Finding an Optimal Label-Splitting to Make a Transition System Petri Net Implementable: a Complete Complexity Characterization*

Ronny Tredup

Universität Rostock, Institut für Informatik, Theoretische Informatik, Albert-Einstein-Straße 22, 18059, Rostock, ronny.tredup@uni-rostock.de

Abstract. Petri net *synthesis* is the task of finding a Petri net N for a given transition system (TS) A that implements A, i.e., N is an (exact net) *realization*, a *language-simulation* or an *embedding* of A according to N's degree of accuracy. Regardless of the sought implementation, there is not always a solution. *Label-splitting* converts a non-implementable TS A into an implementable one by giving edges with the same label now different labels. Since increasing the number of labels increases the complexity of the net synthesized, it is desired to keep the number of split labels small. This means that label-splitting can be considered as decision problem that asks for a given TS A and natural number κ whether there is an implementable TS B with at most κ labels that is derived from A by splitting labels. Recently, Schlachter and Wimmel (2020) [18] showed that this problem is NP-complete if an embedding is sought. In this paper, we show that this remains true if A is 2-bounded, i.e., every state has at most two incoming and two outgoing edges, and that this bound is tight. Schlachter and Wimmel also alleged that label-splitting aiming at exact realization is NP-complete. In this paper, we prove this conjecture and show that label-splitting aiming at language-simulation or realization is NP-complete even if A is 1-bounded.

1 Introduction

Petri net *synthesis* [2, 3, 5, 6] is the task of finding a Petri net N for a given transition system (TS) A that implements A, i.e., N is an (exact net) *realization*, a *language-simulation* or an *embedding* of A according to N's degree of accuracy.

Synthesis of Petri nets has applications in many areas like extracting concurrency from sequential specifications like TS and languages [4], process discovery [1], supervisory control [14] or the synthesis of speed independent circuits [12].

However, regardless of the implementation sought, there is not always an implementing Petri net. In this case, *Label-splitting* [3, 11, 17] is an option, i.e, converting a non-implementable TS A into an implementable one by giving edges with the same label now different labels. Label-splitting has been originally introduced in [10] for the synthesis of 1-bounded Petri nets and was extended to k-bounded Petri nets by the same authors in [9]. The corresponding heuristics have been implemented in the synthesis tool PETRIFY [12]. In [8], a novel view of the label-splitting techniques of [9, 10] has been shown, relating them to some NP-complete problems like *chromatic number* and *weighted set cover*. In *process mining*, label-splitting is used to handle imprecise labels of *event-logs* to allow better fitting models instead of strongly over-approximating ones [15, 16]. Other applications of label-splitting can be found in [19], where it supports synthesis to get concise representations of service compositions, exhibiting concurrency.

Label-splitting is a powerful transformation that always guarantees an implementable TS. In fact, an exact implementation is possible at the latest when every edge of the resulting TS is labeled differently. However, increasing the number of labels split, increases the complexity of the net derived and yields an over-fitting model. For applications in process-mining, this prevents users from getting a better insight about the behavior of the process [16]. Thus, it is desired to keep the number of labels split as small as possible. This means that label-splitting can be considered as an optimization problem that consist of converting A into an implementable TS that has as few labels as possible. Equivalently [18], in this paper, we understand *label-splitting* as decision problem that asks for a given TS A and natural number κ whether there is an implementable TS B with at most κ labels that is derived from A by splitting labels.

In [18], it has been shown that label-splitting is NP-complete if an embedding is sought. The presented reduction can be modified, such that the resulting TS is 3-bounded, i.e., every state has at most three incoming and three outgoing edges. In this paper, we strengthen this result and show that label-splitting aiming at embedding remains NP-complete if A is 2-bounded, and that this bound is tight.

In [18], it is also conjectured that label-splitting for realization is intractable. In this paper, we prove this conjecture and show also that label-splitting for language-simulation and (exact) realization is NP-complete even if A is 1-bounded.

We obtain all of our NP-completeness results by reductions from the vertex-cover problem on cubic graphs, which will be introduced in Section 3.

This paper is organized as follows. Section 2 introduces necessary definitions and supports them with examples. Section 3 and Section 4 present the complexity of label-splitting aiming at exact realization, language-simulation and embedding, respectively. Finally, Section 5 briefly closes the paper.

2 Preliminaries

This section introduces all necessary definitions and supports them with examples.

Transition Systems. A (deterministic) *initialized transition system* (TS, for short) $A = (S, E, \delta, \iota)$ is a directed labeled graph with the set of nodes S (called *states*), the set of labels E (called *events*), the partial *transition function* $\delta : S \times E \longrightarrow S$ and the initial state $\iota \in S$, where every state $s \in S$ is *reachable* from ι by a directed labeled path. The *set of arcs of* A is defined by $\Delta_A = \{(s, e, s') \mid s, s' \in Se \in E : \delta(s, e) = s'\}$. Event e *occurs* at state

s, denoted by $s\xrightarrow{e}$, if $\delta(s,e)$ is defined. We abridge $\delta(s,e)=s'$ by $s\xrightarrow{e}s'$. If $w=e_1\dots e_n\in E^*$, then $s\xrightarrow{w}$ denotes that there are states $s=s_0,\dots,s_n\in S$ such that $s_i\xrightarrow{e_{i+1}}s_{i+1}\in A$ for all $i\in\{0,\dots,n-1\}$. The *language* of A is defined by $L(A)=\{w\in E^+\mid \iota\xrightarrow{w}\}\cup\{\varepsilon\}$. Let $b\in\mathbb{N}$. A is called b-bounded if, for every state $s\in S$, there are at most b incoming or outgoing arcs, i.e., $|\{e\in E\mid \xrightarrow{e}s\}|\le b$ and $|\{e\in E\mid s\xrightarrow{e}\}|\le b$. A cycle-free 1-bounded TS A is called *linear*. If A is not explicitly defined, then we refer to its components by $S(A)$ (states), $E(A)$ (events) δ_A (function), ι_A (initial state).

Simulations. Let A and B be TS with the same set of events E. We say B simulates A, if there is a mapping $\varphi: S(A)\to S(B)$ such that $\varphi(\iota_A)=\iota_B$ and $s\xrightarrow{e}s'\in A$ implies $\varphi(s)\xrightarrow{e}\varphi(s')\in B$; such a mapping is called a *simulation* (between A and B). φ is an *embedding*, denoted by $A\hookrightarrow B$, if it is injective; φ is a *language-simulation*, denoted by $A\triangleright B$, if $\varphi(s)\xrightarrow{e}$ implies $s\xrightarrow{e}$, implying $L(A)=L(B)$ [3, p. 67]; φ is an *isomorphism*, denoted by $A\cong B$, if it is bijective and $\delta_A(s,e)=s'$ if and only if $\delta_B(\varphi(s),e)=\varphi(s')$ for all $s,s'\in S(A), e\in E(A)$.

