Models of natural computation : gene assembly and membrane systems

Brijder, R.

Chapter 5

How Overlap Determines Reduction Graphs for Gene Assembly

Abstract
Ciliates are unicellular organisms having two types of functionally different nuclei: micronucleus and macronucleus. Gene assembly transforms a micronucleus into a macronucleus, thereby transforming each gene from its micronuclear form to its macronuclear form. Within a formal intramolecular model of gene assembly based on strings, the notion of reduction graph represents the macronuclear form of a gene, including byproducts, given only a description of the micronuclear form of that gene. For a more abstract model of gene assembly based on graphs, one cannot, in general, define the notion of reduction graphs. We show that if we restrict ourselves to the so-called realistic overlap graphs (which correspond to genes occurring in nature), then the notion of reduction graph can be defined in a manner equivalent to the string model. This allows one to carry over from the string model to the graph model several results that rely on the notion of reduction graph.

5.1 Introduction

Gene assembly is a process that takes place in unicellular organisms called ciliates, which have two types of functionally different nuclei: micronucleus (MIC) and macronucleus (MAC). Gene assembly transforms the genome of the MIC into the genome of the MAC. The two genomes are dramatically different in both the global form of their chromosomes and in the local form of single genes. During gene assembly each gene in its MIC form gets transformed into the same gene in its MAC form. See [12] for a detailed account of the biology of gene assembly.

In this chapter we consider only intramolecular models of gene assembly – thus here we do not consider the intermolecular models initiated by Landweber and Kari [20], and further developed by Daley et al. [10, 9]. Among the formal models of intramolecular gene assembly the string pointer reduction system (SPRS) and the graph pointer reduction system (GPRS), see [12], are of interest for this chapter. The SPRS consist of three types of string rewriting rules operating on so-called legal strings, while the GPRS consist of three types of graph rewriting rules operating on so-called overlap graphs. The GPRS is an abstraction of the SPRS: some information present in the SPRS is lost in the GPRS.

Realistic strings are strings that represent genes in their micronuclear form. Legal strings are an abstraction of realistic strings. The reduction graph, which is defined for legal strings, is a notion that describes the gene corresponding to the legal string in its macronuclear form (along with its waste products: the substrings "spliced out" in the process) – it is unique for a given legal string. It has been shown that the reduction graph retains the information needed to characterize which string negative rules (one of the three types of string rewriting rules) can be used during the transformation of a MIC form of a gene to its MAC form [6, 4]. Therefore it would be useful to have a notion of the reduction graph also for the GPRS. However, this is not so straightforward. We will demonstrate that, since the GPRS loses some information concerning the application of string negative rules, in general there is no unique reduction graph for a given overlap graph, cf. Example 6. However, as we will show, when we restrict ourselves to "realistic" overlap graphs then one gets a unique reduction graph. These overlap graphs are called realistic since they correspond to (micronuclear) genes. In this chapter, we explicitly define the notion of reduction graph for realistic overlap graphs (within the GPRS) and show that it is equivalent to the notion of reduction graph for legal strings (within the SPRS). Finally, we give a number of direct corollaries of this equivalence, including an answer to an open problem formulated in Chapter 13 in [12].

In Section 5.2 we recall some basic notions and notation concerning sets, strings and graphs. In Section 5.3 we recall notions used in models for gene assembly, such as legal strings, realistic strings and overlap graphs. In Section 5.4 we recall the notion of reduction graph within the framework of SPRS and we prove some elementary properties of this graph for legal strings. In particular we establish a calculus for the sets of overlapping pointers between vertices of the reduction graph. In Section 5.5 we prove properties of the reduction graph for a more restricted type of legal strings, the realistic strings. It is shown that reduction graphs of realistic strings have a subgraph of a specific structure, the root subgraph. Moreover, we show (using the calculus from Section 5.4) that the existence of the other edges in the reduction graph depends directly on the overlap graph. In Section 5.6 we provide a convenient function for reduction graphs that allows one to simplify reduction graphs without losing any information. In Section 5.7 we define the reduction graph for realistic overlap graphs, and prove the main theorem of this chapter: the equivalence of reduction graphs defined for realistic strings with

the reduction graphs defined for realistic overlap graphs. In Section 5.8 we discuss some immediate consequences of this theorem. A conference version of this chapter, which does not contain any proofs, was presented at FCT '07 [7].

5.2 Notation and Terminology

In this section we recall some basic notions concerning functions, strings, and graphs. We do this mainly to set up the basic notation and terminology for this chapter.

The cardinality of a set X is denoted by $|X|$. The symmetric difference of sets X and Y, $(X\backslash Y) \cup (Y\backslash X)$, is denoted by $X \oplus Y$. Since symmetric difference is associative, we extend it to (finite) families of sets $(X_i)_{i\in A}$, and denote this by $\bigoplus_{i\in A} X_i$. The *composition* of functions $f : X \to Y$ and $g : Y \to Z$ is the function $gf : X \to Z$ such that $(gf)(x) = g(f(x))$ for every $x \in X$. The restriction of f to a subset A of X is denoted by $f|A$.

We use λ to denote the empty string. For strings u and v, we say that v is a *substring of* u if $u = w_1vw_2$, for some strings w_1, w_2; we also say that v *occurs in* u. Also, v is a *cyclic substring of* u if either v is a substring of u or $u = v_2wv_1$ and $v = v_1v_2$ for some strings v_1, v_2, w. We say that v is a *conjugate of* u if $u = w_1w_2$ and $v = w_2w_1$ for some strings w_1 and w_2. For a string $u = x_1x_2\cdots x_n$ over Σ with $x_i \in \Sigma$ for all $i \in \{1, \ldots, n\}$, we say that $v = x_nx_{n-1}\cdots x_1$ is the *reversal of* u. A *homomorphism* is a function $\varphi : \Sigma^* \to \Delta^*$ such that $\varphi(uv) = \varphi(u)\varphi(v)$ for all $u, v \in \Sigma^*$.

We move now to graphs. A *labelled graph* is a 4-tuple $G = (V, E, f, \Gamma)$, where V is a finite set, $E \subseteq \{\{x, y\} \mid x, y \in V, x \neq y\}$, and $f : V \to \Gamma$. The elements of V are called *vertices* and the elements of E are called *edges*. Function f is the *labelling function* and the elements of Γ are the *labels*. Note that our graphs are not directed and do not have loops.

We say that G is *discrete* if $E = \varnothing$. Labelled graph $G' = (V', E', f|V', \Gamma)$ is a *subgraph of* G if $V' \subseteq V$ and $E' \subseteq E_{V'} = E \cap \{\{x, y\} \mid x, y \in V', x \neq y\}$. If $E' = E_{V'}$, we say that G' is the *subgraph of* G *induced by* V'. A string $\pi = e_1e_2\cdots e_n \in E^*$ with $n \geq 1$ is a *path in* G if there is a $v_1v_2\cdots v_{n+1} \in V^*$ such that $e_i = \{v_i, v_{i+1}\}$ for all $1 \leq i \leq n$. Labelled graph G is *connected* if there is a path between every two vertices of G. A subgraph H of G induced by V_H is a *component of* G if both H is connected and for every edge $e \in E$ we have either $e \subseteq V_H$ or $e \subseteq V\backslash V_H$.

Labelled graphs $G = (V, E, f, \Gamma)$ and $G' = (V', E', f', \Gamma)$ are *isomorphic*, denoted by $G \approx G'$, if there is a bijection $\alpha : V \to V'$ such that $f(v) = f'(\alpha(v))$ for all $v \in V$, and $\{x, y\} \in E$ iff $\{\alpha(x), \alpha(y)\} \in E'$ for all $x, y \in V$. Any such bijection α is then called an *isomorphism from* G *to* G'. It is important to realize that we require that the labels of vertices identified by an isomorphism are equal.

In this chapter we will consider graphs with two sets of edges. Therefore, we need the notion of 2-edge coloured graphs. A 2-*edge coloured graph* is a 5-tuple $G = (V, E_1, E_2, f, \Gamma)$, where both (V, E_1, f, Γ) and (V, E_2, f, Γ) are labelled graphs.

