course: CSC 135-01 - Computing Theory and Programming Languages

instructor: Ted Krovetz

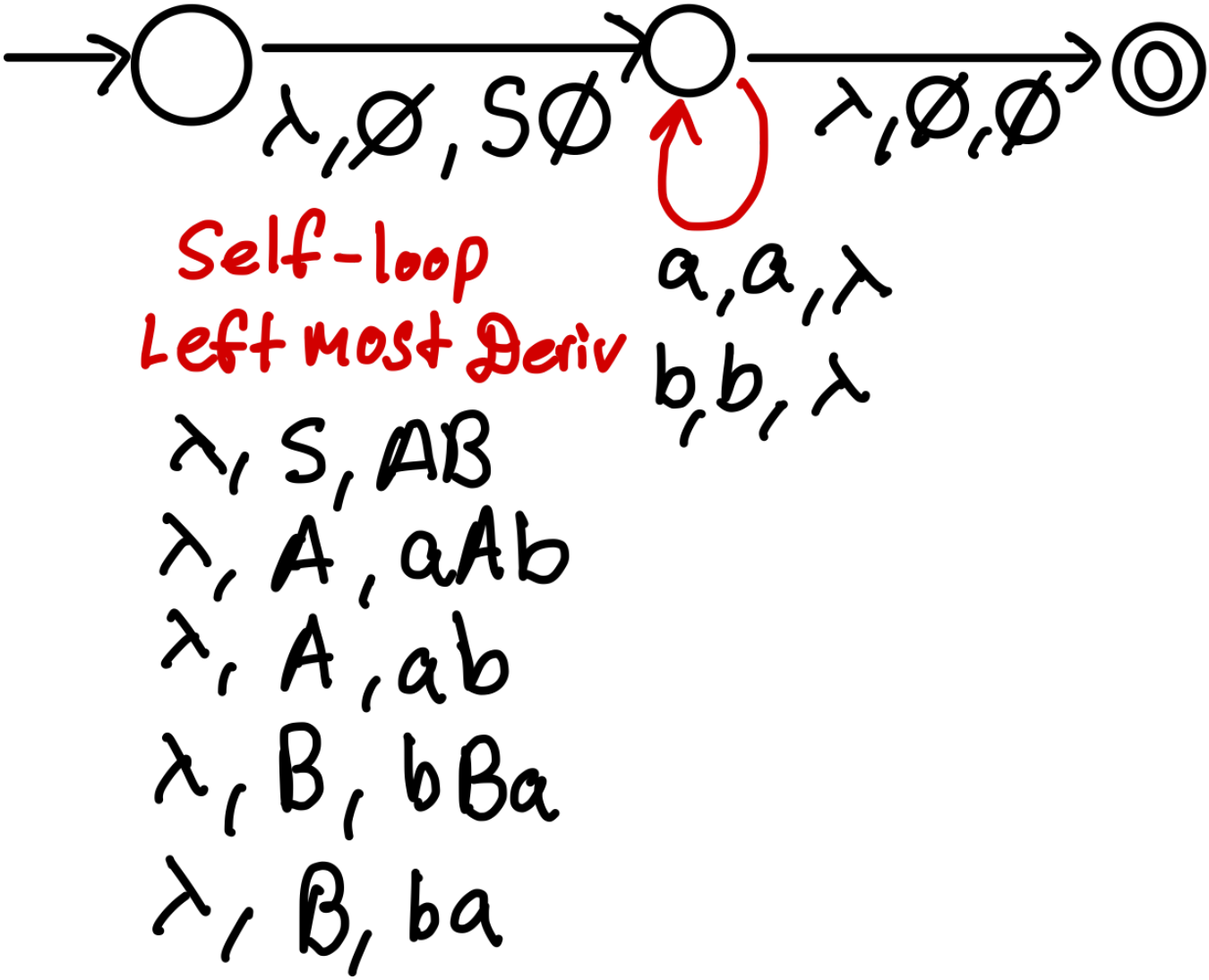
related notes: 2022-05-10

Review

W19.2 | Tuesday, May 10, 2022 | 09:02 AM

Context Free Grammar (CFG)

