REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
13-07-2005

2. REPORT TYPE
Final Report

3. DATES COVERED *(From – To)*
1 January 2004 - 10-Nov-05

4. TITLE AND SUBTITLE
Efficient Reasoning with Large Knowledge Bases

5a. CONTRACT NUMBER
FA8655-04-1-3007

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Professor Andrei Voronkov

5d. PROJECT NUMBER

5d. TASK NUMBER

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Manchester Informatics Ltd
Department of Computer Science
Oxford Rd
Manchester M13 9PL
United Kingdom

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
EOARD
PSC 802 BOX 14
FPO 09499-0014

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
Grant 04-3007

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This report results from a contract tasking Manchester Informatics Ltd as follows: The grantee will investigate relevance determination for reasoning on systems that contain more than 100,000 first-order axioms. Relevance determination refers to techniques for examining very large knowledge bases to distinguish between relevant, possibly relevant, and not relevant information. The best existing approaches (from other researchers) are unable to cope with knowledge bases of 10,000 axioms. He will investigate these techniques in two phases. First, at six-months he will deliver an extension to his existing system, Vampire, capable of resolving queries within seconds on knowledge bases over 30,000 axioms. By the conclusion of the research, he will improve his relevance filtering techniques to enable Vampire to reason on knowledge bases with over 100,000 axioms within seconds. Complete details described in the attached technical proposal. We tested the new strategy again to find out inconsistencies in SUMO 1.72 with row variables expanded to sequences of the length 50 (that is, a knowledge base with about 30,000 first-order axioms). When we used the negation of an axiom causing inconsistency as the query, inconsistency was always proved in less than one second. We believe query answering can be done much faster in less than 0.1 second. Our experiments discovered the following problem. When a knowledge base contains many similar atoms (e.g., ground facts with the instance predicate) just passing the knowledge base to Vampire's kernel may take over a second. After profiling, we have found out that the time is essentially spent not on query answering at all but on building some indexes. Indexes in Vampire were not designed with the aim of handling large signatures and should be reimplemented for experiments with ontologies. Moreover, we think that indexes should be pre-compiled rather than built by the kernel. However, this is a subject for a future research.

15. SUBJECT TERMS
EOARD, Automated Reasoning, Theorem Prover, Validating Compiler

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18, NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			PAUL LOSIEWICZ, Ph. D.
UNCLAS	UNCLAS	UNCLAS	UL		**19b. TELEPHONE NUMBER** *(Include area code)* +44 20 7514 4474

Grant 043007: Final Report

Andrei Voronkov

1 Quotations from the Application

Our aim is to push the current technology of query answering in large knowledge bases far beyond its current capabilities. We are going to implement a system able to answer in a few second queries to first-order knowledge bases containing over 100,000 axioms (with equality) while currently the best systems can usually cope only with about 1,000 such axioms.

After the first six months we will deliver an extension of Vampire by a relevance tester able to cope within seconds with knowledge bases over 30,000 axioms. After this we will make extensive experiments and improve our relevance filtering techniques to make it scale to knowledge bases over 100,000 axioms.

2 First Phase

In the first six months of the project we worked on embedding a relevance strategy into the inference mechanism of Vampire.

The result of the first phase was an intermediate version of the system. The improved relevance testing was achieved by introducing new techniques that allow for a more goal-oriented reasoning. They are mentioned below.

1. A new real-valued parameter `--nongoal_weight_coefficient` that puts a penalty on clauses not related to the goal.

2. A new version of the Knuth-Bendix ordering that tries to select symbol weights and precedence relation on symbols in such a way that top-down goal-oriented reasoning will be preferred to bottom-up reasoning.

Unfortunately, we could not test the system directly on knowledge bases with a huge number of axioms, since the largest currently available KIF-based ontology SUMO has about 5,000 axioms (but this number of axioms already makes it unmanageable for other systems). Nonetheless, we could obtain a knowledge base with 30,000 axioms using the row variables of SUMO. Row variables in SUMO can be expanded to a sequence of variables of an arbitrary length. By using expansion to sequences of the length up to 50 we obtained a knowledge base of the required size.

