Mathematical Structures in Programming

Zhenjiang Hu

Summer Term, 2008

Chapter 1: Predicate Calculus

Predicates on State Space

- A predicate is a function on a state space $\mathcal{X}$ (set of program variables):

$$P : \mathcal{X} \rightarrow \{\text{false}, \text{true}\}$$

  which actually defines a subset of $\mathcal{X}$.

- An Example
  - Suppose we have two program variables

$$x, y : \text{integer}$$

    then the state space $\mathcal{X}$ is

$$\mathcal{X} = \mathcal{Z} \times \mathcal{Z}.$$

    A predicate on $\mathcal{X}$ is

$$x \geq 2 \wedge y = 3.$$

Predicate Logic

- Predicates
  - Constants: true, false
  - Negation: $\neg P$
  - Conjunction: $P \wedge Q$
  - Disjunction: $P \vee Q$
  - Implication: $P \Rightarrow Q$
  - Equivalence: $P \equiv Q$
- Note: Negation has the highest priority while equivalence has the lowest.

$$P \Rightarrow Q \equiv \neg P \vee Q$$

should be read as

$$(P \Rightarrow Q) \equiv ((\neg P) \vee Q)$$

Validity

- P is true for all states is denoted by $[P]$.
- Examples
  - [true]
  - $[Q \equiv Q]$
  - $[\neg(P \wedge Q) \equiv \neg P \vee \neg Q]$, many and many
  - $[(x+1)^2 = x^2 + 2x + 1]$
  - $[x \geq 1 \Rightarrow x \geq 0]$
- If $[P \Rightarrow Q]$, then
  - P is stronger than Q, or
  - Q is weaker than P.

Q: What is the weakest predicate and what is the strongest predicate?

Substitution

- Substitution of expression E for variable x in expression Q is denoted by

$$Q(x := E)$$

- Examples
  - $[(x^2 + 2 * x)(x := x + 1) \equiv (x + 1)^2 + 2 * (x + 1)]$
  - $[(x + 2 * y = z)(x, y := y, x) \equiv y + 2 * x = z]$
  - $[(x = E)(x := E) \equiv E = E(x := E)]$
  - $[(P(x := y))(x := y) \equiv P(x := y)]$
  - $[(P(x := y))(y := x) \equiv P(y := x)]$
  - $[(P \wedge Q)(x := E) \equiv P(x := E) \wedge Q(x := E)]$

Quantification

- Existential quantification:

$$(\exists i : R : P)$$

  - i: a bound variable, or a dummy (of type $\mathcal{Z}$)
  - R: a range
  - P: a term

  Examples:

  - $(\exists i : i \in \mathcal{Z} \wedge i \geq 0 : x = 2i)$
  - $[(\exists i : \text{false} : P) \equiv \text{false}]$

- Universal quantification:

$$(\forall i : R : P)$$

Chapter 2: The Guarded Command Language

Hoare's Triples

$$\{P\}S\{Q\}$$

Each execution of S in a state satisfying P terminates in a state satisfying Q.

- P: precondition or assertion
- S: statement (program)
- Q: postcondition or assertion

General Rules of Programs

- Trivial (S will not be executed.)

$$\{\text{false}\}S\{P\}$$

- Termination

$$\{P\}S\{\text{true}\}$$

- Execluded Miracles

$$\{P\}S\{\text{false}\} \text{ is equivalent to } [P \equiv \text{false}]$$

- Strengthening of Precondition

$$\{P\}S\{Q\} \text{ and } [P_0 \Rightarrow P] \text{ implies } \{P_0\}S\{Q\}$$

- Weakening of Postcondition

$$\{P\}S\{Q\} \text{ and } [Q \Rightarrow Q_0] \text{ implies } \{P\}S\{Q_0\}$$

- Conjunctivity

$$\{P\}S\{Q\} \text{ and } \{P\}S\{R\} \text{ is equivalent to } \{P\}S\{Q \wedge R\}$$

- Disjunctivity

$$\{P\}S\{Q\} \text{ and } \{R\}S\{Q\} \text{ is equivalent to } \{P \vee R\}S\{Q\}$$

Predicate Transformer

- $wp.S.Q$ denotes the weakest precondition of S with respect to Q.

$$\{P\}S\{Q\} \text{ is equivalent to } [P \Rightarrow wp.S.Q]$$

- Examples:
  - $[wp.S.\text{false} \ \equiv \ \text{false}]$
  - $[wp.S.Q \wedge wp.S.R \ \equiv \ wp.S.(Q \wedge R)]$
  - $[wp.S.Q \vee wp.S.R \Rightarrow wp.S.(Q \vee R)]$

Skip

- Execution of skip does not have any effect.

