Recurrence Relations, Succession Rules, and the Positivity Problem

Stefano Bilotta[1], Elisa Pergola[1], Renzo Pinzani[1], and Simone Rinaldi[2]

[1] Dipartimento di Matematica e Informatica "Ulisse Dini"
University of Florence
Viale Morgagni 65, 50134 Firenze, Italy
{stefano.bilotta,elisa.pergola,renzo.pinzani}@unifi.it
[2] Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche
University of Siena
Via Roma 56, 53100 Siena, Italy
rinaldi@unisi.it

Abstract. In this paper we present a method which can be used to investigate on the positivity of a number sequence defined by a recurrence relation having constant coefficients (in short, a C-recurrence).

Keywords: C-recurrences, positive numbers sequence

1 Introduction

Succession rules (sometimes called *ECO-systems*) have been proved to be an efficient tool in order to solve several combinatorial problems. The concept of a succession rule was introduced in [6] to study reduced Baxter permutations, and only later this has been recognized as an extremely useful tool for the ECO method, a methodology applied for the enumeration of various combinatorial structures [2].

An *(ordinary) succession rule* Ω is a system constituted by an *axiom* and a set of *productions*. A production constructs the *successors* of any given label (k). The rule Ω can be represented by means of a *generating tree* having the axiom as the label of the root and each node labelled (k) at level n has k sons at level $n+1$.

A succession rule Ω defines a sequence of positive integers $\{f_n\}_{n\geq 0}$ where f_n is the number of the nodes at level n in the generating tree defined by Ω. By convention the root is at level 0, so $f_0 = 1$. The function $f_\Omega(x) = \sum_{n\geq 0} f_n x^n$ is the *generating function* determined by Ω.

More recently, there have been some efforts in developing methods to pass from a recurrence relation defining an integer sequence to a succession rule defining the same sequence; in this case we say that the succession rule and the recurrence relation are *equivalent*.

Our work fits into this research line, and tries to deepen the relations between succession rules and recurrence relations.

It is worth mentioning that almost all studies realized until now on this topic have regarded linear recurrence relations with a finite number of integer coefficients [4, 7]. We will address to these ones as *C-finite recurrence relations*, and to the defined sequences as *C-finite sequences* [18].

Accordingly, our work will start considering C-finite recurrences. Compared with the methods presented in [4, 7], our approach is completely different.

To achieve this goal, we first translate the given C-finite recurrence relation into an *extended succession rule*, which differs from the ordinary succession rules since it admits both jumps and marked labels. Then we recursively eliminate jumps and marked labels from such an extended succession rule, thus obtaining an ordinary succession rule equivalent to the previous one. We need to point out that this translation is possible only if a certain condition – called *positivity condition* – is satisfied. Such a condition ensures that all the labels of the generating tree are non marked, hence the sequence defined by the succession rule has all positive terms.

If the recurrence relation has degree k with coefficients $a_1, \ldots, a_k$, such a condition can be expressed in terms of a set of k inequalities which can be obtained from a set of quotients and remainders given by the coefficients. To the authors' knowledge, such a condition is completely new in literature. It directly follows that our positive condition provides a sufficient condition for testing the positivity of a C-finite sequence, then it is related to the so called *positivity problem*.

Positivity Problem: given a C-finite sequence $\{f_n\}_{n\geq 0}$, establish if all its terms are positive.

This problem was originally proposed as an open problem in [3], and then re-presented in [16] (Theorems 12.1-12.2, pages 73-74), but no general solution has been found yet.

It is worth mentioning that the positivity problem can be solved for a large class of C-finite sequences, precisely those whose generating function is a $\mathbb{N}$-rational series. We also recall that the class of $\mathbb{N}$-rational series is precisely the class of the generating functions of regular languages, and that a Soittola's Corollary in [17] states that the problem of establishing whether a rational generating function is $\mathbb{N}$-rational is decidable.

$\mathbb{N}$-rational series have been recently revisited using modern combinatorial techniques in [4, 15], using different approaches and some algorithms to pass from an $\mathbb{N}$-rational series to a regular expression enumerated by such a series have been proposed [1,13]. However, none of these techniques provides a method to face C-finite recurrence relations which are not $\mathbb{N}$-rational.

