REGULAR LANGUAGES AND THEIR GENERATING FUNCTIONS: THE INVERSE PROBLEM

CHRISTOPH KOUTSCHAN

ABSTRACT. The technique of determining a generating function for an unambiguous context-free language is known as the Schützenberger methodology. For regular languages, Elena Barcucci et al. proposed an approach for inverting this methodology. This idea allows a combinatorial interpretation (by means of a regular language) of certain positive integer sequences that are defined by C-finite recurrences.

In this paper we present a Maple implementation of this inverse methodology and describe various applications. We give a short introduction to the underlying theory, i.e., the question of deciding $\mathbb{N}$-rationality. In addition, some aspects and problems concerning the implementation are discussed; some examples from combinatorics illustrate its applicability.

1. INTRODUCTION

This paper essentially deals with sequences of positive integers that are defined by a linear recurrence with constant coefficients (C-finite recurrence), and which can be identified with the power series expansion of some rational function. The focus of attention is the interrelation between such sequences and regular languages: A formal power series S corresponds to a formal language L, if

$$s_n = \Big|\{w \in L : |w| = n\}\Big|, \quad \text{where } S = \sum_{n=0}^{\infty} s_n x^n,$$

i.e., if the n^{th} coefficient s_n gives the number of words in L with length n.

How to find the power series corresponding to a given language (of certain type) is known as the Schützenberger methodology: Let $G = (V, \Sigma, P, I)$ be an unambiguous context free grammar of the language L_G, where V denotes the set of nonterminals, Σ the set of terminals, P the set of production rules, and I the initial symbol. The morphism Θ is defined by

$$\begin{aligned} \Theta(a) &= x && \forall a \in \Sigma \\ \Theta(\lambda) &= 1 && (\lambda \text{ denotes the empty word}) \\ \Theta(A) &= A(x) && \forall A \in V \end{aligned}$$

and is applied to all elements of P. Any production rule $A \to e_1|e_2|\dots|e_k$ yields an algebraic equation in the $A(x), B(x), \dots$:

$$\Theta(A) = \sum_{i=1}^{k} \Theta(e_i).$$

This system has to be solved for $I(x)$ and gives the generating function corresponding to L_G. In [CS63] it is proved that if G is an unambiguous regular grammar, then the corresponding generating function is rational.

In this paper, we consider the inverse problem by using the approach of Barcucci et al. [BLFR01]: Given a formal power series generated by some rational

Koutschan@risc.uni-linz.ac.at, supported by grant SFB F1305 of the Austrian FWF.

function, how can we obtain a regular expression for the corresponding regular language (in the case that such a language exists at all), and thus get a combinatorial interpretation of the series?

Section 2 is a short introduction to the underlying theory which will end up with an exact characterization of $\mathbb{N}$-rational series; solely for such series our inverse problem is solvable. In Section 3 we discuss some aspects concerning the implementation of the inverse Schützenberger methodology. At some points the methods described in [BLFR01] are sketchy and needed deeper investigation in order to describe them algorithmically. Also some mistakes had to be corrected. Summarizing, our work can be viewed as an algorithmic streamline of the approach of Barcucci et al. In addition, it resulted in a computer algebra implementation; the first one we know of. In order to illustrate the described algorithms and to demonstrate the functionality of our implementation, some examples from combinatorics are presented in Section 4.

2. Formal Power Series and Regular Languages

This section presents a condensed list of definitions and resulats, stated without proofs, that set the stage for the discussion of the method and its application. All these results can be found in [Niv69], [SS78], and [BR88].

2.1. **General setting.** We will mainly deal with the *free monoid* Σ^* generated by an *alphabet* Σ. Σ^* contains all finite sequences $x_1 \dots x_n$ of elements $x_i \in \Sigma$, including also the empty sequence denoted by λ. The elements of Σ^* are called *words*, which can be linked by concatenation. Of course, the empty word λ acts as neutral element of Σ^*. Every subset $L \subseteq \Sigma^*$ is referred to as a *formal language*. Further we need the notion of a *semiring*, i.e., roughly speaking, a ring without subtraction. For example, the natural numbers $\mathbb{N}$ form a semiring.

Definition 2.1. *Given an alphabet Σ and a semiring $\mathbb{K}$. A* formal power series *(or formal series) S is a function $S : \Sigma^* \to \mathbb{K}$. The image of a word w under S is called the* coefficient *of w in S and is denoted by s_w. S is written as a formal sum*

$$S = \sum_{w \in \Sigma^*} s_w w.$$

The set of formal power series over Σ^* with coefficients in $\mathbb{K}$ is denoted by $\mathbb{K}\langle\langle\Sigma^*\rangle\rangle$.