The basic notions and notation for labelled graphs carry over to 2-edge coloured graphs. However, to extend the notion of isomorphism care must be taken that the two sorts of edges are preserved. Thus, if $G = (V, E_1, E_2, f, \Gamma)$ and $G' = (V', E_1', E_2', f', \Gamma')$ are 2-edge coloured graphs, and α is an isomorphism from G to G', then $(x, y) \in E_i$ iff $(\alpha(x), \alpha(y)) \in E_i'$ for $x, y \in V$ and $i \in \{1, 2\}$.

5.3 Gene Assembly in Ciliates

Two models that are used to formalize the process of gene assembly in ciliates are the string pointer reduction system (SPRS) and the graph pointer reduction system (GPRS). The SPRS consist of three types of string rewriting rules operating on *legal strings* while the GPRS consist of three types of graph rewriting rules operating on *overlap graphs*. For the purpose of this chapter it is not necessary to recall the string and graph rewriting rules; a complete description of SPRS and GPRS, as well as a proof of their "weak" equivalence, can be found in [12]. We do recall the notions of legal string and overlap graph, and we also recall the notion of realistic string.

We fix $\kappa \geq 2$, and define the alphabet $\Delta = \{2, 3, \ldots, \kappa\}$. For $D \subseteq \Delta$, we define $\bar{D} = \{\bar{a} \mid a \in D\}$ and $\Pi_D = D \cup \bar{D}$; also $\Pi = \Pi_\Delta$. The elements of Π are called *pointers*. We use the "bar operator" to move from Δ to $\bar{\Delta}$ and back from $\bar{\Delta}$ to Δ. Hence, for $p \in \Pi$, $\bar{\bar{p}} = p$. For $p \in \Pi$, we define $\mathbf{p}$ to be p if $p \in \Delta$, and $\bar{p}$ if $p \in \bar{\Delta}$, i.e., $\mathbf{p}$ is the "unbarred" variant of p. For a string $u = x_1 x_2 \cdots x_n$ with $x_i \in \Pi$ $(1 \leq i \leq n)$, the *complement of* u is $\bar{x}_1 \bar{x}_2 \cdots \bar{x}_n$. The *inverse of* u, denoted by $\bar{u}$, is the complement of the reversal of u, thus $\bar{u} = \bar{x}_n \bar{x}_{n-1} \cdots \bar{x}_1$. The *domain of* u, denoted by $\mathrm{dom}(u)$, is $\{\mathbf{p} \mid p \text{ occurs in } v\}$. We say that u is a *legal string* if for each $p \in \mathrm{dom}(u)$, u contains exactly two occurrences from $\{p, \bar{p}\}$. For a pointer p and a legal string u, if both p and $\bar{p}$ occur in u then we say that both p and $\bar{p}$ are *positive in* u; if on the other hand only p or only $\bar{p}$ occurs in u, then both p and $\bar{p}$ are *negative in* u. So, every pointer occurring in a legal string is either positive or negative in it. Therefore, we can define a partition of $\mathrm{dom}(u) = \mathrm{pos}(u) \cup \mathrm{neg}(u)$, where $\mathrm{pos}(u) = \{p \in \mathrm{dom}(u) \mid p \text{ is positive in } u\}$ and $\mathrm{neg}(u) = \{p \in \mathrm{dom}(u) \mid p \text{ is negative in } u\}$.

Let $u = x_1 x_2 \cdots x_n$ be a legal string with $x_i \in \Pi$ for $1 \leq i \leq n$. For a pointer $p \in \Pi$, the *p-interval of* u is the substring $x_i x_{i+1} \cdots x_j$ with $\{x_i, x_j\} \subseteq \{p, \bar{p}\}$ and $1 \leq i < j \leq n$. Substrings $x_{i_1} \cdots x_{j_1}$ and $x_{i_2} \cdots x_{j_2}$ *overlap in* u if $i_1 < i_2 < j_1 < j_2$ or $i_2 < i_1 < j_2 < j_1$. Two distinct pointers $p, q \in \Pi$ *overlap in* u if the p-interval of u overlaps with the q-interval of u. Thus, two distinct pointers $p, q \in \Pi$ overlap in u iff there is exactly one occurrence from $\{p, \bar{p}\}$ in the q-interval, or equivalently, there is exactly one occurrence from $\{q, \bar{q}\}$ in the p-interval of u. Also, for $p \in \mathrm{dom}(u)$, we denote the set of all $q \in \mathrm{dom}(u)$ such that p and q overlap in u by $O_u(p)$, and for $0 \leq i \leq j \leq n$, we denote by $O_u(i, j)$ the set of all $p \in \mathrm{dom}(u)$ such that there is exactly one occurrence from $\{p, \bar{p}\}$ in $x_{i+1} x_{i+2} \cdots x_j$. Also, we define $O_u(j, i) = O_u(i, j)$. Intuitively, $O_u(i, j)$ is the set of $p \in \mathrm{dom}(u)$ for which the substring between "positions" i and j in u contains exactly one representative

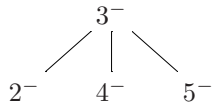

Figure 5.1: The overlap graph of legal string $u = 24535423$.

from $\{p, \bar{p}\}$, where position i for $0 < i < n$ means the "space" between x_i and x_{i+1} in u. For $i = 0$ it is the "space" to the left of x_1, and for $i = n$ it is the "space" to the right of x_n. A few elementary properties of $O_u(i, j)$ follow. We have $O_u(i, n) = O_u(0, i)$ for i with $0 \leq i \leq n$. Moreover, for $i, j, k \in \{0, \ldots, n\}$, $O_u(i, j) \oplus O_u(j, k) = O_u(i, k)$; this is obvious when $i < j < k$, but it is valid in general. Also, for $0 \leq i \leq j \leq n$, $O_u(i, j) = \varnothing$ iff $x_{i+1} \cdots x_j$ is a legal string.

Definition 1
Let u be a legal string. The *overlap graph of* u, denoted by γ_u, is the labelled graph $(\mathrm{dom}(u), E, \sigma, \{+, -\})$, where for $p, q \in \mathrm{dom}(u)$ with $p \neq q$, $\{p, q\} \in E$ iff p and q overlap in u, and σ is defined by: $\sigma(p) = +$ if $p \in \mathrm{pos}(u)$, and $\sigma(p) = -$ if $p \in \mathrm{neg}(u)$. ■

Example 1
Let $u = 24535423$ be a legal string. The overlap graph of u is

$$\gamma = (\{2, 3, 4, 5\}, \{\{2, 3\}, \{4, 3\}, \{5, 3\}\}, \sigma, \{+, -\}),$$

where $\sigma(v) = -$ for all vertices v of γ. The overlap graph is depicted in Figure 5.1.

Let γ be the overlap graph of a legal string u. We define $\mathrm{dom}(\gamma)$ as the set of vertices of γ, $\mathrm{pos}(\gamma) = \{p \in \mathrm{dom}(\gamma) \mid \sigma(p) = +\}$, $\mathrm{neg}(\gamma) = \{p \in \mathrm{dom}(\gamma) \mid \sigma(p) = -\}$, and for $q \in \mathrm{dom}(u)$, $O_\gamma(q) = \{p \in \mathrm{dom}(\gamma) \mid \{p, q\} \in E\}$. We have $\mathrm{dom}(\gamma) = \mathrm{dom}(u)$, $\mathrm{pos}(\gamma) = \mathrm{pos}(u)$, $\mathrm{neg}(\gamma) = \mathrm{neg}(u)$, and $O_\gamma(q) = O_u(q)$ for all $q \in \mathrm{dom}(\gamma) = \mathrm{dom}(u)$.

We define the alphabet $\Theta_\kappa = \{M_i, \bar{M}_i \mid 1 \leq i \leq \kappa\}$, and say that $\delta \in \Theta_\kappa^*$ is a *micronuclear arrangement* if for each i with $1 \leq i \leq \kappa$, δ contains exactly one occurrence from $\{M_i, \bar{M}_i\}$. With each string over Θ_κ, we associate a unique string over Π through the homomorphism $\pi_\kappa : \Theta_\kappa^* \to \Pi^*$ defined by: $\pi_\kappa(M_1) = 2$, $\pi_\kappa(M_\kappa) = \kappa$, $\pi_\kappa(M_i) = i(i+1)$ for $1 < i < \kappa$, and $\pi_\kappa(\bar{M}_j) = \overline{\pi_\kappa(M_j)}$ for $1 \leq j \leq \kappa$. A string u is a *realistic string* if there is a micronuclear arrangement δ such that $u = \pi_\kappa(\delta)$. We then say that δ is a *micronuclear arrangement for* u.

