Tero Harju | Chang Li | Ion Petre | Grzegorz Rozenberg

Parallelism in gene assembly

Turku Centre *for* Computer Science

TUCS Technical Report
No 611, July 2004

Parallelism in gene assembly

Tero Harju
Department of Mathematics, University of Turku,
Turku Center for Computer Science,
FIN-20014 Turku, Finland,
email: harju@utu.fi

Chang Li
Department of Computer Science, Åbo Akademi University,
Turku Center for Computer Science,
FIN-20520 Turku, Finland,
email: lchang@abo.fi

Ion Petre
Department of Computer Science, Åbo Akademi University,
Turku Center for Computer Science,
FIN-20520 Turku, Finland,
email: ipetre@abo.fi

Grzegorz Rozenberg
Leiden Institute for Advanced Computer Science,
Leiden University, 2333 CA Leiden, the Netherlands, and
Department of Computer Science, University of Colorado,
Boulder, Co 80309-0347, USA,
email: rozenber@liacs.nl

TUCS Technical Report

No 611, July 2004

Abstract

The process of gene assembly in ciliates, an ancient group of organisms, is one of the most complex instances of DNA manipulation known in any organisms. This process is fascinating from the computational point of view, with ciliates even using the linked lists data structure. Three molecular operations (ld, hi, and dlad) have been postulated for the gene assembly process. We initiate here the study of parallelism in this process, raising several natural questions, such as: when can a number of operations be applied in parallel to a gene pattern; or how many steps are needed to assemble (in parallel) a micronuclear gene. In particular, this gives rise to a new measure of complexity for the process of gene assembly in ciliates.

> "One of the oldest forms of life on Earth has been
> revealed as a natural born computer programmer."
> BBC, September 10, 2001.

TUCS Laboratory
Discrete Mathematics for Information Technology

1 Introduction

The ciliates (ciliated protozoa) are an ancient and diverse group of unicellular organisms. Their diversity can be appreciated by comparing their genomic sequences – some ciliate types differ genetically more than humans differ from fruit flies! A unique feature of the ciliates is their nuclear dualism: each ciliate possesses two kinds of nuclei in the same cell: a *micronucleus* and a *macronucleus*, see [11], [12], and [13]. The micronucleus is a germline nucleus and has no known function in the growth or the division of the cell. The RNA transcripts are provided by the macronucleus – the somatic nucleus. The two types of nuclei are however interrelated: at some stage, in the process of sexual reproduction, the genome of the micronucleus develops into the genome of the macronucleus, in a process called *gene assembly*. What makes this process unusual is the sophisticated rearrangement that a family of ciliates, the *Stichotrichs*, have engineered in the DNA sequence of their micronuclear genome. Thus, while genes in the macronucleus are contiguous sequences, placed (with very few exceptions) on their own short DNA molecules, the DNA in the micronucleus is organized in long molecules, with genes occurring individually or in groups, separated by long stretches of non-coding DNA. Moreover, the genes in the micronucleus are broken into pieces called *MDSs*, separated by non-coding segments called *IESs*. During gene assembly, the IESs are excised and MDSs are ligated to form transcriptionally competent macronuclear genes. The complexity of this process is best illustrated in *Stichotrichs* ciliates (which we consider in this paper), where the MDSs may be scrambled, i.e., the sequence of MDSs is permuted in the micronucleus, with some MDSs being inverted.

The gene assembly process is highly interesting from the computational point of view. One of the amazing features of this process is that ciliates apparently know *linked lists* and use them in an elegant pattern matching mechanism.

Three molecular operations, ld, hi, and dlad, have been postulated in [8] and [14] for the gene assembly process – they were successfully used to give a uniform explanation to all known experimental data. The gene structure and the operations themselves have been modelled and formally investigated on three levels of abstraction based on permutations, strings, and graphs see [1], [6], [8], and [14]. A detailed discussion on the methodology of model forming can be found in [4]. This line of research has already answered a number of natural questions, such as the assembly power of these operations, invariants of the gene assembly, or micronuclear gene patterns that can be assembled using a subset of operations, see, e.g. [2], [3], [5], and [7]. We refer to the recent monograph [4] for details and further topics in this research area.

In our research so far, the process of gene assembly has been mostly considered as a *sequence* of folding and recombination operations. While this approach was adequate for the type of research questions that have been considered, in order to gain more insight into the gene assembly process, a more general *parallel* application of molecular operations must be investigated – *parallelism* is a natural phenomenon in biomolecular processes. In this paper we initiate a systematic study of parallelism in our model for gene assembly. Intuitively, a number of operations can be applied in parallel to a gene pattern if each operation's applicability is independent of the other's. In other words, a set of operations can be applied in parallel to a gene pattern if and only if they can be (sequentially) applied to that pattern in any order – this is consistent with how *concurrency* and *parallelism* are usually defined in Computer Science.

Our notion of parallelism naturally leads to a new measure of complexity for the gene assembly process, given by the minimal number of steps required to assemble a gene in parallel. E.g., micronuclear genes having the MDSs in the orthodox order, such as *C2* and βTP in *S. nova*, should be intuitively equally easy to assemble (if the number of MDSs only differs slightly). This is indeed the case, as we discuss in this paper: the signed graphs associated to such genes can be reduced to the empty graph (the abstraction of the completion of the gene assembly process for signed graphs) in one parallel step. This clearly leads to another question: how many steps are needed in general to reduce a signed graph (or to assemble a gene pattern)? We conjecture a stunning answer to this question: any negative graph can be assembled in parallel in at most two steps! Note however that we assume here maximal parallelism: any operation that can be applied in a given step of the reduction must be applied at that stage. Whether or not ciliates actually operate in this way is clearly a different question that can be answered only through well-designed laboratory experiments. Nevertheless, one can assume the hypothesis of maximal parallelism, as we do here, without loss of generality: if a number of operations is applicable at some stage of the reduction, then these operations will remain applicable throughout the reduction.

2 Molecular operations for gene assembly

Three molecular operations were postulated in [8] and [14] for the gene assembly in ciliates. We only show here in Figs. 1-3 the foldings required by each operation and the recombinations that take place in each case. We refer to [4] for a detailed discussion.

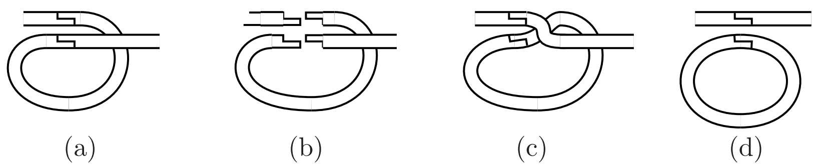

Figure 1: Illustration of the ld molecular operation.

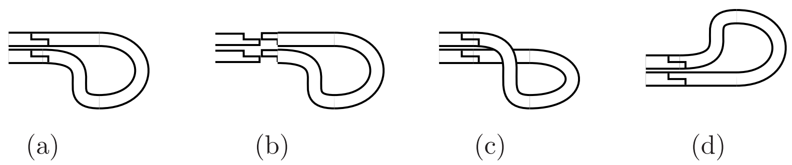

Figure 2: Illustration of the hi molecular operation.

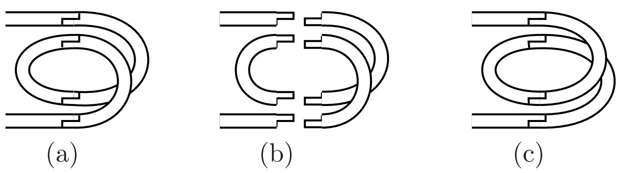

Figure 3: Illustration of the dlad molecular operation.

The central role in gene assembly is played by characteristic short sequences at the ends of MDSs, called *pointers* – the pointer in the end of an MDS M coincides (as a nucleotide sequence) with the pointer in the beginning of the MDS succeeding M in the macronuclear gene. Each micronuclear gene and its intermediary successors in the gene assembly process can be thus described by *signed permutations* (denoting the sequence and the orientation of the MDSs), *signed double occurrence strings* (denoting the sequence and the orientation of the pointers), and *signed graphs* (denoting the overlap of the pointers). Surprisingly enough, it has been proved in [1] and [6] that the information given by the overlap relations among pointers is sufficient for analyzing the whole process of gene assembly. We refer to [1], [4], and [6] for all details concerning the various levels of abstraction and for the methodology of model forming. We focus in this paper on the string and on the graph levels, and formalize the notion of parallelism in these two frameworks.

3 String reduction rules for gene assembly

We recall in this section some basic definitions related to signed double occurrence strings – we refer to [4] for more details.

Let $\Sigma = \{a_1, a_2, \ldots\}$ be a set of symbols. The set of all strings over the alphabet Σ is denoted by Σ^*. If $\overline{\Sigma} = \{\overline{a} \mid a \in \Sigma\}$ is a signed copy of Σ, $\Sigma \cap \overline{\Sigma} = \emptyset$, then $\Sigma \cup \overline{\Sigma}$ is a *signed alphabet* and the set of all strings over $\Sigma \cup \overline{\Sigma}$ is denoted by $\Sigma^{\maltese} = (\Sigma \cup \overline{\Sigma})^*$. A string $v \in \Sigma^{\maltese}$ is called a *signed string over* Σ.

For two strings $v, w \in \Sigma^{\maltese}$, we say that v is a *substring* of u if $u = w_1 v w_2$, for some strings $w_1, w_2 \in \Sigma^{\maltese}$.

Let $v \in \Sigma^{\maltese}$ be a signed string over Σ. We say that a letter $a \in \Sigma \cup \overline{\Sigma}$ *occurs* in v, if a or $\overline{a}$ is a substring of v. Let $dom(v) \subseteq \Sigma$, called the *domain* of v, be the set of (unsigned) letters that occur in v.

We say that a string $v \in \Sigma^{\maltese}$ is a *signed double occurrence string*, if every letter $a \in dom(v)$ occurs exactly twice in v. In this case, we also say that v is a *legal string.* For $a \in \Sigma \cup \overline{\Sigma}$, if v contains both substring a and $\overline{a}$, then a is *positive* in v; otherwise, a (or $\overline{a}$) is *negative* in v.

