G may be reach in G' (the constraint associated may be false). We need to show that a state is π_0-compatible in G if and only if this state is π_0-compatible in G'. Therefore, we will have the equality of the reachable trees under π_0.

Lemma 4. *$s = (q, C) \in G$ is π_0 compatible iff $s' = (q, C \wedge \sigma) \in G'$ is $\pi_0 -$ compatible.*

Proof : Let $s = (q, C) \in G$ be a π_0-compatible state and $s' = (q, C \wedge \sigma$ be its associated state in G'.
s is π_0-compatible meaning that $\pi_0 \models \exists X : C$. The assucmption was that $\pi_0 \models \sigma$. Therefore, $\pi_0 \models \exists(X : C) \wedge \sigma$.
The proof of Lemma 1 showed that $\exists(X : C) \wedge \sigma = \exists(X : C \wedge \sigma)$.
Therefore, s' is π_0-compatible.
Let $s = (q, C) \in G$ be a π_0-incompatible state and $s' = (q, C \wedge \sigma$ be its associated state in G'.
s is π_0-incompatible, meaning that $\pi_0 \not\models \exists(X : C)$, therefore $\pi_0 \not\models \exists(X : C) \wedge \sigma = \exists(X : C \wedge \sigma)$.. Therefore s' is π_0-incompatible. □

This lemma shows that the reachability graph of $\mathcal{A}$ and $\mathcal{A}_{/\sigma}$ under π_0 are the same (with the assumption $\pi_0 \models \sigma$).

Theorem 1. $\mathrm{IM}(\mathcal{A}_{/\sigma}, \pi_0) = \mathrm{IM}(\mathcal{A}, \pi_0)_{/\sigma}$ *where* $\mathrm{IM}(\mathcal{A}, \pi_0)_{/\sigma} = \mathrm{IM}(\mathcal{A}, \pi_0) \wedge \sigma$

Proof : Lemma 4 states that the reachability graphs, under π_0 are the same. We know that $\mathrm{IM}(\mathcal{A}, \pi_0)$ returns $\bigwedge_{(q,C)\in S}(\exists X : C)$. Therefore, an instantiation σ) a posteriori of some parameters gives the constraint :

$$(\bigwedge_{(q,C)\in S} (\exists X : C)) \wedge \sigma$$

By the preceding Lemmas, we have that,

$$\mathrm{IM}(\mathcal{A}_{/\sigma}, \pi_0) = \bigwedge_{(q,C\wedge\sigma)\in S} (\exists X : (C \wedge \sigma)$$

By Lemma 1, we have :

$$\mathrm{IM}(\mathcal{A}_{/\sigma}, \pi_0) = \bigwedge_{(q,C\wedge\sigma)\in S} (\exists X : C) \wedge \sigma$$

Therefore

$$\mathrm{IM}(\mathcal{A}_{/\sigma}, \pi_0) = \mathrm{IM}(\mathcal{A}, \pi_0) \wedge \sigma$$

□

This theorem shows that the instantiation of some parameters can be done before or after the inverse method analysis without any difference on the result.

2 Example : Flip-flop

Flip-flop [10] is a case study example for IMITATOR II [2, 3]. It provides a small example, with 12 parameters, 5 clocks and 4 automaton with a complex behavior depending on the choice of π_0. For a full description of the model, see [10]. For this study, we chose the following π_0 :

- tHI = 40
- tLO = 20
- tSetup = 19
- tHold = 6
- $dG1_u = 18$
- $dG1_l = 18$
- $dG2_l = 5$
- $dG2_u = 19$
- $dG3_l = 8$
- $dG3_u = 10$
- $dG4_l = 3$
- $dG4_u = 7$

The reachability tree under this π_0 is prensented in figure 2. The constraint

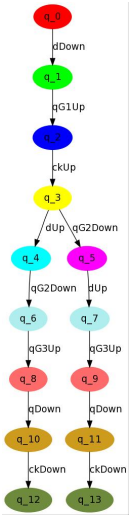

Figure 2: Reachability under π_0, no instanciation

obtained, without any instanciation, is :

