

Research Article

The MATCHIT Automaton: Exploiting Compartmentalization for the Synthesis of Branched Polymers

**Mathias S. Weyland,¹ Harold Fellermann,² Maik Hadorn,³ Daniel Sorek,⁴
Doron Lancet,⁴ Steen Rasmussen,^{2,5} and Rudolf M. Fuchsli^{1,6}**

¹ European Centre for Living Technology, S. Marco 2940, 30124 Venice, Italy

² Center for Fundamental Living Technology (FLinT), Department of Physics, Chemistry and Pharmacy, University of Southern Denmark, 5230 Odense, Denmark

³ Department of Chemistry and Applied Biosciences, ETH Zurich, 8093 Zurich, Switzerland

⁴ The Lancet Lab, Department of Molecular Genetics, Weizmann Institute of Science, 76100 Rehovot, Israel

⁵ Santa Fe Institute, Santa Fe, NM 87501, USA

⁶ Institute of Applied Mathematics and Physics, School of Engineering, Zurich University of Applied Sciences, 8401 Winterthur, Switzerland

Correspondence should be addressed to Mathias S. Weyland; mathias@weyland.ch

Received 31 October 2013; Accepted 8 December 2013

Academic Editor: Roberto Serra

Copyright © 2013 Mathias S. Weyland et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose an automaton, a theoretical framework that demonstrates how to improve the yield of the synthesis of branched chemical polymer reactions. This is achieved by separating substeps of the path of synthesis into compartments. We use chemical containers (chemtainers) to carry the substances through a sequence of fixed successive compartments. We describe the automaton in mathematical terms and show how it can be configured automatically in order to synthesize a given branched polymer target. The algorithm we present finds an optimal path of synthesis in linear time. We discuss how the automaton models compartmentalized structures found in cells, such as the endoplasmic reticulum and the Golgi apparatus, and we show how this compartmentalization can be exploited for the synthesis of branched polymers such as oligosaccharides. Lastly, we show examples of artificial branched polymers and discuss how the automaton can be configured to synthesize them with maximal yield.

1. Introduction

Recently, small scale personal manufacturing has seen a rapid increase in popularity with emerging technologies such as 3D printing. In place of central production and physical distribution of goods, personal manufacturing offers a transfer of information (e.g., designs, protocols) followed by in-place, customizable production. Although mainly discussed in the context of macroscale personal manufacturing (i.e., 3D printing), personal manufacturing is also found in the domain of (bio-)chemistry. In custom oligonucleotide and peptide synthesis, the information, that is, DNA or protein sequence, is sent by the customer to the supplier. Even though the synthesized DNA oligonucleotides or peptides are shipped back to the customer (i.e., transfer of goods), this

example demonstrates how transferred information can lead to in-place, customizable production of (bio)chemical goods. However, often the desired product cannot be synthesized via either one-pot synthesis or sequential one-pot synthesis. Consequently, distinct confinement and transport of substances and an elaborate temporal and spatial reaction management are needed.

The European Commission funded project MATCHIT (Matrix for Chemical IT [1, 2]) aims to open the domain of chemistry for distributed manufacturing by implementing unconventional embedded computation systems. By employing addressable soft colloid supermolecular chemical containers (chemtainers) that act both as transport and reaction vessels and that are interfaced with electronic computers via microelectromechanical systems (MEMS), the topological

organization of the cytoplasm of natural cells is mimicked, where a multitude of chemicals is organized in addressable compartments and where transport and fusion of compartments trigger reactions of previously separated chemicals. By utilizing a hybrid biochemical and information technological system, an integrated programmable material transportation, information processing and material production system was implemented resulting in a generic programmable platform for complex molecular processing tasks.

In contrast to other approaches that either employ “traditional” 3D printing techniques to implement small scale personal manufacturing in (bio)chemistry [3] or use of simpler building blocks like lipid-coated aqueous droplets in oil [4], in the MATCHIT machine, DNA oligonucleotides attached to the surface of the chemtainers (DNA addresses) are employed to guide the chemtainer’s assembly, fusion promoting chemical reactions, and reversible docking on defined spots of the microfluidic environment through reversible hybridization. The DNA addresses thereby dynamically change and adapt as a result of ongoing DNA-computation operations such as relabeling and simple Boolean operations [5–7]; the current set of DNA addresses therefore both portrays the chemtainers history and defines the next steps to take (e.g., with whom to bind and fuse, where to dock). In this context, the MEMS technology is perfectly suited not only for precise and accurate guidance the chemtainers creation [8], loading, and manipulation but also for an on-chip pre- and postprocessing (e.g., electrophoretic separation) of substances released from the chemtainers or waiting for reencapsulation.

