Simplifying Notes

- To simplify your life, we are using a suffix notation, e.g.:

A -> R(A, A)
S -> C(A,A)

Or we could have used an infix notation:

A -> (A)R(A)
S -> (A)C(A)

Grammar

Rule #	Production Rule	Rule #	Production Rule
1	S -> A	16	R -> *from_to*
2	S -> C(A,A)	17	N -> *not*
3	S -> U	18	D -> *younger_than*
4	A -> B	19	D -> *older_than*
5	A -> R(B,B)	20	I -> *indeed*
6	A -> I B	21	I -> *hardly*
7	B -> E	22	V -> *extremely*
8	B -> N E	23	H -> *very*
9	B -> D E	24	H -> *more_or_less*
10	E -> T	25	T -> *young*
11	E -> H T	26	T -> *middle_aged*
12	E -> V T	27	T -> *old*
13	C -> *and*	28	U -> *any_age*
14	C -> *or*	29	U -> *no_age*
15	R -> *in_between*	30	U -> *unknown_age*

Algorithm

- 1) Initialize string from Starting symbol
- 2) Scan current string and identify all non-terminals
- 3) if there aren't any, STOP
- 4) Scan grammar and identify all rules containing non-terminals in Left Hand Side

		Randomly	
Current String	**Possible Rules**	**selected**	**New**
		rule	**string**
S	(1,2,3)		

- 5) Select one random rules from set of Possible Rules

		Randomly	
Current String	**Possible Rules**	**selected**	**New**
		rule	**string**
S	(1,2,3)	2	

- 6) Apply selected rule to first instance of non_terminal

		Randomly	
Current String	**Possible Rules**	**selected**	**New**
		rule	**string**
S	(1,2,3)	2	C(A,A)

- 7) Go to (2)

Example

Current String	Possible Rules	Randomly selected rule	New string
S	(1,2,3)	2	C(A,A)
C(A,A)	(13,14,4,5,6)	13	*and(*A,A*)*
*and(*A,A*)*	(4,5,6)	4	*and(*B,A*)*
*and(*B,A*)*	(7,8,9,4,5,6)	8	*and(*N E,A*)*
*and(*N E,A*)*	(17,10,11,12,4,5,6)	12	*and(*N V T,A*)*
*and(*N V T,A*)*	17,22,25,26,27,4,5,6	22	*and(*N *extremely* T,A*)*
*and(*N *extremely* T,A*)*	(17,25,26,27,4,5,6)	4	*and(*N *extremely* T,B*)*
*and(*N *extremely* T,B*)*	(17,25,26,27,7,8,9)	7	*and(*N *extremely* T,E*)*
*and(*N *extremely* T,E*)*	17,25,26,27,10,11,12	10	*and(*N *extremely T,T)*
*and(*N *extremely T,T)*	(17,25,26,27)	25	*and(*N *extremely young,* T*)*
*and(*N *extremely young,* T*)*	(17,25,26,27)	17	*and(not extremely young,* T*)*
and(not extremely young, T*)*	(25,26,27)	25	*and(not extremely young,young)*
and(not extremely young,young)	()		

Semantic interpereter

- Define functions and functions composition so that you can evaluate

  and(not extremely young,young):

  or

  and(not (extremely (young)),young):

  MIN[(1- young^4),young]

  where *young* is a vector of dimension D