Static Probabilistic Timing Analysis of Random Replacement Caches using Lossy Compression

David Griffin, Benjamin Lesage
Alan Burns, Rob Davis

RTS York

THE UNIVERSITY *of York*

Introduction

- Static analysis gives absolute guarantees
- … but it's massively pessimistic
- … and most people don't need absolute guarantees

Static Probabilistic Timing Analysis (SPTA)

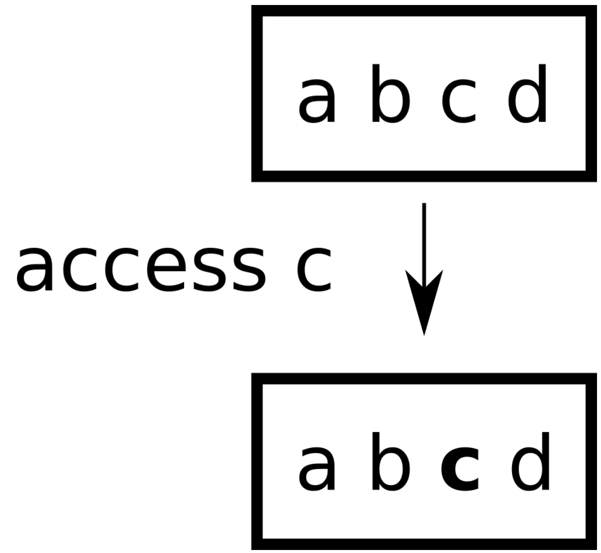
- Determine probability that a system would fail
- Find a probability of failure that is sufficiently low
- (Hopefully less pessimistic)

Randomised Hardware

- Can be analysed by SPTA
- Idea: By making a lot of truly random choices, expected behaviour is predictable
- This paper examines the Random Replacement cache

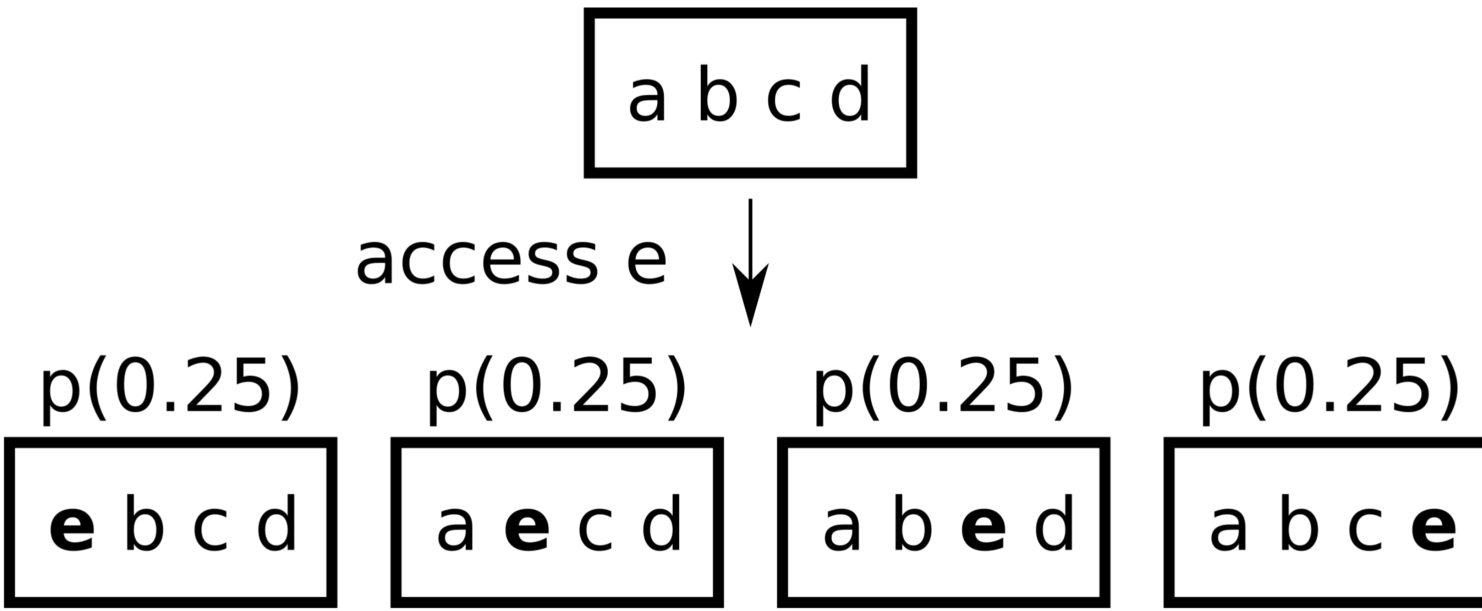
Random Replacement Cache

- If memory access is a hit, do nothing



Random Replacement Cache

- If memory access is a miss, evict a random element



Current Analysis

- Original methods assumed independence of hit probabilities
  - They're not
- Corrected version by Davis et al. [5]
  - But this simple analysis is outperformed by LRU cache in all cases [Reineke, 2014]

State of the Art Techniques

- Focus Blocks approach proposed by Altmeyer and Davis [6]
- A number of memory blocks are focused and analysed by exhaustive search
- Others are analysed by previous method
- Not dominated by LRU analysis

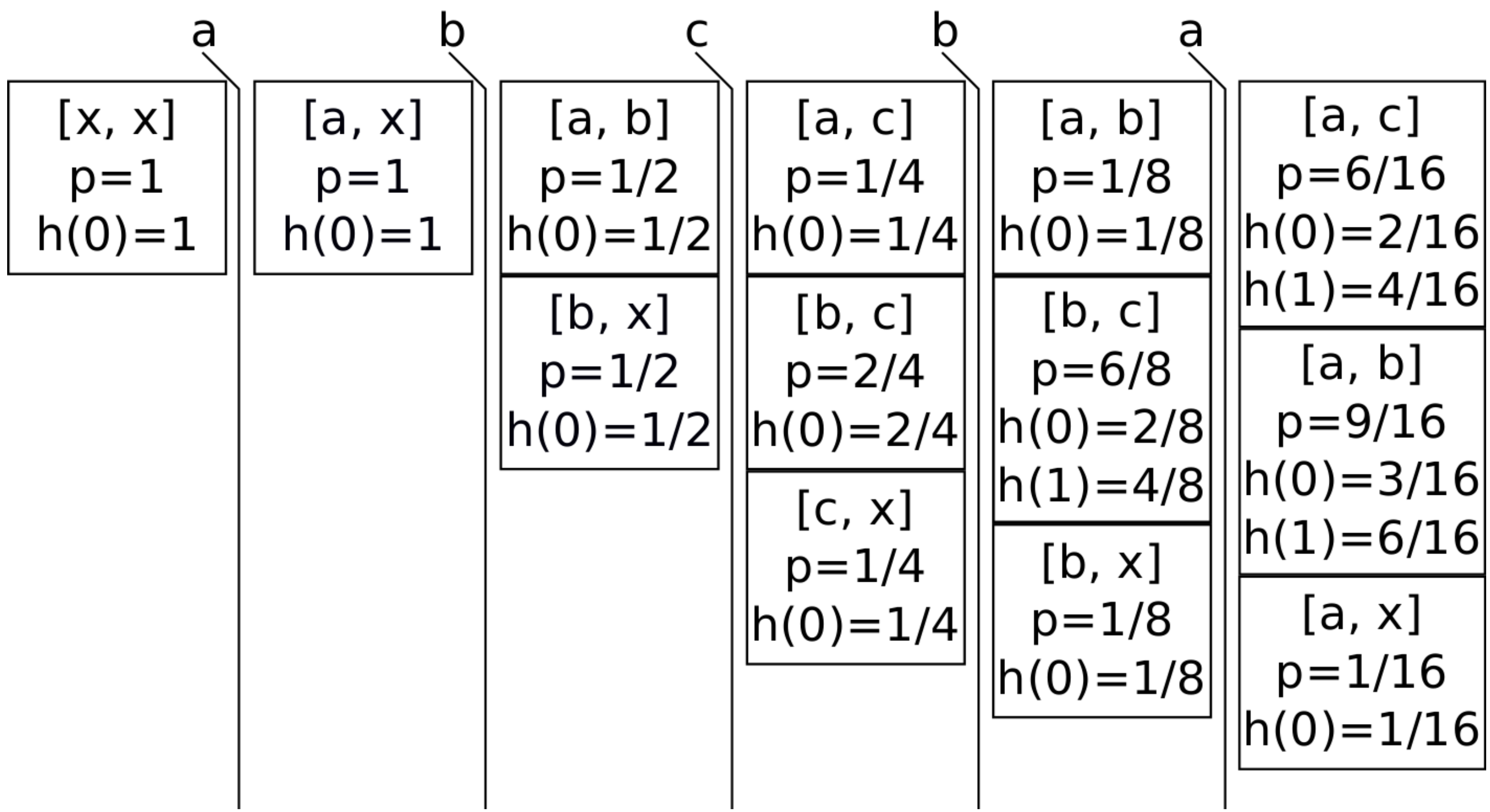
State of the Art Techniques

- Weaknesses of Focus Blocks
  - Computationally expensive
  - Increasing number of focus blocks can decrease accuracy
  - Size of distributions modelled increases with size of input
  - No May analysis

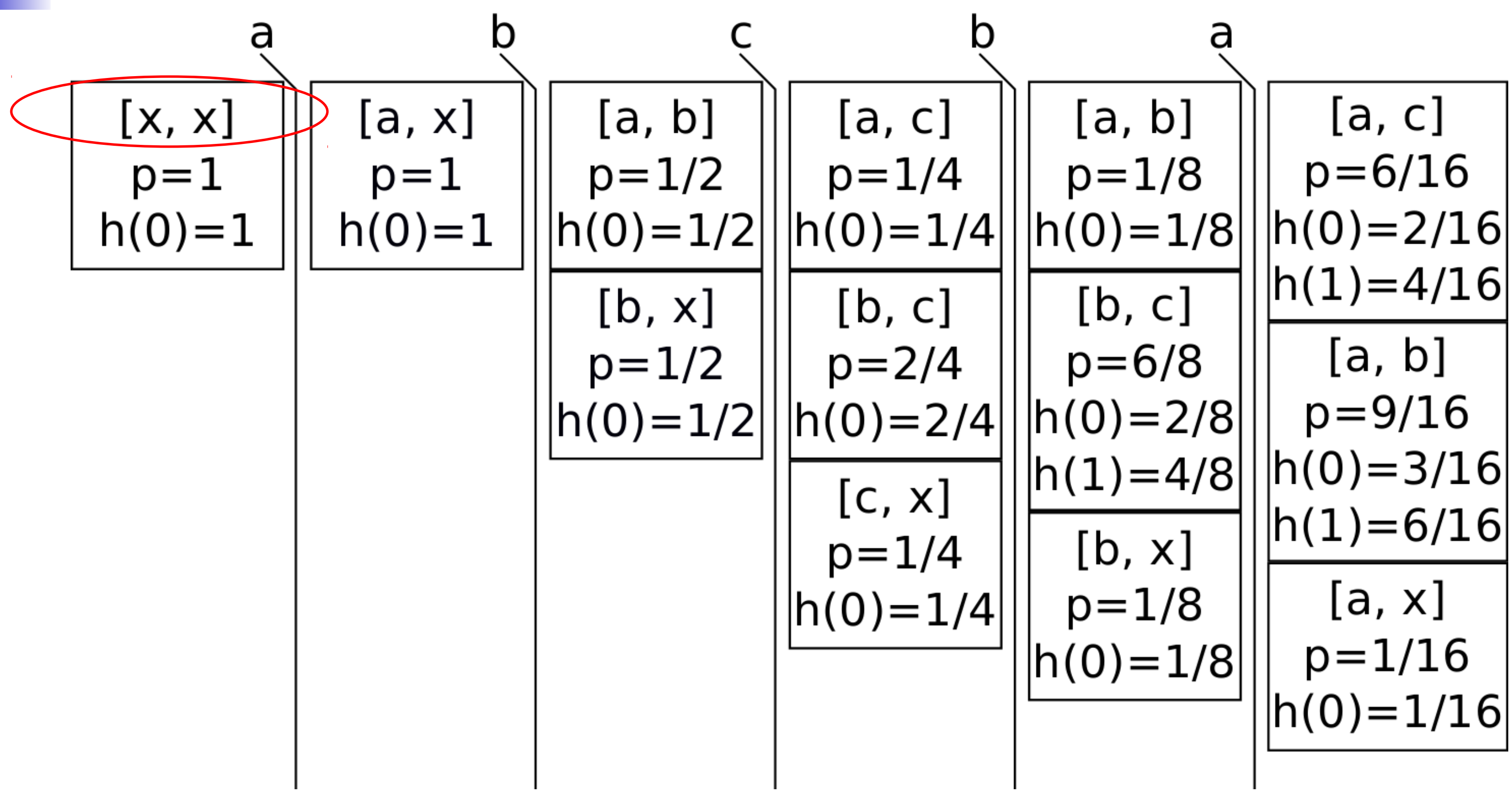
Lossy Compression

- Lossy Compression is used to find what can be removed with little consequence
- In this case, apply lossy compression to states in exhaustive search

