Concurrent
Compact-fit

Concurrency & Scalability
versus
Fragmentation & Compaction

Questions

- Does allocation/deallocation throughput scale with multiple processors?

Questions

- Does allocation/deallocation throughput scale with multiple processors?
- Which aspects influence scalability?

Questions

- Does allocation/deallocation throughput scale with multiple processors?
- Which aspects influence scalability?
- Does compaction of large objects harm system latency?

Questions

- Does allocation/deallocation throughput scale with multiple processors?
- Which aspects influence scalability?
- Does compaction of large objects harm system latency?
- Does concurrency and incrementality affect memory consumption?

Partial Compaction

- Per-size-class partial compaction bound κ bounds size-class fragmentation:
  - $\kappa = 1$: fully compacting
  - $1 < \kappa < \infty$: partially compacting
  - $\kappa = \infty$: non-compacting

Partial Compaction

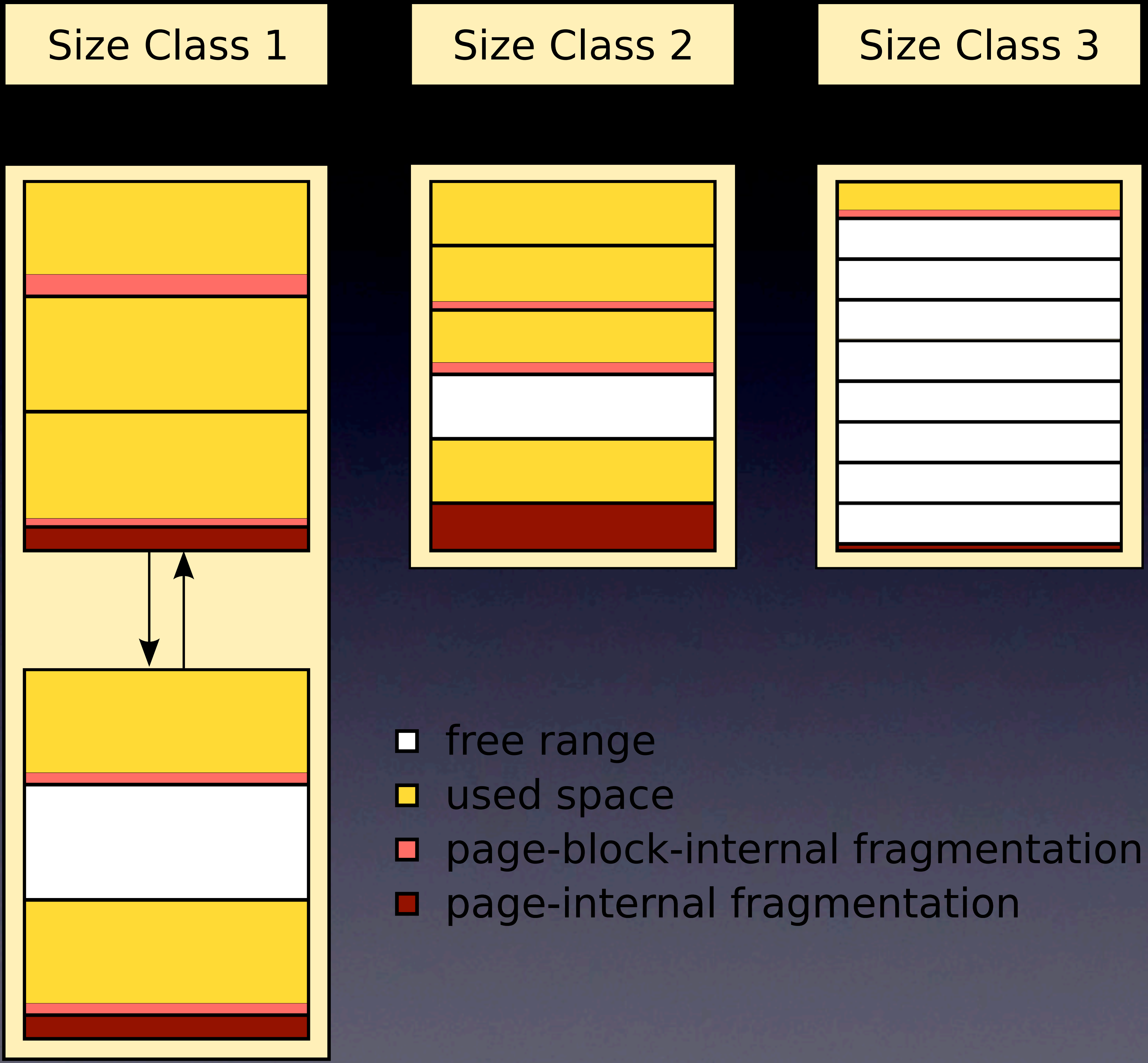
- Per-size-class partial compaction bound κ bounds size-class fragmentation:
  - $\kappa = 1$: fully compacting
  - $1 < \kappa < \infty$: partially compacting
  - $\kappa = \infty$: non-compacting
- Non-compacting CF can be optimized by not using abstract addresses