- $\text{tSetup} \geq \text{dG1}_\text{u}$
- $\text{tHold} + \text{dG1}_\text{l} \geq \text{dG2}_\text{u}$
- $\text{tHold} \geq \text{dG2}_\text{l}$
- $\text{dG2}_\text{l} \geq 0$
- $\text{tLO} \geq \text{tSetup}$
- $\text{dG4}_\text{l} \geq 0$
- $\text{dG4}_\text{u} \geq \text{dG4}_\text{l}$
- $\text{dG3}_\text{u} \geq \text{dG3}_\text{l}$
- $\text{tHI} \geq \text{dG2}_\text{u} + \text{dG3}_\text{u} + \text{dG4}_\text{u}$
- $\text{dG1}_\text{u} \geq \text{dG1}_\text{l}$
- $\text{dG1}_\text{l} \geq \text{tHold}$
- $\text{dG2}_\text{l} + \text{dG3}_\text{l} \geq \text{tHold}$
- $\text{dG2}_\text{u} \geq \text{tHold}$

An instance of every parameters but $\text{dG1}_\text{u}, \text{dG2}_\text{u}, \text{dG3}_\text{u}, \text{dG4}_\text{u}$ gives the following constraint :

- $19 \geq \text{dG1}_\text{u}$
- $24 \geq \text{dG2}_\text{u}$
- $\text{dG4}_\text{u} \geq 3$
- $\text{dG3}_\text{u} \geq 8$
- $40 \geq \text{dG2}_\text{u} + \text{dG3}_\text{u} + \text{dG4}_\text{u}$
- $\text{dG1}_\text{u} \geq 18$
- $\text{dG2}_\text{u} \geq 6$

This constraint corresponds, of course, exactly to the one, with an instanciation a priori. The reachability tree is exactly the same as well. The time needed to do the analysis was (on the average of 10 runs) : $0.378s$ for the fully parametred version, $0.352s$ for the instanced one. This corresponds to a speed-up of 6.9%. Therefore, the a priori instanciation is not recommended as the speed-up is small compared to the lost of information. With a fully-parametred analysis, one can then isntance whaterver subset of parameters one wants, for a slight overcost.

An intuitive idea of the proof is the following. If a state was π_0 compatible, in the fully-parametred analysis, the adjunction of a π_0-compatible constraint does not change the reachability graph, in the same way, if a state was π_0-incompatible, the adjunction of a π_0-compatible constraint does not change its status. The commutation inside the intersection overall states is quite obvious, since the instanciation is only on the parameters.

3 Root Contention Protocol

An other advantage of postponing the instanciation until after the analysis has been showned in [4]. In this case for the Root Contention Protocol [12, 11], an early instance of the parameters prevented the use of PRISM for an additional analysis because some of the instancied parameters were too big (~ 80) to be handled by PRISM. Instead, a fully-parametred analysis allowed the authors to rescale all the parameters to as low as desired (~ 1) so PRISM would be able to handle the analysis. The figure is shown in figure 3. In this cartography, we see that with this particular cartography that we can decrease the *s_min* parameter to as low as 80 and the delay to as low as we want. If the cartography would have been done with 5 parameters we could have seen that every parameter can be decreased to as low as desired.

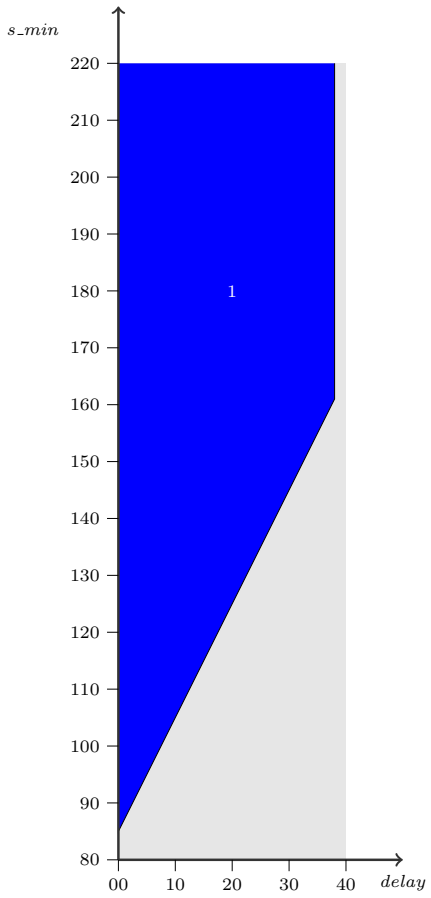

Figure 3: Result of the Inverse Method for RCP according *delay* and *s_min*

4 Behavioral Cartography

The behavioral cartography [4, 6] is a feature of IMITATOR II that allows to create tiles in which the behavior of the given PTA is the same, allowing for a fast verification of large and dense area of the parameters space. See [4, 6] for more details.

