Constructive Operation Research

Bridging Logic, Mathematics, Computer Science, and Philosophy

Klaus Mainzer
TUM Emeritus of Excellence
TUM Graduate School of Computer Science
Technische Universität München (TUM)

William of Ockham – Early Munich Advocate of CORE

William of Ockham (Ockham 1287 – Munich 1347), Franciscan friar and scholastic philosopher, was a pioneer of *nominalism* and father of *modern epistemology*, because of his strongly argued position that *only individuals* exist, rather than supra-individual universals, essences, or forms. He denied the real existence of metaphysical universals and advocated the *reduction of ontology*.

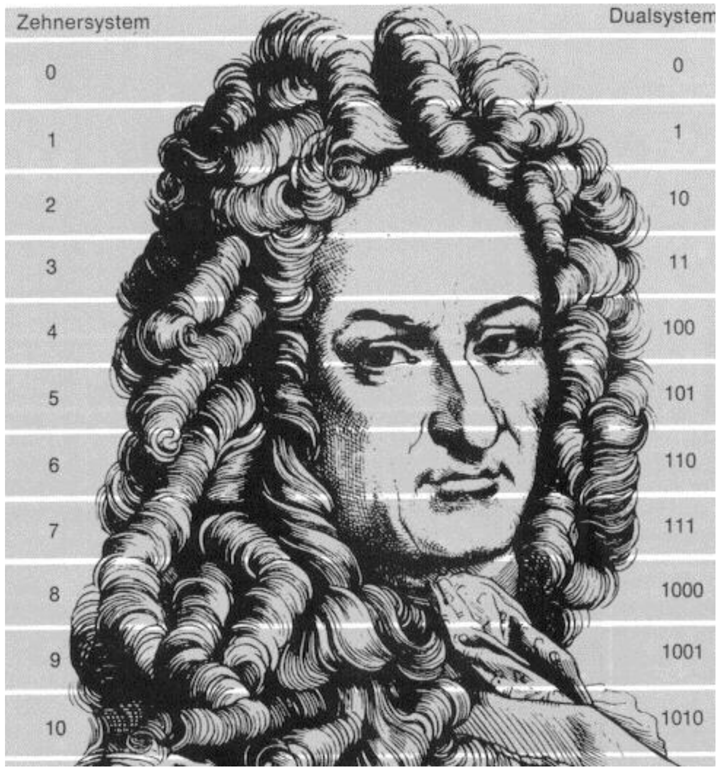
With respect to *constructive operation research*, he argued for *efficient reasoning* with the *principle of parsimony* in explanation and theory building that came to be known as *Occam's Razor*:

This maxim, as interpreted by Bertrand Russell, states that one should always opt for an explanation in *terms of the fewest possible causes, factors, or variables*. He turned this into a concern for *ontological parsimony*.

1. From Turing Machines to Information Systems
2. Constructive Proof Mining
3. From Brouwer‘s Creative Subject to Fan Theorem
4. Constructive Reverse Mathematics
5. Constructive Foundations of Financial Mathematics
6. Real Computing in Numerical Analysis
7. Bridging Logic, Mathematics, Computer Science and Philosophy

1. From Turing Machines to Information Systems

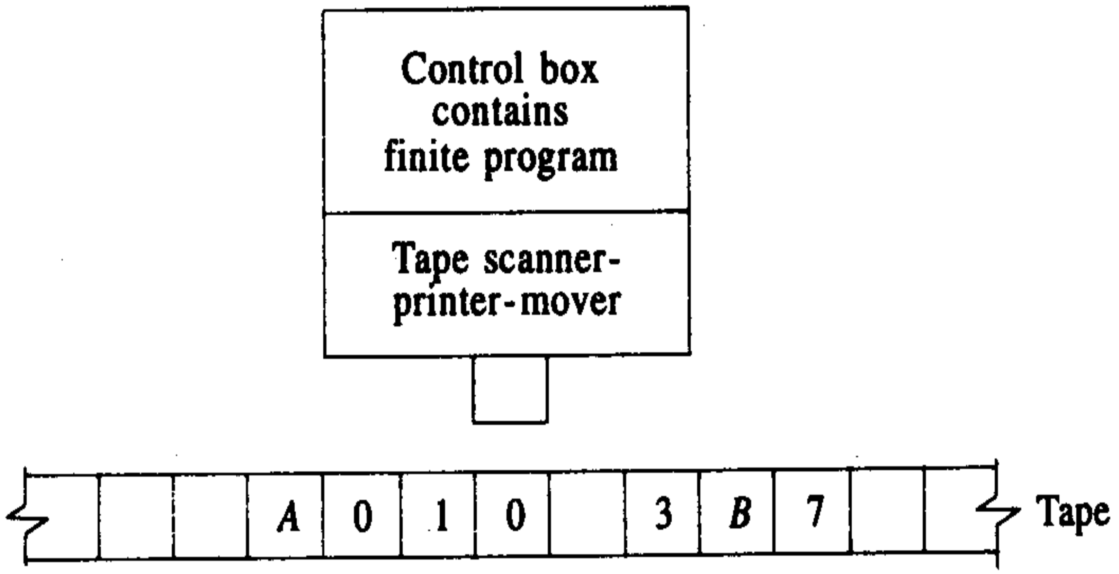
G. W. Leibniz: Knowledge Representation and Universal Computation



In his *"mathesis universalis"* G. W. Leibniz (1646-1716) designed a *universal formal language (lingua universalis)* to *represent human thinking* by *calculation procedures* (*"algorithms")* and to implement them on *mechanical calculating machines*.

An *"ars iudicandi"* should allow every problem to be *decided* by an *algorithm* after *representation* in *numeric symbols*. An *"ars inveniendi"* should allow users to *seek* and *enumerate desired data* and *solutions of problems*. *Struggle on preferences of values* should be decided *"by machines"* (*"ad abacos"*).

Turing's Theory of Computability


Control box contains finite program
Tape scanner-printer-mover
A
0
1
0
3
B
7
Tape

Every *computable procedure (algorithm)* can be realized by a *Turing machine (Church's thesis)*. Every *Turing program* can be simulated by a *universal Turing machine (general purpose computer)*.

A *Turing machine* is a *formal procedure*, consisting of

a) a *control box* in which a *finite program* is placed,

b) a potential *infinite tape*, divided lengthwise into squares,

c) a device for *scanning*, or *printing* on one square of the tape at a time, and for *moving* along the tape or *stopping*, all under the command of the *control box*.

Computability of Functions

A number-theoretical function f **is** ***computable*** **(according to** ***Church's thesis*****) if and only if (iff)** ***f*** **is** ***computable*** **by a** ***Turing machine*** **TM.**

i.e. there is a ***TM-program stopping*** **for** ***numerical inputs*** $x_1, \ldots, x_n$ **as arguments of a function** f **(e.g.,** $x_1=3$, $x_2 = 5$ **of the additional function** $f(x_1, x_2) = x_1 + x_2$**) after finitely many steps and** ***printing*** **the functional value** $f(x_1, \ldots, x_n)$
(e.g. $f(3, 5) = 8$**).**

Approximations of Computable Functionals in Information Systems

In order to describe *approximations* of *abstract objects* like *functionals* by *finite* ones, we use an *information system* with a *countable set A* of *bits* of *data* ("*tokens*") (Scott 1982, Schwichtenberg/Wainer 2012). *Approximations* need *finite sets U* of *data* which are *consistent* with each other. An "*entailment relation*" expresses the fact that the *information* of a *consistent set U* of data is *sufficient* to compute a *bit of information* (*"token"*) :

An *information system* is a *structure* $(A, \mathrm{Con}, \vdash)$ with a *countable set A* ("*tokens*"), *non-empty set* Con of *finite* ("*consistent*") *subsets* of *A* and *subset* $\vdash$ of $\mathrm{Con} \times A$ ("*entailment relation*") with

i. $U \subseteq V \in \mathrm{Con} \Longrightarrow U \in \mathrm{Con}$

ii. $\{a\} \in \mathrm{Con}$

iii. $U \vdash a \in \mathrm{Con} \Longrightarrow U \vdash a$

iv. $a \in U \in \mathrm{Con} \Longrightarrow U \vdash a$

v. $U, V \in \mathrm{Con} \Longrightarrow \forall a \in V\ (U \vdash a) \Longrightarrow (V \vdash b \Longrightarrow U \vdash b)$

The *ideals* („*objects*“) of an *information system* $(A, \mathrm{Con}, \vdash)$ are defined as *subjects x of A* with

i. $U \subseteq x \Longrightarrow U \in \mathrm{Con}$ (x is *consistent*)

ii. $x \supseteq U \vdash a \Longrightarrow a \in x$ (x is *deductively closed*)

<u>Example</u>: The *deductive closure* $\overline{U} := \{a \in A | U \vdash a\}$ of $U \in \mathrm{Con}$ is an *ideal.*

Computable Partial Continuous Functionals of Finite Type

Types are built from *base types* by the formation of *function types* $\rho \rightarrow \sigma$. For every *type* ρ, the *information system* $\mathcal{C}_\rho = (C_\rho, \mathrm{Con}_\rho, \vdash_\rho)$ can be *defined*.
The *ideals* $x \in |\mathcal{C}_\rho|$ are the *partial continuous functionals of type* ρ .
Since $\mathcal{C}_{\rho\rightarrow\sigma} = \mathcal{C}_\rho \rightarrow \mathcal{C}_\sigma$, the *partial continuous functionals* of *type* $\rho \rightarrow \sigma$ will correspond to the *continuous functions* from $|\mathcal{C}_\rho|$ to $|\mathcal{C}_\sigma|$ with respect to the *Scott topology*.

A *partial continuous functional* $x \in |\mathcal{C}_\rho|$ is *computable* iff it is *recursive enumerable* as *set of tokens*.

Partial continuous functionals of *type* ρ can be used as *semantics* of a *formal functional programming language*:
Every *closed term* of *type* ρ in the *programming language* denotes a *computable partial continuous functional* of type ρ, i.e. a *recursive enumerable consistent* and *deductively closed set* of *tokens*.

Another approach uses *recursive equations* to *define computable functionals* (Berger, Eberl, Schwichtenberg 2003).

