Formal languages, normal forms, and Hilbert series

Lars Hellström
Lars.Hellstrom@mdh.se

Division of Applied Mathematics, Mälardalen University, Västerås, Sweden

SNAG, Linköpings universitet, 28 september 2018

Definition

Let (for simplicity) $\mathbb{K}$ be a field and $\mathcal{A}$ be a $\mathbb{K}$-algebra, generated by finite-dimensional vector space V. Define a filtration $\{F_n\}_{n\in\mathbb{N}}$ of $\mathcal{A}$ as follows:

- $F_0 = \{r1\}_{r\in\mathbb{K}}$ if $\mathcal{A}$ has a unit 1, otherwise $F_0 = \{0\}$,
- $F_1 = F_0 + V$,
- $F_n = F_{n-1} + \sum_{k=1}^{n-1} F_k \cdot F_{n-k}$ for $n > 1$.

Then the Hilbert series for $\mathcal{A}$ (when generated by V) is

$$H(t) = \sum_{n=0}^{\infty} (\dim F_n) t^n.$$

This can converge for small t, but mostly we'll regard it as a formal power series.

If $\mathcal{A}$ is associative, then

$$F_n = F_{n-1} + V^n, \text{ where } \quad V^n = \left\{ \prod_{k=1}^{n} a_k \,\middle|\, a_1, \dots, a_n \in V \right\}.$$

If furthermore $\mathcal{A}$ is graded so that $\mathcal{A} = \bigoplus_{n=0}^{\infty} V^n$, authors may prefer to work with a graded variant of the Hilbert series

$$H_{\text{graded}}(t) = \sum_{n=0}^{\infty} (\dim V^n) t^n,$$

but a point of making them *series* is that translations between the two are easy; $\dim V^n = \dim F_n - \dim F_{n-1}$ would imply

$$H_{\text{graded}}(t) = \dim F_0 + \sum_{n=1}^{\infty} (\dim F_n - \dim F_{n-1}) t^n = (1-t)H(t)$$

If $\mathcal{A}$ is associative, then

$$F_n = F_{n-1} + V^n, \text{ where } \quad V^n = \left\{ \prod_{k=1}^{n} a_k \,\middle|\, a_1, \dots, a_n \in V \right\}.$$

If furthermore $\mathcal{A}$ is graded so that $\mathcal{A} = \bigoplus_{n=0}^{\infty} V^n$, authors may prefer to work with a graded variant of the Hilbert series

$$H_{\text{graded}}(t) = \sum_{n=0}^{\infty} (\dim V^n) t^n,$$

but a point of making them *series* is that translations between the two are easy; $\dim V^n = \dim F_n - \dim F_{n-1}$ would imply

$$H_{\text{graded}}(t) = \dim F_0 + \sum_{n=1}^{\infty} (\dim F_n - \dim F_{n-1}) t^n = (1-t) H(t)$$

Some examples

For a commutative polynomial algebra in r variables,

$$\dim F_n = \binom{n+r}{r} \quad \text{and} \quad H(t) = \frac{1}{(1-t)^{r+1}}.$$

For a free associative algebra in r variables,

$$\dim F_n = \sum_{k=0}^{n} r^k \quad \text{and} \quad H(t) = \frac{1}{(1-rt)(1-t)}.$$

For a free nonassociative algebra generated by r elements,

$$H(t) = \frac{1-\sqrt{1-4rt}}{2(1-t)} \quad \text{and} \quad \dim F_n = \sum_{k=1}^{n} \frac{1}{k}\binom{2k-2}{k-1} r^k.$$

Counting variables

The variable t is called a counting variable; its powers are primarily used as labels (distinguishing different counts within a single series).
Having several counting variables can let one keep a more fine-grained track of things.