The modular design of the MATCHIT machine allowed for an individual testing and optimization of each of its components (i.e., chemtainers, MEMS, and computer science). In the context of the chemtainers, we reported the first DNA-computation operations on the surface of artificial giant unilamellar vesicles (GUVs), DNA-mediated self-assembly [9, 10] and controlled fusion [11–13] of GUVs and oil-in-water droplets, and increase in the complexity of available soft colloid chemtainers by preparing hierarchically organized GUVs [14]. The research efforts of integrating chemical and electronic systems in a MEMS device resulted in unconventional embedded computation systems that performed complex nanoscale chemical tasks autonomously [15]. Traditional computer science was not only used for programing the MEMS devices. By using the same language for simulating the inherent complexity of MATCHIT objects and operations and to program the MEMS devices, solutions found *in silico* were directly transferred to control the MEMS devices. In this context, the MATCHIT automaton (MA) is able to optimize the microfluidics design in terms of structure, MEMS and chemtainer DNA addresses, chemtainer types, and interaction rules.

Our motivation to demonstrate how the MA can be applied to the synthesis of branched polymers (e.g., branched oligosaccharides) is as follows: often oligosaccharides are attached to proteins and impart an additional level of information content to the underlying protein structures [16]. The glycosylation is a critical function of the biosynthetic-secretory pathway in the endoplasmic reticulum (ER) and

the Golgi apparatus. Glycoproteins associated with cell surface and intracellular proteins have crucial biological and physiological roles, from contributions in protein folding and quality control to involvement in a large number of biological recognition events [16, 17]. In the ER, for example, glycosylation is used to monitor the status of protein folding, acting as a quality control mechanism to ensure that only proteins properly folded are trafficked to the Golgi. Oligosaccharides are composed of monosaccharides each having four to five binding sites. This large number of binding sites and the vast number of distinct monosaccharides constitute the large variety of oligosaccharides. Thus, the glycan structures attached to proteins can be highly complex, with numerous possibilities for branching and anomeric linkage. Consequently, glycoproteins have much greater structural diversity than linear nucleic or polypeptide structures. Hence, the custom synthesis of such complex branched molecules is a demanding task because the control of potential side reactions is still not solved [18]. In nature, glycosylation occurs in a stepwise fashion by trafficking glycoproteins to distinct spatially separated compartments (i.e., Golgi cisternae) that contain a distinct set of enzymes.

One approach to tackle the problem of side reactions is to use linker structures to increase the yield in one-pot reactions. Such small linker structures have been implemented and are known by the term click chemistries. The concept of click chemistries is for example, described in Kolb et al. [19–21] and addresses powerful, highly reliable, and selective reactions for the rapid synthesis of new compounds through heteroatom links. In the context of oligosaccharide synthesis however, the number of such click-chemistry linkers is highly limited, leading to a significant, but moderate, increase of yield [22]. Hence, our approach is to augment the effect of these small linker structures by means of additional compartmentalization. Specifically, the substances (e.g., oligosaccharide) are enclosed by chemtainers which can be transported between compartments and which can be addressed by tags.

In this paper, we study an artificial chemistry [23] that models the synthesis of branched polymers. The focus of this work lies in the description of a compiler for a defined reaction environment, the MA. By a compiler we understand a method that takes the description of a branched oligomer as input and maps it onto system parameters of the MA such that this automaton synthesizes the input structure. We refer to these system parameters as the configuration of the MA. With such a compiler, we present an example of what is often called ChemBio-IT [24], the merger of computer science and (bio)chemistry. The MA offers a platform with which such combinatorial variety of different types of branched oligomers—a promising tool in a future personalized medicine—can be generated. This work presents a mathematical formalism to encode a large and biologically important class of such oligomers and thereby enables automatized generation of protocols for the synthesis of specific structures.

The contribution of this work is twofold. First, we present a mathematical framework that describes the MA. Second,

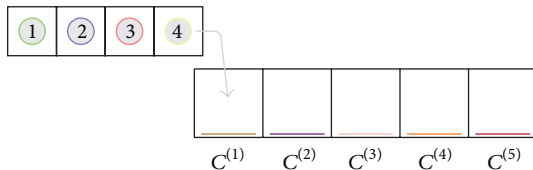


FIGURE 1: Overview of the elements of the MATCHIT automaton. The inlet is populated with four chemtainers, carrying the substances 1 to 4, respectively, and equipped with different chemtainer tags (color of circle). The tube consists of five cells which are equipped with individual cell tags. The color of the bar at the bottom of each cell implies which particular cell tag is used. The arrow shows how chemtainer at the front of the inlet would be inserted into the tube.

we discuss an algorithm—the compiler—to configure the MA in order to synthesize branched polymers. The remainder of this paper is structured as follows. In Section 2, we describe the MA in mathematical terms (Section 2.1), we suggest how branched polymers can be encoded into a mathematical graph structure (Section 2.2), and we describe how the MA configuration can be derived from a given target polymer (Section 2.3). In Section 3, we discuss some results obtained using these methods. We compare the MA to another artificial reactor and we propose improvements for the shortcomings of the MA. We finish this paper with a conclusion in Section 4 where we hypothesize how real instances of the MA could be used in the future.

