Recurrence-Driven Summations in Automated Deduction

Visa Nummelin, Jasmin Blanchette, Sander Dahmen

Recurrence-Driven Summations in Automated Deduction

Visa Nummelin[1] [0000−0001−7049−6847],
Jasmin Blanchette[1,2] [0000−0002−8367−0936], and
Sander R. Dahmen[1] [0000−0002−0014−0789]

[1] Vrije Universiteit Amsterdam, Amsterdam, the Netherlands
{visa.nummelin,j.c.blanchette,s.r.dahmen}@vu.nl
[2] Ludwig-Maximilians-Universität München, Munich, Germany
jasmin.blanchette@lmu.de

Abstract. Many problems in mathematics and computer science involve summations. We present a procedure that automatically proves equations involving finite summations, inspired by the theory of holonomic sequences. The procedure is designed to be interleaved with the activities of a higher-order automatic theorem prover. It performs an induction and automatically solves the induction step, leaving the base cases to the theorem prover.

1 Introduction

Finite summations—that is, summations $\sum_{i=m}^{n} t_i$ over finitely many terms t_i—are ubiquitous in mathematics and computer science, but they are poorly supported by automatic theorem provers. One reason is that summations are higher-order, whereas most theorem provers are first-order.

In recent years, we have seen the rise of higher-order provers [2,3,16–18]. With these provers, $\sum_{i=m}^{n} t_i$ can be represented as $\mathsf{sum}\ m\ n\ (\lambda i.\ t_i)$; the traditional $\sum$ syntax can be seen as syntactic sugar. But despite the use of heuristics [17, Sect. 4], higher-order provers are ill-equipped to reason inductively. A simple problem such as $\sum_{i=0}^{\mathsf{n}} i = \mathsf{n}(\mathsf{n}+1)/2$ is a formidable challenge for them, even if we include axioms for $+$, $\cdot$, $/$, and $\sum$ together with an induction principle.

In this paper, we introduce a procedure for proving such equations in a higher-order prover. The procedure is triggered by a proof goal of the form $k\sum s + t = u$, possibly with some conditions (Sect. 2). In a refutational prover, the equation would be negated, as $k\sum s + t \neq u$, and would correspond to the negated conjecture, a problem axiom, or some clause derived by the prover.

Our procedure translates facts about summations to linear recurrences. These recurrences have almost the same form as multivariate holonomic sequences [20], which, while not being a prerequisite for reading this paper, strongly inspired our work. Each recurrence is associated with a multivariate sequence—a sequence with one or more indices. In this paper, the word "sequence" generally means "multivariate sequence."

The procedure has three steps.

1. *Initialization* (Sect. 3): Heuristically choose terms in the goal to generalize and perform induction on. Among the problem axioms, select those of a suitable form as initial recurrences for the procedure.
2. *Propagation* (Sect. 4): From the initial recurrences, compute recurrences corresponding to the goal. For $+$, $\cdot$, and $\sum$ expressions occurring in the goal, recurrences are computed from the recurrences of their direct subexpressions.
3. *Induction* (Sect. 5): If the final recurrences for the goal involve only the goal and no other sequences, use them for induction. If they make the difference of successive values of $k\sum s+t-u$ constantly 0, this establishes the induction step. Then reduce the goal to a set of base cases and give these to the prover.

Propagation and induction apply holonomic-style techniques almost as a black box. Initialization connects them to the overall proof search.

For example, to prove $\sum_{i=0}^{\mathsf{n}} i = \mathsf{n}(\mathsf{n}+1)/2$, the procedure would transform the equation into recurrences and find out that the difference $\sum_{i=0}^{\mathsf{n}} i - \mathsf{n}(\mathsf{n}+1)/2$ remains constant as n increases, thereby establishing the induction step. If that difference is constantly 0, we get $\sum_{i=0}^{\mathsf{n}} i = \mathsf{n}(\mathsf{n}+1)/2$; in general, it suffices to prove a number of base cases, which are left to the prover. This example is very simple, but the procedure scales up to more sophisticated problems (Sect. 6). An implementation is under way in the Zipperposition prover [17].

The procedure treats $\sum$ as an interpreted (built-in) symbol. The summation expression evaluates to a value in a commutative group, or a ring if ring multiplication is present. The commutative group or ring gives us $+$, $\cdot$, and $-$. These are also interpreted, as are numerals. Integers, including indices, can multiply group elements. Based on the interpretation, we use the forms $t = u$ and $t - u = 0$ interchangeably.

Compared with Wilf–Zeilberger pairs [19] and other methods (Sect. 7), the main benefit of our procedure is that it goes beyond holonomic sequences and supports both uninterpreted functions and an infinite number of base cases. Our procedure is widely applicable and may help prove not only difficult summations in a restrictive form but also easier summations in a more general form, which is useful in a general-purpose theorem prover. At the heart of our work is the novel combination of techniques from superposition and holonomic sequences, which is visible both in the prover integration (Sect. 2) and in the computation of so-called excess terms (Sect. 4). We refer to our technical report [14] for more details.

2 Inference Rule

Our procedure can be integrated into a theorem prover, where it takes the form of an inference rule that complements the prover's existing rules. Our technical report discusses an integration with satisfiability modulo theories (SMT) and tableaux; here, we present a rule for superposition:

$$\frac{C_1 \;\cdots\; C_l \quad C' \vee t[\vec{s}] \neq 0}{D \vee C' \vee \bigvee_{\vec{b} \in B} t[\vec{b}] \neq 0}\ \textsc{Summation}$$

These side conditions apply:

- $t[\vec{s}]$ is an expression that can be brought into the general form $k\sum_{i=m}^{n} t' + t''$;
- the procedure selects, generalizes, and performs an induction on the subterms $\vec{s}$ of t (Sect. 3);
- the procedure succeeds at proving the induction step based on initial recurrences derived from $C_1, \ldots, C_l$ (Sect. 3) and their propagation (Sect. 4);
- the procedure identifies B as the finite set of base cases of the induction, where each case is a vector $\vec{b}$ of terms of the same length as $\vec{s}$ (Sect. 5); and
- the subclause D captures potential conditions determined by the procedure.

The intuition behind the rule is that the conclusion should be easier to refute than the rightmost premise. As for the premises $C_1, \ldots, C_l$, they can contain useful information about $\vec{s}$, often about bounds.

3 Initialization

The first step of our procedure is to recognize the structure of recurrences. Variables on which we can perform induction appear as Skolem constants in the negated goal. Further opportunities for induction can be created by generalizing complex terms. Also as part of this step, we must choose which terms represent (multivariate) sequences and which clauses represent their recurrences.

Theory Detection. We require the necessary theory of summation to be predefined. Specifically, this refers to the inductive theory of integers, axioms for commutative groups (including multiplication by integers), and the definition of summation from 0 by $\sum_{n=0}^{-1} f_n = 0$ and $\sum_{n=0}^{m+1} f_n = \sum_{n=0}^{m} f_n + f_{m+1}$ even for negative $m \in \mathbb{Z}$. Other finite intervals than $[0, m]$ are expressible as differences.

Ring multiplication may be absent, so we do not take it as predefined. Instead, we search candidate binary operators from the negated goal. For each candidate, we can try to prove left and right distributivity by syntactically looking for that axiom or by running another instance of the prover. Distributivity is the only necessary property to apply the procedure, but associativity, commutativity, and the unit element can also be used in simplifications.

Term Generalization. Term generalization transforms Skolem constants or complex terms into variables and then performs an induction on the variables. We propose a straightforward heuristic: For each nonnumeral subterm s of type $\mathbb{Z}$ occurring in the negated goal, generalize s if s stays variable-free even after recursively applying this heuristic on the proper subterms of s itself. For example, in the following variable-free integer terms, the underlined subterms would be generalized: $\underline{\mathsf{a}}$, 123, $\underline{\mathsf{f}\ 0\ 2}$, $2\mathsf{f}\ \underline{(\mathsf{g}\ (-1))}\ (-3\underline{\mathsf{a}})$, $\mathsf{f}\ 1\ (\mathsf{g}\ (\underline{\mathsf{a}} + 1))$, $\mathsf{f}\ (\mathsf{g}\ \underline{\mathsf{a}})\ 7\underline{\mathsf{a}}$.

