*A conjunction i*s the official part of speech that serves to connect independent units within a simple sentence, as well as proposals for communication among themselves. Communication can be expressed in equal units or master and slave units. *Conjunctions* can be simple in structure, component, or phrase, and may represent a form this is desemantised (losingmeaning) in other parts of speech (*providing, supposing*). *Conjunctions* transmit differentiated grammatical relations, and these relationships to some extent can be abstracted. Be aware that unions are auxiliary words, precisely because they are deprived of the nomination and therefore they cannot be members of a sentence. However, it should be recognized that some of the discharge unions replenished bydesemantised verbal nouns actually function as unions. Thus some ***desemantizatsiya*** obviously takes place due to the fact that new grammatical meaning has been purchased. For example: *the moment, the way.* At the level of a complex sentence, unions are used as subordinate and coordinative.It should be noted that there are allied words which are defined as pronouns and adverbs and that have retained their original values. These words have also developed the ability to enterthe sensitive predicative unit, for example: *which, who, what* (pronoun) and adverbs (*how, when, where, why.*) Distinction of Union speech and association is that, they do not losetheir meaning and therefore do not change the status of the sentence. They may take positions or additions. *I'll tell you why I am late.* Allied adverbs participate in compound simple sentences, and some of them (*thus, therefore, then)* connect parts of a complex sentence. They can occupy subject positionor additions, and they administer identification, and more subordinate members. This is shown in the examples above. Wehave analyzed morphemes borrowed from the Latin language, and we have traced the adaptation of Latin morphemes in the system of English.

REFERENCES:


1. Banshhikova, M.A. (2013). Ob anglojazychnom zaimstvovanii kak jazykovom fenomene (naprimere anglojazychnogo professional'nogo zhargona v nemeckom reklamnom mediadiskurse).//Filologicheskie nauki. Voprosy teorii I praktiki. – Tambov: Gramota ,№11(29) v 2-ch.Ch.1. – Pp. 27-30
2. Rozental'D. (1991). Je.Sovremennyj russkij jazyk. Uchebnoe posobie dlja studentov-filologov zaochnogo obuchenija / D.Je.Rozental', I.B.Golub, M.A.Telenkova. – M.:Vysshajashkola. – 559 p.
3. Steblin-Kamenskij, M.I. (1974). K voprosu o chastjah rechi. Spornoe jazykoznanie. – L., Pp.19-34
4. Smirnickij, A.I. (1957). Sintaksis anglijskogo jazyka M. – 284 p.
5. Jelektoronnyj resurs:http:// thefreelibrary.com /


## ESOTERIC PROGRAMMING LANGUAGES AS A STATE-OF-THE-ART SEMIOTIC TREND


***A.A. Zabenkov*, *D.A. Morel Morel***

Russia, Belgorod National Research University



The article gives an overview of esoteric programming languages features dwelling upon LOLCODE language. It is shown that such languages emergence is natural process in semiotic systems of the information-oriented society.

Everybody knows that information technologies are among the most dynamic and important spheres of modern knowledge and practice. But unsophisticated users scarcely meditate upon the contribution programming languages make to such dynamics and significance. Being an inconspicuous but indispensable keystone of all IT sphere programming languages – and methods, techniques, principles of programming consequently – are emerging nearly every year.

At the dawn of programming languages the binary machine code was the only means of the human-computer communication. The revolutionary moment was the appearance of a system coding machine commands with the help of special symbols. After that almost all improvements of the program code's syntax have been reduced to the achievement of the convenience and simplicity of handling it that allows programmers to abstract away from hardware peculiarities of their computers [Programming Language 2012].

"Study of approaches to semantics building allows to deduce an often ignored but very important feature of programming languages, namely the fact that the semantics of these languages must be analyzed with respect to both a computer and a human working on developing programs. Expressions recorded in any language must be conceptualized primarily by a person creating them. This feature has become noticeable at the moment when to run a computer a person has no longer needed detailed knowledge about organization of the hardware component of the device for which the program is intended. It was one of the primary tasks for which solutions "high level" languages were developed" [Kazakova 2008: 134].