4.1 Acyclic case

First, we would like to recall some results presented in [4]

Lemma 5 (Finite number of tiles)**.** *Let $\mathcal{A}$ be an acyclic PTA. The set of tiles $\{IM(\mathcal{A}, \pi) \mid \pi \in \mathbb{Q}^p_{\geq 0}\}$, where p is the number of parameters in $\mathcal{A}$, is finite.*

This proof is the one presented by the author in [4]

Proof : First, the number of possible trace sets is finite from the acyclicity of $\mathcal{A}$. Moreover, the set $Post^*_{\mathcal{A}(K)}(\{s_0\})$ of reachable states is finite for any K, due to the acyclicity of $\mathcal{A}$. As a consequence, for a given K, the number of π_0-inequalities is also finite. Thus, there is a finite number of possibilities to refine K by the addition of the negation of a π_0-inequality. As a consequence, there is a finite number of possible constraints K at the end of the algorithm. And, from the finiteness of the number of possible trace sets, the number of possible intersections of K with the constraints associated to the reachable states (i.e., K_0) is finite, which proves the result.

Now, one can show that BC' covers the whole parametric space, for a "sufficiently large" V_0 and a step "sufficiently small", for acyclic PTAs. Formally:

Proposition 1. *Let $\mathcal{A}$ be an acyclic PTA. Then there exist a rectangle V_0 and a step ϵ such that $BC'(\mathcal{A}, V_0, \epsilon)$ covers the whole real-valued parametric space.*

Proof : From Lemma 5 we know that,
$T = \{IM(A, \pi) | \pi \in \mathbb{Q}^P_{\geq 0}\}$ is finite
Meaning that there exists $(\pi_1, ..., \pi_n)$ such that

$$\mathbb{Q}^P_{\geq 0} \subseteq \{IM(\mathcal{A}, \pi) | i \in (1, ..n)\} = T$$

Let $T_i = \{\pi \in \mathbb{R}^P_{\geq 0} | \pi \in IM(\mathcal{A}, \pi_i\}$
Let $x \in \mathbb{R}^P_{\geq 0} \backslash (\bigcup_{i \in \{1,...,n\}} T_i)$, this corresponds to the set of points that may not covered by the tiles.
We know that the tiles are convex, therefore this points are necessarily in the intersection of the boarders of some tiles.
Therefore, $\exists K \subseteq \{1, ..., n\} \mid x \in \bigcap_{k \in K} \delta(T_k)$.
Every vertices of the possible tiles are in $\mathbb{Q}^P_{\geq 0}$ as a solution of a linear system with constants in $\mathbb{Q}^P_{\geq 0}$. The Cramer method for a resolution of a linear system gives a straight forward proof of this claim.
The only points left that may be uncovered are the ones on the facets of each tiles that are not a vertex of this tile. Even those can be easily covered by using the same type of proof as before by aproaching them with a sequence of barycenters of the vertices of the tile they belong to with some rational weights.

This way, we have covered everything in the inner set of each tile, then the vertices of each tiles and finally the rest of the boarder of each tile.
The only way to have an uncovered point would be an isolated irrational points, but then it would be an intersection of some constraints and we have shown that these points are always rational.
Therefore, the cartography cartography may cover the whole real-valued parametric space if applied to the right points.
The number of points being finite, we can bound them in some rectangle V_0, if choosen correctly.
Regarding the step of the cartography needed in order to cover the whole real-valued parametric space, we need to hit all the $(\pi_1, ..., \pi_n)$.
Let us recall that $\forall i \in \{1, ..., n\}, \pi_i \in \mathbb{Q}_{\geq 0}$, therefore
$\forall i \in \{1, ..., n\}, \exists p_i, q_i \in \mathbb{N} | \text{gcd}(p_i, q_i) = 1 and \pi_i = p_i/q_i$.
Therefore with a step of $\epsilon = 1/(gcf(q_1, ..., q_n)$ will hit all these points, therefore $BC'(\mathcal{A}, V_0, \epsilon)$ will cover the whole real valued parametric space.

