# Conceptual Graphs Self-Tutoring System

Albena Strupchanska[1], Milena Yankova[1] and Svetla Boytcheva[2]

[1] Bulgarian Academy of Sciences, Linguistic Modelling Lab, CLPP,
25A Acad. G. Bonchev Str., 1113 Sofia, Bulgaria,
{albena,myankova}@lml.bas.bg

[2] Department of Information Technologies, FMI, Sofia University,
5 J. Bauchier Blvd., 1164 Sofia, Bulgaria,
svetla@fmi.uni-sofia.bg

**Abstract.** The present paper describes a self-tutoring system `CG-EST` (Conceptual Graphs Elearning Self-Tutoring system). The purpose of this system is to generate questions to the user from given text lessons, to process user answers and to check their correctness. `CG-EST` includes modules for: parsing and producing logical forms (LF); generating conceptual graphs (CG) based on LF; generating questions based on LF; defining the scope of correct answer using CG operations; matching the user's and correct answers. This approach assists the automatic eLearning systems being developed for various domains with some requirements: all sentences in the lessons have to use Controlled English (CE); to each lesson a simple concept type hierarchy and a synonyms set have to be attached. However these restrictions are useful for semantic interpretation of Natural Language (NL) texts. Thus `CG-EST` presents a solution for self-service eLearning that is ideal for training in a specific field.

## 1 Introduction

Nowadays the eLearning industry faces an immense development. In order to satisfy the increasing user interest and demands of eLearning systems it is necessary to ease the process of development of such kind of systems. One possible step towards is to automate the knowledge acquisition and the testing of a user comprehension. Many different theories for cognitive aspects of body of knowledge/knowledge acquisition exists nowadays. They distinguish different levels of knowledge with different granularity. We shall mention only two of them as our research is closer to these paradigms. Benjamin Bloom's famous "Taxonomy of Educational Objectives" [1] identifies six levels: knowledge (rote memory skills - facts, terms, procedures, classification systems), comprehension (the ability to translate, paraphrase, interpret or extrapolate material), application (the capacity to transfer knowledge from one setting to another), analysis (the ability to discover and differentiate the component parts of a larger whole), synthesis (the ability to weave component parts into a coherent whole) and evaluation (the ability to judge the value or use of information using a set of standards). Another theory claims that research in educational psychology identifies three major types of knowledge: rote (i.e., memorization), declarative (i.e., knowledge of concepts) and procedural (i.e., knowledge of a physical or intellectual process, method, or skill). Knowledge does not automatically transfer between declarative and procedural knowledge [16]. The procedural type of knowledge is more complicated to be automatically examinated since it requires deep semantic inferences so we focus only on the rote and the declarative ones. Usually the learning process includes set of lessons presented as texts and some drills that test user knowledge comprehension.

The presented system `CG-EST` (Conceptual Graphs Elearning Self-Tutoring system) translates sentences from the given text lessons into CG and automatically generates questions from them. User comprehension is tested by processing user answers and checking their correctness. Scope of correctness is presented as *minimal required* and *maximal expected* CG sets. Once the answer is translated into CG, `CG-EST` tries to match it to the so defined scope. The answer is correct only if it is a subset of *maximal* and superset of *minimal* CG sets.

Automatic extraction of formal knowledge specifications from free-text still looks too hard, effort consuming and time expensive. Automatic knowledge acquisition requires preliminary defined description of each (important) word expected in the input text. In addition an upper layer type hierarchy and some semantic primitives are required. That's why the task of automatic knowledge acquisition from free text is shifted to extraction from restricted NL input only. For our purposes we define "controlled English" as "complete declarative sentences in English". This ease the automatic processing of the lesson texts in `CG-EST` as well as the user answers.

This paper is structured as follows. Section 2 overviews briefly some related research. Section 3 describes the architecture of `CG-EST` in general. Section 4 presents the preprocessing module for generation of CG from the lessons. Section 5 focuses on self-tutoring module for questions generation and on defining the scope of correct answer by using of CG operations. Section 6 presents answering module and illustrates in more details the matching algorithm between user answer and the scope of correct ones. Section 7 contains the conclusion and sketches future work.

## 2 Related work

A lot of research is carried out concerning the usage of CG for the semantic parsing of NL sentences and storing the semantic information in CG. The first algorithm was proposed by Sowa [17–19] and it was further applied to several prototypes for different languages as Italian [20], French, German and English [14, 15].

