On Abstract Shapes of RNA

Markus E. Nebel, Anika Scheid*
Fachbereich Informatik, Technische Universität Kaiserslautern
Gottlieb-Daimler-Straße 48, D-67663 Kaiserslautern, Germany
{nebel,a_scheid}@informatik.uni-kl.de

Abstract

As any RNA sequence can be folded in many different ways, there are lots of different possible secondary structures for a given sequence. Most computational prediction methods based on free energy minimization compute a number of suboptimal foldings and we have to identify the native structures among all these possible secondary structures. For this reason, much effort has been made to develop approaches for identifying good predictions of RNA secondary structure. Using the abstract shapes approach as introduced by Giegerich et al. [GVR04], each class of similar secondary structures is represented by one shape and the native structures can be found among the top shape representatives. In this article, we derive some interesting results answering enumeration problems for abstract shapes and secondary structures of RNA. We start by computing asymptotical representations for the number of shape representations of length n. Our main goal is to find out how much the search space can be reduced by using the concept of abstract shapes. To reach this goal, we analyze the number of secondary structures and shapes compatible with an RNA sequence of length n under the assumption that base pairing is allowed between arbitrary pairs of bases analytically and compare their exponential growths. Additionally, we analyze the number of secondary structures compatible with an RNA sequence of length n under the assumptions that base pairing is allowed only between certain pairs of bases and that the structures meet some appropriate conditions. The exponential growth factors of the resulting asymptotics are compared to the corresponding experimentally obtained value given in [GVR04].

1 Introduction

Ribonucleic acid (RNA) is a single-stranded nucleic acid. The basis structural units of such nucleic acids are formed by nucleotides, where each nucleotide is composed of a phosphate group, a sugar group (ribose) and one of the four bases adenine (A), cytosine (C), guanine (G) and uracil (U). An RNA single-strand is formed by linking together the nucleotide units. In fact, the linear structure of the RNA molecule, modeled as a word over the alphabet $\Sigma = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}$ representing the four different types of nucleotides, is formed by creating *phosphodiester bonds*. The specific sequence of bases along this chain is called the *primary structure* of the molecule.
Any of these linear primary structures may form a lot of different more complex structures by folding. The reason for folding is that in addition to the phosphodiester bonds between neighboured nucleotides of the primary stucture, two bases that are not neighboured may form other bonds. More precisely, the complementary bases adenine (A) and uracil (U) resp. cytosine (C) and guanine (G) form stable base pairs with each other by creating hydrogen bonds. These base pairs are called *Watson-Crick pairs*. In addition to these stable Watson-Crick base pairs, there may occur weaker base pairs formed by the non-complementary bases guanine (G) and uracil (U), which are called *wobble GU pairs*. Other pairs may also occur, but they are not as stable as the Watson-Crick and wobble GU pairs.
By pairing of nucleotides according to these rules, the linear primary structure of an RNA molecule is folded into a three-dimensional conformation, with helices in three dimensions. This three-dimensional conformation is called the *tertiary structure* of the molecule, which in many cases determines the function of the molecule. It is costumary in science to allow only non-crossing (nested) base pairs, such that the primary structure is folded into a two-dimensional conformation, called the *secondary structure*.

As determining the tertiary structure is computationally complex, it has proven convenient to first search for the secondary structure, for which only a subset of the hydrogen bonds is considered, such that the

*Corresponding author.

molecule can be modeled as a planar graph. Investigating the secondary structure of RNA molecules is important, as much of the 3D structure is determined by the base-pairing interactions *in the plane*. In addition, the experimental determination of RNA tertiary structures is usually time-consuming and expensive and therefore, much effort has been made to create approaches for the computational prediction of RNA secondary structures over the last decades.

The most common approach for predicting the secondary structure of an RNA molecule is free energy minimization. As in nature every RNA molecule seeks to achieve a minimum of free energy by folding into a higher-dimensional conformation, it is assumed that the correct structure is the one with the lowest free energy. Hence, many prediction methods use free energy as their metric and try to compute a conformation of minimum free energy.

The most successful and popular method for energy minimization over the last 30 years has been the use of dynamic programming algorithms. The pioneering work in this domain was published in 1978 by Nussinov et al. [NPGK78]. In this paper, the authors introduced an efficient dynamic programming algorithm which used a simple free energy function **E** that is minimized when the secondary structure contains the maximum number of complementary base pairs. Hence, this approach simplified the problem of folding a primary structure into a structure with minimum free energy to the problem of finding a structure with maximum number of base pairs, and the computed folding contained the maximum number of base pairs that could be found for the entire primary structure.

By utilizing a simple method for estimating the free energy of loops found in RNA secondary structures based on their sequence [TUL71], the folding rules of this dynamic programming algorithm for maximal matching were modified to allow an estimate of the free energy of loop structures based on sequence data [NJ80]. This means that hydrogen bond potential energies are computed for each base pair, such that the algorithm computes one structure with the lowest free energy **E**.

A dynamic programming algorithm for folding an RNA molecule that finds a conformation of minimum free energy using thermodynamics and auxiliary information [ZS81] was presented in 1981 by Zuker and Stiegler. This algorithm uses loop-dependent energy rules to compute the free energy of each loop, and the overall free energy of a secondary structure is given by the sum of the free energies of its loops. During the following years, this dynamic programming algorithm based on thermodynamic parameters has been improved several times [SKMC83, ZS84, Zuk89a].

However, due to imprecisions in the energy rules and the thermodynamic parameters, as well as the fact that certain chemical aspects, like for example the influence of enzymes or the effect of cotranscriptional folding, have not been incorporated into dynamic programming algorithms, the predicted optimal (minimum free energy) structure was often not the native one. Therefore, there was an urgent need to additionally predict suboptimal foldings. For this reason, in 1989, an algorithm for determining RNA secondary structures within any prescribed increment of the computed global minimum free energy was introduced [Zuk89b].

Finally, it remains to mention that all these algorithms only work for secondary structures without pseudoknots, as they cannot predict crossing base pairs. Pseudoknots [PB89, AvdBvBP90, GW90, DPD92, Ple94], formed by two crossing base pairs, are often considered as belonging to the tertiary structure and are usually not permitted in definitions of secondary structures. But as pseudoknots are important to the function of several kinds of RNAs, much effort has been made to develop algorithms for predicting RNA secondary structures that contain pseudoknots. In fact, a rather general algorithm for predicting structures with pseudoknots, which is capable of predicting nearly all known classes of pseudoknots, was presented in 1999 by Rivas and Eddy [RE99]. But for N the size of the primary structure this algorithm has a theoretical worst-case complexity of $\mathcal{O}(N^6)$ in time and $\mathcal{O}(N^4)$ in storage and is thus only practical for very short RNA sequences. The currently most general algorithm (which has a better theoretical worst-case complexity) has been presented recently by Metzler and Nebel [MN08]. However, this algorithm is not based on free energy minimization.

A review on how RNA folding algorithms work and why they can't deal with pseudoknots can be found in [Edd04].

Using an RNA folding algorithm for the computational prediction of RNA secondary structures which additionally creates suboptimal solutions, we have to search a huge set for native solutions. However, this set of suboptimal foldings usually contains lots of similar structures and we are only interested in structures with more fundamental differences. For this reason, the concept of *abstract shapes* was introduced by Giegerich et al. [GVR04]. Abstract shapes are homomorphic images of secondary structures and each shape comprises a class of similar structures. Furthermore, an abstract shape class has a representative structure with minimum free energy.

Consequently, using this concept of abstract shapes, we can find the native structures among the top shape representatives. This means that we do not have to search for native structures in the huge set of

suboptimal minimum free energy structures anymore, but in the much smaller set of shape representatives. Based on this approach, an integrated RNA analysis package called RNAShapes has been developed [SVR+06b, SVR+06a]. This software package integrates three analysis tools based on the abstract shape approach: the analysis of shape representatives [GVR04], the calculation of shape probabilities [VGR06] and the consensus shapes approach [RG05]. It also has a number of useful features like for example the ability to compute suboptimal foldings.

2 Formal Framework

In this section we present the formal framework needed for our investigations.

2.1 RNA Secondary Structures

As secondary structures are two-dimensional, they can be modeled as planar graphs. A formal definition is given as follows:

Definition 2.1 ([Wat78]) *A* secondary structure *of size n is a loop free graph on the set of n labeled points $\{1, 2, \ldots, n\}$ such that the adjacency matrix $A = (a_{ij})$ (which is defined in the usual way by $a_{ij} = 1$ if i and j are adjacent, and $a_{ij} = 0$ otherwise, with $a_{ii} = 0$) has the following three properties:*

1. *$a_{i,i+1} = 1$ for $1 \leq i \leq n-1$.*
2. *For each fixed i, $1 \leq i \leq n$, there is at most one $a_{i,j} = 1$ where $j \neq i \pm 1$.*
3. *If $a_{i,j} = a_{k,l} = 1$, where $i < k < j$, then $i \leq l \leq j$.*

Note that constraint 3 of Definition 2.1 ensures that these graph representations remain planar, as pseudoknots are not permited in secondary structures due to this constraint.
Any secondary structure consists of several substructures and therefore can be decomposed into different structural components. The simplest substructures are introduced by the following definition.

Definition 2.2 ([Wat78]) *Suppose A is the adjacency matrix for a secondary structure of size n.*

1. *The point j is said to be* paired *if there is some point $i \neq j \pm 1$ such that $a_{i,j} = 1$.*
2. *The sequence $i+1, i+2, \ldots, j-1$ is a* loop*, if $i+1, i+2, \ldots, j-1$ are all unpaired and $a_{i,j} = 1$. The pair (i, j) is said to be the* foundation of the loop*.*
3. *The sequence $i+1, i+2, \ldots, j-1$ is a* bulge *if $i+1, i+2, \ldots, j-1$ are all unpaired, i and j are both paired ,and $a_{i,j} \neq 1$.*
4. *An* interior loop *is two bulges $i+1, i+2, \ldots, j-1$ and $k+1, k+2, \ldots, l-1$ such that $a_{i,l} = 1$ and $a_{j,k} = 1$. (Here $i < j < k < l$.)*
5. *A* join *is a bulge $i, i+1, \ldots, j$ such that $a_{k,l} = 1$ for $k < i$ implies $l \leq i$, and $a_{k,l} = 1$ for $k > j$ implies $l \geq j$.*
6. *A* tail *is a sequence $1, 2, \ldots, j$ resp. $j, j+1, \ldots, n$, where $1, 2, \ldots, j$ resp. $j, j+1, \ldots, n$ are unpaired and $j+1$ resp. $j-1$ is paired.*
7. *A* ladder *(or* helical region*) is built by two sequences $i+1, i+2, \ldots, i+j$ and $k+1, k+2, \ldots, k+j$ such that $i+j+1 < k$, $a_{i+l,k+j-l+1} = 1$ for $1 \leq l \leq j$ and $a_{i,k+j+1} = a_{i+j+1,k} = 0$. If $i+j+3 = k+1$, this last requirement is dropped.*
8. *A hairpin is the longest sequence $i+1, i+2, \ldots, j-1$ containing exactly one loop such that $a_{i+1,j-1} = 1$ and $a_{i,j} = 0$. The paired points $i+1$ and $j-1$ will be called the* foundation *of the hairpin.*

The next theorem shows that every RNA secondary structure can be built using the previously defined structural components.

Theorem 2.1 ([Wat78]) *Any secondary structure can be uniquely decomposed into loops, ladders, bulges, and tails. Alternatively, every secondary structure can be uniquely decomposed into hairpins and ladders, bulges, and tails which are not members of a hairpin.*

Some of the previously defined structural components of RNA secondary structures together might form more complex substructures, which are called *multiloops*. A definition of these structural components can be given as follows:

Definition 2.3 *Suppose A is the adjacency matrix for a secondary structure of size n.*
For every $k \geq 2$, a multiloop *is a sequence $j_0 + 1, \ldots, i_1, \ldots, j_1, \ldots, i_2, \ldots, j_k, \ldots, i_{k+1} - 1$ such that $a_{j_0,i_{k+1}} = 1$, $a_{i_1,j_1} = 1$, …, $a_{i_k,j_k} = 1$ and each of the k sequences $i_1, \ldots, j_1$, …, $i_k, \ldots, j_k$ contains at least one loop. Furthermore, if $j_l < i_{l+1}$ for $l \in \{0, \ldots, k\}$, then $j_l + 1, \ldots, i_{l+1} - 1$ are all unpaired. (Here $1 \leq j_0 \leq i_1 < j_1 \leq i_2 < \ldots < j_k \leq i_{k+1} \leq n$.)*
The pair (j_0, i_{k+1}) is said to be the foundation *of the multiloop and the k sequences $i_1, \ldots, j_1$, …, $i_k, \ldots, j_k$ are called the* helices *of the multiloop.*

Furthermore, every secondary structure of an RNA molecule that is not completely unpaired forms an *external loop*, which can be a seen as a list of adjacent substructures or adjacent structural components of this secondary structure.