Exhaustive Search Example

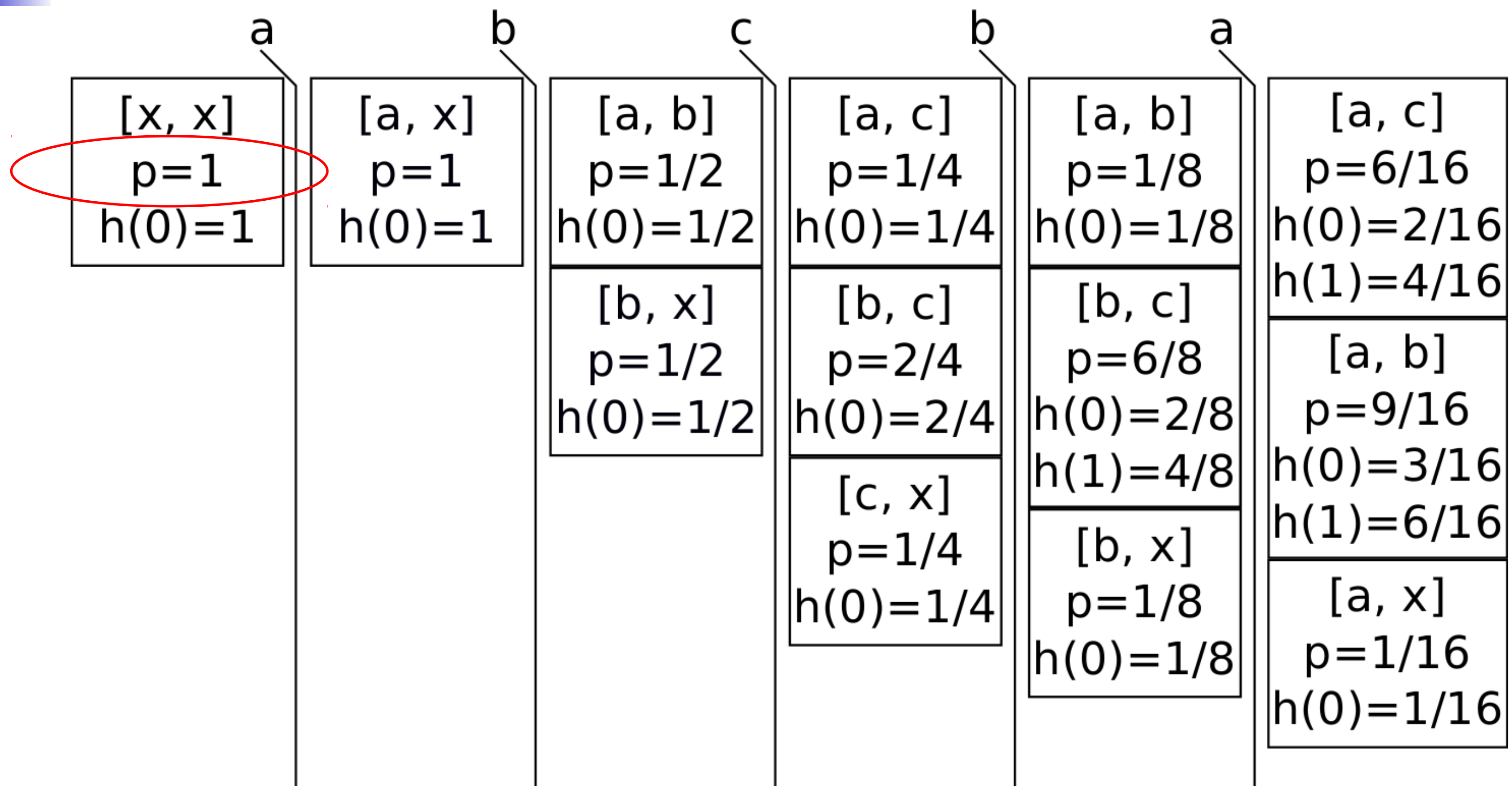


Exhaustive Search Example

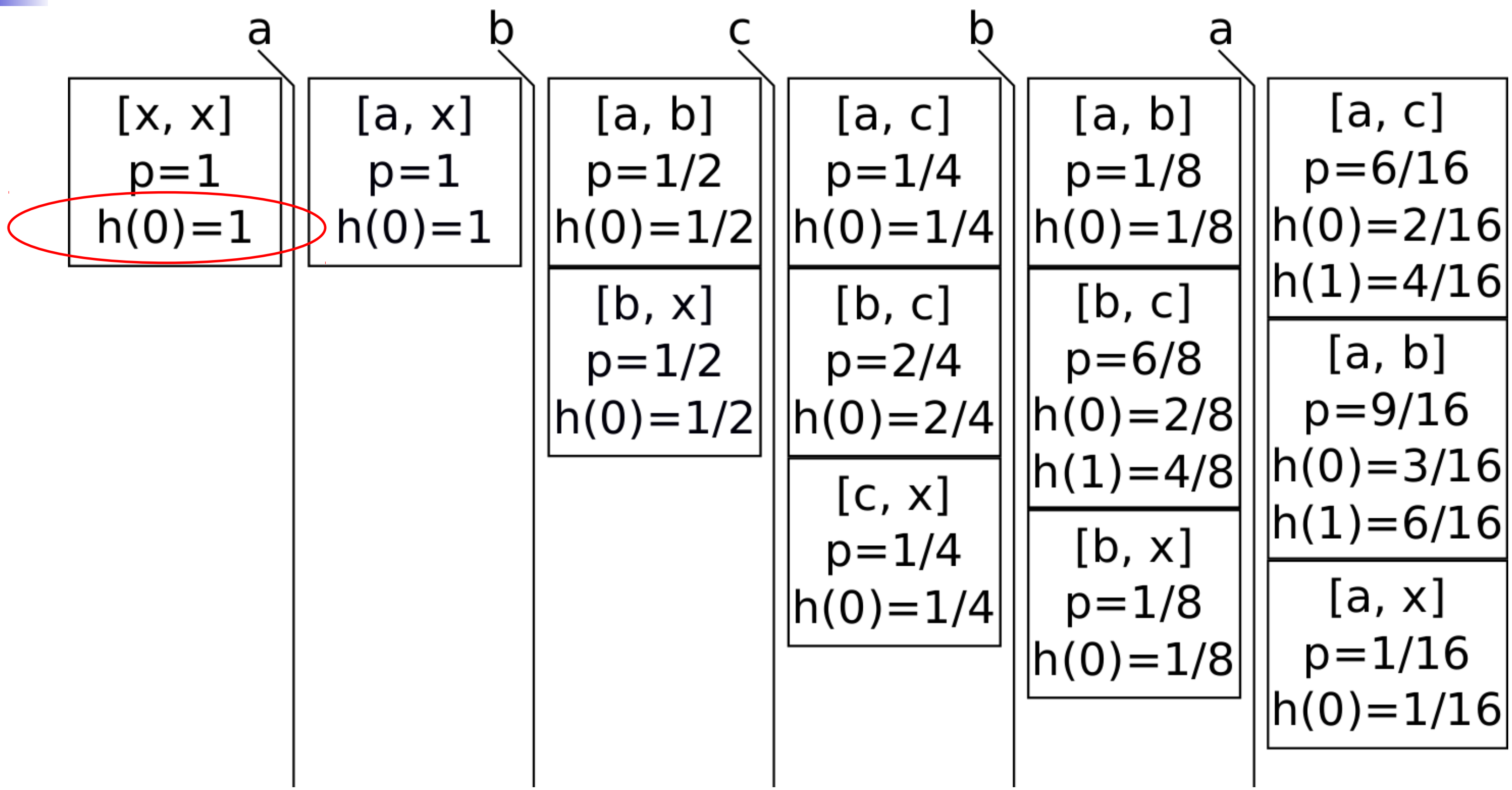


THE UNIVERSITY *of York*

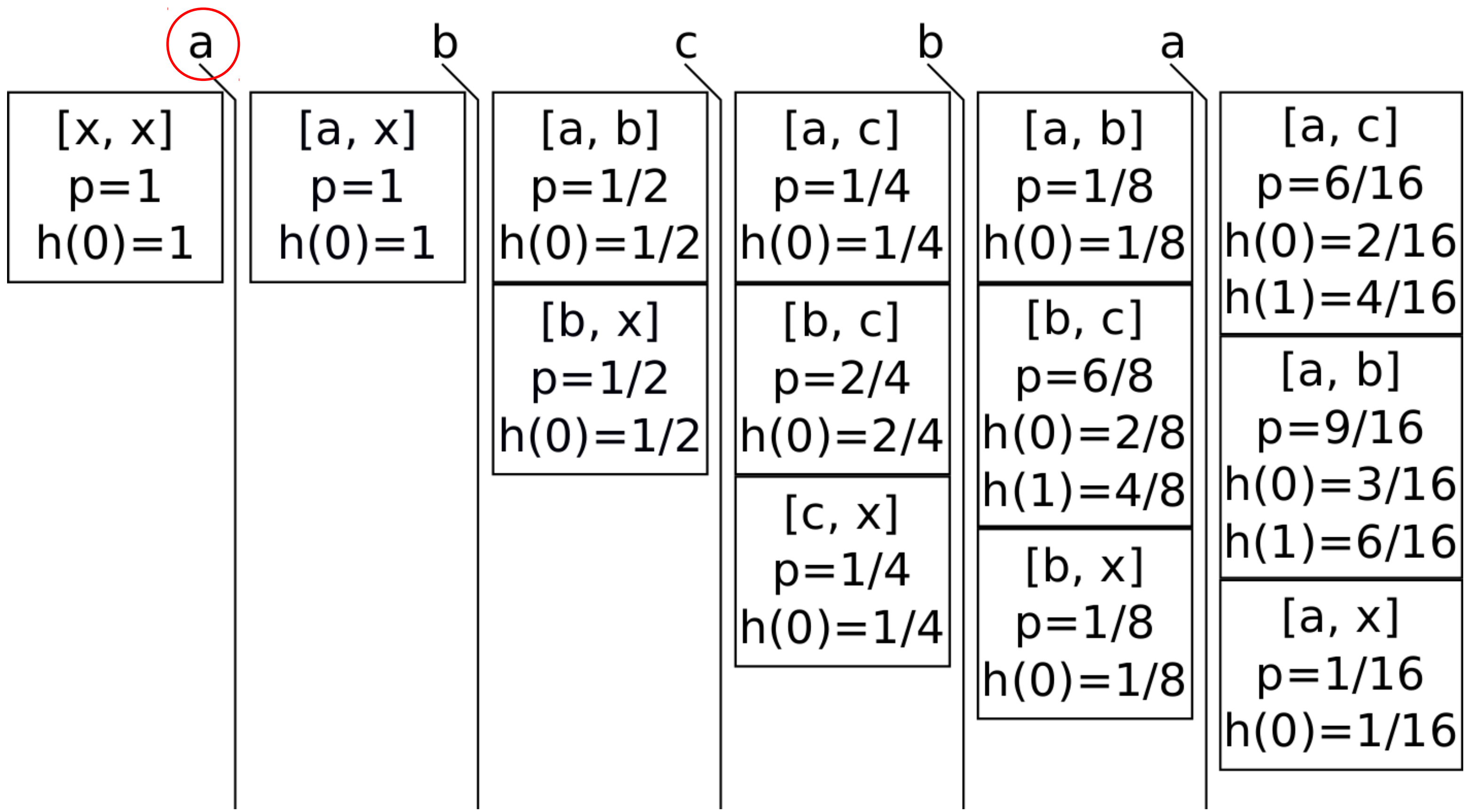
Exhaustive Search Example



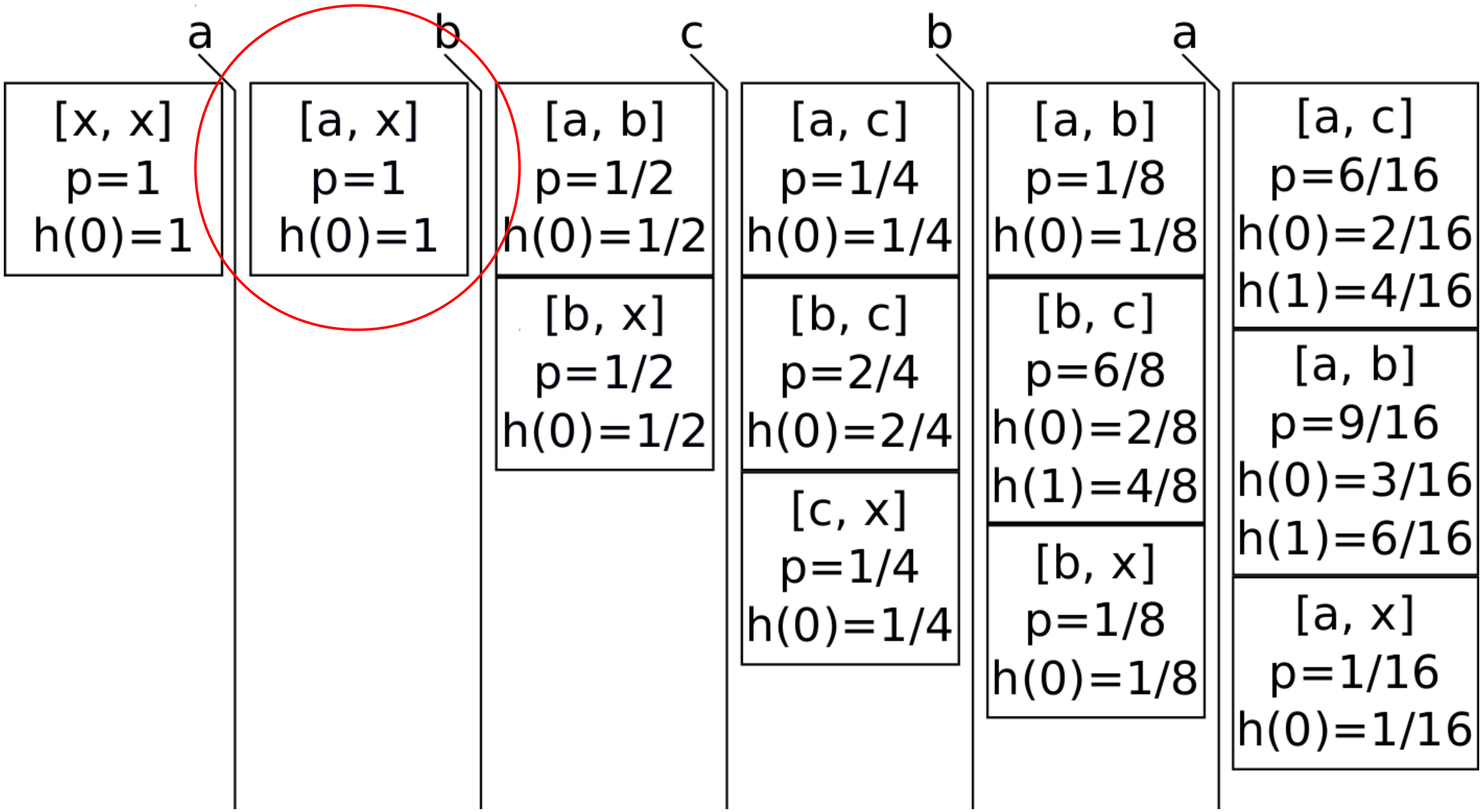
Exhaustive Search Example



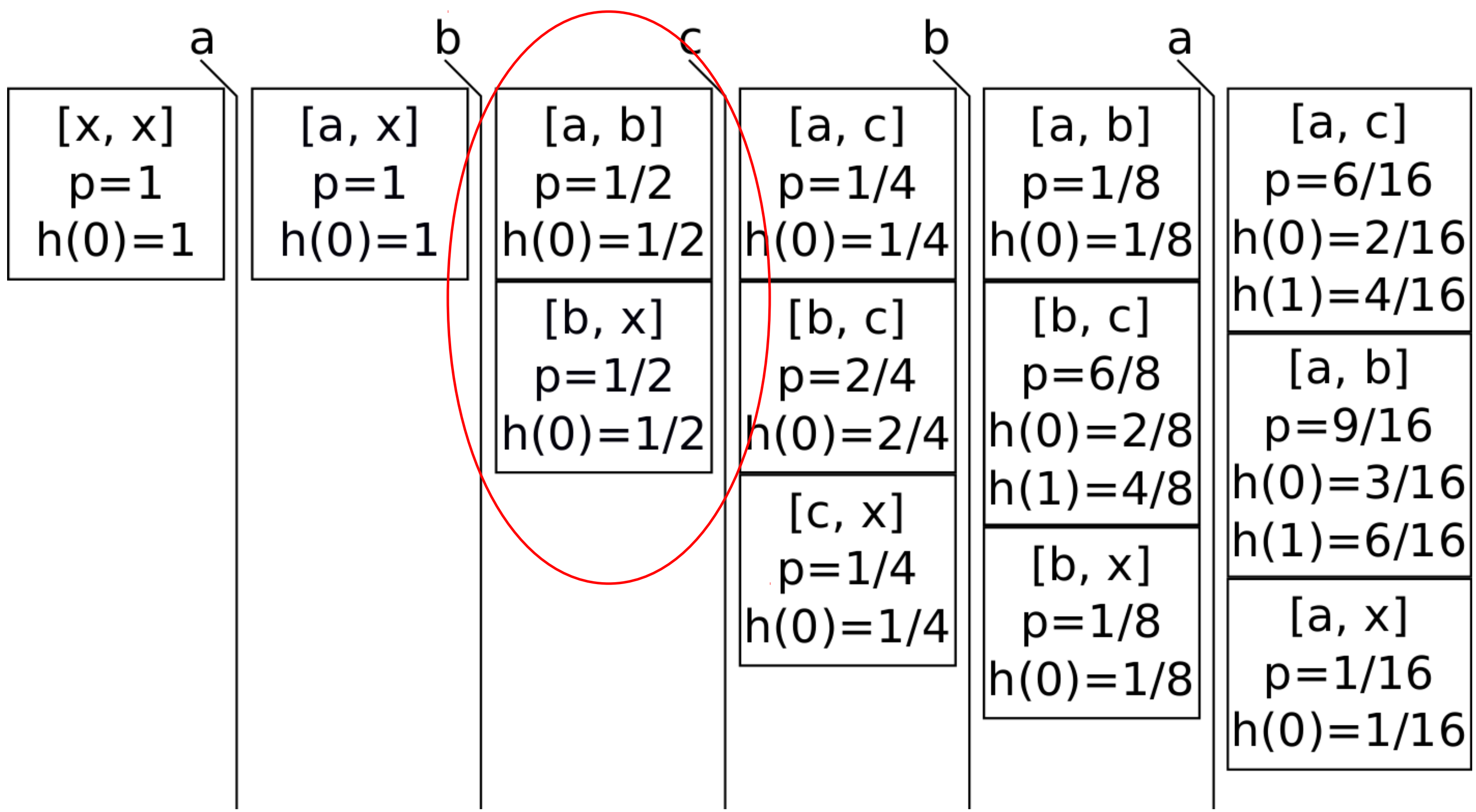
Exhaustive Search Example



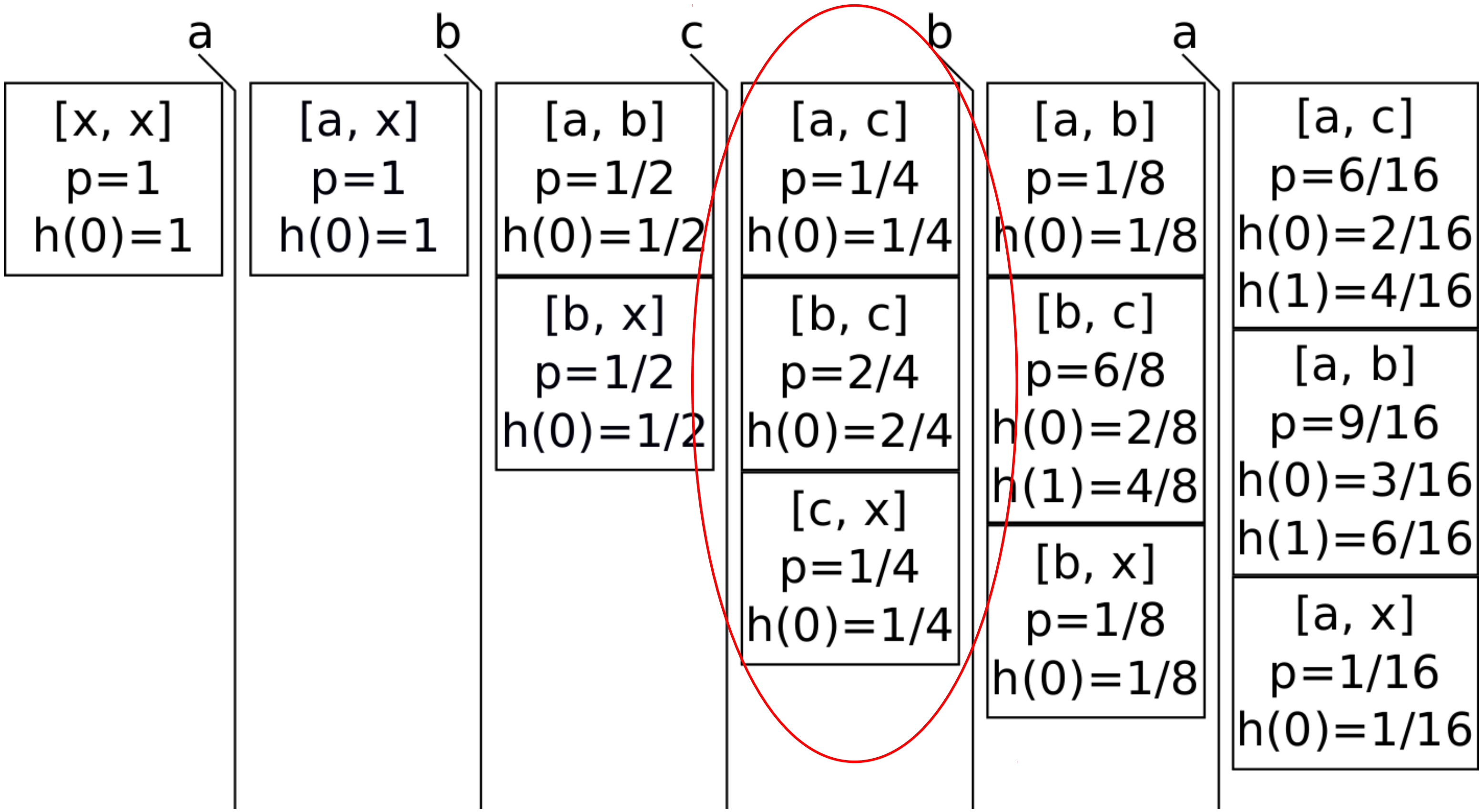
