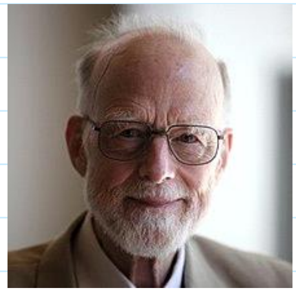
# 17 Axiomatic Semantics

Wednesday, November 6, 2024 12:36 PM

- Hoare Logic.
  '69 C.A.R. Hoare
  "Tony"



What? A Formal system of Logic Rules to reason rigurously about programs.

Formal = mathematical.

Objectives :-
- Give meaning to programs
- Prove that a program has certain properties
- Prove correctness of a program

Dijkstra, Wirth, Knuth, Lamport agree.

Program is a mathematical object.

- REVIEW of Mathematical Logic

Logic: the mathematics of correct reasoning
- "true" - "False"
- Evaluate sentences/formulas to know wheather they are true or false

- PROPOSITIONAL LOGIC (Boolean Logic)
  - Propositions

p q r s

Propositions are assigned True/False

p = true
q = true.

- Logical Connectives $\wedge \ \vee \ \neg \ \rightarrow$
  the meaning stipulated by truth tables

| $p$ | $\neg p$ |
|---|---|
| T | F |
| F | T |

| $p$ | $q$ | $p \wedge q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | F |
| F | F | F |

| $p$ | $q$ | $p \rightarrow q$ |
|---|---|---|
| T | T | T |
| T | F | F |
| F | T | T |
| F | F | T |

- Propositional Formulas
  To be evaluated

$p = T$
$q = F$
$r = T$

$$((p \vee q) \vee \neg r) \wedge (r \rightarrow q) = \text{False}$$

T F F
T
F

- Inference Rules
  rules of pre-evaluated formulas

Premises → $S_1 \ S_2 \ S_3 \ldots S_k$
Conclusion → $C$

- if all $S_i$ are true then $C$ is true.

e.g. $\dfrac{p \wedge q}{p}$ $\qquad \dfrac{p \rightarrow q \quad p}{q}$ $\qquad \dfrac{\quad}{p \vee \neg p}$

Limitations:
e.g.: "All men are mortal,
Aristotle is a man
therefore Aristotle is mortal"

- PREDICATE LOGIC
  - Objects a b aristotle
    1 7 garfield
  - Variables over objects
    X Y Z
  - Predicates

X Y Z

- Predicates
  - represent properties of objects or relationships between objects
  - have an "arity": fixed number of arguments

  e.g. red(a)  man(aristotle)  cat(garfield)
  odd(7)  friend(aristotle, garfield)

  lessThan(1, 7)  likes(garfield, X)

  "Ground" predicates (those without variables) can be assigned true/false.

  man(aristotle) true  lessThan(1, 7) true
  cat(aristotle) False  red(a) False.

- Quantifiers :- to handle variables.
  $\forall x\ (\ p(x)\ )$ :- true if p is true for every object
  $\exists x\ (\ p(x)\ )$ :- True if p is true for at least one object

- Logical Connectives
  $\wedge \quad \vee \quad \neg \quad \rightarrow$

- Predicate Formulas
  - Objects, variables, predicate
  - Quantifiers, connectives.

eg. objects {a, b}  predicates {p q r}

$p(a) = T$
$p(b) = T$
$q(a) = F$
$q(b) = T$
$r(a) = T$
$r(b) = F$

$$\underbrace{\underbrace{\forall X\, p(x)}_{T} \vee \underbrace{\exists Y\, q(Y)}_{T}}_{T} \wedge \underbrace{\neg \underbrace{\forall X\, r(X)}_{F}}_{T} \;=\; T$$

- Rules of inference.

$$\frac{p(a),\ p(b)}{p(a) \vee p(b)} \qquad \frac{\forall X\, p(x)}{p(c)} \qquad \frac{p(c)}{\exists X\, p(x)}$$

$$\frac{p(a),\ p(b)}{p(a) \vee p(b)} \qquad \frac{\forall x\, p(x)}{p(c)} \qquad \frac{p(c)}{\exists x\, p(x)}$$

$$\frac{\forall x\, p(x) \to q(x),\ \ p(c)}{q(c)}$$ "modus ponnens"

- The power of predicate logic

1) all men are mortal
2) aristotle is a man
   therefore
3) aristotle is mortal

objects {aristotle, garfield}

1) $\forall x$ man(x) $\to$ mortal(x) = True
2) man(aristotle) = true

(plug into modus ponners

mortal(aristotle) = true.

- HOARE LOGIC

A logic for programs.

- Objects

Program Variables and their assignments.