Alternatively, RNA secondary structures can be modeled as strings over the alphabet $\Sigma := \{(,), .\}$, where a dot represents an unpaired nucleotide and a pair of corresponding brackets ( ) represents two bases in the RNA molecule that are paired. A formal definition of these dot-bracket representations of RNA secondary structures is given as follows:

Definition 2.4 ([VC85]) *For $\Sigma := \{(,), .\}$ and $w \in \Sigma^*$ let $|w|_x$ for $x \in \Sigma$ denote the number of occurrences of symbol x in w. Then a word $w \in \Sigma^n$ is a secondary structure of size n if w satisfies the three following conditions:*

1. *For every factorization $w = u \cdot v$, $|u|_(\geq |u|_)$.*
2. *$|w|_(= |w|_)$.*
3. *w has no factor ( ).*

It remains to mention that words over the alphabet $\{(,)\}$ which satisfy the first two conditions are known as *semi-Dyck words*, whereas words over the alphabet Σ satisfying these first two conditions are known as *Motzkin words*. The third condition ensures that a hairpin loop consists of at least one unpaired nucleotide. Thus, in this model a hairpin loop has a minimum length of one, although in reality, hairpin loops of length less than tree are impossible and do not form. Nevertheless, the planar graph model also commits a minimum loop length of one for hairpin loops, and there is a one-to-one correspondence between the planar graph model for secondary structures and this model of string representations, which yields the following definition, partially given in [Neb04]:

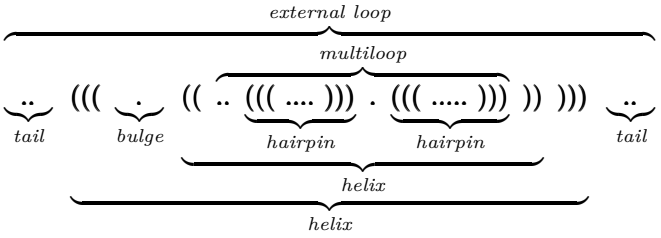
Definition 2.5 *Let w be a secondary structure of size n and let w_i denote the i-th symbol of w, $1 \leq i \leq n$.*

1. *The subword $v = w_{i+1} \ldots w_{j-1}$ is a* (hairpin) loop*, if $v \in \{.\}^+$ and $w_i w_j =$ ( ) is a corresponding pair of brackets of w.*
2. *The subword $v = w_{i+1} \ldots w_{j-1}$ is a* bulge*, if $v \in \{.\}^+$ and $w_i w_j \in \{(,)\}^2$ but $w_i w_j$ does not represent a pair of corresponding brackets of w.*
3. *An* interior loop *is two subwords (bulges) $u = w_{i+1} \ldots w_{j-1}$ and $v = w_{k+1} \ldots w_{l-1}$ such that $u \in \{.\}^+$, $v \in \{.\}^+$ and $w_i w_l =$ ( ), $w_j w_k =$ ( ) are corresponding pairs of brackets of w, where $i < j < k < l$.*
4. *A* join *is a subword (bulge) $v = w_i w_{i+1} \ldots w_j$ such that $v \in \{.\}^+$ and a corresponding pair of brackets $w_k w_l =$ ( ) of w for $k < i$ resp. $k > j$ implies $l \leq i$ resp. $l \geq j$.*
5. *A* tail *is a prefix $v = w_1 \ldots w_i$ resp. a suffix $v = w_j \ldots w_n$ such that $v \in \{.\}^+$ and w_{i+1} resp. w_{j-1} is in $\{(,)\}$.*
6. *A* ladder *(or* helical region*) consists of two maximal subwords u, v such that $u = w_i \ldots w_{i+c}$ and $v = w_j \ldots w_{j+c}$ and $w_{i+k} w_{j+c-k}$ is a pair of corresponding brackets, $0 \leq k \leq c$. The length of a ladder is given by $c + 1$.*
7. *A* hairpin *is a subword $v = w_{i+1} \ldots w_{j-1}$ such that v contains exactly one loop, $w_{i+1} w_{j-1}$ is a corresponding pair of brackets of w, but $w_i w_j$ is none.*

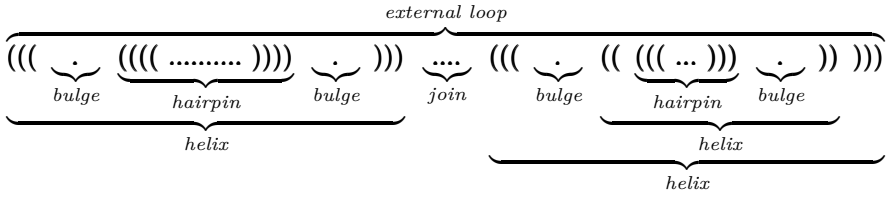
8. *For every $k \geq 2$, a* multiloop *is a subword $u = w_{j_0+1} \dots w_{i_1} \dots w_{j_1} \dots w_{i_2} \dots w_{j_k} \dots w_{i_{k+1}-1}$ such that $w_{j_0} w_{i_{k+1}}$, $w_{i_1} w_{j_1}$, …, $w_{i_k} w_{j_k}$ are pairs of corresponding brackets of w and each of the k subwords $w_{i_1} \dots w_{j_1}$, …, $w_{i_k} \dots w_{j_k}$ contains at least one loop. Furthermore, if $j_l < i_{l+1}$ for $l \in \{0, \dots, k\}$, then $w_{j_l+1} \dots w_{i_{l+1}-1} \in \{.\}^+$. (Here $1 \leq j_0 \leq i_1 < j_1 \leq i_2 < \dots < j_k \leq i_{k+1} \leq n$.) The k subwords $w_{i_1} \dots w_{j_1}$, …, $w_{i_k} \dots w_{j_k}$ are called the* helices *of the multiloop.*

Example 2.1 *Four different (decompositions of) dot-bracket representations of secondary structures are given as follows:*

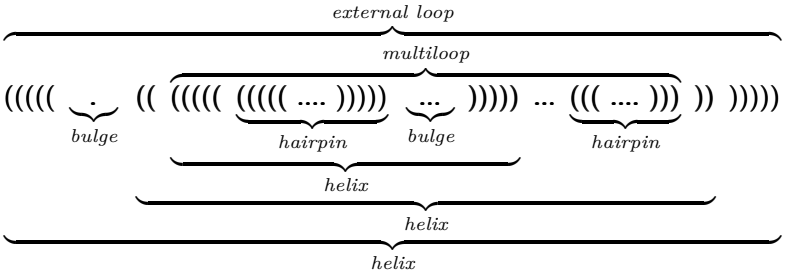
1. secondary$_1$*:*



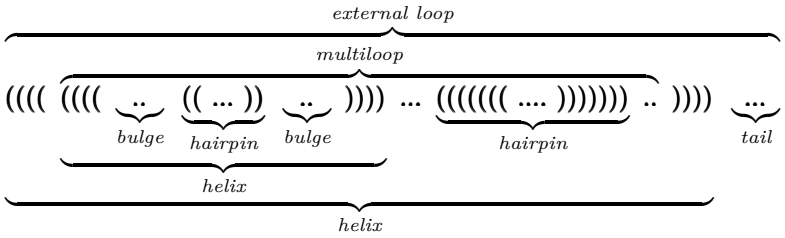
2. secondary$_2$*:*



3. secondary$_3$*:*



4. secondary$_4$*:*



Note that the shown decompositions of these dot-bracket representations of secondary structures will be used to illustrate the construction of abstract shapes *of RNA in the sequel.*

The reading order of secondary structures in dot-bracket representation is from left to right, which corresponds to the reading order of the primary structure. The reading order of the primary structure is due to the chemical structure of the RNA molecule. In the sequel, the terms secondary structure, secondary structure encoding or secondary structure in dot-bracket representation will be used interchangeably.
It should be clear that these dot-bracket representations abstract from the primary structure, as they only consider the number of base pairs and unpaired bases and their positions.

2.2 Abstract Shapes of RNA

In this subsection, we want to give all the definitions and ideas concerning abstract shapes that will be needed for our further investigations.

2.2.1 Shape Definitions

There are five shape types for five different levels of abstraction. While two of them, namely type 1 and type 5 shape abstractions (also called π' and π shapes, respectively), were formally defined by a tree homomorphism, more precisely by a shape abstraction mapping π' and π, respectively, in [GVR04], all five different shape levels were informally described in [SVR$^+$06a].
Common to all levels is that they abstract from loop and ladder lengths, while generally retaining nesting and adjacency of helices, but disregarding their size and concrete position in the primary structure. In general, helical regions are depicted by a pair of opening and closing squared brackets **[** resp. **]** and unpaired regions are represented by a single underscore _. In the most accurate shape type (type 1), all structural components contribute to the shape representation, the succeeding shape types gradually increase abstraction by not including certain unpaired regions or combining nested helices.
We now want to describe each of the five types (as defined in [SVR$^+$06a]) separately, ordererd by their degree of abstraction.

Type 1 (Most accurate):

nesting pattern for all loop types and all unpaired regions

This means that all helical regions are depicted by a pair of opening and closing squared brackets and all unpaired regions are represented as a single underscore. Thus, all structural components contribute to this shape representation, nesting and adjacency of helices are retained. Accordingly, this shape type only abstracts from loop and ladder lengths.

Type 2:

nesting pattern for all loop types and unpaired regions in external loop and multiloop

Consequently, all helical regions (ladders) are depicted by a pair of opening and closing squared brackets and unpaired regions in external loops (tails and joins) and multiloops (bulges) are represented as a single underscore. This means that in this shape representation, nesting and adjacency of helices is still retained, but in difference to type 1 shape representations, not all structural components contribute to this shape representation, since underscores representing hairpin loops are omitted, as well as underscores representing single bulges and internal loops which are interruptions of ladders. Thus, this type does not only abstract from loop and ladder lengths, but also from unpaired regions which close ladders (hairpin loops) and interrupt ladders (some bulges and internal loops).

Type 3:

nesting pattern for all loop types, but no unpaired regions

Shape representations of type 3 thus also retain nesting and adjacency of helices, since all helical regions are depicted by a pair of opening and closing squared brackets. But in contrast to the previously introduced two types, no unpaired regions are considered (except in the case of the completely unpaired structure). This means that this shape representation completely abstracts from single-stranded regions. The difference between type 2 shape representations and those of type 3 is that in the latter, all underscores representing unpaired regions in external loops (tails and joins) and multiloops (bulges) are omitted. Equally, the difference between type 1 and type 3 shapes is that in type 1 shape representations, all single-stranded regions are represented as an underscore, whereas in type 3 shape representations, all these underscores are omitted.

Type 4:

helix nesting pattern and unpaired regions in external loop and multiloop

Accordingly, nested helices are combined in this shape representation. This means that even interrupted ladders are depicted by only one pair of opening and closing squared brackets and like for type 2 shapes, only unpaired regions in external loops (tails and joins) and multiloops (bulges) are represented as a single underscore in this shape representation. Thus, the sole difference to type 2 shapes is that in this

representation, nested helices are combined and therefore, this shape type does additionaly abstract from nesting and adjacency of helices.

Type 5: (Most abstract)

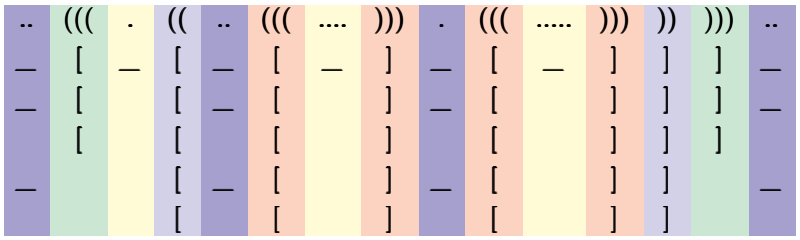
helix nesting pattern and no unpaired regions

In this shape abstraction, (interrupted) ladders are depicted by a pair of opening and closing squared brackets, since nested helices are combined again. This combination of nested helices is the sole difference to type 3 shape representations and the difference to type 4 shape representations is that all underscores representing unpaired regions (in external loops and multiloops) are omitted here (except in the case of the completely unpaired structure).

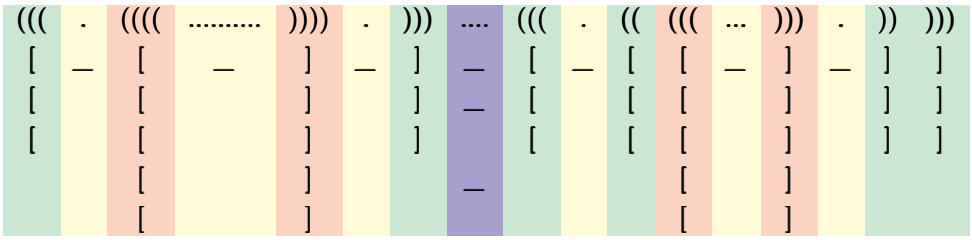
The differences between these five abstraction levels are illustrated in Example 2.2.