Label-splitting. Let $A=(S,E,\delta,\iota)$ be a TS and $\mathfrak{E}=\{e_1,\dots,e_n\}\subseteq E$ a set of events. The *label-splitting* of $e_1,\dots,e_n$ into the (pairwise distinct) events $e_1^1,\dots,e_1^{m_1},\dots,e_n^1,\dots,e_n^{m_n}$, where $m_i\ge 2$ for all $i\in\{1,\dots,n\}$, yields the event set $E'=(E\setminus\mathfrak{E})\cup\bigcup_{i=1}^n\{e_i^1,\dots,e_i^{m_i}\}$. A TS $B=(S,E',\delta',\iota)$ is an E'-*label-splitting* of A if there is a bijective mapping $\psi:\Delta_B\to\Delta_A$ such that $\psi((s,e,s'))=(s,e_i,s')$ if $e=e_i^j$ for some $i\in\{1,\dots,n\}, j\in\{1,\dots,m_i\}$ and, otherwise, $\psi((s,e,s'))=(s,e,s')$ for all $(s,e,s')\in\Delta_B$. We say $\mathfrak{E}$ is *the set of events of A that occur split in B*.

Petri nets. A *Petri net* $N=(P,T,f,M_0)$ consists of finite and disjoint sets of *places* P and *transitions* T, a (total) *flow function* $f:((P\times T)\cup(P\times T))\to\mathbb{N}$ and an *initial marking* $M_0:P\to\mathbb{N}$. A transition $t\in T$ can *fire* or *occur* in a marking $M:P\to\mathbb{N}$, denoted by $M\xrightarrow{t}$, if $M(p)\ge f(p,t)$ for all places $p\in P$. The firing of t in marking M leads to the marking $M'(p)=M(p)-f(p,t)+f(t,p)$ for all $p\in P$, denoted by $M\xrightarrow{t}M'$. Again, this notation extends to sequences $w\in T^*$ and the *reachability set* $RS(N)=\{M\mid \exists w\in T^*: M_0\xrightarrow{w}M\}$ contains all of N's reachable markings. The *reachability graph* of N is the TS $A_N=(RS(N),T,\delta,M_0)$, where for every reachable marking M of N and transition $t\in T$ with $M\xrightarrow{t}M'$ the transition function δ of A_N is defined by $\delta(M,t)=M'$.

The following definitions relate TS and Petri nets via the reachability graph.

Implementations. A Petri net N is a *realization*, respectively a *language-simulation*, respectively an *embedding* of a TS A if there is a simulation such that $A\cong A_N$, respectively $A\triangleright A_N$, respectively $A\hookrightarrow A_N$.

Regions. If a TS A is implementable by a Petri net N, then we want to construct N purely from A. TS represents the behavior of a modeled system by means of *global states* (states of TS) and transitions between them (events). Dealing with a Petri net, we operate with *local states* (places) and their changing (transitions), while the global states of a net are markings, i.e., combinations of local states. Since A_N has to simulate A, N's transitions correspond to A's events. The connection between global states in TS and local states in the sought net is given by *regions of TS* that mimic places: A region $R = (sup, con, pro)$ of $A = (S, E, \delta, \iota)$ consists of the mappings $sup : S \to \mathbb{N}$ and $con, pro : E \to \mathbb{N}$ such that for edge $s \xrightarrow{e} s'$ of A it holds that $con(e) \leq sup(s)$ and $sup(s') = sup(s) - con(e) + pro(e)$. Notice that R is *implicitly* completely defined by $sup(\iota)$, con and pro: Since A is reachable, for every state $s \in S$, there is a path $\iota \xrightarrow{e_1} \dots \xrightarrow{e_n} s_n$ such that $s = s_n$. Thus, we inductively obtain $sup(s_{i+1})$ by $sup(s_{i+1}) = sup(s_i) - con(e_{i+1}) + pro(e_{i+1})$ for all $i \in \{0, \dots, n-1\}$ and $s_0 = \iota$. Since we can compute sup and, thus, R purely from $sup(\iota)$, con and pro, we often present regions only implicitly. A region $R = (sup, con, pro)$ models a place p and the corresponding part of the flow function f, i.e., $con(e)$ models $f(p, e)$ (the number of tokens that e consumes from p), $pro(e, p)$ models $f(e, p)$ (the number of tokens that e produces on p) and $sup(s)$ models (the number of tokens) $M(p)$ (that are on p) in the marking $M \in RS(N)$ that corresponds to $s \in S(A)$ via the simulation between A and A_N. Every set $\mathcal{R}$ of regions of A defines the *synthesized net* $N_A^{\mathcal{R}} = (\mathcal{R}, E, f, M_0)$ with $f(R, e) = con(e)$, $f(e, R) = pro(e)$ and $M_0(R) = sup(\iota)$ for all $R = (sup, con, pro) \in \mathcal{R}$ and all $e \in E$.

State and Event State Separation. To ensure that the input behavior is captured by the synthesized net, we have to distinguish global states, and prevent the firings of transitions when their corresponding events are not present in TS. This is stated as so called *separation atoms* and *separation properties*. A pair (s, s') of distinct states of A defines a *states separation atom* (SSP atom). A region $R = (sup, con, pro)$ *solves* (s, s') if $sup(s) \neq sup(s')$. If every SSP atom of A is solvable then A has the *state separation property* (SSP, for short). A pair (e, s) of event $e \in E$ and state $s \in S$ where e does not occur, that is $\neg s \xrightarrow{e}$, defines an *event/state separation atom* (ESSP atom). A region $R = (sup, con, pro)$ *solves* (e, s) if $sup(s) < con(e)$. If every ESSP atom of A is solvable then A has the *event state separation property* (ESSP, for short). A set $\mathcal{R}$ of regions of A is called a *witness* for A's SSP, respectively ESSP, if for each SSP atom, respectively ESSP atom, there is a region R in $\mathcal{R}$ that solves it.

The next lemma ([3, p. 162], Proposition 5.10) establishes the connection between the existence of witnesses and the existence of an implementing net N in dependence of the testified property. Notice that Petri nets correspond to the type of nets τ_{PT} in [3, p. 130].

Lemma 1 ([3]). *Let A be a TS. There is a Petri net N such that $A \hookrightarrow A_N$, respectively $A \triangleright A_N$, respectively $A \cong A_N$ if and only if there is a witness $\mathcal{R}$ that testifies A's SSP, respectively ESSP, respectively both SSP and ESSP, and $N = N_A^{\mathcal{R}}$.*

By Lemma 1, deciding the existence of an implementing net is equivalent to deciding if the input TS has the property that corresponds to the implementation. Moreover, there is an implementing Petri net for a given TS A if and only if there is a witness $\mathcal{R}$ of regions that testifies A's SSP or ESSP or both according to the sought-for implementation.

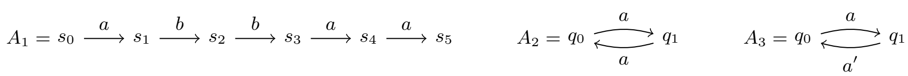

Fig. 1: The TSs A_1 (Example 1), A_2 (Example 2) and A_3 (Example 3). All of them are 1-bounded, but only A_1 is linear.

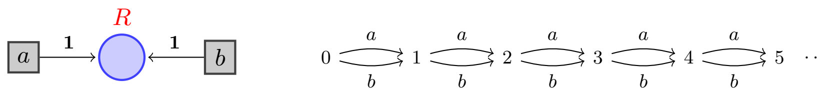

Fig. 2: Left: The net $N = N^{\mathcal{R}}_{A_1}$ (Example 1), where zero-valued flow arcs are omitted. Right: A sketch of the (infinite) reachability graph A_N, which embeds A_1.

