UNARY REPRESENTATION OF FIBONACCI NUMBERS AS AN UNRESTRICTED GRAMMAR

The Fibonacci sequence, 1, 1, 2, 3, 5, …, seems to pop up many times in different capacities is software development courses, so it is not a surprise to take a look at where is fits in the study of grammars. In a 1991 paper Moll and Venkatsan demonstrated that the language of Fibonacci numbers is not a Context Free Language. In a follow up to the Moll and Vekatsan paper Mootha looks at the unary representations of Fibonacci numbers and demonstrates that the language of unary representations of Fibonacci numbers is Context Sensitive. Along with this paper we provide JFLAP and JFLAP v8 copies of Mootha's unrestricted grammar for the language of unary representations of Fibonacci numbers.

	LHS		RHS
1	S	⟶	0
2	S	⟶	AE0B0CD
3	AE	⟶	AH
4	H0	⟶	F0
5	F00	⟶	0F0
6	F0B	⟶	BF0
7	F0C	⟶	GC0
8	0G	⟶	G0
9	BG	⟶	GB
10	AG	⟶	AH
11	AHB	⟶	ABJ
12	BJ0	⟶	0BK
13	K0	⟶	0K
14	KC	⟶	LC0
15	0L	⟶	L0
16	BL	⟶	BJ
17	BJC	⟶	BM
18	M0	⟶	0M
19	MD	⟶	NCD
20	0N	⟶	N0
21	BN	⟶	NB
22	AN	⟶	AE
23	AE	⟶	P
24	P0	⟶	0P
25	PB	⟶	P
26	PCD	⟶	λ

UNRESTRICTED GRAMMAR

This module makes direct us of the notation in Mootha's article so that the reader can move directly from the this module to the article. The productions, illustrated in the table at the right, come directly from Mootha's article and have been entered into JFLAP (both v7 and v8) to provide students with an opportunity to work with this grammar. Mootha uses "0" as his unary symbol, meaning that a number n is represented by a string of n 0s.

Paraphrasing from Mootha's article, productions 1 generates the first two Fibonacci numbers and production 2 generates F_n for n >2. Each application of production 3 eventually leads to a string of the form,

AE0...B0...CD

where between E and B is F_{n-1}, the unary representation the n-1 Fibonacci number and between B and C is F_{n-2}, the unary representation of the n-2 Fibonacci number. Productions 4 through 22 process the string translating

AE F_{n-2} B F_{n-1} CD into AE F_{n-1} B F_n CD.

The productions successfully terminate with an application of production 23, which along with productions 24, 25, and 26, remove all markers (non terminal symbols) leaving the unary representation of a Fibonacci number.

Derivation View

Step | Complete | Reset

Derivation Tree | Deriv

Production	Derivation
	S
S->A E 0 B 0 C D	A E 0 B 0 C D
A E->A H	A H 0 B 0 C D
H 0->F 0	A F 0 B 0 C D
F 0 B->B F 0	A B F 0 0 C D
F 0 0->0 F 0	A B 0 F 0 C D
F 0 C->G C 0	A B 0 G C 0 D
0 G->G 0	A B G 0 C 0 D
B G->G B	A G B 0 C 0 D
A G->A H	A H B 0 C 0 D
A H B->A B J	A B J 0 C 0 D
B J 0->0 B K	A 0 B K C 0 D
K C->L C 0	A 0 B L C 0 0 D
B L->B J	A 0 B J C 0 0 D
B J C->B M	A 0 B M 0 0 D
M 0->0 M	A 0 B 0 M 0 D
M 0->0 M	A 0 B 0 0 M D

The productions for the unrestricted grammar for unary Fibonacci numbers is in the file Fibonacci.jflap. To illustrate consider the unary representation of the Fibonacci string, 00000. After applying the brute force parse that leads to 00000 go to JFLAP's **Derivation View** to see the step by step application of productions leading to the string 00000. The sequence of figures on the right, and on subsequent pages, illustrate the complete

Derivation View.

The productions 3 through 21, inclusive, perform the translation of the string from the nth to the n+1st Fibonacci number, which is completed with the application of production 23, as highlighted in the figure ot the right. The highlighted entry from the **Derivation Table** illustrates that the 4th Fibonacci number,

AE0B00CD

has been formed from the 2nd and 3rd Fibonacci numbers, along with their intermediate markers.