In the case of a hom-algebra, one typically wants a separate counting variable for how many times the hom α has been applied, since otherwise the pieces might be infinite-dimensional.

$$F_{n,m} := F_{n-1,m} + F_{n,m-1} + \alpha(F_{n,m-1}) + \sum_{k=1}^{n-1} \sum_{l=0}^{m} F_{k,l} F_{n-k,m-l}$$

$$H(s,t) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} (\dim F_{n,m}) s^m t^n$$

Hilbert series for subspaces

One can also define a Hilbert series $H_{\mathcal{I}\subseteq\mathcal{A}}$ for a subspace $\mathcal{I}$ of an algebra $\mathcal{A}$:

$$H_{\mathcal{I}\subseteq\mathcal{A}}(t) = \sum_{n=0}^{\infty} \dim(\mathcal{I} \cap F_n) t^n$$

If the filtration on the quotient algebra $\mathcal{A}/\mathcal{I}$ is generated by $\{x + \mathcal{I} \,|\, x \in V\}$, then

$$H_{\mathcal{A}}(t) = H_{\mathcal{I}\subseteq\mathcal{A}}(t) + H_{\mathcal{A}/\mathcal{I}}(t).$$

An application: the Ore domain condition

Recall that in commutative algebra, it is a sufficient and necessary condition for an algebra to have a field of fractions that it has no zero divisors.

In noncommutative algebra, having no zero divisors is no longer sufficient; one must also require that two arbitrary denominators a and b have a common same-side multiple

$$\exists c, d \neq 0 \colon ac = bd.$$

An Ore domain is an associative ring without zero divisors and with common same-side multiples.

Theorem

Every associative algebra $\mathcal{A}$ without zero divisors for which the Hilbert series has radius of convergence 1 is an Ore domain.

An application: the Ore domain condition

Recall that in commutative algebra, it is a sufficient and necessary condition for an algebra to have a field of fractions that it has no zero divisors.

In noncommutative algebra, having no zero divisors is no longer sufficient; one must also require that two arbitrary denominators a and b have a common same-side multiple

$$\exists c, d \neq 0 \colon ac = bd.$$

An Ore domain is an associative ring without zero divisors and with common same-side multiples.

Theorem

Every associative algebra $\mathcal{A}$ without zero divisors for which the Hilbert series has radius of convergence 1 is an Ore domain.

Proof

For the contrapositive, assume $a, b \in \mathcal{A} \setminus \{0\}$ have no common right multiple other than 0. Then for every $n \in \mathbb{N}$,

$$(a \cdot F_n) \cap (b \cdot F_n) = \{0\}.$$

This implies that the sum $a \cdot F_n + b \cdot F_n$ is direct, so

$$\dim(a \cdot F_n + b \cdot F_n) = \dim F_n + \dim F_n = 2 \dim F_n.$$

Now let $m > 0$ be minimal such that $a, b \in F_m$. Since $\{F_n\}_{n \in \mathbb{N}}$ is a filtration, we have

$$a \cdot F_n + b \cdot F_n \subseteq F_{m+n} \quad \text{and thus}$$
$$2 \dim F_n = \dim(a \cdot F_n + b \cdot F_n) \leqslant \dim F_{m+n}$$

for all $n \in \mathbb{N}$. It follows that the terms of $H(t)$ do not decrease for $t \geqslant 2^{-1/m}$, implying in particular that the radius of convergence is strictly less than 1.

Coordinate view

Some traditions emphasize stating everything in a coordinate-free form – the above definitions adhere to this – but in order to get anywhere calculation-vise, it is often necessary to introduce coordinates.

Let X be some basis of the degree 1 subspace V.
Let Y be the multiplicative closure of X. (Include multiplicative unit if there is one. Ditto other multiplication-like operations.)

By construction, $Y \cap F_n$ spans F_n.

We can (abstractly) pick a basis $B \subseteq Y$ such that $B \cap F_n$ is a basis of F_n. Then

$$H(t) = \sum_{n=0}^{\infty} |B \cap F_n| \, t^n$$

which is a discrete formula for the Hilbert series.

Coordinate view

Some traditions emphasize stating everything in a coordinate-free form – the above definitions adhere to this – but in order to get anywhere calculation-vise, it is often necessary to introduce coordinates.

Let X be some basis of the degree 1 subspace V.
Let Y be the multiplicative closure of X. (Include multiplicative unit if there is one. Ditto other multiplication-like operations.)

By construction, $Y \cap F_n$ spans F_n.