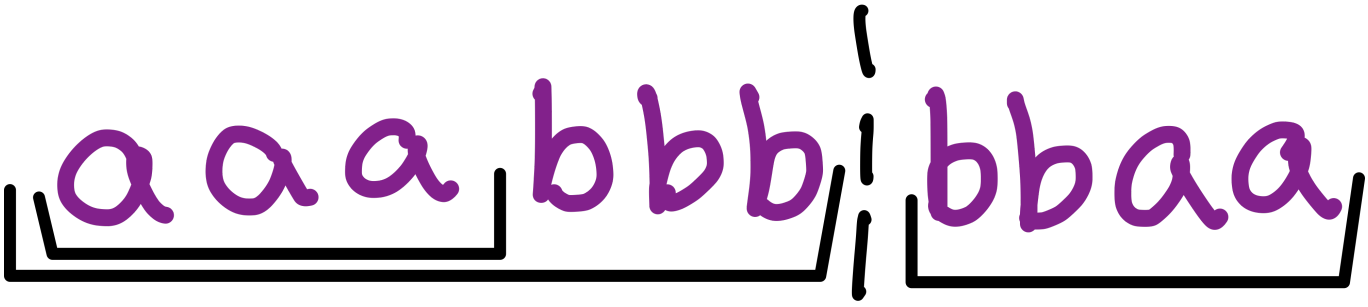
> **Info**
>
> $$S \rightarrow AB$$
> $$A \rightarrow aAB \mid ab$$
> $$B \rightarrow bBa \mid ba$$



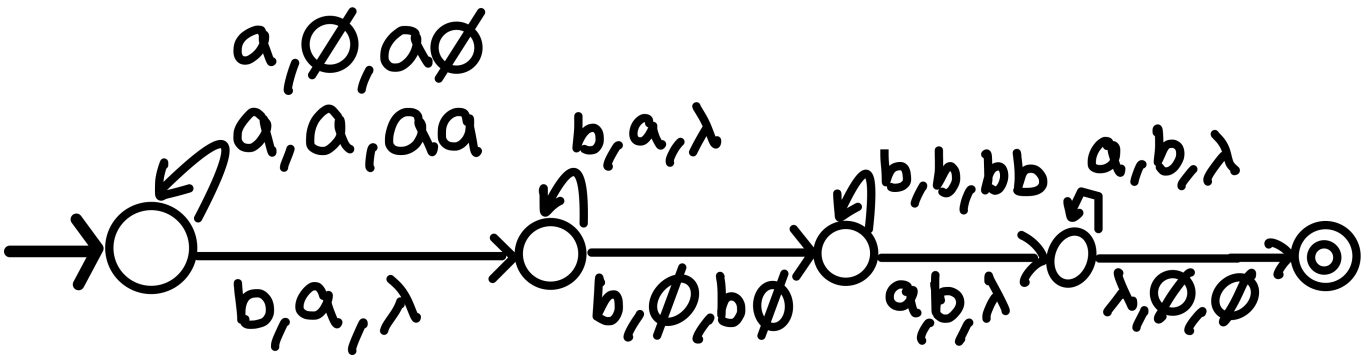
CFG to Pushdown Automata

> **Info**
>
> $\{a^i\, b^j\, a^k \mid i + k = j\}$

If you had a stack how would you fit a string in language



1. Consume and Push a's
2. Consume b's and Pop a's
3. Consume and Push b's
4. Consume a's and Push b's



Is Suitable For Recursion?

1. No left recursion
2. Not ambiguous
3. No prediction table conflicts
    1. Fails because both have first of a
        1. $A \rightarrow aAB \mid ab$
        2. $B \rightarrow bBa \mid ba$

