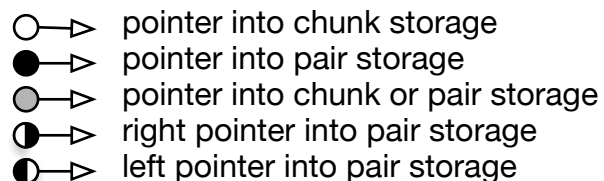
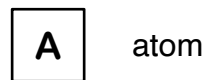
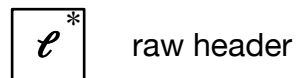
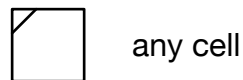
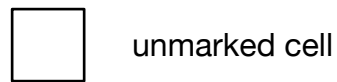
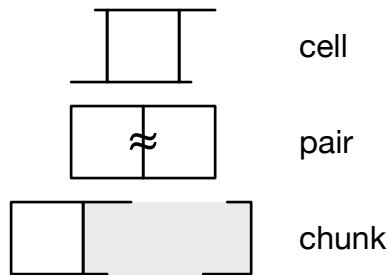


# Jonkers-Schorr-Waite mark-thread



chunks, pairs  $\subset \mathbb{N}$   
 chunks  $\cap$  pairs =  $\emptyset$   
 pointers = chunks  $\cup$  pairs

types = {  $a(\text{tom})$ ,  $h(\text{eader})$ ,  $p(\text{ointer})$  }

marks = {  $m(\text{arked})$ ,  $u(\text{nmarked})$  }

cells = pointers  $\times$  types  $\times$  markers

$*$ : pointers  $\longleftrightarrow$  cells :  $p \longleftrightarrow [\pi, \tau, \mu]$

$\uparrow$ : pointers  $\longrightarrow$  pointers :  $p \longmapsto p\uparrow \equiv *p_\pi$

$\uparrow$ : pointers  $\longrightarrow$  types :  $p \longmapsto p\uparrow \equiv *p_\tau$

$\downarrow$ : pointers  $\longrightarrow$  markers :  $p \longmapsto p\downarrow \equiv *p_\mu$