2. Methods

2.1. Description of the MATCHIT Automaton. The MA can be seen as a model of the spatially separated compartments of the ER and Golgi that contain distinct sets of enzymes. In particular, the MA consists of the following entities. (a) *Substances*. That is, monomers, polymers or a mixture thereof. (b) *Reactions*. That is, translation rules that determine how two substances react into a third substance (cf. enzymes). (c) *Chemtainers*. That is, compartments containing a substance and equipped with one or more chemtainer tags. (d) *Cells*. That is, compartments containing chemtainers, equipped with a cell tag. (e) *Tags*. That is, a mechanism that induces chemtainers to stick to cells (chemtainer cell interaction) or to each other (chemtainer chemtainer interaction). (f) *Tube*. That is, a chain of concatenated cells. (g) *Inlet*. That is, the mechanism to stage chemtainers and insert them into the tube.

The MA operates in a discretized time domain. Figure 1 shows an overview of the MA and its entities. The general working principle is as follows. Chemtainers containing substances are staged inside an inlet and inserted into the tube consisting of various cells. The chemtainers move from cell to cell and can fuse under some conditions controlled by a tag system. In the event of a chemtainer fusion, the substances carried inside the fused chemtainers mix and can react with each other. The product of such a reaction remains inside the chemtainer, but this chemtainer can fuse with other chemtainers in subsequent steps. As we will demonstrate, this allows for a very granular control of the sequence of reaction steps contained inside the chemtainers.

The following sections describe the various aspects of the system in more detail.

2.1.1. Inlet, Tube, and Cells. The tube is a list of m cells $C^{(1)}, \dots, C^{(m)}$. The movement of chemtainers from cell to cell is governed by the following rules. (a) Chemtainers from the inlet are always inserted into $C^{(1)}$. (b) Chemtainers can only move from $C^{(j)}$ to $C^{(j+1)}$. (c) Chemtainers in $C^{(m)}$ remain there (end of the tube).

The inlet is a list of chemtainers; its purpose is to insert chemtainers into the tube. There is only one inlet and it is inserting chemtainers into the first cell of the tube, that is, $C^{(1)}$.

2.1.2. Chemtainer. A chemtainer c_i is a tuple (s_i, T_i, μ_i) where $s_i \in S$ is a substance and $T_i \subset T^{\text{Chem}}$ is a set of tags assigned to the chemtainer. Each chemtainer maintains a counter μ_i which tells how many time steps are left for the chemtainer to stick to the current cell before moving to the next cell. This allows for colocation and subsequent fusion of chemtainers.

2.1.3. Substance and Reaction. Exactly two substances can react with each other; the result of a reaction is a product substance. The notation for such a reaction is $(s_i + s_j \rightarrow s_p)$ where s_i and s_j react to the product s_p . Note that a reaction exists for any pair s_i and s_j ; if the reactants cannot react chemically, the resulting product is a mixture of s_i and s_j which is considered to be waste. S denotes the set of all substances and $S' \subset S$ is the set of all monomers.

2.1.4. Tag and Affinity. Tags are used to determine whether chemtainers can interact with each other and whether they can stick to cells. There are two kinds of tags: chemtainer-tags $t^{\text{Chem}} \in T^{\text{Chem}}$ and cell tags $t^{\text{Cell}} \in T^{\text{Cell}}$. The former are a property of chemtainers; the latter are a property of cells.

For $T = T^{\text{Chem}} \cup T^{\text{Cell}}$ there is an affinity function $a : T \times T \rightarrow \mathbb{N}_0$ which assigns a nonnegative integer to each combination of tags t_i and t_j such that $a(t_i, t_j) = a(t_j, t_i)$. This affinity a between two tags is interpreted as follows. (a) If $t_i \in T^{\text{Chem}}$ and $t_j \in T^{\text{Cell}}$, a indicates the total number of time steps the chemtainer takes to stick to the cell. $a = 0$ means that the chemtainer does not stick to the cell at all. (b) If $t_i, t_j \in T^{\text{Chem}}$, $a > 0$ indicates that two chemtainers can potentially fuse, and $a = 0$ indicates that two chemtainers cannot fuse. (c) If $t_i, t_j \in T^{\text{Cell}}$, a has no meaning since the cells are static.

