CS 839: Systems Verification

Fall 2025

Today's agenda

1. Rocq demo / review
2. Informal proofs
3. Induction

$P(n) \triangleq 1+2 \cdots + n = n(n+1)/2$

$P(1) \quad 1 = 1(1+1)/2$

$= 1$

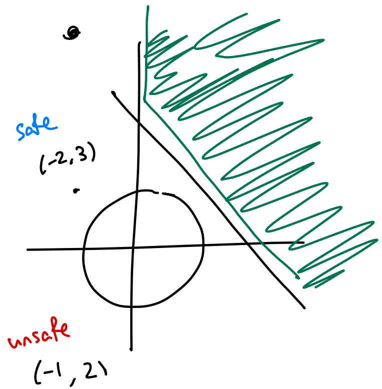
$\forall k, \quad P(k) \to P(k+1)$

$$\frac{IH: \quad 1 + \cdots + k = k(k+1)/2}{\boxed{1 + \cdots + k} + (k+1) = (k+1)(k+2)/2}$$

$= k(k+1)/2 + (k+1) = (k+1)(k+2)/2$

$= \dfrac{k(k+1)}{2} + \dfrac{2(k+1)}{2}$

$\dfrac{(k+2)(k+1)}{2} = \dfrac{(k+1)(k+2)}{2}$



$$e = \sigma_1, \sigma_2, \sigma_3, \ldots$$

eg. $(5,3)$

$$tr(\sigma, \sigma') = \sigma \searrow \sigma' \;\lor\; \sigma \uparrow \sigma' \;\lor\; \text{noop}$$

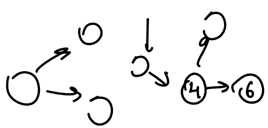
$$\sigma_1 = (0,5)$$

$$\frac{\forall i,\quad tr(e(i), e(i+1))}{\forall i,\quad e(i) \text{ safe}}$$

$$\frac{\forall \sigma.\ init(\sigma) \to P(\sigma) \qquad \forall \sigma, \sigma'.\ P(\sigma) \to tr(\sigma, \sigma') \to P(\sigma')}{\forall e,\ valid(e) \to \forall i,\ P(e(i))}$$

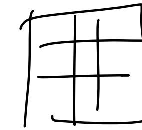
induction for transition systems

valid(e): follows init and tr



Lecture 4: Abstraction

- Most of today will be a fun (?!) activity
- Read the notes



```
location := {
    row: int  [0, 2]
    col: int  [0, 2]
}

player := black | white
cell   := empty | full (p: player)
state  := location → cell

game_over (s: state)

move (s, s') : Prop
```

---

contents of box: three 1's, two 2's, two 3's, two 4's, one 5 of each color

8 "Clock" tokens (hints)
4 "Black Fuse" tokens (lives/misplays)
assume 3 players, hand size of 5 if you like

Lecture 5: Hoare Logic (part 1)

Learning Outcomes:

1. Explain what pre- and post-conditions mean
2. Formally analyze a "whiteboard" programming language

$\{P\}\ e\ \{Q\}$ — if P holds and we run e, and it finishes, then Q $\Big]$ soundness

- semantics: "run e"?
- logic: set of rules for $\{P\}\ e\ \{Q\}$
- soundness

$\{P\}\ e\ \{\lambda v.\ Q(v)\}$

$e \rightsquigarrow v'$

$Q(v')$

code | proof

recursive | inductive

calls mod | proof of euclid "call" proof of mod

euclid(a, b)

mod(a, b)

$\{\_\_\}\ \mathrm{mod}(a, b)\ \{c.\ \_\_\_\}$

$\{\_\_\}\ \mathrm{euclid}(a, b)\ \{c.\ \mathrm{gcd}(a, b, c)\}$

$3 : \text{nat}$

$\bar{3} : \text{val}$

expr $\quad e ::= x \mid v \mid \lambda x. e \mid e_1\ e_2$

$\quad\quad \mid \underline{\text{if}}\ e\ \underline{\text{then}}\ e_1\ \text{else}\ e_2$

$\quad\quad \mid e_1 + e_2$

$\quad\quad \mid (e_1, e_2) \mid \pi_1\ e \mid \pi_2\ e$

values $\quad v ::= \lambda x. e \mid \bar{n} \mid \text{true} \mid \text{false} \mid (v_1, v_2)$

$\underline{\text{let}}\ x := e_1\ \underline{\text{in}}\ e_2 \quad ::= \quad (\lambda x. e_2)\ e_1$

$(\lambda x. e)\ v \longrightarrow e[v/x]$ β-reduction $(\lambda x.\ x + \bar{3})\ \bar{5}$

$\downarrow$

$\bar{5} + \bar{3}$

$\text{if false then } e_1 \text{ else } e_2 \longrightarrow e_2$

step

$\boxed{e_1 \longrightarrow e_2}$

$\pi_1\ (v_1, v_2) \longrightarrow v_1$

$\pi_2\ (v_1, v_2) \longrightarrow v_2$

$e_1 \longrightarrow^{*} e_2$

$\bar{n_1} + \bar{n_2} \longrightarrow \overline{(n_1 + n_2)\ \%\ 2^{64}}$

we have products (tuples)

add sums (inductives / enums)

$e ::= \ldots \mid$

$\underline{ok}\ e \mid \underline{err}\ e \mid$

$\underline{match}\ e\ \underline{with}$
$\mid ok\ x \Rightarrow e_1$
$\mid err\ x \Rightarrow e_2$

$\underline{case}\ (e, ok\_f, err\_f)$

$case\ (ok\ e, ok\_f\ err\_f) \longrightarrow ok\_f\ e$
$case\ (Err\ e, ok\_f, err\_f) \longrightarrow err\_f\ e$

$A + B$

Either a b

Result ⟨A, B⟩
| Ok (a : A)
| Err (b : B)

$$e ::= x \mid v \mid \lambda x.e \mid e_1 + e_2 \mid \dots$$

$$v ::= \bar{n} \mid \text{true} \mid \text{false} \mid \dots$$

$e_1 \rightarrow e_2$ step relation

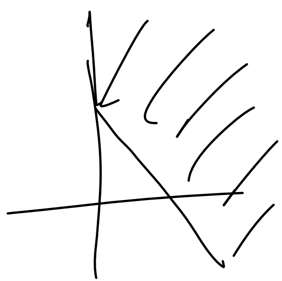
$e_1 \rightarrow^* e_2$ semantics

$\{P\}\ e\ \{\lambda v.\ Q(v)\}$

if

$(\sigma, pc) \rightarrow^* (\sigma', pc')$ then

$Q(\sigma')$ $\quad \sigma'(r_6) = 0 + \dots + 10$



$$\boxed{\forall v',\ P \wedge e \rightarrow^* v' \Rightarrow Q(v')}$$

$$\{Q(v)\} \quad v \quad \{v.\ Q(v)\}$$

$$\frac{\overset{2}{}\quad P \vdash Q(v)}{\{P\} \quad v \quad \{v.\ Q(v)\}}$$

1 2 3

$$\frac{\{P\}\ e_1\ \{v.\ Q(v)\} \qquad \forall v,\ \{Q(v)\}\ e_2[v/x]\ \{R\}}{\{P\}\ \underline{\text{let}}\ x := e_1\ \underline{\text{in}}\ e_2\ \{R\}}\ \text{hoare-let}$$

Lecture 6: Hoare Logic (part 2)

Hoare logic ×2

Separation logic ×2

Iris Proof Mode (Rocq)

Learning outcomes:

1. Prove reasoning principles in Hoare Logic
2. Analyze pre- and post-conditions

$\{P\}\ e\ \{v.\ Q(v)\}$

reasonable?

$\{P\}\ 3\ \{Q\}$

1. $\forall v',\ P \wedge (e \to^* v') \Rightarrow Q(v')$ ✓

2 ✓ maybe

4 too strong

3 ✗

5 ✗

5. $(P \wedge e \to^* 17) \Rightarrow Q(17)$

$P \wedge \forall v'',$ $(\text{let } x := e_1 \text{ in } e_2) \longrightarrow^* v''$

prove: $R(v'')$

use $\{P\}\ e_1\ \{Q\}$ prove P ✓

$e_1 \longrightarrow^* v' \implies$

$Q(v')$ ✓

conclude $R(v'')$

$(\text{let } x := e_1 \text{ in } e_2) \longrightarrow^*$

$(\text{let } x := v' \text{ in } e_2) \longrightarrow^*$

$\boxed{e_2[v'/x] \longrightarrow^* v''}$ ✓

$e_1 \longrightarrow^* v'$

2. $\{\text{True}\}\ \text{add}\ \bar{n}\ \bar{m}\ \{v.\ v = \overline{n+m}\}$

$\{n < 2^{64} - 1\}$

$3 + \text{true} \not\to \cdot$

$f\ \bar{n}$

$\{\exists p.\ y = \bar{p} \land p \leq 1\}$

What did you learn?

What are you confused about?

Lecture 7: Separation Logic (part 1)

- syntax, semantics ← add a heap
- P, Q $\quad \forall x,\ P(x) \quad P \lor Q \quad$ Prop ← heap predicates
- $\{P\}\ e\ \{v.\ Q(v)\}_{SL}$
  → soundness
  → rules for constructing proofs

$\text{alloc } e^{:T} \quad : \text{ref } T \qquad *T \qquad e : \text{loc}$

$!\,e \qquad *e$

$e \leftarrow v \qquad *e = v$

$h : \text{loc} \rightarrow \text{option val}$

"->" on slides

$(e, h) \rightsquigarrow (e', h')$

$!5$

$Q : \text{val} \to (\text{heap} \to \text{Prop})$

$\text{val} \to \text{heap} \to \text{Prop}$ "currying"

$Q \; v \; h$

SL propositions

model as heap $\to$ Prop

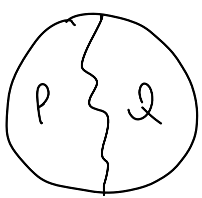
$$(\ell \mapsto v)(h) \triangleq h(\ell) = v \wedge \mathrm{dom}(h) = \{\ell\}$$

$$h = \{\ell \mapsto v\}$$

$$P \vdash Q \triangleq \forall h,\ P(h) \to Q(h)$$

$$(P * Q)(h) \triangleq \exists h_1, h_2,\ h = h_1 \cup h_2 \wedge \underbrace{h_1 \perp h_2}_{\text{disjoint}} \wedge P(h_1) \wedge Q(h_2)$$

h



$$\mathrm{emp}(h) \triangleq \mathrm{dom}(h) = \emptyset$$

$$\mathrm{True}(h) = \mathrm{True}$$

$$\frac{(P * Q)(h) \qquad \forall h,\ P(h) \to P'(h)}{\text{goal}: \quad (P' * Q)(h)}$$

$h = h_1 \cup h_2$

$P(h_1) \qquad Q(h_2)$

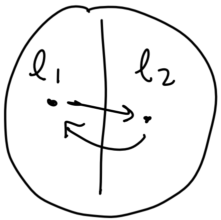
$\downarrow$

$P'(h_1)$

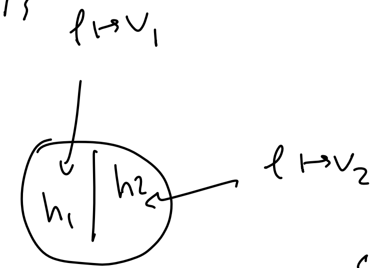
$P * Q \vdash Q * P$ sep-comm

$\vdash Q * P'$ sep-monotone-right

$\vdash P' * Q$ sep-comm



$\{ l_1 \mapsto l_2 ; \quad l_2 \mapsto l_1 \}$



$\boxed{l \mapsto v_1 \ * \ l \mapsto v_2} \vdash \text{False}$

$\mathrm{dom}(h_1) = \{l\}$

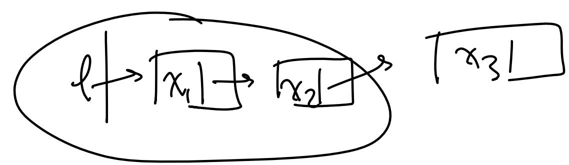
$\mathrm{dom}(h_2) = \{l\}$

$h_1 \perp h_2$

$l_1 \mapsto v_1 \ * \ l_2 \mapsto v_2 \quad \vdash \quad l_1 \neq l_2$

$\text{llist}(l_1, \ xs) \ * \ \text{llist}(l_2, ys)$

list int

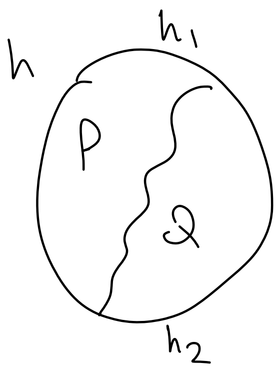


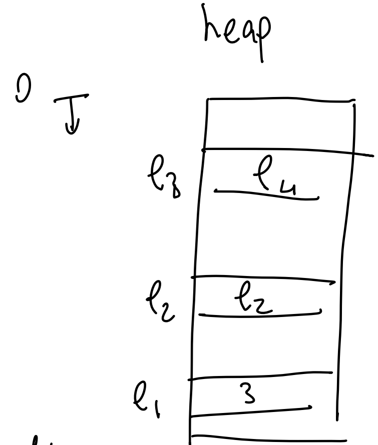
Lecture 8

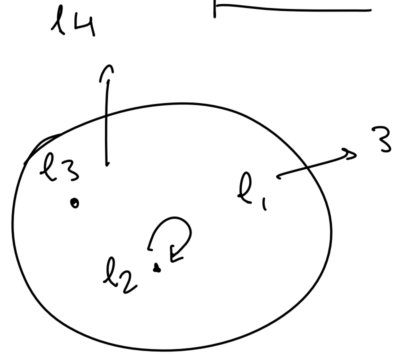
discriminate missing

caution in using induction tactic

Strategy for even proof is not obvious







$\{P\}\ e\ \{v.\ Q(v)\}$

$$\varphi : \mathrm{Prop}$$

$$\lceil \varphi \rceil : \mathrm{hProp}$$

$w = \#\ell$

val   loc

$$\lceil \varphi \rceil (h) \triangleq \varphi \wedge h = \{\}$$

$$\mathrm{emp} \dashv\vdash \lceil \mathrm{True} \rceil$$

$$P, Q : \mathrm{hProp}$$

$$(P \wedge Q)(h) \triangleq P(h) \wedge Q(h)$$

$$\lceil \varphi \rceil * P \dashv\vdash \lceil \varphi \rceil \wedge P$$

$$\frac{P' \vdash P \qquad Q \vdash Q' \qquad \{P\}\ e\ \{Q\}}{\{P'\}\ e\ \{Q'\}}$$

$$\frac{\{emp\}\ \text{alloc } 42\ \{y \mapsto 42\}}{\{x \mapsto 0\}\ \text{alloc } 42\ \{x \mapsto 0 * y \mapsto 42\}}$$

RET #();

$\{\ell_1 \mapsto 0\}\ f(\ell_1, \ell_2)\ \{\ell_1 \mapsto 42\}$

$\{emp\}$ assert #true $\{emp\}$

let t := $!\ell_1$ in
let t_2 := $!\ell_2$ in
$\ell_1 \leftarrow t_2$;
$\ell_2 \leftarrow t$

$\{emp\}$
let x := alloc 0 in
$\{\boxed{x \mapsto 0}\}$
let y := alloc 42 in
$\{\boxed{x \mapsto 0} * \boxed{y \mapsto 42}\}$
$\boxed{f(x, y)}$
$\{\boxed{x \mapsto 42} * \boxed{y \mapsto 42}\}$
let a := $!x$ in
$\{\boxed{\ulcorner a = 42 \urcorner} * \boxed{x \mapsto 42} * y \mapsto 42\}$
let b := $!y$ in
$\{\boxed{\ulcorner b = 42 \urcorner * x \mapsto 42 * y \mapsto 42}\}$
assert #(bool_decide $(a = b)$)
$\{x \mapsto 42 * y \mapsto 42\}$
$\{True\}$

$\{\ell_1 \mapsto a \ast \boxed{\ell_2 \mapsto b}\}$

let $t := !\ell_1$ in

$\{\boxed{\ulcorner t = a \urcorner \ast \ell_1 \mapsto a} \ast \ell_2 \mapsto b\}$

let $t_2 := !\ell_2$ in

$\{\boxed{\ulcorner t_2 = b \urcorner} \ast \boxed{\ell_2 \mapsto b} \ast \ell_1 \mapsto a\}$

$\ell_1 \leftarrow t_2;$

$\{\boxed{\ell_1 \mapsto t_2} \ast \ell_2 \mapsto b\}$

$\ell_2 \leftarrow t$

$\{\ell_2 \mapsto \cancel{t}\ a \ast \ell_1 \mapsto \cancel{t_2}\ b\}$

$\{\ell_1 \mapsto b \ast \ell_2 \mapsto a\}$