Note that every realistic string is a legal string. However, not every legal string is a realistic string. For example, a realistic string cannot have "gaps" (missing pointers): thus 2244 is not realistic while it is legal. It is also easy to produce examples of legal strings which do not have gaps but still are not realistic — 3322 is such an example. Realistic strings are most useful for the gene assembly models, since only these legal strings can correspond to genes in ciliates.

An overlap graph γ is called *realistic* if it is the overlap graph of a realistic string. Not every overlap graph of a legal string is realistic. For example, it can be shown that the overlap graph γ of $u = 24535423$ depicted in Figure 5.1 is not realistic. In fact, one can show that it is not even *realizable* — there is no isomorphism α such that $\alpha(\gamma)$ is realistic.

5.4 The Reduction Graph

We now recall the notion of a (full) reduction graph, which was first introduced in [6].

Remark
Below we present the notion of reduction graph in a slightly modified form: we omit the special vertices s and t, called the source vertex and target vertex respectively, which did appear in the definition presented in [6]. As shown in Section 5.5, in this way a realistic overlap graph corresponds to exactly one reduction graph. Fortunately, several results concerning reduction graphs do not rely on the special vertices, and therefore carry over in a straightforward way to reduction graphs as defined here.

Definition 2
Let $u = p_1 p_2 \cdots p_n$ with $p_1, \ldots, p_n \in \Pi$ be a legal string. The *reduction graph of* u, denoted by $\mathcal{R}_u$, is the 2-edge coloured graph

$$(V, E_1, E_2, f, \mathrm{dom}(u)),$$

where

$$V = \{I_1, I_2, \ldots, I_n\} \cup \{I'_1, I'_2, \ldots, I'_n\},$$

$$E_1 = \{e_1, e_2, \ldots, e_n\} \text{ with } e_i = \{I'_i, I_{i+1}\} \text{ for } 1 \leq i \leq n-1, e_n = \{I'_n, I_1\},$$

$$\begin{aligned} E_2 = \quad & \{\{I'_i, I_j\}, \{I_i, I'_j\} \mid i, j \in \{1, 2, \ldots, n\} \text{ with } i \neq j \text{ and } p_i = p_j\} \cup \\ & \{\{I_i, I_j\}, \{I'_i, I'_j\} \mid i, j \in \{1, 2, \ldots, n\} \text{ and } p_i = \bar{p}_j\}, \text{ and} \end{aligned}$$

$$f(I_i) = f(I'_i) = \mathbf{p}_i \text{ for } 1 \leq i \leq n.$$

■

The edges of E_1 are called the *reality edges*, and the edges of E_2 are called the *desire edges*. Intuitively, the "space" between p_i and p_{i+1} corresponds to the reality edge $e_i = \{I'_i, I_{i+1}\}$. Hence, we say that i is the *position of* e_i, denoted by $\mathrm{posn}(e_i)$, for all $i \in \{1, 2, \ldots, n\}$. Note that positions are only defined for reality edges. Since for every vertex v there is a unique reality edge e such that $v \in e$, we also define the *position of* v, denoted by $\mathrm{posn}(v)$, as the position of e. Thus, $\mathrm{posn}(I'_i) = \mathrm{posn}(I_{i+1}) = i$ (while $\mathrm{posn}(I_1) = n$).

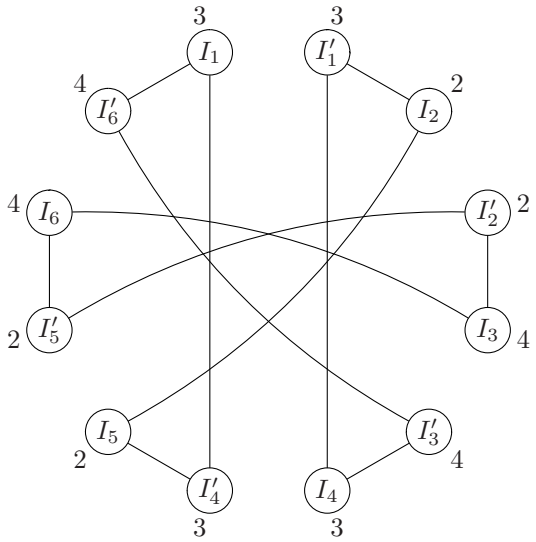

Figure 5.2: The reduction graph of u from Example 2.

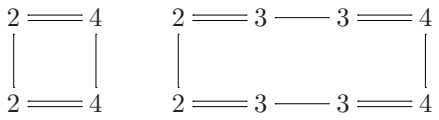

Figure 5.3: The reduction graph of u from Example 2 in the simplified representation.

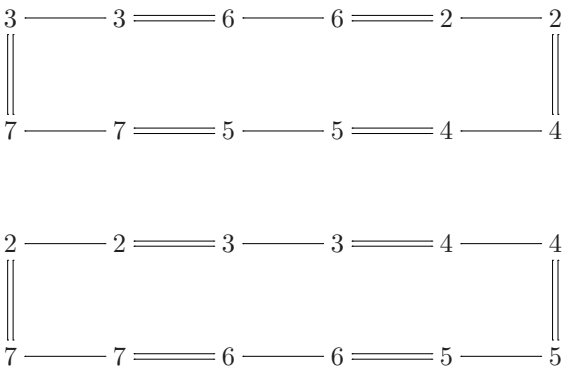

Figure 5.4: The reduction graph of u from Example 3.

Example 2
Let $u = 32\bar{4}3\bar{2}4$ be a legal string. Since $\bar{4}3\bar{2}$ can not be a substring of a realistic string, u is not realistic. The reduction graph $\mathcal{R}_u$ of u is depicted in Figure 5.2. The labels of the vertices are also shown in this figure. Note that the desire edges corresponding to positive pointers (here 2 and 4) cross (in the figure), while those for negative pointers are parallel.

We consider reduction graphs up to isomorphism. Therefore, the exact identity of the vertices in a reduction graph is not essential for the problems considered in this chapter, and in pictorial representations of reduction graphs we denote the vertices by their labels. We also depict reality edges by double line segments to distinguish them from the desire edges. Figure 5.3 shows the reduction graph of Figure 5.2 in this simplified representation.

Example 3
Let $u = \pi_7(M_7M_1M_6M_3M_5\overline{M_2}M_4) = 7267345\bar{6}\bar{3}\bar{2}45$. Thus, unlike in the previous example, u is a realistic string. The reduction graph is given in Figure 5.4. Note that according to our convention, the vertices are represented by their labels.

The reduction graph is defined for legal strings. In this chapter, we show how to directly construct the reduction graph of a realistic string from its overlap graph. In this way we can define the reduction graph for realistic overlap graphs in a direct way.

Next we consider sets of overlapping pointers corresponding to pairs of vertices in reduction graphs, and we begin to develop a calculus for these sets that will later enable us to characterize the existence of certain edges in the reduction graph, cf. Theorem 15.

Lemma 3
Let u be a legal string. Let $e = \{v_1, v_2\}$ be a desire edge of $\mathcal{R}_u$ and let p be the

label of both v_1 and v_2. Then

$$O_u(\mathrm{posn}(v_1), \mathrm{posn}(v_2)) = \begin{cases} O_u(p) & \text{if } p \text{ is negative in } u, \\ O_u(p) \oplus \{p\} & \text{if } p \text{ is positive in } u. \end{cases}$$

Proof
Let $u = p_1 p_2 \dots p_n$ with $p_1, p_2, \dots, p_n \in \Pi$ and let i and j be such that $i < j$ and $p = p_i = p_j$. Without loss of generality, we can assume $\mathrm{posn}(v_1) < \mathrm{posn}(v_2)$. Then, $v_1 \in \{I_i, I_i'\}$ and $v_2 \in \{I_j, I_j'\}$, hence $\mathrm{posn}(v_1) \in \{i-1, i\}$ and $\mathrm{posn}(v_2) \in \{j-1, j\}$.

First, assume that p is negative in u. Then, by the definition of reduction graph, the following two cases are possible:

1. $e = \{I_i, I_j'\}$, thus $O_u(\mathrm{posn}(I_i), \mathrm{posn}(I_j')) = O_u(i-1, j) = O_u(p)$,
2. $e = \{I_i', I_j\}$, thus $O_u(\mathrm{posn}(I_i), \mathrm{posn}(I_{j-1}')) = O_u(i, j-1) = O_u(p)$,

Thus in both cases we have $O_u(\mathrm{posn}(v_1), \mathrm{posn}(v_2)) = O_u(p)$.

