Efficient Synthesis of Weighted Marked Graphs with Circular Reachability Graph, and Beyond

Raymond Devillers[1], Evgeny Erofeev(✉)*[2], and Thomas Hujsa**[3]

[1] Département d'Informatique, Université Libre de Bruxelles,
B-1050 Brussels, Belgium (rdevil@ulb.ac.be)
[2] Department of Computing Science, Carl von Ossietzky Universität Oldenburg,
D-26111 Oldenburg, Germany (evgeny.erofeev@informatik.uni-oldenburg.de)
[3] LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France (thujsa@laas.fr)

Abstract. In previous studies, several methods have been developed to synthesise Petri nets from labelled transition systems (LTS), often with structural constraints on the net and on the LTS. In this paper, we focus on Weighted Marked Graphs (WMGs) and Choice-Free (CF) Petri nets, two weighted subclasses of nets in which each place has at most one output; WMGs have the additional constraint that each place has at most one input.

We provide new conditions for checking the existence of a WMG whose reachability graph is isomorphic to a given circular LTS, i.e. forming a single cycle; we develop two new polynomial-time synthesis algorithms dedicated to these constraints: the first one is LTS-based (classical synthesis) while the second one is vector-based (weak synthesis) and more efficient in general. We show that our conditions also apply to CF synthesis in the case of three-letter alphabets, and we discuss the difficulties in extending them to CF synthesis over arbitrary alphabets.

Keywords: Weighted Petri net, weighted marked graph, choice-free net, synthesis, weak synthesis, labelled transition system, cycle, cyclic word, circular solvability, polynomial-time algorithm, P-vector, T-vector, Parikh vector.

1 Introduction

Petri nets form a highly expressive and intuitive operational model of discrete event systems, capturing the mechanisms of synchronisation, conflict and concurrency. Many of their fundamental behavioural properties are decidable, allowing to model and analyse numerous artificial and natural systems. However, most interesting model checking problems are worst-case intractable, and the efficiency of synthesis algorithms varies widely depending on the constraints imposed on the desired solution. In this study, we focus on the Petri net synthesis problem

* Supported by DFG through grant Be 1267/16-1 ASYST.

** Supported by the STAE foundation/project DAEDALUS, Toulouse, France.

from a labelled transition system (LTS), which consists in determining the existence of a Petri net whose reachability graph is isomorphic to the given LTS, and building such a Petri net solution when it exists.

In previous studies on analysis or synthesis, structural restrictions on nets encompassed *plain* nets (each weight equals 1; also called ordinary nets) [1], *homogeneous* nets (for each place p, all the output weights of p are equal) [2, 3], *free-choice* nets (the net is plain, and any two transitions sharing an input have the same set of inputs) [4, 2], *join-free* nets (each transition has at most one input place) [2, 5, 6, 3]. Recently, another kind of restriction has been considered, limiting the number of distinct labels of the LTS [7–10].

Depending on the constraints on the solution to be constructed, the complexity of the synthesis problem can vary widely: the problem can be solved in polynomial-time for bounded Petri nets [11], while aiming at *elementary net systems*, or at various other Petri net subclasses with fixed marking bound, makes the problem NP-complete [12, 13].

In this paper, we study the solvability of LTS with weighted marked graphs (WMGs; each place has at most one output and one input) and choice-free nets (CF; each place has at most one output). Both classes are important for real-world applications, and are widely studied in the literature [14–21]. We focus mainly on finite *circular LTS*, i.e. strongly connected LTS that contain a unique *cycle*[4]. In this context, we investigate the *cyclic solvability* of a word w, meaning the existence of a Petri net solution to the finite circular LTS induced by the infinite *cyclic word* w^∞. These restrictions appear in practical situations, since various complex applications can be decomposed into subsystems satisfying such constraints [22, 7, 9, 23, 19, 24–26].

Contributions. We study further the links between simple LTS structures and the reachability graph of WMGs and CF nets, as follows. First, we show that a binary (i.e. over a two-letter alphabet) LTS is CF-solvable if and only if it is WMG-solvable. Then, we develop new conditions for WMG-solving a cyclic word over an arbitrary alphabet, with a polynomial-time synthesis algorithm.

We show that a word over a three-letter alphabet is cyclically WMG-solvable iff it is cyclically CF-solvable, and that this result does not hold with four-letter alphabets. More generally, we discuss the difficulties of extending these results to CF synthesis over arbitrary alphabets.

We introduce the notion of *weak synthesis*, which aims at synthesising a Petri net from a given transition-vector Υ instead of a sequence: the solution obtained enables some sequence whose Parikh vector equals Υ. This allows to be less restrictive on the solution design. Then, we provide a polynomial-time algorithm for the weak synthesis of WMGs with circular reachability graphs.

[4] A set A of k arcs in a LTS G defines a cycle of G if the elements of A can be ordered as a sequence $a_1 \dots a_k$ such that, for each $i \in \{1, \dots, k\}$, $a_i = (n_i, \ell_i, n_{i+1})$ and $n_{k+1} = n_1$, i.e. the i-th arc a_i goes from node n_i to node n_{i+1} until the first node n_1 is reached, closing the path.

Finally, we show that our weak synthesis algorithm performs generally much faster than the sequence-based algorithm.

Comparing with [27], we provide more details, we add the equivalence result on CF nets for three-letter alphabets in Subsection 4.4 and the new Section 5 on weak synthesis, with a new synthesis algorithm and the study of its complexity.

Organisation of the paper. After recalling classical definitions, notations and properties in Section 2, we present the equivalence of CF- and WMG-solvability for 2-letter words in Section 3.

In Section 4, we focus on circular LTS: we give a new characterisation of WMG-solvability and a dedicated polynomial-time synthesis algorithm. We prove the equivalence between cyclic WMG and CF synthesis for three-letter alphabets. We also provide a number of examples showing that some of our results cannot be applied to the class of CF-nets over arbitrary alphabets.

Section 5 contains our study of the weak synthesis problem for WMGs with a circular reachability graph, with a new polynomial-time synthesis algorithm. Finally, Section 6 presents our conclusions and perspectives.

2 Classical Definitions, Notations and Properties

LTS, sequences and reachability. A *labelled transition system with initial state*, *LTS* for short, is a quadruple $TS = (S, \rightarrow, T, \iota)$ where S is the set of *states*, T is the (finite) set of *labels*, $\rightarrow \subseteq (S \times T \times S)$ is the *transition relation*, and $\iota \in S$ is the *initial state*. A label t is *enabled* at $s \in S$, written $s[t\rangle$, if $\exists s' \in S \colon (s, t, s') \in\rightarrow$, in which case s' is said to be *reachable* from s by the firing of t, and we write $s[t\rangle s'$. Generalising to any (firing) sequences $\sigma \in T^*$, $s[\varepsilon\rangle$ and $s[\varepsilon\rangle s$ are always true, with ε being the empty sequence; and $s[\sigma t\rangle s'$, i.e., σt is *enabled* from state s and leads to s' if there is some s'' with $s[\sigma\rangle s''$ and $s''[t\rangle s'$. For clarity, in case of long formulas we write $\lfloor_r \sigma \lfloor_s \tau \lfloor_q$ instead of $r[\sigma\rangle s[\tau\rangle q$, thus fixing some intermediate states along a firing sequence. A state s' is *reachable* from state s if $\exists \sigma \in T^* \colon s[\sigma\rangle s'$. The set of states reachable from s is noted $[s\rangle$.

Petri nets and reachability graphs. A (finite, place-transition) *weighted Petri net*, or *weighted net*, is a tuple $N = (P, T, W)$ where P is a finite set of *places*, T is a finite set of *transitions*, with $P \cap T = \emptyset$ and W is a *weight* function $W \colon ((P \times T) \cup (T \times P)) \rightarrow \mathbb{N}$ giving the weight of each arc. A *Petri net system*, or *system*, is a tuple $\mathcal{S} = (N, M_0)$ where N is a net and M_0 is the *initial marking*, which is a mapping $M_0 \colon P \rightarrow \mathbb{N}$ (hence a member of $\mathbb{N}^P$) indicating the initial number of *tokens* in each place. The *incidence matrix* I of the net is the integer $P \times T$-matrix with components $I(p, t) = W(t, p) - W(p, t)$.

A place $p \in P$ is *enabled by* a marking M if $M(p) \geq W(p, t)$ for every transition $t \in T$. A transition $t \in T$ is *enabled by* a marking M, denoted by $M[t\rangle$, if for all places $p \in P$, $M(p) \geq W(p, t)$. If t is enabled at M, then t can *occur* (or *fire*) in M, leading to the marking M' defined by $M'(p) = M(p) - W(p, t) + W(t, p)$, denoted $M[t\rangle M'$. A marking M' is *reachable* from M if there is a sequence of firings leading from M to M'. The set of markings reachable from M is denoted by $[M\rangle$. The *reachability graph of* $\mathcal{S}$ is the labelled transition system $RG(\mathcal{S})$ with the set of vertices $[M_0\rangle$, the set of labels T, initial state M_0 and transitions $\{(M, t, M') \mid M, M' \in [M_0\rangle \wedge M[t\rangle M'\}$. A system $\mathcal{S}$ is *bounded* if $RG(\mathcal{S})$ is finite.

Vectors. The *support* of a vector is the set of the indices of its non-null components. Consider any net $N = (P, T, W)$ with its incidence matrix I. A *T-vector* (respectively *P-vector*) is an element of $\mathbb{N}^T$ (respectively $\mathbb{N}^P$); it is called *prime* if the greatest common divisor of its components is one (i.e., it is non-null and its components do not have a common non-unit factor). A *T-semiflow* ν of the net is a non-null T-vector such that $I \cdot \nu = \mathbb{0}$. A T-semiflow is called *minimal* when it is prime and its support is not a proper superset of the support of any other T-semiflow [14].

The *Parikh vector* $\mathbf{P}(\sigma)$ of a finite transition sequence σ is a T-vector counting the number of occurrences of each transition in σ, and the *support* of σ is the support of its Parikh vector, i.e., $supp(\sigma) = supp(\mathbf{P}(\sigma)) = \{t \in T \mid \mathbf{P}(\sigma)(t) > 0\}$.

Strong connectedness and cycles in LTS. The LTS is said *reversible* if, $\forall s \in [\iota\rangle$, we have $\iota \in [s\rangle$, i.e., it is always possible to go back to the initial state; reversibility implies the strong connectedness of the LTS.

A sequence $s[\sigma\rangle s'$ is a *cycle*, or more precisely a *cycle at (or around) state* s, if $s = s'$. A non-empty cycle $s[\sigma\rangle s$ is called *small* if there is no non-empty cycle $s'[\sigma'\rangle s'$ in TS with $\mathbf{P}(\sigma') \lneqq \mathbf{P}(\sigma)$ (the definition of Parikh vectors extends readily to sequences over the set of labels T of the LTS). A cycle $s[\sigma\rangle s$ is *prime* if $\mathbf{P}(\sigma)$ is prime. TS has the *prime cycle property* if each small cycle has a prime Parikh vector.

A *circular LTS* is a finite, strongly connected LTS that contains a unique cycle; hence, it has the shape of an oriented circle. The circular LTS *induced by* a word $w = w_1 \dots w_k$ is defined as $s_0[w_1\rangle s_1[w_2\rangle s_2 \dots [w_k\rangle s_0$ with initial state s_0. All notions defined for labelled transition systems apply to Petri nets through their reachability graphs.