2. Constructive Proof Mining

Proofs as Verification of Truth

Theorem: ***There are infinitely many prime numbers.***

The predicate $P(x) \equiv' x\ is\ a\ prime\ number'$ can be expressed in a *quantifier-free* way as *primitive recursive predicate.*

Euclid's Proof (reductio ad absurdum): Elements IX Prop. 20; M. Aigner/G.M. Ziegler 2001, pp. 3-6; U. Kohlenbach 2008, p. 15)

Assume **there are** ***finitely many prime numbers*** $p \leq x$

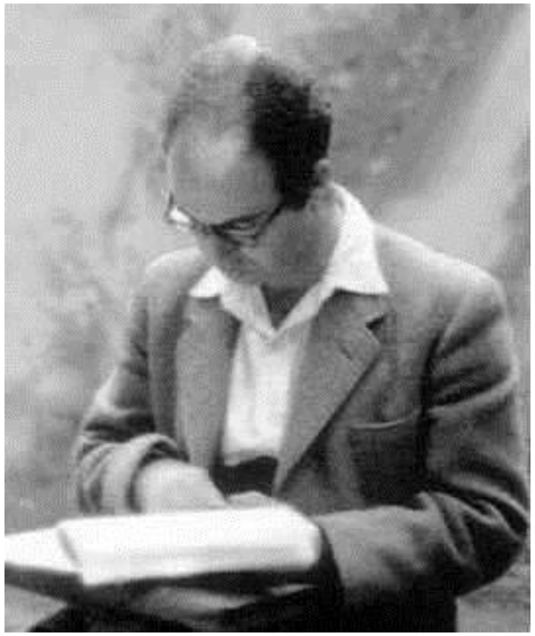
construct $a := 1 + \prod_{\substack{p \leq x \\ p\ prime}} p$

⇨ a ***cannot*** **be** ***prime*** **(because** $a > p$ **for all** $p \leq x$**)**

⇨ a **contains a** ***prime factor*** **(by the decomposition of every number into prime factors)** $q \leq a$ **with** $q > x$ **(otherwise** q **is a prime factor** $q \leq x$ **)**

⇨ ***contradiction to assumption*** **!**

Proofs are more than Verification!



„What more do we know if we have proved a theorem by restricted means than if we merely know that it is true?“

Georg Kreisel: *Unwinding of Proofs*

Ulrich Kohlenbach: *Proof Mining* (cf. „Applied Proof Theory“ 2008, Chapter 2)

Consider an *existential theorem* $A \equiv \exists x\, B(x)$ (closed):

A weaker requirement is to *construct* a *list of terms* $t_1, \dots, t_n$ which are candidates for *A*, such that $B(t_1) \vee \cdots \vee B(t_n)$ holds. More general:

If $A \equiv \forall x \exists y\, B(x, y)$, then one can ask for an *algorithm p* such that

$\forall x\, B\big(x, p(x)\big)$ holds

or – weaker - for a *bounding function b* that $\forall x\, \exists y \leq b(x)\, B(x, y)$.

Euclid's Proof yields a Computable Bounding Function!

Euclid's proof yields the *bound* $g(x) := 1 + x! \geq 1 + \prod_{\substack{p \leq x \\ p\ prime}} p$

$$\Rightarrow g(x) \sim 1 + (2\pi x)^{\frac{1}{2}} \left(\frac{x}{e}\right)^{x} \quad \text{(by Stirling formula)}$$
$$= 1 + \sqrt{2\pi} \cdot e^{x\ln x - x + \frac{1}{2}\ln x}$$

We aim at an *upper bound* on the (*r*+1)-th *prime number* p_{r+1} which *only* depends on *r* instead of $x \geq p_r$:
Euclid's proof yields $p_{r+1} \leq p_1 \cdot \ \ldots \ \cdot p_r + 1$

$$\Rightarrow p_r < 2^{2^r}\ for\ all\ r \geq 1\ (by\ induction\ on\ r).$$

<u>Remark</u>: *Euler's proof yields a much better bound!* (cf. U. Kohlenbach 2008, p. 15)

Brouwer-Heyting-Kolmogorov (BHK) Proof Interpretation of the Intuitionistic Logical Constants

BHK interpretation explains the *meaning of logical constants* in terms of *proof constructions*: (S.C. Kleene 1945; U. Kohlenbach 2008, p. 43)

i. There is *no proof* for $\perp$.

ii. A *proof* of $A \wedge B$ is a pair (*q,r*) of proofs, where *q* is a proof of *A* and *r* is a *proof* of *B*.

iii. A *proof* of $A \vee B$ is a pair of (*n,q*) consiting of an *integer n* and a *proof q* which proves *A* if $n = 0$ and resp. *B* if $n \neq 0$.

iv. A *proof p* of $A \to B$ is a *construction* which *transforms* any *hypothetical proof q* of *A* into a *proof p*(*q*) of *B*.

v. A *proof p* of $\forall x A(x)$ is a *construction* which *produces* for *every construction* c_d of an *element d* of the *domain* a proof $p(c_d)$ of *A*(*d*).

vi. A *proof* of $\exists x A(x)$ is a pair (c_d, q), where c_d is the *construction* of an *element d* of the *domain* and *q* is a *proof* of *A*(*d*).

Computable Functionals instead of Constructive Proofs

The *disadvantage* of the *BHK-interpretation* is the *unexplained notion* of *construction* resp. *constructive proof*. K. Gödel wanted that *constructive proofs* of *existential theorems* provide *explicit realizers*. Therefore, he replaced the notion of *constructive proof* by the more definite and less abstract concept of *computable functionals* of *finite type*.

The notion of a *computable functional of finite type* can be *mathematically defined* as an *ideal* in an *information system*.

But *Gödel's proof interpretation* is largely *independent* of a *precise definition* of *computable functionals* : One only needs certain basic functionals as computable (e.g., primitive recursion in finite types) and their closure under composition.

Following Gödel, every *formula A* is assigned with the *existential formula* $\exists x A_1(x)$ with $A_1(x)$ $\exists$-free. Then, a realizing term *r* with $A_1(r)$ must be *extracted* from a *derivation of A* (‚*Dialectica-Interpretation*' 1958)

Modified Realizability in $\mathcal{L}(\mathrm{E}-\mathrm{HA}^{\omega})$

For each formula *A* of $\mathcal{L}(\mathrm{E}-\mathrm{HA}^{\omega})$, formula $\underline{x}\ mr\, A$ (‚*x modified realizes A*') is defined in $\mathcal{L}(\mathrm{E}-\mathrm{HA}^{\omega})$: (G. Kreisel 1959, 1962; U. Kohlenbach 2008, Chapter 5)

i. $\underline{x}\ mr\ A :\equiv A$ with *empty* tuple $\underline{x}$ and *A prime formula*

ii. $\underline{x},\underline{y}\ mr\ (A \wedge B) :\equiv \underline{x}\ mr A \wedge \underline{y}\ mr \mathrm{B}$

iii. $z^0, \underline{x}, \underline{y}\ mr\ (A \vee B :\equiv ((z =_0 \rightarrow \underline{x}\ mr\ A) \wedge (z \neq_0 0 \rightarrow \underline{y}\ mr\ B)$

iv. $\underline{y}\ mr\ (A \rightarrow B) :\equiv \forall \underline{x}(\underline{x}\ mr\ A \ \rightarrow \underline{y}\underline{x}\ mr\ B)$

v. $\underline{x}\ mr\ (\forall y^{\rho} A(y) :\equiv \forall y^{\rho}(\underline{x}\underline{y}\ mr\ A(y))$

vi. $z^{\rho}, x\ mr\ (\exists y^{\rho} A(y) :\equiv \underline{x}\ mr\ A(z)$

Remark: $\underline{x}\ mr\ t\ A$ (‚ *x modified realizes A with truth* ') is defined analogously to $x\ mr\ A$ except (iv)' $\underline{y}\ mr\ t\ (A \rightarrow B) :\equiv \forall \underline{x}(\underline{x}\ mr\ t\ A \rightarrow \underline{y}\underline{x}\ mr\ t\ B) \wedge (A \rightarrow B)$

Soundness and Characterization of Modified Realization in $\mathcal{L}(\mathrm{E}-\mathrm{HA}^{\omega})$

Let *A* be a *formula* in $\mathcal{L}(\mathrm{E}-\mathrm{HA}^{\omega})$, Δ_{ef} set of $\exists$-*free sentences.* Then the following rule holds:

$\mathrm{E}-\mathrm{HA}^{\omega}+\mathrm{AC}+\Delta_{ef}\vdash A \Rightarrow \mathrm{E}-\mathrm{HA}^{\omega}+\Delta_{ef}\vdash \underline{t}\, mr\, A$

where *t* is a tuple of *terms* of $\mathrm{E}-\mathrm{HA}^{\omega}$ with $FV(\underline{t}) \subseteq FV(A)$ which can be *extracted from proof of A*.

Proof: *Induction* on the length of *A* (A.S. Troelstra 1973; U. Kohlenbach 2008, p. 98)

Let *A* be a formula of $\mathcal{L}(\mathrm{E}-\mathrm{HA}^{\omega})$.
Then $\mathrm{E}-\mathrm{HA}^{\omega}+\mathrm{AC}\vdash A \leftrightarrow \exists \underline{x}(\underline{x}\, mr\, A)$

Proof: *Induction* on the logical structure of *A* (A.S. Troelstra 1973; U. Kohlenbach 2008, p. 100)

Program Extraction of Modified Realization

Let $\forall x^\rho \exists y^\tau A(x,y)$ be a sentence of $\mathcal{L}(\mathrm{E}-\mathrm{HA}^\omega)$ with types ρ, τ, Δ_{ef} set of $\exists$-*free sentences*, and IP_{ef} *independence-of-premise scheme*. Then the following rule holds:

$$\mathrm{E}-\mathrm{HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_{ef} + \Delta_{ef} \vdash \forall x^\rho \exists y^\tau A(x,y) \Rightarrow$$
$$\mathrm{E}-\mathrm{HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_{ef} + \Delta_{ef} \vdash \forall x^\rho A(x,tx),$$

where t is a *closed term* of $\mathcal{L}(\mathrm{E}-\mathrm{HA}^\omega)$ which is *extracted from* a *proof* of the *premise* by *modified realizability*.

It follows:

$\mathcal{L}^\omega \vDash \Delta_{ef} \Rightarrow \mathcal{L}^\omega \vDash \forall x^\rho A(x,tx)$.

<u>Proof</u>: By *soundness* and *characterization* theorem (cf. U. Kohlenbach 2008, p. 100)

Automatic Program Extraction with MINLOG

MINLOG is an *interactive proof system* which is equipped with *tools* to extract *functional programs* directly from *proof terms*. The system is supported by *automatic proof search* and *normalization* by *evaluation* as an *efficient term rewriting device*.

<u>Example: Existence proof for "list reversal"</u>

Write vw for the result $v * w$ of appending the list w to the list v,
Write vx for the result $v * x$ of appending one element list x: to the list v,
Write xv for the result $x :: v$ of writing one element x in front of a list v
Assume Init Rev: R(nil, nil)
GenRev: $\forall v, w, x\ (Rvw \rightarrow R(vx, xw))$

<u>Proposition</u>: $\forall v \in T\ \exists w \in T\ Rvw$ („Existence of list *w* with reverse order of given list *v*")
<u>Proof</u>: Induction on the length of v

Link of Proof Mining to Computer Science: *Practical Use of Soundness Proofs*

Customers want *software* which *solves a problem*. Thus, they require a *proof* that it *works*. *Suppliers* answer with a *proof of the existence of a solution* to the specification of the problem. The proof has been *automatically extracted* from the *formal specification* of the problem by a *proof mining software* (e.g., MINLOG).