Now, assume that p is positive in u. Then, by the definition of reduction graph, the following two cases are possible:

1. $e = \{I_i, I_j\}$, thus $O_u(\mathrm{posn}(I_i), \mathrm{posn}(I_j)) = O_u(i-1, j-1) = O_u(p) \oplus \{p\}$,
2. $e = \{I_i', I_j'\}$, thus $O_u(\mathrm{posn}(I_i'), \mathrm{posn}(I_j')) = O_u(i, j) = O_u(p) \oplus \{p\}$,

Thus in both cases we have $O_u(i_1, i_2) = O_u(p) \oplus \{p\}$. ∎

Let u be a legal string. For $P \subseteq \mathrm{dom}(u)$, we define $\Pi_u(P) = (\mathrm{pos}(u) \cap P) \oplus \left(\bigoplus_{t \in P} O_u(t)\right)$. Similarly, we define $\Pi_\gamma(P)$ for an overlap graph γ (by replacing u by γ in the definition).

The following result follows by iteratively applying Lemma 3 and using the definition of $\Pi_u(P)$.

Corollary 4
Let u be a legal string. Let

$$p_0 \;\overline{=}\; p_1 \;\text{—}\; p_1 \;\overline{=}\; p_2 \;\text{—}\; p_2 \;\overline{=}\; \cdot\cdot \;\overline{=}\; p_n \;\text{—}\; p_n \;\overline{=}\; p_{n+1}$$

be a subgraph of $\mathcal{R}_u$, and let e_1 (e_2, resp.) be the leftmost (rightmost, resp.) edge. Then $O_u(\mathrm{posn}(e_1), \mathrm{posn}(e_2)) = \Pi_u(P)$ with $P = \{p_1, \dots, p_n\}$.

Note that, in the above, e_1 and e_2 are reality edges and therefore $\mathrm{posn}(e_1)$ and $\mathrm{posn}(e_2)$ are defined.

By the definition of reduction graph the following lemma holds.

Lemma 5
Let u be a legal string. If I_i and I_i' are vertices of $\mathcal{R}_u$, then $O_u(\mathrm{posn}(I_i), \mathrm{posn}(I_i')) = \{p\}$, where p is the label of I_i and I_i'.

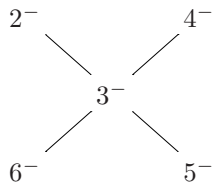

Figure 5.5: The overlap graph of both legal strings u and v from Example 5.

Example 4
We again consider the legal string $u = 32\bar{4}3\bar{2}4$ and its reduction graph $\mathcal{R}_u$ from Example 2. Desire edge $e = \{I'_2, I'_5\}$ with vertices labelled by 2 is connected to reality edges $\{I'_2, I_3\}$ and $\{I'_5, I_6\}$ with positions 2 and 5 respectively. By Lemma 3, we have $O_u(2,5) = O_u(2) \oplus \{2\} = \{2,3,4\}$. This can of course also be verified by directly considering the corresponding substring $\bar{4}3\bar{2}$ between positions 2 and 5 of u. Also, since I_2 and I'_2 with positions 1 and 2 respectively are labelled by 2, by Lemma 5 we have $O_u(1,2) = \{2\}$.

5.5 The Reduction Graph of Realistic Strings

The next theorem asserts that the overlap graph γ for a realistic string u retains all information of $\mathcal{R}_u$ (up to isomorphism). In this chapter, we give a method to determine $\mathcal{R}_u$ (up to isomorphism), from γ. Of course, the naive method is to first determine a legal string u corresponding to γ and then to determine the reduction graph of u. However, we present a method that allows one to construct $\mathcal{R}_u$ in a direct way from γ.

Theorem 6
Let u and v be realistic strings. If $\gamma_u = \gamma_v$, then $\mathcal{R}_u \approx \mathcal{R}_v$.

Proof
By Theorem 1 in [19] (or Theorem 10.2 in [12]), we have $\gamma_u = \gamma_v$ iff v can be obtained from u by a composition of reversal, complement and conjugation operations. By the definition of reduction graph it is clear that the reduction graph is invariant under these operations (up to isomorphism). Thus, $\mathcal{R}_u \approx \mathcal{R}_v$.■

This theorem does *not* hold for legal strings in general — the next two examples illustrate that legal strings having the same overlap graph can have different reduction graphs up to isomorphism.

Example 5
Let $u = 2653562434$ and $v = h(u)$, where h is the homomorphism that interchanges 5 and 6. Thus, $v = 2563652434$. Note that both u and v are not realistic, because substrings 535 of u and 636 of v can obviously not be substrings of realistic strings. The overlap graph of u is depicted in Figure 5.5. From Figure 5.5

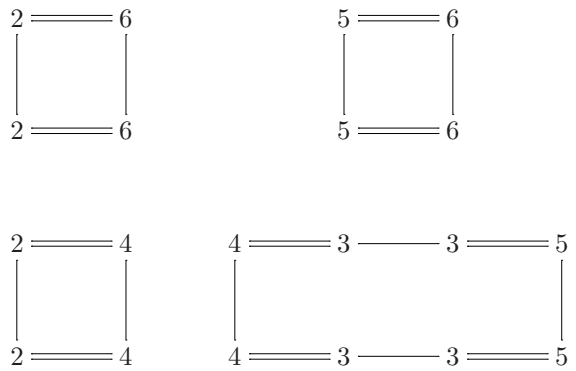

Figure 5.6: The reduction graph of u from Example 5.

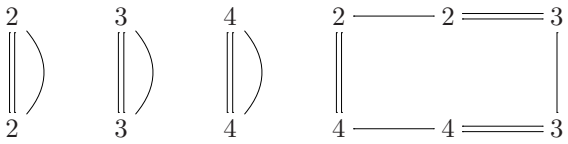

Figure 5.7: The reduction graph of u from Example 6.

and the fact that v is obtained from u by renumbering 5 and 6, it follows that the overlap graphs of u and v are equal. The reduction graph $\mathcal{R}_u$ of u is depicted in Figure 5.6. The reduction graph $\mathcal{R}_v$ of v is obtained from $\mathcal{R}_u$ by renumbering the labels of the vertices according to h. Clearly, $\mathcal{R}_u \not\approx \mathcal{R}_v$.

Example 6
Let $u = \pi_\kappa(M_1M_2M_3M_4) = 223344$ be a realistic string and let $v = 234432$ be a legal string. Note that v is not realistic. Legal strings u and v have the same overlap graph γ ($\gamma = (\{2,3,4\}, \varnothing, \sigma, \{+,-\})$, where $\sigma(v) = -$ for $v \in \{2,3,4\}$). The reduction graph $\mathcal{R}_u$ of u is depicted in Figure 5.7, and the reduction graph $\mathcal{R}_v$ of v is depicted in Figure 5.8. Note that $\mathcal{R}_u$ has a component consisting of six vertices, while $\mathcal{R}_v$ does not have such a component. Therefore, $\mathcal{R}_u \not\approx \mathcal{R}_v$.

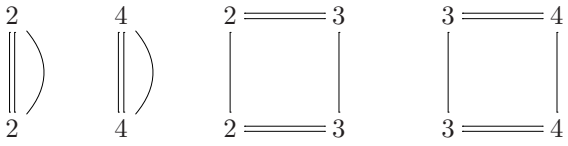

Figure 5.8: The reduction graph of v from Example 6.

For realistic strings the reduction graph has a special form. This is seen as follows. For $1 < i < \kappa$ the symbol M_i (or $\bar{M}_i$) in the micronuclear arrangement defines two pointers p_i and p_{i+1} (or $\bar{p}_{i+1}$ and $\bar{p}_i$) in the corresponding realistic string u. At the same time the substring $p_i p_{i+1}$ (or $\bar{p}_{i+1}\bar{p}_i$, respectively) of u corresponding to M_i (or $\bar{M}_i$, respectively) defines four vertices $I_j, I'_j, I_{j+1}, I'_{j+1}$ in $\mathcal{R}_u$. It is easily verified (cf. Theorem 8 below) that the "middle" two vertices I'_j and I_{j+1}, labelled by p_i and p_{i+1} respectively, are connected by a reality edge and I'_j (I_{j+1}, respectively) is connected by a desire edge to a "middle vertex" resulting from M_{i-1} or $\bar{M}_{i-1}$ (M_{i+1} or $\bar{M}_{i+1}$, respectively). This leads to the following definition.

