Andrew Schumann

A NOVEL TENDENCY IN PHILOSOPHICAL LOGIC

In this paper we consider perspectives of application of coinductive and corecursive methods of non-well-founded mathematics to philosophical logic. So, it is shown that the problem of analysis can be solved by using greatest fixed points. Means of well-founded mathematics are enough only for an explication of the trivial analysis. We claim that the nontrivial analysis should be explicated by means of non-well-funded mathematics. Further, we build a non-well-founded propositional logic with syntax and semantics whose objects are defined by coinduction as streams. We also survey perspectives of relationship between non-well-founded logics and unconventional computing.

Department of Philosophy and Science Methodology,
Belarusian State University, Minsk, Belarus
e-mail: Andrew.Schumann@gmail.com

1. Non-well-foundedness and the main problem of analytical philosophy

A non-well-founded (non-WF) set theory belongs to axiomatic set theories that violate the rule of WF-ness and, as an example, allow sets to contain themselves: $X \in X$. In non-WF set theories, the foundation axiom of Zermelo-Fraenkel set theory is replaced by axioms implying its negation. The theory of non-WF sets has been explicitly applied in diverse fields such as logical modelling non-terminating computational processes and behavior of interactive systems in computer science (process algebra, coalgebra, logical programming based on coinduction and corecursion), linguistics and natural language semantics (situation theory), logic (analysis of semantic paradoxes).

Non-WF sets have been also implicitly used in non-standard (more precisely, *non-Archimedean*) analysis like infinitesimal and p-adic analysis [8], [29], [30], [49], [55], [58], [80]. Main advantages of non-WF sets consist in that we get an extension of standard sets such that the way of setting ma-

ISBN 978–83–7431–163–2 ISSN 0860-150X

thematical objects changes and we have a more general approach to computation without classical induction and recursion. The other extensions just as fuzzy or interval sets are defined by non-Boolean valuations of their membership relations, but the way of setting mathematical objects and the basic computing principles keep fulfilling. Non-WF sets suppose a unique way of extending conventional sets, changing the nature of setting the set hierarchy. We can define fuzzy, interval, continuous, and probabilistic systems on the base of non-WF sets. Therefore we can claim that non-WF sets are better for formalizing computing processes in natural systems than other extensions of conventional sets.

The *axiom of foundation* asserts that the membership relation $\in$ is WF (there is no descending sequence for $\in$), i.e. that any nonempty collection Y of sets has a member $y \in Y$ which is disjoint from Y. We can deny this axiom in order to postulate a set that has an infinite descending $\in$-chain, i.e. that is not WF. The particular case of such a set is one of the form $X = \{X\}$ with the circular membership relation. The set theory with the anti-foundation axiom (with denying the axiom of foundation) is considered in [4], [13]. Replacing the axiom of foundation in classical set theory with an alternative is not a new idea. For instance, in 1917 Miramanoff formulated the fundamental distinction between WF sets and hypersets.

The interest in non-WF phenomena is mainly motivated by some developments in computer sciences. Indeed, in this area, many objects and phenomena do have non-WF features: self-applicative programs, self-reference, graph circularity, looping processes, transition systems, paradoxes in natural languages, etc. Some others like strings, streams, and formal series are potentially infinite, and can only be approximated by partial and progressive knowledge. Also, it is natural to use universes containing adequate non-WF sets as frameworks to give semantics for these objects or phenomena. Moreover, it is often not relevant to use the classical principles of definition and reasoning by induction or recursion to define and reason about these objects. Therefore they assume some new metamathematical (logical) properties to be used.

So, instead of recursion, one applies *corecursion* as a type of operation that is dual to recursion. Corecursion is typically used to generate infinite data structures. The rule for primitive corecursion on codata is the dual to that for primitive recursion on data. Instead of descending on the argument, we ascend on the result. Notice that corecursion creates potentially infinite codata, whereas ordinary recursion analyzes necessarily finite data.

Induction and recursion are firmly entrenched as fundamentals for proving properties of inductively defined objects, e.g. of finite or enumerable objects. Discrete mathematics and computer science abound with such objects, and mathematical induction is certainly one of the most important tools. However, we cannot use the principle of induction for non-WF objects. Instead of this principle, the notion of *coinduction* appears as the dual to induction. Unfortunately, coinduction is still not fully established in the collective mathematical consciousness. A contributing factor is that coinduction is often presented in a relatively restricted form. Coinduction is often considered synonymous with bisimulation and is used to establish equality or other relations on infinite data objects such as streams [86] or recursive types [36]. But the sphere of applications of coinductive methods permanently became wider. For coinductive representation of real numbers see [16], [27]. Also, [56] shows possibilities of using the metric coinduction principle in the context of infinite streams as an alternative to traditional methods involving bisimulation (it is exemplified there by novel proof methods in theories of Markov chains and Markov decision processes). For applications of coinduction in logic see [61], [63], [67], [70], [77], [82], [83].

The difference between induction and coinduction may be well defined as follows. Firstly, let an operation $\Phi : \mathcal{P}(A) \to \mathcal{P}(A)$, where $\mathcal{P}(A)$ is the powerset of A, be defined as monotone iff $X \subseteq Y$ implies $\Phi(X) \subseteq \Phi(Y)$ for $X, Y \subseteq A$. Any monotone operation Φ has the least and the greatest fixed point, X_Φ and X^Φ respectively, that is, $\Phi(X_\Phi) = X_\Phi$, $\Phi(X^\Phi) = X^\Phi$, and for any other fixed point $Y \subseteq A$ of Φ (i.e. $\Phi(Y) = Y$) we have $X_\Phi \subseteq Y \subseteq X^\Phi$. The sets X_Φ and X^Φ can be defined by $X_\Phi := \bigcap\{Y\colon Y \subseteq A, \Phi(Y) \subseteq Y\}$, $X^\Phi := \bigcup\{Y\colon Y \subseteq A, Y \subseteq \Phi(Y)\}$. It is easy to see that the monotonicity of Φ implies the required properties of X_Φ and X^Φ.

On the one hand, by definition of X_Φ, we have for any set $Y \subseteq A$ that $\Phi(Y) \subseteq Y$ implies $X_\Phi \subseteq Y$. This principle is called *induction*. On the other hand, by definition of X^Φ, we have for any set $Y \subseteq A$ that $Y \subseteq \Phi(Y)$ implies $Y \subseteq X^\Phi$. This principle is called *coinduction*.

The non-WF set can be exemplified as the following stream of seasons that unfolds without end but with a cyclic pattern to their nature: $seasons = \langle spring, \langle summer, \langle fall, \langle winter, seasons\rangle\rangle\rangle\rangle$. Over the past twenty years or so, ever more intrinsically circular phenomena have come to the attention of researchers in the areas of artificial intelligence and computer science. For instance, the broad fields of artificial intelligence and computer science have given urgency to the need to formulate models of self-referential structures (see [12]).

Denying the foundation axiom in number systems implies setting the *non-Archimedean ordering structure*. Remind that Archimedes' axiom affirms the existence of an integer multiple of the smaller of two numbers which exceeds the greater: for any positive real or rational number y, there exists a positive integer n such that $y = 1/n$ or $ny = 1$. The informal sense of Archimedes' axiom is that anything can be measured by a ruler. Refusing the Archimedean axiom entails the existence of infinitely large numbers.

A connection between denying the axiom of foundation and denying the Archimedean axiom may be shown as follows. In the beginning we demonstrate an informal meaning of the axiom of foundation. Imagine that all initial objects of mathematics (e.g. numbers) are ways and the operations over those initial objects are motions on them. Then this axiom says that there exist finite ways; in this case, we use the induction principle: it is possible to achieve an aim at the shortest distance between points. The negation of the axiom of foundation causes that all ways are infinite. Then we cannot apply the *induction principle*: there are no shortest distances. Therefore one uses there the *coinduction principle*: it is possible to achieve an aim at the largest distance between points. Taking into account the existence of infinitely large numbers in non-Archimedean mathematics (e.g. in p-adic analysis or in analysis of infinitesimals), we can state that initial objects of non-Archimedean mathematics are objects obtained implicitly by denying the axiom of foundation. Non-Archimedean numbers may be represented only as infinite ways. These objects are non-WF.