On the set $\mathbb{K}\langle\langle\Sigma^*\rangle\rangle$ the sum and Cauchy product are defined in the usual way; these operations induce the structure of a semiring on $\mathbb{K}\langle\langle\Sigma^*\rangle\rangle$. The set of words which have nonzero coefficients is referred to as the *support* of a series. The set of all series with finite support, i.e., all polynomials, is denoted by $\mathbb{K}\langle\Sigma^*\rangle$, which is a semiring, too. If $\mathbb{K}$ is a ring, then so are $\mathbb{K}\langle\langle\Sigma^*\rangle\rangle$ and $\mathbb{K}\langle\Sigma^*\rangle$. The support of any series $S \in \mathbb{K}\langle\langle\Sigma^*\rangle\rangle$ is a formal language over the alphabet Σ. On the other hand for any formal language L we define its *characteristic series* $\operatorname{char}(L) = \sum s_w w$ by

$$s_w = \begin{cases} 1 & \text{if } w \in L \\ 0 & \text{if } w \notin L \end{cases} \quad \forall w \in \Sigma^*$$

Definition 2.2. *A power series (especially a polynomial) $S \in \mathbb{K}\langle\langle\Sigma^*\rangle\rangle$ is called* quasiregular *if the coefficient of the neutral element of Σ^* vanishes, i.e., if $s_\lambda = 0$.*

Now let S be a quasiregular series. Then the limit

$$S^* = \lim_{m \to \infty} \sum_{n=0}^{m} S^n$$

exists (the sequence $S^0, S^1, S^2, \dots$ is called *summable*) and is named the *star* of S. In the theory of formal languages this expression is termed *Kleene closure*.

Definition 2.3. *The* rational operations *in* $\mathbb{K}\langle\langle\Sigma^*\rangle\rangle$ *are the sum, the product, and the star. A subsemiring of* $\mathbb{K}\langle\langle\Sigma^*\rangle\rangle$ *is* rationally closed *if it is closed for the rational operations. The* rational closure *of a subset* $M \subseteq \mathbb{K}\langle\langle\Sigma^*\rangle\rangle$ *is the smallest subset of* $\mathbb{K}\langle\langle\Sigma^*\rangle\rangle$ *containing* M *and being rationally closed. A formal series* $S \in \mathbb{K}\langle\langle\Sigma^*\rangle\rangle$ *is* $\mathbb{K}$-rational *if it is an element of the rational closure of* $\mathbb{K}\langle\Sigma^*\rangle$*. This set of all* $\mathbb{K}$*-rational series is denoted by* $\mathbb{K}^{\mathrm{rat}}\langle\langle\Sigma^*\rangle\rangle$*.*

The following two theorems state the strong connection between regular languages and $\mathbb{K}$-rational series.

Theorem 2.4. [SS78, Chap. II, Theorem 5.1] *Let* L *be a regular language and* $\mathbb{K}$ *a semiring. Then char*(L) *is* $\mathbb{K}$*-rational.*

Theorem 2.5. [SS78, Chap. II, Theorem 5.3] *The support of any formal power series* $S \in \mathbb{N}^{\mathrm{rat}}\langle\langle\Sigma^*\rangle\rangle$ *is a regular language.*

We will now leave this general multivariate and noncommutative setting.

2.2. **Rational Series in One Variable.** From now on we examine rational series over an alphabet that consists only of one single letter: $\Sigma = \{x\}$. Instead of $\mathbb{K}\langle\langle\Sigma^*\rangle\rangle$ we write $\mathbb{K}\langle\langle x^*\rangle\rangle$. Of course, x^* is a commutative monoid that is isomorphic to $(\mathbb{N}_0, +)$. Therefore, a series $S \in \mathbb{K}\langle\langle x^*\rangle\rangle$ is written as $S = \sum_{n=0}^{\infty} s_n x^n$.

A rational function $p(x)/q(x)$ is called *normalized* if p and q have no common factor in $\mathbb{K}\langle x^*\rangle$ and if $q(0) = 1$. In the following we consider rational functions to be given always in normalized form.

