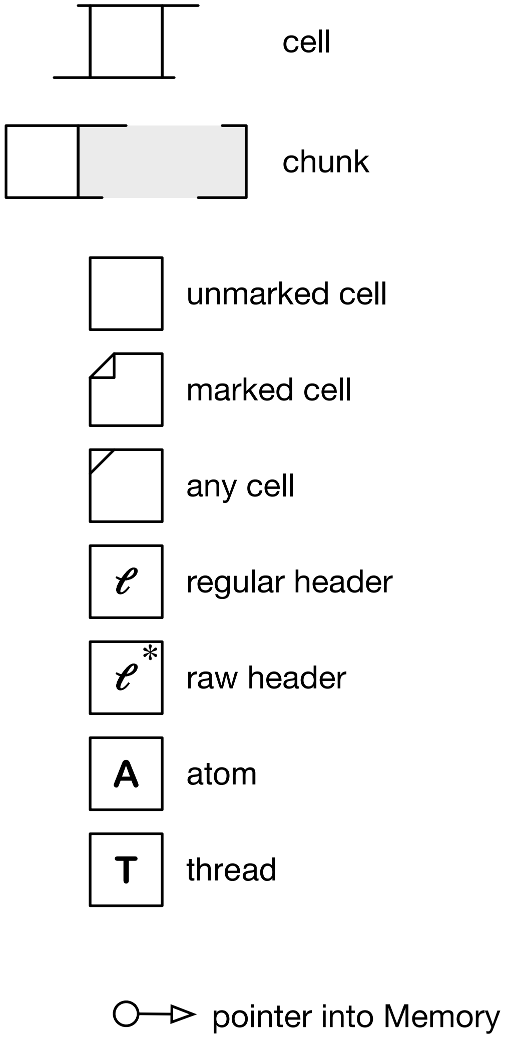
Jonkers mark-thread



cell

chunk

unmarked cell

marked cell

any cell

regular header

raw header

atom

thread

pointer into Memory

pointers = $\mathbb{N}$

flags = { a(tom), m(arked), u(nmarked) }
cells = pointers × flags

$*$: pointers $\longleftrightarrow$ cells : $\quad p \longleftrightarrow [\pi , \varphi]$

