



An Efficient Characterization of Petri Net Solvable Binary Words

David de Frutos Escrig¹, Maciej Koutny², and Łukasz Mikulski³(✉)

¹ Dpto. Sistemas Informáticos y Computación, Facultad de Ciencias Matemáticas, Universidad Complutense de Madrid, Madrid, Spain

`defrutos@sisip.ucm.es`

² School of Computing, Newcastle University, Newcastle upon Tyne NE4 5TG, UK

`maciej.koutny@ncl.ac.uk`

³ Faculty of Mathematics and Computer Science, Nicolaus Copernicus University in Toruń, Chopina 12/18, Toruń, Poland

`lukasz.mikulski@mat.umk.pl`

Abstract. We present a simple characterization of the set of Petri net solvable binary words. It states that they are exactly the extensions of the prefixes of Petri net cyclic solvable words, by some prefix x^k , where x is any letter of the binary alphabet being considered, and k is any natural number. We derive several consequences of this characterization which, in a way, shows that the set of solvable words is ‘smaller than expected’. Therefore, the existing conjecture that all of them can be generated by quite simple net is not only confirmed, but indeed reinforced. As a byproduct of the characterization, we also present a linear time algorithm for deciding whether a binary word is solvable. The key idea is that the connection with the cyclic solvable words induces certain structural regularity. Therefore, one just needs to look for possible irregularities, which can be done in a structural way, resulting in a rather surprising linearity of the decision algorithm. Finally, we employ the obtained results to provide a characterization of reversible binary transition systems.

Keywords: Petri net · Binary word · Word solvability · Reversibility
Binary transition system

1 Introduction

In the past few years several authors investigated finite labelled transition systems (flts) which can be *solved* by Petri nets (i.e., flts’s which are isomorphic to the reachability graphs of Petri nets). The solvability problem turned out to be more complicated than expected and a very particular case of the general problem, where the flts’s are *linear* and defined by *binary* words, was studied first. Based mainly on the theory of regions [1] many useful properties of the set of solvable words have been obtained. In particular, [5] presented two decision algorithms (with quadratic time complexity) for the solvability of finite and

cyclic words. Both algorithms are based on a *quantitative* analysis that relates the number of a 's and b 's in any 'convenient' *piece* of the tested word w , checking whether in all cases the obtained quantities are *close enough*. In this way, the sets of solvable and cyclic solvable words are totally characterized. [5] was a continuation of [2], that introduced quantitative techniques used also here.

Studying the two (collections of) conditions that characterize both solvable and cyclic solvable words, we observed that they are related, and in fact look quite *similar*. However, following a closer investigation we *discovered* some important differences that make it difficult to bring to light the close connection between these two sets of solvable words. As discussed in Sect. 4, the conditions verifying the solvability of (plain) words compare *any* two consecutive parts β and γ , which are connected producing a full sequence $y\beta x\gamma y$, where $x \neq y$ are letters of the alphabet $\{a, b\}$ being considered. However, in the cyclic case, the two compared parts cover the whole verified word, i.e., $w' = y\beta x\gamma$ for a circular permutation w' of the verified word w . The first property can be seen as *local*, since we need to compare two (consecutive) pieces that can be arbitrarily long/short, whereas the second is *global*.

This paper focuses on the *structural* results. Some are recalled from [2, 5], and prove that solvable words can be decomposed into *blocks* ab^{x_i} . We reformulate the original conditions giving more precise description of block decomposition which is easier to verify. An informal statement of the resulting criteria is as follows: 'Any two adequate pieces, consecutive or not, of a solvable word must contain *quite similar* proportion of a 's and b 's'. Moreover, an *adequate* piece can be understood as a sequence $ab^{x_i}ab^{x_{i+1}} \dots ab^{x_{i+m-1}}$ of full blocks.

The only difference between the two sets of conditions is the way in which they treat the beginning of a word. In the plain case, a prefix x^k (followed by $y \neq x$) will not be considered in most cases. In the cyclic case, any part of the word is subjected to the same procedure due to the *circularity* of cyclic words.

As long as we test (nearly) any two pieces of a word in order to verify its solvability, the corresponding algorithm remains quite costly. In fact, a direct check of the restated criteria would be even worse in the general case, since we need to check also nonconsecutive pieces. Having said that, the algorithm could be used to *catch* unsolvable words in a fast way, by exhibiting two *unbalanced* (possibly nonconsecutive) sequences of blocks.

The structural properties stated in [6] imply that no cyclic solvable word w contains both aa and bb , which could be seen as the simplest version of the procedure confirming unsolvability, and is also a basic test when we check solvability. And, if w passes it, we apply the block decomposition results. More precisely, we take advantage of the result that there are no two full blocks ab^{x_i} and ab^{x_j} with $|x_i - x_j| \geq 2$, which is another example of our linear time local test.

If both simple tests are passed, we need to check more sophisticated conditions. In particular, if w is solvable and contains two consecutive ab^{e-1} blocks, then it cannot contain two consecutive ab^e blocks. This corresponds to the *structural periodicity* of solvable words: if there are no two unbalanced pieces at the level of single letters, then we proceed at the level of blocks. We then continue

the analysis in a similar hierarchical way, checking larger and larger parts of the word, until either we find a witness of unsolvability, or we terminate arriving at a trivial solvable word which proves the solvability of the original word. Fortunately, each step of this recursive procedure at least halves the length of the checked word. As a result, even if the number of iterations is logarithmic, the full cost of the algorithm is linear, as each of the iterations is linear in the current size of the checked word.

We adopted from [6] the idea to generate recursive reductions, called here *derivatives*. The authors of [6] introduced a *compression* operator that applies our general derivation operator in a particular case of the generation of a minimal unsolvable word from another, longer word. In fact, along with the compression mechanism, [6] presented an *extension* procedure that played a dual role. In a similar vein, we will introduce a general *integration* operator, and show that derivation/integration preserves and reflects solvability (and so also unsolvability!). Its application will lead, as announces above, to a pair of optimal (linear time) decision algorithms, which verify both plain and cyclic solvability.

Our initial motivation was to extend the existing results on solvability, continuing the study of the reversibility of linear transition systems initiated in the conference version of [4], where questions about decidability of several related notions are studied. In this paper, we show that a linear transition system can be reversed if and only if both the word w generated by the system, and its reversed version w^{rev} are solvable. Then, we look for a more explicit definition of the set of reversible words. In [3] it was already demonstrated that not every solvable word is reversible. However, we show that all the cyclic solvable words are reversible. Hence, towards the end of in this paper, we will look for the connections and differences between these two classes. We show that there are a *few* ‘simple’ reversible words that are not cyclic solvable, but all the other ‘more complex’ reversible words are.

The paper is organized as follows. After recalling the basic notions in Sect. 2, Sect. 3 contains new characterizations of cyclic solvable words, including the efficient algorithm for detect them. Sections 4 and 5 deal with plain solvable words and reversible words, respectively. A brief section containing conclusions ends the paper.

2 Basic Notions

The sets of all integers and non-negative integers are denoted by \mathbb{Z} and \mathbb{N} , respectively.

Words. A word over an alphabet T is a finite sequence $w \in T^*$, and it is *binary* if $|T| = 2$. The empty word is denoted by ε . The *reverse* of a word $w = t_1 \dots t_n$ is $w^{rev} = t_n \dots t_1$. For a word $w \in T^*$ and a letter $t \in T$, $|w|$ denotes the length of w , and $|w|_t$ denotes the number of occurrences of t in w . Moreover, $|w|_y^x = |w|_x / |w|_y$ and if $|w|_y = 0$, then $|w|_y^x = \infty$. A word $w' \in T^*$ is called a *subword* of $w \in T^*$ if $w = uw'v$, for some $u, v \in T^*$. In particular, w' is a *prefix*

of w if $u = \varepsilon$, a *suffix* of w if $v = \varepsilon$, and an *infix* of w if $u \neq \varepsilon \neq v$. The concatenation of k copies of a word $w \in T^*$ is denoted by w^k . Moreover, w^ω denotes an infinite word obtained by concatenating infinitely many copies of w . Note that in the denotation w^ω we *explicitly indicate the period* w which is used to construct the infinite word $ww\dots$. Similarly, uw^ω denotes an infinite word $uww\dots$ constructed using the prefix u and period w .

For two alphabets T and V , a mapping $\phi : T^* \rightarrow V^*$ is a *morphism* if $\phi(\varepsilon) = \varepsilon$ and $\phi(uv) = \phi(u)\phi(v)$, for all $u, v \in T^*$. A morphism ϕ is uniquely determined by its application to the members of T .