Production	Derivation
M D->N C D	A 0 B 0 0 N C D
0 N->N 0	A 0 B 0 N 0 C D
0 N->N 0	A 0 B N 0 0 C D
B N->N B	A 0 N B 0 0 C D
0 N->N 0	A N 0 B 0 0 C D
A N->A E	A E 0 B 0 0 C D
A E->A H	A H 0 B 0 0 C D
H 0->F 0	A F 0 B 0 0 C D
F 0 B->B F 0	A B F 0 0 0 C D
F 0 0->0 F 0	A B 0 F 0 0 C D
F 0 0->0 F 0	A B 0 0 F 0 C D
F 0 C->G C 0	A B 0 0 G C 0 D
0 G->G 0	A B 0 G 0 C 0 D
0 G->G 0	A B G 0 0 C 0 D
B G->G B	A G B 0 0 C 0 D
A G->A H	A H B 0 0 C 0 D
A H B->A B J	A B J 0 0 C 0 D
B J 0->0 B K	A 0 B K 0 C 0 D
K 0->0 K	A 0 B 0 K C 0 D
K C->L C 0	A 0 B 0 L C 0 0 D
0 L->L 0	A 0 B L 0 C 0 0 D

Application of production 3 yields,

AE0B00CD

and begins the set of production that transforms the 4th Fibonacci number into the 5th Fibonacci number, which is highlighted below in the figure to the right.

B L->B J	A 0 B J 0 C 0 0 D
B J 0->0 B K	A 0 0 B K C 0 0 D
K C->L C 0	A 0 0 B L C 0 0 0 D
B L->B J	A 0 0 B J C 0 0 0 D
B J C->B M	A 0 0 B M 0 0 0 D
M 0->0 M	A 0 0 B 0 M 0 0 D
M 0->0 M	A 0 0 B 0 0 M 0 D
M 0->0 M	A 0 0 B 0 0 0 M D
M D->N C D	A 0 0 B 0 0 0 N C D
0 N->N 0	A 0 0 B 0 0 N 0 C D
0 N->N 0	A 0 0 B 0 N 0 0 C D
0 N->N 0	A 0 0 B N 0 0 0 C D
B N->N B	A 0 0 N B 0 0 0 C D
0 N->N 0	A 0 N 0 B 0 0 0 C D
0 N->N 0	A N 0 0 B 0 0 0 C D
A N->A E	A E 0 0 B 0 0 0 C D
A E->P	P 0 0 B 0 0 0 C D
P 0->0 P	0 P 0 B 0 0 0 C D
P 0->0 P	0 0 P B 0 0 0 C D
P B->P	0 0 P 0 0 0 C D
P 0->0 P	0 0 0 P 0 0 C D

A careful analysis of the productions between the two highlighted productions illustrates that starting with the string,

AE F_{n-2} B F_{n-1} CD

the productions are doing two things:

1. F_{n-1} is ***copied*** to the left to eventually appear to the left of the E marker, A E F_{n-1} F_{n-2} B F_{n-1} CD.
2. The B marker is moved to the left of the yielding, A E F_{n-1} B F_{n-2} F_{n-1} CD.

Since F_{n-2} F_{n-1} forms the unary representation of F_n the string is correctly formed.

The only remaining task is removing the markers from the string, which is started with the application of production 23 followed by applications of productions 24 and 25, and, finally, leading to the application of production 26, producing the desired result, which appear on the figure at the top of the next page.

CONTEXT SENSITIVE GRAMMAR

Because of productions 23, 25, and 26, the grammar shown above is an unrestricted grammar. Many formal language texts present methods for changing an unrestricted grammar into a context sensitive grammar. For example, the classic

Hopcroft and Ullman text, see Exercise 9.5, present a process for eliminating productions that replaces productions where the right had side of the production is smaller that the left hand side with context sensitive productions.

Mootha's article outlines the process of creating the new set of productions based on the original productions by reorganizing the original productions into a new set of productions that satisfy the context sentitive requirement and mimic the productions of the original grammar.The table below shows the productions of the contexts sensitive grammar grouped together to demonstrate the relationship between the context sensitive grammar and original unrestricted grammar.

