NTIN066 - solutions 7

David Mareček

March 29, 2023

1. Prove that for every cache size C and every $\epsilon > 0$, there exists a sequence of requests for which $T_{LRU} \geq (1-\epsilon) \cdot C \cdot T_{OPT}$.

The sequence will consist of K copies of $1, \ldots, C+1$; e.g., for $C = 5$, we will have the following sequence:

$$1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, \ldots$$

If LRU is used and the cache is empty a the beginnning, there will be always the cache miss. The OPT algorithm here would always substitute the last recently used item instead of the least recently used, so when loading number "6", it kicks out the previous "5". Then the following $C-1$ blocks are in cache and the cache miss will be generally only every Cth block. The only exception are the first C blocks that have to be cached into the empty cache first. But for each ϵ, we can find such K, for which $T_{OPT} \leq (1+\epsilon) \cdot N/C)$, where N is the size of the sequence. Then, $T_{LRU} = N \geq T_{OPT} \cdot C \cdot (1-\epsilon)$.

2. Prove that if H is (k, c)-independent for $k > 1$, then it is also $(k-1, c)$-independent

$$P[h(x_1) = a_1 \wedge h(x_2) = a_2 \wedge \cdots \wedge h(x_{k-1}) = a_{k-1}] =$$

$$= \sum_{i=1}^{m} P[h(x_1) = a_1 \wedge h(x_2) = a_2 \wedge \cdots \wedge h(x_{k-1}) = a_{k-1} \wedge h(x_k) = i] \leq$$

$$\leq m \frac{c}{m^k} = \frac{c}{m^{k-1}}$$

3. Prove that if H is $(2, c)$-independent, then it is c-universal.

$$\forall x_1, x_2, a_1, a_2 : P[h(x_1) = a_1 \wedge h(x_2) = a_2] \leq \frac{c}{m^2}$$

$$P[h(x_1) = h(x_2)] = \sum_{i=1}^{m} P[h(x_1) = i \wedge h(x_2) = i] \leq m \frac{c}{m^2} = \frac{c}{m}$$