Lingua-T — a language of a growing D-Theory

Andrzej Blikle
February 19th, 2026

Lingua-T is a quasi-programming language of a formalized theory of denotations — **D-Theory**. Originally, D-Theory was defined as a formalized theory of the denotations of **Lingua-V**. It is uniquely determined by a set of functional symbols of different sorts and a set of axioms that characterize these symbols. In **Lingua-T,** they are the names of the constructors of AlgDen-V. However, every "living theory" developed to be used rather than to be an object of investigation is usually growing by adding new functional symbols together with their expected properties. Although every enrichment of a theory is formally a new theory, we intuitively think of it as "the same theory but with new concepts". **D-Theory** is an example of such a living theory, since whenever we intend to write a program associated with some phenomenon, e.g., a growing market for fast-moving goods or bridge spans vibrating during an earthquake, we have to introduce new concepts, thereby enriching our theory. An example of such a situation is described in [1].

Whenever we introduce a new functional symbol into a language, we have to indicate its signature, i.e., the category of its values and the categories of its arguments. For instance, a signature of a constructor of conditional specinstructions has the following signature:

if-ins : ValExp x Sin x Sin ⟼ Sin

which means that in our language we can use instructions of the following shape:

if-ins(ValExp, Sin, Sin).

In practice, instead of writing signatures of constructors, we introduce appropriate clauses to the grammar of the language, which may also allow us to use infix notations rather than prefix notation. In our case, an infix-notation clause for conditional specinstructions looks as follows:

if ValExp **then** Sin **else** Sin **fi**.

In our ecosystem, we have yet another mechanism to introduce new functional symbols — the initial repository of temporary schemes (Sec. 11.2.2 of [2]). In this repository, our signature is represented by the following typed pattern:

(sin, **if** vex **then** sin1 **else** sin2 **fi**)

which allows for the generation of all grammatically correct conditional instructions.

Our investigation concludes that the introduction of a new functional symbol into **Lingua-T**, or, more precisely, a new grammatical structure, should consist in adding a new typed pattern to the initial set of typed patterns in the repository of temporary schemes. Note that in this way, we may add not only new patterns of existing sorts, but also new patterns of new sorts. Of course, whenever we add a new pattern, we should include some axioms for it.

References

[1] A. Blikle, *Programming in a Circular Arithmetics,* a manuscript available at https://moznainaczej.com.pl/spotkania-robocze-2026

[2] A. Blikle, P. Chrząstowski-Wachter, J. Jabłonowski, A. Tarlecki, A. Denotational Engineering of Programming Languages, a book in statu nascendi, available at https://moznainaczej.com.pl/what-has-been-done/the-book.