P 0->0 P	0 0 0 0 P 0 C D
P 0->0 P	0 0 0 0 0 P C D
P C D->λ	0 0 0 0 0

1) $[S] \to 0$

2) $[S] \to [AE0][B0CD]$

3) $[AE0] \to [AH0]$

4) $[AH0] \to [AF0]$

5) $[ABF0][0CD] \to [AB0][F0CD]$
$[ABF0]0 \to [AB0][F0]$
$[F0][0CD] \to 0[F0CD]$
$[AF0]0 \to [A0][F0]$
$[BF0]0 \to [B0][F0]$
$[F0]0 \to 0[F0]$

6) $[AF0][B0CD] \to [ABF0][0CD]$
$[AF0][B0] \to [ABF0]0$
$[F0][B0] \to [BF0]0$

7) $[F0CD] \to [GC0D]$
$[F0][C0D] \to [GC0][0D]$

8) $[AB0][GC0D] \to [ABG0][C0D]$
$0[GC0D] \to [G0][C0D]$
$[AB0][G0] \to [ABG0]0$
$0[G0] \to [G0]0$
$[B0][G0] \to [BG0]0$
$[A0][GB0] \to [AG0][B0]$
$0[GC0] \to [G0][C0]$

14) $[BKC0D] \to [BLC0][0D]$
$[KC0D] \to [LC0][0D]$
$[BKC0] \to [BLC0]0$
$[KC0] \to [LC0]0$

15) $[B0][LC0] \to [BL0][C0]$
$0[LC0] \to [L0][C0]$
$[B0][L0] \to [BL0]0$

16) $[BLC0] \to [BJC0]$
$[BL0] \to [BJ0]$

17) $[BJC0] \to [BM0]$

18) $[BM0][0D] \to [B0][M0D]$
$[M0D] \to [0MD]$
$[BM0]0 \to [B0][M0]$
$[M0][0D] \to 0[M0D]$
$[M0]0 \to 0[M0]$

19) $[0MD] \to [0NCD]$

20) $[0NCD] \to [N0CD]$
$[B0][N0CD] \to [BN0][0CD]$
$[A0][NB0] \to [AN0][B0]$
$0[N0CD] \to [N0][0CD]$
$[B0][N0] \to [BN0]0$
$0[NB0] \to [N0][B0$
$[A0][N0] \to [AN0]0$
$0[N0] \to [N0]0$

9) $[ABG0] \to [AGB0]$
$[BG0] \to [GB0]$

10) $[AGB0] \to [AHB0]$
$[AG0] \to [AH0]$

11) $[AHB0] \to [ABJ0]$

12) $[ABJ0][C0D] \to [A0][BKC0D]$
$[ABJ0]0 \to [A0][BK0]$
$[A0][BJ0][C0] \to [A0]0[BKC0]$
$[BJ0]0 \to 0[BK0]$
$[BJ0][C0] \to 0[BKC0]$

13) $[BK0][C0D] \to [B0][KC0D]$
$[BK0]0 \to [B0][K0]$
$[K0][C0] \to 0[KC0]$
$[BK0][C0] \to [B0][KC0]$
$[K0]0 \to 0[K0]$

21) $[BN0] \to [NB0]$

22) $[AN0] \to [AE0]$

23) $[AE0] \to [P0]$

24) $[P0]0 \to 0[P0]$
$[P0][B0] \to 0[PB0]$
$[P0][0CD] \to 0[P0CD]$
$[P0CD] \to [0PCD]$

25) $[PB0] \to [P0]$

26) $[0PCD] \to 0$

SUMMARY

This module is based on three references:

1. J. Hopcroft & J. Ullman. *Introduction to Automata Theory, Languages, and Computation*. New York: Addison-Wesley, 1979.
2. R. Mo; & Venkatesan. "Fibonacci Numbers are Not Context-Free." *Fibonacci Quarterly* **29.1** (1991): 59-61.
3. V K Mootha. "Unary Fibonacci Numbers are Context Sensitive". *Fibonacci Quarterly* **31.1** (1993): 41-43.

Mootha's article was written when he was a student at Stanford. He is currently (2114) a computational biologist at Harvard's Howard Hughes Medical Institute.