$$\{P\}\text{skip}\{Q\} \text{ is equivalent to } [P \Rightarrow Q]$$

- Example

$$
\begin{array}{l}
|[\\
\mathbf{var}\ x, y : \text{int}; \\
\{x \geq 1\} \\
\text{skip} \\
\{x \geq 0\} \\
]|
\end{array}
$$

- Weakest precondition: $wp.\text{skip}.Q \equiv Q$

Assignment

- Any change of state is due to the execution of an assignment statement.

$$x := E$$

replaces the value of x by the value of E.

$$\{P\}x := E\{Q\} \text{ is equivalent to } [P \Rightarrow \text{def}.E \wedge Q(x := E)]$$

Here def.E is defined for which values of its variables in E is defined.

$$\text{def}.(a \ \mathbf{mod}\ b) = b \neq 0$$

- Weakest precondition

$$[wp.(x := E).Q \ \equiv \ \mathbf{def}.E \ \wedge \ Q(x := E)]$$

- Example

$$\{x \geq 3\}x := x + 1\{x \geq 0\}$$

follows from

$$
\begin{array}{ll}
 & \text{def.}(x+1) \wedge (x \geq 0)(x := x+1) \\
\equiv & \quad \{\text{ def }\} \\
 & \mathit{true} \wedge (x \geq 0)(x := x+1) \\
\equiv & \quad \{\ \wedge,\text{ substitution }\} \\
 & x + 1 \geq 0 \\
\equiv & \quad \{\text{ arithmetic }\} \\
 & x \geq -1 \\
\Leftarrow & \quad \{\text{ arithmetic }\} \\
 & x \geq 3
\end{array}
$$

Catenation

- Catenation allows us to describe sequence of actions.

$$S;T$$

S is executed after which T is executed.

$$\{P\}S;T\{Q\} \text{ is equivalent to } \exists R.\{P\}S\{R\} \text{ and } \{R\}T\{Q\}$$

- Weakest precondition

$$[wp.(S;T).Q \;\equiv\; wp.S.(wp.T.Q)]$$

i.e., semi-colon corresponds to function composition.

- Prove

$$
\begin{array}{l}
|[\\
\textbf{var } a, b : \text{bool}; \\
\{(a \equiv A) \wedge (b \equiv B)\} \\
a := a \equiv b; \\
b := a \equiv b; \\
a := a \equiv b \\
\{(a \equiv B) \wedge (b \equiv A)\} \\
]|
\end{array}
$$

  ▶ Hint: Compute the weakest preconditions backwards.

Selection

$$\textbf{if } B.0 \rightarrow S.0 \ [] \ \cdots \ [] \ B.n \rightarrow S.n \textbf{ fi}$$

where

- $B.i$: a boolean expression (a guard)
- $S.i$: a statement
- $B.i \rightarrow S.i$: a guarded command

1. All guards B_i are evaluated.
2. If none of the guards evaluates to true then execution *aborts*, otherwise one of the guards that has the value true is chosen *non-deterministically* and the corresponding statement is executed.

An Example

Derive a statement S that satisfies

$$
\begin{array}{l}
|[\\
\textbf{var } x, y, z : \text{int}; \\
\{\text{true}\} \\
S \\
\{z = x \ \textbf{max} \ y\} \\
]|
\end{array}
$$

where **max** is defined by

$$z = x \ \textbf{max} \ y \ \equiv \ (z = x \vee z = y) \wedge z \geq x \wedge z \geq y$$

We conclude that $z := x$ is a candidate for S. As a precondition we can have

$$
\begin{aligned}
& ((z = x \vee z = y) \wedge z \geq x \wedge z \geq y)(z := x) \\
\equiv \quad & \quad \{ \text{ substitution } \} \\
& (x = x \vee x = y) \wedge x \geq x \wedge x \geq y \\
\equiv \quad & \quad \{ \text{ calculus } \} \\
& x \geq y
\end{aligned}
$$

So

$$x \geq y \rightarrow z := x$$

Symmetrically,

$$y \geq x \rightarrow z := y$$

So, the definition of S is

$$\mathbf{if}\ x \geq y \rightarrow z := x\ [\,]\ y \geq x \rightarrow z := y\ \mathbf{fi}$$

Formulation of Selection Statement

$$\{P\}\mathbf{if}\ B_0 \rightarrow S_0\ []\ B_1 \rightarrow S_1\ \mathbf{fi}\{Q\}$$

is equivalent to

1. $[P \rightarrow B_0 \vee B_1]$ and
2. $\{P \wedge B_0\}S_0\{Q\}$ and $\{P \wedge B_1\}S_1\{Q\}$

- Examples
  - Prove $\{x = 0\}\mathbf{if}\ \mathit{true} \rightarrow x := x + 1\ []\ \mathit{true} \rightarrow x := x + 1\ \mathbf{fi}\{x = 1\}$.
  - Prove $\{x = 0\}\mathbf{if}\ \mathit{true} \rightarrow x := 1\ []\ \mathit{true} \rightarrow x := -1\ \mathbf{fi}\{x = 1 \vee x = -1\}$.