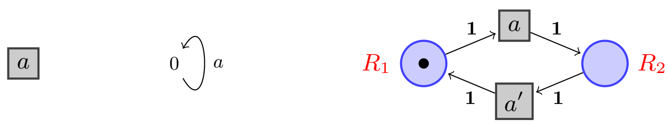

Fig. 3: Left: $N_1 = N^{\mathcal{R}}_{A_2}$ (Example 2). Middle: The reachability graph A_{N_1}. Right: The Net $N_2 = N^{\mathcal{R}}_{A_3}$ (Example 3), whose reachability graph is isomorphic to A_3.

Example 1 (Embedding). The TS A_1 of Figure 1 has the SSP, since the following region $R = (sup, con, pro)$ solves all SSP atoms on one blow: $sup(s_i) = i$ for all $i \in \{0, \dots, 5\}$ and $con(a) = con(b) = 0$ and $pro(a) = pro(b) = 1$. In particular, the set $\mathcal{R} = \{R\}$ is a witness of A_1's SSP. The (infinite) reachability graph of the net $N^{\mathcal{R}}_{A_1} = (\mathcal{R}, \{a, b\}, f, M_0)$ synthesized from $\mathcal{R}$, where $f(R, a) = f(R, b) = con(a) = con(b) = 0$ and $f(a, R) = f(b, R) = pro(a) = pro(b) = 1$ and $M_0(R) = 0$, embeds A_1, cf. Figure 2. An embedding φ is given by $\varphi(s_i) = i$ for all $i \in \{0, \dots, 5\}$.

The TS A_1 does not have the ESSP, since the atom $\alpha = (a, s_2)$ is not solvable. This has already been proven in [7, p. 42], for self-containment we provide the proof: If $R = (sup, con, pro)$ is a region that solves α, then (1) $con(a) \leq sup(s_0)$, since a occurs at s_0; (2) $sup(s_2) < con(a)$, implying $sup(s_0) - (con(a) + con(b)) + (pro(a) + pro(b)) < con(a)$, since R solves α; (3) $con(a) \leq sup(s_4)$, implying $con(a) \leq sup(s_0) - 2(con(a) + con(b)) + 2(pro(a) + pro(b))$, since a occurs at s_4. If we combine (1) and (2), then we get $-(con(a) + con(b)) + (pro(a) + pro(b)) < 0$; if we combine (2) and (3), then we get $0 \leq -(con(a) + con(b)) + (pro(a) + pro(b))$. Since this is a contradiction, R cannot exist and thus α is not solvable.

Example 2 (Language-Simulation). The TS A_2 of Figure 1 does not have any ESSP atom, since the only event a occur at all states. Hence, A_2 has the ESSP. The set $\mathcal{R} = \emptyset$ is a witness of A_2's ESSP. The reachability graph A_{N_1} of the synthesized net $N_1 = N^{\mathcal{R}}_{A_2} = (\emptyset, \{a\}, f, M_0)$ simulates A_2 up to language equivalence, cf. Figure 3. A language-simulation φ is defined by $\varphi(q_0) = \varphi(q_1) = 0$.

On the other hand, A_2 does not have the SSP, since the SSP atom $\alpha = (s_0, s_1)$ is not solvable. This can be seen as follows: If $R = (sup, con, pro)$ is a region that solves α, then (1) $sup(s_0) \neq sup(s_1)$; (2) $sup(s_0) = sup(s_1) - con(a) + con(a)$; (3) $sup(s_1) = sup(s_0) - con(a) + con(a)$. One easily verifies that the combination of (2) and (3) implies that $sup(s_0) = sup(s_1)$, which contradicts (1). Hence, R cannot exist and α is not solvable.

Example 3 (Realization of an E'-Label Splitting). Let A_2 and A_3 be in accordance to Figure 1. The TS A_2 has the event set $E = \{a\}$. The label-splitting of the event a into the events a and a' yields the event set $E' = (E \setminus \{a\}) \cup \{a, a'\} = \{a, a'\}$. The TS A_3 is an E'-label-splitting of A_2: For $q_0 \xrightarrow{a} q_1 \in A_2$ there is exactly one $x \in \{a, a'\}$, namely $x = a$, such that $q_0 \xrightarrow{x} q_1 \in A_3$ and, for $q_1 \xrightarrow{a} q_0 \in A_2$, there is exactly one $y \in \{a, a'\}$, namely $y = a'$, such that $q_1 \xrightarrow{y} q_0 \in A_3$. The set $\{a\}$ is the set of events of A_2 that occur split in A_3.

The TS A_3 has both the SSP and the ESSP. More exactly, A_3 has the SSP atom $\alpha_1 = (q_0, q_1)$ and the ESSP atoms $\alpha_2 = (a, q_1)$ and $\alpha_3 = (a', q_0)$. The region $R_1 = (sup_1, con_1, pro_1)$, where $sup_1(q_0) = 1$, $sup_1(q_1) = 0$, $con_1(a) = pro_1(a') = 1$ and $pro_1(a) = con_1(a') = 0$ solves α_1 and α_2. Moreover, the region $R_2 = (sup_2, con_2, pro_2)$, where $sup_2(q_0) = 0$, $sup_2(q_1) = 1$, $con_2(a') = pro_2(a) = 1$ and $pro_2(a') = con_2(a) = 0$ solves α_3. Thus, $\mathcal{R} = \{R_1, R_2\}$ is a witness for A_3's SSP and ESSP. Figure 3 depicts the synthesized net $N_2 = N^{\mathcal{R}}_{A_3}$ and it is easy to see that its reachability graph A_{N_2} is isomorphic to A_3.

3 Label Splitting for Language-Simulation and Exact Realization

The following theorem presents our main result and states that label-splitting aiming at language-simulation or realization is NP-complete.

Theorem 1. *Let A be a linear TS and $\kappa \in \mathbb{N}$. Deciding whether there is an E'-label-splitting B of A that satisfies $|E'| \leq \kappa$ and allows a Petri net N such that $B \triangleright A_N$ or $B \cong A_N$ is NP-complete.*

The proof of Theorem 1 bases on a polynomial-time reduction of the vertex cover (VC) problem on cubic graphs, which is known to be NP-complete from [13]:
CUBIC VERTEX COVER (CVC)

Input: a Graph $G = (V, \Sigma)$ with set of vertices V and edges Σ such that every $v \in V$ is a member of exactly three distinct $\sigma_0, \sigma_1, \sigma_2 \in \Sigma$, a number $\lambda \in \mathbb{N}$.

Decide: whether there is a (λ-VC) $M \subseteq V$ satisfying $|M| \leq \lambda$ and $M \cap \sigma \neq \emptyset$ for all $\sigma \in \Sigma$.

Example 4 (CVC). The instance $(G, 3)$, where $G = (V, \Sigma)$ such that $V = \{v_0, v_1, v_2, v_3\}$ and $\Sigma = \{\{v_0, v_1\}, \{v_0, v_2\}, \{v_0, v_3\}, \{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}\}$, allows a positive decision, since $M = \{v_0, v_1, v_2\}$ is a 3-VC of G.

In the remainder of this paper, if not explicitly stated otherwise, let (G, λ) be an arbitrary but fixed input of CVC, where $G = (V, \Sigma)$ has n vertices $V = \{v_0, \dots, v_{n-1}\}$ and m edges $\Sigma = \{\sigma_0, \dots, \sigma_{m-1}\}$ such that $\sigma_i = \{v_{i_0}, v_{i_1}\}$ for all $i \in \{0, \dots, m-1\}$. For technical reasons, we assume without loss of generality that $i_0 < i_1$ for the vertices v_{i_0}, v_{i_1} of the edge σ_i for all $i \in \{0, \dots, m-1\}$.