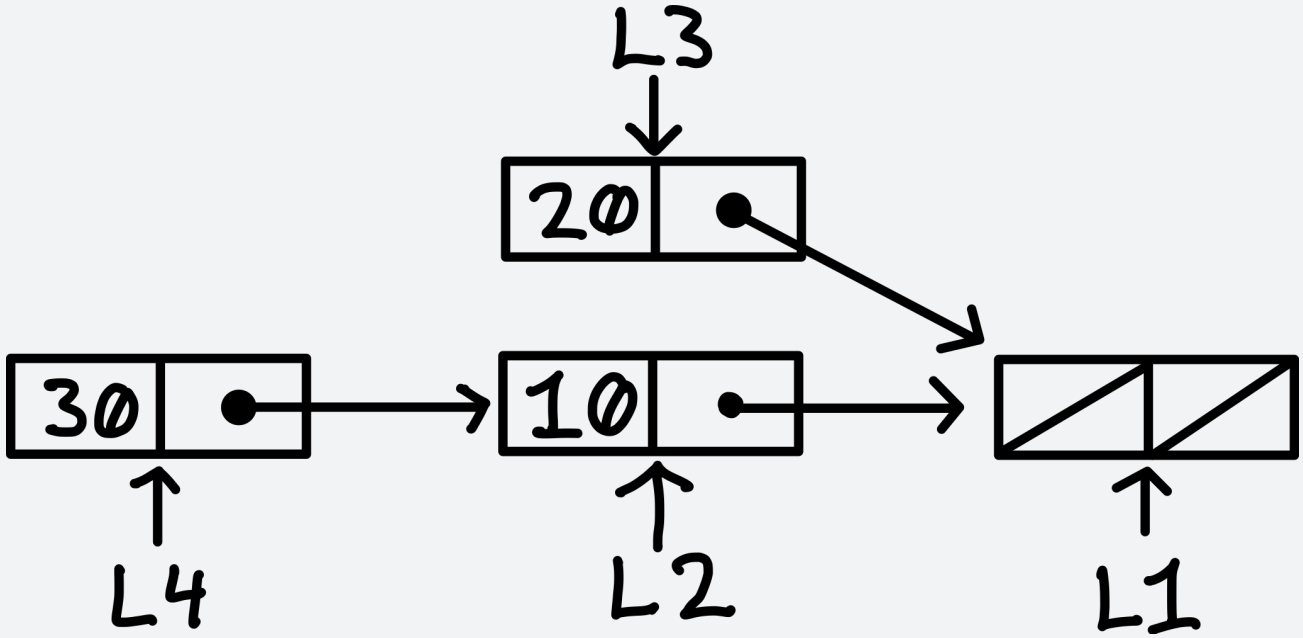
Persistent Structures (`list135`) - Everything is $O(1)$