Following the attempt of enlightening some questions on positive sequences, some researches have recently focused on determining sufficient conditions to establish the possible positivity of a given C-finite recurrence relation, as interestingly described in [11]. As a matter of fact, up to now, we only know that the positivity problem is decidable for C-finite recurrences of two [12] or three terms [14]. Another approach to tackle the positivity problem is to develop

algorithms to test possible positivity of recursively defined sequences (and, in particular, C-finite sequences) by means of computer algebra, as in [10].

Our work fits into this research line, since the positive condition we propose is a sufficient condition for testing the positivity of a C-finite sequence.

2 Basic Definitions and Notations

In this section we present some basic definitions and notations related to the concept of succession rule. For further definitions and examples we address the reader to [6].

Two succession rules are *equivalent* if they have the same generating function. A succession rule is *finite* if it has a finite number of labels and productions.

For example, the two succession rules:

$$\begin{cases} (2) \\ (2) \rightsquigarrow (2)(2) \end{cases} \qquad \begin{cases} (2) \\ (k) \rightsquigarrow (1)^{k-1}(k+1) \end{cases}$$

are equivalent rules, and define the sequence $f_n = 2^n$. The one on the left is a finite rule, since it uses only the label (2), while the one on the right is an infinite rule.

According to the technique of *colored label* [8], in a succession rule there can be labels $(k)_1$ and $(k)_2$ having the same number k of sons but having different productions, in this case we refer to *colored succession rules*.

A slight generalization of the concept of ordinary succession rule is provided by the so called *jumping succession rule* [9]. Roughly speaking, the idea is to consider a set of succession rules acting on the objects of a class and producing sons at different levels.

The usual notation to indicate a jumping succession rule Ω is the following:

$$\begin{cases} (a) \\ (k) \overset{j_1}{\rightsquigarrow} (e_{11}(k))(e_{12}(k)) \dots (e_{1k}(k)) \\ (k) \overset{j_2}{\rightsquigarrow} (e_{21}(k))(e_{22}(k)) \dots (e_{2k}(k)) \\ \quad \vdots \\ (k) \overset{j_m}{\rightsquigarrow} (e_{m1}(k))(e_{m2}(k)) \dots (e_{mk}(k)) \end{cases}$$

The generating tree associated with Ω has the property that each node labelled (k) lying at level n produces m sets of sons at level $n + j_1$, $n + j_2$, ..., $n + j_m$, respectively and each of such set has labels $(e_{i1}(k)), (e_{i2}(k)), \dots, (e_{ik}(k))$ respectively, $1 \leq i \leq m$.

We need to point out that a node labelled (k) has precisely k sons, according to the above definitions. A rule having this property is said to be *consistent*. However, in many cases we can relax this constraint and consider rules, where the number of sons is a function of the label k.

Another generalization is used in [5], where the authors deal with *jumping and marked succession rules*. In this case the labels appearing in a jumping

succession rule can be marked, and the *marked* labels are considered together with the unmarked ones.

A *jumping and marked generating tree* is a rooted labelled tree where there appear marked and unmarked labels according to the corresponding succession rule. The main property is that in the generating tree a marked label $(\overline{k})$ kills or annihilates the unmarked label (k) lying on the same level n. In particular, the enumeration of the combinatorial objects in a class is the difference between the number of unmarked and marked labels lying on a given level.

For any label (k), we introduce the following notation for generating tree specifications:

$$(\overline{\overline{k}}) = (k); \quad (k)^n = \underbrace{(k)\dots(k)}_{n} \quad n > 0; \quad (k)^{-n} = \underbrace{(\overline{k})\dots(\overline{k})}_{n} \quad n > 0.$$

3 A Method to Translate C-sequences into Succession Rules

The main purpose of our research is to develop a general formal method to translate a given recurrence relation into a succession rule defining the same number sequence. In this case we will say that the recurrence relation and the succession rules are *equivalent* by abuse of language.

