

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 in software development courses, so it is not a surprise to take a look at where it 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 Venkatsan 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.

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, production 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

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

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.

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

	LHS	RHS
1	S	→ 0
2	S	→ AE0B0CD
3	AE	→ AH
4	H0	→ F0
5	F00	→ OF0
6	F0B	→ BF0
7	F0C	→ GC0
8	0G	→ G0
9	BG	→ GB
10	AG	→ AH
11	AHB	→ ABJ
12	BJ0	→ OBK
13	K0	→ OK
14	KC	→ LCO
15	0L	→ LO
16	BL	→ BJ
17	BJC	→ BM
18	M0	→ OM
19	MD	→ NCD
20	0N	→ N0
21	BN	→ NB
22	AN	→ AE
23	AE	→ P
24	P0	→ OP
25	PB	→ P
26	PCD	→ λ

Derivation View	
Step	Complete Reset
Derivation Tree Deriv	
Production	Derivation
	S
S → AE0B0CD	AE0B0CD
AE → AH	AH0B0CD
H0 → F0	AF0B0CD
F0B → BF0	ABF00CD
F00 → OF0	AB0F0CD
F0C → GC0	AB0GC0D
0G → G0	ABG0C0D
BG → GB	AGB0C0D
AG → AH	AHB0C0D
AHB → ABJ	ABJ0C0D
BJ0 → OBK	A0BK0C0D
KC → LCO	A0BLC00D
BL → BJ	A0BJC00D
BJC → BM	A0BM00D
M0 → OM	A0B0M0D
M0 → OM	A0B00MD

Derivation View.

The productions 3 through 21, inclusive, perform the translation of the string from the n th to the $n+1$ st Fibonacci number, which is completed with the application of production 23, as highlighted in the figure to 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.

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.

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

Production	Derivation
MD->NCD	A0B00NCD
ON->NO	A0B0N0CD
ON->NO	A0BN00CD
BN->NB	A0NB00CD
ON->NO	AN0B00CD
AN->AE	AE0B00CD
AE->AH	AH0B00CD
H0->F0	AF0B00CD
F0B->BF0	ABF000CD
F00->0F0	AB0F00CD
F00->0F0	AB00F0CD
F0C->GCC0	AB00GC0D
0G->G0	AB0G0C0D
0G->G0	ABG00C0D
BG->GB	AGB00C0D
AG->AH	AHB00C0D
AHB->ABJ	ABJ00C0D
BJ0->0BK	A0BK0C0D
K0->0K	A0B0KC0D
KC->LC0	A0B0LC00D
0L->L0	A0BL0C00D
BL->BJ	A0BJ0C00D
BJ0->0BK	A00BK0C00D
KC->LC0	A00BLC000D
BL->BJ	A00BJC000D
BJC->BM	A00BM000D
M0->0M	A00B0M00D
M0->0M	A00B00M0D
M0->0M	A00B000MD
MD->NCD	A00B000NCD
ON->NO	A00B00N0CD
ON->NO	A00B0N00CD
ON->NO	A00BN000CD
BN->NB	A00NB000CD
ON->NO	A0N0B000CD
ON->NO	AN00B000CD
AN->AE	AE00B000CD
AE->P	P00B000CD
P0->0P	0P0B000CD
P0->0P	00PB000CD
PB->P	00P000CD
P0->0P	000P00CD

Hopcroft and Ullman text, see Exercise 9.5, present a process for eliminating productions that replaces productions where the right hand side of the production is smaller than 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 sensitive requirement and mimic the productions of the original grammar. The table below shows the productions of the context sensitive grammar grouped together to demonstrate the relationship between the context sensitive grammar and original unrestricted grammar.

