Narrowing down the Hardness Barrier of Synthesizing Elementary Net Systems

Ronny Tredup
Universität Rostock, Germany
ronny.tredup@uni-rostock.de

Christian Rosenke
arivis AG, Munich, Germany
christian.rosenke@arivis.com

Abstract

Elementary net system feasibility is the problem to decide for a given automaton A if there is a certain boolean Petri net with a state graph isomorphic to A. This is equivalent to the conjunction of the state separation property (SSP) and the event state separation property (ESSP). Since feasibility, SSP and ESSP are known to be NP-complete in general, there was hope that the restriction of graph parameters for A can lead to tractable and practically relevant subclasses. In this paper, we analyze event manifoldness, the amount of occurrences that an event can have in A, and state degree, the number of allowed successors and predecessors of states in A, as natural input restrictions. Recently, it has been shown that all three decision problems, feasibility, SSP and ESSP, remain NP-complete for linear A where every event occurs at most three times. Here, we show that these problems remain hard even if every event occurs at most twice. Nevertheless, this has to be paid by relaxing the restriction on state degree, allowing every state to have two successor and two predecessor states. As we also show that SSP becomes tractable for linear A where every event occurs at most twice the only open cases left are ESSP and feasibilty for the same input restriction.

2012 ACM Subject Classification Software and its engineering → Petri nets, Theory of computation → Problems, reductions and completeness

Keywords and phrases Elementary net systems, Petri net synthesis, NP-completeness, Parameterized Complexity

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2018.16

Related Version A technical report covering all results of the paper is available at [11], `https://arxiv.org/abs/1711.00220`.

1 Introduction

In this paper we investigate the complexity of synthesizing elementary net systems (ENS), which are the most fundamental type of Petri nets [12]. ENSs are a powerful language for describing processes in digital hardware and provide lots of methods for specification, verification and synthesis of particularly asynchronous or self-timed circuits [5][15]. Moreover, equipped with basic concepts like choice and causality, ENSs are the formal foundation of business process modeling languages, as for instance the Business Process Modeling Notation (BPMN) [9], Event Driven Process Chains (EPC) [6] or activity diagrams in the UML standard [7]. Especially because of their simpleness ENSs are useful for the specifications of workflow management systems like MILANO [1].

29th International Conference on Concurrency Theory (CONCUR 2018).
Editors: Sven Schewe and Lijun Zhang; Article No. 16; pp. 16:1–16:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

ENS synthesis for a given automaton A, called transition system (TS) in this context, means to find a Petri net N with a state graph isomorphic to A. More precisely, N has to be a directed graph on place nodes P and transition nodes T linked by flow arcs F such that, starting from the given initial marking $M_0 \subseteq P$, the net can transform its current marking $M \subseteq P$ into $M' \subseteq P$ by a transition $t \in T$, if $(p, t) \in F$ for all deallocated places $p \in M \setminus M'$ and $(t, q) \in F$ for all occupied places $q \in M' \setminus M$. The reachable markings of N are required to exactly cover A's states S while the transitions T embody A's events E. Every t-transition from M to M' has to correspond to an A-arc $s \xrightarrow{e} s'$ from the state s of M to the state s' of M' and labeled by the event e standing for t.

To assess the complexity of ENS synthesis, this paper analysis the corresponding decision problem, called feasibility. For a given TS A, feasibility asks if there is an ENS N with a state graph isomorphic to A. As not every TS can be synthesized into an ENS, feasibility is a problem worth studying. Usually, it is approached by the *state separation property* (SPP) and the *event state separation property* (ESSP) as, according to [3], A is feasible if and only if it satisfies both properties.

This does not mean that the SSP and the ESSP are not of interest when considered alone. Synthesizing TSs A having only the ESSP leads to Petri nets implementing all event sequences of A by their transitions but with less states [3]. Being able to efficiently decide the SSP, on the other hand, could serve as a quick-fail preprocessing mechanism for synthesis.

Hiraishi [8] shows that both, SSP and ESSP, are NP-complete. Feasibility is NP-complete [2], too. Nevertheless, considerable efforts have been made to find practically relevant tractable cases. For example, feasibility becomes tractable for Flip-Flop nets, a superclass of ENSs [13]. Workflow net models as defined in [1] are a subclass of ENS that allow polynomial time feasibility, too.

Rather than generalizing or restricting the set of nets, this paper restricts the problem input to learn about synthesis complexity. We propose the following two natural and fundamental parameters of TSs that, when controlled, should have a resounding positive impact on synthesis complexity:

State degree of a TS A is the maximum amount g of incoming and, respectively, outgoing edges at the states of A. The decision problems where input is restricted to so called g-grade TSs are referred to as g-grade SSP, g-grade ESSP, and g-grade feasibility. If $g = 1$ and A is not a cycle we use the term *linear*.

Event manifoldness of a TS A is the maximum amount k of edges in A that can be labeled with the same event. Accordingly, we speak about k-fold TSs and the problems k-SSP, k-ESSP, and k-feasibility.

Benchmarks of the digital hardware design community show that practical TSs often have limited state-degree [4]. If restricted event manifoldness is practical relevant has not been evaluated, yet, but it is a straight forward TS parameter.

In [14], we already show that even simultaneous and extreme restrictions of event manifoldness and state degree do not help reducing complexity. In fact, SSP, ESSP, and feasibility remain NP-complete for linear 3-fold input TSs. In this paper, we draw a more precise picture of the problems' hardness. All three of them remain NP-complete for g-grade k-fold TSs if $g \geq 2$ and $k \geq 2$.

On the other hand, 1-SSP, 1-ESSP, and 1-feasibility, that is, when events occur only once, are trivially tractable for every state degree. As this paper also shows that linear 2-SSP can be solved in polynomial time the only remaining open questions concern linear 2-ESSP and linear 2-feasibility. Figure 1 shows an overview of our findings.

This paper is organized as follows: For a start, the following two sections introduce preliminary notions used throughout the paper. In Section 4, we introduce our main result, a polynomial time reduction of cubic monotone one-in-three 3-SAT to 2-grade 2-ESSP. Our

g	problem \ k	1	2	3	4	...
1	SSP	P	P	NPC	NPC	...
	ESSP	P	open	NPC	NPC	...
	Feasibility	P	open	NPC	NPC	...
2	SSP	P	NPC	NPC	NPC	...
	ESSP	P	NPC	NPC	NPC	...
	Feasibility	P	NPC	NPC	NPC	...
⋮	⋮	...	...	...	...	...

$k > 2$ shown in [14] (rows $g = 1$)

shown in this paper (rows $g \geq 2$)

trivial cases (column $k = 1$)

Figure 1 Overview of our results regarding the complexity of the SSP, the ESSP, and feasibility depending on the parameters state degree g and event manifoldness k. Considering the parameters individually, we have already determined the exact borderline between tractable and intractable cases.

reduction makes sure that the produced TS instances always have the SSP. In this way, ESSP and feasibility become the same problem with respect to the generated instances and, hence, we simultaneously show the NP-completeness of g-grade k-ESSP and g-grade k-feasibility for all $g \geq 2$ and $k \geq 2$.

That g-grade k-SSP is also hard to solve for $g, k \geq 2$ is provided in Section 5. Although usually perceived differently, we thereby imply that SSP is not easier than ESSP for TSs with limited state degree and event manifoldness.

Because of space limitations, some technical proofs have been omitted. For a complete presentation of all technical details, we refer to our technical report [11].