This section is organized as follows.

i) We deal with C-finite recurrences of the form

$$f_n = a_1 f_{n-1} + a_2 f_{n-2} + \ldots + a_k f_{n-k} \qquad a_i \in \mathbb{Z}, 1 \leq i \leq k \tag{1}$$

with *default* initial conditions, i.e. $f_0 = 1$ and $f_h = 0$ for all $h < 0$. First, we translate the given C-finite recurrence relation into an extended succession rule, possibly using both jumps and marked labels (Section 3.1).

ii) Then, we recursively eliminate jumps and marked labels from such an extended succession rule, thus obtaining a finite succession rule equivalent to the previous one (Section 3.2). We remark that steps i) and ii) can be applied independently of the positivity of $\{f_n\}_{n\geq 0}$, but at this step we cannot be sure that all the labels of the obtained rule are nonnegative integers.

iii) We state a condition to ensure that the labels of the obtained succession rule are all nonnegative. If such a condition holds, then the sequence $\{f_n\}_{n\geq 0}$ has all positive terms, thus we refer to this as *positivity condition* (Section 3.3).

3.1 C-sequences with Default Initial Conditions

Let us consider a C-finite recurrence relation expressed as in (1), with default initial conditions and the related C-finite sequence $\{f_n\}_{n\geq 0}$. We recall that the generating function of $\{f_n\}_{n\geq 0}$ is rational, and precisely it is $f(x) = \sum_{n\geq 0} f_n x^n = \frac{1}{1-a_1 x - a_2 x^2 - \ldots - a_k x^k}$.

The first step of our method consists into translating the C-finite recurrence relation (1) into an extended succession rule. The translation is rather straightforward, since in practice it is just an equivalent way to represent the recurrence relation.

Proposition 1. *The recurrence relation (1) with default initial conditions is equivalent to the following extended succession rule:*

$$\begin{cases} (a_1) \\ (a_1) \overset{1}{\rightsquigarrow} (a_1)^{a_1} \\ (a_1) \overset{2}{\rightsquigarrow} (a_1)^{a_2} \\ \quad \vdots \\ (a_1) \overset{k}{\rightsquigarrow} (a_1)^{a_k} \end{cases} \tag{2}$$

For example, the recurrence relation $f_n = 3f_{n-1} + 2f_{n-2} - f_{n-3}$ with default initial conditions, defines the sequence $1, 3, 11, 38, 133, 464, 1620, 5655, \ldots$, and it is equivalent to the following extended succession rule:

$$\begin{cases} (3) \\ (3) \overset{1}{\rightsquigarrow} (3)^3 \\ (3) \overset{2}{\rightsquigarrow} (3)^2 \\ (3) \overset{3}{\rightsquigarrow} (\overline{3}) \end{cases} \tag{3}$$

Figure 1 shows the first few levels of the associated generating tree.

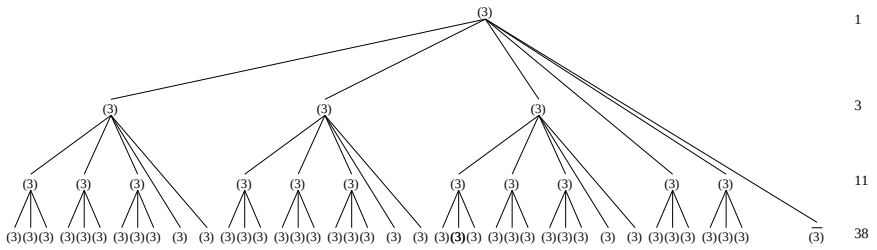

Fig. 1. Four levels of the generating tree associated with the succession rule (3)

3.2 Elimination of Jumps and Marked Labels

The successive step of our method consists into recursively eliminating jumps from the extended succession rule (2) in order to obtain a finite succession rule which is equivalent to the previous one. Once jumps have been eliminated we will deal with marked labels.