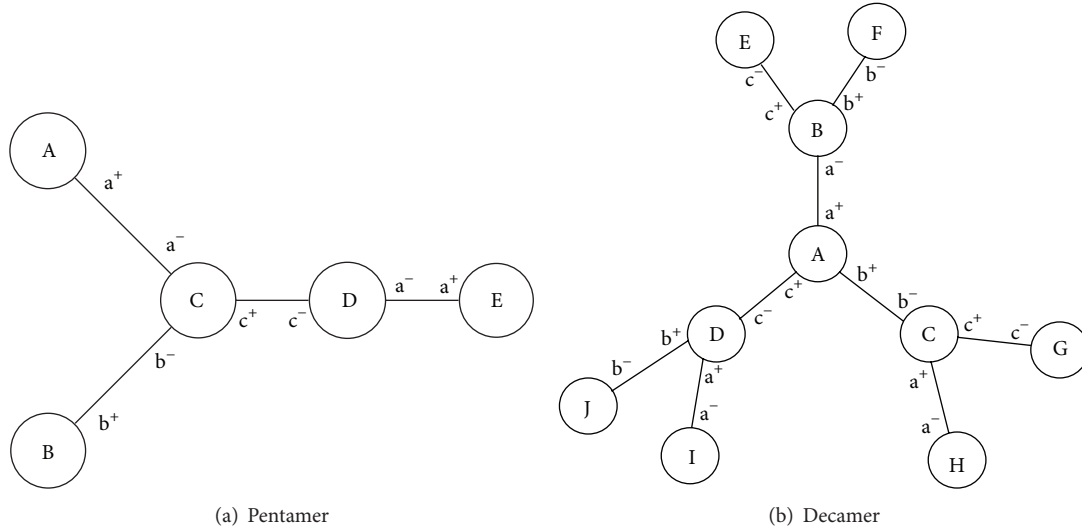


FIGURE 2: Examples of target polymers. Monomers are denoted by capital letters, and linkers associated with the bonds between monomers are denoted by lowercase letters and a sign; linkers with the same letter and opposite signs are matching linkers.

and cell tags never interact with each other. Hence this affinity is not defined.

2.1.5. Dynamics. The time domain is discretized into time steps $\tau \in \mathbb{N}_0$. At time step $\tau = 0$, the inlet is filled with n chemtainers and no chemtainer has moved or reacted yet.

Starting from time step $\tau = 1$ onwards, the following steps are executed in the order stated. (a) Move chemtainers from one cell to the next: the counter μ_i of each chemtainer c_i is decreased by 1. If $\mu_i = 0$ before decreasing, c_i is moved to the next cell. The chemtainer counter μ_i is reset according to the largest chemtainer cell affinity; that is,

$$\mu_i = \max_{t_j} (a(t_j, t_c)) \quad (1)$$

over all the chemtainer tags t_j of chemtainer c_j ; t_c is the chemtainer tag of the next cell. (b) Move one chemtainer from the front of the inlet to the first cell in the tube $C^{(1)}$. (c) Chemtainer fusion and reactions: for all chemtainers that are both sticking to the same cell j (i.e., with $\mu_j > 0$) and sticking together by the means of chemtainer chemtainer interaction, fuse the two chemtainers to a single one and apply the appropriate reaction to the chemtainer contents. Note that a cell may potentially contain more than two chemtainers, in which case the order of fusion has to be defined. The algorithm described below (see Section 2.3.3) however ensures that such a situation never occurs; hence we do not define the order of fusion for such a case.

After the consecutive execution of these three steps, τ is incremented by 1.

2.2. Representation of Chemtainer Content and Fusion Thereof. A graph structure is used to represent substances. The vertices of the graph represent monomers. Each vertex is associated with a label that is an element of S^l (i.e., the vertex color)

denoting a specific type of monomer. Edges between vertices represent bonds between monomers. In addition to these properties of a colored graph, each vertex contains a tuple of linkers. Edges are associated with these linkers such that the degree of a vertex cannot be greater than its number of linkers. If the degree of the vertex is less than the number of linkers, some linkers are considered to be active, allowing for a reaction.

The ordered nature of the linker tuple allows for a limited preservation of the geometric arrangement of the substance. Two substances are considered equal if and only if their graph representation is isomorphic (i.e., the structure of the substances is equal) and their linker tuples match.

In the event of a chemtainer fusion, the chemtainer resulting from the fusion inherits both sets of chemtainer tags from the original chemtainers. The following cases are to be distinguished when generating the graph of a product substance. (a) The active linkers of both substances do not match. (b) The active linkers of both substances match unambiguously. (c) The active linkers of both substances match ambiguously, such that different product substances may result depending on chance.