The non-Archimedean version of non-WF mathematics (i.e. mathematics without the axiom of foundation) is a new branch of modern mathematics. These unconventional mathematics have found the wide application in the p-adic case of non-WF physics [51], [52], [106], [107], in the p-adic case of non-WF probability theory [53]–[54], and in the non-Archimedean case of non-WF logic [92]–[98].

In this section we will consider the possibilities of application of non-WF ideas to analytical philosophy. Ones of the first works, devoted to the similar topic, were written by Barwise [11], [12]. In those works the ideas of non-WF-ness are used for an explanation of semantic paradoxes, mainly the Liar proposition. We shall try to show that in the semantic analysis it is impossible to avoid non-WF phenomena.

The reasoning is an initial concept of any (informal or symbolic) logic. A speech which is characterized by the simultaneous realization of the following conditions is called **reasoning**: (1) it is attributive – something is affirmed in relation to something (an attributive speech is called also an *analysis*; if B is affirmed in relation to A, then A is called an *analyzed expression*, B an *analyzing expression*); (2) it is informative – it explicitly shows the recipient the content, in other words, it refers him to real or invented objects and speaks about them something nontrivial; such a speech is not wholly reducible to the statement "$A \approx A$" (it is read as follows: A is analytically equal to A) and has the form "$A \approx B$"[1]; (3) it is inferable – it explicitly or implicitly includes certain deductions which substantiate its content; such a speech can be reduced to the expression "$A \approx B$, because…"; (4) it is convincing – it can convince an interlocutor of a truth-validity of any point of view; it is possible to reduce such a speech to the expression "somebody thinks that $A \approx B$ and it is possible to agree with it."

Thus, an informative, inferable and convincing analysis is called a reasoning. It is readily seen that we have: (4) $\Rightarrow$ (3) $\Rightarrow$ (2) $\Rightarrow$ (1). Indeed, every convincing speech is inferable, every inferable speech is informative, and every informative speech is attributive, but no vice versa, because there exists an attributive speech which is not informative, an informative speech, which is not inferable, and an inferable speech, which is not convincing.

Logic is a science which studies various modes of modelling reasonings.

It is possible to distinguish three levels in a reasoning: (1) syntactical, i.e. relations between signs used in a reasoning; (2) semantical, i.e. relations between denotations, truth-values, senses, etc. of signs; (3) pragmatical, i.e. relations between agents of a language competence, namely those who depending on a concrete situation are capable to ascribe semantical objects to well-formed combinations of signs. The class of all syntactical relations of an arbitrary language L (for example, English) is called syntax of L. The class of all semantical relations of a language L is called semantics of L. The class of all pragmatical relations of a language L is called pragmatics of L. A semiotic system of L is formed as a triple of syntax, semantics, and pragmatics of L.

Semantics of any language consists of rules of the following classes: (1) rules according to which real or invented object are ascribed to well-formed combinations of signs; (2) rules according to which state of affairs are ascribed to well-formed combinations of signs; the latter are called relations of semantical superposition.

A *semantical superposition* is a correspondence relation that maps states of affairs onto finite combinations of words. Such a combination is called a proposition. For example, the state of affairs 'Socrates goes for a walk' is assigned to the combination of five words "Socrates goes for a walk". A semantical superposition is formed due to special functions. For example, the proposition "Socrates is a man" is formed due to the function "…is…".

For the successful modelling of reasonings it is necessary to have an algorithm of constructing a semantical superposition. Notice that a convincing

[1] We have $A = B \supset A \approx B$, but no vice versa, where $=$ is a sign for the equality.

attributive speech can satisfy the principle of induction: (1) the analysis "$A \approx A$" is convincingly inferable (i.e. this analysis is obvious without deductions); (2) if the analysis "$A \approx B$" is convincingly inferable by means of any argument C, then it is convincingly inferable by means of any other number of arguments among which C occurs; (3) every analysis is convincingly inferable if and only if it satisfies the two previous conditions. The inductive treatment of a convincing substantiation shows that the analysis in logic should contain the least type of semantical connection. It means that in the logical analysis, the natural logical function should assume a WF semantical superposition and it should satisfy the principle of induction. Thus, if A is a semantical superposition, then B is a semantical superposition if and only if $B \subseteq A$.

Conventional logical functions (negation, conjunction, disjunction, implication, etc.) are a variety of WF semantical superposition. For example, the implication is understood as the least type of semantical connection in a conditional proposition. Recall that the implication satisfies the following condition: $A \supset B$ is false if and only if A is true and B is false, and $A \supset B$ is true in all other cases.

Rules of WF semantical superpositions for appropriate propositions allow us to set semantical superpositions by finite algorithms. Using the example of implication we can show that indeed, the implication is a WF semantical superposition (i.e. it assumes the least fixed point, the least type of semantical connection). Firstly, from the false antecedent everything follows: $\{\text{false}\} \supset A$, i.e. any A, expressing any state of affairs. Secondly, we infer the true from any proposition: $A \supset \{\text{true}\}$. Consequently, the expression "If $2 \times 2 = 4$, then the moon is spherical" is a true implication, though any causal relationship between the antecedent and the consequent is not seen. The similar absurdities implied from the condition of WF semantical superposition for a conditional proposition are called in logic the paradoxes of implication.

Traditionally, the relations which assume a WF semantical superposition are regarded as logical relations. Let $\mathfrak{R}_L$ be the set of all logical relations and $\mathfrak{R}_S$ be the set of all semantical relations, i.e. the relations supposing any semantical superposition (WF and as well as non-WF). It is known that logical relations form the least nonempty subset of the set of semantical relations. This property is formulated as follows: $\mathfrak{R}_L \subset \mathfrak{R}_S$, and if $\mathfrak{R}_X \subset \mathfrak{R}_S$ and $\mathfrak{R}_X \subset \mathfrak{R}_L$, then $\mathfrak{R}_X = \mathfrak{R}_L$. In other words, for every $R_S \in \mathfrak{R}_S$ it is possible to find an appropriate $R_L \in \mathfrak{R}_L$ such that R_L is the least semantical connection in the semantical relation R_S.

Such an understanding of logical relations was proposed in the logical positivism (its representatives are Carnap, Frege, Hilbert, Quine, Russell, Tarski, and many others). We will call this understanding the **hypothesis of WF-ness of logical relations**.

The analytical philosophy (as a special direction of logical-and-philosophical researches) studies conditions of constructing attributive speeches, i.e. the expressions of the form "$A \approx B$". Logic studies not any, but only convincing attributive speeches. Therefore logic has the more restricted subject than analytical philosophy. However, algorithms of building the analysis "$A \approx B$" are exclusively studied within the framework of logic.

The logical positivism is the best known tradition of analytical philosophy. Within the framework of this tradition, the problem of analysis is solved only by means of logical methods, therefore the semantic analysis here is completely reduced to WF semantical superpositions. The main problem of the logical positivism is formulated as follows: what the least subset of the given set of terms, sufficient for analyzing all the remaining terms of this set? Thus, in the logical positivism, the correct analysis is reduced to the identity, because, according to logical positivists, every analysis is a WF semantical superposition.

The research of the problem of analysis should be carried out in frameworks of a system of postulates. For example, in the system of nonlogical postulates, in which the term 'triangle' is defined, we can conclude: "X is an equilateral triangle if and only if X is an equiangular triangle". Therefore "$A \approx B$" can be treated as "$A = B$" only in frameworks of a system L in which terms A and B with the corresponding semantical relations are well defined.

Let X be an analyzed expression, Y be an analyzing expression. If the analysis is correct, then, according to logical positivists, X is identical to Y; but in this case an opinion expressed by the expression $X = Y$ is identical to an opinion expressed by the expression $X = X$. In other words, if the analysis, containing a certain identity, is correct, then it is trivial. This refers to the paradox of analysis: the expression $X = Y$ cannot simultaneously be true and nontrivial [60], [75].