Fragmentation through Partitioning

- Fragmentation through partitioning is fixed at compile time and is not controlled by partial compaction:
  - Page-block-internal fragmentation
  - Page-internal fragmentation

Fragmentation through Partitioning

- Fragmentation through partitioning is fixed at compile time and is not controlled by partial compaction:
  - Page-block-internal fragmentation
  - Page-internal fragmentation
- May dominate overall fragmentation

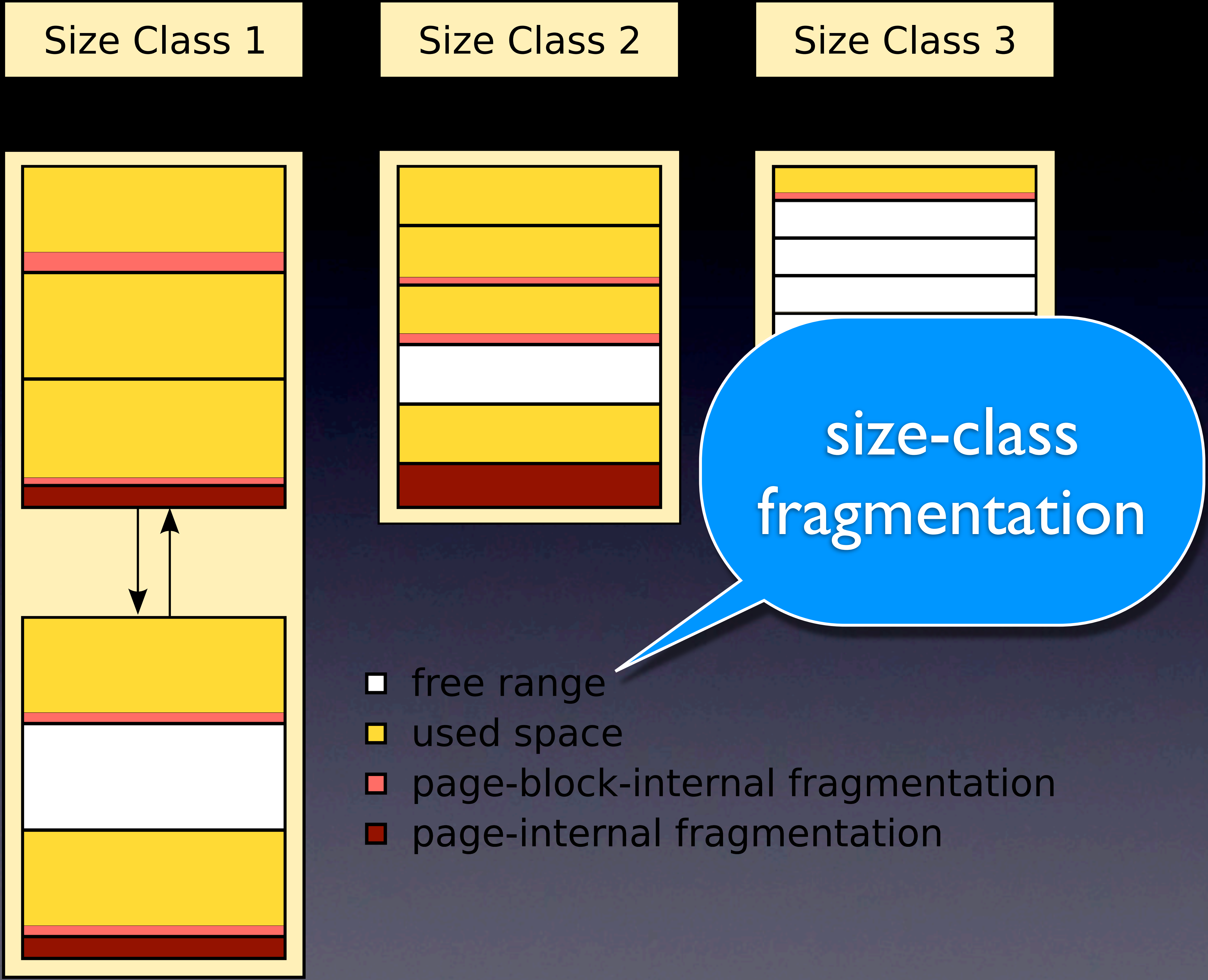


Size Class 1

Size Class 2

Size Class 3

- □ free range
- 🟨 used space
- 🟥 page-block-internal fragmentation
- 🟫 page-internal fragmentation