In case (a), the resulting substance is the graph union of both reactant substances. An example for this case is the reaction of the polymers shown in Figures 2(a) and 2(b). Both these reactants do not have any matching linkers. In fact, they do not have open linkers at all. In case (b), an edge is introduced between the two matching linkers in the graph union. For example, consider the polymer shown in Figure 2(a) and remove the edge between D and E. The resulting two reactants each have one exposed linker, a^- and a^+ , respectively. A reaction of these two reactants thus leads to the original polymer. Note that there is no ambiguity; the only site the linker a^- can bind to is the exposed linker a^+ . Case (c) however might be a source of waste because the substances may react to a product undesired for the synthesis.


```

(1) build_reaction_tree() begin
(2)    $g \leftarrow$  target polymer
(3)    $n \leftarrow$  root node
(4)   recursion( $g, n$ )
(5)   return  $n$ 
(6) recursion(graph  $g$ , node  $n$ ) begin
(7)   for each edge  $e$  of  $g$  do
(8)      $(v_1, v_2) \leftarrow$  vertices of  $e$ 
(9)      $(g_1, g_2) \leftarrow$  split( $g, e$ )
(10)     $P \leftarrow$  estimate_probability( $g_1, g_2, e$ )
(11)     $r \leftarrow$  add_child( $n, P$ )
(12)     $s_{v_1} \leftarrow$  add_child( $r, v_1$ )
(13)     $s_{v_2} \leftarrow$  add_child( $r, v_2$ )
(14)    recursion( $g_1, s_{v_1}$ )
(15)    recursion( $g_2, s_{v_2}$ )

```

ALGORITHM 1: Algorithm to recursively find possible paths of synthesis for a target polymer.

For example, consider the larger of the two reactants in the previous example, that is, the one consisting of the monomers A, B, C, and D (but not E). Following the same procedure and removing the edge between A and C result in two reactants that can form different products. Particularly The monomer A can bind not only to C but also to D. Since our goal is to maximize yield (i.e., to minimize waste), the algorithm presented in Section 2.3.3 ensures that case (c) is not encountered.

2.3. Compiling a Suitable MA for a Given Target Polymer. In this section, we present the steps that are required to find a suitable MA configuration (i.e., cells, tags, and affinities) to synthesize a given target polymer starting from its monomers. The goal is to only allow for steps that have unambiguously matching linkers and to ensure that chemtainer fusion is only possible for pairs of chemtainers in each cell, thus avoiding any waste. Note that we first need to identify the proper sequential chemical pathway at which point we can implement this pathway by configuring the MA appropriately.

Thus, this is a two-step process. First, the possible paths of synthesis are identified. Second, one of these paths is chosen and the MA configuration is derived. Due to the huge number of synthesis paths, the first step is time consuming and requires a lot of storage space. We show at the end of this section that a suitable path of synthesis can be found without enumerating all possible options.

2.3.1. Identifying Paths of Synthesis. Algorithm 1 generates possible paths of synthesis in a recursive top-down approach. The result is a tree of *reaction nodes* and *edge-split nodes*. Reaction nodes are used to estimate the quality of a particular reaction, and edge-split nodes are used to track the edge that is to be connected during the synthesis.

First, a root node is created and the recursion is started with the assembled target polymer as input graph (lines 2–4). One edge $e = (v_1, v_2)$ is removed from this input graph (lines 8–9), resulting in two disconnected subgraphs g_1 and g_2 representing two intermediate products. In general, there

is no guarantee that the removal of an edge will disconnect the graph; for example, removing the first edge from a cyclic structure will lead to an acyclic, connected graph. In the case of branched polymers however, disconnected sub-graphs can be assumed by definition. The probability p for a successful synthesis of the target polymer given the two sub-graphs is computed (line 10). A new reaction node is added as child of the current node in the tree and labelled with p . Furthermore, two edge-split nodes are added to the newly inserted reaction node and labelled with v_1 and v_2 , respectively (lines 11–13).

The same procedure is repeated recursively with each of the sub-graphs until no more edges can be removed, meaning that the polymer was decomposed to one of its monomers. On each level of the recursion, every edge of the respective input graph is removed as described above.

The runtime of this brute force algorithm scales factorial with the size of the target structure, which is worse than exponential. The runtime could be improved by dynamic programming techniques, where results of the recursion for sub-graphs are memorized in order to avoid multiple computations for the same structure. We have not taken this route, however, as we will introduce an additional constraint in the next section that will prune the search space such that the algorithm performs in linear time.

2.3.2. Selecting Suitable Path of Synthesis. A suitable path can be selected by walking the tree resulting from the algorithm in a manner that never adds a node with $p < 1$ to the fringe. This approach fails if no solution with 100% yield exists. In the case of branched polymers however, this case can be ruled out if the number of distinct linker pairs is at least as large as the degree of any vertex. This allows us to impose the constraint that the linkers associated with each vertex shall be unique. Under this constraint, we prove our claim as follows.