A striking example of a programming language close to natural ones is SQL language, providing with the opportunity of interacting with databases. As an example: SELECT NAME FROM TABLE WHERE SURNAME=IVANOV [SQL Tutorial 2014]. But it cannot be called a full-fledged programming language, since it allows only to make queries to databases but not to create full-function programs.

If we review the whole history of programming languages we can see two extreme points that are machine code as the departure and esoteric languages – or esolangs for short – as a state-of-the-art trend.

While traditional programming languages developers try to make the code syntax maximally intelligible and easy-to-use, esolang creators usually set themselves another aim. Esolangs are conceived as a joke, to parody "genuine" programming languages or to reduce "serious" programming concepts to an absurdity. Some of them are intentionally limited while others are all-purpose, able to solve the same tasks as traditional programming languages. Thus, some esoteric languages could be compared to machine code – and low level languages in general – being unreadable due to their pure abstract-symbolic form when others could not be, being as close to informal "natural" speech as possible [Connors 2001].

The subject of the present study is LOLCODE, an esoteric language created by Adam Lindsay in 2007 and proved to be Turing complete (see: [Connors 2002]).

Considering the language in more detail, we can identify the following language constructions [Meza 2007; LOLCODE Specification 2007]:

- The program is bracketed with the keywords HAI and KTHXBYE.

- I HAS A <varname> – declaration of a variable (LOLCODE is dynamically typed, so there is no need to declare also the type of the variable).
- <varname> R <value> – assignment of a variable (there are the following types: NOOB (non-typified, before initialization), NUMBR (integer), NUMBAR (float), YARN (string), TROOF (Boolean)).
- Mathematical operations (work with types NUMBR and NUMBAR, conversion from YARN is possible if necessary): SUM OF (addition), DIFF OF (subtraction), PRODUKT OF (multiplication), QUOSHUNT OF (division), MOD OF (remainder), BIGGR OF (maximum), SMALLR OF (minimum).
- Boolean operations (work with TROOF type taking values WIN and FAIL): BOTH OF (AND), EITHER OF (OR), WON OF (XOR), NOT, ALL OF ... MKAY (AND for any number of arguments), ANY OF ... MKAY (OR for any number of arguments).
- Comparisons: BOTH SAEM (equality), DIFFRINT (inequality). At present other kinds of comparisons are not supported and must be constructed using BIGGR OF and SMALLR OF.
- VISIBLE – printing a string or a number.
- GIMMEH – reading a string from an input stream.
- Basic if-then-else structure is as follows:

<expression>
RLY?
YA RLY
<code block>
NO WAI
<code block>
OIC

- Case-statement structure is as follows:

<expression>
WTF?
OMG <value literal>
<code block>
[OMG <value literal>
<code block> …]
[OMGWTF
<code block>]
OIC

The comparison uses the implicit variable IT and a constant. OMG-block can contain any number of statements and ends with GTFO. If GTFO-statement is absent, the following comparison is executed after execution of this block.

- While-loop:

IM IN YR <label> <operation> YR <variable> [TIL|WILE <expression>]
<code block>
IM OUTTA YR <label>

The loop uses a temporary local variable and any unary operation for its change, including UPPIN and NERFIN (increment and decrement). The statement is used as the condition of loop completion: TIL ends when it becomes true (WIN), WILE ends when it becomes false (FAIL).

- Function declaration:

HOW DUZ I <function name> [YR <argument1> [AN YR <argument2>]]
<code block>
IF U SAY SO

The return from the function is possible in three ways. FOUND YR <expression> returns the value of an expression. GTFO returns NOOB. If the function reaches its end the value of the variable IT is returned.

- GTFO – similar to the break-statement in other languages; used to interrupt loops, cases and functions.

For illustration we can give a sample of LOLCODE program code:

```
HAI
CAN HAS STDIO?
PLZ OPEN FILE "LOLCATS.TXT"?
AWSUM THX
VISIBLE FILE
O NOES
VISIBLE "ERROR!"
KTHXBYE
```

As succeeding in opening the given file this simple program displays its content on the screen, otherwise it shows an error message. As we can see from this sample the code bears some resemblance to a certain informal conversation.

In conclusion, the following should be mentioned.

