Mathematical Problems of Computer Science 25, 2006, 27-38.

A Rule System for Translation of Universal Networking Language (UNL) Expressions into Armenian

Vahan Avetisyan

Institute for Informatics and Automation Problems of NAS of RA
e-mail va@ipia.sci.am

Abstract

In this paper a system of rules is described for automatic translation (deconversion) of Universal Networking Language (UNL) expressions into Armenian sentences. Some classification of rules is proposed for provision of proper coverage of Armenian grammar. Also some approaches to the representation of formalized grammar and morphology of the Armenian Language are described.

Introduction

The system of automatic translation described in this paper is a part of a project called "Development of Armenian Module of UNL". This project was started in 2003 in the Institute for Informatics and Automation Problems of National Academy of Sciences of RA. The general purpose of the project is the creation of an Armenian language module for automatic translation from Armenian to UNL (Enconversion) and from UNL to Armenian (Deconversion). The project includes, in particular, the creation of dictionaries of the Armenian language and Enconversion and Deconversion rules, as well as development of several supplementary software tools for automation of these processes. In this paper the main approaches to the development of deconversion rules are described. The linguistic properties of Armenian are used and some classification of rules is given. Also a system of rules created according to those classifications is described. This system of rules was created using the preceding works implemented during the realization of the project ([1],[2],[3],[6],[7]).

Some Characteristics of UNL

UNL is developed by a group of Japan scientists in the Institute of United Nations in Tokyo ([14],[15]). It was developed as an artificial formalized language for the representation of texts in a language independent format. The connection with natural languages is established by means of corresponding language modules ([1],[10],[11]).
UNL was created on the base of English language.
Various works are being implemented for automatic translation from UNL to different natural languages and from natural languages to UNL ([10],[11],[12],[13]).
The sentences in UNL are presented as hyper graphs consisting of concepts called Universal Words (UW) which play the role of nodes. The arches of these graphs describe binary relations among nodes, expressing the connectivity between words of the sentence. 45 relations are

defined to cover all types of relations among members of the sentence. Besides, a Knowledge Base is provided as a repository of global knowledge. It provides UWs (concepts) which are organized into a hierarchic structure. UWs are notions specific for UNL. They denote the words considered with some fixed meanings.
UW is a character-string usually made up of English words that can be split into two different parts: a headword and a constraint list. The headword can be a word, an expression, a phrase or even an entire sentence. It should be interpreted as a label for a set of concepts: the set made up of all the concepts that may correspond to that in its original language. The headword indicates hence an entire range of references that can be referred to. The constraint list is used to delimit a concept within that range as to be clearly and unambiguously indicated by the UW. It restricts the interpretation of a headword to a subset or to a specific concept included within.

> For example the UW **"broadcasting(icl>activity)"** indicates the concept "broadcasting" as kind of activity (icl> = included in), and another UW **"activity(icl>abstract thing)"** indicates "activity" as an abstract thing. This hierarchy goes up to several root concept which correspond to the main types of words such as noun, verb adjective etc.

In other words a triple model is implemented where relations play the role of predicates for concepts. Also a set of universal attributes is defined which serves for description of such concept attributes as time, mood, direction etc. Two major processes of Enconversion and Deconversion are considered. Enconversion is a process of analysis of sentences in a natural language and generation of UNL relation sets called UNL sentences. Deconversion is analysis of UNL sentences and generation of natural language sentences. For each of these activities software tools are provided called` EnConverter and DeConverter. These software tools are designed to work with resource files of natural language they apply to. Natural language module developers are responsible to provide these resources, which include:

- *Natural language word dictionary*
- *Natural language deconversion rules*
- *Natural language enconversion rules*

Below natural language word dictionaries and deconversion rules are described.

