Towards Empirical Computer Science
Peter Wegner, Draft, June 10

Table of Contents

Part I: Empirical Computer Science
1. **Interaction Machines**
2. **Incompleteness**
3. **Observation Equivalence and Expressiveness**
4. **On-Line Models of Observation**

Part II: Physics and Computation
5. **Observability in Physics and Computation**
6. **Interactive Realism**
7. **Interaction as Observation**
8. **Explanatory Power and Computational Expressiveness**
9. **Quantum Turing Machines and Church's Thesis**

Part III: Philosophy of Empirical Computation
10. **Rationalism, Empiricism, and Realism**
11. **Platonic Empiricism**
12. **The Interactive Turing Test**
13. **Interactive Extensions of Logic**
14. **Acknowledgements**
15. **References**

Abstract: Part I presents a model of interactive computation and a metric for expressiveness, part II relates physical and computational models by observation equivalence, and part III considers empirical models from a philosophical perspective. Interaction machines, which extend Turing machines to interaction, are shown in Part I to be more expressive than Turing machines by a direct proof, by adapting Godel's incompleteness result, and by observability metrics. Observational equivalence provides a tool and metric for comparing the expressiveness of algorithms and interactive systems that also captures the explanatory power of physical theories. The change of focus from algorithms specified by computable functions to interaction specified by observation equivalence captures the essence of empirical computer science.

Part II relates interaction in models of computation to observation in the natural sciences. Explanatory power in physics is specified by the same observability metric as expressiveness in interactive systems. The relation between observation and realist models of inner structure is characterized in terms of induction, abduction, and Occam's razor (sections 5 and 10). Interactive realism extends the hidden-variable model of Einstein to hidden interfaces that provide extra degrees of freedom to formulate hypotheses with testable predictions conforming with quantum theory. The multiple-hidden-interface model corresponds to the multiple-concurrent-writers problem, showing that models of computation provide a framework for the study of fundamental properties of physical models. Greater expressiveness of collaborative computational observers (writers) than single observers implies that hidden-interface models are more expressive than hidden-variable models.

Part III relates the evolution in computing from algorithms to interaction to that in physics from rationalism to empiricism. Interactive extensions of Plato's cave metaphor show how Platonic rationalism can be extended to empiricism. Turing test machines with hidden interfaces express interactive and collaborative thinking richer than the traditional Turing test. Interactive (nonmonotonic) extensions of logic such as the closed-world assumption suggest that interactiveness is incompatible with monotonic logical inference. Procedure call, atomicity of transactions, and taking a fixed point are techniques for closing open systems similar to "preparation" followed by "observation" of a physical system. Pragmatics is introduced as a framework for extending logical models with a fixed syntax and semantics to multiple-interface models that support collaboration among clients sharing common resources.

Towards Empirical Computer Science
Peter Wegner, Draft, June 10

Part I: Empirical Computer Science

Part I introduces a model of interactive computation and a metric for expressiveness. Section 1 extends Turing machines (TMs) to interaction machines (IMs) that interact dynamically with an external environment, showing that IMs are not reducible to or expressible by TMs and that they can solve a larger class of problems than TMs. Section 2 shows that IMs cannot be completely specified by first-order logic, extending to interactive computing Godel's result that syntactic proof theory is too weak to formalize mathematical semantics. Section 3 develops models of observation and testing that provide a uniform metric of expressiveness for TMs and IMs, while section 4 reviews computational models of on-line observation as a prelude to exploring physical observability in Part II.

Interactive systems express problems like modeling the behavior of bank accounts and airline reservation systems involving sequences or patterns of interactions that cannot be modeled by computable functions, TMs, or noninteractive algorithms. Algorithms express the problem-solving power of computing agents by their inner cleverness, while problems solved by a person or a physical computing agent require interaction with unpredictable, open environments. Exams and IQ tests, which measure the inner cleverness of people, are generally poor predictors of success in solving real-world problems because they fail to measure interactive problem-solving ability. Computable functions, which express the inner cleverness of TMs and noninteractive algorithms, are too weak a model for interactive problem solving. Church's thesis, which equates the intuitive notion of computing to formal computability by TMs, is too weak in its intuitive notion of the nature of computing.

Observational equivalence provides a metric of expressiveness according to which interactive systems are more expressive than algorithms, playing a role in specifying the behavior of interactive computing comparable to that of computable functions for algorithms. Interactive systems are distinguished from algorithms by the fact that they allow more complex kinds of observations. For TMs TM1, TM2 that compute distinct functions there exist single observations for which TM1 and TM2 have different output, while distinct interaction machines IM1, IM2 may require sequences or patterns of observation to distinguish them. The requirement of TMs that they always start in the same initial state precludes sequences and patterns of input-output pairs from expressing richer behavior than single observations. Defining system expressiveness of systems by observable behavior rather than by inner system-transformation power captures the essence of empirical computer science. We show in part II that expressiveness in physics and other empirical sciences is likewise captured by specifying all permitted observations.

1. Interaction Machines

Algorithms are recipes for computing functions that noninteractively transform inputs into outputs. They are modeled by *Turing machines (TMs)* with instructions that read a symbol from their tape, perform a state transition determined by their state and input symbol, write a symbol on the tape, and move one tape position to the right or left. TMs start in an initial state and compute a finite output from a finite input by a finite number of computation steps. Functions computable by TMs can be equivalently specified by other computing formalisms like the lambda calculus, recursively enumerable sets, and rewriting rules of phrase-structure grammars. All programming languages are equivalent to TMs in their algorithmic problem-solving power. This robustness of expressive power determines a stable, formal notion of computability that lies at the core of algorithm analysis and complexity theory. Church's thesis equates the robust formal notion of computable functions specified by TMs with the intuitive notion of computability.

However, algorithms fail to express the interactive behavior of objects, agents, distributed systems, personal computers, and embedded systems. They focus on inner processes of algorithmic transformation, while the technology of the 1990s increasingly focuses on the interaction of agents with their environment and collaboration among multiple users. Interactive computation is modeled by *interaction machines*

(IMs), which extend TMs with input and output actions that interact dynamically with the external environment. IMs can solve a larger class of problems than TMs, including problems like driving and process control that are inherently interactive [We2]. IMs extend the notion of computation from a single algorithmic interaction to sequences of history-dependent interactions and more generally to patterns of concurrent interactions that share common system resources. We distinguish interactive from noninteractive systems, define IMs as a representative class of interactive systems that extend TMs, and focus on persistent TMs, a restricted class of IMs that concretely specifies an important class of practical interactive systems.

Definition 1 (interaction property): A computing agent has the interaction property if it has input and output actions that interact with an external environment not under its control.

TMs do not have the interaction property since they noninteractively compute outputs from inputs, while objects in object-oriented programming, agents in artificial intelligence, components in distributed systems, and physical computers have the interaction property. The interaction property is an effectively testable property of computing agents, though the class of all computing agents with the interaction property cannot be enumerated. Interaction machines that extend TMs with the interaction property play a role in empirical computer science analogous to that of TMs for algorithms.

Definition 2 (interaction machines): An interaction machine (IM) is a TM extended with input and output actions having the interaction property. Whereas TMs have finite input tapes, IMs have input streams whose elements are dynamically supplied by an external mechanism not under its control.

Definition 2 encompasses many forms of interaction, while definition 3 specifies a narrow class of machines that model sequential interaction by extending TMs with a persistent working tape.

Definition 3 (persistent Turing machines (PTMs)): A PTM is a multitape TM with a persistent work tape. It executes sequences of algorithms, corresponding to sequences of observations, without reinitializing its working tape for successively executed algorithms.

PTMs model sequences of history-dependent interactions of objects and agents. Expressing sequential interaction by persistence may at first seem surprising, but follows from the way in which IMs are defined. TMs are a specialized form of PTM in which the state does not persist because of reinitialization for each input action. From the viewpoint of an observer, PTMs that preserve their state between interactions are simpler than TMs that reinitialize their state (erase their memory) for every interaction.

PTMs are a robust and useful class of IMs that model stacks, queues, and bank accounts. They are equivalent in their expressive power to sequential interaction machines (SIMs) that sequentially transform elements of a single input stream to an output stream. Once the idea of forms of computing stronger than TMs is introduced, many forms of interactive engagement that differ in their interactive expressiveness can be identified. There is strong evidence that multiple-stream IMs (MIMs) that support multiple concurrent input streams and collaborative access to shared resources are more expressive than SIMs, contrasting with the fact that multiple-tape TMs are no more expressive than single-tape TMs [We3]. The greater expressiveness of MIMs over SIMs and PTMs plays an important role in exploring the expressiveness of physical systems in part II and in the analysis of the interactive Turing test in part III.

IMs are not reducible to TMs because interaction cannot be modeled by finite input tapes.

Proposition 1 (irreducibility): IMs are not expressible by or reducible to TMs.

Proof: IMs cannot be modeled by TMs with a finite initial input tape because any finite input stream can always be extended by an adversary in the environment [We3]. Thus IMs are as powerful as TMs with infinite initial input tapes, which are known to be more expressive than TMs.

An alternative proof of irreducibility based on Godel incompleteness is given in section 2 below. Arguments based on observability metrics that show IMs to be more expressive even for finite computations [We2] are developed in section 3.

The failure of TMs to model interactive computation was already evident to Turing, who mentions choice machines (c-machines) that allow externally controlled choices [Tu1] but focuses exclusively on automatic machines.TMs model automatic computation, while IMs extend the notion of what is computable to nonautomatic (interactive) computation. Irreducibility is corroborated by theoretical evidence from Turing and Godel and by practical evidence from software engineering and artificial intelligence [We2]. IMs can model TMs with oracles, which were shown by Turing himself to be more powerful than TMs [Tu3, Pa]. They can model reactive (embedded) systems that provide services over time, shown in [MP] to be more expressive than TMs, as well as objects, agents, and distributed systems.

2. Incompleteness

Logic expresses both syntactic rules for noninteractively proving theorems from axioms by rules of inference (proof theory) and semantic notions of soundness and completeness that relate syntactic processes of proof to semantic properties of a modeled domain in which formulae are interpreted (model theory). There are trade-offs between reasoning and modeling [We4]: purely rule-based syntactic reasoning is too weak to capture semantic properties of mathematical, computational, or physical systems. Model builders must choose between complete formalizability and modeling power. TMs focus on formalizability at the expense of modeling power while IMs choose modeling power over formalizability.

A logic is sound if all its theorems are true and complete if all true assertions of the domain being modeled are theorems. Soundness and completeness are desirable mathematical properties that make logic mathematically tidy but restrict expressiveness. Godel showed that mathematically well-behaved autonomous systems like arithmetic over the integers cannot be completely formalized by logic because not all their properties are syntactically expressible as theorems. We extend Godel's reasoning to show that interactive and empirical systems are likewise incomplete because they have too many properties to be expressible as theorems of a sound and complete logic.