2 Preliminaries

This paper deals with (deterministic) *transition systems* (TS) $A = (S, E, \delta, s_0)$ which are determined by finite disjoint sets S of states and E of events, a partial transition function $\delta : S \times E \to S$, and an initial state $s_0 \in S$. Usually, we think of A as an edge-labeled directed graph with node set S where every triple $\delta(s, e) = s'$ is interpreted as an e-labeled edge $s \xrightarrow{e} s'$. For readability, we say that an event e *occurs* at state s if $\delta(s, e) = s'$ for some state s' and abbreviate this with $s \xrightarrow{e}$. Moreover, TSs are required to be *simple*, that is, there are no multi-edges $s \xrightarrow{e} s'$ and $s \xrightarrow{e'} s'$, *loop-free*, which rules out instant state recurrence like $s \xrightarrow{e} s$, *reachable*, where every state can be reached from s_0 by a directed path, and *reduced*, which means free of unused events in E.

Key concept of this paper are *g-grade* TSs A where both, the predecessor set $\{s' \mid \exists e \in E : \delta(s', e) = s\}$ and the successor set $\{s' \mid \exists e \in E : \delta(s, e) = s'\}$, contain at most g elements for every state $s \in S$. We use *linear* for 1-grade TSs that are not a cycle. Moreover, A is called *k-fold* if the set $\{(s, s') \mid \delta(s, e) = s'\}$ of e-connected states contains at most k pairs for every event $e \in E$.

Fundamental to the following notions are *regions* of TSs. A set $R \subseteq S$ is called a region of A if it permits a so-called *signature* $sig : E \to \{-1, 0, 1\}$. This means, all edges $s \xrightarrow{e} s'$ have to satisfy $R(s') = sig(e) + R(s)$, where, by a little abuse of notation, $R(s) = 1$ if $s \in R$ and otherwise $R(s) = 0$ for all $s \in S$. It is easy to see that every region R has a unique signature which is therefor called *the* signature sig_R of R. We say that an event e enters region R, respectively exits or obeys R, if $sig_R(e) = 1$, respectively $sig_R(e) = -1$ or $sig_R(e) = 0$.

Based on the previous definition, two states $s, s' \in S$ are *separable* in A if $R(s) \neq R(s')$ for some region R of A. Moreover, an event $e \in E$ is *inhibitable* at state $s \in S$ if there is a region R of A with either $R(s) = 0$ and $sig_R(e) = -1$ or $R(s) = 1$ and $sig_R(e) = 1$. Using this, a TS A has the *state separation property* (SSP), if all states of A are pairwise separable and it has the *event state separation property* (ESSP) if all events e of A are inhibitable at all states s where $s \xrightarrow{e}$ is not fulfilled. Then, A is *feasible* if and only if it has the SSP and the ESSP.

To study feasibility, ESSP and SSP for restricted TSs we define the g-grade (k-fold) problem for all naturals g (k) where the input is restricted to g-grade (k-fold) TSs. Notice that the set of g-grade k-fold TSs is a subclass of g'-grade k'-fold TSs in case $k \leq k'$ and $g \leq g'$. Hence, hardness results propagate up the problem hierarchy and efficient algorithms are legitimate for all lower classes.

As we approach feasibility by SSP and ESSP, which are defined on top of TSs, we omit a formal definition of ENSs and rather refer to, e.g., [3].

3 Unions, Transition System Containers

For our NP-completeness proofs this section introduces *unions*, a gadget concept to modularize our arguments. In a union, individual TSs are grouped together and treated as if being part of one TS. Moreover, we develop a *joining* operation to merge union parts and preserve their (E)SSP and feasibility.

Formally, if $A_0 = (S_0, E_0, \delta_0, s_0^0), \dots, A_m = (S_m, E_m, \delta_m, s_0^m)$ are TSs with pairwise disjoint states then we say that $U(A_0, \dots, A_m)$ is their *union*. By $S(U)$ we denote the entirety of all states in $A_0, \dots, A_m$ and $E(U)$ is the aggregation of all events. The joint transition function $\Delta^U = \bigcup_{i=0}^{m} \delta_i$ of U is defined as

$$\Delta^U(s, e) = \begin{cases} \delta_i(s, e), & \text{if } s \in S_i \text{ and } e \in E_i, \\ \text{undefined}, & \text{else} \end{cases}$$

for all $s \in S(U)$ and all $e \in E(U)$. If every event in $E(U)$ occurs at most k times in U, not necessarily as part of the same TS, we say that U is k-fold.

For simplicity, we build unions recursively: Firstly, every TS A is identified with the union containing only A, that is, $A = U(A)$. Next, if $U_1 = U(A_0^1, \dots, A_{m_1}^1), \dots, U_n = (A_0^n, \dots, A_{m_n}^n)$ are unions then $U(U_1, \dots, U_n)$ is the union $U(A_0^1, \dots, A_{m_1}^1, \dots, A_0^n, \dots, A_{m_n}^n)$ that flattens out the parent unions by cumulating all their TSs.

As we want to combine independent TSs $A_0, \dots, A_m$ in a union $U = U(A_1, \dots, A_m)$ and treat U as one TS, we need to lift regions, the SSP and the ESSP to U: We say that $R \subseteq S(U)$ is a region of U if it permits a signature $sig_R : E \to \{-1, 0, 1\}$. Hence, for all $i \in \{0, \dots, m\}$ the subset $R_i = R \cap S_i$, coming from the states S_i of A_i, has to be a region of A_i with signature $sig_{R_i}(e) = sig_R(e)$ for all $e \in E_i$. Then, U has the SSP if for all states $s, s' \in S(U)$ coming from the same TS A_i there is a region R of U with $R(s) \neq R(s')$. Moreover, U has the ESSP if for all events $e \in E(U)$ and all states $s \in S(U)$ with $\neg(s \xrightarrow{e})$ there is a region R of U such that $R(s) = 0$ and $sig_R(e) = -1$ or $R(s) = 1$ and $sig_R(e) = 1$. Naturally, U is called feasible if it has both, the SSP and the ESSP.

To merge a union $U = U(A_0, \dots, A_m)$ back into a single TS, we define the joining $A(U)$ as follows: If $s_0^0, \dots, s_0^m$ are the initial states of U's TSs then $A(U) = (S(U) \cup Q, E(U) \cup Y \cup Z, \delta, q_0)$ is a TS with additional connector states $Q = \{q_0, \dots, q_m\}$ and fresh events

$Y = \{y_0, \ldots, y_m\}, Z = \{z_0, \ldots, z_{m-1}\}$ joining the loose elements of U by

$$\delta(s,e) = \begin{cases} \Delta^U(s,e), & \text{if } s \in S(U) \text{ and } e \in E(U), \\ s_0^i, & \text{if } s = q_i \text{ and } e = y_i \\ q_{i+1}, & \text{if } s = q_i \text{ and } e = z_i. \end{cases}$$

Notice that $A(U)$ preserves k-foldness and, if every initial state s_0^i has at most one predecessor in A_i, it preserves g-gradeness for $g \geq 2$. The following lemma certifies the validity of joining (most) unions:

▶ **Lemma 1.** *Let $U = U(A_0, \ldots, A_m)$ be a union of TSs $A_0, \ldots, A_m$ which fulfill for every event e that there is at least one state s with $\neg(s \xrightarrow{e})$. Then U has the (E)SSP, respectively is feasible, if and only if the joining $A(U)$ has the (E)SSP, respectively is feasible.*

Proof. *If*: Projecting a region separating s and s', respectively inhibiting e at s, in $A(U)$ to the component TSs yields a region separating s and s', respectively inhibiting e at s in U. Hence, the (E)SSP of $A(U)$ trivially implies the (E)SSP of U.

Only if: A region R of U separating s and s', respectively inhibiting e at s, can be completed to become an equivalent region of $A(U)$ by setting

$$R(q_i) = 0, sig_R(z_j) = 0, \text{ and } sig_R(y_i) = R(s_0^i)$$

for all $i, j \in \{0, \ldots, m\}, j < m$.