Let us consider two equations: $4 = IV$ and $4 = \sqrt[3]{64}$. We can always state that the equation $4 = IV$ is trivial, whereas the equation $4 = \sqrt[3]{64}$ is informative, as various numbers occur both on the right-side and on the left-side. To avoid the paradox of analysis, logical positivists distinguish two modes of identity: ontological and semantical. The expression "X and Y are identical concepts (or properties)" reflects an *ontological mode of identity.* A *semantical mode of identity* is formulated as follows: "X and Y are synonymous expressions". So, the equation $4 = \sqrt[3]{64}$ contains an ontological mode of identity, and the equation $4 = IV$ a semantical mode of identity.

Nevertheless, the paradox of analysis again arises in a semantic statement if we accept the following principle:

(i) *Let S_1 be a statement containing expression X and S_2 be a statement obtaining from S_1 if X is replaced by a synonymous expression Y; then S_1 is synonymous with S_2.*

This principle directly entails the paradox if we consider S_1 as a statement expressing the analysis, X as an analyzable expression, and Y as an analyzing expression. For example, while the word 'father' is synonymous with the expression 'male parent', the phrase 'the father is a male parent' means (i.e. is synonymous with) that 'father is father'. In the formal notation:

(ii) $X \in S_1$ *and* $Y \in S_1$, *therefore* $(X = Y) \in S_1$ *and* $(X = X) \in S_2$.

Let us consider another example. Let the analysis S_1 be represented by the expression "the attribute of being a brother = the attribute of being a male sibling". It allows us to paradoxically infer:

> *The statement "the attribute of being a brother = the attribute of being a male sibling" is identical to the statement "the attribute of being a brother = the attribute of being a brother".*

If we regard S_2 as statement "somebody knows that brother = brother", then we infer that "somebody knows that brother = male sibling". It is obvious that this deduction cannot be logically correct. So, from my knowledge that $4 = 4$ does not follow that I know all the true equations of arithmetics concerning the number 4. This version of the paradox of analysis is called the epistemic paradox of analysis. It is formulated as follows:

(iii) *If $p = q$, $(p = p) = (p = q)$ and everybody knows that $p = p$, then everybody knows that $p = q$.*

In Church's opinion, we can deny paradox (ii), in particular paradox (iii), if we accept Frege's difference between sense (Sinn, connotation) and meaning (Bedeutung, denotation). The names 'father' and 'male parent', designating the same concept, have, nevertheless, a different sense, e.g. in the equation $2 + 2 = 4$ the expressions both on the left-side and on the right-side designate the same number (they have a common denotation), but they have a different sense. Clearly, the substitution rule formulated in (i) holds if the synonymy relation is understood as identification by sense.

Another way of denying paradoxes (ii) and (iii) was proposed by R. Carnap [23], [24]. He noticed that statements "Scott is identical to Scott" and "Scott is identical to the author of Waverly" are not identical, because the first is a tautology, and the second is a factual statement (assuming that the name 'Scott' is not an abbreviation of the expression 'the author of Waverly').

The most adequate way of denying the paradox of analysis consists in the difference between $X = Y$ (i.e. a nontrivial informative analysis), as a non-WF synonymous connection for X (*salva necessitate* is its traditional name), and $X = X$ (i.e. a trivial uninformative analysis), as a WF synonymous connection for X (*salva veritate* is its traditional name). Let i designate the members identical among themselves. The analysis $X = Y_i$ which takes place for some i is called a *non-WF synonymous connection for* X. This means that the identity is closed by using the greatest fixed point (coinduction). As a result, the nontrivial analysis "Scott is identical to the author of Waverly" should be considered as non-WF synonymous connection. The analysis $X = Y_i$ which takes place for all i is called a *WF synonymous connection for* X. This means that the identity is closed by using the least fixed point (induction). In this case the trivial analysis of the form "Scott is identical to Scott" should be described. It is obvious that the problem of analysis in relation to WF synonymous connection for X is solved by using conventional computing (e.g. using determined algorithms). We can show that the problem of analysis in relation to non-WF synonymous connection for X is solved by using unconventional computing (in particular, probabilistic algorithms).

Let C be an analyzable expression containing a certain theoretical term and the formula $Q_i \supset R_i$ be an analyzing expression of the form of conditional statement described in the language of i observations, namely in the language in which the theoretical term included in C is regarded. For instance, take as C the expression "At the moment t the electric light has a power equal l", and take as $Q_i \supset R_i$ the expression "At the observation i if at the moment t the given electric cable is connected to the ammeter, then the arrow of this ammeter moves at this moment to l from the initial position".

Let us assume that we have a WF synonymous connection for C in the analysis $C = Q_i \supset R_i$. This means that $Q_i \supset R_i$ logically follows from C for all i. Suppose further that $Q_i \supset R_i$ follows from C for some i, i.e. the expression $Q_i \wedge \neg R_i$ entails $\neg C$ for some i and $Q_j \wedge \neg R_j$ entails C for some $j \neq i$. But it cannot be correct for a WF synonymous connection for C. Suppose now that in the analysis $C = Q_i \supset R_i$, we should use a non-WF synonymous connection for C. Indeed, it is impossible to show that $C = Q_i \supset R_i$ for every observation i, because we should carry out infinitely many observations. In the given situation it is better to compare C with a degree of its falsification or confirmation. In other words, the analysis $C = Q_i \supset R_i$ should be considered not in all possible worlds (not for all i), but in some ones (for some i). Thus, the problem of the analysis $C = Q_i \supset R_i$ can be solved here by using probabilistic algorithms.

Let us also show that the problem of analysis is not solved by means of determined algorithms or other WF methods of logical analysis in the natural language. Let C be an analyzing expression, $Q_i \supset R_i$ be an analyzed expression, and the latter is pronounced by an agent of the language competence i with the following sense: "If the expression C is correct, then C is convincing for me." Assume that in the analysis $C = Q_i \supset R_i$, we have a WF synonymous connection for C. It means that $Q_i \supset R_i$ should be inferred from C for all i. Assume also that if the expression "C is correct and C is convincing for me" for some i, then we have $\neg C$. But it is not obvious for a WF synonymous connection for C. Thus, we have the condition of non-WF connection for C in the analysis $C = Q_i \supset R_i$. Interviewing all agents i to make clear, whether the expression C convinces them, is not a realizable task. As we see, the analysis $C = Q_i \supset R_i$ is necessary for considering not in all probable worlds, but at concrete i. Therefore the problem of the analysis $C = Q_i \supset R_i$ is always contextually solved in the natural language, in relation to the concrete native speaker.

So, paradoxes of analysis are bright witnesses that using WF methods it is not possible to explicate all the semantical relations originating in the natural language. Within the framework of logical positivism, the reduction of all semantical connections to WF was proposed, and the initial WF semantical superpositions were called logical functions (negation, conjunction, disjunction, implication). However, as we were convinced, there are the semantical relations which are not expressed by means of WF methods, namely the nontrivial analysis "$A \approx B$." Therefore we can assume that there are the natural logical functions producing the greatest type of semantical connection (non-WF logical relations). It means that each this logical function assumes a non-WF semantical superposition. In this case, if A is a semantical superposition and $A \subseteq B$, then B also is a semantical superposition.

Hence, we can put forward the **hypothesis of non-WF-ness of logical relations**. So, the task of construction of logical calculi not on the basis of WF semantical superpositions, but on the basis of non-WF ones seems to be very promising nowadays.

2. Non-well-founded logical predications

One of the most useful non-WF mathematical object is a stream – a recursive data-type of the form $s = \langle a, s' \rangle$, where s' is another stream. The notion of *stream calculus* was introduced by Escardó and Pavlović [72] as a means to do symbolic computation using the coinduction principle instead of the induction one. Let A be any set. We define the set A^{ω} of all streams over A as $A^{\omega} = \{\sigma\colon \{0, 1, 2, \ldots\} \to A\}$. For a stream σ, we call $\sigma(0)$ the initial value of σ. We define the *derivative* $\sigma(0)$ of a stream σ, for all $n \geq 0$, by $\sigma'(n) = \sigma(n + 1)$. For any $n \geq 0$, $\sigma(n)$ is called the n-th element of σ. It can also be expressed in terms of higher-order stream derivatives, defined, for all $k \geq 0$, by $\sigma^{(0)} = \sigma$; $\sigma^{(k+1)} = (\sigma^{(k)})'$. In this case the n-th element of a stream σ is given by $\sigma(n) = \sigma^{(n)}(0)$. Also, the stream is understood as an infinite sequence of derivatives. It will be denoted by an infinite sequence of values or by an infinite tuple: $\sigma = \sigma(0) :: \sigma(1) :: \sigma(2) :: \ldots :: \sigma(n-1) :: \sigma^{(n)}$, $\sigma = \langle \sigma(0), \sigma(1), \sigma(2), \ldots \rangle$.