Sound and complete first-order logics have a recursively enumerable (RE) number of theorems and, because completeness requires all true formulae to be theorems, can formalize only semantic domains with an RE number of distinct properties. Arithmetic over the integers is incomplete because its true properties cannot be recursively enumerated (for any sound formalization we can find a true property of the integers that cannot be proved). Godel's incompleteness result holds for any domain whose true properties cannot be recursively enumerated, including empirical domains modeled by interactive systems.

Incompleteness occurs when the number of true facts or observable properties of a system cannot be recursively enumerated by theorems. For interactive systems, properties correspond to observations and the number of properties corresponds to the number of distinct observations. Algorithms and TMs have only an RE number of properties, while interactive systems have an open-ended number of properties that is generally not RE and cannot therefore be expressed as theorems of a complete first-order logic.

Proposition 2 (completeness): A sound and complete first-order logic (SCL) can model only domains with an RE number of properties.

Proof: An SCL allows only an RE number of theorems to be proved. Since SCLs are sound and complete, the number of true assertions is in one-to-one correspondence with the number of theorems. Thus, the number of true assertions of domains modeled by an SCL is at most RE.

Completeness restricts the expressiveness of domains modeled by an SCL: domains whose number of properties is not RE are necessarily incomplete.

Corollary 2.1 (incompleteness): Any system with a non-RE number of properties cannot be represented by an SCL.

Proof: Immediate.

Incompleteness of arithmetic over the integers was proved directly by Godel using a diagonalization argument. By viewing this result indirectly as a corollary of the more general result that SCLs cannot model systems with a non-RE number of properties we gain additional insights into the nature of incompleteness. Incompleteness results apply equally to mathematical systems with an autonomous semantics like the integers and to nonmathematical intuitive notions like interactive computability.

Corollary 2.2 (intuition): Any intuitive notion about which the number of questions that can be asked is not RE cannot be formalized by the first-order predicate calculus or any SCL.
Proof: Immediate.

Incompleteness distinguishes interactive systems from algorithms. In part II, formal incompleteness of computing systems is related to descriptive incompleteness of physical models of the real world and empiricism is distinguished from rationalism by incompleteness.

Corollary 2.3 (interaction): Interactive systems that interact with environments over unbounded time horizons cannot be formalized by the first-order predicate calculus or any SCL.
Proof: The number of interactive binary input streams for unbounded time horizons is nonenumerable.

Algorithms are carefully defined so that their number of true properties is RE. We show in section 10 that algorithms, in restricting the intuitive notion of problem solving to inner cleverness, are a rationalist abstraction of computing and that the transition from algorithmic to interactive models (abstractly captured by incompleteness) corresponds to that from rationalism to empiricism.

Proposition 3 (algorithms): Algorithms are abstractions with an RE number of true properties.
Proof: The number of true properties of an algorithm corresponds to the number of elements in the argument domain and is RE if the input domain is RE.

Incompleteness shows the limitations of syntactic formalisms in expressing semantic behavior. Hilbert's program for formalizing mathematics assumed it was possible to express any mathematical problem as a decision problem (predicate) in a sound and complete logic. Incompleteness, which implies that arithmetic over the integers cannot be expressed as a decision problem, is here extended to show that interactive computational and physical systems cannot be so expressed. The common property of nonformalizable systems is that the domain being formalized has a non-RE number of true properties. Incompleteness plays a role for non-RE domains that is analogous to the role of undecidability for RE domains. Undecidability is a limitation on the computability of dynamic properties of RE domains, while incompleteness is a limitation on the specifiability of static properties of non-RE domains.

Irreducibility of IMs to TMs was proved in section 1 by showing that IMs cannot be expressed by TMs with a finite initial tape. Here it is proved by appealing to a general limitation of formal systems that lies at the core of Godel's incompleteness result. Both methods of proof are based on the nonenumerability of the set of subsets of an enumerably infinite domain. The mathematical distinction between rational and real numbers is the basis for the physical distinction between rationalist and empiricist (realist) models of physics and computation. The mathematical notion of incompleteness is here related to the intuitive notion of descriptive incompleteness and is singled out as a characterizing property of both the algorithm/interaction and rationalism/empiricism dichotomies.

Interactive problem solving requires a choice at each interactive step which, if the choice is supplied (by an oracle), reduces interactive to algorithmic problem solving. Interactive computations can be described algorithmically after they have occurred, just as solutions to problems in NP have polynomial-time solutions in P. Interactive computation is asymmetric with regard to time because moving forward in time requires considering an exponentially increasing number of contingencies that in the limit becomes nonenumerable, while viewing a computation after it has happened does not require exponential effort and is in the limit enumerable.

TMs are defined extensionally by finite specifications. The set of all TMs can be enumerated, while IMs cannot be enumerated because the interaction property is testable but not constructive. Expressiveness for interactive systems is defined in section 3 in terms of the ability of observers to make observational distinctions. From this viewpoint, TM behavior is completely characterized by a single observation, while IM behavior may depend on unbounded sequences or patterns of observations. Observational expressiveness allows behavior of algorithms and interactive systems to be measured by the same metric. Part II shows that this same metric expresses the explanatory power of physical theories.

Algorithms specify sequences of inner actions whose observable effects are computable functions. Functionally-equivalent algorithms are observation-equivalent when observation of inner actions is precluded. Algorithms that compute the same function can be very different: bubblesort, mergesort, and quicksort are substantially different observation-equivalent algorithms for sorting. Equivalence of algorithms (programs) is known to be undecidable and not even partially decidable: the class of algorithms that realize a given function is not RE. Observation equivalence of algorithms cannot be verified by an SCL.

3. Observation Equivalence and Expressiveness

Equivalence relations are a mathematical tool for specifying abstractions that ignore properties judged irrelevant and focus on properties judged relevant. Computing deals with behavioral (observational) abstractions, specified by observation equivalence, that ignore unobservable inner actions and focus on observationally distinguishable behavior. Observation equivalence provides a uniform metric for specifying behavior of both procedure and data abstractions. Procedure abstractions determine history-independent input-output behavior that ignores inner interactions, while data abstractions determine history-dependent sequences of interactions and ignore both inner actions and data structures.

Definition 4 (observation equivalence): Two systems S1, S2 are observation-equivalent with respect to an observer O if all possible observations that O can make on S1 have the same effect when made on S2 and conversely. Indistinguishable systems are said to be observation-equivalent for O, while distinguishable (inequivalent) systems can be distinguished by observations of O called distinguishability certificates.

Observation equivalence is a constraint on behavior determined by the set of observations of O. Each observation of O determines the constraint that observed systems behave the same for that observation, while equivalence for O specifies the union of such constraints. Enlarging the set O imposes a stronger equivalence constraint with greater discriminating power and finer equivalence classes.

Interaction is more naturally specified by constraints on all possible behaviors than by composition of inner state-transition actions. Compositional behavior is useful when components are created to specify how they behave for a single interaction, but once components have been created their interactive behavior is better specified by constraints. Interactive composition P1||P2 of two components P1, P2 can be specified by constraints imposed by each component on the behavior of the other [We3]: P2 causes the behavior of P1 to be constrained by restricting its interaction. An agent constrained to perform a specific task, or a person who works for an organization, loses interactive freedom by focusing (specializing) their energies to interact with a specific external component. If the composite system is closed, then composite behavior is algorithmic even though unconstrained behavior of the open components is nonalgorithmic [We3]. Providing a system with an environment constrains system behavior to that environment.

Observation equivalence abstractly expresses indistinguishability for all possible application domains, while nonequivalence abstractly expresses distinguishability. Refinement of observation equivalence expresses greater distinguishability (greater expressiveness), while selection of a representative element of an equivalence class corresponds to selection of an implementation for models of computation and to an explanation of observed behavior by unobservable inner structure for models of the natural sciences.

The idea of behavior specification by constraints arises in contexts such as [Sa], where system behavior is modeled by ask and tell constraints. Because constraint models of computation are naturally extensible to interaction and can handle partial (incomplete) specifications, they are likely to become increasingly important as a tool for interactive problem solving.

Correctness can be defined as observation equivalence between a system and an ideal behavior specification that defines a correctness criterion, while testing can be defined as an observation equivalence between a system and a partial behavior specification.

Definition 5 (correctness and testing): A system S satisfies a behavior specification Sp defined by a set of observations O if S is observation-equivalent to Sp for all observations of O. Correctness is an observation equivalence relation between a system specification and observable system behavior. Testing is an observation equivalence relation between a partial system specification and observable system behavior.

Specifications Sp for algorithms are computable functions. Correctness of an algorithm S for a specification Sp can sometimes be proved but is in general undecidable. Specifications Sp for interactive systems describe behavior for sequences and patterns of observations that cannot be given in closed form because the set of all possible observations with distinct observable behavior cannot be recursively enumerated. Correctness for interactive systems is unprovable in principle because of incompleteness. Testing of S for a subset of observations becomes correspondingly more important, since complete correctness is meaningless and partial correctness through testing is the only game in town.

Dijkstra used the fact that testing can show only the presence of bugs but not their absence as an argument against testing and in favor of correctness proofs. This argument is shaky even for algorithms but becomes untenable for incomplete interactive systems for which correctness is in principle unachievable and testing of selected observations is the only mechanism for gaining confidence in system behavior. This relative view of correctness echoes that of the theory of relativity that physical quantities like mass can be defined only relative to the observer's frame of reference (see Part II). Empirical computer science focuses on testing partial observable behavior against incomplete specifications. TMs elevate correctness for single observations into an absolute correctness criterion by their initial-state requirement, while PTMs gain expressiveness by eliminating their initial-state requirement.

Correctness is generally viewed as an absolute rather than relative notion, being identified with the functional correctness of algorithms. When correctness is extended to encompass interactive as well as algorithmic expressiveness, absolute correctness must be abandoned, correctness becomes an entirely relative notion, and the distinction between correctness and testing disappears. Algorithmic correctness becomes partial correctness relative to observers that can make only single observations, a viewpoint natural for algorithms but inadequate for PTMs and interactive systems.

Observed behavior of systems depends on the class of systems being observed and the kinds of observations that observers are allowed to make. Rules of engagement for observers can differ along many dimensions. However, one of the most significant distinctions is that between off-line observers that can make only single observations and on-line observers that can make sequences and patterns of observations.

Definition 6 (off-line and on-line observers): On-line (interactive) observers can interact with the system they observe while they are observing it, while off-line (algorithmic) observers cannot.

TMs are carefully defined so that on-line observers cannot gain any information over and above that gained by off-line observers. Observation of interactive behavior requires both systems that can interact, such as PTMs, and on-line observers that observe the behavior of interactive systems. Observers of PTMs can elicit new information by sequences of on-line observations and demonstrate that interactive systems are more expressive than algorithms.

Expressive system behavior cannot be detected if observers cannot observe it. Off-line observers cannot detect interactive behavior because they cannot distinguish it from algorithmic behavior, just as a blind person cannot distinguish balls of different colors. Failure to detect that interactive systems have richer behavior than algorithms was due in part to the fact that observers were restricted to be off-line.

Observers that distinguish a greater range of behavior can measure the expressive power of systems that perform a greater range of tasks. Defining expressiveness by the distinguishing power of observers rather than by system transformation power captures the essence of empirical computer science.