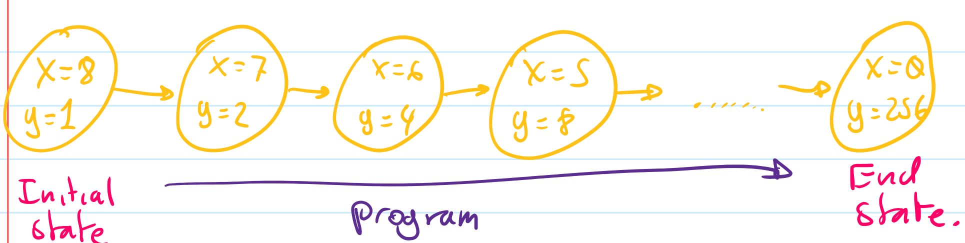
- State :-

An assignment of values to variables

- Program Execution :-

A sequence of states.

A sequence of transitions from an initial state to a final state.




- Predicate Formulas.
  - To talk about states.
  - Not to assign them true or false
  - Represent the set of all states in which

- Not to assign them True or False
- Represent the set of all states in which the formula is true

e.g: Objects $\{x, y\}$ $\mathbb{N} \cup \{Q\} = \mathbb{N}_Q$

$$even(x) = \{ \ \{x=2, y=7\} \quad \{x=4, y=29\} \quad \{x=8, y=7\} \ \ldots\ldots \ \}$$

$$lessthan(x,y) = \{ \ \{x=2, y=4\} \quad \{x=4, y=8\} \quad \{x=6, y=28\} \ \ldots\ldots \ \}$$

$x < y$

- "Hoare Triples": Formulas in Hoare Logic

$$\{P\} C \{Q\}$$

where

P, Q : Predicate formulas — P: "precondition", Q: "post-condition"

C : "Command" : a piece of code

In english:

if $\{P\} C \{Q\} = True$

what I mean is: If you execute C in a state in which P is true, the program C will finish is a state in which Q is true

¿ How is this usefull ?

C your program

$\{P\}$ : a formula : spec of the input

$\{Q\}$ : a formula : spec of the output.

If you prove $\{P\} C \{Q\}$, you prove your program is correct

# Semantics through logic Rules.

- A program is a mathematical object
- A program can be analysed mathematically
- Use mathematics to prove properties of programs.

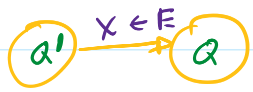
- Rules of inference

  Cover.
  - assignment
  - conditionals
  - loops

## – Axiom of Assignment

$$\overline{\{Q_{\underbrace{x \to E}}\}\ x \leftarrow E\ \{Q\}}$$

replace every occurance of x in Q by E

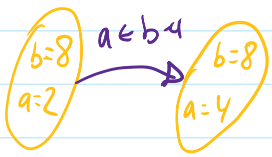



E.g.

$$\{\ ?\ \}\ \underbrace{a}_{x} \leftarrow \overset{C}{\underbrace{b-4}_{E}}\ \overset{Q}{\{\underline{a} > 0\}}$$

$\{\underline{b-4} > 0\}$

$\{b > 4\}\ a \leftarrow b-4\ \{a > 0\}$ = True.




E.g. $\{\ ?\ \}\ a \leftarrow 2*(b-2)\ \{0 \le \underline{a} \le 10\}$

$\{0 \le 2\cdot(b-2) \le 10\}$

$0 \le 2b-4 \le 10$

$4 \le 2b \le 14$

$\{2 \le b \le 7\}\ a \leftarrow 2\cdot(b-2)\ \{0 \le a \le 10\}$ = True.

## – Rule of Composition

$$\frac{\{P\}C_1\{R\}\ ,\ \{R\}C_2\{Q\}}{\{P\}C_1;C_2\{Q\}}$$

E.g:

$$\{?\}\ a \leftarrow 3*b-1\ ;\ b \leftarrow 4*a-22\ \overset{Q}{\{b > 10\}}$$

$\{4a-22>10\}\ b \leftarrow 4*a-22\ \{b>10\}$

$\{4a>32\}$

$\{a>8\}\ b \leftarrow 4*a-22\ \{b>10\}$ = true

R C$_2$ Q

$\{3b-1>8\}\ a \leftarrow 3*b-1\ \{a>8\}$

$\{3b>9\}$

$\{b>3\}\ a \leftarrow 3*b-1\ \{a>8\}$ = True.

P C$_1$ R

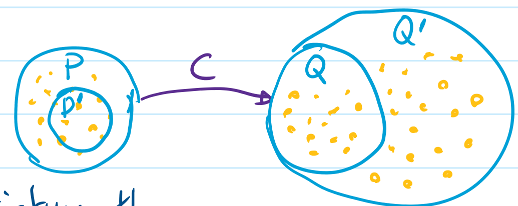
$\{b>3\}\ a \leftarrow 3\cdot b-1\ ;\ b \leftarrow 4*a-22\ \{b>10\}$ = true.