Size Class 1

Size Class 2

Size Class 3

size-class fragmentation

- □ free range
- 🟨 used space
- 🟥 page-block-internal fragmentation
- 🟫 page-internal fragmentation

Incremental Compaction

- Global compaction increment ι bounds size of memory involved in any atomic compaction operation:
  - $1 < \iota < \infty$: incremental compaction of objects larger than ι
  - $\iota = \infty$: non-incremental compaction

Incremental Compaction

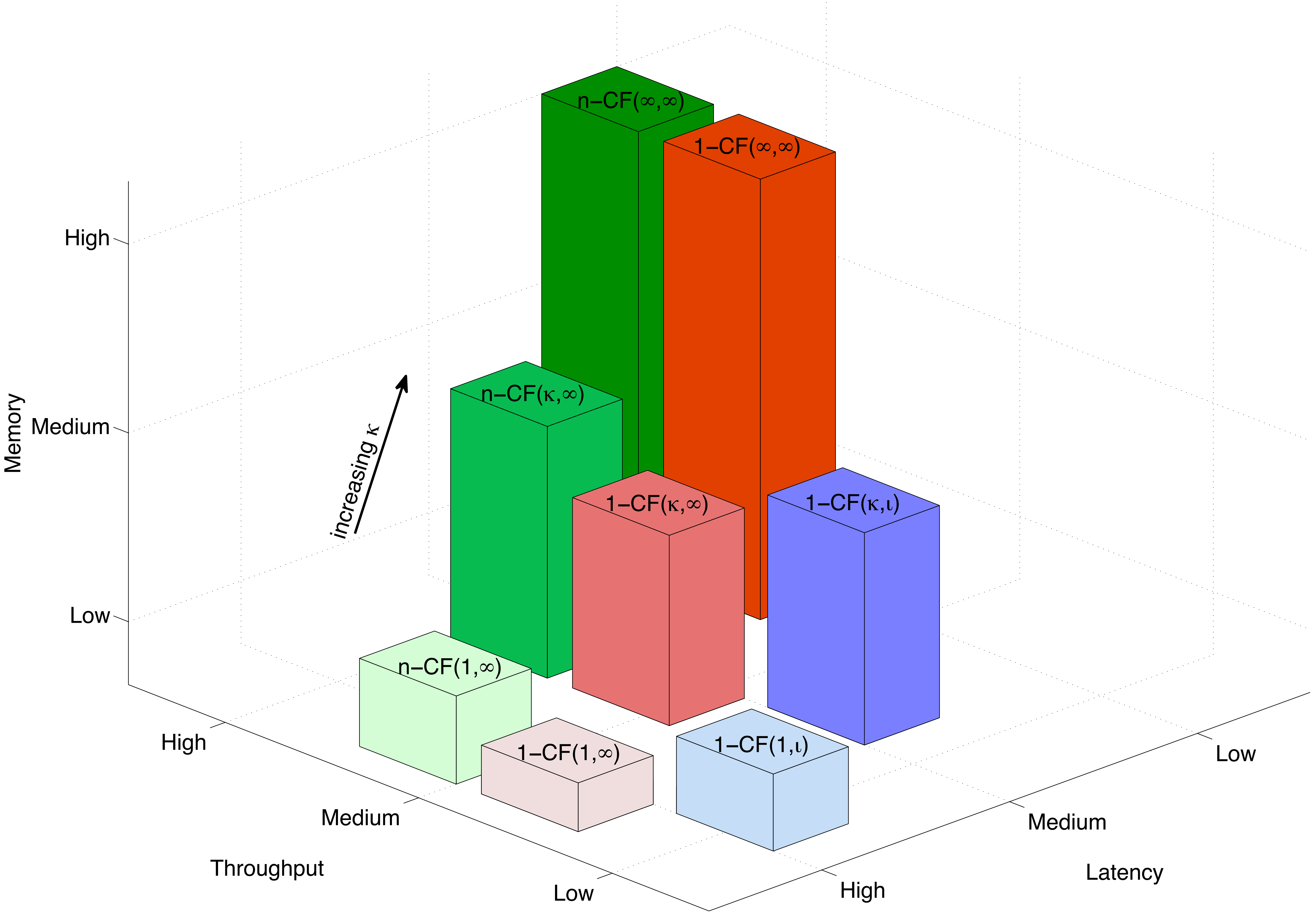
- Global compaction increment ι bounds size of memory involved in any atomic compaction operation:
  - $1 < \iota < \infty$: incremental compaction of objects larger than ι
  - $\iota = \infty$: non-incremental compaction
- Incremental compaction creates transient size-class fragmentation