Exhaustive Search Example



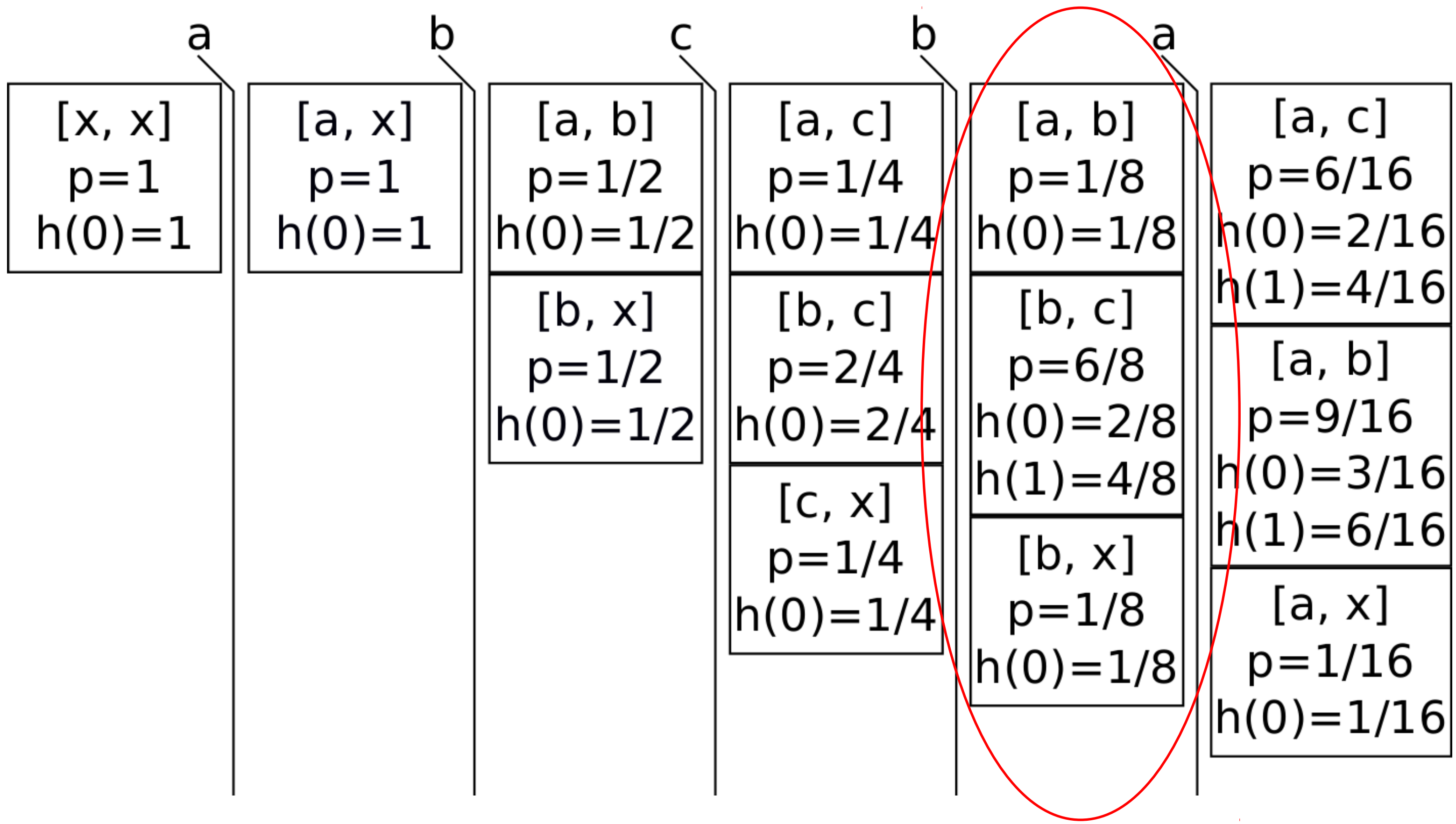
Exhaustive Search Example



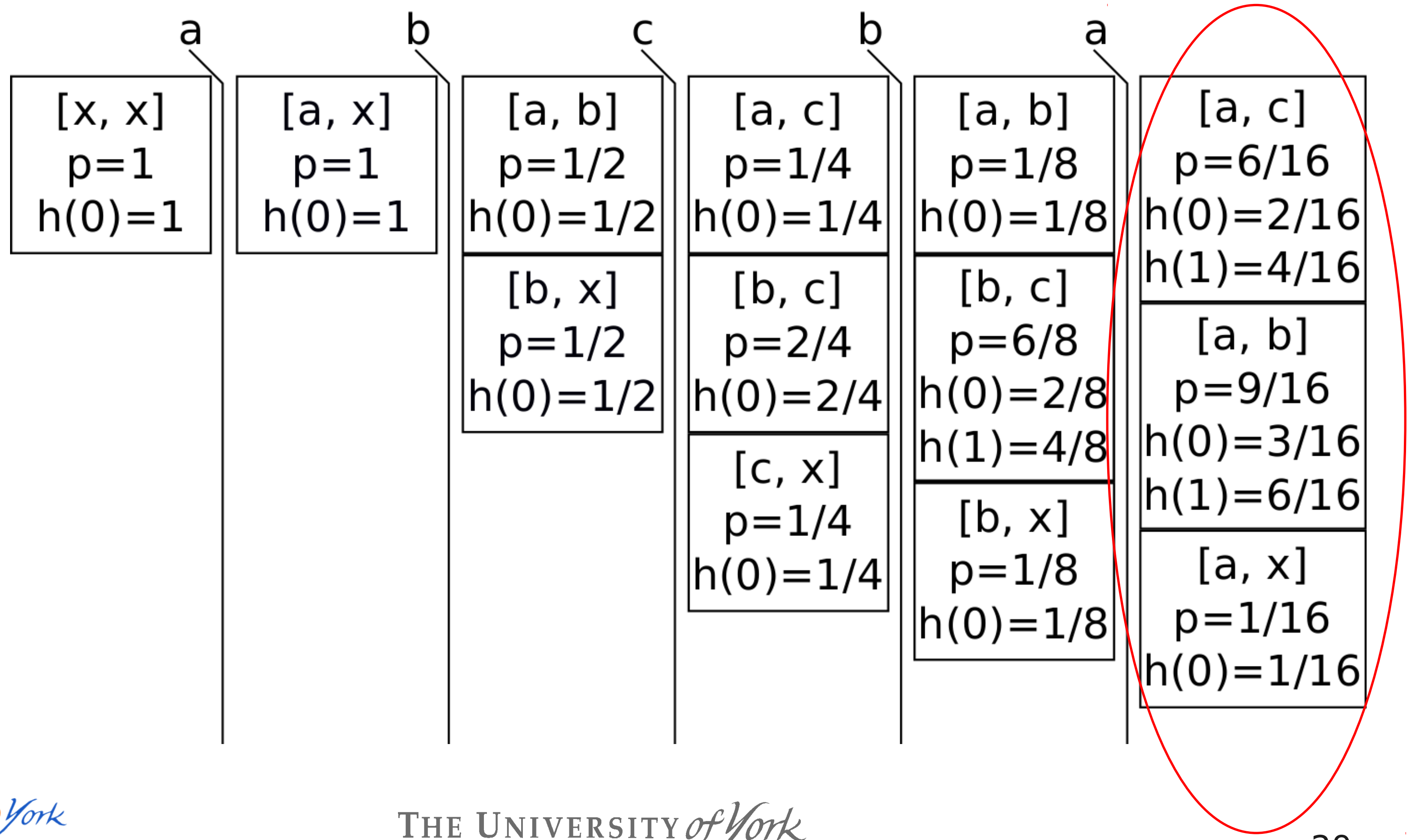
Exhaustive Search Example



Exhaustive Search Example



Exhaustive Search Example



What to compress

- Problem: All information is potentially valuable
- Solution: Decide based on context when information is not likely to be valuable