Transition Systems. A *finite labelled transition system* (or *flts*) is a tuple $TS = (S, T, \rightarrow, s_0)$ with a finite set of *states* S , a finite set of *labels* T , a set of *arcs* $\rightarrow \subseteq (S \times T \times S)$, and an *initial state* $s_0 \in S$. A label t is *enabled* at $s \in S$, denoted by $s[t]$, if $(s, t, s') \in \rightarrow$, for some $s' \in S$. A state s' is *reachable* from s through the execution of $\sigma \in T^*$, denoted by $s[\sigma]s'$, if there is a directed path from s to s' whose arcs are labelled consecutively by σ . The set of states reachable from s is denoted by $[s]$. A sequence $\sigma \in T^*$ is *enabled* at a state s , denoted by $s[\sigma]$, if there is some state s' such that $s[\sigma]s'$.

$t_{TS}^\bullet = \{s \in S \mid \exists s' \in S : (s', t, s) \in \rightarrow\}$ and $\bullet t_{TS} = \{s \in S \mid \exists s' \in S : (s, t, s') \in \rightarrow\}$ are respectively the sets of all states having an incoming arc labelled with t , and an outgoing arc labelled with t . The set of all arcs labelled by t is denoted by \vec{t} . We assume that each \vec{t} is nonempty, and each state is reachable from s_0 .

Two flts's, (S, T, \rightarrow, s_0) and $(S', T, \rightarrow', s'_0)$, are *isomorphic* if there is a bijection $\zeta : S \rightarrow S'$ with $\zeta(s_0) = s'_0$, and $(s, t, s') \in \rightarrow \Leftrightarrow (\zeta(s), t, \zeta(s')) \in \rightarrow'$, for all $s, s' \in S$ and $t \in T$.

Petri Nets. A (*place/transition*) *net* is a tuple $N = (P, T, F, M_0)$, where P is a finite set of *places*, T is a disjoint finite set of *transitions* (or actions), F is the *flow function* $F : ((P \times T) \cup (T \times P)) \rightarrow \mathbb{N}$ specifying the arc weights, and M_0 is the *initial marking* (where a marking is a mapping $M : P \rightarrow \mathbb{N}$).

A transition $t \in T$ is *enabled* at a marking M , denoted by $M[t]$, if $M(p) \geq F(p, t)$, for every $p \in P$. The *firing* of an enabled t at marking M leads to M' , denoted by $M[t]M'$, where $M'(p) = M(p) + F(t, p) - F(p, t)$, for every $p \in P$. The notions of enabledness and firing, $M[\sigma]$ and $M[\sigma]M'$, are extended in the usual way to sequences $\sigma \in T^*$, and $[M]$ denotes the set of all markings reachable from M . For an infinite sequence of transitions, $\sigma \in T^\omega$, we write $M[\sigma]$ if $M[\sigma']$, for infinitely many finite prefixes σ' of σ . We assume that each transition is enabled in at least one reachable marking.

Transition enabledness is *monotonic*, i.e., if a transition t is enabled at a marking M and $M(p) \leq M'(p)$, for every $p \in P$, then t is also enabled at M' .

The net N can also be specified as (N', M_0) , where $N' = (P, T, F)$.

Synthesis of Words. A Petri net $N = (P, T, F, M_0)$ net is *bounded* if the set of reachable markings $[M_0]$ is finite, and its *reachability graph* is then defined as the flts $RG(N) = ([M_0], T, \{(M, t, M') \mid M, M' \in [M_0] \wedge M[t]M'\}, M_0)$. If an

fts TS is isomorphic to the reachability graph of N , then N solves TS , and TS is synthesizable (to N).

Let $w = t_1 \dots t_m$ and $u = t_{m+1} \dots t_{m+n}$ be nonempty words, $T = \{t_1, \dots, t_m\}$, and $T' = \{t_1, \dots, t_{m+n}\}$. Then $w/w^\omega/wu^\omega$ is solvable if respectively

$$\begin{aligned} TS_w &= (\{0, \dots, m\}, T, \{(i-1, t_i, i) \mid 1 \leq i \leq m\}, 0) \\ TS_w^c &= (\{0, \dots, m-1\}, T, \{(i-1, t_i, i) \mid 1 \leq i < m\} \cup \{(m-1, t_m, 0)\}, 0) \\ TS_{w,u}^c &= (\{0, \dots, m+n-1\}, T', \\ &\quad \{(i-1, t_i, i) \mid 1 \leq i < m+n\} \cup \{(m+n-1, t_m, m)\}, 0) \end{aligned}$$

is a synthesizable fts. Moreover, a Petri net N solves $w/w^\omega/wu^\omega$ if respectively the fts $TS_w/TS_w^c/TS_{w,u}^c$ is synthesizable to N (see Fig. 1).

In the rest of this paper, T is a binary alphabet and all the finite words considered belong to T^* . Moreover, x, y, t_i typically stand for the letters in T , and unless stated otherwise, $T = \{a, b\}$.

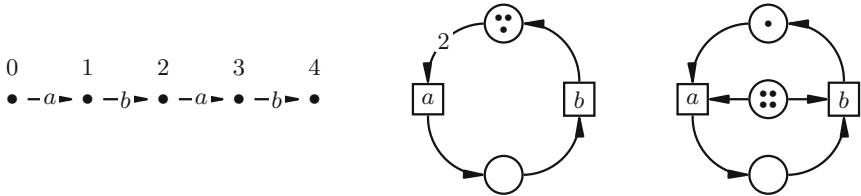


Fig. 1. The fts TS_{abab} , and two Petri nets solving $abab$. The second net without the middle place solves $(ab)^\omega$.

3 A Finer Characterisation of Cyclic Solvable Words

Two words, w and w' , are *circular equivalent* if $w = uv$ and $w' = vu$, for some u and v . We denote this by $w \stackrel{\circ}{=} w'$. One can show that $\stackrel{\circ}{=}$ is an equivalence relation. The equivalence classes of $\stackrel{\circ}{=}$ are *circular words*, and the circular word containing $w = t_1 \dots t_n$ is $[w]_{\stackrel{\circ}{=}} = \{t_i \dots t_n t_1 \dots t_{i-1} \mid 1 \leq i \leq n\}$. Hence $|[w]_{\stackrel{\circ}{=}}| \leq n$, and if $|[w]_{\stackrel{\circ}{=}}| = n$, then w is *prime*. One can show that, for each word w , there is a unique prime word u and $k \geq 1$ such that $w = u^k$. The *subwords* of $[w]_{\stackrel{\circ}{=}}$ are all the subwords of the members of $[w]_{\stackrel{\circ}{=}}$.

A word w is *cyclic solvable* if u^ω is solvable, where u is the prime satisfying $w = u^k$, for some $k \geq 1$. In such a case, every member of $[w]_{\stackrel{\circ}{=}}$ is cyclic solvable, and $[w]_{\stackrel{\circ}{=}}$ itself is *solvable*. A set of nets \mathcal{N} solves $[w]_{\stackrel{\circ}{=}}$ if they cyclic solve all the members of $[w]_{\stackrel{\circ}{=}}$.

The above definitions are more subtle than it might appear at a first glance. First, notice that u^ω and $(u^k)^\omega$ ‘construct’ the same infinite word, but do so using different periods and the solvability is tested using different fts’s, viz. TS_u^c and $TS_{u^k}^c$. Moreover, whenever u is not prime, no Petri net can solve u^ω ,

and so generate the infinite cyclic word w denoted by it using u as period. In particular, if we take $(u^k)^\omega$ with $k > 1$, then $TS_{u^k}^c$ would include at least two different states generating w , and all such states should be represented by the same marking in the reachability graph of a Petri net to which $TS_{u^k}^c$ might be synthesizable. This, however, is impossible.

Situations describe above are not a problem for the definition of cyclic solvability since we only consider prime periods u . For example, $w = aa$ is cyclic solvable because $w = a^2$, a is prime, and the fits $TS_a^c = (\{s\}, \{a\}, \{(s, a, s)\}, s)$ can be synthesized to the net $N_a = (\{p\}, \{a\}, F, M_0)$ with $M_0(p) = 1$ and the non-zero entries of F being $F(a, p) = F(p, a) = 1$. On the other hand, the fits $TS_{aa}^c = (\{s_0, s_1\}, \{a\}, \{(s_0, a, s_1), (s_1, a, s_0)\}, s_0)$ is not synthesizable.

Fact 1 (follows from Proposition 5 in [5]). *A solvable circular word does not have both aa and bb as subwords.*

Hence, we will usually assume (wlog) that aa is not a subword of cyclic solvable words. And, in order to avoid cumbersome trivial cases in proofs, we will disregard the solvable circular words $[a]_\pm$, $[b]_\pm$, and $[ab]_\pm$. As a result, we will concentrate on the remaining solvable circular words $[w]_\pm$, where $w = ab^{x_1}ab^{x_2} \dots ab^{x_n}$ and $x_1, \dots, x_n \geq 1$. Each such $[w]_\pm$ is a *circular block-word*, each w is a *block-word*, and each ab^{x_i} is a *block*. Also, a *block-subword* of $[w]_\pm$ is ε or $ab^{x_i}ab^{x_{i+1}} \dots ab^{x_j}$ (for $i \leq j$) or $ab^{x_i} \dots ab^{x_n}ab^{x_1} \dots ab^{x_j}$ (for $i > j$). Note that not all such $[w]_\pm$ are solvable, but only those that are also ‘periodic’, in the sense to be made precise later.

