Discrete Mathematics with Applications I

COMPSCI&SFWRENG 2DM3

McMaster University, Fall 2019

Wolfram Kahl

2019-09-04

Mathematical Modelling

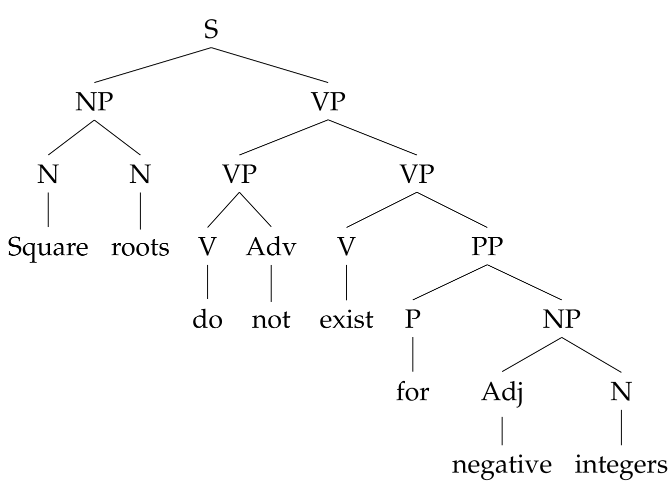
Textbook p. 2: How to specify an algorithm to compute b, an integer approximation to $\sqrt{n}$ for some integer n?

- Square roots do not exist for negative integers!
  Therefore, the algorithm must only be used for non-negative n.
  Precondition: $n \geq 0$
- To compute *an* approximation???
  42 is *an* approximation of $\sqrt{1000}$!
  "Reasonable" approximations (candidates for the *postcondition*):
  - $b^2 \leq n \leq (b+1)^2$
  - $abs(b^2 - n) \leq abs((b+1)^2 - n)$ and $abs(b^2 - n) \leq abs((b-1)^2 - n)$
  - $(b-1)^2 \leq n \leq b^2$

Now step back, and do "grammatical analysis"!

Grammatical Analysis: Sentence Structure Trees

Square roots do not exist for negative integers.



Mathematical Modelling uses Mathematical Expressions

Textbook p. 2: How to specify an algorithm to compute b, an integer approximation to $\sqrt{n}$ for some integer n?

- Square roots do not exist for negative integers!
  Therefore, the algorithm must only be used for non-negative n.
  Precondition: $n > 0$
- To compute *an* approximation??? — 42 is *an* approximation of $\sqrt{1000}$!
  "Reasonable" approximations (candidates for the *postcondition*):
  - $b^2 \le n \le (b+1)^2$
  - $abs(b^2 - n) \le abs((b+1)^2 - n)$ and $abs(b^2 - n) \le abs((b-1)^2 - n)$
  - $(b-1)^2 \le n \le b^2$

Now step back, and do "grammatical analysis"!

- How is all that math put together?
- What are the different kinds of atoms ("words")?
- What are the different kinds of composite structures ("phrases")?
- What are the rules for analysis/synthesis of composite structures?

Syntax of Conventional Mathematical Expressions

- A **constant** (e.g., 231) or **variable** (e.g., x) is an expression
- If E is an expression, then (E) is an expression
- If $\circ$ is a **unary prefix operator** and E is an expression, then $\circ E$ is an expression, with operand E.

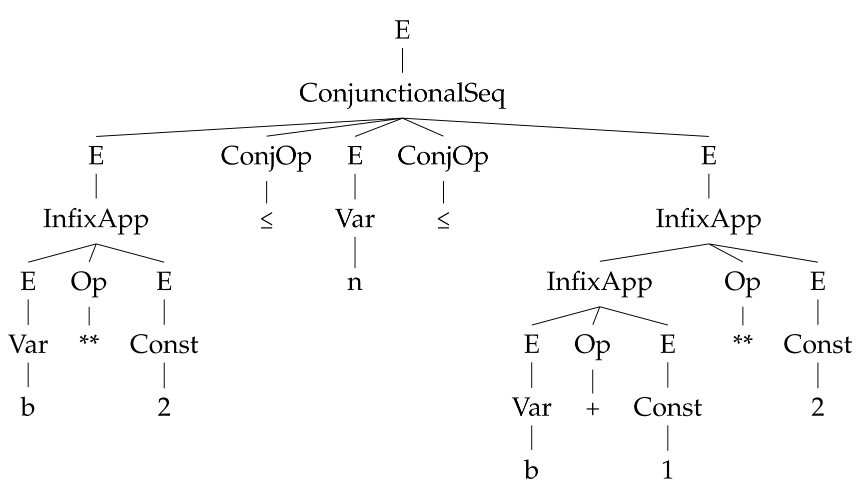
  For example, the negation symbol $-$ is used as a unary prefix operator, so -5 is an expression.
- If $\otimes$ is a **binary infix operator** and D and E are expressions, then $D \otimes E$ is an expression, with operands D and E.