But, the question arises whether the *extraction mechanism of the proof* is *itself*, *in general*, *correct*. The *soundness theorem* guarantees that *every formal proof* can be realized by a *normalized extracted term*.

Application of Proof Mining to Mathematics
(cf. U. Kohlenbach 1993; 2008, Chapter 15.1)

Task:

Constant solutions $x \in K$ **for equations**

$$A(x) :\equiv (F(x) = 0)$$

with *K compact metric space* and $F: K \to \mathbb{R}$ *continuous function*.

Solution:

1) ***Construct approximate solutions*** $x_n \in K$ **satisfying**

$$A_n(x_n) :\equiv (|F(x_n)| < 2^{-n})$$

2) ***Conclude*** **that either** $(x_n)_{n\in\mathbb{N}}$ **itself or some *subsequence converges* to a *solution* of** $A(x)$ **, using the *compactness* of *K* and the *continuity* of *F*.**

Proof Mining:

a) **If *F* has *exactly one root*** $\hat{x} \in K$**, *proof-theoretic analysis* of a given *proof of uniqueness* of** $\hat{x}$ **can be applied to *extract* an *effective rate of convergence* under quite general circumstances.**

b) **If the *solution*** $\hat{x}$ **is *not necessary unique*, one often *cannot effectively* obtain a *solution* but weaker tasks like obtaining *effective rates* of *asymptotic regularity* might be solvable.**

3. From Brouwer‘s Creative Subject to FanTheorem

Intuitionistic Philosophy of Creative Subject

According to Brouwer, *mathematical truth* is founded by *construction of a creative subject* . Following Kant, *mathematical construction* can only be realized in a *finite process, step by step in time* like counting in arithmetic. Thus, for Brouwer, *mathematical truth* depends on *finite stages of realization in time by a creative subject* (in a definition of Kripke and Kreisel 1967) :

The creative subject has a proof of proposition *A* at stage *m* $(\sum \vdash_m A)$ **iff**

(CS1) For any proposition *A* , $\sum \vdash_m A$ **is a *decidable* function of *A* , i.e.**
$\forall \boldsymbol{x} \in \mathbb{N} \;\; (\sum \vdash_{x} \boldsymbol{A} \vee \neg \sum \vdash_{x} \boldsymbol{A})$

(CS2) $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{N} \; (\sum \vdash_{x} \boldsymbol{A} \rightarrow (\sum \vdash_{x+y} \boldsymbol{A})$

(CS3) $\exists \boldsymbol{x} \in \mathbb{N} \; (\sum \vdash_{x} \boldsymbol{A}) \leftrightarrow \boldsymbol{A}$

A weaker version of CS3 is G. Kreisel's "*Axiom of Christian Charity*" (1967)

(CC) $\neg\exists x \in \mathbb{N} \; (\sum \vdash_{x} A) \rightarrow \neg A.$

Fan Principle and Fan Theorem

The *fan principle* states that for every fan *T* in which every *branch* at some point satisfies a property *A*, there is a *uniform bound* on the depth at which this property is met. Such a property is called a *bar* of *T*.

FAN Principle:

$$\forall\alpha \in T\ \exists n\ A\big(\alpha(\overline{n})\big) \rightarrow \exists m\ \forall\alpha \in T\ \exists n \leq m\ A\big(\alpha(\overline{n})\big)$$

with α *choice sequences* and $\alpha(\overline{n})$ the *initial segment* of α with the first m elements.

FAN Theorem:

Every *continuous real function* on a *closed interval* is *uniformly continuous*.

Proof: *Fan Principle*

Brouwer‘s Bar Principle for the Universal Spread

The *bar principle* provides *intuitionistic mathematics* with an *induction principle* for trees. It expresses a *well-foundedness principle* for *spreads* with respect to *decidable properties* :

$\forall\alpha\,\forall n\, A(\alpha(\overline{n})) \vee \neg A(\alpha(\overline{n}))) \wedge \forall\alpha\,\exists n\, A(\alpha(\overline{n})) \wedge \forall\alpha\,\forall n\, A(\alpha(\overline{n})) \to B(\alpha(\overline{n}))) \wedge$
$\forall\alpha\,\forall n(\forall m\, B(\alpha(\overline{n})\cdot m) \to B(\alpha(\overline{n})))) \to B(\varepsilon)$

with ε empty sequence.

The Fan Principle is not universally true in Classical Mathematics

The *simplest form* of the *fan principle* is the *contraposition* of *binary König's lemma* :

$\mathrm{FAN_{KL}}$:

$$T(f) \wedge \forall b \leq_1 1\, \exists x^0 \left(f\left(\overline{b}x\right) \neq_0 0\right) \rightarrow \exists x^0 \forall b \leq_1 1\, \exists \tilde{x} \leq x \left(f\left(\overline{b}\tilde{x}\right) \neq_0 0\right)$$

where $T(f)$ expresses that *f* is the *characteristic function* of *binary tree* and $1 := \lambda x^0 . 1^0$

In this form, the *fan principle* is *classically valid* as WKL (Weak König's Lemma). In *intuitionistic mathematics*, there is also the much more *general form* :

FAN:

$$\forall f \leq_1 1\, \exists x^0\, A(f, x) \rightarrow \exists x^0\, \forall f \leq_1 1\, \exists \tilde{x} \leq_0 x\, A(f, \tilde{x})$$

with arbitrary formula *A*.

FAN is *inconsistent* with *classical logic* :

$$\forall f \leq_1 1\, \exists x^0 \underbrace{(\exists y (f(y) =_0 0) \rightarrow f(x) =_0 0)}_{A(f,\, x)}$$ *classical valid*

$$\underset{FAN}{\Longrightarrow} \quad \exists x^0 \forall f \leq_1 1\, (\exists y (f(y) =_0 0) \rightarrow \exists y \leq x\, (f(y) =_0 0))$$ *classically false .*

4. Constructive Reverse Mathematics

Reverse Mathematics in Antiquity

Since Euclid (Mid-4th century – Mid 3rd century BC), *axiomatic mathematics* has started with *axioms* to *deduce* a *theorem*. But the "*forward*" *procedure* from *axioms* to *theorems* is not always obvious. How can we *find appropriate axioms* for a proof starting with a *given theorem* in a „*backward*" (*reverse*) *procedure*?

Pappos of Alexandria (290-350 AC) called the "*forward*" *procedure* as "*synthesis*" with respect to Euclid's *logical deductions* from *axioms* of geometry and *geometric constructions* (Greek: "*synthesis*") of corresponding figures. The *reverse search procedure of axioms* for a given theorem was called "*analysis*" with respect to *decomposing* a *theorem* in its *necessary* and *sufficient conditions* and the *decomposition* of the *corresponding figure* in its *building blocks*.

Classical Reverse Mathematics

Reverse mathematics is a modern *research program* to determine the *minimal axiomatic system* required to *prove theorems*. In general, it is *not possible* to start from a *theorem* τ to prove a *whole axiomatic subsystem* T_1. A *weak base theory* T_2 is required to *supplement* τ:

If $T_2 + \tau$ can prove T_1, this *proof* is called a *reversal*.
If T_1 *proves* τ and $T_2 + \tau$ is a *reversal*, then T_1 and τ are said to be *equivalent over* T_2.

Reverse mathematics allows to determine the *proof-theoretic strength* resp. *complexity* of *theorems* by *classifying* them with respect to *equivalent theorems* and *proofs*. Many *theorems* of *classical mathematics* can be *classified* by *subsystems* of *second-order arithmetic* $\mathbb{Z}_2$ with *variables* of *natural numbers* and *variables* of *sets of natural numbers*.

The Subsystems of Second-Order Arithmetics $\mathcal{Z}_2$

Arithmetical formulas can be *classified* according to the *arithmetical hierarchy* $\sum_n^0, \prod_n^0$, and Δ_n^0. We can distinguish $\sum_n^0, \prod_n^0$, and Δ_n^0- schemas of *induction* and *comprehension*. That is also possible for the *analytical hierarchy* $\sum_n^1, \prod_n^1$, and Δ_n^1

A *structure* of an (*arithmetical*) *set M* defines its *variables* and *non-logical symbols* (constants, operations) satisfying relations between *variables*: e.g., $\mathbb{Q} = (M, +_{\mathbb{Q}}, -_{\mathbb{Q}}, \cdot_{\mathbb{Q}}, \mathbf{0}_{\mathbb{Q}}, \boldsymbol{I}_{\mathbb{Q}}, <_{\mathbb{Q}}, =_{\mathbb{Q}})$ *structure* of *rational numbers*.

A *model* of a *set* of (*arithmetical*) *formulas* is a *structure* with the *same non-logical symbols* and all *formulas* in the set are in the *model* as well.

The *arithmetical* and *analytical hierarchies* yield *classifications* of *axiomatic subsystems* of $\mathcal{Z}_2$ with *increasing proof-theoretic power* and corresponding *structures* of $\mathcal{Z}_2$-models.

Distinguished $\mathcal{Z}_2$-Subsystems of Reverse Mathematics

The *following subsystems* have *increasingly proof-theoretic power* starting with the *weakest subsystem* RCA_0:
Recursive Comprehension Axiom (RCA_0):

$\mathrm{RCA}_0 = \mathit{Peano\ axioms} + \Sigma_1^0 - \mathit{induction} + \Delta_1^0 -$ *comprehension*.

$\omega -$*model* *S* satisfies *(1)* $S \neq \emptyset$

(2) $A \in S$ and $B \in S$ imply $A \oplus B \in S$

(3) $A \in S$ and $B \leq_T A$ (*B* Turing-reducible to *A*) imply $B \in S$

The *minimum* $\omega -$*model* of RCA_0 is the *computable sets*.

Examples of theorems provable in RCA_0 :

– *Intermediate value theorem*

– *Soundness theorem*

– *Gödel's completeness theorem*

– *Baire category theorem*

Distinguished $\mathcal{Z}_2$-Subsystems of Reverse Mathematics

Arithmetical Comprehension Axiom (ACA_0):

$$\mathrm{ACA}_0 = \mathrm{RAC}_0 + \mathit{Arithmetical\ Comprehension}$$

ACA_0 has a *comprehension scheme* for *all arithmetical formulas* of the *arithmetical hierarchy*.