The prototypes for acquisition of CG from text de facto process controlled English. The restrictions for most systems can be summarized as: *(i)* limited vocabulary of the NL input, *(ii)* processing of phrases and simple sentences, *(iii)* often missing syntactic analysis as a separate module. For example `CG Mars Lander` [7, 10] skips unknown words in the input sentences exemplified in [7], which is a flexible strategy to define a controlled language. The creators of the workbench `WebKB` developed so called Formalized English that provides acquisition of different ontological structures from text statements [11]. Another prototype `CGExtract` [2] is restricted in a similar manner since it deals with restricted vocabulary, grammar rules and restricted set of semantic primitives.

Several prototypes have been recently developed for generating CG from text in CE. `CGExtract` [2] constructs Knowledge Base (KB) of CG corresponding to the input text using the NL analysis and Understanding (NLU) module `Parasite` [13], an initial type hierarchy and a relevant lexicon. `Parasite` processes coreferences in extended discourse in a few sentences. The extraction of CG is automatic but the filling of lexicon and the acquitting, encoding and testing of the meaning postulates is time consuming. On the other hand `CGExtract` checks for semantic consistency of an extracted graph to already created CG KB, which is an attempt for achieving a better functionality. Another prototype [12] extracts CG from English texts, where the syntactic structure is analyzed by `Conexor` syntactic parser and represented as a "grammatical CG". The system uses concepts and relations type hierarchy, `WordNet`, and manually created rules for transformation of "grammatical CG" to CG.

Zhang and Yu present in [21] another method for generating CG from domain specific sentences using a machine-learning based approach. Each sentence in the text is parsed by a link parser. This approach uses a vocabulary of words, types of links that can be attached to them and labels indicating how they are attached in a particular sentence.

Methods for automatic generation of questions are presented in [9]. Kunichica et al. extracts syntactic and semantic information of sentences using Definite Clause Grammar, sematic categories of nouns and verbs with corresponding interrogative pronouns. They also define a correspondence between roles of prepositional phrases and interrogative pronouns. Their approach for asking questions about the content of a sentence is to determine the sematic role which will be questioned, to find appropriate interrogative pronoun for it using mentioned above correspondences, to replace query part with interrogative pronoun and to translate the sentence into interrogative one. In order to generate more complicated questions synonyms and antonyms dictionaries are used for replacement of words in sentences. Using modifiers of different instance or different words is another way of generating questions with different contents from the original sentences. Methods for questioning about contents represented by plural sentences using a relative pronoun and questioning about relationship of time and space are also described. All methods generate approximately 93% semantically correct sentences due to the rough classification of semantic categories. Moreover, the words in parsed sentences should be classified according to the defined categories in advance.

## 3 CG-EST System

`CG-EST` is an intelligent eLearning system for self-tutoring in specific scientific domains. `CG-EST` has two main modules: preprocessing module and self-tutoring module (fig. 1). The main purpose of the preprocessing module is to generate a KB of the system from given by the tutor *(i)* lessons in NL, *(ii)* type hierarchy and *(iii)* terminology. The self-tutoring module trains the user in a chosen domain by presenting some lessons and asking questions related to them to check his/her comprehension. The innovative strategy implemented in this module is an automatic questions generation from a dynamically extracted KB. In addition the self-tutoring module applies another intelligent technique that handles checking process of user answers in CE using dynamically generated scope of the correct answer. The preprocessing module is activated once but the self-tutoring module could be used many times by different users.

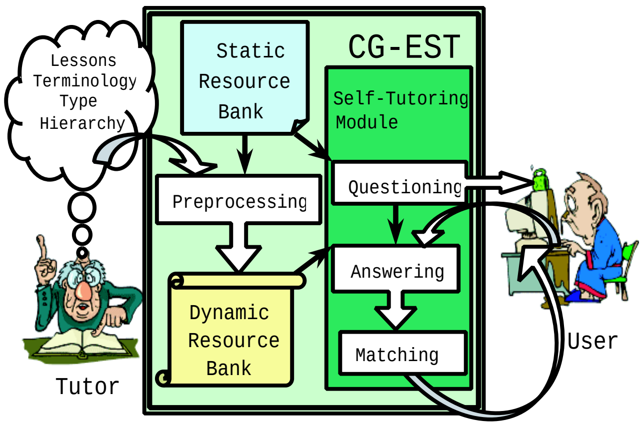


**Fig. 1.** Architecture of CG-EST system

## 4 Preprocessing Module for KB Generation

This section describes the preprocessing module (fig. 2). Its purpose is to generate a KB of CG from lessons in CE. At the beginning a lexical (morphological) analysis of the input is needed to automate the process. Then a syntactic analysis is needed. At this stage we use the system GATE 2.1 beta 1 (GATE - General Architecture for Text Engineering) [6] which provides several NLP modules. The most important for this work are the Lexical Analysis modules. They turn a text into a sequence of sentences, each of them is a sequence of lexical items (tokens). Each token is looked up in the dictionary to determine its possible features. Also part-of-speech types are assigned here. The syntactic analysis translates

