DYNAMICAL COMPUTING, COMMUNICATION, DEVELOPMENT AND HIERARCHICAL INFERENCE

H. M. HUBEY

Montclair State University, Upper Montclair, New Jersey, 07043, USA
E-mail: hubeyh@mail.montclair.edu

P. B. IVANOV

International Science and Technology Center, 9 Luganskaya Street, P.O. Box 25, Moscow, 115516, Russia
E-mail: unism@narod.ru

A model of computing is suggested, combining the approach of analytical mechanics with the principles of a general psychological theory of activity. Thus reformulated, the traditional picture of computation allows generalizations of interest for distributed and parallel computing, artificial intelligence, or consciousness studies. The notion of hierarchical computing is discussed, stressing the communicative aspect; the directions of increasing the complexity of both computational universe and the computing agents are indicated. The idea of computability is reconsidered in the light of the new approach. The basic principles of hierarchical logic are presented as a tool for constructing generic formal systems.

1 Introduction

Using a computer, one has to arrive to useful results starting from some raw material. The principal question is that of computability. First computers were relatively simple, and the famous Gödel theorems reformulated for various formal systems [1–3] indicated the limits of primitive sequential computing. With the development of the Internet, the problem of computers talking to each other gained importance, and the rapid development of parallel computing and peer-to-peer technologies requires a different theoretical picture reflecting the present situation. The inherent insufficiency of the traditional logical systems in a complex environment has been demonstrated by Hubey [4].

In studying human behavior, computer analogies are still popular, which may hinder the inverse process, understanding computation as a primitive analog of consciousness. A general theory of activity that has been developed in Russian psychology since 1920s [5,6] could provide a solid framework for analysis of the communities of computers. The key principle of that theory, the sociality of development, perfectly reflects the practices of the World Wide Web and may serve as a source of ideas in designing efficient computer protocols approaching conscious communication.

Hierarchical structures and systems are necessary for efficient computation in a developing world [7]. However, the general principles of hierarchical organization are still poorly explicated in the literature, and the relation of hierarchy to development is far from being well understood.

In this paper, we present a summary of a hierarchical approach to computation. A general model of dynamical computing serves to translate the traditional static notions into a language more suitable for description of motion and development. Then we consider communicating computers and demonstrate how the opposition of the inner and outer world appears. We also present a formal scheme of a hierarchy, replacing the traditional idea of inference with purpose-directed construction.

2 Dynamics of computation

Traditionally, theories of computing were developed as formal models of an isolated computer operating in an essentially static world. Such an approach complies with the classical paradigm of mathematical study, but its application to real computation can only be limited, since the results of one computation serve to shape many other computation processes. That is why alternative pictures of computing may be useful.

2.1 The configuration space

Every computation occurs in some universe, so that successive operations would change the state of that universe. We admit that its distinct states can be somehow specified, and the collection of all the possible states forms what physicists usually call a *configuration space* **X**, which may be modeled with some mathematical structure (e.g. a finite set, a Euclidean space, a Hilbert space, a functional space, or a manifold). Points x of the configuration space **X** represent both the possible initial data and the possible outcomes of computation.

2.2 The agent

The agent is a device that can perform computation. According to A. N. Leontiev's theory [6,8], we distinguish the following levels of any agent's functioning.

2.2.1 Operations

An elementary operation changes the state of the universe, which is naturally represented as transition from one point of the configuration space to another. In every particular state (point x) there is a variety of admissible changes; by analogy to analytical mechanics [9,10], we will call it the *tangent space* to $\mathbf{X}$ in point x, $\mathbf{T}_x$; the union of all the $\mathbf{T}_x$ is called the tangent space to $\mathbf{X}$ and denoted with $\mathbf{T}$. Configuration space $\mathbf{X}$ together with all the tangent spaces $\mathbf{T}_x$ forms the *phase space* of the system, similar to a stratified manifold. Different agents are represented by different tangent spaces $\mathbf{T}$.