For the proof of Theorem 1, we reduce (G, λ) to a pair (A, κ) of *linear* TS A and natural number κ as follows. If there is an E'-label-splitting B of A satisfying $|E'| \leq \kappa$ that has the ESSP, then G has a λ-VC. Hence, if B allows a language-simulation (implying its ESSP) or a realization (implying its SSP and its ESSP), then G has a λ-VC. Conversely, if G has a λ-VC, then there is an E'-label-splitting B of A satisfying $|E'| \leq \kappa$ that has a set $\mathcal{R}$ of regions that witnesses both B's ESSP and SSP, which, by Lemma 1, implies both a language-simulation and a realization for B. Thus, (G, λ) is a yes-instance if and only if (A, κ) is a yes-instance, according to the implementation sought.

For a start, we define $\kappa = n + 2(m-1) + \lambda$, where $n + 2(m-1)$ is the number of events of the announced TS A. Hence, λ corresponds to the maximum number of events of A that could possibly be split for a sought E'-label-splitting B. For every $i \in \{0, \dots, m-1\}$, the TS A has the following directed path T_i that uses the vertices of the edge $\sigma_i = \{v_{i_0}, v_{i_1}\}$ as events:

$$T_i = t_{i,0} \xrightarrow{v_{i_0}} t_{i,1} \xrightarrow{v_{i_1}} t_{i,2} \xrightarrow{v_{i_1}} t_{i,3} \xrightarrow{v_{i_0}} t_{i,4} \xrightarrow{v_{i_0}} t_{i,5}$$

We use the states $\bot_1, \dots, \bot_{m-1}$ and events $y_1, \dots, y_{m-1}, z_1, \dots, z_{m-1}$ and, for all $i \in \{1, \dots, m-1\}$, the edges $t_{i-1,5} \xrightarrow{y_i} \bot_i$ and $\bot_i \xrightarrow{z_i} t_{i,0}$ to connect $T_0, \dots, T_{m-1}$ to finally build A. The resulting linear TS can be sketched as follows:

$$A = \quad T_0 \xrightarrow{y_1} \bot_1 \xrightarrow{z_1} T_1 \xrightarrow{y_2} \bot_2 \xrightarrow{z_2} \cdots \xrightarrow{y_{m-1}} \bot_{m-1} \xrightarrow{z_{m-1}} T_{m-1}$$

We summarize events and states by $Y = \{y_1, \dots, y_{m-1}\}$, $Z = \{z_1, \dots, z_{m-1}\}$ and $\bot = \{\bot_1, \dots, \bot_{m-1}\}$. Notice that A has $|V \cup Y \cup Z| = n + 2 \cdot (m-1)$ events.

Lemma 2. *If there is an E'-label-splitting B of A that satisfies $|E'| \leq \kappa$ and has the ESSP, then G has a λ-vertex cover.*

Proof. Let B be an E'-label-splitting of A that satisfies $|E'| \leq \kappa$ and has the ESSP. Let $\mathfrak{E} \subseteq E'$ be the set of events of A that occur split in B. By definition of label-splitting, for every $e \in \mathfrak{E}$, there are at least two distinct events e_1, e_2 in E' that originate from the splitting of e and do not correspond to the splitting of another event $e' \in \mathfrak{E} \setminus \{e\}$. By $|E| = n + 2 \cdot (m-1)$ and $|E'| \leq n + 2 \cdot (m-1) + \lambda$, this implies $|\mathfrak{E}| \leq \lambda$.

Let $i \in \{0, \dots, m-1\}$. Notice that the TS A_1 of Figure 1 and the path T_i (when considered as a TS) are isomorphic, that is, with the exception of names for states and events, they are structurally the same. Hence, just like in Example 1, one argues that if a TS has the path T_i, then it cannot have the ESSP, since the atom $(v_{i_0}, t_{i,2})$ is not solvable. Consequently, since B has the ESSP, the path T_i

$$t_{0,0} \xrightarrow{v_0} t_{0,1} \xrightarrow{v_1} t_{0,2} \xrightarrow{v_1} t_{0,3} \xrightarrow{v_0} t_{0,4} \xrightarrow{v_0} t_{0,5} \qquad t_{1,0} \xrightarrow{v_0} t_{1,1} \xrightarrow{v_2} t_{1,2} \xrightarrow{v_2} t_{1,3} \xrightarrow{v_0} t_{1,4} \xrightarrow{v_0} t_{1,5}$$

$$t_{2,0} \xrightarrow{v_0} t_{2,1} \xrightarrow{v_3} t_{2,2} \xrightarrow{v_3} t_{2,3} \xrightarrow{v_0} t_{2,4} \xrightarrow{v_0} t_{2,5} \qquad t_{3,0} \xrightarrow{v_1} t_{3,1} \xrightarrow{v_2} t_{3,2} \xrightarrow{v_2} t_{3,3} \xrightarrow{v_1} t_{3,4} \xrightarrow{v_1} t_{3,5}$$

$$t_{4,0} \xrightarrow{v_1} t_{4,1} \xrightarrow{v_3} t_{4,2} \xrightarrow{v_3} t_{4,3} \xrightarrow{v_1} t_{4,4} \xrightarrow{v_1} t_{4,5} \qquad t_{5,0} \xrightarrow{v_2} t_{5,1} \xrightarrow{v_3} t_{5,2} \xrightarrow{v_3} t_{5,3} \xrightarrow{v_2} t_{5,4} \xrightarrow{v_2} t_{5,5}$$

Fig. 4: The paths $T_0, \dots, T_5$ of TS A for Theorem 1 that originates from Example 4.

$$t_{0,0} \xrightarrow{v_0'} t_{0,1} \xrightarrow{v_1} t_{0,2} \xrightarrow{v_1'} t_{0,3} \xrightarrow{v_0} t_{0,4} \xrightarrow{v_0} t_{0,5} \qquad t_{1,0} \xrightarrow{v_0'} t_{1,1} \xrightarrow{v_2} t_{1,2} \xrightarrow{v_2'} t_{1,3} \xrightarrow{v_0} t_{1,4} \xrightarrow{v_0} t_{1,5}$$

$$t_{2,0} \xrightarrow{v_0} t_{2,1} \xrightarrow{v_3} t_{2,2} \xrightarrow{v_3} t_{2,3} \xrightarrow{v_0'} t_{2,4} \xrightarrow{v_0} t_{2,5} \qquad t_{3,0} \xrightarrow{v_1'} t_{3,1} \xrightarrow{v_2} t_{3,2} \xrightarrow{v_2'} t_{3,3} \xrightarrow{v_1} t_{3,4} \xrightarrow{v_1} t_{3,5}$$

$$t_{4,0} \xrightarrow{v_1} t_{4,1} \xrightarrow{v_3} t_{4,2} \xrightarrow{v_3} t_{4,3} \xrightarrow{v_1'} t_{4,4} \xrightarrow{v_1} t_{4,5} \qquad t_{5,0} \xrightarrow{v_2} t_{5,1} \xrightarrow{v_3} t_{5,2} \xrightarrow{v_3} t_{5,3} \xrightarrow{v_2'} t_{5,4} \xrightarrow{v_2} t_{5,5}$$

Fig. 5: According to the reduction for Theorem 1, the paths $T_0', \dots, T_5'$ of the E'-label-splitting B of A that originates from the VC $M = \{v_0, v_1, v_2\}$ of Example 4.