Definition 2.6. *Let* S *be a rational power series and* $f(x) = p(x)/q(x)$ *its normalized generating function with* $q(x) = 1 - q_1 x - \cdots - q_k x^k$*. Then the roots of* q *are called* poles *of* S*. The inverse values of the poles, i.e. the roots of the reciprocal polynomial* $\bar{q}(x) = x^k - q_1 x^{k-1} - \cdots - q_{k-1}x - q_k$ *are called* roots *of* S*.*

In the case that $\mathbb{K}$ is a commutative ring, there are three different characterizations of $\mathbb{K}$-rational series which are quite easy to verify. Let $\mathbb{K}\langle\langle x^*\rangle\rangle \ni S = \sum s_n x^n$. Then $S \in \mathbb{K}^{\mathrm{rat}}\langle\langle x^*\rangle\rangle$ if and only if one of the following three properties holds:

(1) S has a rational generating function $p(x)/(1 - q(x))$, where $p, q \in \mathbb{K}\langle x^*\rangle$ are polynomials and q is quasiregular.
(2) There is a C-finite recurrence for the coefficients of S, i.e., $s_n = q_1 s_{n-1} + \cdots + q_k s_{n-k}$, $q_i \in \mathbb{K}$, which holds for large n.
(3) The coefficients s_n can be expressed by the *exponential polynomial*:
$$s_n = \sum_{i=0}^{r} P_i(n)\lambda_i^n \quad \text{(for large } n\text{)},$$
where $\lambda_0, \ldots, \lambda_r$ are the distinct roots of S with multiplicities $m_0, \ldots, m_r$, and the P_i's are complex nonzero polynomials with $\deg P_i = m_i - 1$ and with coefficients that are algebraic over $\mathbb{K}$.

Note that changing finitely many coefficients of a $\mathbb{K}$-rational series preserves $\mathbb{K}$-rationality.

2.3. **Positive Series.** We demonstrated that for a formal power series the question of $\mathbb{K}$-rationality is not difficult to decide if $\mathbb{K}$ is a commutative ring. Let now $\mathbb{K}$ be a subring of $\mathbb{R}$. We want to examine series over $\mathbb{K}_+$ which is only a semiring (e.g. $\mathbb{K}_+ = \mathbb{N}$ or $\mathbb{K}_+ = \mathbb{R}_+$). This case is much more difficult and it will take some efforts to work out a criterion for deciding $\mathbb{K}_+$-rationality. In general it is not sufficient to show that a series in $\mathbb{K}^{\mathrm{rat}}\langle\langle x^*\rangle\rangle$ has positive coefficients. It may well happen that such a series is not $\mathbb{K}_+$-rational.

Example 2.7. *Consider the series* [Slo, A094423]*:*

$$x + 4x^2 + x^3 + 144x^4 + 361x^5 + 484x^6 + 19321x^7 + 28224x^8 + 128881x^9 + \dots$$

which is generated by the rational function $(x+5x^2)/(1+x-5x^2-125x^3)$. Although all coefficients of this series are positive integers it is not $\mathbb{N}$-rational.

The following theorem states a very important property of $\mathbb{K}_+$-rational series.

Theorem 2.8. [BR88, Chap. V, Theorem 2.2] *Let $S \in \mathbb{K}_+^{\mathrm{rat}}\langle\langle x^* \rangle\rangle \setminus \mathbb{K}_+\langle x^* \rangle$ have the generating function $f(x)$ and the roots $\lambda_0, \dots, \lambda_r$ and let $\varrho := \min_{0 \le i \le r} |\lambda_i^{-1}|$. Then:*

$$\left.\begin{array}{l}\varrho \text{ is a pole of } S \text{ (let } m_\varrho \text{ be its multiplicity) and all other poles of modulus} \\ \varrho \text{ have the form } \varrho\vartheta \text{ and a multiplicity} \le m_\varrho.\ \vartheta \text{ denotes a complex root of} \\ \text{unity, i.e., } \exists\, p \in \mathbb{N} : \vartheta^p = 1.\end{array}\right\} (*)$$

We now introduce the operations of decomposing and merging series. The next theorem states that these operations preserve $\mathbb{K}$-rationality, and additionally characterizes the roots of the decomposed series.

Definition 2.9. *Given a formal series $S = \sum s_n x^n$. For any $p \in \mathbb{N}$ the list of subseries $S_0, \dots, S_{p-1}$ is called a* decomposition *of S if*

$$S_i = \sum_{n=0}^{\infty} s_{i+np} x^n.$$

On the other hand S is termed the merge *of $S_0, \dots, S_{p-1}$:*

$$S(x) = \sum_{i=0}^{p-1} x^i S_i(x^p).$$

Thus to build up the subseries S_i one has to take every p^{th} coefficient, beginning at index i. How to obtain a generating function for the S_i will be described in Theorem 3.1.