Notice that R, defined as above, also inhibits e at all connector states. Hence, to inhibit an event $e \in E(U)$ at all connector states of $A(U)$, we choose any U-region R_e that inhibits e at any state $s \in S(U)$. As we require that every $e \in E(U)$ has $s \in S(U)$ with $\neg(s \xrightarrow{e})$, the ESSP of U implies the existence of R_e. Thus, in $A(U)$ every event of U can be inhibited at all required states.

For the (E)SSP of $A(U)$ it is subsequently sufficient to analyze (event) state separation concerning the connector states (events). By the uniqueness of the connector events $Y \cup Z$, it is easy to see that each connector state q_i on its own defines a region $R_i = \{q_i\}$ of $A(U)$ that inhibits y_i, z_i and separates q_i in $A(U)$. ◀

4 The Hardness of the ESSP and Feasibility for 2-grade 2-fold Transition Systems

This section presents our main result and answers the question if restricting the event manifoldness to $k = 2$ helps reducing the complexity of synthesizing ENS:

▶ **Theorem 2.** *Deciding the ESSP or feasibility is NP-complete on g-grade k-fold transition systems for all $g \geq 2$ and all $k \geq 2$.*

The rest of this section is devoted to the proof of this theorem.

That the g-grade k-fold versions of the ESSP and feasibility are contained in NP is clearly not a proof obligation here, as this already follows from the NP-completeness of the unrestricted problems [2][8].

For the proof of completeness in NP, we basically reduce cubic monotone one-in-three 3-SAT, which is NP-complete [10], to 2-grade 2-ESSP in polynomial time. Therefore, we start the reduction from a cubic monotone boolean CNF expression $\varphi = \{C_0, \ldots, C_{m-1}\}$, a set of negation-free 3-clauses where every variable occurs in exactly three clauses. The result is a union U^φ of gadget TSs that has the ESSP if and only if φ has a one-in-three model

M, that is, a subset of φ's variables $V(\varphi)$ with $|M \cap C_i| = 1$ for all $i \in \{0, \dots, m-1\}$. This means that M exactly covers all clauses of φ. For example, the expression

$$\varphi_0 = \big\{\{x_0, x_1, x_2\}, \{x_0, x_1, x_4\}, \{x_0, x_2, x_3\}, \{x_1, x_4, x_5\}, \{x_2, x_3, x_5\}, \{x_3, x_4, x_5\}\big\}$$

has six clauses over the variables $V(\varphi_0) = \{x_0, \dots, x_5\}$ which are satisfied by the one-in-three model $M_0 = \{x_0, x_5\}$. Unfortunately, we cannot use the expression φ_0 as a running example since the union U^{φ_0} resulting from our construction would already have 1500 states. Thus, its presentation as a whole would go far beyond the scope of this paper.

The construction of U^φ makes sure that even in the joining $A^\varphi = A(U^\varphi)$ every event is used at most twice and has at most two predecessors and two successors. By design, the ESSP of U^φ implies the SSP, too. This makes ESSP and feasibility the same problem, even for A^φ as stated in Lemma 1. Consequently, our proof provides the NP-hardness for both problems on 2-grade 2-fold TSs.

In the following, we start with the details of constructing U^φ. The union consists of several functional components. Firstly, it installs a TS H, called the *head*, which initializes the connection between the satisfiability problem and the ESSP. It introduces the key event k that is supposed to be inhibitable at a certain key state if and only if φ has a one-in-three model. In order to achieve this behavior, U^φ adds a so-called translator T_i for every clause C_i. For a key region, one that inhibits k at the key state, the purpose of T_i is to implement one-in-three behavior for C_i. More precisely, T_i applies events to represent the three variables of C_i and assures that exactly one of them has a positive signature while the other two have to obey. This means for a key region that every gadget $T_0, \dots, T_{m-1}$ has exactly one entering variable event which exactly translates into a one-in-three model for φ. Reversely, every one-in-three model tells us how we can define a key region by choosing exactly one entering variable in every translator.

The main problem so far is to get along with the restriction of using every event only twice. We solve this problem by adding more TSs to U^φ that, for a key region, generate helper and replacement events with predefined signatures, leaving, entering, or obeying. To create a better picture of our method, the following introduces the details of all applied gadget TSs. See Figure 2 to also visualize the technical details of the description.

Head. H is a TS having two responsibilities. Firstly, it introduces the key event k and the key state $h_{0,8}$. We add the name affix *key* to regions R_{key} of U^φ that inhibit k at $h_{0,8}$, or more precisely, where $sig_{R_{key}}(k) = -1$ and $h_{0,8} \notin R_{key}$. Secondly, H cooperates with the subsequent duplicator gadgets to prepare sufficient amounts of events with negative signature. The reason is that our reduction has to get along with applying k just twice. To duplicate the negative signature of k to other events, the so-called key copies, H works with a production line of $14m$ submodules H_j, each cooperating with a duplicator D_j to initialize one key copy. More precisely, for a key region, H_j prepares three events for D_j, two so-called *vice* events v_{2j}, v_{2j+1}, which have a positive signature (that is, vice with respect to the signature of k) and one obeying *wire* event w_{2j}. In return, the duplicator provides two key copies k_{3j}, k_{3j+1} and one obeying *accordance* event a_j. In H_{j+1} these three result events are used for the synchronization of the next vice and wire events. The main result of D_j, however, is k_{3j+2}, one of the $14m$ key copies that are free to be applied in the other reduction gadgets.

See Figure 2 (a) for a definition of H together with an illustration of H's part of a key region, R^H. Observe that there are *reachability* events $r_0, \dots, r_{14m-2}$ which have the only purpose to make every state of H reachable from initial $h_{0,0}$. Moreover, for R^H, every module H_j receives key copies k_{3j-3}, k_{3j-2} and accordance event a_{j-1} from D_{j-1}. Thanks to a_{j-1}, the state $h_{j,8}$ behaves according to the key event $h_{0,8}$ and is excluded from R^H. Because of exiting k_{3j-3} the state $h_{j,1}$ is out of R^H, too. This imprints a zero signature on the *zero* events z_{2j}, z_{2j+1}. Exiting k_{3j-2} puts $h_{j,4}$ into R^H and excludes $h_{j,5}$ which, together with z_{2j}, z_{2j+1}, makes v_{2j}, v_{2j+1} entering and w_{2j}, w_{2j+1} obeying.

Duplicators. D_j are TSs that, for a key region, generate three key copies $k_{3j}, k_{3j+1}, k_{3j+2}$ and one obeying accordance event a_j using the vice events v_{2j}, v_{2j+1} with positive signature and the obeying wire event w_{2j}. Figure 2 (b) defines D_j and demonstrates the duplicator fraction R^{D_j} of a key region. The entering vice events force $d_{j,2}, d_{j,4}$ into R^{D_j} and exclude $d_{j,1}, d_{j,3}$. The obeying wire event signals the condition of $d_{j,4}$ to $d_{j,0}$ putting it into R^{D_j}. By design, $k_{3j}, k_{3j+1}, k_{3j+2}$ are exiting and a_j becomes obeying. As H consumes only k_{3j}, k_{3j+1} and a_j, we keep the remaining duplicate k_{3j+2} of k. Creating $14m$ duplicators in total, we get $14m$ free key copies.

