CS 520
Theory of Programming Language

03/31 – 04/07, 2021

3. Program Specification and Proofs.

1. Motivation.

① $C_{div3} \equiv$

```
a := 0;
b := x;
while (b ≥ 3) do
    b := b - 3;
    a := a + 1
```

... (1) Formally specify the intended behaviour.

(2) Formally prove that the program indeed satisfies the spec.

② Hoare logic. ✓ $\{\underbrace{x \geq 0}_{\text{precondition}}\}\ \underbrace{C_{div3}}_{\text{program}}\ \{\underbrace{x = 3 \times a + b \wedge 0 \leq b < 3}_{\text{postcondition}}\}$.

✓

③ Partial correctness. Total correctness.

2. Specification.

Hoare triple. , triple.

partial correctness.

① Abstract syntax.

$$\langle spec \rangle ::= \{\langle assert \rangle\} \langle comm \rangle \{\langle assert \rangle\}$$

$$\mid \ [\langle assert \rangle] \langle comm \rangle [\langle assert \rangle]$$

total correctness.

$\{p\}\ c\ \{q\}$ $\qquad$ $[p]\ c\ [q]$ $\qquad$ " $\qquad$ triple.

if p true initially
$\wedge$ c terminates
then q true at
the final state.

if p true at the initial state.
then c terminates.
& the final state satisfies q.

② Semantics.

$$[\![-]\!]_{\underbrace{spec}_{\text{omit this}}} : \langle spec \rangle \to \mathbb{B} = \{tt, ff\}.$$

$[\![\{p\} \, c \, \{q\}]\!] = tt$ iff for all states $\sigma \in \Sigma$.
if $[\![p]\!]\sigma = tt$ and $[\![c]\!]\sigma \neq \perp$,
then $[\![q]\!]([\![c]\!]\sigma) = tt$.

(1) ex. $[\![[p] \, c \, [q]]\!] = tt$ valid. iff for all states $\sigma \in \Sigma$
if $[\![p]\!]\sigma = tt$,
then $[\![c]\!]\sigma \neq \perp$ and $[\![q]\!]([\![c]\!]\sigma) = tt$.

which ones are valid?

(2) {false} while true do skip {false}. → valid
{true} " {false}. → valid.
[false] " [false] → valid.
[true] " [false]. → not valid.

{p}. C $\{\underset{\sim}{\text{false}}\}$ ...... C doesn't terminate when p holds initially.

e.g.

$\{x \geq 0\} \; C_{div3} \; \{x = 3 \times a + b \wedge 0 \leq b < 3\}$.

$\{n \geq 1\} \; C_{fib} \; \{x = fib(n)\}$.

math. notation.

3. Proof rules.

p true always.

$\{p\}c\{q\}$

① Have the form of. $$\frac{\varphi_1 \quad \varphi_2 \quad \cdots \quad \varphi_n}{\varphi}$$

p

$\{p'\}d\{q'\}$

$x := e$

$c_1 ; c_2$

$\vdots$

non-structural rules.

② Hoare-logic proof rules. .... two groups, the first tied to prog. const. and the second not tied.

structural rules.

(1) Non-structural rules.

$$\frac{}{\{p\}\,\text{skip}\,\{p\}} \qquad \frac{}{\{q[x \to e]\}\ x := e\ \{q\}}$$

SP. assignment rule.

$$\frac{\{p_1\}\ c_1\ \{q\} \qquad \{p_2\}\ c_2\ \{q\}}{\{(b \Rightarrow p_1) \land (\neg b \Rightarrow p_2)\}\ \text{if}\ b\ c_1\ c_2\ \{q\}}$$

$$\frac{\{p\}\ c_1\ \{r\} \qquad \{r\}\ c_2\ \{q\}}{\{p\}\ c_1 ; c_2\ \{q\}} \qquad \frac{\{p \land b\}\ c_1\ \{q\} \qquad \{p \land \neg b\}\ c_2\ \{q\}}{\{p\}\ \text{if } b \text{ then } c_1 \text{ else } c_2\ \{q\}}$$

$$\frac{\{i \land b\}\ c\ \{i\}}{\{i\}\ \text{while } b \text{ do } c\ \{\underline{i \land \neg b}\}}$$

loop inv.

(2) Structural rule.

$$\frac{p' \Rightarrow p \qquad \{p\}\ c\ \{q\} \qquad q \Rightarrow q'}{\{p'\}\ c\ \{q'\}}$$

ex Prove:

(i) $\{x \geq 0\}$. $a := 0 ; b := x$ $\{x = 3 \times a + b \wedge b \geq 0\}$.

(ii) $\{x = 3 \times a + b \wedge b \geq 0 \wedge b \geq 3\}$

$b := b - 3 ; a := a + 1$

$\{x = 3 \times a + b \wedge b \geq 0\}$.

$x=3a+b$ $\quad$ $x=3(a+1)+b-3.$

$\wedge b\geq 0$ $\quad \Rightarrow \quad \wedge\ b-3\geq 0.$

$\wedge b\geq 3$

$\{x=3(a+1)+b-3 \wedge b-3\geq 0.\}$ $b:=b-3$ $\{$ " $\}.$

$\{$ " $\}$ $b:=b-3.$ $\{x=3(a+1)+b \wedge b\geq 0.\}$ $\quad$ $\{x=3(a+1)+b \wedge b\geq 0\}$ $a:=a+1$ $\{$ " $\}$

$\{x=3a+b \wedge b\geq 0 \wedge b\geq 3\}$ $b:=b-3;$ $a:=a+1$ $\{x=3a+b \wedge b\geq 0\}.$

$\{x=3a+b \wedge 0\leq b \wedge b\geq 3\}$ ..... $\{x=3a+b \wedge 0\leq b\}.$

$\{x\geq 0\}$ $a:=0;b:=x$ $\{x=3a+b \wedge 0\leq b\}$ $\quad$ $\{x=3a+b \wedge 0\leq b\}$ while ..... $\{$ " $\}.$

$\{x\geq 0\}.$ $C_{div3.} \equiv$ $a:=0;b:=x$; while $b\geq 3$ do $b:=b-3;$ $a:=a+1$ $\{x=3a+b \wedge 0\leq b \wedge \neg(b\geq 3)\}$

$\{x=3a+b \wedge 0\leq b\}.$

ex:

(1)

$$\frac{}{\{p\}\ x := e\ \{\ \ldots\ \}}.$$

↑ what ... should be?

(2). Write specs & proofs of fib. & Euclid.