Lecture 01/25/17

Lecturer: Xiaodi Wu

Reading Assignment: Chap 17.1-17.3, Note on Amortized Analysis, Chap 17.4 (optional)
The Accounting Method

Principle

► Every primitive operation costs 1-unit money.
► Deposit money whenever performing an operation (amortized complexity). Money spent after every primitive operation.
► Your bank starts with zero-balance and remains non-negative during the whole procedure. No loan!

Correctness

\[
\text{#all primitive ops} \leq \text{#all money deposited} \\
= \text{amortized complexity} \times \# \text{ ops}
\]

\leq \text{due to your balance being non-negative all the time!}
The Accounting Method: Example

Push() & Multi-pop()

- deposit 2$ for each Push(): 1$ is spent to execute the push operation, 1$ is left in the bank for later.
- deposit 0$ for each Multi-pop(): its cost is paid for by the deposit made at the push operation.
- A formal proof requires showing the non-negativity of your balance.

Credit Invariant

- Invariant: # of (bank) credits = # of items in the stack.
- Prove the invariant for each operation: push(), multi-pop().
The Potential Function Method

Principle

- Every primitive operation costs 1-unit energy.
The Potential Function Method

Principle

▶ Every primitive operation costs 1-unit energy.
▶ For each operation, energy cost + potential energy change = amortized complexity.

Mathematics

Let \( \Phi_i \) denote the potential energy right after the \( i \)th op.
\[ \Phi_0 = 0, \Phi_i \geq 0, \forall i. \]

Let \( t_i \) denote the actual running time of the \( i \)th op. Then its amortized running time \( t'_i \) is defined to be
\[ t'_i = t_i + \Phi_i - \Phi_{i-1}. \]
The Potential Function Method

Principle

- Every primitive operation costs 1-unit energy.
- For each operation, energy cost + potential energy change = amortized complexity.
- Potential energy starts with 0 and remains non-negative.
The Potential Function Method

Principle

- Every primitive operation costs 1-unit energy.
- For each operation, energy cost + potential energy change = amortized complexity.
- Potential energy starts with 0 and remains non-negative.

Mathematics

- Let $\Phi_i$ denote the potential energy right after the $i$th op. $\Phi_0 = 0$, $\Phi_i \geq 0$, $\forall i$.
- Let $t_i$ denote the actual running time of the $i$th op. Then its amortized running time $t_i'$ is defined to be

$$t_i' = t_i + \Phi_i - \Phi_{i-1}$$
The Potential Function Method: cont’d

Correctness: total actual running time \( \leq \) total amortized running time

\[
T = \sum_i t_i
= \sum_i (t'_i + \Phi_{i-1} - \Phi_i)
= \sum_i t'_i + \sum_i (\Phi_{i-1} - \Phi_i)
= T' + (\Phi_0 - \Phi_n)
\leq T'
\]

where \( T' = \sum_i t'_i \), the total amortized time of all operations. The second summation simplifies to \((\Phi_0 - \Phi_n)\) due to the telescoping sum.
The Potential Function Method: Example

Setup

▶ Set $\Phi_i =$ # of items in the stack. $\Phi_0 = 0$ and $\Phi_i \geq 0, \forall i$. 
The Potential Function Method: Example

Setup

- Set $\Phi_i =$ # of items in the stack. $\Phi_0 = 0$ and $\Phi_i \geq 0, \forall i$.
- Push() : $t' = t_i + \Phi_i - \Phi_{i-1} = 1 + 1 = 2$, which is $O(1)$. The change in the potential is an increase in one, which combines with the constant-time operations of push to yield a total amortized cost of 2.
The Potential Function Method: Example

Setup

- Set \( \Phi_i = \# \) of items in the stack. \( \Phi_0 = 0 \) and \( \Phi_i \geq 0, \forall i \).
- Push() : \( t' = t_i + \Phi_i - \Phi_{i-1} = 1 + 1 = 2 \), which is \( O(1) \). The change in the potential is an increase in one, which combines with the constant-time operations of push to yield a total amortized cost of 2.
- Multi-pop(): Multipop(k): \( t' = t_i + \Phi_i - \Phi_{i-1} = k + -k = 0 \), which is \( O(1) \). The potential decrease cancels the running time of multi-pop().