TEMPORAL LOGIC FOR MAN

Lukáš Žilka
Bachelor Degree Programme (3), FIT BUT
E-mail: xzilka07@stud.fit.vutbr.cz

Supervised by: Aleš Smrčka
E-mail: smrcka@fit.vutbr.cz

ABSTRACT

The paper deals with the automated translation of a natural language to temporal logic. Existing research attempts are summarized and built upon. For specificating the temporal properties a subset of English is introduced. The main contribution of the work is the proposed algorithm of translation of a property in the given language to LTL temporal logic, based on processing of and finding patterns in grammatical dependencies of the Stanford natural language parser. Future research directions are discussed at the end.

1 INTRODUCTION

Hardware and software development advances rapidly but verification of the developed systems, namely the use of formal methods, lacks wide spread due to its need of an experienced user. With those systems gradually becoming an indispensable part of our lives, everyone can experience the consequences of the development process that suffers from insufficient verification. As a result, the developers of these systems pay more attention to automatic verification and validation methods.

A lot of effort has been put into development of tools supporting the use of formal methods in practice. In general, the verification process constist of preprocessing the verified systems (e.g. creating a model of it, or a simulation of an environment) and specifying its properties. The process of processing the system and its subsequent verification are mature enough to be used in practice thus the last thing that prevents model checking from being widely used is the property definition process.

The properties are usually expressed by temporal logic formulas or assertions, that are verified on a model of the verified system. Formulas of temporal logic, in particular, offer great power but also pose a great difficulty for those who builds them. The developers should not have to know all about temporal logic when they want to express how their system is supposed to behave. This work proposes an automatic translation of a property specified in a natural language, English in particular, to LTL temporal logic.

The goal is to create a program that would translate the input sentence from English to the LTL temporal logic. The most widely used formalisms for specifying temporal properties are CTL and LTL. In this paper, I aimed at LTL since it is more intuitive and provides better error trace support than CTL. The program should also present to the user how it understood the input sentence and illustrate the resulting temporal formula in a time diagram. Additionally, the program's architecture should offer great extensibility so that the range of the input language can be broadened comfortably and without a change to the main program.

2 PREVIOUS WORK

The translation from a natural language to temporal logic has been the main topic of several papers already — I analyzed them and used in my research. Although each of them is very thorough and a high-quality work, their pragmatic use is due to lack of technical documentation generally unclear and impossible.

Natural language for hardware verification: semantic interpretation and model checking [3] was created as a part of the PROSPER project (which is very interesting but dead for almost ten years now). It goes through the process of creating a controlled language for the system and behavior specification. I used the information provided to create the limited input language (subset of English) appropriately and to design the mechanisms for its analysis.

Translating from a Natural Language to a Target Temporal Formalism [1] deals with the translation from English to temporal logic. It considers several temporal formalism, explains the pros and cons of each, and describes the process of their verification via the Kripke's structure. At the end the author illustrates problems that need to be faced when translating from English to temporal logic but does not propose any algorithmic solution.

Automatic Translation of Natural Language System Specifications Into Temporal Logic [4] is considering the translation of the natural language to temporal logic through DRS. The author claims to have created a tool that can translate English to temporal logic but its technical documentation or sources are nowhere to be found.

3 THE SUBSET OF NATURAL LANGUAGE

The English language has been chosen as the input language due to the fact that it is a language that the software and hardware developers are familiar with, and it is common and also simple enough to facilitate its computer processing. In addition, it has to be reduced to a so called *controlled English* (a subset of English) to allow for its algorithmic processing. To achieve that, I created a corpus of sentences from various specification documents to limit the language appropriately and for it to remain as natural for the developers as possible.

I used the iterative and hierarchical process for controlled language creation described in [2]. It creates a new language in each step. The first language is created by literal translation of the constitutive elements of LTL to English (e.g. "The signal WR or the signal RD are not set until the EN signal is set."). The second one is created by adding synonymous constructs to the first one (e.g. set = assert, write to one, ...). To the third language aliases are added to cover more complicated patterns (compositional mapping; e.g. "change on the signal A") and the fourth one contains even abstract language structures that do not necessarily describe behavior of signals but rather represent an abstract pattern of the behavior of signals (e.g. "Every request has to be eventually acknowledged.").

The basic version of the program must be able to process sentences from the first language because it has a full expressing power of LTL already, and it must be extensible.

4 THE TRANSLATION PROCESS

The whole process consists of a chain of a parser, analyzers and a synthesizer that interact with a database of predicates. Each of the analyzer gets a set of *grammatical dependencies* and the current state of the database of predicates as an input. It has five steps: (i) The input sentence (system specification in English) is firstly processed by the Stanford Parser which decomposes it into a set of *grammatical dependencies* (e.g. "subj(assert, A); nn(A, signal)"). (ii) The individual phrases in the sentence are separated by looking at the *subject dependency* relations. (iii) The phrases are classified according to their function in the sentence (e.g. "If the signal WREN is set" → *precondition*; "until the signal RD has been asserted" → *consequence*; "the line CLK is zero" → *main phrase*). (iv) The phrase-meaning-extraction analyzers are launched. The (iv) is the subject of future extension in order to be able to process a richer language. Finally, after each of the analyzers has finished, (v) the synthesizer is launched. It searches the database of predicates for predicates that are the constituents of temporal formulas and consequently puts them together, creating a temporal formula as a result.

5 CONCLUSION AND FUTURE DIRECTIONS

The whole field of automated translation of a natural language to temporal logic is vast and yet to be fully discovered. Although, I designed and I am in the process of implementation of a program that can translate a subset of English into temporal logic. The subset of English was created by taking the expressing capabilities of temporal logic and mapping them to English. The program also offers instruments through which the range of the input language can be broadened.

In the future I will aim at the customization of the Stanford Parser so that it can process HW/SW specification specific language structures that are not grammatically correct English structures but would be useful for the rest of the specifications translation process. Then I will try to engage ontology systems to enlarge the range of accepted language and aim at the processing of discourse of subsequent specification sentences.

REFERENCES

[1] R. Fazaeli. Translating from a Natural Language to a Target Temporal Formalism. Final Year Project, May 2002.

[2] C. Grover, A. Holt, E. Klein, and M. Moens. Designing a controlled language for interactive model checking. In *Proceedings of the Third International Workshop on Controlled Language Applications*, 2000.

[3] Alexander Holt, Ewan Klein, and Claire Grover. Natural language for hardware verification: semantic interpretation and model checking. In *Proc. ICoS-1: Inference in Computational Semantics, Amsterdam*. University of Amsterdam, 1999.

[4] R. Nelken and N. Francez. Automatic translation of natural language system specifications into temporal logic. In *8th International Computer Aided Verification Conference*, 1996.