Fact 2 (Theorem 3 in [5]). *A circular word $[w]_\pm$ is solvable iff $|x\alpha|_y^x > |w|_y^x$, for its every subword $x\alpha y$ with $x \neq y$.*

Following the above result, a circular word $[w]_\pm$ is *balanced* if, for its every subword $x\alpha y$ with $x \neq y$:

$$|x\alpha|_y^x > |w|_y^x. \tag{1}$$

Moreover, if (1) does not hold for $x\alpha y$, then $[w]_\pm$ is *$x\alpha$ -unbalanced*. It turns out that for circular block-words it suffices to concentrate on block-subwords.

Proposition 1. *A circular block-word $[w]_\pm$ is not balanced iff there is a block-subword w' of $[w]_\pm$ such that $[w]_\pm$ is w' -unbalanced or bw' -unbalanced.*

Proof. Suppose that $[w]_\pm$ is $\alpha\alpha$ -unbalanced. Then we can extend $\alpha\alpha$ by some b^j ($j \geq 0$) obtaining a block-subword $w' = \alpha\alpha b^j$ such that $[w]_\pm$ is w' -unbalanced.

Suppose that $[w]_\pm$ is $b\alpha$ -unbalanced. If $|\alpha|_a = 0$, then $[w]_\pm$ cannot be $b\alpha$ -unbalanced. Hence $b\alpha = b^j w'$, where $j \geq 1$ and w' is a block-subword, and $[w]_\pm$ is bw' -unbalanced. \square

Let w be a block-word and $\alpha\gamma\beta\delta \doteq w$, where $\alpha, \gamma, \beta, \delta$ are block-subwords of $[w]_\pm$. Then α and β are *complementary* if $\gamma = \delta = \varepsilon$, and *mutually unbalanced*, denoted by $\alpha \frown \beta$, if $\frac{|\alpha|_b+1}{|\alpha|_a} \leq \frac{|\beta|_b-1}{|\beta|_a}$ or $\frac{|\beta|_b+1}{|\beta|_a} \leq \frac{|\alpha|_b-1}{|\alpha|_a}$.

An alternative presentation of the *balancing property* of circular words is obtained by considering complementary block-subwords. It will later be used to show that for any unbalanced block-word one can construct a pair of mutually unbalanced complementary block-subwords.

Proposition 2. *Let α and β be complementary block-subwords of $[w]_{\underline{}}$.*

- *If $[w]_{\underline{}}$ is $b\alpha$ -unbalanced or α' -unbalanced, where $\alpha'b = \alpha$, then $\alpha \prec \beta$.*
- *If $\alpha \prec \beta$, then w is not balanced.*

Proof. Let m and k be the numbers of blocks in α and β , respectively. Then $|b\alpha|_a^b = \frac{|\alpha|_b+1}{m} \leq |w|_a^b = \frac{|\beta|_b-1+|\alpha|_b+1}{m+k}$ is equivalent to

$$m|\alpha|_b + m + k|\alpha|_b + k \leq m|\alpha|_b + m + m(|\beta|_b - 1)$$

which is equivalent to $k(|\alpha|_b + 1) \leq m(|\beta|_b - 1)$. And the latter is equivalent to $\frac{|\alpha|_b+1}{m} \leq \frac{|\beta|_b-1}{k}$.

Moreover, we have that $|a\alpha'|_b^a = \frac{m}{|a\alpha'b|_b-1} \leq |w|_b^a = \frac{m+k}{|a\alpha'b|_b-1+|\beta|_b}$ is equivalent to $m(|\beta|_b + 1) \leq k(|a\alpha'b|_b - 1)$ which is equivalent to $\frac{|a\alpha'b|_b-1}{m} \geq \frac{|\beta|_b+1}{k}$. □

Next we show that as soon as we have two mutually unbalanced block-subwords, one can extend one of them by the block-subword positioned ‘in between’ the original sub-words.

Proposition 3. *Let $w = \alpha\gamma\beta\delta$ be a block-word such that $\alpha, \beta, \gamma, \delta$ are block-subwords of $[w]_{\underline{}}$ and $\alpha \prec \beta$. Then: (i) $\alpha\gamma \prec \beta$ or $\alpha \prec \gamma\beta$; and (ii) $\alpha \prec \beta\delta$ or $\delta\alpha \prec \beta$.*

Proof. (i) It is a consequence of the following general result. Whenever we have $\frac{m_1-1}{n_1} \geq \frac{m_2+1}{n_2}$, then for all p, q with $q \neq 0$, we have either $\frac{(m_1+p)-1}{n_1+q} \geq \frac{m_2+1}{n_2}$ or $\frac{m_1-1}{n_1} \geq \frac{(m_2+p)+1}{n_2+q}$. Indeed, the former is clearly obtained when $\frac{p}{q} > \frac{m_1}{n_1}$, and the latter when $\frac{p}{q} < \frac{m_2}{n_2}$. In general, if $\frac{p}{q} \geq \frac{m_2+1}{n_2}$ then, by $\frac{m_1-1}{n_1} \geq \frac{m_2+1}{n_2}$, we have $\frac{m_1-1}{n_1} \geq \frac{(m_2+p)+1}{n_2+q}$. And, by symmetry, if $\frac{p}{q} \leq \frac{m_1-1}{n_1}$ then $\frac{m_1-1}{n_1} \geq \frac{(m_2+p)+1}{n_2+q}$.

Note: The two cases considered in the second part of the proof cover all possibilities, but the previous easy cases are included to improve readability.

(ii) The proof is similar as for (i) due to the fact that δ lies ‘in between’ β and α in the circular sense. □

A block-word $w = ab^{x_1}ab^{x_2} \dots ab^{x_n}$ is *block-balanced* if, for all $1 \leq j \leq k \leq l \leq m \leq n$:

$$\frac{\sum_{i=j}^k x_i}{k-j+1} > \frac{\sum_{i=l}^m x_i}{m-l+1} \implies \frac{\sum_{i=j}^k x_i - 1}{k-j+1} < \frac{\sum_{i=l}^m x_i + 1}{m-l+1} \tag{2}$$

and

$$\frac{\sum_{i=j}^k x_i}{k-j+1} < \frac{\sum_{i=l}^m x_i}{m-l+1} \implies \frac{\sum_{i=j}^k x_i + 1}{k-j+1} > \frac{\sum_{i=l}^m x_i - 1}{m-l+1}. \tag{3}$$

Moreover, if all block-words in $[w]_{\leq}$ are block-balanced, then w is *circular block-balanced*. (Note that in the case of circular block-balancing, if all block-words in $[w]_{\leq}$ satisfy (2), then they satisfy (3) as well.) The block-word w is *contiguously block-balanced* if the conditions (2) and (3) are satisfied for $l = k + 1$. And w is *contiguously circular block-balanced* if all the members of $[w]_{\leq}$ are contiguously block-balanced.

Theorem 1. *A block-word is cyclic solvable iff it is circular block-balanced.*

Proof. Applying Proposition 3 we only need to observe that once we have $w = \alpha\gamma\beta\delta$, which is circular block unbalanced because $\alpha \frown \beta$, then $\alpha\gamma \frown \beta$ or $\alpha \frown \gamma\beta$, will be also mutually unbalanced. Suppose (wlog) that the former holds. Then another application of Proposition 3 implies that $\alpha\gamma \frown \beta\delta$ or $\alpha \frown \gamma\beta\delta$ holds, and an application of Proposition 2 concludes the proof. \square

Corollary 1. *A block-word is cyclic solvable iff it is contiguously circular block-balanced.*

Proof. Follows immediately from Theorem 1 and Proposition 3.

Propositions 8 and 9 in [5] imply that solvable circular words can be solved by ‘circular’ nets with two places and two transitions. For all $A, B \geq 1$, let \mathcal{N}_{AB} be the set of all nets $N_{AB} = (\{p_a, p_b\}, \{a, b\}, F, M_0)$ such that $M_0(p_a) + M_0(p_b) = A + B - 1$ and the non-zero values for F are $F(p_a, a) = F(a, p_b) = A$ and $F(p_b, b) = F(b, p_a) = B$.