Example 2.2 *The following four examples show the differences between the five shape types resp. the five abstraction levels. In each example, the first line shows one of the secondary structures* secondary_i, $1 \leq i \leq 4$ *(in dot-bracket representation, as given in Example 2.1). The following lines show the resulting shapes, starting with the type 1 shape in the second line and ending with the type 5 shape for this secondary structure in the last line, respectively.*

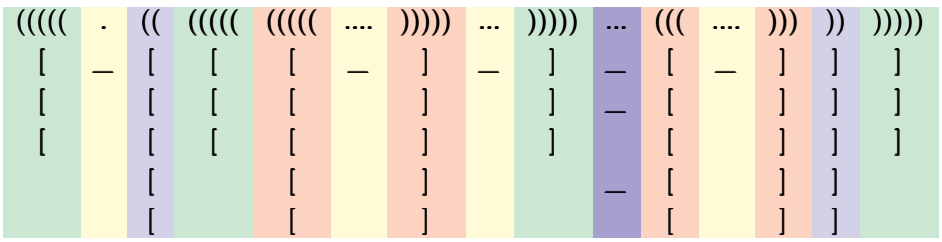
1. secondary_1*:*



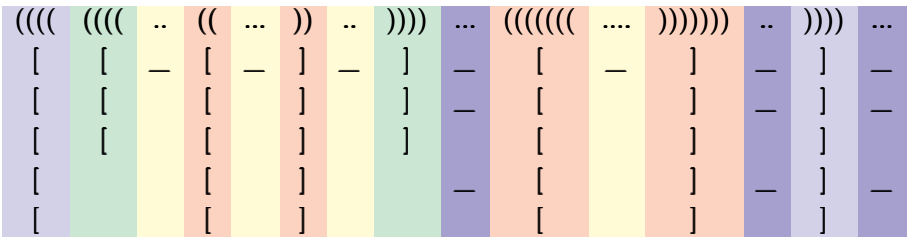
2. secondary_2*:*



3. secondary_3*:*



4. secondary_4*:*



2.2.2 Shape Languages

Now, we want to give formal definitions of the languages containing exactly all shapes of a certain type. At this point as well as for our further studies, we assume the reader has basic knowledge of the notions concerning context-free languages and grammars. There is plenty of literature that could be used to get an introduction, e.g. [HMU01] or [Har78]. Again, we start with the most accurte shape type 1 and will consider type 5 at last.

Hence, our first goal is to give a formal definiton of the language $\mathcal{L}_1$ containing exactly all type 1 shape representations of any possible RNA secondary structure. To reach this goal, we first observe that for this type, the shape representation of a totally unpaired secondary structure is given by a single underscore, so $\{\_\}$ must be a subset of the language $\mathcal{L}_1$. On the other hand, each secondary structure that is not totally unpaired represents an external loop containing at least one helical region. The first helical region in this external loop may be preceeded by a tail and equally, the last helical region in this external loop may be followed by a tail. Furthermore, there may be a join between two helical regions. This means that every secondary structure that is not completely unpaired and whose external loop contains $n \geq 1$ adjacent helices can be represented as a word

$$t_0({}^{a_1}u_1)^{a_1} \cdots t_{n-1}({}^{a_n}u_n)^{a_n}t_n,$$

where $a_i \geq 1$, $1 \leq i \leq n$, $t_j \in \{.\}^*$, $0 \leq j \leq n$, and each of the subwords $u_1, \ldots, u_n$ must contain at least one (hairpin) loop. As by definition, helical regions, tails and joins contribute to this shape representation, any such secondary structure is mapped to a type 1 shape $v_0[w_1] \cdots v_{n-1}[w_n]v_n$, where each of the words w_i is the homomorphic image of subword u_i of the secondary structure, $1 \leq i \leq n$, and every word $v_i \in \{\_, \epsilon\}$. Thus, let $\mathcal{L}_u = \{\_, \epsilon\}$ be the language of the two possible homomorphic images of unpaired regions and let $\mathcal{L}_{l_1}$ be the language containing exactly all homomorphic images of helices. Furthermore, let $\mathcal{L}_{l_1u} := \mathcal{L}_{l_1}\mathcal{L}_u$ be the concatenation of these two languages $\mathcal{L}_{l_1}$ and $\mathcal{L}_u$. Obviously, any type 1 shape of the form $v_0[w_1] \cdots v_{n-1}[w_n]v_n$, $n \geq 1$, is contained in the language $\mathcal{L}_u\mathcal{L}_{l_1u}^+$ and thus, every possible secondary structure is mapped to a type 1 shape in $\{\_\} \cup \mathcal{L}_u\mathcal{L}_{l_1u}^+$.
We now want to define the language $\mathcal{L}_{l_1}$ containing all type 1 shapes that are homomorphic images of helices. Therefore, we first observe that a helix may be a hairpin loop, which is represented by a word $({}^a.^+)^a$ in the secondary structure and mapped to the word $[\_]$. But a helix may also be decomposed into a ladder, one or two bulges interrupting this ladder and another helix, whose helical region is the second part of this interrupted ladder. Hence, a given secondary structure may contain some of the subwords $({}^a.^+({}^bu)^b)^a$, $({}^a({}^bu)^b.^+)^a$ and $({}^a.^+({}^bu)^b.^+)^a$, for some $a, b \geq 0$, where $({}^bu)^b$ is again a helix. As both ladders and bulges interrupting loops contribute to this shape representation, their homomorphic images are given by $[\_[w]]$, $[[w]\_]$ and $[\_[w]\_]$, respectively, where the subwords $[w]$ are again contained in the language $\mathcal{L}_{l_1}$. Finally, a helix may be a multiloop and thus, the language $\mathcal{L}_1$ can be defined as follows:

Definition 2.6 *The language $\mathcal{L}_1$ containing exactly all type 1 shapes is given by $\mathcal{L}_1 := \{\_\} \cup \mathcal{L}_u\mathcal{L}_{l_1u}^+$, where $\mathcal{L}_{l_1u} := \mathcal{L}_{l_1}\mathcal{L}_u$, $\mathcal{L}_u := \{\_, \epsilon\}$ and $\mathcal{L}_{l_1}$ is the smallest language satisfying the following conditions:*

1. $[\_] \in \mathcal{L}_{l_1}$.
2. *If $w \in \mathcal{L}_{l_1}$, then $[\_w], [w\_], [\_w\_] \in \mathcal{L}_{l_1}$.*
3. *If $w_1, \ldots, w_n \in \mathcal{L}_{l_1}$, $v_0, \ldots, v_n \in \mathcal{L}_u$ and $n \geq 2$, then $[v_0w_1v_1w_2 \ldots v_{n-1}w_nv_n] \in \mathcal{L}_{l_1}$.*

Alternatively, a formal definition of the language $\mathcal{L}_1$ could be given as follows:

Definition 2.7 *The language $\mathcal{L}_1$ containing exactly all type 1 shapes is given by $\mathcal{L}_1 := \{\_\} \cup \mathcal{L}_u\mathcal{L}_{l_1u}^+$, where $\mathcal{L}_{l_1u} := [\mathcal{L}_{l_1}]\mathcal{L}_u$, $\mathcal{L}_u := \{\_, \epsilon\}$ and $\mathcal{L}_{l_1}$ is the smallest language satisfying the following conditions:*

1. $\_ \in \mathcal{L}_{l_1}$.
2. *If $w \in \mathcal{L}_{l_1}$, then $\_[w], [w]\_, \_[w]\_ \in \mathcal{L}_{l_1}$.*
3. *If $w_1, \ldots, w_n \in \mathcal{L}_{l_1}$, $v_0, \ldots, v_n \in \mathcal{L}_u$ and $n \geq 2$, then $v_0[w_1]v_1[w_2] \ldots v_{n-1}[w_n]v_n \in \mathcal{L}_{l_1}$.*

We will use the second characterization, since it will be more useful for our further investigations.

As by definition, hairpin loops, single bulges interrupting ladders and internal loops do not contribute to shape representations of type 2, a characterization of the language $\mathcal{L}_2$ containing exactly all type 2 shapes can easily be obtained from that of the language $\mathcal{L}_1$. Thus, we immediately obtain:

Definition 2.8 *The language $\mathcal{L}_2$ containing exactly all type 2 shapes is given by $\mathcal{L}_2 := \{\_\} \cup \mathcal{L}_u\mathcal{L}_{l_2u}^+$, where $\mathcal{L}_{l_2u} := [\mathcal{L}_{l_2}]\mathcal{L}_u$, $\mathcal{L}_u := \{\_, \epsilon\}$ and $\mathcal{L}_{l_2}$ is the smallest language satisfying the following conditions:*

1. $\epsilon \in \mathcal{L}_{l_2}$.
2. *If $w \in \mathcal{L}_{l_2}$, then $[w] \in \mathcal{L}_{l_2}$.*
3. *If $w_1, \ldots, w_n \in \mathcal{L}_{l_2}$, $v_0, \ldots, v_n \in \mathcal{L}_u$ and $n \geq 2$, then $v_0[w_1]v_1[w_2]\ldots v_{n-1}[w_n]v_n \in \mathcal{L}_{l_2}$.*

The language $\mathcal{L}_3$ containing exactly all type 3 shapes can easily be characterized by taking into account that all single-stranded regions (except for the completely unpaired structure) are ignored in these shape representations. Hence, considering the definition of the language $\mathcal{L}_1$ resp. $\mathcal{L}_2$, we obtain the following language definition for type 3 shapes:

Definition 2.9 *The language $\mathcal{L}_3$ containing exactly all type 3 shapes is given by $\mathcal{L}_3 := \{\_\} \cup \mathcal{L}_{l_3u}^+$, where $\mathcal{L}_{l_3u} := [\mathcal{L}_{l_3}]$ and $\mathcal{L}_{l_3}$ is the smallest language satisfying the following conditions:*

1. $\epsilon \in \mathcal{L}_{l_3}$.
2. *If $w \in \mathcal{L}_{l_3}$, then $[w] \in \mathcal{L}_{l_3}$.*
3. *If $w_1, \ldots, w_n \in \mathcal{L}_{l_3}$ and $n \geq 2$, then $[w_1][w_2]\ldots[w_n] \in \mathcal{L}_{l_3}$.*

Now, we want to give a formal definiton of the language $\mathcal{L}_4$ containing exactly all type 4 shape representations. As the only difference to type 2 shapes is that nested helices are combined, it is obvious that a characterization of the language $\mathcal{L}_4$ is given as follows:

Definition 2.10 *The language $\mathcal{L}_4$ containing exactly all type 4 shapes is given by $\mathcal{L}_4 := \{\_\} \cup \mathcal{L}_u\mathcal{L}_{l_4u}^+$, where $\mathcal{L}_{l_4u} := [\mathcal{L}_{l_4}]\mathcal{L}_u$, $\mathcal{L}_u := \{\_, \epsilon\}$ and $\mathcal{L}_{l_4}$ is the smallest language satisfying the following conditions:*

1. $\epsilon \in \mathcal{L}_{l_4}$.
2. *If $w_1, \ldots, w_n \in \mathcal{L}_{l_4}$, $v_0, \ldots, v_n \in \mathcal{L}_u$ and $n \geq 2$, then $v_0[w_1]v_1[w_2]\ldots v_{n-1}[w_n]v_n \in \mathcal{L}_{l_4}$.*

Finally, the language $\mathcal{L}_5$ containing exactly all type 5 shapes can easily be characterized by modifying the defnition of the language $\mathcal{L}_4$, such that no underscores representing single-stranded regions (except in the case of the completely unpaired structure) are retained. This yields the following characterization:

Definition 2.11 *The language $\mathcal{L}_5$ containing exactly all type 5 shapes is given by $\mathcal{L}_5 := \{\_\} \cup \mathcal{L}_{l_5u}^+$, where $\mathcal{L}_{l_5u} := [\mathcal{L}_{l_5}]$ and $\mathcal{L}_{l_5}$ is the smallest language satisfying the following conditions:*

1. $\epsilon \in \mathcal{L}_{l_5}$.
2. *If $w_1, \ldots, w_n \in \mathcal{L}_{l_5}$ and $n \geq 2$, then $[w_1][w_2]\ldots[w_n] \in \mathcal{L}_{l_5}$.*

2.2.3 Shape Grammars

The next goal is to find five unambiguous[1] context-free grammars $\mathcal{G}_i$ with $\mathcal{L}(\mathcal{G}_i) = \mathcal{L}_i$, $1 \leq i \leq 5$. This means we want to construct the five grammars $\mathcal{G}_i$ such that for each $i \in \{1, \ldots, 5\}$ the grammar $\mathcal{G}_i$ unambiguously generates exactly the language $\mathcal{L}_i$.

