

Lexical-Functional Correspondences and Their Use in the System of Machine Translation ETAP-3

A.S. Andreyeva

Moscow State University, Philological Faculty,
Department of Theoretical and Applied Linguistics
Moscow, Vorobjevi Gori, 1st Building of the Humanities
andreyevs@mtu-net.ru

Abstract

ETAP-3 is a system of machine translation consisting of various types of rules and dictionaries. Those dictionaries, being created especially for NLP system, provide for every lexeme not only data about its characteristics as a separate item, but also different types of information about its syntactic and semantic links to other lexemes.

The paper shows how the information about certain types of semantic links between lexemes represented in the dictionaries can be used in a machine translation system. The paper deals with correspondences between lexical-functional constructions of different types in the Russian and the English languages.

Lexical-functional construction is a word-combination consisting of an argument of a lexical function and a value of this lexical function for this argument.

The paper describes the cases when a lexical functional construction in one of these languages corresponds to a lexical-functional construction in the other language, but lexical functions represented by these two constructions are different. The paper lists different types of correspondences and gives the reasons for their existence. It also shows how the information about these correspondences can be used to improve the work of the linguistic component of the machine translation system ETAP-3.

1 Introduction

The concept of lexical function (LF) was proposed in Igor Mel'čuk's "Meaning \Leftrightarrow Text Theory" (Mel'čuk, 1974; Mel'čuk & Zholkovsky, 1984; Mel'čuk et al., 1984, 1988, 1992) as the means of description of certain types of lexeme meaning correlations. "Lexical function f describes the dependence that determines for the certain word or word-combination such a multitude of words or word-combinations $\{Y_i\}=f(X)$, that for every X_1, X_2 the following statement is true: if $f(X_1)$ and $f(X_2)$ exist, then there is always the same semantic correlation between $f(X_1)$ and X_1 , on the one hand, and between $f(X_2)$ and X_2 , on the other hand". (Mel'čuk, 1974)

Soon lexical-functional description turned out to be of great value for the systems of natural language processing. Different ways the LF description can be used in NLP system are described in (Apresjan et al., 2003). As far as machine translation is concerned, lexical functions play an important role in it, being used, in particular, for providing translation equivalents.

The mechanism of their usage is the following: if in one language (L1) X_1 is an argument of the lexical function lf_1 , and $lf_1(X_1)=Y_1$, and X_1 has a translation equivalent X_2 in another language (L2), and X_2 is an argument of the same lexical function lf_1 , and $lf_1(X_2)=Y_2$, then if in the process of translation from L1 to L2 a word-combination " X_1+Y_1 " turns out to be a lexical-functional construction² representing lf_1 , X_1 is replaced with X_2 , and Y_1 is replaced with Y_2

© 2008. Licensed under the *Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported* license (<http://creativecommons.org/licenses/by-nc-sa/3.0/>). Some rights reserved.

² Lexical-functional construction, or lexical-functional word-combination, is a word-combination consisting of an argument of a lexical function and a value of given lexical function for the same argument.

irrespective of the fact what trivial translation equivalent³ it has.

- (1) IncepReal1 (bus) = take
IncepReal1 (avtobus) = sadit'sja na
(*avtobus* – bus)
(*sadit'sja na* – sit on, *brat'* - take)

To take a bus is translated as *sadit'sja na avtobus*, not *brat' avtobus*.

In the system of machine translation ETAP-3 information about LF links between the words is stored in the dictionaries⁴. If a lexeme is an argument of one or several lexical functions, the list of these LFs is written in the dictionary entry of this lexeme along with the values of these LFs for this argument. Thus, dictionary entry of the word *bus* includes the following fragment:

IncepReal1: take

Such a way of storage allows the information about LF links between words to be easily used in the process of translation.

The described above mechanism of usage of lexical functions in the process of translation is very useful, but it can be implemented only if X1 and X2 are arguments of the same lexical function, and Russian and English do not provide such a correspondence in 100% of cases.

If such a direct correspondence between two languages does not exist, information about lexical functions can still be used for providing proper translation equivalents. In many cases “X1+Y1” and its translation equivalent are both LF constructions but representing different lexical functions.

