Converting indexed languages to functional equations

Marni Mishna

March 9, 2005

Indexed languages were introduced in the thesis of Aho in the late 60s to model a more natural subclass of context sensitive languages that still had interesting closure properties. [1, 4]

The goal here is to determine an *automated* way to convert the grammar production rules of an indexed language into meaningful functional equations satisfied by its generating series. Here is a sample of some known indexed languages and their generating series. In the second section we take a stab at converting the equations.

Indexed grammars

An indexed grammar is a 5-tuple (V, Σ, I, P, S), where V is the set of variables, Σ the set of terminals, I the set of indices, $S \in V$ the start symbol, and P a finite set of productions of the form

$$A \to \alpha \quad A \to B_f \quad A_f \to \alpha,$$

where $A, B \in V$, $f \in I$ and $\alpha \in (V \cup \Sigma)^*$. Derivations are similar to those in a CFG except variables may be followed by strings of indices. When a production such as $A \to BC$ is applied, the string of indices for A is attached to B and C.

Properties of indexed grammars

- properly includes all context-free grammars
- proper subset of the class of context-sensitive languages
- the class of indexed languages is *full abstract family of languages*, htat is it is closed under union, concatenation, Kleene closure (*), homomorphism, inverse homomorphims and intersection with regular sets.
- the set of indexed languages *is not* closed under intersection or complement. i

1. $L_1 = \{a^n b^n c^n : n > 0\}$

Grammar

$$\begin{array}{lll} S \to T_g & T \to T_f & T \to ABC \\ A_f \to aA & B_f \to bB & C_f \to cC \\ A_g \to a & B_g \to b & C_g \to c \end{array}$$

Sample Derivation

$$\begin{aligned} A_{f^n g} &\to aA_{f^{n-1}g} \to a^2 A_{f^{n-2}g} \cdots \to a^n A_g \to a^{n+1} \\ S &\to T_g \to T_{fg} \to^* T_{f^n g} \to A_{f^n g} B_{f^n g} C_{f^n g} \\ &\to a^{n+1} b^{n+1} c^{n+1} \end{aligned}$$

Generating series

$\sum_n z^{3n} = \frac{1}{1-z^3}$ (rational)

2. $L_2 = \{a^{n^2} : n \geq 1\}$

Grammar

$$S \rightarrow T_g \quad T \rightarrow T_f A | A \quad A_f \rightarrow aaA$$
$$A_g \rightarrow a$$

Sample Derivations

$$\begin{aligned} S &\rightarrow T_g \ \rightarrow A_g \ \rightarrow a \\ A_{f^n g} &\rightarrow a^{2n+1} \\ S &\rightarrow T_{fg} A_g \ \rightarrow T_{ffg} A_{fg} A_g \ \rightarrow^* T_{f^n g} A_{f^{n-1} g} \ldots A_g \\ &\rightarrow A_{f^n g} A_{f^{n-1} g} \ldots A_g \ \rightarrow^* a^{2(n+1)+1} a^{2n+1} \ldots a = a^{\sum_{i=0}^n 2i+1} = a^{(n+1)^2} \end{aligned}$$

Generating series

$L_2(z) = \sum_n z^{n^2}$

- Sloane number: A010052
- $L_2(z)$ satisfies $0 = f(L_2(z), L_2(z^2), L_2(z^4))$ where $f(u, v, w) = (u-w)^2 - (v-w)(v+w-1)$ - Michael Somos, Jul 19 2004
- See note below for differential equation.
- **Comments:**
  For n ¿= 1 another formula for a(n) is: a(n) = d(n) mod 2 where d(n) is the number of divisors of n, A000005. - Ahmed Fares (ahmedfares(AT)my-deja.com), Apr 19 2001
- **References:**
  J.-P. Allouche and J. Shallit, Automatic Sequences, Cambridge Univ. Press, 2003, p. 4.
  T. M. Apostol, Introduction to Analytic Number Theory, Springer-Verlag, 1976, page 48, Problem 20.
  Y. Puri and T. Ward, Arithmetic and growth of periodic orbits, J. Integer Seqs., Vol. 4 (2001), #01.2.1.