Let us start with the graph of the target polymer, P_0 , and remove an arbitrary edge. This leads to two disconnected sub-graphs P_1 and P_2 , each of which has one active linker. Because there is only one active linker per subgraph, mixing the substances represented by the two sub-graphs would

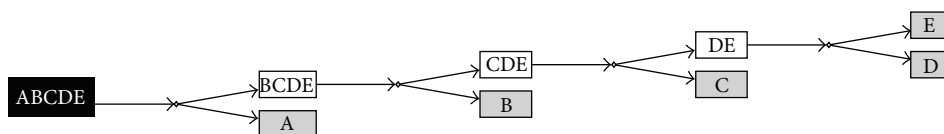


FIGURE 3: Example of a path of synthesis by sequential insertion for the target polymer shown in Figure 2(a). Edge-split nodes are drawn as rectangles and reaction nodes as small diamonds. The black rectangle denotes the target polymer, white rectangles denote intermediate products, and gray rectangles are monomers. The nodes are added to the tree from left to right as the algorithm progresses. Because the algorithm starts at the target polymer, decomposing it into its monomers, the tree shows the reverse of a synthesis.

result in the unambiguous synthesis of the target polymer. Without loss of generality, let us now continue with the first of the two sub-graphs, P_1 , and remove an edge from the vertex with the active linker. This decomposition leads to two sub-graphs P_3 and P_4 . As before, P_1 can be synthesized by mixing P_3 and P_4 because the newly exposed active linkers are different from the previously exposed ones due to the constraint. Moreover, we can ensure that neither P_3 nor P_4 can react with P_1 by isolating subreactions in different chemtainers. In general, the repeated application of this procedure eventually leads to a monomer which cannot be decomposed any further. Because the constraint ensures that each decomposition can be uniquely synthesized by mixing the substances represented by the resulting sub-graphs, $p = 1$ is guaranteed for each step.

2.3.3. Configuration of the MA. Once a suitable path of synthesis is chosen, the MA is configured. A suitable configuration results in the synthesis of the target polymer from monomers and can be found as follows. First, the reaction nodes of the selected path are visited in a breadth-first manner. This means that the tree is visited level by level, starting from the target polymer at the root, visiting the reactions of the intermediate products. For each node that is visited, a cell is created. Because the root represents the last step of the synthesis, the first node that is visited corresponds to the last cell of the tube and the last node that is visited corresponds to the first cell of the tube. The m cells created in that manner are equipped with a unique cell tag.

Second, all monomers of the target polymer are filled into chemtainers. If the molecular graph corresponding to the target polymer consists of n nodes, n chemtainers are created, each containing the respective monomer. Note that two chemtainers are created if two nodes with a particular color are present. The n chemtainers created in this manner are equipped with a unique chemtainer tag.

Third, the affinities are defined. As a consequence of the breadth-first traversal, the first cell corresponds to a reaction node that has two external children. These two children correspond to two chemtainers c_i and c_j containing monomers, which are to be staged inside the inlet. Hence, a positive affinity $a(t_i, t_j) > 0$ is defined for the interaction between the two tags of the chemtainers c_i and c_j . Furthermore, c_i and c_j shall stick to the first cell, and therefore an affinity a for the interaction between the cell tag of cell 1 and the tag of chemtainer c_i is defined. The same affinity is also defined for the interaction between the cell tag and the tag of c_j , such that

(a) both chemtainers stick together and (b) both chemtainers stick to cell 1, thus satisfying the requirements for the fusion of c_i and c_j in the first cell. The selection of a suitable value for a is discussed below.

In general, a cell is responsible for either (a) the fusion of chemtainers containing a monomer each, (b) the fusion of a chemtainer containing a monomer with a chemtainer containing an intermediate product, or (c) two chemtainers containing intermediate products each. If a chemtainer contains a monomer, its tag is known because it was assigned while staging the inlet and can therefore readily be used as discussed above. If a chemtainer contains an intermediate product, it is the result of a chemtainer fusion since all original chemtainers were containing monomers. As a consequence, such a chemtainer will carry each of the tags of the fused original chemtainers and a tag can be found by a lookup in the subtree of the corresponding reaction node.

The following considerations are made to compute the affinity of chemtainer-cell interaction a . If a is too small, the counter μ_i of a chemtainer waiting for fusion with another chemtainer may elapse before the other chemtainer reaches the cell. Therefore, the rule

$$a_i = n + \sum_{j=1}^{i-1} a_j \quad (2)$$

is suggested for any affinity a_i involving the cell tag of the i th cell. The rule takes into account the release of the n chemtainers from the inlet, as well as the maximal time a chemtainer could spend sticking in previous cells.

To summarize, the procedure described above ensures that (a) there is a bijection between reaction nodes and cells, (b) the reactions occur in the proper order, (c) the proper conditions with respect to tag affinity are created for successful chemtainer fusion and subsequent reaction of chemtainer content, and (d) chemtainer fusion does not occur in the absence of a planned reaction.