(or an isomorphic version of it) cannot be present in B. Since B is an E'-label-splitting of A, this implies that there is an event $e \in \{v_{i_0}, v_{i_1}\}$ such that $e \in \mathfrak{E}$. Since i was arbitrary, this is simultaneously true for all $T_0, \dots, T_{m-1}$. Hence, if $M = V \cap \mathfrak{E}$, then $M \cap E(T_i) \neq \emptyset$, implying $M \cap \sigma_i \neq \emptyset$, for all $i \in \{0, \dots, m-1\}$. Finally, by $|M| = |V \cap \mathfrak{E}| \leq |\mathfrak{E}| \leq \lambda$, we get that M defines a λ-VC of G. □

Conversely, we have to show that if G has a λ-VC, then there is a searched E'-label-splitting for A. Let $M = \{v_{j_0}, \dots, v_{j_{\lambda-1}}\} \subseteq V$ be a λ-VC of G. For every $i \in \{0, \dots, \lambda - 1\}$, we split the event v_{j_i} into the two events v_{j_i} and v_{j_i}'. This yields $E' = (E \setminus M) \cup \bigcup_{i=0}^{\lambda-1} \{v_{j_i}, v_{j_i}'\}$. To define the announced E'-label-splitting $B = (S, E', \delta', t_{0,0})$ of A, it is sufficient to define δ' on the states of $T_0, \dots, T_{m-1}$. For all $i \in \{0, \dots, m-1\}$, δ' restricted to $S(T_i)$ yields the path T_i' as follows:

- if $v_{i_0} \in M$ and $v_{i_1} \notin M$, then $T_i' = t_{i,0} \xrightarrow{v_{i_0}} t_{i,1} \xrightarrow{v_{i_1}} t_{i,2}, \xrightarrow{v_{i_1}} t_{i,3} \xrightarrow{v_{i_0}'} t_{i,4} \xrightarrow{v_{i_0}} t_{i,5}$;
- if $v_{i_0}, v_{i_1} \in M$, then $T_i' = t_{i,0} \xrightarrow{v_{i_0}'} t_{i,1} \xrightarrow{v_{i_1}} t_{i,2}, \xrightarrow{v_{i_1}'} t_{i,3} \xrightarrow{v_{i_0}} t_{i,4} \xrightarrow{v_{i_0}} t_{i,5}$;
- if $v_{i_0} \notin M$ and $v_{i_1} \in M$, then $T_i' = t_{i,0} \xrightarrow{v_{i_0}} t_{i,1} \xrightarrow{v_{i_1}} t_{i,2}, \xrightarrow{v_{i_1}'} t_{i,3} \xrightarrow{v_{i_0}} t_{i,4} \xrightarrow{v_{i_0}} t_{i,5}$.

By the following lemma, mappings *sup* and *con*, *pro* defined on the states and events of $T_0', \dots, T_{m-1'}$ that, for all $i \in \{0, \dots, m-1\}$, behave like a region of T_i' (when restricted to T_i') can be extended to a region $R = (sup', con', pro')$ of B:

Lemma 3. *Let* $sup : S \setminus \bot \to \mathbb{N}$ *and* $con, pro : E' \setminus (Y \cup Z) \to \mathbb{N}$ *be mappings such that if* $e \in \{v_{i_0}, v_{i_0}', v_{i_1}, v_{i_1}'\}$ *and* $t_{i,j} \xrightarrow{e} t_{i,j+1} \in B$, *then* $sup(t_{i,j}) \leq con(e)$ *and* $sup(t_{i,j+1}) = sup(t_{i,j}) - con(e) + pro(e)$ *for all* $i \in \{0, \dots, m-1\}$ *and* $j \in \{0, \dots, 4\}$. *If* sup', con', pro' *are defined as follows, then* $R = (sup', con', pro')$ *is* *a region of* B*: for all* $s \in S$*, if* $s \in \perp$*, then* $sup'(s) = 0$*, otherwise* $sup'(s) = sup(s)$*; for all* $e \in E'$ *and all* $i \in \{0, \dots, m-1\}$*, if* $e = y_i$*, then* $con'(e) = sup(t_{i-1,4})$ *and* $pro'(e) = 0$*; if* $e = z_i$*, then* $con'(e) = 0$ *and* $pro'(e) = sup(t_{i,0})$*; otherwise* $con'(e) = con(e)$ *and* $pro'(e) = pro(e)$*.*

Proof. By the assumption about sup, con and pro, it is easy to see that $s \xrightarrow{e} s' \in B$ implies $sup'(s) \leq con'(e)$ and $sup'(s') = sup'(s) - con'(e) + pro'(e)$. □

The next lemma states that linear TSs and thus especially B have the SSP:

Lemma 4. *If* $A = s_0 \xrightarrow{e_1} \dots s_{i-1} \xrightarrow{e_i} s_i \xrightarrow{e_{i+1}} \dots \xrightarrow{e_n} s_n$ *is a linear TS, then* A *has the SSP. Moreover, if the event* e_i *occurs exactly once, then the atom* (e_i, s) *is solvable for all* $s \in \{s_0, \dots, s_n\} \setminus \{s_{i-1}\}$*.*

Proof. The following region $R = (sup, con, pro)$ solves all SSP atoms on one blow: $sup(s_0) = 0$; for all $j \in \{1, \dots, n\}$, $con(e_j) = 0$ and $pro(e_j) = 1$.

The following region $R = (sup, con, pro)$ solve (e_i, s) for all $s \in \{s_0, \dots, s_{i-2}\}$: $sup(s_0) = 0$; for all $j \in \{1, \dots, n\}$, if $j = i$, then $con(e_j) = i$ and $pro(e_j) = 0$; otherwise $con(e_j) = 0$ and $pro(e_j) = 1$.

The following region $R = (sup, con, pro)$ solve (e_i, s) for all $s \in \{s_i, \dots, s_n\}$: $sup(s_0) = 1$; for all $j \in \{1, \dots, n\}$, if $j = i$, then $con(e_j) = 1$ and $pro(e_j) = 0$; otherwise $con(e_j) = pro(e_j) = 0$. □

Lemma 5. *There is a set* $\mathcal{R}$ *of regions witnessing the ESSP and the SSP of* B*.*

Proof. By Lemma 4, B has the SSP and the atom (e, s) is solvable for all $e \in Y \cup Z$ and (relevant) $s \in S$.

Let $x \in E' \setminus (Y \cup Z)$ be arbitrary but fixed. Since G is cubic, there are exactly three indices $i < j < k \in \{0, \dots, m-1\}$, such that $x \in T'_\ell$ for all $\ell \in \{i, j, k\}$. Let $no_\ell(x)$ denote the number of x's occurrences in T'_ℓ for all $\ell \in \{i, j, k\}$. The following region solves (x, s) for all $s \in B \setminus (S(T'_i) \cup S(T'_j) \cup S(T'_k))$: If $i = 0$, then $sup(t_{0,0}) = no_0(e)$, otherwise $sup(t_{0,0}) = 0$. For all $e \in E'$ and $\ell \in \{i, j, k\}$, if $e = z_\ell$, then $con(e) = 0$ and $pro(e) = no_\ell(x)$; if $e = x$, then $con(e) = 1$ and $pro(e) = 0$; otherwise $con(e) = pro(e) = 0$.