Let $\vec{s} = (s_1, \ldots, s_d)$ be the subterms chosen for generalization. Then, based on the negated goal $C' \vee t[\vec{s}] \neq 0$ (as in the SUMMATION rule), generalization sets up the goal $\forall \vec{n} \in N.\ t[\vec{n}] = 0$ where $N \subseteq \mathbb{Z}^d$ collects the bounds of $\vec{s}$ (often $N = \mathbb{N}^d$). We try to prove this goal up to base cases and other mild conditions.

The generalization makes it possible to use induction to prove that the goal sequence term $t[\vec{n}]$—a function of $\vec{n}$—equals zero on N. We try to prove the generalized goal assuming $\neg C'$ and some extra conditions E such as the base cases of the induction. Then, instantiating $\vec{n} := \vec{s}$, we conclude $C' \vee \neg E \vee t[\vec{s}] = 0$. This, together with the negated goal $C' \vee t[\vec{s}] \neq 0$, implies a conclusion of the form $C' \vee \neg E$ for the SUMMATION rule. Note that C' is not generalized.

The set N embodies knowledge about $\vec{s}$ that we find among existing clauses $C_1, \ldots, C_l$ and the condition $\neg C'$. The free variables of $\neg C'$ are interpreted as constants, and they can also occur in $\vec{s}$. For example, assume that $\vec{s} = \mathsf{f}\ s'$ and $\vec{n} = n$ and that the generalized goal contains the factorial $n!$. Its recurrence must be in a conditional clause—e.g., $(m+1)! = (m+1)m! \vee 0 \not\leq m$. To use this recurrence for $n!$, we need $n \geq 0$, which we can ensure using N if we find a bounding clause $\mathsf{f}\ s' \geq 0$ or its generalization such as $\mathsf{f}\ m \geq 0$ where m is a free variable. The more we know about $\vec{s}$, the more recurrences we can get. At the same time, N must allow induction, so we keep it convex by considering only coordinatewise bounds of $\vec{s}$.

Form of sequence terms. Sequence terms are terms of the underlying higher-order logic that our procedure can work with. From their structure, we distinguish (pointwise) addition and multiplication, summation, and affine substitution. This gives a first-order grammar to express the sequence terms.

Definition 1. *Sequence terms* on a ring A are inductively defined as follows. The logic's terms of type A with distinguished integer variables $\vec{n}$ are sequence terms. If $f_{\vec{n}}$ and $g_{\vec{n}}$ are sequence terms with d variables $\vec{n}$, then so are $f_{\vec{n}} + g_{\vec{n}}$, $f_{\vec{n}} \cdot g_{\vec{n}}$, $\sum_{i=0}^{\vec{c} \bullet \vec{n} + a} f_{\{n_j \mapsto i\}\vec{n}}$, and $\sigma f_{\vec{n}} = f_{\sigma\vec{n}}$ where $\vec{c}$ is a vector, a is an integer, $\vec{c} \bullet \vec{n} = c_1 n_1 + \cdots + c_d n_d$, and σ is an affine substitution (meaning $\sigma\vec{m} = q\vec{m} + \vec{b}$ for a matrix q and a vector $\vec{b}$); a, the entries of $\vec{c}$, and the entries of σ (meaning the entries of q and $\vec{b}$) must be numerals.

Remark 2. In Definition 1 and in the sequel, a commutative group can be used instead of a ring if ring multiplication is absent. In this case, all formulas involving ring multiplication (e.g., $f_{\vec{n}} \cdot g_{\vec{n}}$) should be ignored.

We view sequence terms as functions $\mathbb{Z}^d \to A$. We then write the sequence terms from the definition compactly as $f + g$, $f \cdot g$, $\sum_j^a f$, and σf, and call $a = \vec{c} \bullet \vec{n} + d$ an affine variable sum. Moreover, since $\cdot$, $\sum_j^a$, and σ all distribute over $+$, we can write any sequence term as $c_1 f^1 + \cdots + c_k f^k$ where the coefficients c_j are numerals and the sequence terms f^j are distinct and do not contain $+$. Finally, we forbid variable shadowing: $\sum_{n_j=0}^{a}$ binds n_j, and while $\sum_j^a \sum_j^b g$ and $\sum_j^{n_j} g$ and other references to n_j outside $\sum_j^a$ are syntactically valid, we avoid such forms by renaming them during encoding and never reintroducing them.

Choice of Initial Recurrences. Semantically, the recurrences we look for are multivariate heterogeneous linear finite-fixed-step equations with polynomial coefficients. An archetypical example is

$$(n^2+1) f_{n+2,m+1} + m f_{n+1,m} = nm f_{n,m+1} - 2h_{n,m} + (m-n) h_{n,m+1} + 1 \quad (1)$$

Here, the sequences $f, h, 1$ are bivariate, and the sequence indices are all of the form $n + k$ or $m + k$ for numerals $k \in \mathbb{Z}$, amounting to finite fixed steps.

The general form is $0 = P_1 g^1 + \cdots + P_k g^k = \vec{P} \bullet \vec{g}$ where $\vec{g} = (g^1, ..., g^k)$ is a tuple of sequence terms and $\vec{P}$ is a tuple of operator polynomials as defined below. If $k = 1$, we have a homogeneous recurrence of g^1; otherwise, it is heterogeneous.

Definition 3. *Operator polynomials* are a $\mathbb{Z}$-algebra with composition as product (meaning closed under addition, composition, and integer multiplication) spanned by the multiplier and shift operators:

- The *multiplier* operator M_j of index j multiplies a multivariate sequence f by the variable n_j of index j: $(M_j f)_{\vec{n}} = n_j f_{\vec{n}}$.
- The *shift* operator $S_j = \{n_j \mapsto n_j + 1\}$ of index j increments the variable n_j of index j of a multivariate sequence f by one: $(S_j f)_{\vec{n}} = f_{\{n_j \mapsto n_j + 1\}\vec{n}}$.

With d index variables, the operator polynomials look like ordinary polynomials $\mathbb{Z}[M_1, ..., M_d, S_1, ..., S_d]$, but the composition product is noncommutative since $S_i M_i = M_i S_i + S_i$ for all $i = 1, \ldots, d$ (a derivation of which is given in the next section directly above equation (2)). As an example of expressing recurrences in terms of operator polynomials, consider the previous archetypical recurrence (1). Taking n as the first and m as the second variable, the recurrence reads

$$((M_1^2 + 1)S_1^2 S_2 + M_2 S_1 - M_1 M_2 S_2) \cdot f + (2 - (M_2 - M_1)S_2) \cdot h + (-1) \cdot 1 = 0$$

Remark 4. The expression $\vec{P} \bullet \vec{g}$ identifying a recurrence is itself a sequence term. It suffices to observe that if f is a sequence term, then so are the substitution $S_j f$ and the product $M_j f = (\vec{n} \mapsto n_j) \cdot f$ with the projection sequence term $\vec{n} \mapsto n_j$.

As sketched in Sect. 1, we must select some of the problem axioms as initial recurrences for the procedure. This is accomplished as follows. Let there be an edge between two axioms of the form $C \vee s = t$ (where C may be empty) if they both contain a top-level occurrence of the same sequence g, i.e., an occurrence of g that is not nested inside an uninterpreted function symbol. The axioms then form a graph. We take as initial recurrences the connected component of the generalized goal.

By a sequence g, we mean the $\mathsf{f}\ \vec{a}$ part of a term of the form $\mathsf{f}\ \vec{a}\ \vec{n}$ where f is an uninterpreted function symbol, $\vec{a}$ is a tuple of variable-free terms, and $\vec{n}$ is a nonempty tuple of integer variables or affine (i.e., linear term + constant term) combinations of them. The tuples $\vec{a}$ and $\vec{n}$ may in general be interleaved.