Definition 7
Let u be a legal string and let $\kappa = |\mathrm{dom}(u)| + 1$. If $\mathcal{R}_u$ contains a subgraph L of the following form:

$$2 \text{ —— } 2 \text{ ═══ } 3 \text{ —— } 3 \text{ ═══ } \cdot\cdot \text{ ═══ } \kappa \text{ —— } \kappa$$

where the vertices in the figure are represented by their labels, then we say that u is *rooted* and L is called a *root subgraph of* $\mathcal{R}_u$. ∎

Example 7
The realistic string u with $\mathrm{dom}(u) = \{2, 3, \ldots, 7\}$ from Example 3 is rooted because the reduction graph of u, depicted in Figure 5.4, contains the subgraph

$$2 \text{ —— } 2 \text{ ═══ } 3 \text{ —— } 3 \text{ ═══ } \cdot\cdot \text{ ═══ } 7 \text{ —— } 7$$

The next theorem shows that indeed every realistic string is rooted.

Theorem 8
Every realistic string is rooted.

Proof
Consider a micronuclear arrangement for a realistic string u. Let $\kappa = |\mathrm{dom}(u)|+1$. By the definition of π_κ, there is a reality edge e_i (corresponding to either $\pi_\kappa(M_i) = i(i+1)$ or $\pi_\kappa(\overline{M_i}) = \overline{(i+1)\,\bar{i}}$) connecting a vertex labelled by i to a vertex labelled by $i+1$ for each $2 \leq i < \kappa$. It suffices to prove that there is a desire edge connecting e_i to e_{i+1} for each $2 \leq i < \kappa - 1$. This can easily be seen by checking the four cases where e_i corresponds to either $\pi_\kappa(M_i)$ or $\pi_\kappa(\overline{M_i})$, and e_{i+1} corresponds to either $\pi_\kappa(M_{i+1})$ or $\pi_\kappa(\overline{M_{i+1}})$. ∎

In the remainder of this chapter, we denote $|\mathrm{dom}(u)| + 1$ just by κ for rooted strings, whenever the rooted string u is understood from the context of considerations. The reduction graph of a realistic string may have more than one root subgraph: it is easy to verify that realistic string $234\cdots\kappa234\cdots\kappa$ for $\kappa \geq 2$ has two root subgraphs.

Example 2 shows that not every rooted string is realistic. The results in the remainder of this chapter that consider realistic strings also hold for rooted strings,

since we will not be using any properties of realistic string that are not true for rooted strings in general.

For a given root subgraph L, it is convenient to uniquely identify every reality edge containing a vertex of L. This is done through the following definition.

Definition 9
Let u be a rooted string and let L be a root subgraph of $\mathcal{R}_u$. We define $rspos_{L,k}$ for $2 \leq k < \kappa$ as the position of the edge of L that has vertices labelled by k and $k+1$. We define $rspos_{L,1}$ ($rspos_{L,\kappa}$, resp.) as the position of the edge of $\mathcal{R}_u$ not in L containing a vertex of L labelled by 2 (κ, resp.). When $\kappa = 2$, to ensure that $rspos_{L,1}$ and $rspos_{L,\kappa}$ are well defined, we additionally require that $rspos_{L,1} < rspos_{L,\kappa}$. ∎

Thus, $rspos_{L,k}$ (for $1 \leq k \leq \kappa$) uniquely identifies every reality edge containing a vertex of L. If it is clear which root subgraph L is meant, we simply write $rspos_k$ instead of $rspos_{L,k}$ for $1 \leq k \leq \kappa$.

The next lemma is essential to prove the main result (Theorem 15) of this chapter.

Lemma 10
Let u be a rooted string. Let L be a root subgraph of $\mathcal{R}_u$. Let i and j be positions of reality edges in $\mathcal{R}_u$ that are not edges of L. Then $O_u(i,j) = \varnothing$ iff $i = j$.

Proof
The reverse implication is trivially satisfied. We now prove the forward implication. The reality edge e_k (for $2 \leq k < \kappa$) in L with vertices labelled by k and $k+1$ corresponds to a cyclic substring $\tilde{M}_k \in \{p_1p_2, p_2p_1 \mid p_1 \in \{k, \overline{k}\}, p_2 \in \{k+1, \overline{k+1}\}\}$ of u. Let k_1 and k_2 with $2 \leq k_1 < k_2 < \kappa$. If $k_1 + 1 = k_2$, then reality edges e_{k_1} and e_{k_2} are connected by a desire edge (by the definition of L). Therefore, pointer k_2 common in $\tilde{M}_{k_1}$ and $\tilde{M}_{k_2}$ originates from two different occurrences in u. If on the other hand $k_1 + 1 \neq k_2$, then $\tilde{M}_{k_1}$ and $\tilde{M}_{k_2}$ do not have a letter in common. Therefore, in both cases, $\tilde{M}_{k_1}$ and $\tilde{M}_{k_2}$ are disjoint cyclic substrings of u. Thus the $\tilde{M}_k$ for $2 \leq k < \kappa$ are pairwise disjoint cyclic substrings of u.

Without loss of generality assume $i \leq j$. Let $u = u_1u_2 \cdots u_n$ with $u_i \in \Pi$. Since u is a legal string, every u_l for $1 \leq l \leq n$ is either part of a $\tilde{M}_k$ (with $2 \leq k < \kappa$) or in $\{2, \bar{2}, \kappa, \bar{\kappa}\}$. Consider $u' = u_{i+1}u_{i+2} \cdots u_j$. Since i and j are positions of reality edges in $\mathcal{R}_u$ that are not edges of L, we have $u' = \tilde{M}_{k_1}\tilde{M}_{k_2} \cdots \tilde{M}_{k_m}$ for some distinct $k_1, k_2, \ldots, k_m \in \{1, 2, \ldots, \kappa\}$, where $\tilde{M}_1 \in \{2, \bar{2}\}$ and $\tilde{M}_\kappa \in \{\kappa, \bar{\kappa}\}$.

It suffices to prove that $u' = \lambda$. Assume to the contrary that $u' \neq \lambda$. Then there is a $1 \leq l \leq \kappa$ such that $\tilde{M}_l$ is a substring of u'. Because $O_u(i,j) = \varnothing$, we know that u' is legal. If $l > 1$, then $\tilde{M}_{l-1}$ is also a substring of u', otherwise u' would not be a legal string. Similarly, if $l < \kappa$, then $\tilde{M}_{l+1}$ is also a substring of u'. By iteration, we conclude that $u' = u$. Therefore, $i = 0$. This is a contradiction, since 0 cannot be a position of a reality edge. Thus, $u' = \lambda$. ∎

Lemma 11
Let u be a rooted string. Let L be a root subgraph of $\mathcal{R}_u$. If I_i and I'_i are vertices of $\mathcal{R}_u$, then exactly one of I_i and I'_i belongs to L.

Proof
By the definition of reduction graph, I_i and I'_i have a common vertex label p but are not connected by a desire edge. Therefore, I_i and I'_i do not both belong to L. Now, if I_i and I'_i both do not belong to L, then the other vertices labelled by p, which are I_j and I'_j for some j, both belong to L – a contradiction by the previous argument. Therefore, either I_i or I'_i belongs to L, and the other one does not belong to L. ∎

The following result captures the main idea that allows for the determination of the reduction graph from the overlap graph only. It relies heavily on the previous lemmas.

Very roughly, the intuition is that there is a reality edge with vertices labelled by p and q outside a fixed root subgraph L precisely when: we can make a "sidestep over" p in the underlying string u "into" L and then "walk over" L to q and finally make a sidestep over q in u in such a way that the accumulated overlap is the empty set.

Theorem 12
Let u be a rooted string, let L be a root subgraph of $\mathcal{R}_u$, and let $p, q \in \mathrm{dom}(u)$ with $p < q$. There is a reality edge e in $\mathcal{R}_u$ with both vertices not in L, one labelled by p and the other by q iff $\Pi_u(P) = \{p, q\}$ where $P = \{p+1, \ldots, q-1\} \cup P'$ for some $P' \subseteq \{p, q\}$.

Proof
We first prove the forward implication. Let $e = \{v_1, v_2\}$ with v_1 labelled by p, v_2 labelled by q, and $\mathrm{posn}(e) = i$. Thus $e = \{I'_i, I_{i+1}\}$. We assume that $v_1 = I'_i$ and $v_2 = I_{i+1}$, the other case is proved similarly. Let $i_1 = \mathrm{posn}(I_i)$ and $i_2 = \mathrm{posn}(I'_{i+1})$. By Lemma 5, $O_u(i, i_1) = \{p\}$ and $O_u(i_2, i) = \{q\}$. By Lemma 11, I_i (labelled by p) and I'_{i+1} (labelled by q) belong to L. Thus $i_1 \in \{\mathit{rspos}_{p-1}, \mathit{rspos}_p\}$ and $i_2 \in \{\mathit{rspos}_{q-1}, \mathit{rspos}_q\}$. By applying Corollary 4 on L, we have $O_u(i_1, i_2) = \Pi_u(P)$ with $P = \{p+1, \ldots, q-1\} \cup P'$ for some $P' \subseteq \{p, q\}$. By definition of $O_u(i, j)$ we have

$$\varnothing = O_u(i, i) = O_u(i, i_1) \oplus O_u(i_1, i_2) \oplus O_u(i_2, i)$$

Since $p \neq q$, we have $\{p\} \oplus \{q\} = \{p, q\}$, and the desired result follows.

We now prove the reverse implication. By applying Corollary 4 on L, we have $O_u(i_1, i_2) = \Pi_u(P)$ for some $i_1 \in \{\mathit{rspos}_{p-1}, \mathit{rspos}_p\}$ and $i_2 \in \{\mathit{rspos}_{q-1}, \mathit{rspos}_q\}$ (depending on P'). By Lemma 5, there is a vertex v_1 (v_2, resp.) labelled by p (q, resp.) with position i (j, resp.) such that $O_u(i, i_1) = \{p\}$ and $O_u(i_2, j) = \{q\}$. By Lemma 11 these vertices are not in L. We have now

$$\varnothing = O_u(i, i_1) \oplus O_u(i_1, i_2) \oplus O_u(i_2, j) = O_u(i, j)$$

By Lemma 10, $O_u(i,j) = \varnothing$ implies that $i = j$. Thus, there is a reality edge $\{v_1, v_2\}$ in $\mathcal{R}_u$ (with position i), such that v_1 is labelled by p and v_2 is labelled by q and both are not vertices of L. ∎

Let γ_u be the overlap graph of some legal string u. Clearly we have $\text{pos}(u) = \text{pos}(\gamma_u)$ and for all $p \in \text{dom}(u) = \text{dom}(\gamma_u)$, $O_u(p) = O_{\gamma_u}(p)$. Thus by Theorem 12 we can determine, given the overlap graph of a rooted string u, if there is a reality edge in $\mathcal{R}_u$ with both vertices outside L that connects a vertex labelled by p to a vertex labelled by q. We will extend this result to completely determine the reduction graph given the overlap graph of a rooted string (or a realistic string in particular).

5.6 Compressing the Reduction Graph

In this section we define the cps function. The cps function simplifies reduction graphs by replacing the subgraph p —— p by a single vertex labelled by p. In this way, one can simplify reduction graphs without "losing information". We define cps for a general family of graphs $\mathcal{G}$ which includes all reduction graphs. The formal definitions of $\mathcal{G}$ and cps are given below.

Let $\mathcal{G}$ be the set of 2-edge coloured graphs $G = (V, E_1, E_2, f, \Gamma)$ such that $f(v_1) = f(v_2)$ for all $\{v_1, v_2\} \in E_2$. Note that for a reduction graph $\mathcal{R}_u$, we have $\mathcal{R}_u \in \mathcal{G}$ because both vertices of a desire edge have the same label. For all $G \in \mathcal{G}$, $\text{cps}(G)$ is obtained from G by considering the second set of edges as vertices in the labelled graph. Thus, for the case when G is a reduction graph, the function cps "compresses" the desire edges to vertices.

Definition 13
The function cps from $\mathcal{G}$ to the set of labelled graphs is defined as follows. If $G = (V, E_1, E_2, f, \Gamma) \in \mathcal{G}$, then

$$\text{cps}(G) = (E_2, E_1', f', \Gamma)$$

is a labelled graph, where

$$E_1' = \{\{e_1, e_2\} \subseteq E_2 \mid \exists v_1, v_2 \in V : v_1 \in e_1, v_2 \in e_2, e_1 \neq e_2 \text{ and } \{v_1, v_2\} \in E_1\},$$

and for $e \in E_2$: $f'(e) = f(v)$ with $v \in e$. ∎

Note that f' is well defined, because for all $\{v_1, v_2\} \in E_2$, it holds that $f(v_1) = f(v_2)$.

Example 8
We are again considering the realistic string u defined in Example 3. The reduction graph of $\mathcal{R}_u$ is depicted in Figure 5.4. The labelled graph $\text{cps}(\mathcal{R}_u)$ is depicted in Figure 5.9. Since this graph has just one set of edges, the reality edges are depicted by single line segments rather than double line segments as we did for reduction graphs.

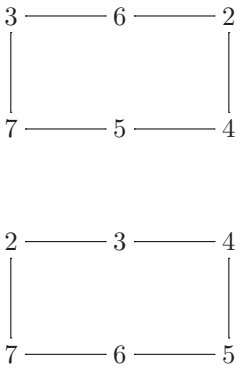

Figure 5.9: The labelled graph $\text{cps}(\mathcal{R}_u)$, where $\mathcal{R}_u$ is defined in Example 8.

It is not hard to see that for reduction graphs $\mathcal{R}_u$ and $\mathcal{R}_v$, we have $\mathcal{R}_u \approx \mathcal{R}_v$ iff $\text{cps}(\mathcal{R}_u) \approx \text{cps}(\mathcal{R}_v)$. In this sense, the cps function allows one to simplify reduction graphs without losing information.

5.7 From Overlap Graph to Reduction Graph

In this section we define (compressed) reduction graphs for realistic overlap graphs, inspired by the characterization from Theorem 12, and then demonstrate their equivalence to reduction graphs for realistic strings.

Definition 14
Let $\gamma = (Dom_\gamma, E_\gamma, \sigma, \{+, -\})$ be a realistic overlap graph and let $\kappa = |Dom_\gamma|+1$. The *reduction graph of* γ, denoted by $\mathcal{R}_\gamma$, is a labelled graph

$$\mathcal{R}_\gamma = (V, E, f, Dom_\gamma),$$

where

$$V = \{J_p, J'_p \mid 2 \leq p \leq \kappa\},$$

$$f(J_p) = f(J'_p) = p, \text{ for } 2 \leq p \leq \kappa, \text{ and}$$

$e \in E$ iff one of the following conditions hold:

1. $e = \{J'_p, J'_{p+1}\}$ and $2 \leq p < \kappa$.
2. $e = \{J_p, J_q\}$, $2 \leq p < q \leq \kappa$, and $\Pi_\gamma(P) = \{p, q\}$, where $P = \{p+1, \ldots, q-1\} \cup P'$ for some $P' \subseteq \{p, q\}$.
3. $e = \{J'_2, J_p\}$, $2 \leq p \leq \kappa$, and $\Pi_\gamma(P) = \{p\}$, where $P = \{2, \ldots, p-1\} \cup P'$ for some $P' \subseteq \{p\}$.

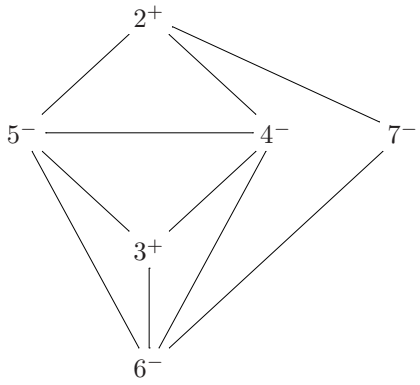

Figure 5.10: The overlap graph γ of a realistic string (used in Example 9).

4. $e = \{J'_\kappa, J_p\}$, $2 \leq p \leq \kappa$, and $\Pi_\gamma(P) = \{p\}$, where $P = \{p+1, \ldots, \kappa\} \cup P'$ for some $P' \subseteq \{p\}$.

