Proof Score Method in CafeOBJ

Kokichi FUTATSUGI

Japan Advanced Institute of Science and Technology
kokichi@jaist.ac.jp
www.jaist.ac.jp/~kokichi

If a specification is expressed as a set of equations and the equations can be used as rewriting rules for getting a simplest form of a given expression, validity of a statement about the specification can be checked by getting a simplest form, which is considered to be an answer, of a boolean expression for the statement. A term **reduction** is used to denote simplification of an expression by rewriting rules.

If a sequence of reductions can be designed to describe a reasoning process for justifying a statement about a specification (a set of equations), and every result of a reduction in the sequence is as expected, the sequence of reductions (and some explanations attached to them) can be used as a proof of the statement. Preparations of this kind of sequences of reductions for "doing proofs about specifications" has been done by OBJ/CafeOBJ users for more than 20 years. Most of them have been a small scale ones for explaining important properties of specifications in OBJ/CafeOBJ by using OBJ/CafeOBJ themselves. An OBJ/CafeOBJ text which is prepared for doing reductions for a specific proof is called **proof score**.

From around 1997, researchers of the CafeOBJ group at JAIST[1] started to extend the proof score method in the directions such as (1) to make the method applicable to distributed and real-time systems, such as classical distributed (and/or real-time) algorithm, railway signal systems, secure protocols, etc, (2) to make the method applicable to practical size problems, and (3) to automate the method. Several achievements have been done, and the proof score method using the CafeOBJ reduction (rewriting) engine has been recognized as a promising way of doing serious proofs about specifications written in CafeOBJ. We are currently intending to use the term proof score to cover more wider concept[2] than just the proof scores in current CafeOBJ. However, many non-trivial proof scores have been written in CafeOBJ for the first time, and the proof scores in CafeOBJ are the most important instances of the proof score at this moment.

The basic principle of proof scores in CafeOBJ can be said as (1) high level planning of a proof is done by user (human) and is described as a sequence of reductions and (2) low level mechanical calculations for the proof are coded into the reductions and should be done automatically. This implies that proof score does not aim at full automation of proofs but aim at the best combination of human and machine capabilities.

Evolution of Proof Scores in CafeOBJ

CAFE project funded by Japanese Government took place from April of 1996 to March of 1998[3]. In this project the current version of the CafeOBJ language and system is designed and implemented. Sufficiently reliable and fast implementation of the CafeOBJ system was available in the second half of the year 1997, and it was used to write several kinds of formal specifications from standard data types, distributed algorithms, railway signal systems, program language interpreters, semantics of programming languages, communication protocols, secure protocols, etc. The current form of proof scores in CafeOBJ is a result of these example writing activities.

The CafeOBJ language and system are also used in lectures at JAIST from 1997, however in 1997 the CafeOBJ system (interpreter) was still under development and the purpose of using the system in the lecture is only to understand the language and system as a part of the study of formal methods. From 1998 the examples written in CafeOBJ including proof scores were used as an important part of lectures on formal methods.

It can be said that the sufficiently reliable and fast implementation of the CafeOBJ interpreter is an important basic factor for writing many specifications and proof scores which congributes to the evolution of proof scores.

From static to dynamic systems

Several ideally well done proof scores for data types are known as folklore in OBJ users already in 80's[4]. They are doing proofs beautifully using reductions for showing induction bases and induction steps based on term structures of initial term algebras. These examples includes proofs of (1) associativity and commutativity of plus operations over Peano natural numbers and (2) the identity $n \times (n+1) = 2 \times (1+2+\cdots+n)$ for any natural number n.

These examples were models for writing proof score in CafeOBJ, and several proof scores of the same nature were written for the proofs including (1) equivalences of functions over natural numbers, (2) equivalences of functions over lists, (3) correctness of simple compilers from expressions to machine codes for stack machines, etc. These proof scores realized almost ideal combination of high level planning and mechanical reductions. However, even in this class of problems, there are some problems which require non-trivial lemma discoveries and/or case splittings.

Dynamic systems (or systems with states) are common in network/computer based systems, but there has not been established model and methodology in algebraic specification languages for coping with this class of problems. The CafeOBJ language is designed for writing formal specifications of dynamic systems based of hidden algebra semantics[5, 6]. Many attempts of writing specifications and proof scores for several kinds of dynamic systems have been done using CafeOBJ based on hidden algebra semantics, and **OTS** (Obsevational Transition Systems) has been recognized as a most promising model. The OTS corresponds to a restricted class of hidden algebras, and has a following nice properties: (1) it is possible to write specifications for OTS in a fixed standard style in CafeOBJ, this makes developments of the specifications easy, (2) this OTS style in CafeOBJ also helps to write proof scores since case splitting can be hinted by the specifications in this style.

The followings are some noticeable publications which show stages in the evolution of proof scores for dynamic systems.

- An attempt to specify and verify (with proof scores) mutual exclusion algorithms by incorporating the UNITY model into CafeOBJ: [7]
- Introduction of a primitive version of OTS: [8]
- Introduction of real-time features into OTS/CafeOBJ and accompanying developments of proof scores: [9, 10]
- A proper introduction of OTS/CafeOBJ and a related proof score writing method: [11, 12]
- Examples of verifications with proof scores in OTS/CafeOBJ: [13–15] (not all)

From explaining to doing proofs

Another major factor distinguishing stages of evolution of proof scores is the extent of automation for necessary reasoning steps by reductions of CafeOBJ. This is the most important direction of evolution of proof scores, although full automation of a proof is not the goal of the proof score method. Automation by a reduction is better to be intended for a mechanical calculation with a focused role and a clear meaning in a context of a whole reasoning process. It is not necessary be a right thing to classify the proof scores with respect to the degree of automation, but it is a good way to analyze what we have done until now.