We can (abstractly) pick a basis $B \subseteq Y$ such that $B \cap F_n$ is a basis of F_n. Then

$$H(t) = \sum_{n=0}^{\infty} |B \cap F_n| \, t^n$$

which is a discrete formula for the Hilbert series.

Constructively picking a basis

It is convenient to switch perspective, and present the algebra $\mathcal{A}$ as a quotient of some suitable free algebra $\mathcal{F}$ generated by X. Then Y is a basis of $\mathcal{F}$, and B is a basis of the subspace of normal forms for elements of the quotient $\mathcal{A}$.

Example

If $\mathcal{A}$ is associative, we can take $\mathcal{F} = \mathbb{K}\langle X\rangle$ and $Y = X^*$ (the set of words on the alphabet X – essentially the set of all finite strings of characters from X).
$Y \cap F_n$ is the set of words of length at most n.

Example

If $\mathcal{A}$ is a hom-algebra, we can take $\mathcal{F}$ to be the free $\mathbb{K}$-algebra with signature $\Omega = \{\alpha(), \mu(,)\}$ (i.e., unary "hom" operation α and binary "multiplication" operation μ) generated by X, and Y to be the set of all formal terms on X with that signature Ω.

Diamond Lemma

If reduction to normal form (for representing elements of $\mathcal{A}$) is described using a confluent rewrite system, then by the Diamond Lemma, a basis B for the set of normal form consists of precisely those elements of Y that no rewrite rule acts nontrivially upon.

If moreover all rewrite rules map each F_n into itself (i.e., doesn't raise the degree), then the above basis B gives the right counts for the Hilbert series.

Effectively, B is characterised as the subset of Y consisting of all words that don't contain certain forbidden subwords.

Example: Noncommutative circle

The noncommutative circle algebra is

$$\mathcal{A} = \mathbb{K}\langle \mathsf{x}, \mathsf{y}\rangle \big/ \langle \mathsf{x}^2 + \mathsf{y}^2 - 1\rangle.$$

The generator of that ideal can be turned into the rewrite rule $\mathsf{y}^2 \to 1 - \mathsf{x}^2$, but that rule alone does not constitute a *confluent* rewrite system. For that, one needs

$$\{\mathsf{y}^2 \to 1 - \mathsf{x}^2, \mathsf{y}\mathsf{x}^2 \to \mathsf{x}^2\mathsf{y}\}.$$

Hence

$$H(t) = \sum_{n=0}^{\infty} \left| \left\{ w \in \{\mathsf{x},\mathsf{y}\}^* \;\middle|\; \begin{array}{l} w \text{ has length at most } n \text{ and does} \\ \text{not have } \mathsf{y}^2 \text{ or } \mathsf{y}\mathsf{x}^2 \text{ as subword} \end{array} \right\} \right| t^n$$

is the Hilbert series for the noncommutative circle algebra.

Formal languages

A subset of X^* is called a formal (word) language.
A subset of the set of all formal terms over a given signature is a formal tree language.

(Since a language is *only* a set of words — a priori there's no grammar, semantics, or anything — one might feel *vocabulary* would be a more apt term, but this is the way it is.)

Formal language theory deals with (amongst other things) finitary descriptions of possibly infinite languages. Interesting classes include:

- regular languages
- context-free languages
- parsing expression languages

Algebra of word languages

Although a language $L \subseteq X^*$ is by definition a set, there are advantages to encoding it as the formal power series

$$\sum_{w \in L} w \in \mathbb{B}\langle\langle X \rangle\rangle.$$

$\mathbb{B} = \{0, 1\}$ denotes the boolean semiring, which has $1 + 1 = 1$; effectively $+$ is OR and $\cdot$ is AND.

Addition in $\mathbb{B}\langle\langle X \rangle\rangle$ corresponds to union of languages. Multiplication is concatenation

$$\left(\sum_{u \in L_1} u\right)\left(\sum_{v \in L_2} v\right) = \sum_{w \in \{w \in X^* \mid w = uv \text{ for some } u \in L_1,\, v \in L_2\}} w$$

The zero $0 \in \mathbb{B}\langle\langle X \rangle\rangle$ is the empty language $\varnothing$. The unit $1 \in \mathbb{B}\langle\langle X \rangle\rangle$ is (the language whose only element is) the empty word. The power series x is the language whose only element is the length 1 word x.