Examples of theorems equivalent over RCA_0 :

– ACA_0

– *Sequential compactness of* [0,1] and *compact metric spaces*

– *Existence of the strong algebraic closure of a countable field*

– *König's lemma for subtrees of* $\mathbb{N}^{\mathbb{N}}$

– *Least upper bound principle for sequences of real numbers.*

Distinguished $\mathcal{Z}_2$-Subsystems of Reverse Mathematics

Weak $\sum_1^1$ –choice ($\sum_1^1$ –WC):

$\sum_1^1$ –WC $= \mathrm{RCA}_0 + \boldsymbol{Weak} \sum_1^1 -$ choice

If $\forall n \exists! x\, \varphi(n,x)$, then $\exists\{y_n | n < \omega\}\, \forall n\, \varphi(n, y_n)$ with φ *arithmetic*

Δ_1^1 – comprehension ($\Delta_1^1 - \mathrm{CA}_0$):

$\Delta_1^1 - \mathrm{CA}_0 \;= \mathrm{RCA}_0 + \Delta_1^1 -$ comprehension

$\Delta_1^1 - \mathrm{CA}_0$ has a *comprehension scheme* for all Δ_1^1 –*formulas* of *hyperarithmetics*

$\sum_1^1$ –choice ($\sum_1^1$ –C):

$\sum_1^1 - \mathrm{C} \;= \mathrm{RCA}_0 + \sum_1^1 -$ choice

Like $\sum_1^1$ –WC *without assuming uniqueness*.

Arithmetic transfinite recursion (ART_0):

$\mathrm{ART}_0 = \mathrm{RCA}_0 + \boldsymbol{arithmetic\ comprehension}$ (*iterated transfinitely*)

$\mathcal{Z}_2$ - Subsystems and Philosophical Research Programs

The *five* most commonly used $\mathcal{Z}_2$ - *subsystems* in *reverse mathematics* correspond to *philosophical programs* in *foundations of mathematics* with *increasing proof-theoretic power* starting with the *weakest* RCA_0*-subsystem*.

RCA_0:	*Turing's computability*
WKL_0:	*Hilbert's finitistic reductionism*
ACA_0:	*Weyl's & Lorenzen's predicativity*
ATR_0:	*Friedman's & Simpson's predicative reductionism*
$\prod_1^1 - \mathrm{CA}_0$:	*impredicativity*

$\Delta_1^1 - CA_0$ yields *systems* of *hyperarithmetic analysis* (Feferman et al.) with Δ_1^1-*predicativism*:

T is a *theory* of *hyperarithmetic analysis* iff

i. its ω-*models* are *closed* under *joins* and *hyperarithmetic reducibility*

ii. it *holds* in $\mathrm{HYP}(x)$ for all *x*

Constructive Reverse Mathematics

Classical reverse mathematics (Friedmann/Simpson) uses *classical logic* and *classification of proof-theoretic strength* with RCA_0 (Δ_1^0-*recursive comprehension*) as *weakest subsystem*.

Constructive reverse mathematics (Ishihara et al.) uses *intuitionistic logic* and *Bishop's constructive mathematics* (BISH) as *weakest subsystem* of a *constructive classification* (Bishop/Bridges/Vita/Richman)

BISH = $\mathcal{Z}_2$ + *Intuitionistic Logic* + *Axioms of Countable*, *Dependent* and *Unique Choice*

Intuitionistic Mathematics (Brouwer, Heyting et al.):

INT = BISH + *Axiom of Continuous Choice* + *Fan Theorem*

Constructive Recursive Mathematics (Markov et al.):

RUSS = BISH + *Markov's Principle* + *Church's Thesis*

Classical Mathematics (Hilbert et al.):

CLASS = BISH + *Principle of Excluded Middle* + *Full Axiom of Choice*

5. Constructive Foundations of Financial Mathematics

Fundamental Theorem of Asset Pricing in Financial Mathematics

According to the *fundamental theorem of asset pricing*, in *arbitrage-free* (“*fair*”) *markets*, the “*fair*” *prices* are given by expectations under *equivalent martingale measures*. A *martingale* can be illustrated by a *fair game* where knowledge of past events never helps predict the mean of the future winnings.

Definition of a discrete-time martingale:

A discrete-time martingale is a discrete-time stochastic process $X_1, X_2, X_3, \ldots$ that satisfies for any time n

$$E(|X_n|) < \infty$$

$$E(X_{n+1} | X_1, \ldots, X_n) = X_n \,.$$

In short, the *fundamental theorem of asset pricing states* that there does not exist an arbitrage strategy (*arbitrage-free market*) iff there exist an *equivalent martingale measure* (Föllmer/Schied 2012).

Lemma:

There exists an *arbitrage strategy* ξ $\Leftrightarrow$ There exists a vector $\mu \in \mathbb{R}^m$ such that

$$\mu \cdot (C - \pi) \in \mathcal{Y}_n$$

with $C - \pi := (c_{ij} - \pi_i)_{i=2,\ldots,m+1; j=1,\ldots,n}$, $\pi = (\pi_1, \ldots, \pi_m, \pi_{m+1}) \in \mathbb{R}^{m+1}$ *prices of the assets* at present time 0, and *price* c_{ij} *of the i-th asset in case j* with $C = (c_{ij})$.

Therefore, *excluding the existence of arbitrage strategies* (*arbitrage-free markets*) can be formalized by $\{\xi \cdot A | \xi \in \mathbb{R}^m\} \cap \mathcal{Y}_n = \emptyset$
with $A := C - \pi$ and $\mathcal{Y}_n := \{(x_1, \ldots, x_n) \in \mathbb{R}^n | \sum_{i=1}^n x_i > 0 \; and \; 0 \leq x_i \text{for } all \; i\}$.

Existence of an equivalent martingale measure means formally that there is a vector $p \in \mathcal{P}_n$ with $C \cdot p = \pi$ and $\mathcal{P}_n := \{(x_1, \ldots, x_n) \in \mathbb{R}^n | \sum_{i=1}^n x_i = 1 \text{ and } 0 < x_i \text{for all } i\}$.

Fundamental Theorem of Asset Pricing :

FTAP

$$\{\xi \cdot A | \xi \in \mathbb{R}^m\} \cap \mathcal{Y}_n = \emptyset \Leftrightarrow \exists p \in \mathcal{P}_n \, (A \cdot p = 0)$$

Constructive Equivalence of FTAP with Markov‘s Principle (RUSS)

Markov‘s Principle :

MP

$\neg\neg\exists n\, A(n) \rightarrow \exists n A(n)$ **with quantifier-free** $A(n)$,

equivalently in terms of ***real numbers*****:**

$\forall x \in \mathbb{R}(\neg(x = 0) \rightarrow |x| > 0)$

Separating Hyperplane Principle :

SEP

Let $(H, \langle , \rangle)$ be a *Hilbert space* with $\mathcal{C} \subseteq H$ *convex*, *closed*, and *located*.
Let $x_1, \dots, x_n \in H$ with $\mathcal{C} \cap C(x_1, \dots, x_n) = \emptyset$.

$\Rightarrow$ **There exists $\varepsilon > 0$ and $p \in H$ such that**
$\langle p, x - c\rangle \geq \varepsilon$ **for all $x \in C(x_1, \dots, x_n)$ and $c \in \mathcal{C}$.**

SEP $\Leftrightarrow$ FTAP $\Leftrightarrow$ MP (Berger/Svindland 2016)

Intuitionism and Constructivism in Financial Mathematics

<u>Brouwer‘s Intuitionistic Fan Theorem</u> :

FAN — **Every *uniformely continuous function* $f : [0,1] \to \mathbb{R}^+$ has *positive infimum*.**

<u>Constructive Version of Fan Theorem</u> :

FAN$_{\text{con}}$ — **Every *uniformely continuous convex function* $f : X \to \mathbb{R}^+$ has *positive infimum* (with X convex hull of finitely many vectors).**

<u>Constructive Foundation of Fundamental Theorem of Asset Pricing</u> :

$$\mathrm{FAN}_{\mathrm{con}} \Rightarrow \mathrm{SEP} \Rightarrow (\mathrm{FTAP} \Leftrightarrow \mathrm{MP})$$

Value of Risk in Financial Mathematics

VaR

Given some *confidence level* $\alpha \in (0,1)$, the *Value at Risk* (VaR) of the *portfolio value X* at the *confidence level* α is given by the smallest number $m \in \mathbb{R}$ such that the probability of a loss is not larger than the *confidence level* α:

$$\mathrm{VaR}_\alpha(X) = \inf\{m \in \mathbb{R} | P(X+m<0) \leq \alpha\} \quad .$$

***Value at Risk* (VaR) is *positively homogeneous*, but *not* in general a *coherent risk measure* (sub-additivity). Hence, it is *not convex*. An immediate consequence is that VaR might *discourage diversification*.**
VaR is, however, *coherent*, under the *assumption of normally distributed losses* when the *portfolio* value is a *linear function of the asset* prices. In this case VaR becomes *equivalent* to a *mean-variance approach* where the risk of a *portfolio* is measured by the *variance of the portfolio's return*. The *average Value at Risk* (AVaR) is defined as

AVaR

$$\mathrm{AVaR}_\lambda = \frac{1}{\lambda}\int_0^\lambda \mathrm{VAR}_\alpha(X)\,d\alpha \text{ at level } \lambda \in (0,1]$$

Definition of Coherent Risk Measure

A functional $\rho : L \to \mathbb{R}$ is said to be a *coherent risk measure* for L if it satisfies the following properties for values $X_1, X_2 \in L$ of *portfolios*:

Monotonicity: If $X_1, X_2 \in L$ and $X_1 \leq X_2$, then $\rho(X_1) \leq \rho(X_2)$.

Sub-additivity: If $X_1, X_2 \in L$, then $\rho(X_1 + X_2) \leq \rho(X_2) + \rho(X_2)$.

The risk of two portfolios together cannot get any worse than adding the two risks separately. This is the *diversification principle*.

Positive homogeneity: If $\alpha \geq 0$ and $X \in L$, then $\rho(\alpha X) = \alpha \rho(X)$.
If you double your portfolio then you *double your risk*.
Translation invariance: If $m \in R$ and $X \in L$, then $\rho(X + m) = \rho(X) - m$.

***Sub-additivity* and *positive homogeneity* can be replaced by the notion of *convexity*:**

If $X_1, X_2 \in L$ and $0 \leq \lambda \leq 1$, then $\rho(\lambda X_1 + (1 - \lambda) X_2) \leq \lambda \rho(X_1) + (1 - \lambda) \rho(X_2)$

Definition of Convex Risk Measure

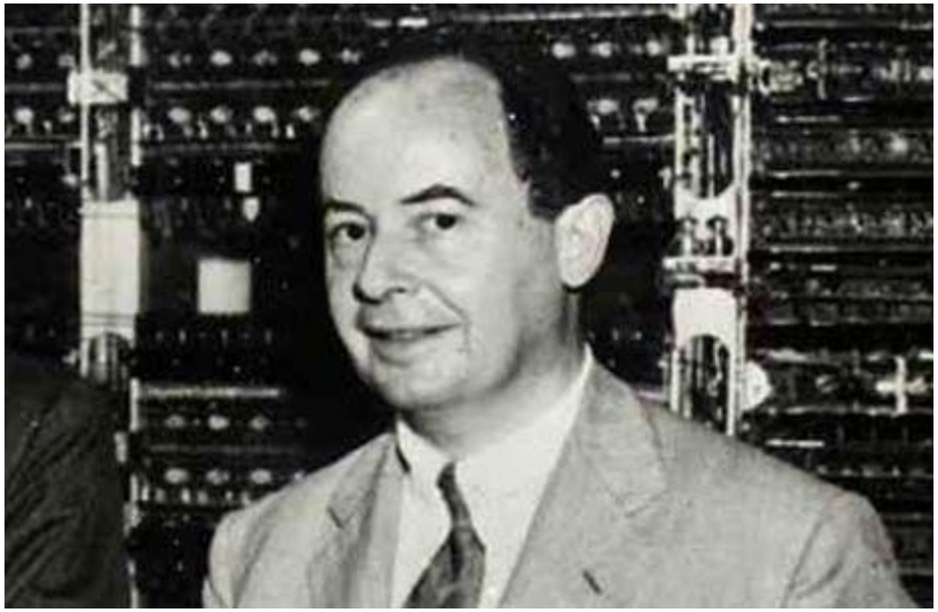
A *dual representation* of a *convex risk measure* (Föllmer/Schied 2008) computes the *worst case expectation* taken over *all models* Q and *penalized* by $\pi(Q)$. The *class M* of *possible probabilistic models* is a *set of probability measures* such that the *expectation* $E_Q(X)$ is *well defined* for *all models* Q and *portfolios* X:

$$\rho(X) = sup_{Q\in M}\ (E_Q(-X) - \pi(Q))$$

The *class of models* serves as *stress tests*. One does *not rely* on a *fixed model*, but chooses the *sure side* for every position and focuses on the corresponding *worst case model*. Thus, the *model ambiguity* is explicitly considered during the procedure.

6. Real Computing in Numerical Mathematics

John von Neumann on Formal Logic, Automata, and Mathematics

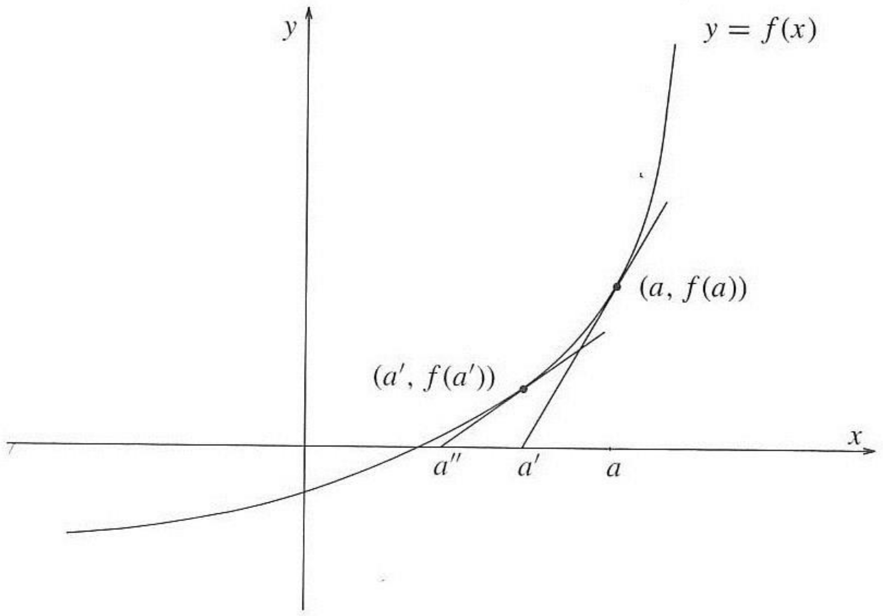


„There exists today a very elaborate system of formal logic, and specifically, of logic as applied to mathematics. This is a discipline with many good sides, but also serious weakness …

The reason for this is that it deals with rigid, all-or-none concepts, and has very little contact with the continuous concept of the real or of the complex number, that is, with mathematical analysis. Yet analysis is the technically most successful and best-elaborated part of mathematics…"

John von Neumann, Hixon Symposium Lecture 1948

Search Algorithm over Real Numbers $\mathbb{R}$ and Complex Numbers $\mathbb{C}$



Newton's method **is a typical** ***search algorithm*** **of** ***numerical analysis*** **and** ***scientific computation*****. It is an** ***iterative method*** **to** ***approximate*** **the roots of** ***nonlinear*** **equations.**

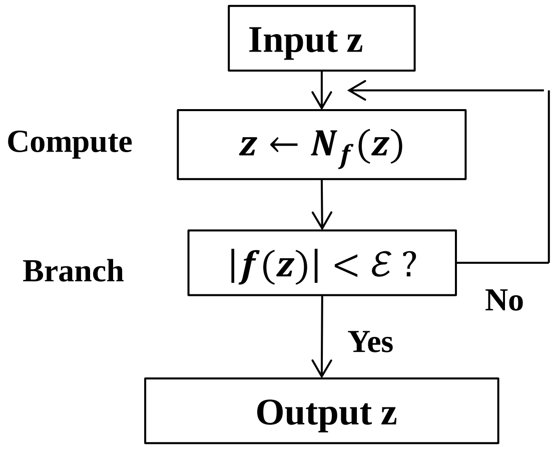
Given an ***initial approximation a*** **to a root of the polynomial equation $f(z) = 0$. Newton's method replaces** ***a*** **by the** ***exact solution a'*** **of the** ***best linear approximation*** **to** ***f*** **which is given by the tangent to the graph of** ***f*** **at point $(a, f(a))$. With** ***a'*** **the approximation is iterated to generate** ***a''*****, etc.**

The ***algorithm*** **is defined by** ***Newton's endomorphism*** **$N_f: \mathbb{C} \longrightarrow \mathbb{C}$ with**

$$N_f(z) = z - \frac{f(z)}{f'(z)}$$

Newton's method is not generally convergent.

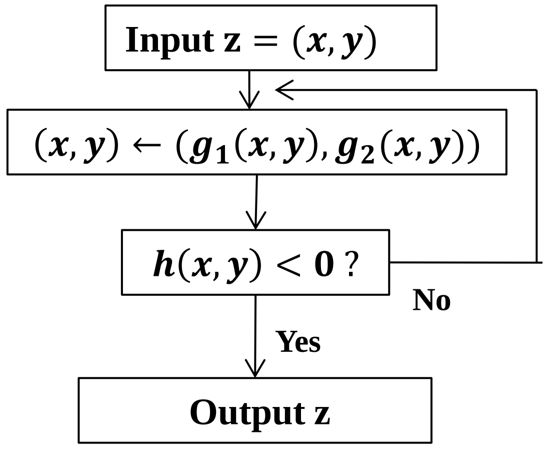
Search Machines over Real Numbers $\mathbb{R}$ and Complex Numbers $\mathbb{C}$



Newton's machine **is represented by a *finite directed graph* with *four types of nodes - input, computation, branch,* and *output* - each with associated functions and conditions on *incoming* and *outcoming edges* (L. Blum, F. Cucker, M. Shub, S. Smale 1998, Chapter 1)**

For an *input* z_0 the machine generates the *orbit*
$$z_0, z_1 = N_f(z_0), z_2 = N_f(z_1), \dots, z_{k+1} = N_f(z_k) = N_f^{k+1}(z_0), \dots$$
The *stopping rule* is "*stop if* $|f(z_k)| < \mathcal{E}$ *and output* z_k." If $N_f(z_k)$ is *undefined* at some stage, there is no output.

The *machine* will *not in general halt* on *all inputs* (e.g. infinite loops). The set $\Omega \subset \mathbb{C}$ is called the *halting set of the machine* iff it contains all inputs for which the machine halts with an output. The input-output map ϕ is defined on Ω with $\phi: \Omega \longrightarrow \mathbb{C}$.



***Newton's machine* is actually a *machine over* $\mathbb{R}$ with $\mathbb{C} = \mathbb{R}^2$ as *input, output* and *state space*. *Newton's endomorphism* (computation node) is given by a *rational function* (quotient of two polynomials)**
$$g = (g_1, g_2): \mathbb{R}^2 \longrightarrow \mathbb{R}^2$$
with $g_1(x, y) = \text{Re } N_f(x + iy)$ and $g_2(x, y) = \text{Im } N_f(x + iy)$.

Computing Endomorphism of Graphic Machine *M* over Ring *R*

Let $\mathcal{N} = \{1, \dots, N\}$ be the *set of nodes* of *M* (with 1 the *input node* and *N* the *output node*) and *S* its *state space*. $\mathcal{N} \times S$ is called the *full state space* of the machine with the *computing endomorphism*

$$H\colon \mathcal{N} \times S \longrightarrow \mathcal{N} \times S,$$

that is, *H* maps each *node/state pair* (η, x) to the *unique next node/state pair* (η', x') determined by the *graph* of *M* and the *associated maps*

(Blum/ Shub/ Smale 1989).

Let $\gamma (= \gamma_x)$ be the *computation path* $\eta^0 = 1, \eta^1, \dots, \eta^k, \dots$ and let $\gamma(k)$ be the *initial computation path* $(\eta^0, \eta^1, \dots, \eta^k) \in \mathcal{N}^{k+1}$ of γ with *length k*. We call $\mathcal{V}_{\gamma(k)} = \{x' \in \mathcal{I}_M | \gamma_{x'}(k) = \gamma(k)\}$ the *initial path set* with all inputs whose computation paths coincide with γ for the first *k* steps.