Fact 3 (follows from Proposition 9 in [5]). *N_{AB} solves the unique solvable circular word $[w_{AB}]_{\leq}$ satisfying $|w_{AB}|_a = B$ and $|w_{AB}|_b = A$. Also, if $\gcd(A, B) = 1$ then w_{AB} is prime, and if $\gcd(A, B) = C > 1$ then $w = u^C$, where u is prime.*

A general result about the extendability of solvable words by prefixes is

Fact 4 (Proposition 4 in [2]). *If aw is solvable, then aaw is also solvable.*

In other words, a prefix a can be extended to a^k without losing solvability. This is not always possible using b instead of a , since otherwise any word would be solvable. However, any cyclic solvable word can be prefixed by a^k or b^k , for any $k \geq 1$, yielding a solvable word (although, in general, not cyclic solvable, and not even a subword of a cyclic solvable word). It seems that this fact was not yet used in the literature, and it has indeed been one of the key observations leading to our new characterization of (plain) solvable words.

Proposition 4. *Let $A, B, k \geq 1$ and $N_{AB} = (\{p_a, p_b\}, \{a, b\}, F, M_0) \in \mathcal{N}_{AB}$ be a net solving w^ω . Moreover, let $N' = (\{p_a, p_b\}, \{a, b\}, F', M'_0)$ be a net such that:*

- $F'(p_a, a) = F'(a, p_b) = A$, $F'(p_b, b) = B + k$, $F'(b, p_y) = kA$, and $F'(b, p_a) = B$ are the non-zero values of F' , and
- $M'_0(p_a) = kA + M_0(p_a)$ and $M'_0(p_b) = kA + M_0(p_b)$.

Then N' solves $a^k w^\omega$.

Note: A symmetric construction solves $b^k w^\omega$.

Proof. The kA additional tokens in p_a allow the firing of a^k . At the same time, b is not enabled since $M_0(p_b) + (k - 1)A < B + kA$ as $M_0(p_b) > A + B$. After the firing of a^k , the kA produced tokens will remain frozen at p_b , and the remaining tokens will ‘constitute’ the initial marking M_0 for N_{AB} , so that the subsequent behaviour of N' will generate w^ω . \square

A direct consequence of the last result is that all the infinite words represented by $x^k w^\omega$, where w is a cyclic solvable word, are solvable.

Given an integer $e \geq 2$, a block-word $ab^{x_1} \dots ab^{x_m}$ is an e -block-word if $x_1, \dots, x_m \in \{e - 1, e\}$. From the results in [5], it immediately follows that each cyclic solvable block-word is an e -block-word, for some $e \geq 2$. We next apply our characterization of cyclic solvable words to demonstrate a duality between the ab^{e-1} blocks and ab^e blocks, which can be swapped without affecting solvability.

Proposition 5. *An e -block-word $ab^{x_1} \dots ab^{x_m}$ is cyclic solvable iff the e -block-word $ab^{2e-1-x_1} ab^{2e-1-x_2} \dots ab^{2e-1-x_m}$ is cyclic solvable.*

Proof. We first observe that $\frac{\sum_{s=i}^{i+m-1} 2e-1-x_s}{m} = 2e-1 - \frac{\sum_{s=i}^{i+m-1} x_s}{m}$. Hence we have that $\frac{\sum_{s=i}^{i+m-1} x_s}{m} > \frac{\sum_{t=j}^{j+k-1} x_t}{k} \implies \frac{\sum_{s=i}^{i+m-1} x_s - 1}{m} < \frac{\sum_{t=j}^{j+k-1} x_t + 1}{k}$ is equivalent to:

$$\frac{\sum_{s=i}^{i+m-1} (2e-1-x_s)}{m} < \frac{\sum_{t=j}^{j+k-1} (2e-1-x_t)}{k} \implies \frac{\sum_{s=i}^{i+m-1} (2e-1-x_s)+1}{m} > \frac{\sum_{t=j}^{j+k-1} (2e-1-x_t)-1}{k}. \square$$

3.1 An Efficient Algorithm to Detect Cyclic Solvable Words

Having observed (wlog) that all the solvable cyclic block-words are e -block-words (i.e., they are built from two kinds of blocks), one can intuitively foresee the internal periodicity of cyclic solvable block-words. This means that the full word w not only appears as period of the corresponding infinite (cyclic) word, but also the internal structure of w is periodic, and the blocks are distributed in a periodic way.

Definition 1. *The derivative words of an e -block-word w are defined as follows:*

- $\partial_1(w)$ is obtained by replacing in w each block ab^{e-1} by 1, and each block ab^e by 2.
- $\partial_2(w)$ is obtained by replacing in w each block ab^{e-1} by 2, and each block ab^e by 1.

We use as alphabet $\{1, 2\}$ in order to reflect the numerical information contained in the blocks. Note that both derivative words ignore the actual value of e .

As shown later in Corollary 2, cyclic solvable e -block-words cannot contain both $ab^{e-1} ab^{e-1}$ and $ab^e ab^e$. Hence, we complete the definition of derivative words in the following way.

Definition 2. An e -block-word w is derivable if it does not contain $ab^{e-1}ab^{e-1}$ or ab^eab^e . Moreover, its derivative is defined as $\partial(w) = \partial_2(w)$ if the former holds, and as $\partial(w) = \partial_1(w)$ otherwise.

A word $b^{x_0}ab^{x_1} \dots ab^{x_m}ab^{x_{m+1}}$ is semi-derivable if $ab^{x_1} \dots ab^{x_m}$ is derivable and $1 \leq x_0, x_{m+1} \leq \min\{x_1, \dots, x_m\} + 1$.

Hence we have $\partial(w) = 1^{y_0}21^{y_1}21^{y_2} \dots 21^{y_n}$, for some $n \geq 1$ and $y_1, \dots, y_n \geq 1$. Moreover, inverse transformations can integrate words like $1^{y_0}21^{y_1}21^{y_2} \dots 21^{y_n}$. Since we can choose any $e \geq 2$, and either ab^{e-1} or ab^e to appear more often this can be done in infinitely many ways.

Definition 3. For each $e \geq 2$ and $v = 1^{y_0}21^{y_1}21^{y_2} \dots 21^{y_n}$, we define:

$$\int_{e,1} v = \phi_{1 \mapsto ab^{e-1}, 2 \mapsto ab^e}(v) \quad \text{and} \quad \int_{e,2} v = \phi_{1 \mapsto ab^e, 2 \mapsto ab^{e-1}}(v)$$

where $\phi_{1 \mapsto z, 2 \mapsto z'}$ is a morphism replacing 1 by z and 2 by z' .

Hence we always have $\partial_i(\int_{e,i} v) = v$, for all $i \in \{1, 2\}$ and $e \geq 2$.

We will now aim at a structural characterization of cyclic solvable words, that will lead to an (optimal!) algorithm detecting such words. In order to see how the relationship between (un)balanced words and their derivatives works, we discuss a simple example.

Consider $v = 212111$ which contains two mutually unbalanced block-subwords 21 and 2111 (note that $\frac{3-1}{1} \not\prec \frac{1+1}{1}$). Let us have a look at its integrals $w_1 = \int_{2,1} v = (ab^2ab)(ab^2ababab)$ and $w_2 = \int_{2,2} v = (abab^2)(abab^2ab^2ab^2)$, where the parenthesis separate contributions of the two blocks in v . Both w_1 and w_2 are unbalanced. In the first case, we can take the block-subwords $w_{11} = ab^2abab^2$ and $w_{12} = ababab$ satisfying:

$$\frac{|w_{11}|_b - 1}{|w_{11}|_a} = \frac{2 \cdot 2 + 1 \cdot 1 - 1}{2 + 1} \not\prec \frac{0 \cdot 2 + 3 \cdot 1 + 1}{0 + 3} = \frac{|w_{12}|_b + 1}{|w_{12}|_a}.$$

In the second case, we can take $w_{21} = ab^2ab^2ab^2$ and $w_{22} = abab^2ab$ satisfying:

$$\frac{|w_{21}|_b - 1}{|w_{21}|_a} = \frac{0 \cdot 1 + 3 \cdot 2 - 1}{0 + 3} \not\prec \frac{2 \cdot 1 + 1 \cdot 2 + 1}{2 + 1} = \frac{|w_{22}|_b + 1}{|w_{22}|_a}.$$

There are two things to be considered. The first is that ∂_2 works in a covariant way, so that ‘wherever’ v has ‘too many’ 1’s, w_2 has in turn ‘too many’ b ’s. On the other hand, ∂_1 works exactly the other way around, i.e., in a contravariant way. Besides, in both cases the block-subwords that now are mutually unbalanced do not coincide exactly with the integrals of the mutually unbalanced block-subwords in v . Indeed, w_{11} and w_{22} (observe again the duality!) need to expand the corresponding block-subword, borrowing the following block, while w_{12} and w_{12} lose their first blocks. In the general case, we have the following:

Theorem 2. An e -block-word w is cyclic solvable iff w is derivable and $\partial(w)$ is cyclic solvable.