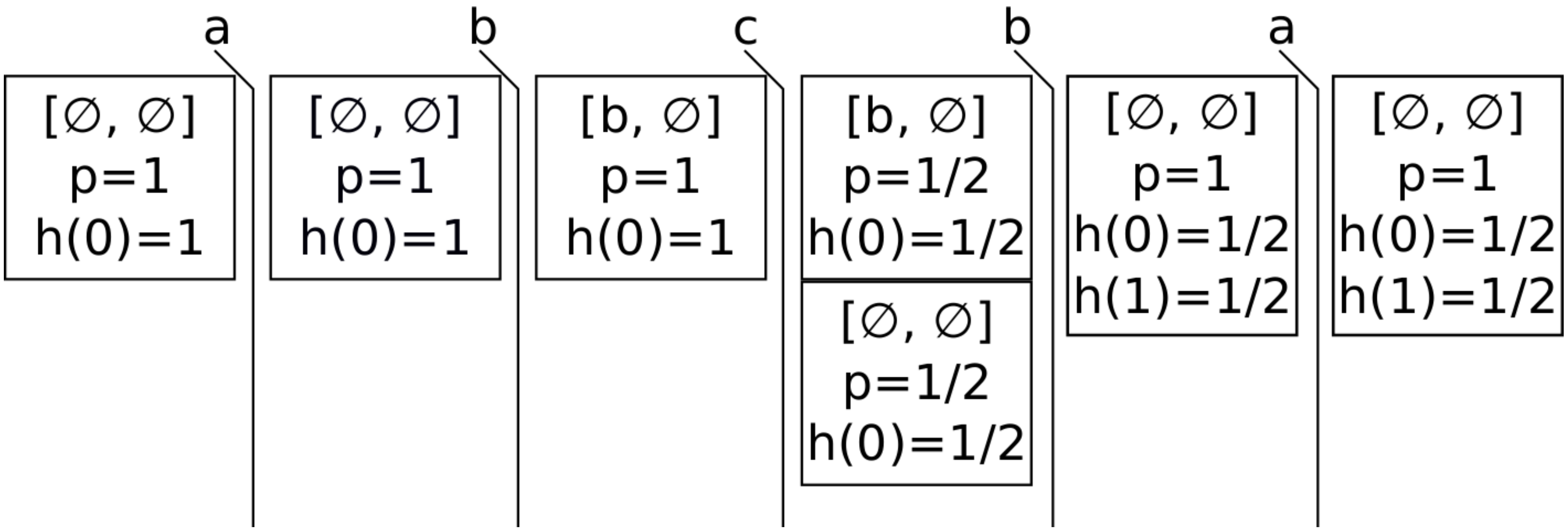
Compressing Memory Blocks

- Want to discard least important memory blocks
- Assume that importance correlates to hit probability
- Define $\emptyset$ to be an unknown memory block

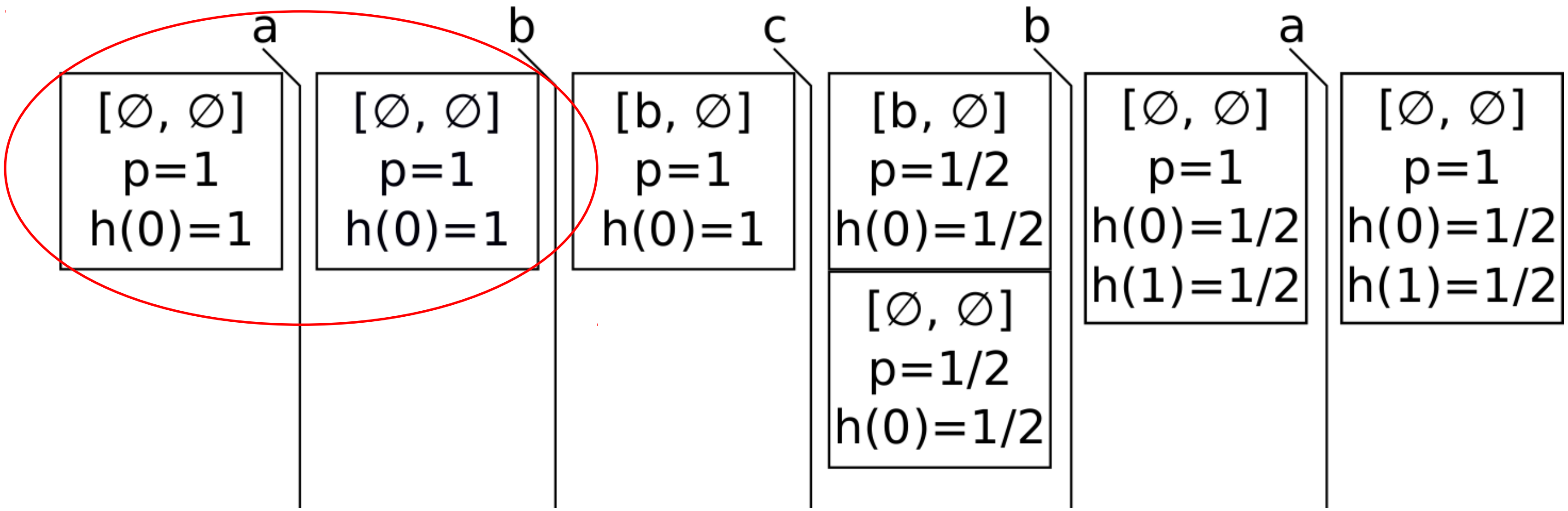
Compressing Memory Blocks

- If a memory block isn't used again in sufficient time, it can be inferred it's hit probability will become low, and therefore it isn't important
  - FRD(x): Replace any memory block whose forward reuse distance is > x with ∅

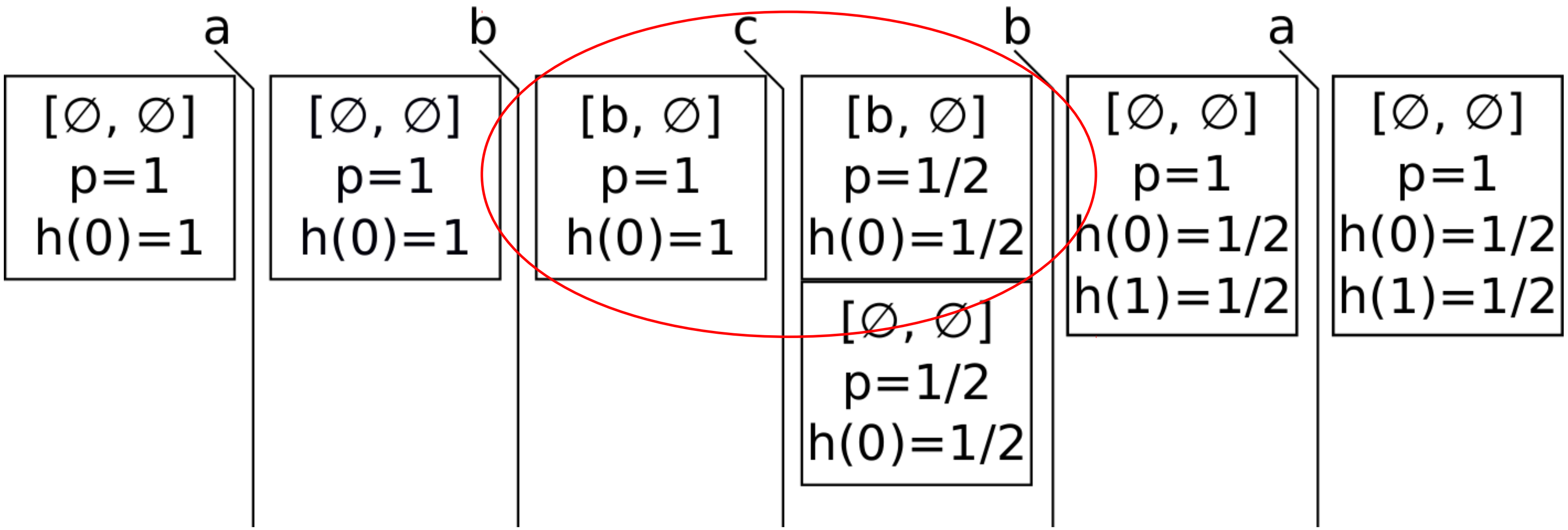
Example - FRD(2)



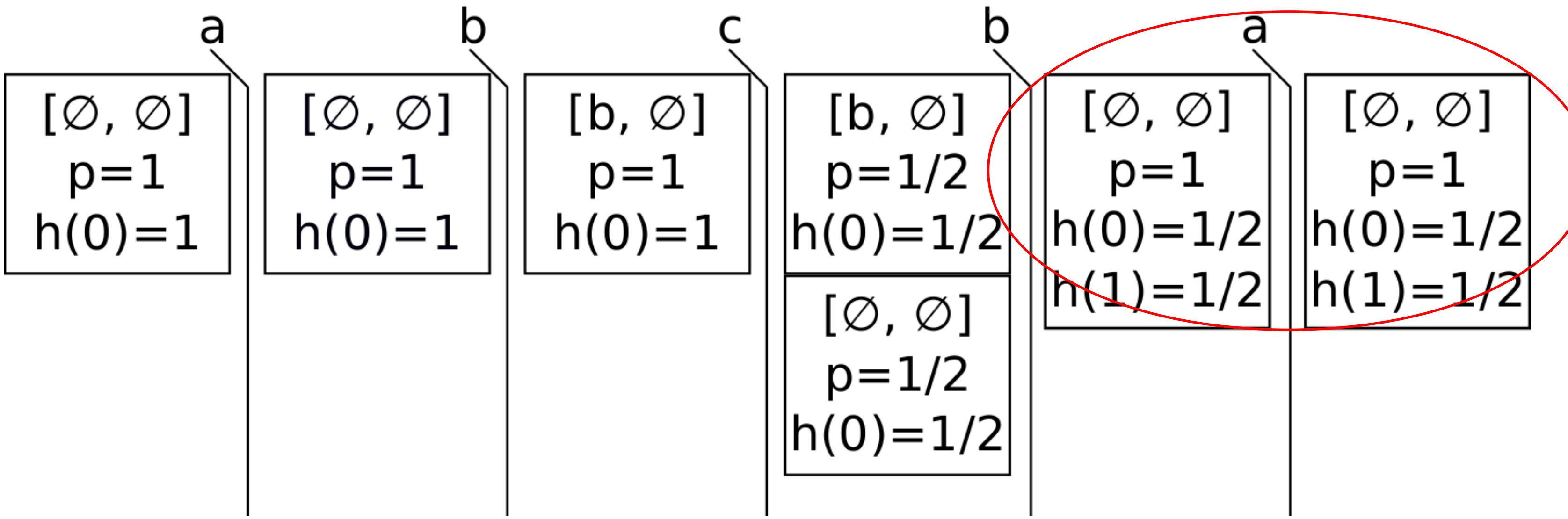
Example - FRD(2)



Example - FRD(2)



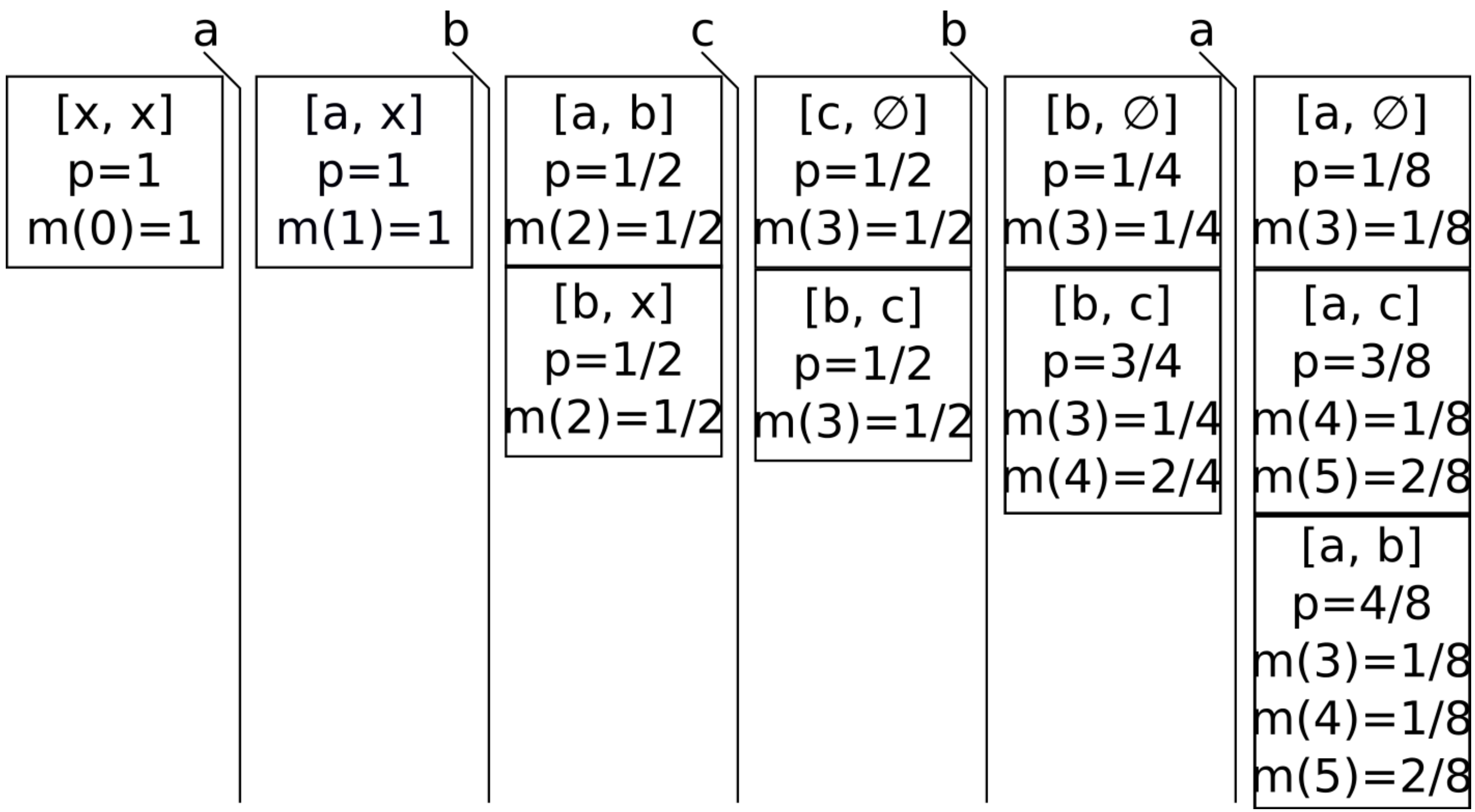
Example - FRD(2)