Barters. B_q are TSs that, for a key region, barter key copies k_{q_1}, k_{q_2} for one obeying so-called *consistency* event c_q. The indices $q_1 = 18m + 6q + 1$ and $q_2 = q_1 + 3$ select two of the last $4 \cdot 2m$ items from the list of free key copies. The use of consistency events is to synchronize three events $x_i^\alpha, x_i^\beta, x_i^\gamma$ for every variable x_i. The reason is that we cannot represent the three occurrences of x_i in the expression φ by an event that can only be used twice. Consequently, we require three generated events of consistent signature to represent x_i. Figure 2 (c) introduces B_q and shows a respective key region part R^{B_q}. As both key copies are leaving, $b_{q,0}, b_{q,2}$ are in R^{B_i} and c_q is obeying. Altogether, we add $4m$ barters that consume $8m$ key copies to generate $4m$ consistency events.

Variable manifolders. X_i are TSs synchronizing three events for every variable $x_i \in V(\varphi)$. If $C_\alpha, C_\beta, C_\gamma$ are the three clauses containing x_i then X_i provides the events $x_i^\alpha, x_i^\beta, x_i^\gamma$. For a key region, they are supposed to have the same signature in order to treat them as manifestations of the same event representing x_i. The definition of X_i as well as an illustration of a possible key region fragment R^{X_i} are given in Figure 2 (d). To create the event equivalence, X_i applies four consistency events, c_{4i}, c_{4i+1}, c_{4i+2}, c_{4i+3}. Their obedience condemns the two state groups $x_{i,0}, x_{i,1}, x_{i,2}$ and $x_{i,3}, x_{i,4}, x_{i,5}$ to a consistent behavior with respect to R^{X_i}, that is, either all states of a group are part of the region or none of them. This brings $x_i^\alpha, x_i^\beta, x_i^\gamma$ into synchronicity.

Translators. T_i are unions $T_i = U(T_{i,0}, T_{i,1}, T_{i,2})$ of three TSs, each. For a key region, T_i implements $C_i = \{x_a, x_b, x_c\}$ by allowing a positive signature for exactly one variable representation, either x_a, x_b or x_c. Figure 2 (e-g) define the three TSs and introduces a possible key region fragment that assigns a positive signature to event x_b^i representing x_b. Apparently, $T_{i,1}$ parenthesizes an event for x_b and the proxy event p_i with two key copies while $T_{i,2}$ does the same for the locum event $\tilde{x}_b^i$ and p_i. For a key region, all key copies exit and the proxy event behaves equally in both TSs. This aligns the signature of x_b^i and $\tilde{x}_b^i$ and makes it non-negative. Furthermore, $T_{i,0}$ is a key copy delimited sequence of events that, for a key region, prevents a negative signature for x_a^i, x_c^i. As the limiting key copies are exiting and as none of $x_a^i, \tilde{x}_b^i, x_c^i$ can be exiting, exactly one of the events x_a^i, x_b^i, x_c^i has to enter.

Altogether, this construction results in the union $U^\varphi = U(U_1^\varphi, U_2^\varphi)$ with

$$U_1^\varphi = U(H, D_0, \dots, D_{14m-1}),$$
$$U_2^\varphi = U(B_0, \dots, B_{4m-1}, X_0, \dots, X_{m-1}, T_0, \dots, T_{m-1}).$$

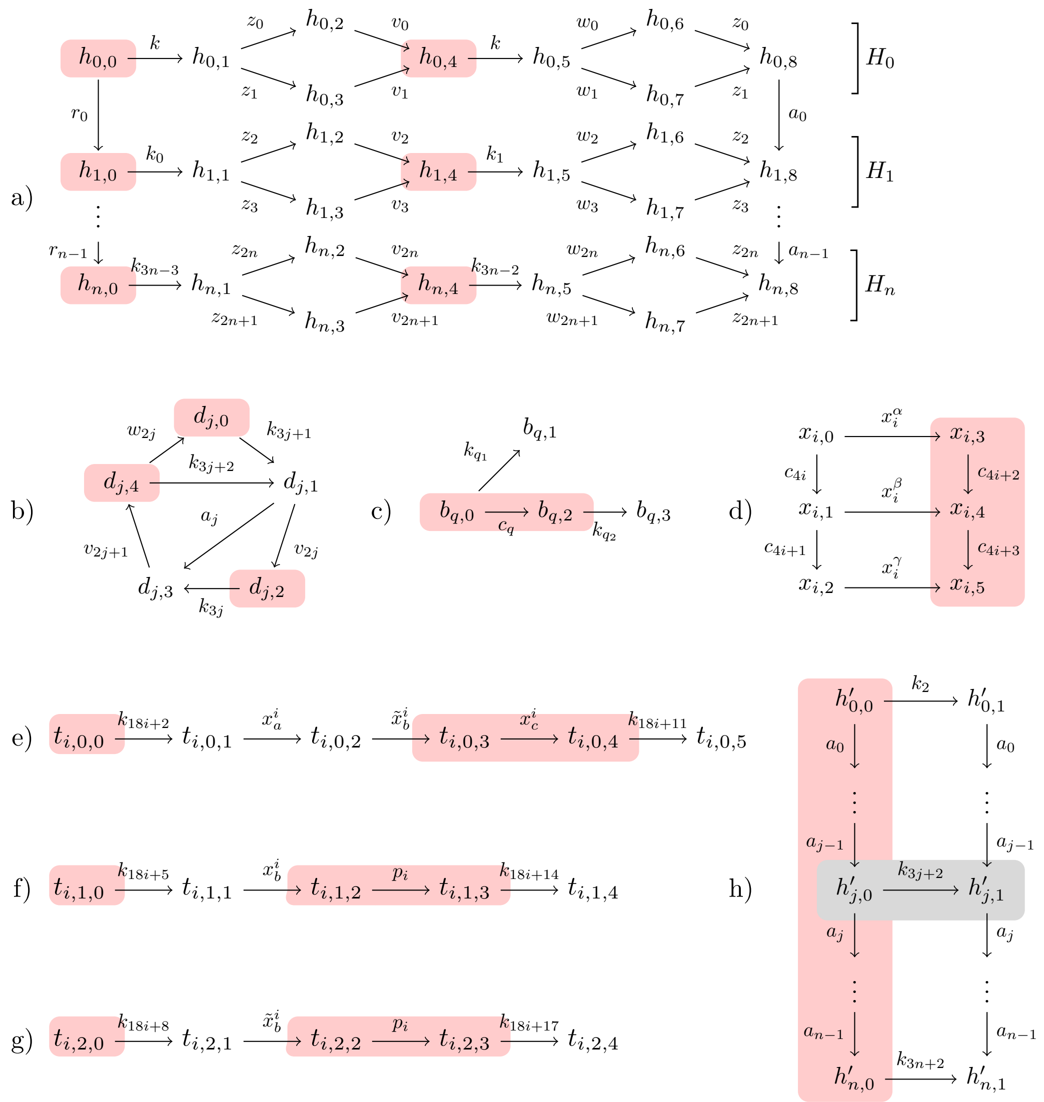

Figure 2 a-g) The gadgets of U^φ with their respective fractions of a key region. The red marked states are included in the key region and the unmarked are excluded. a) The head H with submodules $H_0, \dots, H_n$ where $n = 14m - 1$. b) D_j, one of the $14m$ duplicators that provide the $14m$ key copies. c) B_q, one out of $4m$ barters trading $8m$ key copies for $4m$ consistency events. Here, $q_1 = 6q+18m+1$ and $q_2 = q_1 + 3$. d) The variable manifolder X_i using four consistency events to synchronize three variable events for x_i. Together, the m variable manifolders consume all $4m$ available consistency events. e-g) The translator T_i consisting of $T_{i,0}$ (e), $T_{i,1}$ (f), and $T_{i,2}$ (g). Using six key copies, T_i implements the clause C_i. All m translators together consume the remaining $6m$ key copies. h) The head H' of the union U^φ_{SSP} from Section 5. The red marked states describe the key region $R^{H'}$ and the gray marked states provide a region of H' that separates the states $h'_{j,0}, h'_{j,1}$ from the remaining states of H'.

