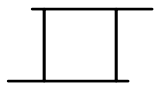
Jonkers-Schorr-Waite compact



cell

chunk

unmarked cell

marked cell

any cell

header

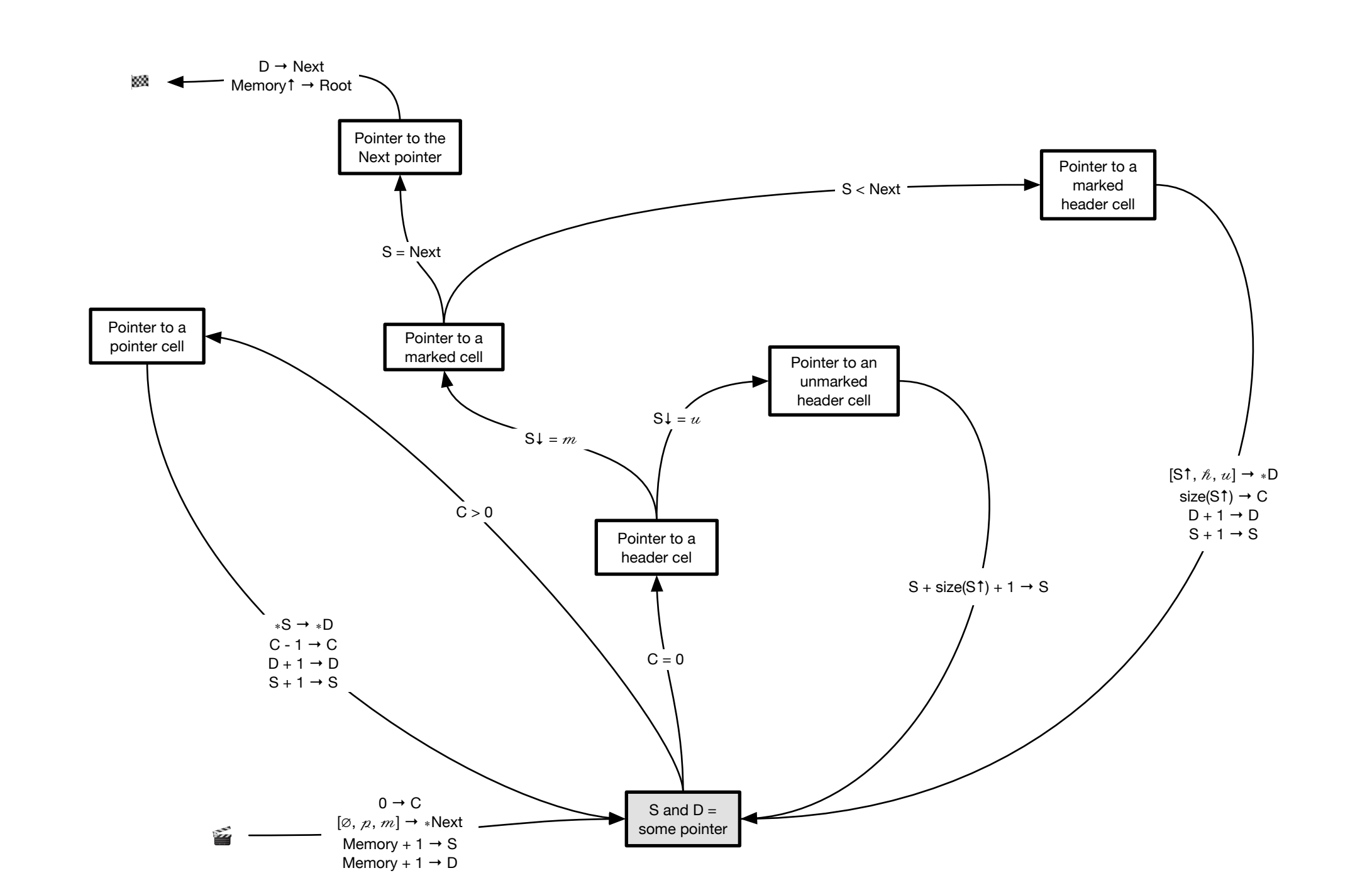
pointer into cell storage

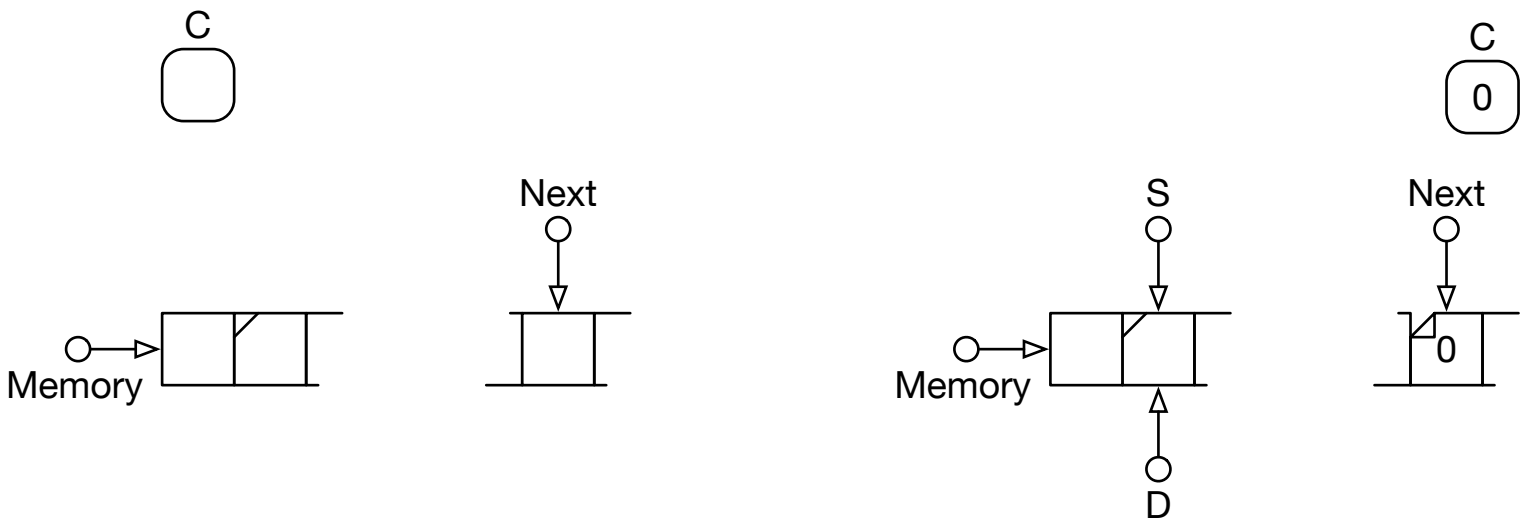
chunks, pairs $\subset \mathbb{N}$
chunks $\cap$ pairs = $\varnothing$
pointers = chunks $\cup$ pairs
types = { a(tom), h(eader), p(ointer) }
marks = { m(arked), u(nmarked) }
cells = pointers × types × markers
$*$: pointers $\longleftrightarrow$ cells : p $\longleftrightarrow$ $[\pi, \tau, \mu]$
$\uparrow$: pointers $\longrightarrow$ pointers : p $\longmapsto$ p$\uparrow$ $\equiv$ $*p_{\pi}$
$\updownarrow$: pointers $\longrightarrow$ pointers : p $\longmapsto$ p$\updownarrow$ $\equiv$ $*p_{\tau}$
$\downarrow$: pointers $\longrightarrow$ markers : p $\longmapsto$ p$\downarrow$ $\equiv$ $*p_{\mu}$

chunk? : pointer $\longrightarrow$ boolean
pair? : pointer $\longrightarrow$ boolean
size : pointer $\longrightarrow$ $\mathbb{N}$
stretch : $\mathbb{N}$ $\longrightarrow$ pointers