Some Petri net subclasses. A net $N = (P, T, W)$ is *plain* if no arc weight exceeds 1; *pure* if $\forall p \in P \colon (p^\bullet \cap {}^\bullet p) = \emptyset$, where $p^\bullet = \{t \in T \mid W(p, t) > 0\}$ and ${}^\bullet p = \{t \in T \mid W(t, p) > 0\}$; *choice-free* (CF) [28, 14] or place-output-nonbranching (ON) [23] if $\forall p \in P \colon |p^\bullet| \leq 1$; a *weighted marked graph* (WMG) if $|p^\bullet| \leq 1$ and $|{}^\bullet p| \leq 1$ for all places $p \in P$. The WMGs form a subclass of the CF nets and contain the weighted T-systems (WTSs) of [18], also known as *weighted event graphs* (WEGs) in [29], in which $\forall p \in P$, $|{}^\bullet p| = 1$ and $|p^\bullet| = 1$. Plain WEGs are also known as *marked graphs* [17] or *T-nets* [4].

Isomorphism and solvability. Two LTS $TS_1 = (S_1, \rightarrow_1, T, s_{01})$ and $TS_2 = (S_2, \rightarrow_2, T, s_{02})$ are isomorphic if there is a bijection $\zeta \colon S_1 \rightarrow S_2$ with $\zeta(s_{01}) = s_{02}$ and $(s, t, s') \in \rightarrow_1 \Leftrightarrow (\zeta(s), t, \zeta(s')) \in \rightarrow_2$, for all $s, s' \in S_1$. If an LTS TS is isomorphic to $RG(\mathcal{S})$, where $\mathcal{S}$ is a net system, we say that $\mathcal{S}$ *solves* TS. Solving a word $w = \ell_1 \dots \ell_k$ amounts to solve the acyclic LTS defined by the single path $\iota[\ell_1\rangle s_1 \dots [\ell_k\rangle s_k$. A finite word w is *cyclically solvable* if the circular LTS induced by w is solvable. An LTS is WMG- (or CF-)solvable if a WMG (or a CF system) solves it.

Separation problems. Let $TS = (S, \rightarrow, T, s_0)$ be a given labelled transition system. The theory of regions [30] characterises the solvability of an LTS through the solvability of a set of *separation problems*. In case the LTS is finite, we have to solve $\frac{1}{2}\cdot|S|\cdot(|S|-1)$ states separation problems and up to $|S|\cdot|T|$ event/state separation problems, as follows:

- A *region* of $(S, \rightarrow, T, s_0)$ is a triple $(\mathbb{R}, \mathbb{B}, \mathbb{F}) \in (\mathbb{N}^S, \mathbb{N}^T, \mathbb{N}^T)$ such that for all $(s, t, s') \in\rightarrow$, $\mathbb{R}(s) \geq \mathbb{B}(t)$ and $\mathbb{R}(s') = \mathbb{R}(s) - \mathbb{B}(t) + \mathbb{F}(t)$. A region models a place p, in the sense that $\mathbb{R}(s)$ models the token count of p at the marking corresponding to s, $\mathbb{B}(t)$ (for *backward*) models $W(p, t)$, and $\mathbb{F}(t)$ (for *forward*) models $W(t, p)$.
- A *states separation problem* (SSP for short) consists of a set of states $\{s, s'\}$ with $s \neq s'$. It is solved by a region $(\mathbb{R}, \mathbb{B}, \mathbb{F})$ when $\mathbb{R}(s) \neq \mathbb{R}(s')$, meaning the region allows to discriminate between s and s'.
- An *event/state separation problem* (ESSP for short) consists of a pair $(s, t) \in S{\times}T$ with $\neg s[t\rangle$. It is solved by a region $(\mathbb{R}, \mathbb{B}, \mathbb{F})$ when $\mathbb{R}(s) < \mathbb{B}(t)$, meaning the region allows to exclude a forbidden transition from some state.

In the rest of this paper, we interpret these two separation problems in terms of places of the hoped-for Petri net system as follows:
− For each SSP $\{s, s'\}$, $s \neq s'$, the two states s and s' must be distinguished by a place p such that $M_s(p) \neq M_{s'}(p)$, i.e. p has a different number of tokens in the markings corresponding to the two states.
− For each ESSP (s, t) with $\neg s[t\rangle$, there must exist a place p such that $M_s(p) < W(p, t)$ for the marking M_s corresponding to state s, where W refers to the arcs of the hoped-for Petri net system.

Notice that if the LTS is infinite, also the number of separation problems (of each kind) becomes infinite.

A synthesis procedure does not necessarily lead to a connected solution. However, the technique of decomposition into prime factors described in [31, 25] can always be applied first, so as to handle connected partial solutions and recombine them afterwards. Hence, in the sequel, we focus on connected nets, w.l.o.g. In the next section, we consider the CF synthesis problem with two distinct labels.

3 Reversible Binary CF Synthesis

In this section, we relate CF- to WMG-solvability for binary reversible LTS.

Lemma 1 (Pure CF-solvability).
If a reversible LTS has a CF solution, it has a pure CF solution.

Proof. Let $TS = (S, \rightarrow, T, \iota)$ be a reversible LTS. If $t \in T$ does not occur in $\rightarrow$, TS is solvable iff $TS' = (S, \rightarrow, T \setminus \{t\}, \iota)$ is solvable and a possible solution of TS is obtained by adding to any solution of TS' a transition t and a fresh place p, initially empty, with an arc from p to t (e.g. with weight 1), so that p is not a side condition[5]. We can thus assume that each label of T occurs in $\rightarrow$.

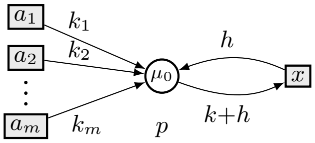

Fig. 1. A general pure ($h = 0$) or non-pure ($h > 0$) choice-free place p with initial marking μ_0. Place p has at most one outgoing transition named x. The set $\{a_1, \ldots, a_m\}$ comprises all other transitions, i.e., $T = \{x, a_1, \ldots, a_m\}$, and k_j denotes the weight of the arc from a_j to p (which could be zero).

The general form of a place in a CF solution is exhibited in Fig. 1. If $h = 0$, we are done, so that we shall assume $h > 0$. If $-h \leq k < 0$, the marking of p cannot decrease, and since x occurs in $\rightarrow$, the system cannot be reversible. If $k = 0$, for the same reason all the k_i's must be null too, $\mu_0 \geq h$, and we may drop p. Hence we assume that $k > 0$ and $\exists i : k_i > 0$.

Once x occurs, the marking of p is at least h, remains so, and since the system is reversible, all the reachable markings have at least h tokens in p. But then, if we replace p by a place p' with initially $\mu_0 - h$ tokens, the same k_i's and $h = 0$, we get exactly the same reachability graph, but with h tokens less in p' than in p. This will wipe out the side condition for p, and repeating this for each side condition, we get an equivalent pure and choice-free solution. □

Theorem 1 (Reversible binary CF-solvability).
A binary reversible LTS is CF-solvable iff it is WMG-solvable.

Proof. If we have two labels, from Lemma 1, if there is a CF solution, there will be one with places of the form exhibited in Fig. 2, hence a WMG solution. □

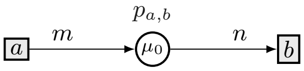

Fig. 2. A generic pure CF-place with two labels.

In the next section, the number of letters is no more restricted.

[5] A place p is a *side condition* if ${}^\bullet p \cap p^\bullet \neq \emptyset$.

4 Cyclic WMG- and CF-solvablity

In this section, we recall and extend conditions for WMG-solvability of some restricted classes of LTS formed by a single circuit, which were developed in [16].

We gradually study the separation problems – SSPs in Subsection 4.1 and ESSPs in Subsection 4.2 – for cyclic solvability with WMGs, leading to a language-theoretical characterisation of cyclically WMG-solvable sequences. The characterisation gives rise to a polynomial-time synthesis algorithm in Subsection 4.3, which is shown to be more efficient than the classical synthesis approach.

Finally, in Subsection 4.4, we study the extensibility of these results to the CF case: for three-letter alphabets, we show that a word is cyclically WMG-solvable iff it is cyclically CF-solvable; unfortunately, for arbitrary alphabets, we show with the help of examples that the other results cannot be directly extended.

In the following, two distinct labels a and b are called *(circularly) adjacent* in a word w if $w = (w_1abw_2)$ or $w = (bw_3a)$ for some $w_1, w_2, w_3 \in T^*$. We denote by $p_{a,*}$ any place $p_{a,b}$ where b is adjacent to a. Also, if $T = \{t_0, t_1, \ldots, t_m\}$ with $m > 0$, at least one label is adjacent to t_0, and at each point at least one label is adjacent to the ones we distinguished so far, until we get the whole set T; we can thus start from any label t_i instead of t_0.

Theorem 2 (Sufficient condition for cyclic WMG-solvability [16]). *Consider any word w over any finite alphabet T such that $\mathbf{P}(w)$ is prime. Suppose the following: $\forall u = w_{|t_1t_2}$ (i.e., the projection[6] of w on $\{t_1, t_2\}$) for some distinct circularly adjacent labels t_1, t_2 in w, $u = v^\ell$ for some positive integer ℓ such that $\mathbf{P}(v)$ is prime, and v is cyclically solvable by a circuit (i.e., a circular Petri net system). Then, w is cyclically solvable with a WMG.*

Theorem 3 (Cyclic WMG-solvability of ternary words [16]). *Consider a ternary word w (with three letters in its alphabet T) with Parikh vector (x, x, y) such that $\gcd(x, y) = 1$. Then, w is cyclically solvable with a WMG if and only if, for any pair $t_1 \neq t_2 \in T$ such that $w = (w_1t_1t_2w_2)$ or $w = (t_2w_3t_1)$, $u = v^\ell$ for some positive integer ℓ with $u = w_{|t_1t_2}$, $\mathbf{P}(v)$ is prime, and v is cyclically solvable by a circuit.*

For a circular LTS, the solvability of its binary projections by circuits is a sufficient condition, as specified by Theorem 2, but it turns out not to be a necessary one. Indeed, for the cyclically solvable sequence $w_1 = aacbbdabd$ (cf. left of Fig. 3), its binary projection on $\{a, b\}$ is $w_{1\,|a,b} = aabbab$ which is not cyclically solvable with a WMG (neither generally solvable). Looking only at the Parikh vector of the sequence is also not enough to establish its cyclic (un)solvability. For instance, sequences $w_2 = abcabdabd$ and $w_3 = abcbadabd$ are Parikh-equivalent: $\mathbf{P}(w_2) = \mathbf{P}(w_3) = (3, 3, 1, 2)$ (and also Parikh-equivalent to

[6] The projection of a word $w \in A^*$ on a set $A' \subseteq A$ of labels, noted $w_{|A'}$, is the word obtained by erasing in w all the occurrences of labels belonging to $A \setminus A'$. For example, the projection of the word $w = \ell_1\,\ell_2\,\ell_3\,\ell_2$ on the set $\{\ell_1, \ell_2\}$ is the word $\ell_1\,\ell_2\,\ell_2$.

