Enumerating RNA Pseudoknots

Markus E. Nebel

University of Kaiserslautern

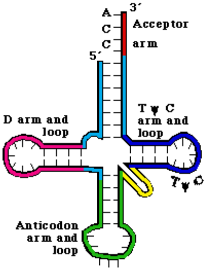
February 2012

Discrete Models of RNA polymers

Secondary structure: Base pairs *leaving the plane* are prohibited.

Example:

untwisting+planarity
$\rightarrow$



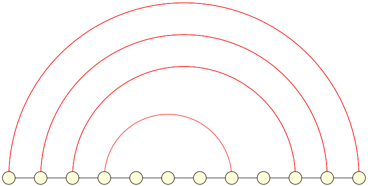
$(((((((\circ\circ\circ((((\circ\circ\circ\circ\circ\circ\circ))))\circ\circ(((((\circ\circ\circ\circ\circ\circ\circ))))))\circ\circ\circ\circ\circ\circ\circ\circ((((\circ\circ\circ\circ\circ\circ\circ))))))))))\circ\circ\circ$

Dot-bracket representation; obvious correspondence to tree structures.

Discrete Models of RNA polymers

Alternative representation: *Contact graph*

- Backbone (primary structure) linear chain of vertices (in natural order) each representing a nucleotide;
- hydrogen bonds are represented by arcs in the upper half-plane.

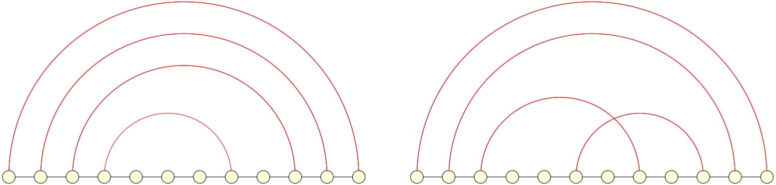


((((∘ ∘ ∘) ∘)))

Discrete Models of RNA polymers

Alternative representation: *Contact graph*

- Backbone (primary structure) linear chain of vertices (in natural order) each representing a nucleotide;
- hydrogen bonds are represented by arcs in the upper half-plane.



$(\; (\; (\; (\; \circ \; \circ \; \circ \;) \; \circ \;) \;) \;)$ $\qquad$ $(\; (\; [\; \circ \; \circ \; (\; \circ \;] \; \circ \;) \;) \;)$

If arcs do cross (right example) $\rightsquigarrow$ **pseudoknot** (otherwise secondary structure).

Algorithmic challenges

Determine the structure from known sequence of bases (primary structure)

- by minimizing free energy (most prominent),
- by stochastic approaches,
- ...

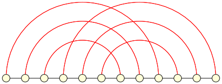
Forbidding pseudoknots this task is computationally feasible (running time in $\mathcal{O}(n^3)$), allowing pseudoknots leads to an $\mathcal{NP}$-complete problem (LYNGSØ AND PEDERSEN 2000) even for a rather simple model of free energy (nearest neighbor model).

Resort: Limit the types of legal pseudoknots.

Algorithmic challenges

Subclasses used for structure prediction:

- LYNGSØ & PEDERSEN: At most one *H*-type pseudoknot not embedded under any arc with nested secondary structures.

  H-type: 

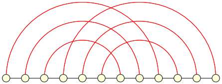
- CAO & CHEN: Any number of *H*-type pseudoknots and secondary structures recursively embedded with restrictions.
- REEDER & GIEGERICH: Any number of *H*-type pseudoknots and secondary structures recursively embedded with less restrictions.
- DIRKS & PIERCE: Any number of *H*-type pseudoknots and secondary structures recursively embedded without restrictions.
- AKUTSU & UEMURA: Any number of simple pseudoknots and secondary structures recursively embedded without restrictions.

Simple pseudoknot:

Algorithmic challenges

Subclasses used for structure prediction:

- LYNGSØ & PEDERSEN: At most one *H*-type pseudoknot not embedded under any arc with nested secondary structures.

  H-type: 

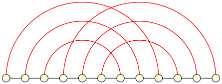
- CAO & CHEN: Any number of *H*-type pseudoknots and secondary structures recursively embedded with restrictions.
- REEDER & GIEGERICH: Any number of *H*-type pseudoknots and secondary structures recursively embedded with less restrictions.
- DIRKS & PIERCE: Any number of *H*-type pseudoknots and secondary structures recursively embedded without restrictions.
- AKUTSU & UEMURA: Any number of simple pseudoknots and secondary structures recursively embedded without restrictions.

