

Proof That FF Cache Eviction Is Optimal (adapted from Clyde Kruskal's lecture)

David Tagatac

dtagatac@cs.umd.edu

<http://www.cs.umd.edu/~dtagatac>

February 23, 2012

1 Theorem

Let FF be the cache maintenance algorithm that, upon any cache miss, always evicts the item in cache that will be used farthest in the future. Let Opt be a cache maintenance algorithm that results in the fewest possible cache misses. FF causes no more cache misses than Opt and, thus, is an optimal cache maintenance algorithm.

2 Proof

For a general program that causes cache misses, denote the order in which items are evicted from the cache by FF as the series

$$FF : e_1, e_2, e_3, e_4, \dots, e_{k-1}, e_k, e_{k+1}, e_{k+2}, \dots, e_n$$

where e_i is an element evicted from the cache.

For the same program, assume for the sake of contradiction that Opt causes fewer cache misses than FF. Opt must then evict items in a different order from FF. Assume, without the loss of generality, that the order in which items are evicted from the cache by Opt first differs from FF at e_k . Denote the order in which items are evicted from the cache by Opt as the series

$$Opt : e_1, e_2, e_3, e_4, \dots, e_{k-1}, f_k, f_{k+1}, f_{k+2}, \dots, f_m$$

where $f_k \neq e_k$ and $m < n$.

By the definition of FF, at the time of the eviction of e_k (equivalent to the time of eviction of f_k in Opt), f_k will be used by the program sooner in the future than e_k . Let us now examine the Opt series, focusing on the time between the eviction of f_k and the program's usage of f_k . We must consider two cases.

2.1 Case 1: Opt evicts e_k before the program uses f_k

In this case, Opt can swap e_k and f_k in their order of eviction without causing any extra cache misses. This contradicts the assumption that Opt causes fewer cache misses than FF.

2.2 Case 2: Opt does not evict e_k before the program uses f_k

Opt must bring f_k back into the cache before it is used by the program. In the case that e_k is not evicted before the program uses f_k , instead of evicting f_k , Opt can evict e_k (and bring it back into the cache before f_k is used). As in Case 1, this does not cause any extra cache misses, contradicting the assumption that Opt causes fewer cache misses than FF.

Q.E.D.