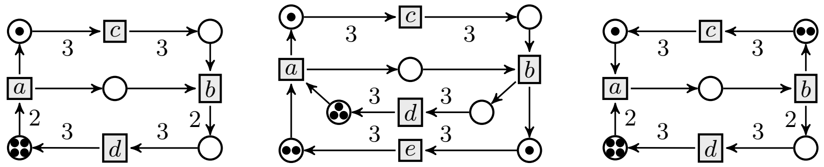

Fig. 3. The WMG on the left solves *aacbbdabd* cyclically, and the WMG in the middle solves *aacbbeabd* cyclically. On the right, the WMG solves *abcabdabd* cyclically.

w_1), but w_2 is cyclically solvable with a WMG (e.g. with the WMG on the right of Fig. 3) and w_3 is not WMG-cyclically solvable.

All the binary projections of w_1 and w_3 are cyclically WMG-solvable, except $w_{i\,|a,b}$. Only the unsolvability of $w_{3\,|a,b}$ implies the unsolvability of w_3. Since all the w_i are Parikh-equivalent, then so are their binary projections. Thus, we have to analyse the sequences themselves, without abstracting to Parikh vectors. Since the projections $w_{1\,|a,b}$ and $w_{3\,|a,b}$ are equivalent (up to cyclic rotation and swapping a and b), it is not sufficient to check the 'problematic' binary projections. We then study the conditions for solvability of separation problems.

4.1 SSPs for Prime Cycles

For any word $w = t_0 \dots t_k$, for $0 \leq i, j \leq k$ such that $i \neq j$, we note $\mathbf{P}_{ij} = \mathbf{P}(t_i t_{i+1} \dots t_{j-1})$ if $i < j$ and $\mathbf{P}_{ij} = \mathbf{P}(t_i t_{i+1} \dots t_{k-1} t_k t_0 t_1 \dots t_{j-1})$ if $i > j$.

Lemma 2 (SSPs are solvable for prime cycles). *For a cyclic transition system $TS = (S, \rightarrow, T, s_0)$ defined by some word $w = t_0 \dots t_k$, where $S = \{s_0, \dots, s_k\}$, $\rightarrow = \{(s_{i-1}, t_{i-1}, s_i) \mid 1 \leq i \leq k\} \cup \{(s_k, t_k, s_0)\}$, if $\mathbf{P}(w)$ is prime then all the SSPs are solvable.*

Proof. If $|T| = 1$, then $k = 0$ and $|S| = 1$, so that there is no SSP to solve. We may thus assume $|T| > 1$.

For each pair of distinct labels $a, b \in T$ that are adjacent in TS, construct places $p_{a,b}$ (and $p_{b,a}$ since adjacency is commutative) as in Fig. 2 with

$$m = \frac{\mathbf{P}(w)(b)}{\gcd(\mathbf{P}(w)(a), \mathbf{P}(w)(b))},\ n = \frac{\mathbf{P}(w)(a)}{\gcd(\mathbf{P}(w)(a), \mathbf{P}(w)(b))}, \tag{1}$$

and $\mu_0 = n \cdot \mathbf{P}(w)(b)$. Clearly, the markings of $p_{a,b}$ reachable by repeatedly firing $u = w_{|ab}$ are always non-negative, and the initial marking is reproduced after each repetition of the sequence u. Consider two distinct states $s_i, s_j \in S$ (w.l.o.g. $i < j$). We now demonstrate that there is at least one place of the form $p_{a,b}$ such that $M_i(p_{a,b}) \neq M_j(p_{a,b})$, where M_l denotes the marking corresponding to state s_l for $0 \leq l \leq k$. If $j - i = 1$, then any place of the form $p_{t_i,*}$ distinguishes states s_i and s_j. The same is true if $j - i > 1$ but $\forall l \in [i, j-1] : t_l = t_i$. Otherwise, choose some letter a from $t_i \dots t_{j-1}$ and an adjacent letter b. Then $M_j(p_{a,b}) = M_i(p_{a,b}) + m \cdot \mathbf{P}_{ij}(a) - n \cdot \mathbf{P}_{ij}(b)$. If $M_i(p_{a,b}) \neq M_j(p_{a,b})$, place $p_{a,b}$

distinguishes s_i and s_j. Otherwise we have $m \cdot \mathbf{P}_{ij}(a) = n \cdot \mathbf{P}_{ij}(b)$, hence, due to the choice of m and n:

$$\frac{\mathbf{P}_{ij}(a)}{\mathbf{P}(w)(a)} = \frac{\mathbf{P}_{ij}(b)}{\mathbf{P}(w)(b)}$$

(so that b also belongs to $t_i \dots t_{j-1}$). Consider some other letter c which is adjacent to a or b. If place $p_{a,c}$ distinguishes s_i and s_j, we are done. Otherwise, due to the choice of the arc weights for these places, we have

$$\frac{\mathbf{P}_{ij}(a)}{\mathbf{P}(w)(a)} = \frac{\mathbf{P}_{ij}(c)}{\mathbf{P}(w)(c)} = \frac{\mathbf{P}_{ij}(b)}{\mathbf{P}(w)(b)}.$$

Since $t_i \dots t_{j-1}$ is finite, by progressing along the adjacency relation, either we find a place which has different markings at s_i and s_j, or for all $a, b \in$ *supp*$(t_i \dots t_{j-1})$ we have

$$\frac{\mathbf{P}_{ij}(a)}{\mathbf{P}(w)(a)} = \frac{\mathbf{P}_{ij}(b)}{\mathbf{P}(w)(b)}.$$

If *supp*$(t_i \dots t_{j-1})$ = *supp*(w), $\mathbf{P}(w)$ is proportional to $\mathbf{P}(t_i \dots t_{j-1})$, but since $t_i \dots t_{j-1}$ is smaller than w (otherwise $s_i = s_j$) this contradicts the primality of $\mathbf{P}(w)$. Hence, there exist adjacent c and d such that $c \in$ *supp*$(w) \setminus$ *supp*$(t_i \dots t_{j-1})$ and $d \in$ *supp*$(t_i \dots t_{j-1})$. For the place $p_{c,d}$ we have $M_j(p_{c,d}) \neq M_i(p_{c,d})$. □

This property has some similarities with Theorem 4.1 in [32], but the preconditions are different. The reachability graph of any CF system, hence of any WMG, satisfies the prime cycle property [33, 34]. Thus, primeness of a sequence avoids solving SSPs when aiming at these two classes of Petri nets.

4.2 ESSPs in Cyclic WMG-Solvability

Now, we develop further conditions for the cyclic WMG-solvability.

Lemma 3 (Special form of WMG solutions for cycles). *If $w \in T^*$ is cyclically solvable by a WMG, there exists a WMG $\mathcal{S} = ((P, T, W), M_0)$, where P consists of places $p_{a,b}$, for each pair of distinct circularly adjacent a and b (i.e., either $w = u_1 a b u_2$ or $w = b u_3 a$).*

Proof. Consider a sequence $w = t_0 \dots t_k$, where $\mathbf{P}(w)$ is prime. Let us assume that the system $((P, T, W), M_0)$ is a WMG solving w cyclically. Due to the definition of WMGs, all the places that we have to consider are of the form schematised in Fig. 4. The arc weights may differ due to the parameter $l > 0$, but the ratio $\frac{W(a, p_{a,b})}{W(p_{a,b}, b)} = \frac{m}{n}$ is determined by the Parikh vector of w and its cyclic solvability; the initial marking is to be defined. Moreover, we have to consider only those places which are connected to the pairs of circularly adjacent transitions in w. Indeed, if $w = u_1 \lfloor_{s_i} a \lfloor_{s_{i+1}} b\ u_2$, where $b \neq a$, s_i is the state reached after performing u_1 and s_{i+1} is the state reached after performing $u_1 a$, then any place that solves the ESSP $\neg M_i[b\rangle$ is an input place for b. On the other hand, any place whose marking at M_{s_i} differs from its marking at $M_{s_{i+1}}$ is connected to a.

Hence, a place $p \in P$ solving $\neg M_i[b\rangle$ is of the form $p_{a,b}$. Since p is only affected by a and b, it also disables b at all the states between s_l and s_i in w when it is of the form $w = u_3 \lfloor_{s_j} t_j \lfloor_{s_{j+1}} b^+ \lfloor_{s_l} u_4 \lfloor_{s_i} abu_2$ with $\mathbf{P}(u_4)(b) = 0$ (in the case there is no b in the prefix between s_0 and abu_2, $s_l = s_0$). Analogously, if $t_j \neq b$, there must be a place $q \in P$ of the form $p_{t_j,b}$ that solves $\neg M_{s_j}[b\rangle$. Doing so, we ascertain that the places of the form schematised in Fig. 4 for the adjacent pairs of transitions are sufficient to handle all the ESSPs.

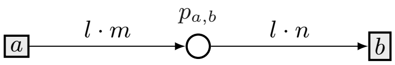

Fig. 4. A general place from a to b in a WMG solution of w: $m = \mathbf{P}(w)(b)$, $n = \mathbf{P}(w)(a)$, l may be any multiple of $1/\gcd(m, n)$.

In fact, for each pair of adjacent transitions a and b in w, a single place of the form $p_{a,b}$ is sufficient. Indeed, assume there are $p_1, p_2 \in P$ of the form $p_{a,b}$. If $\frac{M_0(p_1)}{\gcd(W(a,p_1),W(p_1,b))} \geq \frac{M_0(p_2)}{\gcd(W(a,p_2),W(p_2,b))}$ then for any $M \in [M_0\rangle$, $M(p_1) < W(p_1, b)$ implies $M(p_2) < W(p_2, b)$. Hence, p_1 is redundant in the system. It means we can choose l as we want (among the multiples of $1/\gcd(m, n)$) and only keep the place of the form $p_{a,b}$ in any solution with the smallest initial marking. Note that it may happen that we need a place $p_{a,b}$, but not $p_{b,a}$. $\square$

The existence of a WMG solution of this special form allows us to establish a necessary condition for the cyclic solvability of sequences.

Lemma 4 (A necessary condition for cyclic solvability with a WMG). *If $w \in T^*$ is cyclically solvable by a WMG, then for any adjacent transitions a and b in w, and any two occurrences of ab in $w = u_1 \lfloor_{s_r} a\, b \ldots \lfloor_{s_q} a\, b\, u_2$, the inequality*

$$\frac{\mathbf{P}_{rj}(b) - 1}{\mathbf{P}_{rj}(a)} < \frac{m}{n} < \frac{\mathbf{P}_{jq}(b) + 1}{\mathbf{P}_{jq}(a)} \tag{2}$$

holds true, where m, n are as in (1), $r \leq j \leq q$, and the right inequality is omitted when $\mathbf{P}_{jq}(a) = 0$ and the left inequality is omitted when $\mathbf{P}_{rj}(a) = 0$.