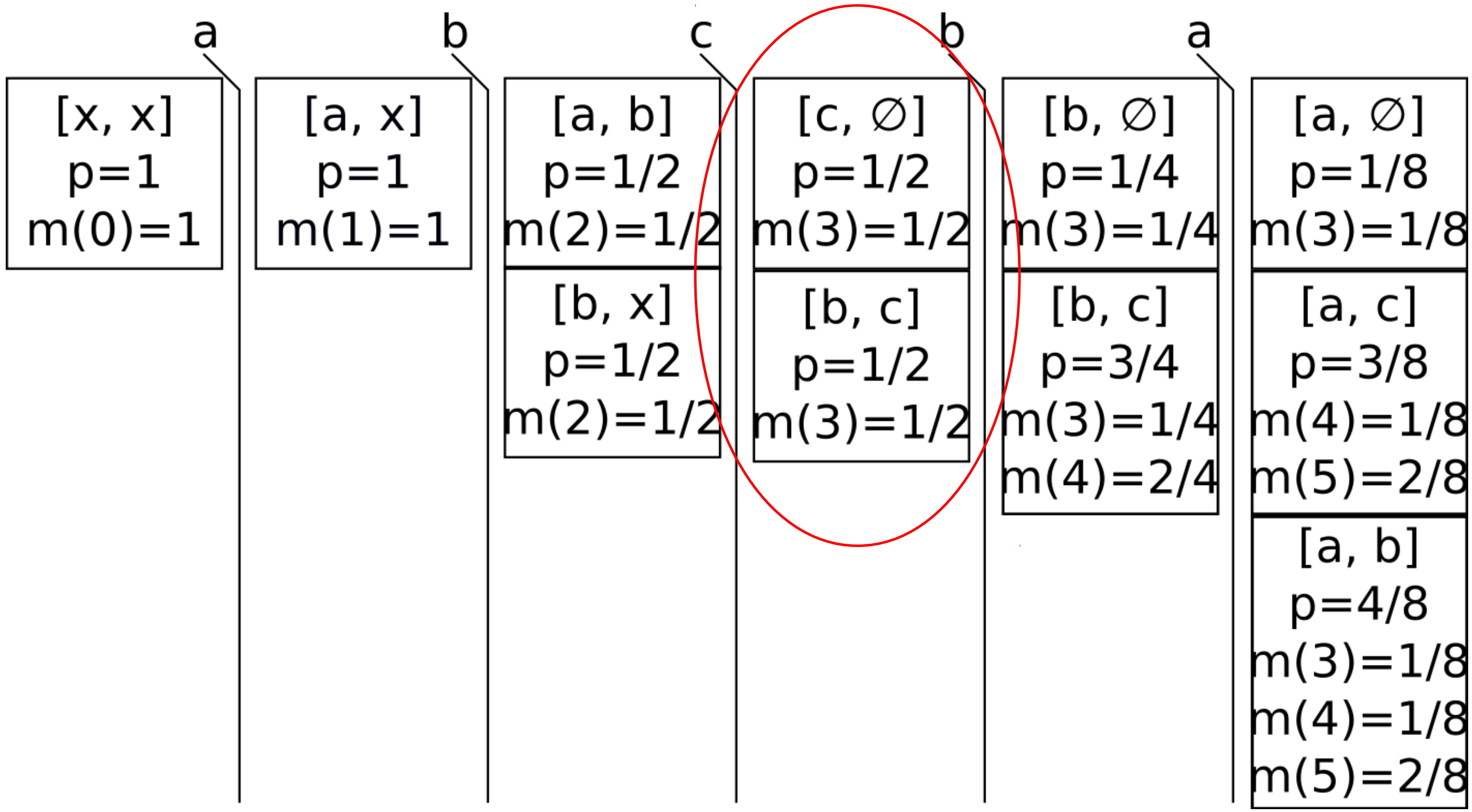
Compressing Memory Blocks

- Or, simply calculate the hit probability of elements in cache
  - PRB(x): Replace any memory block with a hit probability of < x with ∅
  - Not as aggressive as FRD

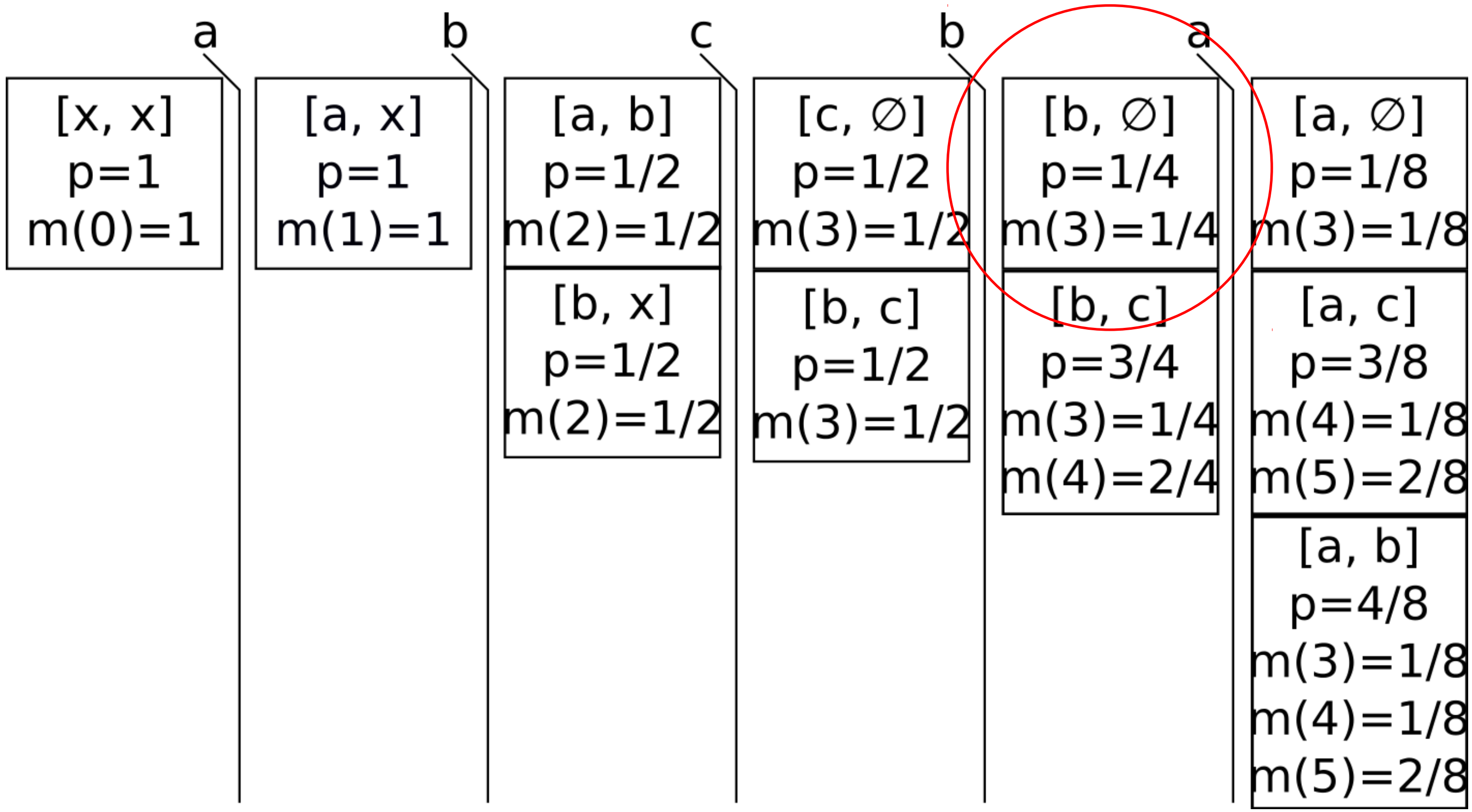
Example – PRB(0.5)



Example – PRB(0.5)



Example – PRB(0.5)



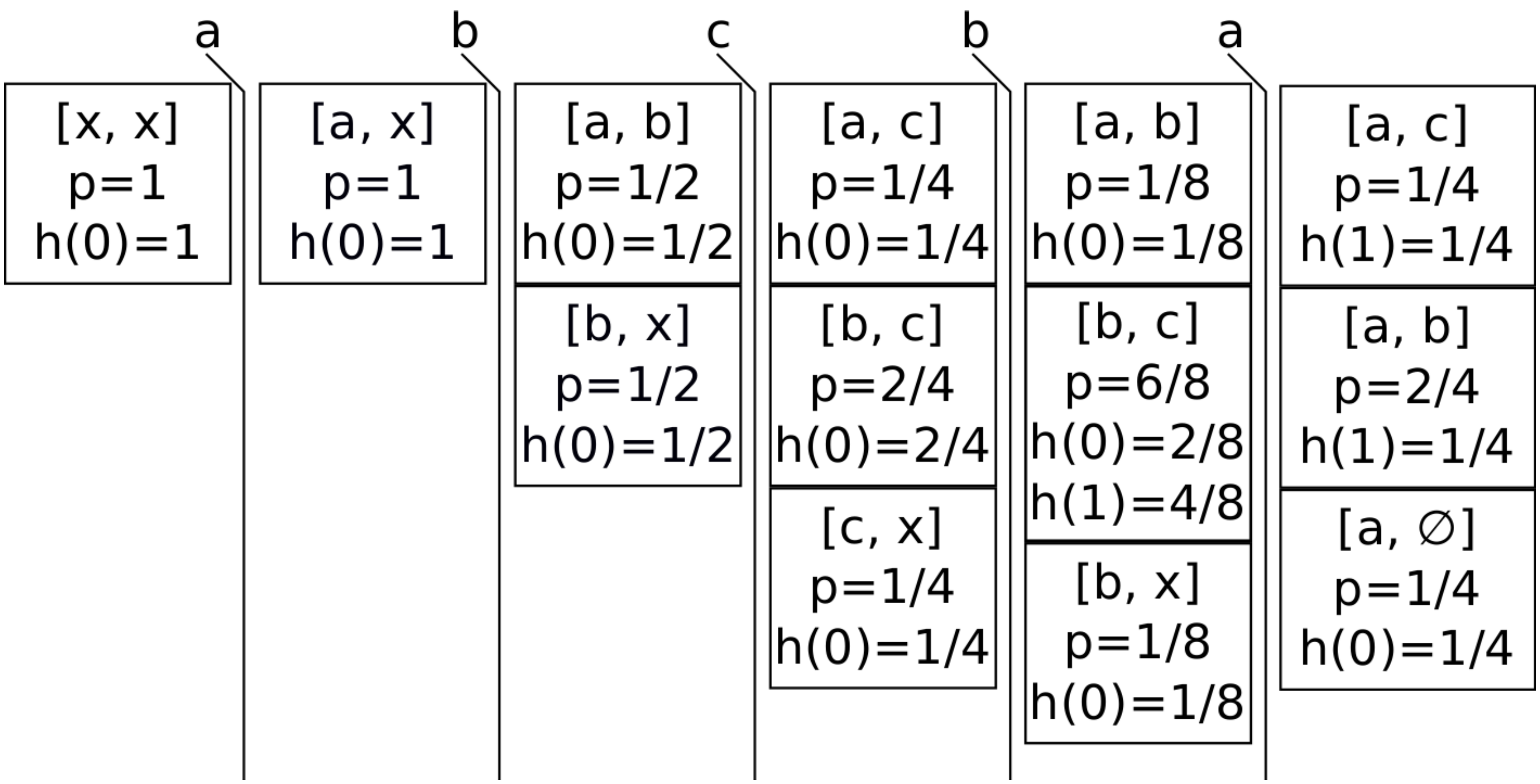
Compressing Distributions

- We are interested in the tail of a distribution
- .. but not interested in the extreme tail of the distribution.
  - Compress any event in distribution that is sufficiently unlikely
    - e.g. probability $< 10^{-9}$

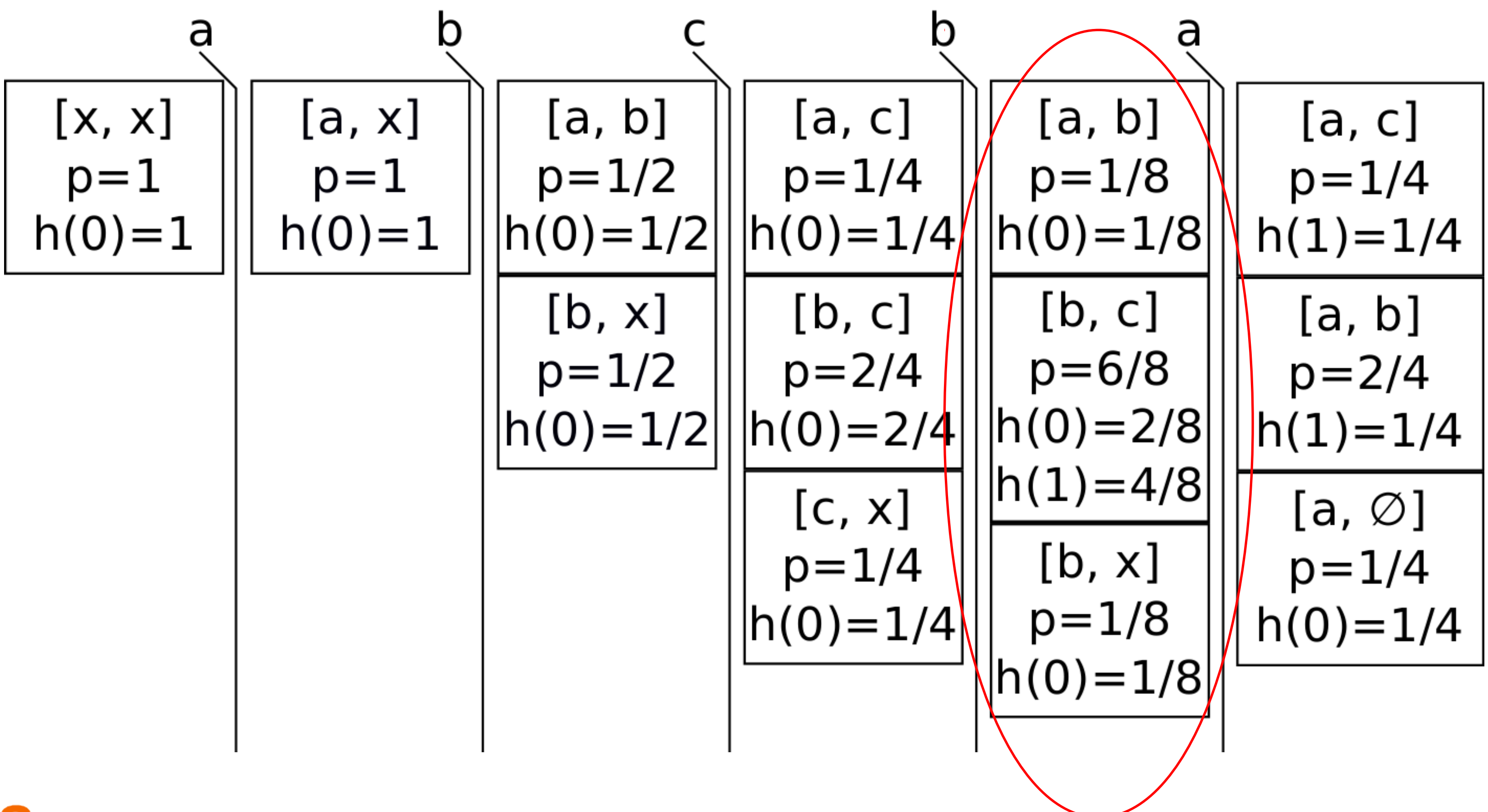
Compressing Distributions

- Use fractions for probabilities $\frac{a}{b}$
- If *b* exceeds a threshold **α**, simplify by a fixed factor *f*
  - $max(\{\frac{x}{\lfloor\frac{b}{f}\rfloor} \Big| \frac{x}{\lfloor\frac{b}{f}\rfloor} < \frac{a}{b}\})$
- Combine all probability 'lost' due to simplification and place into an upper bound state