It remains to argue that (x, s) is also solvable when both x and s belong to the same gadget T'_i, $i \in \{0, \dots, m-1\}$. To do so, we proceed as follows. Let $i \in \{0, \dots, m-1\}$; We argue, for all the (possible) cases $v_{i_0} \in M, v_{i_1} \not\in M$ and $v_{i_0} \not\in M, v_{i_1} \in M$ and $v_{i_0}, v_{i_1} \in M$, that (x, s) is solvable for all relevant events $x \in \{v_{i_0}, v_{i_1}, v'_{i_0}, v'_{i_1}\}$ and states $s \in S(T'_i)$. By the arbitrariness of i, this finally proves the ESSP for B. By Lemma 3, it suffices to define mappings for $T'_0, \dots, T'_{m-1}$. Let $S = \{t_{0,0}, \dots, t_{m-1,0}\}$ and $E = E' \setminus (Y \cup Z)$. Let's start with the case $v_{i_0} \in M, v_{i_1} \not\in M$, which implies $T'_i = t_{i,0} \xrightarrow{v_{i_0}} t_{i,1} \xrightarrow{v_{i_1}} t_{i,2}, \xrightarrow{v_{i_1}} t_{i,3} \xrightarrow{v'_{i_0}} t_{i,4} \xrightarrow{v_{i_0}} t_{i,5}$.

(v_{i_0} and v'_{i_0}): Let $j \neq k \in \{0, \dots, m-1\} \setminus \{i\}$ be such that $v_{i_0} \in \sigma_j \cap \sigma_k$. The following region $R = (sup, con, pro)$ solves (v_{i_0}, s) for all $s \in \{t_{i,1}, t_{i,2}, t_{i,3}, t_{i,5}\}$: For all $s \in S$, if $s = t_{i,0}$, then $sup(s) = 1$, if $s \in \{t_{j,0}, t_{k,0}\}$, then $sup(s) = 2$; otherwise $sup(s) = 0$. For all $e \in E$, if $e = v_{i_0}$, then $con(e) = 1$ and $pro(e) = 0$

(notice that $no_j(e), no_\ell(e) \leq 2$, since $v_{i_0} \in M$); if $e = v'_{i_0}$, then $con(e) = 0$ and $pro(e) = 1$; otherwise, $con(e) = pro(e) = 0$.

The following region $R = (sup, con, pro)$ solves the SSP atom (v'_{i_0}, s) for all $s \in \{t_{i,0}, t_{i,1}, t_{i,2}, t_{i,4}, t_{i,5}\}$: For all $s \in S$, if $s \in \{t_{j,0}, t_{k,0}\}$, then $sup(s) = 2$; otherwise $sup(s) = 0$. For all $e \in E$, if $e = v'_{i_0}$, then $con(e) = 2$ and $pro(e) = 0$ (notice that $no_j(e), no_\ell(e) \leq 1$); if $e = v_{i_1}$, then $con(e) = 0$ and $pro(e) = 1$; otherwise, $con(e) = pro(e) = 0$.

(v_{i_1}): Let $j \neq k \in \{0, \dots, m-1\} \setminus \{i\}$ be such that $v_{i_1} \in \sigma_j \cap \sigma_k$. The following region $R = (sup, con, pro)$ solves (v_{i_1}, s) for all $s \in \{t_{i,0}, t_{i,3}, t_{i,4}\}$: For all $s \in S$, if $s \in \{t_{j,0}, t_{k,0}\}$, then $sup(s) = 3$; otherwise $sup(s) = 0$. For all $e \in E$, if $e = v_{i_1}$, then $con(e) = 1$ and $pro(e) = 0$ (notice that $no_j(e), no_\ell(e) \leq 3$); if $e = v_{i_0}$, then $con(e) = 0$ and $pro(e) = 2$; otherwise, $con(e) = pro(e) = 0$.

The following region $R = (sup, con, pro)$ solves $(v_{i_1}, t_{i,5})$: For all $s \in S$, if $s = t_{i,0}$, then $sup(s) = 2$; if $s \in \{t_{j,0}, t_{k,0}\}$, then $sup(s) = 3$; otherwise $sup(s) = 0$. For all $e \in E$, if $e = v_{i_1}$, then $con(e) = 1$ and $pro(e) = 0$; otherwise, $con(e) = pro(e) = 0$.

Similarly, one finds solving regions for all relevant ESSP atoms (x, s) of T'_i for the remaining cases $v_{i_0} \notin M, v_{i_1} \in M$ and $v_{i_0}, v_{i_1} \in M$. By the arbitrariness of i, this proves the lemma. $\square$

4 Label Splitting for Embedding

In [18], it has been shown that label-splitting aiming at embedding is NP-complete. The six *strands* of the reduction presented in [18] can be consecutively arranged such that the resulting TS is 3-bounded. Thus, the problem is also NP-complete for 3-bounded inputs. In this section, we strengthen this result and show that label-splitting aiming at embedding remains NP-complete even for 2-bounded inputs and that this bound is tight.

Theorem 2. *Let A be a b-bounded TS and $\kappa \in \mathbb{N}$. Deciding if there is an E'-label-splitting B of A that satisfies $|E'| \leq \kappa$ and allows a Petri net N such that $A \hookrightarrow A_N$ is NP-complete if $b > 1$, otherwise it is polynomial.*

The following lemma paves us the way for being able to prove easily the polynomial-time statement of Theorem 2:

Lemma 6. *Let $A = s_0 \xrightarrow{e_1} \dots \xrightarrow{e_n} s_n$ be a directed cycle TS, i.e., $s_0 = s_n$ and $s_i \neq s_j$ for all $i \neq j \in \{0, \dots, n-1\}$. If there is an $i \in \{0, \dots, n-1\}$ such that $\neg s_j \xrightarrow{e_i}$ for all $j \in \{0, \dots, n\} \setminus \{i-1\}$, that is, e_i occurs exactly once in A, then A has a set $\mathcal{R}$ of regions that witnesses its SSP.*

Proof. We argue that any SSP atom (s, s') of A is solvable. To do so, we assume without loss of generality that $s_0 = s$ and $s_j = s'$ and $i > j$, that is,

$$A = s_0 \xrightarrow{e_1} \dots \xrightarrow{e_j} s_j \xrightarrow{e_{j+1}} \dots \xrightarrow{e_{j+k}} s_{i-1} \xrightarrow{e_i} s_i \xrightarrow{e_{i+1}} \dots \xrightarrow{e_{i+\ell}} s_n$$

where $i-1 = j+k$ and $n = j+k+\ell+1$. This assumption is no essential restriction, since A is a cycle and thus can be transformed in the desired form by a simple renaming of its states. The following region $R = (sup, con, pro)$ solves (s_0, s_j): $sup(s_0) = \ell$; for all $e \in E(A)$, if $e \neq e_i$, then $con(e) = 0$ and $pro(e) = 1$; if $e = e_i$, then $con(e) = j+k+\ell$ and $pro(e) = 0$. This implies $sup(s_0) = \ell \neq \ell + j = sup(s_j)$, since $\ell > 0$. Thus, R separates (s, s'). $\square$

If $A = (S, E, \delta, \iota)$ is a directed cycle that does not have the SSP, implying that every event occurs at least twice by Lemma 6, then we obtain an embeddable E'-label-splitting B of A as follows: Let $q \xrightarrow{x} q' \in A$ be arbitrary but fixed and $E' = (E \setminus \{x\}) \cup \{x, x'\}$. The E'-label-splitting $B = (S, E', \delta, \iota)$ that originates from A by relabeling the edge $q \xrightarrow{x} q'$ by $q \xrightarrow{x'} q'$ (and nothing else), obviously, satisfies $|E'| = |E| + 1$. Moreover, since x' occurs only once, B has the SSP by Lemma 6. Hence, if A is a 1-bounded TS that has the SSP, which is particularly implied if it is linear, then (A, κ) is a yes-instance if and only if $|E| \leq \kappa$. Otherwise, (A, κ) is a yes-instance if and only if $|E| + 1 \leq \kappa$.