Definition 7 (expressiveness of observers): An observer O1 is as expressive as O2 for a class of systems S if observation equivalence for O1 implies observation equivalence for O2 - that is, if O2 cannot make any distinctions that cannot also be made by O1. If the behavior distinguishable by O1 is a superset of that of O2, O1 is more expressive than O2 and O2 is less expressive than O1.

Expressiveness is a property of observers defined relative to a class of systems, while equivalence is a property of systems defined relative to a class of observers. Expressiveness is inversely related to abstraction: greater expressiveness means less abstraction and conversely. The goal of observation in both computation and physics is to gain knowledge by reducing abstraction and increasing expressiveness.

Expressiveness is the ability to interact with an environment. It is more naturally defined by the power to observe behavior for multiple interactions than by transformation power for single observations. This accords with expressiveness for physical objects, where inner state transitions of objects or people cannot be observed and rules of engagement for interacting with observed objects determine behavior. Interactive systems are more expressive than algorithms because their observable behavior specifies interactive tasks not describable by algorithms by refining observation equivalence relations.

Algorithms and TMs permit only atomic input-output observations of computing systems. Persistent TMs permit sequences of input-output pairs to make observational distinctions that are more fine-grained than individual observations. IMs that admit concurrent (collaborative) observers at multiple interfaces, like airline reservation systems, have nonsequential patterns of observation that are strictly more expressive than PTMs, confirming the intuition that collaboration cannot be expressed by sequential interactive computation. Multiple collaborative observers (writers) cause systems to appear nondeterministic from the viewpoint of any single observer, though they are globally deterministic (see part II).

4. On-Line Models of Observation

Functional equivalence relations that express off-line system behavior for single observations can be refined by on-line observation of outer interaction histories or inner implementation properties:

outer behavior: observe sequences and patterns of interaction that refine input-output equivalence

inner behavior: observe inner properties of implementations by looking inside black boxes

Observation of both outer behavior patterns and inner system properties refine knowledge gained from single input-output observations. Both kinds of knowledge can be modeled by a refinement of observation equivalence for single observations. Sequences and patterns of observations express interactive behavior, while inner algorithmic steps express interaction during the execution of a given algorithm. Physics permits both outer and inner observation is some contexts, while contexts that allow only outer behavior to be observed require inner properties to be inferred by abduction. Sequential observers are modeled by PTMs, while inner observers are illustrated for finite-state grammars (FSGs) and finite-state automata (FSAs).

By extending multitape TMs to preserve the content of their working tape between observations (algorithm executions), PTMs allow on-line observers to detect new behavior not distinguishable by single observations of off-line observers. A distinguishability certificate (DC) of length k for two PTMs S1, S2 consists of k input-output pairs of the form {(i1, o1), (i2,o2), ... , (ik,ok)} observable by one of the systems but not by the other. We show in [GW] that for any k>0 there exist PTMs whose shortest DC (SDC) is of length k. PTMs that store results of their first k computations in a buffer of length k and output the buffer at the (k+1)th step have an SDC of length k+1. In contrast, TMs that compute distinct functions have an SDC of length 1, showing that TMs can be distinguished by less expressive obsevers than PTMs.

Off-line observers cannot distinguish systems whose SDCs are greater than 1 and are therefore less expressive for the class of PTMs than on-line sequential observers. A hierarchy of levels of successively more expressive sequential observers exists for PTMs such that observers that can distinguish among systems with SDC k+1 are more expressive than those that distinguish only among systems with SDC k. The idea that observers can improve their discriminating power by asking sequences of questions is very natural: interviewers use this technique to learn more about job applicants. PTMs capture the intuitive expressiveness of observers that ask sequences of questions and the associated ability to perform tasks like driving or process control that require sequences of interactive steps, while TMs do not.

TM behavior can be expressed by off-line observers that do not interact during a computation. TMs impose a strong constraint on the mode of observation by the requirement of a fixed initial state. PTMs allow sequences of observations such that the working tape contents depends on the history of previous observations. Systems that allow multiple concurrent observers, asynchronous interaction, and noninstantaneous events with duration further increase expressiveness.

Techniques for refining observation equivalence due to inner interaction during algorithm execution complement refinement techniques arising from outer interaction of systems with external observers. The goal of distinguishing among different ways of implementing a function is less ambitious than that of capturing the interactive behavior of patterns of algorithms. The ability to look inside black boxes refines functional equivalence classes to distinguish among ways of performing the same algorithmic task.

We examine refinement of equivalence classes for finite-state grammars and automata. The two grammars “G1: S -> aX, X -> b | c” and “G2: S -> aX | aY, X -> b, Y -> c” determine the same off-line language L(G1) = L(G2) = {ab, ac}. G1 and G2 are equivalent when observed by off-line observers that observe the set of generated strings. However, when observed by an on-line observer, G1 allows choice between b and c after generating a, while G2 is committed to either b or c after generating a. This distinction is expressed by a.(b+c) versus a.b+a.c, where . is generative concatenation and + is nondeterministic choice.

On-line observers of grammars have greater expressive power because they can observe that G2 has fewer choices than G1 after generating a, while off-line observers cannot distinguish between G1 and G2 because the off-line strings they generate are equal. Preserving freedom of choice is important in two-person games and in warfare, where premature commitment by one side can be exploited by the other.

A strong form of structural equivalence called generative equivalence can be defined for FSGs whose productions are of the form X -> aY (transition from state X to state Y on generating (reading) a), or X -> a (transition to a final state on generating (reading) a).

Definition 7 (generative equivalence): Two FSGs G1 and G2 are generatively equivalent if there is a one-to-one correspondence between their nonterminals such that the language generated by each nonterminal of G1 is the same as the language generated by the corresponding nonterminal of G2.

Generative equivalence is finer than language equivalence but coarser than tree equivalence (with renamed nonterminals): X -> b|b and X -> b are generatively equivalent though the generating trees are different. The generative behavior of an FSG can be modeled as the set of all paths of a directed labeled graph from a designated initial (root) node to a designated final (sink) node. Labeled graphs derived from an FSG can be interpreted as FSAs whose states correspond to nodes of the graph (nonterminals of the grammar) and whose state transitions correspond to edges of the graph (productions of the grammar). Generative equivalence for FSGs induces an equivalence on FSAs: generating steps for grammars become computation steps of automata and generative equivalence becomes computational equivalence.

Definition 8 (finite-state automaton): The FSA A(G) associated with a finite-state grammar G has states corresponding to the nonterminals of G and a final state F. The start symbol of A(G) corresponds to the start symbol of G. For each production X -> aY, A(G) has a transition from X to Y labeled with the input symbol a. For each production X -> a, A(G) has a transition from X to F labeled a.

Proposition 5 (automaton simulation): If two FSGs G1, G2 are generatively equivalent, then the FSAs A(G1), A(G2) can simulate each other's step-by-step behavior.

Proof: If G1, G2 are generatively equivalent, then every labeled path in A(G1) corresponds to a labeled path in A(G2) with equivalent labeled continuations.

Observation equivalences for inner interaction have been extensively studied in the context of process models. Milner [Mi] introduced a form of observation equivalence for processes called strong bisimulation that views two systems S1, S2 as equivalent if, at each intermediate observable state, the choices of S1 are the same as those of S2. Bisimulation was first introduced to model properties of system failure and divergence: bisimilar systems simulate each other's failures, divergence, and success. Later it was found that preservation of freedom of choice precisely expresses the semantics of two-person games.

Bisimulation is stronger than generative equivalence because systems may exhibit equivalence of choice when they are not step-by-step equivalent. Nondeterministic automata (ambiguous grammars) may be choice equivalent though not generatively equivalent. The grammars G3: S -> aX|aY|bZ, X -> c, Y -> c, Z -> c, and G4: S -> aX|bY|bZ, X -> c, Y -> c, Z -> c, are bisimilar but not generatively equivalent.

Nondeterminism restricts nonobservability since the target state associated with an input symbol cannot be observed. Relations based on equivalence of choice diverge from relations based on equivalence of computational steps when inner behavior cannot be completely observed. But for deterministic finite-state grammars and automata, where inputs uniquely determine state transitions, bisimulation corresponds to generative equivalence: there is a one-to-one correspondence between generating and choice steps.

The divergence between choice-based and step-based equivalence, already evident for nondeterminism, is even stronger for systems that contain silent transitions of the form X -> Y (in addition to X -> aY and X -> a). We call finite-state systems with silent transitions X -> Y extended finite-state systems. The three extended FSGs G5: S -> a, G6: S -> X -> a, and G7: S -> X|Y, X -> a, Y -> a are bisimilar though not generatively equivalent. However, G8: S -> X|Y, X -> a, Y -> b is not bisimilar to G9: S -> a|b, because the transition S -> X causes G8 to enter a state where only a can be chosen and G9 does not have such a state.

Whereas equivalence of deterministic finite-state processes when all actions are observable is a robust notion for which strong bisimulation corresponds to generative equivalence, equivalence of processes with inner silent actions is less robust and has spawned a variety of observation equivalence relations. Milner proposed a notion of observable choice equivalence called weak bisimilarity [Mi] such that choices associated with each observable a are equivalent. [VW] showed that weak bisimilarity did not capture choice associated with silent moves and introduced a finer equivalence called branching bisimulation that changed the rules of observation (engagement), so that silent transitions that cause new choices to become available are observable. Whereas weak bisimulation faithfully captures observation of nonsilent moves, branching equivalence more faithfully captures the unobservable inner branching structure of processes and is more directly relation to step-by-step simulation of computation. Weak equivalence that treats sequences of silent moves as atomic makes sense from the viewpoint of external observers but is an ad hoc form of equivalence from the viewpoint of observers of generating steps of grammars. Weak bisimulation cannot be captured by a simple grammar equivalence because the treatment of sequences of silent (X -> Y) moves as unobservable cannot be expressed by a simple grammar equivalence.

Larsen and Skou [LS] show that the set of choices of systems whose actions are governed by probability distributions are, under plausible assumptions, indirectly observable in that transition probabilities can be determined with arbitrarily high probability. Nonzero probabilities are required to be above a certain threshold and testing multiple copies is permitted. Though explicit enumeration of the nondeterministic alternatives of hidden state transitions (testing of weather conditions) is precluded, the set of alternatives and their probability of occurrence can be determined by multiple runs of copies of finite-state structures.

The analysis of modes of observation for process models has been an active research area, especially in Europe, but has not been viewed as mainstream by the algorithms community. Our analysis suggests that the study of modes of observation expressed by observation equivalence will be a mainstream research area of empirical computer science. The effect of information hiding on observability is subtle and deserves further study. The distinction between nondeterministic transitions X -> aY, X -> aZ that associate nonunique actions with a and hide the next state, and silent moves X -> Y that hide the action itself, arises in other contexts such as Markov processes.

Better understanding of on-line modes of observation is central to empirical computer science, providing a starting point for a comprehensive model of interactive expressiveness that extends to models of physical expressiveness. We expect the study of observation equivalence relations associated with on-line observation to play a central role in future research on interactive computing.

