On Quantitative Effects of RNA Shape Abstraction

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

Abstract

Over the last decades, much effort has been made to develop approaches for identifying good predictions of RNA secondary structure. This is due to the fact that most computational prediction methods based on free energy minimization compute a number of suboptimal foldings and we have to identify the native folding among all these possible secondary structures. 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 compute precise asymptotics for the number of different shape representations of size n and for the number of different shapes showing up when abstracting from secondary structures of size n under a combinatorial point of view. A more realistic model taking primary structures into account remains an open challenge – we give some arguments why the present techniques cannot be applied in this case.

1 Introduction

Ribonucleic acid (RNA) is a single-stranded molecule. Nucleotides, composed of a phosphate group, a sugar group (ribose) and one of the four bases adenine (A), cytosine (C), guanine (G) and uracil (U), are the basic structural units of such nucleic acids. An RNA single-strand is formed by *phosphodiester bonds* linking together nucleotides and typically is modeled as a word over the alphabet $\Sigma = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}$ representing the four different bases. 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 neighbored nucleotides, two bases that are not neighbored within the primary structure may form hydrogen bonds to each other. Those bonds are most likely between the complementary bases adenine (A) and uracil (U) resp. cytosine (C) and guanine (G) yielding stable *Watson-Crick pairings*. In addition, weaker base pairs formed by the non-complementary bases guanine (G) and uracil (U) – so called *wobble GU pairs* – may occur. Other pairings are possible, but they are much less likely and 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 *tertiary structure* of the molecule, which in many cases determines its function. It is customary in science to allow only non-crossing (nested) base pairs, such that the primary structure remains planar, i.e. a two-dimensional conformation, called the *secondary structure*.
The experimental determination of RNA tertiary structures is usually time-consuming and expensive and therefore, much effort has been made to predict a molecules structures by means of computational methods. However, determining the tertiary structure is computationally complex, but it has proven convenient to first search for the less complex secondary structure which can be determined efficiently. As much of the 3D structure is determined by the base-pairing interactions *in the plane* this allows for useful conclusions towards the tertiary structure. 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. The most successful and popular method for energy minimization over the last 30 years has been the use of dynamic programming algorithms. While some of the first algorithms in this field computed only one secondary structure with the lowest free energy according to different energy models (see, e.g. [NPGK78, NJ80, ZS81, SKMC83, ZS84]), most of the currently used algorithms

*Corresponding author.

additionally predict suboptimal foldings (see, e.g. [Zuk89, Zuk03, WFHS99, DL03, DCL04]). 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 at the end. However, this set of suboptimal foldings usually contains lots of similar structures and we are only interested in structures with significant structural differences. For this reason, the concept of *abstract shapes* was introduced by Giegerich et al. [GVR04]. Abstract shapes are morphic images of secondary structures, where 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 RNA analysis software package called *RNAShapes* has been developed [SVR$^+$06b, SVR$^+$06a]. This 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.

It should also be mentioned that recently, abstract shapes (and also certain corresponding asymptotical numbers, of which some will be derived in this article) gained significantly in importance, as a new approach for faster searching of RNA family databases based on shape abstractions, which is called *shape based indexing*, has been invented [JRG08]. In fact, asymptotics as calculated in this article (irrespective of primary structure) are correctly seen as upper bound to the size of shape indices for large databases.

Obviously, in order to analyze the expected complexity of algorithms dealing with abstract shapes, the first question is to know the size of the search space those algorithms have to deal with. Accordingly, enumeration results with respect to the number of different abstract shapes of a given size are of interest. Furthermore, since an abstract shape serves as a representative of numerous secondary structures, the (average) number of different foldings (secondary structures) represented by a single shape should be known (e.g. in connection with shape based indexing) and can be concluded from our results.

The rest of this paper is organized as follows: First, we will compute asymptotical representations for the number of different shape representations of size n. Afterwards, we will analyze the number of different shapes showing up when abstracting from secondary structures of size n under a complete combinatorial point of view (ignoring primary structure and complementarity of bases). Finally, we will discuss problems arising when switching to more realistic models taking primary structure into account.

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 2 of Definition 2.1 implies that even nucleotides only one position apart may form a hydrogen bond[1], condition 3 ensures that these graph representations remain planar. In fact, according to this constraint, *pseudoknots* are prohibited. 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[2]. An example for a secondary structure in planar graph representation is shown in Figure 1.

[1] Later, we will speak of the *minimal length of hairpin loops* being 1, but so far we don't have the right vocabulary.

[2] Allowing pseudoknots makes secondary structure prediction to become $\mathcal{NP}$ complete, which probably is the reason for their exclusion.

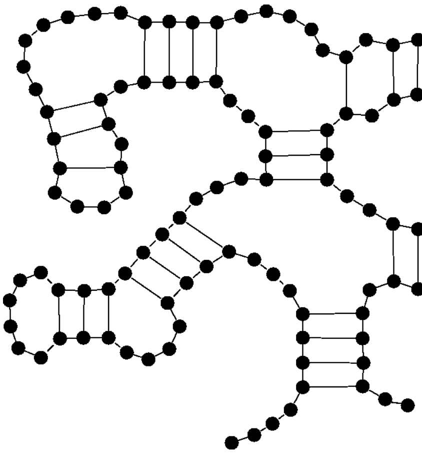

Figure 1: An RNA secondary structure.

Besides this rather descriptive model, many other ways of formalizing RNA folding have been described in literature. One example is the so called *dot-bracket representation*, where a secondary structure is modeled as a string over the alphabet $\Sigma := \{(,),.\}$; a dot represents an unpaired nucleotide and a pair of corresponding brackets ( ) represents two paired bases. As abstract shapes build on this representation, we proceed with the following definition:

Definition 2.2 ([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 ( ).*

We mention by passing that words over the alphabet $\{(,)\}$ which satisfy the first two conditions of the previous definition are known as *semi-Dyck words*, whereas words over the alphabet Σ satisfying these first two conditions are known as *Motzkin words.* The model of dot-bracket representations is 1-to-1 to Waterman's planar graphs. It should be clear that both models abstract from primary structure, as they only consider the number of base pairs and unpaired bases and their positions.
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, partially given in [Neb04]:

Definition 2.3 *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 $v = w_{i+1} \dots w_{j-1}$ is a* (hairpin) loop, *if $v \in \{.\}^+$ and $w_i w_j =$ ( ) is a corresponding pair of brackets of w.*
2. *The subword $v = w_{i+1} \dots w_{j-1}$ 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. A bulge for which $w_i =$) and $w_j =$ (holds is called* join.
3. *An* interior loop *is two subwords (bulges) $u = w_{i+1} \dots w_{j-1}$ and $v = w_{k+1} \dots w_{l-1}$ 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* tail *is a prefix $v = w_1 \dots w_i$ resp. a suffix $v = w_j \dots w_n$ such that $v \in \{.\}^+$ and w_{i+1} resp. w_{j-1} is in $\{(,)\}$.*
5. *A* ladder *(or* helical region*) consists of two maximal subwords u, v such that $u = w_i \dots w_{i+c}$ and $v = w_j \dots w_{j+c}$ 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 $c + 1$.*
6. *A* hairpin *is a subword $v = w_{i+1} \dots w_{j-1}$ 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.*

7. *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.*

Note that the third condition of Definition 2.2 only ensures that a hairpin loop consists of at least one unpaired nucleotide in accordance with Waterman's model. Though in reality, hairpin loops of length less than three are impossible and do not form[3].
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.*

Furthermore, every secondary structure of an RNA molecule that is not completely unpaired forms an *exterior loop*, which can be a seen as a list of adjacent substructures or adjacent structural components of this secondary structure.
An illustration of the different structural components is given in Figure 2, as well as in the following example.