This gives us a first case where, the algorithm for the behavioral cartography will cover the whole V_0 and moreover the whole real-valued parametric space.

We want to illustrate this result with the following example ; SPSMALL memory. For more details about the model used see [4, 7, 8, 9].

The cartography has been performed over two parameters and all the others being instancied. Despite what has been previously said about a priori instance, the fully-parametred analysis takes too much time to perform a fully parametred behavioral cartography. The cartography with an arbitrary step of 1 is presented in figure 1. We can see that only about 25% of the original V_0 is covered by the

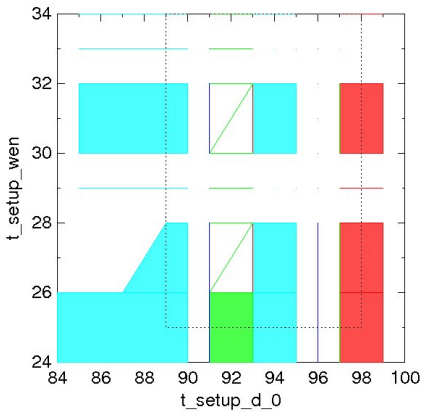

Figure 4: Cartography of the SPSMALL memory with an arbitrary step of 1

cartography, but of course every integer points is covered. The use of a more apt step, here 1/4, returns the carography shown in figure 1. We can see that with this step, the whole V_0 is covered. This step has been chosen because, in this case study, the minimal tile with a non empty inner set that can be found is a unit triangle. Therefore, it's easy to see that with a step of 1 all the vertices of these triangle are going to be covered but we may miss the inner set of these triangles. With a step of 1/4 we are sure to cover those triangles by testing the points $(n + 1/4, m + 3/4)$ and $(n + 3/4, m + 3/4)$ where $n, m \in \mathbb{N}$.

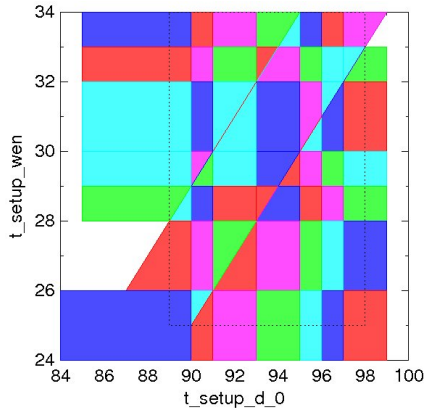

Figure 5: Cartography of the SPSMALL memory with an suitable step of 1/4

4.2 General Case

The proposition 1 has a strong hypothesis that is the acyclicity of the automaton studied. In a more general case, we show in this report that the behavioral cartography may cover almost the whole V_0.

We recall that the Inverse Method only generates convex tiles, i.e. conjunction of linear inequalities. A linear constraint is $\sum_{i\in 1..n}(\alpha_i * p_i)$, with $\prec c\ \alpha_i \in \mathbb{Z}$ for every $i \in I$ and $c \in \mathbb{Z}$.

Notation 3. *Let* $(\alpha_1, ...\alpha_n) \in \mathcal{P}^n \subset \mathbb{Z},\ c \in \mathbb{Z}$. *Let* $\prec \in \mathcal{O} = \{<, \leq, >, \geq\}$
Let C *the constraint* $\sum_{i\in 1..n}(\alpha_i * p_i) \prec c$.
We denote by $\mathcal{C}_{(\alpha_1,...\alpha_n,c,\prec)} = \{(v_1, ..., v_n) \in \mathbb{R}^n | \sum_{i\in 1..n}(\alpha_i * v_i) \prec c\}$ *the sets of points of* $\mathbb{R}^n$ *satisfying* C.