Simple pseudoknot:

Algorithmic challenges

Subclasses used for structure prediction:

- LYNGSØ & PEDERSEN: At most one *H*-type pseudoknot not embedded under any arc with nested secondary structures.

H-type: 

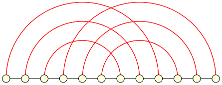
- CAO & CHEN: Any number of *H*-type pseudoknots and secondary structures recursively embedded with restrictions.
- REEDER & GIEGERICH: Any number of *H*-type pseudoknots and secondary structures recursively embedded with less restrictions.
- DIRKS & PIERCE: Any number of *H*-type pseudoknots and secondary structures recursively embedded without restrictions.
- AKUTSU & UEMURA: Any number of simple pseudoknots and secondary structures recursively embedded without restrictions.

Simple pseudoknot:

Algorithmic challenges

Subclasses used for structure prediction:

- LYNGSØ & PEDERSEN: At most one *H*-type pseudoknot not embedded under any arc with nested secondary structures.

  H-type: 

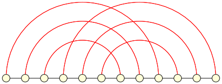
- CAO & CHEN: Any number of *H*-type pseudoknots and secondary structures recursively embedded with restrictions.
- REEDER & GIEGERICH: Any number of *H*-type pseudoknots and secondary structures recursively embedded with less restrictions.
- DIRKS & PIERCE: Any number of *H*-type pseudoknots and secondary structures recursively embedded without restrictions.
- AKUTSU & UEMURA: Any number of simple pseudoknots and secondary structures recursively embedded without restrictions.

Simple pseudoknot:

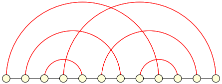
Algorithmic challenges

Subclasses used for structure prediction:

- Lyngsø & Pedersen: At most one *H*-type pseudoknot not embedded under any arc with nested secondary structures.

  H-type: 

- Cao & Chen: Any number of *H*-type pseudoknots and secondary structures recursively embedded with restrictions.
- Reeder & Giegerich: Any number of *H*-type pseudoknots and secondary structures recursively embedded with less restrictions.
- Dirks & Pierce: Any number of *H*-type pseudoknots and secondary structures recursively embedded without restrictions.
- Akutsu & Uemura: Any number of simple pseudoknots and secondary structures recursively embedded without restrictions.

  Simple pseudoknot: 

Algorithmic challenges

Subclasses used for structure prediction (cont.):

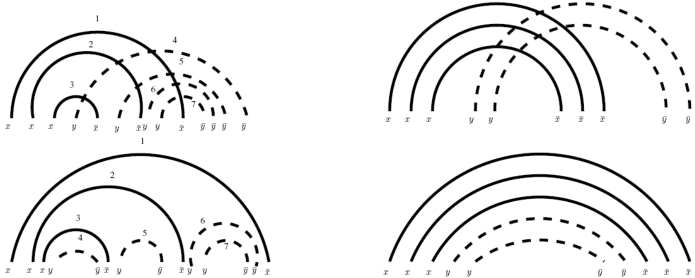
- REIDYS & … & NEBEL: So-called *shadows* of the pseudoknots have topological genus at most 1.
- RIVAS & EDDY: Generalized notion of *nesting* where a single arc may be replaced by a complete contact graph (instead of placing it between two vertices).

Those restriction lead to prediction algorithms with a running time between $\mathcal{O}(n^4)$ and $\mathcal{O}(n^6)$ making inputs of moderate size manageable.

Mathematical challenges

Question: Quantify the portion of all possible pseudoknots ($\sim \sqrt{2} \cdot 2^n \cdot \left(\frac{n}{e}\right)^n$) covered by the various subclasses.

Approach: (SAULE ET AL. 2010) bijective combinatorics



(a) Simple pseudoknot (b) H-type pseudoknot

Remark: In the before mentioned investigation and in all what follows, unpaired nucleotides (symbols $\circ$) are neglected!!

Mathematical challenges

SAULE ET AL. were not able to apply this approach to the RIVAS & EDDY class; the corresponding enumeration problem was again left open (for almost 12 years). Furthermore, the newly introduced REIDYS & NEBEL class asked for enumeration.

Our solution:

- Get unified descriptions of the different classes first,
- translate them into multiple context-free grammars,
- which are used for enumeration purposes (by means of generating functions).

Stay tuned for details!

