

CS 4110 - 10/3/25

Lab #17 - Separation Logic

Recap

Last time we extended IMP with commands for manipulating data on the heap.

$C ::= \dots$

$x := \text{new}(e)$	// allocate
$\text{free}(e)$	// deallocate
$x := *e$	// load
$*e_1 := e_2$	// store

We also defined a big-step operational semantics:

$$\langle \sigma, h, c \rangle \Downarrow (\sigma', h')$$

Finally, we defined heap assertions

$H ::= [P]$

| emp

| $e_1 \hookrightarrow e_2$

| $H_1 * H_2$

| $H_1 \frown H_2$

| $H_1 \vee H_2$

| $\dots$

Defn: Assertions H_1 and H_2 are equivalent, $H_1 \equiv H_2$, if they are either both valid, or not valid.

$$H_1 \equiv H_2 \stackrel{\Delta}{=} \models H_1 \Leftrightarrow \models H_2$$

Examples

- $H * J \equiv J * H$
 - $(H * J) * K \equiv H * (J * K)$
 - $H * \text{emp} \equiv H$
 - $[P] * H \neq [P] \wedge H$, but
 $[P] * \text{emp} \equiv [P] \wedge \text{emp}.$
-

Example

What does this program do?

```
t := *p;  
u := *q;  
*p := u;  
*q := t;
```

Answer: swap!

What would a good specification be?

Pre: $\{ p \leftrightarrow a * q \leftrightarrow b \}$

Post: $\{ p \leftrightarrow b * q \leftrightarrow a \}$

Yes!

Is this the only specification? No!

No!

For instance, what if p and q are aliases?
Then the function is the identity!

Pre: $\{ [p = q] * p \leftrightarrow a \}$

Post $\{ [p = q] * p \leftrightarrow a \}$

Example: What does this program do?

$x = p1;$

$r = p1;$

while ($p1 \neq 0$) do (

$x := p1;$

$p1 := *(p1 + 1);$

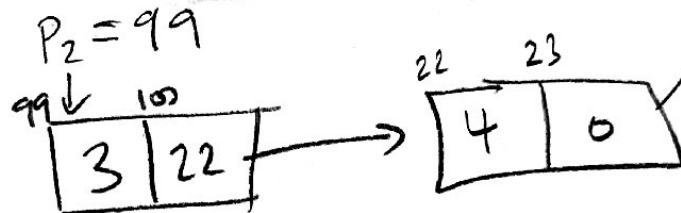
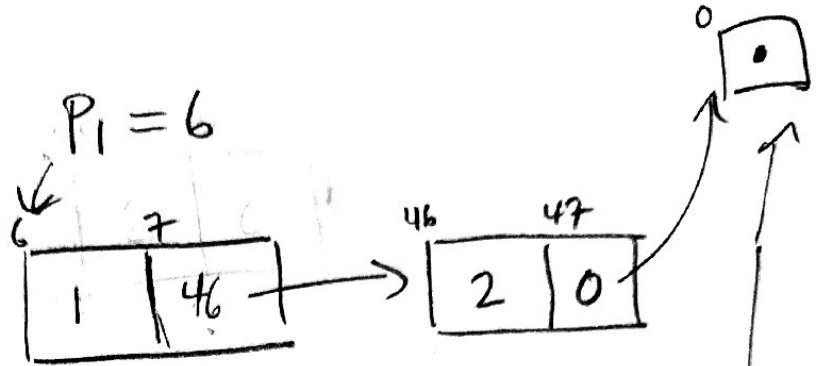
);

$*(x + 1) := p2;$

List append!

Before

6	1
7	46
/ / / / /	
22	4
23	0
/ / / / /	
46	2
47	0
/ / / / /	
99	3
100	22
/ / / / /	



After

6	1
7	46
/ / / / /	
22	4
23	0
/ / / / /	
46	2
47	99
/ / / / /	
99	3
100	22
/ / / / /	

