A software engineer's appraisal of $e = m + c$

J.N. Oliveira

Departamento de Informática, Escola de Engenharia
Universidade do Minho

SDDI'06 — PhD symposium
U. Minho, Braga, Feb. 22-24

Reflections of a PhD student

Listen to the suffering PhD student . . .

Reflections of a PhD student

Listen to the suffering PhD student . . .

- Is my research worthwhile?
- Have **I** got the 'right' theory (if any)?
- Will **I** ever get a PhD?
- What am **I** doing here . . . ??

Doing a PhD project:

Reflections of a PhD student

Listen to the suffering PhD student . . .

- Is my research worthwhile?
- Have I got the 'right' theory (if any)?
- Will I ever get a PhD?
- What am I doing here . . . ??

Doing a PhD project:

- # Has it always been like this?

Reflections of a PhD student

Listen to the suffering PhD student …

- Is my research worthwhile?
- Have I got the ‘right’ theory (if any)?
- Will I ever get a PhD?
- What am I doing here …??

Doing a PhD project:

- Has it always been like this?
- Wasn’t science nicer in the old days?

Reflections of a PhD student

Software engineering

- Is there a software engineering?
- Are we condemned to trial and error?
- If not: what is missing?

$$e \quad = \quad ?$$

About science and engineering

Science

Science is about understanding how things work

About science and engineering

Science

Science is about understanding how things work

Engineering

This is about ensuring that some desirable things happen repetitively and reliably.

About science and engineering

Science

Science is about understanding how things work

Engineering

This is about ensuring that some desirable things happen repetitively and reliably.

Theodore Von Karman, an aerospace engineer quoted in `http://www.discoverengineering.org`, puts it in this way: *"Scientists discover the world that exists; engineers create the world that never was."*

About science and engineering

Science

Science is about understanding how things work

Engineering

This is about ensuring that some desirable things happen repetitively and reliably.

Theodore Von Karman, an aerospace engineer quoted in `http://www.discoverengineering.org`, puts it in this way: *"Scientists discover the world that exists; engineers create the world that never was."*

In both cases

Need for a modelling strategy

Problem-solving strategy

Software technology is becoming a mature discipline in its (however late) adoption of the **universal problem solving** strategy (UPS) which one is taught at school:

- **understand** your problem
- build a mathematical **model** of it
- **reason** in such a model
- upgrade your model, if necessary
- **calculate** a final solution and implement it.

School maths example

The problem

My three children were born at a 3 year interval rate. Altogether, they are as old as me. I am 48. How old are they?

School maths example

The problem

My three children were born at a 3 year interval rate. Altogether, they are as old as me. I am 48. How old are they?

The model

$$x + (x + 3) + (x + 6) = 48$$

School maths example

The problem

My three children were born at a 3 year interval rate. Altogether, they are as old as me. I am 48. How old are they?

The model

$$x + (x + 3) + (x + 6) = 48$$

The calculation

$$\begin{array}{ll} & 3x + 9 = 48 \\ \equiv & \{ \text{“al-djabr” rule} \} \\ & 3x = 48 - 9 \\ \equiv & \{ \text{“al-hatt” rule} \} \\ & x = 16 - 3 \end{array}$$

School maths example

The solution

$$\begin{aligned} x &= 13 \\ x+3 &= 16 \\ x+6 &= 19 \end{aligned}$$

School maths example

The solution

$$\begin{aligned} x &= 13 \\ x+3 &= 16 \\ x+6 &= 19 \end{aligned}$$

Questions....

- "al-djabr" rule ?
- "al-hatt" rule ?

School maths example

The solution

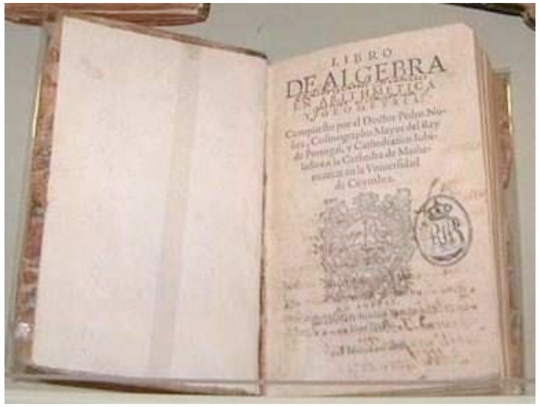
$$\begin{aligned} x &= 13 \\ x+3 &= 16 \\ x+6 &= 19 \end{aligned}$$

Questions....

- "al-djabr" rule ?
- "al-hatt" rule ?

You should read Pedro Nunes (1502-1578) *Libro de Algebra en Arithmetica y Geometria* dated 1567 ...