The points of $\mathbf{X}$ that can be connected with a single operation are considered as *adjacent*. With thus introduced notion of *relative* adjacency, points adjacent for one agent may be not adjacent for another. For instance, different processors may emulate each other's functionality on the microprogrammatic level.

2.2.2 Actions

In this model, a computation process is represented by a trajectory in the phase space. Formally, there is a mapping d: $\mathbf{X} \to \mathbf{T}$, so that every point x of $\mathbf{X}$ corresponds to a single element dx from $\mathbf{T}_x$. Given the initial and final states, x_i and x_f , one can choose an admissible trajectory to arrive from x_i to x_f ; the class of such trajectories is called an *action*. That is, contrary to operations that connect only adjacent points, actions link distant points via a sequence of operations.

Different agents may have different classes of admissible trajectories, and the same action may either be unavailable to some agents, or be implemented in different ways. The range of possible actions is intimately related to the nature of the agent, and it can usually be derived from a few fundamental principles. Thus, in classical mechanics, the principle of minimum action normally selects a single trajectory for any fixed x_i and x_f . The same holds for quantum mechanics, but the trajectory in a functional or operator space is considered instead of the common 3-dimesional space.

2.2.3 Activities

In simple configuration spaces, only operations and actions are possible. In a more complex case, the points of the space $\mathbf{X}$ form a number of classes $\mathbf{X}_k$; any trajectory connecting the points of the same classes $\mathbf{X}_1$ and $\mathbf{X}_2$ belongs to the same action class, which is called an *activity*. That is, an activity is like a higher-level operation, connecting adjacent classes; on the other hand, any activity is non-local, since it implies actions.

For an example of activity, one can consider an infinite trajectory in some configuration space $\mathbf{X}$: the points on the trajectory belong to the same class, and any action that can be represented by a finite segment of that trajectory connects that class to itself, hence belonging to the same activity.

Yet another important example: if the initial and final states are structured, any action transforming a component of the initial state to some component of the final state will be a representative of the same activity. Such partial actions (iterations) may fail to converge; however, such an activity often leads to quite acceptable results (e.g. using asymptotic series expansions in special function approximation).

2.3 The computable world

Every agent encounters certain initial (boundary) conditions and operates following its built-in logic. However, due to the limited operational capacity, the agent cannot achieve any point of the world at all. Some points are unachievable because they are not connected to the initial state by any admissible trajectory; some other points are only asymptotically approached; there may also be dynamical singularities that cannot lie on any admissible trajectory regardless of the initial conditions. That is, any single agent can only span a subspace of the full configuration space $\mathbf{X}$; this subspace is called *the world* $\mathbf{W}$ of that particular agent, in the given dynamical conditions.

A world is an analog of a dynamic flow (a bunch of trajectories) in analytical mechanics. In the simplest case, the world can be reduced to a single trajectory.

This individual world may have a structure quite different from the structure of the configuration space in general, actualizing only a part of the possibilities available. For instance, in a Euclidean configuration space, the individual world may form a sphere, a torus, or a fractal. Also, the worlds spanned by the same agent with different initial conditions may be quite different. The agent can never break out of its individual world unless there are other agents, and hence a hierarchy of agents operating in a hierarchical world.

3 Hierarchical computing

The very distinction of the levels of operation, action and activity is already introducing hierarchy in the model, implying a hierarchical organization of both the configuration space and the agent. In this section, we consider communication as the source of hierarchical development, which gives way to numerous implications of importance in distributed computing and artificial intelligence.

Let there be two agents A_1 and A_2 operating in the same computational universe. Since a point in the configuration space $\mathbf{X}$ is a distinct state of the universe, and since, in this model, any operation changes the state of the whole universe, the two agents cannot act simultaneously, save in the trivial cases, and sequential operation is the only possibility:

$$... \to x_1 \to A_1 \to x_2 \to A_2 \to x_3 \to A_1 \to x_4 \to \ ...$$

This means that, from the viewpoint of each agent, the state of the universe between successive operations or actions may "spontaneously" change, which is impossible in the traditional approach to computability. That is, the activity of another agent results in discontinuities of individual trajectories, up to switching to an entirely different class of trajectories (activity).