Example 1. The signed string $u = 345\overline{2}3\overline{5}24$ over $\{3, 4, 5\}$ is legal. Pointers 2 and 5 are positive in u, while 3 and 4 are negative in u. On the other hand, the string $w = 345\overline{25}24$ is not legal, since 3 has only one occurrence in w.

Let $u = a_1 a_2 \ldots a_n \in \Sigma^{\maltese}$ be a legal string over Σ, where $a_i \in \Sigma \cup \overline{\Sigma}$ for each i. For each $a \in dom(u)$, there are indices i and j with $1 \leq i < j \leq n$ such that $||a_i|| = a_i = ||a_j||$. The substring $u_{(a)} = a_i a_{i+1} \ldots a_j$ is the a*−interval* of u. Two different letters $a, b \in dom(u)$ are said to *overlap* in u if the a-interval and the b-interval of u overlap, i.e., if $u_{(a)} = a_{i_1} \ldots a_{j_1}$ and $u_{(b)} = a_{i_2} \ldots a_{j_2}$, then either $i_1 < i_2 < j_1 < j_2$ or $i_2 < i_1 < j_2 < j_1$.

Example 2. The string $u = 24353\overline{265}7467$ is legal. The 2-interval of u is the substring $u_{(2)} = 24353\overline{2}$. Also, $u_{(3)} = 353$, $u_{(4)} = 4353\overline{265}74$, $u_{(5)} = 53\overline{265}$, $u_{(4,5)} = 4353\overline{265}74$, and $u_{(5,6)} = 53\overline{265}746$. Thus, pointer 7 overlaps with pointers 4 and 6, and pointer 3 overlaps with pointer 5. On the other hand, pointer 2 does not overlap with pointer 7.

As noted above, the essential information to trace the gene assembly process lays in the sequence of pointers of the considered gene. Thus, the MDS structure of genes can be represented only by the sequence of its pointers. Each MDS M_i, $1 < i < k$, can be represented as the string $i\,(i+1)$, and the inverse MDS $\overline{M_i}$

can be represented as the string $\overline{i+1}\,\overline{i}$. The first MDS M_1 and the last one, M_k are special: M_1 is represented as 2, M_k is represented as the string k, while $\overline{M_1}$ is represented as $\overline{2}$ and $\overline{M_k}$ is represented as $\overline{k}$. Clearly, this associates a unique legal string to each MDS sequence. We refer to [4] and [10] for formal details on this abstraction.

Example 3. (a) The legal string corresponding to the *actin I* gene in *S.nova*, see Fig. 4 for its MDS sequence, is $3\,4\,4\,5\,6\,7\,5\,6\,7\,8\,9\,\overline{3}\,\overline{2}\,2\,8\,9$.

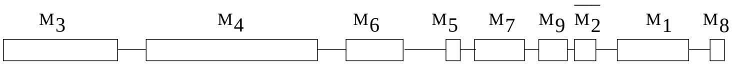

Figure 4: The MDS sequence of the micronuclear gene encoding the *actin I* protein in *S.nova*

(b) The legal string corresponding to the αTP gene in *S.nova*, see Fig. 5 for its MDS sequence, is $2\,3\,4\,5\,6\,7\,8\,9\,10\,11\,12\,2\,3\,4\,5\,6\,7\,8\,9\,10\,11\,12\,13\,13\,14\,14$.

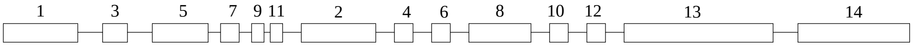

Figure 5: Structure of the micronuclear gene encoding αTP protein in *S. nova*.

The three molecular operations ld, hi, and dlad, conjectured in [8] to carry the gene assembly in ciliates can be formalized in the framework of legal strings through the string rewriting rules snr, spr and sdr, resp., defined below. In the following, let $k \geq 2$, $\Delta_k = \{2, 3, \ldots, k\}$, and $\Pi_k = \Delta_k \cup \overline{\Delta}_k$.

- The *string negative rule* snr_p for a pointer $p \in \Pi_k (k \geq 2)$ is applicable to a legal string u of the form $u = u_1 pp u_2$, where $u_1, u_2 \in \Delta_k^{\maltese}$ and
$$\mathsf{snr}_p(u_1 pp u_2) = u_1 u_2.$$
Let $\mathsf{Snr} = \{\mathsf{snr}_p \mid p \in \Pi_k,\ k \geq 2\}$.

- The *string positive rule* spr_p for a pointer $p \in \Pi_k (k \geq 2)$ is applicable to a legal string u of the form $u = u_1 p u_2 \overline{p} u_3$, where $u_1, u_2, u_3 \in \Delta_k^{\maltese}$ and
$$\mathsf{spr}_p(u_1 p u_2 \overline{p} u_3) = u_1 \overline{u_2} u_3.$$
Let $\mathsf{Spr} = \{\mathsf{spr}_p \mid p \in \Pi_k,\ k \geq 2\}$.

- The *string double rule* $\mathsf{sdr}_{p,q}$ for pointers $p, q \in \Pi_k (k \geq 2)$ is applicable to a legal string u of the form $u = u_1 p u_2 q u_3 p u_4 q u_5$, where $u_i \in \Delta_k^{\maltese}$ for each i, and
$$\mathsf{sdr}_{p,q}(u) = u_1 u_4 u_3 u_2 u_5.$$
Let $\mathsf{Sdr} = \{\mathsf{sdr}_{p,q} \mid p, q \in \Pi_k,\ k \geq 2\}$.

A composition $\varphi = \varphi_n \ldots \varphi_1$ of operations from $\mathsf{Snr} \cup \mathsf{Spr} \cup \mathsf{Sdr}$ is a *string reduction* of u, if φ is applicable to u. Also, φ is a *successful reduction* for u if $\varphi(u)$ is the empty string Λ.

Example 4. (i) Consider the legal string $u = 233\overline{4}\overline{2}545$. Then snr_3 is applicable to u and $\mathsf{snr}_3(u) = 2\overline{4}\overline{2}545$. Also, spr_4 is applicable to u and $\mathsf{spr}_4(u) = 233\overline{5}25$. Similarly, $\mathsf{spr}_2(u) = 4\overline{3}\overline{3}545$. Two successful reductions of u are the following:

$$(\mathsf{spr}_2 \circ \mathsf{spr}_5 \circ \mathsf{spr}_4 \circ \mathsf{snr}_3)(u) = (\mathsf{spr}_2 \circ \mathsf{spr}_5 \circ \mathsf{spr}_4)(2\overline{42}545) =$$
$$= (\mathsf{spr}_2 \circ \mathsf{spr}_5)(2\overline{5}25) = (\mathsf{spr}_2)(2\overline{2}) = \Lambda$$

and

$$(\mathsf{sdr}_{4,5} \circ \mathsf{spr}_2 \circ \mathsf{snr}_3)(u) = (\mathsf{sdr}_{4,5} \circ \mathsf{spr}_2)(2\overline{42}545) = (\mathsf{sdr}_{4,5})(4545) = \Lambda.$$

(ii) Let $w = 34\overline{3}2\overline{5}65426$ be a legal string. Then $\mathsf{spr}_3(w) = \overline{4}2\overline{5}65426$ and $\mathsf{sdr}_{4,6}(w) = 325\overline{3}2\overline{5}$. Two successful reductions of w are the following:

$$(\mathsf{snr}_6 \circ \mathsf{spr}_5 \circ \mathsf{spr}_2 \circ \mathsf{spr}_4 \circ \mathsf{spr}_3)(w) = (\mathsf{snr}_6 \circ \mathsf{spr}_5 \circ \mathsf{spr}_2 \circ \mathsf{spr}_4)(\overline{4}2\overline{5}65426) =$$
$$= (\mathsf{snr}_6 \circ \mathsf{spr}_5 \circ \mathsf{spr}_2)(\overline{56}5\overline{2}26) = (\mathsf{snr}_6 \circ \mathsf{spr}_5)(\overline{56}56) = (\mathsf{snr}_6)(66) = \Lambda$$

and

$$(\mathsf{snr}_{\overline{5}} \circ \mathsf{spr}_2 \circ \mathsf{spr}_3 \circ \mathsf{sdr}_{4,6})(w) = (\mathsf{snr}_{\overline{5}} \circ \mathsf{spr}_2 \circ \mathsf{spr}_3)(325\overline{3}2\overline{5}) =$$
$$= (\mathsf{snr}_{\overline{5}} \circ \mathsf{spr}_2)(\overline{52}2\overline{5}) = (\mathsf{snr}_{\overline{5}})(\overline{55}) = \Lambda.$$

4 Parallelism in the string-based model

We consider in this section the notion of parallelism in gene assembly, seen here in the framework of signed double occurrence (i.e., legal) strings.

Definition 1. *Let $S \subseteq \mathsf{Snr} \cup \mathsf{Spr} \cup \mathsf{Sdr}$ be a set of rules and let u be a legal string. We say that the rules in S can be applied in parallel to u if for any ordering $\varphi_1, \varphi_2, \ldots, \varphi_k$ of S, the composition $\varphi_k \circ \cdots \circ \varphi_1$ is applicable to u. In particular, two rules $\varphi, \psi \in \mathsf{Snr} \cup \mathsf{Spr} \cup \mathsf{Sdr}$ can be applied in parallel to u if both $\varphi \circ \psi$ and $\psi \circ \varphi$ are applicable to u.*

We consider the following question: given two reduction rules and a legal string u, can those rules be applied in parallel to u? As it turns out, the answer is straightforward unless we have two sdr rules.

Let $u = a_1 a_2 \ldots a_n \in \Sigma^{\maltese}$ be a legal string over Σ, where $a_i \in \Sigma \cup \overline{\Sigma}$ for each i, and let $\varphi, \psi \in \mathsf{Snr} \cup \mathsf{Spr} \cup \mathsf{Sdr}$ be two rules applicable to u. We consider in the following all possible cases when φ and ψ are applied in parallel to u.