Theorem 2.10. [BR88, Chap. V, Theorem 2.5] *Let $\mathbb{K}$ be a semiring. $S \in \mathbb{K}\langle\langle x^* \rangle\rangle$ is $\mathbb{K}$-rational if and only if there exist for any $p \in \mathbb{N}$ a set of $\mathbb{K}$-rational power series $S_0, S_1, \dots, S_{p-1}$ and their merge is S. Moreover, if $\mathbb{K}$ is commutative and $\lambda_0, \dots, \lambda_r$ are the roots of S with multiplicities $m_0, \dots, m_r$, then each of the S_j's has the following properties: The roots $\mu_0, \dots, \mu_s$ $(s \le r)$ of S_j are among the numbers $\lambda_0^p, \dots, \lambda_r^p$, and any root μ_l of S_j has the multiplicity $m_l' \le \max_{0 \le i \le r} \{m_i : \lambda_i^p = \mu_l\}$.*

The notion of a dominating root will play an extremely important role:

Definition 2.11. *Let $\lambda_0, \dots, \lambda_r$ be the roots of S. λ_0 is called the* dominating root *of S if $\lambda_0 \in \mathbb{R}_+$ and $\lambda_0 > |\lambda_i|, 1 \le i \le r$ holds.*

The following two theorems give us the complete characterization of $\mathbb{K}_+$-rational series. The important case for our work is $\mathbb{K}_+ = \mathbb{N}$.

Theorem 2.12. [SS78, Chap. II, Theorem 10.4] *Let $S \in \mathbb{K}_+\langle\langle x^* \rangle\rangle$ be $\mathbb{K}$-rational with dominating root λ_0. Then S is $\mathbb{K}_+$-rational.*

Theorem 2.13. [BR88, Chap. V, Theorem 2.10] *A series $S \in \mathbb{K}_+\langle\langle x^* \rangle\rangle$ is $\mathbb{K}_+$-rational if and only if it is a merge of $\mathbb{K}$-rational series each of them having a dominating root.*

If a series generated by f is $\mathbb{N}$-rational then there is a regular language L corresponding to it. We want to compute a regular expression for L. In fact, we transform f into an expression which we shall call pseudoregular, since it is not the same as one unterstands by regular expression in the narrow sense of the definition.

An expression is called pseudoregular if it involves only polynomials from $\mathbb{N}[x]$, connected by addition, multiplication and star operation. This can be translated into a regular expression using the procedure *ren* proposed in [BLFR01, p. 133]:

First of all, an alphabet Σ is initialized with the empty set.

$$\begin{aligned}
ren(1) &= \lambda \\
ren(a) &= \begin{cases} a & \text{if } a \notin \Sigma. \text{ Then set } \Sigma := \Sigma \cup \{a\} \\ b, b \notin \Sigma & \text{if } a \in \Sigma. \text{ Then set } \Sigma := \Sigma \cup \{b\} \end{cases} \\
ren(X+Y) &= ren(X) \vee ren(Y) \\
ren(X \cdot Y) &= ren(X) ren(Y) \\
ren(X^*) &= (ren(X))^*
\end{aligned}$$

Herein X and Y denote arbitrary pseudoregular expressions.

Example 2.14. *Consider the pseudoregular expression* $2\,(x^*)^3 \cdot x\,\left(x^* x^2 + 1\right)$. *Applying the procedure ren yields the regular expression* $(a^*b^*c^* \vee d^*e^*f^*)\,g\,(h^*ij \vee \lambda)$ *and the alphabet* $\Sigma = \{a, b, c, d, e, f, g, h, i, j\}$.

3. Realization with Maple

This section describes some problems we had to overcome in our implementation which is presented in more details in [Kou05]. This thesis contains also the manpages for the usage of our package. Throughout this section we consider the following rational function to demonstrate our program:

```
> f:= 1/(1-2*x)^2/(1-10*x^2);
```

$$f := \frac{1}{(1-2x)^2(1-10x^2)}$$

We use our procedure `getCoefficients` for a fast computation of the first coefficients of the corresponding series:

```
> getCoefficients(f, 10);
```

$$[1, 4, 22, 72, 300, 912, 3448, 10144, 36784, 106560, 379104]$$

3.1. Getting the Roots.

First of all, we need a procedure for determining all (different) roots of a polynomial (or a rational function). The multiplicities of the roots need not be respected. Instead of using the Maple command `solve`, which can lead to time-consuming computations and unwieldy results (think of the general case of a polynomial with degree 4), we use without exception Maple's `RootOf` expressions. For this purpose the polynomial is made squarefree, and then factorized. We implemented this in the procedures `getRoots` (for polynomials) and `getRootsRat` (for rational functions); both return the roots as an unsorted list:

```
> lambda:= getRootsRat(f);
```

$$\lambda := \left[2, \frac{1}{\mathrm{RootOf}(-1+10\_Z^2, index=1)}, \frac{1}{\mathrm{RootOf}(-1+10\_Z^2, index=2)}\right]$$