The goal of this paper is to describe different types of such correspondences, to explain the reasons of their existence, and to show the ways they can be used in a machine translation system.

2 Translation and false homonymy

The first type of lexical-functional correspondences I would like to mention is described in an earlier paper (Andreyeva, 2007).

That paper is devoted to homonymous word-combinations which are lexical-functional at least in one of its meanings. It describes different

types of homonymy, and one of them is so-called false homonymy.

This type of homonymy characterizes LF constructions which are not actually homonymous having only one meaning each, but every such construction can be described with the help of at least two lexical functions.

- (2) *to conclude an agreement*

As I note in (Andreyeva, 2007), the word-combination from (2) can be described with the help of two LFs (IncepOper1 and CausFunc0), but the way of description does not change the meaning: “to begin to have an agreement” (IncepOper1) or “to cause an agreement to take place” (CausFunc0) are two descriptions of the same situation, not descriptions of two different situations.

In (Andreyeva, 2007) I show that (2) is not unique, there exist quite big groups of non-homonymous word-combinations which can also be described with the help of the same pair of lexical functions. These are, for example, arguments of LF IncepOper1 with the value *begin* (*to begin an argument, a battle, a struggle* and so on).

There exists also a much larger group of words denoting different objects that can be created this or that way (*to grow plants, to write music* etc). In this case the word-combinations can be described with the help of both CausFunc0 and Oper1.

The work also shows one more pair of lexical functions describing the same constructions - FinOper1 and LiqueFunc0 (*to stop the battle*, for example).

In (Andreyeva, 2007) it is claimed that there are several reasons of the existence of false homonymy.

First, the descriptions of lexical functions are quite general and approximate. The creators of the system of LF did not have an aim to divide all the possible situations into non-crossing classes, the aim was to describe the main prototypical semantic correspondences.

Second, lexical functions were initially created for the description of situations. Their usage for the description of objects produced additional cases of false homonymy.

To sum it up, it is possible to list three pairs of lexical functions, and each pair can be used for the description of a non-homonymous word-combination:

³ Trivial translation equivalent of a word from L1 is its default translations equivalent in L2, or translation equivalent this word from L1 has as a separate word.

⁴ The system has a separate dictionary for every language.

№	English ↔	English
1.	IncepOper1 (X) ↔	CausFunc0 (X)
2.	Oper1 (X) ↔	CausFunc0 (X)
3.	FinOper1 (X) ↔	LiquFunc0 (X)

Table 1. False homonymy correspondences in the English language

All the examples in this section were given for the English language. The fact is that for the majority of their translation equivalents in Russian the situation is the same – being non-homonymous they can be described with the help of two lexical functions (the same ones as their English equivalents).

So, table 1 can be transformed into the following one:

№	L1 ↔	L2
4.	IncepOper1 (X) ↔	CausFunc0 (X)
5.	Oper1 (X) ↔	CausFunc0 (X)
6.	FinOper1 (X) ↔	LiquFunc0 (X)

Table 2. False homonymy correspondences⁵

So, there are a lot of word-combinations in both languages that have only one meaning but can be described with the help of two different lexical functions. This fact is interesting from the point of view of theoretical semantics, but it causes difficulties for a machine translation system.

For simplification of the situation in (Andreyeva, 2007) I propose to use only one of these lexical-functional descriptions in every case. It is a good decision for every word-combination in particular, but being implemented for the system in general it still causes difficulties.

The fact is that the number of these word-combinations is quite big. The ETAP-3 system is developed by many linguists and it is changed all the time. In the cases described in this section there is no or nearly no difference between the meanings of lexical functions in the pairs, so each of these LFs can be chosen for the description of a word-combination. It is impossible to guarantee that the same functions will be chosen to describe translation equivalents. Besides, there can be a slight difference between the meanings of equivalents in two languages, and different functions can seem preferable for

their description. If different lexical functions are used to describe translation equivalents, the information about their equivalency will be lost.

To avoid such a situation and to be free to choose the best corresponding lexical function for the description of an LF construction without taking into consideration the material of the other language it seems reasonable to add special rules to the ETAP-3 system. These rules allow to replace the value of a lexical function from the pair not only with the value of the same function but also with the value of the other LF from the same pair. The technique of this replacement is described in section 4.

