Lect 20:

Representation independence

→ Prove that **no** client code can ever tell the difference between

O(n) size ← D and C → O(1) size

D.dict C.dict

are different

types

D.dict

has

Node(l, (k, v), r)

C.dict

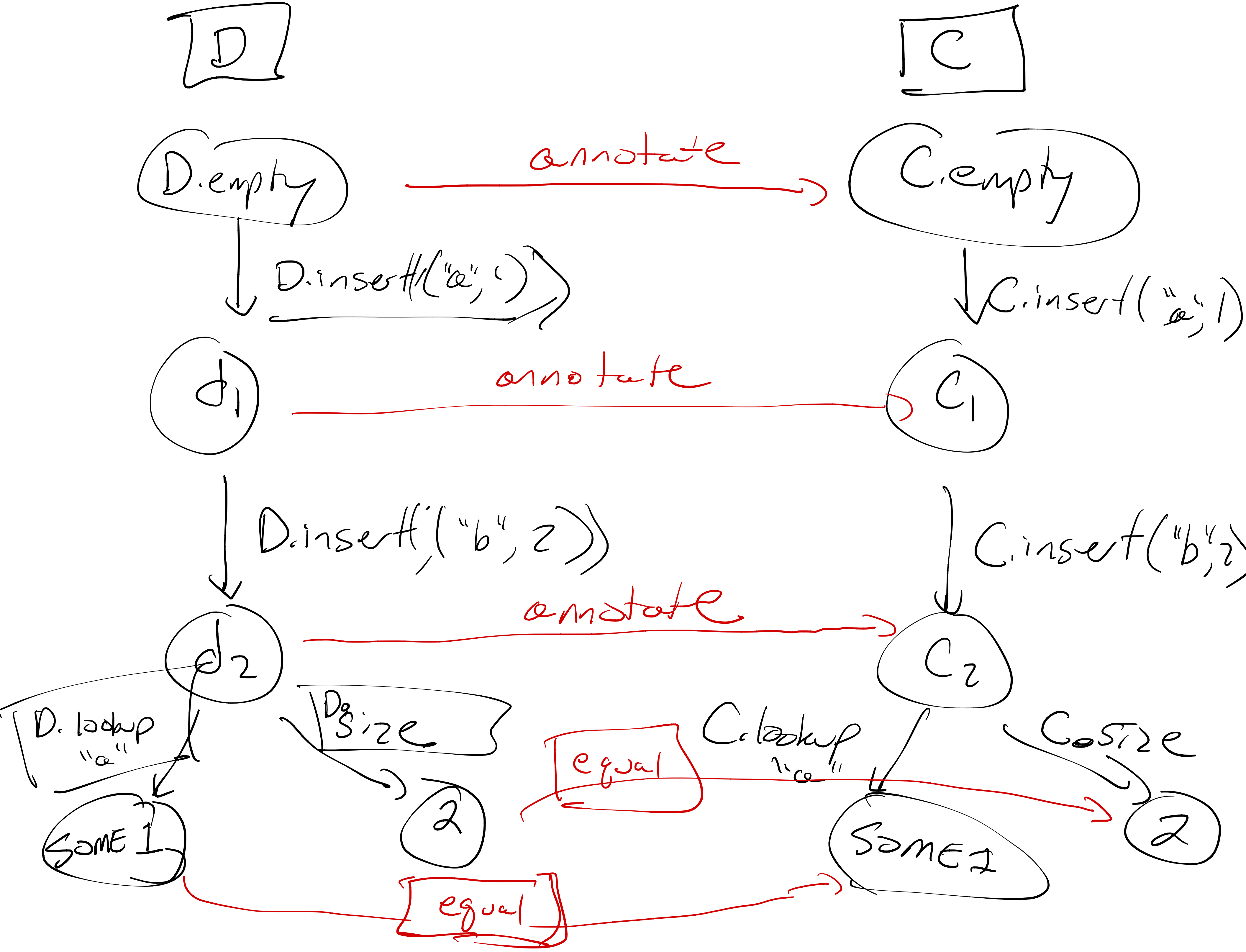
has

Node(l, ((k, v), s), r)

set up a

correspondence/

simulation



D
C
D.empty
annotate
C.empty
D.insert("a", 1)
C.insert("a", 1)
d1
annotate
c1
D.insert("b", 2)
C.insert("b", 2)
d2
annotate
c2
D.lookup "a"
D.size
C.lookup "a"
C.size
equal
Some 1
2
Some 1
2
equal

Have to prove:

- every operation that creates a dictionary preserves the simulation
  - empty
  - insert

  annotate

- when two dictionaries are in simulation, the observations are equal
  - lookup
  - size

Then: D and C behave the same for all client code!