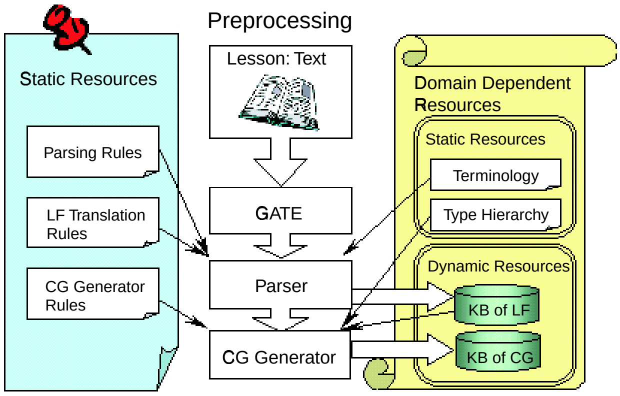


**Fig. 2.** CG-EST: Preprocessing Module

sentences into LF using parsing rules. LF and CG are two different forms used for formal representation of the syntactic and the sematic structure of sentences. LF consist of lemma-predicates corresponding to the words and $\theta$-predicates corresponding to the thematic roles in sentences. CG have concepts and relations between them. The concepts correspond to the words and the relations correspond to the thematic roles in a sentence. Thus LF and CG have very similar structure except for different granularity of lemma-predicates in LF and concepts in CG.

### 4.1 Logical Forms

A specially developed left-recursive, top-down depth-first parser, implemented in Sicstus Prolog, is used for translation of sentences in CE to LF. This parser uses grammar rules and rules for translation into LF from our static resource bank. In LF we represent all words as predicates (*lemma-predicates*) with predicate symbol (the corresponding base form (lemma) of the word) and two arguments, where the first one is a variable and the other is a Part-Of-Speech (POS) tag attached to the word. POS tag indicates which word-form of the lemma is used in the raw text, this is necessary for the further analysis. For example the word "*trades*" will be represented in LF as $trade(X,'VBZ')$, where $'VBZ'$ is tag for "verb - 3rd person, singular, present". For thematic roles we add predicates with predicate symbol $\theta$ and three arguments $\theta(X, agnt, Y)$. The constant that represents the thematic role is placed as a second argument. The rest of the arguments are bound with the lemma-predicates or constants, concerning this thematic role (see example 1). These predicates are called *$\theta$-predicates*. All proper names are represented as constants that occur as arguments of the corresponding thematic roles.

**Example 1:**
Text in NL: *Primary market trades newly issued stocks and bonds.*
LF: $logical\_form(primary\_market(\_451,'NN')\&trades(\_425,'VBZ')\&$
$\theta(\_425, agnt, \_451)\&\theta(\_425, with, \_1569\&\_1570)\&newly(\_1169,'RB')\&$
$\theta(\_1569\&\_1570, attr, \_1169)\&issued(\_1367,'RB')\&\theta(\_1569\&\_1570, attr, \_1367)\&$
$stocks(\_1569,'NNS')\&bonds(\_1570,'NNS'))$.

The adverbial relations in thematic roles are not specified (e.g. location, instrument, time etc.), because this task is extremely difficult and there is no need of such kind of "deep" semantic (dependency and context) analysis. Therefore all adverbial relations are marked with prepositions as they are met in the text.

An important resource given by the knowledge engineer (tutor) with the lesson text is a terminology bank. The terminology bank is a list of words and phrases used as terms in the given text, which should be represented as single concepts. During the parsing process the compound terms from the terminology bank are recognized in the text and the consisting words are concatenated with "_" in a single predicate name. For instance "*primary market*" becomes "*primary_market*". This allows correct interpretation of terms as concepts in the next step of generating CG form LF. Once the lesson text is parsed, all LF are stored into the KB of LF for further processing. All LF in this KB are in conjunctive normal form (CNF).