CF Configurations

- 1-CF(κ, ι)
  - one CF instance for multiple threads
  - partial compaction bound κ
  - compaction increment ι

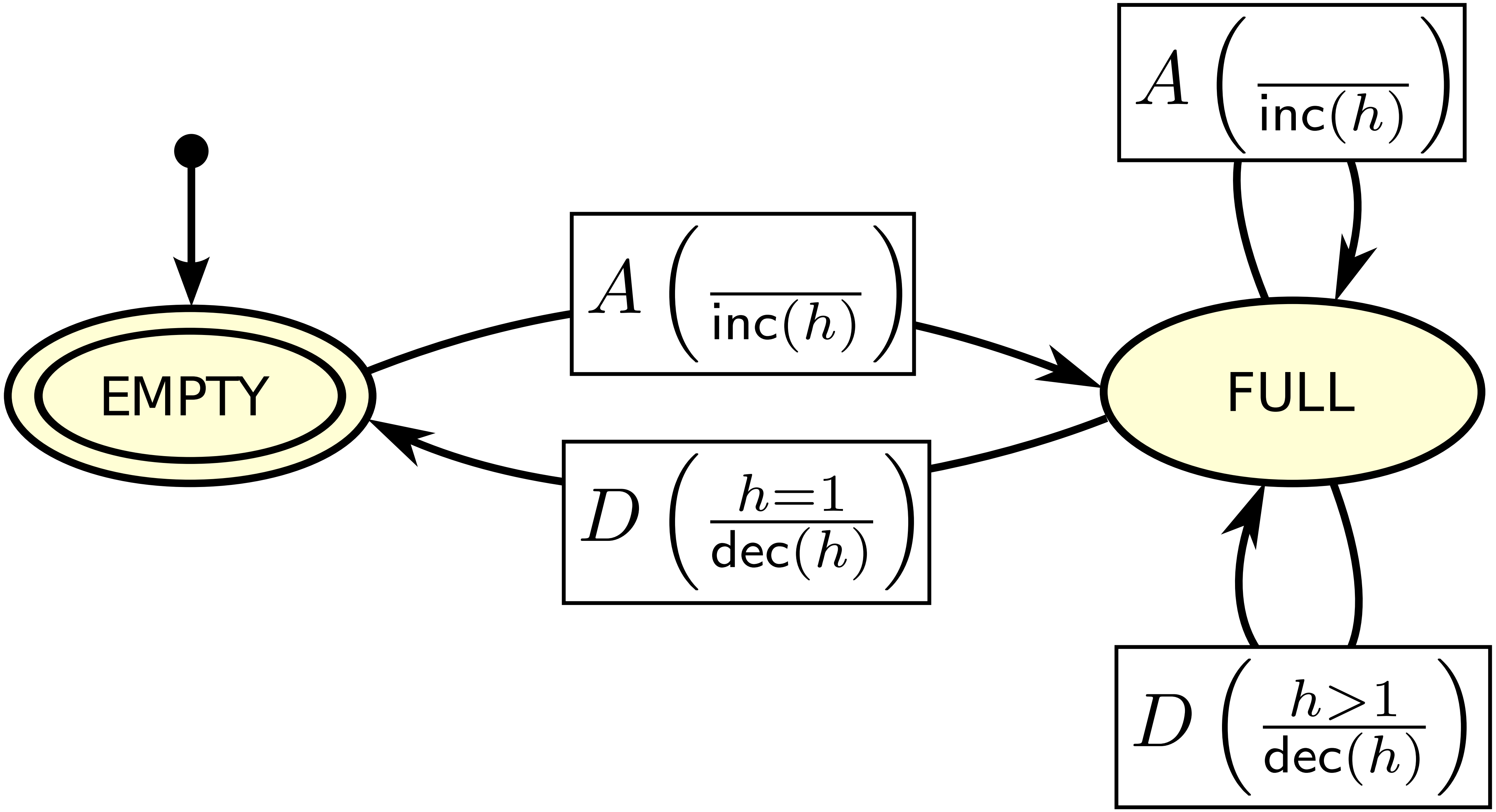
CF Configurations

- 1-CF(κ, ι)
  - one CF instance for multiple threads
  - partial compaction bound κ
  - compaction increment ι
- n-CF(κ, ι)
  - n CF instances for n threads
  - allows to control degree of sharing