Computation Paths and Branching Conditions

Initial path sets **are characterized by the** ***branching conditions*** **along the** ***path*** $\boldsymbol{\gamma(k)}$ **with two** ***sets of left and right step-k branching functions***

$$L_{\gamma(k)} = \{f_{\gamma(k')} \mid k' < k, k' \text{ branch step in } \gamma, \text{and } \eta^{k'+1} = \beta^{-}(\eta^{k'})\}$$
$$R_{\gamma(k)} = \{f_{\gamma(k')} \mid k' < k, k' \text{ branch step in } \gamma, \text{and } \eta^{k'+1} = \beta^{+}(\eta^{k'})\}.$$

Then, in case ***R*** **is** ***ordered***, **the** ***initial path set*** **is**

$$\mathcal{V}_{\gamma(k)} = \{x \in \mathcal{I}_M \mid f(x) < 0, g(x) \geq 0, f \in L_{\gamma(k)}, g \in R_{\gamma(k)}\}$$

or, in case ***R*** **is** ***unordered***,

$$\mathcal{V}_{\gamma(k)} = \{x \in \mathcal{I}_M \mid f(x) \neq 0, g(x) = 0, f \in L_{\gamma(k)}, g \in R_{\gamma(k)}\}$$

A subset ***S*** **of** $\boldsymbol{R^n}$ **is** ***basic semi-algebraic*** **over** ***R*** **in the** ***ordered case*** **(or** ***basic quasi-algebraic***, **in the** ***unordered case*****) if** ***S*** **is the set of elements in** $\boldsymbol{R^n}$ **that satisfy a** ***finite system of polynomial equalities*** **and** ***inequalities*** **over** ***R*****. A** ***semi-algebraic*** **(or** ***quasi-algebraic*****) set is a** ***finite union*** **of** ***basic semi-algebraic*** **(or** ***basic quasi-algebraic*****) sets.**

Halting Sets as Semi-Algebraic Sets

(1) If R is an *ordered ring* or *field*, then the *initial path set* $\mathcal{V}_{\gamma(k)}$ is *basic semi-algebraic* (or *basic quasi-algebraic* in the *unordered case*).

(2) If $\gamma_1(k) \neq \gamma_2(k)$, then $\mathcal{V}_{\gamma_1(k)} \cap \mathcal{V}_{\gamma_2(k)} = \emptyset$.

Let $\Gamma_T = \{\gamma_x(T) | T_M(x) \leq T \ for \ x \in \mathcal{I}_M\}$ be the *set of time-T halting paths* with *halting time* $T_M(x)$, i.e. the *least* such T with $T_M\colon \Omega_M \longrightarrow \mathbb{Z}^+$.
The *set of halting paths* of M is $\Gamma_M = \bigcup_{T<\infty} \Gamma_T$ and the *set of minimal halting paths* $\Gamma'_M = \{\gamma \in \Gamma_M | N$ occurs only once in $\gamma\}$.

For *any machine M* over R the *halting set* $\Omega_M = \bigcup_{\gamma \in \Gamma'_M} \mathcal{V}_\gamma$ is a *countable disjoint union* of *basic semi-algebraic* (resp. basic quasi-algebraic) *sets*.

Computability and Decidability over Ring *R*

For $l, m \leq \infty$, a (partial) map $\varphi\colon R^l \to R^m$ is *computable* over *R* iff there is a *machine M* over *R* such that its *halting set* $\Omega_M = \Omega_\varphi$, the *domain* of φ, and $\phi_M(x) = \varphi(x)$ for all $x \in \Omega_\varphi$ and *input-output map* $\phi_M\colon \Omega_M \to \mathcal{O}$ (*output space*).

A set $Y \subset R^n$ is called *recursive enumerable* over *R* iff $Y = \Omega_M$ (*halting set*) for some *machine M* over *R*. (*Y* is *not computable* for $R = \mathbb{R}$.) It is said to be *decidable* iff it and its *complement* are *both recursive enumerable* over *R*.

Undecidability of Newton's Method

Theorem: The *set of points* that *converge under Newton's method* is generally *undecideable* over $\mathbb{R}$.

Proof: It is sufficient to use the *cubic* $f(x) = x^3 - 2x + 2$ to obtain the *undecidability result.*
It is known that the *set of points* that do *not converge* to a *root* of *f* under *iteration by Newton's method* is exactly a *Cantor set*, i.e. *uncountable*.
A *Cantor set cannot* be the *countable union of semi-algebraic sets*.

(L. Blum, F. Cucker, M. Shub, S. Smale 1998, p. 55)

7. Bridging Logic, Mathematics, Computer Science, and Philosophy

***Proof Theory*:**:
- intuitionistic/minimal logic
- functional interpretation
- information system
- proof mining

***Mathematics*:**
- numerical analysis
- functional analysis
- mathematical physics
- financial mathematics

CORE:
- constructive mathematics
- reverse mathematics (*„degrees of constructivity"*)

***Computer Science*:**
- functional programing
- scientific computing
- real (analog) computing
- program extraction

***Philosophy*:**
- nominalism
- intuitionism
- predicativism
- constructivism

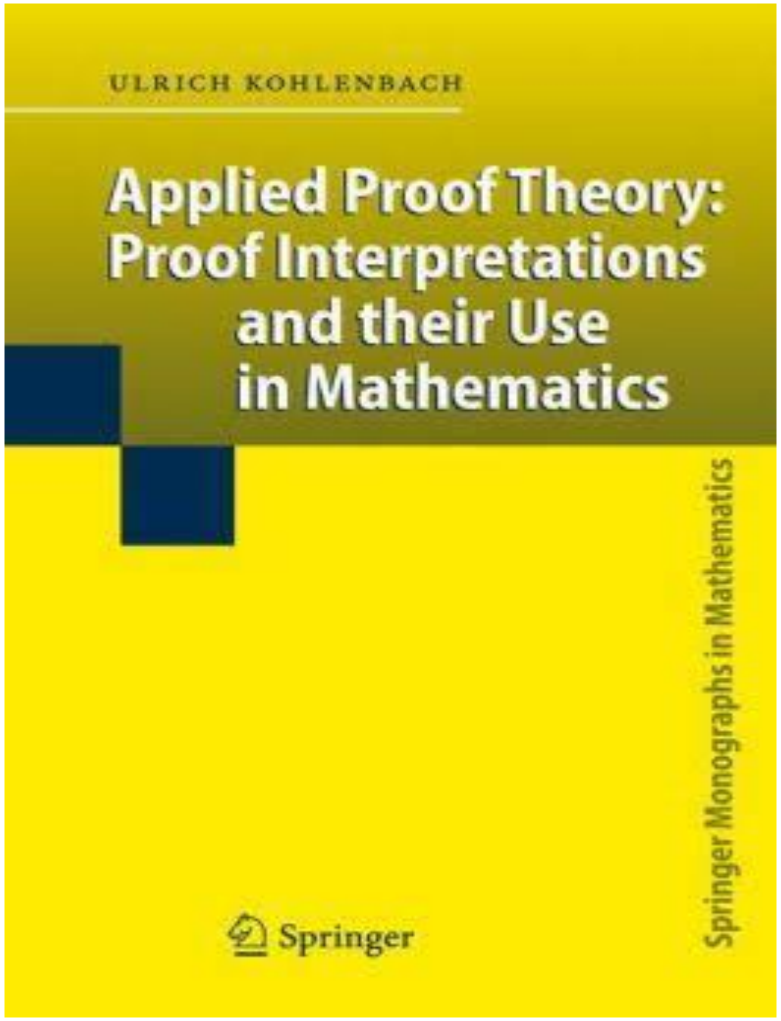
CORE, Ockham, and „The Name of the Rose“

Ockham's razor **and plea of** ***ontological reductionism*** **did not only influence** ***modern logic*** **and** ***science*****, but also** ***literature*****. In Umberto Eco's novel „*****Il nome della rosa*****“ (1980), the central character, the Franciscan friar** ***William of Baskerville*****, alludes both to the philosopher** ***William of Ockham*** **and the fictional detective** ***Sherlock Holmes*** **(compare Conan Doyle's novel „*****The Hound of Baskerville*****“).**

According to ***Ockham's razor*****,** ***William of Baskerville*** **only follows** ***logical rigorousness*** **and** ***facts*****, and** ***avoids speculative hypotheses*** **in his detective work. As philosopher, he is a** ***nominalist*** **in the famous** ***debate on universals*****. Thus, he is a** ***forerunner*** **of** ***constructive foundations in science*** **and even** ***politics*****, starting with** ***basic individuals*** **(in mathematics „natural numbers“), arguing** ***step by step without circular conclusions*** **(„predicativism“), and** ***economical*** **with** ***ontological abstractions*** **(*****principle of parsimony*****):**