Note that this knowledge base has very complex statements abnormal for an ordinary knowledge base or ontology. For example, it contains relations of arity 50, while in a normal ontology most (if not all) of the relations have arity 1 or 2. To confirm that the knowledge base with such relations is very complex we give an example of an inconsistency proof found by Vampire in version 1.72 of SUMO in the enclosed file `inconsistency.vam`. This proof uses relations of large arities that are very difficult to handle for all systems.

We ran Vampire to check the knowledge base for consistency. The previous version of Vampire shipped to Teknowledge Inc. could not find inconsistencies in SUMO 1.72. The intermediate version running in the default mode could detect inconsistency in 53.5 seconds on a 2GHz PC. We could also find a mode in which inconsistency has been detected in 26.2 seconds.

3 Second Phase

In the second phase three subtasks have been implemented:

1. Extensive testing of the system using about 200 departmental computers. The purpose of the testing was
   (a) to check the best combinations of the new parameters of the system with the previously available ones;
   (b) to find out possible weaknesses of the system.
2. Providing using interface for working with knowledge bases, and in particular multiple knowledge bases.
3. Further research on improving relevance detection.

Let us note that by the time of the second part of the project (and independently of it) we implemented a completely new version of the Knuth-Bendix ordering (KBO) in which atoms are first compared using so-called *levels* (a level is an integer) of predicate symbols, and then, if two predicate symbols have the same level, using the ordinary Knuth-Bendix ordering.

We have implemented an algorithm that is based on the following idea. Let us call two symbols (function, predicate, or constant) *connected* if there exists a clause in the knowledge base that contains both symbols. For example, if the knowledge base contains an atom $p(a, b)$, then p, a, b are connected to each other. Define the *connection graph* of the knowledge base as the graph whose nodes are symbols occurring it the knowledge base and there is an edge between two nodes if and only if these nodes are connected. Call the *distance* between two symbols the distance between them in the connection graph.

We use the negation of a distance between symbols in the query and a symbol in the knowledge base as the level of this symbol. This means that literals containing the symbols "closer" to the query get selected with a higher probability.

The new literal ordering mechanism turned out to behave better than the previous one. However, we also discovered some problems with it. Essentially, ontologies may contain a large number of statements using the predicates `instance` and `subclass`.

Many symbols having nothing in common get connected in just two steps using these symbols. In the next experiment, we excluded these symbols from the connection graph. This made a drastic influence on the speed of query answering.

We tested the new strategy again to find out inconsistencies in SUMO 1.72 with row variables expanded to sequences of the length 50 (that is, a knowledge base with about 30,000 first-order axioms). When we used the negation of an axiom causing inconsistency as the query, inconsistency was always proved in less than one second.

Note. We believe query answering can be done much faster in less than 0.1 second. Our experiments discovered the following problem. When a knowledge base contains many similar atoms (e.g., ground facts with the `instance` predicate) just passing the knowledge base to Vampire's kernel may take over a second. After profiling, we have found out that the time is essentially spent not on query answering at all but on building some indexes. Indexes in Vampire were not designed with the aim of handling large signatures and should be reimplemented for experiments with ontologies. Moreover, we think that indexes should be pre-compiled rather than built by the kernel. However, this is a subject for a future research.

4 Other

The intermediate version of the system took part in the annual world cup in theorem proving CASC. As in the previous two years, the system has won the two main categories of the competition:

- MIX - consisting of arbitrary problems in clausal form;
- FOF - consisting of arbitrary problems in the first-order form.

Vampire was considerably better than the last year version (180:157 in MIX and 80:75 in FOF, in the number of solved problems). Moreover, it is the first time in the history of the competition that in FOF Vampire was better than all other systems, even if one runs them in parallel (80:76).

5 How to run Vampire

This version of Vampire can be run as explained in the enclosed document `kif_mode.pdf`. As an example, try

```
vampire < test.xml
```

Note that the file `test.xml` loads a knowledge base from the file `sumo.kif`. This file contains version 1.72 of the SUMO ontology.