$\boxed{\text{For size:}}$

If $\underset{\text{d}}{d} : (\text{'k}, \text{'v})\ D.\text{dict} \xrightarrow{\text{annotate}} c : (\text{'k}, \text{'v})\ C.\text{dict}$

thus $D.\text{size}(d) = C.\text{size}(c)$

---

for all $d : D.\text{dict}$ $\boxed{\text{Theorem}}$

$$D.\text{size}(d) = C.\text{size}(\underbrace{\text{annotate}}_{\text{ann}}\ d)$$

$\boxed{\text{Size}}$ case for D.Empty = E

$\boxed{\text{TS}}$ D.size (D.E) = C.Size (arr D.E)

$\mapsto$ 0 $\qquad\qquad$ $\mapsto$ C.size(C.E)

$\qquad\qquad\qquad\qquad$ $\mapsto$ 0

$\boxed{\text{Size}}$ case for $D.N(l, \overset{(k,v)}{x}, r)$:

$\boxed{\text{To show:}}$

$$D.\mathrm{size}(D.N(l, x, r)) = C.\mathrm{size}(\mathrm{ann}(D.N(l, x, r)))$$

$$\mapsto 1 + D.\mathrm{size}(l) + D.\mathrm{size}(r)$$

$$=_{IH} 1 + C.\mathrm{size}(\mathrm{ann}\ l) + C.\mathrm{size}(\mathrm{ann}\ r)$$

C.size t
ann t
valuable

RHS

$$\mapsto C.\mathrm{size}(C.\mathrm{Node}'(\mathrm{ann}\ l, x, \mathrm{ann}\ r))$$

$$\mapsto C.\mathrm{size}\left(C.N\left(\mathrm{ann}\ l, \left(x, 1 + C.\mathrm{size}(\mathrm{ann}\ l) + C.\mathrm{size}(\mathrm{ann}\ r)\right), \mathrm{ann}\ r\right)\right)$$

$$\mapsto 1 + C.\mathrm{size}(\mathrm{ann}\ l) + C.\mathrm{size}(r)$$

lookup

$$d \xrightarrow{\text{annotate}} c$$

$$D.\mathrm{lookup}(d, k) = C.\mathrm{lookup}(c, k)$$

For any $d : ('k, 'v)\ D.\mathrm{dict}$ and $\mathrm{cmp} : 'k * 'k \to \mathrm{ord}$, $k : 'k$

$$D.\mathrm{lookup}(\mathrm{cmp}, d, k) = C.\mathrm{lookup}(\mathrm{cmp}, \mathrm{annotate}\ d, k)$$

[both NONE
or SOME v and SOME v'
with v = v']

$\boxed{\text{For empty}}$

To show: D.empty $\xrightarrow{\text{annotate}}$ C.empty

$\mapsto$ D.Empty $\qquad \mapsto$ C.Empty

✓

For insert:

If $d \xrightarrow[\text{ann}]{\text{annotate =}} c$

then $D.\underbrace{\text{insert}}_{\text{ins}}(cmp, d, (k', v')) \xrightarrow[\text{ann}]{\text{annotate}} C.\underbrace{\text{insert}}_{\text{ins}}(cmp, c, (k', v'))$

If $d : ('k, 'v)\ D.\text{dict}$ and $k' : 'k$ and $cmp : 'k * 'k \to \text{order}$
$v' : 'v$

then

$$\text{ann}(D.\text{ins}(cmp, d, (k', v'))) = C.\text{ins}(cmp, \text{ann}\ d, (k', v'))$$

Insert

Case for D.E:

$$\mathrm{ann}(D.\mathrm{ins}(D.E, (k', v'))) = C.\mathrm{ins}(\mathrm{ann}\, D.E, (k', v'))$$

$$\mapsto \mathrm{ann}(D.N(D.E, (k', v'), D.E))$$

$$\mapsto^{*} C.\mathrm{Node}'(C.E, (k', v'), C.E)$$

RHS

$$\mapsto C.\mathrm{ins}(C.E, (k', v'))$$

$$\mapsto C.\mathrm{Node}'(C.E, (k', v'), C.E)$$

✓

$\boxed{\text{Insert}}$

Case $D.N(l, (k,v), r)$

To show:

$$ann(D.ins(D.N(l, (k,v), r), (k', v')))$$

$$=$$

$$C.ins(ann(D.N(l, (k,v), r)), (k', v'))$$

$\boxed{\text{insert}}$ $\boxed{\text{Node}}$ ① $cmp(k, k') = EQUAL$

LHS

$\mapsto ann(D.Node(l, (k, v'), r))$

$\mapsto C.Node'(ann\ l, (k, v'), ann\ r)$

RHS

ann total
size total

$\mapsto C.ins(C.Node'(ann\ l, (k, v), ann\ r), (k', v'))$

$\mapsto C.ins(C.N(ann\ l, (k, v), \underbrace{\text{compute size}}_{\text{valuable}}, ann\ r), (k', v'))$

$\mapsto C.Node'(ann\ l, (k, v'), ann\ r)$

$\boxed{\text{Insert}}$ $\boxed{\text{Node}}$ $\mathrm{Cmp}(k', k) = \mathrm{LESS}$

LHS

$$\mapsto \mathrm{ann}(D.N(D.\mathrm{ins}(l, (k', v')), (k, v), r))$$

$$\mapsto C.\mathrm{Node}'(\mathrm{ann}(D.\mathrm{ins}(l, (k', v'))), (k, v), \mathrm{ann}\ r)$$

$$=_{IH} C.\mathrm{Node}'(C.\mathrm{ins}(\mathrm{ann}\ l, (k', v')), (k, v), \mathrm{ann}\ r)$$

RHS

$$\mapsto C.\mathrm{ins}(C.\mathrm{Node}'(\mathrm{ann}\ l, (k, v), \mathrm{ann}\ r), (k', v'))$$

$$\mapsto C.\mathrm{Node}'(C.\mathrm{ins}(\mathrm{ann}\ l, (k', v')), (k, v), \mathrm{ann}\ r)$$