Natural language word dictionary

The word entries of each language are stored in the **Word Dictionary**. Each entry of the Word Dictionary is composed of three kinds of elements: the **Headword**, the **Universal Word (UW)** and the **Grammatical Attributes**. A headword is a notation/surface of a word of a natural language which is to be used in composing a sentence. An UW expresses the meaning of the word which is to be used as a trigger or link for obtaining equivalent words or expressions. Grammatical Attributes describe the word in scope of natural language grammar and are referenced in deconversion and enconversion rules. The entry of the dictionary has the following format:

[HW] "UW" (ATTR , ...) <FLG , FRE , PRI>;

Where:

HW - The headword of a native language
UW - The Universal Word of UNL referencing to the concept in UW Knowledge base.
ATTR - The Grammatical Attributes of the headword defined for the natural language.
FLG - The Language Flag of one character of English in ASCII code.
FRE - The Frequency of the headword. Used for defining the retrieval order in DeConverter.
PRI - The Priority of the headword. Used for defining the retrieval order in EnConverter.

This format is the same for all natural languages. The only thing that varies is the content of the dictionary.

Natural language deconversion rules

The algorithm of deconversion of UNL sentences into Armenian is mainly defined by deconversion rules. These rules are being fed to the DeConverter software and compiled into binary files which are then used for deconversion. As the format of the deconversion rules is predefined, the rule developer needs to formalize the algorithms for generation of sentences in natural language using this format. So the deconversion software (DeConverter) only performs a set of predefined actions over the rules.
DeConverter operates on sentences represented by UNL expressions as directed hyper graph structures called Node-nets. The root node of a Node-net is called Entry Node and represents the main predicate of the sentence. Then generation rules are applied to every node in the Node-net respectively, the word list (Node-list) in the target language is generated.
A Deconversion Rule is composed of Conditions for the nodes placed on Generation Windows and Condition Windows, and Actions and/or Operations for the nodes placed on Generation Windows. Such deconversion rules describe what kind of actions and/or operations should be carried out for all phenomena of a language and on what conditions. DeConverter will find out the most suitable rule every time, insert a new word or add a suffix or prefix to complete a word at a certain place, and so on. A sentence will finally be completed after having applied a set of all necessary rules.
A Deconversion Rule has the following syntax:

<TYPE> {<COND1>:<ACTION1>:<RELATION1>:<ROLE1>}
{<COND2>:<ACTION2>:<RELATION2>:<ROLE2>}

When the node on the left Generation Window satisfies **<COND1>** attributes, the node on the right Generation Window satisfies **<COND2>** attributes the grammatical attributes of the nodes in the Generation Windows are rewritten according to **<ACTION1>** and **<ACTION2>** respectively. If either node in the Generation Windows is indicated as "Insertion", the node linked to the other node by **<RELATION1>** or **<RELATION2>** in the Node-net is inserted into the Node-list. And the operations are carried out on the Node-list depending on the type of rule shown in field **<TYPE>**. The **<ROLE1>** and **<ROLE2>** fields are used for references to co-occurrence dictionary which is used as a supplementary resource for semantic knowledge. We will not discuss this resource in this paper as it is in stage of development now.

UNL is designed to be independent from the peculiarities of natural language. So the algorithms of EnConverter and DeConverter tools are mostly derived from corresponding rule definitions

implemented by the language module providers. This is done considering the variety of natural languages, and approaches for their analysis and generation.

Some linguistic properties of the Armenian Language

The Armenian language belongs to agglutinative-analytical languages and is notable for variety of different morphological categories. A large amount of syntactic and semantic level information can be obtained on morphological level. The first phases of this project considered the creation of some formalization of the description of modern Eastern Armenian morphology and development of attributes for its presentation in UNL format ([3]).

For the formalized description of morphology the following tasks were considered and implemented:

- Development of rules for word-form articulation on morphemes; on the basis of these rules the vocabularies and tables of morphemes are created;
- Determination of dependences of grammatical categories from morphemes, that constitute the word form;
- Determination of the calculus of all word forms, produced by the given stem;
- Development rules for definition of the word form grammatical characteristics.

The solution of these tasks were discussed in previous papers ([3],[4]).
In addition to linguistic issues that are mentioned, also some practical problems were met during the development of dictionaries. One the most essential problems that we deal with during the presentation of Armenian morphology is the ***Variation of root words for some alternated and homonymic forms.*** This brings to some restrictions for usage of standard approaches that are usual for such languages as English. The problem arises when an alternation of the stem of the word occurs. A typical example of such situation is the following:

> The word "unnıū" which means "house" is a root word itself. But if in English the genitive form of this word is generated simply by adding an "s" at the end in Armenian it would be "unuū" (the second letter changed from "ı" to "u").