In an early phase, the proof scores in CafeOBJ were restricted to subsidiary uses for assisting a part of verification by doing reductions for showing necessary logical statements for a specification. As several techniques for writing proof scores are developed, it is gradually intended to codify as many logical statements as possible into reductions in CafeOBJ. Recently, as one of ultimate point of this direction, a full automatic (i.e. algorithmic) verification method for a subset of OTS is developed. This algorithm can be seen as an unification of several techniques for proof scores in OTS.

The followings are several noteworthy publications which shows stages of automation of proof scores.

- Using CafeOBJ mainly for writing formal specification and subsidiary for proof score: [7]
- Reporting examples with sufficiently complete proof scores: [13–15] (not all)
- Another kind of attempt for automation of proof scores: [16]
- A full automatic (algorithmic) method of verification for OTS: [17]

Future issues

Proof scores have high potential for providing a practical new way of doing proofs for specifications in CafeOBJ (or in other algebraic specification languages). The followings are important issues for future research for making the proof score method more efficient.

- Introducing interactions into the Creme algorithm[17] for guiding high level planning by users while keeping other parts including reductions automatic.
- Farther development of the TATAMI/KUMO project of University of California, San Diego[18] (done as a subproject of the CAFE project) to realize a web (or hypertext) based proof score writing environment.

References

1. CafeOBJ: CafeOBJ web page. `http://www.ldl.jaist.ac.jp/cafeobj/` (2005)
2. Futatsugi, K., Goguen, J., Ogata, K.: Theory and practice of proof scores. in preparation (2005)
3. Futatsugi, K., Nakagawa, A.T.: An overview of CAFE specification environment - an algebraic approach for creating, verifying, and maintaining formal specifications over networks. In: Proc. of 1st International Conference on Formal Engineering Methods (ICFEM '97), November 12-14, 1997, Hiroshima, JAPAN, IEEE (1997) 170–182

4. Goguen, J., Winkler, T., Meseguer, J., Futatsugi, K., Jouannaud, J.P.: Introducing OBJ. Technical Report SRI-CSL-92-03, SRI International, Computer Science Laboratory (1992)
5. Diaconescu, R., Futatsugi, K.: CafeOBJ report: The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification. AMAST Series in Computing, 6. World Scientific, Singapore (1998)
6. Futatsugi, K.: Formal methods in CafeOBJ. In Hu, Z., Rodríguez-Artalejo, M., eds.: FLOPS. Volume 2441 of Lecture Notes in Computer Science., Springer (2002) 1–20
7. Ogata, K., Futatsugi, K.: Specification and verification of some classical mutual exclusion algorithms with CafeOBJ. In: Proceedings of OBJ/CafeOBJ/Maude Workshop at Formal Methods '99, Theta, (1999) 159–177
8. Ogata, K., Futatsugi, K.: Specifying and verifying a railroad crossing with CafeOBJ. In: Proceedings of the 6th International Workshop on Formal Methods for Parallel Programming: Theory and Applications (6th FMPPTA); Part of Proceedings of the 15th IPDPS, IEEE Computer Society Press (2001) 150
9. Ogata, K., Futatsugi, K.: Modeling and verification of distributed real-time systems based on CafeOBJ. In: Proceedings of the 16th International Conference on Automated Software Engineering (16th ASE), IEEE Computer Society Press (2001) 185–192
10. Futatsugi, K., Ogata, K.: Rewriting can verify distributed real-time systems. In: Proc. of International Symposium on Rewriting, Proof, and Computation (PRC2001), Tohoku Univ. (2001) 60–79
11. Ogata, K., Futatsugi, K.: Rewriting-based verification of authentication protocols. In: Proceedings of the 4th International Workshop on Rewriting Logic and its Applications (4th WRLA). ENTCS 71, Elsevier (2002)
12. Ogata, K., Futatsugi, K.: Proof scores in the OTS/CafeOBJ method. In: Proceedings of the 6th IFIP WG6.1 International Conference on Formal Methods for Open Object-Based Distributed Systems (6th FMOODS). LNCS 2884, Springer (2003) 170–184
13. Ogata, K., Futatsugi, K.: Formal analysis of the iKP electronic payment protocols. In: Proceedings of the 1st International Symposium on Software Security (ISSS2002). LNCS 2609, Springer (2003) 441–460
14. Ogata, K., Futatsugi, K.: Formal verification of the Horn-Preneel micropayment protocol. In: Proceedings of the 4th International Conference on Verification, Model Checking, and Abstract Interpretation (4th VMCAI). LNCS 2575, Springer (2003) 238–252
15. Ogata, K., Futatsugi, K.: Equational approach to formal verification of SET. In: Proceedings of the 4th International Conference on Quality Software (4th QSIC), IEEE Computer Society Press (2004) 50–59
16. Mori, A., Futatsugi, K.: CafeOBJ as a tool for behavioral system verification. In Okada, M., Pierce, B.C., Scedrov, A., Tokuda, H., Yonezawa, A., eds.: ISSS. Volume 2609 of Lecture Notes in Computer Science., Springer (2002) 461–470
17. Nakano, M., Ogata, K., Nakamura, M., Futatsugi, K.: Automating invariant verification of behavioral specifications. submitted for publication (2005)
18. Goguen, J.A., Lin, K., Mori, A., Rosu, G., Sato, A.: Distributed cooperative formal methods tools. In: Proc. of 1997 International Conference on Automated Software Engineering (ASE '97), November 02-05, 1997, Lake Tahoe, CA, IEEE (1997) 55–62