Proof. First, if w contains both $ab^{e-1}ab^{e-1}$ and ab^eab^e , then it is not cyclic solvable. Let us suppose that it contains ab^eab^e , which corresponds to the covariant case, as explained in the introduction to the proof. Now, if $\partial(w)$ is not cyclic solvable, then its alternative presentation $v' = 21^{y_1}21^{y_2} \dots 21^{y_n+y_0}$ is not either. To simplify the notation, we consider first the case $y_0 = 0$. Applying Theorem 1, we should have two block-subwords $21^{y_i}21^{y_{i+1}} \dots 21^{y_{i+p-1}}$ and $21^{y_j}21^{y_{j+1}} \dots 21^{y_{j+k-1}}$, with $\frac{\sum_{s=i}^{i+p-1} y_s - 1}{p} \geq \frac{\sum_{t=j}^{j+k-1} y_t + 1}{k}$. If we consider the corresponding block-subwords in $\int_{e,2} v$:

$$\begin{aligned} - w_1 &= ab^{e-1}(ab^e)^{y_i}ab^{e-1}(ab^e)^{y_{i+1}} \dots ab^{e-1}(ab^e)^{y_{i+p-1}} \quad \text{and} \quad w'_1 = ab^{e-1}w'_1 \\ - w_2 &= ab^{e-1}(ab^e)^{y_j}ab^{e-1}(ab^e)^{y_{j+1}} \dots ab^{e-1}(ab^e)^{y_{j+k-1}} \quad \text{and} \quad w'_2 = w_2ab^{e-1} \end{aligned}$$

$$\text{we have } \frac{|w'_1|_b - 1}{|w'_1|_a} = \frac{(p-1)(e-1) + e \sum_{s=i}^{i+p-1} y_s - 1}{(p-1) + \sum_{s=i}^{i+p-1} y_s} \geq \frac{(k+1)(e-1) + e \sum_{t=j}^{j+k-1} y_t + 1}{(k+1) + \sum_{t=j}^{j+k-1} y_t} = \frac{|w'_2|_b + 1}{|w'_2|_a}.$$

And the contravariant case can be treated in a similar way, but reversing the inequality.

For the converse, let us consider an e -block-word $w = ab^{x_1} \dots ab^{x_m}$ and its two block-subwords, w'_1 and w'_2 , satisfying $\frac{|w'_1|_b - 1}{|w'_1|_a} \geq \frac{|w'_2|_b + 1}{|w'_2|_a}$. Reasoning similarly as in Proposition 1, we obtain that $w'_1 = (ab^e)^{y_1} \dots ab^{e-1}(ab^e)^{y_m}$, and $w'_2 = ab^{e-1}(ab^e)^{y_p} \dots (ab^e)^{y_{p+r-1}}ab^{e-1}$. Also, $w_1 = ab^{e-1}w'_1$ and w_2 such that $w'_2 = w_2ab^{e-1}$, are block-subwords of w , so that $\partial(w)$ contains the block-subwords $21^{y_1} \dots 21^{y_m}$ and $21^{y_p} \dots 21^{y_{p+r-1}}$. Hence we can see that all this corresponds to the situation that we had in the first part of the proof, so that reversing our arguments we conclude as required. \square

Theorem 3 (efficient recursive algorithm checking cyclic solvability).
The following PROLOG-like algorithm checks in linear time the cyclic solvability of an e -block-word $w = ab^{x_1}ab^{x_2} \dots ab^{x_m}$.

- *if* $x_1 = \dots = x_m$
 then $CyclicSolvable(w)$
- *if* $x_i = x_{i+1} \neq x_j = x_{j+1}$, for some i, j
 then $\neg CyclicSolvable(w)$
- *if* $w = (ab^e)^f(ab^{e-1}ab^e)^n(ab^{e-1})^g$, for some $f, g \in \{0, 1\}$ and $n \geq 1$
 then $CyclicSolvable(w)$
 else $CyclicSolvable(w) = CyclicSolvable(\partial(w))$

Proof. The algorithm checks the cyclic solvability of w by Theorem 2. It terminates in linear time since $|w| \geq 2 \cdot |\partial(w)|$. \square

4 Relating Solvable Words and Cyclic Solvable Words

Here we present the main result of this paper, showing that all solvable words can be obtained by prefixing with x^k prefixes of cyclic solvable words.

Theorem 4. $\{x^k w \mid k \geq 0 \text{ and } w \text{ is a prefix of a cyclic solvable word}\}$ is the set of all solvable words.

In order to prove the (\supseteq) inclusion (note that the reverse inclusion holds by Proposition 6), we give an algebraic characterization of the set of prefixes of cyclic solvable words. This is very similar to that of the cyclic solvable words in Theorem 1. We start with recalling an important result.

Theorem 5 (Theorem 2 in [5]). A word w is solvable if, for any decomposition $w = \alpha y \beta x \gamma y \delta$, with $x \neq y$, we have $|y\beta|_x^y > |x\gamma|_x^y$.

The above condition is equivalent to $\frac{|\beta|_y+1}{|x\gamma|_x} > \frac{|x\gamma|_y}{|\gamma|_x+1}$, and this is indeed very close to the conditions (2) and (3) in the definition of block-balanced words. We have found that these are exactly the prefixes of cyclic solvable words:

Theorem 6. A word $w = ab^{x_1} ab^{x_2} \dots ab^{x_m} a$ is a prefix of a cyclic solvable word iff $w' = ab^{x_1} ab^{x_2} \dots ab^{x_m}$ is block-balanced.

It is worth to note that the only differences between the statements in Theorem 1 are first Theorem 6 is that in the latter we need to introduce a final a in the word, in order to signal the completion of the previous block ab^{x_m} . On the other hand, we have the circular character of circular words, which ‘forces’ us to consider also block-subwords that can ‘glue’ the end of the word with its beginning.

As it is the case for cyclic solvable words, the characterization above remains valid when we only check contiguous block-subwords.

Theorem 7. A word $w = ab^{x_1} ab^{x_2} \dots ab^{x_m} a$ is a prefix of a cyclic solvable word iff $w' = ab^{x_1} ab^{x_2} \dots ab^{x_m}$ is contiguously block-balanced.

To prove the last two theorems we need several auxiliary results.

Proposition 6. For any $e \geq 2$, no solvable word w contains the subword $ab^e ab^e$ before the subword $ab^{e-1} ab^{e-1} a$.

Proof. Let us consider an occurrence of $ab^e ab^e$ before, and as close as possible, the occurrence of $ab^{e-1} ab^{e-1} a$. Then $w = \alpha(ab^e ab^e)(ab^{e-1} ab^e)^k(ab^{e-1} ab^{e-1})a\delta$. Let y be the first a after α , $\beta = (b^e ab^e)(ab^{e-1} ab^e)^{k-1}(ab^{e-1} ab^{e-1})$, x the last b in the subsequence $(ab^{e-1} ab^e)^k$, $\gamma = ab^{e-1} ab^{e-1}$, and the second y be the last a . Then we get $|a\beta|_a \cdot |b\gamma|_b = (2e-1) \cdot (2k+2) \cdot 2 = (k+1) \cdot (2e-1) \cdot 2 = |a\beta|_b \cdot |b\gamma|_a$, and thus $|a\beta|_a \cdot |b\gamma|_b \not\geq |a\beta|_b \cdot |b\gamma|_a$. Hence, by Fact 5 w is unsolvable and thus we obtain an obvious contradiction, finishing the proof. \square

Corollary 2. No solvable circular block-word has $ab^e ab^e$ and $ab^{e-1} ab^{e-1}$ as block-subwords.

Proof. In such a case we could always ‘put’ the occurrence of $ab^e ab^e$ ‘before’ that of $ab^{e-1} ab^{e-1}$, due to the cyclic character of circular words.

Proposition 7. A solvable word cannot have both $ab^e ab^e$ and $ab^{e-1} ab^{e-1} a$ as subwords, after any previous occurrence of b . \square

Proof. By Proposition 6, the only possibility is that we will have $ab^{e-1}ab^{e-1}a$ before ab^eab^e , possibly sharing the last a of the first, but then, reasoning as in the proof of Proposition 6, we should have $w = ab(ab^{e-1}ab^{e-1})(ab^eab^{e-1})^k(ab^eab^e)\delta$. Taking as y the first b after α , $\beta = (ab^{e-1}ab^{e-1})(ab^eab^{e-1})^k$, as x the first a in the last block (ab^eab^e) , $\gamma = b^eab^{e-1}$, and as second y the last b , we get $|b\beta|_b \cdot |a\gamma|_a = (2e - 1) \cdot (2k + 2) \cdot 2 \not\geq (2k + 2) \cdot (4e - 2) = |b\beta|_a \cdot |a\gamma|_b$, obtaining again the unsolvability of w . \square

Remark 1. It is important to observe that the total duality we had in the case of cyclic solvable words, as shown by Proposition 5, is partially lost when we consider solvable (plain) words. In particular, from Proposition 6 we conclude that ab^2ab^2abab is not a solvable word, while $ababab^2ab^2$ is solvable, because it can be decomposed as $a(babab^2ab^2)$, and $babab^2ab^2ab$ (or $babab^2ab$) is cyclic solvable. However, when considering prefixes of cyclic solvable words, this duality remains valid, since it can be obtained as an immediate consequence of the following more general result.