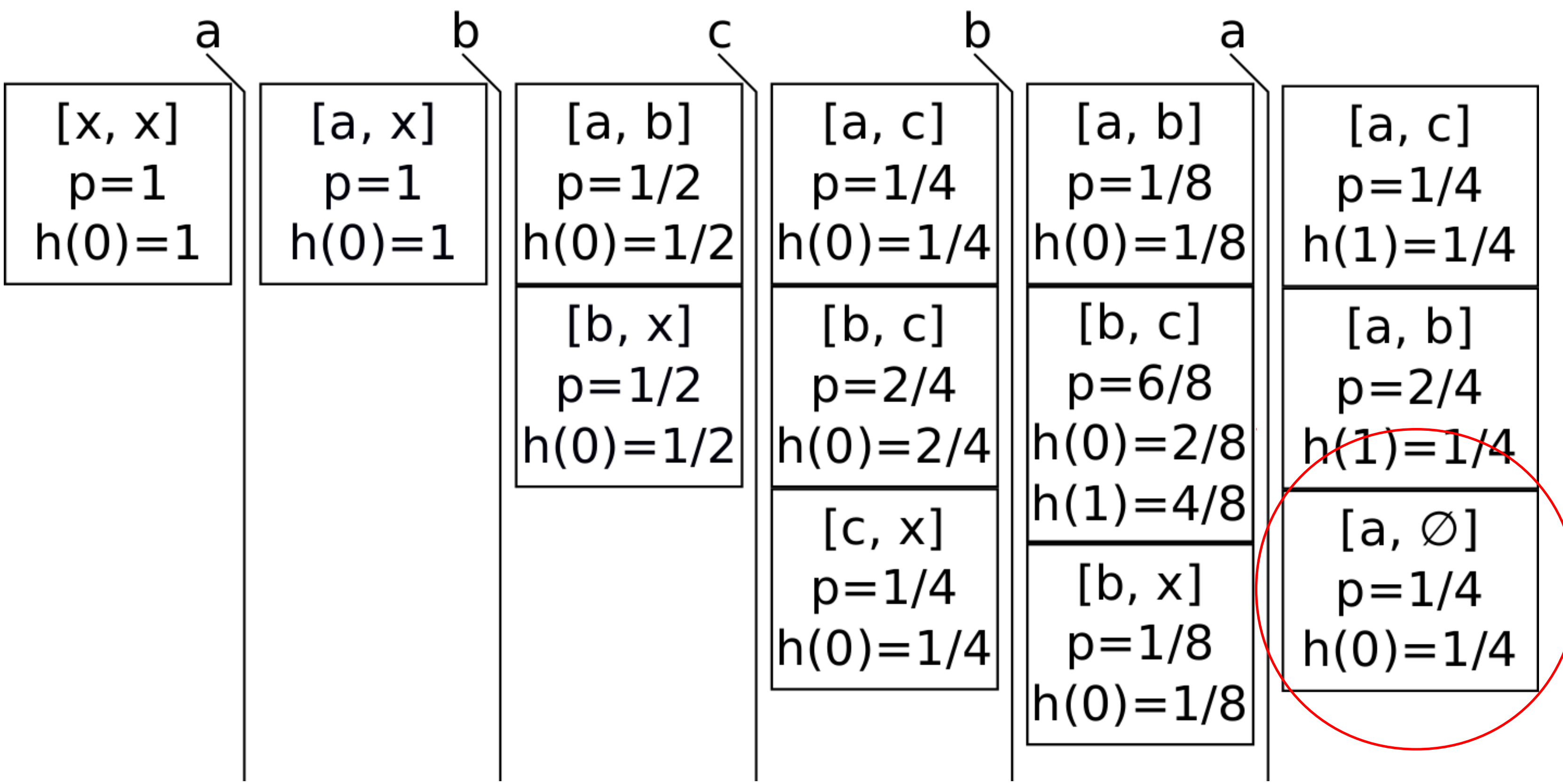
Example $\alpha = 8, f = 2$



Example $\alpha = 8, f = 4$



Example $\alpha = 8, f = 2$



New Technique

- Two proposed techniques
  - FRD – Forward Reuse Distance
  - PRB – Hit Probability
- Combines the FRD/PRB memory block compression strategies with distribution compression

Evaluation

- 16-way Random Replacement Cache with cache line size 8
- Traces from Mälardalen Benchmarks
- Fixed parameters $\alpha = 10^9$, $t = 10^6$
- Variety of parameters for PRB and FRD methods

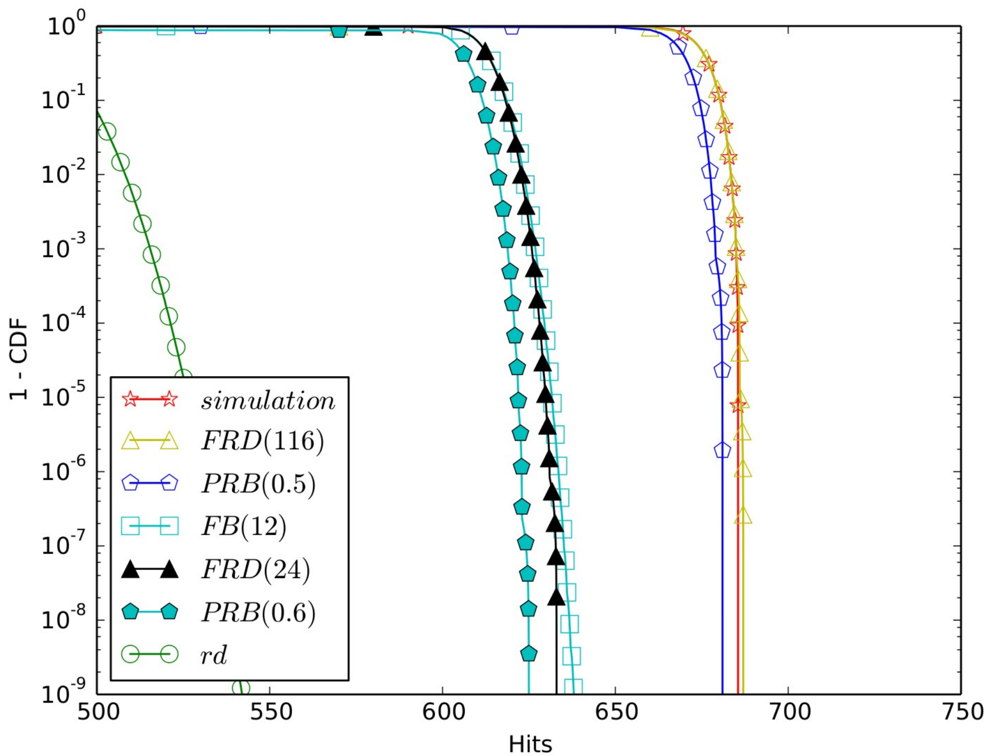
RTS York

THE UNIVERSITY of York

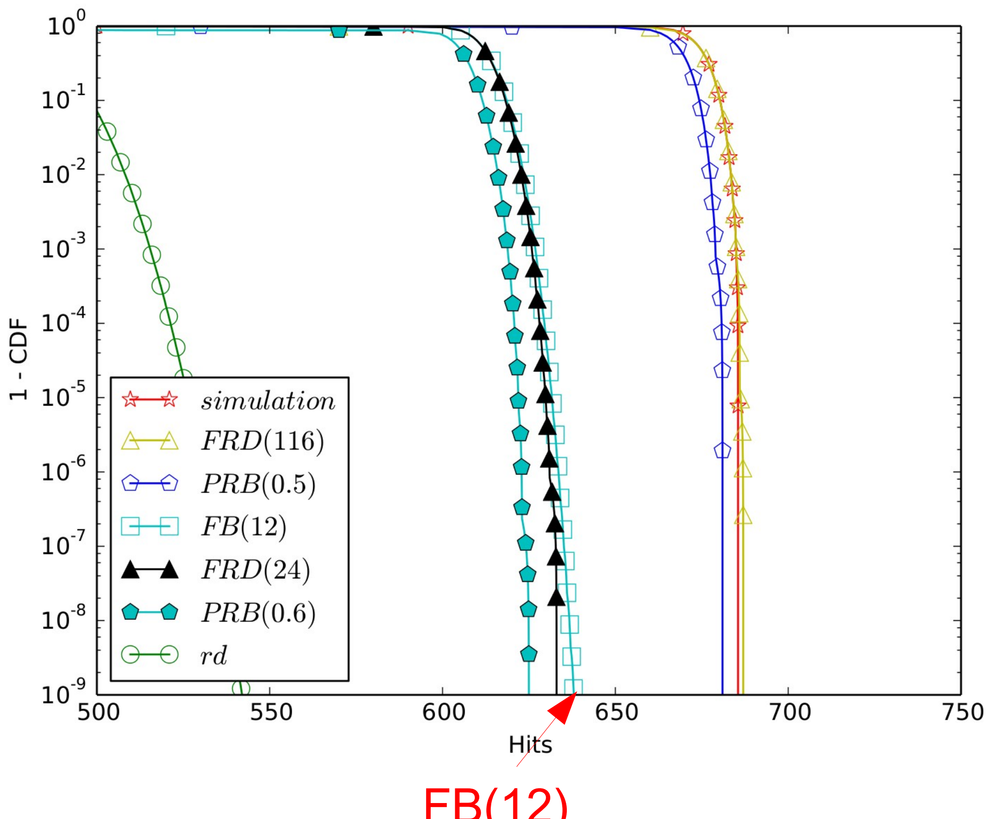
Evaluation

- Compared against
  - 1 Billion Simulator Runs
  - Altmeyer and Davis' Focus Blocks method [6]
- All analyses run on moderately powerful laptop
  - (Does not include simulator runs)