Part II: Physics and Computation

Observation equivalence provides a tool for measuring the expressiveness of physical and computational systems whose role in interactive modeling is comparable to that of computable functions for algorithms. Section 5 examines physical models of observation, section 6 proposes the hidden-interface model as an alternative to the hidden-variable model, section 7 explores relations between interaction and observation, section 8 relates explanatory power of physical theories to computational expressiveness, and section 9 reviews variants of Church's thesis for quantum computing and interaction.

Physics deals with more elaborate observers and instruments of observation than computing for a fixed semantic domain (the real world R). Since all physical theories are about the same semantic domain, their expressiveness (explanatory power) can be directly compared. Relativistic and quantum observers have more fine-grained observation equivalence relations than Newtonian observers. Though a finest possible observation equivalence relation for physics is in principle conceivable, it cannot in practice be specified both because of incompleteness and because physics can always think of new actual and potential ways of observing physical systems that refine observation equivalence.

The question whether subjective nondeterminism of quantum observers can be causally explained by deterministic (realist) models has been a subject of debate since 1935, when Einstein proposed the hidden-variable model as a plausible explanation of subjective nondeterminism. Though the hidden-variable model is inconsistent with observation, the existence of realist models consistent with quantum theory has not been ruled out. We provide a framework for discussing such issues in terms of interactive models of computation and propose a realist model that potentially can explain subjective nondeterminism observed by a primary observer by the effect of secondary observations through hidden interfaces.

Nondeterminism due to inherent incompleteness of locally observed behavior is compatible with objectively deterministic physical laws. Interactive realism permits physical laws to appear nondeterministic because individual observers cannot inherently have complete information about external influences acting on the objects they observe, though the laws are objectively deterministic to metaobservers (God) to whom complete information is available. According to interactive realism, God does not play dice, but observers see a universe in which God appears to play dice because they cannot perceive the universe the way that God does. Interactive realism can be viewed as an extension of Einstein's principle of relativity for observers that imposes inherent limitations on observability of individual observers due to the inherent openness of the systems they observe. The assumption that an observer and observed system form a closed system during the process of observation (section 7) does not hold because forces of gravity and electromagnetism cause subsystems in nature to be inherently open and interconnected.

5. Observation in Physics and Computation

Physical observers have more degrees of freedom than observers of computing systems: they can refine observation equivalence by instruments like telescopes, microscopes, infrared sensors, magnetic resonance devices, etc. Physical theories determine observation equivalence relations over the observations they specify and explain that are continually refined by new forms of observation. Though computational and physical distinguishability differ in their quality, modeling expressiveness by observational distinguishability applies equally to computational and physical systems.

Newtonian, relativity, and quantum theories can be distinguished by their models of observability. Newtonian models view the observer as an external recorder of behavior of a preexisting modeled world with absolute notions of space and time. Relativity theory views the observer as a part of the physical system being observed, distinguishing between observer-independent properties like causality that are invariant for all observers and observer-dependent properties like mass, length, and simultaneity that depend on the frame of reference of the observer. Relativity refers to the fact that properties of objects (like length) and relations among events (like simultaneity) have a meaning only relative to an observer, losing their absolute Newtonian meaning. Absolute notions of space and time are discarded but the space-time geometry introduced by Minkowski is viewed as a feature of reality. Relativity disengages from observer-independent absolute reality but retains a belief in an objective space-time reality whose curvature is determined by the objects it contains. Quantum theory takes the relativity of observers to its logical conclusion, viewing the assumption of an independently existing absolute reality as unnecessary (metaphysical).

Quantum theorists have moved away from the view that events are caused by an independently existing reality to the view that observed events do not require a causal explanation. According to [Per], “quantum theory is a set of rules for observing probabilities for the outcomes of tests that follow specified preparations”. Tests (observations) involve an irreversible transfer of knowledge from the observed system to the observer, increasing the entropy of the observed system and decreasing the entropy of the observer.

Newtonian physics allows observers to observe objective reality without changing it. The quantum-theoretical position that such one-way interaction is not possible can be viewed as a violation of Newton’s first law, which in this context becomes: “every action (observation) has an opposite interaction”. The notion of an absolute reality that may be observed without changing it, which underlies both Newtonian and relativistic physics, violates Newton’s first law in its model of the relation between observers and observed systems. Quantum theory may be viewed as an extension of physics to the observation of microsystems whose energy is comparable to acts of observation, so that Newton’s first law has measurable consequences on the observed system. However, conservation laws of classical systems are supplemented by conservation of information (entropy) in quantum systems. An observation causes the observer to gain information (decrease entropy) and the observed system to lose information (increase entropy).

Realism has proved very useful as an explanatory hypothesis: the assumption of the reality of space, time, mass, atoms, and electrons provides a comfortable framework for models of physics that should not be lightly discarded. Unobservable genes, chromosomes, and double helices provide a realist model for biology. Realists construct reality from observed behavior by methods like abduction and Occam’s razor whose deductive legitimacy is even more questionable than induction. Bohr and the Copenhagen school accept induction but not abduction as a basis for interpreting quantum theory (see section 10).

Abduction is a process of selecting an explanation from a set of explanations consistent with an observation equivalence relation, possibly with the aid of additional criteria like Occam’s razor. Choice among a set of abductively equivalent explanations for a fixed set of observations is in principle metaphysical. But in practice, observation equivalence relations are refined to distinguish new theories from their predecessors by experiments that falsify previous explanations. Science is a process of developing theories that refine observation equivalence relations by falsifying abductive explanations, as described by Popper’s “logic of scientific discovery” [Po]. Counterexamples are observations inconsistent with predictions of abductive explanatory theories (see section 8).

If physics could define an absolute observation equivalence for which equivalent systems cannot be distinguished by any observation whatsoever, then equivalent systems would be truly indistinguishable and abduction that selects one system over another would be metaphysical. Quantum theory in fact claims to be a theory capturing such an ultimate observation equivalence relation, but it claims to explain only aggregate rather than individual observations. Whether there are any models of reality consistent with quantum theory is still an open question. The notion that no strongest observation equivalence relation exists is a possibility if there is no limit on the sensitivity of observation. Though physics appears to place an absolute limit on the sensitivity of measuring instruments, it is possible that the universe is structured like a Mandelbrot set, with an infinite regress of levels of structure and explanation.

The explanatory success of theories of the solar system or the atom does not justify the inference that heavenly bodies are “real”. The assumption that we can deduce (abduce) the existence of atoms, electrons, or other unobservable abstractions because they are the simplest explanations of phenomena has even less legitimacy than the assumption that general laws of observation can be induced from regular patterns of observation. In fact, particular models of reality explain phenomena only for limited experimental contexts. Newtonian reality of objects in space and time becomes invalid for large velocities and for microphenomena where the energy of the observed system is comparable to the energy of observation. Newton's first law causes acts of observation to have an equal and opposite reaction in the observed system.

Einstein believed in an independent reality that observations are designed to explain. The Einstein-Podolski-Rosen (EPR) Gedankenexperiment argues that measurement of a component of an entangled state violates either locality (spatially separated components of a state cannot influence each other instantaneously) or completeness (the wave function completely describes the state). Einstein found it inconceivable that locality could be violated and proposed a hidden-variable (incomplete) model of the state. Bell showed that the hidden-variable model had observable consequences at variance with the predictions of quantum theory. Experimental results showed that observed behavior supports quantum theory [Per].

These results discredited Einstein's belief in the existence of an independent deterministic reality (where God does not play dice), along with his reputation for clarity of thought. The view of Bohr and the Copenhagen school that quantum theoretic "reality" is inherently nondeterministic and that no plausible deterministic model of reality exists has been generally accepted. Strict Copenhagen quantum theorists do not accept abduction as a legitimate principle for deriving causal explanations of experimentally observed correlations, believing that even if a model conforming to observations could be found it would be metaphysical and therefore unacceptable. The principle of relativity of observers that Einstein himself introduced appears experimentally inconsistent with Einstein's observer-independent realism.

6. Interactive Realism

Bell's arguments against hidden-variable models [Per] involve a number of assumptions that rule out some but not all models of reality. D'Espagnat proposes a more general catch-all class of models of reality called veiled reality [Es] that has been criticized on the grounds that it is insufficiently specific. We propose a class of "interactive-reality" models, broader than Einstein's hidden-variable models but more specific than d'Espagnat's veiled-reality models, whose parameters may in principle be massaged so that hypotheses with testable consequences conforming with quantum predictions can be formulated.

Interactive realism generalizes the hidden-variable model so that a) values of hidden variables are complex numbers whose rules of combination satisfy "wave rules", as in the slit experiment, rather than particle rules, and b) hidden variables are subject to active modification by secondary observers through hidden interfaces that cause nondeterminism from the viewpoint of a primary observer.

By allowing values of interactive-reality models to combine according to wave rules, Bell's inequalities do not apply, and consistency with quantum observations can be reestablished. Nondeterministic collapse of the wave function is accounted for by hidden interfaces that capture the impossibility of separating regions of space into closed noninteractive subsystems (discrete closed observable subsystems are an inadequate approximation at the quantum level to a continuous interconnected space).

Interactive realism is an explanatory model of reality with complex-valued hidden variables obeying quantum rules of combination and multiple hidden interfaces whose behavior appears nondeterministic to a primary observer because it can be concurrently observed and modified by multiple secondary observers. Whereas hidden-variable models view objects as disconnected components that become connected only by an overt act of observation, hidden-interface models view objects as continuously connected to the environment and subject to modification even when there is no primary observation. Multiple interfaces that approximate to continuous interfaces between sybsystems and their environment are assumed adequate to model quantum behavior. Objects are subject to perturbation by secondary observers over which the primary observer has no control (Figure 1). The probabilistic collapse of the wave function observed in quan-

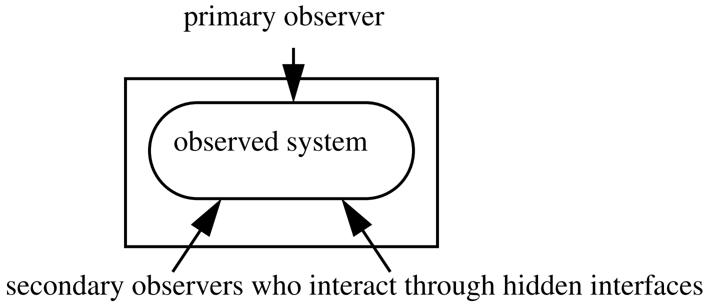

Figure 1: Primary Observer with Hidden Interfaces and Secondary Observers

tum theory could be explained by perturbations of secondary observers, such as random noise.

The view that objects have hidden interfaces through which they interact with their environment even when not being observed may be contrasted with the quantum-theory view that unobserved objects evolve according to a deterministic wave function that collapses nondeterministically when observed, and to traditional realism, which asserts that the world consists of a collection of disconnected objects that may be observed without disturbing them. Interactive realism specializes d'Espagnat's notions of open realism and veiled reality [Es]. The quantum rules of combination of hidden variables and the modification of variables through hidden interfaces impose constraints on veiled reality that can be further parameterized.