Proof. Let w be cyclically solvable with a WMG $\mathcal{S} = ((P, T, W), M_0)$ as in Lemma 3, and place $p \in P$ be of the form $p_{a,b}$ (as in Fig. 4, with $l = 1$ and a well chosen initial marking) for an adjacent pair ab. Choose two ab's in $w = u_1 \lfloor_{s_r} a \lfloor_{s_{r+1}} b \lfloor_{s_{r+2}} \cdots \lfloor_{s_q} a \lfloor_{s_{q+1}} b\, u_2$ with possibly other letters between s_{r+2} and s_q (if there is only one ab, apply the argumentation while wrapping around w circularly, i.e., $s_r[w\rangle s_q$). Since p solves ESSPs $\neg s_r[b\rangle$ and $\neg s_q[b\rangle$, the next inequalities hold true, where μ_r denotes the marking of $p_{a,b}$ at state s_r:

$$\begin{array}{rll} \neg s_r[b\rangle : & \mu_r & < n \\ s_{r+1}[b\rangle : & \mu_r + m & \geq n \\ \forall j:\ r \leq j \leq q : & \mu_r + \mathbf{P}_{rj}(a) \cdot m - \mathbf{P}_{rj}(b) \cdot n & \geq 0 \\ \neg s_q[b\rangle : & \mu_r + \mathbf{P}_{rq}(a) \cdot m - \mathbf{P}_{rq}(b) \cdot n & < n \end{array} \tag{3}$$

From the first and the third line of (3) we get $\mathbf{P}_{rj}(a) \cdot m - \mathbf{P}_{rj}(b) \cdot n > -n$. This implies:

$$\frac{\mathbf{P}_{rj}(b) - 1}{\mathbf{P}_{rj}(a)} < \frac{m}{n} \text{ when } r < j \leq q. \tag{4}$$

From the third and the fourth line of (3) we obtain

$$(\mathbf{P}_{rq}(a) - \mathbf{P}_{rj}(a)) \cdot m - (\mathbf{P}_{rq}(b) - \mathbf{P}_{rj}(b)) \cdot n < n.$$

If $\mathbf{P}_{jq}(a) \neq 0$, since $\mathbf{P}_{rq} = \mathbf{P}_{rj} + \mathbf{P}_{jq}$ this inequality can be written as

$$\frac{m}{n} < \frac{\mathbf{P}_{jq}(b) + 1}{\mathbf{P}_{jq}(a)}. \tag{5}$$

Thus, from (4) and (5) we have a necessary condition for solvability. □

In particular, Lemma 4 explains the cyclic unsolvability of the word $w_3 = \lfloor_{s_r} abcb \lfloor_{s_j} ad \lfloor_{s_q} abd$. Indeed, $\mathbf{P}(w_3)(b) = 3 = \mathbf{P}(w_3)(a)$, so that $m/n = 1$ and $1 \not< \frac{0+1}{1} = \frac{\mathbf{P}_{jq}(b)+1}{\mathbf{P}_{jq}(a)}$. Moreover, the necessary condition for cyclic sovability from Lemma 4 extends to a sufficient one in the following sense.

Lemma 5 (A sufficient condition for cyclic solvability by a WMG). *If $w \in T^*$ has a prime Parikh vector, and for each circularly adjacent ab pair in $w = \ldots \lfloor_{s_q} a\, b \ldots$, the inequality*

$$\frac{m}{n} < \frac{\mathbf{P}_{jq}(b) + 1}{\mathbf{P}_{jq}(a)} \tag{6}$$

holds true for any s_j such that $\mathbf{P}_{jq}(a) \neq 0$, then w is cyclically WMG-solvable.

Proof. We have proved in Lemma 2 that all SSPs are solvable for prime cycles. Let us consider the ESSPs at states s as in $w = \ldots \lfloor_s ab \ldots$, i.e. $\neg s[b\rangle$. Since we are looking for a WMG solution, all the sought places are of the form $p_{a,b}$ (see Lemma 3 and Fig. 4) with m, n as in (1). To define the initial marking of $p_{a,b}$, let us put $n \cdot \mathbf{P}(w)(b)$ tokens on it and fire the sequence w once completely. Choose some state s' in $w = \ldots \lfloor_{s'} a \ldots$ such that the number k of tokens on $p_{a,b}$ at state s' is minimal (it may be the case that such an s' is not unique; we can choose any such state). Define $M_0(p_{a,b}) = n \cdot \mathbf{P}(w)(b) - k$ as the initial marking of $p_{a,b}$. By construction, the firing of w reproduces the markings of $p_{a,b}$ and M_0 guarantees their non-negativity. Let us show that the constructed place $p_{a,b}$ solves all the ESSPs $\neg s[b\rangle$, where $w = \ldots \lfloor_s ab \ldots$. Consider such a state s in w (w.l.o.g. we assume $s \neq s'$, since s' certainly disables b). From $w = u_1 \lfloor_{s'} a \ldots \lfloor_s a\, b\, u_2$ (circularly) and from inequality (6) for $s_j = s'$ and $s_q = s$, we get $\mathbf{P}_{s's}(a) \cdot m - \mathbf{P}_{s's}(b) \cdot n < n$ since $\mathbf{P}_{s's}(a) > 0$. Since $M_{s'}(p_{a,b}) = 0$, $M_s(p_{a,b}) = M_{s'}(p_{a,b}) + \mathbf{P}_{s's}(a) \cdot m - \mathbf{P}_{s's}(b) \cdot n < n$, i.e., $p_{a,b}$ disables b at state s.

Now, we show that places of the form $p_{a,b}$ also solve the other ESSPs against b, i.e., at the states where b is not the subsequent transition. Sequence w (up to rotation) can be written as $w = u_1\, b^{x_1}\, u_2\, b^{x_2}\, \ldots\, u_l\, b^{x_l}$, $1 \leq l \leq \mathbf{P}(w)(b)$, and for $1 \leq i \leq l$: $x_i > 0$, $u_i \in (T \setminus \{b\})^+$. Transition b has to be disabled at all the states between successive b-blocks. Consider an arbitrary pair of such blocks b^{x_j} and $b^{x_{j+1}}$ in $w = \ldots\, b^{x_j}\, u_j\, b^{x_{j+1}}\, \ldots = \ldots\, b^{x_j} \lfloor_s u'_j \lfloor_r t\, b^{x_{j+1}}\, \ldots$, with $u_j = u'_j t$. Place $p_{t,b}$ does not allow b to fire at state r. We have to check that b is not enabled at any state between s and r, i.e., it is not enabled 'inside' u'_j. If u'_j is empty, then $s = r$, and we are done. Let $u'_j \neq \varepsilon$. Due to $\mathbf{P}(u'_j)(b) = \mathbf{P}(u_j)(b) = 0$, the marking of place $p_{t,b}$ cannot decrease from s to r, i.e., $M_s(p_{t,b}) \leq M_{s''}(p_{t,b}) \leq M_r(p_{t,b})$ for any s'' 'inside' u'_j. Since $p_{t,b}$ disables b at r, it then disables b at all states between s and r, inclusively. □

From Lemma 4 and Lemma 5 we can deduce the following characterisation.

Theorem 4 (A characterisation of cyclic WMG-solvability). *A sequence $w \in T^*$ is cyclically solvable with a WMG iff $\mathbf{P}(w)$ is prime and, for any pair of circularly adjacent labels in w, for instance $w = \ldots \lfloor_{s_q} ab \ldots$,*

$$\frac{m}{n} < \frac{\mathbf{P}_{jq}(b) + 1}{\mathbf{P}_{jq}(a)}$$

holds true with m, n as in (1) for any s_j such that $\mathbf{P}_{jq}(a) \neq 0$. A WMG solution can be found with the places of the form $p_{a,b}$ for every such pair of a and b.

4.3 A Polynomial-time Algorithm for Cyclic WMG-Solvability

From the characterisation given by Theorem 4 and the considerations above, Algorithm 1 below synthesises a cyclic WMG solution for a given sequence $w \in T^*$, if one exists.

The algorithm works as follows. Initially, the Parikh vector of the input sequence is calculated and checked for primeness in lines 2-3. If the Parikh vector is prime, we consecutively consider all the pairs of adjacent letters and examine the inequality from Theorem 4 for them. To achieve it, we take the first two letters in the sequence (lines 4-11), check if the inequality is satisfied for all the states (lines 12-18), and if so, construct a new place connecting the two letters under consideration (19-25). Then, the sequence is cyclically rotated such that the initial letter goes to the end and the second letter becomes initial (line 8). In the new sequence, we take again the first two letters (lines 9-11) and repeat the procedure for them. The algorithm stops after a complete rotation of the initial sequence, and by this moment all the pairs of adjacent letters have been checked.

The ordered alphabet is stored in the array T, and the sequence is stored in v. We use variables a and b to store the letters under consideration in each step, ia and ib to store their indices in the alphabet, and na and nb are used for counting their occurrences during the check of the inequality from Theorem 4. Variables M and $Mmin$ are used to compute the initial marking of the sought place.

Algorithm 1: Synthesis of a WMG solving a cyclic word

```
input  : $w \in T^n$, $T = \{t_0, \dots, t_{m-1}\}$
output: A WMG system $(N, M_0)$ cyclically solving $w$, if it exists
var: $T[0 \mathinner{..} m-1] = (t_0, \dots, t_{m-1})$, $v[0 \mathinner{..} n-1]$, $a$, $b$, $na$, $nb$, $ia$, $ib$, $M$, $Mmin$;
compute the Parikh vector $\mathbf{P}[0 \mathinner{..} m-1]$ of $w$;
if $\mathbf{P}$ is not prime then return unsolvable ;          // Parikh-primeness
$b \leftarrow w[0]$;
for $j = 0$ to $m-1$ do                                            // index of $b$
    if $b = T[j]$ then $ib \leftarrow j$ ;
for $i = 0$ to $n-1$ do
    $v \leftarrow w[i] \dots w[n-1] w[0] \dots w[i-1]$ ;                 // rotation of $w$
    $a \leftarrow b$, $b \leftarrow v[1]$, $ia \leftarrow ib$ ;                 // fix first adjacent pair
    for $j = 0$ to $m-1$ do
        if $b = T[j]$ then $ib \leftarrow j$ ;
    $na \leftarrow 1$, $nb \leftarrow 1$ ;
    if $a \neq b$ then
        for $k = 2$ to $n-1$ do
            if $\frac{\mathbf{P}[ib]}{\mathbf{P}[ia]} \geq \frac{\mathbf{P}[ib]-nb+1}{\mathbf{P}[ia]-na}$ then
                return unsolvable ;            // check solvability condition
            if $v[k] = T[ia]$ then $na \leftarrow na + 1$ ;
            if $v[k] = T[ib]$ then $nb \leftarrow nb + 1$ ;
        $M \leftarrow \mathbf{P}[ia] \cdot \mathbf{P}[ib]$, $Mmin \leftarrow M$;
        for $k = 0$ to $n-1$ do                              // find initial marking
            if $w[k] = a$ then $M \leftarrow M + \mathbf{P}[ib]$ ;
            if $w[k] = b$ then $M \leftarrow M - \mathbf{P}[ia]$ ;
            if $M < Mmin$ then $Mmin \leftarrow M$ ;
        add new place $p_{T[ia],T[ib]}$ to $N$ with
        $W(T[ia], p) = \mathbf{P}[ib]$, $W(p, T[ib]) = \mathbf{P}[ia]$, $M_0 = \mathbf{P}[ia] \cdot \mathbf{P}[ib] - Mmin$;
return $(N, M_0)$
```

Polynomial-time complexity of Algorithm 1. For a sequence of length n over an alphabet with m labels, the Parikh vector can be computed in $\mathcal{O}(n)$ and its primeness can be checked using e.g. the Euclidean algorithm, with a running time in $\mathcal{O}(m \cdot \log_2^2 n)$. The main `for`-cycle of Algorithm 1 involves the enumeration of all pairs of distinct states of the cycle. For each pair of adjacent labels, a run of the `for`-cycle consists of a lookup for an index in $\mathcal{O}(m)$, the verification of the inequality in $\mathcal{O}(n)$ and the construction of a place in $\mathcal{O}(n)$, which sums up to $\mathcal{O}(m+n)$. Thus, the main `for`-loop requires a runtime in $\mathcal{O}(n(n+m))$. Taking into account that $m \leq n$, and that n growths much faster than $\log_2^2 n$, the overall running time of the algorithm does not exceed $\mathcal{O}(n^2)$.