In addition the high flectivity of the Armenian language makes such cases more complicated.

> If the ablative for "house" in English would be "from house", then in Armenian it is "unūhg". In case of Armenian the third type of root comes into play. It's "unū", which gained the "hg" flexion afterwards.

Now considering the fact that UNLs deconversion rule format doesn't support variation of word strings that are already loaded from the dictionary, we can see that the solution must be found in a creation of corresponding optimal construction of the dictionary. In such cases two approaches can be considered for implementation.
The first approach supposes that the root words are narrowed from the right side to match the longest fixed stem. In such case the new root for the word "ÏáõÝ" will be "Ï". The obvious weakness of such approach is the fact that in some irregular cases not only one or more letters can change but a totally different root can be used. For instance:

The past tense for the word "գալ" which means "come" will be "եկա" ("came").

Another argument against this approach is that most of the roots that will be shortened will become a single letter words. And the generation of all supletive forms will need a big amount of rules, which will bring to more complexity.
The second approach that can be applied is called grammatical root selection. In this case each word is represented with several entries in the dictionary. They share the same UW and most of the grammatical attributes. The differences are in the way the words are written and attributes indicating the stem. So for the word in the example above we will have 3 entries:

[տուն]{}"house(icl>building)"(N,NC,Sg,StB,GNP33)<A,0,0>;
[տա]{}"house(icl>building)"(N,NC,Sg,StC,GNP33)<A,0,0>;
[տնի]{}"house(icl>building)"(N,NC,Sg,StD,GNP33)<A,0,0>;

Note that each stem in the first list has a different attribute indicating its stem (**StB, StC, StD**).

The benefit of this approach becomes more evident when taking into account the fact that the maximum number of stems for any word is limited to 3. The downside of this approach will be the raised number of word entries in the dictionary. But this is not so essential while considering that the dictionary is being compiled only once and the speed of work with it is very fast even with very large number of entries in it.

So the second approach is taken for the development of Armenian word dictionary. Of course the definition of different stems for the words in the dictionary would be worthless if it wasn't taken into account in the deconversion rules. The rules now can have a condition of certain stem to match it. So for genitive and ablative forms if the word "ïáõÝ" we'll have the following rules:

:{N,GNP33,StC,^Pl,Gen,^STEM:+STEM::}"[ ]:::"P10;
:{N,GNP33,StD,^Pl,Abl,^STEM:+STEM::}"[Çó]:::"P10;

Note that in the first rule no flexion is being added to the stem as it is already in a ablative form.

Classification of rules

As it is mentioned above the algorithm of deconversion is mainly defined by the deconversion rules. This approach helps to create the deconversion software independently from considered natural language. All properties of a natural language are reflected only in its rules and dictionaries.
For a more complete and precise definition of this algorithms a classification of the rules is proposed here, to guarantee a proper coverage of the grammar of the Armenian language.
Thus, a set of classifications is described that will cover the basic formal properties of the Armenian language. They form a multidimensional space which includes characteristics (i.e. coordinates) of all possible deconversion rules.

So, the multidimensional space of rules for the Armenian Language will consist of the following dimensions:

- **Rule Type**

- **Insertion**
- **Generality**
- **Relation Type**
- **Attribute set**

Rule Type
The format that UNL defines for deconversion rules consists of several basic types. They correspond to the actions that can performed by the Deconversion software. Below the list of deconversion **rule types** defined in UNL is given:

- *Attribute Changing ":"*
- *Right-shift "R"*
- *Left-shift "L"*
- *Backtrack "?"*
- *Left Node Assignment Backtrack "?L"*
- *Right Node Assignment Backtrack "?R"*
- *Left Node Copy "C" or "CL"*
- *Right Node Copy "CR"*
- *Left Node Deletion "DL"*
- *Right Node Deletion "DR"*

Though this is one of the initial classifications predefined in UNL, in most cases it can be omitted, as most of the rule types are playing auxiliary role. The *Attribute Changing* rules are supposed to carry out main actions of generation and insertion of morphemes in the list which eventually become a sentence in natural language. Also the *Right-shift* and *Left-shift* are used for controlling the direction the deconversion algorithm moves to.