  For example, the symbols $+$ and $\cdot$ are binary infix operators, so $1+2$ and $(-5)\cdot(3+x)$ are expressions.

Textbook 1.1, p. 7

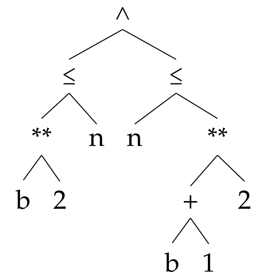
Grammatical Analysis for Mathematical Expression

$b^2 \le n \le (b+1)^2$



Term Tree Presentation of Mathematical Expression

$$b^2 \le n \le (b+1)^2$$



We write strings, but we think trees.

All the rules we have for implicit parentheses only serve to encode the tree structure.

Syntax of Conventional Mathematical Expressions

- A **constant** (e.g., 231) or **variable** (e.g., x) is an expression
- If E is an expression, then (E) is an expression
- If $\circ$ is a **unary prefix operator** and E is an expression, then $\circ E$ is an expression, with operand E.
- If $\otimes$ is a **binary infix operator** and D and E are expressions, then $D \otimes E$ is an expression, with operands D and E.

Therefore, each expression (for now) is **exactly one** of the following alternatives:

- **either some constant**
- **or some variable**
- **or some simpler expression** in parentheses
- **or** the application of **some unary prefix operator** to **some simpler expression**
- **or** the application of **some binary infix operator** to **two simpler expressions**

Recognising the Syntax of Conventional Mathematical Expressions

- A **constant** (e.g., 231) or **variable** (e.g., x) is an expression
- If E is an expression, then (E) is an expression
- If $\circ$ is a **unary prefix operator** and E is an expression, then $\circ E$ is an expression, with operand E.
- If $\otimes$ is a **binary infix operator** and D and E are expressions, then $D \otimes E$ is an expression, with operands D and E.

```
Calculation:
    7 · 8
  =⟨ Fact `8 = 7 + 1` ⟩
    7 · (7 + 1)
  =⟨ Fact `7 = 10 - 3` ⟩
    (10 - 3) · (7 + 1)
  =⟨ “Distributivity of · over +” ⟩
    (10 - 3) · 7 + (10 - 3) · 1
  =⟨ “Distributivity of · over -” ⟩
    10 · 7 - 3 · 7 + 10 · 1 - 3 · 1
  =⟨ “Identity of ·” ⟩
    10 · 7 - 3 · 7 +   10    -   3
  =⟨ Fact `3 · 7 = 21` ⟩
    10 · 7 -    21  +   10    -   3
  =⟨ Fact `10 · 7 = 70` ⟩
    70     -    21  +   10    -   3
  =⟨ Fact `10 - 3 = 7` ⟩
    70     -    21  +         7
  =⟨ Fact `21 + 7 = 28` ⟩
    70     -        28
  =⟨ Fact `70 - 28 = 42` ⟩
    42
```

Which are expressions?

1. $6 \cdot 7 - 8 \cdot 9$ $\surd$
2. $-3 - -4 \cdot -5$ — really bad style
3. $3 \cdot +7$ $\surd$ — we may use $+$ as a unary prefix operator.
4. $x + \cdot 11$ $\times$
5. $21 + 22 \cdot 23$ $\surd$
6. $31 + 32 \cdots 39$ $\times$

Why is this an expression?

$$2 \cdot 3 + 4$$

- If $\otimes$ is a **binary infix operator** and D and E are expressions, then $D \otimes E$ is an expression, with operands D and E.

- **or** the application of **some binary infix operator** to **two simpler expressions**

Which expression is it? Why?

$\Longrightarrow$ The multiplication operator $\cdot$ has higher **precedence** than the addition operator $+$.