Complexity comparison: the known general approach is less efficient. For a comparison, solving a cycle of length n over m labels with a WMG amounts to solve $n(n-1)$ SSPs and $n(m-1)$ ESSPs at most. In the special case of WMG synthesis from a prime cycle, we know that all the SSPs are solvable (Lemma 2) and that solving first the other problems avoids to consider the SSPs (see [20]). Since each of the sought places has at most one input and one output, each of the separation problems seeks for 3 unknown variables, namely the initial marking of a place, the input and the output arc weights. For an ESSP, the output transition is clearly the one which has to be disabled and the input transition is to be found. So, there are $m-1$ possibilities to define a concrete ESSP, which in the worst case gives us up to $n(m-1)^2$ systems of inequalities to solve all the ESSPs.

The general region-based synthesis typically uses ILP-solvers, and using e.g. Karmarkar's algorithm [35] (which is known to be efficient) for solving an ILP-problem with k unknowns, we expect a running time of $\mathcal{O}(k^{3.5} \cdot L^2 \cdot \log L \cdot \log\log L)$ where L is the length of the input in bits. For the case of a cycle, the input of each separation problem is the matrix with the range of $(n+1) \times m$ and the vector of right sides with the range of $n+1$, where each component of the vector and of the matrix is a natural number not greater than n. Hence, the length of the input for a single separation problem can be estimated as $L = (m+1) \cdot (n+1) \cdot \log_2 n$ bits, implying a runtime of $\mathcal{O}(n^2 \cdot m^2 \cdot \log_2^2 n \cdot \log L \cdot \log\log L)$ for solving a single separation problem (the number of unknowns being equal to 3, i.e. constant). Thus the general synthesis approach would need a runtime of $\mathcal{O}(n^3 \cdot m^4 \cdot LF)$ with the logarithmic factor $LF = \log_2^2 n \cdot \log((m+1) \cdot (n+1) \cdot \log_2 n) \cdot \log\log((m+1) \cdot (n+1) \cdot \log_2 n)$. Note that, with this general approach, some redundant places may be constructed, but they can be wiped out in a post-processing phase.

4.4 CF-solvability vs WMG-solvability of Cycles

Let us now relate cyclic WMG-solvability to cyclic CF-solvability.

Theorem 5. *A sequence $u \in \{a,b,c\}^*$ is cyclically WMG-solvable iff u is cyclically CF-solvable.*

Proof. WMGs form a proper subclass of CF nets, hence the direct implication. Let now $TS = (S, T = \{a,b,c\}, \to, s_0)$ be a CF-solvable circular LTS obtained from u and $\Upsilon = \mathbf{P}(u)$. By contraposition, assume that TS is not solvable by a WMG. Then, due to Theorem 4, for some distinct states $j, q \in S$ and distinct labels $a, b \in T$

$$\frac{\Upsilon(a)}{\Upsilon(b)} \geq \frac{\mathbf{P}_{jq}(a)+1}{\mathbf{P}_{jq}(b)}. \tag{7}$$

W.l.o.g. we can choose the leftmost j satisfying (7). Then, in TS we have $r[a\rangle j$ for some $r \in S$ preceding j. Indeed, if this is not the case and either $r[b\rangle j$ or

$r[c\rangle j$, then (7) holds true for r and q, contradicting the choice of j. On the other hand, since $\mathbf{P}_{jq}(a)+1=\mathbf{P}_{rq}(a)$ and $\mathbf{P}_{jq}(b)=\mathbf{P}_{rq}(b)$, the inequality (7) implies

$$\frac{\Upsilon(a)}{\Upsilon(b)} \geq \frac{\mathbf{P}_{rq}(a)}{\mathbf{P}_{rq}(b)}. \tag{8}$$

Consider a place p which is an input place of a in a cyclic CF solution of u. From Lemma 1, we can assume pureness, i.e., the place has the form illustrated on the right of Fig. 5 with $x=a$, $y=b$, $z=c$. We must have the following constraints for p:

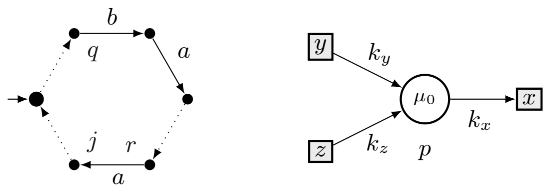

Fig. 5. u (left) is cyclically solvable by a CF system; a CF place over $\{x, y, z\}$ (right).

$$\begin{array}{lll} \text{cycle} & : & k_b \cdot \Upsilon(b) + k_c \cdot \Upsilon(c) = k_a \cdot \Upsilon(a) \\ r[a\rangle & : & M_r(p) \geq k_a \\ r[a\ldots\rangle q & : & M_q(p) = M_r(p) + k_b \cdot \mathbf{P}_{rq}(b) + k_c \cdot \mathbf{P}_{rq}(c) - k_a \cdot \mathbf{P}_{rq}(a). \end{array} \tag{9}$$

If $\mathbf{P}_{rq}(c) \geq \mathbf{P}_{rq}(a) \cdot \frac{\Upsilon(c)}{\Upsilon(a)}$, then due to (8) and (9),

$$\begin{aligned} M_q(p) = \quad & M_r(p) + k_b \cdot \mathbf{P}_{rq}(b) + k_c \cdot \mathbf{P}_{rq}(c) - k_a \cdot \mathbf{P}_{rq}(a) \\ \geq \quad & k_a + \big(k_b \cdot \frac{\Upsilon(b)}{\Upsilon(a)} + k_c \cdot \frac{\Upsilon(c)}{\Upsilon(a)} - k_a\big) \cdot \mathbf{P}_{rq}(a) = k_a, \end{aligned}$$

implying $q[a\rangle$ which contradicts $q[b\rangle$. Hence, $\mathbf{P}_{rq}(c) < \mathbf{P}_{rq}(a) \cdot \frac{\Upsilon(c)}{\Upsilon(a)}$. Together with (8), we have

$$\frac{\mathbf{P}_{rq}(b)}{\Upsilon(b)} \geq \frac{\mathbf{P}_{rq}(a)}{\Upsilon(a)} > \frac{\mathbf{P}_{rq}(c)}{\Upsilon(c)}.$$

which is equivalent to

$$\frac{\mathbf{P}_{qr}(b)}{\Upsilon(b)} \leq \frac{\mathbf{P}_{qr}(a)}{\Upsilon(a)} < \frac{\mathbf{P}_{qr}(c)}{\Upsilon(c)}. \tag{10}$$

For an arbitrary input place of b, hence of the form illustrated on the right of Fig. 5 with $x=b$, $y=a$, $z=c$,

$$\begin{array}{lll} \text{cycle} & : & k_a \cdot \Upsilon(a) + k_c \cdot \Upsilon(c) = k_b \cdot \Upsilon(b) \\ q[b\rangle & : & M_q(p) \geq k_b \\ q[b\ldots\rangle r & : & M_r(p) = M_q(p) + k_a \cdot \mathbf{P}_{qr}(a) + k_c \cdot \mathbf{P}_{qr}(c) - k_b \cdot \mathbf{P}_{qr}(b). \end{array} \tag{11}$$

Then, due to (10) and (11),

$$\begin{aligned} M_r(p) = \quad & M_q(p) + k_a \cdot \mathbf{P}_{qr}(a) + k_c \cdot \mathbf{P}_{qr}(c) - k_b \cdot \mathbf{P}_{qr}(b) \\ \geq \quad & k_b + \big(k_a \cdot \frac{\Upsilon(a)}{\Upsilon(b)} + k_c \cdot \frac{\Upsilon(c)}{\Upsilon(b)} - k_b\big) \cdot \mathbf{P}_{qr}(b) = k_b \end{aligned}$$

which implies $r[b\rangle$, a contradiction. Thus, TS is solvable by some WMG. □

When the alphabet has more than three elements, the inclusion of WMGs into CF nets is strict, i.e., there are cyclically CF-solvable sequences that are not cyclically WMG-solvable: the sequence $w = abcbad$ has a cyclic CF solution (cf. Fig. 6); for $a \lfloor_r b\, c \lfloor_q b\, a\, d$ we have $\frac{\mathbf{P}(w)(a)}{\mathbf{P}(w)(b)} = \frac{2}{2} \not< \frac{0+1}{1} = \frac{\mathbf{P}_{rq}(a)+1}{\mathbf{P}_{rq}(b)}$ which, by Theorem 4, implies the cyclic unsolvability of w by a WMG.

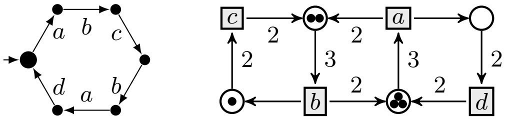

Fig. 6. Sequence *abcbad* is cyclically solved by the CF system on the right.

By Lemma 3, using places only between adjacent transitions is sufficient for cyclic WMG-solvability. For the sequence *abcbad* in Fig. 6, b follows a and c, and the input place of b in the CF solution is an output place for both a and c. The situation is similar for a, which follows b and d. However, this is not always the case when we are looking for a solution in the class of CF nets. For instance, the sequence *cabdaaab* is cyclically solvable by a CF system (see Fig. 7). In this sequence, b always follows a. But in order to solve ESSPs against b, we need an output place for c (in addition to a). Indeed, if there is a place $p_{a,b}$ as on the right of Fig. 7 which solves ESSPs against b, then for $ca \lfloor_s bdaa \lfloor_q ab$ we get

$$\begin{aligned} s[b\rangle \quad & : \mu_0 + 2 & \geq 4 \\ \neg q[b\rangle \quad & : \mu_0 + 3 \cdot 2 - 4 & < 4 \end{aligned}$$

Subtracting the first inequality from the second one, we get $4 - 4 < 0$, a contradiction. Hence, $p_{a,b}$ cannot solve all ESSPs against b in the cycle *cabdaaab*.

In a WMG, a place has at most one input. This restriction is relaxed for CF nets: multiple inputs are allowed. Let us show that a single input place for each transition is not always sufficient. For instance, consider the cyclically CF-solvable sequence *bcafdeaaabcdaafdcaaa* and Fig. 8. Assume we can solve all ESSPs against transition a with a single place p as on the right of the same figure; due to Lemma 1, we do not need any side-condition. Then, for p and $w = \lfloor_{s_0} b\ c\ a\ f\ d \lfloor_{s_5} e \lfloor_{s_6} a\ a\ a\ b\ c \lfloor_{s_{11}} d \lfloor_{s_{12}} a\ a\ f\ d \lfloor_{s_{16}} c \lfloor_{s_{17}} a\ a\ a$, the following

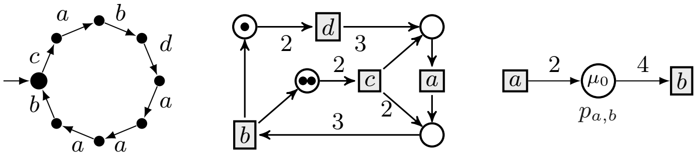

Fig. 7. *cabdaaab* is cyclically CF-solvable (middle), but is not cyclically WMG-solvable.