P Q

- Rule of consequence.

Intuition:

Suppose $\{P\}C\{Q\}$ true.




Restricting the precondition.

Relaxing the Postcondition.

$$\frac{P' \subseteq P,\quad \{P\}C\{Q\},\quad Q \subseteq Q'}{\{P'\}\,C\,\{Q'\}}$$

E.g.

$\{b>4\}\ a \leftarrow b-4\ \{a>0\}$ = True.

Restrict the precondition

- $\{b>10\}\ a \leftarrow b-4\ \{a>0\}$ = true $\quad \{b>10\} \subseteq \{b>4\}$
- $\{4<b<8\}\ a \leftarrow b-4\ \{a>0\}$ = true. $\quad \{4<b<8\} \subseteq \{b>4\}$

Relax the postcondition

- $\{b>4\}\ a \leftarrow b-4\ \{a \geq 0\}$ = true $\quad \{a>0\} \subseteq \{a \geq 0\}$
- $\{b>4\}\ a \leftarrow b-4\ \{True\}$ = true.

Q

- Conditional Rule.

$$\frac{\{B \wedge P\}\, C_1 \,\{Q\} \quad , \quad \{\neg B \wedge P\}\, C_2 \,\{Q\}}{\{P\}\ \text{IF } B \text{ THEN } C_1 \text{ ELSE } C_2\ \{Q\}}$$

Prove:

$\underbrace{\{0 \leq x \leq 12\}}_{P}$ IF $x < 12$ THEN $x \leftarrow x+1$ ELSE $x \leftarrow 0$ $\underbrace{\{0 \leq x \leq 12\}}_{Q}$ = True

We need to prove:

1) $\{0 \leq x \leq 12 \wedge x < 12\}$ $x \leftarrow x+1$ $\{0 \leq x \leq 12\}$ = true
   $\{0 \leq x \leq 11\}$

2) $\{0 \leq x \leq 12 \wedge \neg(x < 12)\}$ $x \leftarrow 0$ $\{0 \leq x \leq 12\}$
   $\{x = 12\}$

Proof of #1

$\{0 \leq x+1 \leq 12\}$ $x \leftarrow x+1$ $\{0 \leq x \leq 12\}$ true
$\{-1 \leq x \leq 11\}$

$\{0 \leq x \leq 11\} \subseteq \{-1 \leq x \leq 11\}$ True

$\{0 \leq x \leq 11\}$ $x \leftarrow x+1$ $\{0 \leq x \leq 12\}$ $\square$

Proof of #2

$\{0 \leq 0 \leq 12\}$ $x \leftarrow 0$ $\{0 \leq x \leq 12\}$ = true
$\{\text{True}\}$ $x \leftarrow 0$ $\{0 \leq x \leq 12\}$
$\{x = 12\}$ $x \leftarrow 0$ $\{0 \leq x \leq 12\}$ $\square$

- WHILE Rule

$$\frac{\{B \wedge P\}\, C \,\{P\}}{\{P\}\ \text{WHILE } \mathbf{B} \text{ DO } C\ \{\underbrace{\neg B}_{\text{Loop Termination}} \wedge P\}}$$

P: loop invariant.

P is something that is true before and after the loop.

## Invariant Properties

- Initialization :- Invariant is true before the loop
- Maintenance :- Invariant is true at the end of an iteration
- Termination :- Invariant is true after the loop is completed

Find the invariant P such that $\{\neg B \wedge P\}$ tells you something useful.

E.g

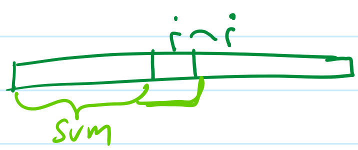
$\underbrace{\{x \leq y\}}_{P}$ WHILE $x<y$ DO $x \leftarrow x+1$ $\{\underbrace{x \leq y}_{P} \wedge \underbrace{\neg(x<y)}_{\neg B}\}$

$\{x \leq y \wedge x<y\}\ x \leftarrow x+1\ \{x \leq y\} = T$ $\qquad \{x = y\}$

Lets Revise this idea with less rigour.

```
FUNCTION sum(a[0..n-1])
    sum ← 0;  i ← 0;
    WHILE i < n DO
        sum ← sum + a[i]
        i ← i + 1
    RETURN sum
```

$\{P: \quad sum = \sum_{0}^{i-1} a[i]\}$

Maintenance: 

$\{P \wedge \neg B: \quad sum = \sum_{0}^{i-1} a[i] \wedge i = n \Rightarrow sum = \sum_{0}^{n-1} a[]$