Discrete Mathematics with Applications I

COMPSCI&SFWRENG 2DM3

McMaster University, Fall 2019

Wolfram Kahl

2019-10-11

What Exactly is Wrong? How Should It Be Done Right?

$f : \mathbb{N} \to \mathbb{N}$
$f\ n\ =\ (\ \sum i : \mathbb{N} \mid i < n \bullet i + 2\ )$

$$
\begin{array}{ll}
 & 2 < f\ 2 \qquad\qquad \text{— This is } \ldots \quad \textit{Exercise!} \\
= & \langle \text{One-point rule for } \forall\text{, Substitution} \rangle \\
 & (\forall\ i : \mathbb{N} \mid i = 2 \bullet i < f\ i\ ) \\
= & \langle \text{Def. } f \text{ with '} n := i \text{'} \rangle \\
 & (\forall\ i : \mathbb{N} \mid i = 2 \bullet i < (\sum i : \mathbb{N} \mid i < n \bullet i + 2)[n := i]) \\
= & \langle \text{Substitution} \rangle \\
 & (\forall\ i : \mathbb{N} \mid i = 2 \bullet i < (\sum i : \mathbb{N} \mid i < i \bullet i + 2)) \\
= & \langle \text{Irreflexivity of } < \rangle \\
 & (\forall\ i : \mathbb{N} \mid i = 2 \bullet i < (\sum i : \mathbb{N} \mid \mathit{false} \bullet i + 2)) \\
= & \langle \text{Empty range for } \sum \rangle \\
 & (\forall i : \mathbb{N} \mid i = 2 \bullet i < 0) \\
= & \langle \text{One-point rule for } \forall\text{, Substitution} \rangle \\
 & 2 < 0
\end{array}
$$

Plan for Today

- **Textbook Chapters 8 and 9: Quantification and Predicate Logic**
  - **Variable Binding:** Interplay between Substitution and Quantification
  - **Universal and Existential Quantification**

Expanding Universal and Existential Quantification

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

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

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

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

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*: $\quad (\,\forall\, x \bullet P\,) \;=\; (\,\forall\, x \mid \mathit{true} \bullet P\,)$
  $\quad (\,\exists\, x \bullet P\,) \;=\; (\,\exists\, x \mid \mathit{true} \bullet P\,)$

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

Generalising De Morgan to Quantification

$$
\begin{array}{ll}
 & \neg(\exists\, i \mid 0 \le i < 4 \bullet P) \\
= & \langle \text{ Expand quantification } \rangle \\
 & \neg(P[i := 0] \vee P[i := 1] \vee P[i := 2] \vee P[i := 3]) \\
= & \langle \text{ (3.47) De Morgan } \rangle \\
 & \neg P[i := 0] \wedge \neg P[i := 1] \wedge \neg P[i := 2] \wedge \neg P[i := 3] \\
= & \langle \text{ Contract quantification } \rangle \\
 & (\forall\, i \mid 0 \le i < 4 \bullet \neg P)
\end{array}
$$

(9.18b,c,a) **Generalised De Morgan**:

$$
\begin{array}{rcl}
\neg(\exists\, x \mid R \bullet P) & \equiv & (\forall\, x \mid R \bullet \neg P) \\
(\exists\, x \mid R \bullet \neg P) & \equiv & \neg(\forall\, x \mid R \bullet P) \\
\neg(\exists\, x \mid R \bullet \neg P) & \equiv & (\forall\, x \mid R \bullet P)
\end{array}
$$

(9.17) **Axiom, Generalised De Morgan**:

$$(\exists\, x \mid R \bullet P) \quad \equiv \quad \neg(\forall\, x \mid R \bullet \neg P)$$

"Trading" Range Predicates with Body Predicates in $\forall$

(9.2) Axiom, Trading: $(\forall x \mid R \bullet P) \quad \equiv \quad (\forall x \bullet R \Rightarrow P)$

Trading Theorems for $\forall$:

(9.3a) $(\forall x \mid R \bullet P) \quad \equiv \quad (\forall x \bullet \neg R \vee P)$

(9.3b) $(\forall x \mid R \bullet P) \quad \equiv \quad (\forall x \bullet R \wedge P \equiv R)$