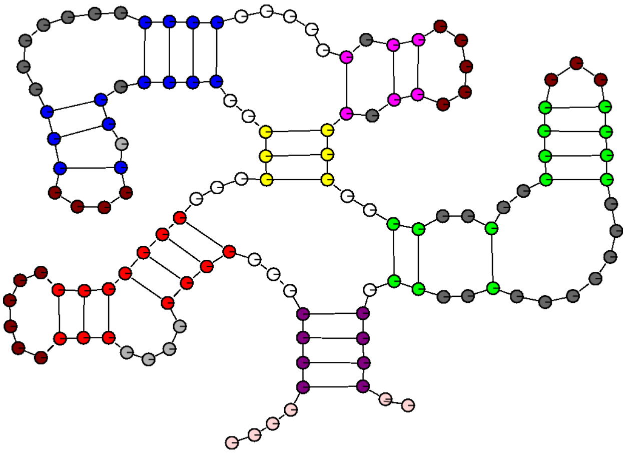

Figure 2: Colored version of the secondary structure shown in Figure 1. For each helical region, paired bases are colored with matching colors. Additionally, hairpin loops are colored brown, single bulges interrupting ladders are colored light gray, interior loops are colored dark gray, and unpaired regions in multiloops and exterior loops are colored white and light pink, respectively. The exterior loop is composed of three adjacent structural components: a tail (light pink), a folded region and another tail (also light pink). The structure possesses two multiloops; the first with three helical regions (colored red, yellow and green) and the second with two helical regions (colored blue and pink).

Example 2.1 *The secondary structure shown in Figure 1 (and in Figure 2) can be represented by a dot-bracket word which can be decomposed into subwords corresponding to basic structural components as follows:*

$$\overbrace{\underbrace{....}_{\textit{tail}}\underbrace{((((\overbrace{\underbrace{...}_{\textit{bulge}}\ \textit{HELIX}_1\ \underbrace{....}_{\textit{join}}\ \textit{HELIX}_2\ \underbrace{..}_{\textit{join}}\ \textit{HELIX}_3\ \underbrace{.}_{\textit{bulge}}}^{\textit{multiloop}}))))}_{\textit{folded region (helix)}}\underbrace{..}_{\textit{tail}}}^{\textit{exterior loop}}$$

Subwords:

[3] Of course, it would be an easy task to change the definition in order to allow loops of length at least 3 only. However, when changing to enumeration and corresponding methods from singularity analysis, such a change would imply polynomials of higher degree and the need to compute their roots. Thus, in order to keep the mathematics behind the model manageable, one probably resigned this modification. Nevertheless, for covariance models, where these reasons do not apply, one sometimes allows loops of length 0 in the consensus.

- HELIX_1:

$$\underbrace{((((\underbrace{....}_{\textit{bulge}}\overbrace{(((\underbrace{......}_{\textit{hairpin loop}})))}^{\textit{hairpin}}))))}_{\textit{helix}}$$

- HELIX_2:

$$\underbrace{(((\overbrace{\underbrace{..}_{\textit{bulge}}\underbrace{((((\underbrace{.}_{\textit{bulge}}((\underbrace{.}_{\textit{bulge}}\overbrace{(\underbrace{....}_{\textit{hairpin loop}})}^{\textit{hairpin}}))\underbrace{.......}_{\textit{bulge}}))))}_{\textit{helix}}\underbrace{.....}_{\textit{join}}\underbrace{(\underbrace{.}_{\textit{bulge}}\overbrace{((\underbrace{.....}_{\textit{hairpin loop}}))}^{\textit{hairpin}}\underbrace{.}_{\textit{bulge}})}_{\textit{helix}}}^{\textit{multiloop}})))}_{\textit{helix}}$$

- HELIX_3:

$$\underbrace{((\underbrace{..}_{\textit{bulge}}(\underbrace{..}_{\textit{bulge}}\overbrace{((((\underbrace{...}_{\textit{hairpin loop}}))))}^{\textit{hairpin}}\underbrace{.......}_{\textit{bulge}})\underbrace{..}_{\textit{bulge}}))}_{\textit{helix}}$$

Note that the shown decomposition of this dot-bracket representation 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 and is due to the chemical structure of the molecule.

2.2 Abstract Shapes of RNA

In this section, 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. Two of them, namely type 1 and type 5 (also called π' and π shapes, respectively), were formally defined by a tree morphism in [GVR04]. All five different shape levels were first introduced and informally described in [SVR$^+$06a] making use of dot-bracket representations. However, since the different shape types were supposed to gradually increase abstraction and it was later observed that the shape definitions given in [SVR$^+$06a] were not appropriate, the different abstraction levels were redefined (informally) in [JRG08].
In this paper, we will consider the renewed shape abstraction types as described in [JRG08]. Results like those of this article related to the original definitions can be found in [SN08]. In fact, the results presented in this previous work provide evidence that the original hierarchy of abstraction levels as introduced in [SVR$^+$06a] was not properly ordered[4].
Common to all levels is their abstraction 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 the most accurate shape type (type 1), all structural components (except hairpin loops) contribute to the shape representation. The succeeding shape types gradually increase abstraction by disregarding certain unpaired regions or combining nested helices.
In general, entire helical regions are represented by a pair of opening and closing squared brackets **[** resp. **]** and unpaired regions are represented by a single underscore **_**, see [GVR04].
According to [JRG08] the different shape types are defined as follows:

Type 1:

Most accurate – all loops and all unpaired

[4] In accordance with observations independently made by R. Giegerich at about the same time (personal communication).

Accordingly, each helical region is depicted by a single pair of opening and closing squared brackets and all unpaired regions (except hairpin loops[5]) are represented as a single underscore. Thus, all structural components contribute to this shape representation[6], nesting and adjacency of helices are retained. As a consequence, this shape type only abstracts from loop and ladder lengths.

Type 2:

Nesting pattern for all loop types and
unpaired regions in ladder interrupting bulges and interior loops

Consequently, all helical regions (ladders) are depicted by a pair of opening and closing squared brackets and furthermore, single ladder interrupting bulges and unpaired regions in interior loops are represented as a single underscore, respectively. 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 single-stranded regions in exterior loops and multiloops are omitted.

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.

Type 4:

No nesting pattern for ladder interruptions by single bulges,
nesting pattern for all other loop types and no unpaired regions

Compared to type 3 shapes, the only difference is that nested helices which are only interrupted by a single bulge are combined and represented by one pair of squared brackets only.

Type 5:

Most abstract – helix nesting pattern and no unpaired regions

In this shape abstraction, we do not account for any helix interruptions (by single bulges or interior loops). This means that (interrupted) ladders are depicted by a pair of opening and closing squared brackets, since nested helices are now always combined.

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

Example 2.2 *Considering the dot-bracket representation of the secondary structure shown in Figure 1 and its decomposition into structural components as presented in Example 2.1, the differences between the five shape types resp. the five abstraction levels are easy to see:*

Sec. str.	....	((((	...	$HELIX_1$	...	$HELIX_2$	..	$HELIX_3$	.	))))	..
Type 1	_	[	_	$HELIX_1$	_	$HELIX_2$	_	$HELIX_3$	_	]	_
Type 2		[		$HELIX_1$		$HELIX_2$		$HELIX_3$		]	
Type 3		[		$HELIX_1$		$HELIX_2$		$HELIX_3$		]	
Type 4		[		$HELIX_1$		$HELIX_2$		$HELIX_3$		]	
Type 5		[		$HELIX_1$		$HELIX_2$		$HELIX_3$		]	

$HELIX_1$:

[5] According to the informal description of level 1 shapes given in [JRG08], it is not clear whether the (one and only but always existing) unpaired region in a hairpin must be recorded on this shape abstraction level or not. Here, we decided to follow the definition used by the *RNAShapes* tool, which is available at `http://bibiserv.techfak.uni-bielefeld.de/rnashapes/welcome.html`. This tool assumes that hairpin loops are not recorded.

