Natural Proof Search and Proof Writing

Dominique Pastre
LIPADE - University Paris Descartes

July 2010

Abstract : MUSCADET is a knowledge-based theorem prover based on natural deduction. Its results show its complementarity with regard to resolution-based provers. This paper presents some MUSCADET results and points out some of its characteristics.

Keywords : automated theorem proving, natural deduction, knowledge-based system

1 Introduction

The first theorem provers used natural methods [Newel al 1957, Wang 1960]. Then most theorem provers were based on the resolution principle [Robinson 1965], which is a simple and very efficient method from a theoretical point of view.

For a long time, many theoretical results regarding resolution were published but the new strategies were not very satisfying from an practical perspective. Moreover, theorems were generally given to the provers as sets of clauses instead of first-order formulas. The language of clauses even became for some *the* language of automated proving.

On the other hand, other methods were also developed [Bledsoe 1977]. In particular the natural-deduction-like heuristic techniques of Bledsoe had some better results than the resolution techniques [Bledsoe 1971].

Since then, several theorem provers based on resolution were improved by many strategies and became powerful. At present few provers are based on natural deduction. One of them in particular, MUSCADET [Pastre 1989, Pastre 1993] which is a natural deduction knowledge-based system proved theorems that resolution-based provers could not prove, but failed to prove some other theorems easily proved by resolution-based provers.

Nowadays, the best generalist theorem provers are based on the resolution principle, as shown by the results of the CASC competitions [Sutcliffe 2006]. The winner of the FOF[1] division has been the resolution-based prover Vampire [Riazanov & Voronkov 2002] since several years.

It is however useful to continue to develop other systems for several reasons:

- If a prover has to communicate with humans, for example as a proof assistant in a mathematical research context or as a part of a tutoring system in education, proofs must be easily read by humans, which is not the case of a resolution proof.
- We can see that different provers based on different techniques may have complementary abilities even if some are in general better than others.

[1] First Order Formula

2 TPTP, CASC and Muscadet

The TPTP (Thousands of Problems for Theorem Provers [Sutcliffe 2009a]) Problem Library is a library of test problems for automated theorem proving (ATP) systems.

The CADE ATP System Competition (CASC [Sutcliffe 2006]) "is an annual evaluation of fully automatic, classical logic Automated Theorem Proving systems."

Muscadet [Pastre 1989, Pastre 1993, Pastre 2001] is a natural deduction knowledge-based prover. It is available under the new BSD license from
http://www.math-info.univ-paris5.fr/∼pastre/muscadet/muscadet.html

Although resolution-based provers are those which prove the largest number of the theorems of the TPTP library and of the theorems of the last CASC competitions, they are not able to prove some theorems which are proved by other provers, like Muscadet.

In 2007, the CASC organizers added a new contribution evaluation measure. "In addition to the ranking measures, the state-of-the-art (SOTA) contribution quantifies the unique abilities of the systems. For each problem solved by a system, its SOTA contribution for the problem is the inverse of the number of systems that solved the problem, and a systems overall SOTA contribution is the average SOTA contribution over the problems it solves." Muscadet had the highest SOTA (three problems were solved by only Muscadet in 2007, two in 2008)[2][Sutcliffe 2008, Sutcliffe 2009].

Some proofs of theorems which were also proved by only Muscadet in 2005 and 2006 may be found in [Pastre 2007]. Sutcliffe [Sutcliffe 2007] noticed the complementarity of the best prover Vampire and of Muscadet, lower ranked, and more recently [Sutcliffe 2009] the complementarity of Muscadet and Zenon [Bonichon & al 2007] which is a theorem prover based on a proof-confluent version of analytic tableaux.

As all provers are continually improved and the TPTP problems are regularly tested on all registered provers, one can see that the theorems which are cited above are now (2010) proved by Vampire or Infinox. Infinox [Claessen & Lillieström 2009] is an automated reasoning tool that can disprove the existence of finite models. It searches for function or predicate symbols with particular properties that imply the infinity, and subsequently uses an automated theorem prover to check if these properties hold. The prover used is E-prover [Schulz 2002] which is the second best prover participating to CASC.

Nevertheless, there are still some theorems that are proved by only Muscadet.
Here are some of them.
Mappings
$f(f^{-1}(Y) \subset Y$ (SET758+4)
Ordered and unordered pairs and cartesian products
$\{A\} \times \{B, C\} = \{(A, B)\ ,\ (A, C)\} \wedge \{A, B\} \times \{C\} = \{(A, C)\ ,\ (B, C)\}$ (SET895+1)
$A \times B \subset \mathcal{P}(\mathcal{P}(A \cup B))$ (SET952+1)
$(A \setminus B) \times C) = (A \times C) \setminus (B \times C)\ \wedge\ C \times (A \setminus B) = (C \times A) \setminus (C \times B)$ (SET972+1)
Ordinal numbers
The product of a nonempty set of ordinal numbers is an ordinal number (SET817+4).