Libro de Algebra en Arithmetica y Geometria (1567)



(...) ho inuẽtor desta arte foy hum Mathematico Mouro, cujo nome era Gebre, & ha em alguãs Liuarias hum pequeno tractado Arauigo, que contem os capitulos de q̃ usamos
(fol. a ij r)

Reference to *On the calculus of al-gabr and al-muqâbala* [1] by Abû Abd Allâh Muhamad B. Mûsâ Al-Huwârizmî, a famous 9c Persian mathematician.

---

[1] Original title: *Kitâb al-muhtasar fi hisab al-gabr wa-almuqâbala*.

Calculus of al-gabr and al-muqâbala

al-djabr

$$x - z \leq y \quad \equiv \quad x \leq y + z$$

Calculus of al-gabr and al-muqâbala

al-djabr

$$x - z \leq y \quad \equiv \quad x \leq y + z$$

all-hatt

$$x * z \leq y \quad \equiv \quad x \leq y * z^{-1} \qquad (z > 0)$$

Calculus of al-gabr and al-muqâbala

al-djabr

$$x - z \leq y \quad \equiv \quad x \leq y + z$$

all-hatt

$$x * z \leq y \quad \equiv \quad x \leq y * z^{-1} \qquad (z > 0)$$

al-muqâbala

Ex:

$$4x^2 + 3 = 2x^2 + 2x + 6 \quad \equiv \quad 2x^2 = 2x + 3$$

Back to geometry and trigonometry . . .

Hot topic in the 16c: revisit old geometrical problems, inc. Euclid's Elements.

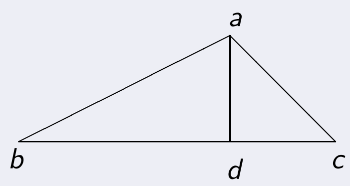
Back to geometry and trigonometry . . .

Hot topic in the 16c: revisit old geometrical problems, inc. Euclid's Elements.

Problem 12 in Johan Müller's (1436-1476) *"De Triangulis"*, vol.II

Given

$$\begin{aligned} bc &= 20 \\ ad &= 5 \\ \frac{ab}{ac} &= \sqrt{5} \end{aligned}$$



find ab, ac and ad.

This is Question 46 in Nunes book (fol. 270r), given as example of problem which Müller could not solve on pure geometric grounds. . .

. . . solved “by algebra”

Nunes model is based on the *inuento Pithagorico* [2]:

Model

(...) Queriendo pos conoscer los lados (...) pornemos .d.c. parte menor ser .1.co. [read: $x = dc$, where *co* is “cousa” = “the thing” (we are looking for)] *(...) Y porque .bd. es .20.m̃.1.co (...) sera el su quadrado 400.p̃.1.ce.m̃.40.co* [read: $20^2 + x^2 - 40x$] *(...)*

Thus he reaches model

$$\frac{ab^2}{ac^2} = \frac{425 - 40x + x^2}{x^2 + 25} = 5$$

[2] “Pythagoras invention”, ie. Prop. 47 of Euclid's Elements — see eg. http://aleph0.clarku.edu/ djoyce/java/elements/bookI/propI47.html

. . . solved "by algebra"

Nunes algebraic calculation

$$\frac{425 - 40x + x^2}{x^2 + 25} = 5$$

$$\equiv \quad \{ \text{ rule } \tfrac{a}{b} = \tfrac{c}{d} \equiv ad = bc \text{ etc } \}$$

$$425 - 40x + x^2 = 5x^2 + 125$$

$$\equiv \quad \{ \text{ "calculus of al-gabr and al-muqâbala" (...) } \}$$

$$75 = x^2 + 10x$$

This leads to the expected

Solution

(...) sera luego *.a.b. R.250.* e *.a.c. R.50* *[read: $ab = \sqrt{250}$ and $ac = \sqrt{5}$]*

"Ut faciant opus signa"

Nunes proof style is regarded by H. Bossmans as unique among his contemporaries in precision and elegance:

> *Ni chez Stifel, ni chez Cardan, on ne trouverait pas une page écrite dans ce style.*

That Nunes marvels at *this most subtle art of Algebra ("esta subtilissima arte de Algebra.")* is apparent from many side comments in his book. (See next slides.)

Were he among us today, he would certainly join the school of thought so nicely captured by Jeuring-Meertens's MPC logo:

Nunes comments

Algebra (...) is thing causing admiration

(...) *Principalmente que vemos algumas vezes, no poder vn gran Mathematico resoluer vna question por medios Geometricos, y resolverla por Algebra, siendo la misma Algebra sacada de la Geometria, q̃ es cosa de admiraciõ.*

that is (literal — not literary — translation):

> *(...) Mainly because we see often a great Mathematician unable to resolve a question by Geometrical means, and solve it by Algebra, being that same Algebra taken from Geometry, which is thing causing admiration.*

[Pedro Nunes (1502-1578) in Libro de Algebra en Arithmetica y Geometria, 1567, fols. 270–270v.]