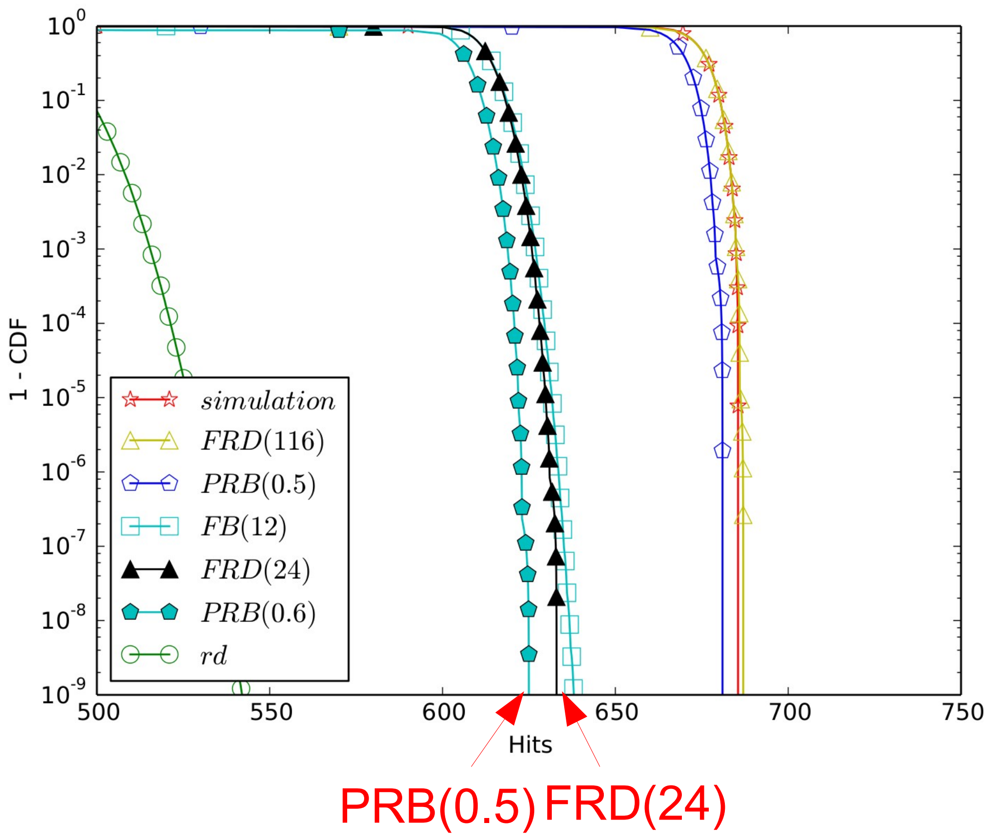
Evaluation – insertsort hits



Evaluation – insertsort hits

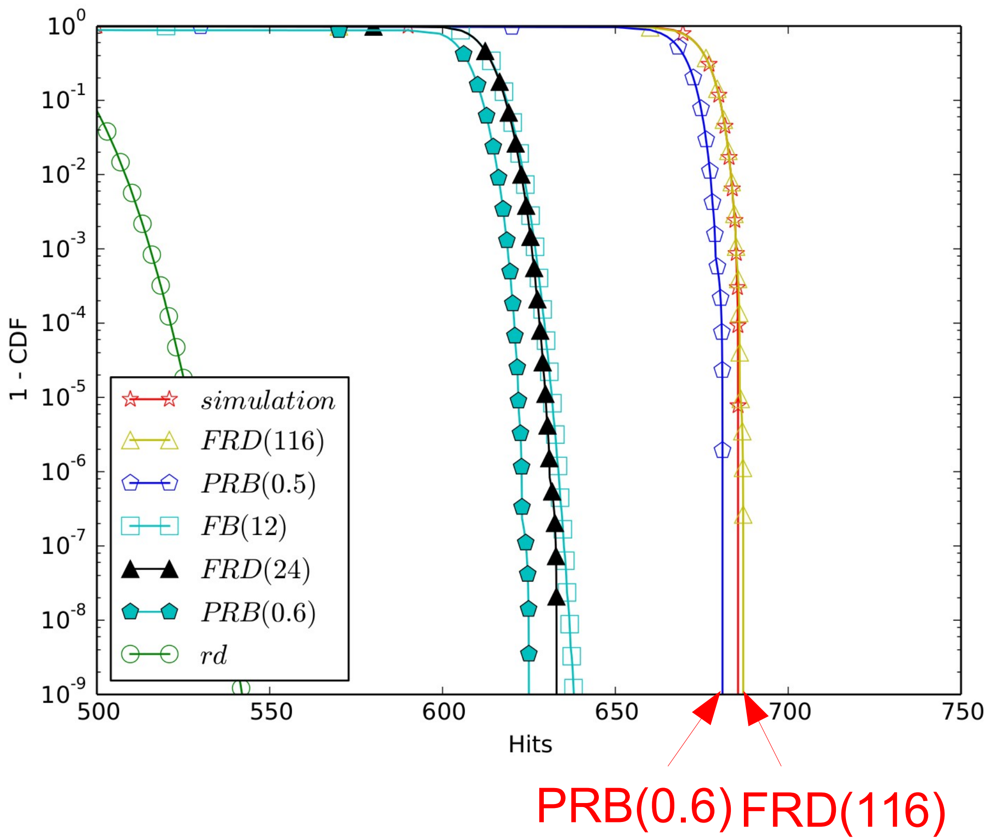


Evaluation – insertsort hits

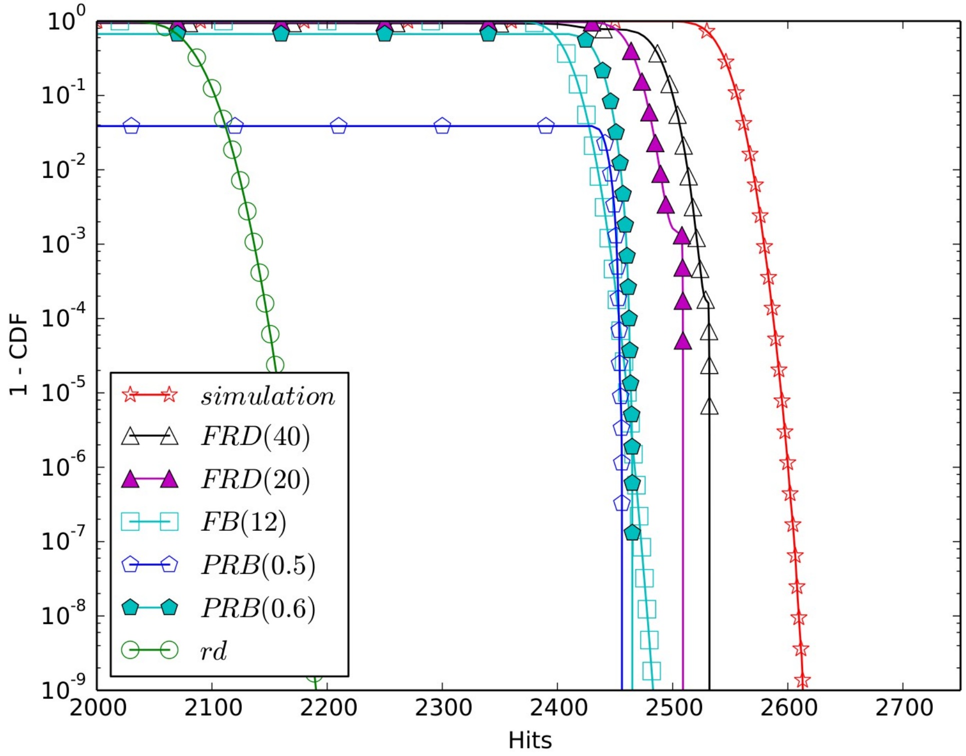


THE UNIVERSITY of York

Evaluation – insertsort hits



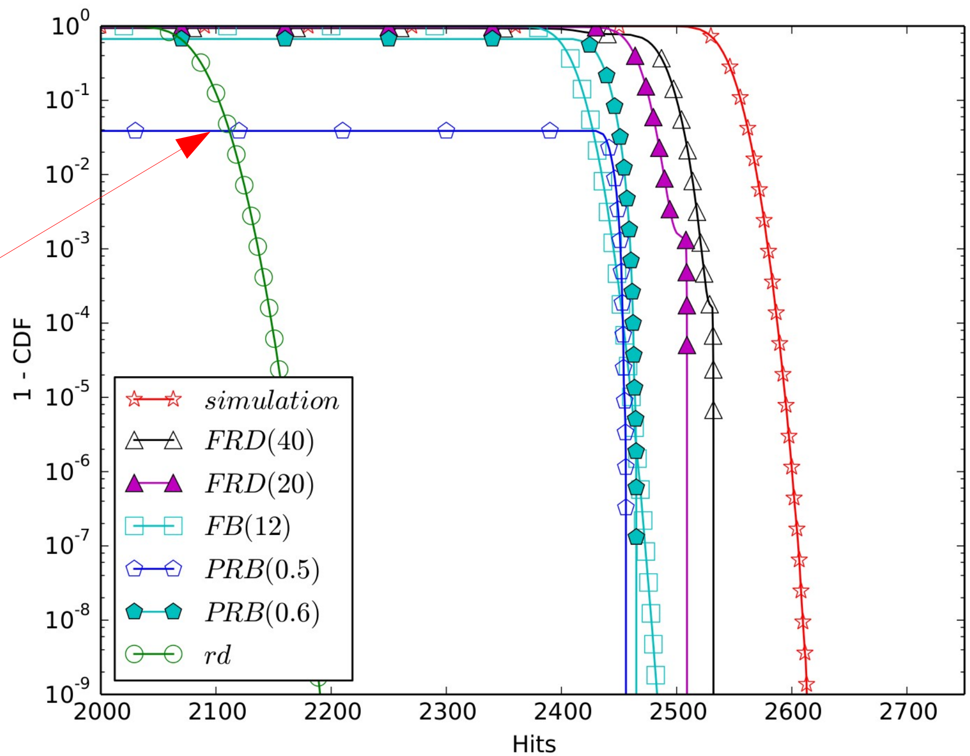
Evaluation – fir hits



THE UNIVERSITY of York

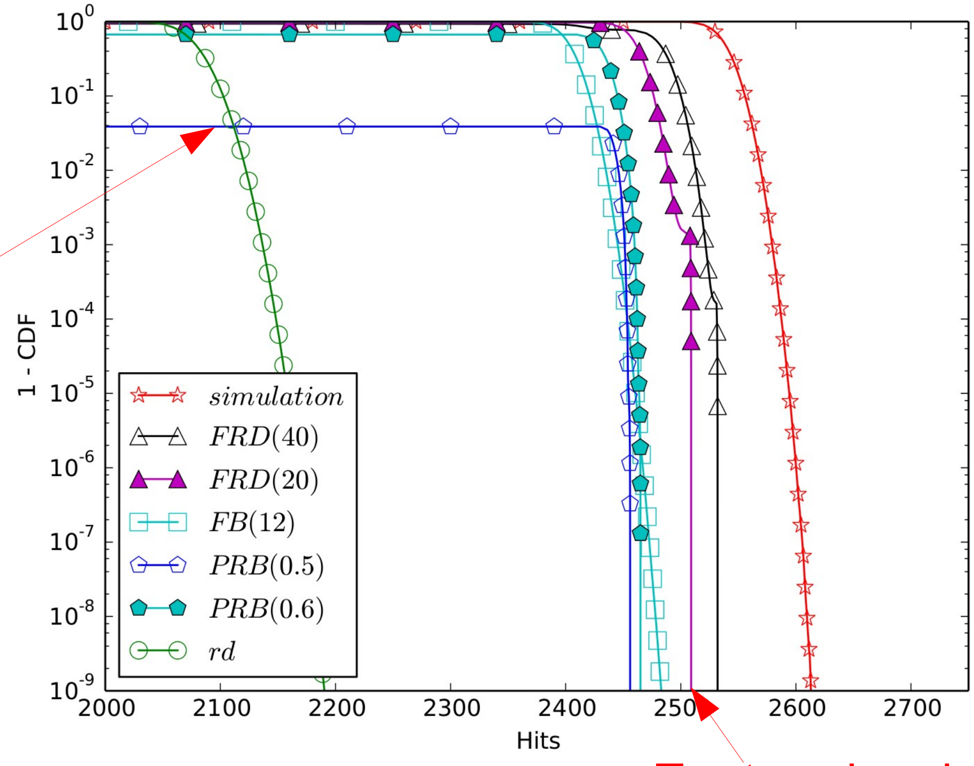
Evaluation – fir hits

At high probability, compression techniques are outperformed



Evaluation – fir hits

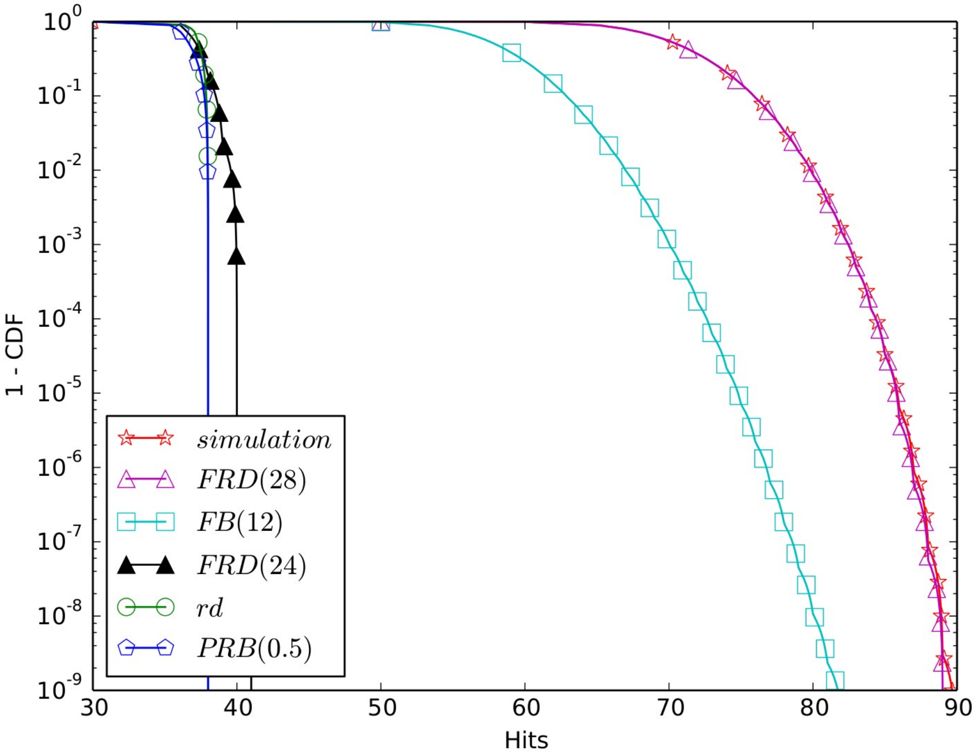
At high probability, compression techniques are outperformed