3. Results and Discussion

The strategy to find all paths of synthesis (recall Algorithm 1) and to configure the MA was implemented and a simulator for the MA was developed. A path of synthesis for the target polymer in Figure 2(a), as computed by Algorithm 1, is displayed in Figure 3. Note that any path of synthesis is a

tree by itself. The one shown in the figure is equivalent to a synthesis by sequential insertion because at least one of the two children of each reaction node is a monomer. Although not investigated further in this paper, it did not escape our attention that such a “bushy” path of synthesis would allow the synthesis of several intermediate products in parallel, leading to faster completion of the target polymer.

We also applied our algorithm to the decamer in Figure 2(b) in which case no path of synthesis by sequential insertion without loss (i.e., $p < 1$) was found. This was expected because the polymer was carefully designed to demonstrate that some polymers cannot be synthesized if the number of available linkers is restricted. However the polymer satisfies the constraint of distinct linkers per monomer, and hence our algorithm was able to derive a path of synthesis that does not result in any waste and was also able to compute a configuration of the MA that led to the successful synthesis in simulation.

Our work shows the advantages of compartmentalization for the synthesis of branched polymers. If the number of linkers is limited, we can suppress unwanted reactions by spatially separating the substances. A disadvantage of our method is that Algorithm 1 is computationally heavy and requires a lot of memory as it lays out the whole tree of all possible paths of synthesis. This allows us to find the path which leads to maximal yield by multiplying the probabilities p as we traverse the tree, but in the case of branched polymers we know that there exists a path for which $p = 1$ holds at every step as long as the constraint of distinct linkers per vertex is met. Hence, building the full tree is not necessary. Instead, we can simply remove an arbitrary edge from the target polymer and continue to remove edges from the same vertices as discussed in Section 2.3.2. This leads to a path of synthesis with maximal yield in linear time.

The representation of substances based on a graph entails some limitations. First, the substances are assumed to be planar. Second, this representation fails to address the selective synthesis of stereoisomers and similar structures. We note however that the substances could be represented in any way as long as products can be split into reactants, and the probability of a successful synthesis of the product out of these reactants can be estimated. This includes extensions of the graph structure used in this paper as well as even physical simulations.

The MA is modeled as a deterministic device. An according stochastic structure has been studied by B. Reller and presented in [22]. Reller proposed, also *in silico*, a self-assembling microreactor, a two-dimensional structure composed of chemtainers that mediate different linking reactions. This microreactor constitutes a spatially heterogeneous reaction environment which, by its spatial structure, controls or at least amplifies desired reaction pathways. In contrast to the MA, no compiler could be given for Reller’s approach.

4. Conclusion

In this paper, we have presented a mathematical framework to optimize chemical reactions of branched polymers at

high yields by means of compartmentalization. It is emphasized that today no complete implementation of an MA exists. However, MATCHIT delivered important technologies towards an MA. As already mentioned in Section 1 (see references given there), candidates for chemtainers investigated in MATCHIT are vesicles and oil droplets (in water) for which a range of functionalities relevant to the implementation of an MA have been demonstrated. Together with novel types of MEMS devices also resulting from MATCHIT [15], a physical implementation of an MA may be within reach in the near future.

Why may one be interested in the construction of such a system instead of using a series of test tubes in a lab? Besides aspects of automation, there is another, fundamental rationale for investigating designs with the potential for miniaturization [22]. We hypothesize that miniaturized version of the MATCHIT automaton could provide opportunities to use types of catalysts not used in conventional chemical process management. Assume that one synthesizes oligomers with some sort of catalytic activity, catalytic activity that is used in a subsequent step. It may well be the case that the environment in which this catalyst is synthesized (or activated) is different from the environment in which the catalyst has to act. In a macroscopic laboratory, this implies that the catalyst has to be transferred from one environment to the other, a process that takes time. This means that catalysts have to fulfill two requirements: first, they have to be efficient, and, second, they have to be stable. On a microscopic length scale, where transport involves transport over micrometers, ten seconds is quite a time (even if this transport happens by diffusion). This means that in microscopic reactors, catalysts, if they are produced *in situ*, only have to be efficient but no longer need to be particularly stable. An average life time of minutes to fractions of a second is sufficient for typical lengths ranging from several ten μm down to part of μm , depending on passive diffusion and active processes such as electrical fields [15]. As a result, many more molecules qualify as potential catalysts. The possibility that microreactors could enable the use of metastable catalysts in a technically feasible manner is a more than sufficient justification for the study of such systems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007–2013) under Grant Agreement no. 249032 (project acronym: MATCHIT), the Danish National Research Foundation, and the University of Southern Denmark through the Center for Fundamental Living Technology (FLinT). M. Hadorn was supported by the Swiss National Science Foundation (SNSF).