3. $L_3 = \{a^n b^{n^2} : n \geq 1\}$

Grammar

$$S \rightarrow T_g \quad T \rightarrow T_f B | AB \quad A_f \rightarrow aA \quad B_f \rightarrow bbB$$
$$A_g \rightarrow a \quad B_g \rightarrow b$$

Sample Derivation

$$\begin{aligned} B_{f^n g} &\rightarrow^* b^{2n+1} \\ S &\rightarrow T_g \ \rightarrow T_{fg} B_g \ \rightarrow T_{f^2 g} B_{fg} B_g \ \rightarrow^* T_{f^n g} B_{f^{n-1} g} \ldots B_g \\ &\rightarrow A_{f^n g} B_{f^n g} B_{f^{n-1} g} \ldots B_g \ \rightarrow a^{n+1} b^{\sum_{i=0}^n 2i+1} = a^{n+1} b^{(n+1)^2} \end{aligned}$$

Generating series

- ID Number: A005369
- Comments: Euler transform of period 4 sequence [0,1,0,-1,...]. Expansion of $q^{-1/4}\eta(q^4)^2/\eta(q^2)$ in powers of q.
- Formula: G.f.: $\prod_{k>0}(1-x^(4k))/(1-x^(4k-2)) = f(x^2,x^6)$ where $f(a,b)$ is Ramanujan's theta function.

4. $L_4 = \{a^{2^n} : n \geq 1\}$

Grammar

$$S \to T_g \quad T \to T_f | UU \quad U_f \to UU \quad U_g \to a$$

Sample Derivation

$$S \to T_g \to^* T_{f^n}g \to U_{f^n g}U_{f^n g} \to U_{f^{n-1}g}U_{f^{n-1}g}U_{f^{n-1}g}U_{f^{n-1}g} \to^* (U_g)^{2^n} \to a^{2^n}$$

Generating series

$L_4(z) = \sum_n z^{2^n}$

- Sloane number: A036987
- **Name:** Fredholm-Rueppel sequence.
- **Comments:**
  a(n+1) = a(floor(n/2)) * (n mod 2); a(0)=1. - Reinhard Zumkeller (reinhard.zumkeller(AT)lhsystems.com), Aug 02 2002 $\sum 0..infinity 1/10^(2^n) = 0.1101000100000001...$ Binary representation of Kempner-Mahler number $\sum(k >= 0, 1/2^(2^k))$.
- **References:**
  H. Niederreiter and M. Vielhaber, Tree complexity and a doubly ..., J. Complexity, 12 (1996), 187-198.
  D. Kohel, S. Ling and C. Xing, Explicit Sequence Expansions
  E. Sheppard, net.math post (1985)
  D. Bailey et al., On the binary expansions of algebraic numbers
  Daniele A. Gewurz and Francesca Merola, Sequences realized as Parker vectors ..., J. Integer Seqs., Vol. 6, 2003.
  Stephen Wolfram, [Page 1092] A New Kind of Science — Online.

5. $\{ww | w \in \{a,b\}^*\}$

Grammar

$$\begin{array}{ccc} S \to T_x & T \to T_f | T_g & T \to RR \\ R_f \to aR & R_g \to bR & R_x \to \epsilon \end{array}$$

Sample Derivation

$$\begin{aligned} S & \to T_x \to T_{fx} \to T_{gfx} \to^* T_{ggfffgfx} \to R_{ggfffgfx}R_{ggfffgfx} \\ & \to^* bbaaabaR_x bbaaabaR_x \to bbaaababbaaaba \end{aligned}$$

Generating series

$L_5(z) = \sum_n 2^n z^{2n} = \frac{1}{1-2z^2}$

Additional notes

Is $\{w^k | w \in \{a,b\}^+, k > 1\}$ an inherently ambiguous indexed grammar?

6. $L_6 = \{0^n | n$ is composite$\}$

Grammar

$$S \to T_{fg} \quad T \to T_f | R \quad R \to RA | AA$$
$$A_f \to aA \quad A_g \to a$$

Is this inherently ambiguous?

Sample Derivations

$$S \to T_{f^n g} \to R_{f^n g} \to R_{f^n g} A_{f^n g} \to R_{f^n g} A_{f^n g} A_{f^n g} \to^* A^m_{f^n g} \to a^{(n+1)m}$$

Generating series

7. $L_7 = \{ab^{i_1} ab^{i_2} \dots ab^{i_k} | 0 \le i_1 \le i_2 \le \cdots \le i_k\}$

Grammar

$$S \to T_g \quad T \to G | GT \quad T \to T_f$$
$$G_f \to Gb \quad G_g \to a$$

Sample Derivations

$$G_{f^n g} \to ab^n$$
$$S \to T_g \to^* T_{f^i g} \to G_{f^{i_1} g} T_{f^{i_1} g} \to^* ab^{i_1} T_{f^{i_2}} \to ab^{i_1} G_{f^{i+j}} T_{f^{i+j}}$$
$$\to ab^{i_1} ab^{i_2} T_{f^{i_2}} \to^* ab^{i_1} ab^{i_2} \dots ab^{i_{k-1}} G_{f^{i_k} g} \to ab^{i_1} ab^{i_2} \dots ab^{i_k}$$