The reason for separating U^φ into two sub unions U^φ_1 and U^φ_2 is that we want to reuse U^φ_2 in Section 5. Here, our last construction step is to join the TSs of U^φ in order to obtain the combined TS $A^\varphi = A(U^\varphi)$. Check Figure 2 to see that the initial states of the gadgets, that is, $h_{0,0}$, $d_{j,0}$, $b_{q,0}$, $x_{i,0}$, $t_{i,0,0}$, $t_{i,1,0}$, and $t_{i,2,0}$, have at most one predecessor state each.

Hence, the definition of joining guarantees that A^φ does not exceed the state degree of two. Moreover, as every event of U^φ occurs at most twice and A^φ just introduces additional unique events, A^φ is a 2-grade 2-fold TS.

Before we can show that A^φ has the ESSP if and only if φ has a one-in-three model, we need the next two lemmas to formalize the properties of key regions in U^φ:

▶ **Lemma 3.** *If R is a region of U_1^φ inhibiting k at $h_{0,8}$, that is, where, without loss of generality, $sig_R(k) = -1$ and $h_{0,8} \notin R$ then*

1. *for all $j \in \{0, \dots, 14m-1\}$ the region contains $h_{j,0}, h_{j,4}$ and excludes $h_{j,1}$, $h_{j,2}$, $h_{j,3}$, $h_{j,5}$, $h_{j,6}$, $h_{j,7}$, $h_{j,8}$,*
2. *for all $j \in \{0, \dots, 14m-1\}$ the region contains $d_{j,0}, d_{j,2}, d_{j,4}$ and excludes $d_{j,1}, d_{j,3}$,*
3. *all key copies exit, that is, for all $j \in \{0, \dots, 14m-1)$ the events k_{3j}, k_{3j+1}, k_{3j+2} have negative signature.*

Proof. Consider the individual gadget regions R^H and R^{D_j} demonstrated in Figure 2 (a) and (b). We show that, combined, they define the only region R of U_1^φ that inhibits k at $h_{0,8}$ with $sig_R(k) = -1$ and $h_{0,8} \notin R$. For this purpose, we use induction over j and simultaneously show that R fulfills (1-3):

For a start, let $j = 0$. We show that $sig_R(k) = -1$ and $R(h_{0,8}) = 0$ force R to coincide with R^H with respect to the part H_0 of H and with R^{D_0}. The requirement $sig_R(k) = -1$ immediately brings $R(h_{0,0}) = R(h_{0,4}) = 1$ and $R(h_{0,1}) = R(h_{0,5}) = 0$. Then, $R(h_{0,1}) = 0$ implies that $sig_R(z_0), sig_R(z_1) \in \{0, 1\}$. The second premise $R(h_{0,8}) = 0$ yields $sig_R(z_0), sig_R(z_1) \in \{-1, 0\}$, which consequently results in $sig_R(z_0) = sig_R(z_1) = 0$. This implies $R(h_{0,2}) = R(h_{0,3}) = 0$ and $R(h_{0,6}) = R(h_{0,7}) = 0$ making v_0, v_1 entering and w_0, w_1 obeying. The entering signature of v_0, v_1 makes R include $d_{0,2}, d_{0,4}$ and exclude $d_{0,1}, d_{0,3}$. As $R(d_{0,4}) = 1$ and w_0 obeys, we get $R(d_{0,0}) = 1$. By $R(d_{j,0}) = R(d_{j,4}) = 1$ and $R(d_{j,1}) = 0$ we obtain that k_1, k_2 exit. By $R(d_{j,1}) = R(d_{j,3}) = 0$ and $R(d_{j,2}) = 1$ we have that k_0 exits and a_0 obeys.

Now assume that R coincides with R^{D_i} and R^H on the parts H_i for all i less than j. Moreover, suppose that (1-3) hold for all indices less j. As $R(h_{j-1,8}) = 0$ and $sig_R(a_{j-1}) = 0$, we have $R(h_{j,8}) = 0$. Furthermore, we get the exiting $k_{3(j-1)}, k_{3(j-1)+1}$. Hence, we basically have the same situation as in H_0 and D_0. Consequently, a similar argumentation as for the induction start yields that R contains exactly the states $h_{j,0}, h_{j,4}$ of H_j and $d_{j,0}, d_{j,2}, d_{j,4}$ of D_j. This makes the vice events v_{2j}, v_{2j+1} enter and the wire events w_{2j}, w_{2j+1} obey. Moreover, D_j lets the key copies k_{3j}, k_{3j+1}, k_{3j+2} have a negative signature and a_j obey. ◀

▶ **Lemma 4.** *If R is a region of U_2^φ having an exiting signature for all key copies, that is, $sig_R(k_{3j+2}) = -1$ for all $j \in \{0, \dots, 14m-1\}$, then*

1. *for all $q \in \{0, \dots, 4m-1\}$ the region R contains $b_{q,0}, b_{q,2}$ and excludes $b_{q,1}, b_{q,3}$ and has $sig_R(c_q) = 0$,*
2. *for all $i \in \{0, \dots, m-1\}$ variable x_i, which occurs in clauses C_α, C_β, C_γ, is represented by events x_i^α, x_i^β, x_i^γ having the same signature*
$$sig_R(x_i^\alpha) = sig_R(x_i^\beta) = sig_R(x_i^\gamma), \text{ and}$$
3. *for all $i \in \{0, \dots, m-1\}$ clause $C_i = \{x_a, x_b, x_c\}$ is realized in translator T_i making exactly one of the events x_a^i, x_b^i, x_c^i enter while the other two obey.*

Proof. Statement (1) means that R coincides with R^{B_q} for every barter B_q. As the key copies k_{q_1} and k_{q_2} are assumed to exit for $q_1 = 18m + 6q + 1$ and $q_2 = q_1 + 3$, this statement trivially follows. First and foremost, this implies that all consistency events $c_q, q \in$

$\{0, \dots, 4m-1\}$ have an obeying signature. In statement (2), this obedience immediately fixes the states $x_{i,0}, x_{i,1}, x_{i,2}$ of variable manifolder X_i to behave consistently, that is, $R(x_{i,0}) = R(x_{i,1}) = R(x_{i,2})$. Analogously, we derive $R(x_{i,3}) = R(x_{i,4}) = R(x_{i,5})$. This implicates $R(x_{i,3}) - R(x_{i,0}) = R(x_{i,4}) - R(x_{i,1}) = R(x_{i,5}) - R(x_{i,2})$. Consequently, the variable events $x_i^\alpha, x_i^\beta, x_i^\gamma$ have the same signature for all $i \in \{0, \dots, m-1\}$.