3.2. Decomposition.

Consider the case that a given series S has no dominating root, but several different roots with maximal modulus; we denote these roots by $\varrho\vartheta_0, \ldots, \varrho\vartheta_k$, where ϱ is a positive real number, and the ϑ_i's are complex numbers with $|\vartheta_i| = 1$. To decide that S is $\mathbb{N}$-rational we must find an integer p such that each subseries of the decomposition $S_0, \ldots, S_{p-1}$ has a dominating root (see Theorem 2.13); this is fulfilled by all numbers p for which $\vartheta_0^p = \cdots = \vartheta_k^p = 1$ holds. By Theorem 2.8 we know that if S is $\mathbb{N}$-rational then the ϑ_i's are complex roots of unity. We first describe how this number p can be found and then how the decomposition itself can be computed.

3.2.1. *The Symmetric Polynomial.* We define the symmetric polynomial R by

$$R(x) := \prod_{\substack{0\le i,j\le r\\ i\ne j}} (\lambda_i - \lambda_j x).$$

where $\lambda_0, \dots, \lambda_r$ are again the roots of S. R can be computed by means of a resultant, and thus has integral coefficients. It has the roots λ_i/λ_j $(0 \le i, j \le r)$, and in the case that S is $\mathbb{N}$-rational among them the roots of unity $\vartheta_0, \dots, \vartheta_k$, since then ϱ itself is a root of S. It is easy to show that if an n^{th} root of unity ϑ_i is a root of R, then R must be divisible by the n^{th} cyclotomic polynomial $\Phi_n(x)$.

Our strategy is the following: The polynomial R is factorized over $\mathbb{Z}$ in order to find all cyclotomic polynomials $\Phi_{n_1}, \dots, \Phi_{n_j}$ that divide it; for this purpose the Maple function `numtheory[invphi]` is used. Then we set $p = \operatorname{lcm}(n_1, \dots, n_j)$. These steps are performed in the procedure `commonUnityRoots`:

```
> commonUnityRoots(denom(f));
```

$$2$$

3.2.2. *Computing the Decomposition.* We know now that we have to decompose S into p subseries, but we need these in an explicit form, i.e., given by their generating functions. This is carried out by means of the multisection formula:

Theorem 3.1. [Rio58, Chap. 4] *Given a series S by its generating function $f(x)$ and an integer p. Let $S_0, \dots S_{p-1}$ denote the decomposition of S. Then*

$$f_i(x) = \frac{1}{px^{i/p}} \sum_{j=1}^{p} s^{p-ij} f(s^j x^{1/p}), \quad s = e^{2\pi i/p}$$

is the generating function for the subseries S_i.

There arise some problems concerning the implementation that we solved in the following way: If the above formula for the f_i's is fed one-to-one into Maple, then in many cases the system does not succeed in simplifying the resulting expression, and even if so, the computation is very slow. A first speed-up is obtained by substituting $x^{1/p}$ by a new variable y. But still, Maple often fails to simplify when roots of unity are involved.

In [Ber89] we get a hint on how to handle this problem: Consider an expression containing several p^{th} roots of unity (let s be a primitive one). Thus our computations take place in the field $\mathbb{Q}[s]$ which is isomorphic to $\mathbb{Q}[x]/\langle\Phi_p(x)\rangle$. For our purposes this means that we introduce a new variable s that represents the root of unity $e^{2\pi i/p}$; then we reduce modulo $\Phi_p(s)$. Thanks to the above isomorphism we obtain the correct result in a fraction of computation time compared to before.

```
> f0:= decomposition(f, 2, 1); f1:= decomposition(f, 2, 2);
```

$$f0 := \frac{-(4x+1)}{(18x + 160x^3 - 96x^2 - 1)}$$

$$f1 := \frac{-4}{(18x + 160x^3 - 96x^2 - 1)}$$

```
> getCoefficients(f0, 5);
```

$$[1, 22, 300, 3448, 36784, 379104]$$

Here we see that in fact every second coefficient of f appears in the subseries f_0. We can as well verify the statement from Theorem 2.10 concerning the roots of the subseries:

```
> lambda0:= getRootsRat(f0);
```

$$\lambda 0 := [10, 4]$$

3.3. **Deciding $\mathbb{N}$-Rationality.** To find out if a given series is $\mathbb{N}$-rational, according to Theorem 2.12, two properties must be verified: The existence of a dominating root and the nonnegativeness of all coefficients.