### 4.2 Conceptual Graphs generator

The CG generator module consists of a preprocessor and a translator submodules. The preprocessor instantiates the arguments of $\theta$-predicates with the lemma-predicates and removes all lemma-predicates from the LF. So after this stage LF contains only $\theta$-predicates which could be almost directly translated to relations. The name of the CG relation is taken from the second argument of a $\theta$-predicate and the other arguments are translated as concepts corresponding to it. The translator uses some rules for translating the $\theta$-predicates into relations and concepts. Most of $\theta$-predicates are translated as relations while others just add some features to already created concepts (see example 2).

**Example 2:**
This part of LF:
$\theta([year,'NN'], count, [one,'CD'])\&\theta([security,'NNS'], maturing, [year,'NN'])$
will be translated as the following CG:

```
[SECURITY]->(MATURING)->[YEAR: @1].
```

Some rules for translation of the POS types assigned by the Lexical Analysis modules to concept referents are also used during the concept creation.

The CG generator creates mostly simple graphs and the only kind of compound graphs are type definitions graphs. They are not created just by looking at the sentence but also by checking the type hierarchy (see example 3). So the

CG generator module takes as input the LF KB, applies some rules and translates the LF KB into CG KB.

**Example 3:**
The sentence:
*"Security is a financial instrument which represents debt of corporation"*
will be translated as the following CG:

```
[SECURITY]->(DEF)->[Definition: [REPRESENT]-
                        (AGNT)-> [FINANCIAL_INSTRUMENT]
                        (OBJ)-> [DEBT]->(POSS)->[CORPORATION] ],
```

if we have a clause in the type hierarchy *isa(security, financial_instrument).*

Using string similarities it is relatively easy to automatic construct "naive" type hierarchy e.g. *primary_market* is a subtype of *market*. For more complex cases as in the example above more powerful and complex methods for generation are needed. Considering that a type hierarchy corresponding to a lesson contains just a few concepts, it would be ineffective to apply any methods. So we chose the type hierarchy to be an input for the CG-EST system. A tutor gives the type hierarchy as clauses that are used in the CG generator and the subsequent modules.

# 5 Self-Tutoring module

CG-EST self-tutoring module automatically generates questions and defines the scope of correct answer. It aims to assist the process of checking user comprehension of material presented as lessons in NL.

## 5.1 Questions generation

The main goal of this module is not just to invert one of the sentences given in the lesson text to questions, but to generate a more complicated question by making some inference. Then it is clear that if the user answer is correct, s/he understands the lesson, not just repeats explicitly said statements in the text.

The module for question generation functions by performing the following steps (see fig. 3 and example 4):

- step 1: Module selects one logical form $L_i$ from the generated KB of LF (lets denote as $\sum$);
- step 2: Module finds the set:
  $S = \{L_j | L_j \in \sum \wedge \exists \; \textit{positive literal} \; l \in L_i \wedge \exists \; \textit{substitution} \; \sigma(l\sigma \in L_j)\}$
  This set contains all correct statements related to the question, which will be given to the user;
- step 3: Then we found the set C of all positive literals that represent concepts and belong to the LFs from S. So we skip all literals for thematic roles, time, place etc. $C = \{c | \exists L_i \in S \wedge \exists \; \textit{substitution} \; \sigma(c\sigma \in L_i)\}$

