Self-Generating Sets, Missing Blocks, and Substitutions

David Failing
Truman State University
Kirksville, Missouri

November 12, 2005

Introduction

- Long-range order is a characterization of a system of objects which exhibits local correlation.
- One such system arises in substitution sequences which generate infinite words.
- We explore properties of such systems.

Substitutions

- A <u>substitution</u> or <u>morphism</u> is a function which maps $\sigma : A^* \rightarrow A^*$ where:
  - the alphabet A is a finite set of symbols
  - A^* is a set of nonempty strings of characters over A, satisfying $\sigma(xy) = \sigma(x)\,\sigma(y)$.

Example 1

- Let σ be a substitution on the finite alphabet $A=\{a,b\}$, defined $\sigma(a) = ab$, $\sigma(b) = a$.
    - $\sigma(a) = ab$
      $\sigma^2(a) = \sigma(ab) = aba$
      $\sigma^3(a) = \sigma(aba) = abaab$
  - $\sigma^n(a)$ approaches abaababaabaabab…, the Fibonacci word.

The (Infinite) Fibonnaci Word

- From previous example, the Fibonacci word is abaababaabaabab…
- $\sigma^n(a) = \sigma^{n-1}(a)\ \sigma^{n-2}(a)$
- $|\ \sigma^n(a)\ | = f_n$, the n-th Fibonacci number.

The Fibonacci Base

- Recall $f_n = f_{n-1} + f_{n-2}$ where $f_0=f_1=1$
- $f_n = \{1, 2, 3, 5, 8, 13, 21, 34,\ldots\}$
- We define the Fibonacci base such that the places are successive terms from the Fibonacci sequence.

Sequence Bases

- Take any sequence a_0, a_1, a_2… We define a sequence base to be an alternative numeration system in which we substitute the terms of the sequence for the places.
- Issues of multiple representations of the integers.

Greedy Expansion

- The *Greedy Algorithm* for integer representation is as follows:
  1. Choose some integer n and an integer sequence a_n.
  2. Find the greatest integer j less than n such that j is a member of some integer sequence.

Greedy Expansion

- (Continued)
  3. Repeat above, replacing n with $n-j$ until the chosen $j = n$.
  4. The chosen integer n may be written as the sum of all j chosen in step 2.

Example 2

- Represent *$n=24$* as a Greedy Expansion of the Fibonacci sequence $a_n=\{1,2,3,5,8,\ldots\}$
  1. 21 is in a, and 21 is less than 24.
  2. 24-21=3, and 3 is in a.
  3. 24=21+3, and in the Fibonacci base, 24=1*21+1*3=1 0 0 0 1 0 0

Lazy Expansion

- The *Lazy Algorithm* also allows us to represent integers in a sequence base uniquely, but requires that the integer sequence chosen be defined by a linear combination of the previous two terms.

Lazy Expansion

- (Continued)
  1. Choose some integer *n*.
  2. Find the Greedy Expansion of n, using some recursive sequence *a*.
  3. For $a=A*a_{n-1}+B*a_{n-2}$, replace all 1 0 0 blocks in the Greedy Expansion with 0 A B blocks.
  4. Repeat 3 until no 1 0 0 blocks remain.

Example 3

- Recall the Greedy Expansion of 24 in the Fibonacci base, 24 = 1 0 0 0 1 0 0.
  1. Recall the Fibonacci sequence, defined $f_n = f_{n-1} + f_{n-2}$.
  2. Replace all 1 0 0 blocks with 0 1 1 blocks, repeating until no 1 0 0 blocks remain.

Example 3 (Continued)

3. 24 = 1 0 0 0 1 0 0
   = 0 1 1 0 0 1 1
   = 0 1 0 1 1 1 1

The *Lazy Expansion* of 24 in the Fibonacci base is 1 0 1 1 1 1.

The Kimberling Sequence

- Define S to be the self-generating set of positive integers determined by the finite generating rules:
  1. 1 is in S;
  2. If x is in S, then $2x$ and $4x-1$ are in S;
  3. Nothing else belongs to S.