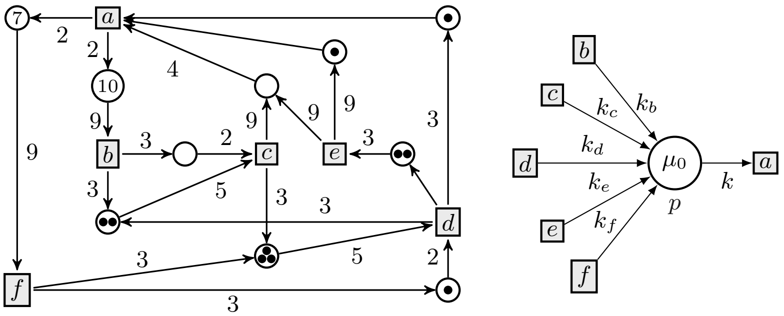

Fig. 8. $w = bcafdeaaabcdaafdcaaa$ is cyclically solved by the CF system on the left; a (pure) place of a CF system with 6 transitions on the right.

system of inequalities must hold true:

$$\begin{array}{lllll}
\text{cycle} & : & 2 \cdot k_b + 3 \cdot k_c + 3 \cdot k_d + k_e + 2 \cdot k_f & = & 9 \cdot k \quad (0) \\
\neg s_5[a\rangle & : & \mu_0 + k_b + k_c + k_d + k_f - k & < & k \quad (1) \\
s_6[aaa\rangle & : & \mu_0 + k_b + k_c + k_d + k_e + k_f - k & \geq & 3 \cdot k \quad (2) \\
\neg s_{11}[a\rangle & : & \mu_0 + 2 \cdot k_b + 2 \cdot k_c + k_d + k_e + k_f - 4 \cdot k & < & k \quad (3) \\
s_{12}[aa\rangle & : & \mu_0 + 2 \cdot k_b + 2 \cdot k_c + 2 \cdot k_d + k_e + k_f - 4 \cdot k & \geq & 2 \cdot k \quad (4) \\
\neg s_{16}[a\rangle & : & \mu_0 + 2 \cdot k_b + 2 \cdot k_c + 3 \cdot k_d + k_e + 2 \cdot k_f - 6 \cdot k & < & k \quad (5) \\
s_{17}[aaa\rangle & : & \mu_0 + 2 \cdot k_b + 3 \cdot k_c + 3 \cdot k_d + k_e + 2 \cdot k_f - 6 \cdot k & \geq & 3 \cdot k \quad (6)
\end{array}$$

From the system above we obtain:

$$\begin{array}{llll}
(2) - (1) & : & k_e & > 2 \cdot k \\
(4) - (3) & : & k_d & > k \\
(6) - (5) & : & k_c & > 2 \cdot k
\end{array}$$

which implies $3 \cdot k_c + 3 \cdot k_d + k_e > 11 \cdot k$, contradicting the equality (0). Hence, the ESSPs against a cannot be solved by a single place.

5 Weak Synthesis of WMGs in Polynomial-time

For any given word w over a set of labels T whose support equals T, each system $\mathcal{S} = ((P, T, W), M_0)$ that cyclically WMG-solves w, when it exists, has a unique minimal (hence prime) T-semiflow Υ with support T, since it is live (meaning that for each transition t, from each reachable marking M, a marking M' is reachable from M that enables t) and bounded (see [18]). In some situations, it might be sufficient to specify only the desired unique minimal T-semiflow, which leads to what we call a *weak synthesis* problem. Given such a prime Parikh vector Υ, the aim is thus to construct a WMG cyclically solving an arbitrary sequence whose Parikh vector equals Υ. In this section, we provide a method for constructing a solution in polynomial-time. To achieve it, we first need to recall known liveness conditions for WMGs and their circuit subclass.

5.1 Previous results on liveness

In [29], a polynomial-time sufficient condition of liveness is developed for the well-formed, strongly connected weighted event graphs (WEGs), equivalent to the well-formed, strongly connected WMGs. Under these assumptions, each place has exactly one ingoing and one outgoing transitions. Variants of this liveness condition for other classes of nets are given in [6], Theorems 4.2 and 5.5.

Additional notions. We introduce the following notions for our purpose:
– For any place p, gcd_p denotes the gcd of all input and output weights of p.
– A marking M_0 satisfies the *useful tokens condition* if, for each place p, $M_0(p)$ is a multiple of gcd_p. Indeed, if $M_0(p) = k \cdot \text{gcd}_p + r$ for some non-negative integers k and r such that $0 < r < \text{gcd}_p$, then r tokens are never used by any firing (see [29, 6] for more details).
– A net (P, T, W) with incidence matrix I is *conservative* if there is a P-vector $X \geq \mathbb{1}^{|P|}$ such that $X \cdot I = \mathbb{0}^{|T|}$, where $\mathbb{1}^{|P|}$ denotes the vector of size $|P|$ in which each component has value 1. Such a P-vector X is called a *conservativeness* vector. The net is 1-*conservative* if $\mathbb{1}^{|P|}$ is a conservativeness vector, i.e. if for each transition, the sum of its input weights equals the sum of its output weights.
– A net N is *structurally bounded* if for each marking M_0, (N, M_0) is bounded.
– By Theorem 4.11 in [18], a live WTS (N, M_0) is bounded iff N is conservative. We focus on live and bounded WMG solutions (which are WTS), hence on conservative, thus structurally bounded (see [36]), solutions.
– The scaling operation (Definition 3.1 in [6]): The multiplication of all input and output weights of a place p together with its marking by a positive rational number α_p is a *scaling* of the place p if the resulting input and output weights and marking are integers. If each place p of a system is scaled by a positive rational α_p, the system is said to be scaled by the vector α whose components are the scaling factors α_p.

Recalling Theorem 3.2 in [6], if $\mathcal{S} = ((P, T, W), M_0)$ is a system and α is a vector of $|P|$ positive rational components, then scaling $\mathcal{S}$ by α preserves the feasible sequences of firings. We deduce from it, and from Theorem 3.5 in the same paper, the following result.

Lemma 6. *Consider a system $\mathcal{S}$, whose scaling by a vector α of positive rationals yields the system $\mathcal{S}'$. Then, for each feasible sequence σ, denote by M the marking reached when firing σ in $\mathcal{S}$ and by M' the marking reached when firing σ in $\mathcal{S}'$; the set of places enabled by M equals the set of places enabled by M'.*

Thus, in conservative systems, we can reason equivalently on feasible sequences and enabled places in the system scaled by a conservativeness vector, yielding a 1-conservative system whose tokens amount remains constant.

The next result is a specialisation of Theorem 4.5 in [6] to circuit Petri nets, using the fact that the liveness of circuits is monotonic, i.e. preserved upon any addition of initial tokens (see [18, 5, 6] and Theorem 7.10 in [37] with its proof).

Proposition 1 (Sufficient condition of liveness [29, 6, 37]). *Consider a conservative circuit system $\mathcal{S} = (N, M_0)$, with $N = (P, T, W)$. $\mathcal{S}$ is live if the following conditions hold:*
– for a place p_0, with $\{t_0\} = p_0^\bullet$, $M_0(p_0) = W(p_0, t_0)$;
– for every place p in $P \setminus \{p_0\}$, with $p^\bullet = \{t\}$, $M_0(p) = W(p, t) - \mathrm{gcd}_p$.
Moreover, for every marking M_0' such that $M_0' \geq M_0$, (N, M_0') is live.

In the particular case of a binary circuit, i.e. with two transitions, we recall the next characterisation condition of liveness, given as Theorem 5.2 in [29].

Proposition 2 (Liveness of binary 1-conservative circuits [29]). *Consider a 1-conservative binary circuit $\mathcal{S} = ((P, T, W), M_0)$ that fulfills the useful tokens condition, with $T = \{a, b\}$ and $P = \{p_{a,b}, p_{b,a}\}$, where $p_{a,b}$ is the output of a and $p_{b,a}$ is the output of b. Let $m = W(a, p_{a,b})$ and $n = W(p_{a,b}, b)$. Then $\mathcal{S}$ is live iff $M_0(p_{a,b}) + M_0(p_{b,a}) > m + n - 2 \cdot \mathrm{gcd}(m, n)$.*

Now, consider any 1-conservative binary circuit system $\mathcal{S}$ whose initial marking M_0 marks one place $p_{a,b}$ with its output weight $W(p_{a,b}, b)$ and the other place $p_{b,a}$ with $W(p_{b,a}, a) - \mathrm{gcd}_{p_{b,a}}$. Each marking reachable from M_0 enables exactly one place; applying Lemma 6, this is also the case for any scaling of $\mathcal{S}$, hence:

Lemma 7 (One enabled place in binary circuits). *Consider a conservative binary circuit system $\mathcal{S} = (N, M_0)$, with $N = (P, T, W)$, $T = \{a, b\}$, such that for the place $p_{a,b}$ with output b, $M_0(p_{a,b}) = W(p_{a,b}, b)$ and for the other place $p_{b,a}$ with output a, $M_0(p_{b,a}) = W(p_{b,a}, a) - \mathrm{gcd}_{p_{b,a}}$. Then each reachable marking enables exactly one place.*

This lemma will help ensuring that the reachability graph of the synthesised WMG forms a circle. Fig. 9 illustrates Lemma 7 and Proposition 1.

We recall a liveness characterisation. Since a place with an output and no input, called a *source-place*, prevents liveness, we assume there is no such place.

Proposition 3 (Liveness of WMGs [18]). *Consider a WMG $\mathcal{S}$ without source places. Then $\mathcal{S}$ is live iff each circuit P-subsystem of $\mathcal{S}$ is live.*

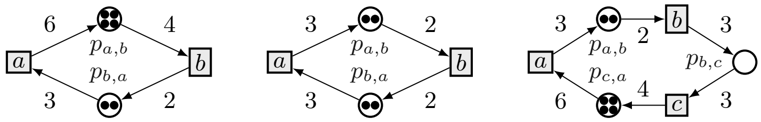

Fig. 9. These circuits are conservative and fulfill the sufficient condition of liveness of Proposition 1. On the left, $\text{gcd}_{p_{a,b}} = 2$ and $\text{gcd}_{p_{b,a}} = 1$. The system in the middle is obtained from the one on the left by scaling $p_{a,b}$ with $\frac{1}{2}$. In this second system, $\text{gcd}_{p_{a,b}} = 1$ and $\text{gcd}_{p_{b,a}} = 1$. On the right, $\text{gcd}_{p_{a,b}} = 1$, $\text{gcd}_{p_{b,c}} = 3$ and $\text{gcd}_{p_{c,a}} = 2$.

5.2 Weak synthesis of WMGs in polynomial-time

Algorithm 2 below constructs a WMG from a given prime T-vector Υ. We prove it terminates and computes a WMG cyclically solving some word with Parikh vector Υ, hence performing weak synthesis. We then show it lies in PTIME.