Proposition 2. *The succession rule:*

$$\begin{cases} (a_1) & \\ (a_1) & \rightsquigarrow (a_1+a_2)(a_1)^{a_1-1} \\ (a_1+a_2) & \rightsquigarrow (a_1+a_2+a_3)(a_1)^{a_1+a_2-1} \\ \vdots & \\ (\sum_{l=1}^{k-1} a_l) & \rightsquigarrow (\sum_{l=1}^{k} a_l)(a_1)^{(\sum_{l=1}^{k-1} a_l)-1} \\ (\sum_{l=1}^{k} a_l) & \rightsquigarrow (\sum_{l=1}^{k} a_l)(a_1)^{(\sum_{l=1}^{k} a_l)-1} \end{cases} \tag{4}$$

is equivalent to the recurrence relation $f_n = a_1 f_{n-1} + a_2 f_{n-2} + \ldots + a_k f_{n-k}$, $a_i \in \mathbb{Z}, 1 \leq i \leq k$, *with default initial conditions.*

Please notice that the numbers inside a label are the coefficients of the recurrence relation and their algebraic sum gives the number of successors of that label. Obviously, the labels $(a_1), (a_1+a_2), \cdots, (\sum_{l=1}^{k} a_l)$ are different labels even if the algebraic sums of the numbers inside labels gives the same value. For example, given the recurrence relation $f_n = 3f_{n-1} + 4f_{n-3}$ with default initial conditions, in this case $a_1 = 3, a_2 = 0, a_3 = 4$, we have the following colored succession rule:

$$\begin{cases} (3)_1 & \\ (3)_1 & \rightsquigarrow (3)_2(3)_1^2 \\ (3)_2 & \rightsquigarrow (7)(3)_1^2 \\ (7) & \rightsquigarrow (7)(3)_1^6 \end{cases}$$

Proof. Let $A_k(x)$ be the generating function of the label $(\sum_{l=1}^{k} a_l)$ related to the succession rule (4). We have:

$$\begin{aligned} A_1(x) &= 1 + (a_1-1)xA_1(x) + (a_1+a_2-1)xA_2(x) + \ldots \\ &\quad \ldots + (a_1+a_2+\ldots+a_k-1)xA_k(x); \\ A_2(x) &= xA_1(x); \\ A_3(x) &= xA_2(x) = x^2A_1(x); \\ &\vdots \\ A_{k-1}(x) &= xA_{k-2}(x) = x^{k-2}A_1(x); \\ A_k(x) &= xA_{k-1}(x) + xA_k(x) = \tfrac{x^{k-1}}{1-x}A_1(x). \end{aligned}$$

Therefore,

$$\begin{aligned} A_1(x) &= 1 + x(a_1-1)A_1(x) + x^2(a_1+a_2-1)A_1(x) + \ldots \\ &\quad \ldots + \tfrac{x^k}{1-x}(a_1+a_2+\ldots+a_k-1)A_1(x), \end{aligned}$$

and we obtain the generating function $A_1(x) = \frac{1-x}{1-a_1x-a_2x^2-\ldots-a_kx^k}$.

At this point we can consider the generating function determined by the succession rule (4) as following:

$$\begin{aligned}\sum_{i=1}^{k} A_i(x) &= A_1(x) + A_2(x) + \ldots + A_{k-1}(x) + A_k(x) = \\ &= A_1(x) + xA_1(x) + \ldots + x^{k-2}A_1(x) + \tfrac{x^{k-1}}{1-x}A_1(x) = \\ &= \tfrac{(1-x)+x(1-x)+\ldots+x^{k-2}(1-x)+x^{k-1}}{1-a_1x-a_2x^2-\ldots-a_kx^k} = \\ &= \tfrac{1}{1-a_1x-a_2x^2-\ldots-a_kx^k}\ .\end{aligned}$$

Following the previous statement, the extended succession rule (3) – determined in the previous section – can be translated into the following succession rule:

$$\begin{cases} (3) \\ (3) \rightsquigarrow (5)(3)^2 \\ (5) \rightsquigarrow (4)(3)^4 \\ (4) \rightsquigarrow (4)(3)^3 \end{cases}$$

We observe that the previously obtained succession rule is an ordinary finite succession rule, but it may happen that the value of the label $(\sum_{l=1}^{i} a_l)$ is negative, for some i with $i \leq k$, then the succession rule (4) contains marked labels.

