| Lec 13 | Mutable Store |

L4 : memory constructs (ptrs , arrays, structs)

Recall :
$S ; \eta \vdash e \triangleright K$
$S ; \eta \vdash s \triangleright K$

Ptrs static semantics

$\tau ::= int \mid bool \mid \tau*$
$e ::= alloc (\tau) \mid *e \mid null$

$$\frac{}{\Gamma \vdash alloc(\tau) : \tau*} \qquad \frac{\Gamma \vdash e : \tau*}{\Gamma \vdash *e : \tau} \qquad \frac{}{\Gamma \vdash null : \tau* \; any*}$$

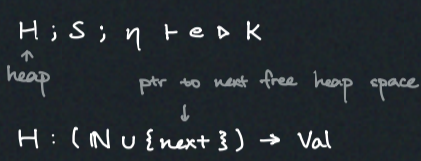↑ problem : can't synthesize type

Polymorphic type $any*$

Ptr comparison : $q == p$ allowed if
 - $p : \tau*$, $q : \tau*$
 - $p : any*$, $q : \tau*$
 - $p : any*$, $q : any*$

Casting $$\frac{\Gamma \vdash e : any*}{\Gamma \vdash e : \tau*} \qquad \frac{\Gamma \vdash e : \tau* \quad \Gamma \nvdash e : any*}{\Gamma \vdash *e : \tau}$$

Ptrs dynamic semantics

Heap in configuration

$$H ; S ; \eta \vdash e \triangleright K$$
↑ heap   ptr to next free heap space
↓
$H : (\mathbb{N} \cup \{ next \}) \to Val$

Evaluation

$H ; S ; \eta \vdash null \triangleright k \to H ; S ; \eta \vdash 0 \triangleright k$

$H ; S ; \eta \vdash alloc (\tau) \triangleright k \to H [a \mapsto default \boxed{1} (\tau) , next \mapsto a + |\tau|] ;$
$\qquad\qquad\qquad\qquad\qquad\qquad S ; \eta \vdash a \triangleright k$ where $a = H(next)$

① default value for all types are 0

Dereferencing

$H ; S ; \eta \vdash *e \triangleright k \to H ; S ; \eta \vdash e \triangleright (*_-, k)$
$H ; S ; \eta \vdash e \triangleright (*_-, k) \to H ; S ; \eta \vdash H(a) \triangleright k$ if $a \neq 0$
$\qquad\qquad\qquad\qquad\qquad$ exception (sth)   if $a = 0$

Assignment
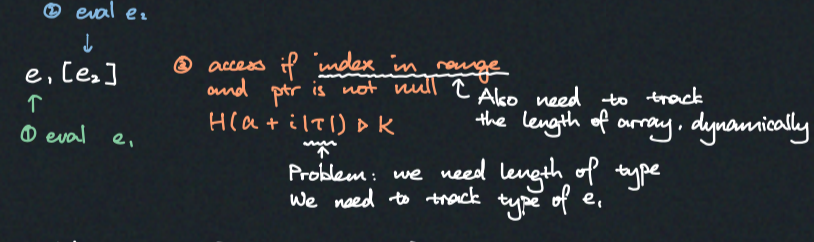
$H ; S ; \eta \vdash assign (*d , e) \triangleright K \to H ; S ; \eta \vdash d \triangleright (assign (*_-, e), k)$
$H ; S ; \eta \vdash a \triangleright (assign (*_-, e), k) \to H ; S ; \eta \vdash e \triangleright (assign (*a, _), k)$
$H ; S ; \eta \vdash c \triangleright (assign (*a, _), k) \to H[a \mapsto c] ; S, \eta \vdash nop \triangleright k$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ if $a \neq 0$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ null dereference otherwise

Array typing
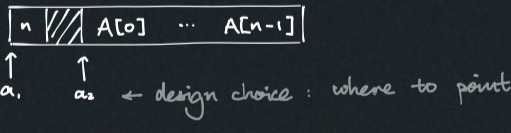
$\tau ::= ... \mid \tau[]$
$e ::= ... \mid alloc\_array (\tau, e) \mid e.[e_2]$
$d ::= ... \mid d[e]$

$$\frac{\Gamma \vdash e_1 : \tau[] \quad \Gamma \vdash e_2 : int}{\Gamma \vdash e_1.[e_2] : \tau} \qquad \frac{\Gamma \vdash e : int}{\Gamma \vdash alloc\_array (\tau, e) : \tau[]}$$

Note array with size 0 is not same as null

   ① eval e_2
   ↓
$e_1.[e_2]$ ③ access if index in range
   ↑   and ptr is not null ↑ Also need to track
   ① eval e_1 $H(a + i|\tau|) \triangleright K$ the length of array, dynamically
   ↑   Problem: we need length of type
   We need to track type of e_1

→ Elaborate $e_1.[e_2] \rightsquigarrow e_1.\{\tau\}[e_2]$ if $e_1 : \tau$
→ keep array header to track size

a_1 a_2 ← design choice : where to point

Cogen ex. $e_1.\{\tau\}[e_2]$ with $|\tau| = k$

   $\rightsquigarrow$ cogen (e_1, a)
      cogen (e_2, i)
      $a_1 \gets a - 8$
      $t_2 \gets M[a_1]$
      if $(i < 0)$ goto error
      if $(i \geq t_2)$ goto error
      $\}$

Allocation
         ① $e \rightsquigarrow n$ ② if $n < 0$, error
         |
         | ③ put address a into continuation
   $alloc\_array (\tau, e) \triangleright k$
      |
      ④ allocate and put things in H

Assignment ② NULL & bound check
      ┌──────┴──────┐
      ①   ③
      | |
   assign ($d\{\tau\}[e_1]$, e_2) $\triangleright k$
      ④   ④
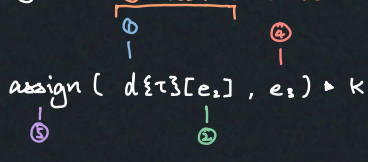
Note : default array needs to be handled differently