Property 1. *Let* $V_0 = I_1 \times ... \times I_n$, *where* I_i *is a bounded interval of* $\mathbb{R}$ *for every* $i \in \{1..n\}$.
Let $E = \{(\alpha_1, ...\alpha_n, c, \prec) \in \mathcal{P}^n \times \mathbb{Z} \times \mathcal{O} | \mathcal{C}_{(\alpha_1,...\alpha_n,c,\prec)} \cap V_0 \neq \emptyset$ et $\mathcal{C}_{(\alpha_1,...\alpha_n,c,\prec)} \cap V_0 \neq V_0\}$

$$\textit{If } \mathcal{P} \textit{ is bounded then } |E| \leq +\infty$$

E corresponds to constraints defining a non-trivial tile in V_0.

Proof : To show that $|E| < +\infty$, we only need to prove that if $(\alpha_1, ...\alpha_n, c, \prec) \in E$ then c is in a bounded interval $\mathbb{Z}$. Indeed, $\mathcal{P}$is bounded and $|\mathcal{O}| = 4$, so if c can only take a finite number of values, we will have that E is finite.
Let $(\alpha_1, ...\alpha_n, c, \prec) \in E$.
$\exists (v_1, ..., v_n) \in V_0 | \sum_{i\in 1..n}(\alpha_i * v_i) \prec c$ because $\mathcal{C}_{(\alpha_1,...\alpha_n,c,\prec)} \cap V_0 \neq \emptyset$
and
$\exists (v'_1, ..., v'_n) \in V_0 | \sum_{i\in 1..n}(\alpha_i * v'_i)(\neg \prec)c$ because $\mathcal{C}_{(\alpha_1,...\alpha_n,c,\prec)} \cap V_0 \neq V_0$
So we can write, wothout loss of generality that, $c \leq \sum_{i\in 1..n}(\alpha_i * v_i)$ and $c \geq \sum_{i\in 1..n}(\alpha_i * v'_i)$, even if it means switching v_i with v'_i.

But, V_0 is bounded because it's a finite cartesian product of bounded intervals,therefore there exists $M \in \mathbb{N}$ such that $V_0 \subseteq [-M; M]^n$.
And,

$$c \leq \sum_{i \in 1..n} (\alpha_i * v_i)$$

That can be rewritten as

$$c \leq \sum_{i \in 1..n | \alpha_i > 0} (\alpha_i * v_i) + \sum_{i \in 1..n | \alpha_i < 0} (\alpha_i * v_i)$$

$$c \leq \sum_{i \in 1..n | \alpha_i > 0} (\alpha_i * v_i)$$

Therefore,

$$c \leq \sum_{i \in 1..n | \alpha_i > 0} (\alpha_i * M)$$

We also have that $\mathcal{P}$ is bounded , therefore there exists $M' \in \mathbb{N}$ such that $\mathcal{P} \subseteq [-M'; M']$.
And,

$$c \leq \sum_{i \in 1..n | \alpha_i > 0} (M' * M)$$

And we have, $|\{i \in 1..n | \alpha_i > 0\}| \leq n$, therefore,

$$c \leq n * M' * M$$

The same way, we have that,

$$c \geq -n * M' * M \square$$

We know that a tile of E is a conjunction of a finite number of constraints on the parameters. But, we have shown that there is only a finite number of constraints delimiting a non-trivial area of V_0, so we can prove the following theorem:

Theorem 2. *If the coefficients on the constraints that can be generated by the Inverse Methode on a PTA are bounded therefore the Behavioral Cartography only has a finite number of tiles*

Proof : The proof is straightforward from Property 1.

4.3 Behavioral Cartography without Tiles with empty inner sets

The former theorem states that, if its conditions are met, the number of tiles is finite. However, there can be some tiles with an empty inner set, as a hyperplane, a plane, a line, or worse a point. This can be inconvenient when we don't know where to run the next analysis but, instead, just testing some pre-determined points. Even a random analysis would not cover almost surely these tiles. If we can make sure that the tiles generated does not have an empty inner set, we can use the following theorem :

Theorem 3. *If the coefficients on the constraints that can be generated by the Inverse Method on PTA $\mathcal{A}$ are bounded and the tiles have non-empty inner sets then*

$$\exists \epsilon \in \mathbb{R}^{+*} | \forall\ T \text{ a tile, } \lambda(T) > \epsilon, \text{ with } \lambda \text{ the Lebesgue measure}$$

