Relative Competitiveness of Cache Replacement Policies

Jan Reineke Daniel Grund
Saarland University

Introduction

In order to fulfill stringent performance requirements, *caches* are now also used in *hard real-time systems*. In such systems, upper and lower bounds on the execution times of a task have to be computed. To obtain tight bounds, timing analyses *must* take into account the cache architecture. However, *developing cache analyses* – analyses that determine whether a memory access is a hit or a miss – is a *difficult problem* for some cache architectures.

Goal

Determine safe bounds on the number of cache hits and misses by a task T under FIFO(k), PLRU(l), or any another replacement policy.

Approach

1. Determine competitiveness of the desired policy P relative to policy Q.

$$\mathbf{m_P \leq k \cdot m_Q + c}$$

2. Compute performance prediction of task T for policy Q by cache analysis.

$$\mathbf{m_Q(T)}$$

3. Calculate upper bounds on the number of misses for P using the cache analysis results for Q and the competitiveness results of P relative to Q.

$$\mathbf{m_P \leq k \cdot m_Q + c} \quad \mathbf{m_Q(T)} \quad = \quad \mathbf{m_P(T)}$$

Relative Competitiveness

Competitiveness (Sleator and Tarjan): worst-case performance of an online policy relative to the *optimal offline policy*.

Relative competitiveness: worst-case performance of an online policy relative to *another online policy*.

Let $m_P(q, s)$ be the number of misses incurred by policy P, starting in cache-set state q, processing memory access sequence s.

Definition 1 (*Relative Miss-Competitiveness*)**.** *Policy P is k-miss-competitive relative to policy Q with additive constant c, if*

$$m_P(p, s) \leq k \cdot m_Q(q, s) + c$$

for all access sequences $s \in S$ and comp. cache-set states $p \in C^P, q \in C^Q$.

> "Policy P will incur at most k times the number of misses of policy Q plus constant c on any access sequence."

Hit-competitiveness is defined analogously.

The *competitive miss ratio* $c^m_{P,Q}$ is the smallest k such that P is k-miss-competitive relative to Q with some additive constant.

Computing Competitive Ratios

P and Q induce a transition system that captures how the two policies act relative to each other, processing the same memory accesses:

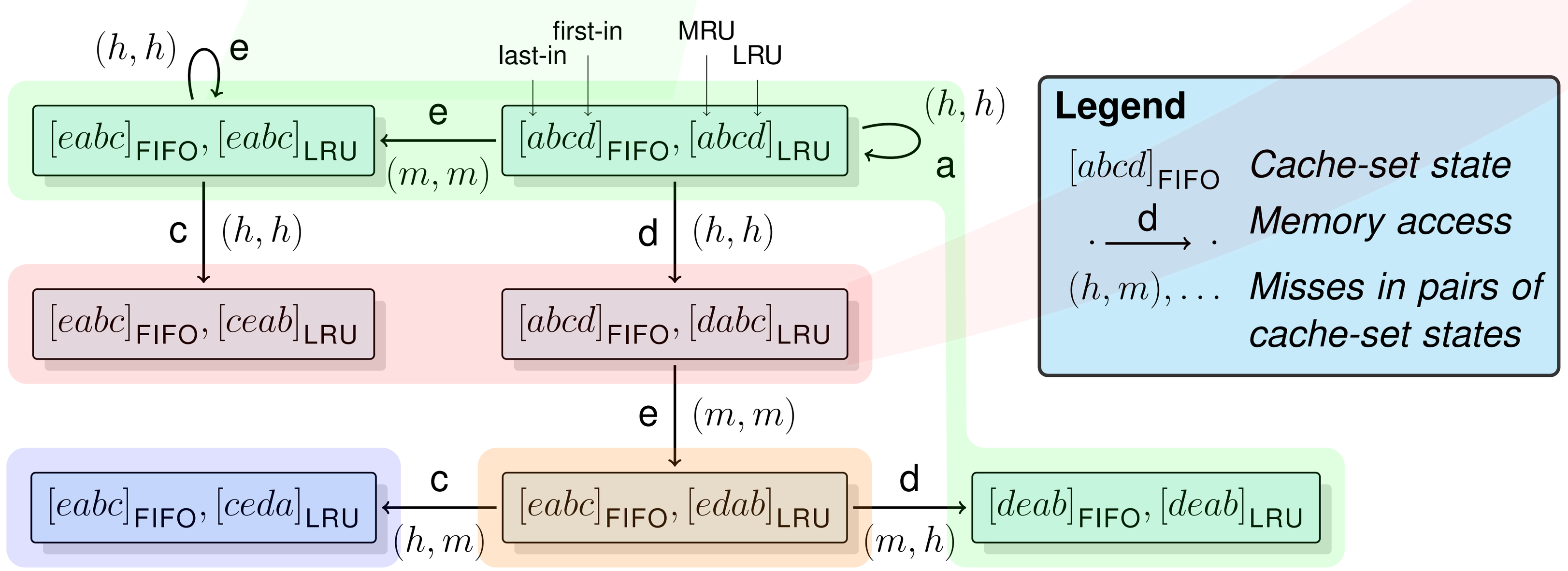
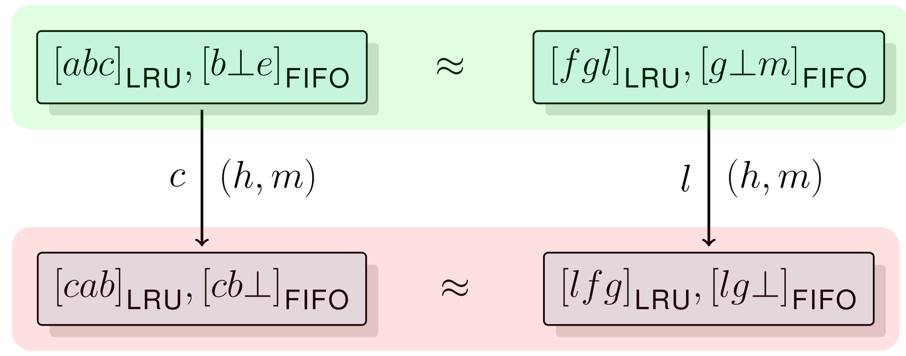


Figure 1: *Small part of the transition system in the computation of competitiveness results for* FIFO(4) *vs.* LRU(4)*.*

Competitive ratio = maximum ratio of misses in policy P relative to the number of misses in policy Q in transition system.

Quotient Transition System

Problem: *The induced transition system is ∞ large.*
Goal: Construct *finite transition system* with same properties.
Observation: Only the *relative positions* of elements matter:



Two pairs of cache-set states are ≈-equivalent, if they can be transformed into each other by a renaming of their contents.

Merging equivalent pairs of cache-set states yields a finite *quotient transition system* that *retains competitiveness properties*:

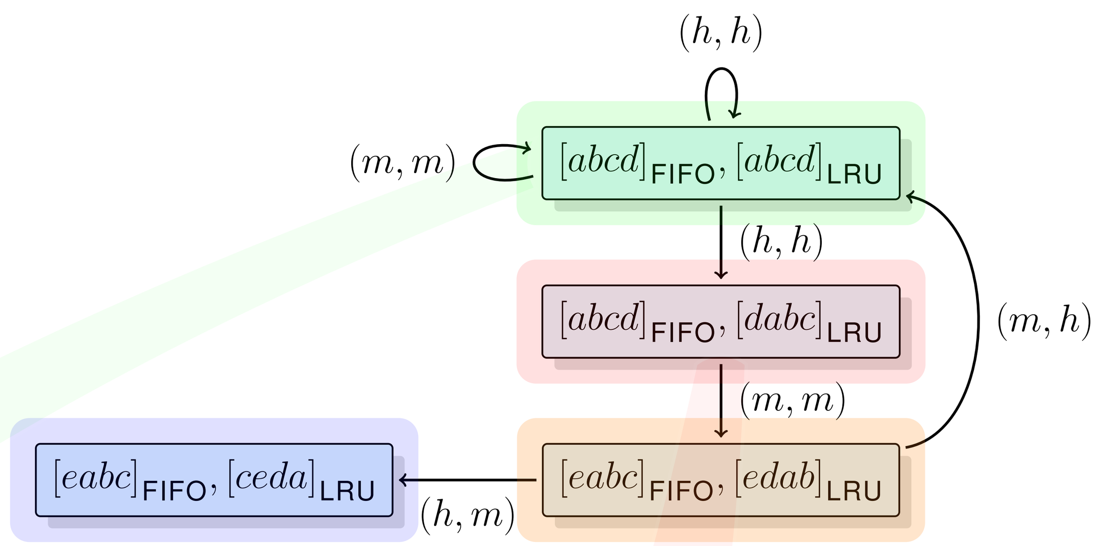


Figure 2: *≈-equivalent states are similarly colored in Figure 1. Merging equivalent states yields the quotient structure depicted here.*

Results

Miss-competitiveness (ratios, constants) (k, c) relating FIFO, PLRU, and LRU:

Associativity:	2	3	4	5	6	7	8
LRU vs FIFO	$2, 1$	$3, 2$	$4, 3$	$5, 4$	$6, 5$	$7, 6$	$8, 7$
FIFO vs LRU	$2, 1$	$3, 2$	$4, 3$	$5, 4$	$6, 5$	$7, 6$	$8, 7$
LRU vs PLRU	$1, 0$	–	$2, 1$	–	–	–	$5, 4$
PLRU vs LRU	$1, 0$	–	∞	–	–	–	∞
FIFO vs PLRU	$2, 1$	–	$4, 4$	–	–	–	$8, 8$
PLRU vs FIFO	$2, 1$	–	∞	–	–	–	∞

Example: LRU(4) is 2-miss-competitive relative to PLRU(4) with constant 1.
PLRU(4) is not miss-competitive relative to LRU(4) at all.

Hit-competitiveness (ratios, constants) (k, c) relating FIFO, PLRU, and LRU:

Associativity:	2	3	4	5	6	7	8
LRU vs FIFO	$0, 0$	$0, 0$	$0, 0$	$0, 0$	$0, 0$	$0, 0$	$0, 0$
FIFO vs LRU	$\frac{1}{2}, \frac{1}{2}$	$\frac{1}{2}, 1$	$\frac{1}{2}, \frac{3}{2}$	$\frac{1}{2}, 2$	$\frac{1}{2}, \frac{5}{2}$	$\frac{1}{2}, 3$	$\frac{1}{2}, \frac{7}{2}$
LRU vs PLRU	$1, 0$	–	$\frac{1}{2}, 1$	–	–	–	$\frac{1}{8}, \frac{15}{8}$
PLRU vs LRU	$1, 0$	–	$\frac{1}{2}, 1$	–	–	–	$\frac{1}{4}, \frac{3}{2}$
FIFO vs PLRU	$\frac{1}{2}, \frac{1}{2}$	–	$\frac{1}{4}, \frac{5}{4}$	–	–	–	$\frac{1}{11}, \frac{19}{11}$
PLRU vs FIFO	$0, 0$	–	$0, 0$	–	–	–	$0, 0$

Generalizations to arbitrary *k*

Previously unknown relations:

- PLRU(k) is 1-competitive relative to LRU($1 + log_2 k$).
- FIFO(k) is $\frac{1}{2}$-hit-competitive relative to LRU(k), whereas
- LRU(k) is not l-hit-competitive relative to FIFO(k) for any l, but
- LRU($2k-1$) is 1-competitive relative to FIFO(k) with constant 0 for all k
  $\longrightarrow$ yields *first may-analysis for* FIFO,
- PLRU(k) is not l-miss-competitive relative to LRU(k) for $k \geq 4$ and any l.
  $\Longrightarrow$ PLRU(k) is not l-miss-competitive (in the classical sense) for $k \geq 4$.

Reference: Jan Reineke and Daniel Grund: *Relative Competitive Analysis of Cache Replacement Policies*. In ACM SIGPLAN/SIGBED 2008 Conference on Languages, Compilers, and Tools for Embedded Systems – LCTES 2008

Contact information:
http://rw4.cs.uni-sb.de/
{reineke|grund}@cs.uni-sb.de

Supported by the
Deutsche Forschungsgemeinschaft