3.3.1. *Existence of a Dominating Root.* The problem of deciding if the absolute values of two roots λ_i and λ_j are equal is nontrivial, because in general Maple is not capable to solve this by symbolic computation. So, let's compute the roots numerically and compare them. But what to do if $\big||\lambda_i| - |\lambda_j|\big|$ is smaller than our numerical precision? We will make use of the following result that tells us how small this distance theoretically can be:

Theorem 3.2. [GS96, p. 9] *Let p be a polynomial over the integers, $\alpha_1, \dots, \alpha_n$ its roots and thus $\deg p = n > 0$ its degree. Define $\kappa(p)$ to be the following quantity*

$$\kappa(p) = \frac{\sqrt{3}}{2}\left(\frac{n(n+1)}{2}\right)^{-\left(\frac{1}{4}n(n+1)+1\right)} \cdot M(p)^{-\frac{1}{2}n\left(n^2+2n-1\right)},$$

then $|\alpha_i| \neq |\alpha_j| \implies \big||\alpha_i| - |\alpha_j|\big| \geq \kappa(p)$ and $\big|\mathrm{Im}(\alpha_i)\big|$ is either 0 or larger than $\kappa(p)$. Herein $M(p)$ is defined by $M(p) := |p_n| \prod_{i=1}^{n} \max\{1, |\alpha_i|\}$.

This formula has to be used carefully; consider the generating function from Section 4.2: Applying Theorem 3.2 we get $\kappa(q) \approx 2.159917528 \cdot 10^{-287579}$. Thus we had to compute with a precision of 287580 digits! But the dominating root (we will see that there is one) differs already in the second digit from the absolute values of all other roots. This gives us reason for proceeding in the following way: First, numerical values with low precision are computed for all roots. If this is not enough for deciding the dominating root property, the precision is increased up to $1 - \lfloor \log_{10} \kappa \rfloor + \lfloor \log_{10} \lambda_0 \rfloor$ digits. The procedure `hasDominatingRoot` carries out these steps and outputs an integer s which has to be interpreted in the following way:

- $s = 0$: There is a dominating root.
- $s = 1$: None of the roots with maximal modulus is positive real.
- $s = 2$: Several roots with maximal modulus and one of them is positive real.

Furthermore a partial sorting on the list of the roots is performed. "Partially sorted" means that the roots having maximal modulus are on the head, and in the case that there is a dominating root, it is followed by one of the roots having second greatest modulus. Some other procedures need this partially sorted list as input.

```
> hasDominatingRoot(reciprocal(denom(f)), lambda, 'lambdaSorted');
```

$$2$$

```
> lambdaSorted;
```

$$\left[\frac{1}{\mathrm{RootOf}(-1+10\_Z^2, \mathit{index}=1)}, \frac{1}{\mathrm{RootOf}(-1+10\_Z^2, \mathit{index}=2)}, 2\right]$$

```
> hasDominatingRoot(reciprocal(denom(f0)), lambda0, 'lambdaSorted0');
```

$$0$$

3.3.2. *Nonnegative Coefficients.* The second important property we have to verify is that all coefficients of the series S are nonnegative. If the series is finite then this is easy to check. In the other case we first compute a boundary n_0 such that all coefficients s_n with $n > n_0$ are nonnegative. The remaining coefficients $s_0, \dots, s_{n_0}$ are tested one by one.

Assume that the given rational function f has a dominating root. We compute the exponential polynomial $P_0(n)\lambda_0^n + \cdots + P_r(n)\lambda_r^n = s_n$ and verify that the leading coefficient of the polynomial $P_0(n)$, which corresponds to the dominating root λ_0, is positive; otherwise no such boundary n_0 exists. Then we distinguish two cases:

(1) f has exactly one root (with multiplicity ≥ 1). Hence the coefficients of the series can be written as $s_n = P_0(n)\lambda_0^n$, $(\lambda_0 > 0)$. We have to choose n_0 such that $P_0(n) > 0$ for all $n > n_0$.

(2) f has several different roots. This case is the most complicated one. Knowing the coefficients of the polynomials P_i we again can compute the bound n_0. Since this is quite technical, we skip it here and refer to [Kou05].