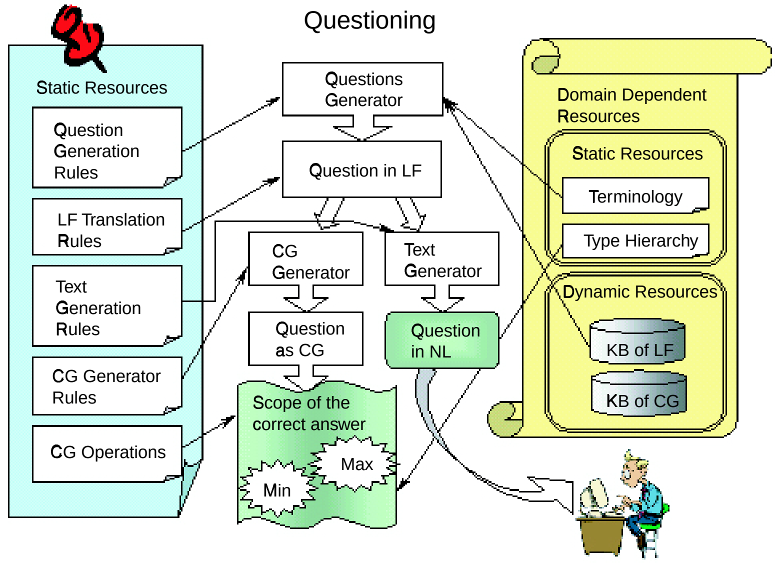


**Fig. 3.** CG-EST: Questions Generation Module

- step 4: Module chooses a non empty subset A $\subseteq$ C. The subset A contains those concepts, which will be questioned;
- step 5: The set S is reduced to the set $S'$ by skipping those LF that do not contain chosen concepts in A, i.e:
  $S' = \{L_p \in S | \exists c \in A \wedge \exists \ substitution \ \sigma(c\sigma \in L_p)\}$
- step 6: Question logical form Q is produced by replacing variables in the terms representing concepts from A with the variable "UNIV" in $S'$ and taking common parts of LF from $S'$ by applying unification algorithm. Further an additional step reduces Q to $Q'$ by skipping all "singeleton" literals i.e literals that do not contain variables for which exists literal from Q that contains any of them. This step is necessary to generate coherent, correct and sensitive questions;
  $Q' = Q \setminus \{l \in Q | X = \text{arg\_set}(l), \forall x \in X : \neg \exists l_i \in Q(l_i \neq l \wedge x \in \text{arg\_set}(l_i))\}$
- step 7: Text Generation module generates a question in NL from question's logical form Q′ by using backward operation for reconstruction of the sentence from its LF. This operation is simple because we do not have discourse and all sentences are universally quantified due to the specific domain.

The algorithm guarantees that we will generate only correct questions because we don't use any kind of generalization, specialization of type hierarchy inference. **Example 4:**
Let the given Lesson from which KB of LF and KB of CG are produced be:
(4.1) *Financial market trades financial instruments. Security is a financial instrument which represents debt of corporation. Bond is a security which repre-*

*sents debt of corporation. A stock is a security that represents an equity interest in a business. The owner of a stock is a stockholder. The stockholder has the right to vote in stockholder meetings. Stockholder meetings take place regularly at the end of the fiscal year. Primary market is a financial market that operates with newly issued financial instruments. Primary market trades newly issued securities and provides new investments. Secondary market is a financial market that operates with already issued financial instruments. Secondary market trades already issued securities.*

For instance let it be chosen LF (see Example 1) of the sentence:
(4.2) *Primary market trades newly issued stocks and bonds.*
Then the set of related sentences to (4.2) that is found after logical inference will contain LF of the following sentences:
(4.3.1) *Financial market trades financial instruments.*
(4.3.2) *Security is a financial instrument which represents debt of corporation.*
(4.3.3) *Bond is a security which represents debt of corporation.*
(4.3.4) *A stock is a security that represents an equity interest in a business.*
(4.3.5) *Primary market trades newly issued stocks and bonds.*
(4.3.6) *Primary market trades newly issued securities and provides new investments.*
(4.3.7) *Secondary market trades already issued securities.*
(4.3.8) *Primary market is a financial market that operates with newly issued financial instruments.*
(4.3.9) *Secondary market is a financial market that operates with already issued financial instruments.*
(4.3.10) *Financial market trades securities.*
(4.3.11) *Financial market trades bonds.*
(4.3.12) *Financial market trades stocks.*
(4.3.13) *Secondary market trades already issued bonds.*
(4.3.14) *Secondary market trades already issued stocks.*

The set of common concepts of (4.3) is:
(4.4) *[already_issued, bond, financial_instrument, financial_market, newly_issued, primary_market, secondary_market, security, stock]*

Let the chosen subset of (4.4) also contains the relation "trade securities":
(4.5) *[financial_market, primary_market, secondary_market]*

Then the reduced set of LF (4.3) contains LF of the following sentences:
(4.6) *Secondary market trades already issued securities. Primary market trades newly issued securities and provides new investments. Financial market trades securities.*

To produce LF of a question all concepts from (4.5) are replaced by "UNIV" and common parts are taken from LF :
(4.7) $logical\_form(trades(\_425,'VBZ')\&\theta(\_425, agnt, UNIV)\&$
$\theta(\_425\, obj, \_1569)\&securities(\_1569,'NNS')$.