In other contexts, an analogous step is known as lemma filtering or premise selection [4, Sect. 2]. Clutter from irrelevant facts is less of an issue in the context of our procedure because it can use only linear recurrences. Beyond this, our simple heuristic does nothing to avoid clutter.

What should we do about conditions such as C in $C \vee \mathsf{f}\ \vec{a}\ \vec{n} = t$? We could forbid them and work only with unit equations such as $\mathsf{f}\ \vec{a}\ \vec{n} = t$. We could collect

them and put them in the D component of the SUMMATION rule's conclusion. Or we could attempt to prove them when the initial recurrences are selected. In our ongoing implementation, we chose the first option, but what the best option is remains an open question.

4 Propagation

Holonomic sequences can be defined by homogeneous recurrences with polynomial coefficients and finitely many base cases. They are closed under the four operations that build sequence terms $(+, \cdot, \sum_j^a, \sigma)$, which especially makes their equality decidable [20]. The closure is realized by four procedures to derive recurrences of a sequence term from the recurrences of its immediate subterms, which we call *propagation*. We can propagate independently of the base cases and hence work on nonholonomic sequence terms [6]. Although we expect the holonomic subcase to be decidable in our setting, in general decidable equality is lost. Additionally, unlike in the holonomic setting, we allow heterogeneous recurrences. We will build this into our noncommutative Gröbner basis setup that is used in the propagation procedures.

Gröbner Bases of Recurrence Operators. A (generalized) Gröbner basis is a certain well-behaved generating set of a left-ideal of (possibly noncommutative) polynomials. Equivalently, we will view it as a system of polynomial equations that is complete for rewriting. Given a polynomial equation $P = 0$, for every monomial M we get a rewrite rule as follows. Decompose MP as $MP = L + R$ where L is the leading monomial of MP w.r.t. a fixed monomial ordering times its coefficient. Then $L = -R$ gives rise to a rewrite rule $L \to -R$. A system of equations is complete for rewriting if every one of its consequences can be proved via rewriting by these rules.

Example 5. The system $\{ab^2 = a + b, a^2b = a + 1\}$ does not prove its consequence $a^2 = b$ by rewriting. (We can see that $a^2 = b$ is a consequence by multiplying the first equation by a and the second equation by b and then by subtracting the two equations.) In the other direction, the system's Gröbner basis $\{a^2 = b, b^2 = a + 1\}$ does give rewrite proofs $ab^2 \xrightarrow[b^2=a+1]{} a^2 + a \xrightarrow[a^2=b]{} b + a$ and $a^2b \xrightarrow[a^2=b]{} b^2 \xrightarrow[b^2=a+1]{} a + 1$.

A theory of Gröbner bases exists for various polynomial algebras [10]. In our setting, a sufficient requirement is that all indeterminates X, Y commute up to lower-order terms: $XY - YX \in \mathbb{Z}X + \mathbb{Z}Y + \mathbb{Z}$. The operator polynomials of Definition 3 fall into this category with the natural choice of taking all multiplier and shift operators as indeterminates. Indeed, for any sequence term f, we have the noncommutation relations

$$(S_j M_j f)_{\vec{n}} = (S_j(\vec{n} \mapsto n_j f_{\vec{n}}))_{\vec{n}} = (n_j + 1)(S_j f)_{\vec{n}} = ((M_j S_j + S_j) f)_{\vec{n}}$$

and all other pairs of multipliers and shifts commute exactly. That is:

$$S_i M_j = M_j S_i + \delta_{i,j} S_i \qquad S_i S_j = S_j S_i \qquad M_i M_j = M_j M_i \tag{2}$$

for all i and j, where $\delta_{i,j}$ equals 1 if $i = j$ and 0 otherwise. When we consider a formal polynomial algebra (necessary to perform Gröbner basis computations), we will usually mean polynomials with integer coefficients and indeterminates $M_1, M_2, \ldots, S_1, S_2, \ldots$ satisfying (2). Exceptionally, when we use propagation to substitution, we will consider compositions of shifts formally as further individual indeterminates, as explained above Procedure 12. Apart from this exceptional setting, we fix a choice of monomials as follows.

Definition 6. In our setting, a *monomial* is a polynomial of the form $M_1^{x_1} \cdots M_d^{x_d} S_1^{y_1} \cdots S_d^{y_d}$ where the exponents $x_j, y_j \in \mathbb{N}$ are numerals.

Due to the (non)commutation relations (2), polynomials can be written as sums of monomials times their integer coefficients. This makes working with these noncommutative polynomials similar to working with commutative ones. A major difference is that monomials are not closed under product, as illustrated by $S_1 \cdot M_1 = M_1 S_1 + S_1$. This complicates the definition of monomial order below, which in turn defines how to interpret a polynomial equation as a rewrite rule.

Definition 7. A *monomial order* $\preccurlyeq$ is a well-founded total order on monomials such that for all monomials A, B, C, if $A \preccurlyeq B$, then the leading monomial of CA is $\preccurlyeq$-smaller than the leading monomial of CB; here, the *leading monomial* of a nonzero polynomial P means the $\preccurlyeq$-largest monomial occurring in P.

Buchberger's algorithm to compute Gröbner bases (also in a noncommutative context) is similar to saturation-based theorem proving. It repeatedly derives from polynomial equations $P = 0$ and $R = 0$ new equations $AP - BR = 0$ where coefficient–monomial products A, B make the leading monomials of AP and BR cancel. It suffices to take A, B with smallest total degree and coprime coefficient. A and B play a similar role to the most general unifier in superposition. Since S_j is semantically bijective, we can and always do cancel it, replacing $S_j R = 0$ by $R = 0$. This modified completion into a Gröbner basis always terminates. The standard termination proof reduces to applying noetherianity of commutative polynomials over $\mathbb{Z}$ or Dickson's lemma [10].

A single operator polynomial P_1 perfectly encodes a linear homogeneous recurrence $0 = P_1 g$ of a sequence term g. However, we allow any heterogeneous recurrence of the form $0 = \vec{P} \bullet \vec{f} = P_1 f^1 + \cdots + P_k f^k$ where $\vec{f} = (f^1, \ldots, f^k)$ is an arbitrary tuple of different sequence terms. We can encode this by a single operator polynomial for the duration of one Gröbner basis computation as follows. Let $\vec{f}$ enumerate exactly once all the sequence terms needed to express the current recurrences with the help of operator polynomials. Let $\vec{f}$ depend on d variables. For each f^j, we consider a shift $F_j := S_{d+j}$ w.r.t. a so far unused variable. Then the operator polynomial $\vec{P} \bullet \vec{F}$ encodes $0 = \vec{P} \bullet \vec{f}$.

This encoding does not respect the semantics of operator polynomials; to recover it, we must apply the substitution $\{\vec{F} \mapsto \vec{f}\}$. However, products such as $F_1 F_2$ remain uninterpretable even with ring-valued sequences because the operator product—function composition—is different from multiplication of f^j's. Hence, we will simply discard uninterpretable polynomials after the Gröbner basis computation. Moreover, from now on, we will freely write f^j for F_j.

Definition 8. Let $X_1, \ldots, X_n$ be an enumeration of all multiplier and shift indeterminates. An $(X_1, \ldots, X_k)$-*elimination order* is a monomial order such that $X_j \succ X_{k+1}^{a_{k+1}} \cdots X_n^{a_n}$ for all indices $j \leq k$ and all exponents $a_{k+1}, \ldots, a_n \in \mathbb{N}$.

Our default choice for the order is to compare total degree in $X_1, ..., X_k$ and break ties using the total degree reverse lexicographical order [7, Chapter 2 §2].