chunk? : pointers  $\longrightarrow$  boolean

pair? : pointers  $\longrightarrow$  boolean

pointer? : pointers  $\longrightarrow$  boolean

raw? : pointers  $\longrightarrow$  boolean

regular? : pointers  $\longrightarrow$  boolean

size : pointers  $\longrightarrow \mathbb{N}$

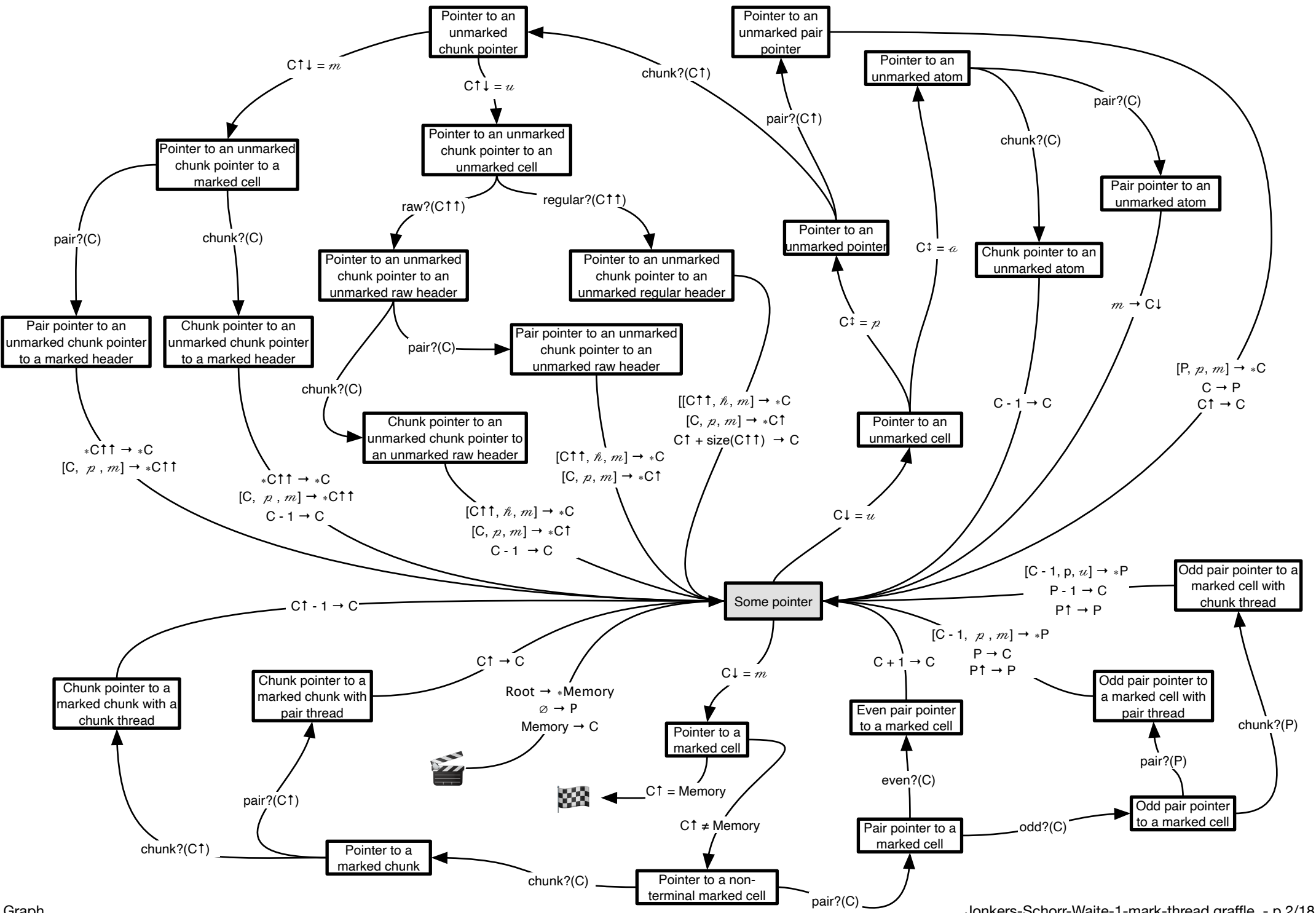
Memory : memory pointer

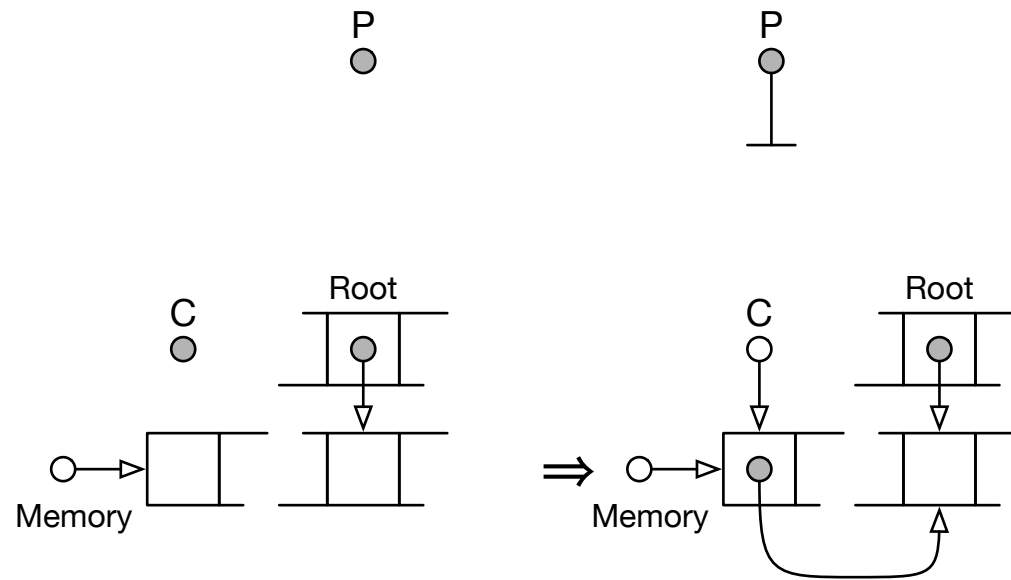
Root : root pointer

C : current pointer

P : previous pointer

T : thread pointer

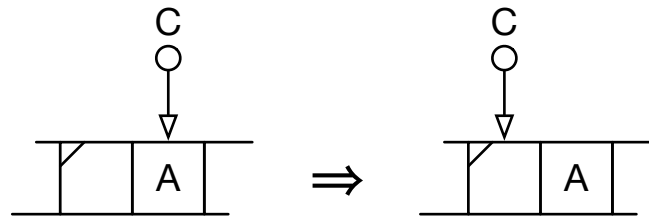




$\{ \text{Root}, \emptyset, \text{Memory} \} \rightarrow \{ * \text{Memory}, P, C \}$