Letting "the symbols do the work" in the 16c

fol. 269r-269v:

Going algebraic

Mas la causa porq̃ obro por Algebra quasi siempre, es que este tratado es hecho para que en el practiquen las Reglas de Algebra en los casos de Geometria.

ie.

> *But the reason why I work by Algebra almost always, is that this treaty is written so that in it you practise the rules of Algebra in case studies of Geometry*

Letting "the symbols do the work" in the 16c

fol. 269r-269v:

Deduction first

Y tambien porque quien obra por Algebra va entendiendo la razon de la obra que haze, hasta la yqualacion ser acabada. (...) De suerte que, quien obra por Algebra, va haziendo discursos demonstrativos.

ie.

> *And also because one performing by Algebra is understanding the reason of the work one does, until the equality is finished. (...) So much so that, who works by Algebra is doing a demonstrative discourse.*

Verdict

(...) De manera, que
quien sabe por Algebra,
sabe scientificamente.

(in this way, who knows by Algebra knows scientifically)

Trend for notation economy

Well-known throughout the history of maths — a kind of "natural language **implosion**" — particularly visible in the syncopated phase (16c), eg.

.40.p̃.2.ce. son yguales a .20.co

(P. Nunes, Coimbra, 1567) for nowadays $40 + 2x^2 = 20x$, or

B 3 in A quad - D plano in A + A cubo æquatur Z solido

(F. Viète, Paris, 1591) for nowadays $3BA^2 - DA + A^3 = Z$

Final touch

Trend for notation economy

Well-known throughout the history of maths — a kind of "natural language **implosion**" — particularly visible in the syncopated phase (16c), eg.

.40.p̃.2.ce. son yguales a .20.co

(P. Nunes, Coimbra, 1567) for nowadays $40 + 2x^2 = 20x$, or

B 3 in A quad - D plano in A + A cubo æquatur Z solido

(F. Viète, Paris, 1591) for nowadays $3BA^2 - DA + A^3 = Z$

Final touch

René Decartes (1596–1650), who studied algebra by the books of Clavius (1538–1612), a student of Nunes at Coimbra.

Later on (18c, 19c, ...)

More demanding problems to be modelled/solved, eg. electrical circuits:

From a simple law ...

$V = R \times I$ by Georg Ohm (1789-1854) ...

Later on (18c, 19c, . . .)

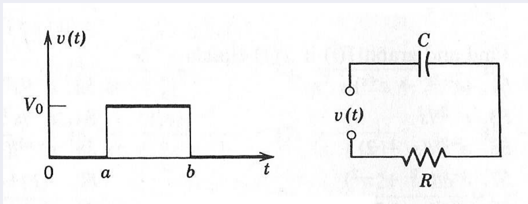
More demanding problems to be modelled/solved, eg. electrical circuits:

From a simple law . . .

$V = R \times I$ by Georg Ohm (1789-1854) . . .

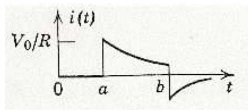
. . . to non-linear RC-circuits

$$v(t) = Ri(t) + \frac{1}{C}\int_0^t i(\tau)d\tau$$
$$v(t) = V_0(u(t-a) - u(t-b)) \qquad (b > a)$$



Calculate $i(t)$

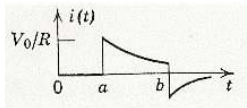
The following $i(t)$ can be observed on an oscilloscope:



Can you explain it?

Calculate $i(t)$

The following $i(t)$ can be observed on an oscilloscope:



Can you explain it?

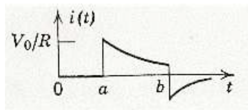
Is 16c maths still enough for the required calculations?

No. Need for the the differential/integral calculus.

But there is more:

Calculate $i(t)$

The following $i(t)$ can be observed on an oscilloscope:



Can you explain it?

Is 16c maths still enough for the required calculations?

No. Need for the the differential/integral calculus.

But there is more:

For the underlying maths to scale up

Need for an *integral transform*, eg. the Laplace transform.

Laplace transform

t-space	s-space
Given problem $y'' + 4y' + 3y = 0$ $y(0) = 3$ $y'(0) = 1$ →	**Subsidiary equation** $s^2 + 4sY + 3Y = 3s + 13$ ↓
Solution of given problem $y(t) = -2e^{-3t} + 5e^{-t}$	← **Solution of subs. equation** $Y = \frac{-2}{s+3} + \frac{5}{s+1}$

An integral transform

$(\mathcal{L}\, f)s = \int_0^\infty e^{-st} f(t) dt$, eg.