„Stat rosa pristina nomine, nomina nuda tenemus.“

„The rose of old remains only in its name; we hold only naked names.“


ULRICH KOHLENBACH

Applied Proof Theory: Proof Interpretations and their Use in Mathematics

Springer Monographs in Mathematics

Springer

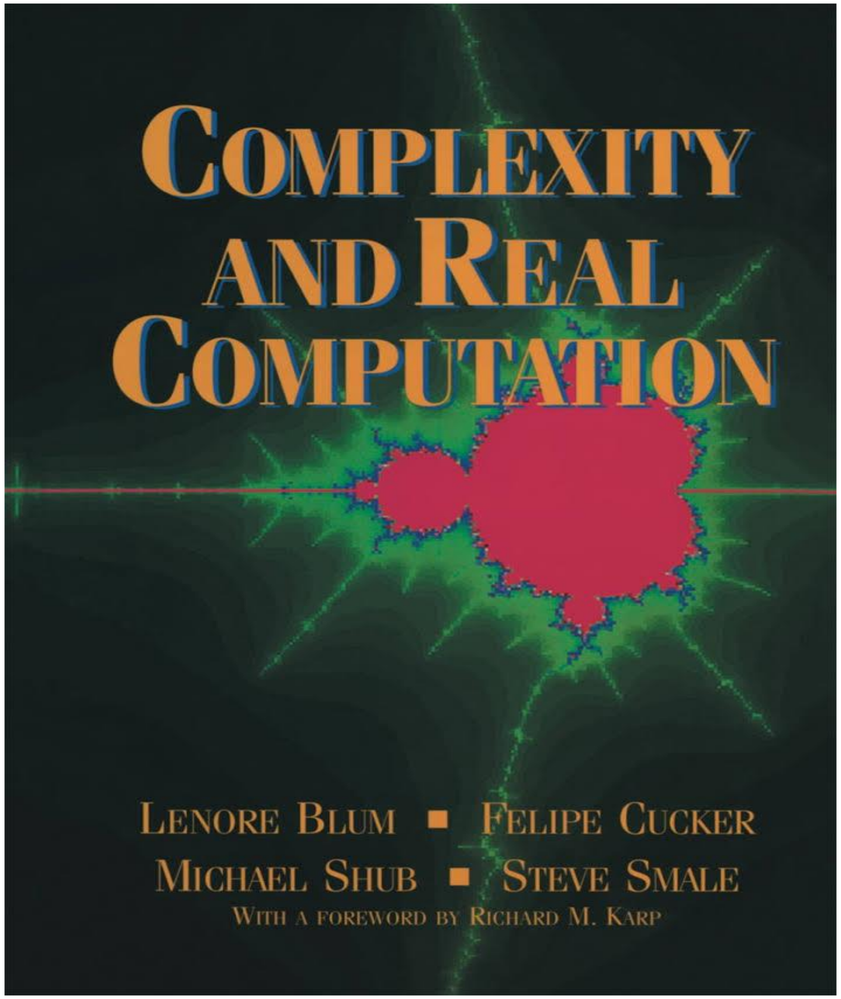
PERSPECTIVES IN LOGIC

Helmut Schwichtenberg
Stanley S. Wainer

PROOFS AND COMPUTATIONS

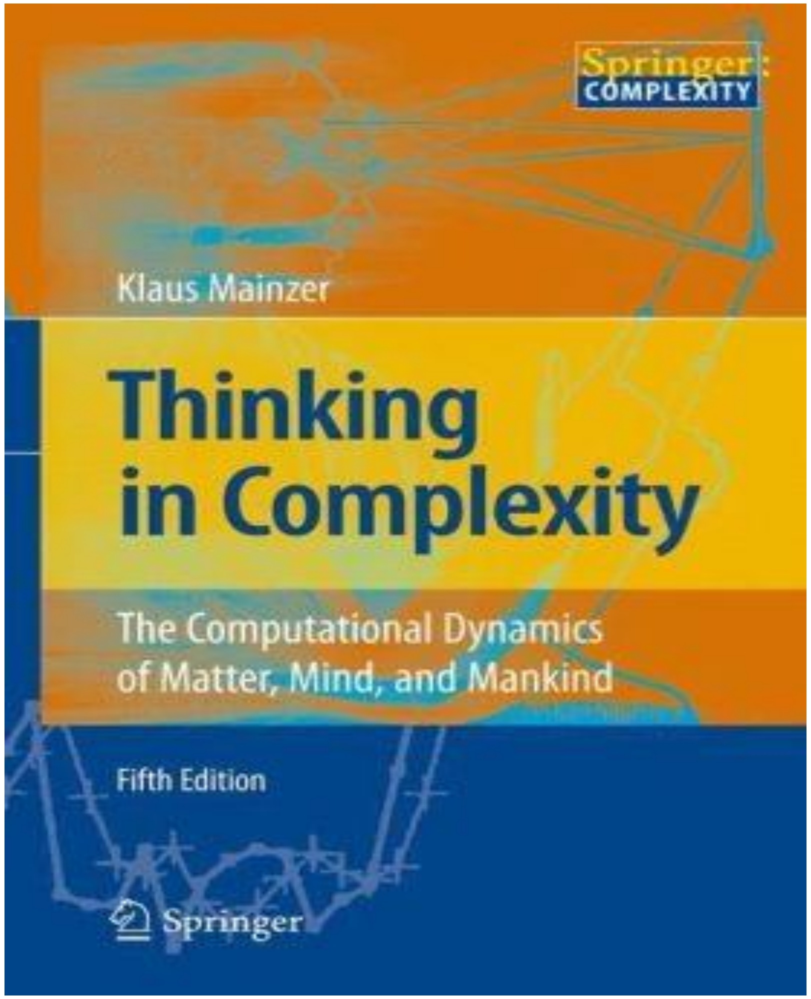
CAMBRIDGE

ASL



Complexity and Real Computation

Lenore Blum ■ Felipe Cucker
Michael Shub ■ Steve Smale
With a foreword by Richard M. Karp



Springer Complexity

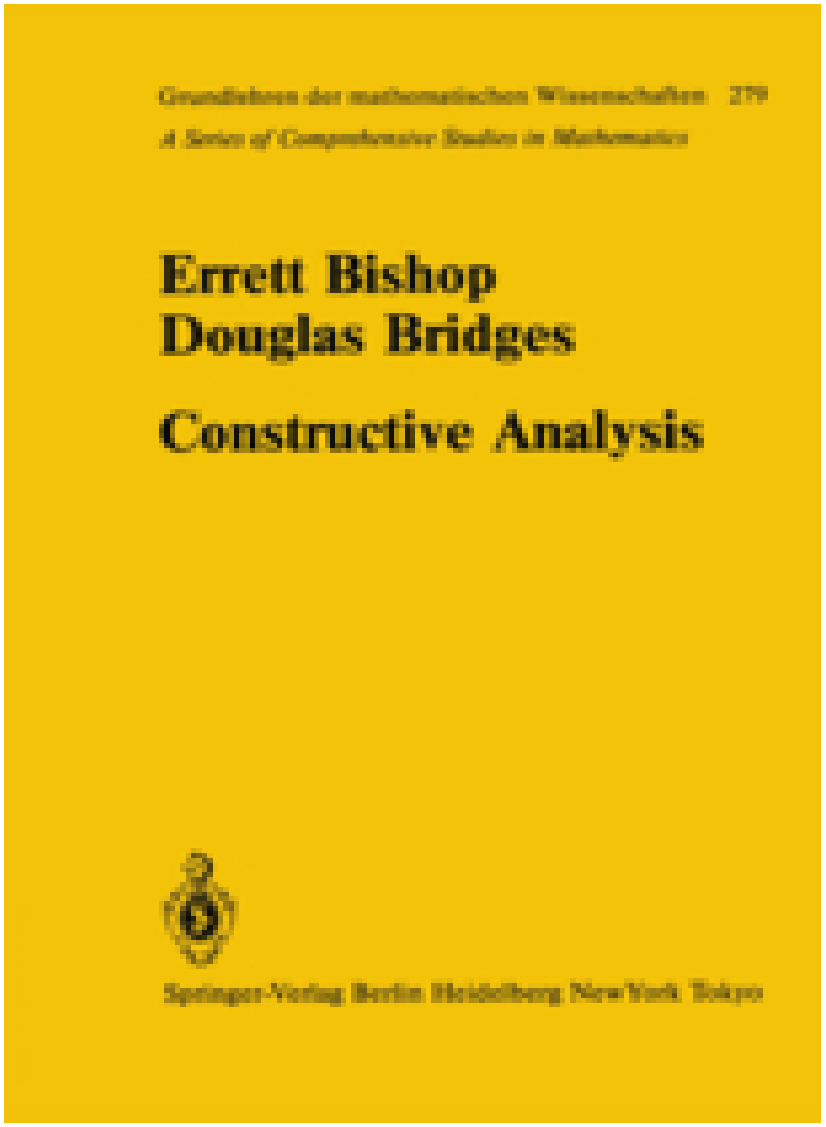
Klaus Mainzer

Thinking in Complexity

The Computational Dynamics of Matter, Mind, and Mankind

Fifth Edition

Springer

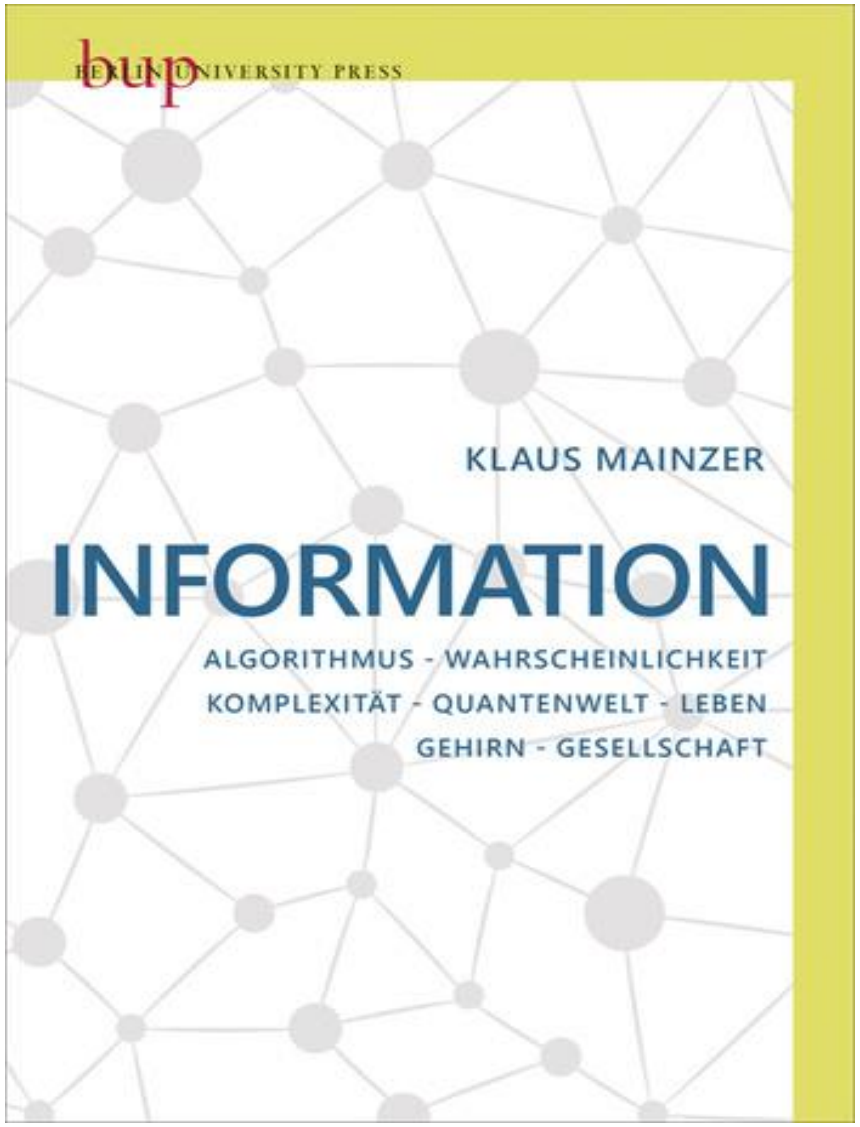


Grundlehren der mathematischen Wissenschaften 279
A Series of Comprehensive Studies in Mathematics