$$(C \downarrow = u) \wedge (C \uparrow = a) \wedge \text{chunk?}(C)$$

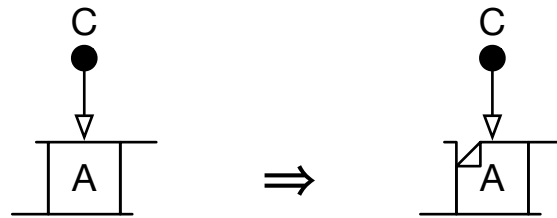
P

$$\{ C - 1 \} \rightarrow \{ C \}$$

$$(C\downarrow = u) \wedge (C\uparrow = a) \wedge \text{pair?}(C)$$

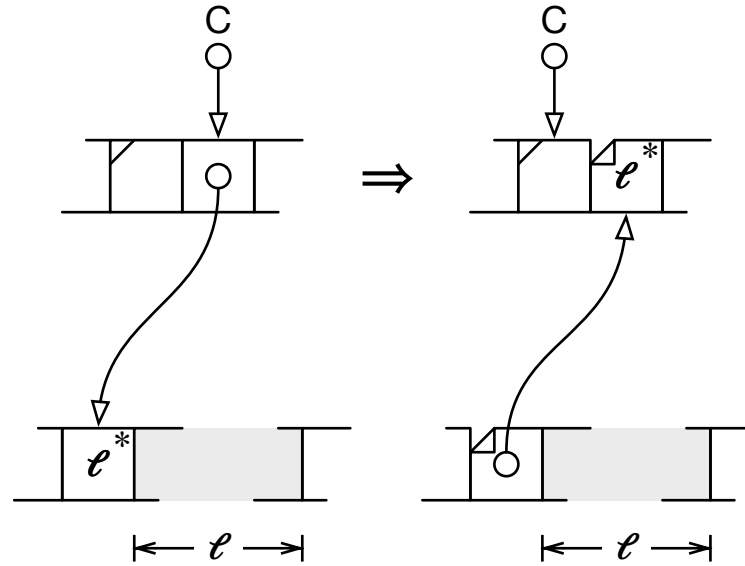
P

$$\{ m \} \rightarrow \{ C\downarrow \}$$

$$(C\downarrow = u) \wedge (C\uparrow \neq a) \wedge \text{chunk?}(C\uparrow) \wedge (C\uparrow\downarrow = u) \wedge \text{raw?}(C\uparrow\uparrow) \wedge \text{chunk?}(C)$$

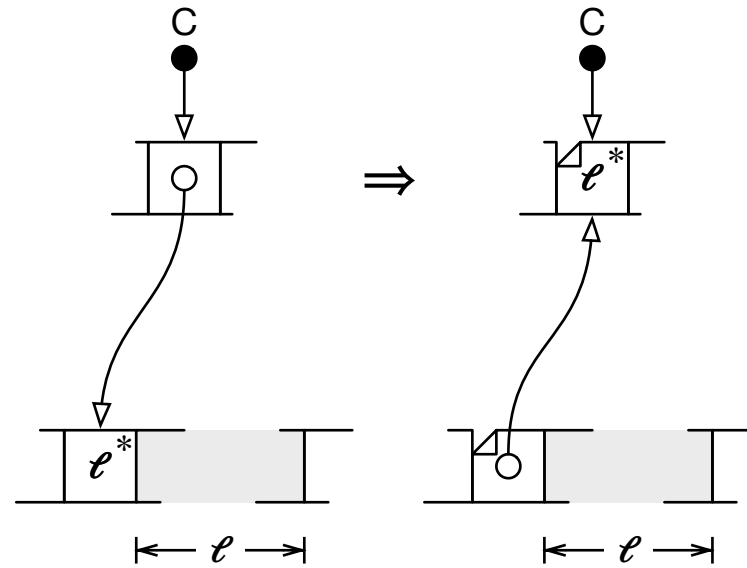
P  
●  
⋮



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

$$(C \downarrow = u) \wedge (C \uparrow \neq \omega) \wedge \text{chunk?}(C \uparrow) \wedge (C \uparrow \downarrow = u) \wedge \text{raw?}(C \uparrow \uparrow) \wedge \text{pair?}(C)$$

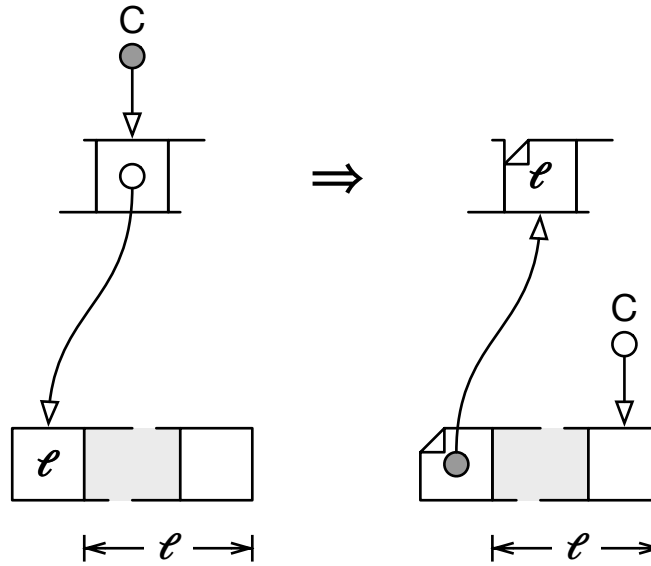
P  
●  
⋮



$$\{ [C \uparrow \uparrow, \hbar, m], [C, p, m] \} \rightarrow \{ *C, *C \uparrow \}$$