Interactive realism is compatible with Einstein's belief that God does not play dice in the sense that an observer who could completely observe all local interactions of an object could construct a local deterministic model. The problem of completely specifying all interactions with a region of space-time arises on a larger scale when the region considered is the universe as a whole. If the universe is globally open there is no globally deterministic model: a God who does not play dice requires the universe to be closed.

Just as computing systems can have a grand designer who controls all interfaces, physical observers can in principle invoke a locally omniscient designer who accounts for local behavior. A global God who does not play dice exists only if the universe is closed, but a local explanation of behavior is possible whether or not the universe is globally deterministic. Local behavior is determined by interactions of the local objects, but may be nondeterministic because the interactions that actually occur cannot be predicted.

Interactive realism adopts the position that incompletely perceivable objects exist and that unperceivable agents may modify them through hidden interfaces. Einstein would certainly have preferred the explanation of interactive realism over models with inherent, unexplained nondeterminism. In fact, interactive realism corroborates Einstein's hypothesis of incompleteness, though the model of incomplete reality is broadened from hidden variables to external secondary observers that interfere in uncontrollable ways with the observed system. Greater expressiveness of collaborative observers (writers) than single observers implies that hidden-interface models are physically more expressive than hidden-variable models.

Quantum theory extends physical models to the case when the assumption that systems are unaffected by observation breaks down and Newton's first law must be strictly applied. The distinction between passive and active observation is nicely captured by distinction between read-only observers of an object and observers who must write whenever they read. Newton's first law may be interpreted computationally as the requirement that observers must write whenever they read: every act of reading by an observer causes an associated act of writing in the observed system. Nondeterminism arises from the fact that multiple concurrent observers of an observed system interfere with one another, just as in computing systems with multiple concurrent writers. Moreover, the primary observer cannot control writing by secondary observers and must model writing by secondary observers nondeterministically. However, they can in principle predict what will happen for any given pattern of writing by secondary observers.

7. Interaction as Observation

Deterministic models of interaction in both physics and computation involve the coming together of an observer and an observed system in an act of observation during which they temporarily function as a closed system. Physicists distinguish reversible preparations - which may be elaborate like a supercollider or simple like a telescope - from irreversible acts of observation. Physical preparation requires not only apparatus for performing tests but also apparatus to produce an objective (intersubjective) record that may involve irreversible amplification of microscopic effects. An algorithm is a computational preparation that, when supplied with a prepared (typed) argument, causes a computation and output of a result.

The supply of an argument to an algorithm is an act of observation. Supplying an arithmetic expression with values of its free variables or supplying arguments for a lambda expression allows reduction of operator-operand combinations to a normal form that yields an observed value. The expression is an open system that is closed by interaction with an argument, causing execution and display of the result in consummating the act of observation. Execution is an irreversible process that consumes the expression with its argument and replaces (substitutes) the result for the closure created by the act of observation.

Algorithms are implemented by procedures, while acts of observation correspond to procedure calls that create a data structure called a closure (activation record), which is consumed and transformed into a result by execution of the procedure. By specifying algorithms as reusable procedure templates instantiated by a closure for each act of observation, observations can be repeated even though instances created by an act of observation are consumed. Physics does not have naturally occurring templates from which instances of physical objects can be created on demand. But repeatability can be achieved by preparations if there is a naturally occurring supply of equivalent objects.

When algorithms and arguments are brought together in an act of computation and observation, both the algorithm and the argument are consumed. This effect is masked by reusable algorithm templates that create activation records for each instance of execution, just as a TM is reused by reinitializing it for each computation. When objects are consumed by an act of observation, then the view that observers observe an independently observable reality is not tenable and the quantum-theory view that observations collapse the observation space must be adopted. In the absence of algorithm templates, observing an algorithm by supplying it with an argument would preclude observing it by supplying it with another argument. This situation obtains when we observe an arithmetic or lambda-calculus expression by supplying values for free variables and reducing the expression to normal form.

Client-server models specify processes of observation with servers as observed systems and clients as observers. Clients play the role of transmitters of observations while servers play the role of receivers that model systems being observed. Synchronous procedure calls treat client-server communication as a closed single act of computation and observation. Rendezvous models synchronous communication where the act of communication coalesces the observed system and the observer into a coupled system that is temporarily closed while the interaction occurs and then separates into two component open systems. Asynchronous communication allows the client to dispatch a message that participates in the act of observation on the client's behalf while allowing the client itself to remain uninvolved in the act of observation.

In both physics and computation, we study observers and observed systems separately in developing models of the act of observation. The observer provides a context or environment for the observed system that allows an interaction with an observed result to occur. Physical preparation can be abstractly interpreted as specification of a class of potential closures for performing certain tests in precisely the way in which algorithms specify a class of potential closures for computing certain results. Closure of a system specializes the system to a specific purpose that precludes its use for other purposes. Client-server systems, like preparations, specify a class of potential closures for interactive systems and should be distinguished from acts of observation (tests) that specialize systems to a specific behavior that precludes other behavior. Computation is viewed as a consequence of an act of observation that specializes a system to a specific behavior; conversely, observation can be viewed as a form of computational specialization.

We can view the observer as a client or transmitter and the observed system as a server or receiver of requests for service. Experimental physicists construct observation tools for observing specially prepared data. In computing, both the client and the server are constructed by the system builder. The designer can have symmetric knowledge of the inner structure of observers and observed systems, though clients can only observe a single interface. Physical observers do not have designer access to observed systems and must model them as servers with inaccessible inner structure and hidden interfaces.

IMs extend observability from single acts of observation to sequences and patterns composed of multiple acts of interrelated observation. Patterns of observation for open systems cannot in general be specified in a computable or RE way (patterns for speech and vision cannot generally be specified in closed form), but classes of interactive patterns among components of closed systems turn out to be computable.

The interaction patterns of on-line agents P that interact with environments E to form closed systems (P, E) can be specified as on-line algorithms [We3]. If we can extend the boundary of an interactive system P to create a closed system (P, E), the interactions of the subsystem become inner algorithmically specifiable computations of the extended system. To observe the pattern of interactions between P and E, we can record this pattern during the computation of the result and output it as a part of the result. On-line algorithms for finding a point on a line or a location in a Manhattan graph require the on-line agent with its environment to be closed [PY, We3].

Supplying an environment for an on-line algorithm is a special case of supplying an argument to an algorithm and may be compared to supplying an initial tape to a TM. Abstractly, we close an open system and observe its algorithmic behavior at an inner interface between two subsystems [We3]. The condition of atomicity (or serializability) for transactions can likewise be viewed as a closure condition that temporarily closes a system for the purpose of performing and observing a computation. Transactions ensure a deterministic result by excluding secondary observers during the process of computation. Such exclusion is possible for computing systems, in which the effect of secondary observers can be controlled by the system, but is not possible for physical systems in which the effect of secondary observers is uncontrollable. The behavior of physical objects between acts of observation can be modeled by that of transactions between acts of execution. But the assumption that physical objects can be treated as atomic between observations does not appear to be warranted. Whereas designers of computing systems can guarantee closure during the execution of atomic transactions, reality must be accepted as it is and cannot be redesigned to ensure atomicity during or between observations.

8. Explanatory Power and Computational Expressiveness

Physical theories specify rules for predicting the outcome of experiments that measure the observable behavior of a system S. A theory T' has greater explanatory power than a theory T for a system S if it correctly predicts a greater range of observed behavior. Explanatory power is formally captured by associating with T an observer Ot for observing the correspondence between predictions of T and observed behavior of S. Ot defines the class of observations of S that T is designed to explain. T is correct if its predicted behavior for S corresponds to observed behavior of S for observations of Ot. The explanatory power of T is specified by the expressiveness of Ot, measured by Ot's ability to make observational distinctions.

Counterexamples that falsify a theory T are distinguishability certificates that show nonequivalence (inadequacy) of T and observable phenomena. Ot must then be replaced by a more expressive observer Ot' with an associated theory T' that can account for a larger class of observations. The theory T is falsified if predictions of T can be distinguished from observations of S by the more sensitive observer Ot'. Nonequivalence is shown by a DC that specifies a counterexample. Ot' extends observable behavior to new kinds of observations not compatible with the predictions of T. A theory T' that correctly predicts the larger class of observations associated with Ot' is said to have greater explanatory power than a theory T that correctly predicts observations of Ot but is falsified by the more expressive observer Ot'.

The explanatory power of Newtonian, relativity, and quantum theory can be expressed by observers with distinct expressiveness. Let N, E, Q represent Newtonian, relativity, and quantum theory, let R represent the observed system (the real world), and let On, Oe, Oq represent Newtonian, relativistic, and quantum observers. On is less expressive than Oe and Oq, since Oe conforms to R for a greater range of observations than On and Oq conforms to R for a distinct greater range of observation. E has well-known counterexamples (DCs) like bending of light rays for N, and Q has counterexamples of relativistic predictions that do not obey quantum theory.

Let U be the universal theory that unifies relativity and quantum theory and Ou be an observer for U. U has the goal of explaining all observations of both E and Q, so that the observation equivalence Ou is the least upper bound of the observation equivalences Oe and Oq. The quest for U has so far proved elusive, but the specification of U in terms of properties of associated observation equivalence relations provides a clear specification of goals.

An observer Ot for a system S determines an equivalence class of theories (or systems) T equivalent to S modulo the observer Ot. Any T equivalent to S can be viewed as an explanation of S modulo Ot, and we may use criteria such as Occam's razor for selecting among theories T that explain S. Realist theories specify S in terms of unobservable independently existing entities that serve as causal explanations of S. For example, Newtonian physics specifies behavior by a realist theory in terms of mass, electrons, gravity, etc. When multiple theories explain observations of Ot equally well, the choice among them makes no observational difference and we call it metaphysical. However, being metaphysical is a notion relative to assumptions about the nature of observers. In fact, progress in science involves narrowing the scope of metaphysics by broadening the expressiveness of observers. Relativity theory and quantum theory both broadened the expressiveness of observation and correspondingly narrowed the scope of metaphysics. The claim that quantum observers Oq are the most sensitive possible observers corresponds to the claim that there is an absolute hard-core metaphysics, but this claim is not universally accepted.

Off-line observers of computing systems have the expressiveness of algorithms specified by computable functions. Equivalence classes of all programs (implementations) of an algorithm are not recursively enumerable. We generally select implementations with minimal (or at least acceptably low) computational complexity as representative elements of algorithmic equivalence classes. Denotational semantics purports to specify a realist abstract denotation for algorithms, but picking a representative implementation provides a more concrete realist model that is nonunique but useful. Observational expressiveness for process models and interaction machines is captured by on-line observers whose equivalence classes are not RE that likewise have realist models determined by implementations.