(9.3c) $(\forall x \mid R \bullet P) \quad \equiv \quad (\forall x \bullet R \vee P \equiv P)$

(9.4a) $(\forall x \mid Q \wedge R \bullet P) \quad \equiv \quad (\forall x \mid Q \bullet R \Rightarrow P)$

(9.4b) $(\forall x \mid Q \wedge R \bullet P) \quad \equiv \quad (\forall x \mid Q \bullet \neg R \vee P)$

(9.4c) $(\forall x \mid Q \wedge R \bullet P) \quad \equiv \quad (\forall x \mid Q \bullet R \wedge P \equiv R)$

(9.4d) $(\forall x \mid Q \wedge R \bullet P) \quad \equiv \quad (\forall x \mid Q \bullet R \vee P \equiv P)$

"Trading" Range Predicates with Body Predicates in $\exists$

(9.2) Axiom, Trading: $(\forall x \mid R \bullet P) \quad \equiv \quad (\forall x \bullet R \Rightarrow P)$

(9.17) **Axiom, Generalised De Morgan**: $(\exists x \mid R \bullet P) \quad \equiv \quad \neg(\forall x \mid R \bullet \neg P)$

(9.19) Trading for $\exists$: $(\exists x \mid R \bullet P) \quad \equiv \quad (\exists x \bullet R \wedge P)$

(9.20) **Trading for** $\exists$: $(\exists x \mid Q \wedge R \bullet P) \equiv (\exists x \mid Q \bullet R \wedge P)$

Bound / Free Variable Occurrences

(8.7) $(\forall i \bullet x \cdot i = 0)$ LADM example expression

Is this true or false? In which states?

We have: $(\forall i \bullet x \cdot i = 0) \quad \equiv \quad x = 0$

The value of (8.7) in a state depends only on x, not on i!

Renaming quantified variables does not change the meaning:

$$(\forall i \bullet x \cdot i = 0) \quad \equiv \quad (\forall j \bullet x \cdot j = 0)$$

- **Occurrences** of quantified variables inside the quantified expression are **bound**
- **Variable occurences** in an expression where they are not bound are **free**

$$i > 0 \vee (\forall i \mid 0 \leq i \bullet x \cdot i = 0)$$

- The variable declarations after the quantification operator may be called **binding occurrences**.

Variable Binding is Everywhere!

- Calculus: $f(y) = \int_0^1 x^2 y^2 dx$
- Imperative Programming (here C):

```
int f(int x)
{
  int q;
  q = x * x;
  return 2 * q;
}
```

- Functional Programming (here Haskell):

```
f x = let q = x * x in 2 * q
```

Variable Binding is Everywhere! Including in Substitution!

Another example expression: $(x + 3 = 5 \cdot i)[i := 9]$

Is this true or false? In which states?

$$\begin{array}{ll} & (x + 3 = 5 \cdot i)[i := 9] \\ \equiv & \langle \text{Substitution}, \ldots \rangle \\ & x = 42 \end{array}$$

The value of $(x + 3 = 5 \cdot i)[i := 9]$ in a state depends only on x, not on i!

Renaming substituted variables does not change the meaning:

$$(x + 3 = 5 \cdot i)[i := 9] \qquad \equiv \qquad (x + 3 = 5 \cdot j)[j := 9]$$

- **Occurrences** of substituted variables inside the target expression are **bound**
- The variable occurrences to the left of := in substitutions may be called **binding occurrences**.
- **Variable occurences** in an expression where they are not bound are **free**.
  $i > 0 \wedge (x + 3 = 5 \cdot i)[i := 7 + i]$
- **Substitution does not bind to the right of** := **!**

Trivial Range Axioms for Universal and Existential Quantification

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

$$(\forall\ x \mid false \bullet P) \quad = \quad true$$

$$(\exists\ x \mid false \bullet P) \quad = \quad false$$

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

(8.14) **Axiom, One-point Rule:** Provided $\neg occurs(\text{‘}x\text{’}, \text{‘}E\text{’})$,

$$(\forall x \mid x = E \bullet P) \quad \equiv \quad P[x := E]$$

$$(\exists x \mid x = E \bullet P) \quad \equiv \quad P[x := E]$$

The *occurs* Meta-Predicate

Definition: $occurs(`v', `e')$ means that at least one variable in the list v of variables occurs **free** in at least one expression in expression list e.