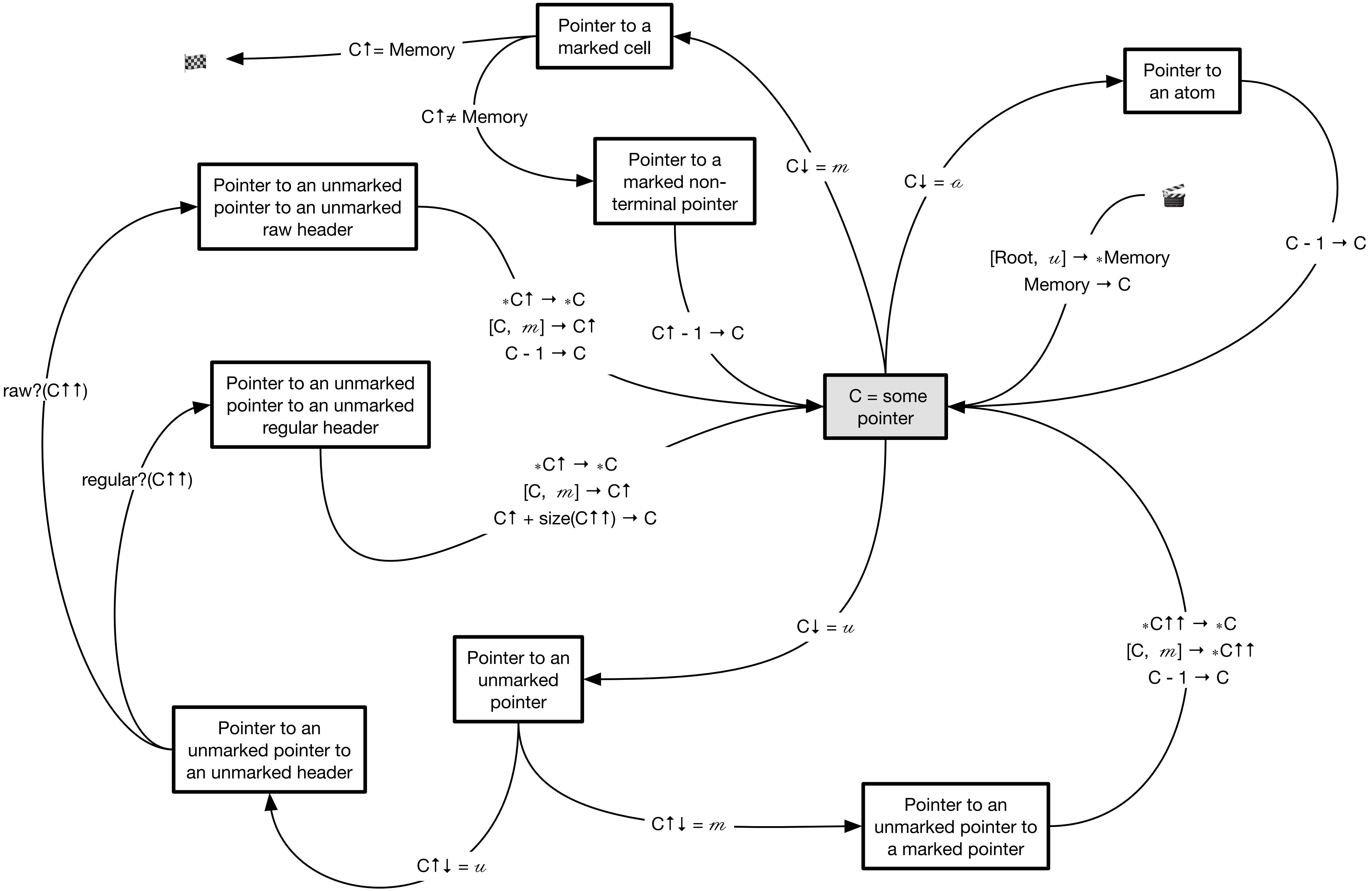
$\uparrow$: pointers $\longrightarrow$ pointers : $\quad p \longmapsto p{\uparrow} \equiv {*p}_{\pi}$

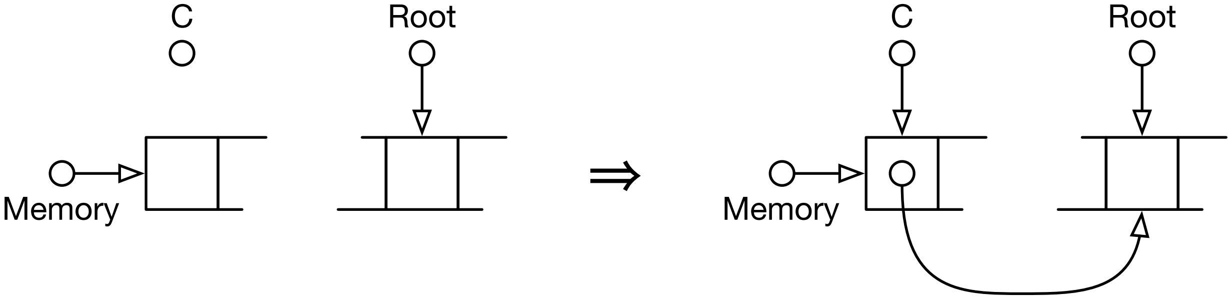
$\downarrow$: pointers $\longrightarrow$ flags : $\quad p \longmapsto p{\downarrow} \equiv {*p}_{\varphi}$

regular? : pointers $\longrightarrow$ boolean
raw? : pointers $\longrightarrow$ boolean
size : pointers $\longrightarrow$ $\mathbb{N}$

Memory : memory pointer
Root : root pointer

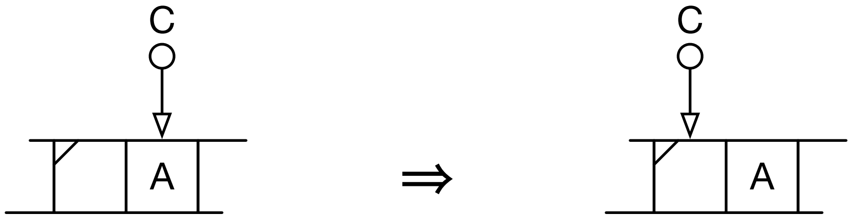
C : current pointer
P : previous pointer