1. Emergence of esoteric programming languages is not an anecdotal evidence, it is within the paradigm of both natural and artificial languages dynamics. On the one hand, programming languages have been generally evolving towards convergence with *formal* registers of natural ones that endeavor to be maximally strict, perspicuous, and unambiguous. Some esolangs like LOLCODE have taken next step forward, their codes being as close to *informal* speech as possible. On the one hand, any developed national languages have sociolects maturing within them. Such sublanguages belonging to different subcultures are intended to be used within a distinct fold and not to be intelligible to the "profane". They can be developed out of conspiracy (thieves' cants case) or as a counterculture or language game manifestation (Internet slangs case, e.g. "olbansky" language, kitty pidgin). Esolangs just reproduce this trend in programming languages medium.

Thus, esoteric programming languages are natural semiotic phenomena fitting in with cultural peculiarities of the Computer Age society.

2. LOLCODE has a developed, full-functional syntax corresponding to the level of traditional general-purpose languages of the third generation at least. The language is based on elements of the English-speaking Internet slang, so people familiar with this slang can understand the code without any knowledge of programming language syntax [Dash 2007]. In fact, if we drew a parallel between

programming languages and natural ones, traditional program codes could correspond to the formal speech, the scientific register, whereas LOLCODE would be analogous to the informal speech, an Internet patois, being an IT-incarnation of the latter (see: [Liberman 2007]).

REFERENCES:


1. Connors, B. (2001). Esoteric Topics in Computer Programming [Electronic Resource] // Cat's Eye Technologies. – URL: http://web.archive.org/web/20020609152409/www.catseye.mb.ca/esoteric/index.html.
2. Connors, B. (2002). The Turing Tarpit [Electronic Resource]. – URL: http://web.archive.org/web/20020808015910/http://www.geocities.com/ResearchTriangle/Station/2266/tarpit/tarpit.html.
3. Dash, A. (2007) .Cats Can Has Grammar [Electronic Resource]. – URL: http://dashes.com/anil/2007/04/cats-can-has-gr.html.
4. Kazakova, A.E. (2008) Metodologicheskie osnovaniya razvitiya yazykov programmirovaniya: PhD in Philosophy Thesis. – M.: MGU. – 146 p.
5. Liberman, M. (2007). Kitty Pidgin and asymmetrical tail-wags [Electronic Resource] // Language Log. – URL: http://itre.cis.upenn.edu/~myl/languagelog/ archives/004442.html.
6. LOLCODE Specification 1.2 (2007) [Electronic Resource]. – URL: http://lolcode.org/1.2_spec.html.
7. Meza, J.J. (2007). Learning the LOLCODE Programming Language [Electronic Resource]. – URL: http://lolcode.org/language.html#title.
8. Programming Language (2012). The Columbia Electronic Encyclopedia, 6th ed., Columbia University Press. [Electronic Resource] // – URL: http://www.factmonster.com/encyclopedia/science/programming-language.html.
9. SQL Tutorial (2014) [Electronic Resource]. – URL: http://www.w3schools.com/sql/.


## DIE VERWENDUNG VON EINJÄHRIGEN PFLANZEN FÜR DIE HERSTELLUNG VON NANOKRISTALLINE CELLULOSE


***J.N. Zencova, L.N. Miroschnitschenko***

Russia, Belgorod State National Research University


Das Ziel dieses Artikels ist ein Überblick über die Möglichkeiten der Gewinnung des Zellstoffes aus verschiedenen Rohstoffen sowie die Qualität der Produkte der Verarbeitung von Cellulose.

Die Ausarbeitung und die Herstellung der ökologisch unschädlichen Produkte ist zurzeit eine wichtige Aufgabe der großen Lebensmittel-Unternehmen. Man projektiert die kostengünstigsten Technologien der Herstellung von biologisch abbaubaren Materialien, Nano-Cellulose-Beschichtungen des ökologisch antioxidativen Papiers.

Die hohe Festigkeit und gute Erneuerbarkeit der Cellulose-Nanofasernerweckt großes Interesse an ihre Herstellung und ihren Einsatz in den Papier- und Kompositwerkstoffen. Die Anwendung der Cellulose-Nanofasern zur Verstärkung der Festigkeit von verschiedenen Polymeren ist ein relativ neues Forschungsgebiet. Die industrielle Produktion der Nano-Cellulose wird bis heute von keinem Werk realisiert.