Memory : memory pointer
Next : next free chunk pointer

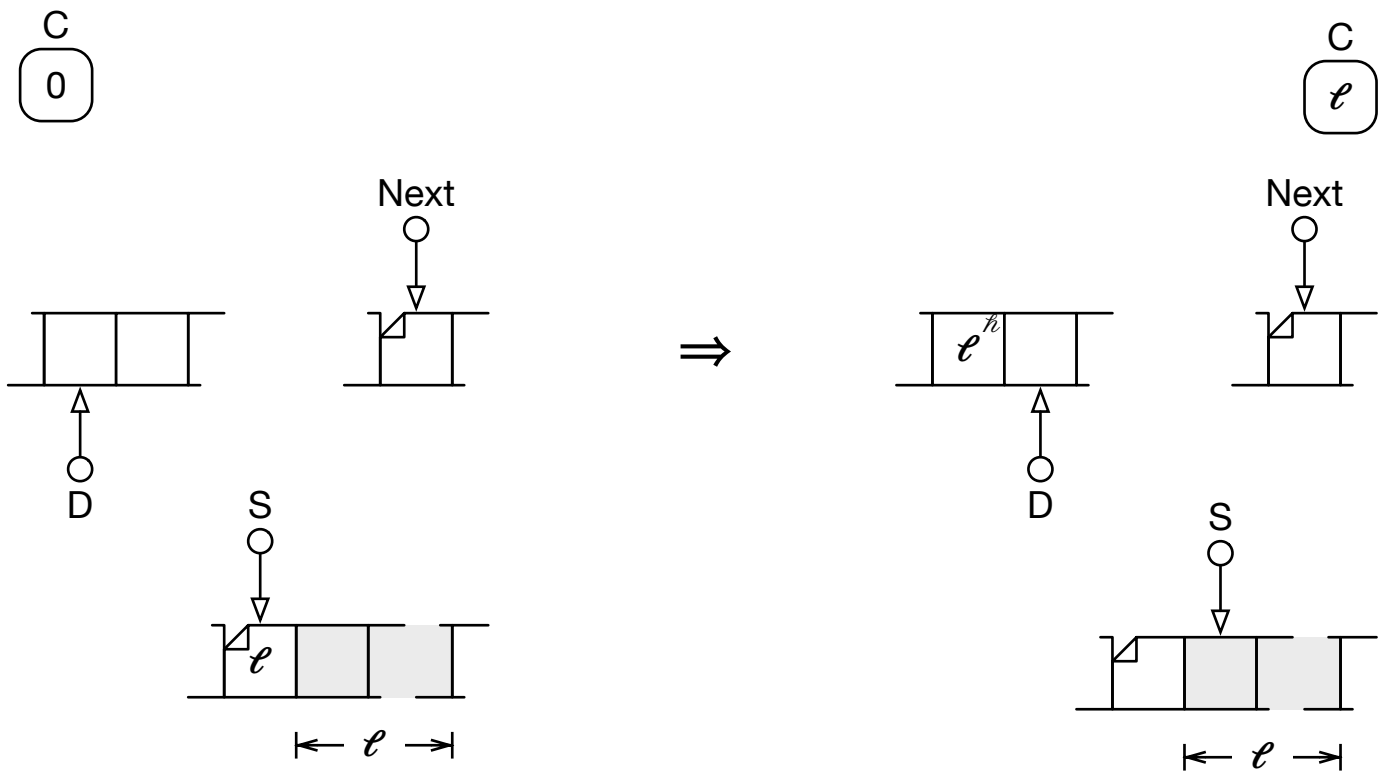
S: source pointer
D: destination pointer
X: any cell
C: counter


D → Next
Memory↑ → Root
Pointer to the Next pointer
S < Next
Pointer to a marked header cell
S = Next
Pointer to a pointer cell
Pointer to a marked cell
Pointer to an unmarked header cell
S↓ = 𝑢
S↓ = 𝑚
[S↑, ℏ, 𝑢] → ∗D
size(S↑) → C
D + 1 → D
S + 1 → S
C > 0
Pointer to a header cel
S + size(S↑) + 1 → S
∗S → ∗D
C - 1 → C
D + 1 → D
S + 1 → S
C = 0
0 → C
[∅, 𝑝, 𝑚] → ∗Next
Memory + 1 → S
Memory + 1 → D
S and D = some pointer



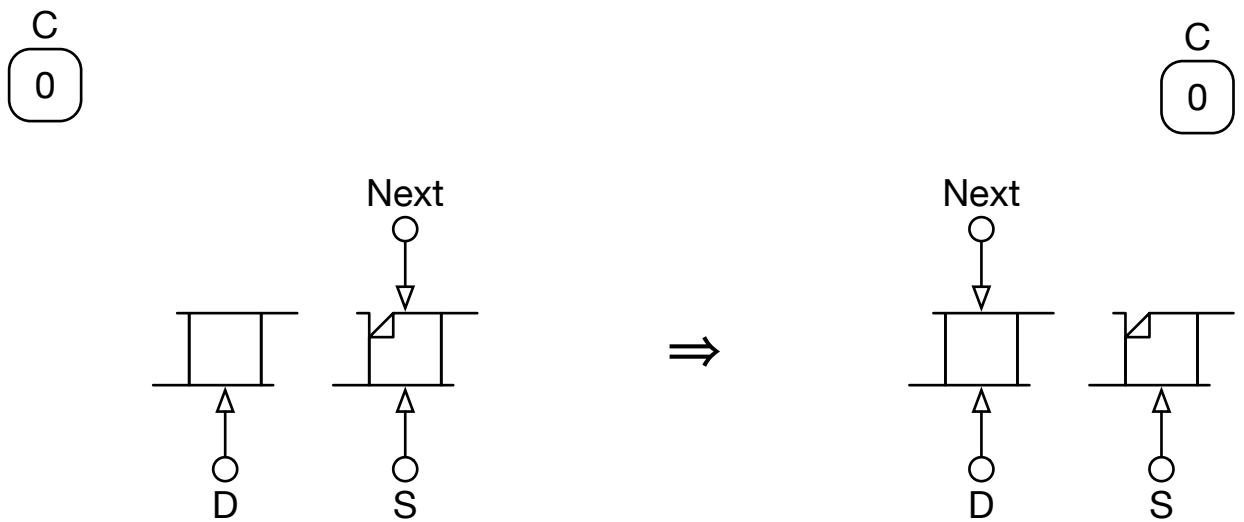
{ 0 , [0, $\hbar$, m] , Memory + 1 , Memory + 1 } → { C , ∗Next , S , D }

(C = 0) ∧ (S↓ = m) ∧ (S < Next)



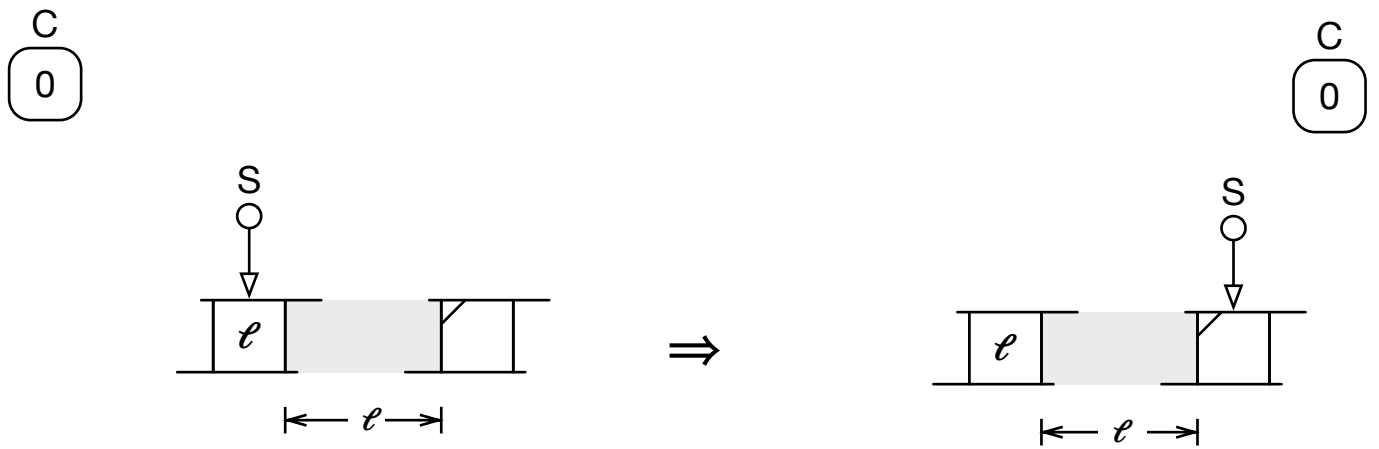
{ [S↑, $\hbar$, u] , size(S↑) , D + 1 , S + 1 } → { ∗D , C , D , S }

(C = 0) ∧ (S↓ = m) ∧ (S = Next)