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

P  
●  
⋮

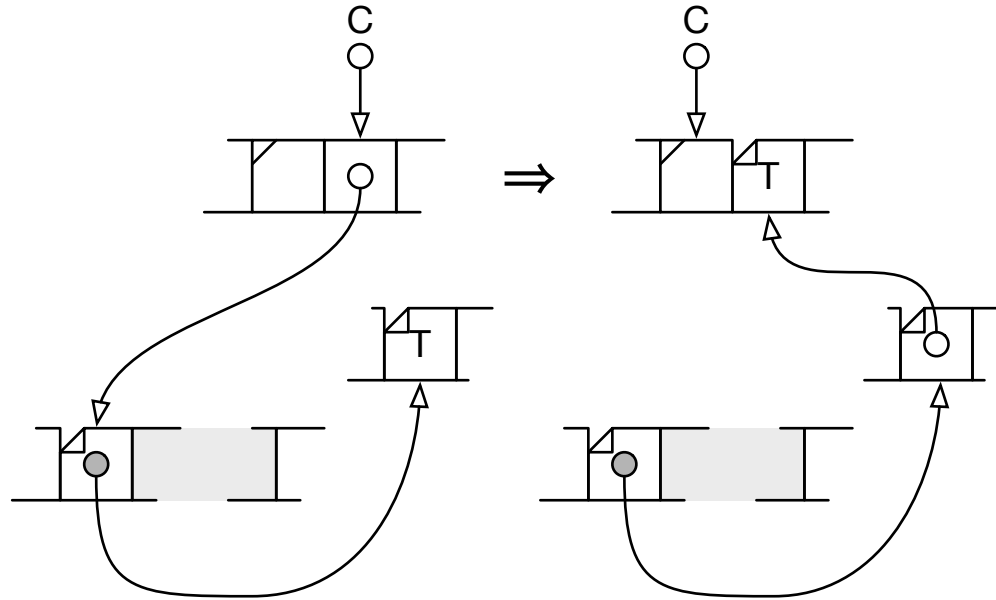


$$\{ [C \uparrow \uparrow, \hbar, m], [C, p, m], C \uparrow + \text{size}(C \uparrow \uparrow) \} \rightarrow \{ *C, *C \uparrow, C \}$$



$$(C \downarrow = u) \wedge (C \uparrow \neq \omega) \wedge \text{chunk?}(C \uparrow) \wedge (C \uparrow \downarrow = m) \wedge \text{chunk?}(C)$$

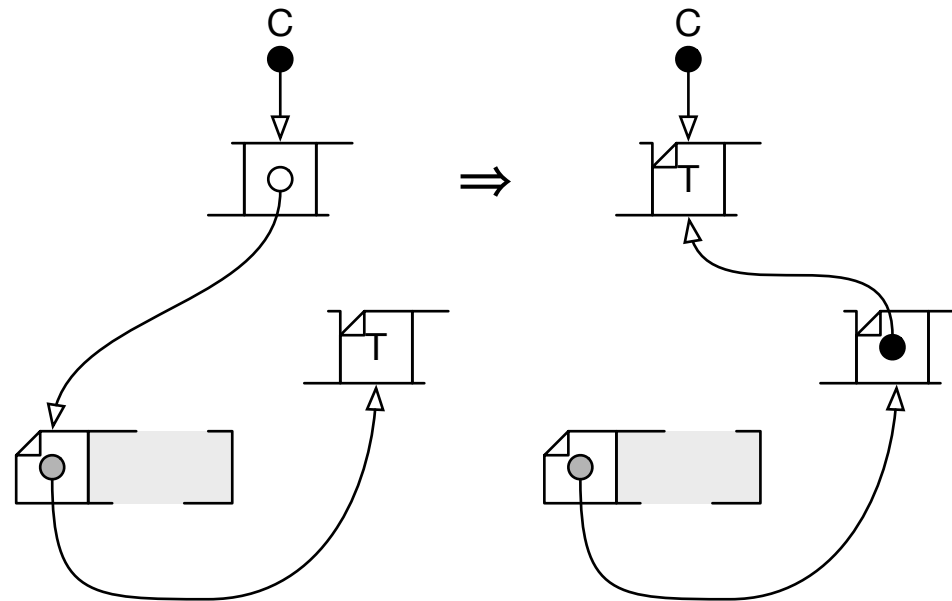
P



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

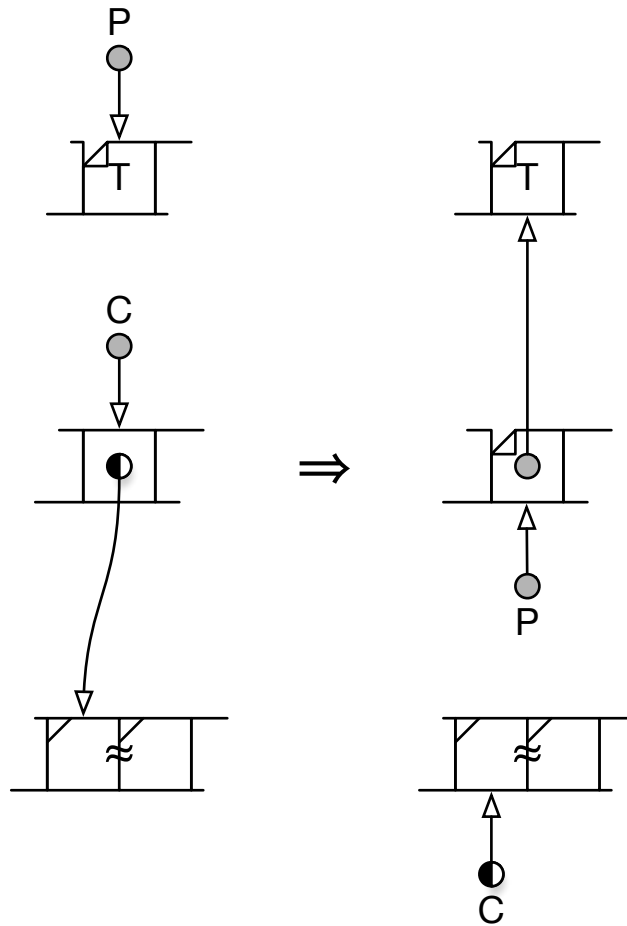
$$(C\downarrow = u) \wedge (C\uparrow \neq a) \wedge \text{chunk?}(C\uparrow) \wedge (C\uparrow\downarrow = m) \wedge \text{pair?}(C)$$