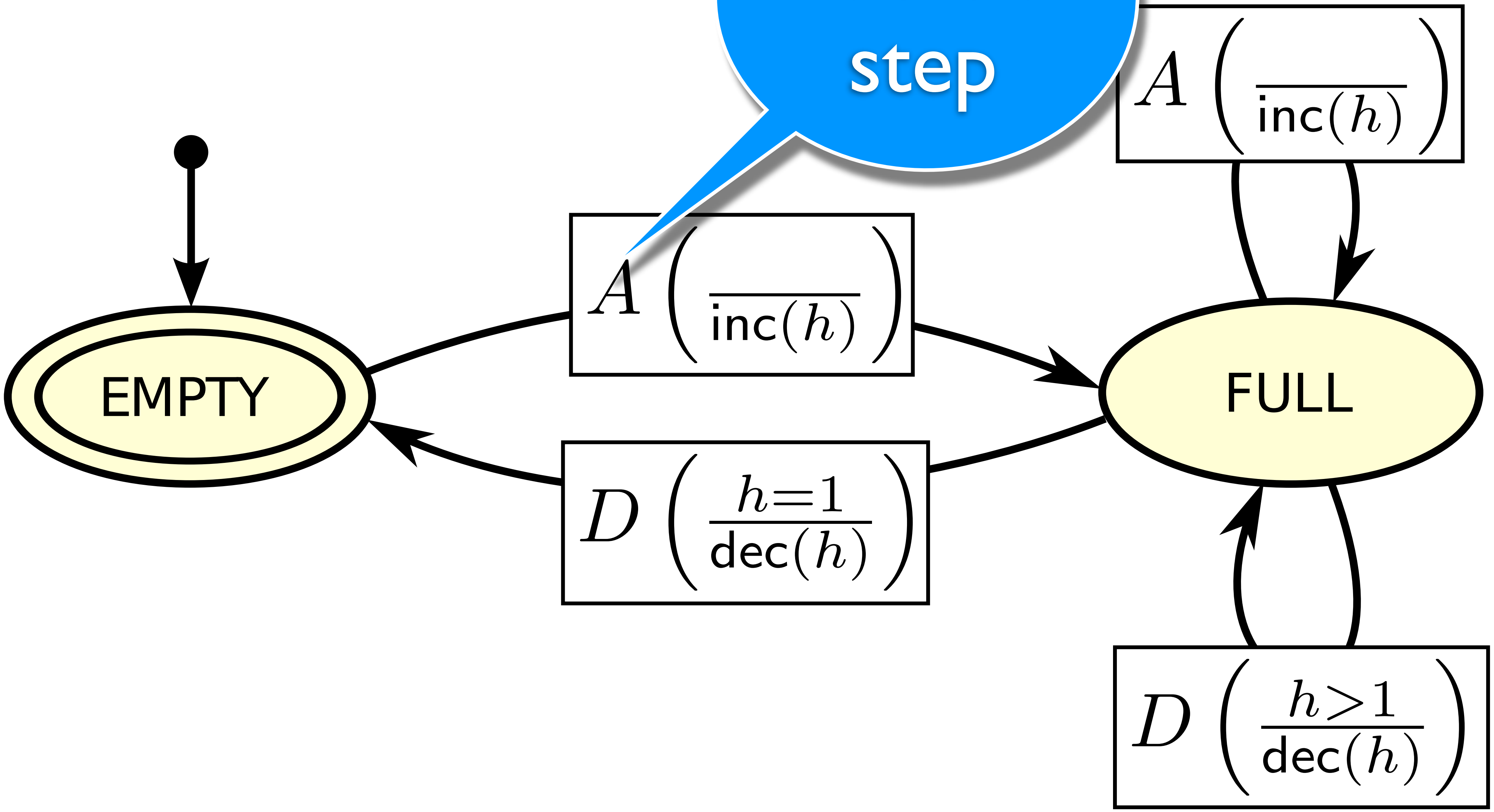

n–CF(∞,∞)
1–CF(∞,∞)
n–CF(κ,∞)
1–CF(κ,∞)
1–CF(κ,ι)
n–CF(1,∞)
1–CF(1,∞)
1–CF(1,ι)
increasing κ
Memory
High
Medium
Low
Throughput
High
Medium
Low
Latency
Low
Medium
High

To make CF
concurrent and incremental
we model the algorithm
as a
finite state machine
whose transitions
must be atomic!