All this is implemented in the procedure `boundaryForNonnegCoeffs` which first computes the boundary n_0, and then identifies the minimal n_1 such that all coefficients s_n with $n > n_1$ are nonnegative, i.e., $n_1 = -1$ in case that there are no negative coefficients at all:

```
> boundaryForNonnegCoeffs(f0, lambdaSorted0);
```

$$-1$$

3.4. **Regular Expressions.** After verifying the $\mathbb{N}$-rationality of a series S generated by f we want to compute a pseudoregular expression for a corresponding regular language. The transformation of f into such a pseudoregular expression is quite complicated. In some cases S has to be decomposed, and in the end, the pseudoregular expressions for the subseries have to be combined. In general, this procedure works recursively on the multiplicity of the dominating root of f. The algorithm that we worked out is described in detail in [Kou05], and is implemented in the procedure `regularExpression` (herein the star operation x^* is denoted by the function `star(x)`):

```
> regularExpression(f0, lambdaSorted0, 0);
```

$$25600 star(36x^2)^2 star(96x^2 + 25600x^6 star(36x^2))x^6 + 264 star(36x^2) star(96x^2 + 25600x^6 star(36x^2))x^2 + 640 star(36x^2) star(96x^2 + 25600x^6 star(36x^2))x^4 + 36x^2 star(36x^2) + 1 + x(563200 star(36x^2)^2 star(96x^2 + 25600x^6 star(36x^2))x^6 + 2656 star(36x^2) star(96x^2 + 25600x^6 star(36x^2))x^2 + 792x^2 star(36x^2) + 22)$$

We see that the coefficients in the resulting pseudoregular expression are often not as small as in Example 2.14, and therefore it does not make sense to implement the procedure *ren*. Otherwise we would get extremely huge regular expressions and alphabets containing thousands of letters.

The above pseudoregular expression is just the first subseries f_0. In the same way we would have to examine f_1 and put both results together in order to get a pseudoregular expression for f.

3.5. **Conclusion.** For convenience we assembled all the steps from the previous sections in the procedure `analyze`: It decides if a given rational function $f(x)$ is $\mathbb{N}$-rational, and in the affirmative case computes a pseudoregular expression for f. It returns `false` if f is not $\mathbb{N}$-rational, and the pseudoregular expression otherwise:

```
> analyze(1/(1-x));
```

$$x\ star(x) + 1$$

Note that this is equivalent to $star(x)$, and thus exactly what we expect here.

4. Further Examples

4.1. **The MIU System.** In [Hof79] Douglas Hofstadter introduces the famous MIU system. This formal system defines a language L_{MIU} over the alphabet $\Sigma = \{\mathsf{M}, \mathsf{I}, \mathsf{U}\}$. Its words can be obtained by starting with the axiom MI and by applying the following rules:

(1) $w\mathsf{I} \to w\mathsf{IU}$
(2) $\mathsf{M}w \to \mathsf{M}ww$
(3) $\mathsf{III} \to \mathsf{U}$
(4) $\mathsf{UU} \to \lambda$

where w denotes an arbitrary word $w \in \Sigma^*$. The language L_{MIU} turns out to be regular, since every word begins with an M, followed by a string containing only I's and U's, where the number of I's is not divisible by 3:

$$w \in L_{MIU} \iff w = \mathsf{M}w' \wedge w' \in \{\mathsf{I}, \mathsf{U}\} \wedge \#_{\mathsf{I}}(w') \not\equiv 0 \bmod 3.$$

By analyzing the finite automaton that accepts the language L_{MIU}, we find out that $(x^2)/(1-3x+3x^2-2x^3)$ is the generating function of the corresponding power series [Slo, A024495]:

$$x^2 + 3x^3 + 6x^4 + 11x^5 + 21x^6 + 42x^7 + 85x^8 + \dots$$

We feed our program with this function and obtain a pseudoregular expression:

```
> analyze(x^2/(1-3*x+3*x^2-2*x^3);
```

$$\begin{gathered} star(x^2)star(2x^2+5x^4+9x^6 star(x^2))x^2(1+3x^2) \\ +x^3 star(x^2)star(2x^2+5x^4+9x^6 star(x^2))(3+2x^2) \end{gathered}$$

By factoring and replacing the unconvenient *star* notation, this simplifies to

$$(x^2)^* \left(x^2 \left(2+5x^2+9x^4(x^2)^*\right)\right)^* x^2(2x+1)(x^2+x+1).$$