[6] Note that it does not matter if a hairpin is represented only by a pair of corresponding squared brackets or by a pair of corresponding squared brackets with an underscore in between, as there must always exist an unpaired region of length at least one in any hairpin.

Sec. str.	((((	....	(((	......	)))	))))
Type 1	[	_	[		]	]
Type 2	[	_	[		]	]
Type 3	[		[		]	]
Type 4	[					]
Type 5	[					]

HELIX$_2$*:*

Sec. str.	(((	..	((((	.	((	.	(	....	)	))	....	))))	.....	(	.	((	.....	))	.	)	)))
Type 1	[	_	[	_	[	_	[		]	]	_	]	_	[	_	[		]	_	]	]
Type 2	[		[	_	[	_	[		]	]	_	]		[	_	[		]	_	]	]
Type 3	[		[		[		[		]	]		]		[		[		]		]	]
Type 4	[		[		[					]		]		[		[		]		]	]
Type 5	[		[									]		[						]	]

HELIX$_3$*:*

Sec. str.	((	..	(	..	((((	...	))))	....	)	..	))
Type 1	[	_	[	_	[		]	_	]	_	]
Type 2	[	_	[	_	[		]	_	]	_	]
Type 3	[		[		[		]		]		]
Type 4	[		[		[		]		]		]
Type 5	[										]

2.2.2 Shape Languages

We now want to invent formal definitions of languages containing exactly all shapes of a certain type. At this point as well as for our further studies, we assume the reader to be familiar with basic notions from formal languages and grammars (see e.g. [HMU01] or [Har78] if needed).

For $\mathcal{L}_i$ the language of all shapes of type i, our first goal is to provide a formal definition of language $\mathcal{L}_1$. We start by observing 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 language $\mathcal{L}_1$. On the other hand, each secondary structure not totally unpaired represents an exterior loop containing at least one helical region. The first (last) helical region in this exterior loop may be preceeded (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 exterior loop contains $n \geq 1$ adjacent helices can be represented as a word

$$s = t_0(^{a_1}u_1)^{a_1}t_1 \cdots t_{n-1}(^{a_n}u_n)^{a_n}t_n,$$

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 like $v_0[w_1] \cdots v_{n-1}[w_n]v_n$, where each of the words w_i is the morphic image of subword u_i of s, $1 \leq i \leq n$, and every word $v_i \in \{\_, \varepsilon\}$, ε the empty word. Thus, let $\mathcal{L}_u = \{\_, \varepsilon\}$ be the language of the two possible morphic images of unpaired regions and let $\mathcal{L}_{l_1}$ be the language containing exactly all morphic images of helices. Furthermore, let $\mathcal{L}_{l_1u} := \mathcal{L}_{l_1}\mathcal{L}_u$ be the concatenation of these two languages. 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 $\{\_\} \cup \mathcal{L}_u\mathcal{L}_{l_1u}^+$.

To get a structural characterization of $\mathcal{L}_{l_1}$, we observe that a helix may be a hairpin, which is represented by a word $(^a.^+)^a$, $a \geq 1$, 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 subwords $(^a.^+(^bu)^b)^a$, $(^a(^bu)^b.^+)^a$ and $(^a.^+(^bu)^b.^+)^a$, for some $a, b \geq 1$, where $(^bu)^b$ is again a helix. As both ladders and bulges interrupting loops contribute to type 1 representations, their morphic 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 characterized as follows:

Definition 2.4 *The language $\mathcal{L}_1$ containing exactly all type 1 shapes is given by $\mathcal{L}_1 := \{\_\} \cup \mathcal{L}_u\mathcal{L}_{l_1u}^+$, $\mathcal{L}_u := \{\_, \varepsilon\}$, for $\mathcal{L}_{l_1u} := \mathcal{L}_{l_1}\mathcal{L}_u$ and $\mathcal{L}_{l_1}$ 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* $[v_0 w_1 v_1 w_2 \ldots v_{n-1} w_n v_n] \in \mathcal{L}_{l_1}$.

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

Definition 2.5 *The language* $\mathcal{L}_1$ *containing exactly all type 1 shapes is given by* $\mathcal{L}_1 := \{\_\} \cup \mathcal{L}_u \mathcal{L}_{l_1 u}^{+}$, $\mathcal{L}_u := \{\_, \varepsilon\}$, *for* $\mathcal{L}_{l_1 u} := [\mathcal{L}_{l_1}]\mathcal{L}_u$ *and* $\mathcal{L}_{l_1}$ *the smallest language satisfying the following conditions:*

1. $\varepsilon \in \mathcal{L}_{l_1}$.
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 and unpaired regions in exterior loops and multiloops 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 $\mathcal{L}_1$. We immediately obtain:

Definition 2.6 *The language* $\mathcal{L}_2$ *containing exactly all type 2 shapes is given by* $\mathcal{L}_2 := \{\varepsilon\} \cup \mathcal{L}_{l_2 u}^{+}$, *where* $\mathcal{L}_{l_2 u} := [\mathcal{L}_{l_2}]$ *and* $\mathcal{L}_{l_2}$ *is the smallest language satisfying the following conditions:*

1. $\varepsilon \in \mathcal{L}_{l_2}$.
2. *If* $w \in \mathcal{L}_{l_2}$, *then* $\_[w], [w]\_, \_[w]\_ \in \mathcal{L}_{l_2}$.
3. *If* $w_1, \ldots, w_n \in \mathcal{L}_{l_2}$ *and* $n \geq 2$, *then* $[w_1][w_2] \ldots [w_n] \in \mathcal{L}_{l_2}$.

The language $\mathcal{L}_3$ containing exactly all type 3 shapes can easily be characterized by taking into account that all single-stranded regions 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 2.7 *The language* $\mathcal{L}_3$ *containing exactly all type 3 shapes is given by* $\mathcal{L}_3 := \{\varepsilon\} \cup \mathcal{L}_{l_3 u}^{+}$, *where* $\mathcal{L}_{l_3 u} := [\mathcal{L}_{l_3}]$ *and* $\mathcal{L}_{l_3}$ *is the smallest language satisfying the following conditions:*

1. $\varepsilon \in \mathcal{L}_{l_3}$.
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 definition of the language $\mathcal{L}_4$ containing exactly all type 4 shape representations. Therefore, recall that their only difference to type 3 shapes is that nested helices are combined if the nesting is due to a helix interruption by a single bulge. As on abstraction level 4, nested helices are not yet combined if the nesting is caused by an interior loop but unpaired regions are always eliminated, it is obvious that a characterization of the language $\mathcal{L}_4$ is given as follows:

Definition 2.8 *The language* $\mathcal{L}_4$ *containing exactly all type 4 shapes is equal to the language* $\mathcal{L}_3$ *containing exactly all type 3 shapes.*

Finally, the language $\mathcal{L}_5$ containing exactly all type 5 shapes can easily be characterized by modifying the definition of the language $\mathcal{L}_3$, such that no nesting patterns for interruptions of ladders (by single bulges or interior loops) are retained. This yields the following characterization:

Definition 2.9 *The language* $\mathcal{L}_5$ *containing exactly all type 5 shapes is given by* $\mathcal{L}_5 := \{\varepsilon\} \cup \mathcal{L}_{l_5 u}^{+}$, *where* $\mathcal{L}_{l_5 u} := [\mathcal{L}_{l_5}]$ *and* $\mathcal{L}_{l_5}$ *is the smallest language satisfying the following conditions:*

