Parsing Problem - Given a Grammar G and a string $w \in T^*$, is $w \in L(G)$?

Younger's Algorithm (This is not a standard parsing algorithm. It is very easy to understand but is not efficient enough to be used in practice.)

Grammar:

$S \rightarrow bA$ $\quad S \rightarrow aB$
$A \rightarrow a$ $\quad B \rightarrow b$
$A \rightarrow aS$ $\quad B \rightarrow bS$
$A \rightarrow bAA$ $\quad B \rightarrow aBB$

String: $w = aaabbaabbb$

(1) First step: put grammar in Chomsky Normal Form.

$S \rightarrow C_1 A$ $\quad C_3 \rightarrow AA$
$A \rightarrow a$ $\quad S \rightarrow C_2 B$
$B \rightarrow C_2 C_4$ $\quad C_4 \rightarrow BB$ $\quad A \rightarrow C_2 S$ $\quad B \rightarrow b$
$C_1 \rightarrow b$ $\quad C_2 \rightarrow a$ $\quad A \rightarrow C_1 C_3$ $\quad B \rightarrow C_1 S$

2. construct recognition matrix REC. let $a_1 a_2 \ldots a_n$ be the sentence we are attempting to recognize. $REC(I,J)$ will contain all nonterminals from which $a_j a_{j+1} \ldots a_{j+i-1}$ can be derived.

J

	1	2	3	4	5	6	7	8	9	10
	a	a	a	b	b	a	a	b	b	b
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

accept string if S is contained in REC(n, 1) after matrix is computed.

amt of work needed:

$$n + \sum_{i=2}^{n} (n-i+1)\,(i-1)$$

$$\sim n^3$$

By noticing the similarity of this algorithm to matrix multiplication and using strassen's trick, a clever Post doctoral student at CMU was able to modify the algorithm to run in time $n^{2.78}$. No one knows if there is a better algorithm or not.