$f(t)$	$\mathcal{L}(f)$
1	$\frac{1}{s}$
t	$\frac{1}{s^2}$
t^n	$\frac{n!}{s^{n+1}}$
e^{at}	$\frac{1}{s-a}$
etc	

Pierre Laplace (1749-1827)

Laplace-transformed RC-circuit model

$\mathcal{L}$(t-space RC model) is

$$RI(s) + \frac{I(s)}{sC} = \frac{V_0}{s}(e^{-as} - e^{-bs})$$

whose *algebraic* solution for $I(s)$ is

$$I(s) = \frac{\frac{V_0}{R}}{s + \frac{1}{RC}}(e^{-as} - e^{-bs})$$

Now, the converse transformation:

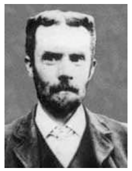
$$\mathcal{L}^{-1}\left(\frac{\frac{V_0}{R}}{s + \frac{1}{RC}}\right) = \frac{V_0}{R}e^{-\frac{t}{RC}}$$

Analytical solution

After some algebraic manipulation we will obtain an analytical answer . . .

$$i(t) = \begin{cases} 0 & if \quad t < a \\ (\frac{V_0 e^{-\frac{a}{RC}}}{R})e^{-\frac{t}{RC}} & if \quad a < t < b \\ (\frac{V_0 e^{-\frac{a}{RC}}}{R} - \frac{V_0 e^{-\frac{b}{RC}}}{R})e^{-\frac{t}{RC}} & if \quad t > b \end{cases}$$

. . . with some help by Oliver Heaviside (1850-1925)



What's new?

While the underlying mathematics has changed,

- from systems of **polynomial** equations, to
- **differential/integral** equations

the overall approach is the same:

$$e = m + c$$

ie.

engineering = *model first, then calculate* . . .

Moreover, via the Laplace transform we get back to **polynomial** equations again.

$e = m + c$ challenges

A "notation problem":

mathematical modelling

requires *descriptive* notations, therefore:

- intuitive
- domain-specific

$e = m + c$ challenges

A “notation problem”:

mathematical modelling

requires *descriptive* notations, therefore:

- intuitive
- domain-specific

calculation

requires *elegant* notations, therefore:

- simple and compact
- generic
- cryptic, otherwise uneasy to manipulate

$e = m + c$ challenges

A "notation problem":

mathematical modelling

requires *descriptive* notations, therefore:

- intuitive
- domain-specific

calculation

requires *elegant* notations, therefore:

- simple and compact
- generic
- cryptic, otherwise uneasy to manipulate

Recall Dijkstra's definition : *elegant* $\equiv$ *simple and remarkably effective*

Quoting Kreyszig's book, p.242

"(...) The Laplace transformation is a method for solving differential equations (...) [which] consists of three main steps:

- 1st step. *The given "hard" problem is transformed into a "simple" equation (subsidiary equation).*
- 2nd step. *The subsidiary equation is solved by* **purely algebraic** *manipulations.*
- 3rd step. *The solution of the subsidiary equation is transformed back to obtain the solution of the given problem.*

In this way the Laplace transformation reduces the problem of solving a differential equation to an **algebraic problem***".*

Question

All we have said applies to physics, mechanical eng., civil eng., electrical and electronic eng.

Question

All we have said applies to physics, mechanical eng., civil eng., electrical and electronic eng.

What about us? (software engineers)

Question

All we have said applies to physics, mechanical eng., civil eng., electrical and electronic eng.

What about us? (software engineers)

Concerning descriptive notation, compare the following (already unified into the "Eindhoven quantifier notation") expressions:

$$\langle \int x \ : \ 0 < x < 10 : \ x^2 \rangle$$

$$\langle \forall \ x \ : \ 0 < x < 10 : \ x^2 \geq x \rangle$$

$$b \sqsubseteq a \quad \equiv \quad \langle \forall \ x \ : \ x \in b : \ x \in a \rangle$$

Need for a transform

Are formulæ such as eg. $b \sqsubseteq a \equiv \langle \forall\ x\ :\ x \in b :\ x \in a \rangle$ amenable to calculation?

Need for a transform

Are formulæ such as eg. $b \sqsubseteq a \equiv \langle \forall\, x : x \in b : x \in a \rangle$ amenable to calculation?

Perhaps not

They are good for descriptive purposes, eg. invariants on data structures, pre-conditions etc.

Is there an *s*-space equivalent for logical quantification?

Need for a transform

Are formulæ such as eg. $b \sqsubseteq a \quad \equiv \quad \langle \forall\, x \;:\; x \in b :\; x \in a \rangle$ amenable to calculation?

Perhaps not

They are good for descriptive purposes, eg. invariants on data structures, pre-conditions etc.