Consequently, to complete the proof of Theorem 2, it remains to consider the case where A is 2-bounded. To do so, we reduce an input (G, λ) of CVC to a pair (A, κ) of 2-bounded TS $A = (S, E, \delta, t_{0,0})$ and natural number κ such that G has a λ-VC if and only if there is an E'-label-splitting B of A that satisfies $|E'| \leq \kappa$ and has a set $\mathcal{R}$ of regions that witnesses the SSP of B.

Let's introduce (A, κ). For a start, we define $\kappa = n+m-1+\lambda$, where $n+m-1$ is the number of events of A. Hence, λ corresponds to the maximum number of events of A that could possibly be split for a sought E'-label-splitting B of A. For every $i \in \{0, \dots, m-1\}$, the TS A has the following gadget T_i that uses the vertices of the edge $\sigma_i = \{v_{i_0}, v_{i_1}\}$ of G as events:

$$T_i = \begin{tikzcd} & t_{i,1} \arrow[dr, bend left, "v_{i_0}"] & \\ t_{i,0} \arrow[ur, bend left, "v_{i_0}"] \arrow[dr, bend right, "v_{i_1}"'] & & t_{i,2} \\ & t_{i,3} \arrow[ur, bend right, "v_{i_1}"'] & \end{tikzcd}$$

We use the events $y_1, \dots, y_{m-1}$ and, for all $i \in \{1, \dots, m-1\}$, the edge $t_{i-1,2} \xrightarrow{y_i} t_{i,0}$ to connect $T_0, \dots, T_{m-1}$ and build A, which can be sketched as follows:

$$A = \quad T_0 \xrightarrow{y_1} T_1 \xrightarrow{y_2} \cdots \xrightarrow{y_{m-2}} T_{m-2} \xrightarrow{y_{m-1}} T_{m-1}$$

The initial state of A is $t_{0,0}$. Let $Y = \{y_1, \dots, y_{m-1}\}$. Notice that A is 2-bounded and has exactly $|V \cup Y| = n + m - 1$ events.

If a TS has any of $T_0, \dots, T_{m-1}$, then it does not have the SSP:

Lemma 7. *If A is a TS that has the paths $P_1 = p_0 \xrightarrow{a} p_1 \xrightarrow{a} p_2$ and $P_2 = p_0 \xrightarrow{b} p_3 \xrightarrow{b} p_2$ with the same starting state p_0 and final state p_2, then A does not have the SSP, since the atom (p_1, p_3) is not solvable.*

Proof. If $R = (sup, con, pro)$ is a region of A, then we have (1) $sup(p_1) = sup(p_0) - con(a) + pro(a)$ and (2) $sup(p_2) = sup(p_0) - 2con(a) + 2pro(a)$ and (3) $sup(p_3) = sup(p_0) - con(b) + pro(b)$ and (4) $sup(p_2) = sup(p_0) - 2con(b) + 2pro(b)$. We subtract (4) from (2), rearrange the resulting equation properly and obtain $-con(b) + pro(b) = -con(a) + pro(a)$. By (1) and (3), this implies $sup(p_1) = sup(p_3)$. Thus, R does not solve (p_1, p_3). R was arbitrary, hence the claim. $\square$

In fact, for all $i \in \{0, \dots, m-1\}$, if a TS has the gadget T_i, then the SSP atom $(t_{i,1}, t_{i,3})$ is not solvable by Lemma 7. By the following lemma, this implies that a searched E'-label-splitting B of A implies a λ-VC of G:

Lemma 8. *If there is an E'-label-splitting B of A that satisfies $|E'| \leq \kappa$ and has the SSP, then G has a λ-VC.*

Proof. Let B be a fitting E'-label-splitting and $\mathfrak{E} \subseteq E$ the set of events of A that occur split in B, implying $|\mathfrak{E}| \leq \lambda$ by $|E'| \leq \kappa$. Since B has the SSP, the atom $(t_{i,1}, t_{i,3})$ is solvable for all $i \in \{0, \dots, m-1\}$. Consequently, by Lemma 7, none of $T_0, \dots, T_{m-1}$ is present in B. Since B is a E'-label-splitting of A, this implies $\mathfrak{E} \cap E(T_i) \neq \emptyset$ for all $i \in \{0, \dots, m-1\}$. Hence, $M = V \cap \mathfrak{E}$ is a λ-VC of G. $\square$

Conversely, we show that if G has a λ-VC, then there is a searched E'-label-splitting for A. Let $M = \{v_{j_0}, \dots, v_{j_{\lambda-1}}\} \subseteq V$ be a λ-VC of G and E' be defined as in Section 3. To obtain the announced E'-label-splitting $B = (S, E', \delta', t_{0,0})$, it is sufficient again to define δ' on the states of $T_0, \dots, T_{m-1}$. In particular, for all $i \in \{0, \dots, m-1\}$, we get T'_i from T_i by δ' as follows:

- if $v_{i_0} \in M$ and $v_{i_1} \notin M$, then $t_{i,0} \xrightarrow{v_{i_0}} t_{i,1} \xrightarrow{v'_{i_0}} t_{i,2}$ and $t_{i,0} \xrightarrow{v_{i_1}} t_{i,3} \xrightarrow{v_{i_1}} t_{i,2}$;
- if $v_{i_0}, v_{i_1} \in M$, then $t_{i,0} \xrightarrow{v_{i_0}} t_{i,1} \xrightarrow{v'_{i_0}} t_{i,2}$ and $t_{i,0} \xrightarrow{v_{i_1}} t_{i,3} \xrightarrow{v'_{i_1}} t_{i,2}$
- if $v_{i_0} \notin M$ and $v_{i_1} \in M$, then $t_{i,0} \xrightarrow{v_{i_0}} t_{i,1} \xrightarrow{v_{i_0}} t_{i,2}$ and $t_{i,0} \xrightarrow{v_{i_1}} t_{i,3} \xrightarrow{v'_{i_1}} t_{i,2}$

Lemma 9. *There is a set $\mathcal{R}$ of regions that witnesses the SSP of B.*

Proof. Let $i \in \{0, \dots, m-1\}$ be arbitrary but fixed. The following region $R = (sup, con, pro)$ solves (s, s') for all states $s \neq s' \in S$ that satisfy $(s, s') \notin \{(t_{i,1}, t_{i,3}), (t_{i,3}, t_{i,1}) \mid i \in \{0, \dots, m-1\}\}$: $sup(t_{0,0}) = 0$; for all $e \in E'$, $con(e) = 0$ and $pro(e) = 1$.