More language algebra: regular expressions

Traditionally one also defines three unary operations on word languages:

$$a^* = \sum_{n=0}^{\infty} a^n \qquad \text{Kleene star; zero-or-more}$$

$$a^+ = \sum_{n=1}^{\infty} a^n \qquad \text{Kleene plus; one-or-more}$$

$$a^? = 1 + a \qquad \text{zero-or-one}$$

a^* and a^+ would not be defined for arbitrary power series in general, but they are defined on the whole of $\mathbb{B}\langle\langle X \rangle\rangle$, since $\mathbb{B}$ has $1 + 1 = 1$.

Theorem

The set of regular word languages on X is the smallest subset of $\mathbb{B}\langle\langle X \rangle\rangle$ which is closed under addition, multiplication, and the Kleene star, and which contains 0, 1, and the elements of X.

Back to the example

The words which are reducible under the rewrite system

$$\{y^2 \to 1 - x^2, yx^2 \to x^2y\}$$

are those of the language

$$(x+y)^*(y^2 + yx^2)(x+y)^*.$$

Its complement — the irreducible words that make up a basis for the normal form — is the regular language

$$x^*(yx)^*y^?$$

Theorem

The set complement of a regular language is also a regular language.

Back to the example

The words which are reducible under the rewrite system

$$\{y^2 \to 1 - x^2, yx^2 \to x^2y\}$$

are those of the language

$$(x + y)^*(y^2 + yx^2)(x + y)^*.$$

Its complement — the irreducible words that make up a basis for the normal form — is the regular language

$$x^*(yx)^*y^?$$

Theorem

The set complement of a regular language is also a regular language.

Automaton description of languages

As is typical in formal language theory, regular languages also admit a description in terms of automata.

A Deterministic Finite Automaton (DFA) is a tuple (S, ϕ, i, T), where

- S is a finite set,
- $\phi\colon X \longrightarrow S^S$ is a function mapping the alphabet X to functions $S \longrightarrow S$, which we extend to a monoid homomorphism $X^* \longrightarrow S^S$,
- $i \in S$ is called the initial state, and
- $T \subseteq S$ is called the set of accepting states.

The language accepted by (S, ϕ, i, T) is

$$\left\{ w \in X^* \,\middle|\, \phi(w)(i) \in T \right\}.$$

(This reads the word w right-to-left. One could do it the other way around.)

Automaton description of languages 2

A nondeterministic finite automaton (NFA) is a tuple (G, ψ, s, t) where

- G is a finite digraph (its vertices are the states of the automaton),
- $s, t \in \mathrm{V}(G)$ are the initial and final respectively states,
- $\psi(e) \subseteq X \cup \{1\}$ for all $e \in \mathrm{E}(G)$.

A word w is in the language accepted by (G, ψ, s, t) if there is a walk $u_0\, e_1\, u_1\, \ldots\, e_n u_n$ in G and factorisation $w = w_1 \cdots w_n$ such that $u_0 = s$, $u_n = t$, and $w_i \in \psi(e_i)$ for all $i = 1, \ldots, n$.

Theorem

The following are equivalent claims about a language $L \subseteq X^*$:

1. *L is the value of a regular expression in $\mathbb{B}\langle\langle X \rangle\rangle$.*
2. *L is the language accepted by some DFA.*
3. *L is the language accepted by some NFA.*

Automaton description of languages 2

A nondeterministic finite automaton (NFA) is a tuple (G, ψ, s, t) where

- G is a finite digraph (its vertices are the states of the automaton),
- $s, t \in \mathrm{V}(G)$ are the initial and final respectively states,
- $\psi(e) \subseteq X \cup \{1\}$ for all $e \in \mathrm{E}(G)$.

A word w is in the language accepted by (G, ψ, s, t) if there is a walk $u_0\, e_1\, u_1 \, \ldots \, e_n u_n$ in G and factorisation $w = w_1 \cdots w_n$ such that $u_0 = s$, $u_n = t$, and $w_i \in \psi(e_i)$ for all $i = 1, \ldots, n$.