P

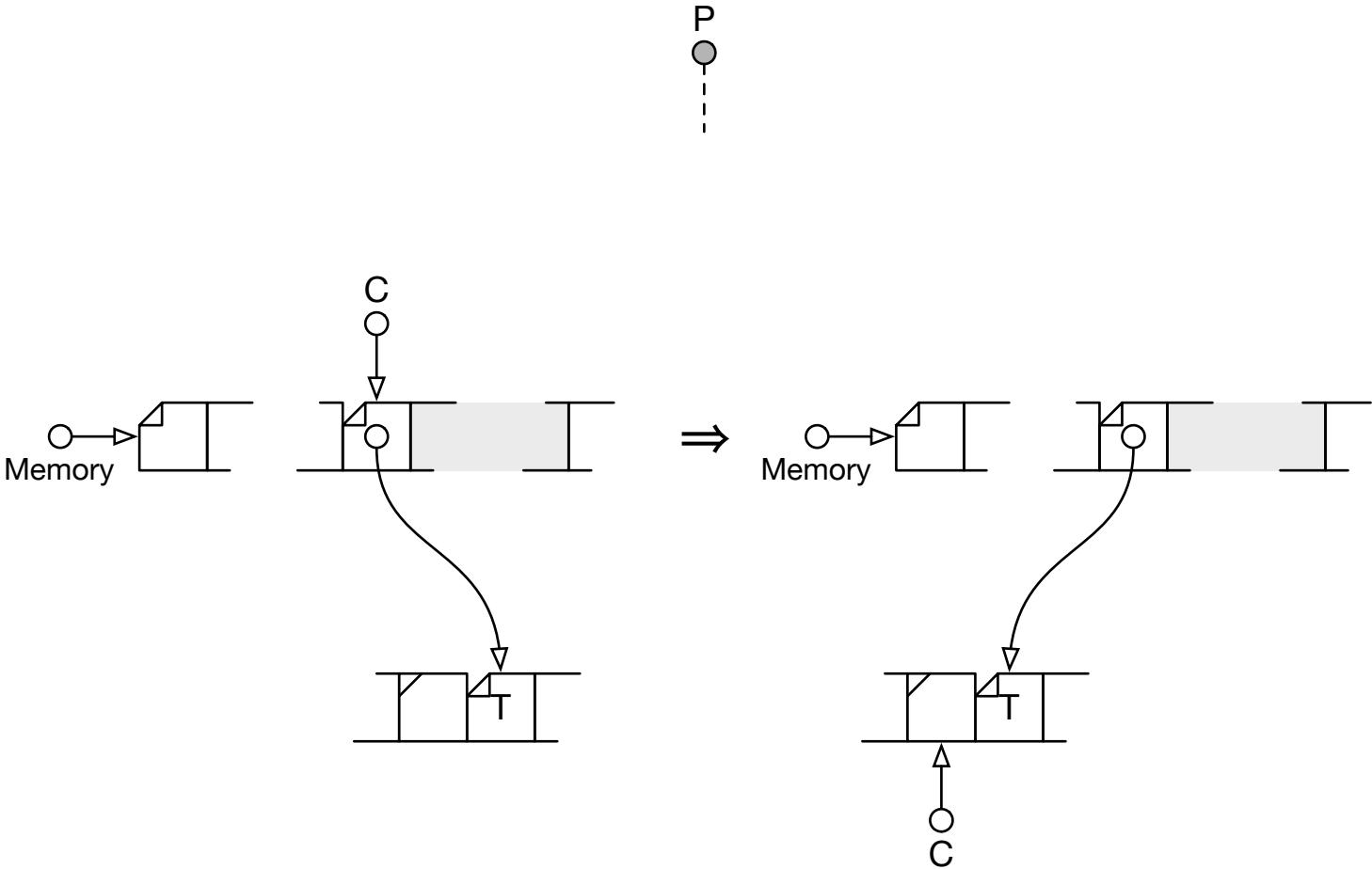
$$\{ *C\uparrow\uparrow, [C, p, m] \} \rightarrow \{ *C, *C\uparrow\uparrow \}$$

$$(C \downarrow = u) \wedge (C \uparrow \neq \omega) \wedge \text{pair?}(C \uparrow)$$