Size-Class Automaton for π = 1

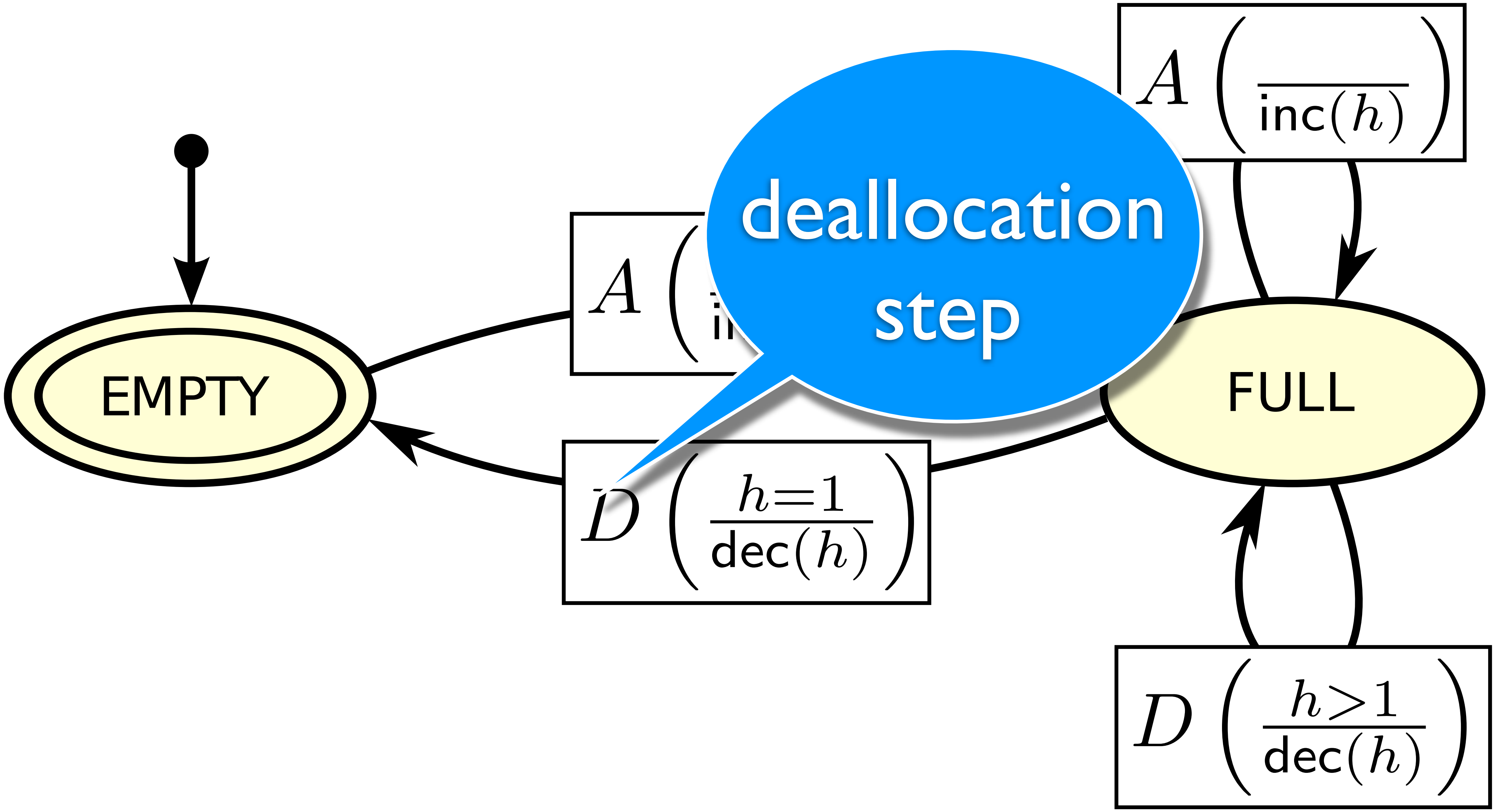

$A\left(\frac{}{\text{inc}(h)}\right)$
EMPTY
$A\left(\frac{}{\text{inc}(h)}\right)$
FULL
$D\left(\frac{h=1}{\text{dec}(h)}\right)$
$D\left(\frac{h>1}{\text{dec}(h)}\right)$

Size-Class Automaton for Π


EMPTY
FULL
$A\left(\frac{}{\text{inc}(h)}\right)$
$A\left(\frac{}{\text{inc}(h)}\right)$
$D\left(\frac{h=1}{\text{dec}(h)}\right)$
$D\left(\frac{h>1}{\text{dec}(h)}\right)$