Proof : From theorem 2, we know that the set of tiles T is finite.
And $\forall T,\ \mathring{T} \neq \emptyset$, i.e. there exists $\pi_T \in T$ such that there exists $\epsilon_T \in \mathbb{R}^{+*}$ such that $\mathcal{B}(x, \epsilon) \subseteq T$.
Let $\epsilon' = \min_T(\epsilon_T)$. We have $\epsilon > 0$ because $\epsilon_T > 0$ for a tile T unand that the numer of tiles is finite.
Therefore $\mathcal{B}(\pi_T, \epsilon') \subseteq T$ for every tile T.
Consequently, $\lambda(T) > \epsilon$ with ϵ the Lebesgue measure of the ball of radius ϵ

This theorem gives us a step such that every point of V_0 is covered by the cartography.

This result show that if the conditions of the theorem are met then there exists a step for the Behavioral Cartography such that the entire V_0 is covered.

References

[1] Rajeev Alur and David Dill. The theory of timed automata. In J. de Bakker, C. Huizing, W. de Roever, and G. Rozenberg, editors, *Real-Time: Theory in Practice*, volume 600 of *Lecture Notes in Computer Science*, pages 45–73. Springer Berlin / Heidelberg, 1992. 10.1007/BFb0031987.

[2] Étienne André. IMITATOR II: A tool for solving the good parameters problem in timed automata. In Yu-Fang Chen and Ahmed Rezine, editors, *Proceedings of the 12th International Workshops on Verification of Infinite State Systems (INFINITY'10)*, volume 39 of *Electronic Proceedings in Theoretical Computer Science*, pages 91–99, Singapore, September 2010.

[3] Étienne André. IMITATOR II user manual. Research Report LSV-10-20, Laboratoire Spécification et Vérification, ENS Cachan, France, November 2010. 31 pages.

[4] Étienne André. *An Inverse Method for the Synthesis of Timing Parameters in Concurrent Systems*. Phd thesis, École Normale Supérieure de Cachan, 2010.

[5] Étienne André, Thomas Chatain, Emmanuelle Encrenaz, and Laurent Fribourg. An inverse method for parametric timed automata. *International Journal of Foundations of Computer Science*, 20(5):819–836, October 2009.

[6] Étienne André and Laurent Fribourg. Behavioral cartography of timed automata. In Antonín Kučera and Igor Potapov, editors, *Proceedings of the 4th Workshop on Reachability Problems in Computational Models (RP'10)*, volume 6227 of *Lecture Notes in Computer Science*, pages 76–90, Brno, Czech Republic, August 2010. Springer.

[7] Rémy Chevallier, Emmanuelle Encrenaz-Tiphène, Laurent Fribourg, and Weiwen Xu. Timing analysis of an embedded memory: SPSMALL. *WSEAS Transactions on Circuits and Systems*, 5(7):973–978, July 2006.

[8] Rémy Chevallier, Emmanuelle Encrenaz-Tiphène, Laurent Fribourg, and Weiwen Xu. Verification of the generic architecture of a memory circuit using parametric timed automata. In Eugène Asarin and Patricia Bouyer, editors, *Proceedings of the 4th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS'06)*, volume 4202 of *Lecture Notes in Computer Science*, pages 113–127, Paris, France, September 2006. Springer.

[9] Rémy Chevallier, Emmanuelle Encrenaz-Tiphène, Laurent Fribourg, and Weiwen Xu. Timed verification of the generic architecture of a memory circuit using parametric timed automata. *Formal Methods in System Design*, 34(1):59–81, February 2009.

[10] Robert Clarisó and Jordi Cortadella. The octahedron abstract domain. In *11th Static Analysis Symposium (SAS'04)*, volume 3148 of *Lecture Notes in Computer Science*, pages 312–327. Springer-Verlag, August 2004.

[11] A. Collomb-Annichini and M. Sighireanu. Parameterized reachability analysis of the ieee 1394 root contention protocol using trex. In *PROCEEDINGS OF THE WORKSHOP ON REAL-TIME TOOLS (RT-TOOLS'2001)*, 2001.

[12] T.S. Hune, J.M.T. Romijn, M.I.A. Stoelinga, and F.W. Vaandrager. Linear parametric model checking of timed automata. *Journal of Logic and Algebraic Programming*, 2002.