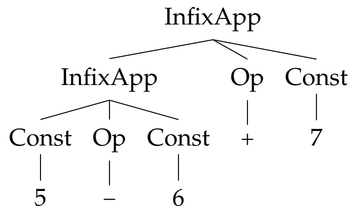
Table of Precedences

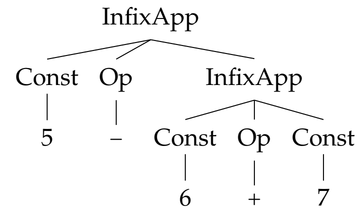
- $[x := e]$ (textual substitution) **(highest precedence)**
- $.$ (function application)
- unary prefix operators $+, -, \neg, \#, \sim, \mathcal{P}$
- $**$
- $\cdot$ $/$ $\div$ **mod** **gcd**
- $+$ $-$ $\cup$ $\cap$ $\times$ $\circ$ $\bullet$
- $\downarrow$ $\uparrow$
- $\#$
- $\lhd$ $\rhd$ $\hat{}$
- $=$ $<$ $>$ $\in$ $\subset$ $\subseteq$ $\supset$ $\supseteq$ $|$ (conjunctional)
- $\vee$ $\wedge$
- $\Rightarrow$ $\Leftarrow$
- $\equiv$ **(lowest precedence)**

All non-associative binary infix operators associate to the left, except $**$, $\lhd$, $\Rightarrow$, which associate to the right.

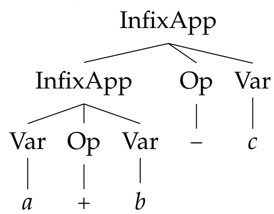
Why are these expressions? Which expressions are these?

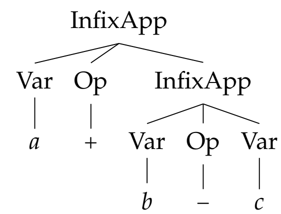
1. $5 - 6 + 7$





2. $a + b - c$





The operators $+$ and $-$ **associate to the left**, also mutually.

Which are expressions? (ctd.)

1. $x + z - 5 - (-3 \cdot y)$ $\surd$
2. $f\ z + 3$ $\times$ — LADM: juxtaposition not used as operator
   $\surd$ — CALCCHECK: juxtaposition used as function application
3. $5 \cdot -(3 \cdot y)$ $\surd$
4. $5 \cdot (-3 \cdot y)$ $\surd$
5. $5(-3 \cdot y)$ $\times$(5 is not a function)
6. $5 = (-3 \cdot y)$ $\surd$
7. $5 = -3 \cdot y$ $\surd$
8. $5 - = 3 \cdot y$ $\times$

Associativity versus Association

- If we write $a + b + c$, there is no need to discuss whether we mean $(a + b) + c$ or $a + (b + c)$, because they are the same:

  $(a + b) + c = a + (b + c)$ $\boxed{\text{“+” is associative}}$

- If we write $a - b - c$, we mean $(a - b) - c$:

  $\boxed{\text{“–” associates to the left}}$ $9 - (5 - 2) \neq (9 - 5) - 2$

- If we write a^{b^c}, we mean $a^{(b^c)}$:

  $\boxed{\text{exponentiation associates to the right}}$ $2^{(3^2)} \neq (2^3)^2$

- If we write $a ** b ** c$, we mean $a ** (b ** c)$:

  $\boxed{\text{“**” associates to the right}}$

- If we write $a \Rightarrow b \Rightarrow c$, we mean $a \Rightarrow (b \Rightarrow c)$:

  $\boxed{\text{“}\Rightarrow\text{” associates to the right}}$ $\mathsf{F} \Rightarrow (\mathsf{T} \Rightarrow \mathsf{F}) \neq (\mathsf{F} \Rightarrow \mathsf{T}) \Rightarrow \mathsf{F}$

Mathematical Expressions, Terms, Formulae ...

"Expression" is not the only word used for this kind of concept.

Related terminology:

- Both "term" and "expression" are frequently used names for the same kind of concept.
- The textbook's "expression" subsumes both "term" and "formula" of conventional first-order predicate logic.

Remember:

- Expressions are **understood** as tree-structures

  — "*abstract syntax*"
- Expressions are **written** as strings

  — "*concrete syntax*"
- Parentheses, precedences, and association rules only serve to disambiguate the encoding of trees in strings.