3 Real lexical-functional correspondences

In the introduction there was given the general definition of lexical function given by Mel'čuk. According to it, to be regarded as an argument of the LF and a value of given LF for the same argument, two words must have a certain correlation between their meanings. But the majority of definitions of concrete lexical functions include not only semantic, but also syntactic conditions: to have a right to be called an argument of the LF and a value of given LF for the same argument, two words must also be connected by a certain syntactic link.

All the pairs of corresponding lexical functions in section 2 have differences in the semantic parts of their definitions, but the syntactic parts are absolutely identical. If they were not, it would be impossible to use these pairs for the description of the same constructions.

But it has already been mentioned that the reasons of adding these correspondences to the system of machine translation are mainly technical. What is really important for the system is the possibility to establish correspondences between different word-combinations, among which one can be described only with the help of lexical function lf1, and the other one represents only lexical function lf2.

Actually, such correspondences have already been described and implemented in the system of machine translation ETAP-3. See, for example, (Apresjan, Tsinman, 2002), where several dozens of such correspondences are listed, including quite rare ones. But the majority of these correspondences were used only in the paraphrasing block of the system, i.e. a block responsible for paraphrasing of sentences of one language only. As far as translation is concerned,

⁵ L1 and L2 can be both the Russian and the English languages.

only a few LF correspondences were implemented in the translation process.

This section shows which real LF correspondences can be found between the Russian and the English languages and how they can be used to improve translation process.

3.1 LF correspondences types

It happens quite often that in one of the described languages X1+Y1 form an LF construction (with X1 as an argument and Y1 as a value of the LF lf1), and the translation equivalent of X1+Y1 in the other language represents some other lexical function. Sometimes it is X2+Y2 (with X2 as a translation equivalent of X1 and Y2 as a value of LF lf2 for X2), but it also happens that it is Y2 only.

In some cases such correspondences represent system differences in strategies two languages use. In these cases the same correspondence describes large groups of word-combinations. In other cases the correspondences are not caused by system differences, but nevertheless can be used for the processing of big groups of constructions. There are also situations when LF correspondences are specific for small groups of constructions.

3.2 System differences

Func - Oper

This subsection describes not a pair, but a whole class of lexical functions dealing with the idea of possession (in its widest meaning, of course).

It is common knowledge that Russian belongs to so-called “be-languages”, and English is a “have-language”. These characteristics of the languages could not help influencing the sphere of lexical functions describing possession.

Of course, both Russian and English have Func and Oper LF constructions of different types. But in the Russian language the number of LF constructions of Func type is approximately two times bigger than in the English language. In the majority of cases, if a Russian LF construction of the Func type cannot be translated into English with the help of the same LF, an Oper correspondence can be found.

This principle can be illustrated by the following list of corresponding pairs of lexical functions:

№	Russian ↔	English
1.	Func1 (X) ↔	Oper1 (X)
2.	IncepFunc1 (X) ↔	IncepOper1 (X)
3.	FinFunc1 (X) ↔	FinOper1 (X)
4.	Func2 (X) ↔	Oper2 (X)

Table 3. Some correspondences of Func type and Oper type lexical functions

Here are some illustrations for LF correspondences from table 3.

- (3) Oper1 (boredom) = feel
Func1 (toska) = glodat'
(*toska* – boredom, *glodat'* – gnaw)
- (4) IncepOper1 (impression) = gain
IncepFunc1 (vpechtlenije) = skladivat'sja u
(*vpechtlenije* – impression, *skladivat'sja u* – form itself at)
- (5) FinOper1 (cold⁶) = shake off
FinFunc1 (nasmork) = prohodit' u
(*nasmork* – cold, *prohodit' u* – be over at)
- (6) Oper2 (threat) = bear
Func2 (ugroza) = navisat' nad
(*ugroza* – treat, *navisat' nad* – hang over)

Examples (3)-(6) show pairs of translation equivalents among which the Russian one is an argument of the Func type lexical function from the corresponding pair of LFs and is not an argument of the LF of Oper type; on the other hand, the English word is an argument of the Oper type LF and is not an argument of the Func type one.