A *bisimulation* on A^{ω} is a relation $R \subseteq A^{\omega} \times A^{\omega}$ such that, for all σ and τ in A^{ω}, if $\langle \sigma, \tau \rangle \in R$ then (i) $\sigma(0) = \tau(0)$ (*initial value*) and (ii) $\langle \sigma', \tau' \rangle \in R$ (*differential equation*).

If there exists a bisimulation relation R with $\langle \sigma, \tau \rangle \in R$ then we write $\sigma \sim \tau$ and say that σ and τ are *bisimilar*. In other words, the *bisimilarity* relation $\sim$ is the union of all bisimulations: $\sim\, := \bigcup\{R \subseteq A^{\omega} \times A^{\omega} \colon R \text{ is a bisimulation relation}\}$. Therewith, this relation $\sim$ is the greatest bisimulation. In addition, the bisimilarity relation is an equivalence relation.

Theorem 1 (Coinduction)

For all $\sigma, \tau \in A^{\omega}$, if there exists a bisimulation relation $R \subseteq A^{\omega} \times A^{\omega}$ with $\langle \sigma, \tau \rangle \in R$, then $\sigma = \tau$. In other words, $\sigma \sim \tau \Rightarrow \sigma = \tau$. □

This proof principle is called *coinduction*. It is a systematic way of proving the statement using bisimularity: instead of proving only the single identity $\sigma = \tau$, one computes the greatest bisimulation relation R that contains the pair $\langle \sigma, \tau \rangle$. By coinduction, it follows that $\sigma = \tau$ for all pairs $\langle \sigma, \tau \rangle \in R$.

Now consider a *non-WF propositional logic* $\mathfrak{L}^{\omega}$, whose syntax and semantics are non-WF, i.e. they are defined by coinduction and their objects are streams. The syntax of $\mathfrak{L}^{\omega}$ is as follows:
Variables: $x ::= p \mid q \mid r \;\ldots$
Constants: $c ::= \top \mid \bot$
Formulas: $\varphi, \psi ::= x \mid c \mid \neg\psi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \supset \psi$

These definitions are coinductive. For instance, a variable x is of the form of a stream $x = x(0) :: x(1) :: x(2) :: \ldots :: x(n-1) :: x^{(n)}$, where $x(i) \in \{p, q, r, \ldots\}$ for each $i \in \omega$; a constant c is of the form of a stream $c = c(0) :: c(1) :: c(2) :: \ldots :: c(n-1) :: c^{(n)}$, where $c(i) \in \{\top, \bot\}$ for each $i \in \omega$; a formula $\varphi \wedge \psi$ has the differential equation $(\varphi \wedge \psi)' = \varphi' \wedge \psi'$ and its initial value is $(\varphi \wedge \psi)(0) = \varphi(0) \wedge \psi(0)$, etc.

Consider now semantics of $\mathfrak{L}^{\omega}$.

Truth-valuation of variables: $v(x) ::= v(p) \mid v(q) \mid v(r) \ \ldots$

Truth-valuation of formulas: $v(\varphi), v(\psi) ::= v(x) \mid v(\neg\psi) \mid v(\varphi \vee \psi) \mid v(\varphi \wedge \psi) \mid v(\varphi \supset \psi)$

These definitions are coinductive too. For example, truth-valuations of formulas are defined as follows:

- the differential equation of $v(\neg\psi)$ is $(v(\neg\psi))' = \top' - v(\psi)'$ and its initial value is $(v(\neg\psi))(0) = \top(0) - v(\psi)(0)$,
- the differential equation of $v(\varphi \vee \psi)$ is $(v(\varphi \vee \psi))' = \sup(v(\varphi)', v(\psi)')$ and its initial value is $(v(\varphi \vee \psi))(0) = \sup(v(\varphi)(0), v(\psi)(0))$,
- the differential equation of $v(\varphi \wedge \psi)$ is $(v(\varphi \wedge \psi))' = \inf(v(\varphi)', v(\psi)')$ and its initial value is $(v(\varphi \wedge \psi))(0) = \inf(v(\varphi)(0), v(\psi)(0))$,
- the differential equation of $v(\varphi \supset \psi)$ is $(v(\varphi \supset \psi))' = \top' - \sup(v(\varphi)', v(\psi)') + v(\psi)'$ and its initial value is $(v(\varphi \supset \psi))(0) = \top(0) - \sup(v(\varphi)(0), v(\psi)(0)) + v(\psi)(0)$.

We used very simple non-WF syntax and non-WF semantics. For non-WF syntax and non-WF semantics in functional programming see [61]. For coinductive definitions and proofs in big-step semantics, using both finite and infinite evaluations, see [63] and the works of Xavier Leroy[2]. The natural semantics of logic $\mathfrak{L}^{\omega}$ are given in [92]–[98] and these semantics assume a *non-Archimedean ordering structure.*

In paper [71], it is shown that Church's higher-order logic (HOL) is perfectly adequate for formalizing both inductive and coinductive definitions, e.g. a theory of recursive and corecursive definitions can be mechanized using Isabelle. Recall that least fixed points express inductive data types such as strict lists; greatest fixed points express coinductive data types, such as lazy lists. WF recursion expresses recursive functions over inductive data types; corecursion expresses functions that yield elements of coinductive data types.

Notice that coinduction has been well established for reasoning in concurrency theory [62]. Abramsky's lazy lambda calculus [1] has made coin-

[2] pauillac.inria.fr/∼xleroy

duction equally important in the theory of functional programming. Later, Milner and Tofte motivated coinduction through a simple proof about types in a functional language [63].

Logical positivists suggested that proof should be WF, e.g. it should be inductively constructed. However, it is shown in [17], [19], [20], and [101] that *cyclic* (non-WF) *proof* provides a promising alternative to traditional inductive proof, modelled on streams (Fermat's infinite descent). WF proof systems are limited by the following principle: if in some case of a proof some inductive definition is unfolded infinitely often, then that case may be disregarded. Essentially, this principle is sound because each inductive definition has a least-fixed point interpretation which can be constructed as the union of a chain of approximations, indexed by ordinals; unfolding a definition infinitely often can thus be seen as inducing an infinite descending chain of these ordinals, which contradicts their WF-ness. In cyclic proof systems, the capacity for unfolding a definition infinitely often is built in to the system by allowing proofs to be non-WF, i.e. to contain infinite paths.

A simple example of *non-WF proof system* is given by Alex Simpson. This system is sound and complete on Borel sets and it captures inclusion between Borel sets of topological spaces. Recall that the Borel sets $\mathcal{B}(X)$ over a topological space X is the smallest σ-algebra containing the open sets $\mathcal{O}(X)$ (i.e. there is the closure of $\mathcal{O}(X)$ under complements and countable unions).

Let us consider a partially ordered set with: (1) finite infima (including top element $\top$), (2) countable suprema (including least element $\bot$), (3) and satisfying the distributive law for countable suprema: $x \wedge \bigvee_i y_i = \bigvee_i x \wedge y_i$. It is called a σ-frame. Further, let F be a σ-frame, and $B \subseteq F$ a base (i.e. every element of F arises as a countable supremum of elements of B). We define formulas for Borels by taking elements $b \in B$ as propositional constants and closing under negation, and countable conjunctions and disjunctions:

$$\psi ::= b \mid \neg\psi \mid \bigvee_i \psi_i \mid \bigwedge_i \psi_i$$

The proof system contains usual sequent proof rules on left and right for each connective, e.g.

$$\frac{\Gamma, \psi_i \vdash \Delta}{\Gamma \vdash \Delta} \bigwedge_i \psi_i \in \Gamma,$$

$$\frac{\{\Gamma \vdash \psi_i, \Delta\}_i}{\Gamma \vdash \Delta} \bigwedge_i \psi_i \in \Delta.$$

This system also includes atomic cuts:

$$\frac{\Gamma, b \vdash \Delta \quad \Gamma \vdash b, \Delta}{\Gamma \vdash \Delta}.$$

A basic entailment, written $C \Rightarrow D$, is given by a finite $C \subseteq B$ and countable $D \subseteq B$ such that: $\bigwedge C \leq \bigvee D$ in F. An infinite branch $(\Gamma_i \vdash \Delta_i)$ in a rule tree is justified if there exist $C \subseteq \bigcup_i \Gamma_i$ and $D \subseteq \bigcup_i \Delta_i$ such that $C \Rightarrow D$. A rule tree is a proof if every infinite branch is justified.