Applying two snr rules in parallel If $\varphi = \mathsf{snr}_a$ and $\psi = \mathsf{snr}_b$, for $a, b \in dom(u)$, then $\{\varphi, \psi\}$ is applicable to u in parallel. Also, $(\varphi \circ \psi)(u) = (\psi \circ \varphi)(u)$.

Applying snr and spr in parallel If $\varphi = \mathsf{snr}_a$ and $\psi = \mathsf{spr}_b$, then $\{\varphi, \psi\}$ is applicable to u in parallel if and only if $u_{(a)}$ is not a substring of $u_{(b)}$. In this case, $(\varphi \circ \psi)(u) = (\psi \circ \varphi)(u)$.

Example 5. (i) For $u = baa\overline{b}$, $\mathsf{spr}_b \circ \mathsf{snr}_a$ is applicable to u, but $\mathsf{snr}_a \circ \mathsf{spr}_b$ is not. Indeed, snr_a is not applicable to $\mathsf{spr}_b(u) = \overline{aa}$.

(ii) For $\nu = aab\overline{b}$, $\{\varphi, \psi\}$ is applicable to ν in parallel and $(\mathsf{spr}_b \circ \mathsf{snr}_a)(\nu) = (\mathsf{snr}_a \circ \mathsf{spr}_b)(\nu)$.

Applying snr and sdr in parallel If $\varphi = \mathsf{snr}_a$ and $\psi = \mathsf{sdr}_{b,c}$, $a, b, c \in dom(u)$, then $\{\varphi, \psi\}$ is applicable to u in parallel. Also, $(\varphi \circ \psi)(u) = (\psi \circ \varphi)(u)$.

Applying two spr rules in parallel If $\varphi = \mathsf{spr}_a$ and $\psi = \mathsf{spr}_b$, $\{\varphi, \psi\}$ is applicable to u in parallel if and only if a and b do not overlap in u. In this case, $(\varphi \circ \psi)(u) = (\psi \circ \varphi)(u)$.

Example 6. (i) For $\gamma = a\bar{a}b\bar{b}$, $\{\varphi, \psi\}$ is applicable to γ in parallel. Indeed, $(\mathsf{spr}_a \circ \mathsf{spr}_b)(\gamma) = (\mathsf{spr}_b \circ \mathsf{spr}_a)(\gamma)$.

(ii) For $\xi = ab\bar{b}\bar{a}$, $\{\varphi, \psi\}$ is applicable to ξ in parallel, although $\xi_{(b)}$ is a substring of $\xi_{(a)}$. Moreover, $(\mathsf{spr}_b \circ \mathsf{spr}_a)(\xi) = (\mathsf{spr}_b)(b\bar{b}) = \Lambda$, and $(\mathsf{spr}_a \circ \mathsf{spr}_b)(\xi) = (\mathsf{spr}_a)(a\bar{a}) = \Lambda$.

(iii)For $\omega = ab\bar{a}\bar{b}$, neither $\mathsf{spr}_a \circ \mathsf{spr}_b$ nor $\mathsf{spr}_b \circ \mathsf{spr}_a$ is applicable to ω.

Applying spr and sdr in parallel If $\varphi = \mathsf{spr}_a$ and $\psi = \mathsf{sdr}_{(b,c)}$, then $\{\varphi, \psi\}$ is applicable to u in parallel if and only if a does not overlap with b or c in u, also neither $u_{(b)}$ nor $u_{(c)}$ is a substring of $u_{(a)}$. In this case, $(\varphi \circ \psi)(u) = (\psi \circ \varphi)(u)$.

Example 7. (i) For $\delta = ba\bar{a}cbc$, $\{\varphi, \psi\}$ is applicable to δ in parallel. Indeed, $(\mathsf{sdr}_{b,c} \circ \mathsf{spr}_a)(\delta) = (\mathsf{spr}_a \circ \mathsf{sdr}_{b,c})(\delta) = \Lambda$.

(ii) For $\zeta = abcbc\bar{a}$, $\mathsf{spr}_a \circ \mathsf{sdr}_{b,c}$ is applicable to ζ, but $\mathsf{sdr}_{b,c} \circ \mathsf{spr}_a$ is not. Indeed, $\mathsf{sdr}_{b,c}$ is not applicable to $\mathsf{spr}_a(\zeta) = \overline{cbcb}$. Note that $\zeta_{(b)}$ and $\zeta_{(c)}$ are substrings of $\zeta_{(a)}$.

(iii) For $\kappa = abc\bar{a}bc$, $\mathsf{spr}_a \circ \mathsf{sdr}_{b,c}$ is applicable to κ, but $\mathsf{sdr}_{b,c} \circ \mathsf{spr}_a$ is not. Indeed, $\mathsf{sdr}_{b,c}$ is not applicable to $\mathsf{spr}_a(\kappa) = \overline{cb}bc$. Note that a overlaps with b and c in κ

(iv) For $\theta = ab\bar{a}cbc$, $\mathsf{spr}_a \circ \mathsf{sdr}_{b,c}$ is applicable to θ, but $\mathsf{sdr}_{b,c} \circ \mathsf{spr}_a$ is not. Indeed, $\mathsf{sdr}_{b,c}$ is not applicable to $\mathsf{spr}_a(\theta) = \bar{b}cbc$. Note that a overlaps with b in θ.

Applying two sdr rules in parallel To simplify the notation, in the following, we use three dots to denote an arbitrarily long string (possibly empty). Thus, e.g., if u is of the form $u = u_1au_2bu_3au_4bu_5cu_6du_7cu_8du_9$, for some $u_i \in \Sigma^{\maltese}$, $1 \leq i \leq 9$, then we denote this as:

$$u = \ldots a \ldots b \ldots a \ldots b \ldots c \ldots d \ldots c \ldots d \ldots .$$

Let $\varphi = \mathsf{sdr}_{a,b}$ and $\psi = \mathsf{sdr}_{c,d}$, where $a, b, c, d \in dom(u)$. Since both φ and ψ are applicable to u, u can only have one of the forms listed below (here, we assume without loss of generality that the pointer from the set $\{\ a, b, c, d\ \}$ which occurs first in u is a). We study in each case whether or not φ and ψ can be applied in parallel to u. We recall that to check this, one needs to verify that both $f = \mathsf{sdr}_{a,b} \circ \mathsf{sdr}_{c,d}$ and $g = \mathsf{sdr}_{c,d} \circ \mathsf{sdr}_{a,b}$ are applicable to u.

- If $u = \ldots a \ldots b \ldots a \ldots b \ldots c \ldots d \ldots c \ldots d \ldots$, then $f(u) = g(u)$;
- if $u = \ldots a \ldots b \ldots a \ldots c \ldots b \ldots d \ldots c \ldots d \ldots$, then $f(u) = g(u)$;
- if $u = \ldots a \ldots b \ldots a \ldots c \ldots d \ldots b \ldots c \ldots d \ldots$, then $f(u) = g(u)$;
- if $u = \ldots a \ldots b \ldots a \ldots c \ldots d \ldots c \ldots b \ldots d \ldots$, then $f(u) = g(u)$;

- if $u = \ldots a \ldots b \ldots a \ldots c \ldots d \ldots c \ldots d \ldots b \ldots$, then $f(u) = g(u)$;
- if $u = \ldots a \ldots b \ldots c \ldots a \ldots b \ldots d \ldots c \ldots d \ldots$, then $f(u) = g(u)$;
- if $u = \ldots a \ldots b \ldots c \ldots a \ldots d \ldots b \ldots c \ldots d \ldots$, then neither f nor g is applicable to u;
- if $u = \ldots a \ldots b \ldots c \ldots a \ldots d \ldots c \ldots b \ldots d \ldots$, then neither f nor g is applicable to u;
- if $u = \ldots a \ldots b \ldots c \ldots a \ldots d \ldots c \ldots d \ldots b \ldots$, then g is not applicable to u;
- if $u = \ldots a \ldots b \ldots c \ldots d \ldots a \ldots b \ldots c \ldots d \ldots$, then $f(u) = g(u)$;
- if $u = \ldots a \ldots b \ldots c \ldots d \ldots a \ldots c \ldots b \ldots d \ldots$, then neither f nor g is applicable to u;
- if $u = \ldots a \ldots b \ldots c \ldots d \ldots a \ldots c \ldots d \ldots b \ldots$, then $f(u) = g(u)$;
- if $u = \ldots a \ldots b \ldots c \ldots d \ldots c \ldots a \ldots b \ldots d \ldots$, then $f(u) = g(u)$;
- if $u = \ldots a \ldots b \ldots c \ldots d \ldots c \ldots a \ldots d \ldots b \ldots$, then g is not applicable to u;
- if $u = \ldots a \ldots b \ldots c \ldots d \ldots c \ldots d \ldots a \ldots b \ldots$, then $f(u) = g(u)$;
- if $u = \ldots a \ldots c \ldots b \ldots a \ldots b \ldots d \ldots c \ldots d \ldots$, then $f(u) = g(u)$;
- if $u = \ldots a \ldots c \ldots b \ldots a \ldots d \ldots b \ldots c \ldots d \ldots$, then neither f nor g is applicable to u;
- if $u = \ldots a \ldots c \ldots b \ldots a \ldots d \ldots c \ldots b \ldots d \ldots$, then neither f nor g is applicable to u;
- if $u = \ldots a \ldots c \ldots b \ldots a \ldots d \ldots c \ldots d \ldots b \ldots$, then g is not applicable to u;
- if $u = \ldots a \ldots c \ldots b \ldots d \ldots a \ldots b \ldots c \ldots d \ldots$, then neither f nor g is applicable to u;
- if $u = \ldots a \ldots c \ldots b \ldots d \ldots a \ldots c \ldots b \ldots d \ldots$, then $f(u) = g(u)$;
- if $u = \ldots a \ldots c \ldots b \ldots d \ldots a \ldots c \ldots d \ldots b \ldots$, then neither f nor g is applicable to u;
- if $u = \ldots a \ldots c \ldots b \ldots d \ldots c \ldots a \ldots b \ldots d \ldots$, then neither f nor g is applicable to u;
- if $u = \ldots a \ldots c \ldots b \ldots d \ldots c \ldots a \ldots d \ldots b \ldots$, then neither f nor g is applicable to u;
- if $u = \ldots a \ldots c \ldots b \ldots d \ldots c \ldots d \ldots a \ldots b \ldots$, then g is not applicable to u;
- if $u = \ldots a \ldots c \ldots d \ldots b \ldots a \ldots b \ldots c \ldots d \ldots$, then $f(u) = g(u)$;
- if $u = \ldots a \ldots c \ldots d \ldots b \ldots a \ldots c \ldots b \ldots d \ldots$, then neither f nor g is applicable to u;
- if $u = \ldots a \ldots c \ldots d \ldots b \ldots a \ldots c \ldots d \ldots b \ldots$, then $f(u) = g(u)$;