In the process of translation word-combinations formed by these arguments of LFs and values of these LFs for these arguments are replaced with each other. Translation equivalents for the material from examples (3)-(6) are the following:

- (3a) Ego glojet toska. ↔ He is feeling boredom.
- (4a) U nego skladivajetsja vpechatlenije, chto...
↔ He is gaining an impression that...
- (5a) Nasmork u nego proshel. ↔ He shook off the cold.
- (6a) Nad nim navisla ugroza iskluchenija. ↔
He bore the threat of exclusion.

⁶ cold3 – a disease

Technical aspects of implementation of LF correspondences described in this section are given in section 4.

Nouns and gerunds

Another system difference between the English and the Russian languages that turns out to be important for the use of lexical functions in the translation process is the difference between forms Russian and English verbs have. In the English language there exists a verb form called gerund which has no analogues in Russian.

The way this difference influences the domain of LF constructions is the following. Among lexical functions a lot (more than a hundred) have nouns as their arguments and verbs (sometimes accompanied by prepositions or adverbs) as their values. For example, all LFs mentioned above belong to this group.

Many of these verbs form verbal nouns which in the majority of cases inherit the meaning of the verbs. Therefore, if a verb V is a value of lexical function lf_1 for the argument X ($lf_1(X) = V$), and V forms a verbal noun N_V , in the majority of cases there will be the same semantic correlation between X and V, on one hand, and X and N_V , on the other hand.

But syntactic links between a verb and a noun (X+V) and a noun and a noun (X+ N_V) are of course different, so word-combinations formed by two nouns cannot be described as representing the same lexical functions as word-combinations formed by a noun and a verb.

This problem was solved by making a special group of lexical functions for the description of word-combinations formed by a verbal noun and another noun. If $lf_1(X) = V$, and semantic correlation between the meanings of V and X and of N_V and X is the same, then $N_V = S0_lf_1(X)$. For example, see (7).

- (7) $IncepOper1$ (compromise) = arrive at
 $S0_IncepOper1$ (compromise) = arrival at

Lexical-functional constructions of $S0_lf$ type exist both in the Russian and the English languages, but their number is bigger in Russian. The reason for this seems to be the fact that the English language has gerund which can be used in a function of a noun. Russian has no verb form of this kind. So, it has to form verbal nouns which are actually formed quite freely. English, being able to use gerunds in many constructions, does not need such a big number of verbal nouns.

The result of this difference is that in many cases Russian LF word-combination representing a lexical function of $S0_lf$ type corresponds to the English construction with gerund. In order not to lose the information about lexical-functional correspondences it is possible to establish the following correlation:

$$S0_lf \text{ (in Russian)} \rightarrow lf \text{ (in English) (gerund)}$$

Such a correspondence can be established for all (or nearly all) the lexical functions of $S0_lf$ type. The article is too small to list them all⁷, so only several examples are given:

№	Russian	→	English
1.	$S0_Oper1(X)$	→	$Oper1(X)$ (gerund)
2.	$S0_IncepOper1(X)$	→	$IncepOper1(X)$ (gerund)
3.	$S0_FinOper1(X)$	→	$FinOper1(X)$ (gerund)

Table 4. Some correspondences of $S0_lf$ type and non $S0_lf$ type lexical functions

Here are some illustrations for LF correspondences from table 4:

- (8) $S0_Oper1$ (sport) = *zanatija*
 $Oper1$ (sport) = go in for
(sport – sport, zanatija – work)
- (9) $S0_IncepOper1$ (*soglashenije*) = *dostizenije*
 $IncepOper1$ (agreement) = arrive at
(soglashenije – agreement)
(dostizenije - reaching)
- (10) $S0_FinOper1$ (*biznes*) = *uhod iz*
 $FinOper1$ (business) = go out of
(biznes – business)
(uhod iz = going away from)

Translation equivalents for examples (8)-(10) are:

- (8a) *zanatija sportom* → going in for sports
(9a) *dostizenije soglashenija* → arriving at an agreement
(10a) *uhod iz biznesa* → going out of business

⁷ There are more than 100 LF of $S0_lf$ type in the ETAP-3 system.