Unified description of pseudoknot classes

Definition

For $\Sigma = \{[_i,]_i \mid i \in \mathbb{N}\}$ let $A, C \subset \Sigma^*$. We call $B \subset \Sigma^*$ resulting from A (resp. A and C) by

- *nesting* if $B = n'(A, C) := A \cup \{w_1 u w_2 \mid w_1 w_1 \in A, u \in C\}$, or $B = n(A) := n'(A, A)$, $B = n''(A) := A \cup \{w_1 u w_2 \mid w_1 w_2, u \in A, (w_1, w_2) = (w_1' [_i,]_j w_2') \Rightarrow i = j\}$.
- *stem extension* if $B = s(A) := A \cup \{w_1 [_i [_i w_2]_i]_i w_3 \mid w_1 [_i w_2]_i w_3 \in A$ and w_2 i-balanced$\}$.
- *knot extension* if $B = k(A) := A \cup \{w_1 [_j [_i]_j w_2]_i w_3 \mid w_1 [_i w_2]_i w_3 \in A, j \neq i$ and w_2 i-balanced$\} \cup \{w_1 [_i w_2 [_j]_i]_j w_3 \mid w_1 [_i w_2]_i w_3 \in A, j \neq i$ and w_2 i-balanced$\} \cup \{w_1 [_i [_j w_2]_i]_j w_3 \mid w_1 [_i w_2]_i w_3 \in A, j \neq i$ and w_2 i-balanced and j-balanced$\}$.

Unified description of pseudoknot classes

Definition

For $\Sigma = \{[_i,]_i \mid i \in \mathbb{N}\}$ let $A, C \subset \Sigma^*$. We call $B \subset \Sigma^*$ resulting from A (resp. A and C) by

- *nesting*: $B = n'(A, C)$, or $B = n(A) := n'(A, A)$, or $B = n''(A)$;
- *stem extension*: $B = s(A)$;
- *knot extension*: $B = k(A)$.

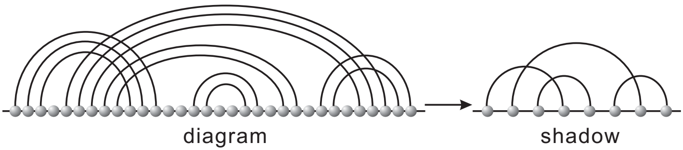
Unified description of pseudoknot classes

Example: For $\mathcal{D} := \{[_1]_1\}$ we find that $k(\mathcal{D})$ is $\mathcal{D}$ extended by the shadow of any H-type pseudoknot $[_1[_2]_1]_2$ (modulo re-indexing).

Shadow: The *shadow* of a contact graph (diagram) is obtained by

- removing all non-crossing arcs,
- collapsing all isolated vertices and
- replacing all remaining stacks (i.e., adjacent parallel arcs) by single arcs.

Example:



Unified description of pseudoknot classes

Lemma

For $\mathcal{S} := \mathcal{D}$ *extended by the shadows of simple pseudoknots and* $\mathcal{G} := \mathcal{D}$ *extended by the shadows of genus-1-pseudoknots we have*

$$
\begin{aligned}
PKF &= n^{\infty}(\mathcal{D}),\\
L\&P &= n'^{\infty}(s^{\infty}(k(\mathcal{D})), n^{\infty}(\mathcal{D})),\\
C\&C &= s^{\infty}(n''^{\infty}(k(\mathcal{D}))),\\
R\&G &= s^{\infty}(n^{\infty}(k(\mathcal{D}))),\\
D\&P &= (s \circ n)^{\infty}(k(\mathcal{D})),\\
A\&U &= (s \circ n)^{\infty}(\mathcal{S}),\\
R\&N &= (s \circ n)^{\infty}(\mathcal{G}),\\
R\&E &= (s \circ n \circ k)^{\infty}(\mathcal{D}).
\end{aligned}
$$

Unified description of pseudoknot classes

Corollary

- $PKF \subset C\&C \subset R\&G \subset D\&P \subset A\&U \subset R\&E$,
- $PKF \subset L\&P \subset R\&N \subset R\&E$,
- $R\&N \# A\&U$.

Proof: Immediate from definitions. □

Unified description of pseudoknot classes

Corollary

- $PKF \subset C\&C \subset R\&G \subset D\&P \subset A\&U \subset R\&E$,
- $PKF \subset L\&P \subset R\&N \subset R\&E$,
- $R\&N \# A\&U$.