- if $u = \ldots a \ldots c \ldots d \ldots b \ldots c \ldots a \ldots b \ldots d \ldots$, then neither f nor g is applicable to u;
- if $u = \ldots a \ldots c \ldots d \ldots b \ldots c \ldots a \ldots d \ldots b \ldots$, then neither f nor g is applicable to u;
- if $u = \ldots a \ldots c \ldots d \ldots b \ldots c \ldots d \ldots a \ldots b \ldots$, then $f(u) = g(u)$;
- if $u = \ldots a \ldots c \ldots d \ldots c \ldots b \ldots a \ldots b \ldots d \ldots$, then $f(u) = g(u)$;
- if $u = \ldots a \ldots c \ldots d \ldots c \ldots b \ldots a \ldots d \ldots b \ldots$, then g is not applicable to u;
- if $u = \ldots a \ldots c \ldots d \ldots c \ldots b \ldots d \ldots a \ldots b \ldots$, then g is not applicable to u;
- if $u = \ldots a \ldots c \ldots d \ldots c \ldots d \ldots b \ldots a \ldots b \ldots$, then $f(u) = g(u)$.

Consequently, two rules $\mathsf{sdr}_{a,b}$ and $\mathsf{sdr}_{c,d}$ are applicable in parallel to a legal string u if and only if u has one of the following forms:

1. $u = \ldots a \ldots b \ldots a \ldots b \ldots c \ldots d \ldots c \ldots d \ldots$, or
2. $u = \ldots a \ldots b \ldots a \ldots c \ldots d \ldots c \ldots d \ldots b \ldots$, or
3. $u = \ldots a \ldots b \ldots c \ldots d \ldots c \ldots d \ldots a \ldots b \ldots$, or
4. $u = \ldots a \ldots c \ldots d \ldots c \ldots d \ldots b \ldots a \ldots b \ldots$, or
5. $u = \ldots a \ldots b \ldots a \ldots c \ldots d \ldots b \ldots c \ldots d \ldots$, or
6. $u = \ldots a \ldots b \ldots c \ldots d \ldots a \ldots b \ldots c \ldots d \ldots$, or
7. $u = \ldots a \ldots b \ldots c \ldots d \ldots a \ldots c \ldots d \ldots b \ldots$, or
8. $u = \ldots a \ldots c \ldots d \ldots b \ldots a \ldots c \ldots d \ldots b \ldots$, or
9. $u = \ldots a \ldots c \ldots d \ldots b \ldots a \ldots b \ldots c \ldots d \ldots$, or
10. $u = \ldots a \ldots c \ldots d \ldots b \ldots c \ldots d \ldots a \ldots b \ldots$, or
11. $u = \ldots a \ldots c \ldots b \ldots d \ldots a \ldots c \ldots b \ldots d \ldots$, or
12. $u = \ldots a \ldots b \ldots a \ldots c \ldots b \ldots d \ldots c \ldots d \ldots$, or
13. $u = \ldots a \ldots b \ldots c \ldots a \ldots b \ldots d \ldots c \ldots d \ldots$, or
14. $u = \ldots a \ldots c \ldots b \ldots a \ldots b \ldots d \ldots c \ldots d \ldots$, or
15. $u = \ldots a \ldots b \ldots a \ldots c \ldots d \ldots c \ldots b \ldots d \ldots$, or
16. $u = \ldots a \ldots b \ldots c \ldots d \ldots c \ldots a \ldots b \ldots d \ldots$, or
17. $u = \ldots a \ldots c \ldots d \ldots c \ldots b \ldots a \ldots b \ldots d \ldots$.

In each of these cases, $(\mathsf{sdr}_{a,b} \circ \mathsf{sdr}_{c,d})(u) = (\mathsf{sdr}_{c,d} \circ \mathsf{sdr}_{a,b})(u)$.

Example 8. (i) For $u = rpqpsrqs$, $\mathsf{sdr}_{r,s} \circ \mathsf{sdr}_{p,q}$ is applicable to u, while $\mathsf{sdr}_{p,q} \circ \mathsf{sdr}_{r,s}$ is not. Indeed, $\mathsf{sdr}_{p,q}$ is not applicable to $\mathsf{sdr}_{r,s}(u) = qpqp$.

(ii) For $\omega = rsprqspq$, neither $\mathsf{sdr}_{r,s} \circ \mathsf{sdr}_{p,q}$ nor $\mathsf{sdr}_{p,q} \circ \mathsf{sdr}_{r,s}$ is applicable to ω. Indeed, $\mathsf{sdr}_{r,s}$ is not applicable to $\mathsf{sdr}_{p,q}(\omega) = rssr$, also $\mathsf{sdr}_{p,q}$ is not applicable to $\mathsf{sdr}_{r,s}(\omega) = qppq$.

(iii) For $\theta = rsrpqspq$, both $\mathsf{sdr}_{r,s} \circ \mathsf{sdr}_{p,q}$ and $\mathsf{sdr}_{p,q} \circ \mathsf{sdr}_{r,s}$ are applicable to θ. Indeed, $(\mathsf{sdr}_{r,s} \circ \mathsf{sdr}_{p,q})(\theta) = (\mathsf{sdr}_{r,s})(rsrs) = \Lambda$ and $(\mathsf{sdr}_{p,q} \circ \mathsf{sdr}_{r,s})(\theta) = (\mathsf{sdr}_{p,q})(pqpq) = \Lambda$. Thus, $\mathsf{sdr}_{r,s} \circ \mathsf{sdr}_{p,q}$ and $\mathsf{sdr}_{p,q} \circ \mathsf{sdr}_{r,s}$ are applicable in parallel to θ.

(iv) For $\zeta = rpsqrpsq$, the following pairs are applicable in parallel to ζ:

$$(\mathsf{sdr}_{r,s} \circ \mathsf{sdr}_{p,q})(\zeta) = (\mathsf{sdr}_{p,q} \circ \mathsf{sdr}_{r,s})(\zeta);$$
$$(\mathsf{sdr}_{r,q} \circ \mathsf{sdr}_{p,s})(\zeta) = (\mathsf{sdr}_{p,s} \circ \mathsf{sdr}_{r,q})(\zeta);$$
$$(\mathsf{sdr}_{r,p} \circ \mathsf{sdr}_{s,q})(\zeta) = (\mathsf{sdr}_{s,q} \circ \mathsf{sdr}_{r,p})(\zeta).$$

The following result follows from the analysis above.

Lemma 1. *If φ, $\psi \in \mathsf{Snr} \cup \mathsf{Spr} \cup \mathsf{Sdr}$ are applicable in parallel to the signed string u, then $(\varphi(\psi(u)) = \psi(\varphi(u))$.*

Using Lemma 1, we can prove now that for any number of rules, if they can be applied in parallel to a string u, then the result of the reduction is the same, regardless of the sequential order in which they are applied to u.

Theorem 2. *Let u be a legal string and let $S \subseteq \mathsf{Snr} \cup \mathsf{Spr} \cup \mathsf{Sdr}$ be a set of rules applied in parallel to u. Then, for any two compositions φ, φ' of the rules in S, $\varphi(u) = \varphi'(u)$.*

Proof. There is a sequence $\varphi = \varphi_0, \varphi_1, \ldots, \varphi_m = \varphi'$ of permutations of φ, where

$$\varphi_i = \varphi_{i2}\alpha_i\beta_i\varphi_{i1} \quad \text{and} \quad \varphi_{i+1} = \varphi_{i2}\beta_i\alpha_i\varphi_{i1},$$

for some compositions φ_{i1} and φ_{i2} and rules α_i and β_i. Therefore, it is sufficient to show the claim for the case, where the compositions are of the form $\varphi = \varphi_2\alpha\beta\varphi_1$ and $\varphi' = \varphi_2\beta\alpha\varphi_1$ for rules α and β. Also, in this case, $\varphi(u) = \varphi'(u)$ if and only if $\alpha\beta(\varphi_1(u)) = \beta\alpha(\varphi_1(u))$. Thus, the claim of the theorem is equivalent to proving that if $\alpha\beta(u)$ and $\beta\alpha(u)$ are both defined, then $\alpha\beta(u) = \beta\alpha(u)$. This however follows from Lemma 1. □

5 Graph reduction rules for gene assembly

We consider now the formalization of gene assembly through signed graphs. As it turns out, the higher level of abstraction given by signed graphs with respect to legal strings is crucial. E.g., while in the string model of gene assembly, 17 cases were needed to describe the parallel applicability of two sdr rules on strings, the same can be described for graphs in terms of avoiding two simple subgraph structures!

We recall in the following some basic definitions related signed graphs – we refer to [15] for more details.