Theorem

The following are equivalent claims about a language $L \subseteq X^*$:

1. *L is the value of a regular expression in $\mathbb{B}\langle\langle X \rangle\rangle$.*
2. *L is the language accepted by some DFA.*
3. *L is the language accepted by some NFA.*

Translations between descriptions

RE to NFA is an easy structural recursion.

NFA to DFA is the famous powerset construction.

DFA to RE can be done with matrices over $\mathbb{B}\langle\langle X \rangle\rangle$. Let $\{\mathbf{e}_s\}_{s\in S}$ be the "standard basis" vectors. Let $\mathbf{t} = \sum_{s\in T} \mathbf{e}_s$. Let $A = \sum_{x\in X} \sum_{s\in S} \mathbf{e}_{\phi(x)(s)} x \mathbf{e}_s^{\mathrm{T}}$. Then the language accepted by the DFA (S, ϕ, i, T) is

$$\mathbf{t}^{\mathrm{T}} A^* \mathbf{e}_i = \mathbf{t}^{\mathrm{T}} (I - A)^{-1} \mathbf{e}_i$$

where $(I - A)^{-1}$ can be calculated using Gaussian elimination in $\mathbb{Q}\langle\langle X \rangle\rangle^{|S|\times|S|}$.

The resulting RE in the last case can also be used to calculate the corresponding Hilbert series; the key is that since it starts out deterministic, it generates words at most once.

Translations between descriptions

RE to NFA is an easy structural recursion.

NFA to DFA is the famous powerset construction.

DFA to RE can be done with matrices over $\mathbb{B}\langle\langle X\rangle\rangle$.
Let $\{\mathbf{e}_s\}_{s\in S}$ be the "standard basis" vectors. Let $\mathbf{t} = \sum_{s\in T}\mathbf{e}_s$. Let $A = \sum_{x\in X}\sum_{s\in S}\mathbf{e}_{\phi(x)(s)}x\mathbf{e}_s^{\mathrm{T}}$.
Then the language accepted by the DFA (S, ϕ, i, T) is

$$\mathbf{t}^{\mathrm{T}}A^*\mathbf{e}_i = \mathbf{t}^{\mathrm{T}}(I-A)^{-1}\mathbf{e}_i$$

where $(I-A)^{-1}$ can be calculated using Gaussian elimination in $\mathbb{Q}\langle\langle X\rangle\rangle^{|S|\times|S|}$.

The resulting RE in the last case can also be used to calculate the corresponding Hilbert series; the key is that since it starts out deterministic, it generates words at most once.

Translations between descriptions

RE to NFA is an easy structural recursion.

NFA to DFA is the famous powerset construction.

DFA to RE can be done with matrices over $\mathbb{B}\langle\langle X\rangle\rangle$. Let $\{\mathbf{e}_s\}_{s\in S}$ be the "standard basis" vectors. Let $\mathbf{t} = \sum_{s\in T}\mathbf{e}_s$. Let $A = \sum_{x\in X}\sum_{s\in S}\mathbf{e}_{\phi(x)(s)}x\mathbf{e}_s^{\mathrm{T}}$. Then the language accepted by the DFA (S, ϕ, i, T) is

$$\mathbf{t}^{\mathrm{T}}A^*\mathbf{e}_i = \mathbf{t}^{\mathrm{T}}(I-A)^{-1}\mathbf{e}_i$$

where $(I-A)^{-1}$ can be calculated using Gaussian elimination in $\mathbb{Q}\langle\langle X\rangle\rangle^{|S|\times|S|}$.

The resulting RE in the last case can also be used to calculate the corresponding Hilbert series; the key is that since it starts out deterministic, it generates words at most once.

Translations between descriptions

RE to NFA is an easy structural recursion.

NFA to DFA is the famous powerset construction.