Let $i \in \{0, \dots, m-1\}$ be arbitrary but fixed. The following region $R = (sup, con, pro)$ solves $(t_{i,1}, t_{i,3})$: $sup(t_{0,0}) = 0$; for all $e \in E'$, if $v_{i_0} \in M$, then $con(v_{i_0}) = 0$, $pro(v_{i_0}) = 1$, $con(v'_{i_0}) = 1$, $pro(v'_{i_0}) = 0$ and $con(e) = pro(e) = 0$ if $e \notin \{v_{i_0}, v'_{i_0}\}$; otherwise, if $v_{i_0} \notin M$, which implies $v_{i_1} \in M$, then $con(v_{i_1}) = 0$, $pro(v_{i_1}) = 1$, $con(v'_{i_1}) = 1$, $pro(v'_{i_1}) = 0$ and $con(e) = pro(e) = 0$ if $e \notin \{v_{i_1}, v'_{i_1}\}$. Since i was arbitrary, the atom $(t_{i,1}, t_{i,3})$ is solvable for all $i \in \{0, \dots, m-1\}$. $\square$

5 Conclusion

In this paper, we completely characterize the label-splitting problem for Petri nets for all types of implementations that have previously been studied in the literature. By doing so, we answer an open question that has been posed in [18, p. 17]. Moreover, we strengthen the result from [18] to 2-bounded inputs, and show that this bound is tight. It remains future work to consider the maximum number κ of labels of the splitting as a parameter in order to investigate the problem from a parameterized complexity point of view and to determine if this parameterization makes the label-splitting problem fixed-parameter-tractable.

Acknowledgements. I would like to thank the unknown reviewers for their valuable comments.

References

1. van der Aalst, W.M.P.: Process Mining - Discovery, Conformance and Enhancement of Business Processes. Springer (2011). https://doi.org/10.1007/978-3-642-19345-3
2. Badouel, E., Bernardinello, L., Darondeau, P.: Polynomial algorithms for the synthesis of bounded nets. In: TAPSOFT. Lecture Notes in Computer Science, vol. 915, pp. 364–378. Springer (1995). https://doi.org/10.1007/3-540-59293-8_207
3. Badouel, E., Bernardinello, L., Darondeau, P.: Petri Net Synthesis. Texts in Theoretical Computer Science. An EATCS Series, Springer (2015). https://doi.org/10.1007/978-3-662-47967-4
4. Badouel, E., Caillaud, B., Darondeau, P.: Distributing finite automata through Petri net synthesis. Formal Asp. Comput. **13**(6), 447–470 (2002). https://doi.org/10.1007/s001650200022
5. Best, E., Devillers, R.R.: Characterisation of the state spaces of live and bounded marked graph Petri nets. In: LATA. Lecture Notes in Computer Science, vol. 8370, pp. 161–172. Springer (2014). https://doi.org/10.1007/978-3-319-04921-2_13
6. Best, E., Devillers, R.R.: Synthesis of bounded choice-free petri nets. In: CONCUR. LIPIcs, vol. 42, pp. 128–141. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2015). https://doi.org/10.4230/LIPIcs.CONCUR.2015.128
7. Best, E., Erofeev, E., Schlachter, U., Wimmel, H.: Characterising petri net solvable binary words. In: Kordon, F., Moldt, D. (eds.) Application and Theory of Petri Nets and Concurrency - 37th International Conference, PETRI NETS 2016, Toruń, Poland, June 19-24, 2016. Proceedings. Lecture Notes in Computer Science, vol. 9698, pp. 39–58. Springer (2016). https://doi.org/10.1007/978-3-319-39086-4_4, https://doi.org/10.1007/978-3-319-39086-4_4
8. Carmona, J.: The label splitting problem. Trans. Petri Nets Other Model. Concurr. **6**, 1–23 (2012). https://doi.org/10.1007/978-3-642-35179-2_1, https://doi.org/10.1007/978-3-642-35179-2_1
9. Carmona, J., Cortadella, J., Kishinevsky, M.: New region-based algorithms for deriving bounded petri nets. IEEE Trans. Computers **59**(3), 371–384 (2010). https://doi.org/10.1109/TC.2009.131, https://doi.org/10.1109/TC.2009.131
10. Cortadella, J., Kishinevsky, M., Kondratyev, A., Lavagno, L., Yakovlev, A.: A region-based theory for state assignment in speed-independent circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **16**(8), 793–812 (Aug 1997). https://doi.org/10.1109/43.644602

11. Cortadella, J., Kishinevsky, M., Lavagno, L., Yakovlev, A.: Deriving petri nets from finite transition systems. IEEE Transactions on Computers **47**(8), 859–882 (1998)
12. Cortadella, J., Kishinevsky, M., Kondratyev, A., Lavagno, L., Yakovlev, A.: A region-based theory for state assignment in speed-independent circuits. IEEE Trans. on CAD of Integrated Circuits and Systems **16**(8), 793–812 (1997). https://doi.org/10.1109/43.644602
13. Fricke, G., Hedetniemi, S.T., Jacobs, D.P.: Independence and irredundance in k-regular graphs. Ars Comb. **49** (1998)
14. Holloway, L.E., Krogh, B.H., Giua, A.: A survey of Petri net methods for controlled discrete event systems. Discrete Event Dynamic Systems **7**(2), 151–190 (1997). https://doi.org/10.1023/A:1008271916548
15. Lu, X., Fahland, D., van den Biggelaar, F.J.H.M., van der Aalst, W.M.P.: Handling duplicated tasks in process discovery by refining event labels. In: Rosa, M.L., Loos, P., Pastor, O. (eds.) Business Process Management - 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings. Lecture Notes in Computer Science, vol. 9850, pp. 90–107. Springer (2016). https://doi.org/10.1007/978-3-319-45348-4_6, https://doi.org/10.1007/978-3-319-45348-4_6
16. de San Pedro, J., Cortadella, J.: Discovering duplicate tasks in transition systems for the simplification of process models. In: Rosa, M.L., Loos, P., Pastor, O. (eds.) Business Process Management - 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings. Lecture Notes in Computer Science, vol. 9850, pp. 108–124. Springer (2016). https://doi.org/10.1007/978-3-319-45348-4_7, https://doi.org/10.1007/978-3-319-45348-4_7
17. Schlachter, U., Wimmel, H.: Relabelling LTS for petri net synthesis via solving separation problems. Trans. Petri Nets Other Model. Concurr. **14**, 222–254 (2019). https://doi.org/10.1007/978-3-662-60651-3_9, https://doi.org/10.1007/978-3-662-60651-3_9
18. Schlachter, U., Wimmel, H.: Optimal label splitting for embedding an LTS into an arbitrary Petri net reachability graph is NP-complete. CoRR **abs/2002.04841** (2020), https://arxiv.org/abs/2002.04841
19. Wang, Y., Nazeem, A., Swaminathan, R.: On the optimal petri net representation for service composition. In: IEEE International Conference on Web Services, ICWS 2011, Washington, DC, USA, July 4-9, 2011. pp. 235–242. IEEE Computer Society (2011). https://doi.org/10.1109/ICWS.2011.40, https://doi.org/10.1109/ICWS.2011.40