Evidently, non-WF logic with non-WF proof system can have unexpected metalogical properties that are not studied yet. The conventional computing is set in the frameworks of WF logics and WF proof systems. We can assume that setting computation using non-WF logics and non-WF proof systems can be provided by new horizons.

3. Unconventional computing

3.1. New computing media

It is a common misconception that computers have been changing in a qualitative sense for the last 50 years. All of the modern computers' basic procedures and architectures are still based on the principles formulated back in the 1930s and 1940s by Alan Turing and John von Neumann. But is it indispensable? Or at least effective? The answers to these questions is the subject of what is called unconventional computing – a relatively young yet many-sided branch of computer science.

Several conceptual question arise on the matter of practical realization of those basic principles. For example, a Turing machine is by definition unrealizable, because it involves an endless tape, whereas every computer's memory is finite. Unconventional models take a different approach to the axiomatization of computing: they try to make physical realizability an indigenous part of the axioms.

Despite having been a topic of discussion since at least as early as the 1970s, the terminology in this field is not quite established yet. Unconventional computing models can be divided into two groups:

1. Mechanical models: billiard ball computers, Domino computers etc., see [14], [37].
2. Biologically-inspired models: neural networks, genetic algorithms, cellular automata etc., see [5], [21], [46], [81].

Some of the models are purely theoretical, like the billiard ball models, and often they are also more important theoretically than being realized in practice. Others, like quantum computers, are variously close to practical implementations.

In the large sense of the word, computation (which is a much broader concept than calculation) is any mapping of input information into an output state. Hence every real-life system does some sort of computation; at the least, it computes its own future state. In other words, all that is needed to make a computer is an excitable medium and a way to channel the propagation of activity in it, i.e. the aim is to design (and, hopefully, construct) systems that do a predictable (or intentionally unpredictable, or probabilistic) computation.

There is also a popular concept of natural computing; most of what is called natural computing can be placed among mechanical or biological models, though there are some approaches that lie somewhere between, like chemical computing.

Here we come to one of the fundamental statements in computation theory, known as the Church-Turing thesis. It claims that any computation that is physically possible (notice that physically possible computations also include human thinking) can also be performed by a *conventional* computer. It is not a provable theorem, but rather an assertion connecting human intuition and the strict concept of algorithm. If this thesis is right, then the unconventional models can only find some applications in making ordinary algorithms more effective. But if it is not, then things get much more tricky: there appear to be problems that are principally unsolvable via conventional computing. Continuing in the same way, we can claim that in the last case we need to use logical systems of the new type as metalanguage, e.g. more expressible non-WF logics.

One of the objectionable points of unconventional computing models is the fact that they rely on many phenomena that are far from being completely described and explained by science, such as quantum physics or brain functioning. So these models inevitably simplify the actual state of affairs, and at some point go beyond the line where they start to bear more and more resemblance to conventional computing.

Human mind and ongoing theoretical research goes far beyond technical possibilities in what concerns unconventional computing – but that's not a reason for scepticism; let us recall that two-valued algebra was invented by George Boole more than a century before it found its application in modern computers. Adherents of mechanical models point at what is sometimes called the semantic gap the difference between visible logic and actual physical principles of computation. Unconventional computing tries to shorten this gap by introducing straightforward analogies between computational systems and the tasks solved by them.

3.1.1. The billiard ball model

It is one of the most ingenious models of unconventional computing [37]. Balls travelling on two-dimensional grid with a constant speed are considered. The grid is somewhat unusual: it is obtained from the ordinary uniform grid by a 45-degree turn; thus every grid point has 4 neighbors with equal distances to each of them, and this spacing is taken as the unit of distance. Time is discrete ($t = 0, 1, 2, \ldots$). Balls have the radius $\frac{1}{\sqrt{2}}$ and thus collide elastically (due to this radius two neighboring grid points cannot be simultaneously occupied by balls, and if two balls occupy two points exactly $\sqrt{2}$ apart, then they touch at exactly one point, because $\frac{1}{2}\sqrt{2} = \frac{1}{\sqrt{2}}$). Certain borders are set on the grid in order to realize conventional logic elements.

The billiard ball model is clearly reversible. But what can it be used for? It is an important universal logic model. It can be shown [37] that every logical circuit can be simulated by means of only one element called the Fredkin gate, which has 3 inputs c, p and q and 3 outputs $x = c$, $y = cp + \bar{c}q$, $z = \bar{c}p + cq$. The Fredkin gate can be built on the basis of a yet simpler element called the S-gate, which has 2 inputs c and x and 3 outputs c, cx and $\bar{c}x$ (the Fredkin gate is composed of two S-gates and two inverse S-gates). So, if we simulate the S-gate, thus we simulate all possible logical circuits. And indeed, the S-gate can be simulated via billiard balls and (in a rather intricate but harmonious manner) by cellular automata [64].

Note that the billiard ball model is a particular case of cellular automata. Using this model, we come to a very important conclusion in the logics of cellular automata: every logical circuit can be simulated by a reversible 2-dimensional 2-state cellular automaton.

3.1.2. Cellular automata

They constitute a young yet prolific field of research, first investigated by John von Neumann and Stanislaw Ulam in the 1940s. It arises at the boundary between several mathematical (like combinatorics or computability theory) and non-mathematical (like microbiology) branches of science, see [5], [21], [64], [111].

Cellular automata are used for modelling synchronous and uniform processes over large arrays, more precisely over infinite d-dimensional arrays of cells. At each iteration in the discrete time, each cell is updated according to a unique local transition function and the states of the neighboring cells.

Thus, in cellular automata there exist objects that may be interpreted as passive data (the neighboring cells in the initial configuration) and objects that may be interpreted as computing devices. In other words, here computation and construction are just two possible modes of activity. The dynamics is given by an explicit local transition rule by which at every step each cell determines its new state from the current state of its neighbors.

Notice that cellular automata are universal, because they can simulate any Turing machine.

3.1.3. Conservation and reversibility

Also, the main originality of unconventional computing is that just as conventional models of computation make a distinction between the structural part of a computer, which is fixed, and the data on which the computer operates, which are variable, so unconventional models assume that both structural parts and computing data are variable ones. Therefore the essential point of conventional computation is that the physics is segregated once and for all within the logic primitives. Once we have the formal specifications of these primitives and perhaps some design constraints, we can forget about the physics, i.e. about the structural part of a computer. However, the structural part has a permanent evolution in billiard ball model and cellular automata.

Notice that the reversibility of physical processes at a microscopic level makes a distinction between physics and logic in the computation. For example, in the AND-gate three of the four possible input configurations, namely $\langle 0, 0\rangle$, $\langle 0, 1\rangle$, and $\langle 1, 1\rangle$, take the same result 0. As we see, the AND-gate is not reversible (e.g., the NOT-element is reversible).

Therefore to combine physics and logic in the computation, one proposed the idea of conservation which consists in the following principles (see [14], [37], [64])

1. each event has as many output signals as input ones;
2. the number of tokens of each kind is invariant.
3. each event establishes a one-to-one correspondence between the collective state of its input signals and that of its output signals.

3.2. Non-well-founded computing

3.2.1. Interactive computing

Interactive computing, developed in [34], [41]–[44], [109], is based on coinductive methods. Although such a computation model is abstract in the same measure as Turing machines, it assumes a combination of physics and logic in the computation, as well as that in cellular automata or in other unconventional computing models.