allocation step

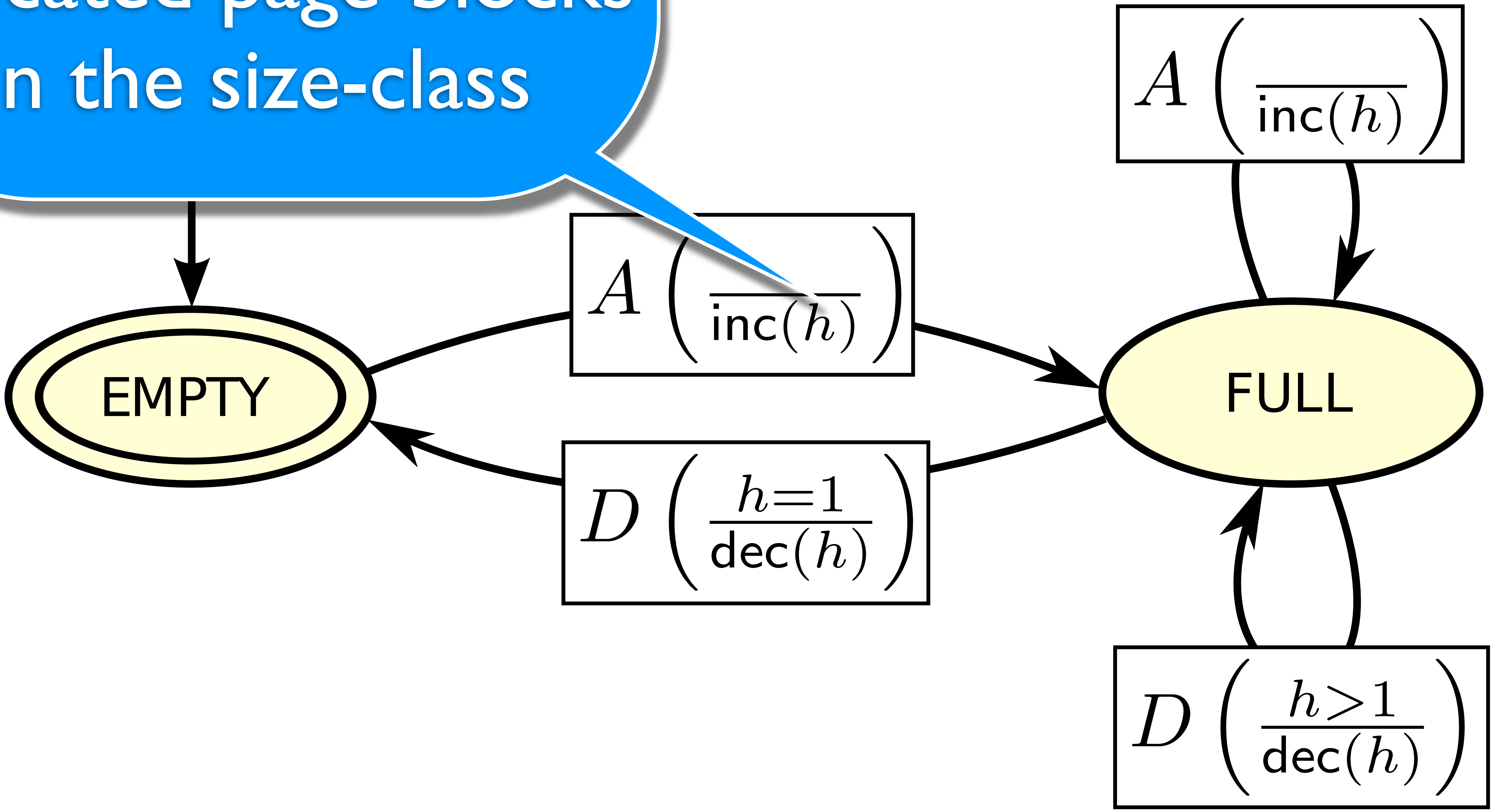
Size-Class Automaton for π = 1


$A\left(\frac{}{\mathsf{inc}(h)}\right)$
deallocation step
$A\Big($
EMPTY
FULL
$D\left(\frac{h=1}{\mathsf{dec}(h)}\right)$
$D\left(\frac{h>1}{\mathsf{dec}(h)}\right)$