$S = \{1, 2, 3, 4, 6, 7, 8, 11, 12, 14, 15, 16, \ldots\}$

Analysis of the Kimberling Sequence

$T = S - 1 = \{0, 1, 2, 3, 5, 6, 7, 10, 11, 13, 14, \ldots\}$

- *T* is also self-generating.
  1. 0 is in *T*;
  2. If *x* is in *T*, then $2x + 1$ and $4x+2$ are in *T*;
  3. Nothing else belongs to *T*.

Further Analysis

- We also note that *T*, with the first term removed, and reduced mod 2 is the infinite Fibonacci word.
- *T* is also the set of integers whose base 2 representations contain no 0 0 block, or the set of valid Lazy Fibonacci Representations.

Generalized Fibonacci Morphisms

- Fibonacci Morphism: $\sigma(0) = 0\ 1$, $\sigma(1) = 0$
- Generalization: $\sigma(0) = 0^n 1$, $\sigma(1) = 0$. We study positive values of n, as n=0 yields an oscillating, non-growing morphism.

Example 4: n=2

- $\sigma(0) = 0\ 0\ 1$, $\sigma(1) = 0$.
  1. $|\ \sigma^n(a)\ | = \{1, 3, 7, 17, 41, \ldots\}$, the sequence $a_n = 2a_{n-1} + a_{n-2}$
  2. Write out the Lazy representations of the integers using the recurrence relation determined above.

Example 4 (Continued)

0 -> 0	7 - > 2 1
1 - > 1	8 - > 2 2
2 - > 2	9 - > 1 0 2
3 - > 1 0	10- > 1 1 0
4 - > 1 1	11- > 1 1 1
5 - > 1 2	12- > 1 1 2
6 - > 2 0	13- > 1 2 0

- Note the Lazy representations using {1,3, 7, 17, …} include no 0 0 or 0 1 blocks, and naturally represent integers in base 3.

Example 4 (Continued)

3. Generate the set T, the integers whose base 3 expansions do not contain a 00 or 01 block.
   T = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, …}
4. Omit the first term of T, and the remaining sequence reduced mod 3, then mod 2 is the infinite word produced by the given morphism 0 -> 0 0 1, 1 -> 0.

Conjecture

- For n greater than or equal to 1:
  1. Generate a few iterations of $\sigma(0) = 0^n1$, $\sigma(1) = 0$. Determine the recurrence relation $a_n=A*a_{n-1}+B*a_{n-2}$ which represents the lengths of successive iterations.
  2. Write out the Lazy representations of the integers using a_n defined above, and look for missing blocks.

Conjecture

- (Continued)

3. Generate the set of integers whose base *n* expansions do not contain the blocks found in step 2. This is the set *T.*
4. Omit the first term of *T*, then reduce the set *T mod (n+1), mod (n), … mod 2*.
5. We arrive at our initial infinite word.

Further Research

- We believe the set $S=T+1$ may be self-generating in a manner similar to the Kimberling sequence.
- For n greater than 2, S reduced repetitively in the manner of our conjecture also generates our infinite word.

Further Research

- We have begun work similar to Kimberling, beginning with self-generating sets and finite generating rules.
- We hope to work from self-generating sets to morphic sequences.

Finite Generating Functions

- For any self generating set *S* where:
  1. 1 is in *S*;
  2. F, a finite family of finite generating functions of form $a^k x-b$, where b is between 0 and a^k
  3. If x is in *S*, and some finite rule *f* is in the family of generating functions, then *f(x)* is in *S*.

Finite Generating Functions

- For any positive integer *a*, it is supposed that S *mod e* may be generated by a substitution.
- We analyze the tree structure of the generating functions, and search for repetitive sub-tree structures to determine our substitution.

Acknowledgements

- Dr. David Garth, MTCS Department, Truman State University
- The Next STEP program, Truman State University
- National Science Foundation, DUE Grant #0431664