A *signed graph* G is a structure $G = (V, E, \sigma)$, where (V, E) is a nondirected graph and $\sigma : V \to \{+, -\}$ is a vertex-labelling function. The graph (V, E) is called the *underlying graph* of G. G is called the *empty graph*, denoted $\emptyset$ if $V = \emptyset$. We denote an edge between vertices u, v as uv – since our graphs are nondirected, we have $uv = vu$ for all edges $uv \in E$. We say that a vertex $v \in V$ is *positive* (*negative*, resp.) if $\sigma(v) = +$ ($\sigma(v) = -$, resp.) We denote $V^+ = \{v \in V \mid \sigma(v) = +\}$ and $V^- = V \setminus V^+$. Let G^+ (G^-, resp.) be the signed subgraph of G induced by V^+ (V^-, resp.). For a vertex $p \in V$ we will also write $p \in G$; if $p \in V^-$ ($p \in V^+$, resp.), then we also write $p \in G^-$ ($p \in G^+$). We say that a signed graph is *all-negative* (*all-positive*, resp.) if $V = V^-$ ($V = V^+$, resp.). The *neighbourhood* of a vertex $v \in V$ is $N_G(v) = \{u \in V \mid uv \in E\}$. The vertex v is *isolated* if $N_G(v) = \emptyset$. The signed graph G is called *discrete* if all its vertices are isolated; in this case, the set V of vertices is called *stable*. G is called a *clique* if $E = \{uv \mid u, v \in V, u \neq v\}$.

For two signed graphs G_1, G_2, with $G_i = (V_i, E_i, \sigma_i)$, $i = 1, 2$, $V_1 \cap V_2 = \emptyset$, we denote by $G_1 \oplus G_2$ the *disjoint union* of G_1 and G_2, i.e., the graph (V_3, E_3, σ_3) with $V_3 = V_1 \cup V_2$, $E_3 = E_1 \cup E_2$, $\sigma_3(u) = \sigma_1(u)$ if $u \in V_1$ and $\sigma_3(u) = \sigma_2(u)$ if $u \in V_2$. We denote by $G_1 \otimes G_2$ the *complete connection* of graphs G_1 and G_2, i.e., the signed graph (V_4, E_4, σ_4) with $V_4 = V_1 \cup V_2$, $E_4 = E_1 \cup E_2 \cup \{uv \mid u \in V_1, v \in V_2\}$, $\sigma_4 = \sigma_3$. Note that the difference between the complete connection of two signed graphs G_1, G_2 and their disjoint unions is that the former has some extra edges: all possible edges between vertices of G_1 and vertices of G_2.

We say that a signed graph is a *complete bipartite graph* (with bipartition (n, m)), denoted by $K_{n,m}$, if there are disjoint discrete graphs G_1 and G_2 with n and m vertices, resp., such that $K_{n,m} = G_1 \otimes G_2$. A graph $K_{1,m}$ is also called a *star*. A signed graph G is called *complete tripartite* if there are discrete graphs G_1, G_2, G_3 such that $G = G_1 \otimes G_2 \otimes G_3$. A signed graph is called a *square*, denoted C_4 (*diamond*, resp., denoted D_4) if its underlying graph is isomorphic to the graph illustrated in Figure 6(a) (Figure 6(b), resp.)

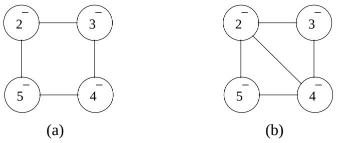

Figure 6: (a) The square C_4; (b) the diamond D_4.

Let u be a legal string over the alphabet Σ. We associate to u a unique signed graph $G_u = (V_u, E_u, \sigma_u)$ as follows:

- $V_u = \{a \in \Sigma \mid a \in \mathsf{dom}(u)\}$;
- $(a, b) \in E_u$ if and only if a and b overlap in u;
- $\sigma_u(a) = +$ ($-$, resp.) if a is positive (negative, resp.) in u.

E.g., the signed graph associated to the micronuclear gene *actin I* in *S. nova*, see Fig. 4 for its MDS sequence, is given in Fig. 7. Also, the signed graph associated to the micronuclear gene αTP in *S. nova*, see Fig. 5 for its MDS sequence, is given in Fig. 8 – the graph consists of one negative clique with vertices $\{2, 3, \ldots, 12\}$ and one negative discrete subgraph with vertices $\{13, 14\}$. We refer to [4] for many other examples.

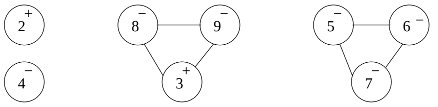

Figure 7: The signed overlap graph associated to the *actin I* gene in *S. nova*.

Let $G = (V, E, \sigma)$ be a signed graph and $S \subseteq V$. We say that the signed graph $G' = (V, E', \sigma')$ is obtained from G by *complementing* on the set of vertices S if G' results from G by replacing the subgraph induced by S with its complement (including the signs of the vertices in S); G' is denoted by $\mathsf{com}_S(G)$.

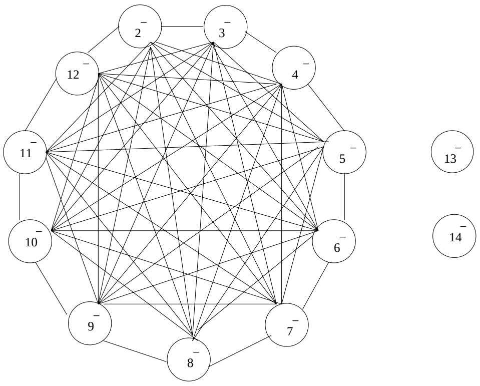

Figure 8: The signed graph associated to αTP protein in *S. nova* consists of one negative clique with vertices $\{2, 3, \ldots, 12\}$ and one negative discrete subgraph with vertices $\{13, 14\}$.

Moreover, if S is the neighbourhood $N_G(v)$ of a vertex $v \in V$, then we get the *local complement* $\mathsf{loc}_v(G)$ at v, i.e., $\mathsf{loc}_v(G) = \mathsf{com}_{N_G(v)}(G)$. For a vertex $u \in V$, we denote by $G - u$ the subgraph of G induced by $V \setminus \{u\}$.

The molecular operations ld, hi, and dlad are modelled on signed graphs by the rules gnr, gpr, and gdr defined bellow.

Let G be a signed graph.

- The *graph negative rule* for a vertex p is applicable to G if $p \in G^-$ is isolated. The result $\mathsf{gnr}_p(G)$ is the signed graph $\mathsf{gnr}_p(G) = G - p$. The *domain* of gnr_p is $\{p\}$.

  Let $\mathsf{Gnr} = \{\mathsf{gnr}_p \mid p \geq 1\}$ be the set of all graph negative rules on signed graphs.

- The *graph positive rule* for a vertex p is applicable to G if $p \in G^+$. The result $\mathsf{gpr}_p(G)$ is the signed graph $\mathsf{gpr}_p(G) = \mathsf{loc}_p(G) - p$. The *domain* of gpr_p is $\{p\}$.

  Let $\mathsf{Gpr} = \{\mathsf{gpr}_p \mid p \geq 1\}$ be the set of all graph positive rules on signed graphs.

- The *graph double rule* for two different vertices p and q is applicable to G if $p, q \in G^-$ are adjacent. The result $\mathsf{gdr}_{p,q}(G)$ is the signed graph where $\mathsf{gdr}_{p,q}(G) = (V \setminus \{p, q\}, E', \sigma')$ is obtained as follows: σ' equals σ restricted to $V \setminus \{p, q\}$, and E' is obtained from E by complementing the edges that join vertices in $N_G(p)$ to vertices in $N_G(q)$. This means that the status of a pair $\{x, y\}$ (for $x, y \in V \setminus \{p, q\}$) as an edge will change if and only if

  $$\begin{array}{lll} x \in N_G(p) \setminus N_G(q) & \text{and} & y \in N_G(q), \\ x \in N_G(p) \cap N_G(q) & \text{and} & y \in (N_G(q) \setminus N_G(p)) \cup (N_G(p) \setminus N_G(q)), \\ x \in N_G(q) \setminus N_G(p) & \text{and} & y \in N_G(p). \end{array}$$

  The *domain* of $\mathsf{gdr}_{p,q}$ is $\{p, q\}$.

  Let $\mathsf{Gdr} = \{\mathsf{gdr}_{p,q} \mid p, q \geq 1\}$ be the set of all graph double rules on signed graphs.

For a signed graph G and some operations $\varphi_1, \varphi_2, \ldots, \varphi_n \in \mathsf{Gnr} \cup \mathsf{Gpr} \cup \mathsf{Gdr}$, we say that $\varphi = \varphi_n \circ \cdots \circ \varphi_2 \circ \varphi_1$ is a *successful strategy* for G if $\varphi(G) = \emptyset$.

The following result is straightforward to prove (see [4], Lemma 11.3 for its counterpart for signed double occurrence strings). We skip the proof here.

Lemma 3. *Let $G = (V, E, \sigma)$ be a signed graph and $p, q \in V$. If $\mathsf{gdr}_{p,q}$ is applicable to G, then $\mathsf{gdr}_{p,q}(G) = \mathsf{gpr}_p(\mathsf{gpr}_q(\mathsf{loc}_p(G)))$.*

6 Parallelism in the graph-based model

In this section we consider the notion of parallelism in the framework of signed graphs.

Let $S \subseteq \mathsf{Gnr} \cup \mathsf{Gpr} \cup \mathsf{Gdr}$ be a set of rules and let $G = (V, E, \sigma)$ be a signed graph. We say that the rules in S *can be applied in parallel* to G if for any ordering $\varphi_1, \varphi_2, \ldots, \varphi_k$ of S, the composition $\varphi_k \circ \cdots \circ \varphi_1$ is applicable to G. In particular, two rules $\varphi, \psi \in \mathsf{Gnr} \cup \mathsf{Gpr} \cup \mathsf{Gdr}$ can be applied in parallel to G if both $\varphi \circ \psi$ and $\psi \circ \varphi$ are applicable to G.

The following result is straightforward to prove and provides a simple criterium for two rules to be applicable in parallel.