Insertion
The classification by insertion side divides the *Attribute Changing* rules into *left* or *right insertion*. In this type of rules a node is considered that is already inserted into the node-list with a candidate node and define from which side the candidate will be inserted, whether it will be before or after the node in the list.

Generality
When implementing the generation of sentences in the Armenian Language it is very difficult to consider all possible variations of expression forms. Although, the amount of possible forms can be considered as a limited value, we can always miss some forms. To decrease the harm caused by such cases and save the quality of generated sentence we introduced a classification by *generality* level.
The classes of rules that correspond to *grammatical generality levels* start from most general and tend to more restricted grammatical situations. These classifications can be presented as trees where roots are general rules for some typical word order used in Armenian (SVO = subject -> verb -> object) and leaves are final descriptions of all regular and irregular structures.
Below is an extract from a such tree. The example is shown for a type of **agt** relation which indicates a thing(subject) that initiates an action(verb) (for example "man walks"):

The most general case on insertion of a subject of action at the left on the verb.
:"::agt:"{:::}P160;

More precise cases with 3 different rules for verbs, nouns and pronouns
:"N::agt:"{V:::}P161
:"NUM::agt:"{V:::}P161
:"Pron::agt:"{V:::}P161

The default case is when the noun is in the straight form (stem B)
:"N,StB::agt:"{V:::}P162
:"N,StB::agt:"{V,StB:::}P163

Rules for particular types of verbs
:"N:+@def:agt:"{V,StC,^GVP20,^GVP11,^GVP17:+PartIpf,+AGT,+aux::}P164;
"Արամ+ը" կարդ+ում+է (էր)
:"N:+@def:agt:"{V,StB,GVP20:+PartAdv,+AGT,+aux::}P164;
"Արամ+ը" տալ+իս+է (էր)
:"N:+@def:agt:"{V,StB,GVP11:+PartAdv,+AGT,+aux::}P164;
"Արամ+ը" գալ+իս+է (էր)
:"N:+@def:agt:"{V,StB,GVP17:+PartAdv,+AGT,+aux::}P164;
"Արամ+ը" լալ+իս+է (էր)

Rules for particular types of numericals
:"NUM:+@def:agt:"{V,StC,^GVP20,^GVP11,^GVP17:+PartIpf,+AGT,+aux::}P162;
:"NUM:+@def:agt:"{V,StB,GVP20:+PartAdv,+AGT,+aux::}P162;
:"NUM:+@def:agt:"{V,StB,GVP11:+PartAdv,+AGT,+aux::}P162;
:"NUM:+@def:agt:"{V,StB,GVP17:+PartAdv,+AGT,+aux::}P162;

Rules for particular types of pronouns
:"Pron::agt:"{V,StC,^GVP20,^GVP11,^GVP17:+PartIpf,+AGT,+aux::}P162;
"ես" կարդ+ում+է (էր)
:"Pron::agt:"{V,StB,GVP20:+PartAdv,+AGT,+aux::}P162;
"ես" տալ+իս+է (էր)
:"Pron::agt:"{V,StB,GVP11:+PartAdv,+AGT,+aux::}P162;
"ես" գալ+իս+է (էր)
:"Pron::agt:"{V,StB,GVP17:+PartAdv,+AGT,+aux::}P162;
"ես" լալ+իս+է (էր)

As it can be seen from the example above, the priority of rules is increasing as the rule becomes more restricted.

In a properly tuned system the need for general rules decreases as all cases are being covered with appropriate rules. This approach also helps to have translation even on early stages of the work, and enhance it's quality as new rules are being added.

Relation Type

As UNL defines 45 types of relations among concepts in the sentence, we also need to define rules to cover all these relations. This requires development of generality trees for each type of relation defined in UNL.