Finally, statement (3) for the translator T_i can be seen as follows: For any region R' of a linear TS it is a simple observation that $\sum_{j=j_1}^{j_2-1} sig_R(e_j) = R(s_{j_2}) - R(s_{j_1})$ for any subsequence $s_{j_1} \xrightarrow{e_{j_1}} \dots \xrightarrow{e_{j_2-1}} s_{j_2}$ within the TS. As the TSs of T_i are linear, we get from $R(t_{i,1,3}) - R(t_{i,1,1}) = 1$ and $R(t_{i,2,3}) - R(t_{i,2,1}) = 1$ that $sig_R(x_b^i) + sig_R(p_i) = sig_R(\tilde{x}_b^i) + sig_R(p_i) = 1$. That means, $sig_R(x_b^i) = sig_R(\tilde{x}_b^i) = 1 - sig_R(p_i)$ which implies that x_b^i has a non-negative signature. As $R(t_{i,0,1}) = 0$ and $R(t_{i,0,4}) = 1$, we have non-negative signature of x_a^i, x_c^i, too. That $sig_R(x_a^i) + sig_R(\tilde{x}_b^i) + sig_R(x_c^i) = 1$ implies that exactly one of these events has a positive signature. ◀

Lemma 3 and Lemma 4 state that the structure of a key region defines a model of φ. That is why we can say that the existence of key region for U^φ implies the one-in-three satisfiability of φ:

▶ **Lemma 5.** *If there is a key region of U^φ then φ has a one-in-three model.*

Proof. Let R_{key} be a key region of U^φ, that is, $sig_{R_{key}}(k) = -1$ and $R_{key}(h_{0,8}) = 0$. First of all, Lemma 3 implies that $sig_{R_{key}}(k_{3j+2}) = -1$ for all $j \in \{0, \dots, 14m-1\}$. As a consequence, we obtain from Lemma 4 for all variables x_i and their three occurrences in clauses $C_\alpha, C_\beta, C_\gamma$ that

$$sig_{R_{key}}(x_i^\alpha) = sig_{R_{key}}(x_i^\beta) = sig_{R_{key}}(x_i^\gamma).$$

Moreover, Lemma 4 means for every clause $C_i = \{x_a, x_b, x_c\}$ that exactly one of the events x_a^i, x_b^i, x_c^i has a positive signature while the other two obey. Hence, if we add a variable x_i to a set M if and only if the corresponding events have positive signature, then we clearly obtain for all clauses C_i that $|M \cap C_i| = 1$. This makes M a one-in-three model. ◀

The other way around, the required equivalence obliges us to derive a key region from any one-in-three model. We argue that working our way backwards through the construction ends up in a region that inhibits k at the key state.

▶ **Lemma 6.** *If φ has a one-in-three model then there is a key region of U^φ.*

Proof. Let $M \subseteq V(\varphi)$ be a one-in-three model of φ. We progressively build a region R by following the requirements of every individual gadget.

Firstly, for every variable x_i occurring in $C_\alpha, C_\beta, C_\gamma$ we take care that $sig_R(x_i^\alpha) = sig_R(x_i^\beta) = sig_R(x_i^\gamma) = M(x_i)$ where $M(x_i) = 1$ if $x_i \in M$ and $M(x_i) = 0$, otherwise. To this end, we let $x_{i,3}, x_{i,4}, x_{i,5} \in R^{X_i}$. Moreover, we set $x_{i,0}, x_{i,1}, x_{i,2} \in R^{X_i}$ if and only if x_i is not in M. This makes the consistency events $c_{4i}, c_{4i+1}, c_{4i+2}, c_{4i+3}$ obey. As different variable manifolders do not share events, the regions $R^{X_0}, \dots, R^{X_{m-1}}$ are pairwise compatible.

For every clause $C_i = \{x_a, x_b, x_c\}$ the model M selects exactly one variable. By the M-conform construction of $R^{X_a}, R^{X_b}, R^{X_c}$ we get that exactly one of the events x_a^i, x_b^i, x_c^i enters and the others obey. Making the key copies of T_i exit, generates a sub region R^{T_i}. That T_i and T_j share events only for $i = j$ makes $R^{T_0}, \dots, R^{T_{m-1}}$ pairwise compatible. As the variable events are selected in compliance with the variable manifolders and as translators and manifolders do not share further events, their sub regions are also compatible.

By the obeying consistency events, we can define a sub region R^{B_q} for every $q \in \{0, \dots, 4m-1\}$. This also makes the used key copies exiting. As different barters have no event in common, share only consistency events with variable manifolders and no events at all with translators, the regions are all compatible.

Head and duplicators just share key copies with translators and barters. As all key copies are exiting and as the provided sub regions meet the conditions of Lemma 3, we can use this lemma as a construction manual for the sub regions $R^H, R^{D_0}, \dots, R^{D_{14m-1}}$. Altogether, we get that the set R formed by

$$R^H \cup R^{D_0} \cup \dots \cup R^{D_{14m-1}} \cup R^{B_0} \cup \dots \cup R^{B_{4m-1}} \cup R^{X_0} \cup \dots \cup R^{X_{m-1}} \cup R^{T_0} \cup \dots \cup R^{T_{m-1}}$$

is a region of U^φ inhibiting k at $h_{0,8}$. ◀

At this point, the previous lemmas have established that φ has a one-in-three model M if and only if there is a key region for U^φ. Although this is basically the foundation of the proof for Theorem 2, it just delivers the *only-if* direction for the NP-completeness of 2-grade 2-fold ESSP by now: If A^φ has the ESSP then Lemma 1 lifts the ESSP to U^φ. By definition, there also has to be a region that inhibits k at $h_{0,8}$, a key region. Then Lemma 5 implies the existence of the one-in-three model M for φ.

Reversely, having M, Lemma 6 only inhibits k at the key state. For the remaining events e and states s of U^φ with $\neg(s \xrightarrow{e})$, we still have to show that e is inhibitable at s:

▶ **Lemma 7.** *If φ has a one-in-three model then $e \in E(U^\varphi)$ is inhibitable at $s \in S(U^\varphi)$ for every event e and state s of U^φ that fulfill $\neg(s \xrightarrow{e})$.*

Lemma 6 already shows the essential part of Lemma 7. But the proof for the remaining non-key event state combinations is very technical and does not lead to further insights. Therefore and for space limitations, we only go into one example here and refer to our technical report [11] for a full analysis:

▶ **Lemma 8.** *If φ has a one-in-three model then the key event k is inhibitable at all states $s \in S(U^\varphi)$ that fulfill $\neg(s \xrightarrow{k})$.*

Proof. Every relevant state s is subsequently provided with a region that inhibits k at s. For brevity however, we omit to repeat the key-region here, which already inhibits k at many states. To define the other regions, we just specify the signature of non-obeying events, as the majority of events are obedient. To this end, we present every required region as one entry in the following listing:

states	*exit*	*enter*	*affected TSs*
remaining states of H except $h_{1,0}$	a_0, k, k_0, r_1	v_1, w_1, z_0	H, D_0
$h_{1,0}$	a_1, k, k_1, k_2, k_3, k_8, r_0, r_2, w_4, w_5	a_2, k_4, k_5, k_6, r_1, v_1, w_1, z_0, z_2, z_3	H, D_0, D_1, D_2, $T_{0,0}$, $T_{0,1}$, $T_{0,2}$
$S(U^\varphi) \setminus S(H)$	k	z_0, z_1	H

The first column *states* lists the states s where k is inhibited by the respective region. The *exit* and *enter* columns provide the exiting, respectively entering, events of that region. To easily find the TSs that contain at least one of these non-obeying events, one can use the *affected TSs* column in the listing.

Notice that the first two lines of the listing show that k is inhibitable at all states of H. The third line presents a region of U^φ where k exits and all non-obeying events occur in H. Therefore, this region inhibits k at all remaining states of U^φ. ◀

The rest of the proof for Lemma 7 works just the same as demonstrated by Lemma 8. This leads to the *if* direction for the NP-completeness of 2-grade 2-fold ESSP: If φ has a one-in-three model M then Lemma 7 tells us that U^φ has the ESSP. Using Lemma 1, we can bring the ESSP down to A^φ, too. Altogether, we may now state that φ has a one-in-three model if and only if A^φ has the ESSP. As our construction is easily done in polynomial time, we have shown the NP-completeness of 2-grade 2-fold ESSP and by that, half of Theorem 2.