Pointer to a marked cell
C↑= Memory
🏁
C↑≠ Memory
Pointer to a marked non-terminal pointer
Pointer to an atom
C↓ = m
C↓ = a
🎬
[Root, u] → ∗Memory
Memory → C
C - 1 → C
Pointer to an unmarked pointer to an unmarked raw header
∗C↑ → ∗C
[C, m] → C↑
C - 1 → C
C↑ - 1 → C
C = some pointer
raw?(C↑↑)
Pointer to an unmarked pointer to an unmarked regular header
∗C↑ → ∗C
[C, m] → C↑
C↑ + size(C↑↑) → C
regular?(C↑↑)
C↓ = u
∗C↑↑ → ∗C
[C, m] → ∗C↑↑
C - 1 → C
Pointer to an unmarked pointer
Pointer to an unmarked pointer to an unmarked header
C↑↓ = m
Pointer to an unmarked pointer to a marked pointer
C↑↓ = u



{ [Root, u] , Memory } → { $*$Memory , C }

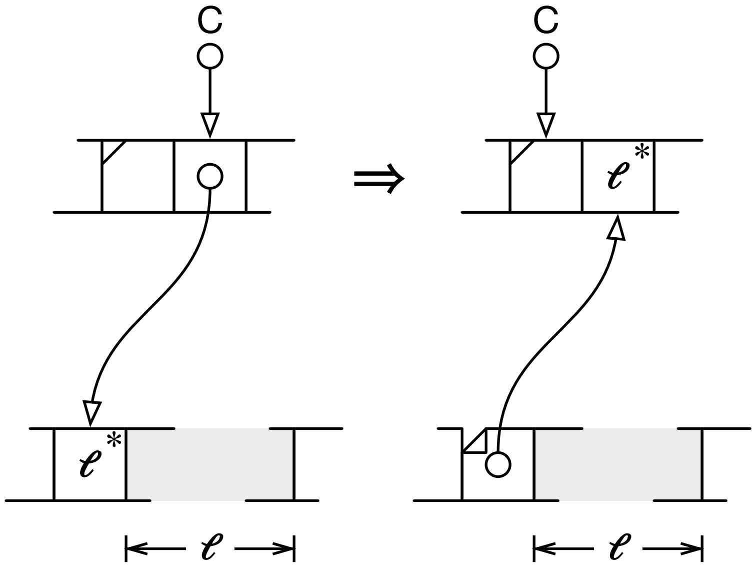
C↓ = 𝒶



{ C - 1 } → { C }

Pointer to an atom

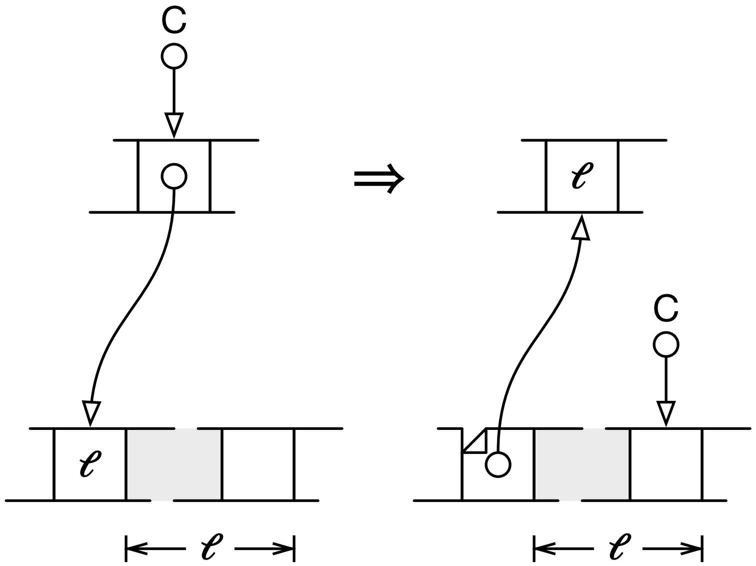
$$(C\downarrow = u) \land (C\uparrow\downarrow = u) \land \text{raw?}(C\uparrow\uparrow)$$



$$\{\, {*}C\uparrow,\ [C,\ m],\ C - 1 \,\} \rightarrow \{\, {*}C,\ {*}C\uparrow,\ C \,\}$$