First, we want to construct an unambiguous context-free grammar $\mathcal{G}_1$ with start symbol S_1 producing exactly all type 1 shape representations. To reach this goal, we first observe that for this type, the shape representation of a totally unpaired secondary structure is given by a single underscore, so the first production rules might be $S_1 \to A$ and $S_1 \to \_$, where A is the start symbol for all type 1 shapes representing a folded secondary structure. Thus, concerning Definition 2.7, we observe that the language that can be generated by starting with nonterminal symbol A must be equal to

$$\begin{aligned}\mathcal{L}_1 \setminus \{\_\} =& \mathcal{L}_u\mathcal{L}_{l_1u}^+ \\ =& \mathcal{L}_u[\mathcal{L}_{l_1}]\mathcal{L}_u\mathcal{L}_{l_1u}^* \\ =& \mathcal{L}_u[\mathcal{L}_{l_1}]\mathcal{L}_u(\{\epsilon\} \cup \mathcal{L}_{l_1u}^+)\end{aligned}$$

[1] Unambiguity is necessary, as we later want to use these five grammars to construct generating functions counting the number of type i shapes, $1 \leq i \leq 5$. If there are more than one leftmost derivations for a type i shape sh, $1 \leq i \leq 5$, then this shape sh is counted more than once in the corresponding generating function.

$$
\begin{aligned}
&= \mathcal{L}_u[\mathcal{L}_{l_1}](\mathcal{L}_u \cup \mathcal{L}_u \mathcal{L}_{l_1 u}^{+}) \\
&= \mathcal{L}_u[\mathcal{L}_{l_1}](\mathcal{L}_u \cup \mathcal{L}_1 \setminus \{\_\}) \\
&= \{\epsilon, \_\}[\mathcal{L}_{l_1}](\{\epsilon, \_\} \cup \mathcal{L}_1 \setminus \{\_\}).
\end{aligned}
$$

Therefore, we use the production rules $A \to C[B]D$, $C \to \epsilon$, $C \to \_$, as well as $D \to \epsilon$, $D \to \_$ and $D \to A$. Obviously, the language that is generated by starting with the nonterminal symbol B on the right hand side of the production rule $A \to C[B]D$ must be equal to the language $\mathcal{L}_{l_1}$. Thus, the expression $[B]$ may generate a hairpin, a bulge interrupting a ladder, an internal loop interrupting a ladder or a ladder whose last pair is the foundation of a multiloop. Concerning Definition 2.7 again, we immediately observe that we have to use the production rules $B \to \_$ (hairpin generating rule), $B \to \_[B]$ (generates a bulge interrupting a ladder on the left), $B \to [B]\_$ (generates a bulge interrupting a ladder on the right), $B \to \_[B]\_$ (interior loop generating rule) and $B \to C[B]A$ (multiloop generating rule). Combining all these production rules, we obtain:

Lemma 2.2 *A context-free grammar $\mathcal{G}_1$ unambiguously generating exactly the language $\mathcal{L}_1$ is given by $\mathcal{G}_1 = (I, \Sigma, R, S_1)$, where $I = \{S_1, A, B, C, D\}$, $\Sigma = \{\_, [,]\}$ and R contains exactly the following rules:*

$$
\begin{array}{lllll}
S_1 \to A, & S_1 \to \_, & A \to C[B]D, & B \to \_, & B \to C[B]A, \\
B \to \_[B], & B \to [B]\_, & B \to \_[B]\_, & C \to \epsilon, & C \to \_, \\
D \to \epsilon, & D \to \_, & D \to A. & &
\end{array}
$$

Shape representations of type 2 can be obtained from shape representations of type 1 by removing all underscores representing hairpin loops as well as all bulges (and internal loops) interrupting ladders. Hence, an unambiguous context-free grammar $\mathcal{G}_2$ creating exactly all type 2 shapes can be obtained from the grammar $\mathcal{G}_1$ for type 1 shapes by removing underscores in the corresponding rules.
First, we observe that hairpins are now represented by $[\,]$ and therefore, the hairpin loop generating rule $B \to \_$ of $\mathcal{G}_1$ has to be changed into $B \to \epsilon$. Furthermore, since bulges and internal loops interrupting ladders have to be removed, the rules $B \to \_[B]$ and $B \to [B]\_$ of $\mathcal{G}_1$ generating such a bulge, and the internal loop generating rule $B \to \_[B]\_$ must all be replaced by the rule $B \to [B]$. As all other rules of $\mathcal{G}_1$ can be maintained, the grammar $\mathcal{G}_2$ is given by:

Lemma 2.3 *A context-free grammar $\mathcal{G}_2$ unambiguously generating exactly the language $\mathcal{L}_2$ is given by $\mathcal{G}_2 = (I, \Sigma, R, S_2)$, where $I = \{S_2, A, B, C, D\}$, $\Sigma = \{\_, [,]\}$ and R contains exactly the following rules:*

$$
\begin{array}{llll}
S_2 \to A, & S_2 \to \_, & A \to C[B]D, & B \to \epsilon, \\
B \to C[B]A, & B \to [B], & C \to \epsilon, & C \to \_, \\
D \to \epsilon, & D \to \_, & D \to A. &
\end{array}
$$

Now, we consider type 3 shapes. By definition, in type 3 shapes, all underscores representing single-stranded regions are omitted and therefore, shape representations of this type are equal to those of type 1 after the elimination of all underscores. Hence, an unambiguous context-free grammar $\mathcal{G}_3$ producing exactly all type 3 shapes can be obtained from the context-free grammar $\mathcal{G}_1$ by removing all underscores (except for the underscore representing a completely unpaired structure) in every rule and then eliminating redundant rules. That way, we obtain the following lemma:

Lemma 2.4 *A context-free grammar $\mathcal{G}_3$ unambiguously generating exactly the language $\mathcal{L}_3$ is given by $\mathcal{G}_3 = (I, \Sigma, R, S_3)$, where $I = \{S_3, A, B, D\}$, $\Sigma = \{[,], \_\}$ and R contains exactly the following rules:*

$$
\begin{array}{llll}
S_3 \to A, & S_3 \to \_, & A \to [B]D, & B \to \epsilon, \\
B \to [B]A, & B \to [B], & D \to \epsilon, & D \to A.
\end{array}
$$

To find an unambiguous context-free grammar $\mathcal{G}_4$ creating exactly all shape representations of type 4, we can modify the context-free grammar $\mathcal{G}_2$ for type 2 shape representations, as the only difference between type 2 and type 4 shapes is that in the latter nested helices are combined and therefore, we can choose $\mathcal{G}_4 = \mathcal{G}_2$ and then eliminate the production rule $B \to [B]$ generating nested helices. That way, we obtain:

Lemma 2.5 *A context-free grammar $\mathcal{G}_4$ unambiguously generating exactly the language $\mathcal{L}_4$ is given by $\mathcal{G}_4 = (I, \Sigma, R, S_4)$, where $I = \{S_4, A, B, C, D\}$, $\Sigma = \{\_, [,]\}$ and R contains exactly the following rules:*

$$
\begin{array}{llllll}
S_4 \to A, & S_4 \to \_, & A \to C[B]D, & B \to \epsilon, & B \to C[B]A, \\
C \to \epsilon, & C \to \_, & D \to \epsilon, & D \to \_, & D \to A.
\end{array}
$$

Considering type 5 shapes, we observe that any shape representation of this type is equal to a type 4 shape without underscores, since in contrast to type 4 shapes where only those underscores are ignored that represent hairpin loops and ladder interrupting bulges and internal loops, in type 5 shapes also underscores representing single-stranded regions in external loops and multiloops are omitted, and therefore no underscores are retained. For this reason, we can construct an unambiguous context-free grammar $\mathcal{G}_5$ producing exactly all type 5 shapes by eliminating all underscores (except for the underscore representing a completely unpaired structure) in every rule of $\mathcal{G}_4$ and then removing redundant rules.
We could alternatively use the context-free grammar $\mathcal{G}_3$ for type 3 shapes to obtain this context-free grammar $\mathcal{G}_5$, as every type 3 shape representation can be transformed into a type 5 shape by eliminating all but one pairs of opening and closing squared brackets representing nested helices. In this case, we must thus only remove the production rule $B \to [B]$ from $\mathcal{G}_3$ to obtain the grammar $\mathcal{G}_5$.
Both alternatives lead to the following lemma:

Lemma 2.6 *A context-free grammar $\mathcal{G}_5$ unambiguously generating exactly the language $\mathcal{L}_5$ is given by $\mathcal{G}_5 = (I, \Sigma, R, S_5)$, where $I = \{S_5, A, B, D\}$, $\Sigma = \{[,], \_\}$ and R contains exactly the following rules:*

$$\begin{array}{llll} S_5 \to A, & S_5 \to \_, & A \to [B]D, & B \to \epsilon, \\ B \to [B]A, & D \to \epsilon, & D \to A. & \end{array}$$

Now, after having constructed the unambiguous grammars $\mathcal{G}_i$, $1 \leq i \leq 5$, the differences of the homomorphisms mapping secondary structures to type i shapes, $1 \leq i \leq 5$, should be clear and we will finally start our analysis of abstract shapes for which we will use these five grammars $\mathcal{G}_i$, $1 \leq i \leq 5$.

3 Number of Shapes

First, we want to derive a simple combinatorial result for shapes, namely the number of type i shapes, for each $1 \in \{1, \ldots, 5\}$. In fact, we aim at determining asymptotical representations of the number of type i shapes of length n, $1 \leq i \leq 5$.
To obtain the desired results, we will use the methods of *generating functions*. In particular, we first want to compute closed forms of the ordinary generating functions $S_i(z)$, $1 \leq i \leq 5$, counting the number s_{i_n} of type i shapes of length n and then apply Darboux's theorem [KW89] to these closed forms to obtain the desired asymptotics for $s_{i_n} = [z^n]S_i(z)$, $1 \leq i \leq 5$[2].
Note that in this article, we will not recall the fundamental definitions and methods concerning generating functions. An introduction to generating functions and some of their uses in discrete mathematics can be found for example in [FS07, Wil94]. Several pretty examples for generating functions can be found in [Com74]. Furthermore, for an introduction to some advanced methods that have to be used for more difficult problems, see for example [GK90].

Now, for every $i \in \{1, \ldots 5\}$, let $\mathcal{S}_i$ be the combinatorial class of all type i shapes and let $s_{i_n} = \mathrm{card}(\mathcal{S}_{i_n})$ be the number of elements in $\mathcal{S}_i$ of length n, $1 \leq i \leq 5$. Our first goal is to find closed forms for the generating functions

$$S_i(z) = \sum_{n \geq 0} s_{i_n} z^n = \sum_{s \in \mathcal{S}_i} z^{|s|},$$

$1 \leq i \leq 5$. As the size of an element $s \in \mathcal{S}_i$, $1 \leq i \leq 5$, is equal to its length, each terminal symbol $t \in \{[,], \_\}$ must be represented by a factor $z^{|t|} = z^1 = z$ and ϵ must be represented by a factor $z^{|\epsilon|} = z^0 = 1$ in generating functions.
We already know that for every $i \in \{1, \ldots 5\}$, the context-free grammar $\mathcal{G}_i$ is unambiguous and generates exactly all type i shape representations and thus all elements in the combinatorial class $\mathcal{S}_i$. Hence, for every $i \in \{1, \ldots 5\}$, we can translate the set of productions of the grammar $\mathcal{G}_i$ into a system of equations as proposed by Chomsky and Schützenberger [CS63]. For every $i \in \{1, \ldots 5\}$, the resulting system can then be solved for the variable S_i corresponding to the start symbol (axiom) of the underlying grammar $\mathcal{G}_i$ to obtain the desired closed form of the ordinary generating function $S_i(z)$.
Considering the grammars $\mathcal{G}_i$, $1 \leq i \leq 5$, the resulting systems of equations are given as follows:

- generating function $S_1(z)$:

$$\begin{aligned} S_1(z) &= A(z) + z, \\ A(z) &= C(z) \cdot z \cdot B(z) \cdot z \cdot D(z), \end{aligned}$$

[2] In this paper we use $[z^n]S(z)$ to denote the coefficient at z^n in the expansion of $S(z)$ around $z = 0$.

$$\begin{aligned} B(z) &= z + C(z)\cdot z\cdot B(z)\cdot z\cdot A(z) + z\cdot z\cdot B(z)\cdot z + z\cdot B(z)\cdot z\cdot z + z\cdot z\cdot B(z)\cdot z\cdot z, \\ C(z) &= 1 + z, \\ D(z) &= 1 + z + A(z). \end{aligned} \tag{1}$$

- generating function $S_2(z)$:

$$\begin{aligned} S_2(z) &= A(z) + z, \\ A(z) &= C(z)\cdot z\cdot B(z)\cdot z\cdot D(z), \\ B(z) &= 1 + C(z)\cdot z\cdot B(z)\cdot z\cdot A(z) + z\cdot B(z)\cdot z, \\ C(z) &= 1 + z, \\ D(z) &= 1 + z + A(z). \end{aligned} \tag{2}$$

- generating function $S_3(z)$:

$$\begin{aligned} S_3(z) &= A(z) + z, \\ A(z) &= z\cdot B(z)\cdot z\cdot D(z), \\ B(z) &= 1 + z\cdot B(z)\cdot z\cdot A(z) + z\cdot B(z)\cdot z, \\ D(z) &= 1 + A(z). \end{aligned} \tag{3}$$