Proposition 8. *Let $ab^{x_1}ab^{x_2} \dots ab^{x_m}$ be a block balanced e -block-word. Then $ab^{2e-1-x_1}ab^{2e-1-x_2} \dots ab^{2e-1-x_m}$ is block balanced.*

Proof. We only need to observe that $\frac{\sum_{i=j}^k x_i - 1}{k - j + 1} < \frac{\sum_{i=l}^m x_i + 1}{m - l + 1}$ is equivalent to $\frac{\sum_{i=j}^k (2e - 1 - x_i) + 1}{k - j + 1} > \frac{\sum_{i=l}^m (2e - 1 - x_i) - 1}{m - l + 1}$, by simply observing that the following holds: $\sum_{i=j}^k (2e - 1 - x_i) = (2e - 1)(k - j + 1) - \sum_{i=j}^k x_i$. \square

As we observed for cyclic solvable block-words, *conjugation* also preserves plain block balancing, and this allows to simplify some proofs, by reducing the number of cases, as in the proof of Theorem 8 below.

Theorem 8. *Let $w = ab^{x_1}ab^{x_2} \dots ab^{x_m}$ be a block balanced derivable word, with $\partial(w) = \partial_2(w)$. Then $\partial(w) = 1^{y_0}21^{y_1}2 \dots 21^{y_{m+1}}$ is semi-derivable and considering $dw = 21^{y_s} \dots 21^{y_t}$, where $s = 0$ if $y_0 = \min\{y_1, \dots, y_m\} + 1$ and $s = 1$ otherwise; and $t = m + 1$ if $y_{m+1} = \min\{y_1, \dots, y_m\} + 1$ and $t = m$ otherwise, the block-subword $\phi_{1 \rightarrow a, 2 \rightarrow b}(dw)$ is block balanced.*

Proof. 1. Suppose that $y_{i_1} < y_{i_2} - 1$, where $i_1, i_2 \in \{1, \dots, m\}$. Then, the corresponding subwords in w , $w_1 = ab^{e-1}(ab^e)^{y_1}ab^{e-1}a$ and $w_2 = (ab^e)^{y_2}a$, are clearly not block balanced since $\frac{ey_1+2(e-1)+1}{y_1+2} \leq \frac{ey_2-1}{y_2}$ is equivalent to $y_1 \leq y_2 - 2$.

2. Exactly as above, once w contains a full sequence of blocks $(ab^e)^{f-1}$, neither y_0 nor y_{m+1} can be greater than f .

3. First notice that $\partial(w')$ can indeed contain a nonempty prefix 1^{y_0} , but at the moment we will ignore this, assuming that $y_0 = 0$. In the same way, we will remove the (possibly) *incomplete* final block $21^{y_{m+1}}$, only keeping it when $y_{m+1} = e$. Note that the obtained word $21^{y_1} \dots 21^{y_t}$ remains block balanced. Let us suppose that this is not the case. Then we have two subwords,

$$21^{y_j} \dots 21^{y_{j+m-1}} \text{ and } 21^{y_k} \dots 21^{y_{k+r-1}}, \text{ with } \frac{\sum_{i=j}^{j+m-1} y_i - 1}{m} \geq \frac{\sum_{i=k}^{k+r-1} y_i + 1}{r}.$$

To simplify, we write $T_1 = \sum_{i=j}^{j+m-1} y_i$ and $T_2 = \sum_{i=k}^{k+r-1} y_i$, so that we have $rT_1 - r \geq mT_2 + m$.

Let us consider the two subwords of w' that generate the two subwords of the derivative given above, $w_1 = ab^{e-1}(ab^e)^{y_j} \dots ab^{e-1}(ab^e)^{y_{j+m-1}}$ and $w_2 = ab^{e-1}(ab^e)^{y_k} \dots ab^{e-1}(ab^e)^{y_{k+r-1}}$. We take w'_1 , where $w_1 = ab^{e-1}w'_1$, and $w'_2 = w_2ab^{e-1}$. Hence $\frac{|w'_1|_b - 1}{|w'_1|_a} \geq \frac{|w'_2|_b + 1}{|w'_2|_a}$.

Since we have $|w'_1|_b = eT_1 + (m - 1)(e - 1)$, $|w'_1|_a = T_1 + (m - 1)$, $|w'_2|_b = eT_2 + (r + 1)(e - 1)$, and $|w'_2|_a = T_2 + r + 1$, we obtained exactly the same situation as in the proof of Theorem 2.

And this gives $(|w'_1|_b - 1)|w'_2|_a \geq (|w'_2|_b + 1)|w'_1|_a \iff rT_1 - r \geq mT_2 + m$. Finally, when the initial prefix is 1^e , the result remains valid for the whole word $21^{y_0} \dots 21^{y_t}$. Note that, although we have introduced an additional 2 at the beginning of the derivative in the case $y_0 = 0$, we have also ‘removed’ the prefix ab^{e-1} of w_1 , and proceeded with its *continuation* w'_1 . Now, we simply avoid that complication, working with the full word w_1 , and the rest of the proof works in the same way as before. \square

Now we can present the proofs of Theorems 6 and 7:

Proof (Theorem 6). By the induction on the number of blocks of w . The base case (i.e., 0) is vacuous. For the inductive case, we first apply Proposition 8 if needed, so that we can assume $\partial(w) = \partial_2(w)$. Hence, we can apply Theorem 8 to conclude that $\phi_{1 \rightarrow a, 2 \rightarrow b}(dw)$ is also block-balanced, so that we can apply the induction hypothesis, getting that it is a prefix of a cyclic solvable word, cw . Now, we can consider $c = \int_{e,2} cw$, that contains w as a subword, since the removed prefix and suffix (if that was the case) would clearly fit into the blocks before and after dw in cw . Finally, to complete the proof, we just need to undo the conjugation, if that was applied at the beginning. By Proposition 8 and Proposition 5 conjugation preserves cyclic solvability. Hence it can be applied to prefixes of the form $w = ab^{x_1}ab^{x_2} \dots ab^{x_m}a$. \square

Proof (Theorem 7). The result follows immediately from Theorem 6, using the arguments from the proof of Proposition 3. \square

We show that the conditions characterizing solvable words in [5], recalled here as Theorem 5, are nearly equivalent with our block balancing property, as the following lemmata states:

Lemma 1

1. Let w be a word and its decomposition $w = \alpha\beta\gamma\delta$ disproves the solvability of w with the use of Theorem 5. Then $\beta = x\beta'y$, and $\delta = \epsilon$ or $\delta = x\delta'$.
2. Let $w = a^k w'$ and $w' = b^{x_0} ab^{x_1} \dots ab^{x_m} ab^{x_{m+1}}$, where $x_1, \dots, x_m \in \{e, e - 1\}$ and $x_0 > 0$, for $e = \min\{x_1, \dots, x_m\} + 1 > 1$. Then:
 - (a) If $y = a$, then $y\beta = ab^{x_i} \dots ab^{x_{i+n_1-1}-1}$ and $x\gamma = bab^{x_{i+n_1}} \dots b^{x_{i+n_1+n_2-1}}$.
 - (b) If $y = b$, then $y\beta = bab^{x_i} \dots ab^{x_{i+n_1-1}}$ and $x\gamma = ab^{x_{i+n_1}} \dots b^{x_{i+n_1+n_2-1}}$.

