Issue Date: 12.01.2014
Version: 2015-01-21 20:54

9th Exercise Sheet for Kombinatorische Algorithmen, WS 14/15

Hand In: *Until Monday, 19.01.2015, 12:00,*
deliver or email to Raphael (`reitzig@cs.uni-kl.de`).

Problem 16

$2 + 2 + 1$ points

Consider the labeled combinatorial class $\mathcal{Q}$ implicitly defined by the specification

$$\Theta\mathcal{Q} \;=\; \mathcal{Q} \star \mathcal{Z} \star \mathcal{Q}\,, \tag{1}$$

where Θ is the *pointing operator* as defined by Flajolet et al. [1].

a) Determine the functional equation for the (exponential) generating function $Q(z)$ for $\mathcal{Q}$ according to specification (1). Translate this equation of generating functions into an equation of coefficients, thereby deriving a recurrence equation for Q_n, i. e. the number of objects of size n in $\mathcal{Q}$.

b) Note that (1) does not specify Q_0. Can you prove a simple closed form of Q_n assuming $Q_0 = 1$?

   For an educated guess you might compute the first few entries, say $Q_0, \ldots, Q_6$, and ask the *On-Line Encyclopedia of Integer Sequences* for help — but do not forget to prove your claims.

c) Show that $\mathcal{Q}$ is isomorphic to a well-known[1] combinatorial class.

[1] As in, you know it from your own studies.

Problem 17

1 + 4 + 3 points

In this exercise, we consider random generation of RNA secondary structures. RNA is an important single-stranded relative of DNA which exhibits rich folding structures. For details, refer to our lecture *Computational Biology II.*

Here, it suffices to know that secondary structures are isomorphic to *Motzkin words*, which are words over $\Sigma = \{\texttt{(},\texttt{)},\texttt{*}\}$ where `(` and `)` form well-nested pairs and `*` can appear anywhere. That is, Motzkin languages are Dyck languages shuffled with $\{\texttt{*}\}^\star$.

For instance, "`*((**(***))*((**))`" is a Motzkin word, but neither "`)(`" nor "`(())(`" are.

Formally, we define the *unlabeled* combinatorial class $\mathcal{S}$ of RNA secondary structures as

$$\mathcal{S} \;=\; \epsilon \;+\; \mathcal{Z}_{*} \times \mathcal{S} \;+\; \mathcal{Z}_{(} \times \mathcal{S} \times \mathcal{Z}_{)} \times \mathcal{S}\;. \tag{2}$$

a) Note that Flajolet et al. only consider labeled classes whereas the secondary structures defined by (2) are unlabeled. Argue why their approach of random generation is also applicable in our case.

b) Use the method described by Flajolet et al. [1] to design an algorithm that generates a secondary structure of given chosen uniformly among all structures $s \in \mathcal{S}_n$ of size n. The algorithm takes n as input.

   Give at least the following intermediate steps explicitly:

   - the *standard specification* for $\mathcal{S}$,
   - recurrence equations for the number of objects of all classes that occur in the standard specification and
   - the generation procedures for all those classes.

   You may directly simplify the generating procedures where possible because of feature of our special case; make sure to explain your simplifications.

c) Implement your procedure from b) in a programming language of your choice.

   Generate 1000 random structures of size $n = 100$ and draw a histogram of the number of unpaired bases U_n, i. e. the number of `*` atoms. Can you guess the distribution of U_n?

References

[1] Philippe Flajolet, Paul Zimmerman, and Bernard Van Cutsem. "A calculus for the random generation of labelled combinatorial structures." In: *Theoretical Computer Science* 132.1-2 (Sept. 1994), pp. 1–35. ISSN: 03043975. DOI: 10.1016/0304-3975(94)90226-7.