- generating function $S_4(z)$:

$$\begin{aligned} S_4(z) &= A(z) + z, \\ A(z) &= C(z)\cdot z\cdot B(z)\cdot z\cdot D(z), \\ B(z) &= 1 + C(z)\cdot z\cdot B(z)\cdot z\cdot A(z), \\ C(z) &= 1 + z, \\ D(z) &= 1 + z + A(z). \end{aligned}$$

- generating function $S_5(z)$:

$$\begin{aligned} S_5(z) &= A(z) + z, \\ A(z) &= z\cdot B(z)\cdot z\cdot D(z), \\ B(z) &= 1 + z\cdot B(z)\cdot z\cdot A(z), \\ D(z) &= 1 + A(z). \end{aligned}$$

After solving these systems for $S_i(z)$, $1 \le i \le 5$, we can use Darboux's theorem to determine asymptotics for the nth coefficients ($n \to \infty$) of the five ordinary generating functions $S_i(z)$, $1 \le i \le 5$. By choosing $m = 0$ for the application of Darboux's theorem and afterwards computing series expansions of the resulting asymptotics about $n \to \infty$, we obtain the following results:

- $s_{1_n} \sim 2.09188^n \cdot 0.783027 \cdot n^{-3/2}$,
- $s_{2_n} \sim 2.40518^n \cdot 0.94913 \cdot n^{-3/2}$,
- $s_{3_n} \sim ((-2)^n + 2^n) \cdot \sqrt{\frac{2}{\pi}} \left(\frac{1}{n}\right)^{3/2} \approx ((-2.)^n + 2.^n) \cdot 0.797885 \cdot n^{-3/2}$,
- $s_{4_n} \sim 2.22293^n \cdot 0.88897 \cdot n^{-3/2}$,
- $s_{5_n} \sim 3^{n/2}\,(1 + (-1)^n) \cdot \sqrt{\frac{3}{2\pi}} \left(\frac{1}{n}\right)^{3/2} \approx 3.^{0.5n}\,(1. + (-1.)^n) \cdot 0.690988 \cdot n^{-3/2}$,

as $n \to \infty$.
Note that the asymptotical representation of the number of type 5 shapes of length n, i.e. the asymptotic for s_{5_n}, has already been determined in [LPC08][3]. It is only presented here for the sake of completeness.

[3] It should be mentioned that in [LPC08], there is also given an asymptotic for the number of type 1 shapes of length n, i.e. for s_{1_n}. However, Lorenz et al. made a little mistake constructing the corresponding context-free grammar, such that their generating function does not count the correct number of type 1 shapes.

4 Reduction of the Search Space

We now want to focus on the main goal of this article, namely on quantifying the reduction of the search space when using the concept of abstract shapes. As a first approach, we want to compare the number of secondary structures of length n to the number of different type i shapes that are homomorphic images of those secondary structures, for every $i \in \{1, \ldots, 5\}$. More precisely, for every $i \in \{1, \ldots, 5\}$, we want to compute the number of different type i shapes sh for which there exists a secondary structure s of length n such that the morphism mapping secondary structures to type i shapes maps this secondary structure s to the shape sh.
Formally, let $\mathcal{S}_s$ be the combinatorial class of all secondary structures and let $h_i : \mathcal{S}_s \to \mathcal{S}_i$ be the morphism mapping a secondary structure s to its type i shape sh, $1 \leq i \leq 5$. Then, for every $i \in \{1, \ldots, 5\}$, let

$$\mathcal{M}_i := \{sh \mid sh \in \mathcal{S}_i \wedge \exists s \in \mathcal{S}_s \ [|s| \geq 1 \wedge h_i(s) = sh]\}$$

be the combinatorial class of all different type i shapes that are homomorphic images of secondary structures of any length, $1 \leq i \leq 5$. Thus, our goal is to compute an asymptotical representation of

$$m_{i_n} := \mathrm{card}(\mathcal{M}_{i_n}),$$

where

$$\mathcal{M}_{i_n} := \{sh \mid sh \in \mathcal{S}_i \wedge \exists s \in \mathcal{S}_s \ [|s| = n \wedge h_i(s) = sh]\}.$$

To reach this goal, we will make use of the fact that the number of different type i shapes that are homomorphic images of secondary structures of length n is equal to the number of different type i shapes that are homomorphic images of secondary structures of length *at most* n, for each $i \in \{1, \ldots, 5\}$. In fact, for each length $n \geq 1$,

$$\mathcal{M}_{i_n} = \{sh \mid sh \in \mathcal{S}_i \wedge \exists s \in \mathcal{S}_s \ [|s| \leq n \wedge h_i(s) = sh]\},$$

$1 \leq i \leq 5$, which results from the observation that every secondary structure can be prolongated to an arbitrary size without changeing its image under h_i, $1 \leq i \leq 5$, e.g. by inserting a run of dots. For details, see [LPC08] and the discussion below.
Using this observation, we now want to construct each of the five generating functions $M_i(z)$, $1 \leq i \leq 5$, separately.

Type 1:

First, we consider the most accurate shape type, i.e. type 1 shape representations. Thus, we aim at constructing the ordinary generating function

$$M_1(z) = \sum_{n \geq 0} m_{1_n} z^n$$

of the couting sequence $(m_{1_n})_{n \geq 0}$. This means that variable z marks length in this ordinary generating function $M_1(z)$, and the length that is marked there is a maximum length of secondary structures $s \in \mathcal{S}_s$. But the coefficient at z^n in this generating function $M_1(z)$ does not count the number of secondary structures with maximum length n, it counts the number of different type 1 shapes that result from these secondary structures with maximum length n.
We now want to construct a system of equations which can be solved for $M_1(z)$ to get a closed form of the generating function $M_1(z)$ by modifying system (1) which we constructed for the ordinary generating funtion $S_1(z)$.
Obviously, any secondary structure s is mapped by the morphism h_1 to a type 1 shape sh with $|sh| = j \leq |s|$. This means that this shape sh would make a contribution of $z^{|sh|} = z^j$ to the generating function $S_1(z) = \sum_{sh' \in \mathcal{S}_1} z^{|sh'|}$ when not "modifying" its size. But we want this shape sh to make a contribution of z^{j+p} for each $p \geq 0$ to the generating function $M_1(z)$. For this reason, we decided to "modify" its size by changing the sum on the right hand side of serveral equations in system (1) by multiplying some factors z to certain summands.
To find out which summands must be changed and why we decided to "modify" the sizes of shapes, we consider the homomorphic image of the secondary structure $\mathrm{secondary}_2$ given in Example 2.1 under the mapping h_1, the type 1 shape $sh = \texttt{[\_[\_]\_]\_[\_[[\_]\_]]}$. As we already know, this shape sh is obtained from any secondary structure s that has the form

$$(^a \ .^+ \ (^b \ .^+ \ )^b \ .^+ \ )^a \ .^+ \ (^c \ .^+ \ (^d \ (^e \ .^+ \ )^e \ .^+ \ )^d \ )^c, \tag{4}$$

$a, b, c, d, e \geq 1$ by using the morphism h_1. Hence, for a large value of n, there are plenty of different secondary structures s with length at most n having this form and all of these secondary structures are mapped by the morphism h_1 to the same shape sh. Therefore, we consider only the secondary structure s with minimum length among all these secondary structures that match the form (4). Obviously, $s = \mathbf{(.(.).).(.((.).))}$, $|s| = |sh| = 17$ and as we can see, this mimimum secondary structure s can easily be obtained from the shape sh by substituting each underscore with a dot, each opening squared bracket **[** with and opening bracket **(** and each closing squared bracket **]** with a closing bracket **)**. Thus, this minimum secondary structure s and the shape sh make the same contribution z^{17} to a generating function, in which z marks length and therefore, by solving the system (1) for $S_1(z)$, we do not only get a closed form for the ordinary generating function $S_1(z)$ counting the number of type 1 shapes with length n, but this solution is also a closed form for an ordinary generating function that counts the number of such minimum secondary structures where all loop and ladder lengths are equal to 1 with length n.
Hence, as sh is obtained from any secondary structure s that has the form (4) by using the morphism h_1, the shape sh must make a contribution of z^k for each $k \geq 17 = |sh|$ to the generating function $M_1(z)$. In fact, if the underlying grammar $\mathcal{G}_1$ generates sh, then we must add a term

$$\sum_{i \geq 0} z^{|sh|+i} = \left(\sum_{i \geq 0} z^i \right) \cdot z^{|sh|} = \frac{1}{1-z} \cdot z^{|sh|}$$

to the generating function $M_1(z)$.
Due to these observations, it is obvious that the ordinary generating function $S_1(z)$ counting the number of type 1 shapes has to be multiplied by the factor

$$\sum_{i \geq 0} z^i = \frac{1}{1-z}$$

to obain the desired generating function in $M_1(z)$ and therefore,

$$M_1(z) = \frac{1}{1-z} \cdot S_1(z).$$

Equivalently, by multiplying the right hand of the first equation of system (1) with the factor $\frac{1}{1-z}$, we obtain the system

$$\begin{aligned}
M_1(z) &= \frac{1}{1-z} \cdot (A(z) + z), \\
A(z) &= C(z) \cdot z^2 \cdot B(z) \cdot D(z), \\
B(z) &= z + C(z) \cdot z^2 \cdot B(z) \cdot A(z) + z^2 \cdot B(z) \cdot (2 \cdot z + z^2), \\
C(z) &= 1 + z, \\
D(z) &= 1 + z + A(z),
\end{aligned}$$

which can be solved for $M_1(z)$ to get the desired closed form of the generating function $M_1(z)$.

Type 2:

To construct the ordinary generating function $M_2(z)$ of counting sequence $(m_{2_n})_{n \geq 0}$, we consider the system of equations (2) for the generating function $S_2(z)$. As before, we want to change the sum on the right hand side of serveral equations in this system of equations by multiplying some factors z to certain summands.
By definition, hairpin loops, bulges interrupting ladders and internal loops are omitted in this shape representation. Thus, considering the type 2 shape $sh = \mathbf{[[\ ]]\_[[[\ ]]]}$ for the secondary structure secondary$_2$ given in Example 2.1, we can conclude that there must be at least five single-stranded regions that are not represented as an underscore in this shape sh. In fact, in the case of the secondary structure secondary$_2$, there are six single-stranded regions omitted in this shape sh, one of them in each of the two hairpin loops, two in the internal loop and one in each of the two bulges interrupting a ladder. Consequently, for a secondary structure s with $h_2(s) = sh$ that has mimimum length among all secondary structures s' with $h_2(s') = sh$, its length $|s|$ must be at least $|sh| + 5$.
This means that we have to multiply a factor z representing a single-stranded region of length 1 to those summands on the right hand sides of system (2) corresponding to the rules of the underlying grammar $\mathcal{G}_2$

that generate type 2 shape abstractions of hairpin loops, ladder interrupting bulges and internal loops, respectively. Finally, we have to multiply the right hand side of the first equation of system (2) by the factor $\frac{1}{1-z}$, as we did before for the construction of the generating function $M_1(z)$. The resulting system of equations is given by

$$\begin{aligned} M_2(z) &= \frac{1}{1-z} \cdot (A(z) + z), \\ A(z) &= C(z) \cdot z^2 \cdot B(z) \cdot D(z), \\ B(z) &= z \cdot 1 + C(z) \cdot z^2 \cdot B(z) \cdot A(z) + z \cdot (z^2 \cdot B(z)), \\ C(z) &= 1 + z, \\ D(z) &= 1 + z + A(z). \end{aligned}$$

It has to be solved for $M_2(z)$ to obtain a closed form of the desired generating function $M_2(z)$.

Type 3:

To obtain the corresponding generating function $M_3(z)$ for type 3 shape representations, we have to change some summands of the sum on the right hand side of serveral equations of system (3).
By definition, single-stranded regions do not contribute to type 3 shape representations and thus, all underscores representing single-stranded regions are omitted in type 3 shapes. Considering the homomorphic image of the secondary structure secondary_2 given in Example 2.1 under the mapping h_3, the type 3 shape $sh = \texttt{[[ ]] [[[ ]]]}$, we can observe that there are exactly seven single-stranded regions omitted, one in each of the two hairpin loops, two in the internal loop, one in each of the two ladder interrupting bulges and one in the external loop, connecting the two adjacent structures (join). As we can see by the definition of secondary structures, hairpin loops, bulges that interrupt ladders and internal loops are obligatory, whereas in multiloops and external loops, there must not exist any single-stranded regions. Consequently, for a (paired) secondary structure s with $h_3(s) = sh$ that has mimimum length among all secondary structures s' with $h_3(s') = sh$, its length $|s|$ must be at least $|sh| + hl + bl + il$, where hl, bl, il are the numbers of hairpin loops, ladder interrupting bulgs and internal loops that are not represented in this shape sh, respectively.
This means that we have to multiply a factor z representing a unpaired region of length 1 to those summands on the right hand sides of system (3) that generated (type 3 shape abstractions of) hairpin loops, bulges interrupting ladders and internal loops in the underlying grammar $\mathcal{G}_3$. Finally, we have to multiply the right hand side of the first equation of system (3) by the factor $\frac{1}{1-z}$, as we did before for the construction of the generating functions $M_1(z)$ and $M_2(z)$. The resulting system of equations is then given by

$$\begin{aligned} M_3(z) &= \frac{1}{1-z} \cdot (A(z) + z), \\ A(z) &= z^2 \cdot B(z) \cdot D(z), \\ B(z) &= z \cdot 1 + z^2 \cdot B(z) \cdot A(z) + z \cdot (z^2 \cdot B(z)), \\ D(z) &= 1 + A(z). \end{aligned}$$

The desired closed form of the ordinary generating function $M_3(z)$ can now be obtained by solving this system of equations for $M_3(z)$.