Is there an *s*-space equivalent for logical quantification?

Yes: just resort the PF-transform

$$\mathcal{PF}(\; b \sqsubseteq a \quad \equiv \quad \langle \forall\, x \;:\; x \in b :\; x \in a \rangle \;) \quad = \quad \sqsubseteq \,=\, \in \setminus \in$$

The variables (points) and the quantification have gone (PF = "point+free")

A transform for logic and set-theory

- The *pointfree transform* adopted in the sequel is at the heels of old reasoning techniques such as the Laplace transform.
- Set-theory-formulated concepts are regarded as "hard" problems to be transformed into "simple", *subsidiary equations* dispensing with points and involving only binary relations.
- As in the Laplace transformation, these are solved by *purely algebraic* manipulations
- The outcome is mapped back to the original (descriptive) mathematical space wherever required.

A transform for set-theory

An old idea:

$$\mathcal{PF}(\text{sets, predicates}) \quad = \quad \text{pointfree binary relations}$$

Calculus of binary relations $B \xleftarrow{R} A$:

- 1860 - introduced by De Morgan, embryonic
- 1870 - Peirce finds interesting equational laws
- 1941 - Tarski's school, cf. *A Formalization of Set Theory without Variables*
- 1980's - coreflexive models of sets (Freyd and Scedrov, Eindhoven MPC group and others)

Binary Relations

- Arrow $B \xleftarrow{R} A$ denotes a binary relation to B (target) from A (source).
- $b \; R \; a$ means that pair $\langle b, a \rangle$ is in R, eg.

$$\begin{array}{ccc} 1 & \leq & 2 \\ \texttt{John} & \mathit{IsFatherOf} & \texttt{Mary} \\ 3 & = (1+) & 2 \end{array}$$

- **Converse** of R — R° such that $a(R^\circ)b$ iff $b \; R \; a$.
- "R at most S" — the obvious $R \subseteq S$ **ordering**.

Some formulæ and their PF-transform

f	$\mathcal{PF}\ f$
$\langle \exists\ a\ ::\ b\ R\ a \wedge a\ S\ c \rangle$	$b(R \cdot S)c$
$\langle \forall\ a, b\ :\ b\ R\ a :\ b\ S\ a \rangle$	$R \subseteq S$
$\langle \forall\ a\ ::\ a\ R\ a \rangle$	$id \subseteq R$
$\langle \forall\ x\ :\ x\ R\ b :\ x\ S\ a \rangle$	$b(R \setminus S)a$
$\langle \forall\ c\ :\ b\ R\ c :\ a\ S\ c \rangle$	$a(S\ /\ R)b$
$b\ R\ a \wedge c\ S\ a$	$(b, c)\langle R, S \rangle a$
$b\ R\ a \wedge b\ S\ a$	$b\ (R \cap S)\ a$
$b\ R\ a \vee b\ S\ a$	$b\ (R \cup S)\ a$
$(f\ b)\ R\ (g\ a)$	$b(f^\circ \cdot R \cdot g)a$
TRUE	$b \top a$
FALSE	$b \perp a$
$b\langle \mu\ X\ ::\ \ldots R \ldots X \ldots S \ldots \rangle a$	$b[\![\ R, S\ ]\!]a$

where id such that $R \cdot id = id \cdot R = R$

Sets and predicates

Meaning of

Predicates

$A \xrightarrow{\phi} Bool$ is transformed into coreflexive relation $[\![\phi]\!] \subseteq id$ such that $b[\![\phi]\!]a \equiv (b = a) \wedge (\phi\ a)$.

Sets and predicates

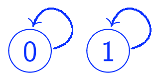
Meaning of

Predicates

$A \xrightarrow{\phi} Bool$ is transformed into coreflexive relation $[\![\phi]\!] \subseteq id$ such that $b[\![\phi]\!]a \equiv (b = a) \wedge (\phi\ a)$.

Example:

$[\![n! \leq 1]\!] =$



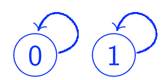
Sets and predicates

Meaning of

Predicates

$A \xrightarrow{\phi} Bool$ is transformed into coreflexive relation $[\![\phi]\!] \subseteq id$ such that $b[\![\phi]\!]a \equiv (b = a) \wedge (\phi\ a)$.

Example:

$[\![n! \leq 1]\!] \quad =$ 

Sets

The meaning of a set $S \subseteq A$ is the meaning of its *characteristic* predicate $[\![\lambda a.a \in S]\!]$, that is,

$$b[\![S]\!]a \quad \equiv \quad (b = a) \wedge a \in S$$

Back to "quien sabe por Algebra, sabe scientificamente"

Useful "al-djabr" rules

$$f \cdot R \subseteq S \equiv R \subseteq f^\circ \cdot S$$
$$R \cdot f^\circ \subseteq S \equiv R \subseteq S \cdot f$$
$$T \cdot R \subseteq S \equiv R \subseteq T \setminus S$$

etc — (nowadays) christened as **Galois connections**.