Algorithm 2: Weak synthesis of a WMG with circular RG.

```
Data: A prime T-vector $\Upsilon$ with support $T = \{t_1, \ldots, t_m\}$.
Result: A WMG cyclically solving a word with Parikh vector $\Upsilon$.
We construct first an unmarked WMG $N = (P, T, W)$ containing all possible
  binary circuits (which we call the complete WMG), as follows:
for each pair of distinct labels $t_i, t_j$ in $T$ do
    Add two new places $p_{i,j}$ and $p_{j,i}$ forming a binary circuit P-subnet with set
      of labels $\{t_i, t_j\}$, such that:
    $W(p_{i,j}, t_j) = W(t_j, p_{j,i}) = \frac{\Upsilon(t_i)}{\text{gcd}(\Upsilon(t_i), \Upsilon(t_j))}$
    $W(p_{j,i}, t_i) = W(t_i, p_{i,j}) = \frac{\Upsilon(t_j)}{\text{gcd}(\Upsilon(t_i), \Upsilon(t_j))}$.
Then, we construct its initial marking $M_0$, visiting the transitions in increasing
  order, as follows:
for $i = 2..m$ do
    Mark each output place $p_{i,h}$ of $t_i$ that is an input of a transition $t_h$ of
      smaller index, i.e. $h < i$, with $M_0(p) = W(p_{i,h}, t_h)$;
    Mark each input place $p_{h,i}$ of $t_i$ that is an output of a transition $t_h$ of
      smaller index, i.e. $h < i$, with
      $M_0(p) = W(p_{h,i}, t_i) - \text{gcd}_{p_{h,i}} = W(p_{h,i}, t_i) - 1$;
return $(N, M_0)$
```

Theorem 6 (Weak synthesis of a WMG). *For every prime T-vector Υ, Algorithm 2 terminates and computes a WMG cyclically solving Υ, i.e. cyclically solving some word $w \in T^*$ such that $\mathbf{P}(w) = \Upsilon$.*

Proof. The proof is illustrated in Fig. 10, 11 and 12. Consider any prime T-vector $\Upsilon \in (\mathbb{N} \setminus \{0\})^m$, where m is the number of transitions. In the first loop, we consider each pair of transitions once. In the second loop, we consider each place once. Thus, the algorithm terminates. Let us prove its correction.

If $|T| = 1$, there is one transition and $\Upsilon = (1)$: the WMG with $T = \{t_1\}$, $P = \emptyset$ and the sequence $w = t_1$ fulfill the claim. Hence, we suppose $|T| \geq 2$.

For each place $p_{i,j}$, we have $\Upsilon(t_j) \cdot W(p_{i,j}, t_j) = \Upsilon(t_i) \cdot W(t_i, p_{i,j})$, so that $-W(p_{i,j}, t_j) \cdot \Upsilon(t_j) + W(t_i, p_{i,j}) \cdot \Upsilon(t_i) = 0$, hence $I \cdot \Upsilon = 0$, where I is the incidence matrix of N. Moreover, each circuit P-subnet of N is conservative (by Corollary 3.6 in [18]).

Now, let us consider the second loop: we prove the next invariant $\mathrm{Inv}(\ell)$ to be true at the end of each iteration ℓ, for each $\ell = 1..m-1$, by induction on ℓ: $\mathrm{Inv}(\ell)$: "At the end of the ℓ-th iteration, the WMG P-subsystem $\mathcal{S}_\ell$ defined by the set of places $P_\ell = \{p_{u,v} \mid u, v \in \{1, \ldots, \ell+1\}, u \neq v\}$ is live, and each binary circuit P-subsystem of $\mathcal{S}_\ell$ has exactly one enabled place".

Before entering the loop, i.e. before the first iteration, the WMG in unmarked.

Base case: $\ell = 1$. At the end of the first iteration, $P_\ell = P_1 = \{p_{1,2}, p_{2,1}\}$, which induces a live binary circuit (by Proposition 1) with exactly one enabled place, since only one output of t_2 is enabled by M_0 and the other place is an output of t_1 considered in the second part of the loop.

Inductive case: $1 < \ell \leq m-1$. We suppose $\mathrm{Inv}(\ell-1)$ to be true, and we prove that $\mathrm{Inv}(\ell)$ is true. Thus, at the end of iteration $\ell - 1$, we suppose that the P-subsystem $\mathcal{S}_{\ell-1}$ induced by $P_{\ell-1}$ is live, and that each binary circuit P-subsystem of $\mathcal{S}_{\ell-1}$ has exactly one enabled place. The iteration ℓ marks only all the input and output places of $t_{\ell+1}$ that are inputs or outputs of transitions in $\{t_1, \ldots, t_\ell\}$. None of these places has been considered in any previous iteration, since each iteration considers only places connected to transitions of smaller index. Thus, these places are newly marked at iteration ℓ, and the only places unmarked at the end of this iteration are connected to transitions of higher index.

We deduce that, at the end of iteration ℓ:
– each binary circuit of $\mathcal{S}_\ell$ has exactly one enabled place: indeed, each such binary circuit either belongs to $\mathcal{S}_{\ell-1}$, on which the inductive hypothesis applies, or to the circuits newly marked at iteration ℓ;
– each circuit P-subsystem of $\mathcal{S}_\ell$ with three places or more is live: indeed, consider any such conservative circuit C; either C is a P-subsystem of $\mathcal{S}_{\ell-1}$, which is live by the inductive hypothesis, hence Proposition 3 applies and C is live, or C contains transition $t_{\ell+1}$, in which case C contains necessarily an output $p_{\ell+1,h}$ of $t_{\ell+1}$ with $h < \ell + 1$: since each place $p_{u,v}$ of $\mathcal{S}_\ell$ is marked with at least $W(p_{u,v}, t_v) - \mathrm{gcd}_{p_{u,v}}$ and $p_{\ell+1,h}$ is marked with $W(p_{\ell+1,h}, t_h)$, C fulfills the sufficient condition of liveness of Proposition 1, hence is live; we deduce that each circuit P-subsystem is live, hence $\mathcal{S}_\ell$ is live by Proposition 3.

We proved that $\mathrm{Inv}(\ell)$ is true for every integer $\ell = 1..m-1$. We deduce that the WMG system $\mathcal{S}_{m-1} = (N, M_0)$ obtained at the end of the last iteration, which is the system returned, fulfills $\mathrm{Inv}(m-1)$. Suppose that some marking M reachable in $\mathcal{S}_{n-1}$ enables two distinct transitions t_i and t_j. Since $\mathcal{S}_{n-1}$ is a complete WMG, there is a binary circuit P-subsystem $C_{i,j} = (N_{i,j}, M_{i,j})$ containing t_i and t_j, in which exactly one place is enabled, applying Lemma 7 (since $M|_{\{p_{i,j},p_{j,i}\}} = M_{i,j}$ is a marking reachable in $(N_{i,j}, M_0|_{\{p_{i,j},p_{j,i}\}})$). We deduce that M cannot enable both t_i and t_j, a contradiction.

Thus the WMG returned is live and each of its reachable markings enables exactly one transition. It is known that, in each live and bounded WMG, a sequence σ is feasible such that $\mathbf{P}(\sigma) = \Upsilon$ which is the unique minimal T-semiflow of the WMG (see [18, 14]). Consequently, its reachability graph is a circle, i.e. the WMG solves Υ (and σ) cyclically. We get the claim. □

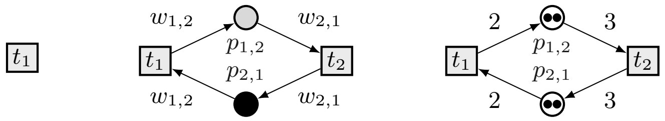

Fig. 10. Sketching Theorem 6 for 1 and 2 transitions. On the left, the circuit system $\mathcal{S}_1$ has no place and solves $\Upsilon = (1)$. In the circuit system $\mathcal{S}_2$ in the middle, the output of t_2 is marked as black and its input as grey. On the right, an instanciation of the binary case, given $\Upsilon = (3, 2)$. These systems are live, $RG(\mathcal{S}_1)$ and $RG(\mathcal{S}_2)$ are circles.

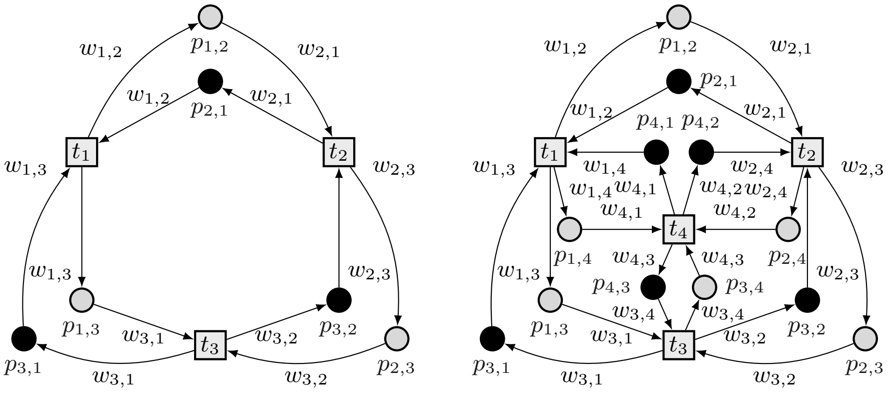

Fig. 11. Sketching Theorem 6 for 3 and 4 transitions (systems $\mathcal{S}_3$, $\mathcal{S}_4$). Each black place $p_{i,j}$ is marked with $W(p_{i,j}, t_j)$, each grey place $p_{i,j}$ is marked with $W(p_{i,j}, t_j) - \gcd_{p_{i,j}}$. On the left, in the circuit induced by $\{p_{1,2}, p_{2,1}\}$, the output of t_2 is black and its input is grey. Then, each output of t_3 is black, each input is grey. Each circuit of $\mathcal{S}_3$ is live, $\mathcal{S}_3$ is live and $RG(\mathcal{S}_3)$ is a circle. In $\mathcal{S}_4$, we keep the marking of $\mathcal{S}_3$ and mark each output of t_4 as black, each of its inputs as grey. Thus, $\mathcal{S}_4$ is live and $RG(\mathcal{S}_4)$ is a circle.