Generating series

$p(n)$ = number of partitions of n $L_7(z) = \sum_n p(n) z^n = \prod_{i>0} \frac{1}{1-z^i}$

Related questions

1. NOT indexed [2, 3]
   - $\{a^{n!} : n \ge 1\}$
   - $\{(ab^n)^n : n \ge 1\}$

   Question: Are these context sensitive??

2. Is $L = \{w \in \{a,b,c\} | |w|_a = |w|_b = |w|_c\}$ an indexed language?
   Generating function: $\sum \frac{(3n)!}{n!n!n!} z^{3n}$

3. Stanley (1999), Exercise 6.63:
   (a) [5] Suppose that $y = \sum_{n \ge 0} a_n z^n \in C[[z]]$ is D-finite. Define the characteristic function $\chi : C \to Z$ by $\chi(a) = 1$ if $a \ne 0$ and 0 otherwise.
   (When) Is $\sum_{n \ge 0} \chi(a_n) z^n$ rational? This question is open even if y is just assumed to be algebraic. (See exercise 4.3)

   (b) [3] Show (a) is false if y is assumed only to satisfy an *algebraic differential equation (ADE).*

(c) [5] Suppose that y satisfies an ADE and that y is not a polynomial. Can y be any more than quadratically lacunary? In otherwords, if $y = \sum b_i x^{n_i}$, can one have $\lim_{i\to\infty} i^2/n_i - =$?

Solution 6.63(b): Jacobi showed that the series $y = 1 + 2\sum_{n\geq 1} x^{n^2}$ satisfies the ADE

$$(y^2 z_3 - 15yz_1z_2 + 30z_1^3)^2 + 32(yz_2 - 3z_1^2)^3 - y^10(yz_2 - 3z_1^2)^2 = 0$$

with $z_1 = xy', z_2 = xy' + x^2y'', z_3 = xy' + 3x^2y'' + x^3y'''$.

An indexed grammar for an inherent ambiguous CFL

$L_8 = \{a^ib^jc^kd^l | i = j \text{ or } k = l\}$

Idea: break it down as $L_8 = A_1 + A_2 + A_3 + A_4 + A_5$ with $A_1 = \{a^ib^ic^jd^k | j < k\}$ $A_2 = \{a^ib^ic^jd^k | k < j\}$ $A_6 = \{a^ib^jc^kd^k | i < j\}$ $A_4 = \{a^ib^jc^kd^k | j < i\}$ $A_5 = \{a^ib^ic^id^i\}$. You cannot do this in the case of context-free languages because A_5 is not context free. The remaining four are context free (unambiguous), for example A_1:

$$A_1 \to XY \quad X \to aXb|ab \quad Y \to cYd|Z$$
$$Z \to dZ|d$$

Automatic conversion: grammar to func. eqs.

There are several types of production rules for an indexed grammar.

1. Straightforward context-free style rule: $A \to BC$
   Hadamard product (to transfer all indices)
2. "push" rule: $A \to A_f$
   When refering to self: Sums
3. "pop" rule: $A_f \to BC$
   Finite recurrences
4. Terminal rule: $A \to a$
   $A(z) = z$. (Base values of recurrences)

Let $I = \{f_1, \ldots, f_n\}$ be the set of indices and $V = \{S, A_1, \ldots, A_k\}$ be the set of non-terminals. Each production rule implies a production rule for every $\mathbf{m} = (m_1, \ldots, m_n), A_i$ pair: $A_i[f_1^{m_1} \cdots f_n^{m_n}] = \ldots.$ (Simplify this to $A_i[\mathbf{m}]$)

1. We are interested in $S(z)$.
2. How do we model the generating function for A_i?
3. Under which conditions does the system yield a solvable series?

Try this out on some easy-cheesy examples.

