COMPLEXITY ANALYSIS OF THE EARLEY ALGORITHM

CONSTANT PARAMETERS
k: number of the rules of grammar G
l: max length of the right part of the rules of G (for example A » AaB has length 3)

VARIABLE PARAMETERS
n: length of the string to be analysed

COMPUTATION OF THE COMPLEXITY
l + 1 max length of a rule of G (right part + left part)
n + 1 max value range of an integer pointer (from 0, included, to n, included)
m = k * (l + 1) * (n + 1): max number of rules with marker and integer pointer
m gives the max number of rules with marker and integer pointer that can stay together in an Earley state
m thus gives the max size of an Earley state

SPACE COMPLEXITY = STATE MAX SIZE * NUMBER OF STATES = m * n = k * (l + 1) * (n + 1) * n = O(n^2)

TIME COMPLEXITY = STATE MAX SIZE * TIME TO SCAN ALL THE RULES IN THE STATE * NUMBER OF STATES =
m * m * n = (k * (l + 1) * (n + 1))^2 * n = O(n^3)

UNDERLYING HYPOTHESIS: a rule with marker and integer pointer is stored in a single memory cell
independently of the lenth of the rule, and in order to mark, unmark and copy rules,
as well as to update the integer pointer, a single time unit is consumed, independently of the length of the rule.

The complexity analysis may be further refined by expressing the costs of the elementary operations in terms
of simpler instructions, but this would simply change the proportionality constant of the cost, not the dependence
on the length n of the string to be analysed, which remains cubic.

It is possible to prove that, if the grammar is not ambiguous, the time complexity would fall to quadratic only
(the proof however requires a non-trivial modification of the algorithm).