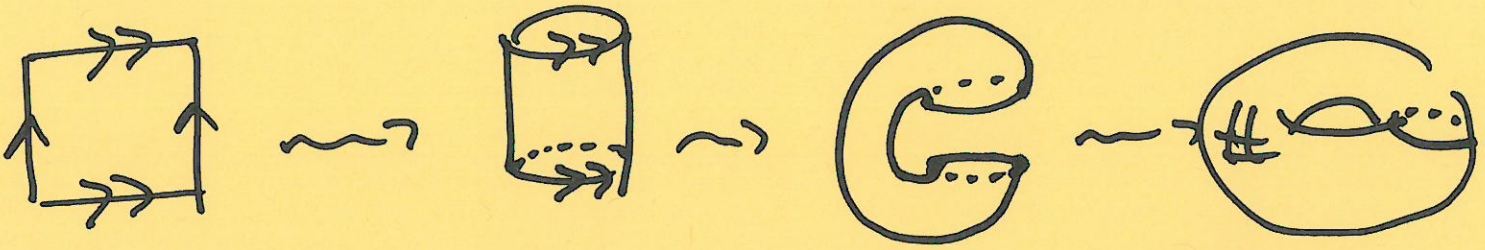
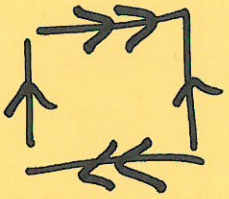
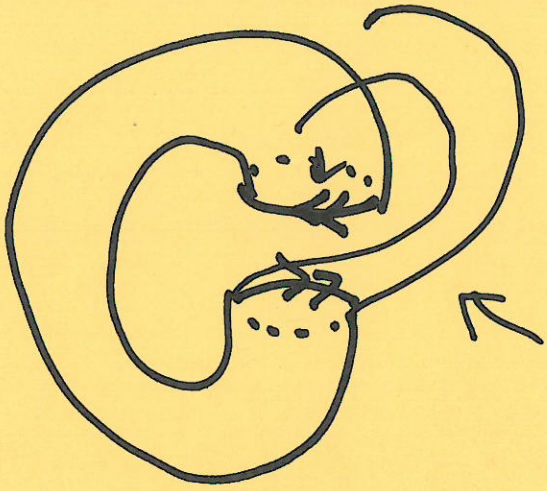
20 Sept 2019



TORUS
Coffee cup
donut





↖ can't be embedded in $\mathbb{R}^3$

((P = pre-condition = what is true before the loop))

for/while G ← loop guard

| ((Li = loop invariant))

| do something

end loop

((Q = post condition))

(what has this loop accomplished)

↳ usu: the strongest thing that can be said

P, Li, Q could be assertions in the code.

P, G, Li, Q ← are all true/false statements, ie. things that can be asserted.

Hints on Loop Invariants:

1. Start w/ post-condition.
2. Think about L_i ... what helps me get "partially" there
3. L_i might "break" mid-loop, but that's ok as long as it becomes true again

```
1: while G
2:     Li must be true
3:     do stuff
4:     Li + Li+1 may/may not be true
5:     do stuff
6:     Li+1 must be true. (would be
           line 2 of next iter through
           the loop)
```

4. ~~Focus~~ In termination, $\{\neg G \wedge L \Rightarrow Q\}$

$\Rightarrow$ if variable appears in Q, it must appear in G or in L.

input: array A of real numbers, $|A| \geq 2$ and A indexed $1 \ldots n$

```
myfcn (A)
1:  temp ← A[1]
2:  i = 2
3:  while (i ≤ n)        ← loop guard!
4:  |   if A[i] > temp
5:  |   |   temp ← A[i]
6:  |   end if                    i++
7:  end while
8:  return temp
```

P = "$A \geq 2$, i is an integer & temp = A[1]"

Q = "temp is the max value in A" = "temp = max(A)"

$G = \{i \leq n\} \Rightarrow \neg G = \{i > n\}$

L_i = "temp is the max value of A[1]... A[i-1]"

Must Check:

① Initialization: $P \Rightarrow L_0$ or L_1 (whatever i is on first entering)

② Maintence: $G \wedge L_i \Rightarrow L_{i+1}$ (looks like induction!)

③ Termination $\neg G \wedge L_i \Rightarrow Q$

proof

① $P \Rightarrow temp = A[1]$

$\Rightarrow temp = \max\{A[1]\}$

$\Rightarrow L_1$ ✓

② $G \wedge L_i \Rightarrow "i \leq n" + L_i$

$\Rightarrow "i \leq n" + "temp = \max_{j=1\ldots i-1}\{A[j]\}"$

Consider 2 cases:

① $temp \geq A[i]$

↳ no need to update temp

② $temp < A[i]$ ~~(greater)~~ (less than)

✓ update temp

So, temp is updated if needed

$\Rightarrow L_{i+1}$.