Code

```
L1 = list135()
L2 = L1.cons(10)
L3 = L1.cons(20)
L4 = L2.cons(30)
```

L.is_empty: L.next == None
L.first(): L.first_element

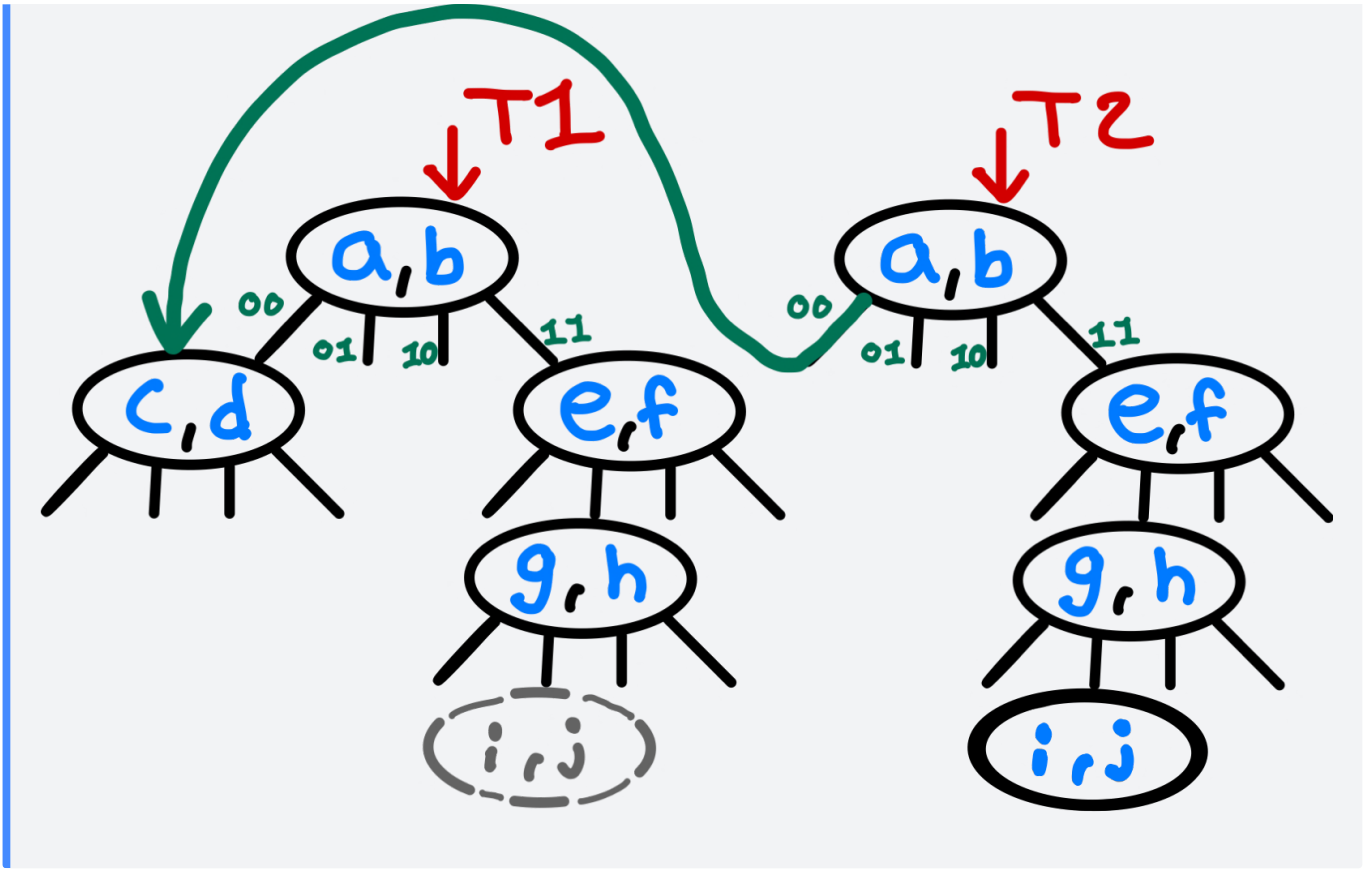
L.rest(): L.rest_of_list



Hash Array Mapped Trie (HAMT)

Code

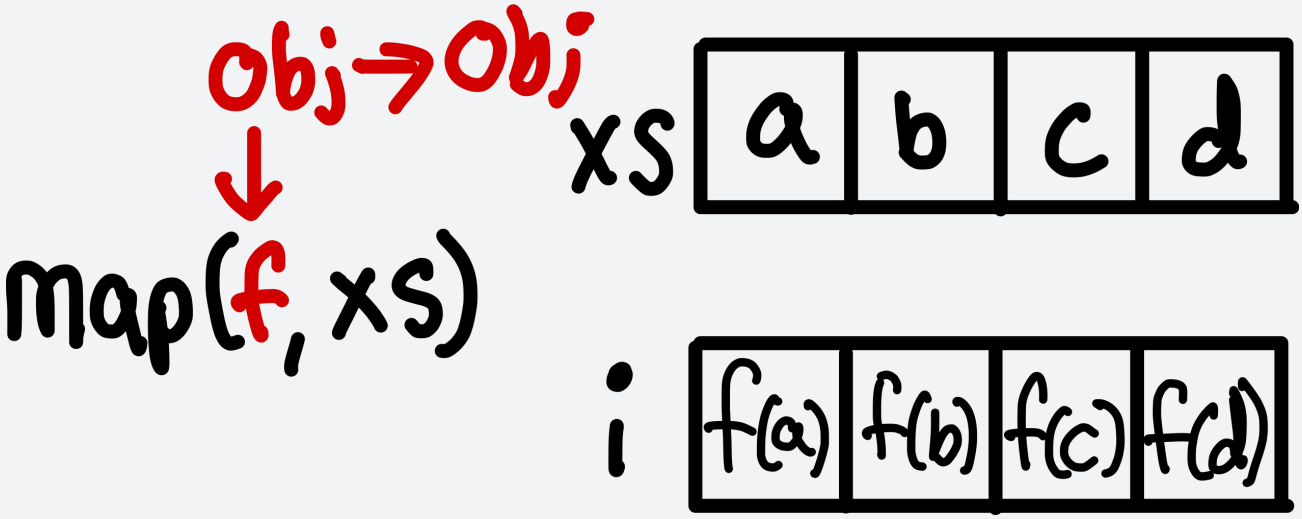
```
T1 = hamt(a, b)
T2 = T1.set(i, j)
hash(i) = ... 010111
```