Attribute Sets

The classification of rules by attribute sets is done mainly to help to estimate the amount of possible rules. The coordinates of this dimension include all possible combinations of both attributes describing Armenian grammar and universal attributes defined in UNL.

System of rules

The practice shows, that non-synchronized implementation of morphological, syntactic and semantic rules often brings to situations when the covering of all possible cases in the sentences of natural language can be implemented only by numerous variations of rules. This is being caused mainly because the algorithm of deconversion gains many paths to the same case, when the rule is being applied. And for each path a different order of nodes (morphemes) must be considered in the rule. As a result the amount of rules increases dramatically while the quality of deconversion becomes poorer. The rules begin to contradict to each other and all this leads the deconversion algorithm in wrong directions.

The solution of this problem is given by the separation and synchronization of morphological, syntactic and semantic rules. The rules are divided into 3 ranges, each of which has it's own execution phase in the work of the algorithm. Namely, we define three phases of generation (Figure 1). This division is done by priority ranges.

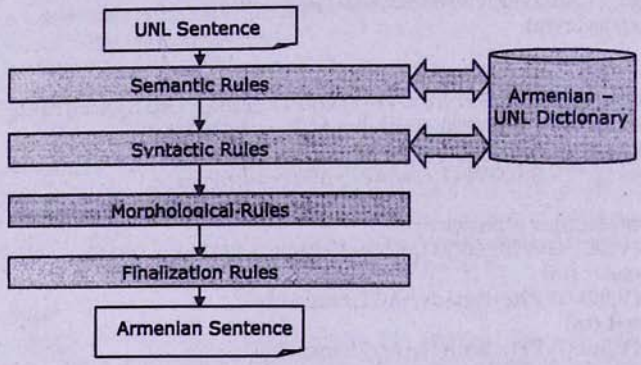


Figure 1. Rules interaction with dictionary

As it is shown in Figure 1 the deconversion phases refer to the Armenian – UNL Dictionary during semantic and syntactic phases of generation only. The information gained during these phases is used then for generation of morphology.

Semantic phase

While the syntactic and morphological rules are required for the Deconverter to generate any result, the semantic rules are optional. Semantic rules contain references to the concepts in condition fields. They are used for coverage of cases where the meaning of the concept plays essential role. The concepts in the rules can be used in both positive and negative forms, triggering or blocking insertion of the corresponding root morphemes. In other words, semantic rules are syntactic rules with additional conditions concerning meanings of morphemes that are considered in the rule.

Let's consider the following example:

The expression *"keep in mind"* can be expressed using the relation *plc:*

plc(keep(agt>thing,obj>thing), mind(icl>abstract thing))

in this case the relation will be processed with the rule

:"V:::"{N,StC,GNP11:+Loc:plc:}P152

This is a typical rule for *plc* relation which inserts the verb (V) of action in front of the noun (N) where the action took place (For example *"cook in the kitchen"*). So for the relation shown above the rule will produce:

պահել մտքում

Although this expression is grammatically and semantically correct, in the Armenian language we can use a special word that will express the meaning more correctly:

մտապահել

But as this case is irregular, a special treatment is needed. This is done by a semantic rule which considers the concepts (Universal Words) that are related.

:{N,StD,GNP11,[[mind(icl>abstract thing)]]:+char_a,+STEM:plc:}"V,StB [[keep(agt>thing,obj>thing)]]:::"P153

In this rule the UWs of the words are considered. For the first UW (*mind(icl>abstract thing)*) the stem D which is the shorter alternated form of the root "ՄՏՔ" is selected from dictionary. The straight form of the verb "պահել" (*keep(agt>thing,obj>thing)*) is being attached to the stem ("մտ") then. Also few temporary attributes are being added to the first morpheme ("մտ"). This Attributes have the following meanings:

char_a – indicates that the "ª" character mast be inserted after.
STEM – indicates that no blank space characters are allowed after.

Each of these attributes will require another rule to perform the actions they suppose.

So, the example above shows that semantic rules are to improve the translation quality by processing special cases that contradict general syntactic rules. This is why they need to have the highest priority, to be executed before the syntactic rules.