Evariste Galois (1811-1832)

Back to basics

Which areas of computing have nowadays well-established, widespread theories taught in undergraduate courses ?

- Parsers and compilers
- Relational databases

What are our "Euclidean Elements" ? Perhaps

- Maier's **The Theory of Relational Databases** (1983)
- Aho and Ullman's **The Theory of Parsing, Translation, and Compiling (1972)**

Back to basics

Which areas of computing have nowadays well-established, widespread theories taught in undergraduate courses ?

- Parsers and compilers
- Relational databases

What are our "Euclidean Elements" ? Perhaps

- Maier's **The Theory of Relational Databases** (1983)
- Aho and Ullman's **The Theory of Parsing, Translation, and Compiling (1972)**

For lesser a number of people, 1976 was a vintage year:

- Dijkstra's **A Discipline of Programming** (1976)
- Wirth's **Algorithms + Data Structures = Programs** (1976)

Back to basics

Which areas of computing have nowadays well-established, widespread theories taught in undergraduate courses ?

- Parsers and compilers
- Relational databases

What are our "Euclidean Elements" ? Perhaps

- Maier's **The Theory of Relational Databases** (1983)
- Aho and Ullman's **The Theory of Parsing, Translation, and Compiling (1972)**

For lesser a number of people, 1976 was a vintage year:

- Dijkstra's **A Discipline of Programming** (1976)
- Wirth's **Algorithms + Data Structures = Programs** (1976)

Let us have a go at Maier's "Elements" . . .

Example 1: FDs in RDB theory

Given relational table

$T =$

...	x	...	y	...
...	a	...	b	...
...	b	...	b	...
...	...	...	...	...

this is said to satisfy functional dependency $x \longrightarrow y$ iff all pairs of tuples $t, t' \in T$ which "agree" on x also "agree" on y, that is,

$$\langle \forall\ t, t'\ :\ t, t' \in T :\quad t[x] = t'[x] \quad \Rightarrow \quad t[y] = t'[y]\ \rangle$$

The role of functions

From **Database Systems: The Complete Book** by Garcia-Molina, Ullman and Widom (2002), p. 87:

> **What Is "Functional" About Functional Dependencies?**
>
> $A_1 A_2 \cdots A_n \rightarrow B$ is called a "functional dependency" because in principle there is a function that takes a list of values [...] and produces a unique value (or no value at all) for B [...] However, this function is not the usual sort of function that we meet in mathematics, because there is no way to compute it from first principles. [...] Rather, the function is only computed by lookup in the relation [...]

However,

- No advantage is taken of the rich calculus of functions

In fact, functions are everywhere in FD theory:

- as attributes
- as the FDs themselves

Functions in one slide

A function f is a binary relation such that

Pointwise	Pointfree	
"Left" Uniqueness		
$b\ f\ a \wedge b'\ f\ a \quad \Rightarrow \quad b = b'$	$\mathrm{img}\, f \quad \subseteq \quad id$	(f is simple)
Leibniz principle		
$a = a' \quad \Rightarrow \quad f\ a = f\ a'$	$id \quad \subseteq \quad \ker f$	(f is entire)

equivalent to GCs

$$f \cdot R \subseteq S \quad \equiv \quad R \subseteq f^\circ \cdot S$$
$$R \cdot f^\circ \subseteq S \quad \equiv \quad R \subseteq S \cdot f$$

(NB: $\ker f = \mathrm{img}\, f^\circ = f^\circ \cdot f$ measures f's injectivity).

Pointfree, generic FDs

In *First steps in pointfree data dependence theory* (PURe project technical report) I calculate the PF-transform of

$$\langle \forall\ t, t' \ :\ t, t' \in T :\quad t[x] = t'[x] \quad\Rightarrow\quad t[y] = t'[y]\ \rangle$$

which shrinks to

PF functional dependency

$$x \xrightarrow{R} y \quad\equiv\quad y \ \leq\ x \cdot R^{\circ}$$

where

$$R \leq S$$

is the "injectivity" preorder.

