Analysis of REGS

BOI Scientific Committee

This task has three different solutions we are aware of:

- back-tracking,
- dynamic programming,
- greedy algorithm with elements of dynamic programming.

The complexity of solutions will be expressed in terms of

- N - number of registers,
- M - number of nodes in the tree,
- K - maximum arity of functions.

The simplest way to solve this task is to have a function $\mathbf{cost}(X, Q)$, which given the number of available registers Q returns the total cost of computing expression X. In its body all possible permutations ρ_X and sets τ_X are generated. Then this function is applied recursively to the children of X and the optimal cost of evaluating X is computed. The complexity of such approach is very high and only very small input data can be solved with it.

The standard approach to improve such solution is to memorize computed values of $\mathbf{cost}(X, Q)$ for each pair of X and Q. This gives complexity $O(N \cdot M \cdot K! \cdot 2^K)$. This solution can be improved based on some observations about the interplay of ρ_X and τ_X. First, there exists an optimal solution where elements of τ_X constitute the prefix of ρ_X. In this case the mutual order of elements of τ_X according to ρ_X does not matter. These observations are summarized on the following diagram:

$$\overbrace{\{T_1, \ldots, T_{|\tau_X|}\}}^{\tau_X}, \underbrace{T_{|\tau_X|+1}, \ldots, T_{K_X}}_{\overline{\tau}_X}$$

Here K_X is the arity of function F_X, $T_i = \rho_X(i)$, and $\overline{\tau}_X$ denotes the complement of τ_X. When using these observations the complexity of the solution is $O(N \cdot M \cdot K!)$.

For the last solution we need to introduce the following definition:

Definition 1. *For given expression X the* perfect number of registers $\alpha(X)$ *is the minimum number of registers which allow computing X without unloading intermediate values.*

Our claim is that there exists an optimal solution such that for each $T' \in \overline{\tau}_X \setminus \{T_{|\tau_X|+1}\}$ when evaluating T' the number of available registers is at least $\alpha(T')$. The proof is as follows: suppose T' gets less than $\alpha(T')$ registers. In this case moving T' to τ_X will not increase the total cost. Another important observation is that there exists an optimal solution where the elements of $\overline{\tau}_X$ have decreasing values of $\alpha(\cdot)$. This leads to the following solution:

1. Compute $\alpha(X)$ for each X. Additionally, precompute $\mathbf{cost}(X, \alpha(X))$ for each X. This can be done bottom-up with complexity $O(M \cdot K \cdot \log K)$.

2. Compute the optimal cost of evaluating the expression using previously defined function $\mathbf{cost}(X, Q)$. Computing $\mathbf{cost}(X, Q)$ involves dynamic programming on the sequence of children of X sorted according to the value of $\alpha(\cdot)$. For each X function $\mathbf{cost}(X, Q)$ is executed at most once with $Q = N$. The total complexity of this step is $O(M \cdot K^2)$.

This gives the total complexity $O(M \cdot K^2 \cdot)$.

Altogether we get four gradually improving solutions having the following complexities:

- impractically high,
- $O(N \cdot M \cdot K! \cdot 2^K)$,
- $O(N \cdot M \cdot K!)$,
- $O(M \cdot K^2)$.

This can serve as a basis for creating a set of tests that gives predetermined amounts of points to different solutions.