Theorem 4. *Let $G = (V, E, \sigma)$ be a signed graph and let $\varphi, \psi \in \mathsf{Gnr} \cup \mathsf{Gpr} \cup \mathsf{Gdr}$ be two rules applicable to G with $\mathsf{dom}(\varphi) \cap \mathsf{dom}(\psi) = \emptyset$.*

(i) If $\varphi \in \mathsf{Gnr}$, then φ and ψ can be applied in parallel to G.

(ii) If $\varphi = \mathsf{gpr}_p$ with $p \in V$, then φ and ψ can be applied in parallel to G if and only if $N_G(p) \cap \mathsf{dom}(\psi) = \emptyset$.

(iii) If $\varphi, \psi \in \mathsf{Gdr}$, then φ and ψ can applied in parallel to G if and only if the subgraph of G induced by $\mathsf{dom}(\varphi) \cup \mathsf{dom}(\psi)$ is not isomorphic to C_4 or D_4.

Proof. Claim (i) is trivial. For (ii), note that for any two positive adjacent vertices p, q, p is negative in the signed graph $\mathsf{gpr}_q(G)$. Thus, gpr_p and gpr_q are applicable in parallel if and only if p and q are not adjacent. Consider also two adjacent vertices $r, s \in G^-$ such that r is also adjacent to p. Then r is positive in $\mathsf{gpr}_p(G)$ and so, gpr_p and $\mathsf{gdr}_{r,s}$ are not applicable in parallel to G.

Case (iii) follows through a simple case analysis observing that for pointers $p, q, r, s \in G^-$, $\mathsf{gdr}_{p,q}$ and $\mathsf{gdr}_{r,s}$ are applicable in parallel to G if and only if p, q remain adjacent in $\mathsf{gdr}_{r,s}(G)$ and r, s remain adjacent in $\mathsf{gdr}_{p,q}(G)$. □

Example 9. Let G be the graph illustrated in Figure 9.

(i) Any two of the rules gpr_2, gnr_7, and $\mathsf{gdr}_{4,5}$ can be applied in parallel to G.

(ii) gpr_2 and gpr_3 cannot be applied in parallel to G, although each of them is applicable to G. Indeed, neither $\mathsf{gpr}_2 \circ \mathsf{gpr}_3$ nor $\mathsf{gpr}_3 \circ \mathsf{gpr}_2$ is applicable to G: in the signed graph $\mathsf{gpr}_2(G)$ ($\mathsf{gpr}_3(G)$, resp.) the vertex 3 (2, resp.) is negative, and thus, gpr_3 (gpr_2, resp.) is not applicable.

(iii) $\mathsf{gdr}_{4,5}$ and $\mathsf{gdr}_{5,6}$ are not applicable in parallel to G since applying one of them removes vertex 5, thus making the other one unapplicable.

(iv) The rules in each of the following sets are applicable in parallel to $G:$ $S_1 = \{\mathsf{gpr}_2, \mathsf{gdr}_{4,5}, \mathsf{gnr}_7\}$, $S_2 = \{\mathsf{gpr}_2, \mathsf{gdr}_{5,6}, \mathsf{gnr}_7\}$, $S_3 = \{\mathsf{gpr}_3, \mathsf{gdr}_{4,5}, \mathsf{gnr}_7\}$, $S_4 = \{\mathsf{gpr}_3, \mathsf{gdr}_{5,6}, \mathsf{gnr}_7\}$.

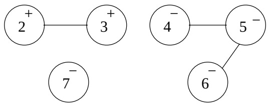

Figure 9: The graph G in Example 9.

Example 10. Let G be the signed overlap graph associated to the *actin I* gene in *S. nova*, illustrated in Figure 7. There are only 6 different maximal parallel strategies to reduce this graph:

$\{\mathsf{gpr}_2, \mathsf{gnr}_4, \mathsf{gdr}_{5,6}, \mathsf{gdr}_{8,9}\}\{\mathsf{gnr}_7, \mathsf{gpr}_3\}$; $\{\mathsf{gpr}_2, \mathsf{gpr}_3, \mathsf{gnr}_4, \mathsf{gdr}_{5,6}\}\{\mathsf{gnr}_7, \mathsf{gpr}_8, \mathsf{gpr}_9\}$;
$\{\mathsf{gpr}_2, \mathsf{gnr}_4, \mathsf{gdr}_{6,7}, \mathsf{gdr}_{8,9}\}\{\mathsf{gnr}_5, \mathsf{gpr}_3\}$; $\{\mathsf{gpr}_2, \mathsf{gpr}_3, \mathsf{gnr}_4, \mathsf{gdr}_{5,7}\}\{\mathsf{gnr}_6, \mathsf{gpr}_8, \mathsf{gpr}_9\}$;
$\{\mathsf{gpr}_2, \mathsf{gnr}_4, \mathsf{gdr}_{5,7}, \mathsf{gdr}_{8,9}\}\{\mathsf{gnr}_6, \mathsf{gpr}_3\}$; $\{\mathsf{gpr}_2, \mathsf{gpr}_3, \mathsf{gnr}_4, \mathsf{gdr}_{6,7}\}\{\mathsf{gnr}_5, \mathsf{gpr}_8, \mathsf{gpr}_9\}$.

Note that there are 3060 sequential strategies to reduce this graph (and assemble the gene) – the reason for this difference is that many sequential strategies coincide modulo commutation of some rules – as it turns out, these rules may be applied in parallel.

According to our definition, if a set of rules is applicable in parallel to a signed graph, then any composition of these rules is applicable to that graph. This definition does not require that the result of applying different compositions of rules must be the same. However, we prove in the following that this is indeed the case.

We consider first the case of two rules and prove that if both $\varphi \circ \psi$ and $\psi \circ \varphi$ are applicable to a graph G, then $(\varphi \circ \psi)(G) = (\psi \circ \varphi)(G)$. For this, we first prove the following lemma.

Lemma 5. *Let G be a signed graph, $G = (V, E, \sigma)$ and $S_1, S_2 \subseteq V$. Then $\mathsf{com}_{S_1}(\mathsf{com}_{S_2}(G)) = \mathsf{com}_{S_2}(\mathsf{com}_{S_1}(G))$.*

Proof. Let $G_1 = \mathsf{com}_{S_1}(\mathsf{com}_{S_2}(G))$ and $G_2 = \mathsf{com}_{S_2}(\mathsf{com}_{S_1}(G))$ with $G_i = (V, E_i, \sigma_i)$, $i = 1, 2$. Then clearly, for any $p, q \in S_1 \cup S_2$, we have the following:

(i) $(p, q) \in E \cap E_i$ if and only if $p, q \in S_1 \cap S_2$;

(ii) $\sigma_i(p) = \sigma(p)$ if and only if $p \in S_1 \cap S_2$.

Also, for any $p \in V \setminus (S_1 \cup S_2)$ we have the following:

(iii) for any $q \in S_1 \cup S_2$, $pq \in E_i$ if and only if $pq \in E$;

(iv) $\sigma_i(p) = \sigma(p)$.

Consequently, $G_1 = G_2$, proving the claim. □

We are now ready to prove the result announced above. We note again that the definition of parallelism only presumes that the rules are applicable in any possible order – this is enough to ensure that the result is always the same regardless of the order in which they are applied, as shown in the next theorem.

Theorem 6. *If $\varphi, \psi \in \mathsf{Gnr} \cup \mathsf{Gpr} \cup \mathsf{Gdr}$ are applicable in parallel to the signed graph G, then $\varphi(\psi(G)) = \psi(\varphi(G))$.*

Proof. If $\varphi \in \mathsf{Gnr}$ or $\psi \in \mathsf{Gnr}$, then the result is trivial. The rest of the cases follow easily from Lemmata 5 and 3 observing that for any $S \subseteq V$ and $p, q \in V$, $\mathsf{com}_S(G) - p = \mathsf{com}_S(G - p)$ and $(G - q) - p = (G - p) - q$. Indeed, all our rules can be expressed as compositions of com and vertex removals. □

The general case follows now easily from Theorem 6, using a similar proof as for Theorem 2, the counterpart result for the string-based model.

Theorem 7. *Let G be a signed graph and let $S \subseteq \mathsf{Gnr} \cup \mathsf{Gpr} \cup \mathsf{Gdr}$ be a set of rules applicable in parallel to G. Then for any two compositions φ, φ' of the rules in S, $\varphi(G) = \varphi'(G)$.*

It is important to note that if the rules in S are applicable in parallel to the signed graph G, then the rules in any subset of S are applicable in parallel to G. However, the reverse is not true, as shown by the following example.

Example 11. Let G be the signed graph in Figure 10. Then any two rules from the set $S = \{\mathsf{gdr}_{2,3}, \mathsf{gdr}_{4,5}, \mathsf{gdr}_{6,7}\}$ are applicable in parallel to G by Theorem 4. However, the rules in S are not applicable in parallel to G. Indeed, applying any two of them to G in an arbitrary order makes the third one unapplicable. E.g., $(\mathsf{gdr}_{2,3} \circ \mathsf{gdr}_{4,5})(G)$ is the isolated all-negative graph on the set of vertices $\{6, 7\}$. Clearly, $\mathsf{gdr}_{6,7}$ is not applicable to this graph.

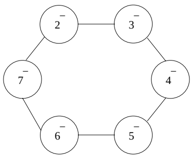

Figure 10: The graph G in Example 11.

The following problem seems to be difficult: check whether or not a given set of rules can be applied in parallel to a given signed graph. In the next theorem we give a simple criterium in the case when at most two gdr-s are among the rules to be applied.

Theorem 8. *Let G be a signed graph and $S \subseteq \mathsf{Gnr} \cup \mathsf{Gpr} \cup \mathsf{Gdr}$ a set of rules containing at most two gdr's. Let P be the union of domains of rules in S with $P^+ = \{p \in P \mid \sigma(p) = +\}$, and $P^- = P \setminus P^+$. Then the rules in S can be applied in parallel to G if and only if the following conditions are satisfied:*

(i) The subgraph induced by P^+ is discrete. Moreover, there is no edge between vertices in P^+ and vertices in P^-.

(ii) The subgraph induced by P^- does not contain induced squares C_4 or diamonds D_4.

Proof. Condition (i) of the theorem is clearly necessary: if there were an edge pq with $p, q \in P^+$, then $\mathsf{gpr}_p \circ \mathsf{gpr}_q$ would not be applicable to G, contradicting the parallelism of the rules in S.

Condition (ii) is also easily seen to be necessary. For this, assume that there is $\mathsf{gdr}_{p,q}, \mathsf{gdr}_{r,s} \in S$ such that the subgraph induced by $\{p, q, r, s\}$ is isomorphic to either C_4, or D_4. Then there is no edge between r and s in $\mathsf{gdr}_{p,q}(G)$ and so, $\mathsf{gdr}_{r,s} \circ \mathsf{gdr}_{p,q}$ is not applicable to G; a contradiction.

Assume now that conditions (i) and (ii) hold and consider an arbitrary composition $\varphi_1 \circ \cdots \circ \varphi_n$ of the rules in S. It is easy to see that property (i) is preserved throughout the reduction $\varphi_1 \circ \cdots \circ \varphi_n$. Indeed, creating an edge between vertices in P^+ or between vertices in P^+ and P^- is only possible if such an edge existed before and gpr or gdr was applied, see the definition of our rules. For the same reason, the edges between the vertices in P^- are not modified throughout the reduction prior to applying a gdr, if one exists in S. Since the vertices in P^- do not induce C_4 or D_4, the second gdr, if it exists in S, will remain applicable to G. □

Note however that Theorem 8 does not hold when more than two gdr-s are in the considered set of rules, as shown in Example 11. Indeed, for the graph illustrated in Figure 10, $\mathsf{gdr}_{2,3}, \mathsf{gdr}_{4,5}, \mathsf{gdr}_{6,7}$ are not applicable in parallel, although no four vertices from $\{2, 3, \ldots, 7\}$ induce a subgraph isomorphic to C_4 or D_4.

7 Parallel complexity of micronuclear genes

A new natural notion of complexity can be defined for the process of gene assembly using the notion of parallelism. The *parallel complexity* of a micronuclear gene (and of its associated signed graph) is the minimal number of steps needed to reduce in parallel the signed graph associated to that gene. We will investigate this notion now and show how this leads to several intriguing question on signed graphs.

Example 12. (i) Consider the micronuclear gene *C2* in *S. nova*, having four MDSs placed in the orthodox order. The signed graph associated to this gene is the all-negative discrete graph with four vertices. Thus, its parallel complexity is 1.

(ii) Consider the micronuclear gene *actin I* in *S. nova* with the associated signed graph illustrated in Figure 7, see also Example 9. Its parallel complexity is two and a parallel strategy in two steps is $\{\mathsf{gpr}_2, \mathsf{gnr}_4, \mathsf{gpr}_3, \mathsf{gdr}_{5,6}\}$ $\{\mathsf{gnr}_7, \mathsf{gpr}_8, \mathsf{gpr}_9\}$, see also Example 10.

Example 13. (a) A discrete graph has parallel complexity one.

(b) An all-positive clique G has parallel complexity at most two. Indeed, for any node u of G, $\mathsf{gpr}_u(G)$ is either empty, or a discrete graph.

(c) An all-negative complete bipartite graph has parallel complexity at most two. Indeed, for any edge uv of G, $\mathsf{gdr}_{u,v}(G)$ is either empty, or a discrete graph.

(d) An all-negative complete tripartite graph has parallel complexity at most two. To see this, consider an arbitrary edge uv of G and note that for any other edge pq of G, $\{p, q, u, v\}$ induces either a subgraph C_4 or a subgraph D_4 in G, see Fig. 11. Thus, p and q will not be adjacent in the graph $\mathsf{gdr}_{u,v}(G)$. Consequently, $\mathsf{gdr}_{u,v}(G)$ is either empty, or a discrete graph.

As it turns out, it is difficult to find signed graphs with parallel complexity higher than 4, or all-negative graphs with parallel complexity higher than 2, see Conjectures 1 and 2 in Section 8. There are at least two types of graphs that seem intuitively "difficult to reduce": graphs on which no two rules can be applied in parallel in the first step, and graphs that avoid a certain rule (such as gdr that reduces two vertices at once) in all parallel reductions. We characterize these two types of graphs in the following.

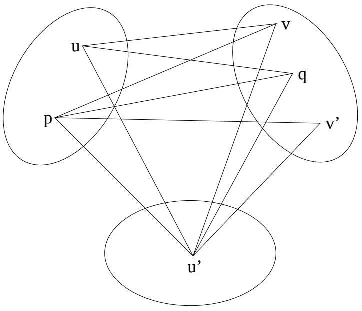

Figure 11: An all-negative complete tripartite graph has parallel complexity at most two.

7.1 Graphs with no parallelism in the first step

The following result is clear by Theorem 4 and Example 13.

Lemma 9. *Let G be a signed graph.*

(i) G has no two parallel applications of gnr*-rules if and only if G has at most one isolated negative vertex.*

(ii) G has no two parallel applications of gpr*-rules if and only if G^+ is a clique. Moreover, if G is all-positive, then its parallel complexity is at most two.*

Lemma 10. *A signed graph G has no two parallel applications of* gdr*-rules if and only if the subgraph G^- induced by the negative vertices is equal to $G^- = (K \otimes S_1) \oplus S_2$, where K is a complete bipartite graph and S_1 and S_2 are (possibly empty) discrete graphs.*

Proof. We assume without loss of generality that the graph G is all-negative.

For the direct implication, note that the result holds for graphs with up to 4 vertices. Consider a larger graph and note that in that graph there can be several isolated vertices and only one non-trivial connected component. We can remove the isolated vertices and thus assume without loss of generality that the graph is connected. Let $G = (V, E, \sigma)$.

Claim 1. Let $x, y \in V$. If $xy \notin E$ then $N(x) = N(y)$.

Proof of Claim 1. Assume there exists a vertex $z \in N(x) \setminus N(y)$. Since G is connected, there exists a vertex $z' \in N(y)$. Now, the set $\{x, y, z, z'\}$ induces a subgraph different from C_4 and D_4 and so, $\mathsf{gdr}_{x,z}$ and $\mathsf{gdr}_{y,z'}$ are both applicable to G. This proves the claim.

Let S be a maximal stable subset of G. Hence each vertex $x \notin S$ is adjacent to a vertex in S, and, by Claim 1, S and $W = V \setminus S$ are completely connected. Since G has no subgraphs K_4, W does not have triangles K_3 and thus no subgraphs D_4. Therefore all four element induced subgraphs are isomorphic to C_4. This is possible only if W induces a complete bipartite subgraph.

The reverse implication is clearly true. □

Note that the above proof could be simplified using a result from [9] stating that a graph G is complete bipartite if and only if it does not have an induced K_3 nor induced $K_2 \oplus K_1$.

Corollary 11. *A signed graph G has no two parallel applications of* gdr*-rules if and only if G^- consists of a discrete graph and a complete tripartite graph, where some of the three components can also be empty. Moreover, if G is all-negative, then its parallel complexity is at most two.*

An induced subgraph H is said to be a *shadow* of a vertex set (or a subgraph) A if for each $x \in A$ and edge uv of H, x is adjacent to u or v or both, and each isolated vertex of H is adjacent to a vertex in A.

Theorem 12. *Let G be a signed graph of at least two vertices. Then G has no parallel applications of the rules (*gnr*,* gpr*, and* gdr*) if and only if*

(i) G^+ is a clique,

(ii) G^- is a shadow of G^+, and

(iii) G^- consists of a discrete graph and a complete tripartite graph.

Moreover, if G is all-negative or all-positive, then its parallel complexity is at most two.

Proof. Condition (i) follows from Lemma 9 and condition (ii) obviously holds since no gpr can be applied in parallel with a gnr or with a gdr. Condition (iii) of the theorem follows from Corollary 11.

□

7.2 Graphs that avoid one type of reduction

The signed graphs that have no reductions using gnr are those that can be reduced using only gpr and gdr. A string-based characterization was given in [3], but giving a similar graph-based characterization remains an open problem.

Lemma 13. *A signed graph G has no reductions using* gpr *if and only if G is all-negative.*

Lemma 14. *Let G be a connected signed graph with no reduction using* gdr*. Then $G = G^+ \otimes G^-$. Moreover, G^+ is either a clique, or a disjoint union of two cliques, and G^- is discrete. That is, $G = (K \oplus K') \otimes S$, where K, K' are all-positive cliques and S is an all-negative discrete graph, where K, K', and S can be empty. Moreover, G can be reduced in at most three parallel steps.*

Proof. If there is an induced path P_3 of three vertices in G^+, then by applying gpr to the middle vertex v we obtain an edge with negative ends. Hence gdr can be applied to $\mathsf{gpr}_v(G)$; a contradiction. Obviously, if a connected graph does not have an induced P_3, then it is a clique. Therefore, G^+ is a disjoint union of cliques.

If $G^- = \emptyset$, then G, being connected, must be an all-positive clique, and the claim follows. Assume thus that G^- is not empty; necessarily, G^- is discrete. Let $x \in G^-$. Assume that $u \in N(x)$ and $u, v \in G^+$ with $uv \in E$. Then also $v \in N(x)$, since xv is an edge in $\mathsf{gpr}_v(G)$ with negative ends. This shows that if $N(x) \cap G_i \neq \emptyset$, then $G_i \subseteq N(x)$, which yields that $N(x)$ is a union of cliques G_i. Now G does not have an induced $K_{1,m}$ with at least three positive leaves. Indeed, if $u, v, w \in N(x)$ are not pairwise adjacent, then in $G' = \mathsf{gpr}_v(G)$, the vertices x, u, w are positive, and they are in the same connected component. However, uw is not an edge in G' and so, $\mathsf{gdr}_{u,w}$ will be applicable to $\mathsf{gpr}_x(\mathsf{gpr}_v(G))$; a contradiction. Consequently, any negative vertex is connected to at most two positive cliques in G.

