Free exactly once
void *p = malloc (...);
...
Free (p);
...
free(p);

Write in at most one thread
void f () {
    (*p)++;
}
pthread_create (f);
f ()

Remember: Structural rules

\[ \frac{\Gamma, e : \tau}{\Gamma, \tau, x : \tau' + e : \tau} \text{ weakening} \]
\[ \frac{\Gamma, x : \tau, y : \tau' + e : \tau}{\Gamma, x : \tau, y : \tau' + e : \tau} \text{ Exchange} \]
\[ \frac{\Gamma, x : \tau, x : \tau' + e : \tau}{\Gamma, x : \tau'} + e : \tau} \text{ Contraction} \]

Get rid of weakening + contraction
⇒ Linear type system
⇒ Use variables exactly once
(just contraction ⇒ Affine T.S. ⇒ At most once)
\[ e ::= x \mid (e) \mid \lambda x : t . e \mid e \cdot e \mid \text{let } (x, y) = e \text{ in } e \]

\[ \tau ::= \text{unit } | \tau \to \tau \]

\[ \Delta ; p : \text{ptr} \mid \Gamma ; c : \text{cap} \mid e : \tau \]

\[ \Delta ; \text{ptr} + \text{cap} \mid e : \tau \]

\[ \Delta ; \tau + \text{free } e : () \]

Return to free, data race examples

```
void * p = malloc (...)
Want to be able to use p!
```

Combine linear + non-linear contexts

Idea: separate pointer + capability

- Linear (affine) use to free (write)
- Non-linear use to read/write (read)

\[ \Delta ; e : \tau \]

\[ \Delta ; \text{ptr} + \text{cap} \mid e : \tau \]

\[ \Delta ; \tau + \text{free } e : () \]
let \((p, c) = \text{malloc }\) in
\[p := p + p;\]
\[\text{free } (p, c) := \text{unit}\]
\[\text{free } (p, c) \times\]

**Linear Logic** (By Curry-Howard Correspondence)

Logic where facts can be used exactly once useful for reasoning about resources

Prop: Logic: \(VM \text{ have } \$1.50 \rightarrow \text{ I have a candy bar}\)

\[\frac{\text{VM}, \$1.50 + \$1.50}{\text{VM}, \$1.50 \text{ and } \$1.50 + \text{Candy}}\]
\[\frac{\text{VM}, \$1.50, \$1.50 \text{ and } \$1.50 \text{ and Candy}}{\text{VM}, \$1.50 + \$1.50 \text{ and Candy}}\]

**Oops**

Linear: \(VM = \$1.50 - \text{candy}\)

\[X = \$1.50 \quad \text{ and } \quad \text{needs both } \$1.50 \text{ and VM}\]

\[\frac{\text{VM}, \$1.50 + \$1.50 \text{ and Candy}}{\text{VM}, \$1.50 + \$1.50 \text{ and Candy}}\]

Thanks to Cornell 6110