4.2. **Look and Say.** A very interesting sequence discovered and examined by John Conway in [Con87] is the so-called *Look and Say Sequence.* It starts with 1, and every subsequent element is the "description" of the previous one. The elements are considered to be strings over the alphabet of digits (it turns out that solely the digits 1, 2, and 3 appear). Then the "description" of an element can be written by the rule

$$x_1^{m_1} x_2^{m_2} \cdots x_k^{m_k} \;\to\; m_1 x_1 m_2 x_2 \cdots m_k x_k.$$

So, the initial string 1 can be described by 11 ("one one"), which itself can be described by 21 ("two ones"). The first elements of the Look and Say Sequence [Slo, A005150] are

$$1, 11, 21, 1211, 111221, 312211, 13112221, 1113213211, \dots$$

We are interested in the formal power series S_{LS} obtained by writing down the lengths of the words in the Look and Say Sequence [Slo, A005341]:

$$S_{LS} = 1 + 2x + 2x^2 + 4x^3 + 6x^4 + 6x^5 + 8x^6 + 10x^7 + \dots$$

The series S_{LS} is generated by the rational function $f_{LS} = p/q$ where p and q are polynomials of degree 78 and 72 respectively. This monstrous function is quite a challenge for our program! We first find out that it has a dominating root:

```
> r:= reciprocal(denom(fLS)):
  hasDominatingRoot(r, getRootsRat(fLS), 'lambdaSorted');
```

$$0$$

```
> evalf(lambdaSorted[1]);
```

$$1.3035772690342963913$$

This number (we denote it by γ) is known as Conway's constant. It indicates that the word lengths in the Look and Say Sequence grow asymptotic to $C\gamma^n$, where C can be computed by our procedure `exponentialPolynomial`:

```
> op(1, exponentialPolynomial(fLS, lambdaSorted));
```

$$2.0421600768578803676$$

We now try to determine a pseudoregular expression for the series S_{LS}. Indeed, after a few hours of computation time, we get a result that fills lots of pages. For computing the pseudoregular expression the series has to be decomposed into 8 subseries which inflates the length of the result by the factor 8. We can verify its correctness by assigning the function $x \mapsto 1/(1-x)$ to the `star` symbol and by subsequent simplifying: Voilà, we obtain the original function f_{LS}!

5. Conclusion

Our Maple package `RLangGFun` is freely available at
http://www.risc.uni-linz.ac.at/research/combinat/software/RLangGFun/

This paper emerged from my master's thesis [Kou05] which is available in the RISC activity database (http://www.risc.uni-linz.ac.at). I want to thank my advisor Volker Strehl, who gave valuable hints and took much time to discuss occurring problems.

References

[Ber89] François Bergeron. A story about computing with roots of unity. In *Proceedings of the third conference on Computers and mathematics*, pages 140–144, New York, 1989. Springer-Verlag.

[BLFR01] Elena Barcucci, Alberto Del Lungo, Andrea Frosini, and Simone Rinaldi. A technology for reverse-engineering a combinatorial problem from a rational generating function. *Advances in Applied Mathematics*, 26(2):129–153, 2001.

[BR88] Jean Berstel and Christophe Reutenauer. *Rational Series and Their Languages.* Springer-Verlag, Berlin, 1988.

[Con87] John H. Conway. The weird and wonderful chemistry of audioactive decay. In Thomas M. Cover and B. Gopinath, editors, *Open Problems in Communication and Computation*, pages 173–188. Springer-Verlag, 1987.

[CS63] Noam Chomsky and Marcel P. Schützenberger. The algebraic theory of context-free languages. In P. Braffort and D. Hirschberg, editors, *Computer Programming and Formal Languages*, pages 118–161. North Holland, 1963.

[GS96] Xavier Gourdon and Bruno Salvy. Effective asymptotics of linear recurrences with rational coefficients. *Discrete Mathematics*, 153(1–3):145–163, 1996. Extended version of an article published in the proceedings of the 5th conference on Formal Power Series and Algebraic Combinatorics, FPSAC'93, Florence, July 1993.

[Hof79] Douglas R. Hofstadter. *Gödel, Escher, Bach: an Eternal Golden Braid.* Basic Books, Inc., New York, 1979.

[Kou05] Christoph Koutschan. Regular languages and their generating functions: The inverse problem. Diplomarbeit, Friedrich-Alexander-Universität Erlangen-Nürnberg, August 2005. http://www.risc.uni-linz.ac.at/people/ckoutsch/research/en_da.html.

[Niv69] Ivan Niven. Formal power series. *American Mathematical Monthly*, 76:871–889, 1969.

[Rio58] John Riordan. *Combinatorial Identities.* John Wiley & Sons, New York, 1958.

[Slo] Neil J. A. Sloane. *The On-Line Encyclopedia of Integer Sequences.* http://www.research.att.com/~njas/sequences/.

[SS78] Arto Salomaa and Matti Soittola. *Automata-Theoretic Aspects of Formal Power Series.* Springer-Verlag, New York, 1978.

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