3.3 Other differences

Lexical-functional correspondences described in the previous section are the result of system differences between the Russian and the English languages. This section is devoted to LF correspondences which are not caused by difference in strategies these languages use, but which are still applicable to large amount of word-combinations.

These LF correspondences consist of functions from Func group.

Lexical functions of Func type include, among others, Func0 and Func1. Func0 describes situations when X takes place, Func1 describes situations when X takes place for something/somebody or characterizes something/somebody (X is an argument of LF and a grammatical subject, something/somebody is a principal complement)⁸.

There are a lot of cases when a word in one of the languages (X1) is an argument of Func1, and its translation equivalent in the other language (X2) is an argument of Func0 and is not an argument of Func1.

In such cases the result of the translation would be much better if we replace the value of Func1 for X1 with the value of Func0 for X2 than if we replace it with the trivial translation equivalent of X1. So, there can be established the following lexical-functional correspondence: Func1→Func0.

It is important to note that, unlike all the other “real” LF correspondences described above, this correspondence works in both Russian-English and English-Russian translation.

Here is the example for the described LF correspondence:

- (11) Func0 (anger) = reign
 Func1 (gnev) = vladet'
 (gnev – anger, vladet' – possess)
 Im vladeet gnev. – Anger reigns.

There is one more reason for the establishment of Func1→Func0 LF correspondence. A lot of words are arguments of both Func0 and Func1. But in many cases information about one of these links is not yet included in the system by mistake, or by chance, or because of the lack of time. In this case Func1→Func0 correspondence works as a technical one, not being able to provide the best

⁸ These definitions of lexical functions were created by Ju.D. Apresjan (Apresjan, Tsinman, 2002).

translation result, but making it as good as possible.

Func1→Func0 is not the only lexical-functional correspondence of Func type. Other LF correspondences for lexical functions of Func group can be established. This is the list of them:

№	L1 →	L2
1.	IncepFunc1 (X) →	IncepFunc0 (X)
2.	FinFunc1 (X) →	FinFunc0 (X)
3.	CausFunc1 (X) →	CausFunc0 (X)
4.	LiquFunc1 (X) →	LiquFunc0 (X)

Table 5. Correspondences of Func type lexical functions

Here are some illustrations for LF correspondences from table 5:

- (12) IncepFunc0 (doubt) = arise
 IncepFunc1 (sommenije) = voznikat' u
 (sommenije – doubt, voznikat' u – appear at)
- (13) FinFunc0 (doubt) = disappear
 FinFunc1 (sommenije) = pokidat'
 (pokidat' – leave)
- (14) CausFunc0 (indignation) = arouse
 CausFunc1 (vozmuschenije) = vyzyvat' u
 (vozmuschenije – indignation)
 (vyzyvat' u – cause at)
- (15) LiquFunc0 (confidence) = shatter
 LiquFunc1 (doverije) = podryvat'
 (doverije – confidence)
 (podryvat' – undermine)

Translation equivalents for examples (12)-(15) are:

- (12a) U nego voznikajet somnenije. →
 Doubt arises.
- (13a) Somnenija pokidajut ego. →
 Doubts leave him.
- (14a) Eto vyzyvajet vozmuschenije u vseh. →
 This arouses everybody's indignation.
- (15a) Eto podryvajet doverije ludej. →
 It shatters people's confidence.

3.4 Rare correspondences

Despite the differences, all lexical-functional correspondences described above have one common feature: they take a word-combination

X1+lf1(X1) and transform it into a word-combination X2+lf2(X2).

But the situation is not always that simple. Let us look at the following examples:

- (16) carry conviction
- (17) privodit' v izumlenije⁹
- (18) prihodit' v izumlenije

All the examples (16)-(18) represent different lexical functions.