We also see, looking at the results of the competition, that Muscadet is faster than other provers on the problems it is able to solve. In case of a success, the proof is obtained at least as quickly as with the others, and much more quickly in many cases. If it fails it often quickly stops itself. The cases of "timeout" are generally due to infinite creations of objects, or to too many sub-theorems.

[2] If $g_o f$ and $h_o g$ are one-to-one, then f is one-to-one (SET742+4).
$f(A \cup f^{-1}(B) = f(A) \cup B$ (SEU069+1)
$\mathcal{P}(A \cap B) = \mathcal{P}(A) \cap \mathcal{P}(B)$ (SET372+4)
$f(A \cup B) = f(A) \cup f(B)$ (SET752+4)
where f and g are functions and $\mathcal{P}(A)$ the power set of A.

3 Main characteristics of Muscadet

3.1 Facts, rules and metarules

Muscadet is a knowledge-based system. It works with facts to which it applies rules.

Facts are the conclusion to be proved, the hypotheses, the objects of a theorem or a sub-theorem to be proved, links to concepts which appeared in the initial conjecture or in the definitions of the preceding concepts, sub-theorems, definitions, axioms and lemmas, and all sorts of facts which give relevant information during the proof searching process. At the beginning, there is no hypothesis and the conclusion to be proved is the first-order formula of the initial conjecture.

Rules are written in the form

rule <name> : if <list of conditions> then <list of actions>

By applying rules, hypotheses are added, objects may be created, the conclusion may be replaced by another one, the theorem to be proved may be split into one or more sub-theorems to be proved, independent or not.

Some rules are general and express logical or mathematical knowledge and usual mathematical know-how.

Other rules are automatically built by metarules from the definitions and lemmas at the beginning of the proof. During the proof of a (sub)theorem, new rules may be dynamically built from universal hypotheses, they are local for this (sub)theorem.

Some actions are elementary, some other actions are more sophisticated and are defined by packs of rules,

3.2 Elimination of functional symbols

Strategies of Muscadet are designed to work with predicates rather than with functional symbols. In a formula with functional symbols, it "eliminates" them by giving names to the terms. These objects will replace these terms in the predicative formula. So, there remains no hypothesis or conclusion such as $p(f(a))$ but instead the hypothesis or conclusion $p(b)$ where b is a constant defined by the hypothesis $f(a) :: b$.

The symbol "::" is used to express that b is the object $f(a)$, and the formula $f(a) :: b$ will be handled as if it were a predicative formula.

The expressions are first transformed by using a new quantifier noted "!", which means

"for the only ... equal to ...".

$p(f(a))$ is replaced by $!A{::}f(a), p(A)$ which means

"for the only A equal to $f(a)$ then $p(A)$ holds"

where A is a variable.

This mechanism is recursive.

Then the expressions $!A$::<term>,<property> are handled by rules specific to hypotheses or to the conclusion, or to building rules from definitions.

3.3 Equality handling

Because of elimination of functional symbols, equality may occur in hypotheses only as equality of objects. Each time two objects of a (sub)theorem to be proved are found to be equal, one of them is replaced everywhere in the sub-theorem by the other, then it is removed, and so is the equality.

3.4 Negation handling

Muscadet works with positive properties as much as possible. Rules built from definition are made to work with positive properties rather than negative properties. Muscadet usually does not add negative properties, on the contrary there are general rules to eliminate negations in most cases.

3.5 Natural writing

With the first versions of Muscadet, the user had to extract the useful steps of the proof by hand. This is now done automatically. All the proof search steps are memorized as facts including all the elements which will be necessary to extract later the useful steps (the name of the executed action or applied rule, the new facts added and the antecedents, but also the rules which have been dynamically built and a brief explanation).

4 Why Muscadet is an efficient system in a number of circumstances

The reasons for the efficiency of Muscadet have been analysed in details with many examples in [Pastre 2007]. This includes
- the fact that the growth of the bases of facts in linear, not exponential ;
- the importance and efficiency of the chosen representations of the problem ;
- the splitting of a (sub)theorem in many subtheorems easier to prove, independent or not ;
- the treatment of functional symbols which flattens the handled expressions ;
- the replacement of definitions and universal hypotheses by natural and efficient rules ;
- the treatment of equalities and negations which removes them as far as possible.

5 Conclusion

Muscadet functions in a manner which is quite different from resolution-based provers. It uses methods based on natural deduction and is a knowledge-based system. Some of these methods are crucial and this explain why Muscadet is able to prove some theorems that resolution-based provers are not yet able to prove. Moreover, in cases where theorems are also proved by other provers, Muscadet proof can be obtained at least as fast as with the other provers and much faster in many cases.

However, Muscadet cannot prove some theorems that resolution-based provers can easily prove.

Muscadet is efficient for everyday mathematical problems which are expressed in a natural manner, for example in naive set theory. It is not efficient for problems which are defined axiomatically, from a logician's point of view, for instance in the fields of axiomatic geometry or axiomatic set theory.

Muscadet is efficient to solve problems which involve many axioms, definitions or lemmas. It is not efficient at all to solve problems which involve only one large conjecture and no intermediary definitions.

Improvements of Muscadet will not come from the increase of computer speed, but from the improvement of the heuristics which are still to be refined to enlarge the scope of situations that Muscadet could handle efficiently.

Another way to improve theorem proving is to have provers cooperate. Muscadet could first analyze the given problem and choose between searching itself for a proof or calling for a resolution-based prover or at least add some resolution-based techniques for subtheorems that it is not able to prove currently.

References

[Bledsoe 1971] W. W. Bledsoe, Splitting and reduction heuristics in automatic theorem proving, *Journal of Artificial Intelligence*, 2:55–77, 1971.

[Bledsoe 1977] W. W. Bledsoe, Non-resolution theorem proving, *Journal of Artificial Intelligence*, 9:1–35, 1977.

[Bonichon & al 2007] R. Bonichon, D. Delahaye and D. Doligez, Zenon: an extensible automated theorem prover producing checkable proofs, *Proceedings of Logic for Programming, Artificial Intelligence, and Reasoning* (LPAR 2007), Lecture Notes in Computer Science, Springer, 151-165, 2007

[Claessen & Lillieström 2009] K. Claessen, A. Lilliestr̈om Automated inference of finite unsatisfiability, CADE-22: *Proceedings of the 22nd International Conference on Automated Deduction*, Springer-Verlag, Montreal, 388-403, 2009, //http://gupea.ub.gu.se/handle/2077/22058

[Newel al 1957] A Newel, J.C. Shaw, and H.A. Simon. Empirical explorations with the logic theory machine: a case study in heuristics. *Proc. Western Joint Computer Conf.*, 1957; In: Feigenbaum and Feldman, editors, *Computers and thought*, 109–133, McGraw-Hill, 1963.

[Pastre 1989] D. Pastre, MUSCADET: an automatic theorem proving system using knowledge and metaknowledge in mathematics, *Journal of Artificial Intelligence*, 38(3):257–318, 1989.

[Pastre 1993] D. Pastre, Automated theorem proving in mathematics, *Annals on Artificial Intelligence and Mathematics*, 8(3-4):425–447, 1993

[Pastre 2001] D. Pastre, Muscadet2.3 : A knowledge-based theorem prover based on natural deduction, *International Joint Conference on Automated Reasoning - Conference on Automated Deduction*, 685–689, 2001,

[Pastre 2007] D. Pastre, Complementarity of a natural deduction knowledge-based prover and resolution-based provers in automated theorem proving, internal report, http://www.math-info.univ-paris5.fr/∼pastre/compl-NDKB-RB.pdf

[Riazanov & Voronkov 2002] A. Riazanov and A. Voronkov, The design and implementation of Vampire, *AI Communications*, 15(2-3):91-110, 2002, http://voronkov.com/vampire.cgi http://www.cs.miami.edu/ tptp/CASC/22/SystemDescriptions.html#Vampire—11.0

[Robinson 1965] J.A. Robinson, A machine oriented logic based on the resolution principle, *J.ACM* 12:23-41, 1965

[Schulz 2002] S. Schulz. E: A Brainiac theorem prover, *AI Communications*, 15(2-3):111-126,2002 http://www.eprover.org/

[Sutcliffe 2006] G. Sutcliffe, C. Suttner, The state of CASC, *AI Communications*,19(1):35-48, 2006

[Sutcliffe 2007] G. Sutcliffe, The 3th IJCAR automated theorem proving competition, *AI Communications*, 20:117-126,2007

[Sutcliffe 2008] G. Sutcliffe, The CADE-21 automated theorem proving competition, *AI Communications*, 21:71:81,2008,

[Sutcliffe 2009] G. Sutcliffe, The 4th IJCAR automated theorem proving competition - CASC-J4, *AI Communications*, 22:59-72,2009,

[Sutcliffe 2009a] G. Sutcliffe, The TPTP problem library and associated infrastructure: the FOF and CNF parts, v3.5.0, *Journal of Automated Reasoning*, 43(4):337-362, 2009 http://www.cs.miami.edu/∼tptp

[Wang 1960] H. Wang. Towards mechanical mathematics, *IBM J.Res.Develop*, 4, 2–22, 1960.