Syntactic phase

Syntactic rules are responsible for building of general grammar of the language. They consider UNL relations among nodes and both grammatical and universal attributes of the nodes and map them to syntactic structure. Actually syntactic rules transform the UNL graph of a sentence into a list of root morphemes that correspond to concepts.

These rules are given higher priority than morphological. This means that the list of roots is generated first. After the order of these roots is established, the morphological rules are being applied.

In the example below the deconversion of a simple sentence is shown:

The UNL sentence for "Women were going home." would be:

agt(go(icl>do).@past.@entry, woman(icl>female person).@pl);
obj(go(icl>do).@past.@entry, home(icl>place));

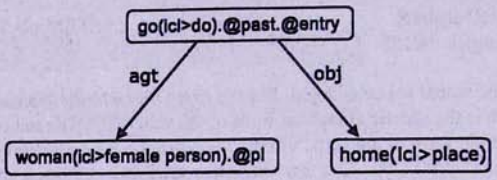
2 relations are used to build the graph of the sentence:

agt - indicates a thing that initiates an action
obj - indicates a thing that is directly affected by an event or state.

The attributes that are attaches to the UWs are:

@past - the past tense of go(icl>do) is used.
@entry - go(icl>do) is the root of the graph (the first node to be inserted in the output list).
@pl - woman(icl>female person) are in plural.

So the graph of the sentence would be:



For the UWs of the graph the following entries are being retrieved from the Armenian – UNL dictionary.

[կին]{}"woman(icl>female person)"(N,Sg,GNP19,StB)<A,0,0>;
[կի]{}"woman(icl>female person)"(N,Sg,GNP19,StC)<A,0,0>;
[կաճ]{}"woman(icl>female person)"(N,Pl,GNP19,StD)<A,0,0>;
[գնալ]{}" go(icl>do)"(V,DoV,AcVc,Inf,GVP2,StB)<A,0,0>;
[գի]{}"go(icl>do)"(V,DoV,AcVc,GVP2,StC)<A,0,0>;
[գնաց]{}"go(icl>do)"(V,DoV,AcVc,GVP2,StD)<A,0,0>;
[տուն]{}"home(icl>place)"(N,NC,Sg,StB,GNP33)<A,0,0>;
[տու]{}"home(icl>place)"(N,NC,Sg,StC,GNP33)<A,0,0>;
[տն]{}"home(icl>place)"(N,NC,Sg,StD,GNP33)<A,0,0>;

Note that 3 stems are available for each UW. Now the rules will specify which stem will be chosen.

The syntactic rules would be:

:"N,GNP19,StD,@pl:+@def:agt:"{V,StC,GVP2:+PartIpf,+Pl,+aux::}P164;
:{V,StC,@past:+Part2,+PartPer::}"N,StB::obj:"P153;

Note that the rules specify that the [կաճ] (StD), [գնաց] (StC) and [տուն] (StB) stems are to be chosen. So after these rules we'll have "[կաճ][գի][տուն]". After morphology

phase the proper endings and spaces will be added. And eventually the translation will be "կաճայք գնում էին տուն:"

Morphological phase

Rules of morphological generation have priorities lower than semantic and syntactic ones. So they are considered only after the semantic and syntactic rules are applied. These rules are mainly performing flexion attachment and auxiliary verb insertion over already inserted and processed root morphemes. During the semantic and syntactic processing roots gain temporary attributes specifying the morphological forms they will get. These can be prefixes, suffixes, auxiliary verbs, articles, and other auxiliary words. Temporary attributes then trigger the execution of corresponding morphology rules.

For example, the syntactic rule in the sample above adds attributes ***PartIpf, Pl, aux*** to the verb. These attributes indicate the following actions:

> ***PartIpf*** *- The verb gains flexion for Participle Imperfect form.*
> ***Pl*** *– The action is initiated by a noun that is in plural form.*
> ***aux*** *- An auxiliary verb need to be inserted after the verb. The type of the auxiliary verb is chosen considering* ***Pl*** *attribute.*

So the morphological rules will do:

*:{V,Part2,StC,**PartIpf**,^STEM:+STEM::}"[ում]:::"P13;*
PartIpf *=> գն + ում*

*:{V,@past,**aux**,**Pl**:-aux::}"[էին],Aux:::"P14;*
Pl, aux *=> գնում + էին*

Finalization Phase

The last phase that is called **Finalization Phase** and has the lowest priority is inserting blank spaces, comas and other punctuation parks according to the attributes that the morphemes gained during previous phases.

