Discrete Mathematics with Applications I

COMPSCI&SFWRENG 2DM3

McMaster University, Fall 2019

Wolfram Kahl

2019-09-13

An Equational Theory of Integers — Axioms — Fill in the Blanks!

(15.1) **Axiom, Associativity:**

(15.2) **Axiom, Symmetry:**

(15.3) **Axiom, Additive identity:**

(15.4) **Axiom, Multiplicative identity:**

(15.5) **Axiom, Distributivity:**

(15.9) **Axiom, Zero of $\cdot$:**

(15.13) **Axiom, Unary minus:**

(15.14) **Axiom, Subtraction:**

Plan for Today

- **Substitution:**
  - **Inference rule Substitution:** Justifies applying instances of theorems:

$$\begin{array}{ll} & 2 \cdot y \ + \ -(2 \cdot y) \\ = & \langle \text{“Unary minus”}\ a \ + \ -a \ = \ 0 \text{ with ‘}a \ := \ 2 \cdot y\text{‘} \rangle \\ & 0 \end{array}$$

  - **Inference rule Leibniz:** Justifies applying (instances of) **equational** theorems deeper inside expressions:

$$\begin{array}{ll} & 2 \cdot x + 3 \cdot (y \ - \ 5 \cdot (4 \cdot x + 7)) \\ = & \langle \text{“Subtraction”}\ a \ - \ b \ = \ a \ + \ -b \text{ with ‘}a, b \ := \ y, 5 \cdot (4 \cdot x + 7)\text{‘} \rangle \\ & 2 \cdot x + 3 \cdot (y \ + \ -(5 \cdot (4 \cdot x + 7))) \end{array}$$

- **Reasoning about Assignment Commands in Imperative Programs**

$$\{ \ Q[x := E] \ \} \ x := E \ \{ \ Q \ \}$$

Textual Substitution

Let E and R be expressions and let x be a variable. We write:

$$E[x := R] \qquad \text{or} \qquad E^x_R$$

to denote an expression that is the same as E but with all occurrences of x replaced by (R).

Examples:

Expression	Result	Unnecessary parentheses removed
$x[x := z+2]$	$(z+2)$	$z+2$
$(x+y)[x := z+2]$	$((z+2)+y)$	$z+2+y$
$(x \cdot y)[x := z+2]$	$((z+2) \cdot y)$	$(z+2) \cdot y$
$x+y[x := z+2]$	$x+y$	$x+y$

Note: Substitution $[x := R]$ is a **highest precedence** postfix operator

Simultaneous Substitution:

$$\begin{array}{ll} & (x+y)[x,y := y-3, z+2] \\ = & \langle \text{ performing substitution } \rangle \\ & ((y-3)+(z+2)) \\ = & \langle \text{ Reflexivity of } = \text{ — removing unnecessary parentheses } \rangle \\ & y-3+z+2 \end{array}$$

Sequential Substitution:

$$\begin{array}{ll} & (x+y)[x := y-3][y := z+2] \\ = & \langle \text{ adding parentheses for clarity } \rangle \\ & ((x+y)[x := y-3])[y := z+2] \\ = & \langle \text{ performing inner substitution } \rangle \\ & (((y-3)+y))[y := z+2] \\ = & \langle \text{ performing outer substitution } \rangle \\ & ((((z+2)-3)+(z+2))) \\ = & \langle \text{ removing unnecessary parentheses } \rangle \\ & z+2-3+z+2 \end{array}$$

Inference Rule: Substitution

(1.1) **Substitution:** $\dfrac{E}{E[x := R]}$

Example:

If $a+0=a$ is a theorem, $\boxed{\text{“Identity of +”}}$

then $3 \cdot b + 0 = 3 \cdot b$ is also a theorem. $\boxed{\text{“Identity of +” with ‘}a := 3 \cdot b\text{‘}}$

$$\frac{a+0=a}{(a+0=a)[a := 3 \cdot b]} \qquad\qquad \frac{a+0=a}{3 \cdot b + 0 = 3 \cdot b}$$

Example:

$$\frac{z \geq x \uparrow y \quad \equiv \quad z \geq x \;\wedge\; z \geq y}{x+y \geq x \uparrow y \quad \equiv \quad x+y \geq x \;\wedge\; x+y \geq y}$$

What is an Inference Rule?

$$\frac{\text{premise}_1 \quad \ldots \quad \text{premise}_n}{\text{conclusion}}$$

- **If all the premises are theorems,**
  then the conclusion is a theorem.
- A thereom is a "proved truth"
- The premises are also called hypotheses.
- The conclusion and each premise all have to be Boolean
- **Axioms** are inference rules with zero premises

Inference Rule Scheme: Substitution

(1.1) **Substitution:** $\frac{E}{E[x := R]}$

Really an **inference rule scheme**:
works for **every combination** of

- expression E,
- variable x, and
- expression R.

$$\frac{a + 0 = a}{3 \cdot b + 0 = 3 \cdot b}$$

Example 1:
If $a + 0 = a$ is a theorem,
then $3 \cdot b + 0 = 3 \cdot b$ is also a theorem.

- expression E is $a + 0 = a$
- the variable x substituted into is a
- the substituted expression R is $3 \cdot b$

Inference Rule Scheme: Substitution

(1.1) **Substitution:** $\frac{E}{E[x := R]}$

Really an **inference rule scheme**:
works for **every combination** of

- expression E,
- variable x, and
- expression R.

Example 2:

$$\frac{a \cdot (b + c) = a \cdot b + a \cdot c}{(2 + x) \cdot (b + c) = (2 + x) \cdot b + (2 + x) \cdot c}$$

If $a \cdot (b + c) = a \cdot b + a \cdot c$ is a theorem,
then $(2 + x) \cdot (b + c) = (2 + x) \cdot b + (2 + x) \cdot c$ is also a theorem.

- expression E is $a \cdot (b + c) = a \cdot b + a \cdot c$
- the variable x substituted into is a
- the substituted expression R is $2 + x$

Inference Rule: Substitution

(1.1) **Substitution:** $\dfrac{E}{E[x := R]}$

Really an **inference rule scheme**:
works for **every combination** of

- expression E,
- variable **list** x, and
- **corresponding** expression **list** R.

Example:

If $x + y = y + x$ is a theorem,
then $b + 3 = 3 + b$ is also a theorem.

- expression E is $x + y = y + x$
- variable list x is x, y
- corresponding expression list R is $b, 3$

Logical Definition of Equality

Two **axioms** (i.e., postulated as theorems):

- (1.2) **Reflexivity of =:** $x = x$
- (1.3) **Symmetry of =:** $(x = y) = (y = x)$

Two **inference rule schemes**:

- (1.4) **Transitivity of =:** $\dfrac{X = Y \qquad Y = Z}{X = Z}$
- (1.5) **Leibniz:** $\dfrac{X = Y}{E[z := X] = E[z := Y]}$

— the rule of "replacing equals for equals"

Using Leibniz' Rule in (15.21)

Given: (15.20) $-a = (-1) \cdot a$

Prove: (15.21) $(-a) \cdot b = a \cdot (-b)$

$$\boxed{\dfrac{X = Y}{E[z := X] = E[z := Y]}}$$

Proving (15.21) $(-a) \cdot b = a \cdot (-b)$**:**

$$
\begin{array}{ll}
 & (-a) \cdot b \\
= & \langle \text{(15.20) } \textbf{(via Leibniz (1.5) with } E \textbf{ chosen as } z \cdot b) \rangle \\
 & ((-1) \cdot a) \cdot b \\
= & \langle \text{Associativity (15.1) and Symmetry (15.2) of } \cdot \rangle \\
 & a \cdot ((-1) \cdot b) \\
= & \langle \text{(15.20)} \rangle \\
 & a \cdot (-b)
\end{array}
$$

Using Leibniz together with Substitution in (15.21)

Given: (15.20) $-a = (-1) \cdot a$

Prove: (15.21) $(-a) \cdot b = a \cdot (-b)$

$$\frac{X = Y}{E[z := X] = E[z := Y]}$$

Proving (15.21) $(-a) \cdot b = a \cdot (-b)$**:**

$(-a) \cdot b$

$=$ ⟨ (15.20) (via Leibniz (1.5) with E chosen as $z \cdot b$) ⟩

$((-1) \cdot a) \cdot b$

$=$ ⟨ Associativity (15.1) and Symmetry (15.2) of $\cdot$ ⟩

$a \cdot ((-1) \cdot b)$

$=$ ⟨ (15.20) with $a := b$ (via Leibniz (1.5) with E chosen as $a \cdot z$) ⟩

$a \cdot (-b)$

Combining Leibniz' Rule with Substitution

(1.5) **Leibniz:** $\dfrac{X = Y}{E[z := X] = E[z := Y]}$

(1.1) **Substitution:** $\dfrac{F}{F[v := R]}$

Using Leibniz' rule:

$E[z := X]$

$=$ ⟨ $X = Y$ ⟩

$E[z := Y]$

Using them together:

$E[z := X[v := R]]$

$=$ ⟨ $X = Y$ ⟩

$E[z := Y[v := R]]$

Justification:

$$\dfrac{\dfrac{X = Y}{X[v := R] = Y[v := R]}\ \text{Substitution (1.1)}}{E[z := X[v := R]] = E[z := Y[v := R]]}\ \text{Leibniz (1.5)}$$

Expression Evaluation

- $2 \cdot 3 + 4$
- $2 \cdot (3 + 4)$
- $2 \cdot y + 4$
- A **state** is a list of variables with associated values. E.g.:

$$s_1 = \langle (x, 5), (y, 6) \rangle$$

- **Evaluating an expression in a state**:
  "Replace variables with their values; then evaluate":
- $x - y + 2$ in state s_1
  $\longrightarrow \quad 5 - 6 + 2 \quad \longrightarrow \quad (5 - 6) + 2 \quad \longrightarrow \quad (-1) + 2 \quad \longrightarrow \quad 1$
- $x \cdot 2 + y$
- $x \cdot (2 + y)$
- $x \cdot (z + y)$

Precondition-Postcondition Specifications

- *Recall:* A **state** is a list of variables with associated values.
- Before and after execution of each command in an imperative program, the program variables with their current values make up such a **state**.
- Boolean expressions in which program variables occur, e.g., $(x = 5 \wedge y = 3)$, will be true or false in each state, and can be used for program specification.
- Program correctness statement in LADM (and much current use):

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

  This is called a "Hoare triple".
- **Meaning:** If command C is started in a state in which the **precondition** P holds (evaluates to *true*),
  then it will terminate in a state in which the **postcondition** Q holds.
- Hoare's original notation: $P\ \{\, C \,\}\ Q$
- **Dynamic logic** notation (will be used in CALCCHECK):

  $$P \Rrightarrow [\, C \,]\ Q$$

Correctness of Assignment Commands

- *Recall:* Hoare triple: $\{\, P \,\}\ C\ \{\, Q \,\}$
- **Dynamic logic** notation (will be used in CALCCHECK): $P \Rrightarrow [\, C \,]\ Q$
- **Meaning:** If command C is started in a state in which the **precondition** P holds, then it will terminate in a state in which the **postcondition** Q holds.
- **Assignment Axiom:** $\{\, Q[x := E] \,\}\ x := E\ \{\, Q \,\}$ $\qquad Q[x := E] \Rrightarrow [\, x := E \,]\ Q$
- **Examples:**
  - $(x = 5)[x := x + 1] \quad \Rrightarrow [\, x := x + 1 \,] \quad x = 5$
  - $$\begin{array}{ll} & true \\ \equiv & \langle \text{Zero of } \vee \rangle \\ & 1 = 0 \vee true \\ \equiv & \langle \text{Reflexivity of } = \rangle \\ & 1 = 0 \vee 1 = 1 \\ \equiv & \langle \text{Substitution} \rangle \\ & (x = 0 \vee x = 1)[x := 1] \\ \Rrightarrow [\, x := 1 \,] & \langle \text{Assignment} \rangle \\ & x = 0 \vee x = 1 \end{array}$$

Sequential Composition of Commands

```
Primitive inference rule “SEQ”:
   `{ P } C₁ { Q }` , `{ Q } C₂ { R }`
  ⊢───────────────────────────────────
           `{ P } C₁ ; C₂ { R }`
```

```
Primitive inference rule “Sequence”:
     `P  ⇛[ C₁ ]  Q`,   `Q  ⇛[ C₂ ]  R`
  ⊢──────────────────────────────────────
       `P  ⇛[ C₁ ; C₂ ]  R`
```

- Activated as transitivity rule
- Therefore used implicitly in calculations, e.g., proving $P \Rrightarrow [\, C_1 ⨾ C_2 \,]\ R$ by:

$$\begin{array}{ll} & P \\ \Rrightarrow [\, C_1 \,] & \langle \ldots \rangle \\ & Q \\ \Rrightarrow [\, C_2 \,] & \langle \ldots \rangle \\ & R \end{array}$$

- No need to refer to this rule explicitly.

Example Proof for a Sequence of Assignments:

Fact: $x = 5 \Rightarrow[\, (y := x + 1 \,;\, x := y + y) \,]\, x = 12$

Proof:

$x = 5$

$\equiv \langle$ "Cancellation of +" $\rangle$

$x + 1 = 5 + 1$

$\equiv \langle$ Fact `$5 + 1 = 6$` $\rangle$

$x + 1 = 6$

$\equiv \langle$ Substitution $\rangle$

$(y = 6)[y \coloneqq x + 1]$

$\Rightarrow[\, y := x + 1 \,]$ $\langle$ "Assignment $\Rightarrow[]$" $\rangle$

$y = 6$

$\equiv \langle$ "Cancellation of $\cdot$" with Fact `$2 \neq 0$` $\rangle$

$2 \cdot y = 2 \cdot 6$

$\equiv \langle$ Evaluation $\rangle$

$(1 + 1) \cdot y = 12$

$\equiv \langle$ "Distributivity of $\cdot$ over +" $\rangle$

$1 \cdot y + 1 \cdot y = 12$

$\equiv \langle$ "Identity of $\cdot$" $\rangle$

$y + y = 12$

$\equiv \langle$ Substitution $\rangle$

$(x = 12)[x \coloneqq y + y]$

$\Rightarrow[\, x := y + y \,]$ $\langle$ "Assignment $\Rightarrow[]$" $\rangle$

$x = 12$