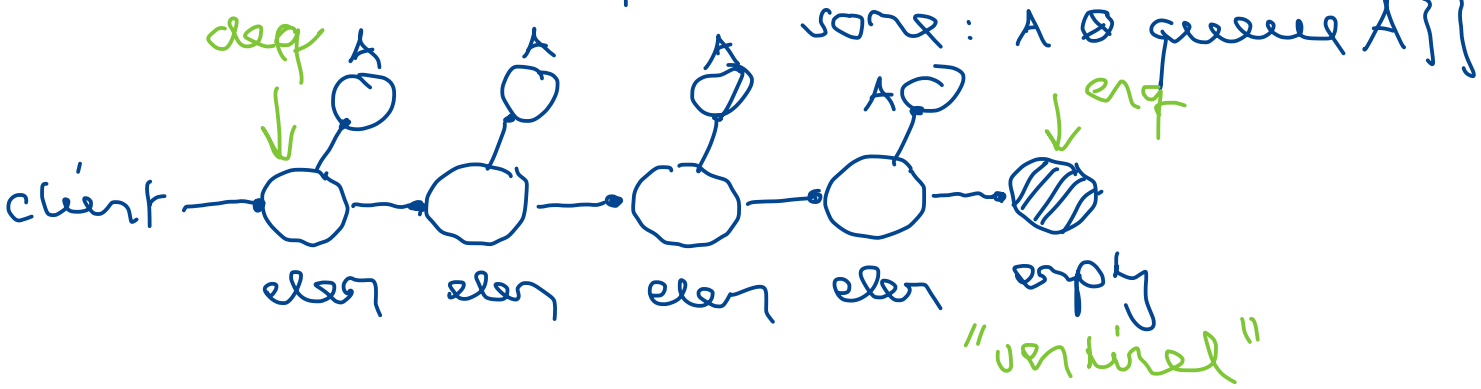
Linear Queue:

queue A = &{enq : A ⊸ queue A,
deq : ⊕{none : 1,
some : A ⊗ queue A}}



• ⊢ empty :: (q : queue A)
q ← empty = // q : queue A
case q of // q : &{...}
| enq → x ← recv q; // x : A ⊢ q : queue A
e ← empty;
q ← elem ← e, x;

| deq → q.none;
case q

```
queue A = &{enq: A ⊸ queue A,
            deq: ⊕{none: 1,
                   some: A ⊗ queue A}}

t: queue A, x: A ⊢ elem :: (q: queue A)
q ← elem ← t, x =
    case q of
      | enq → y ← recv q;
              t.enq; send t y;
              q ← elem ← x, t;

      | deq → q.some;
              send q x;
              fwd q t;
```