Errett Bishop
Douglas Bridges

Constructive Analysis

Springer-Verlag Berlin Heidelberg New York Tokyo

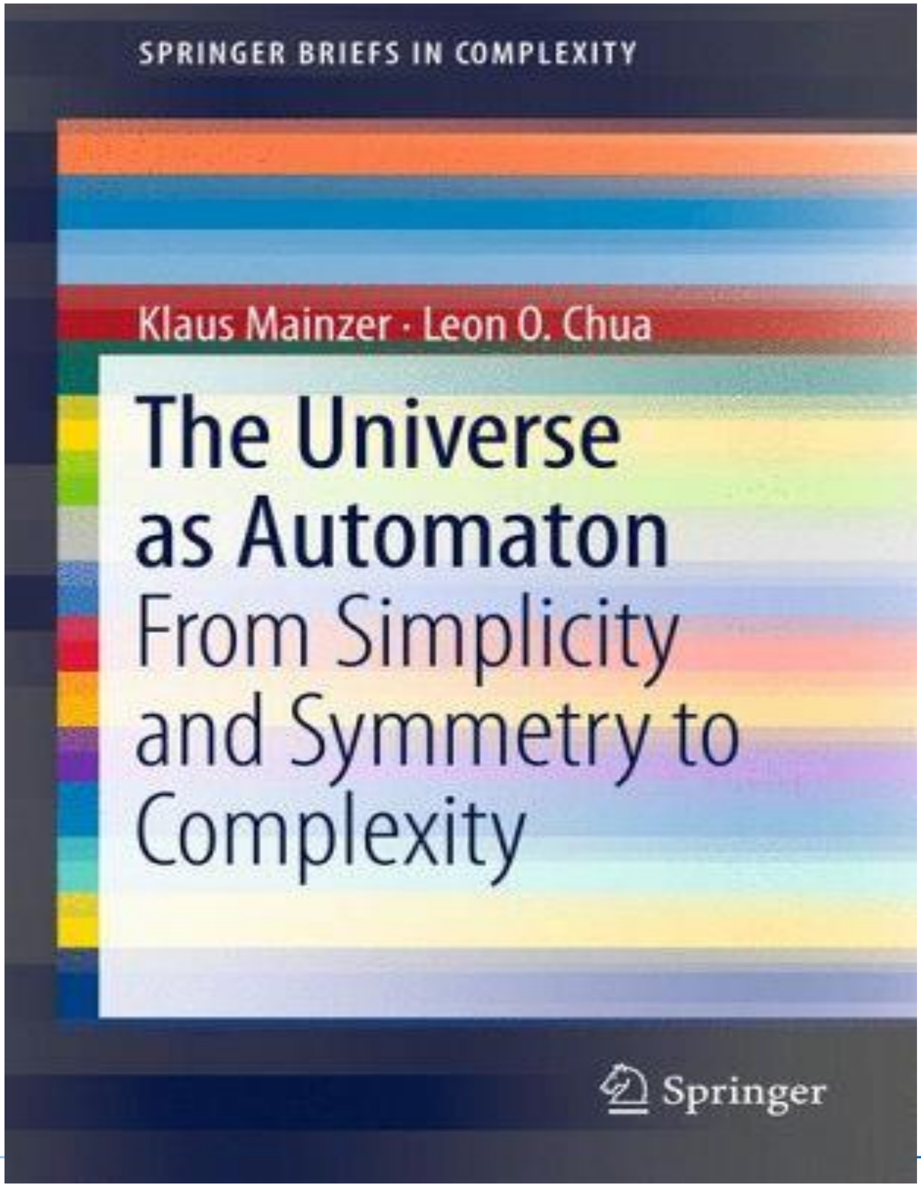


bup BERLIN UNIVERSITY PRESS

KLAUS MAINZER

INFORMATION

ALGORITHMUS - WAHRSCHEINLICHKEIT
KOMPLEXITÄT - QUANTENWELT - LEBEN
GEHIRN - GESELLSCHAFT



SPRINGER BRIEFS IN COMPLEXITY

Klaus Mainzer · Leon O. Chua

The Universe as Automaton

From Simplicity and Symmetry to Complexity

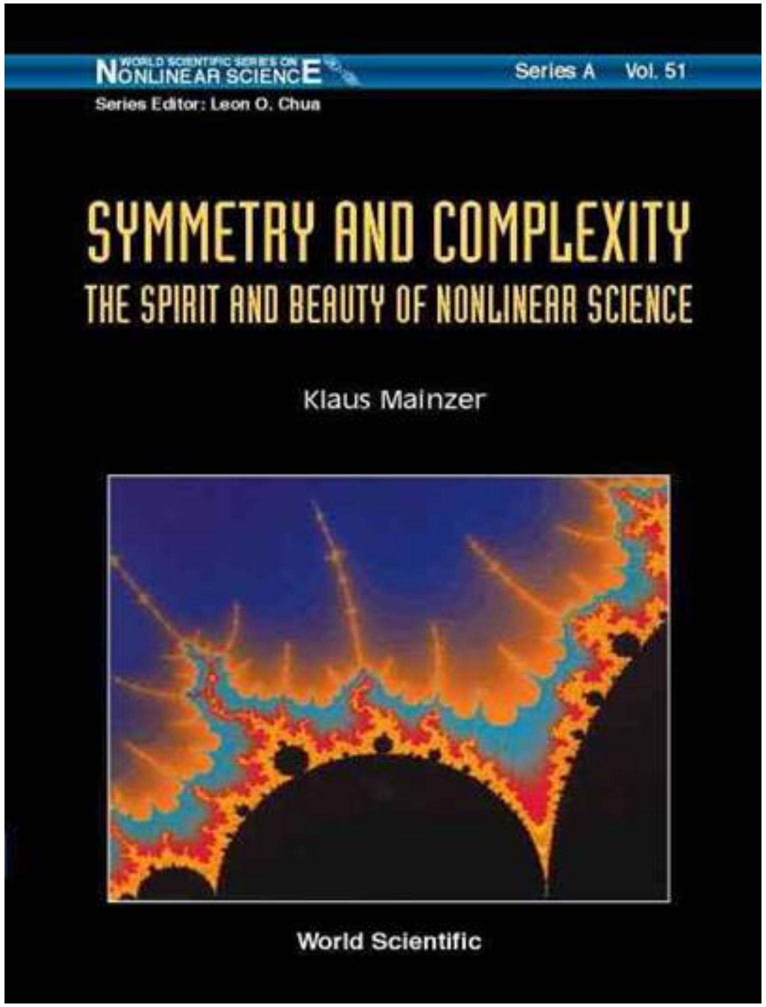
Springer

Klaus Mainzer

Die Berechnung der Welt

Von der Weltformel zu Big Data

C.H.Beck



NONLINEAR SCIENCE Series A Vol. 51

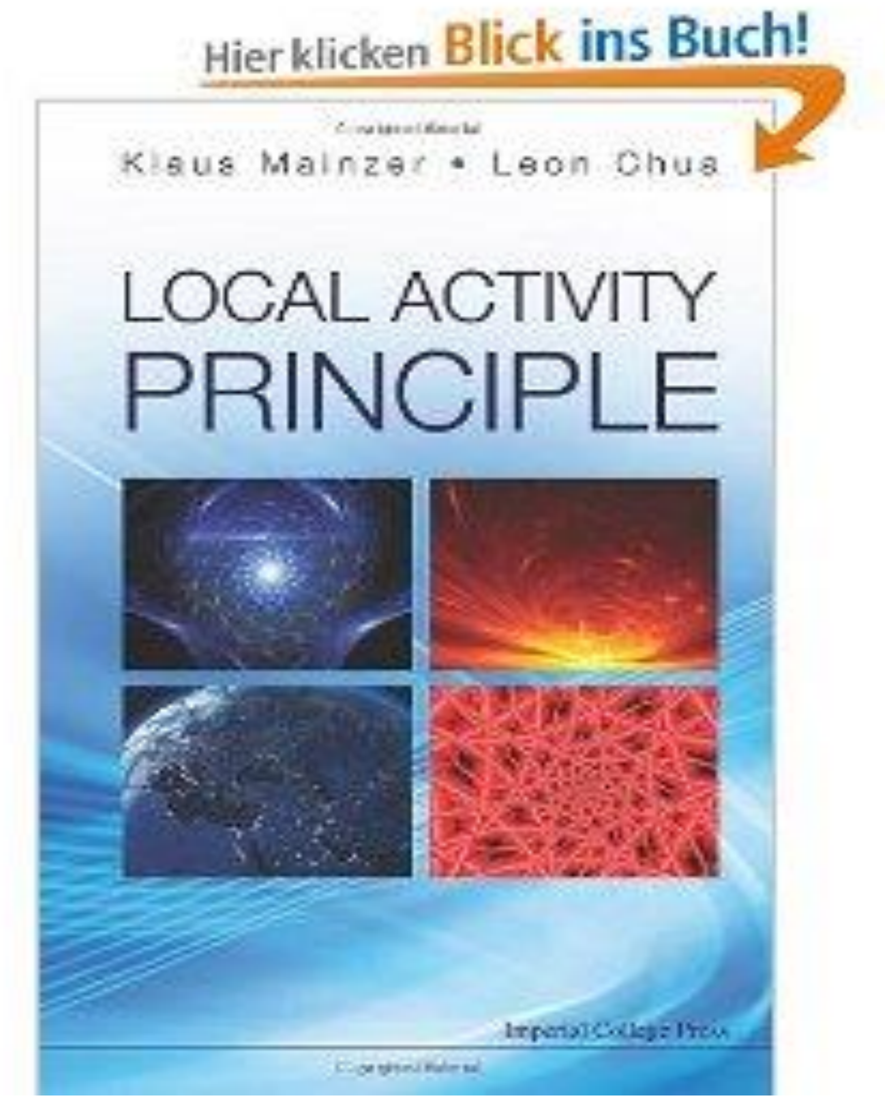
Series Editor: Leon O. Chua

SYMMETRY AND COMPLEXITY

THE SPIRIT AND BEAUTY OF NONLINEAR SCIENCE

Klaus Mainzer

World Scientific



Hier klicken Blick ins Buch!

Klaus Mainzer • Leon Chua

LOCAL ACTIVITY PRINCIPLE

Imperial College Press

Künstliche Intelligenz – Wann übernehmen die Maschinen?

Jeder kennt sie. Smartphones, die mit uns sprechen, Armbanduhren, die unsere Gesundheitsdaten aufzeichnen, Arbeitsabläufe, die sich automatisch organisieren, Autos, Flugzeuge und Drohnen, die sich selber steuern, Verkehrs- und Energiesysteme mit autonomer Logistik oder Roboter, die ferne Planeten erkunden, sind technische Beispiele einer vernetzten Welt intelligenter Systeme. Sie zeigen uns, dass unser Alltag bereits von KI-Funktionen bestimmt ist.

Auch biologische Organismen sind Beispiele von intelligenten Systemen, die in der Evolution entstanden und mehr oder weniger selbstständig Probleme effizient lösen können. Gelegentlich ist die Natur Vorbild für technische Entwicklungen. Häufig finden Informatik und Ingenieurwissenschaften jedoch Lösungen, die sogar besser und effizienter sind als in der Natur.

Seit ihrer Entstehung ist die KI-Forschung mit großen Visionen über die Zukunft der Menschheit verbunden. Löst die „künstliche Intelligenz" also den Menschen ab? Dieses Buch ist ein Plädoyer für Technikgestaltung: KI muss sich als Dienstleistung in der Gesellschaft bewähren.

ISBN 978-3-662-48452-4

9 783662 484524

▸ springer.com

Mainzer

Künstliche Intelligenz – Wann übernehmen die Maschinen?

Klaus Mainzer

Künstliche Intelligenz – Wann übernehmen die Maschinen?

Springer