To complete the theorem's proof, it remains to establish that φ has a one-in-three model M if and only if A^φ is feasible. However, this is fairly easy reusing the work we have already done. In fact, the first direction is already there: If A^φ is feasible then it also has the ESSP, which we know implies the existence of M. Reversely, if M exists, then it is sufficient to show that, beside the already established ESSP, A^φ has the SSP, too:

▶ **Lemma 9.** *If φ has a one-in-three model then U^φ has the SSP.*

Proof. If s, s' are two states that do not belong to the same TS of U^φ then they are separable by definition. Hence, let s, s' be two distinct states of one TS $A \in U^\varphi$. If there is an event e that occurs at s, that is, $s \xrightarrow{e}$, but not s', that is, $\neg(s' \xrightarrow{e}$) then s, s' are separable. The reason for this comes from Lemma 7, which states that e is inhibitable at s' by a region R of U^φ. This means, that, without loss of generality, $R(s') = 0$ and $sig_R(e) = -1$, which implies $R(s) = 1$.

Using this condition, we get the separability for all state pairs s, s' that are in one of our TSs except for H as follows: As Figure 2 shows, every event of $A \neq H$ occurs only once within A and (ii) there is only one state of A without a successor. Hence, without loss of generality, there is an event e that occurs at s but not at s'.

Figure 2 also demonstrates that all the events $\{v_{2j}, v_{2j+1}, w_{2j}, w_{2j+1} \mid 0 \leq j < 14m\}$ and $\{a_j, k_{3j}, k_{3j+1}, r_j \mid 0 \leq j < 14m-1\}$ occur only once in H and that $h_{14m-1,8}$ is the only state of H without a successor. Consequently, if $s, s' \in S(H)$ and s is neither in $\{h_{j,1}, h_{j,6}, h_{j,7} \mid 0 \leq j < 14m\}$ nor in $\{h_{0,4}, h_{14m-1,8}\}$ then the above condition makes s and s' separable, too. Moreover, notice that k is the only event occurring at $h_{0,4}$ and that no event occurs at $h_{14m-1,8}$ at all. Hence, our argument works to separate these two states from all states in $S(H)$, too.

As seen in Figure 2, z_{2j} and z_{2j+1} occur only within the part H_j of H. Applying the above condition again, we get for all $0 \leq j < 14m$ that $s \in \{h_{j,1}, h_{j,6}, h_{j,7}$ is separable from $s' \in S(H) \setminus \{h_{j,0}, \dots, h_{j,8}\}$.

It remains to show that the states $\{h_{j,1}, h_{j,6}, h_{j,7}\}$ are pairwise separable for all $j \in \{0, \dots, 14m-1\}$. Notice that z_{2j} occurs at $h_{j,1}$ and $h_{j,6}$ but not at $h_{j,7}$. Hence, using the region inhibiting z_{2j} at $h_{j,7}$ coming from the above argumentation, separates both, $h_{j,1}$ and $h_{j,6}$ from $h_{j,7}$. Similarly, z_{2j+1} occurs at $h_{j,7}$ but not at $h_{j,6}$ which leads to their separability, too. ◀

As a last step, we can use Lemma 1 again, to bring the *SSP* of U^φ down to A^φ, too. This finally proves Theorem 2.

5 The Hardness of SSP for 2-grade 2-fold Transition Systems

This section completes our complexity analysis for the synthesis of TSs having event manifoldness less than three:

▶ **Theorem 10.** *Deciding the SSP is NP-complete on g-grade k-fold transition systems for all* $g \geq 2$ *and all* $k \geq 2$.

Proof. We reuse most of the reduction from Section 4 and create yet another union $U^{\varphi}_{\text{SSP}} = U(H', U^{\varphi}_2)$ by simply replacing U^{φ}_1 with the new head TS H' shown in Figure 2 (h). We prove that φ has a one-in-three model if and only if the two key states $h'_{0,0}, h'_{1,0}$ are separable by a key region R'_{key} if and only if U^{φ}_{SSP} has the SSP.

If U^{φ}_{SSP} has the SSP then there is a key region R'_{key} where, without loss of generality, $R'_{key}(h'_{0,0}) = 1$ and $R'_{key}(h'_{0,1}) = 0$. This implies $sig_{R'_{key}}(k_2) = -1$. Using this as a start, induction over j infers from $R'_{key}(h'_{j,0}) = 1$ and $R'_{key}(h'_{j,1}) = 0$ that a_j is obeying and that $R'_{key}(h'_{j+1,0}) = 1, R'_{key}(h'_{j+1,1}) = 0$ and that $sig_{R'_{key}}(k_{3j+2}) = -1$. Hence, on H' the key region is just $R^{H'}$ from Figure 2 (h). By Lemma 4, the exiting key copies $k_{3j+2}, j \in \{0, \dots, 14m-1\}$ imply that every variable $x_i \in V(\varphi)$ is represented by three synchronized variable events and for every clause $C_j = x_a, x_b, x_c$ exactly one of x^j_a, x^j_b, x^j_c enters. Hence, taking just the variables of entering events, gets φ a one-in-three model.

Reversely, if φ has a one-in-three model, Lemma 6 provides a key region R_{key} for U^{φ}. As all key copies exit, we can easily transform R_{key} into a key region R'_{key} for U^{φ}_{SSP} by making the accordance event a_j obeying and defining $R'_{key}(h'_{j,0}) = 1, R'_{key}(h'_{j,1}) = 0$ for all $j \in \{0, \dots, 14m-1\}$ as well as removing the region's definition on states of $S(U^{\varphi}_1)$. To argue the SSP of U^{φ}_{SSP} consider for all $i \in \{0, \dots, m-1\}$ and all $j \in \{1, 2\}$ the region $R_{i,j}$, where all events obey but $sig_{R_{i,j}}(p_i) = -1$. This regions separates every state in $\{t_{i,j,0}, \dots, t_{i,j,2}\}$ from every state in $\{t_{i,j,3}, t_{i,j,4}\}$. Analogously, let R_i be the region where just $sig_{R_i}(\tilde{x}^i_b) = -1$. This region separates states of $\{t_{i,0,0}, \dots, t_{i,0,2}\}$ from states $\{t_{i,0,3}, \dots, t_{i,0,5}\}$ as well as states of $\{t_{i,2,0}, t_{i,2,1}\}$ from $\{t_{i,2,2}, \dots, t_{i,2,4}\}$. It is easy to see that the remaining state pairs of $S(U^{\varphi}_2)$ are either separated by the key region or by a region where all events obey except for one variable event or one consistency event. Finally, as no accordance event of H' occurs in U^{φ}_2, taking the key region and for all $j \in \{0, \dots, 14m-1\}$ the region $\{h'_{j,0}, h'_{j,1}\}$ solves the remaining separation problems in H'.

Using Lemma 1, it is again possible to transfer the SSP from U^{φ}_{SSP} to its joined TS $A^{\varphi}_{\text{SSP}} = A(U^{\varphi}_{\text{SSP}})$ and back. As the polynomial time construction of A^{φ}_{SSP} is obvious just as its state degree and event manifoldness of two, the proof is complete. ◀

6 The Tractability of SSP for Linear 2-fold Transition Systems

This section shows that 2-fold SSP becomes tractable if we turn to linear TSs:

▶ **Theorem 11.** *Deciding the SSP can be done in polynomial time on linear* 2*-fold transition systems.*

To pove this theorem, we provide the following SSP-equivalent property for linear 2-fold TSs: If $A = s_0 \xrightarrow{e_1} \dots \xrightarrow{e_t} s_t$ is a linear TS then $A^i_j = s_i \xrightarrow{e_{i+1}} \dots \xrightarrow{e_j} s_j$ is called a *subsequence* of A for all $0 \leq i < j \leq t$ and A^i_j is *exactly* 2*-fold* if every contained event occurs exactly twice within A^i_j.

▶ **Lemma 12.** *A linear* 2*-fold TS* A *has the SSP if and only if* A^i_j *is not an exactly* 2*-fold subsequence for any* $0 \leq i < j \leq t$.