MVD definition — Maier (1983)

An n-ary relation T is said to satisfy the *multi-valued* dependency (MVD) $x \twoheadrightarrow y$ iff, for any two tuples $t, t' \in T$ which "agree" on x there exists a tuple $t'' \in T$ which "agrees" with t on xy and "agrees" with t' on $S - xy$, that is,

$$\langle \forall\ t, t' \ :\ t, t' \in T : \quad t[x] = t'[x] \Rightarrow \langle \exists\ t'' \ :\ t'' \in T : \ t[xy] = t''[xy] \wedge t''[S - xy] = t'[S - xy] \rangle \rangle$$

	x	y	$S - xy$
t	α	β	γ
t''	α	β	γ'
t'	α	β'	γ'

MVD definition — Beeri, Fagin & Howard (1977)

Given subsets $x, y \subseteq S$ of the relation scheme S, let $z = S - xy$. An n-ary relation T is said to satisfy the *multi-valued* dependency (MVD) $x \twoheadrightarrow y$ iff, for every xz-value ab, that appears in T, we have $Y(ab) = Y(a)$, where for every $k \subseteq S$ and k-value c, we define

$$Y(c) \quad = \quad \{y \mid \langle \exists\, u \;:\; u \in T :\; u[k] = c \wedge u[y] = y \rangle\}$$

MVD definition — Beeri, Fagin & Howard (1977)

Given subsets $x, y \subseteq S$ of the relation scheme S, let $z = S - xy$. An n-ary relation T is said to satisfy the *multi-valued* dependency (MVD) $x \twoheadrightarrow y$ iff, for every xz-value ab, that appears in T, we have $Y(ab) = Y(a)$, where for every $k \subseteq S$ and k-value c, we define

$$Y(c) = \{y \mid \langle \exists\, u : u \in T : u[k] = c \wedge u[y] = y \rangle\}$$

Question

Are these "the same" definition?

$$\begin{array}{lcl} \text{Maier} & \Rightarrow & \text{Beeri, Fagin \& Howard?} \\ \text{Maier} & \Leftarrow & \text{Beeri, Fagin \& Howard?} \\ \text{Maier} & \equiv & \text{Beeri, Fagin \& Howard?} \end{array}$$

A PF-transform check of MVD definition

Bear in mind the following two PF-transform rules:

"Guarded" composition

Given

- binary relations $B \xleftarrow{R} A$ and $A \xleftarrow{S} C$
- predicate $Bool \xleftarrow{\phi} A$ (ie. coreflexive Φ)

Then, for all $b \in B$ and $c \in C$,

$$\langle \exists\ a\ :\ \phi\ a :\ b\ R\ a \wedge a\ S c \rangle$$

PF-transforms to

$$b(R \cdot \Phi \cdot S)c$$

A PF-transform check

"Guarded 'at most"'

Given

- binary relations $B \xleftarrow{R,S} A$
- predicates $Bool \xleftarrow{\psi} A$ and $Bool \xleftarrow{\phi} B$ (ie., coreflexives Ψ and Φ, respectively)

Then

$$\langle \forall\ b, a\ :\ (\phi\ b) \wedge (\psi\ a) :\ b\ R\ a \Rightarrow b\ S\ a \rangle$$

PF-transforms to

$$\Phi \cdot R \cdot \Psi \subseteq S$$

MVDs PF-transformed

Let us abbreviate $S - xy$ to z in

Maier's definition

$$\langle \forall\ t, t' \ :\ t, t' \in T :\ \begin{array}{c} t[x] = t'[x] \\ \Downarrow \\ \langle \exists\ t'' \ :\ t'' \in T :\ \begin{array}{l} t[xy] = t''[xy] \wedge \\ t''[z] = t'[z] \end{array} \rangle \end{array} \rangle$$

and recall $\dfrac{f \quad | \quad \mathcal{PF}\ f}{(f\ b)\ R\ (g\ a) \ |\ b(f^{\circ} \cdot R \cdot g)a}$ that is,

$(f\ b) = (f\ a) \quad \equiv \quad b(f^{\circ} \cdot f)a$. Thanks to the "guarded rules" above and abbreviating functional split $\langle x, y \rangle$ by xy we obtain, in sequence:

MVDs PF-transformed

PF-transform, Step 1

$$
\begin{array}{ll}
x \stackrel{T}{\rightharpoondown\!\!\rightharpoondown} y & \equiv \langle \forall\ t, t' \ :\ t, t' \in T :\ t[x] = t'[x] \Rightarrow \langle \exists\ t'' \ :\ \ldots :\ \ldots \rangle \rangle \\
& \equiv \quad \{ \text{ ``guarded rules'' + point removal } \} \\
& \quad T \cdot (\ker x) \cdot T \subseteq (\ker xy) \cdot T \cdot (\ker z) \\
& \equiv \quad \{ \text{ kernel unfolding } \} \\
& \quad T \cdot x^{\circ} \cdot x \cdot T \ \subseteq\ xy^{\circ} \cdot xy \cdot T \cdot z^{\circ} \cdot z \\
& \equiv \quad \{ \text{ ``al-djabr'' } \} \\
& \quad xy \cdot T \cdot x^{\circ} \cdot x \cdot T \cdot z^{\circ} \ \subseteq\ xy \cdot T \cdot z^{\circ}
\end{array}
$$

