Compiler Construction

Describe an efficient tree pattern-matching algorithm that could be used to find a cheapest covering of an abstract syntax tree by pattern templates with given costs. Illustrate your algorithm using the following templates:

#1 \( R \leftarrow k \) \hspace{1cm} \text{cost: 1}
#2 \( R \leftarrow f(R, k) \) \hspace{1cm} \text{cost: 2}
#3 \( R \leftarrow f(R, R) \) \hspace{1cm} \text{cost: 2}
#4 \( R \leftarrow f(R, f(R, k)) \) \hspace{1cm} \text{cost: 3}
#5 \( R \leftarrow f(f(R, k), R) \) \hspace{1cm} \text{cost: 4}

and the following tree:

\[ f(f(k, k), f(k, k)) \]