Weakest Precondition for Selection

$$
\begin{aligned}
&[wp.(\mathbf{if}\ B_0 \rightarrow S_0\ []\ B_1 \rightarrow S_1\ \mathbf{fi}).Q \\
&\quad \equiv (B_0 \vee B_1)\ \wedge \\
&\qquad (B_0 \rightarrow wp.S_0.Q)\ \wedge \\
&\qquad (B_1 \rightarrow wp.S_1.Q) \\
&]
\end{aligned}
$$

Repetition

$$\mathbf{do}\ B.0 \rightarrow S.0\ [] \ \cdots \ []\ B.n \rightarrow S.n\ \mathbf{od}$$

1. All guards B_i are evaluated.
2. If none of the guards evaluates to true then execution *skip*, otherwise one of the guards that has the value true is chosen *non-deterministically* and the corresponding statement is executed, after which the repetition is executed again.

Formulation of Repetition Statement

$$\{P\}\mathbf{do}\ B_0 \rightarrow S_0\ []\ B_1 \rightarrow S_1\ \mathbf{od}\{Q\}$$

is equivalent to

$$\begin{array}{l} \{P\} \\ \mathbf{if}\ (\neg B_0 \wedge \neg B_1) \rightarrow \mathrm{skip} \\ \ []\ B_0 \rightarrow S_0;\ \mathbf{do}\ B_0 \rightarrow S_0\ []\ B_1 \rightarrow S_1\ \mathbf{od} \\ \ []\ B_1 \rightarrow S_1;\ \mathbf{do}\ B_0 \rightarrow S_0\ []\ B_1 \rightarrow S_1\ \mathbf{od} \\ \mathbf{fi} \\ \{Q\} \end{array}$$

So

(i) $[P \wedge (\neg B_0 \wedge \neg B_1) \Rightarrow Q]$ and

(ii) $\{P \wedge B_0\}S_0\{P\}$ and $\{P \wedge B_1\}S_1\{P\}$

implies

$$\{P\}\mathbf{do}\ B_0 \rightarrow S_0\ [\!]\ B_1 \rightarrow S_1\ \mathbf{od}\{Q\}$$

provided that this repetition terminates.

Note: A predicate P that satisfies (ii) is called an invariant of $\mathbf{do}\ B_0 \rightarrow S_0\ [\!]\ B_1 \rightarrow S_1\ \mathbf{od}$.

An Example

Prove that

$$
\begin{array}{l}
|[\\
\textbf{var } x, y : \text{int}; \\
\{x = X \wedge y = Y \wedge x > 0 \wedge y > 0\} \\
\textbf{do } x > y \rightarrow x := x - y \;[]\; y > x \rightarrow y := y - x \textbf{ od} \\
\{x = X \textbf{ gcd } Y\} \\
]|
\end{array}
$$

Here, X **gcd** Y denotes the greatest common divisor of X and Y, and **gcd** has the following properties:

$$
\begin{array}{l}
x \textbf{ gcd } x = x \\
x \textbf{ gcd } y = y \textbf{ gcd } x \\
x > y \Rightarrow x \textbf{ gcd } y = (x - y) \textbf{ gcd } y \\
y > x \Rightarrow x \textbf{ gcd } y = x \textbf{ gcd } (y - x)
\end{array}
$$

The point is to define a suitable invariant P.

$$
\begin{array}{l}
|[\\
\textbf{var } x, y : \text{int}; \\
\{x = X \wedge y = Y \wedge x > 0 \wedge y > 0\} \\
\{P\} \\
\textbf{do } x > y \rightarrow x := x - y \ [] \ y > x \rightarrow y := y - x \textbf{ od} \\
\{x = X \textbf{ gcd } Y\} \\
]|
\end{array}
$$

Proof Sketch.

1. Define an invariant P as

$$P: \; x > 0 \wedge y > 0 \wedge x \;\mathbf{gcd}\; y = X \;\mathbf{gcd}\; Y$$

   satisfying $x = X \wedge y = Y \wedge x > 0 \wedge y > 0 \;\Rightarrow P$.

2. Prove:

   - $P \wedge \neg(x > y) \wedge \neg(y > x) \;\Rightarrow\; x = X \;\mathbf{gcd}\; Y$
   - $\{P \wedge (x > y)\} x := x - y \{P\}$
   - $\{P \wedge (y > x)\} y := y - x \{P\}$

3. Show the termination of the repetition.
   - Let $t = x + y$. $t \geq 0$ and t decreases in each step of repetition.
     - $P \wedge (x > y \vee y < x) \Rightarrow t \geq 0$
     - $\{P \wedge (x > y) \wedge t = C\} x := x - y \{t < C\}$
     - $\{P \wedge (y > x) \wedge t = C\} y := y - x \{t < C\}$

Constants

The following is not satisfactory:

$$
\begin{array}{l}
[\![\\
\textbf{var } A, B, x : int; \\
\{A > 0 \wedge B > 0\} \\
\quad gcd \\
\{x = A \; gcd \; B\} \\
]\!]
\end{array}
$$

as $A, B, x := 1, 1, 1$ is a possible solution.