$occurs(`i', `5 \cdot i')$ ✓

$occurs(`i', `0 \cdot i')$ ✓

$occurs(`i', `5 \cdot k')$ ×

$occurs(`i', `(\sum i \mid 0 \le i < k \bullet n^i)')$ ×

$occurs(`n', `(\sum i \mid 0 \le i < k \bullet n^i)')$ ✓

$occurs(`i,n', `(\sum i \mid 0 \le i < k \bullet n^i)')$ ✓

$occurs(`i,n', `(\sum i,n \mid 1 \le i \cdot n \le k \bullet n^i)')$ ×

$occurs(`i,n', `(\sum i,n \mid 1 \le i \cdot n \le k \bullet n^i), (\sum n \mid 0 \le n < k \bullet n^i)')$ ✓

$occurs(`i', `(i \cdot (5+i))[i := k+2]')$ × **Substitution is a variable binder, too!**

$occurs(`i', `(i \cdot (5+i))[i := i+2]')$ ✓

The $\neg$*occurs* Proviso for the One-point Rule

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

$$(\forall x \mid x = E \bullet P) \quad\equiv\quad P[x := E]$$

$$(\exists x \mid x = E \bullet P) \quad\equiv\quad P[x := E]$$

Examples:

- $(\forall x \mid x = 1 \bullet x \cdot y = y) \quad\equiv\quad 1 \cdot y = y$
- $(\exists x \mid x = y+1 \bullet x \cdot x > 42) \quad\equiv\quad (y+1)\cdot(y+1) > 42$

Counterexamples:

- $(\forall x \mid x = x+1 \bullet x = 42) \quad ? \quad x+1 = 42$ — **"≡" not valid!**
- $(\exists x \mid x = 2 \cdot x \bullet y + x = 42) \quad ? \quad y + 2 \cdot x = 42$ — **"≡" not valid!**

Automatic extraction of $\neg$*occurs* Provisos

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

$$(\forall x \mid x = E \bullet P) \quad\equiv\quad P[x := E]$$

$$(\exists x \mid x = E \bullet P) \quad\equiv\quad P[x := E]$$

Investigate the binders in scope at the metavariables P and E:

- P on the LHS occurs in scope of the binder $\forall\, x$
- P on the RHS occurs in scope of the binder $\_[x := \ldots]$

Therefore: Whether x occurs in P or not does not raise any problems.

- E on the LHS occurs in scope of the binder $\forall\, x$
- E on the RHS occurs in scope no binders

Therefore: An x that is free in E would be **bound** on the LHS, but **escape** into freedom on the RHS!

CalcCheck derives and checks $\neg$*occurs* provisos automatically.

Textual Substitution Revisited

Let E and R be expressions and let x be a variable. **Original definition:**

> We write: $E[x := R]$ or E^x_R
> to denote an expression that is the same as E but with all occurrences of x replaced by (R).

This was for expressions E built from **constants, variables, operator applications** only!

In presence of **variable binders**, such as $\sum$, $\prod$, $\forall$, $\exists$ and substitution,

- only **free** occurrences of x can be replaced
- **and** we need to avoid **"capture of free variables"**:

(8.11) Provided $\neg occurs(`y', `x, F')$,

$$(\star y \mid R \bullet P)[x := F] \quad = \quad (\star y \mid R[x := F] \bullet P[x := F])$$

LADM Chapter 8:

"$\star$ is a **metavariable** for operators $\_+\_$, $\_\cdot\_$, $\_\wedge\_$, $\_\vee\_$ (resp. $\sum$, $\prod$, $\forall$, $\exists$)

(8.11) is part of the Substitution keyword in CalcCheck.

Read LADM Chapter 8!

