```
    TaskInfo take() {
1:    h := head;
2:    t := tail;
3:    if (h = t) return EMPTY;
4:    task := tasks.array[h%tasks.size];
5:    head := h+1;
6:    return task;
    }

    TaskInfo steal() {
      Order read in 1 before read in 2
      Order read in 1 before read in 4
      Order read in 5 before CAS in 6
1:    h := head;
2:    t := tail;
3:    if (h = t) return EMPTY;
4:    a := tasks;
5:    task := a.array[h%a.size];
6:    if !CAS(head,h,h+1) goto 1;
7:    return task;
    }

    void expand() {
      Order writes in 2 and 4 before write in 5
      Order write in 5 before write in put:5
1:    size := tasks.size;
2:    a := new TaskArrayWithSize(2*size);
3:    for i = head:tail−1,
4:       a.array[i%a.size] := tasks.array[i%tasks.size];
5:    tasks := a;
}
```