5. $e = \{J'_2, J'_\kappa\}$, $\kappa > 3$, and $\Pi_\gamma(P) = \varnothing$, where $P = \{2, \ldots, \kappa\}$. ∎

An algorithm that constructs $\mathcal{R}_\gamma$ is efficiently implemented by observing that $\Pi_\gamma(P)$ only needs to be calculated for all intervals $P = [i, j] = \{i, \ldots, j\}$ with $i \leq j$ and $i, j \in \{2, \ldots, \kappa\}$. These values can be stored in an upper-triangular matrix $A = (a_{i,j})$ having $a_{i,j} = \Pi_\gamma([i, j])$. Note that A can be defined recursively as follows: we have $a_{i,j} = a_{i,j-1} \oplus a_{j,j}$ if $i < j$, and $a_{i,i} = O_\gamma(i)$ if i is negative in γ, and $a_{i,i} = O_\gamma(i) \oplus \{i\}$ if i is positive in γ. After calculating A, we can obtain the edges of $\mathcal{R}_\gamma$. If $2 < i$ and $j < \kappa$, then $a_{i,j} \in \{\{i, j\}, \{i-1, j\}, \{i, j+1\}, \{i-1, j+1\}\}$ iff there is an edge $e = \{J_p, J_q\}$ with $a_{i,j} = \{p, q\}$. The cases $2 = i$ or $j = \kappa$ are handled similarly.

Example 9
The overlap graph γ in Figure 5.10 is realistic. Indeed, realistic string $u = \pi_7(M_7 M_1 M_6 M_3 M_5 \overline{M_2} M_4) = 726734563\bar{3}\bar{2}45$ introduced in Example 3 has this overlap graph. Clearly, the reduction graph $\mathcal{R}_\gamma$ of γ has the edges $\{J'_p, J'_{p+1}\}$ for $2 \leq p < 7$.

Now to obtain the remaining edges, we construct the upper-triangular matrix $A = (a_{i,j})$ having $a_{i,j} = \Pi_\gamma([i, j])$ with $i \leq j$ and $i, j \in \{2, \ldots, \kappa\}$ (as discussed above, this can be done recursively). This matrix is given below, where the entries corresponding to edges of $\mathcal{R}_\gamma$ are underlined.

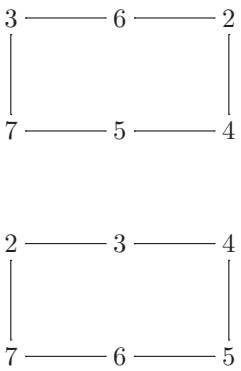

Figure 5.11: The reduction graph $\mathcal{R}_\gamma$ of the overlap graph γ from Example 9.

$\Pi_\gamma([i,j])$	2	3	4	5	6	7
2	$\{2,4,5,7\}$	$\{2,3,6,7\}$	$\{5,7\}$	$\{2,3,4,5,6,7\}$	$\underline{\{2,6\}}$	$\underline{\varnothing}$
3		$\{3,4,5,6\}$	$\underline{\{2,4\}}$	$\underline{\{3,6\}}$	$\{4,5,6,7\}$	$\{2,4,5,7\}$
4			$\{2,3,5,6\}$	$\underline{\{4,5\}}$	$\underline{\{3,7\}}$	$\{2,3,6,7\}$
5				$\{2,3,4,6\}$	$\{2,5,6,7\}$	$\underline{\{5,7\}}$
6					$\{3,4,5,7\}$	$\{2,3,4,5,6,7\}$
7						$\{2,6\}$

From matrix A we see that, $a_{2,7} = \varnothing$ corresponds to edge $\{J'_2, J'_7\}$, while the other underlined values $\{2,4\}$, $\{4,5\}$, $\{5,7\}$, $\{3,7\}$, $\{3,6\}$, and $\{2,6\}$ correspond to edges $\{J_2, J_4\}, \{J_4, J_5\}, \ldots, \{J_2, J_6\}$, respectively.

We have now completely determined $\mathcal{R}_\gamma$; it is shown in Figure 5.11 (again, as we have done for reduction graphs of legal strings, in the figures the vertices of reduction graphs of realistic overlap graphs are represented by their labels).

Figures 5.9 and 5.11 show that, for $u = 726734563\bar{3}\bar{2}45$, $\mathrm{cps}(\mathcal{R}_u) \approx \mathcal{R}_\gamma$. The next theorem shows that this is a general property for realistic strings u.

Theorem 15
Let u be a realistic string. Then, $\mathrm{cps}(\mathcal{R}_u) \approx \mathcal{R}_{\gamma_u}$.

Proof
Let $\kappa = |\mathrm{dom}(u)| + 1$, let $\gamma = \gamma_u$, let $\mathcal{R}_\gamma = (V_\gamma, E_\gamma, f_\gamma, \mathrm{dom}(\gamma))$, let $R_u = \mathrm{cps}(\mathcal{R}_u) = (V_u, E_u, f_u, \mathrm{dom}(u))$, and let L be a root subgraph of $\mathcal{R}_u$. Recall that the elements of V_u are the desire edges of $\mathcal{R}_u$.

Let $h : V_u \to V_\gamma$ defined by

$$h(v) = \begin{cases} J_{f_u(v)} & \text{if } v \text{ is not an edge of } L, \\ J'_{f_u(v)} & \text{if } v \text{ is an edge of } L. \end{cases}$$

We will show that h is an isomorphism from R_u to $\mathcal{R}_\gamma$. Since for every $l \in \mathrm{dom}(u)$ there exists exactly one desire edge v of $\mathcal{R}_u$ that belongs to L with $f_u(v) = l$ and there exists exactly one desire edge v of $\mathcal{R}_u$ that does not belong to L with $f_u(v) = l$, it follows that h is one-to-one and onto. Also, it is clear from the definition of f_γ that $f_u(v) = f_\gamma(h(v))$. Thus, it suffices to prove that $\{v_1, v_2\} \in E_u \Leftrightarrow \{h(v_1), h(v_2)\} \in E_\gamma$.

We first prove the forward implication $\{v_1, v_2\} \in E_u \Rightarrow \{h(v_1), h(v_2)\} \in E_\gamma$. Let $\{v_1, v_2\} \in E_u$, let $p = f_u(v_1)$ and let $q = f_u(v_2)$. Clearly, $v_1 \neq v_2$. By the definition of cps, there is a reality edge $\tilde{e} = \{\tilde{v}_1, \tilde{v}_2\}$ of $\mathcal{R}_u$ with $\tilde{v}_1 \in v_1$ and $\tilde{v}_2 \in v_2$ (and thus $\tilde{v}_1$ and $\tilde{v}_2$ are labelled by p and q in $\mathcal{R}_u$, respectively). Let i be the position of $\tilde{e}$. We consider four cases (remember that v_1 and v_2 are both desire edges of $\mathcal{R}_u$):

1. Assume that $\tilde{e}$ belongs to L. Then clearly, v_1 and v_2 are edges of L. Without loss of generality, we can assume that $p \leq q$. From the structure of root subgraph and the fact that $\tilde{e}$ is a reality edge of $\mathcal{R}_u$ in L, it follows that $q = p+1$. Now, $h(v_1) = J'_p$ and $h(v_2) = J'_q = J'_{p+1}$. By the first condition of Definition 14, it follows that $\{h(v_1), h(v_2)\} = \{J'_p, J'_{p+1}\} \in E_\gamma$. This proves the first case. In the remaining cases, $\tilde{e}$ does not belong to L.

2. Assume that v_1 and v_2 are both not edges of L (thus $\tilde{e}$ does not belong to L). Now by Theorem 12 and the second condition of Definition 14, it follows that $\{h(v_1), h(v_2)\} = \{J_p, J_q\} \in E_\gamma$. This proves the second case.

3. Assume that either v_1 or v_2 is an edge of L and that the other one is not an edge of L (thus $\tilde{e}$ does not belong to L). We follow the same line of reasoning as we did in Theorem 12. Without loss of generality, we can assume that v_1 is not an edge of L and that v_2 is an edge of L. Clearly,

$$\varnothing = O_u(i, i) = O_u(i, i_1) \oplus O_u(i_1, i)$$

   for each position i_1. By the structure of L we know that $q = 2$ or $q = \kappa$. Let $q = 2$ ($q = \kappa$, resp.). By Lemma 5 and Lemma 11, we can choose $i_1 \in \{\mathit{rspos}_{p-1}, \mathit{rspos}_p\}$ such that $O_u(i_1, i) = \{p\}$. By applying Corollary 4 to L, we get $O_u(i, i_1) = \Pi_u(P)$ with $P = \{2, \ldots, p-1\} \cup P'$ ($P = \{p+1, \ldots, \kappa\} \cup P'$, resp.) for some $P' \subseteq \{p\}$. By the third (fourth, resp.) condition of Definition 14, it follows that $\{h(v_1), h(v_2)\} = \{J'_2, J_q\} \in E_\gamma$ ($\{h(v_1), h(v_2)\} = \{J'_\kappa, J_q\} \in E_\gamma$, resp.). This proves the third case.