1. $\varepsilon \in \mathcal{L}_{l_5}$.
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[7] context-free grammars $\mathcal{G}_i$ with $\mathcal{L}(\mathcal{G}_i) = \mathcal{L}_i$, $1 \leq i \leq 5$, where $\mathcal{L}(\mathcal{G})$ denotes the language generated by $\mathcal{G}$.
First, we consider $\mathcal{G}_1$ with axiom S_1, observing that for this abstraction type, the shape representation of a totally unpaired secondary structure is given by a single underscore. So the first production rules might be $S_1 \rightarrow A$ and $S_1 \rightarrow \_$, where A is the start symbol for all type 1 shapes representing a folded secondary structure. According to Definition 2.5, we observe that the language generated by nonterminal A must be equal to

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

Therefore, we use the productions $A \rightarrow C[B]D$, $C \rightarrow \varepsilon$, $C \rightarrow \_$, as well as $D \rightarrow \varepsilon$, $D \rightarrow \_$ and $D \rightarrow A$. Obviously, the language that is generated by starting with nonterminal B on the right-hand side of the production $A \rightarrow C[B]D$ must be equal to $\mathcal{L}_{l_1}$. Thus, the expression $[B]$ may generate a hairpin, a bulge interrupting a ladder, an interior loop interrupting a ladder or a ladder whose last pair is the foundation of a multiloop. Resorting to Definition 2.5 again, we immediately observe that we have to use the production rules $B \rightarrow \varepsilon$ (hairpin generating rule), $B \rightarrow \_[B]$ (generates a bulge interrupting a ladder on the left), $B \rightarrow [B]\_$ (generates a bulge interrupting a ladder on the right), $B \rightarrow \_[B]\_$ (interior loop generating rule) and $B \rightarrow C[B]A$ (multiloop generating rule). Combining all these productions, we find the following grammar which, after a moment's reflection, proofs to be unambigous:

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:*

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

We may now proceed the same way we derived the characterizations of $\mathcal{L}_i$ from that of $\mathcal{L}_{i-1}$ in order to deduce grammar $\mathcal{G}_i$ from $\mathcal{G}_{i-1}$, $2 \leq i \leq 5$. All we have to take care of is that we don't introduce ambiguity while modifying the grammars.

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 $I = \{S_2, A, B, D\}$, $\Sigma = \{\_, [,]\}$ and R contains exactly the following rules:*

$$\begin{array}{lllll} S_2 \rightarrow A, & S_2 \rightarrow \varepsilon, & & & \\ A \rightarrow [B]D, & & & & \\ B \rightarrow \varepsilon, & B \rightarrow [B]A, & B \rightarrow \_[B], & B \rightarrow [B]\_, & B \rightarrow \_[B]\_, \\ D \rightarrow \varepsilon, & D \rightarrow A. & & & \end{array}$$

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:*

$$\begin{array}{lll} S_3 \rightarrow A, & S_3 \rightarrow \varepsilon, & \\ A \rightarrow [B]D, & & \\ B \rightarrow \varepsilon, & B \rightarrow [B]A, & B \rightarrow [B], \\ D \rightarrow \varepsilon, & D \rightarrow A. & \end{array}$$

Lemma 2.5 *A context-free grammar $\mathcal{G}_4$ unambiguously generating exactly the language $\mathcal{L}_4$ is given by the grammar $\mathcal{G}_3$ of Lemma 2.4.*

[7] Unambiguity is necessary, as we later want to use these grammars to construct generating functions counting the numbers 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 sh is counted more than once by the corresponding generating function.

Type	Number of shapes
1	$s_{1_n} \sim 2.40591^n \cdot 0.989959 \cdot n^{-3/2}$
2	$s_{2_n} \sim 2.0523^n \cdot 0.88639 \cdot n^{-3/2}$
3, 4	$s_{3_n} = s_{4_n} \sim ((-2)^n + 2^n) \cdot \sqrt{\frac{2}{\pi}} \left(\frac{1}{n}\right)^{3/2} \approx ((-2.)^n + 2.^n) \cdot 0.797885 \cdot n^{-3/2}$
5	$s_{5_n} \sim 3^{n/2} \left(1 + (-1)^n\right) \cdot \sqrt{\frac{3}{2\pi}} \left(\frac{1}{n}\right)^{3/2} \approx 1.73205^n \left(1. + (-1.)^n\right) \cdot 0.690988 \cdot n^{-3/2}$

Table 1: Precise asymptotics of the numbers s_{i_n} of type i shapes of size n, $1 \le i \le 5$.

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:*

$$\begin{array}{ll}
S_5 \to A, & S_5 \to \varepsilon, \\
A \to [B]D, & \\
B \to \varepsilon, & B \to [B]A, \\
D \to \varepsilon, & D \to A.
\end{array}$$

3 Number of Shapes

We start this section by deriving combinatorial result for shapes, namely the number of type i shapes of size n, for each $i \in \{1, \ldots, 5\}$. In fact, we aim at determining asymptotical representations of the numbers s_{i_n} of type i shapes of size n, $1 \le i \le 5$. Note that asymptotics for s_{1_n} and s_{5_n} have already been determined in [LPC08]. We present them for the sake of completeness only.

Theorem 3.1 *The numbers s_{i_n} of type i shapes, $1 \le i \le 5$, of size n, $n \to \infty$, are asymptotically given in Table 1.*

Proof: To obtain the desired results, we are going to use the method of *generating functions*[8]. In particular, we first want to compute closed forms of ordinary generating functions $S_i(z)$, $1 \le i \le 5$, counting the numbers s_{i_n} of type i shapes of size n and then apply Darboux's theorem [KW89] to obtain the desired asymptotics for $s_{i_n} = [z^n]S_i(z)$, $1 \le i \le 5$.[9] According to Chomsky and Schützenberger [CS63], those generating functions can be derived by

- translating the grammars $\mathcal{G}_i$ into systems of equations for generating functions, and
- solving this system for the generating function $S_i(z)$ associated with the axiom S_i of the grammar $\mathcal{G}_i$.

During this translation, we have to keep track of the size of words to properly translate into the exponent of our generating functions' variable z, such that each shape of size n contributes to the coefficient at z^n. However, since every terminal symbol of our language contributes 1 to the size of the shape represented, this can easily be done by introducing a factor z for each terminal symbol and a factor 1 for the empty word ε.

Considering the grammars $\mathcal{G}_i$, $1 \le i \le 5$, the resulting systems of equations are given as follows:

- generating function $S_1(z)$:

$$\begin{aligned}
S_1(z) &= A(z) + z, \\
A(z) &= C(z) \cdot z^2 \cdot B(z) \cdot D(z), \\
B(z) &= 1 + C(z) \cdot z^2 \cdot B(z) \cdot A(z) + z \cdot z^2 \cdot B(z) + z^2 \cdot B(z) \cdot z + z \cdot z^2 \cdot B(z) \cdot z, \\
C(z) &= 1 + z, \\
D(z) &= 1 + z + A(z).
\end{aligned} \tag{1}$$

- generating function $S_2(z)$:

$$S_2(z) = A(z) + 1,$$

[8] 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 [FS09, 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].

