Discrete Mathematics with Applications I

COMPSCI&SFWRENG 2DM3

McMaster University, Fall 2019

Wolfram Kahl

2019-10-09

Formalise:

- The sum of the first n odd natural numbers is equal to n^2.
- The product of k consecutive positive integers is always divisible by $k!$.
- For some non-trivial interval of positive integers, the product is less than the sum.

Plan for Today

- Conditional expressions: `if_then_else_fi`
- **Textbook Chapters 8 and 9: Quantification and Predicate Logic**
  - **Sums and Products** — continued
  - **Universal and Existential Quantification**

Conditional Commands

- Pascal:

```
if condition then
  statement₁
else
  statement₂
```

- Ada:

```
if condition then
  statement₁
else
  statement₂
end if;
```

- C/Java:

```
if (condition)
  statement₁
else
  statement₂
```

- Python:

```
if condition:
  statement₁
else:
  statement₂
```

- sh:

```
if condition
then
  statement₁
else
  statement₂
fi
```

Conditional Expressions — at Any Type

- Haskell/Elm:

  `if` *condition* `then` $expr_1$ `else` $expr_2$

- C/Java:

  condition `?` $expr_1$ `:` $expr_2$

- Python:

  $expr_1$ `if` *condition* `else` $expr_2$

- CALCCHECK (Exercise 6.4):

  if *condition* then $expr_1$ else $expr_2$ fi

  (*Library-defined* mixfix operator if_then_else_fi)

Using Conditional Expressions

Exercise 6.4 introduces the *Library-defined* mixfix operator if_then_else_fi for conditional expressions:

$$\frac{condition : \mathbb{B} \qquad expr_1 : t \qquad expr_2 : t}{\text{if } condition \text{ then } expr_1 \text{ else } expr_2 \text{ fi } : t}$$

```
Declaration: fact : ℕ → ℕ
Axiom “Definition of `fact`”:
    fact n  =  if n = 0
               then 1
               else n · fact (pred n)
               fi

Declaration: fib : ℕ → ℕ
Axiom “Definition of `fib`”:
    fib n  =  if n < 2 then 1
              else fib (pred n) + fib (pred (pred n))
              fi
```

Reasoning about Conditional Expressions

Exercise 6.4 introduces the *Library-defined* mixfix operator if_then_else_fi for conditional expressions:

$$\frac{\mathit{condition} : \mathbb{B} \qquad \mathit{expr}_1 : t \qquad \mathit{expr}_2 : t}{\text{if } \mathit{condition} \text{ then } \mathit{expr}_1 \text{ else } \mathit{expr}_2 \text{ fi} \ : t}$$

Definition allows reasoning about conditionals occurring "deep inside P":

```
Axiom "Definition of `if`"   "`if` to ∧":
    P[z ≔ if b then x else y fi]
  ≡ (b ⇒ P[z ≔ x]) ∧ (¬ b ⇒ P[z ≔ y])
```

Use this after "backwards Substitution":

```
    (u = if b then x else y fi)
  ≡⟨ Substitution ⟩
    (u = z)[z ≔ if b then x else y fi]
  ≡⟨ "Definition of `if`" ⟩
    (b ⇒ (u = z)[z ≔ x]) ∧ (¬ b ⇒ (u = z)[z ≔ y])
  ≡⟨ Substitution ⟩
    (b ⇒ u = x) ∧ (¬ b ⇒ u = y)
```

General Shape of Sum and Product Quantifications

$(\sum x : t_1;\ y,z : t_2 \mid R \bullet E)$ $\qquad\qquad$ $(\prod x : t_1;\ y,z : t_2 \mid R \bullet E)$

- Any number of **variables** x, y, z can be quantified over
- The quantified variables may have **type annotations** (which act as **type declarations**)
- Expression $R : \mathbb{B}$ is the **range** of the quantification
- Expression E is the **body** of the quantification
- E will have a number type ($\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$)
- Both R and E may refer to the **quantified variables** x, y, z
- The type of the whole quantification expression is the type of E.
- The range defaults to *true*: $\quad (\sum x \bullet E) = (\sum x \mid \mathit{true} \bullet E)$
  $\quad (\prod x \bullet E) = (\prod x \mid \mathit{true} \bullet E)$

  ("syntactic sugar", covered by reflexivity of =)

LADM/CalcCheck Quantification Notation

Conventional sum quantification notation: $\displaystyle\sum_{i=1}^{n} e \quad = \quad e[i := 1] + \ldots + e[i := n]$

The textbook uses a different, but systematic **linear** notation:

$$(\sum i \mid 1 \le i \le n : e) \qquad \text{or} \qquad (+ i \mid 1 \le i \le n : e)$$

We use a variant with a "spot" "•" instead of the colon ":" and only use "big" operators:

$$(\sum i \mid 1 \le i \le n \bullet e)$$

Reasons for using this linear quantification notation:

- Clearly delimited introduction of **quantified variables** (**dummies**)
- **Arbitrary** Boolean expressions can define the **range** of the quantified variables
  $$(\sum i \mid 1 \le i \le 7 \wedge \mathit{even}\ i \bullet i) = 2 + 4 + 6$$
- Extends easily to multiple quantified variables:
  $$(\sum i, j : \mathbb{Z} \mid 1 \le i < j \le 4 \bullet i/j) = 1/2 + 1/3 + 1/4 + 2/3 + 2/4 + 3/4$$

The sum of the first n odd natural numbers is equal to n^2

```
Theorem “Odd-number sum”:
    (∑ i : ℕ ❙ i < n • suc i + i) = n · n
Proof:
  By induction on `n : ℕ`:
    Base case:







    Induction step:
```

The sum of the first n odd natural numbers is equal to n^2

```
Theorem “Odd-number sum”:
    (∑ i : ℕ ❙ i < n • suc i + i) = n · n
Proof:
  By induction on `n : ℕ`:
    Base case:
        (∑ i : ℕ ❙ i < 0 • suc i + i)
      =⟨ ? ⟩



      =⟨ ? ⟩
        0 · 0
    Induction step:
        (∑ i : ℕ ❙ i < suc n • suc i + i)
      =⟨ ? ⟩





      =⟨ ? ⟩
        suc n · suc n
```

Empty Range Axioms

(8.13) **Axiom, Empty Range**:

$$(\sum x \mid false \bullet E) \quad = \quad 0$$

$$(\prod x \mid false \bullet E) \quad = \quad 1$$

Manipulating Ranges

(8.23) **Theorem Split off term**: For $n : \mathbb{N}$ and dummies $i : \mathbb{N}$,

$$(\sum i \mid 0 \le i < n+1 \bullet P) \;=\; (\sum i \mid 0 \le i < n \bullet P) \;+\; P[i := n]$$

$$(\sum i \mid 0 \le i < n+1 \bullet P) \;=\; P[i := 0] \;+\; (\sum i \mid 0 < i < n+1 \bullet P)$$

- Typical use: Verification of loops
- Generalisation: $\mathbb{N} \longrightarrow \mathbb{Z}$, $\quad 0 \longrightarrow m : \mathbb{Z}$ (with $m \le n$)

The following work both with $m, n, i : \mathbb{N}$ and with $m, n, i : \mathbb{Z}$:

Theorem: Split off term from top:

$$m \le n \quad \Rightarrow \quad (\sum i \mid m \le i < n+1 \bullet P) = (\sum i \mid m \le i < n \bullet P) \;+\; P[i := n]$$

Theorem: Split off term from bottom:

$$m \le n \quad \Rightarrow \quad (\sum i \mid m \le i < n+1 \bullet P) = P[i := m] \;+\; (\sum i \mid m+1 \le i < n+1 \bullet P)$$

Disjoint Range Split

(8.16) **Axiom, Range Split:**

$$(\Sigma x \mid Q \vee R \bullet P) = (\Sigma x \mid Q \bullet P) + (\Sigma x \mid R \bullet P)$$

provided $Q \wedge R = \mathit{false}$ and each sum is defined.

(8.16) **Axiom, Range Split:**

$$(\Pi x \mid Q \vee R \bullet P) = (\Pi x \mid Q \bullet P) \cdot (\Pi x \mid R \bullet P)$$

provided $Q \wedge R = \mathit{false}$ and each product is defined.

That is: Summing up over a large range can be done by adding the results of summing up two disjoint and complementary subranges.

$\Longrightarrow$ "**Divide and conquer**" algorithm design pattern

DIVIDE ET IMPERA

— Gaius Julius Caesar

Proving Split-off Term

(8.16) **Axiom, Range Split:**

$$(\Sigma x \mid Q \vee R \bullet P) = (\Sigma x \mid Q \bullet P) + (\Sigma x \mid R \bullet P)$$

provided $Q \wedge R = \mathit{false}$ and each sum is defined.

```
Theorem “Split off term” “Split off term at top”:
    (∑ i : ℕ ❙ i < suc n • E) = (∑ i : ℕ ❙ i < n • E) + E[i ≔ n]
```

Axioms for One-element Ranges

(8.14) **Axiom, One-point Rule:** Provided $\neg occurs('x', 'D')$,

$$(\sum x \mid x = D \bullet E) \quad = \quad E[x := D]$$

$$(\prod x \mid x = D \bullet E) \quad = \quad E[x := D]$$

Example:

$$\begin{aligned}
& (\sum i : \mathbb{N} \bullet 5 + 2 \cdot i < 7 \mid 5 + 7 \cdot i) \\
= \; & \langle \ldots \rangle \\
& (\sum i : \mathbb{N} \bullet i = 0 \mid 5 + 7 \cdot i) \\
= \; & \langle \text{One-point rule} \rangle \\
& (5 + 7 \cdot i)[i := 0] \\
= \; & \langle \text{Substitution} \rangle \\
& 5 + 7 \cdot 0
\end{aligned}$$

Important Quantification Laws I

(8.13) **Empty Range**:

$$(\Sigma x \mid false \bullet E) \quad = \quad 0$$
$$(\Pi x \mid false \bullet E) \quad = \quad 1$$

(8.14) **One-point Rule:** Provided $\neg occurs('x', 'E')$,

$$(\Sigma x \mid x = E \bullet F) \quad \equiv \quad F[x := E]$$
$$(\Pi x \mid x = E \bullet F) \quad \equiv \quad F[x := E]$$

(8.16) **Disjoint range split:** Provided $Q \wedge R = false$ and each sum is defined:

$$(\Sigma x \mid Q \vee R \bullet P) \quad = \quad (\Sigma x \mid Q \bullet P) + (\Sigma x \mid R \bullet P)$$
$$(\Pi x \mid Q \vee R \bullet P) \quad = \quad (\Pi x \mid Q \bullet P) \cdot (\Pi x \mid R \bullet P)$$

(8.23) **Split off term**: For $n : \mathbb{N}$ and dummies $i : \mathbb{N}$,

$$(\Sigma i \mid 0 \le i < n + 1 \bullet P) \quad = \quad (\sum i \mid 0 \le i < n \bullet P) + P[i := n]$$

$$(\Sigma i \mid 0 \le i < n + 1 \bullet P) \quad = \quad P[i := 0] + (\sum i \mid 0 < i < n + 1 \bullet P)$$

Universal and Existential Quantification

$(\forall x \bullet P)$

- "For all x, we have P"

$(\forall x \mid R \bullet P)$

- "For all x with R, we have P"

$(\exists x \bullet P)$

- "There exists an x such that P (holds)"
- "For some x, we have P"

$(\exists x \mid R \bullet P)$

- "There exists an x with R such that P (holds)"
- "For some x with R, we have P"

Universal and Existential Quantification

$(\forall\, x \bullet p(x))$

- "For all x, we have $p(x)$"

$(\forall\, x \mid r(x) \bullet p(x))$

- "For all x with $r(x)$, we have $p(x)$"

$(\exists\, x \bullet p(x))$

- "There exists an x such that $p(x)$ (holds)"
- "For some x, we have $p(x)$"

$(\exists\, x \mid r(x) \bullet p(x))$

- "There exists an x with $r(x)$ such that $p(x)$ (holds)"
- "For some x with $r(x)$, we have $p(x)$"

Expanding Universal and Existential Quantification

Universal quantification ($\forall$) is
"conjunction ($\wedge$) with arbitrarily many conjuncts":

$$\begin{aligned} & (\forall\, i \mid 1 \le i < 3 \bullet i \cdot d \ne 6) \\ = \; & \langle \text{ Quantification expansion, substitution } \rangle \\ & 1 \cdot d \ne 6 \quad \wedge \quad 2 \cdot d \ne 6 \end{aligned}$$

Existential quantification ($\exists$) is
"disjunction ($\vee$) with arbitrarily many disjuncts":

$$\begin{aligned} & (\exists\, i \mid 0 \le i < 21 \bullet b[i] = 0) \\ = \; & \langle \text{ Quantification expansion, substitution } \rangle \\ & b[0] = 0 \quad \vee \quad b[1] = 0 \quad \vee \quad \ldots \quad \vee \quad b[20] = 0 \end{aligned}$$

General Shape of Universal and Existential Quantifications

$$(\forall\, x : t_1 ; y, z : t_2 \mid R \bullet P)$$

$$(\exists\, x : t_1 ; y, z : t_2 \mid R \bullet P)$$

- Any number of **variables** x, y, z can be quantified over
- The quantified variables may have **type annotations** (which act as **type declarations**)
- $R : \mathbb{B}$ is the **range** of the quantification
- $P : \mathbb{B}$ is the **body** of the quantification
- Both R and P may refer to the **quantified variables** x, y, z
- The type of the whole quantification expression is $\mathbb{B}$.
- The range defaults to *true*: $(\forall\, x \bullet P) \;=\; (\forall\, x \mid \mathit{true} \bullet P)$
  $(\exists\, x \bullet P) \;=\; (\exists\, x \mid \mathit{true} \bullet P)$

  ("syntactic sugar", covered by reflexivity of $\equiv$)