V. B. Borščev - M. V. Chomjakov

NEIGHBOURHOOD DESCRIPTION OF FORMAL LANGUAGES

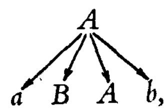
1. *Preamble.* There exists a huge amount of formal languages which are used for many different purposes. These languages differ both as regards their syntax (rules for forming propositions) and their semantics (the way of interpreting propositions). The authors' last two papers in this field were devoted to completely different topics. One of them (V. B. Borščev, M. V. Chomjakov, 1973), delivered at the last conference in Debrecen, concerns formal grammars. The other (V. B. Borščev, M. V. Chomjakov, 1972) deals with computable functions and relations. To the authors' surprise, it appeared that constructions arising in both papers had a lot of similar features. The aim of this report is to investigate these similarities. We hope that this discussion will help to throw light upon problems related to the semantics of formal languages.

2. *Grammars.* Nowadays a language in mathematical linguistics is regarded as a set of texts (for example, strings, syntactical structures, programmes, etc.), and a grammar as a way of describing this set. It proved to be convenient to represent any text as a finite model over a suitable signature, consisting of the names of relations that actually occur in the text (the dependence relation, the order from left to right, etc.). Vertices (points of the model's carrier) correspond to elements of the text (words, immediate constituents, etc.). Usually a grammar is a generative process, enumerating all proper texts of the language. We were tempted to determine the properness of a text by means of its intrinsic features. It turned out to be sufficient to consider only the immediate environment of every vertex of any proper text. All the types of such environments can be reduced to a small number of neighbourhoods. A neighbourhood is a small model in which a certain vertex – the centre – is marked. We say that a neighbourhood holds for a vertex of a text, if it can be mapped in the text by an isomorphism

*

in such a way that the centre is mapped on the vertex. We describe the environment of any vertex by pointing out which neighbourhoods must hold for it and which must not. Thus, any grammar is a logical formula over a finite number of neighbourhood. This formula has to hold for every vertex of any proper text.

For example, in the case of the context-free grammar, for every rule of the form $A \to aBAb$ there corresponds a neighbourhood



the whole grammar being the disjunction of all such neighbourhoods. Different kinds of grammars may be obtained by imposing restrictions on the shape of the neighbourhoods and formulas.

3. *Algorithms.* Algorithms is a method of defining functions of relations. Several equivalent forms are known (the recursive definition, Turing machines, Markov's algorithms and programming languages). In every case the function or relation in question is constructed on some carrier (the set of natural numbers, the set of strings on some alphabet, etc.), and one can use certain basic relations and functions (such as 0 and the successor function on the set of natural numbers, standard preludes in ALGOL-68, etc.) and construct auxiliary functions and relations if necessary. Thus emerges the model on the suitable carrier over a signature consisting of the name of our function and the names of the basic and auxiliary functions and relations. Unlike usual algorithmic systems which describe the process of the generation of such a model, we can attempt to characterise it by means of its intrinsic local features. For every pair of points $\langle x, f(x) \rangle$ (generally for every n-tuple for which $R(x_1, ..., x_n)$ holds once again it was sufficient to consider its immediate environment in the model. We introduced the neighbourhood again as any finite model over the same signature having a centre, a marked *n-tuple*. A neighboorhood holds for a certain n-tuple of the model, if there is a homomorphism from the neighbourhood to the model (the centre mapped on the n-tuple). The finite collection of neighbourhoods is called a scheme. A model is proper according to the scheme, if for every pair $\langle x, f(x) \rangle$ (every n-tuple $\langle x_1, ..., x_n \rangle$), where $f(x)$ or $R(x_1, ..., x_n)$ is the main or auxiliary function (relation),

there exists the scheme's suitable neighbourhood which holds for it. It was proved that with the help of the scheme it is possible to define every computable function, i.e. for every algorithm one can point out such a scheme that the model of this algorithm will be covered properly by the neighbourhoods of the scheme. Moreover, unlike algorithms, schemes may describe classes of models as well. The usage of schemes makes possible simpler descriptions of the semantics of programming languages.

4. It is easy to see that these two works possess a number of similar features. Let us try to formulate them in a more exact manner. In both cases the objects we deal with are models over some signature. On the other hand, the language for describing these models is introduced each time. Then we construct the correspondence between every proposition of the language and the models which satisfy it (the semantics of the language). The language we use is the set of neighbourhood formulas. Its semantics is the rule for covering any model by the formulas' neighbourhoods. The models that prove to have the proper covering are the models of the formula.

Some differences between these two works should be mentioned. In the first case, we deal with infinite sets of finite models – the texts and neighbourhoods were stated for vertices of the texts. However, for the algorithms the models are infinite and the neighbourhoods are stated for all the occurences of functions or relations (pairs, triples of points and so on).

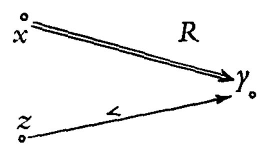
But the main point is that in both cases our description consists of 1) finding out certain finite collections of pieces of these models – neighbourhoods – and 2) the rule by means of which the models are covered by these neighbourhoods.

5. It seems for us that this method may be of use for investigating many formal languages. The following is a tentative attempt to apply this method to the predicate calculus of the first order.

Given some signature, let us consider all the models and propositions over it. Every formula signifies the set of models for which it is true. Let us try to determine the intrinsic features of the models belonging to this set. Consider our formula in prenex form, i.e. as $Qx_1\ Qx_2\ \ldots\ Qx_n\ F(x_1\ \ldots\ x_n)$, Q being $\forall$ or $\exists$, and F being quantifier-free. Let F be in disjunctive normal form ($F = F_1 v\ F_2 v \ldots v F_n$). Then for every F_i we construct a finite model – the neighbourhood. The carrier of this model is the set of variables occuring in F_i, while the relations are those that hold on the variables in F_i. The neighbourhood holds for an n-tuple of points in the model, if the relations in the model holding on these points are arranged in the same way as in the neighbourhood. The rule of covering a model by the neighbourhoods is rather complicated, dictated as it is by quantifiers of the formula. It can be explained on the example of the formula

$$\exists z \forall x \exists y \; [R(x, y) \; \& \; (y < z)]$$

which works for the set of natural numbers and defines the relation R (the relation $<$ being basic). Associate the neighbourhood



with the formula. The quantifiers require the covering of any proper model to satisfy the following: the point z should exist such that "turning" the neighbourhood around it the vertex x would cover all sets of natural numbers. Although the rule of covering does not appear to be too simple, we can note the same features again – the existence of neighbourhoods and the rule of covering.

6. *Conclusions.* As a rule a formal language is used for describing either large (infinite) objects or large (infinite) classes of objects. The description is stated as a proposition about the objects. The proposition itself is finite (as a matter of fact it is comparatively small enough to satisfy in a sense the law 7 ± 2). It is convenient to divide the meaning of the proposition into two parts. Firstly it contains a collection of neighbourhoods – the small pieces which may be included in a large object or a large class of objects. The second part of the proposition dictates the way of covering the models by neighbourhoods. We hope that this viewpoint may turn out to be useful for describing the semantics of formal languages.

REFERENCES

V. B. Borščev, M. V. Chomjakov, *Schemes for Functions and Relations*, Reports of the CMEA conference *Automatic Processing of Texts in Natural languages* (in Russian), Yerevan, 1972.

V. B. Borščev, M. V. Chomjakov, *An Axiomatic Approach to the Description of Formal Languages*, in *Mathematical Linguistics* (in Russian), Moscow, 1973.