- (16a)CausFunc0 (conviction) = carry
- (17a)CausOper1 (izumlenije) = privodit'
- (18a)IncepOper1 (izumlenije) = prihodit'
(izumlenije – astonishment)
(prihodit' – come, privodit' – lead)

The fact is that none of the examples (16)-(18) can be translated with the help of any of LF correspondences described above in this article. These word-combinations are transformed in the process of translation into one verb. Fortunately this verb is a value of CausV0 for the translation equivalent of X1, so it is possible to establish the following LF correspondences:

№	L1	→	L2
1.	CausFunc0 (X) + X	→	CausV0 (X)
2.	CausOper1 (X) + X	→	CausV0 (X)
3.	IncepOper1 (X) + X	→	CausV0 (X) (passive voice)

Table 6. Some rare LF correspondences

Here are translation equivalents for table 6.

- (16b) to carry conviction → ubejdat'
(ubejdat' – convince)
- (17b) privodit' v izumlenije → to astonish
- (18b) prihodit' v izumlenije → to be astonished

4 Mechanism of translation

In sections 2 and 3 different types of lexical-functional correspondences were described. This chapter shows how the use of these LF correspondences is realised in the system of machine translation ETAP-3.

“Linguistically, ETAP-3 consists of various sets of rules and dictionaries... All the rules ... are subdivided into three main types: (i) general rules that apply to all the sentences in the course

of their processing; (ii) class-specific rules that hold for compact groups of words and are referred to by their names in the dictionary entries of the respective items; (iii) word-specific rules that are characteristic of individual lexical items and are stored directly in their dictionary entries. The second and third types of rules are activated only on condition that the processed sentence contains the relevant lexical items.” (Apresjan et al, 2003). As for general rules, it is important to note that they work one after another, in the fixed order, so the order they are listed in the system is very important.

To implement the above-described lexical-functional correlations we have to include them into the system in a form of translation rules. It means two main problems to be solved:

- 1) what type these rules must belong to,
- 2) if they are general, what their order must be.

As for the type, the decision seems to be the following: all the correspondences except ones from section 3.4 (rare correspondences) must become general rules, and those from 3.4 must become class-specific ones. The latter are very rare and can be implemented, perhaps, only for several words each. There is no use in making them general, and they must not be word-specific, too, because they describe groups of constructions, not singular cases. So, class-specific type is ideal for them.

All the other types of LF correspondences described above are worth being implemented with the help of general rules. First, they describe big groups of constructions. Second and the most important is the reason for their usage: we need them to work automatically in case the main rule of translation with the help of a lexical function (described in the introduction) does not work, and the only way to provide this is to make them general.

As these correspondences are implemented with the help of general rules, it is very important to put them into the proper order. I would like to propose the following one.

- 1) The first rule in the list of rules responsible for the translation with the help of LFs is of course the main rule described in introduction. All the other rules can work only in case the first one did not work. So, the first rule is:

$$lf1 (X) \rightarrow lf1 (X)$$

⁹ Russian examples (17) and (18) will be translated below.

- 2) The second block of rules is the block responsible for Func-Oper correspondences. In case the general rule does not work this block provides the most correct translation equivalents. So, the second block is:

№	Russian	↔	English
1.	Func1 (X)	↔	Oper1 (X)
2.	IncepFunc1 (X)	↔	IncepOper1 (X)
3.	FinFunc1 (X)	↔	FinOper1 (X)
4.	Func2 (X)	↔	Oper2 (X)

The order of rules inside this block (as well as inside all the other blocks) is not of great importance. It can be the same as in the table.

- 3) The third block is the one responsible for correspondences of Func type.

№	L1	→	L2
1.	Func1 (X)	→	Func0 (X)
2.	IncepFunc1 (X)	→	IncepFunc0 (X)
3.	FinFunc1 (X)	→	FinFunc0 (X)
4.	CausFunc1 (X)	→	CausFunc0 (X)
5.	LiquFunc1 (X)	→	LiquFunc0 (X)

It is very important for the third block to be implemented only after the second one, because there are lexical functions both blocks work with (Func1, IncepFunc1, and FincFunc1). If it is impossible to replace the value of one of these functions with the value of the same one, we must first try to replace it with its Oper equivalent and only in case it is impossible pass to Func0 correspondence. Oper equivalent is better than Func0 one because the former allows to preserve the information about all the actants of the verb – value of an LF, while the latter loses the information about one of the actants.

- 4) The fourth block is the one transforming nouns into gerunds.