Proof: Immediate from definitions. $\square$

A context-free grammar

- is a mechanism to generate sets of strings (language $\mathcal{L}$),
- which allows to derive a system of equations for the generating function $\sum_{w \in \mathcal{L}} z^{|w|}$.

Such a grammar is given by

- two disjoint alphabets I and T of intermediate and terminal symbols respectively, and
- a distinguished intermediate symbol S called axiom, and
- a set of rules/productions P specifying how intermediate symbols can be replaced by strings over $I \cup T$.

Enumeration by multiple context-free grammars

Basic concepts and definitions

A context-free grammar

- is a mechanism to generate sets of strings (language $\mathcal{L}$),
- which allows to derive a system of equations for the generating function $\sum_{w \in \mathcal{L}} z^{|w|}$.

Such a grammar is given by

- two disjoint alphabets I and T of intermediate and terminal symbols respectively, and
- a distinguished intermediate symbol S called axiom, and
- a set of rules/productions P specifying how intermediate symbols can be replaced by strings over $I \cup T$.

Basic concepts and definitions

Example

- CFG $G = (I, T, P, S) = \big(\{S\}, \{(,\ )\}, \{S \rightarrow (S)S, S \rightarrow \epsilon\}, S\big)$
- $S \Rightarrow (S)S \Rightarrow (S)(S)S \Rightarrow \underbrace{((S)S)(S)S}_{\text{sentential form}} \Rightarrow (())()$
- $S \rightarrow (S)S, S \rightarrow \epsilon \quad \equiv \quad S(z) = z^2 S(z)^2 + 1$

Bad news: Representation of pseudoknot classes cannot be generated (without bijection) by context-free grammar, i.e. use of CFGs leads to bijective combinatorics.

Basic concepts and definitions

Example

- CFG $G = (I, T, P, S) = \big(\{S\}, \{(,)\}, \{S \to (S)S, S \to \epsilon\}, S\big)$
- $S \Rightarrow (S)S \Rightarrow (S)(S)S \Rightarrow \underbrace{((S)S)(S)S}_{\text{sentential form}} \Rightarrow (())()$

- $S \to (S)S, S \to \epsilon \quad \Longrightarrow \quad S(z) = z^2 S(z)^2 + 1$

Bad news: Representation of pseudoknot classes cannot be generated (without bijection) by context-free grammar, i.e. use of CFGs leads to bijective combinatorics.

Basic concepts and definitions

Example

- CFG $G = (I, T, P, S) = \big(\{S\}, \{(,)\}, \{S \to (S)S, S \to \epsilon\}, S\big)$
- $S \Rightarrow (S)S \Rightarrow (S)(S)S \Rightarrow \underbrace{((S)S)(S)S}_{\text{sentential form}} \Rightarrow (())()$
- $S \to (S)S, S \to \epsilon \quad \equiv \quad S(z) = z^2 S(z)^2 + 1$

Bad news: Representation of pseudoknot classes cannot be generated (without bijection) by context-free grammar, i.e. use of CFGs leads to bijective combinatorics.

Basic concepts and definitions

Example

- CFG $G = (I, T, P, S) = \big(\{S\}, \{(,)\}, \{S \to (S)S, S \to \epsilon\}, S\big)$
- $S \Rightarrow (S)S \Rightarrow (S)(S)S \Rightarrow \underbrace{((S)S)(S)S}_{\text{sentential form}} \Rightarrow (())()$
- $S \to (S)S, S \to \epsilon \quad \equiv \quad S(z) = z^2 S(z)^2 + 1$

Bad news: Representation of pseudoknot classes cannot be generated (without bijection) by context-free grammar, i.e. use of CFGs leads to bijective combinatorics.

Basic concepts and definitions

Example

- CFG $G = (I, T, P, S) = \big(\{S\}, \{(,)\}, \{S \to (S)S, S \to \epsilon\}, S\big)$
- $S \Rightarrow (S)S \Rightarrow (S)(S)S \Rightarrow \underbrace{((S)S)(S)S}_{\text{sentential form}} \Rightarrow (())()$
- $S \to (S)S, S \to \epsilon \quad \equiv \quad S(z) = z^2 S(z)^2 + 1$

Bad news: Representation of pseudoknot classes cannot be generated (without bijection) by context-free grammar, i.e. use of CFGs leads to bijective combinatorics.

Basic concepts and definitions

Example

- CFG $G = (I, T, P, S) = \big(\{S\}, \{(,)\}, \{S \to (S)S, S \to \epsilon\}, S\big)$
- $S \Rightarrow (S)S \Rightarrow (S)(S)S \Rightarrow \underbrace{((S)S)(S)S}_{\text{sentential form}} \Rightarrow (())()$
- $S \to (S)S, S \to \epsilon \quad \equiv \quad S(z) = z^2 S(z)^2 + 1$

Bad news: Representation of pseudoknot classes cannot be generated (without bijection) by context-free grammar, i.e. use of CFGs leads to bijective combinatorics.

Basic concepts and definitions

Multiple context-free grammars (MCFGs): Intermediate symbols now have multiple components

- which are replaced in a coupled way,
- while appearing detached in the sentential forms.

Example: $\ldots\alpha A_1\beta A_2\gamma\ldots \Rightarrow \alpha[\beta]\gamma$ for $A_1, A_2 \to [,] \in P$.

Goal: Find unambiguous (one tree per word) MCFGs which generate the (language of the) R&E class (without bijection).

Key ideas:

- Represent pair of corresponding brackets $[\cdots]$ by two dimensional intermediate $A_1 \cdots A_2$ (plus rule $A_1, A_2 \to [,]$).
- To enforce unambiguity, different intermediates are used depending on which operation introduces the brackets.
- Find set of rules to generate the different ways *gapped structures* may be combined.

Basic concepts and definitions

Multiple context-free grammars (MCFGs): Intermediate symbols now have multiple components

- which are replaced in a coupled way,
- while appearing detached in the sentential forms.

Example: $\dots\alpha A_1 \beta A_2 \gamma \dots \Rightarrow \alpha[\beta]\gamma$ for $A_1, A_2 \rightarrow [,] \in P$.

Goal: Find unambiguous (one tree per word) MCFGs which generate the (language of the) R&E class (without bijection).

Key ideas:

- Represent pair of corresponding brackets $[\cdots]$ by two dimensional intermediate $A_1 \cdots A_2$ (plus rule $A_1, A_2 \rightarrow [,]$).
- To enforce unambiguity, different intermediates are used depending on which operation introduces the brackets.
- Find set of rules to generate the different ways *gapped structures* may be combined.

Basic concepts and definitions

Multiple context-free grammars (MCFGs): Intermediate symbols now have multiple components

- which are replaced in a coupled way,
- while appearing detached in the sentential forms.

Example: $\ldots\alpha A_1\beta A_2\gamma\ldots \Rightarrow \alpha[\beta]\gamma$ for $A_1,A_2 \to [,] \in P$.

Goal: Find unambiguous (one tree per word) MCFGs which generate the (language of the) R&E class (without bijection).

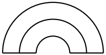
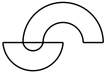
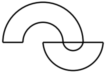
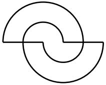
Key ideas:

- Represent pair of corresponding brackets $[\cdots]$ by two dimensional intermediate $A_1 \cdots A_2$ (plus rule $A_1,A_2 \to [,]$).
- To enforce unambiguity, different intermediates are used depending on which operation introduces the brackets.
- Find set of rules to generate the different ways *gapped structures* may be combined.

Grammar for the R&E class

Modulo ambiguity, the following grammar allows to *simulate* the recursive decomposition of R&E:

$$
\begin{array}{rcl}
S & \rightarrow & T_1\, T_2 \\
T_1, T_2 & \rightarrow & [S,]S \\
 & & T_1\, T_1', T_2'\, T_2 \\
 & & T_1\, T_1'\, T_2, T_2' \\
 & & T_1, T_1'\, T_2\, T_2' \\
 & & T_1\, T_1', T_2\, T_2'
\end{array}
$$

T_1', T_2' a distinguished copy of T_1, T_2.

Note that for enumeration purposes we do not need to distinguish between different kinds of brackets.

Grammar for the R&E class

$$
\begin{aligned}
S &\rightarrow \epsilon + \circ S + [S]S + K_1[SK_2]S,\\
K_1, K_2 &\rightarrow K_1 I_1, I_2 K_2 + I_1 K_1 I_2, K_2 + D_1, I_1 D_2 I_2 + L_1 K_1, L_2 K_2 + [S,]S,\\
I_1, I_2 &\rightarrow I_1 K_1 I_2, K_2 + D_1, I_1 D_2 I_2 + L_1 K_1, L_2 K_2 + [S,]S,\\
D_1, D_2 &\rightarrow K_1 I_1, I_2 K_2 + D_1, I_1 D_2 I_2 + L_1 K_1, L_2 K_2 + [S,]S,\\
L_1, L_2 &\rightarrow K_1 I_1, I_2 K_2 + I_1 K_1 I_2, K_2 + D_1, I_1 D_2 I_2 + [S,]S,
\end{aligned}
$$

automatically derived from above algebraic description of the pseudoknot class.

A general enumeration lemma

What do we gain? A lot when considering the ring of formal power series.

Advantage: Going forth and back from grammar to series to grammar allows to

- *reorder* the terminal symbols in the words generated (by commutativity of the series) without affecting their numbers (similar to bijective combinatorics) and the size of the language generated,
- *find* a plain context-free grammar which generates the corresponding (reordered) language of exactly the same sizes,
- use well-known enumeration techniques for context-free languages.

A general enumeration lemma

What do we gain? A lot when considering the ring of formal power series.

Advantage: Going forth and back from grammar to series to grammar allows to

- *reorder* the terminal symbols in the words generated (by commutativity of the series) without affecting their numbers (similar to bijective combinatorics) and the size of the language generated,
- *find* a plain context-free grammar which generates the corresponding (reordered) language of exactly the same sizes,
- use well-known enumeration techniques for context-free languages.

A general enumeration lemma

What do we gain? A lot when considering the ring of formal power series.

Advantage: Going forth and back from grammar to series to grammar allows to

- *reorder* the terminal symbols in the words generated (by commutativity of the series) without affecting their numbers (similar to bijective combinatorics) and the size of the language generated,
- *find* a plain context-free grammar which generates the corresponding (reordered) language of exactly the same sizes,
- use well-known enumeration techniques for context-free languages.

This heuristic explanation opens the way to prove the following lemma.

A general enumeration lemma

Lemma

Let $G = (I, d, T, P, \vec{X}^{<1>})$, $I = \{\vec{X}^{<1>}, \ldots, \vec{X}^{<k>}\}$, *an unambiguous* MCFG *without* ε*-rules*[1] *and* $\mathcal{SE}$ *the system of equations where for each* $\vec{X}^{<i>} \in I$ *the following variable and corresponding equation is introduced:*

$$X^{(i)} = \sum_{\vec{\alpha}:\vec{X}^{<i>} \to \vec{\alpha} \in P} \prod_{1 \leqslant k \leqslant d(X^{<i>})} h(\alpha_k),$$

where h *is the substitution that maps* $a \in T$ *to variable* z, *the first component* $X_1^{<j>}$ *of any intermediate symbol to* $X^{(j)}$ *and its other components to* 1 *(replacing concatenation of symbols by multiplication). If* $X^{(1)}(z)$ *denotes the generating function obtained from solving* $\mathcal{SE}$ *for* $X^{(1)}$, *choosing the unique solution compatible with initial conditions, then* $[z^n]X^{(1)}(z) = |\mathcal{L}(G) \cap T^n|$ *holds.*

[1] For MCFGs an ε-rule is given by a rule like $\vec{A} \to (\varepsilon, \ldots, \varepsilon)^T$.

Application

Using the lemma provides systems of equations like e.g. (Rivas&Eddy)

$$
\begin{aligned}
S &= zS^2 + 1 + GzS^2, \\
G &= A + B + D + zS^2, \\
A &= G(B + D + zS^2), \\
B &= G(A + D + zS^2), \\
D &= (G^2 + 2G)(A + B + zS^2).
\end{aligned}
$$

No need to solve it; expansion at dominant singularity is sufficient.

$\rightsquigarrow -0.0176036308\ldots\sqrt{1 - 15.7923959885\ldots z} + \mathcal{O}\big((1 - 15.7923959885\ldots z)\big)$.

Under mild conditions (isolated singularity, expansion valid beyond disc of convergence) we have for $f(z) = \sum_{n\geqslant 0} f_n z^n$ and $g(z) = \sum_{n\geqslant 0} g_n z^n$

$$[z^n]\big(f(z) + \mathcal{O}(g(z)\big) = f_n + \mathcal{O}(g_n).$$

Application

Using the lemma provides systems of equations like e.g. (Rivas&Eddy)

$$\begin{aligned}
S &= zS^2 + 1 + GzS^2,\\
G &= A + B + D + zS^2,\\
A &= G(B + D + zS^2),\\
B &= G(A + D + zS^2),\\
D &= (G^2 + 2G)(A + B + zS^2).
\end{aligned}$$

No need to solve it; expansion at dominant singularity is sufficient.

