SSC1 exercise sheet: parsing

Consider the following grammar:

\[
P \rightarrow \ P \ast \ P \\
P \rightarrow \ 2
\]

The symbols \( \ast \) and \( 2 \) are terminals. The start symbol is \( P \).

(a) Translate the grammar above into Java classes suitable for representing abstract syntax trees.

(b) Explain what is wrong with the grammar above for the purpose of predictive parsing.

(c) Rewrite the grammar so that it becomes suitable for predictive parsing and translate the new grammar to Java methods. (The methods need not build the parse tree, so they can have return type void.) Hint: you need a new non-terminal \( M \) and a rule

\[
P \rightarrow \ 2 \ M
\]

Here the non-terminal \( M \) represents strings of the form

\[
\ast \ 2 \ast \ 2 \ \ldots \ \ast \ 2
\]

You will need two rules for \( M \), one for reading more \( \ast \ P \) and one for stopping. You can assume that the symbol $ marks the end of the input, which will provide one of the case labels.