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

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

- generating functions $S_3(z)$ and $S_4(z)$:

$$\begin{aligned}
S_3(z) &= A(z) + 1,\\
A(z) &= z^2 \cdot B(z) \cdot D(z),\\
B(z) &= 1 + z^2 \cdot B(z) \cdot A(z) + z^2 \cdot B(z),\\
D(z) &= 1 + A(z).
\end{aligned}$$

- generating function $S_5(z)$:

$$\begin{aligned}
S_5(z) &= A(z) + 1,\\
A(z) &= z^2 \cdot B(z) \cdot D(z),\\
B(z) &= 1 + z^2 \cdot B(z) \cdot A(z),\\
D(z) &= 1 + A(z).
\end{aligned}$$

After solving these systems for $S_i(z)$, $1 \leq i \leq 5$, we can use Darboux's theorem [KW89] to determine precise asymptotics for the nth coefficients ($n \to \infty$) of the five ordinary generating functions $S_i(z)$, $1 \leq i \leq 5$. By choosing $m = 0$[10] for the application of Darboux's theorem and afterwards computing series expansions of the resulting asymptotics about $n \to \infty$, we finally obtain the desired results. □

Remark 3.1 There is an alternative approach to find the generating functions $S_i(z)$, $1 \leq i \leq 5$: From our previous discussion, it should be obvious that a type i shape, $2 \leq i \leq 5$, results from a type 1 shape by deleting symbols according to the higher level of abstraction. This is in line with our grammars, where $\mathcal{G}_i$, $2 \leq i \leq 5$, can be (re)constructed from $\mathcal{G}_1$ by deleting production rules and nonterminal symbols. As a consequence, by making use of a multivariate generating function $S(u, v, w, \dots)$, which marks the different nonterminals on the right-hand side of different production rules of $\mathcal{G}_1$ by different variables, it would have been possible to simulate the construction of grammars $\mathcal{G}_i$, $2 \leq i \leq 5$, by appropriate substitutions for the variables of $S(u, v, w, \dots)$. Assume for example that variable u (v, w resp.) has been used to uniquely mark symbol _ within $S_1 \rightarrow$ _ ($C \rightarrow$ _, $D \rightarrow$ _ resp.). Then, setting $(u, v, w) = (1, 0, 0)$ (instead of (z, z, z) for $S_1(z)$) would give us S_2. This explains why all the asymptotics given in Table 1 are of the same pattern having varying constants only; being able to trace back all the different shape levels to essentially the same generating function implies an algebraic transformation of variable z when changing from one level to the next. In our case, the transformation falls into the supercritical case of singularity analysis (see [FS09] for details), in which the singular type for the function (and therefore the design of the asymptotic) remains unchanged.

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. Therefore, we want to compare the number of secondary structures of size n – also termed type 0 shapes of size n in the sequel – to the number of different type i shapes that are morphic images of those secondary structures, for every $i \in \{1, \dots, 5\}$. More precisely, for every $i \in \{1, \dots, 5\}$, we want to compute the number of different type i shapes sh for which there exists a secondary structure s of size n such that s is mapped to sh when applying shape abstraction of level i.

Besides assuming the unrealistic minimal number of 1 for the length of a hairpin loop and a minimum number of 1 base pair in ladders, we will compute the desired results also under the assumption of a minimum number of 3 unpaired bases in hairpin loops and under the assumption that no isolated base pairs, i.e. no ladders consisting of less than 2 base pairs, can occur (see e.g. [GVR04]).

It should be obvious that by not allowing hairpin loops of length less than 3 or by avoiding ladders of length less than 2, the number of feasible secondary structures is reduced significantly. The precise quantitative effect of these additional restrictions is captured by the following result:

[10] In the considered version of Darboux's theorem as given in [KW89], the variable m is used to choose the number of terms for the computed asymptotic. In fact, by choosing $m = 0$, the resulting asymptotic consists of the leading term only.

	Type	Number of different shapes for all secondary structures of size n
$\text{minL}_{\text{ladder}} = 1$ and $\text{minL}_{\text{hairpin}} = 1$	0	$2.61803^n \cdot 1.10437 \cdot n^{-3/2}$
	1	$2.09188^n \cdot 1.50017 \cdot n^{-3/2}$
	2	$1.84277^n \cdot 1.65267 \cdot n^{-3/2}$
	3	$1.66034^n \cdot 1.71055 \cdot n^{-3/2}$
	4	$1.60804^n \cdot 1.79677 \cdot n^{-3/2}$
	5	$1.51243^n \cdot 1.84657 \cdot n^{-3/2}$
$\text{minL}_{\text{ladder}} = 1$ and $\text{minL}_{\text{hairpin}} = 3$	0	$2.28879^n \cdot 0.71312 \cdot n^{-3/2}$
	1	$1.80776^n \cdot 1.27613 \cdot n^{-3/2}$
	2	$1.65404^n \cdot 1.50643 \cdot n^{-3/2}$
	3	$1.4616^n \cdot 1.85429 \cdot n^{-3/2}$
	4	$1.42194^n \cdot 2.04493 \cdot n^{-3/2}$
	5	$1.32218^n \cdot 2.44251 \cdot n^{-3/2}$
$\text{minL}_{\text{ladder}} = 2$ and $\text{minL}_{\text{hairpin}} = 1$	0	$1.96798^n \cdot 2.1614 \cdot n^{-3/2}$
	1	$1.56947^n \cdot 3.4426 \cdot n^{-3/2}$
	2	$1.43537^n \cdot 3.88212 \cdot n^{-3/2}$
	3	$1.33966^n \cdot 4.03737 \cdot n^{-3/2}$
	4	$1.32321^n \cdot 4.17456 \cdot n^{-3/2}$
	5	$1.26585^n \cdot 4.37739 \cdot n^{-3/2}$
$\text{minL}_{\text{ladder}} = 2$ and $\text{minL}_{\text{hairpin}} = 3$	0	$1.84892^n \cdot 1.48483 \cdot n^{-3/2}$
	1	$1.47667^n \cdot 3.04214 \cdot n^{-3/2}$
	2	$1.37736^n \cdot 3.61323 \cdot n^{-3/2}$
	3	$1.27614^n \cdot 4.19348 \cdot n^{-3/2}$
	4	$1.26197^n \cdot 4.42176 \cdot n^{-3/2}$
	5	$1.20259^n \cdot 5.12777 \cdot n^{-3/2}$

Table 2: Asymptotics for the numbers $s_0(n, \text{minL}_{\text{ladder}}, \text{minL}_{\text{hairpin}})$ of all secondary structures (type 0 shapes) of size n, as well as for the numbers $m_i(n, \text{minL}_{\text{ladder}}, \text{minL}_{\text{hairpin}})$ of different type i shapes, $1 \leq i \leq 5$, that are morphic images of secondary structures of size n, assuming a minimum hairpin length $\text{minL}_{\text{hairpin}} \in \{1, 3\}$ and a minimum ladder length $\text{minL}_{\text{ladder}} \in \{1, 2\}$.

Theorem 4.1 *Under the assumption of each possible combination of a minimum hairpin loop length* $\text{minL}_{\text{hairpin}} \in \{1, 3\}$ *and a minimum helix length* $\text{minL}_{\text{ladder}} \in \{1, 2\}$*, the numbers* $m_i(n, \text{minL}_{\text{ladder}}, \text{minL}_{\text{hairpin}})$ *of different type i shapes,* $1 \leq i \leq 5$*, that are morphic images of secondary structures of size n,* $n \to \infty$*, are asymptotically given in Table 2.*

Note that for type 1 and type 5 shapes, the corresponding results under the assumption of a minimum hairpin length $\text{minL}_{\text{hairpin}} = 3$ and a minimum ladder length $\text{minL}_{\text{ladder}} = 1$ have already been determined by Lorenz et. al [LPC08].

Proof: For the sake of simplicity, let us assume that $\text{minL}_{\text{ladder}} = 1$ and $\text{minL}_{\text{hairpin}} = 1$.
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 morphic images of secondary structures of any length, $1 \leq i \leq 5$. Thus, our goal is to compute an asymptotical representation of

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

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 morphic images of secondary structures of size n is equal to the number of different type i shapes that are morphic images of secondary structures of size *at most* n, for each $i \in \{1, \ldots, 5\}$. In fact, for each length $n \geq 1$,

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

$1 \leq i \leq 5$, holds, which results from the observation that every secondary structure can be prolongated to an arbitrary size without changing its image under h_i, $1 \leq i \leq 5$, e.g. by inserting a run of dots in the dot-bracket representation. For details, see [LPC08] and the discussion below.