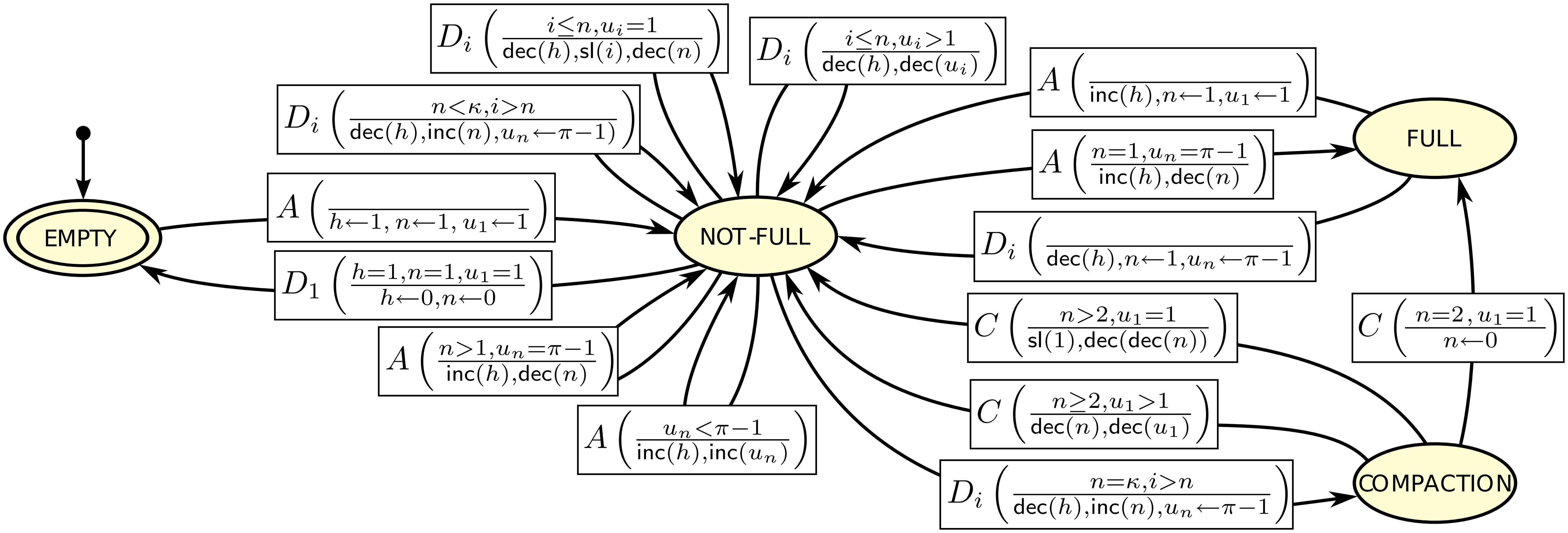
Size-Class Automaton

π = 1

h is the total # of allocated page-blocks in the size-class


EMPTY
$A\left(\frac{}{\mathsf{inc}(h)}\right)$
FULL
$A\left(\frac{}{\mathsf{inc}(h)}\right)$
$D\left(\frac{h=1}{\mathsf{dec}(h)}\right)$
$D\left(\frac{h>1}{\mathsf{dec}(h)}\right)$

Size-Class Automaton for π > 1


EMPTY
NOT-FULL
FULL
COMPACTION
$D_i\left(\frac{i\leq n, u_i=1}{\mathsf{dec}(h), \mathsf{sl}(i), \mathsf{dec}(n)}\right)$
$D_i\left(\frac{i\leq n, u_i>1}{\mathsf{dec}(h), \mathsf{dec}(u_i)}\right)$
$D_i\left(\frac{n<\kappa, i>n}{\mathsf{dec}(h), \mathsf{inc}(n), u_n\leftarrow\pi-1)}\right)$
$A\left(\frac{}{h\leftarrow 1, n\leftarrow 1, u_1\leftarrow 1}\right)$
$D_1\left(\frac{h=1, n=1, u_1=1}{h\leftarrow 0, n\leftarrow 0}\right)$
$A\left(\frac{n>1, u_n=\pi-1}{\mathsf{inc}(h), \mathsf{dec}(n)}\right)$
$A\left(\frac{u_n<\pi-1}{\mathsf{inc}(h), \mathsf{inc}(u_n)}\right)$
$A\left(\frac{}{\mathsf{inc}(h), n\leftarrow 1, u_1\leftarrow 1}\right)$
$A\left(\frac{n=1, u_n=\pi-1}{\mathsf{inc}(h), \mathsf{dec}(n)}\right)$
$D_i\left(\frac{}{\mathsf{dec}(h), n\leftarrow 1, u_n\leftarrow\pi-1}\right)$
$C\left(\frac{n>2, u_1=1}{\mathsf{sl}(1), \mathsf{dec}(\mathsf{dec}(n))}\right)$
$C\left(\frac{n=2, u_1=1}{n\leftarrow 0}\right)$
$C\left(\frac{n\geq 2, u_1>1}{\mathsf{dec}(n), \mathsf{dec}(u_1)}\right)$
$D_i\left(\frac{n=\kappa, i>n}{\mathsf{dec}(h), \mathsf{inc}(n), u_n\leftarrow\pi-1}\right)$

h is the total # of allocated page-blocks in the size-class

n is the # of not-full pages

u$_i$ is the # of used page-blocks in a not-full page *i*