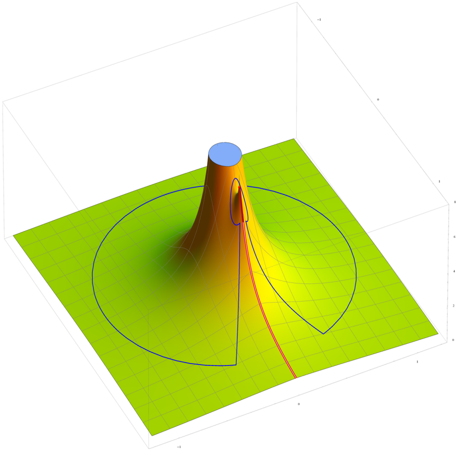
$\rightsquigarrow -0.0176036308\ldots\sqrt{1 - 15.7923959885\ldots z} + \mathcal{O}\big((1 - 15.7923959885\ldots z)\big)$.

Under mild conditions (isolated singularity, expansion valid beyond disc of convergence) we have for $f(z) = \sum_{n\geqslant 0} f_n z^n$ and $g(z) = \sum_{n\geqslant 0} g_n z^n$

$$[z^n]\big(f(z) + \mathcal{O}(g(z))\big) = f_n + \mathcal{O}(g_n).$$

Asymptotic of coefficients

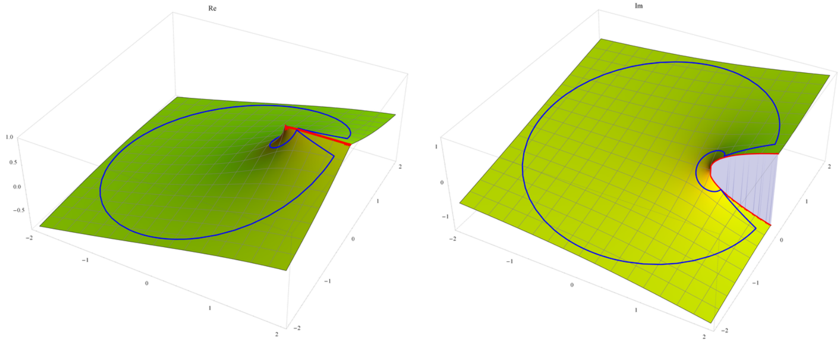
For $f(z) = \sum_{n \geqslant 0} f_n z^n$ we have $[z^n] f(z) = \frac{1}{2\pi i} \oint \frac{f(z)}{z^{n+1}} dz$.



Plot for Catalan generatingfunction (discontinuities marked red).

Asymptotic of coefficients

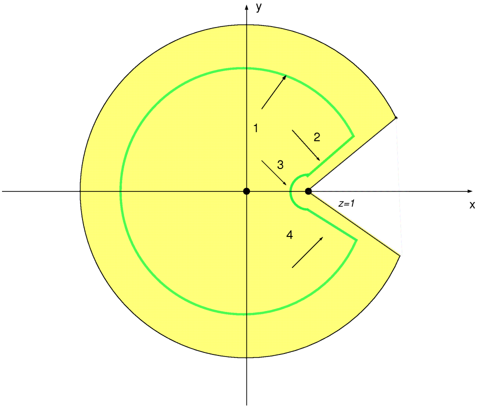
For $f(z) = \sum_{n \geqslant 0} f_n z^n$ we have $[z^n]f(z) = \frac{1}{2\pi i} \oint \frac{f(z)}{z^{n+1}} dz$.



Plots for $f(z) = 1 - \sqrt{1-z}$ (discontinuities marked red).

Asymptotic of coefficients

If function is analytic in a *pac-man* like domain,validity of termwise translation (asymptotic and error term) can be proved by integrating along the so-called Hannkel-contour (green line).



Results

Table: The asymptotical number of pseudoknot structures of size n.