References

- [1] MATCHIT Matrix for Chemical IT, <http://fp7-matchit.eu>.
- [2] M. Amos, P. Dittrich, J. McCaskill, and S. Rasmussen, "Biological and chemical information technologies," in *Proceedings of the 2nd European Future Technologies Conference and Exhibition (FET '11)*, vol. 7, pp. 56–60, May 2011.
- [3] M. D. Symes, P. J. Kitson, J. Yan et al., "Integrated 3D-printed reactionware for chemical synthesis and analysis," *Nature Chemistry*, vol. 4, no. 5, pp. 349–354, 2012.
- [4] G. Villar, A. D. Graham, and H. Bayley, "A tissue-like printed material," *Science*, vol. 340, no. 6128, pp. 48–52, 2013.
- [5] G. Seelig, D. Soloveichik, D. Y. Zhang, and E. Winfree, "Enzyme-free nucleic acid logic circuits," *Science*, vol. 314, no. 5805, pp. 1585–1588, 2006.
- [6] D. Y. Zhang and E. Winfree, "Control of DNA strand displacement kinetics using toehold exchange," *Journal of the American Chemical Society*, vol. 131, no. 47, pp. 17303–17314, 2009.
- [7] L. Qian and E. Winfree, "Scaling up digital circuit computation with DNA strand displacement cascades," *Science*, vol. 332, no. 6034, pp. 1196–1201, 2011.
- [8] D. van Swaay and A. deMello, "Microfluidic methods for forming liposomes," *Lab on a Chip*, vol. 13, no. 5, pp. 752–767, 2013.
- [9] M. Hadorn and P. Eggenberger Hotz, "DNA-mediated self-assembly of artificial vesicles," *PLoS ONE*, vol. 5, no. 3, article e9886, 2010.
- [10] M. Hadorn, E. Boenzli, K. T. Sffrensen, H. Fellermann, P. E. Hotz, and M. M. Hanczyc, "Specific and reversible DNA directed self-assembly of oil-in-water emulsion droplets," *Proceedings of the National Academy of Sciences*, vol. 109, no. 50, pp. 20320–20325, 2012.
- [11] F. Caschera, S. Rasmussen, and M. M. Hanczyc, "An oil droplet division-fusion cycle," *ChemPlusChem*, vol. 78, no. 1, pp. 52–54, 2013.
- [12] T. Sunami, F. Caschera, Y. Morita et al., "Detection of association and fusion of giant vesicles using a fluorescence-activated cell sorter," *Langmuir*, vol. 26, no. 19, pp. 15098–15103, 2010.
- [13] F. Caschera, T. Sunami, T. Matsuura, H. Suzuki, M. M. Hanczyc, and T. Yomo, "Programmed vesicle fusion triggers gene expression," *Langmuir*, vol. 27, no. 21, pp. 13082–13090, 2011.
- [14] M. Hadorn, E. Boenzli, P. E. Hotz, and M. M. Hanczyc, "Hierarchical unilamellar vesicles of controlled compositional heterogeneity," *PLoS One*, vol. 7, no. 11, article e50156, 2012.
- [15] P. F. Wagler, U. Tangen, T. Maeke, and J. S. McCaskill, "Field programmable chemistry: integrated chemical and electronic processing of informational molecules towards electronic chemical cells," *BioSystems*, vol. 109, no. 1, pp. 2–17, 2012.
- [16] R. D. Cummings, "The repertoire of glycan determinants in the human glycome," *Molecular BioSystems*, vol. 5, no. 10, pp. 1087–1104, 2009.
- [17] A. Varki, "Biological roles of oligosaccharides: all of the theories are correct," *Glycobiology*, vol. 3, no. 2, pp. 97–130, 1993.
- [18] K. M. Koeller and C.-H. Wong, "Complex carbohydrate synthesis tools for glycochemists: enzyme-based approach and programmable one-pot strategies," *Glycobiology*, vol. 10, no. 11, pp. 1157–1169, 2000.
- [19] H. C. Kolb, M. G. Finn, and K. B. Sharpless, "Click chemistry: diverse chemical function from a few good reactions," *Angewandte Chemie—International Edition*, vol. 40, no. 11, pp. 2004–2021, 2001.
- [20] H. C. Kolb and K. B. Sharpless, "The growing impact of click chemistry on drug discovery," *Drug Discovery Today*, vol. 8, no. 24, pp. 1128–1137, 2003.
- [21] M. D. Best, "Click chemistry and bioorthogonal reactions: unprecedented selectivity in the labeling of biological molecules," *Biochemistry*, vol. 48, no. 28, pp. 6571–6584, 2009.
- [22] R. M. Fuchsli, A. Dzyakanchuk, D. Flumini et al., "Morphological computation and morphological control: steps toward a formal theory and applications," *Artificial Life*, vol. 19, no. 1, pp. 9–34, 2013.
- [23] P. Dittrich, J. Ziegler, and W. Banzhaf, "Artificial chemistries—a review," *Artificial Life*, vol. 7, no. 3, pp. 225–275, 2001.
- [24] COBRA Project: EU FP7 project, <http://www.cobra-project.eu>.