In conventional computing, the computation is performed in a closed-box fashion, transforming a finite input, determined by the start of the computation, to a finite output, available at the end of the computation, in a finite amount of time. Therefore physical implementation does not play role in such a computation that may be considered as process in a black box.

Consider now a simple example of interactive computing provided by Peter Wegner to show the role of interactive factors in the computation. Let us consider an automatic car whose task is to drive us across town from point W (work) to point H (home); we shall refer to it as the WH problem. The output for this problem should be a time-series plot of signals to the car's controls that enable it to perform this task autonomously. At issue is what form the inputs should take. We have two possible approaches to solve the WH problem: conventional and interctive/unconventional. Firstly, the car can be equipped with a map of the city, i.e. in the algorithmic scenario using conventional methods, where all inputs are provided a priori of computation. In a static world, such a map is in principle obtainable but ours is a dynamic environment. Secondly, the WH problem can be solved interactively. In this scenario, the inputs, or percepts, consist of a stream of images received in the car, as it is driving from W to H.

Thus, we see the main difference between conventional and unconventional computing. In the first case the computation is off-line (it takes place before the driving begins), and in the second on-line (it takes place as the car drives). This property allows Peter Wegner to claim that interactive/unconventional computation falls outside the bounds of the Church Turing thesis and it is shown to be a more powerful computational paradigm by allowing us to solve computational tasks that cannot be solved algorithmically [41], [109].

Recall that the Church-Turing thesis was initially formulated as follows: the intuitive notion of effective computability for functions and algorithms is formally expressed by Turing machines (Turing) or the lambda calculus (Church). This thesis equated WF logic, lambda calculus, Turing machines, and algorithmic computing as equivalent mechanisms of problem solving. Later, it was reinterpreted as a uniform complete mechanism for solving all computational problems. However, the simple example of the WH problem confirms that Turing machines are inappropriate as a universal foundation for computational problem-solving, because they are too weak to express interaction of object-oriented and distributed systems. Therefore Dina Goldin and Peter Wegner proposed interaction machines as a stronger model that better captures computational behavior for finite interactive computing agents [34], [41]–[44], [109]. Probably, for these machines, the following extension of the Church-Turing thesis should hold: the intuitive notion of sequential interaction is formally modelled by non-WF methods (coinduction, corecursion, etc.). For example, the least fixed point of the equation $S = A \times S$ is the empty set (i.e. it is so from the standpoint of conventional computing), while the greatest fixed point is the set of all streams over S (i.e. it is so from the standpoint of non-WF computing).

Notice, the statement that Turing machines completely express the intuitive notion of computing is a common misinterpretation of the Church-Turing thesis. For instance, Turing asserted in [104] that Turing machines could not provide a complete model for all forms of computation, but only for algorithms. Therefore he defined choice machines as an alternative model of computation, which added interactive choice as a form of computation, and later, he also defined unorganized machines as another alternative that modelled the brain.

Induction and recursion determine enumerable collections of finite structures, while coinduction and corecursion determine non-enumerable collections of infinite structures. As a result, WF logic cannot model interactive/unconventional computing. For example, sound and complete (first-order) logics have a recursively enumerable set of theorems and can formalize only semantic domains with a countable number of distinct properties. Therefore, in particular, the means of WF logics are not sufficient for a syntactical expressibility of all properties of arithmetic over the integers (Gödel's two incompleteness theorems). Thus, Gödel's reasoning may be extended to show that coinductive and corecursive systems are likewise incomplete because they have too many properties to be expressible as theorems of WF logics. However, the question how non-WF logics can syntactically express interactive/unconventional computing is still open.

Dina Goldin and Peter Wegner affirm that the gap between least- and greatest-fixed-point semantics is also the gap between operational (algorithm) and denotational (observation) semantics. It is also the same as the gap between deduction and abduction. It seems to be a restriction, because greatest-fixed-point semantics may be used in deductive non-WF logics, in particular in cyclic proof systems.

3.2.2. Coalgebras and hidden algebras

Instead of algebraic methods, their dual is applied to interactive systems defined by coinductive rules. This dual is called a theory of coalgebras. *Co-*

algebras, developed in [50], [73], [78], [88], [102], consider a notion of observational indistinguishability as bisimulation, a characterization of abstract behaviors as elements of final coalgebras and coinduction as a definition/proof principle for system behavior. *Hidden algebra*, introduced in [38] and further developed in [39], [40], combines algebraic and coalgebraic techniques in order to provide a semantic foundation for the object paradigm. Recall that the object paradigm is described as having: 1. objects with local state and operations that modify or observe them; 2. classes that classify objects through an inheritance hierarchy; and 3. concurrent distributed execution. The theory of hidden algebras is an extension of the theory of many sorted algebras that uses both constructor and destructor operations and a loose behavioral semantics over a fixed data universe for the states of objects.

Hidden algebras were introduced to give algebraic semantics for the object paradigm. One distinctive feature is a split of sorts into visible and hidden, where visible sorts are for data and hidden sorts are for objects. *Hidden logic* is the generic name for various logics closely related to hidden algebra, giving sound rules for behavioral reasoning that are easily automated [84], [85].

Let us remember that algebras and its associated inductive techniques have been successfully used for the specification of data types. Data types can be presented as **F**-algebras using *constructor operations* going into the type, i.e. tuples $\langle A, \alpha \rangle$, where A is an object and $\alpha : \mathbf{F}A \to A$ is a morphism in some category C, with $\mathbf{F} : \mathrm{C} \to \mathrm{C}$. Among **F**-algebras, initial ones $\iota : \mathbf{F}I \to I$ (least fixed points of **F**) are most relevant, their elements denote closed programs. Initial algebras come equipped with an induction principle stating that no proper subalgebras exist for initial algebras. This principle constitutes the main technique used in algebraic specifications for both definitions and proofs: defining a function on the initial algebra by induction amounts to defining its values on all the constructors; and proving that two functions on the initial algebra coincide amounts to showing that they agree on all the constructors.

The theory of coalgebras is viewed as a dualization of the theory of algebras. Object systems are presented as **G**-coalgebras using *destructor operations* going out of the object types, i.e. tuples $\langle C, \beta \rangle$, where C is an object and $\beta : C \to \mathbf{G}C$ is a morphism in some category C, with $\mathbf{G} : C \to C$. Final **G**-coalgebras $\zeta : Z \to \mathbf{G}Z$ (greatest fixed points of **G**) are in this case relevant, they incorporate all **G**-behaviors. The unique coalgebra homomorphism from a coalgebra to the final one maps object states to their behavior. A bisimulation between two coalgebras is a relation on their carriers, carrying itself coalgebraic structure. Bisimulations relate states that exhibit the same behavior. Final coalgebras come equipped with a coinduction principle stating that no proper bisimulations exist between a final coalgebra and itself; that is, two elements of a final coalgebra having the same behavior coincide.

While universal algebras [28] were applied to (WF) logics due to Lindenbaum and Tarski's well-known construction [76], coalgebras begin to be used in (non-WF) logics too [57], [67], [70], [82], [83].

4. Conclusion

A novel computational paradigm concerning a computation in non-WF systems is a burgeoning research area with much potential. The methodological frameworks of the future researches could be non-WF logical calculi, non-Archimedean mathematics, coalgebras, and their application to unconventional computing. Within these frameworks, non-WF computing could be regarded as a new model of non-Church-Turing computation. Advanced techniques for non-WF computing will include:

- Analysis of expressive powers of non-WF formal arithmetic.
- Setting probabilistic algorithms of non-WF computation.
- Applying these algorithms to unconventional computing media.

References

[1] Abramsky, S., The lazy lambda calculus [in:] D. A. Turner, editor, *Resesarch Topics in Functional Programming*, Addison-Wesley, 1977, 65–116.

[2] Abramsky, S., Semantics of Interaction [in:] A. Pitts and P. Dibyer, editors, *Semantics and Logic of Computation*, Cambridge, 1997.

[3] Abramsky, S., A domain equation for bisimulation, *Information and Computation* 92, 1991, 161–218.

[4] Aczel, P., *Non-Well-Founded Sets.* Stanford, 1988.

[5] Adamatzky, A., *Identification of Cellular Automata.* London, Taylor & Francis, 1995.