Types 4 and 5:

In the same way, we can determine closed forms of the generating functions $M_4(z)$ and $M_5(z)$. We obtain the following systems:

- generating function $M_4(z)$:

$$\begin{aligned} M_4(z) &= \frac{1}{1-z} \cdot (A(z) + z), \\ A(z) &= C(z) \cdot z \cdot B(z) \cdot z \cdot D(z), \\ B(z) &= z \cdot 1 + C(z) \cdot z \cdot B(z) \cdot z \cdot A(z), \\ C(z) &= 1 + z, \\ D(z) &= 1 + z + A(z). \end{aligned}$$

- generating function $M_5(z)$:

$$\begin{aligned} M_5(z) &= \frac{1}{1-z} \cdot (A(z) + z), \\ A(z) &= z \cdot B(z) \cdot z \cdot D(z), \\ B(z) &= z \cdot 1 + z \cdot B(z) \cdot z \cdot A(z), \\ D(z) &= 1 + A(z). \end{aligned}$$

Applying Darboux's theorem (with the choice $m = 0$) to these five generating functions $M_i(z)$, $1 \leq i \leq 5$, and afterwards computing series expansions of the resulting asymptotics about $n \to \infty$, we obtain:

- $m_{1_n} \sim 2.09188^n \cdot 1.50017 \cdot n^{-3/2}$,
- $m_{2_n} \sim 1.97491^n \cdot 1.50531 \cdot n^{-3/2}$,
- $m_{3_n} \sim 1.66034^n \cdot 1.71055 \cdot n^{-3/2}$,
- $m_{4_n} \sim 1.8879^n \cdot 1.52786 \cdot n^{-3/2}$,
- $m_{5_n} \sim 1.51243^n \cdot 1.84657 \cdot n^{-3/2}$,

as $n \to \infty$.

An asymptotical representation of the number $s_{0_n} := \text{card}(\mathcal{S}_{s_n})$ of secondary structures of length n has already been determined in [SW78]. It is given by

$$s_{0_n} \sim \sqrt{\frac{15 + 7\sqrt{5}}{8\pi}} n^{-3/2} \left(\frac{3 + \sqrt{5}}{2} \right)^n \approx 2.61803^n \cdot 1.10437 \cdot n^{-3/2}.$$

It is easy to observe that all these asymptotics grow exponentially in the variable n, where the base of the exponential expression in the asymptotical representation of the number of secondary structures of length n, i.e. of s_{0_n}, is significantly greater than the base of the exponential expression in the asymptotical representations of m_{i_n}, $1 \leq i \leq 5$.

We now want to compare the number of secondary structures of length n to the number of different type i shapes that are homomorphic images of those secondary structures, for every $i \in \{1, \ldots, 5\}$, by considering a plot containing all the resulting asymptotics for s_{0_n} and m_{i_n}, $1 \leq i \leq 5$. As all these asymptotics grow exponentially in n, it is appropriate to plot them using a logarithmic scale. The resulting logarithmic plot is shown in Figure 1.
It can easily be seen that the derived results are conform to the definition of type 1 shapes as the most accurate and of type 5 shapes as the most abstract shape type. However, there is not the expected order of plots. In fact, we can observe that type 3 shapes are the second most abstract shape type, and not as expected type 4 shape representations.
Moreover, the consideration of Figure 1 leads to the conclusion that abstracting from loop and stack lengths (mapping secondary structures to type 1 shapes) is not only the first, but also the biggest step for reducing the search space. Furthermore, abstracting from single-stranded regions that are not contained in multiloops and external loops (difference from type 1 to type 2 shape abstractions) yields only a comparatively small additional reduction. However, additionaly combining nested helices (difference from type 2 to type 4 shape abstractions) yields an even smaller reduction of the search space. But we make the possibly largest step for reducing the search space by not abstracting from combined helices but abstracting from single-stranded regions in multiloops and external loops (difference from type 2 to type 3 shape abstractions). Also a comparatively large final reduction of the search space is reached by abstracting from nesting of helices and combining nested helices (difference from type 3 to type 5 shape abstractions).

5 Taking Primary Structure into Account

In the last section, we have considered all secondary structures of length n, i.e. all possible two-dimensional foldings of a random primary structure s of length n, under the assumption that any base can basepair with any other base.

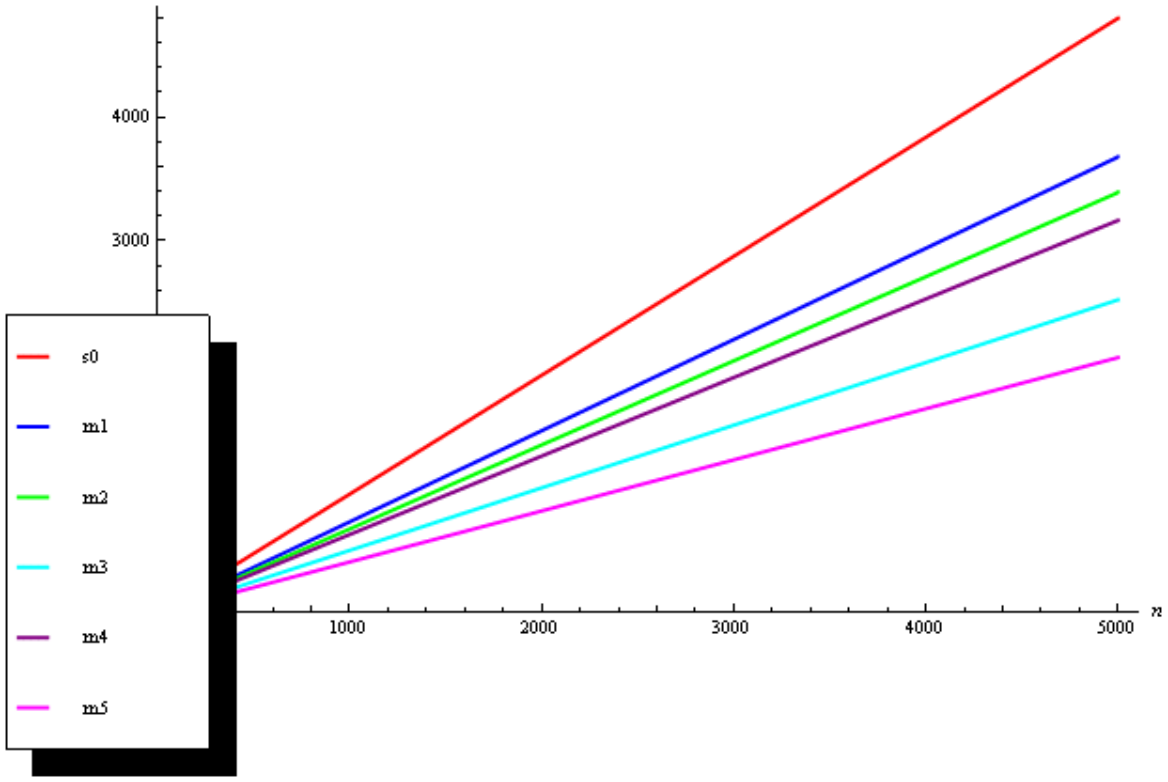


Figure 1: Number of secondary structures of length n and number of different type i shapes that are homomorphic images of secondary structures of length n, for all $i \in \{1, \ldots, 5\}$, logarithmically scaled.

This means we have taken a complete combinatorial point of view and counted the number of possible foldings unter the assumption of the combinatorial model for RNA secondary structures which has been considered by many authors (see for example [SW78, VC85, Neb02]). In the combinatorial model for RNA secondary structures, a uniform distribution of those structures is assumed, which means that all secondary structures are equiprobable. In fact, the combinatorial model considers only the topolgy of the planar secondary structure and completely abstracts from the possible primary structures of which a secondary structure could have been formed.
But as we already know, hydrogen bonding can occur only between the bases A and U or between G and C, with a weaker bond possible between the two bases G and U. Thus, by considering all secondary structures of length n for a random primary structure s of length n, where any two bases are allowed to pair, we consider an exponential number of biologically impossible foldings. For this reason, given an RNA primary structure s of length n, it seems appropriate to consider only those secondary structures of length n that are compatible with s, i.e. that contain only stable base pairs, when searching for the correct folding.
Similarly, using the abstract shapes approach to search for the correct folding of an RNA primary structure s of length n, it seems appropriate to consider only those shapes that are compatible with s, i.e. that have a preimage (secondary structure) such that this preimage contains only stable base pairs (is compatible with s).
For these reasons, Giegerich and co-workers introduced the terms (concrete) folding space and (abstract) shape space. According to [GVR04], they can be defined as follows:

Definition 5.1 *For a given RNA sequence (primary structure) s, its (concrete) folding space $F(s)$ is the set of all legal secondary structures according to the rules of base pairing. For each $i \in \{1, \ldots, 5\}$, its (abstract) shape space is $P_i(s) = \{h_i(x) \mid x \in F(s)\}$, where $h_i : \mathcal{S}_s \to \mathcal{S}_i$ is the morphism mapping secondary structures to type i shapes, $1 \leq i \leq 5$.*

As for a given RNA primary structure s of length n, the number of suboptimal minimum free energy secondary structures grows exponentially in n and the number of corresponding shapes is significantly smaller, Giegerich et al. [GVR04] derived some results on the growth behaviour of the folding space $F(s)$ and the shape space $P_5(s)$ for type 5 shapes. In fact, they assumed that the size of the folding space $F(s)$

and the size of the shape space $P_5(s)$ for a random primary structure s of length n can be represented as $\text{card}(F(s)) = c_F \cdot a^n$ and $\text{card}(P_5(s)) = c_{P_5} \cdot b^n$, respectively, and estimated the base of the exponential expression relating the number of secondary structures (without isolated base pairs) and type 5 shapes, to the length n of the primary structure s. Therefore, they computed the sizes of the folding spaces $F(s)$ and of the shape spaces $P_5(s)$ for random primary structures s of various lengths and obtained the desired estimates for $\text{card}(F(s))$ and $\text{card}(P_5(s))$ by approximating the parameters c_F, c_{P_5}, a and b.
Estimates for the exponential growths of the shape space sizes $\text{card}(P_i(s))$ for a random primary structure s of length n, $i \in \{1, \ldots, 5\}$, are given in Table 5 of [VGR06].
As all these results on the growth behaviour of the folding space and the five different shape spaces have been determined by heuristic approximations, it is a further task to analyze their sizes analytically. In fact, it would be interesting to compute asymptotics for the size of the folding shape $F(s)$ and the shape spaces $P_i(s)$, $i \in \{1, \ldots, 5\}$, for a random primary structure of length n and afterwards compare the derived results to the experimentally obtained values given in [GVR04] and [VGR06].