Similarly, assuming a universe developing according to some natural laws, we arrive to the necessity of accounting for the regularities of such development in the individual computation processes. However, in this work, we are mostly interested in agent-produces changes in the universe and do not consider naturally developing worlds.

Agents A_1 and A_2 exist in their individual worlds $\mathbf{W}_1$ and $\mathbf{W}_2$, in general, spanning different parts of the whole configuration space $\mathbf{X}$. This leads to a number of useful notions characterizing the possible relations between the worlds.

Non-intersecting worlds $\mathbf{W}_1$ and $\mathbf{W}_2$ imply that agents A_1 and A_2 cannot operate together; if one of them works, the other must be stopped.

An operation of agent A_1 is *A_2-compliant* iff the resulting state of the universe belongs to $\mathbf{W}_2$. Such operations do not change the activity of agent A_2 , rather influencing the timing of an action; alternatively, they could be called *boosts*.

An operation of agent A_1 is *A_2-compatible* iff it results in a point x that can lie on some trajectory of A_2 , maybe with different initial conditions. In other words, there are points of $\mathbf{X}$ adjacent to x in A_1 . Obviously, all A_2-compliant operations of A_1 are also A_2-compatible.

Existence of non-compliant operations means that the actual configuration space of an agent does not coincide with the whole $\mathbf{X}$ and hence is reducible to some subspace of $\mathbf{X}$. However, as it will be shown below, there are no such domain limitations in hierarchical agents.

Indeed, one can consider sequential operations performed by different agents as a higher-level operation performed by an agent A consisting of both A_1 and A_2:

$$... \to x_1 \to (A_1 \to x_2 \to A_2) \to x_3 \to ...$$

The intermediate state x_2 (point of $\mathbf{X}$) can be interpreted as *internal* for A, and the elementary operation of A transforming x_1 into x_3 is a composition of the operations of A_1 and A_2 ; the point x_2 of $\mathbf{X}$, beside being a specific state of the universe, *represents* a particular composition of operations. Points of $\mathbf{X}$ that serve as internal for some agent A (and hence mediate communication of lower-level agents) are called *products*. State s of the universe that is exclusively used to switch activity from one agent to another is called a *symbol*.

Alternatively, one can consider a hierarchy of operations. The original tangent space $\mathbf{T}$ now contains only *direct* operations, while there also are *indirect* operations mediated by other agents. In the above example, x_3 may be unachievable for A_1 with any direct operation, but it becomes achievable with a second-order operation involving another agent.

Hierarchical agents imply hierarchical worlds composed of many individual worlds, plus the points x achievable via collective actions. In the above scheme, the points x_i and agents A_i become interchangeable:

$$\ldots \to A_1 \to (x_1 \to A_2 \to x_2) \to A_3 \to \ldots$$

Like the points x may become internal for hierarchical agents, transformations of the world (operations and agents) can be considered as occurring in the interior of a higher-level point of the hierarchical configuration space. The difference between agents and the states of the computational universe hence becomes relative.

From the hierarchical viewpoint, one could consider any action as an operation of a higher-level agent arising from self-communication:

$$\ldots \to x_1 \to (A \to x_2 \to A) \to x_3 \to \ldots$$

The agent A thus becomes composed of two specialized components: one of them (the afferent component) transforming an outer state of the universe x_1 into an inner state of the agent x_2 , and the other (efferent) component producing an outer state x_3 from the inner state.