[6] Adamatzky, A., Hierarchy of fuzzy cellular automata. *J. Fuzzy Sets and Systems*, 62, 1994, 167–174.

[7] Ambler, S., Crole, R., and A. Momigliano. Combining higher order abstract syntax with tactical theorem proving and (co)induction [in:] V.

A. Carreño, editor, *TPHOLs 2002*, vol 2342 of LNCS. Springer Verlag, 2002.

[8] Bachman, G., *Introduction to p-adic numbers and valuation theory*, Academic Press, 1964.

[9] Banks, E.R., Universality in Cellular Automata. *IEEE Ann. Symp. Switching and Automata Theory*. Santa Monica. 11, 1970, 194–215.

[10] Bartels, F., Generalized coinduction, *Math. Structures Comput. Sci.* 13, 2003, 321–348.

[11] Barwise, J., Etchemendy, J., *The Liar*. Oxford UP, New York, 1987.

[12] Barwise, J., Moss, L., *Vicious Circles*. Stanford, 1996.

[13] Barwise, J., Moss, L., *Hypersets*. Springer Verlag, New York, 1992.

[14] Bennett, Ch., Logical reversibility of computation. *IBM J. Res. Develop.* 6, 1973, 525–532.

[15] Bennett, Ch., The thermodynamics of computation – a review. *Int. J. Theor. Phys.* 21, 1982, 905–940.

[16] Bertot, Y., Affine functions and series with co-inductive real numbers. *Mathematical Structures in Computer Science*, Volume 17, Issue 1, 2007, 37–63.

[17] Brotherston, J., Cyclic proofs for first-order logic with inductive definitions [in:] B. Beckert, editor, *TABLEAUX 2005*, volume 3702 of LNAI, Springer-Verlag, 2005, 78–92.

[18] Brotherston, J., Formalised inductive reasoning in the logic of bunched implications, *SAS-14*, volume 4634 of LNCS, Springer-Verlag, August 2007, 87–103.

[19] Brotherston, J., *Sequent Calculus Proof Systems for Inductive Definitions*. PhD thesis, University of Edinburgh, November 2006.

[20] Brotherston, J., Simpson, A., Complete sequent calculi for induction and infinite descent. *LICS-22*, IEEE Computer Society, July 2007, 51–60.

[21] Burks, A., *Essays on Cellular Automata*. Chicago, University of Illinois Press, 1970.

[22] Carnap, R., *Der logische Aufbau der Welt*, Berlin, 1928.

[23] Carnap, R., Empiricism, Semantics, and Ontology, *Revue internationale de philosophie*, 4, 1950, 20–40.

[24] Carnap, R., *Meaning and Necessity*, Chicago, University of Chicago Press, 1947.

[25] Cattaneo, G., Flocchini, P., Mauri, G., Santoro, N., Fuzzy cellular automata and their chaotic behavior. *Proc. International Symposium on Nonlinear Theory and its Applications*, Vol. 4. Hawaii, IEICE, 1993, 1285–1289.

[26] Cattaneo, G., Flocchini, P., Mauri, G., Vogliotti, C. Q., Santoro, N., Cellular automata in fuzzy backgrounds. *Physica D.* 105, 1997, 105–120.

[27] Ciaffaglione, A., Gianantonio, Di P., A certified, corecursive implementation of exact real numbers. *Theoretical Computer Science*, Volume 351, 2006, 39–51.

[28] Cohn, P.M., *Universal algebra.* New York, Evanston, and London, 1965.

[29] Cutland, N. J. (edit.), *Nonstandard Analysis and its Applications.* Cambridge University Press, Cambridge, 1988.

[30] Davis, M., *Applied Nonstandard Analysis*, John Wiley and Sons, New York, 1977.

[31] De Moura, F. L. C., Ayala-Rincón, M., Kamareddine, F., Higher-Order Unification: A structural relation between Huet's method and the one based on explicit substitutions. *J. Applied Logic* 6(1), 2008, 72–108.

[32] Deutsch, D., Quantum theory, the Church-Turing principle and the universal quantum computer. *Proc. R. Soc. London* A400, 1985, 97–117.

[33] Deutsch, D., *The Fabric of Reality.* Allen Lane, New York, 1997.

[34] Eberbach, E., Wegner, P., Beyond Turing Machines. *Bulletin of the EATCS*, 81, 2003, 279–304.

[35] Feynman, R., Quantum-Mechanical Computers. *Foundations of Physics*, 16, 1986, 507–531.

[36] Fiore, M.P., A coinduction principle for recursive data types based on bisimulation. *Proc. 8th Conf. Logic in Computer Science (LICS'93).* 1993, 110–119.

[37] Fredkin, E., Toffoli, T., Conservative Logic. *Int. J. Theor. Phys.* 21, 1982, 219–253.

[38] Goguen, J., Types as theories [in:] G. M. Reed, A. W. Roscoe, and R. F. Wachter, editors, *Topology and Category Theory in Computer Science*, Oxford, 1991, 357–390.

[39] Goguen, J., and R. Diaconescu. Towards an algebraic semantics for the object paradigm [in:] Hartmut Ehrig and Fernando Orejas, editors, *Proceedings Tenth Workshop on Abstract Data Types*, Springer, 1994, 1–29.

[40] Goguen, J., and G. Malcolm, A hidden agenda, *Theoretical Computer Science*, 245 (1), August 2000. 55–101. Also UCSD Dept. Computer Science & Eng. Technical Report CS97–538, May 1997.

[41] Goldin, D., Smolka, S., Wegner, P., (eds.) *Interactive Computation: the New Paradigm.* Springer 2006.

[42] Goldin, D., Wegner, P., The Church-Turing Thesis: Breaking the Myth. *CiE*, 2005, 152–168.

[43] Goldin, D. Q., Smolka, S. A., Wegner, P., Turing Machines, Transition Systems, and Interaction. *Electr. Notes Theor. Comput. Sci.* 52(1), 2001.

[44] Goldin, D. Q., Smolka, S. A., Attie, P, Sonderegger, E., Turing Machines, Transition Systems, and Interaction, *Information and Computation Journal*, Volume 194, Issue 2, Nov. 2004, 101–128.

[45] Gordon, A. D., Bisimilarity as a theory of functional programming, *Theor. Comput. Sci.* 228, 1999, 5–47.

[46] Hertz, J., Krogh, A., Palmer, R. G., *Introduction to the Theory of Neural Computation.* Redwood City, Addison-Wesley, 1991.

[47] Howe, D. J., Equality in lazy computation systems, *Proc. 4th IEEE Symposium on Logic in Computer Science*, 1989, 198–203.

[48] Howe, D. J., Proving congruence of bisimulation in functional programming languages, *Inf. and Comp.* 124, 1996.

[49] Hurd, A., Loeb, P. A., *An Introduction to Nonstandard Real Analysis*, Academic Press, New York.

[50] Jacobs, B., and J. Rutten, A tutorial on (co)algebras and (co)induction, *EATCS Bulletin* 62, 1997, 222–259.

[51] Khrennikov, A. Yu., p-adic quantum mechanics with p-adic valued functions, *J. Math. Phys.*, 32 (4), 1991, 932–937.

[52] Khrennikov, A. Yu., *p-adic valued distributions in mathematical physics.* Kluwer Academic Publishers, Dordrecht, 1994.

[53] Khrennikov, A. Yu., *Interpretations of Probability*, VSP Int. Sc. Publishers, Utrecht/Tokyo, 1999.

[54] Khrennikov, A. Yu., and Schumann, A., Logical Approach to p-adic Probabilities, *Bulletin of the Section of Logic*, 35 (1), 2006, 49–57.

[55] Koblitz, N., *p-adic numbers, p-adic analysis and zeta functions*, second edition, Springer-Verlag, 1984.

[56] Kozen, D., Coinductive proof principles for stochastic processes [in:] Alur, R., ed.: *Proc. 21st Symp. Logic in Computer Science (LICS'06)*, IEEE, 2006, 359–366.

[57] Kurz, A., Specifying coalgebras with modal logic. *Theor. Comp. Sci.*, 260(1–2), 2001, 119–138.