Proof. (1) If $\beta = y\beta'$, then β' makes $|\beta'|_y < |\beta|_y$, and thus can be used instead of β . And, similarly, if $\beta = \beta'x$, then we can take $\gamma' = x\gamma$ instead of γ ; while if $\delta = y\delta'$, we can take $\gamma' = \gamma y$ instead of γ .
 (2) This is an application of part (1) to this particular class of words. \square

Lemma 2. *Let $w = a^k w'$ and $w' = b^{x_0} ab^{x_1} \dots ab^{x_m} ab^{x_{m+1}}$, where $x_1, \dots, x_m \in \{e, e - 1\}$ and $x_0 > 0$, for $e = \min\{x_1, \dots, x_m\} + 1 > 1$, and the decomposition $w = \alpha y \beta x \gamma y \delta$ disproves its solvability with the use of Theorem 5. Then:*

1. *If $y\beta = ab^{x_i} \dots ab^{x_{i+n_1}-1}$ and $x\gamma = bab^{x_{i+n_1}} \dots b^{x_{i+n_1+n_2}-1}$, then $y\beta b \frown \gamma$, and so $ab^{x_i} \dots ab^{x_{m+1}}$ is not a prefix of a cyclic solvable word.
 Conversely, if $u \frown v$ are sequences of blocks with $\frac{|u|_b-1}{|u|_a} \geq \frac{|v|_b+1}{|v|_a}$, then $\alpha uv \delta$ is unsolvable, for all α and δ .*
2. *If $y\beta = bab^{x_i} \dots ab^{x_{i+n_1}-1}$ and $x\gamma y = ab^{x_{i+n_1}} \dots b^{x_{i+n_1+n_2}-1}$, then $\beta \frown a\gamma b$, and so $ab^{x_i} \dots ab^{x_{m+1}}$ is not a prefix of a cyclic solvable word.
 Conversely, if $u \frown v$ are sequences of blocks with $\frac{|u|_b+1}{|u|_a} \leq \frac{|v|_b-1}{|v|_a}$, then $\alpha uv \delta$ is unsolvable, for all α and δ .*

Proof. (1) $\frac{|\alpha\beta|_a}{|\alpha\beta|_b} = \frac{|u|_a}{|u|_b-1} \leq \frac{|v|_a}{|v|_b+1} = \frac{|b\gamma|_a}{|b\gamma|_b}$.
 (2) $\frac{|b\beta|_b}{|b\beta|_a} = \frac{|u|_b-1}{|u|_a} \leq \frac{|v|_b+1}{|v|_a} = \frac{|a\gamma|_b}{|a\gamma|_a}$.

Note that these two cases are not totally symmetric, because in the second case introducing at least one b before u is needed to obtain an unsolvable word. \square

And finally we can present the proof of our main theorem.

Proof (Theorem 4). Let $w = a^k w'$ and $w' = b^{x_0} ab^{x_1} \dots ab^{x_m} ab^{x_{m+1}}$, where $x_1, \dots, x_m \in \{e, e - 1\}$ and $x_0 > 0$, for $e = \min\{x_1, \dots, x_m\} + 1 > 1$. We consider two possible cases:

Case 1: $k = 0$. Then $w = b^{x_0} \dots ab^{x_m} ab^{x_{m+1}}$. Suppose that the word $w' = ab^{x_1} \dots ab^{x_m} ab^{x_{m+1}}$ cannot be extended to a cyclic solvable word. Then, by Theorem 7, there exists a pair of contiguous block-subwords $u \frown v$ of w' . This gives us $w = \alpha uv \delta$, for some words α and β , and we can apply one of the cases of Lemma 2, whether we have $\frac{|u|_b-1}{|u|_a} \geq \frac{|v|_b+1}{|v|_a}$ or $\frac{|u|_b+1}{|u|_a} \leq \frac{|v|_b-1}{|v|_a}$, getting in both cases that w would be unsolvable, against the hypothesis.

Case 2: $k > 0$. Then we have $w = a^k b^{x_0} ab^{x_1} \dots ab^{x_m} ab^{x_{m+1}}$, and applying Lemma 2(2) again, if $ab^{x_1} \dots ab^{x_m} ab^{x_{m+1}}$ would not be a prefix of a cyclic solvable word, then $bab^{x_1} \dots ab^{x_m} ab^{x_{m+1}}$ would not be solvable, which cannot be the case. Therefore, $ab^{x_1} \dots ab^{x_m} ab^{x_{m+1}}$ must be a prefix of a cyclic solvable word, and then $b^{e-1} ab^{x_1} \dots ab^{x_m} ab^{x_{m+1}}$ is also a prefix of a cyclic solvable word.

Then, the only remaining case to check corresponds to $x_0 = e$, but then $w' = b^{x_0} \dots ab^{x_m} ab^{x_{m+1}}$ can be extended to a cyclic solvable word iff $aw = w' = ab^{x_0} \dots ab^{x_m} ab^{x_{m+1}}$ can be extended. But if this would not be the case, since w is assumed to be solvable, there should exist a decomposition disproving its solvability with $i = 0$. This corresponds to the case $y = a$. But then, by Lemma 2(2), we conclude again that w is unsolvable. \square

4.1 An Efficient Algorithm to Detect Solvable Words

Having established that the extensions of prefixes of cyclic solvable words with an additional prefix of the form x^k are the only solvable words, we can now verify solvability. For this purpose we use the constructive proof of Theorem 8, combined with Theorem 2, that states that derivation both preserves and reflects cyclic solvability.

Theorem 9 (efficient recursive algorithm to check solvability). *The following PROLOG-like algorithm with clauses applied in indicated order checks in linear time the solvability of w .*

- *if* $w = x^k y w'$ and $k > 0$
 then $Solvable(w) = PrefixCyclicSolvable(y w')$
- *if* $w = \alpha x \beta y \gamma$ and $x \neq y$
 then $\neg PrefixCyclicSolvable(w)$
- *if* $w = x^{k_1} y^j x^{k_2}$, for $k_1, k_2 \in \mathbb{N}$ and $j \in \{0, 1\}$
 then $PrefixCyclicSolvable(w)$

Now $w = b^{x_0} a b^{x_1} a b^{x_2} \dots a b^{x_m} a b^{x_{m+1}}$, where $m \geq 1$ and $x_1, \dots, x_m \geq 1$ and $x_0, x_{m+1} \in \mathbb{N}$.

- *if* $x_i > x_j + 1$, for some $j \in \{1, \dots, m\}$ and $i \in \{0, \dots, m+1\}$
 then $\neg PrefixCyclicSolvable(w)$
- *if* $x_1 = \dots = x_m$
 then $PrefixCyclicSolvable(w)$
- *if* $x_0 < x_j$, for some $j \in \{1, \dots, m\}$
 then $PrefixCyclicSolvable(w) = PrefixCyclicSolvable(ab^{x_1} \dots ab^{x_{m+1}})$
 else $PrefixCyclicSolvable(w) = PrefixCyclicSolvable(ab^{x_0} \dots ab^{x_{m+1}})$
- *if* $x_{m+1} < x_j$, for some $j \in \{1, \dots, m\}$
 then $PrefixCyclicSolvable(w) = PrefixCyclicSolvable(ab^{x_1} \dots ab^{x_m} a)$
 else $PrefixCyclicSolvable(w) = PrefixCyclicSolvable(ab^{x_1} \dots ab^{x_{m+1}} a)$
- *if* $w = (ab^e)^f (ab^{e-1} ab^e)^k (ab^{e-1})^g$, for some $f, g \in \{0, 1\}$ and $k \in \mathbb{N}$
 then $PrefixCyclicSolvable(w)$
 else $PrefixCyclicSolvable(w) = PrefixCyclicSolvable(\partial(w))$

Proof. Follows immediately from Theorems 2 and 8. Applying the latter we know that whenever the algorithm declares w not extendable to a cyclic solvable word, because its derivative was proved not to be such, the decision is sound. And if the algorithm, after some derivatives, finally reaches some trivial prefix of a cyclic solvable word, then the application of the former theorem proves that after the corresponding number of integrations, we would obtain a cyclic solvable word that extends the considered word w .

The algorithm terminates in linear time, since $|w| \geq 2 \cdot |\partial w|$. □

Let us consider $w = b^8(ab)^2 ab^2(ab)^4 ab^2(ab)^3 ab^2(ab)^4 ab^2(ab)^4 ab^2 ab$. After removing the prefix b^8 , we obtain w' with $\partial(w') = 1^2 21^4 21^3 21^4 21^4 21$, which once the short prefix and suffix are removed, can be translated to $w'' = ab^4 ab^3 ab^4 ab^4 a$.

Now we have $\partial(w'') = 121^2$, which produces the simple word ab^2 , which clearly is a prefix of a cyclic solvable word. Thus w is solvable.

One can show in an ‘effective’ way that the analysis was correct by constructing the primitives of the involved words, that will be cyclic solvable words extending them (see Fig. 2). In particular, we see first that 21^2 proves that $\partial(w'')$ is a prefix of $(1)(21^2)(21)$, while the first integration produces the cyclic solvable word $cw'' = ab^4ab^3ab^4$, by means of which we see that w'' is a prefix of $cw'' \cdot cw''$. And finally, a new integration produces the cyclic solvable word $cw' = (ab)^2ab^2(ab)^4ab^2(ab)^3ab^2(ab)^2$, by means of which we see that taking $w = b^8w'$, w' is a prefix of $cw' \cdot cw'$.

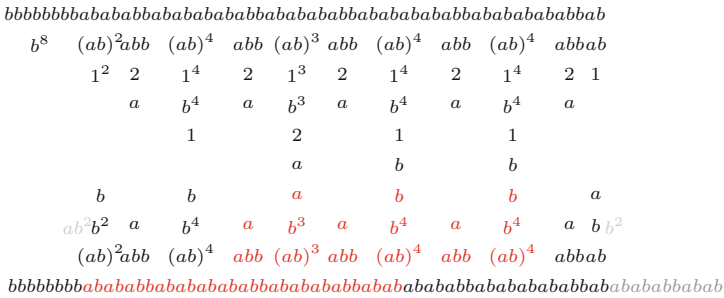


Fig. 2. Procedure of checking solvability and deriving a circular word for suffix.