Higher Order Functions: `map`, `filter`, `reduce`; `lambda`

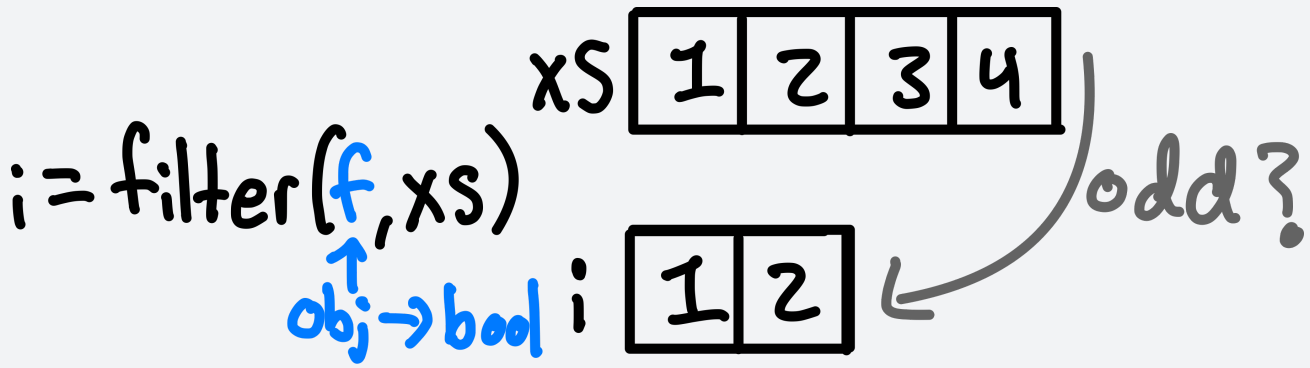
map()

```
i = map(f, xs)
L = list(i)
```



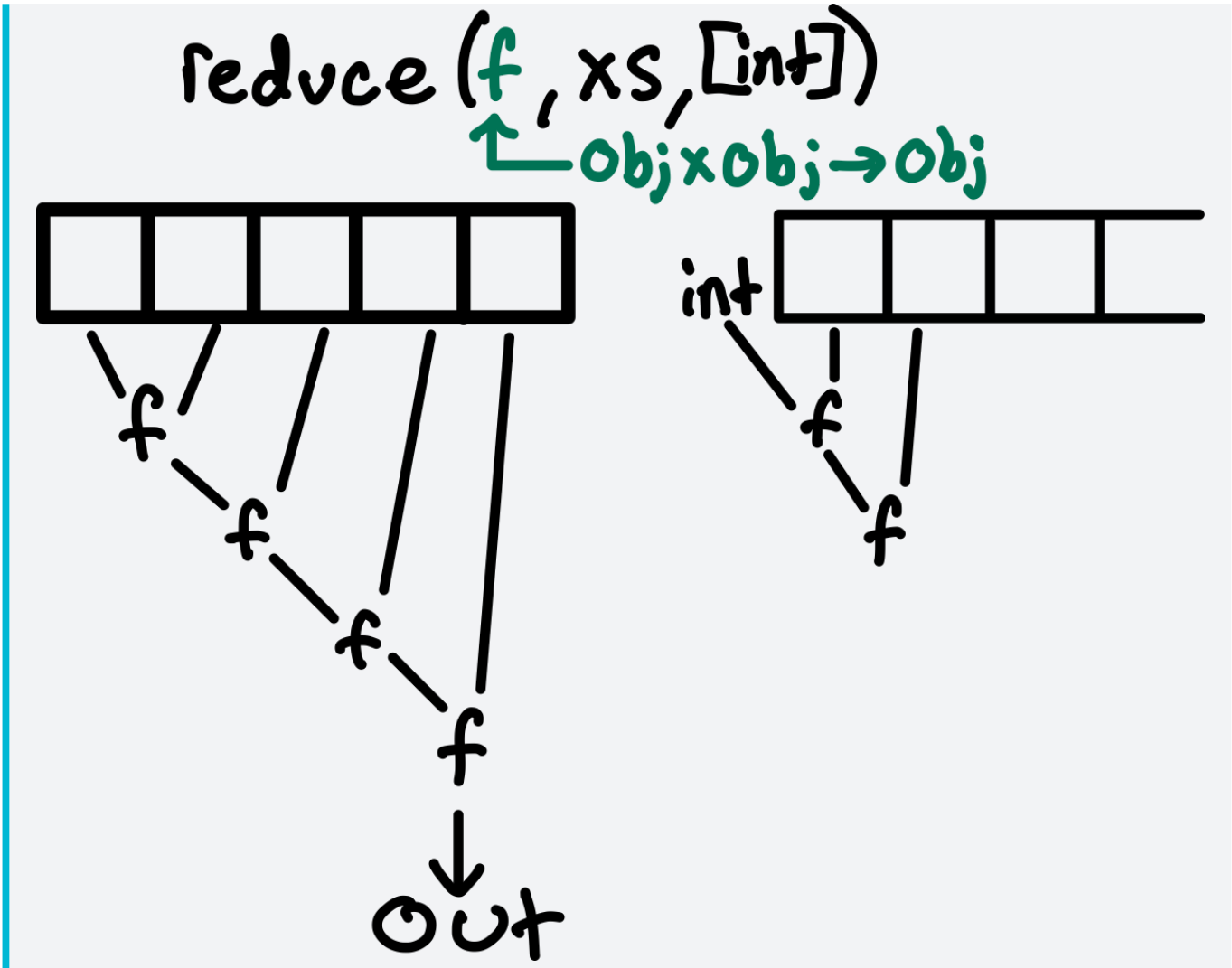
filter()

```
i = filter(f, xs)
```



reduce()