Using this observation, it becomes easy to construct the ordinary generating function

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

of the counting sequence $(m_{1_n})_{n \geq 0}$. Here, variable z keeps track of the maximum size of secondary structures $s \in \mathcal{S}_s$, while the coefficient at z^n provides the number of different type 1 shapes that result from these secondary structures. By modifying system (1), we construct a system of equations which can be solved for $M_1(z)$ to get a closed form of the generating function in question. 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 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 need sh to make a contribution of $z^{(|sh|+p_0)+p} = z^{(j+p_0)+p}$ for each $p \geq 0$, where $p_0 \geq 0$ is the number of additional symbols needed to construct a secondary structure s of minimum size with $h_1(s) = sh$. For this reason, we adapt the size of shapes by changing the contribution of the right-hand side of several equations in system (1) to their generating function counterparts by multiplying appropriate factors z to certain summands.

To find out which of the summands must be adapted in which way, consider for example the morphic image of the secondary structure *sec* given in Example 2.1 and the corresponding type 1 shape sh presented in Example 2.2. As we already mentioned, by using the morphism h_1, this shape sh is obtained from any secondary structure s' that has

- the same number of unpaired regions,
- the same number of paired regions and the
- same order of adjacent and nested substructures as the secondary structure *sec*.

The only differences are the lengths of the unpaired and paired regions. Hence, for a large value of n, there are plenty of different secondary structures s' of size at most n having these properties and being mapped to shape sh. Therefore, we start by considering a secondary structure s with minimum size among all these secondary structures; shape sh will then be made to contribute to any size $n \geq |s|$, as we may stretch s by inserting symbols **.** without changing its image with respect to h_1.

Obviously, this minimum secondary structure s could easily be obtained from the shape sh as follows: First, substitute each underscore with a dot, each opening squared bracket **[** with and opening bracket **(** and each closing squared bracket **]** with a closing bracket **)**. Afterwards, the desired secondary structure obviously consists of exactly $|sh|$ symbols. But we must recall that hairpin loops are not recorded explicitly in type 1 shapes, i.e. are not given as an underscore representing an unpaired region. For this reason, we have to additionally add a dot for each hairpin loop during the construction of the minimum secondary structure s, as each hairpin loop must consist of at least one unpaired nucleotide. Assuming that the shape sh contains p_0 subwords **[ ]**, then the constructed structure s has length $|s| = |sh| + p_0$. Thus, this minimum secondary structure s would make a contribution to the coefficient at $z^{|s|} = z^{|sh|+p_0}$ to a generating function in which z marks size.

Furthermore, as the type 1 shape sh is generated by the morphism h_1 for any secondary structure s' which differs only in the lengths of the unpaired and paired regions from the minimum secondary structure s, the shape sh must make a contribution to the coefficient at $z^{|sh|+p_0+p}$ of generating function $M_1(z)$ for each $p \geq 0$. In fact, if the underlying context-free grammar $\mathcal{G}_1$ generates a single type 1 shape sh, then sh must imply a term

$$\sum_{p \geq 0} z^{|sh|+p_0+p} = \left(\sum_{p \geq 0} z^p \right) \cdot z^{|sh|+p_0} = \left(\sum_{p \geq 0} z^p \right) \cdot z^{|s|} = \frac{1}{1-z} \cdot z^{|s|}$$

to generating function $M_1(z)$.

Due to these observations, it is easy to see that we first have to multiply a factor z *inserting* a single-stranded region of length 1 to each summand on the right-hand sides of system (1) that correspond to a rule of $\mathcal{G}_1$ that generates a pair of brackets[11] **[ ]** representing the type 1 abstraction of a hairpin loop. By multiplying these factors, we ensure that the considered shape sh contributes to the coefficient at $z^{|sh|+p_0} = z^{|s|}$. Second,

[11] Within our grammar, the rule $B \to \varepsilon$ generates from a sentential form $\ldots [B] \ldots$ such a pair of brackets and therefore has to be weighted by a factor z.

we obviously have to multiply the right-hand side of the first equation of the resulting modified system (1) by the factor $\frac{1}{1-z}$ representing an arbitrary blow-up of size. The final system of equations is then given as follows:

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

It has to be solved for $M_1(z)$ to obtain a closed form of the desired generating function $M_1(z)$.
Now, after the previous discussion, it should be clear that when we are assuming variable minimum lengths $\mathrm{minL}_{\mathrm{ladder}}$ and $\mathrm{minL}_{\mathrm{hairpin}}$ for helical regions and hairpin loops, respectively, we have to consider the following system, which can immediately be obtained from the previously presented one:

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

In the same way, we can easily construct the corresponding four systems of equations for the remaining four different shape abstraction levels $i \in \{2, \ldots, 5\}$. These four systems are given as follows:

- Type 2 shapes:

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

- Type 3 shapes:

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

- Type 4 shapes:

$$\begin{aligned}
M_4(z) &= \frac{1}{1-z} \cdot (A(z) + z \cdot 1), \\
A(z) &= z^{2 \cdot \mathrm{minL}_{\mathrm{ladder}}} \cdot B(z) \cdot D(z), \\
B(z) &= z^{\mathrm{minL}_{\mathrm{hairpin}}} \cdot 1 + z^{2 \cdot \mathrm{minL}_{\mathrm{ladder}}} \cdot B(z) \cdot A(z) + z \cdot z^{2 \cdot \mathrm{minL}_{\mathrm{ladder}}} \cdot B(z) \cdot z, \\
D(z) &= 1 + A(z).
\end{aligned}$$

- Type 5 shapes:

$$\begin{aligned}
M_5(z) &= \frac{1}{1-z} \cdot (A(z) + z \cdot 1), \\
A(z) &= z^{2 \cdot \mathrm{minL}_{\mathrm{ladder}}} \cdot B(z) \cdot D(z), \\
B(z) &= z^{\mathrm{minL}_{\mathrm{hairpin}}} \cdot 1 + z^{2 \cdot \mathrm{minL}_{\mathrm{ladder}}} \cdot B(z) \cdot A(z), \\
D(z) &= 1 + A(z).
\end{aligned}$$

Note that now, the cases of type 3 and type 4 shapes differ, since the structural identical shapes of both abstraction levels possess different preimages.
Now, for each $i \in \{1, \ldots, 5\}$, we can solve the respective system for type i shapes for the variable $M_i(z)$. This way, we obtain the generating functions $M_i(z, \mathrm{minL}_{\mathrm{ladder}}, \mathrm{minL}_{\mathrm{hairpin}})$, $1 \le i \le 5$. Applying Darboux's theorem (with the choice $m = 0$) and computing series expansions of the resulting asymptotics about $n \to \infty$ afterwards, we obtain the asymptotics for $m_i(n, \mathrm{minL}_{\mathrm{ladder}}, \mathrm{minL}_{\mathrm{hairpin}})$, $1 \le i \le 5$, as given in the theorem. □

Remark 4.1 Note that building on Remark 3.1, the inflation of shapes in order to recover its structural parents again could have been traced back to appropriate substitutions for a multivariate generating function derived from $\mathcal{G}_1$. Like before, this would yield the supercritical case of singularity analysis explaining the constant pattern of all the asymptotics given in Table 2.

To be able to quantify the reduction of the search space when using the concept of abstract shapes, we have to consider the following result:

Theorem 4.2 *Under the assumption of each possible combination of a minimum hairpin loop length* $\mathrm{minL}_{\mathrm{hairpin}} \in \{1, 3\}$ *and a minimum helix length* $\mathrm{minL}_{\mathrm{ladder}} \in \{1, 2\}$*, the resulting asymptotical numbers* $s_0(n, \mathrm{minL}_{\mathrm{ladder}}, \mathrm{minL}_{\mathrm{hairpin}})$ *of secondary structures (type 0 shapes) of size n, $n \to \infty$, are those given in Table 2.*

This theorem can easily be proven using the same techniques as before. Furthermore, most (if not all) of the asymptotics presented can be found somewhere in the literature. Accordingly, we omit a proof.

