Treiber's stack

[Treiber, 1986]

```
struct node {
    struct node *next;
    value data;
};
struct stack { struct node *Top; };
struct stack *S;

void init() {
    S = alloc();
    S->Top = NULL;
}

value pop() {
  struct node *t, *x;
  do {
    t = S->Top;
    if (t == NULL) return EMPTY;
    x = t->next;
  } while (!CAS(&S->Top,t,x));
  return t->data;
}

void push(value v) {
  struct node *t, *x;
  x = alloc();
  x->data = v;
  do {
      t = S->Top;
      x->next = t;
  } while (!CAS(&S->Top,t,x));
}
```

Typical lock-free structure

If the CAS fails
(thread might starve)

then another thread succeeded
(global progress)

A proof of lock-freedom or fair termination needs to catpure that fact.

Treiber's stack simplified

```
    void op() {
        do {
Op1:        t = S->Top;
            x = ... ;
Op2:    } while (!CAS(&S->Top,t,x));
    }
```