Substitution Examples

(8.11) Provided $\neg occurs(`y', `x, F')$,

$$(\star y \mid R \bullet P)[x := F] \quad = \quad (\star y \mid R[x := F] \bullet P[x := F])$$

- $(\sum x \mid 1 \leq x \leq 2 \bullet y)[y := y + z]$
  $= \langle$ substitution $\rangle$
  $(\sum x \mid 1 \leq x \leq 2 \bullet y + z)$

- $(\sum x \mid 1 \leq x \leq 2 \bullet y)[y := y + x]$
  $= \langle$ (8.21) Variable renaming $\rangle$
  $(\sum z \mid 1 \leq z \leq 2 \bullet y)[y := y + x]$
  $= \langle$ substitution $\rangle$
  $(\sum z \mid 1 \leq z \leq 2 \bullet y + x)$

Renaming of Bound Variables

(8.21) **Axiom, Dummy renaming** (α-conversion):

$$(\star x \mid R \bullet P) \quad = \quad (\star y \mid R[x := y] \bullet P[x := y])$$

provided $\neg occurs(`y', `R, P')$.

$(\sum i \mid 0 \leq i < k \bullet n^i)$
$= \langle$ Dummy renaming (8.21), $\neg occurs(`j', `0 \leq i < k,\ n^i') \rangle$
$(\sum j \mid 0 \leq j < k \bullet n^j)$

$(\sum i \mid 0 \leq i < k \bullet n^i)$
$?\ \langle$ Dummy renaming (8.21)$\rangle$ $\times$
$(\sum k \mid 0 \leq k < k \bullet n^k)$

In CalcCheck, renaming of bound variables is part of "Reflexivity of =",
but can also be mentioned explicitly.

Substitution Examples (ctd.)

(8.11) Provided $\neg occurs(`y', `x, F')$,

$$(\star\, y \mid R \bullet P)[x := F] \quad = \quad (\star\, y \mid R[x := F] \bullet P[x := F])$$

- $$\begin{aligned} & (\sum x \mid 1 \le x \le 2 \bullet y)[x := y + x] \\ = & \ \langle \text{(8.21) Variable renaming} \rangle \\ & (\sum z \mid 1 \le z \le 2 \bullet y)[x := y + x] \\ = & \ \langle \text{Substitution} \rangle \\ & (\sum z \mid 1 \le z \le 2 \bullet y) \\ = & \ \langle \text{(8.21) Variable renaming} \rangle \\ & (\sum x \mid 1 \le x \le 2 \bullet y) \end{aligned}$$

(8.11f) Provided $\neg occurs(`x', `E')$,

$$E[x := F] \quad = \quad E$$

Predicate Logic Laws You Really Need To Know

(9.2) **Trading for $\forall$:** $(\forall x \mid R \bullet P) \quad \equiv \quad (\forall x \bullet R \Rightarrow P)$

(9.4a) **Trading for** $\forall$: $(\forall x \mid Q \wedge R \bullet P) \quad \equiv \quad (\forall x \mid Q \bullet R \Rightarrow P)$

(9.19) **Trading for $\exists$:** $(\exists x \mid R \bullet P) \quad \equiv \quad (\exists x \bullet R \wedge P)$

(9.20) **Trading for** $\exists$: $(\exists x \mid Q \wedge R \bullet P) \equiv (\exists x \mid Q \bullet R \wedge P)$

(9.13) **Instantiation:** $(\forall x \bullet P) \quad \Rightarrow \quad P[x := E]$

(9.28) **$\exists$-Introduction**: $P[x := E] \quad \Rightarrow \quad (\exists x \bullet P)$

(9.17) **Generalised De Morgan**: $(\exists x \mid R \bullet P) \quad \equiv \quad \neg(\forall x \mid R \bullet \neg P)$

(8.13) **Empty Range**: $(\forall x \mid false \bullet P) \quad = \quad true$
$(\exists x \mid false \bullet P) \quad = \quad false$

(8.14) **One-point Rule:** Provided $\neg occurs(`x', `E')$, $(\forall x \mid x = E \bullet P) \quad \equiv \quad P[x := E]$
$(\exists x \mid x = E \bullet P) \quad \equiv \quad P[x := E]$

…and correctly handle substitution, Leibniz, renaming of bound variables, and monotonicity/antitonicity …