$$\{ [P, p, m], C, C \uparrow \} \rightarrow \{ *C, P, C \}$$

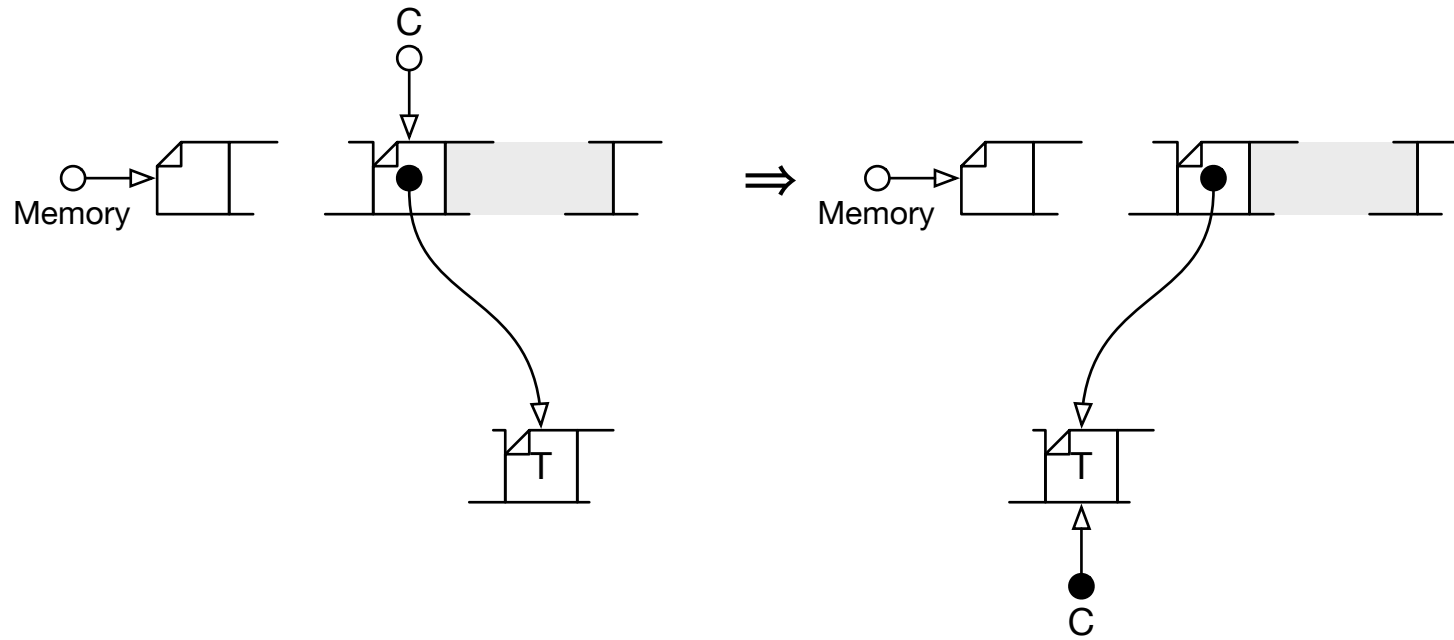
$$(C\downarrow = m) \wedge (C\uparrow \neq \text{Memory}) \wedge \text{chunk?}(C) \wedge \text{chunk?}(C\uparrow)$$



$$\{ C\uparrow - 1 \} \rightarrow \{ C \}$$

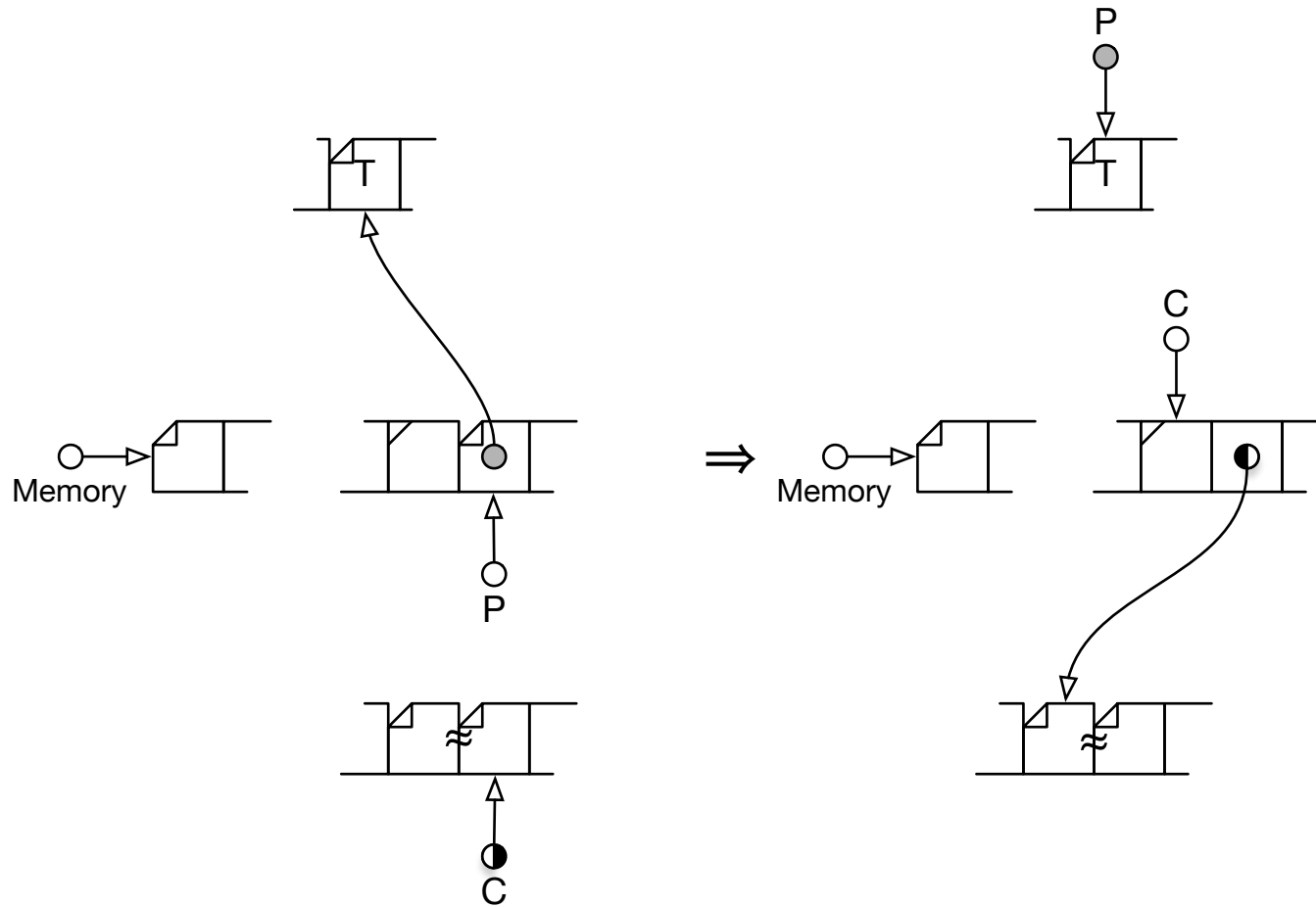
$$(C\downarrow = m) \wedge (C\uparrow \neq \text{Memory}) \wedge \text{chunk?}(C) \wedge \text{pair?}(C\uparrow)$$