The text generation module produces: (4.8) *Who does trade securities?*

### 5.2 Scope of correct answer

One approach for automatic check-up of answers is to assign scores to them. In [8] the answer is matched against three types of patterns: QA patterns (manually prepared), Qtargets (sematic type of determined answer) and Qargs (additional information related to Qtargets), and a score is calculated according to the matching. The standard approach for checking user answer is to compare it with a given beforehand correct answer. Both approaches need some manual work in advance. Our approach is to check the answer correctness by using an automatic generated scope. To have the user freedom in answering and according to [3] we chose to define correct answer scope as two sets: a *minimal required* and a *maximal expected* answer set. The approach adopted in [3] checks user answer both syntactically and semantically, it also gives a useful feedback to the user. On the other hand a lot of manually prepared meaning postulates are needed.

Since the knowledge representation model in `CG-EST` is CG, our approach consists of translating the question into CG (query graph) and applying CG operations to determine the scope of a user answer. The LF of the question obtained by the previous module is processed by the CG generator and a query CG is created (see example 5).

**Example 5: Query graph for (4.8)**

```
[UNIV]<-(AGNT)<-[TRADE]->(OBJ)->[SECURITY: {*}].
```

The projection operation projects one graph into another if the latter is a specialization of the former one. So by projecting the query graph to the knowledge base, `CG-EST` extracts all CG that fulfill the query graph (see example 6). These graphs also contain minimal information that should be included in the answer. Thus the resulting graphs are considered as the minimal required answer.

**Example 6: Minimal answer graphs for the question (4.8)**

```
[PRIMARY_MARKET]<-(AGNT)<-[TRADE]->(OBJ)->[BOND: {*}].
[PRIMARY_MARKET]<-(AGNT)<-[TRADE]->(OBJ)->[STOCK: {*}].
[PRIMARY_MARKET]<-(AGNT)<-[TRADE]->(OBJ)->[SECURITY: {*}].
[SECONDARY_MARKET]<-(AGNT)<-[TRADE]->(OBJ)->[SECURITY: {*}].
```

In addition these CG are processed as in [4] to obtain the corresponding pairs (query concept/ KB concept). For each retrieved KB concept, all CG from minimal required answer are obtained and a special maximal join operation is executed. "Special" here means that we perform maximal join on batch of CG until only one CG is obtained. The algorithm for the operation consists of two steps:

- step 1: arbitrary selection of two graphs, maximal join carried out on them and replacement of these two graphs by the result of the join in order to receive new CG batch;
- step 2: performing step 1 over the CG batch until the CG batch consists of one CG.

Thus, by applying the algorithm we reduce the set of minimal answer graphs.

**Example 7: Reduced minimal answer graphs for the question (4.8)**

```
(7.1) [PRIMARY_MARKET]<-(AGNT)<-[TRADE]-(OBJ)-[BOND: {*}]
                                       -(OBJ)-[STOCK: {*}].
(7.2) [SECONDARY_MARKET]<-(AGNT)<-[TRADE]->(OBJ)->[SECURITY: {*}].
```

The maximal answer set has to contain all true statements related to the question. These are not only explicitly said in the lesson text facts but are also facts that can be extracted with some inferences. A set containing all these truth utterances, is already created by the question generation module in step 2 (see example 4, especially (4.3)) . The representation of facts in this set is in LF and the CG generator from the preprocessing module is used again to translate them into CG and create maximal answer CG set.

## 6 Answering module

This module expects user answer in a few full declarative sentences in English as an input. Answering module performs similar techniques as the preprocessing module but instead of generating KB of LF and CG, it produces user answer in LF and user answer as CG (fig. 4).

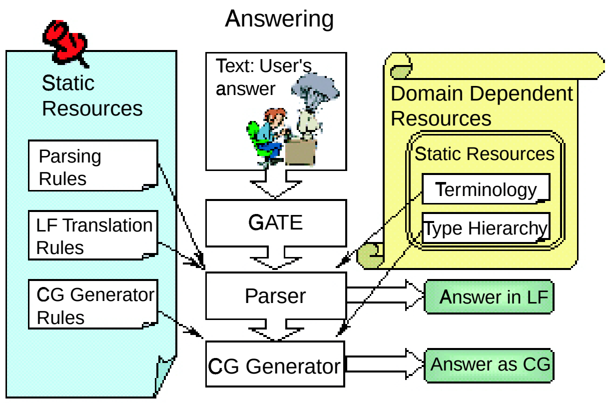


**Fig. 4.** CG-EST: User Answer Processing Module

### 6.1 Matching algorithm

The matching algorithm realizes the actual checking of the user utterance correctness against the scope (fig.5). The main goal of this step is to verify the user answer and to give him/her detailed feedback. The feedback given by `CG-EST` for truth to the user is elaborated taking into account the feedbacks used in [5].