$P0 \rightarrow 0P$	0 0 0 0 P 0 C D
$P0 \rightarrow 0P$	0 0 0 0 0 P C D
$PCD \rightarrow \lambda$	0 0 0 0 0

1) $[S] \rightarrow 0$	14) $[BKC0D] \rightarrow [BLC0][0D]$ $[KCO] \rightarrow [LCO][0D]$ $[BKC0] \rightarrow [BLC0]0$ $[KCO] \rightarrow [LCO]0$	9) $[ABG0] \rightarrow [AGB0]$ $[BG0] \rightarrow [GB0]$	21) $[BN0] \rightarrow [NB0]$
2) $[S] \rightarrow [AE0][B0CD]$		10) $[AGB0] \rightarrow [AHB0]$ $[AG0] \rightarrow [AH0]$	22) $[AN0] \rightarrow [AE0]$
3) $[AE0] \rightarrow [AH0]$	15) $[B0][LCO] \rightarrow [BL0][CO]$ $0[LCO] \rightarrow [L0][CO]$ $[B0][L0] \rightarrow [BL0]0$	11) $[AHB0] \rightarrow [ABJ0]$	23) $[AE0] \rightarrow [P0]$
4) $[AH0] \rightarrow [AF0]$	16) $[BLC0] \rightarrow [BJCO]$ $[BL0] \rightarrow [BJ0]$	12) $[ABJ0][C0D] \rightarrow [A0][BKC0D]$ $[ABJ0]0 \rightarrow [A0][BK0]$ $[A0][BJ0][C0] \rightarrow [A0]0[BKC0]$ $[BJ0]0 \rightarrow 0[BK0]$ $[BJ0][C0] \rightarrow 0[BKCO]$	24) $[P0]0 \rightarrow 0[P0]$ $[P0][B0] \rightarrow 0[PB0]$ $[P0][0CD] \rightarrow 0[P0CD]$ $[P0CD] \rightarrow [0PCD]$
5) $[ABF0][0CD] \rightarrow [AB0][F0CD]$ $[ABF0]0 \rightarrow [AB0][F0]$ $[F0][0CD] \rightarrow 0[F0CD]$ $[AF0]0 \rightarrow [A0][F0]$ $[BF0]0 \rightarrow [B0][F0]$ $[F0]0 \rightarrow 0[F0]$	17) $[BJCO] \rightarrow [BM0]$		25) $[PB0] \rightarrow [P0]$
6) $[AF0][B0CD] \rightarrow [ABF0][0CD]$ $[AF0][B0] \rightarrow [ABF0]0$ $[F0][B0] \rightarrow [BF0]0$	18) $[BM0][0D] \rightarrow [B0][M0D]$ $[M0D] \rightarrow [0MD]$ $[BM0]0 \rightarrow [B0][M0]$ $[M0][0D] \rightarrow 0[M0D]$ $[M0]0 \rightarrow 0[M0]$	13) $[BK0][C0D] \rightarrow [B0][KCO]$ $[BK0]0 \rightarrow [B0][K0]$ $[K0][C0] \rightarrow 0[KCO]$ $[BK0][C0] \rightarrow [B0][KCO]$ $[K0]0 \rightarrow 0[K0]$	26) $[0PCD] \rightarrow 0$
7) $[F0CD] \rightarrow [GC0D]$ $[F0][C0D] \rightarrow [GC0][0D]$	19) $[0MD] \rightarrow [0NCD]$		
8) $[AB0][GC0D] \rightarrow [ABG0][C0D]$ $0[GC0D] \rightarrow [G0][C0D]$ $[AB0][G0] \rightarrow [ABG0]0$ $0[G0] \rightarrow [G0]0$ $[B0][G0] \rightarrow [BG0]0$ $[A0][GB0] \rightarrow [AG0][B0]$ $0[GC0] \rightarrow [G0][C0]$	20) $[0NCD] \rightarrow [N0CD]$ $[B0][N0CD] \rightarrow [BN0][0CD]$ $[A0][NB0] \rightarrow [AN0][B0]$ $0[N0CD] \rightarrow [N0][0CD]$ $[B0][N0] \rightarrow [BN0]0$ $0[NB0] \rightarrow [N0][B0]$ $[A0][N0] \rightarrow [AN0]0$ $0[N0] \rightarrow [N0]0$		

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.