To stay ahead at higher accuracy, would need greater **α** parameter

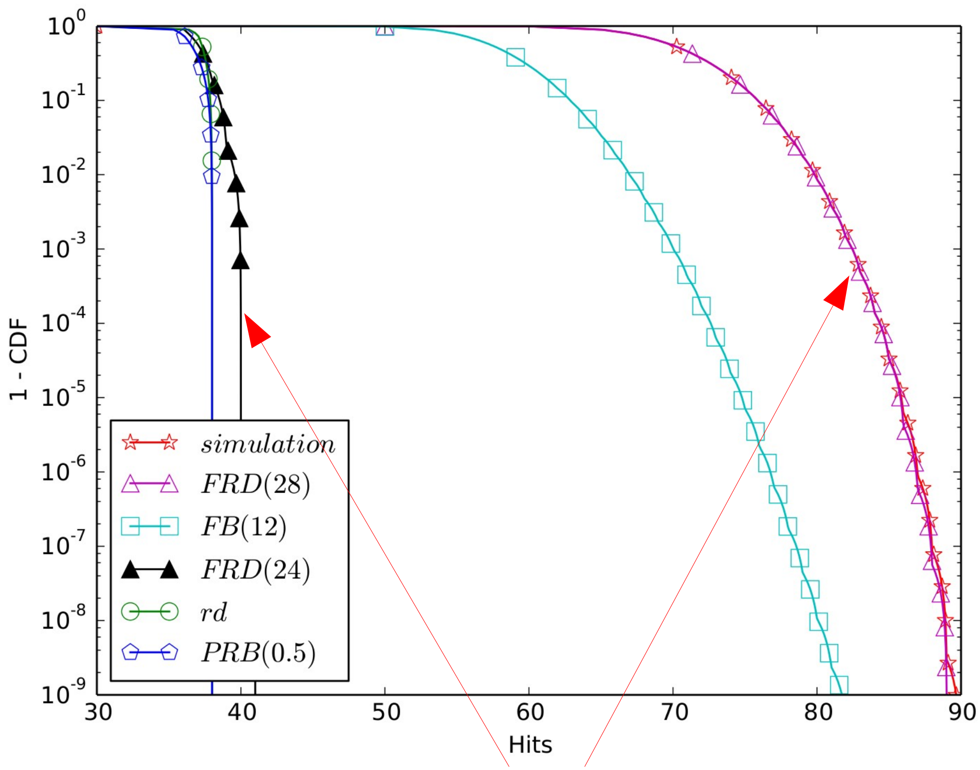
THE UNIVERSITY of York

Evaluation – bs hits



THE UNIVERSITY of York

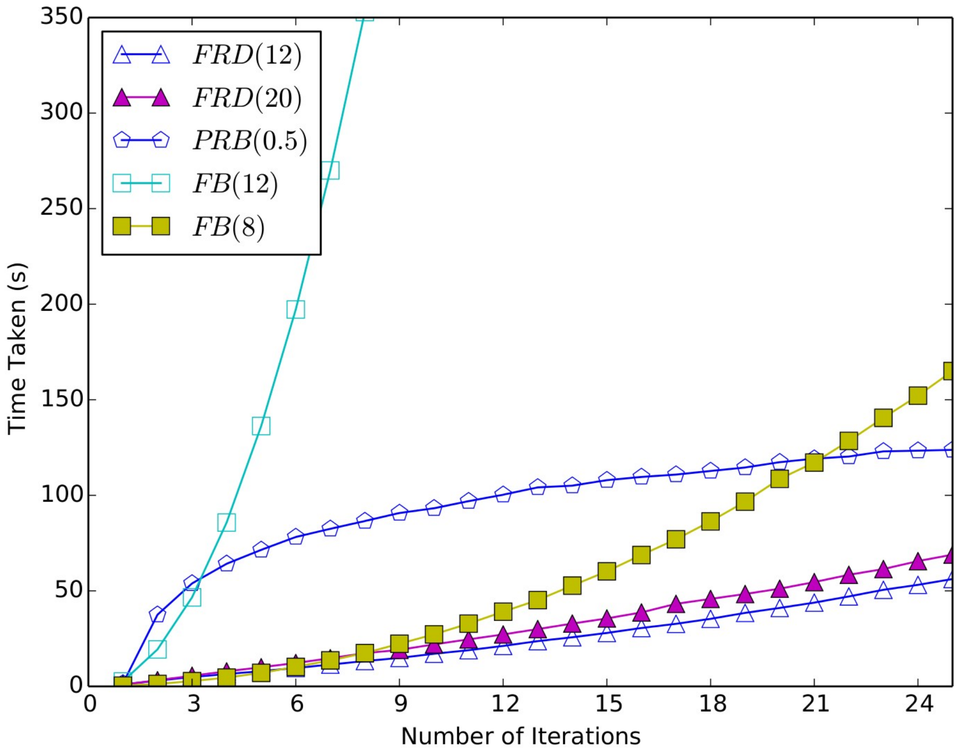
Evaluation – bs hits



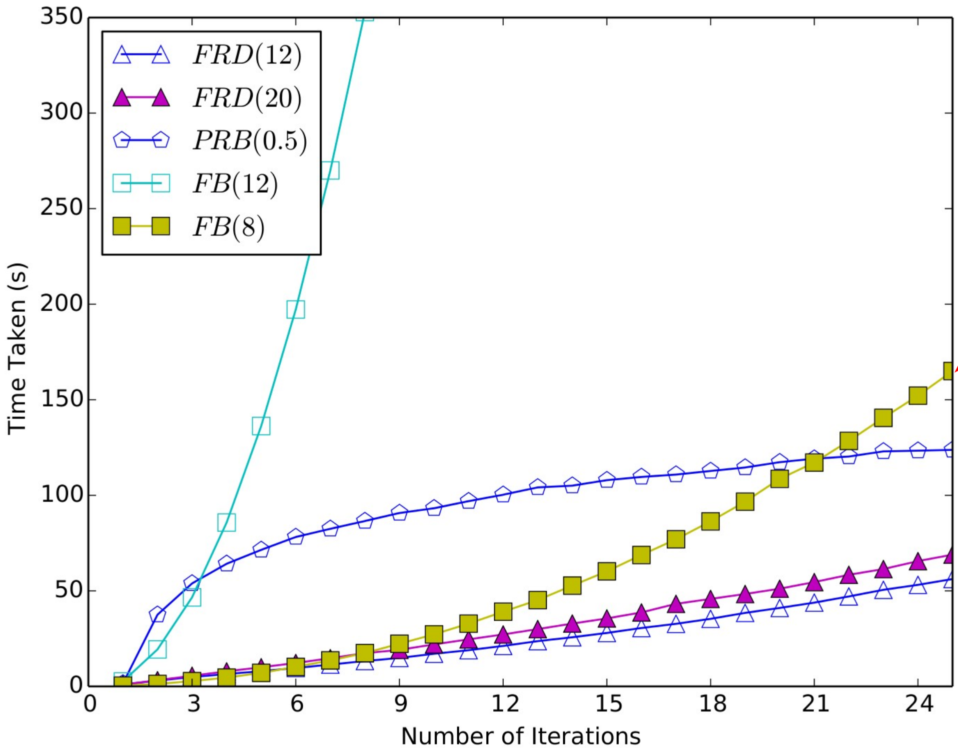
Small difference of parameter can make a large difference

THE UNIVERSITY of York

Evaluation – execution time for iterations of fibcall

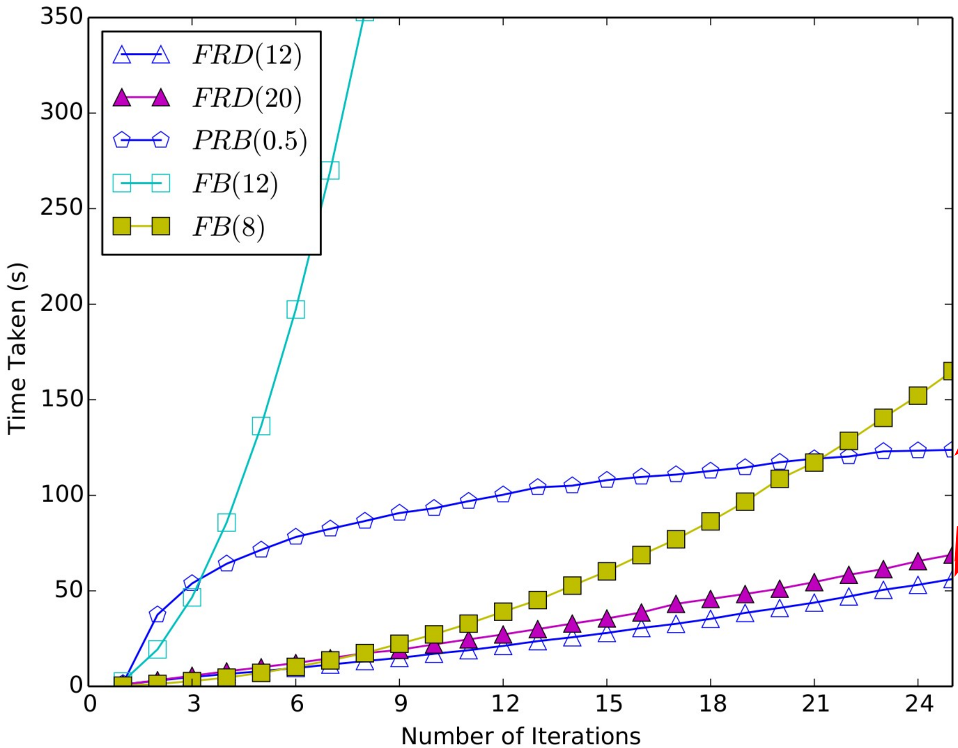


Evaluation – execution time for iterations of fibcall



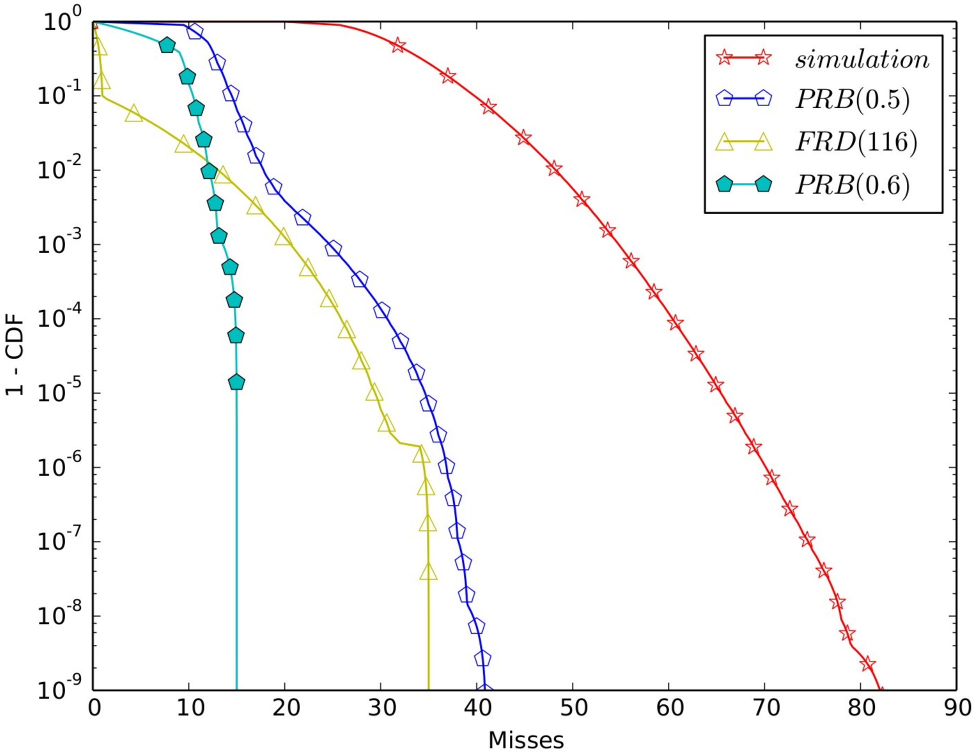
All parameters have very similar accuracy, apart from FB(8)

Evaluation – execution time for iterations of fibcall



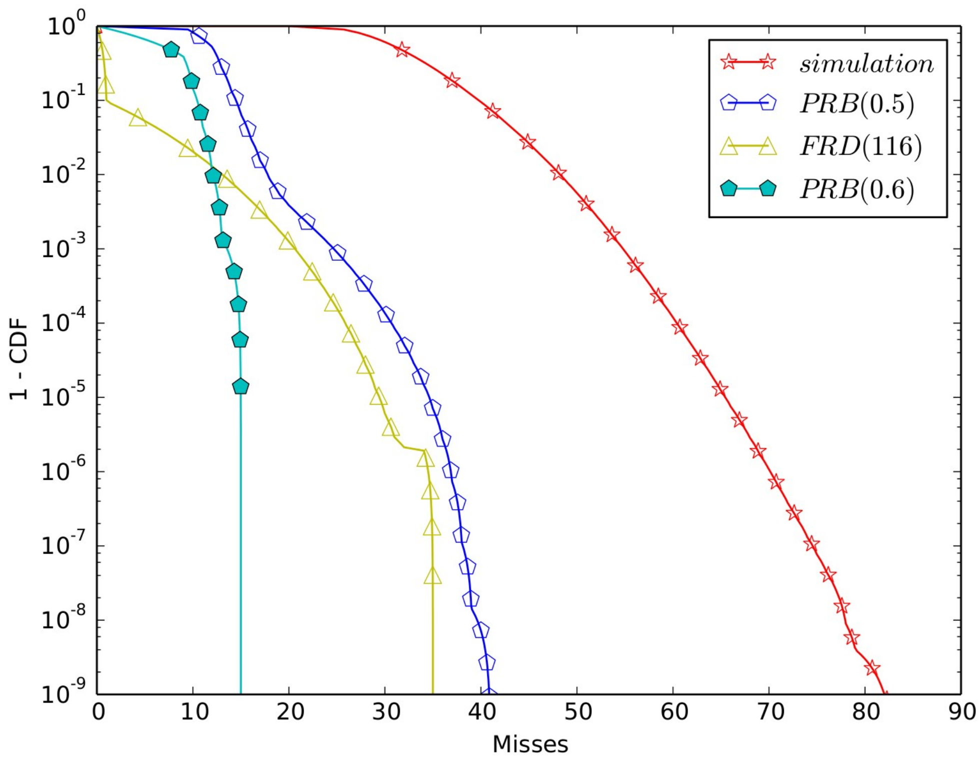
Lower complexity

Evaluation – insertsort misses



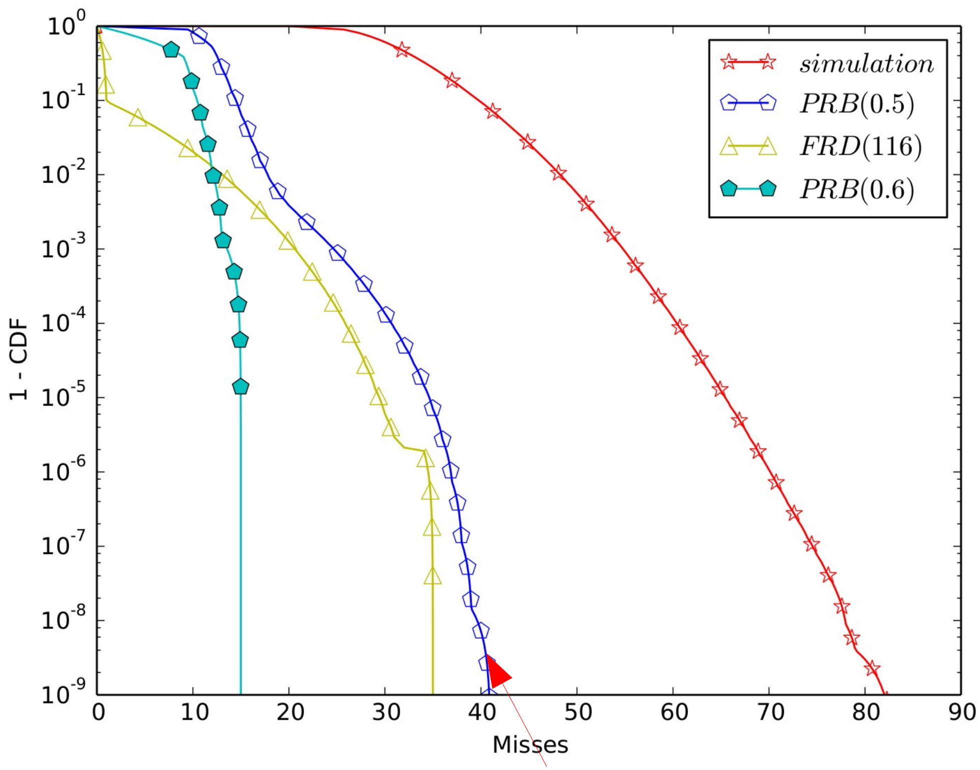
Evaluation – insertsort misses

FRD < 116 gives near zero Misses



THE UNIVERSITY of York

Evaluation – insertsort misses



PRB now outperforms FRD

Conclusions

- Parameters improve accuracy in a predictable way
- Significant improvements over previous methods in accuracy, speed and memory usage
- Demonstrates a May Analysis for a Random Replacement cache is possible

Future Work

- “Fixed Effort” compression
  - Current technique is analogous to “Fixed Quality” compression
  - “Fixed Effort” is analogous to “Fixed Bitrate”
  - Idea: Given finite time, calculate best possible result
- Extension to multipath

Any Questions?