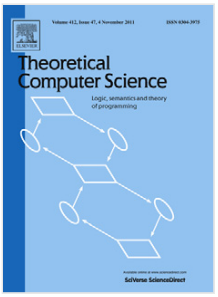
Accepted Manuscript



A comparison of graph-theoretic DNA hybridization models

Robert Brijder, Joris J.M. Gillis, Jan Van den Bussche

PII: S0304-3975(11)00989-3
DOI: 10.1016/j.tcs.2011.12.023
Reference: TCS 8634

To appear in: *Theoretical Computer Science*

Please cite this article as: R. Brijder, J.J.M. Gillis, J. Van den Bussche, A comparison of graph-theoretic DNA hybridization models, *Theoretical Computer Science* (2011), doi:10.1016/j.tcs.2011.12.023

A comparison of graph-theoretic DNA hybridization models

Robert Brijder*, Joris J.M. Gillis[1], Jan Van den Bussche

Hasselt University and Transnational University of Limburg, Belgium

Abstract

We show that the graph-theoretic DNA hybridization models of pot tiles [4, 5] and of sticker complexes [3, 2] are equivalent. This allows one to carry over known results from one model to the other. In addition, we introduce the concept of "greedy" hybridization and compare it to "regular" hybridization.

Keywords: DNA hybridization, multiset-based graph grammars, self-assembly, database theory

1. Introduction

During the process of self-assembly, intricate structures are obtained by spontaneous assembly of smaller building blocks. Self-assembly may appear at various scales [8], e.g., it may appear on the level of molecules or on the level of organisms [1]. A prime example of the former is self-assembly of algorithmically-designed DNA molecules through hybridization (see, e.g., [9, 6, 7]).

Hybridization, as a self-assembly process, has been studied in [10] from a computational point of view, where the families of regular, context-free, and recursively enumerable languages are characterized by three different hybridization models. The hybridization model for the family of recursively enumerable languages relies on two-dimensional sheets to provide the necessary context-sensitivity. On the other hand, the hybridization model for the family of context-free only uses branched junction DNA molecules for its computation.

In [4, 5], a hybridization model similar to the context-free model of [10] is used to characterize whether or not complexes without sticky ends may be obtained through hybridization from a given set of branched junction DNA molecules. Intuitively, as hybridization in this model is context-free, the precise physical structure of a complex is irrelevant — only the multiset of sticky ends

*Corresponding author

Email addresses: robert.brijder@uhasselt.be (Robert Brijder), joris.gillis@uhasselt.be (Joris J.M. Gillis), jan.vandenbussche@uhasselt.be (Jan Van den Bussche)

[1]Ph.D. Fellow of the Research Foundation Flanders (FWO).

matters. The model in [4], which we refer to as the *pot tile model*, is therefore quite general and may potentially be used for other types of self-assembly as well.

Inspired by the potential of DNA computing for database applications, a formal database model using DNA complexes is presented in [3]. A crucial operation within this database model is a context-free type of hybridization. This context-free type of hybridization, which we refer to as the *sticker complex hybridization model*, is more thoroughly studied in [2]. There it is shown, using graph-theoretical arguments, that it can be decided in polynomial time whether or not the set of DNA complexes obtainable (up to isomorphism) through hybridization is infinite for a given initial complex (appearing in a unbounded multiplicity).

In this paper, we show that the hybridization models of pot tiles [4] and sticker complexes [2] are equivalent. In this way, results from either of the two models may be carried over to the other model. Of course, either model may then be chosen to present the results of the two models uniformly — we have chosen in this paper for (a streamlined version of) the pot tile model. Furthermore, to capture the fact that sticky ends within a complex are much more likely to interact than sticky ends between two complexes, we introduce in this paper the notion of greedy hybridization. Finally, we fit this notion into a result of [4].

This paper is organized as follows. In Section 2 we recall basic notions and notation regarding multisets and graphs. We recall the pot tile model of [4] in Section 3, and the sticker complex hybridization model of [3, 2] in Section 4. In Section 5 we show that both models are equivalent. In Section 6 we motivate and define the notion of greedy hybridization, and in Sections 7 and 8 we present known results of both hybridization models within a common framework and moreover carry over a result to greedy hybridization. Finally, a discussion is given in Section 9.

2. Preliminaries

In order to fix notation and terminology, we recall in this section some basic notions concerning multisets and graphs.

A *multiset* m (over set S) is a function $S \to \mathbb{N}_0$. Intuitively, m "is" the set S where elements of S can appear more than once. For $a \in S$, we write $|m|_a = m(a)$, i.e., the number of occurrences of a in m. A multiset m can be represented by a set $S_m = \cup_{a \in S}\{(a,1), \ldots, (a, m(a))\}$. We say that S_m is the *set corresponding to* m. We may also write m as a string, e.g., *aaab* over $\{a, b, c\}$ means $m(a) = 3$, $m(b) = 1$, and $m(c) = 0$ (the ordering of the letters in the string is irrelevant). The set of multisets over S is denoted by $\mathcal{M}_S$.

As usual, an (undirected) *multigraph* (thus allowing both parallel edges and loops) is formalized as a tuple (V, E, ϵ) where V is the set of vertices, E is the set of edges, and $\epsilon : E \to \Omega_V$ the endpoint mapping, where $\Omega_V = \{\{u, v\} \mid u, v \in V, u \neq v\} \cup \{\{u\} \mid u \in V\}$. For $e \in E$, if $\epsilon(e) = \{u\}$, then e is a *loop* on

u. Let Σ be some alphabet. A *labelled multigraph* (over Σ) is a tuple (V, E, ϵ, l) where (V, E, ϵ) is a multigraph and $l : V \to \Sigma$ is a vertex labelling function.

3. Pots and Hybridization

In this section we recall the hybridization model described in [4]. This model is also defined in its extended version [5], however the definition in [4] is more concise. For notational convenience, the formulation below is slightly altered w.r.t. [4].

We fix some finite alphabet Δ, and let $\bar{\Delta} = \{\bar{x} \mid x \in \Delta\}$ be a disjoint copy of Δ. We let $\bar{\bar{x}} = x$, i.e., the bar operator is an involution. We define $\Sigma = \Delta \cup \bar{\Delta}$. The elements of Σ are called *sticky-end types*. We say that a and $\bar{a}$ are *complementary* for $a \in \Sigma$. The sticky-end types represent DNA sequences, where complementary sticky-end types are the Watson-Crick complement of each other. As the same DNA sequence may occur multiple times in the environment, we make a distinction between a sticky-end type, which is a DNA sequence, and a sticky-end, which is an *occurrence* of a particular DNA sequence in the environment.

No particular physical structure is assumed for the DNA building blocks that contain the sticky ends. Hence, a building block, called *tile*, is simply a multiset of sticky-end types.

Definition 1. A *pot type* H (over Σ) is a finite set of tiles. A *pot* P of H is a multiset of tiles in H.

A pot type H is called *proper* if for any x that appears in some tile $m_1 \in H$, $\bar{x}$ also appears in some tile $m_2 \in H$.

Example 2. Let us fix $\Delta = \{a, b\}$. Then $H = \{aab, b\bar{b}, \bar{a}\bar{b}\}$ is a pot type of Σ, and $P = \{aab, \bar{a}\bar{b}, \bar{a}\bar{b}\}$ is a pot of H. Note that H is proper.

Although a tile does not fix a particular physical interpretation, the definition of a tile is motivated in [4] by one such interpretation: the branched junction DNA molecules. These molecules are star shaped with a sticky-end at the end of each arm, and these arm are flexible enough such that complementary free sticky-ends (between molecules or within a molecule) can always engage in a bonding. Intuitively, this accounts for the context-freeness of the model. As an example, the branched junction DNA molecule for the tile $t = aab$, is given in Figure 1. The intuition of a pot P is then a test tube containing these branched junction DNA molecules in the multiplicity given by the multiset P.

With the definition of pot P in place, hybridization is now simply a matching of complementary sticky ends between the tiles in P. We will now formalize this notion.

Let P be a pot of some type H. Let S_P be the standard set representation of P, where thus $(m, i) \in S_P$ denotes the i^{th} occurrence of $m \in H$ in P.

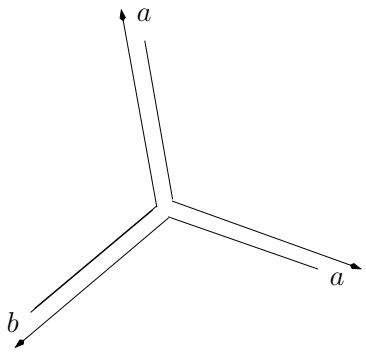


Figure 1: Branched junction DNA molecule for tile $t = aab$.

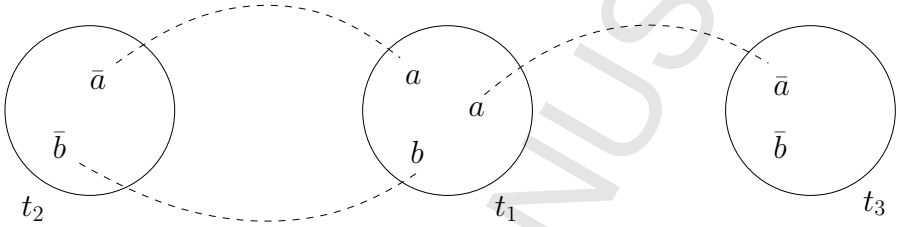


Figure 2: A matching of the pot of Example 4.

Definition 3. Let P be a pot of some type H. A *matching* over P is a set B of unordered pairs $c = \{(t, a), (t', \bar{a})\}$ where $t, t' \in S_P$, $a \in \Sigma$, and such that for all $x \in S_P$ and $b \in \Sigma$, we have $|\{c \mid (x, b) \in c \in B\}| \leq |h|_b$ where $x = (h, i)$ for some integer i.

If a pot type H is not proper, say $x \in \Sigma$ appears in some tile of H while $\bar{x}$ does not, then x is not "interacting". That is, each of occurrence of x may never engage in a matching. Consequently, the family of proper pot types may be seen as a normal form for the family of all pot types. Therefore, we assume without loss of generality that *each pot type under consideration is proper*.

The next example illustrates the concept of matching over a pot.

Example 4. Consider again the pot type $H = \{aab, b\bar{b}, \bar{a}\bar{b}\}$ and pot $P = \{aab, \bar{a}\bar{b}, \bar{a}\bar{b}\}$ from Example 2. Then $S_P = \{t_1, t_2, t_3\}$ where $t_1 = (aab, 1)$, $t_2 = (\bar{a}\bar{b}, 1)$, and $t_3 = (\bar{a}\bar{b}, 2)$. Let $B = \{c_1, c_2, c_3\}$ where $c_1 = \{(t_1, a), (t_2, \bar{a})\}$, $c_2 = \{(t_1, b), (t_2, \bar{b})\}$, and $c_3 = \{(t_1, a), (t_3, \bar{a})\}$. By Definition 3, P is a matching over P. Indeed, e.g., the ordered pair $(t_1, a) = ((aab, 1), a)$ appears not more than 2 times in B, and similarly the ordered pair $(t_1, b) = ((aab, 1), b)$ appears not more than once in B. Matching B is visualized in Figure 2, where the tiles are represented by vertices and the matching pairs by dashed edges.

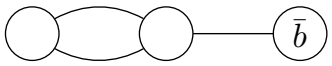


Figure 3: The complex of the matching of Figure 2.

Figure 2 suggests a graph representation of a matching, which is called a complex, where the vertices are the tiles and the edges between the tiles are the matchings. For administrative purposes the unused sticky ends appear as labels of the vertices. Intuitively, a complex is a graph that represents the structure obtained when hybridizing the branched junction DNA molecules associated with the tiles. The formal definition given here is much more concise than the formal definition in [5] which explicitly uses star graphs that join together according to a specific type of graph homomorphism.

Definition 5. Let B be a matching over some pot P. The *complex* of B and P is the labelled multigraph $G = (V, E, \epsilon, l)$ where $V = S_P$ is the set corresponding to P, $E = B$, for $\{(u, a), (v, \bar{a})\} \in E$, $\epsilon(e) = \{u, v\}$ if $u \neq v$ and $\epsilon(e) = \{u\}$ otherwise, and $l : V \to \mathcal{M}_\Sigma$ such that for $v \in V$ and $a \in \Delta$, $|l(v)|_a = |t_v|_a - |\{z \in B \mid (v, a) \in z\}|$, where t_v is the tile corresponding to v.

Note that the label of a vertex v of a complex in Definition 5 is obtained from the tile t_v corresponding to v by removing the sticky ends of t_v that appear in B.

Note also that as the case $t = t'$ is allowed in the definition of matching B, a complex may have loops.

A complex G of a matching B is called *terminal* if each vertex of G is labelled by the empty multiset. As we assume that pot types are proper, terminal complexes represent precisely those DNA complexes which cannot be extended any further (i.e., these DNA complexes cannot engage in a bonding with other DNA complexes through sticky ends).

Remark 6. It is, of course, possible to extend the notion of a complex by introducing an edge labelling function which assigns to each edge the label in Δ that was used in the matching. For our purposes in this paper we do not need this labelling.

Example 7. The complex of the matching B in Example 4 (visualized in Figure 2) is given in Figure 3. Note that the labels of the vertices representing t_1 and t_2 are the empty multiset as the number of occurrences of each letter l in t_1 (t_2, resp.) is exactly the number of occurrences of (t_1, l) ((t_2, l), resp.) in pairs in B. Also, the label of the vertex representing t_3 contains one occurrence of $\bar{b}$ since the number of occurrences of $\bar{b}$ in t_3 is one more than the number of occurrences of $(t_3, \bar{b})$ in pairs in B (which, in fact, is zero).

The next definition illustrates that a pot type H can be considered as a particular kind of graph grammar.

Definition 8. Let H be a pot type. The *language* of H, denoted by $L(H)$, is the set of connected complexes of matchings of a pot of type H. Moreover, the *terminal language* of H, denoted by $TL(H)$, is the set of terminal complexes of $L(H)$.

4. Sticker complex data model

We now recall the DNA sticker complex model defined in [3]. It is used in [3] as a data model for database computations in which hybridization is one of several DNA operations defined by the model. In this paper we do not recall the other DNA operations which form together with hybridization the full database model, as these other DNA operations are not used or considered in this paper. Hybridization in this model is studied in more detail in [2], and for notational convenience we recall now the more succinct description of this model formulated in [2].

Recall from Section 3 that we let $\Sigma = \Delta \cup \bar{\Delta}$ where Δ is some fixed finite alphabet.

Definition 9. A *pre-complex* is a 4-tuple $C = (V, L, \lambda, \mu)$ where

1. $D = (V, L)$ is a digraph without loops (i.e., $(v, v) \notin L$ for all $v \in V$),
2. $\lambda : V \to \Sigma$ is a vertex labelling function, and
3. $\mu \subseteq \{\{v, w\} \mid v, w \in V, v \neq w, \text{ and } \lambda(v) = \overline{\lambda(w)}\}$ is a partial matching on the set of vertices, i.e., each vertex occurs in at most one pair in μ.

A *strand* of a pre-complex $C = (V, L, \lambda, \mu)$ is a connected component of the digraph $D = (V, L)$ (so ignoring μ). A *component* of C is a connected component of C regarding both L and μ as edges (so not ignoring μ). The *length* of a strand is its number of vertices. The vertices of C that do not appear in μ are called *free* (in C).

Definition 10. A *sticker complex* is a pre-complex with the following restrictions on the strands:

1. Each node has at most one incoming and at most one outgoing edge. Thus, each strand has the form of a chain or a cycle.
2. Strands are homogeneously labeled: the vertex labels of a strand are either all in Δ (a *positive* strand) or all in $\bar{\Delta}$ (a *negative* strand).
3. Every negative strand has length one or two, and if it has length two, then it must have a single edge (i.e., it cannot be a 2-cycle).

Example 11. A sticker complex C is given in Figure 4. The dashed lines indicate the unordered pairs of μ. The sticker complex consists of five strands: two positive strands (both of length 3) and three negative strands (two of length 2 and one of length 1). There are three components. The component in the middle of the figure has two free vertices, one labelled by $\bar{b}$ and the other by b, and all vertices of the other two components are free.

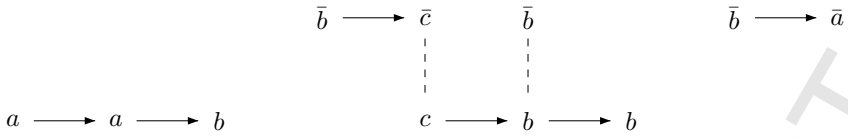


Figure 4: A sticker complex.

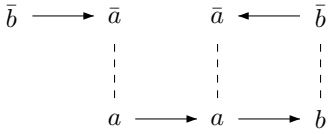


Figure 5: A MHE of the sticker complex of Figure 4.

We define now hybridization for sticker complexes. Intuitively, we start with a sticker complex C that specifies the types of complexes that appear in arbitrary multiplicity in the environment (e.g., a test tube). The components obtained during hybridization are the components obtained by extending the partial matching μ within and between copies of components of C. Formally, for sticker complexes C and C', C' is a *redundant variation* of C if for each component D' of C' there is a component D of C isomorphic to D'. For a sticker complex $C = (V, L, \lambda, \mu)$, a *hybridization extension* of C is a sticker complex (V, L, λ, μ') with $\mu' \supseteq \mu$. A *multiplying hybridization extension* (MHE for short) for C is a hybridization extension of a redundant variation of C. The *hybridization* of C, denoted by $\mathrm{hybr}(C)$, is the set of components of MHE's of C. We, similar as for the pot tile model, assume without loss of generality that a sticker complex is *proper*, i.e., if a vertex v is free in C, then there is a vertex w free in C with $\lambda(v) = \overline{\lambda(w)}$. A component Q of an MHE of a sticker complex C is called *finished* if there is no free vertex v of Q and free vertex w of C such that $\lambda(v) = \overline{\lambda(w)}$. The set of finished components of $\mathrm{hybr}(C)$ is denoted by $\mathrm{fhybr}(C)$.

Example 12. Consider again the sticker complex C of Figure 4. Let us denote the three components of C from left to right by Q_1, Q_2 and Q_3. The MHE of C given in Figure 5 is obtained from a redundant variation of C, consisting of one isomorphic copy of Q_1 and two isomorphic copies of Q_3, by a hybridization extension where three pairs are added to μ.

5. Equivalence of Hybridization Models

We now show that the pot tile hybridization model recalled in Section 3 is equivalent to hybridization in the sticker complex data model recalled in Section 4. By equivalent we mean that although the basic building blocks for hybridization are different, hybridization operates in exactly the same way: building blocks may stick to other building blocks if they have complementary sticky ends. The correspondence is more precisely described as follows.

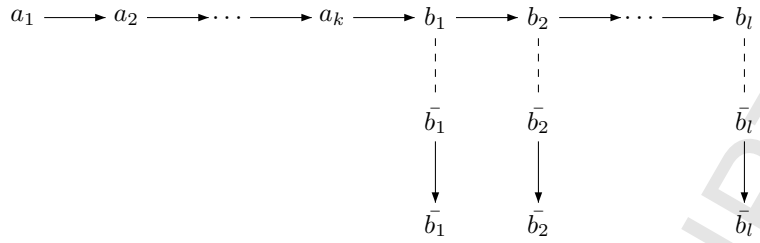


Figure 6: A sticker complex that simulates a tile.

Let C be a sticker complex, and let Q be a component of C. Let F_Q be the set of free vertices of Q. Let tile(Q) be the multiset over Σ where $|m_Q|_a = |\{v \in F_Q \mid \lambda(v) = a\}|$. In this way, the sticker complex C corresponds to the pot type pottype(C) consisting of the tiles tile(Q) for Q a component of C. A redundant variation C' of C corresponds to a pot, denoted by pot(C'), which is the multiset of tiles tile(Q) with Q a component of C'. A hybridization extension of C' (an MHE) is a matching of free vertices of C', and corresponds therefore to a matching B over pot(C'). Therefore, the set of components of MHE's of C (i.e., hybr(C)) corresponds then to the connected components of the complexes G of matchings B (recall that a complex is the natural graph representation of a matching B) over pots of type H_C, i.e., the language $L(H_C)$. Moreover, the set of finished components of MHE's of C (i.e., fhybr(C)) corresponds to the set of terminal complexes of $L(H_C)$, i.e., $TL(H_C)$.

Example 13. Consider again the sticker complex C of Figure 4 containing the components Q_1, Q_2 and Q_3. Then tile(Q_1) = aab, tile(Q_2) = $b\bar{b}$, and tile(Q_3) = $\bar{a}\bar{b}$. Hence H = pottype(C) = $\{aab, b\bar{b}, \bar{a}\bar{b}\}$. Consider the redundant variation C' of C of Example 12. The pot P corresponding to C' is pot(C') = $\{aab, \bar{a}\bar{b}, \bar{a}\bar{b}\}$. Note that pot type H and pot P are equal to the pot type and pot of Example 2.

Consider now the hybridization extension C'' of C' of Example 12, i.e., the MHE given in Figure 5. We find that C'' corresponds to the matching B over pot(C') given in Figure 2.

Conversely, we start with a tile t. Let $t = a_1 \cdots a_k \bar{b_1} \cdots \bar{b_l}$ with all the a_i's and b_i's in Δ. In Figure 6 a connected sticker complex compSticker(t) corresponding to t is constructed. Note that the labels of the free vertices of compSticker(t) form precisely the multiset t. Hence, tile(compSticker(t)) = t. Given a pot type H, we construct in this way a complex sticker(H) such that the components Q of sticker(H) correspond one-to-one with occurrences of tiles $t \in H$ where Q is isomorphic to compSticker(t). Consequently, pottype(sticker(H)) = H. Evidently, a pot P of type H corresponds (in exactly the same way as before) to a redundant variation C' of C_H, and moreover a matching B of a pot P corresponds to a hybridization extension of C'. Again, the (terminal) connected components of the complexes G of matchings B correspond to the (finished) components of MHE's of C.

From this equivalence the next result follows.

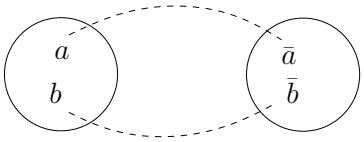


Figure 7: A greedy matching.

Theorem 14. *Let C be a sticker complex. Then* hybr(C) *is infinite (empty, resp.) iff* L(pottype(C)) *is infinite (empty, resp.). Similarly,* fhybr(C) *is infinite (empty, resp.) iff* TL(pottype(C)) *is infinite (empty, resp.).*

As pottype(sticker(H)) = H for each pot type H, we have the following corollary to Theorem 14.

Corollary 15. *Let H be a pot type. Then* hybr(sticker(H)) *is infinite (empty, resp.) iff* $L(H)$ *is infinite (empty, resp.). Similarly,* fhybr(sticker(H)) *is infinite (empty, resp.) iff* $TL(H)$ *is infinite (empty, resp.).*

We may use either of these two hybridization models to unify known results of these models. In the rest of this paper we will work with the pot tile hybridization model (of Section 3).

6. Greedy hybridization

As noted in [5] it is natural to assume that sticky ends within a connected DNA complex are much more likely to interact than sticky ends between two complexes. In this way, when two DNA complexes meet to form a single connected DNA complex, all complementary sticky ends in the newly formed DNA complex will hybridize before interactions with other DNA complexes occur. Sticky ends will thus bond in a "greedy" way. To this aim, it is assumed in [5] that all complexes in $L(H)$ (for pot type H) are stable, where the notion of stable is defined as follows.

Definition 16. Let $C = (V, E, \epsilon, l)$ be a complex (as defined in Definition 5). Then C is called *stable* when, for all $v, v' \in V$, if $a \in l(v)$, then $\bar{a} \notin l(v')$ (we allow $v = v'$).

Example 17. The complex C of Figure 3 is stable. Indeed, $\bar{b}$ is the only sticky end of C, and thus there are no two complementary sticky ends in C.

However, one may argue that the condition of stable is not enough to capture the above described greedy way of hybridizing. Indeed, consider the pot $P = \{ab, ab, \bar{a}\bar{b}\}$. When a tile of type ab hybridizes in a greedy way with a tile of type $\bar{a}\bar{b}$, all sticky ends will bond to obtain a stable complex of two vertices, see Figure 7. (Note that the obtained complex is indeed stable as it has no sticky ends — they are all used in the matching.) As a consequence, the (perfectly

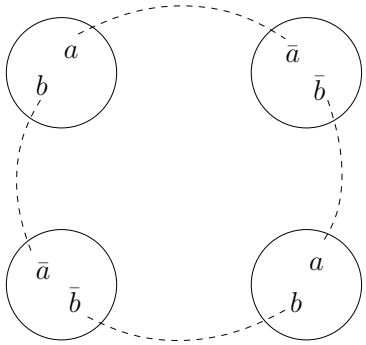


Figure 8: A non-greedy matching.

valid) matching B of P given in Figure 8 may not be obtained by hybridizing in a greedy way.

We now define the class of greedy matchings. This class formalizes the intuitive notion of greedy hybridization: two different complexes may only glue together when they are both stable.

Definition 18. Let P be a pot (of some type H). A matching B over P is called *greedy* if there is a linear ordering $c_1, \ldots, c_n$ of B such that

1. C_n is stable, and
2. if C_{j+1} has less connected components than C_j for some $j \in \{1, \ldots, n-1\}$, then C_j is stable,

where C_i for $i \in \{1, \ldots, n\}$ denotes the complex of the matching $\{c_1, \ldots, c_i\}$ over P.

Note that "C_{j+1} has less connected components than C_j" is equivalent to "C_{j+1} has one connected component less than C_j" in Condition 2 in Definition 18.

Evidently, the complex of a greedy matching is stable, but we have seen in Figure 8 that the converse does not hold.

We (may) now trivially define the language and terminal language w.r.t. this greedy way of hybridization.

Definition 19. Let H be a pot type. The *greedy language* of H, denoted by $GL(H)$, is the set of connected complexes of greedy matchings of a pot of type H. Moreover, the *terminal greedy language* of H, denoted by $TGL(H)$, is the set of terminal complexes of $GL(H)$.

7. Existence of terminal graphs

In [5], among other related problems, the problem of whether or not the terminal language of a pot type H is empty (i.e., whether or not $TL(H) = \varnothing$) is investigated. In this section we recall the characterization of emptiness of

$TL(H)$ in [5] and show that this result carries over when restricting to greedy languages.

We associate to each multiset m over $\Sigma = \Delta \cup \bar{\Delta}$ the row vector v_m over the integers indexed by Δ where, for each $a \in \Delta$, $v_m(a) = |m|_a - |m|_{\bar{a}}$. Here $v_m(a)$ denotes the value on the position a in vector v_m.

Example 20. Let $\Delta = \{a, b, c\}$ and $m = aa\bar{a}\bar{b}\bar{b}$. Then $v_m = (1, -2, 0)$ where the values are in the order a, b, c, e.g., $v_m(b) = -2$.

The following result links the emptiness conditions of $TL(H)$ and $TGL(H)$ to basic linear algebra. The second "iff" statement in this result is essentially from [5]. For convenience and clarity of exposition we give here a full proof without resorting to [5]. We denote by $\mathbb{Q}^+$ the set of nonnegative rational numbers.

Theorem 21. *Let $H = \{m_1, m_2, \ldots, m_n\}$ be a pot type. Then $TGL(H) = \varnothing$ iff $TL(H) = \varnothing$ iff $v_{m_1}, v_{m_2}, \ldots, v_{m_n}$ are linearly independent over $\mathbb{Q}^+$.*

Proof. Let us denote the three conditions in the theorem by (i), (ii), and (iii) respectively.

We first prove that (i) implies (iii). If $v_{m_1}, v_{m_2}, \ldots, v_{m_n}$ are linearly dependent over $\mathbb{Q}^+$, then there are $k_1, \ldots, k_n \in \mathbb{Q}^+$ (not all zero) such that $k_1 v_{m_1} + k_2 v_{m_2} + \cdots + k_n v_{m_n}$ is equal to the null vector. Let z be least common multiple of the denominators of $k_1, \ldots, k_n$. Then $zk_1, \ldots, zk_n \in \mathbb{N}_0$ and $(zk_1)v_{m_1} + (zk_2)v_{m_2} + \cdots + (zk_n)v_{m_n}$ is equal to the null vector (thus, $v_{m_1}, v_{m_2}, \ldots, v_{m_n}$ are linearly dependent over $\mathbb{N}_0$). Let P be the pot with $P(m_i) = zk_i$ for all $i \in \{1, \ldots, n\}$. By the construction of P the number of sticky ends of every type $x \in \Sigma$ among the tiles in P is equal to the number of sticky ends of type $\bar{x} \in \Sigma$. Let B be an arbitrary maximal greedy matching of P, i.e., B is a greedy matching of P but no proper superset of B is a greedy matching. Now, as the number of sticky ends of type $x \in \Sigma$ and of type $\bar{x} \in \Sigma$ are equal, B matches every sticky end. Hence, the complex C_B of B is stable and thus $C_B \in TGL(H)$. Therefore, $TGL(H) \neq \varnothing$.

Next we prove that (iii) implies (ii). Assume $TL(H) \neq \varnothing$. Then there is a pot $P : H \to \mathbb{N}_0$ of type H and a matching B of P such that the complex C_B of B is stable. As each sticky end in each tile in P is matched with a complementary sticky end, the number of occurrences of $x \in \Sigma$ in P is equal to the number of occurrences of $\bar{x} \in \Sigma$ in P. Consequently, $P(m_1)v_{m_1} + P(m_2)v_{m_2} + \cdots + P(m_n)v_{m_n}$ is the null vector and $v_{m_1}, v_{m_2}, \ldots, v_{m_n}$ are linearly dependent.

Finally, (ii) implies (i) as the set of greedy matchings is a subset of the set of matchings. □

We illustrate Theorem 21 by two examples.

Example 22. Consider the pot type $H_1 = \{ab, \bar{a}\bar{b}\}$. Then $v_{ab} = (1, 1)$ and $v_{\bar{a}\bar{b}} = (-1, -1)$, where the vectors are indexed by a, b (in this order). Clearly, v_{ab} and $v_{\bar{a}\bar{b}}$ are linearly dependent over $\mathbb{Q}^+$ (their sum is the null vector). By Theorem 21, $TGL(H_1) \neq \varnothing$. Indeed, a greedy matching of a pot of H_1 is given in Figure 7.

Example 23. Consider now the pot type $H_2 = \{aab, \overline{b}\overline{b}\}$. Then $v_{aab} = (2, 1)$ and $v_{\overline{b}\overline{b}} = (0, -2)$, where again the vectors are indexed by a, b (in this order). Clearly, v_{aab} and $v_{\overline{b}\overline{b}}$ are linearly independent over $\mathbb{Q}^+$. By Theorem 21, $TL(H_2) = \varnothing$. Hence, there is no matching of a pot of type H_2 for which the complex is stable.

Let A be the matrix where the columns of A are the vectors $v_{m_1}, v_{m_2}, \ldots, v_{m_n}$ of Theorem 21 (given as column vectors). Then $v_{m_1}, v_{m_2}, \ldots, v_{m_n}$ are linearly dependent over $\mathbb{Q}^+$ iff there is a vector x with nonnegative entries such that $Ax = 0$ over $\mathbb{Q}$. This problem can be formulated as a linear programming problem. Such problems (over $\mathbb{Q}$) are known to be solvable in time polynomial in the number of variables (which in our case is equal to the number of vectors n). Hence, the following holds.

Corollary 24 ([5])**.** *Let H be a pot type. It is decidable in polynomial time whether or not $TL(H) = \varnothing$ (and whether or not $TGL(H) = \varnothing$).*

8. Termination

We say that H is *terminating* if $L(H)$ is finite and H is otherwise called *nonterminating*. This notion is characterized in [2] in the sticker complex model of Section 4. We refer to [2] for a detailed motivation for studying this notion. By the equivalence shown in Section 4 of the sticker complex model and hybridization model of [4] (recalled in Section 3), the results of [2] carry over to the latter model essentially unchanged. The main differences are notational. Therefore, in this section, we omit proofs. They can be found, w.r.t. the sticker complex model, in [2].

We say that a cycle C in a graph is *simple* if all edges of C are distinct. The next result is (implicitly) shown in [2].

Theorem 25 ([2])**.** *Let H be a pot type. Then H is nonterminating iff there is a $G \in L(H)$ containing a simple cycle.*

Moreover, [2] shows that a simple cycle in a graph $G \in L(H)$ is equivalent to an alternating cycle in a graph, called the *hybridization graph*, of a pot type H — this latter condition is computable in polynomial time. Combining this with Theorem 25 the following result is obtained.

Corollary 26 (Corollary 1 of [2])**.** *Let H be a pot type. It is decidable in polynomial time whether or not H is nonterminating.*

Although by Corollary 26 it is decidable in polynomial time whether or not $L(H)$ is finite, it is an open problem whether or not the finiteness of $GL(H)$ is decidable in polynomial time. In fact, it is also open whether or not the finiteness of $TL(H)$ or of $TGL(H)$ is decidable in polynomial time.

9. Discussion

We have shown that the hybridization models of pot tiles [4, 5] and of sticker complexes [3, 2] are equivalent and we illustrated this by fitting results from each of these models into a single framework. We also considered greedy hybridization to model the greedy nature of hybridization.

Although the precise physical structure of DNA building blocks for context-free hybridization is irrelevant, it, of course, may matter for certain applications. For example, for the "full" sticker complex database model of [3], which consists of several operations in addition to hybridization, the ordering of the sticky ends in the sticker complexes is important.

Open problems remain on the decidability of various graph language classes, in particular those associated with greedy hybridization.

Acknowledgements

The authors are indebted to the anonymous referees for valuable comments on this paper.

References

[1] E. Ben-Jacob and H. Levine. Self-engineering capabilities of bacteria. *Journal of Royal Society Interface*, 3(6):197–21, 2006.

[2] R. Brijder, J.J.M. Gillis, and J. Van den Bussche. Graph-theoretic formalization of hybridization in DNA sticker complexes. In L. Cardelli and W.M. Shih, editors, *Proceedings of the 17th International Conference on DNA Computing and Molecular Programming (DNA 17)*, volume 6937 of *Lecture Notes in Computer Science*, pages 49–63. Springer, 2011.

[3] J.J.M. Gillis and J. Van den Bussche. A formal model of databases in DNA. In K. Horimoto, M. Nakatsui, and N. Popov, editors, *Algebraic and Numeric Biology 2010*, volume 6479 of *Lecture Notes in Computer Science*. Springer, 2011. To appear; for a preprint see `http://alpha.uhasselt.be/~vdbuss/dnaql.pdf`.

[4] N. Jonoska, G.L. McColm, and A. Staninska. Spectrum of a pot for DNA complexes. In C. Mao and T. Yokomori, editors, *Proceedings of the 12th International Meeting on DNA Computing (DNA 12)*, volume 4287 of *Lecture Notes in Computer Science*, pages 83–94. Springer, 2006.

[5] N. Jonoska, G.L. McColm, and A. Staninska. On stoichiometry for the assembly of flexible tile DNA complexes. *Natural Computing*, 10(3):1121–1141, 2011.

[6] P.W. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 2006.

[7] P.W.K. Rothemund, N. Papadakis, and E. Winfree. Algorithmic self-assembly of DNA sierpinski triangles. *PLoS Biol*, 2(12):e424, 2004.

[8] G.M. Whitesides and B. Grzybowski. Self-assembly at all scales. *Science*, 295:2418–2421, 2002.

[9] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman. Design and self-assembly of two dimensional DNA crystals. *Nature*, 394:539–544, 1998.

[10] E. Winfree, X. Yang, and N.C. Seeman. Universal computation via self-assembly of DNA: Some theory and experiments. In L.F. Landweber and E.B. Baum, editors, *DNA Based Computers II: DIMACS Workshop, held June 10–12, 1996*, pages 191–213. American Mathematical Society, 1998.