For this reason, we now want to compute an asymptotical representation of the size $\text{card}(F(s))$ of the folding space $F(s)$ for a random primary structure of length n.
To reach this goal, we consider the combinatorial class $\mathcal{S}_s$ of all secondary structures. We can model this combinatorial class as formal language $\mathcal{L}_s$. Considering (the derivation of) Definition 2.7, a formal definition of this language can immediately be given as follows:

Definition 5.2 *The language $\mathcal{L}_s$ containing exactly all secondary structures is given by $\mathcal{L}_s := \{.\}^+ \cup \mathcal{L}_u\mathcal{L}_{l_su}^+$, where $\mathcal{L}_{l_su} := (\mathcal{L}_{l_s})\mathcal{L}_u$, $\mathcal{L}_u := \{.\}^*$ and $\mathcal{L}_{l_s}$ is the smallest language satisfying the following conditions:*

1. $\{.\}^+ \subset \mathcal{L}_{l_s}$.
2. *If $w \in \mathcal{L}_{l_s}$, then $(w) \in \mathcal{L}_{l_s}$.*
3. *If $w \in \mathcal{L}_{l_s}$, then $\{.\}^+(w) \subset \mathcal{L}_{l_s}$, $(w)\{.\}^+ \subset \mathcal{L}_{l_s}$ and $\{.\}^+(w)\{.\}^+ \subset \mathcal{L}_{l_s}$.*
4. *If $w_1, \ldots, w_n \in \mathcal{L}_{l_s}$ and $n \geq 2$, then $\mathcal{L}_u(w_1)\mathcal{L}_u(w_2)\cdots\mathcal{L}_u(w_n)\mathcal{L}_u \subset \mathcal{L}_{l_s}$.*

Considering the differences between Definition 2.7 and Definition 5.2, we can easily obtain an unambiguous context-free grammar $\mathcal{G}_s$ with $\mathcal{L}(\mathcal{G}_s) = \mathcal{L}_s$ by modifing the grammar $\mathcal{G}_1$ given in Lemma 2.2. This grammar $\mathcal{G}_s$ is given in the following lemma:

Lemma 5.1 *A context-free grammar $\mathcal{G}_s$ unambiguously generating exactly the language $\mathcal{L}_s$ is given by $\mathcal{G}_s = (I, \Sigma, R, S_s)$, where $I = \{S_s, A, B, C, D\}$, $\Sigma = \{(,), .\}$ and R contains exactly the following rules:*

$$\begin{array}{llll} S_s \to A, & S_s \to .C, & A \to C(B)D, & B \to .C, \\ B \to C(B)A, & B \to .C(B), & B \to (B)C., & B \to .C(B)C., \\ B \to (B), & C \to \epsilon, & C \to .C, & D \to \epsilon, \\ D \to .C, & D \to A & & \end{array}$$

We will now use the grammar $\mathcal{G}_s$ to compute an asymptotic for the expected size $\text{card}(F(s))$ of the folding space for a random primary structure s of length n, i.e. for the expected number of different secondary structures of length n that are compatible with a random primary structure s of length n.
First, by translating the rule set R of $\mathcal{G}_s$ into a system of equations, we obtain:

$$\begin{aligned} S_s(z) &= A(z) + z \cdot C(z), \\ A(z) &= C(z) \cdot (z \cdot B(z) \cdot z) \cdot D(z), \\ B(z) &= z \cdot C(z) + C(z) \cdot (z \cdot B(z) \cdot z) \cdot A(z) + \\ &\quad z \cdot C(z) \cdot (z \cdot B(z) \cdot z) + (z \cdot B(z) \cdot z) \cdot C(z) \cdot z + \\ &\quad z \cdot C(z) \cdot (z \cdot B(z) \cdot z) \cdot C(z) \cdot z + (z \cdot B(z) \cdot z), \\ C(z) &= 1 + z \cdot C(z), \\ D(z) &= 1 + z \cdot C(z) + A(z). \end{aligned} \tag{5}$$

By solving this sytem for $S_s(z)$, we obtain an ordinary generating function counting the number of secondary structures of length n, and by computing an asymptotic for the nth coefficient of this generating function, we obain an asymptotic for s_{0_n}.

Now, recall that for a given primary structure s of length n, each secondary structure in the folding space $F(s)$ is compatible with the primary structure s and a number of other primary structures $s' \neq s$ (for example by replacing symbol A with C and symbol U with G in s, we obtain a compatible primary structure $s' \neq s$).
To obtain the desired result for $\mathrm{card}(F(s))$ for a fixed length n, we will count all the different compatible primary structures for each secondary structure of length n. This means that for each secondary structure $\mathrm{sec} \in \mathcal{S}_{s_n}$, we will determine the number $\mathrm{c}(\mathrm{sec})$ of all primary structures that are compatible with sec (i.e. the number of all primary structures s with $\mathrm{sec} \in F(s)$). By computing the sum of these numbers of compatible primary structures over all secondary structures of length n, we obtain

$$f_n := \sum_{\mathrm{sec} \in \mathcal{S}_{s_n}} \mathrm{c}(\mathrm{sec}) = \sum_{\mathrm{sec} \in \mathcal{S}_{s_n}} \mathrm{card}(\{s \mid \mathrm{sec} \in F(s)\}),$$

which is the number of possible primary structures of length n that are compatible with the different secondary structures in $\mathcal{S}_{s_n}$, where each primary structure s is counted exactly $\mathrm{card}(F(s))$ times. Consequently, by dividing this sum f_n by the number of all possible primary structures of length n, we obtain the expected size $\mathrm{card}(F(s))$ of the folding space $F(s)$ for a random primary structure s of length n under the assumption of a uniform distribution on $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}$.
As primary structures are modeled as words over $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}$, the number of all possible primary structures of length n is obviously given by $\mathrm{card}(\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\})^n = 4^n$.

Hence, we first aim at computing an asymptotic for f_n for a random primary structure s of length n. To reach this goal, we want to transform system (5) into a new system of equations for the construction of the generating function

$$F(z) := \sum_{n \geq 0} f_n z^n = \sum_{n \geq 0} \left(\sum_{\mathrm{sec} \in \mathcal{S}_{s_n}} \mathrm{card}(\{s \mid \mathrm{sec} \in F(s)\}) \right) \cdot z^n$$

In this generating function, variable z marks length of primary structures and for each secondary structure which is generated by the underlying grammar $\mathcal{G}_s$, we have to count all the possible primary structures that are compatible with this secondary structure.
As there are always the 4 different possible choices A, C, G and U for an unpaired nucleotide, each dot representing an unpaired nucleotide must be represented by $a := 4 \cdot z$ in the generating function $F(z)$. Furthermore, each pair of corresponding brackets **( )** in all rules of the underlying grammar $\mathcal{G}_s$ representing a base pair must be translated into a factor $\mathrm{numBP} \cdot z \cdot z$ for the generating function $F(z)$, where numBP is the number of base pairs that are allowed to form. In fact, often it is assumed that only complementary bases can pair, which means that only Watson-Crick base pairs are allowed and thus $\mathrm{numBP} = \mathrm{card}(\{\mathrm{AU}, \mathrm{CG}, \mathrm{GC}, \mathrm{UA}\}) = 4$. If not only Watson-Crick but also wobble GU pairs are possible, then $\mathrm{numBP} = \mathrm{card}(\{\mathrm{AU}, \mathrm{CG}, \mathrm{GC}, \mathrm{UA}, \mathrm{GU}, \mathrm{UG}\}) = 6$.
According to the observations, the resulting system of equations is given as follows:

$$\begin{aligned}
F_s(z) &= A(z) + a \cdot C(z), \\
A(z) &= C(z) \cdot (\mathrm{numBP} \cdot z \cdot B(z) \cdot z) \cdot D(z), \\
B(z) &= a \cdot C(z) + C(z) \cdot (\mathrm{numBP} \cdot z \cdot B(z) \cdot z) \cdot A(z) + \\
&\quad a \cdot C(z) \cdot (\mathrm{numBP} \cdot z \cdot B(z) \cdot z) + \\
&\quad (\mathrm{numBP} \cdot z \cdot B(z) \cdot z) \cdot C(z) \cdot a + \\
&\quad a \cdot C(z) \cdot (\mathrm{numBP} \cdot z \cdot B(z) \cdot z) \cdot C(z) \cdot a + \\
&\quad (\mathrm{numBP} \cdot z \cdot B(z) \cdot z), \\
C(z) &= 1 + a \cdot C(z), \\
D(z) &= 1 + a \cdot C(z) + A(z).
\end{aligned} \tag{6}$$

By solving system (6) for $F_s(z)$, we obtain a closed form of the desired generating function $F(z)$. Hence, by computing an asymptotic for $[z^n]F(z) = f_n$ and afterwards dividing it by the term 4^n, we obtain an asymptotical representation of the expected size $\mathrm{card}(F(s))$ of the folding space $F(s)$ for a random primary structure s of length n assuming a minimum number of 1 unpaired bases in hairpin loops and a minimum number of 1 base pairs in ladders.
But as in nature, hairpin loops of length less than 3 do not form, it seems appropriate to compute the desired results under the assumption of a minimum number of 3 unpaired bases in hairpin loops. Similarly,

	Only Watson-Crick Pairs (numBP = 4)	Watson-Crick and Wobble GU Pairs (numBP = 6)
$\text{minL}_{\text{ladder}} = 1$ and $\text{minL}_{\text{hairpin}} = 1$	$2^{-n-\frac{1}{2}}\left(2-\sqrt{3}\right)^{-n-2} \cdot \left(\frac{1}{n}\right)^{3/2}\sqrt{\frac{-12+7\sqrt{3}}{\pi}}$ $\approx 1.86603^n \cdot 1.95947 \cdot n^{-3/2}$	$2^{\frac{1}{2}-n}3^n(2+\sqrt{6}-2\sqrt{1+\sqrt{6}})^{-n-2} \cdot \left(\frac{1}{n}\right)^{3/2}\sqrt{\frac{-4\left(9+4\sqrt{6}\right)+\sqrt{6\left(481+201\sqrt{6}\right)}}{\pi}}$ $\approx 2.04101^n \cdot 1.6374 \cdot n^{-3/2}$
$\text{minL}_{\text{ladder}} = 1$ and $\text{minL}_{\text{hairpin}} = 3$	$1.72139^n \cdot 1.54195 \cdot n^{-3/2}$	$1.85479^n \cdot 1.22479 \cdot n^{-3/2}$
$\text{minL}_{\text{ladder}} = 2$ and $\text{minL}_{\text{hairpin}} = 1$	$1.36247^n \cdot 5.8205 \cdot n^{-3/2}$	$1.49265^n \cdot 4.16417 \cdot n^{-3/2}$
$\text{minL}_{\text{ladder}} = 2$ and $\text{minL}_{\text{hairpin}} = 3$	$1.33089^n \cdot 5.11834 \cdot n^{-3/2}$	$1.44358^n \cdot 3.45373 \cdot n^{-3/2}$

Table 1: Asymptotics for the expected sizes $\text{card}(F(s))$ for a random primary structure s of length n assuming a minimum hairpin length $\text{minL}_{\text{hairpin}}$ and a minimum ladder length $\text{minL}_{\text{ladder}}$, for each possible combination of $\text{minL}_{\text{hairpin}} \in \{1,3\}$, $\text{minL}_{\text{ladder}} \in \{1,2\}$ and $\text{numBP} \in \{4,6\}$, respectively.

as the size of the folding space $F(s)$ for a random primary structure s of length n has been estimated by Giegerich et al. [GVR04] under the assumption that no isolated base pairs, i.e. no ladders consisting of less than 2 base pairs, can occur, it seems appropriate to determine the desired results for a minimum length of 2 for ladders. In fact, under the assumption of a minimum number of $\text{minL}_{\text{hairpin}} \geq 1$ unpaired bases for hairpin loops and a minimum number of $\text{minL}_{\text{ladder}} \geq 1$ base pairs for ladders, we have to consider the following system of equations, which can immediately be obtained from system (6):

$$\begin{aligned}
F_s(z) &= A(z) + a \cdot C(z), \\
A(z) &= C(z) \cdot (\text{numBP}^{\text{minL}_{\text{ladder}}} \cdot z^{\text{minL}_{\text{ladder}}} \cdot B(z) \cdot z^{\text{minL}_{\text{ladder}}}) \cdot D(z), \\
B(z) &= a^{\text{minL}_{\text{hairpin}}} \cdot C(z) + C(z) \cdot (\text{numBP}^{\text{minL}_{\text{ladder}}} \cdot z^{\text{minL}_{\text{ladder}}} \cdot B(z) \cdot z^{\text{minL}_{\text{ladder}}}) \cdot A(z) + \\
&\quad a \cdot C(z) \cdot (\text{numBP}^{\text{minL}_{\text{ladder}}} \cdot z^{\text{minL}_{\text{ladder}}} \cdot B(z) \cdot z^{\text{minL}_{\text{ladder}}}) + \\
&\quad (\text{numBP}^{\text{minL}_{\text{ladder}}} \cdot z^{\text{minL}_{\text{ladder}}} \cdot B(z) \cdot z^{\text{minL}_{\text{ladder}}}) \cdot C(z) \cdot a + \\
&\quad a \cdot C(z) \cdot (\text{numBP}^{\text{minL}_{\text{ladder}}} \cdot z^{\text{minL}_{\text{ladder}}} \cdot B(z) \cdot z^{\text{minL}_{\text{ladder}}}) \cdot C(z) \cdot a + \\
&\quad (\text{numBP} \cdot z \cdot B(z) \cdot z), \\
C(z) &= 1 + a \cdot C(z), \\
D(z) &= 1 + a \cdot C(z) + A(z).
\end{aligned} \tag{7}$$

Finally, solving system (7) for $F_s(z)$, we obtain the desired closed form of the generating function $F(z, \text{minL}_{\text{hairpin}}, \text{minL}_{\text{ladder}})$, where $F(z,1,1) = F(z)$. An asymptotic for the expected size of $F(s)$ for a random primary structure s of length n assuming a minimum number of $\text{minL}_{\text{hairpin}}$ unpaired bases in hairpin loops and a minimum number of $\text{minL}_{\text{ladder}}$ base pairs in ladders can be obtained by using Darboux's theorem for $F(z, \text{minL}_{\text{hairpin}}, \text{minL}_{\text{ladder}})$ and afterwards dividing by the term 4^n.
Table 1 contains the resulting asymptotics[4] for the expected folding space sizes $\text{card}(F(s))$ for a random primary structure s of length n for each possible combination of $\text{minL}_{\text{hairpin}} \in \{1,3\}$, $\text{minL}_{\text{ladder}} \in \{1,2\}$ and $\text{numBP} \in \{4,6\}$.