Though physical and computational models have common notions of expressiveness and equivalence, physical systems are specified by nature rather than constructed by man. Physical systems have a greater variety of modes of observation and models of inner reality than computing systems. The goals of modeling are also different: computer models aim to support the construction of efficient, reliable systems from given primitive structural elements, while physical models aim to explain unknown inner structure by progressive refinement of models of observation. Models of computation generally fix both the mode of observation and the model of inner structure prior to analysis, while physical models treat modes of observation as variable in order to refine them and treat inner structure as unknown out of necessity.

Whereas algorithms have a fixed mode of off-line observation, interactive systems have a variety of modes of on-line observation that correspond to modes of interactive physical observation. Though models of physics and computation differ in the domain-specific inner properties they assign to objects, they have similar models of the interactive relations among objects. There are many interactive properties that do not depend on whether the objects that interact are physical or computational. The notion of synchronous systems with a global notion of time, asynchronous systems with a local frame of reference, and on-line observers that change the state of observed systems is equally valid for both physical and computational systems and forms the basis for a common infrastructure for physics and interaction.

Viewing synchronous, asynchronous, and nonserializable behavior as computational analogs of physical, relativistic, and quantum-theoretic models brings out strong structural similarities between models of computation and physics and shows that computing systems share interactive attributes with physical systems. Mapping physical to computational systems is in this case even more interesting, since it brings out that deep results of physics are a consequence of computational properties of interaction.

Since computer science is a newer discipline than physics, it is natural to try to reduce computational expressiveness to the older notion of physical explanatory power. However, more insight is obtained by describing both expressiveness and explanatory power in terms of the more abstract concept of observational equivalence. By developing an abstract foundation for empirical computer science in terms of observation equivalence, we gain insight into the computational foundations of physical theories.

9. Quantum Computers and Church's Thesis

Church's thesis has the form "the intuitive notion X corresponds to a formally definable equivalence class Y of computing devices". In Church's original thesis, X is the intuitive notion of computability and Y is the equivalence class of devices and problems specifiable by a TM. Part I extended both X and Y, replacing TMs by the more powerful class of IMs, introducing the metric of observability as a basis for comparing the expressiveness of TMs and IMs, and showing TMs to be distinguishable by single observations while IM behavior required observers able to distinguish sequences and patterns of observations.

Complexity theorists narrow the scope of X and Y, restricting the intuitive notion of computing X to X' = "physically realizable computation" and replacing Y by Y' = "computable by TMs with polynomial space and time resources". The property Y' cannot be verified by observers who observe functional relations between inputs and outputs. However, if the notion of observation is extended from observing input-output relations to observing the variation in time required to compute outputs from inputs as problem size increases, then the difference between polynomial-time and exponential-time computation is observable, though the difference for memory and other nontemporal resources is not. Church's thesis for physically realizable computing is more restrictive both in its notion of computing and in its definition of acceptable computing devices.

Strong (Physical) Church's Thesis: Any physical computing device can be simulated by a TM in a number of steps polynomial in the resources used by the computing device.

The physical Church thesis appears pragmatically true for machines that obey the laws of classical physics, though harnessing turbulence or chaos could possibly extend classical computing power. However, Feynman [Fe] conjectured that quantum computers could not be simulated by TMs in polynomial time. There is strong evidence that quantum Turing machines (QTMs) [De, BV] are a counterexample to the physical Church thesis because they allow a larger class of functions for which there are no known polynomial algorithms (such as prime factorization) to be computed in polynomial time [Sch]. Their ability to update a superposition of configurations at each state transition can in principle be harnessed to perform an exponential amount of work in polynomial time. The idea that state transitions of natural devices can exhaustively explore an exponential space of possibilities in polynomial time is plausible, though only a polynomial subset of such spaces can be explored, and the trick is to organize the computation so that the space explored by the computational device corresponds to the space for the particular problem.

We make the stronger claim that IMs are more expressive than TMs according to the original Church thesis. IMs express interactive computations that are not computable by TMs, where expressiveness is defined in terms of the ability to make observational distinctions. Both Church's original thesis and the strong thesis restrict observable behavior of computing devices to be noninteractive (off-line). When observed behavior is broadened to include interactive behavior, then computing devices can no longer be simulated by TMs and a broader notion of computing machine or computing device is needed.

QTMs are defined in [BV] so they can be observed only once. They determine noninteractive nondeterministic machines more expressive than TMs in the sense of the physical Church thesis. Interactive QTMs (IQTMs) that model sequences of QTM observations would be more expressive than QTMs in the sense of traditional Church thesis, just as IMs are more expressive than TMs. The hidden-interface model, which can be implemented by IMs, can model QTMs and IQTMs without requiring nondeterminism or probabilistic collapse on observation as an explicit additional primitive.

States of a QTM are superpositions of states in a Hilbert space that evolve deterministically according to Schrodinger's wave equation provided they are not observed (do not interact). Observations cause the collapse of the wave function to a specific state with probability determined by the amplitude of the superposed quantum states. A complete definition of QTMs can be found in [De, BV], while a fuller discussion of the underlying physics is given in [Per].

The collapse of potential to actual behavior caused by observation already occurs in algorithms. Observation of an algorithm by supplying an argument causes collapse of the set of potential computations to a single computation and observation of an output. Collapse in both physics and computation is due to selection of what to observe and projection on sensors of the observer. For algorithms, selection is determined by the argument while projection is accomplished by computation. Observation of an algorithm usually returns a unique result, but if execution depends on additional inputs or interaction during computation then the output can be nondeterministic.

Observation of an object by supplying a message is a better analogy to physical observation than observing an algorithm. Nondeterminism arises because observed behavior may depend on hidden effects through interfaces not under the control of the observer. Observation of both physical and computing systems causes collapse of potential to actual behavior by selection and projection. Observation is deterministic for algorithms and objects that interact with only a single observer at a time, but is nondeterministic for primary observers of systems with multiple secondary observers. Collapse of potential to actual behavior is an entropy-reducing step, in contrast to entropy-preserving noninteractive behavior that obeys the second law of thermodynamics. Gaining knowledge in both physics and computation reduces entropy by collapsing potential behaviors to a narrower range or a unique value. The study of observation, measurement, reversible computation, and entropy of computation is discussed in [Fe] and more technically in [Per].

Part III: Philosophy of Empirical Computation

The evolution in computing from algorithms to interaction parallels that in physics from rationalism to empiricism. The relation between empiricism and realism is examined in section 10, Plato's cave metaphor is extended by in section 11 by allowing cave dwellers to interact with the external world, the Turing test is interactively extended in section 12, and extensions of logic to interactive models are explored in section 13. Extension principles for Platonic, Turing test, and logic models correspond to principles for modeling observers in physics and computation.

10. Rationalism, Empiricism, and Realism

The paradigm shift in computing from algorithms to interaction parallels the shift in physics from rationalism to empiricism in the 17th and 18th centuries (see Figure 2). Rationalism views reason as the chief

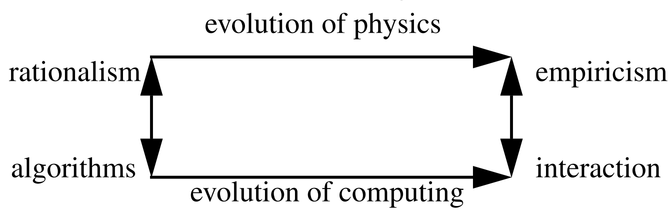

Figure 2: Parallel Evolution of Physics and Computing

source of knowledge and in its extreme form asserts that all knowledge can be derived by reasoning from necessarily true axioms by noninteractive rules of inference. Descartes' "cogito ergo sum" further equates thinking with reasoning, succinctly asserting that existence derives from thinking and by implication that facts about the world can be deduced entirely by reasoning.

Newtonian physics was paradoxically interpreted by rationalists as confirmation that physical reality could be rationally explained, but empiricists like Hume showed that neither inductive inference nor causality could be validated deductively. Kant's "Critique of Pure Reason" implies by its title that pure reasoning about necessarily true propositions cannot entirely express contingently true knowledge of the real world. However, rationalism continued to have great appeal in the 19th century in spite of the triumph of empiricism in science. Boole's "Laws of Thought" was rationalist in equating reasoning with thought. Hegel, whose dialectical logic extended reasoning beyond its legitimate domain, influenced political philosophers like Marx to develop rationalist social systems as well as mathematical thinkers like Russell to develop rationalist mathematical systems. Mathematical thought in the early 20th century was dominated by Russell's and Hilbert's rationalist attempts to reduce mathematics to logic.

Godel's incompleteness result showed the limitations of rationalist mathematics, but its implications for computer science have not yet been fully appreciated. Whereas mathematicians accept the implication that mathematics cannot be reduced to logic, many computer scientists continue to believe that the intuitive notion of computing reduces to the formal notion of Turing computability. Godel's result is strongly antireductionist, implying that neither mathematics nor computing can be reduced to noninteractive logic.

Absolute reality is a rationalist notion whose potential contradictions were already evident to the early Greek philosopher Parmenides who, even before Plato, believed reality was unchangeable, that empirical (sensory) perceptions were deceptive, and that the appearance of change was illusory because it contradicted the belief in an independently existing reality. The deeper analysis of quantum theory in section 5 above corroborates Parmenides' intuition by showing that an autonomous reality is not only an unnecessary (metaphysical) notion but is actually incompatible with empiricism.

Interaction in computing, like observation in science, distinguishes empirical from rationalist models. By showing that interactive behavior is not expressible by algorithms, we demonstrate that empiricism is not expressible by rationalism, validating arguments by Hume and Kant that contingent facts cannot be inferred by "pure reason". The irreducibility of interaction to algorithms overturns the established belief that algorithms express computing and bolsters an interdisciplinary antireductionist stance.

Noninteractiveness is an unstated assumption that underlies both Turing models of computation and rationalist models of the world. The Church-Turing thesis is a computational analog of the rationalist idea that reasoning from axioms can completely model experience and existence: both assert that intuitively definable autonomous domains can be completely expressed by noninteractive models. Both algorithms and reasoning express behavior by rules that noninteractively transform an initial state of affairs through a sequence of steps to a final state. Inner behavior for a single act of observation is an episode in the evolution over time of interactive systems. Algorithms express inner processes of noninteractive computation, while interactive models are grounded in external reality that is both more demanding and richer in behavior than the rule-based world of noninteractive algorithms.