For example, the recurrence relation $f_n = 5f_{n-1} - 6f_{n-2} + 2f_{n-3}$, with default initial conditions, which defines the sequence 1,5,19,67,231,791,2703, ..., (sequence A035344 in the The On-Line Encyclopedia of Integer Sequences) is equivalent to the following succession rule:

$$\begin{cases} (5) \\ (5) \rightsquigarrow (-1)(5)^4 \\ (-1) \rightsquigarrow (1)(\overline{5})^2 \\ (1) \rightsquigarrow (1) \end{cases}$$

Therefore our next goal is to remove all possible marked labels from the succession rule. We observe that in order to obtain this goal, the recurrence relation $f_n = a_1f_{n-1} + a_2f_{n-2} + \ldots + a_kf_{n-k}$ with default initial conditions needs $a_1 > 0$. We assume that this condition holds throughout the rest of the present section.

In order to furnish a clearer description of our method, we start considering the case $k = 2$.

Proposition 3. *The C-finite recurrence $f_n = a_1f_{n-1} + a_2f_{n-2}$, with default initial conditions, and having $a_1 > 0$, is equivalent to*

$$\begin{cases} (a_1) \\ (a_1) \rightsquigarrow (0)^{q_2}(r_2)(a_1)^{a_1-(q_2+1)} \\ (r_2) \rightsquigarrow \Big((0)^{q_2}(r_2)\Big)^{q_2}(0)^{q_2}(r_2)(a_1)^{r_2-(q_2+1)^2} \end{cases} \tag{5}$$

where, by convention, the label (0) *does not produce any son, and* q_2, r_2 *are defined as follows:*
- if $a_1 + a_2 \leq 0$ *then* $q_2, r_2 > 0$ *such that* $|a_1 + a_2| = q_2 a_1 - r_2$*;*
- otherwise $q_2 = 0$*,* $r_2 = a_1 + a_2$*.*

Proof. We have to distinguish two cases: in the first one $a_1 + a_2 \leq 0$ and in the second one $a_1 + a_2 > 0$.

If $a_1 + a_2 \leq 0$, we have to prove that the generating tree associated to the succession rule (5) is obtained by performing some actions on the generating tree associated to the extended succession rule (6) which is obviously equivalent to the recurrence $f_n = a_1 f_{n-1} + a_2 f_{n-2}$ having $a_1 > 0$ and $a_2 < 0$, with $f_0 = 1$ and $f_h = 0$ for each $h < 0$.

$$\begin{cases} (a_1) \\ (a_1) \overset{1}{\rightsquigarrow} (a_1)^{a_1} \\ (a_1) \overset{2}{\rightsquigarrow} (a_1)^{a_2} \end{cases} \tag{6}$$

The proof consists in eliminating jumps and marked labels at each level of the generating tree associated with succession rule (6), sketched in Figure 2, by modifying the structure of the generating tree, still maintaining f_n nodes at level n, for each n.

Let (a_1) be a label at a given level n. We denote by B_1 the set of a_1 labels (a_1) at level $n+1$ and by B_2 the set of a_2 labels (a_1) at level $n+2$, see Figure 2. We remark that $(a_1)^{a_2} = \underbrace{\overline{(a_1)} \ldots \overline{(a_1)}}_{-a_2}$.

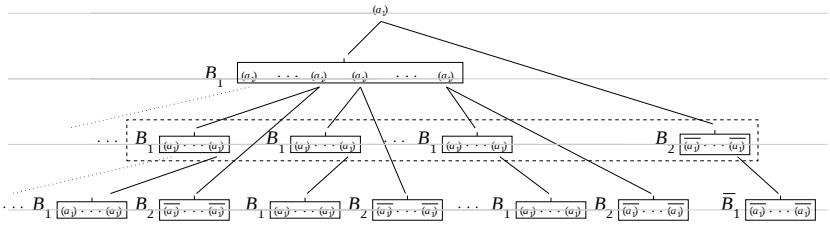

Fig. 2. Step 1

In order to eliminate both jumps and marked labels in B_2 at level 2 produced by the root (a_1) at level 0, we have to consider the set of a_1 labels (a_1) in B_1 at level 2 obtained by (a_1) which lie at level 1. At level 2, each label (a_1) in a given set B_1 kills one and only one marked label $\overline{(a_1)}$ in B_2. At this point $|a_1 + a_2|$ labels $\overline{(a_1)}$ in B_2 always exist at level 2.