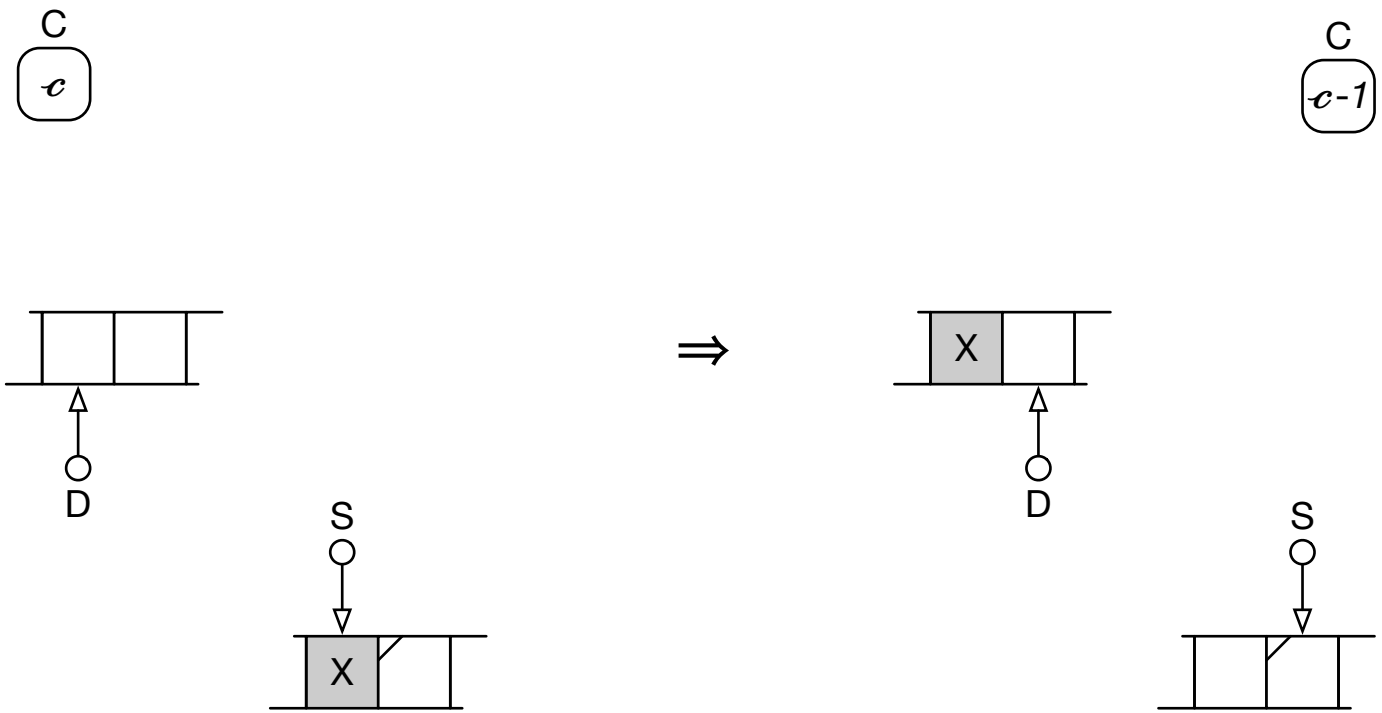
{ D } → { Next }

$(C = 0) \wedge (S{\downarrow} = u)$



$\{\ S + \text{size}(S{\uparrow}) + 1\ \} \rightarrow \{\ S\ \}$

C > 0



{ ∗S , C - 1, D + 1 , S + 1 } → { ∗D , C , D , S }

```
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;

ptr Memory, Next, Null;

unsigned size(ptr);

void Jonkers_Schorr_Waite_compact(void)
  { ptr D, S, S_;
    unsigned C;
    C = 0;                                  // C <- 0
    *Next = (cel){ Null, h, m };            // *Next = [Null, h, m]
    for (S = D = Memory + 1;;)              // S <- D <- Memory + 1
      { S_ = S->P;                          // S^
        if (C == 0)                         // C = 0
          if (S->M == m)                    // Sv = m
            if (S < Next)                   // S < Next
              { *D = (cel){ S_, h, u };     // *D <- [S^, h, u]
                C = size(S_);               // C <- size(S^)
                D += 1;                     // D <- D + 1
                S += 1; }                   // S <- S + 1
            else                            // S = Next
              { Next = D;                   // free <- D
                break; }                    // stop
          else                              // Sv = u
            S += size(S_) + 1;              // S <- S + size(S^) + 1
         else                               // C > 0
            { *D = *S;                      // *D <- *S
              C -= 1;                       // C <- C - 1
              D += 1;                       // D <- D + 1
              S += 1; }}}                   // S <- S + 1
```