Finally, if also $y \in G^-$, $v \in N(x)$ and $u \in N(x) \cap N(y)$, then also $v \in N(y)$. Indeed, otherwise, $\mathsf{gdr}_{v,y}$ is applicable to $\mathsf{gpr}_x(\mathsf{gpr}_u(G))$. Consequently, G^+ is either a clique, or a disjoint union of two cliques, proving the first part of the lemma.

For the second part, note that if G^+ is a clique, then applying any gpr to G transforms it into a disjoint union of a positive clique and a discrete all-negative graph, reducible in two steps. If G^+ is a disjoint union of two cliques, then applying two gpr rules in parallel, one in each clique, transforms G into a discrete all-negative graph, thus reducible in one parallel step. □

7.3 Some bounds on the parallel complexity

Some upper and lower bounds on the parallel complexity of a signed graph are given in the next result.

Lemma 15. *(i) The parallel complexity of a signed graph with n vertices is at most $n/2+3$. If the graph is all-negative, then its complexity is at most $n/4+2$.*

(ii) There are signed graphs with parallel complexity four. There are all-negative graphs with parallel complexity two.

Proof. (i) Consider a signed graph $G = (V, E, \sigma)$ on n vertices. If at least one rule gdr or two rules gpr are applicable to G, then the number of vertices is decreased by at least two in one step. Assume then that no gdr and only at most one gpr is applicable to G. We may also assume without loss of generality that G has no isolated vertices. Then G^+ is a clique, G^- is discrete, and G^- is a shadow of G^+. Consider then an arbitrary $u \in G^+$ and let $A = V^+ \setminus \{u\}$, $B = N_G(u) \cap V^-$, and $C = V^- \setminus B$. If $G_1 = \mathsf{gpr}_u(G)$, then A and C induce all-negative discrete graphs in G_1, B induces an all-positive clique, and there are no edges between B and C.

If at least two gdr rules, or one gpr and one gdr are applicable to G_1, then at least 3 vertices are eliminated. Thus, we eliminate at least 4 vertices from G in two steps and we can continue the same reasoning as above with G_1 instead of G. If this is not the case, then in G_1 no two rules can be applied in parallel and by Theorem 12, A is completely connected in G_1 to B and to C. Thus, in the signed graph G, A is completely connected to C and no edges exist between A and B, or between B and C. Consequently, G can be reduced in at most 3 steps. Indeed, the first step of such a strategy for G applies rule gpr_p for some $p \in A$. Then, in $\mathsf{gpr}_p(G)$, A and B induce all-negative discrete graphs, C an all-positive clique, and the only edges are those completely connecting $\{u\}$ and B. Clearly, this graph can be then reduced in two steps.

Assume now that the graph G is all-negative. If at least two rules gdr are applicable to G in parallel, then the order of G is decreased by at least 4 in one step. If no gdr are applicable to G, then G is discrete and thus reducible in one step. If exactly one gdr is applicable to G, then, by Lemma 10, $G = T \oplus D$, with T a complete tripartite graph and D a discrete graph. Such a graph can be reduced in at most two steps, see Example 13.

(ii) The square C_4 cannot be reduced using less than two steps. Also, the graph in Figure 12 cannot be reduced using less than four steps. One strategy reducing the graph in four steps is the following: $\{\mathsf{gpr}_2\}\{\mathsf{gdr}_{3,6}\}\{\mathsf{gpr}_4, \mathsf{gnr}_7\}\{\mathsf{gnr}_5\}$.

□

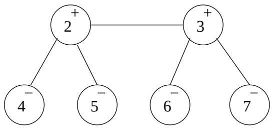

Figure 12: A signed graph irreducible in less than four parallel steps.

8 Conclusions

We have investigated in this paper a notion of parallelism for reducing signed graphs such as those associated to micronuclear gene patterns. The fact that parallelism could only be tediously formalized for strings is not surprising: signed graphs give a higher level of abstraction making easier the analysis of the gene assembly process.

We also introduced a notion of parallel complexity for micronuclear genes and their signed graphs, given by the minimal number of steps needed in a parallel reduction. Surprisingly, we have been unable to find examples of graphs with high parallel complexity; we conjecture that no such graphs exist. More specifically, we state the following two conjectures:

Conjecture 1. *Any all-negative graph can be reduced in parallel in at most two steps.*

Conjecture 2. *Any signed graph can be reduced in parallel in at most four steps.*

Notably, Conjecture 1 is open even for "well-structured" graphs, such as bipartite graphs. We also state the following interesting graph-theoretical conjecture.

Conjecture 3. *Let G be a black and white graph of vertices $v_1, ..., v_n, x$, where x is a special sink vertex (output). Let loc_i be the local complementation at v_i, which is applicable if v_i is black in the current graph. If π_1 and π_2 are any two permutations of $\mathsf{loc}_1, \mathsf{loc}_2, ..., \mathsf{loc}_n$ that are applicable to G, then x has the same color in both $\pi_1(G)$ and $\pi_2(G)$.*

Another interesting question is how one can find an optimal (minimal) parallel reduction strategy for a given signed graph. Also, the computational complexity of this optimization problem is yet-to-be established. Clearly, using a parallel reduction following a greedy strategy (maximize the number of nodes to be reduced in each step) does not necessarily lead to an optimal result, as shown by the following example.

Example 14. Let G be the signed diamond D_4 in Figure 13. Maximizing the number of rules to be applied in parallel in the first step of the reduction we get a parallel strategy reducing G in 3 steps: $\{\mathsf{gpr}_3, \mathsf{gpr}_4\}\{\mathsf{gpr}_5\}\{\mathsf{gpr}_2\}$. Note however that there is another strategy using less rules in the first step, still reducing the graph in two parallel steps: $\{\mathsf{gpr}_5\}\{\mathsf{gpr}_2, \mathsf{gdr}_{3,4}\}$.

Acknowledgements

The authors are supported by European Union project MolCoNet, IST-2001-32008. T. Harju gratefully acknowledges support of the Academy of Finland, project 39802. I. Petre gratefully acknowledges support of the Academy of Finland, project 105128. G. Rozenberg gratefully acknowledges support by NSF grant 0121422.

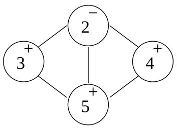

Figure 13: The graph G in Example 14.

References

[1] Ehrenfeucht, A., Harju, T., Petre, I., Prescott, D. M., and Rozenberg, G., Formal systems for gene assembly in ciliates. *Theoret. Comput. Sci.* **292** (2003) 199–219

[2] Ehrenfeucht, A., Harju, T., Petre, I., and Rozenberg, G., Patterns of micronuclear genes in cliates. *Lecture Notes in Comput. Sci.* **2340** (2002) 279–289

[3] Ehrenfeucht, A., Harju, T., Petre, I., and Rozenberg, G., Characterizing the micronuclear gene patterns in ciliates. *Theory of Comput. Syst.* **35** (2002) 501–519

[4] Ehrenfeucht, A., Harju, T., Petre, I., Prescott, D. M., and Rozenberg, G., *Computation in Living Cells: Gene Assembly in Ciliates*, Springer (2003).

[5] Ehrenfeucht, A., Petre, I., Prescott, D. M., and Rozenberg, G., Universal and simple operations for gene assembly in ciliates. In: V. Mitrana and C. Martin-Vide (eds.) *Words, Sequences, Languages: Where Computer Science, Biology and Linguistics Meet*, Kluwer Academic, Dortrecht (2001) pp. 329–342

[6] Ehrenfeucht, A., Petre, I., Prescott, D. M., and Rozenberg, G., String and graph reduction systems for gene assembly in ciliates. *Math. Structures Comput. Sci.* **12** (2001) 113–134

[7] Ehrenfeucht, A., Petre, I., Prescott, D. M., and Rozenberg, G., Circularity and other invariants of gene assembly in cliates. In: M. Ito, Gh. Păun and S. Yu (eds.) *Words, semigroups, and transductions*, World Scientific, Singapore (2001) 81–97

[8] Ehrenfeucht, A., Prescott, D. M., and Rozenberg, G., Computational aspects of gene (un)scrambling in ciliates. In: L. F. Landweber, E. Winfree (eds.) *Evolution as Computation*, Springer, Berlin, Heidelberg, New York (2001) pp. 216–256

[9] Hage, J., Harju, T., and Welzl, E., Euler graphs, triangle-free graphs and bipartite graphs in switching classes. *Fundamenta Inf.*, to appear.

[10] Harju, T., Petre, I., and Rozenberg, G., Gene assembly in ciliates: formal frameworks. In: G.Paun, G. Rozenberg, A.Salomaa (Eds.) *Current Trends in Theoretical Computer Science*, (2004).

[11] Jahn, C. L., and Klobutcher, L. A., Genome remodeilng in ciliated protozoa. *Ann. Rev. Microbiol.* **56** (2000), 489–520.

[12] Prescott, D. M., The evolutionary scrambling and developmental unscabling of germlike genes in hypotrichous ciliates. *Nucl. Acids Res.* **27** (1999), 1243 – 1250.

[13] Prescott, D. M., Genome gymnastics: unique modes of DNA evolution and processing in ciliates. *Nat. Rev. Genet.* 1(3) (2000) 191–198

[14] Prescott, D. M., Ehrenfeucht, A., and Rozenberg, G., Molecular operations for DNA processing in hypotrichous ciliates. *Europ. J. Protistology* **37** (2001) 241–260

[15] West, D. B., *Introduction to Graph Theory*, Prentice Hall, Upper Saddle River, NJ (1996)

Turku Centre *for* Computer Science

Lemminkäisenkatu 14 A, 20520 Turku, Finland | www.tucs.fi

University of Turku

- Department of Information Technology
- Department of Mathematical Sciences

Åbo Akademi University

- Department of Computer Science
- Institute for Advanced Management Systems Research

Turku School of Economics and Business Administration

- Institute of Information Systems Sciences

ISBN 952-12-1352-3
ISSN 1239-1891