P  
●  
⋮



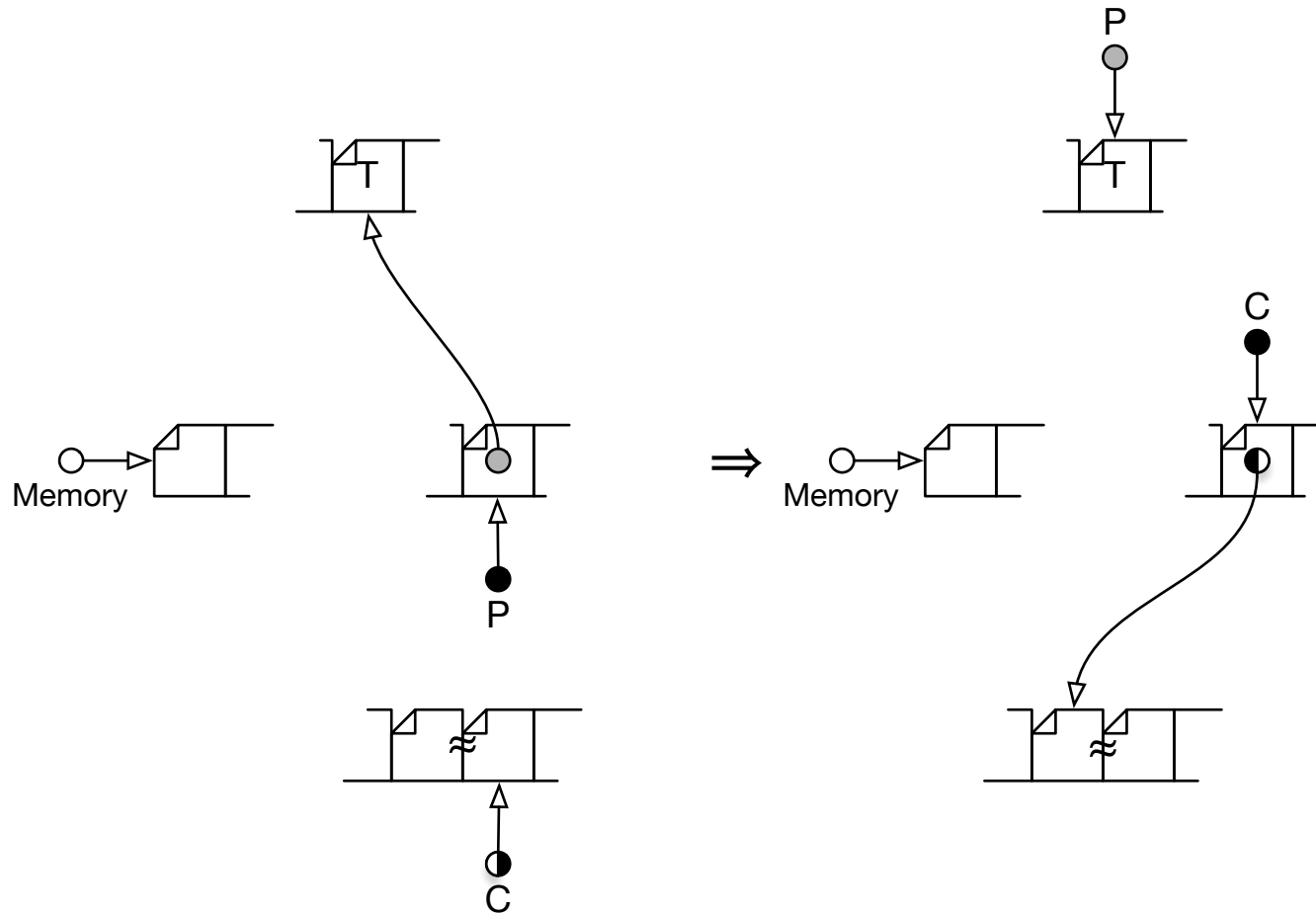
$$\{ C\uparrow \} \rightarrow \{ C \}$$

$$(C \downarrow = m) \wedge (C \uparrow \neq \text{Memory}) \wedge \text{pair?}(C) \wedge \text{right?}(C) \wedge \text{chunk?}(P)$$



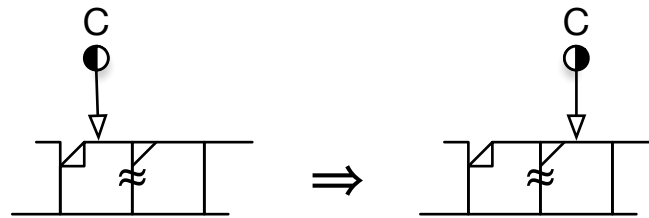
$$\{ [C - 1, \mu, u], P - 1, P \uparrow \} \rightarrow \{ *P, C, P \}$$

$$(C \downarrow = m) \wedge (C \uparrow \neq \text{Memory}) \wedge \text{pair?}(C) \wedge \text{right?}(C) \wedge \text{pair?}(P)$$



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

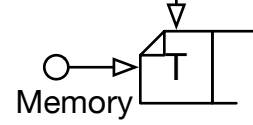
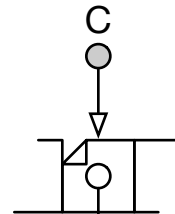
$$(C \downarrow = m) \wedge \text{pair?}(C) \wedge \text{left?}(C)$$



$$\{ C + 1 \} \rightarrow \{ C \}$$



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



```

typedef struct CEL * ptr;
typedef enum { a, h, p } typ;
typedef enum { m, u } mrk;
typedef struct CEL { ptr P; typ T; mrk M; } cel;

```

```

const ptr Null = 0;

```

```

ptr Memory;

```

```

static unsigned is_chunk(ptr), is_right(ptr), is_raw(ptr), size(ptr);

```

```

void Jonkers_Schorr_Waite_mark_thread(cel Root)

```

```

{ ptr C, C_, C__, P, P_;
  *Memory = Root;
  P = Null;
  for (C = Memory;;)
  { C_ = C->P;
    if (C->M == u)
    { if (C->T == a)
      { if (is_chunk(C))
        { C -= 1;
          else
            C->M = m;
        }
      }
      if (is_chunk(C_))
      { C__ = C->P;
        if (C->M == u)
        { *C = (cel){ C__, h, m };
          *C_ = (cel){ C, p, m };
          if (is_raw(C__))
            if (is_chunk(C))
              C -= 1;
            else;
          else
            C = C_ + size(C__); }
        else