5 Discussion

Now, we are finally able to discuss and compare the derived results (where primary structure is not considered). Later on, we will discuss why the derivation of similar results under the assumption of a more realistic model (which takes primary structure into account) remains on open challenge.

5.1 Abstract Shapes – Combinatorial Point of View

Considering Table 2, it is easy to observe that all asymptotics grow exponentially in n, where the base of the exponential growth of the number of secondary structures of size n is significantly larger than that for the different types of shapes – as expected.
Next, we want to compare the number of secondary structures of size n to the number of different type i shapes that are morphic images of those secondary structures, for every $i \in \{1, \ldots, 5\}$. Therefore, for each of our four possible choices of $\mathrm{minL}_{\mathrm{ladder}}$ and $\mathrm{minL}_{\mathrm{hairpin}}$, we consider a plot containing all the resulting asymptotics for $s_0(n, \mathrm{minL}_{\mathrm{ladder}}, \mathrm{minL}_{\mathrm{hairpin}})$ and $m_i(n, \mathrm{minL}_{\mathrm{ladder}}, \mathrm{minL}_{\mathrm{hairpin}})$, $1 \le i \le 5$. As all these asymptotics grow exponentially in n, it is appropriate to plot them using a logarithmic scale. The resulting plots are shown in Figure 3.
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. Moreover, there is the expected order of graphs, i.e. the different shape levels are in fact ordered by their degree of abstraction. Furthermore, the consideration of the logarithmic plot for the choice of $\mathrm{minL}_{\mathrm{ladder}} = \mathrm{minL}_{\mathrm{hairpin}} = 1$ shown in the upper left corner of Figure 3 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. Additionally, abstracting from single-stranded regions in multiloops and exterior loops (difference from type 1 to type 2 shape abstractions) yields only a smaller but still comparatively large additional reduction. However, abstracting from bulges interrupting ladders and single-stranded regions in interior loops (difference from type 2 to type 3 shape abstractions) results in a search space reduction of similar size as the previous one. The definitely smallest step is made when reducing the search space by only partially abstracting from nesting of helices, more precisely by combining nested helices only if the nesting is due to an interruption by a single bulge (difference from type 3 to type 4 shape abstractions). Also a significant but comparatively small final reduction of the search space is reached by additionally abstracting from nesting of helices caused by interior loop interruptions and hence combining all nested helices (difference from type 4 to type 5 shape abstractions). As expected, the other three plots shown in Figure 3 lead to similar conclusions.

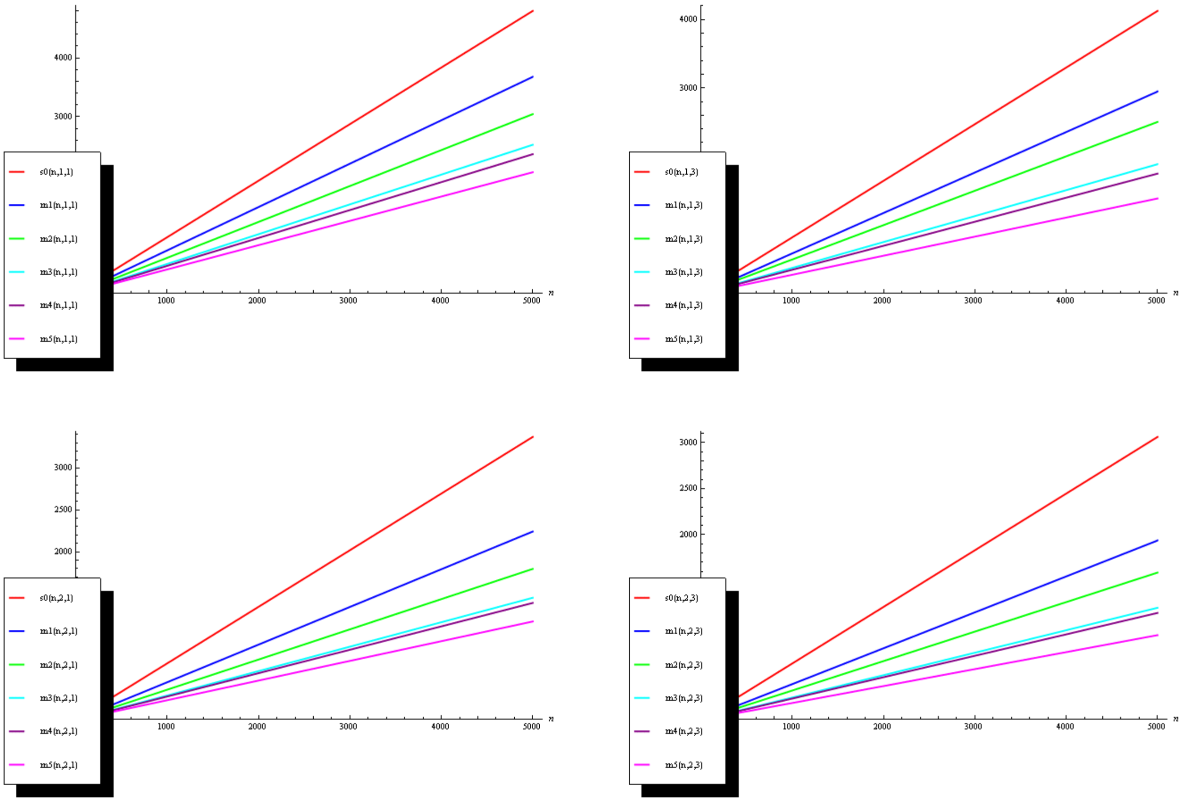

Figure 3: Asymptotics of the number $s_0(n, \mathrm{minL}_{\mathrm{ladder}}, \mathrm{minL}_{\mathrm{hairpin}})$ of secondary structures of size n and of the numbers $m_i(n, \mathrm{minL}_{\mathrm{ladder}}, \mathrm{minL}_{\mathrm{hairpin}})$ of different type i shapes, $1 \leq i \leq 5$, that are morphic images of secondary structures of size n, for each possible combination of $\mathrm{minL}_{\mathrm{hairpin}} \in \{1, 3\}$ and $\mathrm{minL}_{\mathrm{ladder}} \in \{1, 2\}$, respectively, logarithmically scaled.

Furthermore, considering all four plots shown in Figure 3, we observe that the reduction of the search space is maximal for $\mathrm{minL}_{\mathrm{ladder}} = 2$ and $\mathrm{minL}_{\mathrm{hairpin}} = 3$. Consequently, the maximum reduction of the search space is reached for the most realistic secondary structure model.

5.2 Taking Primary Structure into Account

In the previous section, we have considered all secondary structures of size n, i.e. all possible two-dimensional foldings, under a pure combinatorial model. That way, primary structure has completely been disregarded. A more realistic consideration should start with a random primary structure s of size n, considering only those secondary structures that are compatible with s according to base pairing. It would then be most interesting to apply shape abstractions to those random foldings and to determine their quantitative behavior. Accordingly, Giegerich and co-workers introduced the terms (concrete) folding space and (abstract) shape space which can be defined as follows [GVR04]:

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 $h_i : \mathcal{S}_s \rightarrow \mathcal{S}_i$ is the morphism mapping secondary structures to type i shapes, $1 \leq i \leq 5$.*

A well-known procedure to keep track of primary structure in enumeration of secondary structures is to make use of the concept of *stickiness*. Assuming that a random primary structure is generated according to a Bernoulli experiment, where symbol X shows up with probability p_{X}, $\mathrm{X} \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{U}\}$, the probability p that two random nucleotides may have a hydrogen bond is given by $2(p_{\mathrm{A}}p_{\mathrm{U}} + p_{\mathrm{C}}p_{\mathrm{G}})$ (Watson-Crick pairings only). This probability is usually called stickiness and of course can easily be adapted to further notions of complementarity. The interesting point about this model is its easy translation into enumeration procedures. Multiplying each terminal representing a paired nucleotide within our grammars by $\sqrt{p}z$ instead of z introduces generating functions whose coefficients are the expected number of secondary structures compatible with a random primary structure (see [Neb04] for details). This provides asymptotic results for the expected size of the folding space which we present in the appendix. From a formal point of view, the same could be done for shapes, weighting each [and] by $\sqrt{p}$. However, this procedure makes no sense because of the following reasons:

When weighting a pair of corresponding brackets [ ] within a shape by p, this – explained intuitively – corresponds to a statistical test which asks for the complementarity of two random nucleotides. However, if this test fails this does not imply that the shape considered may not be compatible with the primary structure at hand. It only says that we might have to shift the location of one or both brackets along the sequence of bases. Consider as an example the primary structure CCCCCGGGGG. Of course, we cannot locate a pair of brackets [ ] of a shape within the block of Cs or the block of Gs, but obviously it is possible to assign a position to each bracket such that the corresponding symbols *match*.

