

Combinator-as-a-Process for Representing the Information Structure of Deep Computing.

Viacheslav Wolfengagen^{*1}, Larisa Ismailova¹, Sergey Kosikov² (jir.vew@gmail.com).

¹NRNU MEPhI, Moscow 115409, Russian Federation, ²NAO "JurInfoR, Moscow 119435, Russian Federation



Viacheslav Wolfengagen

Larisa Ismailova

Sergey Kosikov

Your
Logo
Here

ABSTRACT

In the present work, the main attention is paid to elucidating the technological features of computations with individual generators, or objects. Their interaction is considered in an applicative environment, which allows us to elucidate the internal structure of ordinary operations, the knowledge of which allows us to understand their properties. The choice of initial constant generators, considered initial and expressed by *combinators*, is discussed. These initial generators are used as the main "building blocks" that enter into the applicative medium in interaction with each other. As a result of the interaction, constructions arise that give representative sets of ordinary operators and immersed computing systems.

INTRODUCTION

Only with the use of applicative computing technologies, as known, it is possible to construct a general theory of computations that operates with an acceptable idea of the individual object in practice. This method was applied to the solution of semantic modeling problems [14], [16], [8], [10] with the formation of a dynamic computational model of the applicative type [15]. An important advance in the construction of a computational model that universally characterizes the interaction of information processes was made in [9], [7], where a layered discussion of the interaction is proposed. This approach in this paper was expressed in the equational construction of the theory of computation [6]. The applicative computational models and technologies obtained on its basis are important for the development of modern programming [11], especially in the context of growing interest in the practical use of the functional approach to programming [5], [17], [2].

METHODS

In this paper, an attempt is made to elucidate the technological foundations of "pure" computing, especially those of them that relate to calculations with objects. Their interaction is considered in an applicative environment, which allows us to elucidate the internal structure of ordinary operations, the knowledge of which allows us to understand their properties. The choice of initial constant entities, considered initial and called *combinators*, is discussed. These initial entities are used as the main "building blocks" that enter into the applicative environment in interaction with each other. As a result of the interaction, constructions arise that give representative sets of ordinary operators and immersed computing systems.

ENTER FINDING #2

Let us try to answer this question using the explanatory system of such computing, the only primitive operation of which is the application { applying } of one object to another. Objects enter into interaction with each other, and what happens during the implementation of their applications is shown from an intuitive point of view in the Fig. 1. The only thing that can be seen from this figure is the fact of applying, or, in another terminology, the em applying of the object a to the object b. In those applications that are currently reasonably understood, it is believed that a plays the role of function, and b plays argument, and the interaction of objects is represented by computation. Considering such an application as a operation, and even better { as a meta-operation, one can raise the question of the result of this operation. Following all the well-known traditions of operating with functions, we assume that a(b) is considered the value of applying the object a to the object b. For reasons of symmetry of objects a and b, in practice, a(b), (ab) and ab are used interchangeably by records of the result of their interaction existence in the environment. In the course of inter-object interaction, a contract is formed, and the original object in the general case is not contained in it, ceasing its explicit existence.

DISCUSSION

C

Interaction with the environment is carried out through parameters, which, in turn, are objects. The number of parameters is not limited, and their effect on the object appears as they are used in calculations. The process leading to the construction of the contract is called reduction and is symbolically denoted by red, and the process leading to the construction of the redex is expansion and is denoted by exp. Redex has the greatest computing potential, and a contract has the smallest potential. The contract contains a virtual representation of the original object, while the redex has its actual representation. The most important point is that the object is in the process of conversion, that is, in the bidirectional reduction-expansion process - the transition from redex state to contract state and vice versa. The reduction of the object is associated with the construction of the computational result, and the expansion is associated with the formation of an object that can lead to the existing computational result. For the feasibility of these transformations, which actually occur, we have to assume for simplicity that there is a fixed supply of primitive objects that can cause changes in the applicative environment. Ideally, in the pure theory of objects { combinatory logic { usually two such combinatory objects are taken { S and K (see [2]).

ENTER FINDING #1

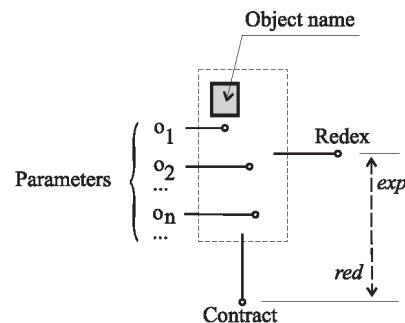


Fig. 1. Diagram of an object.

ENTER FINDING #3

Only with applicative forms it is not possible for many useful constructions be formed. Some objects interact with others, they must be exposed by their neighbors in applicative forms, and also act on other objects themselves. As a result of interaction, transformations of objects should occur, and these expected intuitive representations of the object are shown in the Fig. 1. The object is supplied with a unique name identifying it in the subject area. In fact, the object is considered as a pair

< redex, contract >.

A redex is considered to be a source object with a sequence of actual parameters used, and a contract is the result of an object interacting with its possible parameters. Redex represents the object before of interaction with the environment, while the contract represents the object after of interaction with the environment. Redex is the place where the original object is born, starting its apparent

CONCLUSIONS

1. Behind the variety of forms in which computing is now manifesting itself, it is not easy to see the unity of information entities and the few general mechanisms that drive the flows of digital information.
2. There is a clear tendency toward the restructuring of computing, the previous forms of which have generated more questions than they have answered. Starting with simple and directly perceived ideas about object diagrams, an extensible computing environment has been built.
3. Structurally, it reproduces itself by cascading component diagrams, and computer control is carried out by a finite set of initial combinatory objects. The "resolution" of the computing system also turns out to be adjustable, allowing not only to analyze the internal structure of the initial operations, which is also a lot, but also to synthesize objects with pre-planned computational behavior.

ACKNOWLEDGMENTS & REFERENCES

This research is supported in part by the Russian Foundation for Basic Research, RFBR grants 20-07-00149-a, 19-07-00326-a, 19-07-00420-a, 18-07-01082-a.