Polynomial-time complexity of Algorithm 2. Let m be the number of transitions (labels). The initial construction of the net N considers a number of transition pairs equal to $m \cdot (m-1)$. The computation of $\gcd(i, j)$ for any two integers i, j can be done using the Euclidean algorithm in $\mathcal{O}(log_2^2(\max(i, j)))$, which remains polynomial in the size of the input vector binary encoding. Computing $\frac{u}{v}$ for two integers u, v, $v \neq 0$, can also be done in $\mathcal{O}(log_2^2(\max(u, v)))$. Thus, con-

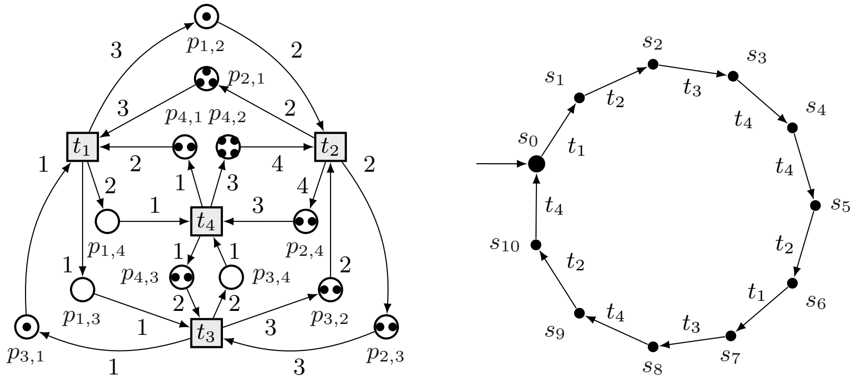

Fig. 12. Illustration of the proof of Theorem 6 for the prime T-vector $\Upsilon = (2, 3, 2, 4)$. On the left, a complete WMG $\mathcal{S}$, with all possible binary circuits. Its marking follows the black and grey places of Fig. 11. Pick any circuit P-subsystem C, e.g. the one induced by $\{p_{4,3}, p_{3,2}, p_{2,4}\}$: it is conservative and fulfills the condition of Proposition 1, hence it is live. On the right, an LTS representing $RG(\mathcal{S})$. The sequence $w = t_1\, t_2\, t_3\, t_4\, t_4\, t_2\, t_1\, t_3\, t_4\, t_2\, t_4$, with $\mathbf{P}(w) = \Upsilon$, is cyclically WMG-solvable.

structing N lies in $\mathcal{O}(m(m-1)3log_2^2(q))$ where q is the highest value in Υ. Then, the algorithm marks all places in $\mathcal{O}(m(m-1))$, knowing that the gcd of each place is 1. Hence the algorithm lies in PTIME: $\mathcal{O}(3m(m-1)log_2^2(q)+m(m-1))$, i.e. $\mathcal{O}(m(m-1)(3log_2^2(q)+1))$ where q is the highest value in Υ.

Comparison with sequence-based synthesis. Algorithm 1 uses $\mathcal{O}(n(m+n))$ steps, where m is the number of labels and n is the length of the input sequence w to be solved cyclically. Since n equals the sum of the components of $\Upsilon = \mathbf{P}(w)$, we get $q \leq n$. Also, $m \leq n$; depending on the weights, n can be exponentially larger than m. Hence, $log_2^2(q) \in \mathcal{O}(log_2^2(n))$, so that $m(m-1)(3log_2^2(q)+1) \in \mathcal{O}(m^2 \cdot log_2^2(n))$. When n is exponential in m, Algorithm 2 operates in time polynomial in m while Algorithm 1 operates in time exponential in m.

6 Conclusions and Perspectives

In this work, we specialised previous methods of analysis and synthesis to the CF nets and their WMG subclass, two useful subclasses of weighted Petri nets allowing to model various real-world applications.

We highlighted the correspondence between CF- and WMG-solvability for binary alphabets. We also tackled the case of an LTS formed of a single cycle with an arbitrary number of letters, for which we developed a characterisation of WMG-solvability together with a dedicated polynomial-time synthesis algorithm. We showed the equivalence between cyclic WMG- and CF-solvability in the case of three-letter alphabets, and that it does not extend to four-letter alphabets. We also discussed the applicability of our conditions to cyclic CF synthesis over arbitrary alphabets.

Finally, we introduced the notion of weak synthesis, allowing to be less restrictive on the solution design, and provided a polynomial-time algorithm weakly synthesising a WMG with circular reachability graph. We showed this second algorithm to often operate much faster than the sequence-based one.

As a natural continuation of the work, we expect extensions of our results in two directions: generalising the class of goal-nets and relaxing the restrictions on the LTS under consideration. Also, we plan to investigate applications of our results, in particular to *process mining* [38]: the algorithm for synthesising a net from a transition system can be applied to classical *process discovery* which is also a kind of synthesis; and the algorithm for weak synthesis can be employed in *streaming process mining*, where only some information about the behaviour of the modelled system can be stored for a later processing.

Acknowledgements

We would like to thank the anonymous referees for their careful proofreading, their relevant comments and their insightful suggestion of addressing the process mining area.

References

1. Murata, T.: Petri nets: properties, analysis and applications. Proceedings of the IEEE **77**(4) (April 1989) 541–580
2. Teruel, E., Silva, M.: Structure theory of Equal Conflict systems. Theoretical Computer Science **153**(1&2) (1996) 271–300
3. Hujsa, T., Devillers, R.: On liveness and deadlockability in subclasses of weighted Petri nets. In van der Aalst, W., Best, E., eds.: Application and Theory of Petri Nets and Concurrency: 38th International Conference, Zaragoza, Spain, Proceedings, Springer (2017) 267–287
4. Desel, J., Esparza, J.: Free Choice Petri Nets. Volume 40 of Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, New York, USA (1995)
5. Delosme, J.M., Hujsa, T., Munier-Kordon, A.: Polynomial sufficient conditions of well-behavedness for weighted join-free and choice-free systems. In: 13th International Conference on Application of Concurrency to System Design. (July 2013) 90–99
6. Hujsa, T., Delosme, J.M., Munier-Kordon, A.: Polynomial sufficient conditions of well-behavedness and home markings in subclasses of weighted Petri nets. ACM Trans. Embed. Comput. Syst. **13**(4s) (July 2014) 141:1–141:25
7. Barylska, K., Best, E., Erofeev, E., Mikulski, L., Piatkowski, M.: On binary words being Petri net solvable. In: Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data, Brussels, Belgium. (2015) 1–15
8. Barylska, K., Best, E., Erofeev, E., Mikulski, L., Piatkowski, M.: Conditions for Petri net solvable binary words. T. Petri Nets and Other Models of Concurrency **11** (2016) 137–159

9. Erofeev, E., Barylska, K., Mikulski, L., Piatkowski, M.: Generating all minimal Petri net unsolvable binary words. In: Proceedings of the Prague Stringology Conference 2016, Prague, Czech Republic. (2016) 33–46
10. Erofeev, E., Wimmel, H.: Reachability graphs of two-transition Petri nets. In: Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data, Zaragoza, Spain. (2017) 39–54
11. Badouel, E., Bernardinello, L., Darondeau, P.: Polynomial algorithms for the synthesis of bounded nets. In Mosses, P.D., Nielsen, M., Schwartzbach, M.I., eds.: TAPSOFT '95: Theory and Practice of Software Development, Berlin, Heidelberg, Springer Berlin Heidelberg (1995) 364–378
12. Badouel, E., Bernardinello, L., Darondeau, P.: The synthesis problem for elementary net systems is NP-complete. Theoretical Computer Science **186**(1) (1997) 107 – 134
13. Tredup, R.: Synthesis of structurally restricted b-bounded Petri nets: Complexity results. In Filiot, E., Jungers, R., Potapov, I., eds.: Reachability Problems, Cham, Springer International Publishing (2019) 202–217
14. Teruel, E., Colom, J.M., Silva, M.: Choice-Free Petri Nets: a Model for Deterministic Concurrent Systems with Bulk Services and Arrivals. IEEE Transactions on Systems, Man, and Cybernetics, Part A **27**(1) (1997) 73–83
15. Hujsa, T., Delosme, J.M., Munier-Kordon, A.: On the reversibility of well-behaved weighted choice-free systems. In Ciardo, G., Kindler, E., eds.: Application and Theory of Petri Nets and Concurrency, Springer (2014) 334–353
16. Devillers, R., Erofeev, E., Hujsa, T.: Synthesis of weighted marked graphs from constrained labelled transition systems. In: Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data, Satellite event of the conferences: Petri Nets and ACSD, Bratislava, Slovakia. (2018) 75–90
17. Commoner, F., Holt, A., Even, S., Pnueli, A.: Marked directed graphs. J. Comput. Syst. Sci **5**(5) (1971) 511–523
18. Teruel, E., Chrzastowski-Wachtel, P., Colom, J.M., Silva, M.: On weighted T-systems. In Jensen, K., ed.: 13th International Conference on Application and Theory of Petri Nets and Concurrency (ICATPN), LNCS. Volume 616., Springer, Berlin, Heidelberg (1992) 348–367
19. Best, E., Hujsa, T., Wimmel, H.: Sufficient conditions for the marked graph realisability of labelled transition systems. Theoretical Computer Science (2017)
20. Devillers, R., Hujsa, T.: Analysis and synthesis of weighted marked graph Petri nets. In Khomenko, V., Roux, O.H., eds.: Application and Theory of Petri Nets and Concurrency: 39th International Conference, PETRI NETS 2018, Bratislava, Slovakia, 2018, Proceedings, Springer (2018) 1–21
21. Best, E., Devillers, R., Schlachter, U., Wimmel, H.: Simultaneous Petri net synthesis. Sci. Ann. Comp. Sci. **28**(2) (2018) 199–236
22. Hujsa, T.: Contribution to the study of weighted Petri nets. PhD thesis, Pierre and Marie Curie University, Paris, France (2014)
23. Best, E., Devillers, R.: Synthesis and reengineering of persistent systems. Acta Inf. **52**(1) (2015) 35–60
24. Devillers, R., Schlachter, U.: Factorisation of Petri net solvable transition systems. In: Application and Theory of Petri Nets and Concurrency - 39th International Conference, Bratislava, Slovakia, June 24-29, 2018, Proceedings. (2018) 82–98
25. Devillers, R.: Factorisation of transition systems. Acta Inf. **55**(4) (2018) 339–362
26. Devillers, R.: Articulation of transition systems and its application to Petri net synthesis. In: Application and Theory of Petri Nets and Concurrency - 40th In-

ternational Conference, Aachen, Germany, June 23-28, 2019, Proceedings. (2019) 113–126

27. Devillers, R., Erofeev, E., Hujsa, T.: Synthesis of weighted marked graphs from circular labelled transition systems. In: Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data, Satellite event of the conferences: Petri Nets and ACSD, Aachen, Germany. (June 2019) 6–22
28. Crespi-Reghizzi, S., Mandrioli, D.: A decidability theorem for a class of vector-addition systems. Inf. Process. Lett. **3**(3) (1975) 78–80
29. Marchetti, O., Munier-Kordon, A.: A sufficient condition for the liveness of Weighted Event Graphs. European Journal of Operational Research **197**(2) (2009) 532–540
30. Badouel, E., Bernardinello, L., Darondeau, P.: Petri Net Synthesis. Springer-Verlag (2015)
31. Devillers, R.: Products of Transition Systems and Additions of Petri Nets. In: Proc. 16th International Conference on Application of Concurrency to System Design (ACSD 2016) J. Desel and A. Yakovlev (eds). (2016) 65–73
32. Devillers, R., Hujsa, T.: Analysis and synthesis of weighted marked graph Petri nets: Exact and approximate methods. Fundamenta Informaticae (2019)
33. Best, E., Devillers, R.: Characterisation of the state spaces of marked graph Petri nets. Information and Computation **253**(3) (2017) 399–410
34. Best, E., Devillers, R., Schlachter, U.: Bounded choice-free Petri net synthesis: Algorithmic issues. Acta Informatica (2017)
35. Karmarkar, N.: A new polynomial-time algorithm for linear programming. Combinatorica **4**(4) (Dec 1984) 373–395
36. Silva, M., Teruel, E., Colom, J.M.: Linear algebraic and linear programming techniques for the analysis of place/transition net systems. In Reisig, W., Rozenberg, G., eds.: Lectures on Petri Nets I: Basic Models. Volume 1491 of LNCS. (1998) 309–373
37. Hujsa, T., Devillers, R.: On deadlockability, liveness and reversibility in subclasses of weighted Petri nets. Fundam. Inform. **161**(4) (2018) 383–421
38. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016)