$L_1 = \{a^nb^nc^n : n > 0\}$

$$\begin{array}{lll} S \to T_g & T \to T_f & T \to ABC \\ A_f \to aA & B_f \to bB & C_f \to cC \\ A_g \to a & B_g \to b & C_g \to c \end{array}$$

$$\begin{aligned} S(z) &= T[0,1](z) \\ T[m,n](z) &= T[m+1,n](z) + A[m,n]B[m,n]C[m,n](z) \\ A[m,n] &= zA[m-1,n] \\ A[0,1] &= z \end{aligned}$$

$$S(z) = T[0,1](z) = T[1,1](z) + A[1,1]B[1,1]C[1,1](z) = \sum A[m,1]B[m,1]C[m,1](z)$$

Now, we solve $A[m,1](z) = z^{m+1}$ (likewise for B and C) and then $A[m,1]B[m,1]C[m,1](z) = z^{3(m+1)}$.

It is possible that if we make some imposition on our grammar that the length of the index sequence is at least as long as all the words it could generate, we can then say something about the coefficients.

$L_2 = \{a^{n^2} : n \geq 1\}$

$$\begin{array}{lll} S \to T_g & T \to T_f|AB & A_f \to AB \\ A_g \to \epsilon & B_f \to aaB & B_g \to a \end{array}$$

$$\begin{aligned} S(z) &= T[0,1](z) \\ T[m,n](z) &= T[m+1,n](z) + A[m,n]B[m,n] \\ A[m,n](z) &= A[m-1,n]B[m-1,n] \\ A[0,1](z) &= 1 \\ B[m,n](z) &= z^2 B[m-1,n] \\ B[0,1](z) &= z \end{aligned}$$

First simplifications:

$$A[0,1](z) = 1 \qquad A[m,n](z) = A[m-1,n]B[m-1,n] \implies A[m,n] = \prod_{i=0}^{m-1} B[i,n]$$

$$B[m,n](z) = z^2 B[m-1,n] \implies B[m,n] = z^{2m+1} \implies A[m,n] = z^{\sum_{i=0}^{m-1} 2i} = z^{m^2}$$

$$S(z) = T[0,1](z) = \sum_{m \geq 0} A[m,1]B[m,1](z) = \sum_{m \geq 0} A[m+1,1](z) = \sum_{m > 0} z^{m^2}$$

$L_4 = \{a^{2^n} : n \geq 1\}$

$$S \to T_g \quad T \to T_f|UU \quad U_f \to UU \quad U_g \to a$$

$$\begin{aligned} S(z) &= T[0,1](z) \\ T[m,n] &= T[m+1,n] + U[m,n]^2 \\ U[m,n] &= U[m-1,n]^2 \\ U[0,1] &= z \end{aligned}$$

Simplifications: $U[m,1] = z^{2^n} \implies S(z) = \sum z^{2^n}$, for much the same reasons as the other examples.

$L_7 = \{ab^{i_1}ab^{i_2}\ldots ab^{i_k} | 0 \leq i_1 \leq i_2 \leq \cdots \leq i_k\}$

$$\begin{array}{lll} S \to T_g & T \to G|GT & T \to T_f \\ G_f \to Gb & G_g \to a & \end{array}$$

$$\begin{aligned} S(z) &= T[0,1](z) \\ T[m,n](z) &= G[m,n](z) + G[m,n]T[m,n](z) + T[m+1,n](z) \\ G[m,n](z) &= zG[m-1,n](z), \qquad G[0,1](z) = z \implies G[m,1](z) = z^{m+1} \end{aligned}$$

Simplifications:

$$T[m,1](z) = G[m,1] + G[m,1]T[m,1] + T[m+1,1] \implies T[m,1](z) = \frac{z^{m+1} + T[m+1,1](z)}{1 - z^{m+1}}$$

$$\implies T[0,1](z) = \frac{z + T[1,1](z)}{1-z} = \frac{z + z^2 + T[2,1](z)}{(1-z)(1-z^2)} = \frac{z + z^2 + z^3 + T[3,1](z)}{(1-z)(1-z^2)(1-z^3)} = \ldots = \frac{z}{1-z}\prod_{i>}\frac{1}{1-z^i}$$

close??? where does the extra factor come in??? Is the grammar ambiguous?

References

[1] Alfred V. Aho. Indexed grammars—an extension of context-free grammars. *J. Assoc. Comput. Mach.*, 15:647–671, 1968.

[2] Robert H. Gilman. A shrinking lemma for indexed languages. *Theoret. Comput. Sci.*, 163(1-2):277–281, 1996.

[3] Robert H. Gilman, Derek F. Holt, and Sarah Rees. Combing nilpotent and polycyclic groups. *Internat. J. Algebra Comput.*, 9(2):135–155, 1999.

[4] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to automata theory, languages, and computation*. Addison-Wesley Publishing Co., Reading, Mass., 1979. Addison-Wesley Series in Computer Science.