Realism accords objects of knowledge a quality of existence. In its broadest sense, all theories and models we have considered are realist and differ only in the quality of reality. Realism, which cuts across rationalism and empiricism, has many different flavors:

Platonic realism: Platonic truths and ideal abstract objects are more real than imperfect actual objects

Cartesian realism: reality is thinking (cogito ergo sum) because that is the only certainty

Newtonian realism: independent reality of space, time, and physical objects

Relativistic realism: space-time has an independent reality determined by the objects it contains

Copenhagen realism: nondeterministic experimental data is the only reality, induction not abduction

Interactive realism: objects observed by primary observers have complex-valued hidden variables composed by quantum laws and hidden interfaces modifiable by secondary observers

Realism is associated with two principles of generalization: induction and abduction. Induction allows laws of observation to be inferred from observed regularities, while abduction is a stronger principle that allows causes to be inferred from correlations. Hume shows that neither principle can be justified by rationalism, focusing on the nondeductive (metarationalist) nature of induction. The belief that abduction can be nonempirical (metaphysical) was anticipated by Hume and its acceptance requires stronger realist assumptions than acceptance of induction. Newtonian, relativistic, and interactive realism accept both induction and abduction and may be called forms of strong realism, while Copenhagen realism accepts induction but not abduction, and Platonic and Cartesian rationalism accept neither induction nor abduction and are progressively weaker forms of realism (see figure 3).

	noninductive	inductive
nonabductive	Platonic Cartesian	Copenhagen
abductive		Newtonian Relativistic Interactive

weak (rationalist) realism = noninductive + nonabductive
empirical (Copenhagen) realism = inductive + nonabductive
strong (computational) realism = inductive + abductive
independent realism = reality exists independently of objects
= Newtonian + relativistic realism

Figure 3: Forms of Realism

Occam's razor further suggests that abductive explanations should be as simple as possible, while Einstein adds that they should not oversimplify. In retrospect, hidden variables appear to oversimplify, while interactive realism appears to satisfy the criterion of being as simple as possible but no simpler.

Interactive models of computation are empirical in modeling behavior by properties of observers and realist in explaining observed behavior by collections of interacting objects. Behavior in computing systems has an indisputably realist cause, since both inner object structure and interactive rules of engagement are implemented by "real" computing systems. Models of computation provide a framework for constructing causal explanations of physics, since physical theories are computational. This was not obvious as long as models of computation were noninteractive, and has caused problems for physicists like Penrose in their quest for computational models of reality (see section 12 below and [Pen]). The solar system and models of atomic physics are continuous models of computation, while hidden-variable and hidden-interface models are discrete interactive models of computation. If, following Einstein, we define realism by the requirement that observable behavior be accounted for (explained) by a deterministic model of computation, then strong realism appears to be the form of realism to which computational models should aspire.

11. Platonic Empiricism

Plato's cave metaphor, which compares humans to cave dwellers who can observe only shadows (projections) of reality on the walls of their cave and not actual objects in the outside world, is a model of relationships between observers and observed systems that has played an important historical role and is still relevant today. We extend Plato's cave metaphor to cave dwellers who interact with the external world by engaging in "shadow boxing", and show that such interactive observers have the power of IMs and express Platonic empiricism.

Empiricists accept Plato's argument that people are like cave dwellers who perceive incomplete projections P(W) of an external world W onto their sensory perceptors. However, they disagree with Plato's conclusions that empirical investigations are therefore not worthwhile and that ideal tables are more real and more interesting than imperfectly perceivable tables in the real world. Plato's belief that incompletely specifiable (imperfect) objects are not worthy of study, based on a faulty analysis of the notion of abstraction, contributed to a 2000-year delay in the emergence of empirical science. Empiricism repudiates the requirement of completeness, accepting that complete knowledge is unnecessary for empirical models of physics because they achieve their goals of prediction and control entirely through incomplete observable shadows (projections) that are observational abstractions of inherently unobservable reality.

Empirical computer science provides a concrete framework for studying observational abstractions in a context where "inner reality" can be specified and systematically varied. Computational abstractions provide insight into realist models of physics, where inner reality is unobservable and uncontrollable.

Plato understates the information that can be gained by shadows on the walls of a cave. Humans can obtain a variety of different images of their environment through multiple senses (vision, sound, touch). They can obtain multiple simultaneous spatial images through two eyes and two ears to obtain a stereo effect. Multiple temporal images can be obtained by sampling the environment at multiple points of time, creating stereo-spatio temporal images that provide richer cues for image reconstruction than geometric two-dimensional images on the walls of a Platonic cave. The richness of images created in this way has been recognized by “caves” in virtual reality.

We not only can select the images we wish to perceive, but can act to modify the external world and observe the effect of our actions. The process of observing how shadows that represent the state of the world change as a result of our actions can be thought of as “shadow boxing”: you can observe, predict, and control the way shadows change when you “punch” them. Shadow boxing may be interpreted as interaction, and occupants of Platonic caves who can shadow-box correspond to full-fledged empiricists.

The empiricist model of Plato’s cave requires traditional realism to be abandoned for the same reason as in quantum theory. Plato, following Parmenides, sensed that empiricism was incompatible with realism and viewed perceived physical objects like tables as imperfect approximations to an abstract reality. The rationalist realism of Plato and the interactive realism of quantum theory are consistent, while the noninteractive empirical realism of Newton and Einstein are inconsistent models. It is, however, fortunate for the progress of science that Newton and Einstein developed realist theories of science that provided remarkable accurate though ultimately inconsistent explanations of empirical phenomena.

Realism is a useful approximation in formulating empiricist theories when the effects of observation are negligible compared with the effects of being observed, just as Newtonian physics is a useful approximation when velocity does not approach the speed of light. Traditional realism breaks down as an explanatory hypothesis when observations can interfere with the system being observed. Einstein showed that a relative realism that abandons the reality of space and time but preserves the reality of space-time has greater explanatory power than Newton’s absolute realism, but quantum theory has shown that even relative realism is inconsistent with traditional empiricism.

12. The Interactive Turing Test

Logic is a rationalist abstraction of thinking that sacrifices interaction for the sake of completeness [We4]. It is too weak to model empirical (human) thinking for the same reasons that algorithms are too weak to model empirical (interactive) computation. Noninteractive theorem proving from axioms by rules of inference can model only limited application domains whose semantic form (ontogeny) recapitulates their syntactic form (phylogeny). We extend the Turing test to interactive thinking, examine interactive extensions of logic that violate its integrity, and extend logics with unique modeled worlds to interface systems with multiple pragmatics that express collaboration among multiple clients sharing system resources. The model for the interactive Turing test is related to the multiple-writers model of interactive realism.

The question “Can machines think?” has different answers for different notions of “machine” and “think”. The Turing test is a Gedankenexperiment that answers this question affirmatively if a machine responds indistinguishably from humans to questions over a broad range of topics. Not surprisingly, Turing assumed question-answering machines to be Turing machines and equated thinking with algorithm execution. The illustrative questions in [Tu2] about arithmetic, chess, and composing poetry do not require the machine to remember earlier questions in answering later ones.

The interactive Turing test [We2] allows question-answering machines to be IMs, including persistent Turing machines and more expressive multiple-interface (multiple-writers) machines of airline reservation systems and interactive realism. It extends the Turing test beyond PTMs so machines can interact with external resources like the Library of Congress, the Web, or human experts. Resources accessible to an IM are hidden from the questioner, who cannot distinguish between communication with IMs and TMs. The interactive Turing test changes the notion of machines and thinking without changing the form of the Turing test, and supports an interactive form of thinking that is closer to human than to algorithmic thinking.

The replacement of TMs by IMs that capture a more powerful intuitive notion of computing requires the relation between thinking and computation to be reexamined. Skeptics who believe that machines cannot think can be divided into *intensional skeptics,* like the philosopher Searle, who believe that passing the test does not constitute thinking because competence does not imply understanding, and *extensional skeptics,* like the physicist Penrose, who believe that machines have inherently weaker behavior than humans. Both Searle and Penrose equate machines with TMs.

Penrose argues that the nondeterminism of physics may be illusory because physical phenomena may be deterministic but noncomputable. He ascribes the nondeterministic action at a distance of the EPR experiment (section 5) to noncomputability of physics and suggests that a noncomputable model of physics could potentially resolve nondeterminism. We agree with Penrose that TM models of computation are too weak to model physics (for the same reasons that rationalism cannot express empiricism). However, IMs can express physical phenomena and physics is computable according to a broader notion of interactive computability. Penrose's arguments about the noncomputability of physics become inapplicable when the notion of computing is broadened to include interactive computation. In fact, IMs can be viewed as the noncomputable models that Penrose needs for his theory, since they can handle phenomena of physics that are not Turing-computable, such as turbulence and action at a distance. We resolve Penrose's dilemma concerning determinism and noncomputability by a natural extension of computability potentially consistent with both deterministic physical laws the observed nondeterminism of quantum theory.

IMs reduce physics to the status of a computable discipline without eliminating subjective nondeterminism of individual observers. But they explain subjective nondeterminism in terms of hidden interfaces that provide a globally deterministic explanation of observed local nondeterminism. Thus Penrose's hypothesis that noncomputable models are needed for deterministic explanation of physical and mental phenomena is replaced by the hypothesis that interactive models of computation can express subjective nondeterminism of observers by deterministic but locally nonobservable hidden interfaces. Whereas Penrose's hypothesis is not testable, we have high hopes that the hidden-interface hypothesis can be tested.

Penrose's dichotomy between computing and physics is based on a misconception of the nature of computing that is shared by Church and Turing and has its historical roots in the rationalism of Plato and Descartes. By identifying interaction as the key ingredient distinguishing empiricism from rationalism and showing that interaction machines express empirical computer science, we can show that the arguments of Penrose, along with those of Church, Turing, Descartes, and other rationalists, are rooted in a common fallacy concerning the role of noninteractive algorithms in modeling the real world. Penrose is a self-described Platonist and his attempt to reduce physics to a form of Platonic realism is understandable and ingenious. His instinct that Turing machines cannot model physics is correct, but his belief that noncomputable models are the key to a unified theory of physical and mental phenomena appears false.

Interactive thinking is a first-class form of thinking used by people for effective problem solving, just as interactive computing is a first-class form of computing. IQ tests that primarily test analytic (algorithmic) skills have been shown to be less effective than tests of interactive intelligence (emotional intelligence) as a predictor of success in life. Logic is a form of noninteractive thinking too weak to model interactive thinking, just as algorithms are a form of noninteractive computing too weak to model interactive computing. The expressive equivalence of algorithms and theorem proving in first-order logic is based on the equivalence of noninteractive reasoning and noninteractive computing.

The interactive Turing test can be further extended so that the system being tested not only asks questions of its environment but is also subject to environmental disturbances. In addition to a primary observer who aims to determine whether the machine can think, there are secondary observers hidden from the primary observer who can interfere in uncontrollable and arbitrary ways with the primary observer. This situation arises in computing when the primary observer wishes to make a reservation on an airline reservation system or when multiple concurrent writers wish to update a document or database through multiple interfaces. Observers of physical systems in nature cannot know whether observed systems can interact through hidden interfaces, so that observation of natural systems is better modeled by this strong version of the interactive Turing test than by a weaker model of observation. The assumption of secondary observers that disturb an observed system through hidden interfaces is stronger than the hidden-variable assumption of the EPR experiment, since it assumes not only hidden information but also hidden agents that can corrupt the hidden information. Nondeterministic behavior of observed systems can be explained by unobserved disturbances of the observed system between one observation and the next. The probabilistic behavior on collapse of the wave function in quantum theory can be explained by the cumulative effect of many disturbances. As previously indicated, it could be due to random disturbances caused by background noise.