Thus, to consider primary structure in context of shape abstractions, we would have to take care of all alternative positionings of the bracket symbols of the shape along the primary structure (recall that shapes abstract from the concrete position of helices). The number of these obviously grows when the sequence gets large compared to the size of the shape and we have to expect that finally, the probability for a shape to be compatible to a random sequence converges to 1. At the moment, it is not clear what the right analytic tool should be for such a study (generating functionology seems unable to capture such a "shifting" constraint). However, we want to stress that a very interesting challenge is discussed here. Besides all progress made by this and further studies, determining the expected size of the shape space is **the problem** relevant for algorithms such as probabilistic shape analysis [VGR06], where for $\mathrm{card}(P_i(s))$ the (unknown) number of type i shapes for a primary sequence s, a runtime of order $\mathcal{O}\left(\mathrm{card}(P_i(s)) \cdot |s|^3\right)$ shows up. Thus, a solution to this problem can be used to determine upper bounds to the size of sequences one can analyze in practice.

Acknowledgements

The authors wish to thank two anonymous reviewers for their careful and helpful remarks and suggestions made for a previous version of this article.

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.

[DCL04] Y. Ding, C. Chan, and C. E. Lawrence. Sfold web server for statistical folding and rational design of nucleic acids. *Nucleic Acids Research*, 32:W135–W141, 2004.

[DL03] Ye Ding and Charles E. Lawrence. A statistical sampling algorithm for RNA secondary structure prediction. *Nucleic Acids Research*, 31(24):7280–7301, 2003.

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

[FS09] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics.* Camebridge University Press, January 2009.

[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: Pseudoknots 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.

[JRG08] Stefan Janssen, Jens Reeder, and Robert Giegerich. Shape based indexing for faster search of RNA family databases. *BMC Bioinformatics*, 9(131), 2008.

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

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

[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.

[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.

[SN08] Anika Scheid and Markus E. Nebel. On abstract shapes of RNA. Technical report, Technische Universität Kaiserslautern, 4 2008.

[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.

[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.

[WFHS99] S. Wuchty, W. Fontana, I. Hofacker, and P. Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49:145–165, 1999.

[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.

[Zuk89] M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244:48–52, 1989.

[Zuk03] M. Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.*, 31(13):3406–3415, 2003.

	Only Watson-Crick Base Pairs $(p = \frac{1}{4})$ $(p = \frac{13}{50})$	Watson-Crick and Wobble GU Pairs $(p = \frac{3}{8})$ $(p = \frac{19}{50})$
$\mathrm{minL}_{\mathrm{ladder}} = 1$ and $\mathrm{minL}_{\mathrm{hairpin}} = 1$	$1.86603^n \cdot 1.95947 \cdot n^{-3/2}$ *$1.88163^n \cdot 1.92488 \cdot n^{-3/2}$*	$2.04101^n \cdot 1.6374 \cdot n^{-3/2}$ *$2.04727^n \cdot 1.6281 \cdot n^{-3/2}$*
$\mathrm{minL}_{\mathrm{ladder}} = 1$ and $\mathrm{minL}_{\mathrm{hairpin}} = 3$	$1.72139^n \cdot 1.54195 \cdot n^{-3/2}$ *$1.73334^n \cdot 1.50771 \cdot n^{-3/2}$*	$1.85479^n \cdot 1.22479 \cdot n^{-3/2}$ *$1.85954^n \cdot 1.21569 \cdot n^{-3/2}$*
$\mathrm{minL}_{\mathrm{ladder}} = 2$ and $\mathrm{minL}_{\mathrm{hairpin}} = 1$	$1.36247^n \cdot 5.8205 \cdot n^{-3/2}$ *$1.37366^n \cdot 5.62474 \cdot n^{-3/2}$*	$1.49265^n \cdot 4.16417 \cdot n^{-3/2}$ *$1.49747^n \cdot 4.12169 \cdot n^{-3/2}$*
$\mathrm{minL}_{\mathrm{ladder}} = 2$ and $\mathrm{minL}_{\mathrm{hairpin}} = 3$	$1.33089^n \cdot 5.11834 \cdot n^{-3/2}$ *$1.34064^n \cdot 4.92109 \cdot n^{-3/2}$*	$1.44358^n \cdot 3.45373 \cdot n^{-3/2}$ *$1.44773^n \cdot 3.41124 \cdot n^{-3/2}$*

Table 3: Asymptotics for the expected sizes of the folding space for a random primary structure s of size n assuming a uniform distribution of the bases A, C, G, U (results in roman) or the skewed distribution $p_{\mathrm{A}} = p_{\mathrm{U}} = {}^2/_{10}$, $p_{\mathrm{C}} = p_{\mathrm{G}} = {}^3/_{10}$ (results in italics), a minimum hairpin length $\mathrm{minL}_{\mathrm{hairpin}} \in \{1, 3\}$ and a minimum ladder length $\mathrm{minL}_{\mathrm{ladder}} \in \{1, 2\}$. Stickiness $p = {}^1/_4$ resp. $p = {}^{13}/_{50}$ corresponds to Watson-Crick parings only, assuming a uniform resp. the skewed distribution. Allowing wobble GU implies $p = {}^3/_8$ resp. $p = {}^{19}/_{50}$.

6 Appendix

During our investigations, we have computed precise asymptotics for the size of the folding space $F(s)$ for different models of secondary structures. The models differ with respect to structural restrictions (minimal length of hairpin loops, isolated base pairs) and the complementary assumed (Watson-Crick pairings only, wobble GU pairs allowed), expecting a uniform distribution for the bases or a skewed one ($p_{\mathrm{A}} = p_{\mathrm{U}} = {}^2/_{10}$, $p_{\mathrm{C}} = p_{\mathrm{G}} = {}^3/_{10}$), according to the experiments performed in [GVR04, VGR06].

Even if they were of no use for our investigations related to abstract shapes due to the problems reported when analyzing shape spaces, we expect those results to be of use for the future and therefore decided to present them in this appendix without proof. Few of those results can already be found in the literature (see e.g. [Neb04]), but there does not exists such a complete presentation.

Theorem 6.1 *Considering a uniform distribution of the bases A, C, G and U resp. the skewed distribution $p_{\mathrm{A}} = p_{\mathrm{U}} = {}^2/_{10}$, $p_{\mathrm{C}} = p_{\mathrm{G}} = {}^3/_{10}$, regarding Watson-Crick pairings only or allowing wobble GU pairs and under the assumption of each possible combination of a minimum hairpin loop length* $\mathrm{minL}_{\mathrm{hairpin}} \in \{1, 3\}$, *and a minimum helix length* $\mathrm{minL}_{\mathrm{ladder}} \in \{1, 2\}$, *the asymptotic expected folding space sizes* $\mathrm{card}(F(s))$ *for a random primary structure s of size n, $n \to \infty$, are those given in Table 3 shown in roman resp. italics.*