The correct user answer has to include the minimal required answer and to be subset of the maximal expected answer. Since the knowledge representation model in `CG-EST` is CG, our approach consists of determining the user answer scope as a set of CG and applying CG operations to define its correctness. There are two steps provided for matching the answer into the scope of correctness. Both steps use the matching rules as described below.

The first step validates if the user answer contains minimal required information. As the only operation that preserves truth is the projection, it is used to match each CG from the minimal answer set to a user answer CG set. If the match succeeds, this step is followed by the second one. Otherwise, the algorithm tries to project the user answer to the minimal answer CG set. If the projection succeeds then the answer is incomplete, see example 8, especially (8.2), because the minimal required information is partially expressed. If the projections mentioned above are impossible, the answer is considered as absolutely wrong, see example 8, especially (8.1).

The second step aims to prove the truth of additional statements made by the user. They should be related to the question and be expressed in the lesson. This means that the user answer CG set should be a subset of the maximal answer CG set. The algorithm checks if every CG from the user answer set can be projected to a CG from the maximal answer set. If the projection succeeds the user answer is absolutely correct, see example 8, especially (8.4). If not - there are some utterances that are false and the answer is only partially correct as we have already proved that user answer contained required statements in minimal answer set, see example 8, especially (8.3). After the user answer is processed all information about the correctness (wrong, incomplete, partially correct and correct) is given to the user as a feedback in NL form.

**Example 8: Matching user answer**
Let's ask the question (4.8) *Who does trade securities?* with minimal required answer set (example 7) and maximal expected answer set translation of (4.3) to CG. The following answers are given by the user:
(8.1) *Stockholder trades securities.*
Step 1: Neither (7.1) nor (7.2) can be projected to the CG of this sentence. Thus the answer does not contain required information and is *absolutely wrong.*
(8.2) *Primary market trades securities.*
Step 1: (7.1) can be projected to the CG of this sentence, but (7.2) can not. Thus the answer contains only part of required information and it is *incomplete.*
(8.3) *Financial market trades securities. However secondary market trades newly issued securities.*
Step 1: Both (7.1) and (7.2) can be projected to the CG of the first sentence. Therefore the answer contains required information and the process continues.
Step 2: The CG of the first sentence can be projected to the CG of (4.3.10), but the CG of the second sentence can not be projected to any of the CG from the

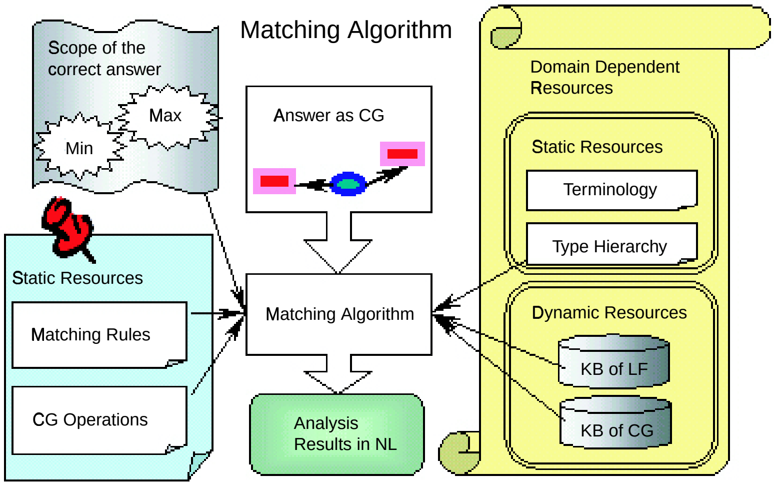


**Fig. 5.** CG-EST: Matching Algorithm

maximal expected answer CG set. This means that the answer contains additional wrong statements and it is only *partially correct*.
(8.4) *Financial market trades securities.*
Step 1: Both (7.1) and (7.2) can be projected to the CG of this sentence. Therefore the answer contains required information and the process continues.
Step 2: The CG of this sentence can be projected to the CG of (4.3.10), so the answer is *correct*.