In order to eliminate such marked labels we have to consider more than a single set B_1 of label (a_1) at level 2. Let q_2 be a sufficient number of sets B_1 at level 2 able to kill all the labels $\overline{(a_1)}$ in B_2 at level 2. Therefore $|a_1+a_2| = q_2 a_1 - r_2$ with $q_2, r_2 > 0$.

We have the desired number of labels (a_1) at level 2 by setting q_2 labels (a_1) at level 1 equal to (0) and one more label (a_1) to (r_2). Note that the marked labels at level 2 are not generated and the labels (a_1) at level 1 are revised in order to have the right number of labels at level 2, see Figure 3.

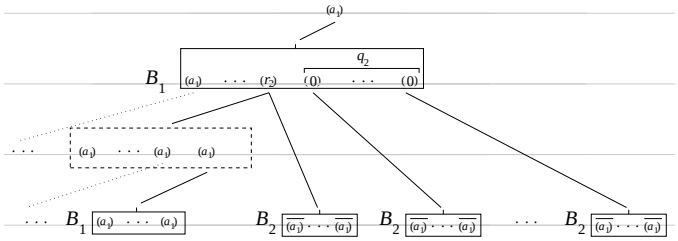

Fig. 3. Step 2

Note that, when a label (a_1) kills a marked label $\overline{(a_1)}$ at a given level n, then the subtree, having such label (a_1) as its root, kills the subtree having $\overline{(a_1)}$ as its root. So, when a label (a_1) of B_1 kills a label $\overline{(a_1)}$ of B_2 at level 2, then the two subtrees having such labels as their roots are eliminated too, see Figure 3.

On the other hand, the q_2+1 sets B_2 at level 3 obtained by the q_2+1 labels at level 1, once labelled with (a_1) and now having value $r_2, 0, \ldots, 0$, respectively, are always present in the tree, see Figure 3. In order to eliminate such undesired marked labels we can only set the production of (r_2). As a set B_2 at a given level is eliminated by using q_2+1 labels at previous level then (r_2) must give $(r_2)\underbrace{(0)\ldots(0)}_{q_2}$ exactly q_2+1 times. This explains the first part of the production rule of the label (r_2) in succession rule (5). Since (r_2) has r_2 sons then the remaining $r_2-(q_2+1)^2$ labels are set to be equal to (a_1) as in the previous case, see Figure 4.

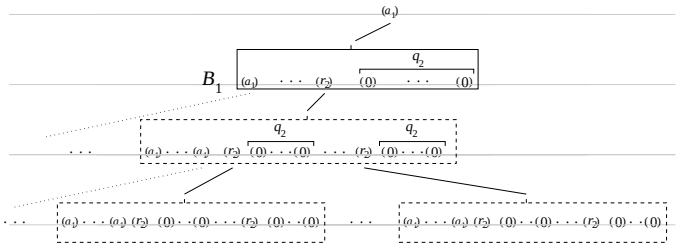

Fig. 4. Step 3

By the way, the modified q_2+1 labels having value $r_2, 0, \ldots, 0$, respectively, at a given level n, produce the labels $\Big((0)^{q_2}(r_2)\Big)^{q_2+1}(a_1)^{r_2-(q_2+1)^2}$ at level $n+1$. Just as obtained for levels 1 and 2, the labels $\Big((0)^{q_2}(r_2)\Big)^{q_2+1}$ automatically annihilate the remaining q_2+1 sets B_2 of marked labels at level $n+2$, once obtained by the modified q_2+1 labels at level n, see Figure 4.

Till now we have modified a portion P of the total generating tree in a way that it does not contain any marked label. Note that, the remaining labels (a_1) will be the roots of subtrees which are all isomorphic to P.

The value f_n defined by the tree associated to the extended succession rule (6), is given by the difference between the number of non-marked and marked labels. The just described algorithm modifies the number of generated non-marked labels and sets to 0 the number of marked ones in a way that f_n is unchanged, for each n, so the succession rule (5) is equivalent to the recurrence $f_n = a_1 f_{n-1} + a_2 f_{n-2}$.

In the case $a_1 + a_2 > 0$ we have marked labels only if $a_2 < 0$. In this case a single set B_1 is sufficient to kill all the marked labels in B_2 at level 2. By the way, both in the case $a_2 < 0$ and $a_2 > 0$ we have that $q_2 = 0$ and $r_2 = a_1 + a_2$, and the succession rule (5) has the same form of the rule (4) which is equivalent to the recurrence $f_n = a_1 f_{n-1} + a_2 f_{n-2}$ having $a_1 > 0$ and $a_2 \in \mathbb{Z}$, with $f_0 = 1$ and $f_h = 0$ for each $h < 0$.

The statement of Proposition 3 can be naturally extended to the general case $k > 2$.

Proposition 4. *The C-finite sequence $\{f_n\}_n$ satisfying $f_n = a_1 f_{n-1} + a_2 f_{n-2} + \ldots + a_k f_{n-k}$, with default initial conditions and $a_1 > 0$ is equivalent to*

$$\begin{cases} (a_1) \\ (a_1) \rightsquigarrow (0)^{q_2}(r_2)(a_1)^{a_1-(q_2+1)} \\ (r_2) \rightsquigarrow \Big((0)^{q_2}(r_2)\Big)^{q_2}(0)^{q_3}(r_3)(a_1)^{r_2-(q_2(q_2+1)+q_3+1)} \\ \quad \vdots \\ (r_i) \rightsquigarrow \Big((0)^{q_2}(r_2)\Big)^{q_i}(0)^{q_{i+1}}(r_{i+1})(a_1)^{r_i-(q_i(q_2+1)+q_{i+1}+1)} \\ \quad \vdots \\ (r_k) \rightsquigarrow \Big((0)^{q_2}(r_2)\Big)^{q_k}(0)^{q_k}(r_k)(a_1)^{r_k-(q_k(q_2+1)+q_k+1)} \end{cases} \tag{7}$$

where the parameters q_i and r_i, with $2 \le i \le k$, can be determined in the following way:
- if $\sum_{l=1}^{i} a_l \le 0$ then $q_i, r_i > 0$ such that $|\sum_{l=1}^{i} a_l| = q_i a_1 - r_i$,
- otherwise $q_i = 0$ and $r_i = \sum_{l=1}^{i} a_l$.

Proof. It is omitted for brevity sake.

We can translate the previously considered recurrence relation $f_n = 5f_{n-1} - 6f_{n-2} + 2f_{n-3}$, with default initial conditions, into the following ordinary succession rule by using Proposition 4:

$$\begin{cases} (5) \\ (5) \rightsquigarrow (0)(4)(5)^3 \\ (4) \rightsquigarrow (0)(4)(1)(5) \\ (1) \rightsquigarrow (1) \end{cases}$$

being $q_2 = 1$, $r_2 = 4$, $q_3 = 0$ and $r_3 = 1$.

3.3 Positivity Condition

The statement of Proposition 4 is indeed a tool to translate C-finite recurrences into finite succession rules. However this property turns out to be effectively applicable only when the labels of the succession rule are all positive, and the reader can easily observe that Proposition 4 does not give us an instrument to test whether this happens or not.

In particular, if the labels of the succession rule are all positive then the terms of the C-finite sequence are all positive. It is then interesting to relate our problem with the so called *positivity problem*, which we have already mentioned in the Introduction.

Corollary 5. *Let us consider the recurrence relation* $f_n = a_1 f_{n-1} + a_2 f_{n-2} + \ldots + a_k f_{n-k}$ *having* $a_1 > 0$ *and* $a_i \in \mathbb{Z}$, $2 \leq i \leq k$, *with* $f_0 = 1$ *and* $f_h = 0$ *for each* $h < 0$. *If*

$$\begin{cases} a_1 - (q_2 + 1) \geq 0 \\ r_2 - (q_2(q_2 + 1) + q_3 + 1) \geq 0 \\ \vdots \\ r_i - (q_i(q_2 + 1) + q_{i+1} + 1) \geq 0 \quad , \quad 3 \leq i \leq k - 1 \\ \vdots \\ r_k - (q_k(q_2 + 1) + q_k + 1) \geq 0 \end{cases} \tag{8}$$

then $f_n > 0$ *for all* n.