№	Russian	→	English
1.	S0_Oper1 (X)	→	Oper1 (X) (gerund)
2.	S0_IncepOper1 (X)	→	IncepOper1 (X) (gerund)
3.	S0_FinOper1 (X)	→	FinOper1 (X) (gerund)

As for the order of the third and the fourth blocks, they do not interfere with each other so it is of no importance which one is the first. The

only problem is that in reality the fourth block is very big (it includes the majority of LFs of S0_If type), so it is just more convenient to have it after the third block.

- 5) And the last one is the block responsible for false homonymy correspondences.

This block was created “just in case”, so it is worth being placed at the end of the list. Besides, it works with Oper functions of different types, so in any case it must be placed after the second block.

This block causes one additional problem. If we list all its correspondences in one column, we will see that CausFunc0 can become both Oper1 and IncepOper1. Establishment of both of these rules in the system will not improve the translation but will produce plenty of wrong translation variants. As the rules of this block are not of great importance, it seems better not to use these two problem rules at all. So, this block will be the following:

№	L1	→	L2
1.	IncepOper1 (X)	→	CausFunc0 (X)
2.	Oper1 (X)	→	CausFunc0 (X)
3.	FinOper1 (X)	→	LiquFunc0 (X)
4.	LiquFunc0 (X)	→	FinOper1 (X)

In the majority of cases the transformation of one LF construction into the other one entails changes in syntactic roles of actants. Information about these changes is also included into the rules.

5 Conclusion

The paper described different types of lexical-functional correspondences between the Russian and the English languages. It showed how the information about LF links included in the dictionaries and translation rules of machine translation system ETAP-3 allowed to consider these correspondences in the process of translation and thus to improve its results.

References

- Andreyeva, A.S. 2007. Lexical Functions and Homonymy. *MTT-2007, Proceedings of the 3rd International Conference on Meaning-Text Theory*. Wiener Slawistischer Almanach, Sonderband 69. München – Wien

- Apresjan, Jury D., Igor M. Boguslavsky, Leonid L. Iomdin, Alexander V. Lazursky, Vladimir Z. Sannikov, Victor G. Sizov, and Leonid L. Tsinman. 2003. ETAP-3 Linguistic Processor: a Full-Fledged NLP Implementation of the MTT. *MTT 2003, First International Conference on Meaning – Text Theory*. Paris, Ecole Normale Supérieure, Paris, 279-288
- Apresjan, Jury D., and Leonid L. Tsinman. 2002. Formal'naja model' perifrazirovanija predlozhenij dlja sistem pererabotki tekstov na estestvennyx jazykax [A Formal Model of Sentence Paraphrasing for NLP Systems]. *Russkij jazyk v nauchnom osveshčenii*, No. 4, pp. 102-146
- Mel'čuk, Igor 1974. *Opyt teorii lingvističeskix modelej "Smysl ⇔ Tekst"* [A Theory of Meaning ⇔ Text Linguistic Models"]. Moscow, Nauka.
- Mel'čuk, Igor, Nadia Arbatchewsky-Jumarie, Lidija Iordanskaja, and Adèle Lessard. 1984. *Dictionnaire explicatif et combinatoire du français contemporain, Recherches lexico-sémantiques I*. Les Presses de l'Université de Montréal.
- Mel'čuk, Igor, Nadia Arbatchewsky-Jumarie, Louise Dagenais, Léo Elnitsky, Lidija Iordanskaja, Marie-Noëlle Lefebvre, and Suzanne Mantha. 1988. *Dictionnaire explicatif et combinatoire du français contemporain. Recherches lexico-sémantiques II*. Les Presses de l'Université de Montréal.
- Mel'čuk, Igor, Nadia Arbatchewsky-Jumarie, Lidija Iordanskaja, and Suzanne Mantha. 1992. *Dictionnaire explicatif et combinatoire du français contemporain. Recherches lexico-sémantiques III*. Les Presses de l'Université de Montréal.
- Mel'čuk, Igor, and Alexander Zholkovskij. 1984. *Tolkovo-kombinatornyj slovar' sovremennogo russkogo jazyka*. [An Explanatory Combinatorial Dictionary of the Contemporary Russian Language] Wiener Slawistischer Almanach, Sonderband 14.