## 7 Conclusion and Further Work

This paper presents an attempt for building a "more intelligent" self-tutoring approach to support eLearning paradigm. The main innovations of our approach are: *(i)* dynamically extraction of a CG KB from the text lessons; *(ii)* automatic generation of complicate questions from the KB by making some inference; *(iii)* distinguishing correct, partially correct, incomplete and wrong answers using formal CG processing. The system helps a tutor to easily create self testing materials for a user and assists the user in studying specific material. Unfortunately there is not known any other similar system to compare our results with it.

It is clear that such kind of systems can process only text in CE because in such a way we use more "formalized" sentences in English. The presented approach shows how only with some poor semantic information (type hierarchy and terms) KB of CG can be automatically constructed. `CG-EST` uses the results of GATE, therefore `CG-EST` performance depends on GATE's performance and the built-in GATE data corpus. However `CG-EST` can be improved by adding more rules in Static resource bank.

## References


1. Bloom, B.S. and Krathwohl, D.R., Taxonomy of Educational Objectives, Addison Wesley, NY, 1984.
2. Boytcheva, S., P. Dobrev and G. Angelova, CGExtract: Toward Extraction of Conceptual Graphs from Controlled English, In Contr. ICCS-2001, pp. 89-102
3. Boytcheva, S., ILP Techniques for Free-Text Processing, In Proc. AIMSA-2002, LNAI 2443, pp. 101-110
4. Boytcheva, S., A. Strupchanska and G. Angelova, Processing Negation in NL Interfaces to Knowledge Bases, In Proc. ICCS-2002, LNAI 2393, pp.137-150
5. Boytcheva, S., O. Kalaydjiev, A. Strupchanska and G. Angelova, Bewveen Language Correcness and Domain Answering on the World Wide Web, In Proc. RANLP-2001, pp. 40-46
6. Cunningaham, H., D. Mayard, K. Boncheva, V. Tab-lan, C. Ursu and M. Dimitrov (2002) "The GATE User Guide". http://gate.ac.uk/.
7. Fuchs, G. and R. Levinson, The CG Mars Lander, In Proc. ICCS-97, LNAI 1257, pp. 611-614.
8. Hovy, E., Gerber, L., Hermjakob, U., Junk, M., and Lin, C.-Y., Question Answering in Webclopedia. In Proc. TREC-9, NIST, Gaithersburg, MD
9. Kunichika, H., Katayama, T., Hrashima, T., and Takeuchi, A., Automated Question Generation Methods for Intelligent English Learning Systems and its Evaluation,In Proc. ICCE 2002, New Zeeland.
10. Levinson, R. Symmetry and the Computation of Conceptual Structures. Proc. ICCS-2000, Darmstadt, Germany, LNAI 1867, pp. 496-509.
11. Martin Ph.,P. Eklund, Large-Scale Cooperatively-Build KDs,ICCS-2001, LNAI 2120, pp. 231-244
12. Nicolas St., Mineau G. and B. Moulin, Extracting Conceptual Structures from English Texts Using a Lexical Ontology and a Grammatical Parser, In Contr. ICCS-2002, pp. 15-28
13. Parasite, http://www.co.umist.ac.uk/∼allan/index.html. Parasite is free for academic applications.
14. Rassinoux, A.-M., Baud, R. H. and J.-R. Scherrer. A Multilingual Analyser of Medical Texts, In Proc. ICCS'94, LNAI 835, pp. 84-96.
15. Schroeder, M. Knowledge Based Analysis of Radiology Reports using Conceptual Graphs, In Proc. 7th Annual Workshop: Conceptual Structures: Theory and Implementation, July 1992, LNAI 754.
16. Smilkstein, Rita, Acquiring Knowledge and Using It, Journal Gamut, Published by Seattle Community College District (Washington) 1993, pp. 16-17,41-43
17. Sowa, J. Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley, Reading, MA, 1984.
18. Sowa, J. and E. Way. Implementing a Semantic Interpreter using Conceptual Graphs, IBM Journal R&D, Vol. 30 (1), 1986, pp. 57-69.
19. Sowa, J. Using a Lexicon of Canonical Graphs in a Semantic Interpreter. Relational Models of the Lexicon, Cambridge University Press, 1988, pp. 113-137.
20. Velardi, P.,M. Pazienza and M. De'Giovanetti. Conceptual Graphs for the Analysis and Generation of Sentences. In: IBM J. Res. and Develop. Vol. 32 (2),March 1988, pp. 251-267.
21. Zhong L., Y. Yu, Learning o Generate CGs from Domain Specific Sentences, ICCS-2001, LNAI 2120, pp. 44-57