For example, to indicate the places for blank space character the STEM attribute is used. All morphemes that have this attribute are considered not to be followed by space.

Conclusion

If we consider the classifications listed above as dimensions, the whole set of rules can be presented as a multidimensional space and each rule will get an exact coordinate set in this space. As each dimension in the space will consist of a discrete set of possible values. The amount of the points in this space will be the product of the amounts of all possible values in all dimensions, which will be the maximum amount of possible rules. Obviously only a small part of them is needed to cover the whole language.

References

[1] V. Avetisyan, V. Sahakyan, L. Petrosyan, R. Urutyan, A Manukyan, L. Hovsepyan. Development of Armenian Language Module of UNL. Convergences'03, Alexandria, Egypt, 2003

[2] V. Avetisyan, A. Manukyan. Development Environment for the Armenian Language Server of UNL. Proceedings of the International Conference on Computer Science and Information Technologies, Abstracts. Yerevan. 2003.

[3] A. Manukyan, V. Avetisyan. Armenian Morphology Model for UNL Language Server. Proceedings of the International Conference on Computer Science and Information Technologies, Abstracts. Yerevan. 2003.

[4] A. Manukyan. Formalized Model for Ancient Armenian Verb//Proceedings of the International Conference on Computer Science and Information Technologies, Abstracts. Yerevan. 2001.

[5] G. Jahukyan. The Universal Linguistic Model. Yerevan. 2000.

[6] V. Avetisyan, T. Grigoryan. The UNL Toolbox CICLING2005 Mexico, Mexico. 2005

[7] V. Avetisyan, R. Urutyan, L. Hovsepyan, S. Tioyan. Development of Deconversion Rules for Generation of Armenian Sentences from UNL. Proceedings of the International Conference on Computer Science and Information Technologies, Abstracts. Yerevan. 2005

[8] H. Uchida, M. Zhu, Tarcisio Della Senta, The UNL, "A Gift for a Millenium", UNU/IAS, Tokyo, 1999.

[9] U. Hiroshi, M. Zhu, "The Universal Networking Language beyond Machine Translation" UNDL Foundation, September 26, 2001.

[10] T. Dhanabalan, K.Saravanan, T.V. Geetha, Tamil to UNL EnConverter. Goa. India.

[11] T. Dhanabalan, T.V. Geetha. UNL Deconverter for Tamil. Convergences'03, Alexandria, Egypt, 2003.

[12] R. Martins, R. Hasegawa, V.Nunnes, M.Graças. HERMETO a NL-UNL enconverting environment. Convergences'03, Alexandria, Egypt, 2003.

[13] Kuntal Dey, Pushpak Bhattacharyya. UNL Based Analysis and Generation for Bengali Case Structure Constructs. Convergences'03, Alexandria, Egypt, 2003.

[14] UW Manual, UNL Center, UNDL Foundation, June 2003.

[15] H. Uchida, M. Zhu, The Universal Networking Language (UNL) specifications version 7 June 2005.

[16] DeConverter Specification Version 2.6, UNL Centre/UNDL Foundation, 2002.UNL Center. Enconverter Specifications. Version 3.3 Tokyo, 2002.

UNL արտահայտություններից հայերեն նախադասությունների սերման կանոնների համակարգ

Վ. Ավետիսյան

Ամփոփում

Այս հոդվածում ներկայացված է UNL արտահայտություններից հայերեն նախադասությունների սերման կանոնների համակարգ: Առաջարկված է կանոնների դասակարգում՝ հայերենի քերականության բավարար ծածկույթ ստանալու համար: Նկարագրված են նաև հայերենի ճևաբանության և շարահյուսության ճևայնացված նկարագրման որոշակի մոտեցումներ: