Compiler Construction

Describe how a parse tree can be translated into a sequence of assembly language instructions based on a pattern matching graph derived from a set of tree rewriting rules where each rule has a cost and a corresponding fragment of code. Illustrate your answer using the following rules:

\[
\begin{align*}
R_i &= K_k & \text{LDI } R_i,K_k & \text{Cost 2} \\
R_i &= \text{add}(R_i,K_k) & \text{ADDI } R_i,K_k & \text{Cost 3} \\
R_i &= \text{add}(R_i,R_j) & \text{ADD } R_i,R_j & \text{Cost 3} \\
R_i &= \text{add}(R_i,\text{add}(R_j,K_k)) & \text{ADD } R_i,R_j,K_k & \text{Cost 4}
\end{align*}
\]

applied to the following parse tree:

\[
\text{add}(K_1,\text{add}(\text{add}(K_2,\text{add}(K_3,K_4)),\text{add}(K_5,K_6)))
\]

[15 marks]

Discuss the advantages and disadvantages of this approach to code generation.

[5 marks]