As previously mentioned, condition (8) ensures that all the labels of the succession rules equivalent to the given C-finite recurrence are positive, hence all the terms f_n are positive. Thus it can be viewed as a sufficient condition to test the positivity of a given C-finite sequence.

Note that our criterion deals with a subclass of C-finite recurrence relations as it requires that $a_1 > 0$.

4 Conclusions and Further Developments

In this paper we have presented a general method to translate a given C-finite recurrence into an ordinary succession rule and we have proposed a sufficient condition for testing the positivity of a given C-finite sequence.

A further development could take into consideration the average complexity necessary to prove the positivity of a given C-finite sequence.

Afterwards, it should be interesting to develop the study concerning the C-finite recurrences with generic initial conditions in order to examine in depth the potentiality of our method.

References

1. Barcucci, E., DelLungo, A., Frosini, A., Rinaldi, S.: A technology for reverse-engineering a combinatorial problem from a rational generating function. Advances in Applied Mathematics 26(2), 129–153 (2001)
2. Barcucci, E., Lungo, A.D., Pergola, E., Pinzani, R.: Eco: a methodology for the enumeration of combinatorial objects. Journal of Difference Equations and Applications 5, 435–490 (1999)
3. Berstel, J., Mignotte, M.: Deux propriétés décidables des suites récurrentes linéaires. Bulletin de la Société Mathématique de France 104(2), 175–184 (1976)
4. Berstel, J., Reutenauer, C.: Another proof of soittola's theorem. Theoretical Computer Science 393, 196–203 (2008)
5. Bilotta, S., Grazzini, E., Pergola, E., Pinzani, R.: Avoiding cross-bifix-free binary words. ACTA Informatica 50, 157–173 (2013)
6. Chung, F.R.K., Graham, R.L., Hoggatt, V.E., Kleimann, M.: The number of baxter permutations. Journal of Combinatorial Theory Series A 24, 382–394 (1978)
7. Duchi, E., Frosini, A., Pinzani, R., Rinaldi, S.: A note on rational succession rules. Journal of Integer Sequences 6(Article 03.1.7) (2003)
8. Ferrari, L., Pergola, E., Pinzani, R., Rinaldi, S.: An algebraic characterization of the set of succession rules. Theoretical Computer Science 281, 351–367 (2002)
9. Ferrari, L., Pergola, E., Pinzani, R., Rinaldi, S.: Jumping succession rules and their generating functions. Discrete Mathematics 271, 29–50 (2003)
10. Gerhold, S.: Sequences: non-holonomicity and inequalities, ph.D. Thesis
11. Gessel, I.: Rational functions with nonnegative integer coefficients, in the 50th seminaire Lotharingien de Combinatoire, page Domaine Saint-Jacques, March 2003. Unpublished, available at Gessels homepage.
12. Halava, V., Harju, T., Hirvensalo, M.: Positivity of second order linear recurrent sequences. Discrete Applied Mathematics 154, 447–451 (2006)
13. Koutschan, C.: Regular languages and their generating functions: The inverse problem. Theoretical Computer Science 391, 65–74 (2008)
14. Laohakosol, V., Tangsupphathawat, P.: Positivity of third order linear recurrence sequences. Discrete Applied Mathematics 157, 3239–3248 (2009)
15. Perrin, D.: On positive matrices. Theoretical Computer Science 94(2), 357–366 (1992)
16. Salomaa, A., Soittola, M.: Automata-Theoretic Aspects of Formal Power Series. Springer-Verlag (1978)
17. Soittola, M.: Positive rational sequences. Theoretical Computer Science 2(3), 317–322 (1976)
18. Zeilberger, D.: A holonomic systems approach to special functions identities. Journal of Computational and Applied Mathematics 32, 321–368 (1990)