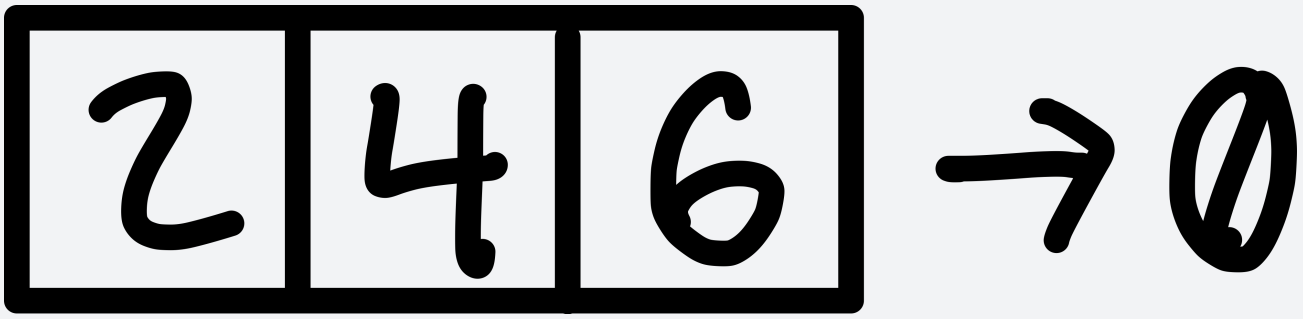
```
from functools import reduce
r= reduce(f, xs, [int])
```



Higher Order Functions Challenge

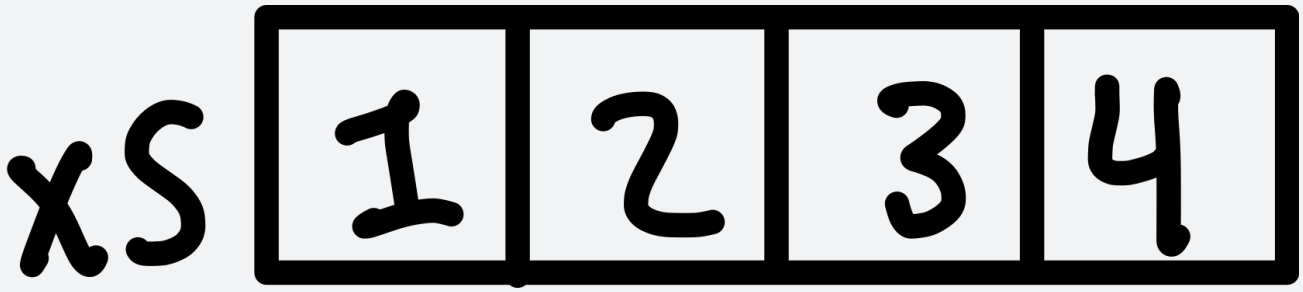
Question

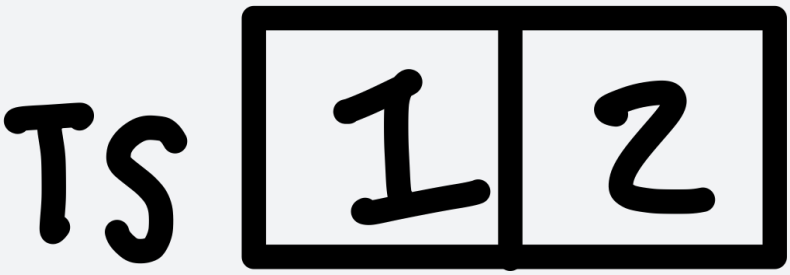
Given list of non-negative integers
return largest odd one or zero if none.



Idea

Filter to keep odds, reduce to find max.






```
TS = filter(lambda x: x % 2 != 0, XS)

def is_odd(x):
        return x % 2 != 0

# Is the same as the anonymous function

lambda: x % 2 != 0
```

Python Turnery Statement: `x if bool else y`

- Simular to C/C++ Turnery: (bool ? x : y)

```
# Solution
res = reduce(lambda a, b: a if a > b and a % 2 != 0 else b, TS, 0)`
```