

Case 10

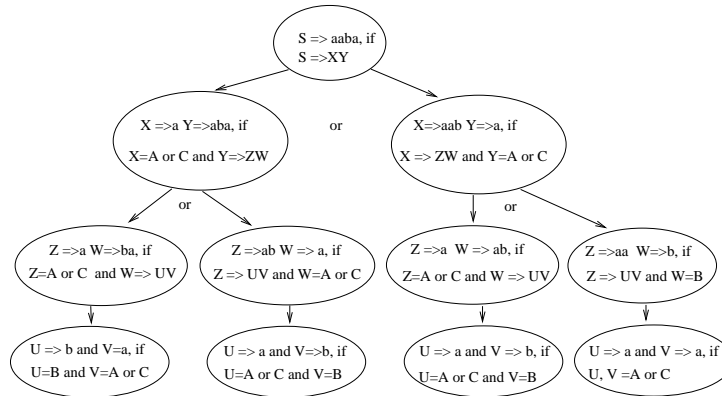
This week we have only one problem, which can give you **6 points**.

Your boss has given you a lot of context-free languages and you should make a parser for each of them in one week! When you get over your worst despair, you notice that all productions of the given languages can be represented in form $A \rightarrow BC$ or $A \rightarrow a$ and at most the starting symbol S is nullable (i.e. for it there can be a production $S \rightarrow \epsilon$).

You first consider some kind of recursive parser, which would study all possible derivations of the string. E.g. in grammar

$$\begin{aligned} S &\rightarrow AB|BA \\ A &\rightarrow CC|a \\ B &\rightarrow BC|b \\ C &\rightarrow AB|a \end{aligned}$$

the string $aaba$ belongs to the language, if holds:



However you notice that this kind of top down parsing is really unefficient and the parser does a lot of useless work. Could the parse trees be constructed from bottom to up, i.e. consider first, from which symbols the leaves (terminals symbols) are derived, then from which symbols these symbols are derived and so on, until you reach the root (which should be S)?

a) Construct a general parsing algorithm for all grammars of the this form (so called *Chomsky normal form*). Test your idea at least with the previous grammar and the strings $aababa$ and $abbba$.

b) What about if the given grammar is not in that form? Could you still somehow transform it into the form your program requires?