DFA to RE can be done with matrices over $\mathbb{B}\langle\langle X\rangle\rangle$.
Let $\{\mathbf{e}_s\}_{s\in S}$ be the "standard basis" vectors. Let $\mathbf{t} = \sum_{s\in T} \mathbf{e}_s$. Let $A = \sum_{x\in X} \sum_{s\in S} \mathbf{e}_{\phi(x)(s)} x \mathbf{e}_s^{\mathrm{T}}$.
Then the language accepted by the DFA (S, ϕ, i, T) is

$$\mathbf{t}^{\mathrm{T}} A^* \mathbf{e}_i = \mathbf{t}^{\mathrm{T}} (I - A)^{-1} \mathbf{e}_i$$

where $(I - A)^{-1}$ can be calculated using Gaussian elimination in $\mathbb{Q}\langle\langle X\rangle\rangle^{|S|\times|S|}$.

The resulting RE in the last case can also be used to calculate the corresponding Hilbert series; the key is that since it starts out deterministic, it generates words at most once.

Again with the example

The normal form basis was

$$\mathsf{x}^*(\mathsf{yx})^*\mathsf{y}^?$$

where $a^* = \sum_{n=0}^{\infty} a^n = (1-a)^{-1}$ and $a^? = 1 + a$. The generators x and y are both degree 1, so they are worth t in the counting picture. The graded Hilbert series is thus

$$H_{\text{graded}}(t) = t^*(t^2)^*t^? = \frac{1}{1-t}\frac{1}{1-t^2}(1+t) = \frac{1}{(1-t)^2}$$

and the non-graded series is $H(t) = \dfrac{1}{(1-t)^3}$, whereas a commutative circle algebra would have

$$H(t) = \frac{1+t}{(1-t)^2}$$

More general classes of word languages

Context-free languages are, thanks to the Backus–Naur form (BNF) of grammars the norm (dogma?) for computer language syntax.

In $\mathbb{B}\langle\langle X\rangle\rangle$, the languages are solutions to systems of polynomial equations that are immediate from the grammar.

But CFL are at heart nondeterministic, so there is no obvious way to describe their complements, and $\mathbb{B}\langle\langle X\rangle\rangle$ equations cannot easily be carried over to counting equations.

Parsing expression languages are similar to CFL, but replaces addition/union/choice by prioritised choice.

Pro: Becomes deterministic. Language class closed under taking complement. Can characterise several languages of mathematical interest that are not CFL.

Con: Not generative. No general link to Hilbert series.

More general classes of word languages

Context-free languages are, thanks to the Backus–Naur form (BNF) of grammars the norm (dogma?) for computer language syntax.
In $\mathbb{B}\langle\langle X \rangle\rangle$, the languages are solutions to systems of polynomial equations that are immediate from the grammar.
But CFL are at heart nondeterministic, so there is no obvious way to describe their complements, and $\mathbb{B}\langle\langle X \rangle\rangle$ equations cannot easily be carried over to counting equations.

Parsing expression languages are similar to CFL, but replaces addition/union/choice by prioritised choice.
Pro: Becomes deterministic. Language class closed under taking complement. Can characterise several languages of mathematical interest that are not CFL.
Con: Not generative. No general link to Hilbert series.

More general classes of word languages

Context-free languages are, thanks to the Backus–Naur form (BNF) of grammars the norm (dogma?) for computer language syntax.

In $\mathbb{B}\langle\langle X \rangle\rangle$, the languages are solutions to systems of polynomial equations that are immediate from the grammar.

But CFL are at heart nondeterministic, so there is no obvious way to describe their complements, and $\mathbb{B}\langle\langle X \rangle\rangle$ equations cannot easily be carried over to counting equations.

Parsing expression languages are similar to CFL, but replaces addition/union/choice by prioritised choice.

Pro: Becomes deterministic. Language class closed under taking complement. Can characterise several languages of mathematical interest that are not CFL.

Con: Not generative. No general link to Hilbert series.

Tree languages

For nonassociative algebra, one rather wants tree languages. Here it turns out that regular languages work pretty much as you're used to context-free languages working, but with all the advantages of determinism. (The bad sides of word CFLs are due to the nesting structure being ambiguous.)

The tree language L of free hom-algebra monomials (in Polish notation) satisfies the equation

$$L = X + \alpha L + \mu LL$$

which translates to the Hilbert series equation

$$H_{\text{graded}}(s,t) = |X|t + sH_{\text{graded}}(s,t) + H_{\text{graded}}(s,t)^2$$

if using $t$as counting variable for generators and s as counting variable for applications of the hom α.