Truth Values

Boolean constants/values: *false*, *true*

The type of Boolean values: $\mathbb{B}$

— This is the type of propositions, for example: $(x = 1) : \mathbb{B}$

— For any type t, equality _=_ can be used on expressions of that type: $\_{=}\_ : t \to t \to \mathbb{B}$

Boolean operators:

- $\neg\_ : \mathbb{B} \to \mathbb{B}$ — negation, complement, "logical not"
- $\_\wedge\_ : \mathbb{B} \to \mathbb{B} \to \mathbb{B}$ — conjunction, "logical and"
- $\_\vee\_ : \mathbb{B} \to \mathbb{B} \to \mathbb{B}$ — disjunction, "logical or"
- $\_\Rightarrow\_ : \mathbb{B} \to \mathbb{B} \to \mathbb{B}$ — implication, "implies", "if ... then ..."
- $\_\equiv\_ : \mathbb{B} \to \mathbb{B} \to \mathbb{B}$ — equivalence, "if and only if", "iff"
- $\_\not\equiv\_ : \mathbb{B} \to \mathbb{B} \to \mathbb{B}$ — inequivalence, "exclusive or"

Table of Precedences

- $[x := e]$ (textual substitution) **(highest precedence)**
- . (function application)
- unary prefix operators $+$, $-$, $\neg$, $\#$, $\sim$, $\mathcal{P}$
- $**$
- $\cdot$ $/$ $\div$ **mod** **gcd**
- $+$ $-$ $\cup$ $\cap$ $\times$ $\circ$ $\bullet$
- $\downarrow$ $\uparrow$
- $\#$
- $\triangleleft$ $\triangleright$ $\hat{}$
- $=$ $\neq$ $<$ $>$ $\in$ $\subset$ $\subseteq$ $\supset$ $\supseteq$ $|$ (conjunctional)
- $\vee$ $\wedge$
- $\Rightarrow$ $\not\Rightarrow$ $\Leftarrow$ $\not\Leftarrow$
- $\equiv$ $\not\equiv$ **(lowest precedence)**

All non-associative binary infix operators associate to the left, except $**$, $\triangleleft$, $\Rightarrow$, which associate to the right.

Some Laws for the Boolean Operators

(3.12) **Double negation**: $\neg\neg p \equiv p$
(3.36) **Symmetry of $\wedge$**: $p \wedge q \equiv q \wedge p$
(3.37) **Associativity of $\wedge$**: $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
(3.38) **Idempotency of $\wedge$**: $p \wedge p \equiv p$
(3.39) **Identity of $\wedge$**: $p \wedge \mathit{true} \equiv p$
(3.40) **Zero of $\wedge$**: $p \wedge \mathit{false} \equiv \mathit{false}$
(3.42) **Contradiction**: $p \wedge \neg p \equiv \mathit{false}$

(3.24) **Symmetry of $\vee$**: $p \vee q \equiv q \vee p$
(3.25) **Associativity of $\vee$**: $(p \vee q) \vee r \equiv p \vee (q \vee r)$
(3.26) **Idempotency of $\vee$**: $p \vee p \equiv p$
(3.29) **Zero of $\vee$**: $p \vee \mathit{true} \equiv \mathit{true}$
(3.30) **Identity of $\vee$**: $p \vee \mathit{false} \equiv p$
(3.28) **Excluded Middle**: $p \vee \neg p$

(3.45) **Distributivity of $\vee$ over $\wedge$**: $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
(3.46) **Distributivity of $\wedge$ over $\vee$**: $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
(3.47) **De Morgan**: $\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\qquad \neg(p \vee q) \equiv \neg p \wedge \neg q$

Binary Boolean Operators: Conjunction

Args.		$\wedge$	
F	F	F	The moon is green, and $2 + 2 = 7$.
F	T	F	The moon is green, and $1 + 1 = 2$.
T	F	F	$1 + 1 = 2$, and the moon is green.
T	T	T	$1 + 1 = 2$, and the sun is a star.

Binary Boolean Operators: Disjunction

Args.		$\vee$	
F	F	F	The moon is green, or $2 + 2 = 7$.
F	T	T	The moon is green, or $1 + 1 = 2$.
T	F	T	$1 + 1 = 2$, or the moon is green.
T	T	T	$1 + 1 = 2$, or the sun is a star.

This is known as "inclusive or" — see textbook p.34.