The interactive Turing test provides a model for explaining quantum-theoretic nondeterminism. Nondeterminism occurs when systems permit observations through interfaces that a given observer cannot observe or control. Such behavior is computable by an IM with multiple interfaces that models the system being observed, but is not completely observable by any given observer. If an observer could observe the system at all, its interfaces it would be deterministic. Nondeterminism is due to the inability of a local observer to have global knowledge rather than, as Penrose conjectured, to noncomputability.

The causes of nondeterminism can be decribed in several equivalent ways. For example, nondeterministic systems have the property that the observed system coupled with the observer does not form a closed system. Nondetermimism arises because of interference among observations of shared resources. However, the view that nondeterminism is caused by failure of an observer to take into account all observations of the system being observed is the most intuitive way of accounting for observed nondeterminism.

13. Interactive Extensions of Logic

Logic is weaker than computation in the sense that every logical problem can be expressed as a computational problem while interactive computational problems cannot be expressed as logical problems. Whereas extending TMs to IMs is natural and intuitive, extending logical inference to interactive reasoning is fraught with contradictions because logic constrains rules of inference (computation rules) by semantic modeling requirements of soundness and completeness. Completeness restricts systems that can be modeled to those whose semantics is isomorphic to their syntax. Semantic domains whose properties cannot be syntactically represented by recursively generated theorems have necessarily incomplete models [We2].

First-order logic has nonlogical constant, function, and predicate symbols, axioms true for all interpretations of nonlogical symbols, and truth-preserving rules of inference. Proof theory deals with syntactic provability while model theory relates provability and truth. Soundness is the property that provability implies truth, while completeness is the more demanding property that truth implies provability (all true statements are provable). Model theory is realist in that syntactic formulae take their meaning in an independently defined semantic domain.

The number of interpretations of function and predicate symbols of first-order logic is nonenumerable. However, the number of theorems for a fixed interpretation is recursively enumerable. Nonenumerability of interpretations is related to nonenumerability of inputs of an interactive system: specific interpretations of constant, function, and predicate symbols have empirical content.

First-order logic in fact has little to say about truth in specific interpretations, dealing only with truth in all possible interpretations, which is itself a specific interpretation corresponding to the free algebra. For this particular interpretation, first-order logic is sound and complete: a formula is semantically true in all interpretations if and only if it can be syntactically proved as a theorem. Truth in all interpretations captures the domain-independent notion of valid reasoning but is rarely useful in reasoning about specific interpretations. When reasoning about specific systems or subject matter, it is not generally possible to preserve soundness and completeness.

Axioms that fix the interpretation of nonlogical symbols such as equality or modal and temporal operators are in some cases sound and complete. Modal logics can reason about logical notions like necessity and possibility, while temporal logics can reason about eventual occurrence of events, though not about real time. The number of properties formalized by modal and temporal logics is limited, while second-order logic, which provides a general framework for formalizing logical properties, is incomplete.

Logic programming support rules of inference with both a procedural and a logical (declarative) reading and can succinctly express certain kinds of computation as processes of logical reasoning. However, logic is an inflexible framework even for algorithmic computation and is too weak as a general framework for interactive computation. The incompatibility of completeness and reactiveness in logic programming follows from the fact that commitment that precludes backtracking (don't-care nondeterminism) realizes reactiveness by sacrificing completeness (We4).

The idea that logic is too weak to model computation was first presented at the closing session of the 10-year fifth-generation computing project in Tokyo in 1992, where the author argued that failure to realize the goal of computation by logic was not due to insufficient resources or ingenuity but to the inherent limitations of logic as a framework for interactive problem solving. Such limitations apply also to the goal of universal methods of proving program correctness. Logic and algorithms can formalize inner cleverness during the process of computation but not the complete process of computation.

Logic can in principle be extended to interaction by allowing nonlogical symbols to be interactively modified (reinterpreted) during the process of inference, for example by updating a database of facts during the execution of logic programs. However, interactive discovery of facts negates the monotonic property that true facts always remain true. Nonmonotonic logics [Re] permit reasoning about incompletely described modeled worlds by the closed-world assumption (cwa) that makes logic locally monotonic by assigning default values to every ground formula that is not a theorem. The cwa restores the ability to reason by a set of default assumptions that may be wrong. By freezing the interpretation and treating the system as closed and complete, reasoning conditional on the default assumptions becomes possible.

Temporary closure of open systems is an interdisciplinary technique for managing empirical systems during observation or interaction. Applying a quantum-theoretic preparation to a physical system closes open physical systems by binding their behavior to an observation tool. The assignment of default values to free variables of expressions is a paradigmatic method of closing open systems similar to assigning default values when invoking the cwa. Temporary closure of open systems is employed to impose atomicity or serializability on transactions. Calling a procedure with parameters or sending a message to an object is a form of closure (the data structure created when a procedure is invoked is actually called a closure). Taking the fixed point of a recursive function or taking the fixed point of a class in establishing its run-time context is a form of closure. On-line algorithms [PY] require closure by providing a context or environment to which the on-line algorithm will be applied (see section 7). Alternative forms of closure that arise in empirical computer science are the subject of a future paper.

Logic provides a framework for expressing the relation between syntactic representations and the semantic worlds that they model. Logic models L = (R, W) express the semantics of modeled worlds W by syntactic representations R only under very restricted assumptions. Because logics are closed systems, domains W have a fixed inductive structure that, for sound and complete models, mirrors the syntactic structure of the space of formulae. Logic can in principle be extended to evolving modeled worlds accessible to multiple observers by extending models to be of the form M = (R,W,P), where P is a collection of "pragmatic" components (P1, P2, ...) associated with clients and observers that can interact with M and modify it. Pragmatic components are open and must be closed by providing a context of use for purposes of interaction. The pragmatic components are associated with interfaces, for example interfaces of an airline reservation system, such that each client can autonomously query and transform the resources of the system. The extension of a logic L = (R, W) to systems M = (R, W, P), where W is a dynamically evolving structure accessible to multiple users, is clearly a radical extension of logic. It provides a paradigmatic model of collaborative sharing of resources that can be viewed as a scaled-up version of sharing of a state by multiple operations of an object. The interactive Turing test as well as interactive realism (Figure 1) may be modeled by extending logic to multiple pragmatic interfaces.

Multiple-interface models are illustrated by airline reservation systems with hundreds of interfaces under the control of travel agents, airline employees, and other classes of clients. Each client effects local closure of a subset of the resources of the system for purposes of interaction and/or observation, but the system as a whole remains open. Collaborative resource sharing appears to be more expressive than computation by a single user using observational distinction as a measure of expressiveness [We3]. It goes beyond analysis of observation of systems by a single user and is an important research area not only for computer science but also for physics.

The extension of logic to multiple pragmatics P specified by multiple interfaces parallels the extension of TMs to IMs. The pragmatics P expresses both the use of a model by external clients and observation by an agent of the external world. First-order logic is a special case M = (R, W0, P0) -> (R, W0) with a fixed modeled world W0 and a pragmatics P0 completely determined by W0. Fixing the modeled world and expressing the pragmatics in terms of the modeled world reduces interactive models to noninteractive closed systems and correspondingly reduces empirical to rationalist models.

14. Acknowledgements

Lecutres by David Mumford and discussions with John Myers helped the author in applying concepts of interative computing to quantum theory (part II). Dina Goldin played a major role in developing the PTM model and her comments on successive drafts of the manuscript helped to improve it.

15. References

[Ag1] Agha, *Actors: A Model of Concurrent Computation in Distributed Systems*, MIT Press, 1986.

[AWY] Gul Agha, Peter Wegner, and Akinori Yonezawa, *Research Directions in Concurrent Object-Oriented Programming*, MIT Press, 1993.

[BV] Ethan Bernstein and Umesh Vazirani, Quantum Complexity Theory, *SIAM Journal on Computing*, October 1997.

[De] D. Deutsch, Quantum Theory, the Church-Turing Principle, and the Universal Quantum Computer, *Proc. Roy. Soc.*, London Series A, 400, 1985.

[Es] Bernard d'Espagnat, *Veiled Reality*, Addison Wesley, 1994.

[Fe] Richard Feynman, *Lectures on Computation*, Addison Wesley, 1996.

[Gl] R. J. van Glabeek, What is Branching-Time Semantics and Why Use It?, rvg@cs.stanford.edu.

[GW] Dina Goldin and Peter Wegner, On the Expressive Power of Interactive Observers, work in progress.

[GW2] Dina Goldin and Peter Wegner, Persistent Turing Machines, work in progress.

[LS] Kim Larsen and Arne Skou, Bisimulation Through Probabilistic Testing, *Information and Computation*, 94, 1-28, 1991.

[MP] Zohar Manna and Amir Pnueli, *The Temporal Logic of Reactive and Concurrent Systems*, Springer Verlag, 1992.

[Mi1] Robin Milner, Elements of Interaction, *CACM*, January 1993.

[Mi2] Robin Milner, Operational and Algebraic Semantics of Concurrent Processes, *Handbook of Theoretical Computer Science*, J. van Leeuwen, editor, Elsevier, 1990.

[Pa] Christos Papadimitriou, *Computational Complexity*, Addison Wesley, 1993.

[Pen] Roger Penrose, *The Emperor's New Mind*, Oxford, 1989.

[Per] Asher Peres, *Quantum Theory: Concepts and Methods*, Kluwer, 1993.

[Po] Karl Popper, The Logic of Scientific Discovery,

[PY] Christos Papadimitriou and Mihalis Yannakakis, Shortest Paths Without a Map, *Theoretical Computer Science* 84-1, July 1991.

[Re] Raymond Reiter, A Logic for Default Reasoning, *Artificial Intelligence*, Vol 13 #1, January 1980.

[Sa] Vivay Saraswat, *Concurrent Constraint Programming*, MIT Press 1993.

[Sch] Peter Schor, Polynomial Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, *SIAM Journal of Computing*, October 1997.

[Tu1] Alan Turing, On Computable Numbers with an Application to the Entscheidungsproblem, *Proc. London Math Soc.*, 2:42, pp. 230-265, 1936.

[Tu2] Alan Turing, Computing Machinery and Intelligence, *Mind*, 1950.

[Tu3] Alan Turing, Systems of Logic Based on Ordinals, *Proc. London Math. Soc.,* 1939.

[VW] Rob Van Glabeek and Peter Weijland, Branching Time and Abstraction in Bisimulation Semantics, *JACM*, May 1995, 43, 3, 555-600.

[We1] Peter Wegner, Interaction as a Basis for Empirical Computer Science, *Comp. Surveys*, March 1995.

[We2] Peter Wegner, Why Interaction is More Powerful Than Algorithms, *CACM*, May 1997.

[We3] Peter Wegner, Interactive Foundations of Computing, *Theoretical Computer Science*, Feb. 1998.

[We4] Peter Wegner, Tradeoffs Between Reasoning and Modeling, in *Research Directions in Concurrent Object-Oriented Programming*, Ed. Agha, Wegner, Yonezawa, MIT Press, 1993.

[We5] Peter Wegner, A Research Agenda for Interactive Computing, work in progress.

[We6] Peter Wegner, Concepts and Paradigms of Object-Oriented Programming, ACM *OOPS Messenger*, August 1990.