[58] Mahler, K., *Introduction to p-adic numbers and their functions*, second edition, Cambridge University Press, 1981.

[59] Martin, O., Odlyzko, A., Wolfram, S., Algebraic properties of cellular automata. *Comm. Math. Phys.* 93, 1984, 219–258.

[60] Martin, R. M., On 'analytic', *Philosophical Studies*, 3, 1952, 42–47.

[61] Matthes and Uustalu, Substitution in non-well-founded syntax with variable binding, *TCS: Theoretical Computer Science* 327, 2004.

[62] Milner, R., *Communication and Concurrency.* Prentice-Hall, 1989.

[63] Milner, R., and Tofte, M., Co-induction in relational semantics. *Theoretical Comput. Sci.*, 87, 1991, 209–220.

[64] Morita, K., Imai, K., *Logical Universality and Self-Reproduction in Reversible Cellular Automata.* ICES, 1996.

[65] Moran, A., *Natural Semantics for Non-Determinism*, Master's thesis, Dept. of Computing Science, Chalmers University of Technology, 1994.

[66] Moss, L., Parametric Corecursion, *Theor. Comput. Sci.* 260, 2001, 139–164.

[67] Moss, L.S., Coalgebraic logic. *Ann. Pure & Appl. Logic*, 96(1–3), 1999, 277–317. Erratum in Ann. Pure & Appl. Logic, 99(1–3), 1999, 241–259.

[68] von Neumann, J., *Theory of Self-Reproducing Automata.* Univ. of Illinois Press, 1966.

[69] Nygaard, M., and G. Winskel, HOPLA – A higher-order process language, Proc., *13th Intl. Conf. on Concurrency Theory*, LNCS 2421, 2002.

[70] Pattinson, P., Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. *Theor. Comp. Sci.*, 309(1-3), 2003, 177–193.

[71] Paulson, L. C., Mechanizing coinduction and corecursion in higher-order logic. *Journal of Logic and Computation*, 7(2), Mar. 1997, 175–204.

[72] Pavlović, D., Escardó, M.H., Calculus in coinductive form, *Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science.* 1998, 408–417.

[73] Pavlović, D., and V. Pratt. The continuum as a final coalgebra, *Theor. Comp. Sci.*, 280 (1-2), 2002, 105–122.

[74] Penrose, R., *The Emperor's New Mind: Concerning Computers, Minds and The Laws of Physics*, 1989.

[75] Quine, W. V. O., Two Dogmas of Empiricism, *The Philosophical Review*, 60, 1951, 20–43.

[76] Rasiowa, H., Sikorski, R., *The mathematics of metamathematics.* Warszawa, Panstwowe Wydawnictwo Naukowe, 1963.

[77] Reichel, H., An approach to object semantics based on terminal co-algebras. *Math. Struct. in Comp. Sci.*, 5, 1995, 129–152.

[78] Reichel, H., and J. Rutten, editors, *Coalgebraic Methods in Computer Science*, Elect. Notes in TCS 11, 1998.

[79] Reiter, C. A., Fuzzy automata and life. *Complexity*, 7 (3), 2002, 19–29.

[80] Robinson, A., *Non-Standard Analysis*, North-Holland Publ. Co., 1966.

[81] Rosenblatt, F., *Principles of Neurodynamics.* Spartan, 1962.

[82] Röiger, M., From modal logic to terminal coalgebras. *Theor. Comp. Sci.*, 260(1-2), 2001, 209–228.

[83] Röiger, M., Coalgebras and modal logic [in:] H. Reichel, editor, *Coalgebraic Methods in Computer Science*, number 33 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 2000.

[84] Roşu, G., *Hidden Logic*, PhD thesis. University of California at San Diego, 2000.

[85] Roşu, G., and J. Goguen. Hidden congruent deduction [in:] Ricardo Caferra and Gernot Salzer, editors, *Automated Deduction in Classical and Non-Classical Logics*, Springer, 2000, 252–267.

[86] Rutten J. J. M. M., A coinductive calculus of streams. *Math. Struct. in Comp. Sci.*, 15(1), 2005, 93–147.

[87] Rutten, J. J. M. M., Behavioral differential equations: a coinductive calculus of streams, automata, and power series, *Theor. Comput. Sci.* 308, 2003, 1–53.

[88] Rutten, J. J. M. M., Universal coalgebra: a theory of systems, *Theor. Comput. Sci.* 249(1), 2000, 3–80.

[89] Rutten, J. J. M. M., and D. Turi, On the foundations of final coalgebra semantics: non-well-founded sets, partital orders, metric spaces, *Math. Structures Comput. Sci.* 8, 1998, 481–540.

[90] Rutten, J. J. M. M., Processes as terms: non-well-founded models for bisimulation. *Math. Struct. in Comp. Sci.*, 2(3), 1992, 257–275.

[91] Santocanale, L., A calculus of circular proofs and its categorical semantics [in:] M. Nielsen and U. Engberg, editors, *Proc. of FoSSaCS 2002*, Grenoble, Apr. 2002, Springer-Verlag LNCS 2303, 357–371.

[92] Schumann, A., Non-Archimedean Valued Sequent Logic, *Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'06)*, IEEE Press, 2006, 89–92

[93] Schumann, A., p-Adic Multiple-Validity and p-Adic Valued Logical Calculi, *Journal of Multiple-Valued Logic and Soft Computing*, 13 (1–2), 2007, 29–60.

[94] Schumann, A., Non-Archimedean Valued Predicate Logic, *Bulletin of the Section of Logic*, 36 / 1–2, 2007, 67–78.

[95] Schumann, A., Non-Archimedean Fuzzy Reasoning, *Fuzzy Systems and Knowledge Discovery (FSKD'07)*, IEEE Press, 2007.

[96] Schumann, A., Smarandache, F., *Neutrality and Many-Valued Logics.* Rehoboth, New Mexico: American Research Press, 2007.

[97] Schumann, A., Non-Archimedean Valued and p-Adic Valued Fuzzy Cellular Automata, *Journal of Cellular Automata*, 2008 (to appear).

[98] Schumann, A., Non-Archimedean Fuzzy and Probability Logic, *Journal of Applied Non-Classical Logics*, 18/1, 2008, 29–48.

[99] Schumann, A., Non-well-Founded Probabilities on Streams, *SMPS'08* (to appear)

[100] Solomon, P., Frank, D. J., The case for reversible computation. *Proc. 1994 International Workshop on Low Power Design.* Napa Valley, CA, 1994 93–98.

[101] Sprenger, Ch., and Mads Dam. On the structure of inductive reasoning: circular and tree-shaped proofs in the μ-calculus, *FOSSACS 2003*, volume 2620 of LNCS, 2003, 425–440.

[102] Tews, H., Coalgebras for binary methods: Properties of bisimulations and invariants. *Inf. Theor. et Appl.*, 35(1), 2001, 83–111.

[103] Turing, A., Intelligent machinery [in:] D. C. Ince, *Mechanical intelligence*, North-Holland, 1992, 107–127.

[104] Turing, A., On Computable Numbers with an Application to the Entscheidungsproblem, *Proceedings of the London Math Society*, 2 (42), 1936, 173–198.

[105] Vink, E.P. de, and J.J.M.M. Rutten. Bisimulation for probabilistic transition systems: a coalgebraic approach. *Theor. Comp. Sci.*, 221, 1999, 271–293.

[106] Vladimirov, V. S., Volovich, I. V. and Zelenov, E. I., *p-Adic Analysis and Mathematical Physics*, World Scientific, Singapore, 1994.

[107] Vladimirov, V. S. and Volovich, I. V., p-Adic Quantum Mechanics, *Commun. Math. Phys.*, 123, 1989, 659–676.

[108] Wee, W.G., Fu, K.S., A Formulation of Fuzzy Automata and its Application as Model of Learning Systems. *IEEE Trans. Systems Man. Cyber*, 5, 1969, 215–223.

[109] Wegner, P., Why Interaction Is More Powerful Than Algorithms. *Commun. ACM* 40(5), 1997, 80–91.

[110] Wolfram, S., *A New Kind of Science.* Wolfram Media, Inc., 2002.

[111] Wolfram, S., *Theory and applications of cellular automata.* World Scientific, 1986.