Procedure 9. *Eliminating* indeterminates $X_1, \ldots, X_k$ from a finite system of equations E means computing a Gröbner basis G of E w.r.t. an $(X_1, \ldots, X_k)$-elimination order and then discarding all polynomials from G that contain any of $X_1, \ldots, X_k$ or that are not linear in the indeterminates encoding sequence terms. (As mentioned above, during the Gröbner basis computation, whenever we derive a polynomial $S_j R$, we replace it by R.)

While in principle any Gröbner basis would suffice for elimination, our default choice is to compute the reduced Gröbner basis (i.e., the fully simplified one). The nonlinear polynomials can be discarded as soon as they are derived during the Gröbner basis computation instead of only at the end. Recurrence equations produced by elimination are logical consequences of the input equations, as we explain in our technical report.

Despite the formally equivalent roles of all sequence terms f^i in the recurrence $0 = \vec{P} \bullet \vec{f}$, we associate with every recurrence a sequence term f^j. It is often convenient to write such a recurrence of f^j as $P_j f^j + e = 0$ where the *excess terms* $e = \vec{P} \bullet \vec{f} - P_j f^j$ contain all sequence terms f^i except f^j. The choice of f^j among $\vec{f}$ will be determined by the definition of excess terms (Definition 18). However, this choice remains irrelevant for the individual propagation steps, described below. We adapt these steps from the four closure properties of holonomic sequences by carrying excess terms along.

Propagation to Addition. Let us start with addition of sequence terms.

Procedure 10. Let f and g be sequence terms, and let h be the formal name of their addition $f + g$. The associated recurrences F of f and G of g are propagated to those of h by eliminating f and g from $F \cup G \cup \{h = f + g\}$. (By Procedure 9, this involves computing a Gröbner basis for these equations and then discarding the equations containing f or g as well as the corresponding nonlinear terms.)

Actually, the same propagation technique works if $f + g$ is replaced by any expression in the general recurrence format $\vec{P} \bullet \vec{l}$ (a dot product of operator polynomials $\vec{P}$ and sequence terms $\vec{l}$). The key is that the defining equation $h = \vec{P} \bullet \vec{l}$ is again a linear recurrence. Such propagations could also be done by iterating more primitive propagations.

Example 11. Consider the goal $\sum_{j=0}^{n} a_j = g_n + a_0$ given $g_0 = 0$ and $g_{n+2} = g_n + a_{n+1} + a_{n+2}$ for all $n \in \mathbb{N}$. The defining recurrence of g can be written using the operator polynomials as $S_1^2 g = g + S_1 a + S_1^2 a$. The defining recurrence of the sum $f_n := \sum_{j=0}^{n} a_j$ is $S_1 f = f + S_1 a$. We must prove that $h_n := g_n + a_0 - f_n$

is 0. To achieve this, we propagate recurrences to h using the elimination procedure described above (Procedure 9) and the total-degree-based (f, g)-elimination order with $f \prec g$. Leading monomials are shown in bold:

$$\begin{array}{lll}
 & 0 = \boldsymbol{S_1^2 g} - g - S_1 a - S_1^2 a & \text{recurrence of } g \\
-S_1^2 & 0 = \boldsymbol{g} + a_0 - f - h & \text{definition of } h \\
\hline
 & 0 = -g - S_1 a - S_1^2 a - a_0 + \boldsymbol{S_1^2 f} + S_1^2 h & \\
-S_1 & 0 = \boldsymbol{S_1 f} - f - S_1 a & \text{recurrence of } f \\
\hline
 & 0 = -g - S_1 a - a_0 + S_1^2 h + \boldsymbol{S_1 f} & \\
- & 0 = \boldsymbol{S_1 f} - f - S_1 a & \text{recurrence of } f \\
\hline
 & 0 = -\boldsymbol{g} - a_0 + S_1^2 h + f & \\
+ & 0 = \boldsymbol{g} + a_0 - f - h & \text{definition of } h \\
\hline
 & 0 = \boldsymbol{S_1^2 h} - h &
\end{array}$$

In this example, $h_{n+2} - h_n = 0$ is the only recurrence that does not contain f and g, so we discard the rest of the Gröbner basis calculation. Since $h_{n+2} - h_n = 0$ contains only the sequence h, we can use it to prove the induction step (of size 2) of a proof of $\forall n.\ h_n = 0$. We are then left with the two base cases $h_0 = 0$ and $h_1 = 0$, which the SUMMATION inference would include in its conclusion without auxiliary symbols (f and h) as $\sum_{j=0}^{0} a_j \neq g_0 + a_0 \vee \sum_{j=0}^{1} a_j \neq g_1 + a_0$.

Propagation to Substitution. Consider a numeral matrix $a = [a_{kj}]_{kj} \in \mathbb{Z}^{d \times D}$ and a vector $\vec{b} \in \mathbb{Z}^d$. They characterize an affine substitution $\sigma = \{\vec{n} \mapsto a\vec{n} + \vec{b}\}$ $= \{n_k \mapsto \sum_{j=1}^{D} a_{kj} n_j + b_k \mid 1 \leq k \leq d\}$. As an operator on sequences, σ performs an affine change of variables: $(\sigma f)_{\vec{n}} = f_{a\vec{n}+\vec{b}}$.

Clearly, any recurrence $Pf = 0$ of f implies $\sigma P f = 0$. Moreover, if $\sigma P = P'\sigma$, then $P'\sigma f = 0$ gives a recurrence of σf. Finding such a P' for a general P can be reduced to finding an operator polynomial P'_X satisfying $\sigma X = P'_X \sigma$ for every indeterminant X. This amounts to pushing all indeterminates X leftwards. For multipliers, we have $\sigma(M_1, \ldots, M_d) = (a(M_1, \ldots, M_D) + \vec{b})\sigma$. In contrast, shifts are easily pushed only rightwards—namely, $S_j \sigma = \sigma S_1^{a_{1j}} \cdots S_d^{a_{dj}}$. Consequently, the recurrences of f must be first expressed in terms of the composite shifts $\mathbb{S}_j := S_1^{a_{1j}} \cdots S_d^{a_{dj}}$. As operators, these satisfy the (non)commutation relations

$$\mathbb{S}_j M_k = (M_k + a_{kj})\mathbb{S}_j \qquad \mathbb{S}_i \mathbb{S}_j = \mathbb{S}_j \mathbb{S}_i \qquad \mathbb{S}_i S_j = S_j \mathbb{S}_i \tag{3}$$

This makes the $\mathbb{S}_j$'s suitable as indeterminates in Gröbner basis computations.

Accordingly, for propagation to substitution, we enlarge our formal polynomial algebra to also contain the indeterminates $\mathbb{S}_1, \mathbb{S}_2, \ldots$ satisfying the relations (3), while also keeping (2). We note that, as operators, the indeterminates further satisfy (essentially by definition) the relations

$$\mathbb{S}_j \prod_{k:\, a_{kj} < 0} S_k^{|a_{kj}|} = \prod_{k:\, a_{kj} > 0} S_k^{a_{kj}} \quad \text{for } j \in \{1, \ldots, D\} \tag{4}$$

We add these new relations to the system of recurrence equations of which we compute the Gröbner basis. Finally, we extend our notion of *monomial* from Definition 6 to mean any polynomial of the form $M_1^{x_1} \cdots M_d^{x_d} S_1^{y_1} \cdots S_d^{y_d} \mathbb{S}_1^{z_1} \cdots \mathbb{S}_D^{z_D}$ where the exponents $x_j, y_j, z_j \in \mathbb{N}$ are numerals.

Procedure 12. Recurrences of a sequence term f are propagated to its affine substitution $(\sigma f)_{\vec{n}} = f_{a\vec{n}+\vec{b}}$ as follows. Eliminate each S_k from the system of polynomial equations containing both the recurrences of f and the relations (4). Every resulting recurrence $P(\vec{M}, \vec{\mathbb{S}})f + e = 0$ implies a recurrence $P(a\vec{M} + \vec{b}, \vec{S})\sigma f + \sigma e = 0$ of σf where we have collected the indeterminates into vectors and where e are excess terms that do not contain f.

Example 13. Consider $\sum_{n_1=0}^{n_2} \binom{n_1}{n_2-n_1} = F_{n_2+1}$ where the Fibonacci numbers are defined by $F_1 = F_2 = 1$ and $(S_1^2 - S_1 - 1)F = 0$. For the binomial coefficient $\binom{\cdot}{n_1,n_2} = \binom{n_1}{n_2} = \frac{n_1!}{n_2!(n_1-n_2)!}$, the recurrence from Pascal's triangle reads as $(S_1S_2 - S_2 - 1)\binom{\cdot}{\cdot} = 0$ and extends $\binom{n_1}{n_2}$ from $0 \le n_2 \le n_1$ to all $n_2 \in \mathbb{Z}$ and $n_1 \in \mathbb{N}$. Moreover, we have $\binom{n_1}{n_2} = \frac{n_1}{n_2}\binom{n_1-1}{n_2-1}$—i.e., $((M_2+1)S_1S_2 - M_1 - 1)\binom{\cdot}{\cdot} = 0$. We want to propagate these recurrences to the substitution $\sigma = \{n_1 \mapsto n_1, n_2 \mapsto n_2 - n_1\}$. We have $S_1\sigma = \sigma S_1 S_2^{-1}$ and $S_2\sigma = \sigma S_2$. So we introduce for $S_1S_2^{-1}$ and S_2 the indeterminates $\mathbb{S}_1$ and $\mathbb{S}_2$ whose characterizing recurrences (4) read

$$(\mathbb{S}_1S_2 - S_1)\tbinom{\cdot}{\cdot} = 0 \quad \text{(i)} \qquad\qquad (\mathbb{S}_2 - S_2)\tbinom{\cdot}{\cdot} = 0 \quad \text{(ii)}$$

Next, we eliminate S_1, S_2 in favor of $\mathbb{S}_1, \mathbb{S}_2$. Here, (ii) immediately rewrites every S_2 to $\mathbb{S}_2$ and then (i) becomes $(-S_1 + \mathbb{S}_1\mathbb{S}_2)\binom{\cdot}{\cdot} = 0$, which rewrites every S_1. The remaining steps to complete a Gröbner basis w.r.t. some total-degree order are irrelevant for what we want to illustrate. We factor the result for readability:

$$\begin{aligned}
(-S_1 + \mathbb{S}_1\mathbb{S}_2)\tbinom{\cdot}{\cdot} &= 0 & (-S_2 + \mathbb{S}_2)\tbinom{\cdot}{\cdot} &= 0\\
(\mathbb{S}_1\mathbb{S}_2^2 - \mathbb{S}_2 - 1)\tbinom{\cdot}{\cdot} &= 0 & ((M_2+1)\mathbb{S}_2 - M_1 + M_2)\tbinom{\cdot}{\cdot} &= 0\\
& & ((M_1+1)\mathbb{S}_1\mathbb{S}_2 - (M_1 - M_2 + 2)\mathbb{S}_1 - M_1 - 1)\tbinom{\cdot}{\cdot} &= 0\\
& & ((M_1 - M_2 + 1)(M_1 - M_2 + 2)\mathbb{S}_1 - M_1M_2 - M_2)\tbinom{\cdot}{\cdot} &= 0
\end{aligned}$$

Now σ maps the lowest four recurrences to recurrences of $f_{n_1,n_2} = \binom{n_1}{n_2-n_1}$ below:

$$\begin{aligned}
(S_1S_2^2 - S_2 - 1)f &= 0 & ((M_2 - M_1 + 1)S_2 - 2M_1 + M_2)f &= 0\\
& & ((M_1+1)S_1S_2 - (2M_1 - M_2 + 2)S_1 - M_1 - 1)f &= 0\\
& & ((2M_1 - M_2 + 1)(2M_1 - M_2 + 2)S_1 - (M_1+1)(M_2 - M_1))f &= 0
\end{aligned}$$

The next step is to propagate to the summation. We postpone it to Example 16.

Propagation to Product. Let $\cdot$ be ring multiplication or more generally a group bihomomorphism. If the sequence terms f and g depend on disjoint sets of variables, recurrences of $fg = f \cdot g$ are essentially a union of recurrences of f and g. Namely, let $Pf + e = 0$ be any recurrence of f where P is an

operator polynomial on the variables of f and the excess terms e do not contain f. Then $P(fg) + eg = 0$ because g is effectively a constant to P, and similarly for recurrences of g. With the help of this special case, propagation to product can be reduced to propagation to substitution, as explained below.

Procedure 14. Let f and g be sequence terms parameterized by the variables $\vec{n} = (n_j)_{j=1}^d$. Let $\vec{m} = (n_{j+d})_{j=1}^d$ be a tuple of fresh variables. The recurrences of f and g are propagated to their pointwise product fg in two steps. First, the recurrences of the variable-disjoint product $f_{\vec{n}}g_{\vec{m}}$ are the union of the recurrences of $f_{\vec{n}}$ multiplied on the right by $g_{\vec{m}}$ and of those of $g_{\vec{m}}$ multiplied on the left by $f_{\vec{n}}$. Then the recurrences of $f_{\vec{n}}g_{\vec{n}} = \{\vec{m} \mapsto \vec{n}\}(f_{\vec{n}}g_{\vec{m}})$ are found by propagating to substitution using Procedure 12.

Propagation to Summation. We finally consider the summations $\sum_{n_1=0}^{n_2} f_{\vec{n}}$. We can assume that the variables are numbered so that the sum acts on the first two. Similarly to above, we consider the consequence $\sum_{n_1=0}^{n_2} Pf_{\vec{n}} + \sum_{n_1=0}^{n_2} e_{\vec{n}} = 0$ of a recurrence $Pf + e = 0$ of the sequence term f where P is an operator polynomial and e are excess terms. We want to find an operator polynomial P' such that $\sum_{n_1=0}^{n_2} P$ becomes $P' \sum_{n_1=0}^{n_2}$ up to excess terms. Like for substitutions, finding such a P' for P can be reduced to finding an operator polynomial P'_X satisfying $\sum_{n_1=0}^{n_2} X = P'_X \sum_{n_1=0}^{n_2}$ up to excess terms for every indeterminant X. The result will be a recurrence $P' \sum_{n_1=0}^{n_2} f_{\vec{n}} + e' = 0$ of $\sum_{n_1=0}^{n_2} f_{\vec{n}}$.

Procedure 15. Recurrences of a sequence term f are propagated to its sum $\sum_{n_1=0}^{n_2} f_{\vec{n}}$ as follows. First, eliminate multipliers M_1 from all recurrences of f. Every resulting recurrence $Pf + e = 0$ implies $\sum_{n_1=0}^{n_2} Pf_{\vec{n}} + \sum_{n_1=0}^{n_2} e_{\vec{n}} = 0$. Here, P is an operator polynomial that does not contain M_1, and the excess terms e do not contain f. Next, each of these recurrences is rewritten into the form $P' \sum_{n_1=0}^{n_2} f_{\vec{n}} + E_0 + E_{n_2} + \sum_{n_1=0}^{n_2} e_{\vec{n}} = 0$ where P' is an operator polynomial and the E_m's are part of excess terms built by applying some operator polynomials and the substitution $\{n_1 \mapsto m\}$ to f. This is achieved by commuting $\sum_{n_1=0}^{n_2}$ with indeterminates other than S_1 and S_2. These two indeterminates are instead handled by

$$\sum_{n_1=0}^{n_2} S_1 g_{\vec{n}} = \sum_{n_1=0}^{n_2} g_{\vec{n}} + \{n_1 \mapsto n_2 + 1\} g_{\vec{n}} - \{n_1 \mapsto 0\} g_{\vec{n}}$$
$$\sum_{n_1=0}^{n_2} S_2 g_{\vec{n}} = S_2 \sum_{n_1=0}^{n_2} g_{\vec{n}} - S_2 \{n_1 \mapsto n_2\} g_{\vec{n}}$$

Example 16. Let us continue the proof of $\sum_{n_1=0}^{n_2} \binom{n_1}{n_2-n_1} = F_{n_2+1}$ from Example 13. There we found for the summand $f_{n_1,n_2} = \binom{n_1}{n_2-n_1}$ a recurrence $(S_1 S_2^2 - S_2 - 1)f = 0$. It is actually the only recurrence after eliminating M_1 as a first step of propagation to summation. Next, we set S_1 to 1 using a telescoping identity:

$$\sum_{n_1=0}^{n_2} S_1 S_2^2 f = \sum_{n_1=0}^{n_2} S_2^2 f + \{n_1 \mapsto n_2\} S_1 S_2^2 f - \{n_1 \mapsto 0\} S_2^2 f$$

Then we push the remaining shifts S_2 leftwards:

$$\begin{aligned}\sum_{n_1=0}^{n_2} (S_2^2 - S_2) f &= S_2 \sum_{n_1=0}^{n_2} (S_2 - 1) f - S_2 \{n_1 \mapsto n_2\} (S_2 - 1) f \\ &= (S_2^2 - S_2) \sum_{n_1=0}^{n_2} f - S_2^2 \{n_1 \mapsto n_2\} f - S_2 \{n_1 \mapsto n_2\} (S_2 - 1) f\end{aligned}$$

Hence, in total we have

$$\begin{array}{llll}
\sum_{n_1=0}^{n_2}(S_1S_2^2 - S_2 - 1)f - (S_2^2 - S_2 - 1)\sum_{n_1=0}^{n_2} f & & & \\
= \{n_1 \mapsto n_2\}S_1S_2^2 f - \{n_1 \mapsto 0\}S_2^2 f - S_2^2\{n_1 \mapsto n_2\}f - S_2\{n_1 \mapsto n_2\}(S_2 - 1)f & & & \\
= \binom{n_2+1}{1} & - \binom{0}{n_2+2} & - \binom{n_2+2}{0} & - \binom{n_2+1}{1} + \binom{n_2+1}{0} \\
= \cancel{\binom{n_2+1}{1}} & - 0 & - 1 & - \cancel{\binom{n_2+1}{1}} + 1 \qquad = 0
\end{array}$$

Since $(S_1S_2^2 - S_2 - 1)f = 0$, we have $(S_2^2 - S_2 - 1)\sum_{n_1=0}^{n_2} f = 0$. Now this is the same recurrence that F_{n_2+1} satisfies and hence the final propagation to difference gives $(S_2^2 - S_2 - 1)(\sum_{n_1=0}^{n_2} f - F_{n_2+1}) = 0$. This proves an induction step of size 2 and leaves two base cases that can be discharged by a theorem prover.

Iteration on Excess Terms. Let g be the term from the negated goal to be proved to be 0. After propagating along the structure of g, we end up with recurrences of the form $Pg = e$ where P is an operator polynomial and the excess terms e do not contain g. In the holonomic case, e will be syntactically 0. We have also observed that e is often 0 in the nonholonomic case as well. But if e is not syntactically 0, then $Pg = e$ cannot immediately be used for a proof by induction. A solution is to iterate a full series of propagations with e in place of g to find $P_2e = e_2$ and conclude $P_2Pg = P_2e = e_2$, then repeat as long as necessary. This process will always terminate, although it might fail to find recurrences.

We will impose an order on the sequence terms to accomplish three things. First, we get a proper definition of which terms in a recurrence are excess. Second, well-foundedness of the order will guarantee termination of the iteration of full propagations to excess terms. Third, the iterations can be interleaved with basic normalizations such as $\{n_1 \mapsto 2n_1\}M_1\{n_1 \mapsto 3n_1{+}1\}f \to 2M_1\{n_1 \mapsto 6n_1{+}2\}f$.

Definition 17. The *spine* of a sequence term f without addition, denoted by $\operatorname{spine} f$, is the sequence term obtained intuitively by erasing operator polynomials from f. Precisely, this means fully reducing f by the rewrite rules $at \to t$, $M_j t \to t$, $\{\vec{n} \mapsto b\vec{n} + \vec{c}\}t \to \{\vec{n} \mapsto b\vec{n}\}t$, and $\sum_j^{\vec{c}\bullet\vec{n}+a} t \to \sum_j^{\vec{c}\bullet\vec{n}} t$ where a, $\vec{c}$, and the matrix b are all numeric.

Shift indeterminates mix with other substitutions, which explains the last two rules. For example, $\operatorname{spine}\{n_1 \mapsto 2n_1\}M_1\{n_1 \mapsto 3n_1 + 1\}f = \{n_1 \mapsto 2n_1\}\{n_1 \mapsto 3n_1\}f$. If we have a sequence term $c_1g^1 + \cdots + c_kg^k$ with addition, it contains multiple spines, one for each g^j. The significance of spines is that when we derive a more complex consequence from a recurrence (during elimination by applying an operator polynomial to it), its spines do not become more complex.

We can easily describe how each propagation step changes the spines of the involved sequence terms. Propagation to the addition $f + g$ produces only spines e_f and e_g in the resulting recurrences, where e_f denotes a spine of a term from a recurrence of f and analogously for e_g. Moreover, propagation to the substitution σf produces σe_f, propagation to the product fg produces $(\operatorname{spine} f)e_g$ and $e_f(\operatorname{spine} g)$, and propagation to the summation $\sum_{n_1=0}^{n_2} f$ produces $\{n_1 \mapsto 0\}$ $(\operatorname{spine} f)$, $\{n_1 \mapsto n_2\}(\operatorname{spine} f)$, and $\sum_{n_1=0}^{n_2} e_f$, where e_f and e_g are as above.

We want propagations to preserve the invariant that excess terms are small. Given how spines change under propagation, a term order on spines offers a way to define smallness. We choose an order that also orients simplifications.

Definition 18. Fix a Knuth–Bendix order with argument coefficients [12] with exactly three weights $W\sum_{n=0}^{a} > W(\cdot) > 3W\sigma > 0$ and all argument coefficients set to 2. Moreover, projection sequence terms corresponding to M_j's (Remark 4) must have equal weights, and substitutions with fewer bindings must have lower precedence. The *excess* (*partial*) *order* on addition-free sequence terms is obtained by comparing the spines of terms using this fixed order. *Excess terms* of a recurrence are all its nonmaximal sequence terms w.r.t. the excess order.

The weights for the excess order are arranged to be compatible with normalization, which pushes substitutions to the leaf nodes of the term tree and pulls summations towards the root. The resulting normal form is simply the typical way of writing terms without explicit substitutions. It is also the normal form of the rewrite system consisting of the applicable associativity and/or commutativity rules of $\cdot$ as well as the following rules:

$$\begin{aligned}
s \cdot \textstyle\sum_j^a t &\to \textstyle\sum_j^a st & 1t, t1, \{\}t &\to t \\
(\textstyle\sum_j^a s) \cdot t &\to \textstyle\sum_j^a st & (\sigma \cup \{n_j \mapsto a\})u &\to \sigma u \\
\sigma \textstyle\sum_j^a t &\to \textstyle\sum_j^{\sigma a} \sigma t & (\sigma \cup \{n_j \mapsto a\})M_j &\to a \\
\textstyle\sum_j^c t &\to \{n_j \mapsto 0\}t + \cdots + \{n_j \mapsto c\}t & \sigma(ts) &\to \sigma t \cdot \sigma s \\
\textstyle\sum_j^{-c} t &\to -\{n_j \mapsto -1\}t - \cdots - \{n_j \mapsto 1-c\}t & \sigma\sigma' t &\to (\sigma \circ \sigma')t
\end{aligned}$$

where s, t, u are sequence terms, u does not contain the variable n_j, a is an affine variable sum, $M_j = \vec{n} \mapsto n_j$ is a projection sequence term, σ, σ' are affine substitutions, and the numeral c is nonnegative.