Obviously, the exponential growth factors of the different expected folding space sizes under the assumption of a minimum ladder length $\text{minL}_{\text{ladder}} = 2$ are closer to the experimentally obtained value of $1.3968912^n \approx 1.4^n$ given in [GVR04] than under the assumption of a minimum ladder length $\text{minL}_{\text{ladder}} = 1$, which is due to the fact that this value has been obtained by considering only secondary structures without isolated base pairs. In fact, for $\text{minL}_{\text{hairpin}} = 1$, $\text{minL}_{\text{ladder}} = 2$ and $\text{numBP} = 4$ (only Watson-Crick base pairs are allowed), the exponential growth factor of the corresponding asymptotic best matches the

[4] These asymptotics have been derived by applying Darboux's theorem with a choice of $m = 0$ to the respective generating functions $F(z, \text{minL}_{\text{hairpin}}, \text{minL}_{\text{ladder}})$.

experimentally obtained exponential growth factor given in [GVR04].

Note that the expected size of the folding space $F(s)$ for a random primary structure s of length n is equal to the expected number of secondary structures of size n under the assumption of the Bernoulli-model for RNA secondary structures, given a uniform distribution of the bases A, C, G and U.
The Bernoulli-model was already considered in some other works, for example in [HSS98, Neb04, ZS84]. It is more realistic than the combinatorial model, as it is capable of incorporating information on the primary structure for a given secondary structure. In fact, it is obtained by a stochastic approach, where a Bernoulli distribution of the bases is assumed and a parameter p is incorporated to specify the probability that two random bases can form a hydrogen bond. In the style of [Les74] and [Neb04], this base-pairing probability p is often called *stickiness*. Thus, using the stickiness p, the coefficients of the resulting generating function describe no longer absolute numbers, but expected values depending on p, i.e. expected values supposing that only structures which are compatible with a random primary structure are counted. It should be noted that an asymptotical representation of the expected number of secondary structures of size n under the assumption of the Bernoulli-model with a stickiness p (for a minimum number of $\text{minL}_{\text{hairpin}} = 1$ unpaired bases in hairpin loops and a minimum number of $\text{minL}_{\text{ladder}} = 1$ base pairs in ladders) can be found in [Neb04]. Thus, for numBP $= 4$ and numBP $= 6$, the expected size of $F(s)$ for a random primary structure s of length n for $\text{minL}_{\text{hairpin}} = 1$ and $\text{minL}_{\text{ladder}} = 1$ could have equivalently been obtained by setting $p = 1/4$ and $p = 3/8$ in this asymptotic.

Now, we would like to extend the Bernoulli-model for RNA secondary structures to shape representations of secondary structures in order to derive the corresponding expected values for the different shape space sizes.
In fact, for every $i \in \{1, \ldots, 5\}$, our aim was to compute asymptotical representations of the expected size $\text{card}(P_i(s))$ of the shape space $P_i(s)$ for a random primary structure s of length n under the assumption of a minimum hairpin length $\text{minL}_{\text{hairpin}}$ and a minimum ladder length $\text{minL}_{\text{ladder}}$ for each possible combination of $\text{minL}_{\text{hairpin}} \in \{1, 3\}$, $\text{minL}_{\text{ladder}} \in \{1, 2\}$ and numBP $\in \{4, 6\}$.
But unfortunately, we are currently not able to derive these results on the expected shape space sizes, so that the only information available so far are the experimental results presented in [GVR04].

6 Summary

In this article, we have derived some interesting results answering enumeration problems for abstract shapes and secondary structures of RNA. In fact, we have computed asymptotical representations for the respective numbers depending on the length n of the corresponding structures for $n \to \infty$. To obtain these asymptotics, we have used context-free grammars, generating functions and Darboux's theorem.
We started our investigations of abstract shapes by deriving asymptotical representations for the number of type i shapes representations of length n, for each $i \in \{1, \ldots, 4\}$[5]. Afterwards, we analyzed the size of the folding space $F(s)$ and of the shape spaces $P_i(s)$, $1 \leq i \leq 5$. First, we considered a combinatorial model for secondary structures and shapes to find out how much the search space can be reduced by using the abstract shape approach under the assumption that base pairing is allowed between arbitrary pairs of bases. In particular, for each $i \in \{1, \ldots, 5\}$, we computed asymptotical representations of the number of type i shapes that are homomorphic images of secondary structures of length n and compared these numbers to the number of secondary structures of length n. This yielded the observation that the search space is reduced significantly by using the concept of abstract shapes; the reduction has been quantified precisely.
To obtain more realistic results, we have considered the Bernoulli-model for RNA secondary structures. In fact, we have described how to derive asymptotics for the expected size $\text{card}(F(s))$ of the folding space $F(s)$ for a random primary structure s of length n assuming a minimum hairpin length $\text{minL}_{\text{hairpin}}$, a minimum ladder length $\text{minL}_{\text{ladder}}$, a number numBP of possible base pairs and a uniform distribution on the nucleotides of a random primary structure. The resulting asymptotics have been presented for each possible combination of $\text{minL}_{\text{hairpin}} \in \{1, 3\}$, $\text{minL}_{\text{ladder}} \in \{1, 2\}$ and numBP $\in \{4, 6\}$, respectively, as these are common choices for the corresponding parameters.
As already mentioned, we tried to extend the Bernoulli-model for RNA secondary structures to a corresponding model for shape representations in order to derive asymptotics for the expected sizes $\text{card}(P_i(s))$ of the shape spaces $P_i(s)$, $i \leq i \leq 5$, for a random primary structure s of length n assuming a minimum hairpin length $\text{minL}_{\text{hairpin}}$, a minimum ladder length $\text{minL}_{\text{ladder}}$, a number numBP of possible

[5] For $i = 5$, the corresponding asymptotic number has already been determined in [LPC08].

base pairs and a uniform distribution on the nucleotides of a random primary structure. The idea behind this approach was to compute the corresponding asymptotics for each possible combination of $\text{minL}_{\text{hairpin}} \in \{1, 3\}$, $\text{minL}_{\text{ladder}} \in \{1, 2\}$ and $\text{numBP} \in \{4, 6\}$, respectively, as we have done for our investigations on the expected size of the folding space in this article, such that the exponential growth factors of the derived asymptotics (for each of the considered models) could be compared to the 6 experimentally obtained values given in [GVR04] and [VGR06].
However, we are currently not able to derive the desired asymptotics for the expected shape spaces sizes. Hence, it will be a further task to find a way to extend the Bernoulli-model for RNA secondary structures to a corresponding model for shape representations such that these results could finally be computed.

References

[AvdBvBP90] J. P. Abrahams, M. van den Berg, E. van Batenburg, and C. W. Pleij. Prediction of RNA secondary structure, including pseudoknotting, by computer simulation. *Nucleic Acids Res.*, 18(10):3035–3044, 1990.

[Com74] Louis Comtet. *Advanced Combinatorics; The art of finite and infinite expansions.* Reidel Publ. Co., Dordrecht, rev. and enl. edition, 1974.

[CS63] N. Chomsky and M. P. Schützenberger. The algebraic theorey of context-free languages. In P. Braffort and D. Hirschberg, editors, *Computer Programming and Formal Systems*, pages 118–161. North-Holland, Amsterdam, 1963.

[DPD92] E. Dam, K. Pleij, and D. Draper. Structural and functional aspects of RNA pseudoknots. *Biochemistry*, 31:11665–11676, 1992.

[Edd04] Sean R. Eddy. How do RNA folding algorithms work. *Nature Biotechnology*, 22(11):1457–1458, 2004.

[FS07] Philippe Flajolet and Robert Sedgewick. Analytic combinatorics. Preliminary version, March 2007.

[GK90] Daniel H. Greene and Donald E. Knuth. *Mathematics for the Analysis of Algorithms.* Birkhäuser Boston, third edition, 1990.

[GVR04] Robert Giegerich, Björn Voß, and Marc Rehmsmeier. Abstract shapes of RNA. *Nucleic Acids Research*, 32(16):4843–4851, 2004.

[GW90] R. R. Gutell and C. R. Woese. Higher order structural elements in ribosomal RNAs: Pseudo-knots and the use of noncanonical pairs. *Proc. Natl. Acad. Sci. USA*, 87:663–667, 1990.

[Har78] Michael A. Harrison. *Introduction to Formal Language Theory.* Addison-Wesley, 1978.

[HMU01] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 2nd edition, 2001.

[HSS98] Ivo L. Hofacker, Peter Schuster, and Peter F. Stadler. Combinatorics of RNA secondary structures. *Discrete Applied Mathematics*, 88:207–237, 1998.

[KW89] Donald E. Knuth and Herbert S. Wilf. A short proof of Darboux's lemma. *Applied Mathematics Letters*, 2:139–140, 1989.

[Les74] A. M. Lesk. A combinatorial study of the effects of admitting non-Watson-Crick base pairings and of base compositions on the helix-forming potential of polynucleotides of random sequences. *J. Theor. Biol.*, 44:7–17, 1974.

[LPC08] W. A. Lorenz, Y. Ponty, and P. Clote. Asymptotics of RNA shapes. *Journal of Computational Biology*, 15(1):31–63, January 2008.

[MN08] Dirk Metzler and Markus E. Nebel. Predicting RNA secondary structures with pseudoknots by MCMC sampling. *Journal of Mathematical Biology*, 56:161–181, 2008.

[Neb02] Markus E. Nebel. Combinatorial properties of RNA secondary structures. *Journal of Computational Biology*, 9(3):541–574, 2002.

[Neb04] Markus E. Nebel. Investigation of the Bernoulli-model of RNA secondary structures. *Bulletin of Mathematical Biology*, 66:925–964, 2004.

[NJ80] R. Nussinov and A. B. Jacobson. Fast algorithms for predicting the secondary structure of single-stranded RNA. *Proceedings of the National Academy of Science of the USA*, 77(11):6309–6313, 1980.

[NPGK78] R. Nussinov, G. Pieczenik, J. R. Griggs, and D. J. Kleitman. Algorithms for loop matchings. *SIAM Journal on Applied Mathematics*, 35:68–82, 1978.

[PB89] C. W. Pleij and L. Bosch. RNA pseudoknots: structure, detection, and prediction. *Methods Enzymol.*, 180:289–303, 1989.

[Ple94] C. W. Pleij. RNA pseudoknots. *Curr. Opin. Struct. Biol.*, 4:337–344, 1994.

[RE99] E. Rivas and S. R. Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.*, 285:2053–2068, 1999.

[RG05] Jens Reeder and Robert Giegerich. Consensus shapes: an alternative to the Sankoff algorithm for RNA consensus structure prediction. *Bioinformatics*, 21(17):3516–3523, 2005.

[SKMC83] D. Sankoff, J. B. Kruskal, S. Mainville, and R. J. Cedergren. Fast algorithms to determine RNA secondary structures containing multiple loops. In *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison*, chapter 3, pages 93–120. Addison-Wesley, Reading, MA, 1983.

[SVR+06a] Peter Steffen, Björn Voß, Marc Rehmsmeier, Jens Reeder, and Robert Giegerich. RNAshapes 2.1.1 manual, February 2006.

[SVR+06b] Peter Steffen, Björn Voß, Marc Rehmsmeier, Jens Reeder, and Robert Giegerich. RNAshapes: an integrated RNA analysis package based on abstract shapes. *Bioinformatics*, 22(4):500–503, 2006.

[SW78] P. R. Stein and M. S. Waterman. On some new sequences generalizing the Catalan and Motzkin numbers. *Discrete Mathematics*, 26:216–272, 1978.

[TUL71] I. Tinoco, O. Uhlenbeck, and M. D. Levine. Estimation of secondary structure in ribonucleic acids. *Nature*, 230:362–367, 1971.

[VC85] G. Viennot and M. Vauchaussade De Chaumont. Enumeration of RNA secondary structures by complexity. *Mathematics in medicine and biology, Lecture Notes in Biomathematics*, 57:360–365, 1985.

[VGR06] Björn Voß, Robert Giegerich, and Marc Rehmsmeier. Complete probabilistic analysis of RNA shapes. *BMC Biology*, 4(5), 2006.

[Wat78] M. S. Waterman. Secondary structure of single-stranded nucleic acids. *Advances in Mathematics Supplementary Studies*, 1:167–212, 1978.

[Wil94] Herbert S. Wilf. *generatingfunctionology*. Academic Press, Inc., second edition, 1994.

[ZS81] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9:133–148, 1981.

[ZS84] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bull. Mathematical Biology*, 46:591–621, 1984.

[Zuk89a] M. Zuker. Computer prediction of RNA structure. In J. E. Dahlberg and J. N. Abelson, editors, *RNA Processing*, volume 180 of *Methods in Enzymology*, pages 262–288. Acad. Pr., San Diego, 1989.

[Zuk89b] M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244:48–52, 1989.