学 位 論 文 内 容 の 要 旨

博士の専攻分野の名称　　博士（情報科学）　　氏名　季　承成

学 位 論 文 題 名

Automated Lemma Generation and Multi-Context Schemes for Rewriting Induction
（書換え帰納法のための補題自動生成と多重文脈スキーム）

In computer science, algebraic structure often appears in various theories as a means of representing systems in a formal way and reasoning about their properties such as their correctness. In the field of software engineering, the related technologies include term rewriting, equational reasoning, and their application to formal methods. An algebraic inductive theorem is a proposition for algebraic specifications defined on inductively-defined data structures such as natural numbers and lists. The proof of such inductive theorems plays a fundamental role in the field of formal verification of information systems. Term rewriting induction (RI) is an automatable proof principle for proving inductive theorems on term rewriting systems. Given a ground-reducible and convergent (terminating and confluent) term rewriting system (TRS), a reduction order, and a set of target inductive theorems or related lemmas, RI tries to conduct a series of inductive proof of target theorems. During the RI procedure, divergence often occurs which stops RI from getting a successful proof. Appropriate lemmas are effective to this problem, but manually supplying such lemmas is difficult to general users or researchers.

In this dissertation, I proposed a new automated lemma generation method called *peripheral sculpture* which can be described in the following steps: 1) detect a potential divergence from a sequence of generated conjectures; 2) generate candidate lemmas by calculating the peripheral sculptures in the diverging sequence. Our method greatly improves the practical usefulness to make the theorem prover more powerful without introducing significant cost increase. RI procedure has several kinds of strategic issues in constructing successful proofs: 1) which reduction order should be applied; 2) which (axiomatic or hypothetical) rules should be applied during rewriting; 3) which variables should be instantiated for induction. In order to solve these issues, I developed multi-context schemes for RI and I also proposed new automated lemma generation methods for such multi-context schemes. In the experiments, we have succeeded in solving several lemma-required benchmark problems which encountered complex differences (i.e., parallel and nested differences) and which the original systems could not solve.

In Chapter 1, we introduce the background of our research which, in the field of Artificial Intelligence, is a subject including term rewriting, equational reasoning, and their application to formal methods.

In Chapter 2, we introduced the basis of term rewriting systems and two term rewriting system based automated reasoning systems. Starting from a well-known completion procedure, I describe the details of rewriting induction procedures.

In Chapter 3, we start with an unsound automated lemma generation method of divergence detection, then propose a new automated lemma generation method, *peripheral sculpture*, to prove inductive theorems which cannot be proved otherwise. We introduce the formal definitions and the inference rules.

In Chapter 4, we review two automated reasoning systems with multi-context scheme. Then we extend our method in the multi-context scheme to develop a practically useful framework featuring multi-context postulations which strengthen flexibility and effectiveness for inductive theorem provers.

In Chapter 5, we discuss the lazy evaluation technique and the other details from a point of view of software engineering which includes system design, construction and optimization of our implementation of multi-context reasoning systems.

In Chapter 6, we discuss the results of our experiments. These results are pleasing in the sense that 1) the newly proposed lemma generation method worked well in combination with other methods in the multi-context framework and solved some lemma-required problems that could not be solved by the previous methods; 2) the new implementations turned out to be more effective than the previous ones.

We give conclusions and future research topics in Chapter 7.