Class	Asymp.	size = arcs α	size = arcs ω	size = all α	size = all ω	Run time algo.
L&P	$\alpha\omega^n$	$\frac{1}{2}$	4	$\frac{1}{4}$	3	$\mathcal{O}(n^5)$
C&C	$\frac{\alpha}{2\sqrt{\pi n^3}}\omega^n$	1.6651	5.857	4.0599	3.2864	$\mathcal{O}(n^6)$
R&G	$\frac{\alpha}{2\sqrt{\pi n^3}}\omega^n$	0.1651	6.576	2.7058	3.5129	$\mathcal{O}(n^4)$
D&P	$\frac{\alpha}{2\sqrt{\pi n^3}}\omega^n$	0.7535	7.315	1.7082	3.7046	$\mathcal{O}(n^5)$
A&U	$\frac{\alpha}{2\sqrt{\pi n^3}}\omega^n$	0.6575	7.547	1.4813	3.7472	$\mathcal{O}(n^6)$
R&N	$\frac{\alpha}{2\sqrt{\pi n^3}}\omega^n$	0.6429	8.284	1.4222	3.8782	$\mathcal{O}(n^6)$
R&E	$\frac{\alpha}{2\sqrt{\pi n^3}}\omega^n$	0.0176	15.792	0.0348	4.9739	$\mathcal{O}(n^6)$

Results

Table: The exact number of structures (roman), the corresponding asymptotic (italics) and their quotient (boldface).

	20	50	100	200
L&P	$3.561142614\ldots \times 10^{11}$ *5.49756×10^{11}* **0.647768**	$4.864448066\ldots \times 10^{29}$ *6.33825×10^{29}* **0.767475**	$6.694392900\ldots \times 10^{59}$ *8.03469×10^{59}* **0.833186**	$1.137720591\ldots \times 10^{120}$ *1.29112×10^{120}* **0.881186**
C&C	$1.112250463\ldots \times 10^{13}$ *1.18424×10^{13}* **0.939212**	$3.128096400\ldots \times 10^{35}$ *3.20821×10^{35}* **0.975028**	$2.704589242\ldots \times 10^{73}$ *2.7391×10^{73}* **0.9874**	$5.611595985\ldots \times 10^{149}$ *5.64734×10^{149}* **0.993671**
R&G	$8.025197758\ldots \times 10^{13}$ *8.40499×10^{13}* **0.954814**	$7.219431757\ldots \times 10^{37}$ *7.35462×10^{37}* **0.981618**	$2.038002911\ldots \times 10^{78}$ *2.05702×10^{78}* **0.990756**	$4.530232923\ldots \times 10^{159}$ *4.55133×10^{159}* **0.995364**
D&P	$4.426689974\ldots \times 10^{14}$ *4.57142×10^{14}* **0.968341**	$9.632041531\ldots \times 10^{39}$ *9.75745×10^{39}* **0.987147**	$5.562927773\ldots \times 10^{82}$ *5.59909×10^{82}* **0.993541**	$5.197760759\ldots \times 10^{168}$ *5.21464×10^{168}* **0.996762**
A&U	$7.251213289\ldots \times 10^{14}$ *7.45787×10^{14}* **0.972291**	$4.022948569\ldots \times 10^{40}$ *4.06881×10^{40}* **0.988729**	$1.109324813\ldots \times 10^{84}$ *1.11565×10^{84}* **0.994333**	$2.365677112\ldots \times 10^{171}$ *2.37242×10^{171}* **0.997158**
R&N	$4.532472408\ldots \times 10^{15}$ *4.70002×10^{15}* **0.964352**	$4.134959215\ldots \times 10^{42}$ *4.19553×10^{42}* **0.985563**	$1.204325100\ldots \times 10^{88}$ *1.21312×10^{88}* **0.992751**	$2.858258052\ldots \times 10^{179}$ *2.86868×10^{179}* **0.996368**
R&E	$5.937168927\ldots \times 10^{19}$ *5.16907×10^{19}* **1.14859**	$1.238646791\ldots \times 10^{55}$ *1.17474×10^{55}* **1.0544**	$3.565692041\ldots \times 10^{114}$ *3.47376×10^{114}* **1.02646**	$8.703449007\ldots \times 10^{233}$ *8.59127×10^{233}* **1.01306**

Conclusion

Making use of multiple context-free grammars, formal power series and generating functions, we have

- introduced a new enumeration lemma applicable to objects with mildly context sensitive encodings,
- making the search for a bijection superfluous.

Those general findings allowed to solve the long open problem to quantify the size of R&E class.
Our ideas can be used to handle a special kind of couplings within recursive decompositions. Accordingly, we also made used of our approach to the analysis of algorithms e.g. to analyze the size of the intersection of two random trees (as a proof of concept).

Conclusion

Thanks for your attention!