5 Characterization of Reversible Binary Words

We start by recalling the definitions of strict reverses and reversible words. A (strict) reverse of a transition $t \in T$ in a net $N = (P, T, F, M_0)$ is a new transition \bar{t} such that $F(p, \bar{t}) = F(t, p)$ and $F(\bar{t}, p) = F(p, t)$. Following [3], we now relate the reversibility of all transitions with the solvability of the reversed sequence.

A binary word $w = t_1 \dots t_n$ is reversible if the following flts is solvable:

$$\overline{TS}_w = (\{0, \dots, n\}, \{a, b\}, \{(i - 1, t_i, i), (i, \bar{t}_i, i - 1) \mid 0 < i \leq n\}, 0).$$

Fact 5. A solvable binary word w is reversible iff w^{rev} is solvable.

We continue by showing that any cyclic solvable word is reversible, and so any prefix of a cyclic solvable word is also reversible.

Proposition 9. A binary word w is cyclic solvable iff w^{rev} is cyclic solvable.

Proof. By Fact 3, any cyclic solvable word w can be generated by a net N_{AB} , with $A = |w|_a, B = |w|_b$. One can show that by reversing the arrows in N_{AB} , and starting from the same initial marking, we generate exactly w^{rev} , recovering again the initial marking, and thus proving that w^{rev} is cyclic solvable. \square

Corollary 3. *Each cyclic solvable binary word is reversible.*

Proof. Follows immediately from Fact 5 and Proposition 9. □

We can further illustrate the last result by showing that in every $N_{AB} = (\{p_a, p_b\}, \{a, b\}, F, M_0) \in \mathcal{N}_{AB}$ one can introduce strict reverses without enlarging the set of reachable markings. We define $\overline{N}_{AB} = (\{p_a, p_b\}, \{a, \bar{a}, b, \bar{b}\}, \overline{F}, M_0)$, where the non-zero values of \overline{F} are $\overline{F}(p_a, a) = \overline{F}(a, p_b) = \overline{F}(p_b, \bar{a}) = \overline{F}(\bar{a}, p_a) = A$, and $\overline{F}(p_b, b) = \overline{F}(b, p_a) = \overline{F}(p_a, \bar{b}) = \overline{F}(\bar{b}, p_b) = B$. Similarly as in N_{AB} , every reachable marking M of \overline{N}_{AB} satisfies the same property as the initial marking, namely $M(p_a) + M(p_b) = A + B - 1$, and the net is reversible (i.e., from every marking one can reach the initial marking). As a consequence, in each reachable marking one can fire either a or b . Similarly, in each reachable marking, one can fire either \bar{a} or \bar{b} . Suppose we can fire \bar{a} at M . Then we cannot fire \bar{b} , hence we go from M_0 to M , firing a as the last transition. The case of the initial marking is treated similarly, thanks to the reversibility of N_{AB} .

The result above leads to a characterization of reversible words.

Proposition 10. *A binary word $w = a^k b \alpha a b^m$ is reversible iff both $w_s = b \alpha a b^m$ and $w_p = a^k b \alpha a$ are prefixes of cyclic solvable words.*

Proof. In order to be reversible, w must be solvable, and so w_s must be a prefix of a cyclic solvable word. Moreover, w^{rev} must also be solvable, so that w_p^{rev} must be a prefix of a cyclic solvable word. Then, by applying Proposition 9, w_p must be also a prefix of a cyclic solvable word. □

Corollary 4. *One can decide in linear time whether a binary flts can be reversed.*

Proof. Given a word $w = a^k b \alpha a b^m$, we can apply the algorithm in Theorem 9 to both $a^k b \alpha a$ and $b \alpha a b^m$.

Note: The result could also be obtained directly, by applying the characterization from Proposition 10, and so applying the algorithm from Theorem 9 to both $a^k b \alpha a$ and $b^m a \alpha^{rev} b$. □

It seems that the above characterization gives us some space to find reversible words that are not prefixes of cycle solvable words, but this can only happen in very few simple cases.

Theorem 10. *Apart from prefixes of cyclic solvable binary words, the only reversible words are those of the form $a^k (ba)^i b^m$ or $b^k (ab)^i a^m$, where $i \in \{0, 1\}$ or $k = 2 = m$.*

Proof. First, it is clear that all the ‘exceptions’ in the statement are indeed solvable and reversible. Moreover one can see that they must appear indeed as exceptional cases, since they do not correspond to prefixes of cyclic solvable words, out of a few trivial instances.

Now we concentrate on the words $w = x^n yuzt^m$ with $x \neq y, z \neq t \in \{a, b\}$, such that $w' = yuz$ contains either aa or bb . We assume the latter. Using the

results about the decomposition of solvable words, we know that all the blocks ab^i , and reversed blocks $b^i a$ in w' , must be either ab^{e-1} or ab^e (resp. $b^{e-1}a$ or $b^e a$), for some $e \geq 2$. Moreover, if $x = a$ then $n = 1$, and when $t = a$ then $m = 1$. While when $x = b$, the cases in which $m < e$ will immediately generate a prefix of a cyclic solvable word, since $w_p = x^n w'$ is such a word; and, similarly, when $t = b$. Hence, we can concentrate in the following on the case $m = e$. Next we distinguish the remaining cases, showing that they never produce reversible words which are not prefixes of cyclic solvable words:

- If $x = b$, then aw_p is a prefix of a cyclic solvable word, and then aw remains solvable. Now, using our characterization of solvable words, w must be a prefix of a cyclic solvable word. The case $t = b$ can be dealt with in a similar (symmetric) way.
- If $x = a$ and $t = a$, we consider the contiguous blocks of b 's, b^{i_x} and b^{i_y} .
 - If $i_x = e$ (resp. $i_y = e$), then the cyclic word containing $w_s = w' y^m$ (resp. w_p) clearly contains one a before (resp. after) its occurrence, and so w itself is contained in that cyclic word.
 - If $i_x = e - 1$ and $i_y = e - 1$ and w is not a prefix of a cyclic solvable word, then w contains two mutually unbalanced contiguous block-subwords that totally cover w_p , since otherwise they would also prove that either w_s or w_p would not be the prefix of some cyclic solvable word, contradicting our assumption. But since the two blocks at the end of these block-subwords will be the same (ab^{e-1}), we could remove them getting two new unbalanced subwords that now do not cover w_p , something that we have already shown to be impossible.

As a result, if w is not a prefix of a cyclic solvable word, then $w' = yuz$ contains neither aa nor bb . Hence it is of the form $(ab)^i$ or $(ba)^i$. □

6 Conclusions

In this paper, we discussed three classes of binary words from the viewpoint of the solvability of the transition systems related to them. For each class, we described a linear algorithm which verifying its membership. Based on our results, one can show every cyclic solvable word is reversible, and every reversible word is solvable (moreover, the two implications cannot be reversed).

A natural direction for further research is to consider larger alphabets, and more sophisticated flts's, for example, those having the shape of directed rooted trees.

Acknowledgement. This research was supported by Cost Action IC1405. The first author was partially supported by the Spanish projects TRACES (TIN2015-67522-C3-3) and N-GREENS (S2013/ICE-2731).

References

1. Badouel, E., Bernardinello, L., Darondeau, P.: Petri Net Synthesis. Texts in Theoretical Computer Science. An EATCS Series. Springer, Heidelberg (2015). <https://doi.org/10.1007/978-3-662-47967-4>
2. Barylska, K., Best, E., Erofeev, E., Mikulski, L., Piątkowski, M.: Conditions for Petri net solvable binary words. In: Koutny, M., Desel, J., Kleijn, J. (eds.) Transactions on Petri Nets and Other Models of Concurrency XI. LNCS, vol. 9930, pp. 137–159. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53401-4_7
3. Barylska, K., Erofeev, E., Koutny, M., Mikulski, L., Piątkowski, M.: Reversing transitions in bounded Petri nets. *Fundam. Inform.* **157**(1–4), 341–357 (2018)
4. Barylska, K., Koutny, M., Mikulski, L., Piątkowski, M.: Reversible computation vs. reversibility in Petri nets. *Sci. Comput. Program.* **151**, 48–60 (2018)
5. Best, E., Erofeev, E., Schlachter, U., Wimmel, H.: Characterising Petri net solvable binary words. In: Kordon, F., Moldt, D. (eds.) PETRI NETS 2016. LNCS, vol. 9698, pp. 39–58. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39086-4_4
6. Erofeev, E., Barylska, K., Mikulski, L., Piątkowski, M.: Generating all minimal Petri net unsolvable binary words. In: Proceedings of PSC 2016, pp. 33–46 (2016)