Proof. We reuse the simple observation that every region R of a linear TS A fulfills for all $0 \leq i < j \leq t$ that $\sum_{k=i}^{j} sig_R(e_k) = R(s_j) - R(s_i)$. Hence, if A has an exactly 2-fold subsequence A^i_j then every region R makes $R(s_j) - R(s_i)$ even, that is, $R(s_j) = R(s_i)$. This means, the two states are not separated by any region of A.

Reversely, assume A is free of exactly 2-fold subsequences and let $0 \leq i < j \leq t$. To see that s_i, s_j are separable, consider the three sequences A_i^0, A_j^i, A_t^j. If A_j^i contains an event that is unique in A then s_i, s_j are clearly separable. Otherwise, we select e_{min} from the events of A_j^i with first occurrence in A_i^0 such that the index i' of $s_{i'} \xrightarrow{e_{min}}$ is minimized. That is, we select the event e_{min} from A_j^i with leftmost first occurrence. If there is an event e in $A_i^{i'}$ that is unique in A or has its first occurrence in $A_{i'}^0$ or its second occurrence in A_t^j then a region R separating s_i, s_j is defined by $sig_R(e) = -sig_R(e_{min}) = 1$ while other events obey. If e does not exist, every event of $A_i^{i'}$ occurs twice in $A_j^{i'}$. In that case, we select e_{max} from the events of A_j^i with second occurrence in A_t^j such that the index j' of $s_{j'} \xrightarrow{e_{max}}$ is maximized. That is, we select the event e_{max} from A_j^i with rightmost second occurrence. If there is an event e in $A_{j'}^j$ that is unique in A or has its first occurrence in A_i^0 or its second occurrence in $A_t^{j'}$ then a region R separating s_i, s_j is defined by $sig_R(e) = -sig_R(e_{max}) = 1$ while other events obey. If e does not exist, every event of $A_{j'}^j$ occurs twice in $A_{j'}^i$.

But now, every event in A_j^i has a second occurrence in $A_{j'}^{i'}$ by the choice of e_{min} and e_{max}. Moreover, we have seen that every event in $A_i^{i'}$ and every event in $A_{j'}^j$ has its second occurrence in $A_{j'}^{i'}$. Hence, A has the exactly 2-fold subsequence $A_{j'}^{i'}$, a contradiction. ◀

For a proof of Theorem 11 it is now sufficient to understand that checking the linear 2-fold TS A for exactly 2-fold subsequences can be done by a straight forward algorithm in $O(t^3)$ time.

Moreover, the proof of Theorem 11 motivates an algorithm to efficiently compute separating regions of linear 2-fold TSs A. This algorithm uses a function $f(k)$ that, given an index $k \in \{0, \dots, t-1\}$, returns the index of the second occurrence of event e_{k+1} that occurs at s_k or -1 if no second occurrence exists. Hence, for two states s_i and s_j we use only $O(t)$ calls to f to parse the sequence A_j^i for the indices i' and j' and to search the event e within $A_i^{i'}$, respectively $A_{j'}^j$. If e is not found, the algorithm denies the separability of s_i and s_j and otherwise, it returns the separating region given in the proof. The function f can be preprocessed as an array in at most $O(t \log t)$ time (depending on the representation of events). After the preprocessing, the algorithm runs in linear time $O(t)$.

7 Conclusion

With the present work on the SSP, the ESSP, and feasibility we consolidate the fact that ENS synthesis is a surprisingly difficult problem. While intractability has been known before when event manifoldness and state degree are limited to small constants, we show that even a tighter restriction of event manifoldness to $k = 2$ has no positive effect on the complexity. Bringing down intractability that close to trivial inputs, makes most considerations of restricting the TS graph structure futile and hampers other promising parameters. Consequently, our results rule out many straight forward approaches from *fixed parameter tractability*, too.

References

1 Alessandra Agostini and Giorgio De Michelis. Improving flexibility of workflow management systems. In Wil M. P. van der Aalst, Jörg Desel, and Andreas Oberweis, editors, *Business Process Management, Models, Techniques, and Empirical Studies*, volume 1806 of *Lecture Notes in Computer Science*, pages 218–234. Springer, 2000. doi:10.1007/3-540-45594-9_14.

2 Eric Badouel, Luca Bernardinello, and Philippe Darondeau. The synthesis problem for elementary net systems is np-complete. *Theor. Comput. Sci.*, 186(1-2):107–134, 1997. `doi:10.1016/S0304-3975(96)00219-8`.

3 Eric Badouel, Luca Bernardinello, and Philippe Darondeau. *Petri Net Synthesis*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2015. `doi:10.1007/978-3-662-47967-4`.

4 Jordi Cortadella. Private correspondance, 2017.

5 Jordi Cortadella, Michael Kishinevsky, Alex Kondratyev, Luciano Lavagno, and Alex Yakovlev. Complete state encoding based on the theory of regions. In *2nd International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC '96), March 18-21, 1996, Aizu-Wakamatsu, Fukushima, Japan*, pages 36–47. IEEE Computer Society, 1996. URL: `http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=3564`, `doi:10.1109/ASYNC.1996.494436`.

6 August-Wilhelm Scheer. Business Process Engineering. Reference Models for Industrial Enterprises. Springer, 1994.

7 UML. Unified Modeling Language (UML). Object Management Group, 2018.

8 Kunihiko Hiraishi. Some complexity results on transition systems and elementary net systems. *Theor. Comput. Sci.*, 135(2):361–376, 1994. `doi:10.1016/0304-3975(94)90112-0`.

9 OMG. Business Process Model and Notation (BPMN). Object Management Group, 2018.

10 Cristopher Moore and J. M. Robson. Hard tiling problems with simple tiles. *Discrete & Computational Geometry*, 26(4):573–590, 2001. `doi:10.1007/s00454-001-0047-6`.

11 Christian Rosenke and Ronny Tredup. The hardness of synthesizing elementary net systems from highly restricted inputs. *CoRR*, abs/1711.00220, 2017. `arXiv:1711.00220`.

12 Grzegorz Rozenberg and Joost Engelfriet. Elementary net systems. In Wolfgang Reisig and Grzegorz Rozenberg, editors, *Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, the volumes are based on the Advanced Course on Petri Nets, held in Dagstuhl, September 1996*, volume 1491 of *Lecture Notes in Computer Science*, pages 12–121. Springer, 1996. `doi:10.1007/3-540-65306-6_14`.

13 Vincent Schmitt. Flip-flop nets. In Claude Puech and Rüdiger Reischuk, editors, *STACS 96, 13th Annual Symposium on Theoretical Aspects of Computer Science, Grenoble, France, February 22-24, 1996, Proceedings*, volume 1046 of *Lecture Notes in Computer Science*, pages 517–528. Springer, 1996. `doi:10.1007/3-540-60922-9_42`.

14 Ronny Tredup, Christian Rosenke, and Karsten Wolf. Elementary net synthesis remains np-complete even for extremely simple inputs. In Victor Khomenko and Olivier H. Roux, editors, *Application and Theory of Petri Nets and Concurrency - 39th International Conference, PETRI NETS 2018, Bratislava, Slovakia, June 24-29, 2018, Proceedings*, volume 10877 of *Lecture Notes in Computer Science*, pages 40–59. Springer, 2018. `doi:10.1007/978-3-319-91268-4_3`.

15 Alex Yakovlev and Albert Koelmans. Petri Nets and Digital Hardware Design. In Wolfgang Reisig and Grzegorz Rozenberg, editors, *Lectures on Petri Nets II: Applications*, volume 1492 of *Lecture Notes in Computer Science*, pages 154–236. Springer, 1998. `doi:10.1007/3-540-65307-4`.