4 Hierarchical inference

So far, we considered hierarchical computing in a static universe, so that only its state could be changed. Beside the already mentioned natural development, this picture can be complicated by new objects produced by the agents. Once the states of the universe become represented by some other states (symbols), the operations on symbols may develop in a very complicated area. After all, agents do not stop on symbolic computation, and they eventually pass to material production, which enormously extends the configuration space and opens new directions of hierarchical development. Formally, this process could be modeled in a peculiar logic, containing the following rules of inference.

1. (Reflexivity) If there is an object O, there is a link → of this object to itself: $O \to O$
2. (Unfolding links) For any link → there is an object O' mediating it, so that → is equivalent to $\to O' \to$; the resulting links are different from the original and denoted with the same arrow merely for brevity.
3. (Folding links) The reverse of (2): any mediated link can be folded in a higher-level link.
4. (Abstraction) For any linked objects $O_1 \to O_2$, $\boxed{O}$ there is an object representing the link.
5. (Unfolding objects) Any object mediating a link $\to O' \to$ is a contracted form of a triad of input, inner state, output: $\to (S' \to C' \to R') \to$; this rule might be replaced with an equivalent: $\to O' \to$ implies $\to (S' \to R') \to$, and then $\to (S' \to C' \to R') \to$ by rule (2).
6. (Refoldability) $\to (O_1 \to O_2) \to$ is equivalent to $\to O_1) \to (O_2 \to$, with a proper re-interpretation of links.

The entity obeying these laws is called a hierarchy.

This set of rules is not minimal, and there may be many equivalent formal systems. Explicitly specifying the levels of hierarchy for both objects and links, one can construct rather complex structures, then fold them into simple schemes, and unfold in a different way. As one can easily see, in a hierarchy as a whole, the objects do not differ much from the links, while they will certainly be different in every particular hierarchical structure derived from that hierarchy.

Obviously, no hierarchy can be complete, since any element can be unfolded in a complex structure, producing additional elements and additional types of links. Hierarchical logic is a method of construction, rather than a description of a completed construction. However, a hierarchy possesses a kind of absolute integrity, since every element is related to each other, and the hierarchy can always be unfolded in a structure, in which these elements are connected with a direct link.

One could put forward the hypothesis that any static formal system, as known in modern mathematics, can be obtained via hierarchical development, as one of the possible unfoldings.

5 Conclusions

We have outlined an alternative approach to computation based on the hierarchical ideas. This approach conveniently links the traditional notions of analytical mechanics to the studies of human behavior within a general psychological theory of activity. Such a synthesis may be productive enough, to give birth to various non-standard theories of computation and inference, efficient methods of distributed and parallel computing, new forms of artificial intelligence. Even if not so, it presents one more possible conceptualization, which is not reducible to any known mathematical structure, rather being a tool for reconstructing any integrity at all.

References

1. Mendelson, E. *Introduction to mathematical logic* (Princeton, NJ: D. van Nostrand, 1964).
2. Cutland, N. *Computability: An introduction to recursive function theory* (Cambridge: Cambridge Univ., 1980)
3. Mesarovic M. D. and Takahara Y. *General Systems Theory: Mathematical Foundations* (N.Y.: Academic, 1975)
4. Hubey H. M. *The Diagonal Infinity* (Singapore: World Scientific, 1998)
5. L.Vygotsky, *Thought and language* (Cambridge, MA: MIT Press, 1986)
6. Leontiev A. N. *Activity, Consciousness and Personality* (Englewood Cliffs, NJ: Prentice Hall, 1978)
7. Efimov E. I. *Intellectual Problem Solvers* (Moscow: Nauka, 1982)
8. Ivanov P. B. A hierarchical theory of aesthetic perception: Scales in the visual arts *Leonardo Music Journal*, **5** (1995) pp. 49–55
9. Arnold V. I. *Mathematical Methods of Classical Mechanics* (Moscow: Nauka, 1979)
10. Dobronravov V. V. *Foundations of Analytical Mechanics* (Moscow: Vysshaya Shkola, 1976)

http://unism.pjwb.net/arc
http://unism.pjwb.org/arc
http://unism.narod.ru/arc