MVDs PF-transformed

Having completed step 1 of the PF-transform, we now "let the symbols do the work": since y and z are interchangeable — see *Pointfree Foundations for Lossless Decomposition* (forthcoming) — we have

Step 2

$$
\begin{array}{ll}
 & xy \cdot T \cdot x^{\circ} \cdot x \cdot T \cdot z^{\circ} \ \subseteq \ xy \cdot T \cdot z^{\circ} \\
\equiv & \quad \{ \text{ swap } y \text{ and } z \text{ and take converses } \} \\
 & y \cdot T \cdot x^{\circ} \cdot x \cdot T \cdot xz^{\circ} \ \subseteq \ y \cdot T \cdot xz^{\circ} \\
\equiv & \quad \{ \ T = T \cdot T^{\circ} \text{ since } T \text{ is coreflexive } \} \\
 & y \cdot T \cdot x^{\circ} \cdot \pi_1 \cdot xz \cdot T \cdot T^{\circ} \cdot xz^{\circ} \ \subseteq \ y \cdot T \cdot xz^{\circ} \\
\equiv & \quad \{ \text{ please turn over } \}
\end{array}
$$

MVDs PF-transformed

Step 2 (cont.)

$$
\begin{array}{ll}
& y \cdot T \cdot x^\circ \cdot \pi_1 \cdot xz \cdot T \cdot T^\circ \cdot xz^\circ \;\subseteq\; y \cdot T \cdot xz^\circ \\
\equiv & \quad \{ \text{ introduce image and the power-transpose } \} \\
& \Lambda(y \cdot T \cdot x^\circ \cdot \pi_1) \cdot \mathrm{img}\,(xz \cdot T) \;\subseteq\; \Lambda(y \cdot T \cdot xz^\circ) \\
\equiv & \quad \{ \text{ define } \lambda_{f,g} T = \Lambda(f \cdot T \cdot g^\circ)\ ;\ \text{“al-djabr”} \} \\
& \mathrm{img}\,(xz \cdot T) \;\subseteq\; (\lambda_{y,x} T \cdot \pi_1)^\circ \cdot (\lambda_{y,xz} T)
\end{array}
$$

This completes step 2 of the PF-transform. Now we go back to points (step 3):

MVDs PF-transformed

Step 3

$$\mathrm{img}\,(xz \cdot T) \quad \subseteq \quad (\lambda_{y,x} T \cdot \pi_1)^\circ \cdot \lambda_{y,xz} T$$

$\equiv$ { reverse PF-transform (for T coreflexive, $xz \cdot T$ is simple) }

$$\langle \forall\ k\ :\ k\ \mathrm{img}\,(xz \cdot T)\ k :\ (\lambda_{y,x} T \cdot \pi_1)k = (\lambda_{y,xz} T)k \rangle$$

$\equiv$ { reverse PF-transform of the image of $xz \cdot T$ }

$$\langle \forall\ k\ :\ \langle \exists\ t\ :\ t \in T :\ xz(t) = k \rangle :\ (\lambda_{y,x} T \cdot \pi_1)k = (\lambda_{y,xz} T)k \rangle$$

$\equiv$ { rename $k := (b, a)$ and simplify }

$$\left\langle \begin{array}{c} \forall\ a, b\ : \\ \langle \exists\ t\ :\ t \in T :\ (x\ t) = a \wedge (z\ t) = b \rangle : \\ (\lambda_{y,x} T)\ a = (\lambda_{y,xz} T)(a, b) \end{array} \right\rangle$$

MVDs PF-transformed

Recognizing $(\lambda_{y,x} T)a$ as $Y(a)$ (etc) in the original definition by Beeri, Fagin & Howard, we have calculated:

Answer

$$\text{Maier} \quad \equiv \quad \text{Beeri, Fagin \& Howard}$$

No cyclic implication, no points — just "al-djabr" equational reasoning.

Lossless decomposition, Armstrong axioms, etc

More details

PF-transformed version of FD/MVD theory (and its generalization beyond database theory) will become available soon in research paper *Pointfree Foundations for Lossless Decomposition* (forthcoming)

On-going PhD theses

- PF transform and applications — Claudia Necco
- '*G*'alculator — Paulo Silva
- PF program refinement — César Rodrigues

Conclusion

. . . A lot to be done!

Summary

- Invest in **perennial** reasoning strategies
- Need for (complex) theory "re-factoring"
- Shift from "implication first" maths to "let the symbols work" maths
- Rôle of **transforms**, **abstract** notation and abstract patterns (easier to spot *al-djabr* rules)
- Stimulate **elegance** in mathematics (it is effective!)
- Learn with the other engineering disciplines