These rules produce additions, which must be interpreted as follows. For any rule above of the general form $t_0 \to c_1 t_1 + \cdots + c_k t_k$, the actual rewrite on the level of entire recurrences is $f[t_0] + R = 0 \to c_1 f[t_1] + \cdots + c_k f[t_k] + R = 0$ where c_j are numerals, the sequence terms $f[t_j]$ are equal except for the distinguished subterm t_j, and R is the sum of the remaining terms in the recurrence.

To conclude termination, it suffices to prove that t_0 dominates each of $t_1, \ldots, t_k$ individually. The proof is in our technical report. It makes apparent our choices of weights and argument coefficients for the transfinite Knuth–Bendix order.

5 Induction

After propagation, we consider all recurrences $Pg = 0$ of the goal sequence term g to be proved to be 0. In exceptionally fortunate cases, the operator polynomial P is ± 1 and we are unconditionally done because, for any group, the multiplication-by-± 1 map is invertible. This happens when the objective is to prove a recurrence that this method derives as a substep anyway. Otherwise, we

apply induction and leave as conditions the base cases as well as invertibility of the multiplication maps associated with the leading monomials' coefficients.

A common case is that variables range over natural numbers and we have a final recurrence with leading shift $S_1^{b_1} \cdots S_d^{b_d}$ w.r.t. any monomial order. Then the values $\bigcup_{j=1}^{d} \{\vec{n} \in \mathbb{N}^d \mid n_j < b_j\}$ suffice for the base cases, as a union of stacked hyperplanes that is infinite unless $d \leq 1$, but it corresponds to only $\sum_{j=1}^{d} b_j$ one-variable substitutions $\{n_j \mapsto a\}$ for $1 \leq j \leq d$ and $0 \leq a < b_j$. If our eager generalization produced variables that do not participate in their induction (i.e., their b_j's are 0), they are replaced back to their original values.

If there is more than one applicable final recurrence, we take the intersection of their base value sets w.r.t. the same monomial order. To see that it works, consider any point outside the intersection. It is a nonbase point w.r.t. some final recurrence and hence the induction step can be taken by the recurrence.

To represent the intersection as substitutions, we distribute it over the hyperplane stack unions. This results in a union of hyperline stacks of the form $N(J, \vec{b}) := \{\vec{n} \in \mathbb{N}^d \mid n_j < b_j \text{ for all } j \in J\}$ where $J \subseteq \{1, \ldots, d\}$ and $\vec{b}$ vary. One such stack is represented by $\prod_{j \in J} b_j$ substitutions $\{n_j \mapsto a_j \mid j \in J\}$ where the a_j's are chosen arbitrarily such that $0 \leq a_j < b_j$. Unfortunately, distribution duplicates some base cases. To compensate, if $I \subseteq J$ and $\vec{b} \geq \vec{c}$ pointwise, then $N(I, \vec{b}) \supseteq N(J, \vec{c})$, so that $N(J, \vec{c})$ can be removed in favor of $N(I, \vec{b})$.

If a variable $n \in \mathbb{Z}$ is unbounded, we perform two inductions on the rays: $0 \leq n$ and $n < b$ if b base cases are needed. The backward induction on $n < b$ can be transformed into an induction on $\mathbb{N}$ by the change of variables $n \mapsto b - 1 - n$.

6 Examples

Our procedure can prove the induction step of holonomic sequence formulas such as Example 13, the binomial formula:

$$(a+b)^h = \sum_{n=0}^{h} \binom{h}{n} a^n b^{h-n} \qquad \binom{a+b}{h} = \sum_{n=0}^{h} \binom{a}{n} \binom{b}{h-n}$$

Heterogeneous recurrences, which are beyond the holonomic fragment, enable proving elementary general sequence formulas such as Example 11 and the following:

$$\sum_{n=0}^{h} f_{h-n} = \sum_{n=0}^{h} f_n \qquad \sum_{h=0}^{k} \sum_{n=0}^{h} f_{h,n} = \sum_{n=0}^{k} \sum_{h=n}^{k} f_{h,n}$$

If we ignore the holonomic base case requirements, we can for example prove the induction steps of Abel's binomial formula and of some Stirling number identities:

$$(a+b)^h = \sum_{n=0}^{h} \binom{h}{n} a(a-n)^{n-1}(b+n)^{h-n} \qquad h^k/h! = \sum_{n=0}^{h} \left\{ {k \atop n} \right\} / (h-n)!$$

Here, the Stirling numbers of the second kind $\left\{ {k \atop n} \right\}$ are one of many special non-holonomic sequences that frequently arise in combinatorics. They count the number of partitions of a k-element set into n subsets.

As further demonstration, we apply our procedure to the last equation. For convenience, we will use the name of a variable also to denote its multiplier operator. Moreover, we will use the uppercase version of the name of a variable to denote its shift operator. The defining recurrence of the Stirling numbers then reads $(KN-(n+1)N-1)\{{k \atop n}\}=0$ for $k,n\geq 0$, where K and N denote the shift operators for the variables k and n, the first n denotes the multiplier for the variable n, and the second n is the variable itself. This recurrence is complemented by the initial values $\{{0 \atop 0}\}=1$ and $\{{n \atop 0}\}=\{{0 \atop n}\}=0$ if $n\neq 0$.

Starting from the right, the inverse $m!^{-1}$ of the factorial satisfies the recurrence $(mM+M-1)m!^{-1}=0$ that holds for all $m\in\mathbb{Z}$ by extension. This must be found in the initialization step because there is no propagation to division. Propagation to the substitution $\{m\mapsto h-n\}$ then gives the following recurrences, factored for clarity:

$$((h-n+1)H-1)(h-n)!^{-1}=0 \qquad (N-h+n)(h-n)!^{-1}=0$$

To propagate to product, we consider $\{{k_1 \atop n_1}\}$ and $(h_2-n_2)!^{-1}$ with variables renamed apart. We must propagate to the substitution $\{n_j\mapsto n,\ h_j\mapsto h,\ k_j\mapsto k \mid j\in\{1,2\}\}$ the recurrences of $\{{k_1 \atop n_1}\}(h_2-n_2)!^{-1}$ given by the following five operator polynomials:

$$\begin{aligned} K_1N_1-(n_1+1)N_1-1 \ \text{ and } \ H_1-1 \quad &\text{from } \{{k_1 \atop n_1}\} \\ (h_2-n_2+1)H_2-1,\ N_2-h_2+n_2, \ \text{ and } \ K_2-1 \quad &\text{from } (h_2-n_2)!^{-1} \end{aligned}$$

We added here the trivial recurrences given by H_1-1 and K_2-1 implied by the independence from h_1 and k_2. Among the defining recurrences (4) of the compound shift indeterminates N,H,K, the recurrence H_1H_2-H simplifies to H_2-H by H_1-1 and K_1K_2-K to K_1-K by K_2-1. (In other words, the factorwise renaming of already disjoint variables h and k amounts to renaming in the entire product.) The third compound shift recurrence, N_1N_2-N, simplifies to $(h_2-n_2)N_1-N$ by $N_2-h_2+n_2$. The part of the Gröbner basis with only compound shifts is then straightforwardly finished with the result $\{KN-(n_1+1)N-h_2+n_2,\ (h_2-n_2+1)H-1\}$. Hence this propagation step yields

$$(KN-(n+1)N-h+n)\frac{\{{k \atop n}\}}{(h-n)!}=0 \qquad ((h-n+1)H-1)\frac{\{{k \atop n}\}}{(h-n)!}=0$$

To sum over n, we first eliminate n from the previous two recurrences and conclude $(H(K-h)+(N-1)(KH-(h+1)H+1))(\{{k \atop n}\}/(h-n)!)=0$. The sum has natural boundaries, meaning that the summand vanishes outside them. This guarantees that there will be no excess terms, which we also tediously discover when pulling out the indeterminates:

$$\begin{aligned} \sum_{n=0}^{h}(N-1)\overbrace{(KH-(h+1)H+1)}^{P}\frac{\{{k \atop n}\}}{(h-n)!} &= P\Big(\frac{\{{k \atop h+1}\}}{(-1)!}-\frac{\{{k \atop 0}\}}{h!}\Big) \\ &= -\{{k+1 \atop 0}\}/(h+1)!+\{{k \atop 0}\}((h+1)H-1)h!^{-1}=0 \\ \sum_{n=0}^{h}H(K-h)\frac{\{{k \atop n}\}}{(h-n)!}-H(K-h)\sum_{n=0}^{h}\frac{\{{k \atop n}\}}{(h-n)!} &= -(K-h)\frac{\{{k \atop h+1}\}}{(-1)!}=0 \end{aligned}$$

Here, by the recurrence of the inverse of the factorial, we get $(-1)!^{-1} = 0$. So we obtain a recurrence $H(K-h)\sum_{n=0}^{h}\{{k \atop n}\}/(h-n)! = 0$ for the left-hand side of our goal. For the right-hand side, we unproblematically obtain $(K-h)(h^k/h!) = 0$. Hence $H(K-h)$ zeros out the difference $h^k/h! - \sum_{n=0}^{h}\{{k \atop n}\}/(h-n)!$. The largest shift HK of the operator $H(K-h)$ determines that the two sets of base cases $h = 0$ and $k = 0$ are sufficient for induction.

7 Related Work

Holonomic sequences [20] are closely related to our work. Unlike our approach, which allows infinitely many base cases as long as they are finitely representable (Sect. 5), they are limited to a finite number of base cases. Relaxing this limitation yields approximately the homogeneous version of our propagation procedure (i.e., without excess terms), whose theory Chyzak, Kauers, and Salvy laid out [6]. Heterogeneity amounts to module Gröbner bases [5, 8, 13]. Its integration into propagations makes elementary identities about general sequences automatically provable, which may be of interest for general-purpose theorem provers.

In practice, hypergeometric sums are common holonomic sequences that have much faster algorithms available. Gosper's indefinite summation [9] can be applied to compute Wilf–Zeilberger pairs [19], which offer compact proof certificates for definite sum identities. These fast methods admit generalizations to the full holonomic setting. See Koutschan's thesis [11] for an overview.

Finding a closed form instead of only checking it for a summation is a different but related task. A common approach is to perform a recurrence solving phase after recurrence computation, as in the Mathematica package Sigma [1, 15].

8 Conclusion

We presented a procedure for proving equations involving summations within an automatic higher-order theorem prover. The procedure is inspired by holonomic sequences and partly generalizes them. It expresses the problem as recurrences and derives new recurrences from existing ones. In case of success, it shows the induction step of a proof by induction, leaving the base cases to the prover.

As future work, we want to continue implementing the procedure in Zipperposition [17]. We hope that the subsequent practical experiments help us to settle how side conditions of initial recurrences ought to be handled.

Acknowledgment. We thank Pascal Fontaine for his ideas on how to integrate our procedure into SMT and tableaux. We also thank Anne Baanen, Pascal Fontaine, Mark Summerfield, and the anonymous reviewers for suggesting improvements.

Nummelin and Blanchette's research has received funding from the Netherlands Organization for Scientific Research (NWO) under the Vidi program (project No. 016.Vidi.189.037, Lean Forward). Dahmen's research has received funding from the NWO under the Vidi program (project No. 639.032.613, New Diophantine Directions).

References

1. Abramov, S.A., Bronstein, M., Petkovsek, M., Schneider, C.: On rational and hypergeometric solutions of linear ordinary difference equations in ΠΣ*-field extensions. J. Symb. Comput. **107**, 23–66 (2021)
2. Barbosa, H., Reynolds, A., Ouraoui, D.E., Tinelli, C., Barrett, C.W.: Extending SMT solvers to higher-order logic. In: Fontaine, P. (ed.) CADE-27. LNCS, vol. 11716, pp. 35–54. Springer (2019)
3. Bhayat, A., Reger, G.: A combinator-based superposition calculus for higher-order logic. In: Peltier, N., Sofronie-Stokkermans, V. (eds.) IJCAR 2020 (1). LNCS, vol. 12166, pp. 278–296. Springer (2020)
4. Blanchette, J.C., Kaliszyk, C., Paulson, L.C., Urban, J.: Hammering towards QED. J. Formaliz. Reason. **9**(1), 101–148 (2016)
5. Bueso, J., Gómez-Torrecillas, J., Verschoren, A.: Gröbner bases for modules. In: Algorithmic Methods in Non-Commutative Algebra: Applications to Quantum Groups, pp. 169–202. Springer (2003)
6. Chyzak, F., Kauers, M., Salvy, B.: A non-holonomic systems approach to special function identities. In: Johnson, J.R., Park, H., Kaltofen, E. (eds.) Symbolic and Algebraic Computation, International Symposium, ISSAC 2009, Seoul, Republic of Korea, July 29-31, 2009, Proceedings. pp. 111–118. ACM (2009)
7. Cox, D.A., Little, J., O'Shea, D.: Ideals, varieties, and algorithms: An introduction to computational algebraic geometry and commutative algebra. Undergraduate Texts in Mathematics, Springer, Cham, 4th edn. (2015)
8. Fajardo, W., Gallego, C., Lezama, O., Reyes, A., Suárez, H., Venegas, H.: Gröbner bases of modules. In: Skew PBW Extensions: Ring and Module-theoretic Properties, Matrix and Gröbner Methods, and Applications, pp. 261–286. Springer (2020)
9. Gosper, R.W.: Decision procedure for indefinite hypergeometric summation. Proc. Natl. Acad. Sci. USA **75**(1), 40–42 (1978)
10. Kandri-Rody, A., Weispfenning, V.: Non-commutative Gröbner bases in algebras of solvable type. J. Symb. Comput. **9**(1), 1–26 (1990)
11. Koutschan, C.: Advanced applications of the holonomic systems approach. ACM Comm. Comput. Algebra **43**(3/4), 119 (2009)
12. Ludwig, M., Waldmann, U.: An extension of the Knuth-Bendix ordering with LPO-like properties. In: Dershowitz, N., Voronkov, A. (eds.) LPAR 2007. LNCS, vol. 4790, pp. 348–362. Springer (2007)
13. Maletzky, A., Immler, F.: Gröbner bases of modules and Faugère's f_4 algorithm in Isabelle/HOL. CoRR **abs/1805.00304** (2018)
14. Nummelin, V., Blanchette, J., Dahmen, S.R.: Automated deduction with recurrence-driven summations (technical report). Technical report (2023), https://lean-forward.github.io/pubs/sums_report.pdf
15. Schneider, C.: Symbolic summation assists combinatorics. Sem. Lothar. Combin. **56**, 1–36 (2007)
16. Steen, A., Benzmüller, C.: Extensional higher-order paramodulation in Leo-III. J. Autom. Reason. **65**(6), 775–807 (2021)
17. Vukmirović, P., Bentkamp, A., Blanchette, J., Cruanes, S., Nummelin, V., Tourret, S.: Making higher-order superposition work. J. Autom. Reason. **66**(4), 541–564 (2022)
18. Vukmirović, P., Blanchette, J., Schulz, S.: Extending a high-performance prover to higher-order logic. In: Sankaranarayanan, S., Sharygina, N. (eds.) TACAS 2023. LNCS, vol. 13994, pp. 111–129. Springer (2023)

19. Wilf, H.S., Zeilberger, D.: Rational functions certify combinatorial identities. J. Am. Math. Soc. **3**(1), 147–158 (1990)
20. Zeilberger, D.: A holonomic systems approach to special functions identities. Journal of Computational and Applied Mathematics **32**(3), 321–368 (1990)