        { *C = *C__;
          *C__ = (cel){ C, p, m };
          if (is_chunk(C))
            C -= 1;
          else; }}
      // *Memory <- Root
      // P <- Null
      // C <- Memory
      // C^
      // Cv = u
      // Cw = a
      // chunk?(C)
      // C <- C - 1
      // pair?(C)
      // Cv <- m
      // Cw ≠ a
      // chunk?(C^)
      // C^v = u
      // *C <- [C^, h, m]
      // *C^ = [C, p, m]
      // raw?(C^v)
      // chunk?(C)
      // C <- C - 1
      // pair?(C)
      // regular?(C^v)
      // C <- C^ + size(C^v)
      // C^v = m
      // *C <- *C^
      // *C^ = [C, p, m]
      // chunk?(C)
      // C <- C - 1
      // pair?(C)
    }
    else
    { *C = (cel){ P, p, m };
      P = C;
      C = C_; }
    // pair?(C^)
    // *C <- [P, p, m]
    // P <- C
    // C <- C^
    // Cv = m
    // C^ ≠ Memory
    // chunk?(C)
    // chunk?(C^)
    // C <- C^ - 1
    // pair?(C^)
    // C <- C^
    // pair?(C)
    // right?(C)
    // P^
    // chunk?(P)
    // *P <- [C - 1, p, u]
    // C = P - 1
    // pair?(P)
    // *P <- [C - 1, p, m]
    // C = P
    // P <- P^
    // left?(C)
    // C <- C + 1
    // C^ = Memory
    // stop
  }
  else
  { if (C_ != Memory)
    { if (is_chunk(C))
      { if (is_chunk(C_))
        { C = C_ - 1;
          else
            C = C_;
          else
            if (is_right(C))
            { P_ = P->P;
              if (is_chunk(P))
              { *P = (cel){ C - 1, p, u };
                C = P - 1; }
              else
              { *P = (cel){ C - 1, p, m };
                C = P; }
              P = P_; }
            else
              C += 1;
          else
            break; }}
    }
  }
}

```