4. Assume that both v_1 and v_2 are edges of L, but $\tilde{e}$ does not belong to L. Again, we follow the same line of reasoning as we did in Theorem 12. Without loss of generality, we can assume that $p \leq q$. By the structure of L, we know that $p = 2$ and $q = \kappa > 3$. By applying Corollary 4 to L, we get $\varnothing = O_u(i, i) = \Pi_u(P)$ with $P = \{2, \ldots, \kappa\}$. By the fifth condition of Definition 14, it follows that $\{h(v_1), h(v_2)\} = \{J'_2, J'_\kappa\} \in E_\gamma$. This proves the last case.

This proves the forward implication.

We now prove the reverse implication $\{v_1, v_2\} \in E_\gamma \Rightarrow \{h^{-1}(v_1), h^{-1}(v_2)\} \in E_u$, where h^{-1}, the inverse of h, is given by:

$h^{-1}(J_p)$ is the unique $v \in V_u$ with $f_u(v) = p$ that is not an edge of L,
$h^{-1}(J'_p)$ is the unique $v \in V_u$ with $f_u(v) = p$ that is an edge of L,

for $2 \leq p \leq \kappa$. Let $e \in E_\gamma$. We consider each of the five types of edges in the definition of reduction graph of an overlap graph.

1. Assume e is of the first type. Then $e = \{J'_p, J'_{p+1}\}$ for some p with $2 \leq p < \kappa$. Since $h^{-1}(J'_p)$ is the desire edge of L with both vertices labelled by p and $h^{-1}(J'_{p+1})$ is the desire edge of L with both vertices labelled by $p+1$, it follows, by the definition of root subgraph, that $h^{-1}(J'_p)$ and $h^{-1}(J'_{p+1})$ are connected by a reality edge in L. Thus, we have $\{h^{-1}(J'_p), h^{-1}(J'_{p+1})\} \in E_u$. This proves the reverse implication when e is of the first type (in Definition 14).

2. Assume e is of the second type. Then $e = \{J_p, J_q\}$ for some p and q with $2 \leq p < q \leq \kappa$ and $\Pi_u(P) = \Pi_\gamma(P) = \{p, q\}$ with $P = \{p+1, \ldots, q-1\} \cup P'$ for some $P' \subseteq \{p, q\}$. By Theorem 12, there is a reality edge $\{w_1, w_2\}$ in $\mathcal{R}_u$, such that w_1 has label p and w_2 has label q and both are not vertices of L. By the definition of cps, we have a $\{w'_1, w'_2\} \in E_u$ such that $f_u(w'_1) = p$ ($f_u(w'_2) = q$, resp.) and w'_1 (w'_2, resp.) is not an edge of L. Therefore $w'_1 = h^{-1}(J_p)$ and $w'_2 = h^{-1}(J_q)$. This proves the reverse implication when e is of the second type.

3. The last three cases are proved similarly.

This proves the reverse implication.

Altogether, we have shown that h is an isomorphism from R_u to $\mathcal{R}_\gamma$. ∎

Example 10
Consider again realistic string $u = 726734563\bar{3}\bar{2}45$ from Example 9 (and Example 3). The reduction graph $\mathcal{R}_\gamma$ of the overlap graph of u is given in Figure 5.11. Recall that the reduction graph $\mathcal{R}_u$ of u is given in Figure 5.4. It is easy to see that after applying cps to $\mathcal{R}_u$ one obtains a graph that is indeed isomorphic to $\mathcal{R}_\gamma$.

Formally, we have not yet constructed (up to isomorphism) the reduction graph $\mathcal{R}_u$ of a realistic string u from its overlap graph. We have "only" constructed $\text{cps}(\mathcal{R}_u)$ (up to isomorphism). However, it is clear that $\mathcal{R}_u$ can easily be obtained from $\text{cps}(\mathcal{R}_u)$ (up to isomorphism) by considering the edges as reality edges and replacing every vertex by a desire edge of the same label.

5.8 Consequences

We can now apply our main theorem, Theorem 15, to carry over results that rely on the notion of reduction graph for legal strings. To illustrate this, we characterize successfulness for realistic overlap graphs in any given $S \subseteq \{Gnr, Gpr, Gdr\}$. To accomplish this, we also use results from [13] (or Chapter 13 in [12]). The notions of successful reduction, string negative rule and graph negative rule used in this section are defined in [12].

First we restate a theorem of [6].

Theorem 16
Let u be a legal string, and N be the number of components in $\mathcal{R}_u$. Then every successful reduction of u has exactly $N-1$ string negative rules.

Due to the "weak equivalence" of the string pointer reduction system and the graph pointer reduction system, proved in Chapter 11 of [12], we can, using Theorem 15, restate Theorem 16 in terms of graph reduction rules.

Theorem 17
Let γ be a realistic overlap graph, and N be the number of components in $\mathcal{R}_\gamma$. Then every successful reduction of γ has exactly $N-1$ graph negative rules.

As an immediate consequence we get the following corollary. It provides an answer to an open problem formulated in Chapter 13 in [12]: to provide a graph theoretic characterization of successfulness in $\{Gpr, Gdr\}$. However, note that our answer is only for the case where γ is a *realistic* overlap graph.

Corollary 18
Let γ be a realistic overlap graph. Then γ is successful in $\{Gpr, Gdr\}$ iff $\mathcal{R}_\gamma$ is connected.

Example 11
Every successful reduction of the overlap graph of Example 9 has exactly one graph negative rule. For example $\mathbf{gnr}_2\ \mathbf{gpr}_4\ \mathbf{gpr}_5\ \mathbf{gpr}_7\ \mathbf{gpr}_6\ \mathbf{gpr}_3$ is a successful reduction of this overlap graph.

With the help of [13] (or Chapter 13 in [12]) and Corollary 18, we are ready to complete the characterization of successfulness for realistic overlap graphs in any given $S \subseteq \{Gnr, Gpr, Gdr\}$.

Theorem 19
Let γ be a realistic overlap graph. Then γ is successful in:

- $\{Gnr\}$ iff γ is a discrete graph with only negative vertices.
- $\{Gnr, Gpr\}$ iff each component of γ that consists of more than one vertex contains a positive vertex.
- $\{Gnr, Gdr\}$ iff all vertices of γ are negative.

- $\{Gnr, Gpr, Gdr\}$.
- $\{Gdr\}$ iff all vertices of γ are negative and $\mathcal{R}_\gamma$ is connected.
- $\{Gpr\}$ iff each component of γ contains a positive vertex and $\mathcal{R}_\gamma$ is connected.
- $\{Gpr, Gdr\}$ iff $\mathcal{R}_\gamma$ is connected.

Proof
The cases where $Gnr \in S$ (cf. the first four cases in the theorem) are known from [13], and case $\{Gpr, Gdr\}$ holds by Corollary 18. Case $\{Gdr\}$ ($\{Gpr\}$, resp.) is obtained by combining the results of cases $\{Gnr, Gdr\}$ ($\{Gnr, Gpr\}$, resp.) and $\{Gpr, Gdr\}$. Note that if γ has an isolated negative vertex, then $\mathcal{R}_\gamma$ is not connected. ■

5.9 Discussion

We have shown a way to directly construct the reduction graph of a realistic string (up to isomorphism) from its overlap graph γ. This allows one to (directly) determine the number n of graph negative rules that are necessary to reduce γ successfully. Surprisingly, although a lot a structural information is lost in the overlap graph (compared to a string representation), this information can be retrieved from the overlap graph via its reduction graph. The main result allows for a complete characterization of successfulness of γ in any given $S \subseteq \{Gnr, Gpr, Gdr\}$ by extending [13] for the cases where $Gnr \notin S$. It remains an open problem to find a (direct) method to determine this number n for overlap graphs γ in general (not just for realistic overlap graphs).