Constant should not be changed.

$$
\begin{array}{l}
[\![\\
\mathbf{con}\ A, B : \mathit{int}; \\
\mathbf{var}\ x : \mathit{int}; \\
\{A > 0 \wedge B > 0\} \\
\quad \mathit{gcd} \\
\{x = A\ \mathit{gcd}\ B\} \\
]\!]
\end{array}
$$

Inner Blocks

$|[$
con $A, B : int; \{A > 0 \wedge B > 0\}$
var $x : int;$
  $|[$
  var $y : int$
  $x, y := A, B;$
  do
    $x > 0 \rightarrow x := x - y$
    $[]y > x \rightarrow y := y - x$
  od
  $\{x = A \;\mathbf{gcd}\; B \wedge y = A \;\mathbf{gcd}\; B\}$
  $]|$
  $\{x = A \;\mathbf{gcd}\; B\}$
$]|$

Used to extend the state (locally) by means of new variables.

$$\{P\}|[\mathbf{var}\ y\ ; S]|\{Q\}$$

is equivalent to

$$\{P\}S\{Q\}$$

provided that y does not occur in both P and Q.

Arrays

Arrays are used to represent a set of variables.

$$f : \mathbf{array}\ [p..q)\ \mathbf{of}\ int$$

defines a program variable f which has as value a function:

$$[p..q) \rightarrow \mathcal{Z}.$$

Exercises

2.1 Show that

$$\{P_0\}S\{Q_0\} \text{ and } \{P_1\}S\{Q_1\}$$

implies

$$\{P_0 \wedge P_1\}S\{Q_0 \wedge Q_1\} \text{ and } \{P_0 \vee P_1\}S\{Q_0 \vee Q_1\}.$$

2.2 Prove

$$\begin{array}{l}
|[\\
\textbf{var } x, y : \mathit{int}; \\
\{x = A \wedge y = B\} \\
x := x - y;\ \ y := x + y;\ \ x := y - x \\
\{x = B \wedge y = A\} \\
]|.
\end{array}$$

2.3 Determine the weakest P such that

$$
\begin{array}{l}
|[\\
\mathbf{var}\ x : \mathit{int}; \\
\{P\} \\
x := x + 1; \\
\mathbf{if}\ x > 0\ \rightarrow\ x := x - 1 \\
\ [\!]\ x < 0 \rightarrow x := x + 2 \\
\ [\!]\ x = 1 \rightarrow \mathit{skip} \\
\mathbf{fi} \\
\{x \geq 1\} \\
]|.
\end{array}
$$

2.4 Prove the correctness of the following program.

$$
\begin{array}{l}
|[\\
\mathbf{var}\ x, y, z : \mathit{int}; \\
\{\mathit{true}\} \\
\mathbf{do}\ x < y \rightarrow x := x + 1 \\
\ [\,]\ y < z \rightarrow y := y + 1 \\
\ [\,]\ z < x \rightarrow z := z + 1 \\
\mathbf{od} \\
\{x = y = z\} \\
]|
\end{array}
$$

2.5 The following problem may be used to compute (non-deterministically) natural numbers x and y such that $x * y = N$. Prove:

$$
\begin{array}{l}
|[\\
\textbf{var } p, x, y, N : \text{int}; \\
\{N \geq 1\} \\
p, x, y := N - 1, 1, 1; \\
\{N = x * y + p\} \\
\textbf{do } p \neq 0 \\
\quad \rightarrow \textbf{if } p \textbf{ mod } x = 0 \rightarrow p, y := p - x, y + 1 \\
\qquad [\,]\ p \textbf{ mod } y = 0 \rightarrow x, p := x + 1, p - y \\
\qquad \textbf{fi} \\
\textbf{od} \\
\{x * y = N\} \\
]|.
\end{array}
$$

2.6 Prove

$$
\begin{array}{l}
|[\\
\textbf{con } N : int \ \{N \geq 0\}; \\
f : \ \textbf{array } [0..N) \textbf{ of } int; \\
\textbf{var } b : bool; \\
\quad |[\\
\quad \textbf{var } n : int; \\
\quad b, n := false, 0; \\
\quad \textbf{do } n \neq N \rightarrow b := b \vee f.n = 0; \ n := n + 1 \textbf{ od} \\
\quad]| \\
\{b \equiv (\exists i : 0 \leq i < N : \ f.i = 0)\} \\
]|.
\end{array}
$$