Pointer to an unmarked pointer to an unmarked raw header

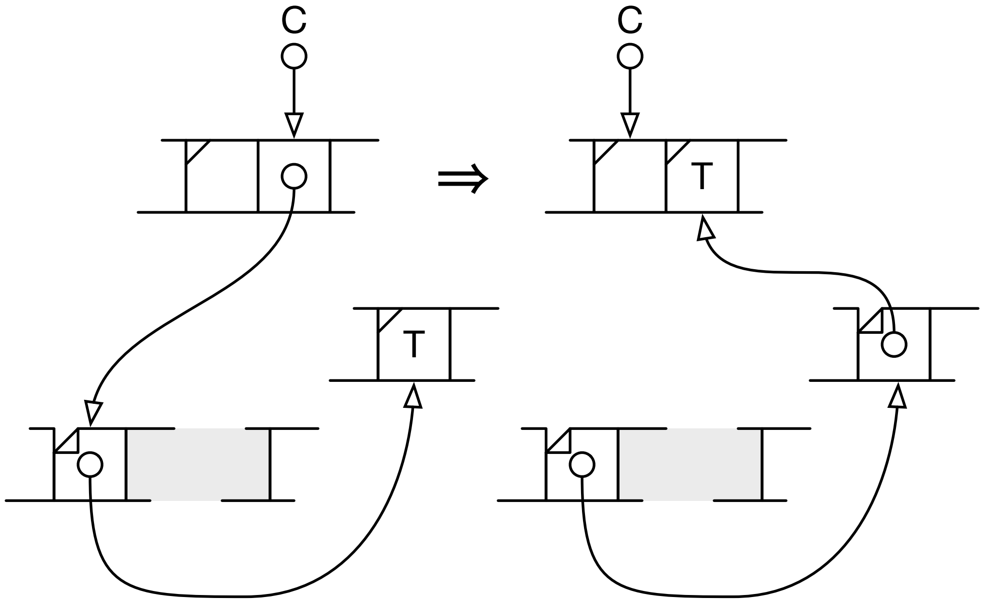
$(C\downarrow = u) \wedge (C\uparrow\downarrow = u) \wedge \text{regular?}(C\uparrow\uparrow) \wedge (\text{size}(C\uparrow\uparrow) \neq 0)$



$\{ *C\uparrow ,\ [C,\ m] ,\ C\uparrow + \text{size}(C\uparrow\uparrow) \} \rightarrow \{ *C ,\ *C\uparrow ,\ C \}$

Pointer to an unmarked pointer to an unmarked regular header

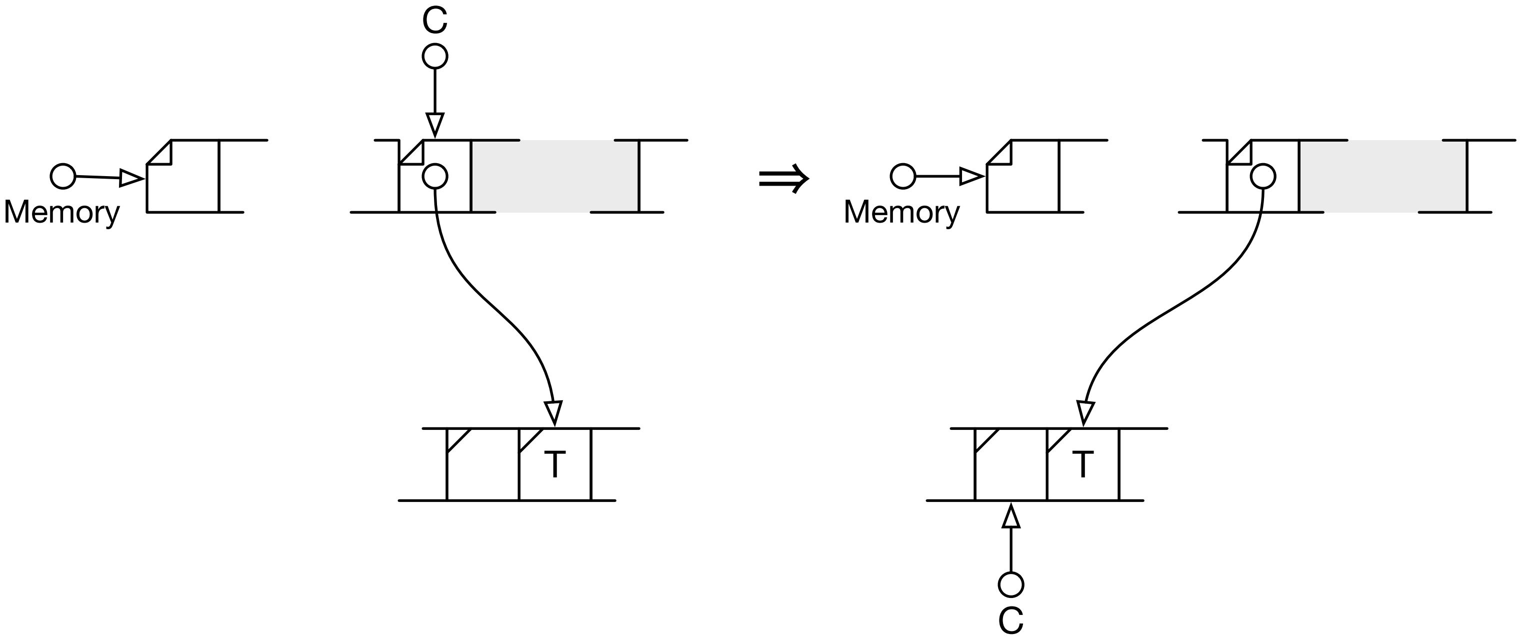
$(C\downarrow = u) \land (C\uparrow\downarrow = m)$



$\{ \ast C\uparrow\uparrow , [C,\ m] , C - 1 \} \to \{ \ast C , \ast C\uparrow\uparrow , C \}$

Pointer to an unmarked pointer to a marked pointer

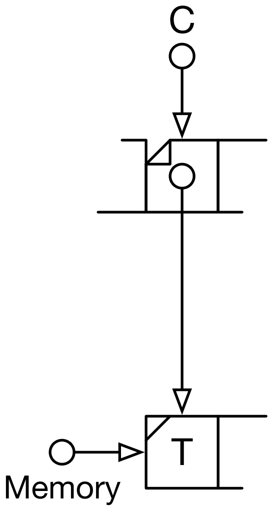
(C↓ = m) ∧ (C↑ ≠ Memory)



{ C↑ - 1 } → { C }

Pointer to a marked non-root pointer

$(C\downarrow = \mathscr{m}) \wedge (C\uparrow = \text{Memory})$



🏁

Stop

```
typedef struct CEL * ptr;
typedef enum {a, m, u} flg;
typedef struct CEL { ptr P; flg F; } cel;

ptr Memory;
unsigned is_raw(ptr);
unsigned size(ptr);

void Jonkers_mark_thread(cel Root)
  { ptr C, C_, C__;
    *Memory = Root;                           // *Memory <- Root
    for (C = Memory;;)                        // C <- Memory
      if (C->F == a)                          // Cv = a
        C -= 1;                               // C = C - 1
      else                                    // Cv ≠ a
        { C_ = C->P;                          // C^
          if (C->F == u)                      // Cv = u
            { C__ = C_->P;                    // C^^
              if (C_->F == u)                 // C^v = u
                { *C = *C_;                   // *C <- *C^
                  *C_ = (cel){ C, m };        // *C_ <- [C, m]
                  if (is_raw(C__))            // raw?(C^^)
                    C -= 1;                   // C <- C - 1
                  else                        // regular?(C^^)
                    C = C_ + size(C__); }     // C <- C^ + size(C^^)
              else                            // C^v = m
                { *C = *C__;                  // *C <- *C^^
                  *C__ = (cel){ C, m };       // *C__ <- [C, m]
                  C -= 1; }}                  // C <- C - 1
          else                                // Cv = m
            if (C_ != Memory)                 // C^ ≠ Memory
              C = C_ - 1;                     // C <- C^ - 1
            else                              // C^ = Memory
              break; }}                       // stop
```