Efficient General Unification for XOR with Homomorphism

Zhiqiang Liu and Christopher Lynch*

Clarkson University, Potsdam, NY 13699, USA
{liuzh,clynch}@clarkson.edu

Abstract. General E-unification is an important tool in cryptographic protocol analysis, where the equational theory E represents properties of the cryptographic algorithm, and uninterpreted function symbols represent other functions. Some important properties are XOR, Abelian groups, and homomorphisms over them. Polynomial time algorithms exist for unification in those theories. However, the general E-unification problem in these theories is NP-complete, and existing algorithms are highly nondeterministic. We give a mostly deterministic set of inference rules for solving general E-unification modulo XOR with (or without) a homomorphism, and prove that it is sound, complete and terminating. These inference rules have been implemented in Maude, and are being incorporated into the Maude NPA. They are designed in such a way so that they can be extended to an Abelian group with a homomorphism.

1 Introduction

In symbolic cryptographic protocol analysis, messages are represented as terms. Actions of principals involved in the protocol are represented with rules, indicating that if a principal receives a message with a given pattern then the principal will send out a message based on the message received. Abilities of malicious intruders are represented by rules indicating how an intruder can manipulate data, where variables in the pattern indicate that the principal will accept any message of that type. A goal state represents an attack, and an analyzer decides whether the goal state is reachable. Generally, the analysis involves working back from the goal state to initial states. If this is possible, then an attack exists. Initial methods of cryptographic protocol analysis were based on the free algebra model[8]. In this method of analysis, two messages are the same only if they are represented by the same term. In this case, during the search back from the goal, a message pattern representing a received message will be compared against a message pattern representing a sent message. Syntactic unification is used to compare them against each other and find the intersection of the patterns.

However, the free algebra model is not precise enough to model properties of cryptographic algorithms[6]. For example, cryptographic algorithms may involve XOR operations, and therefore two messages may be equivalent in the theory

* Both authors are supported by NSF Grant CNS 09-05378.

N. Bjørner and V. Sofronie-Stokkermans (Eds.): CADE 2011, LNAI 6803, pp. 407–421, 2011.

of XOR but not syntactically equivalent. Abelian groups are also important, because they can model products, such as the product of exponents in Diffie Hellman. Another common property of cryptographic algorithms is a homomorphisms over an XOR or an Abelian group operator. For example, RSA, has the property ${m_1}^e {m_2}^e = (m_1 m_2)^e$, where raising to the power of e is a homomorphism, and the product of the messages forms an Abelian group. Unfortunately, the free algebra approach fails to detect attacks in protocols using cryptographic algorithms with these properties. Therefore, to conduct a more precise analysis, unification must be performed modulo these equational theories.

In conclusion, unification algorithms for the theory of XOR (with homomorphism) and Abelian groups (with homomorphism) are essential for cryptographic protocol analysis. It is important that these algorithms are efficient. Efficient unification algorithms have been developed for these theories[4,7]. However, cryptographic protocol analysis also must deal with uninterpreted function symbols. So it is important to have unification algorithms for these theories in combination with uninterpreted function symbols. When uninterpreted function symbols occur in combination with these theories, the complete set of unifiers is not always a singleton, but it is finite. It is crucial that the unification algorithm creates a complete set of unifiers that is as small as possible. If this set is too large, the search space of searching for an attack quickly blows up and cryptographic protocol analysis becomes infeasible.

The goal then is to build equational unification algorithms for these theories that are both efficient and create a small complete set of unifiers. There are two standard techniques for dealing with these equational theories in combination with uninterpreted function symbols. Let us consider the theory of exclusive OR in particular, because the other theories suffer from the same issues. One way to deal with XOR with uninterpreted function symbols is to create an efficient algorithm to solve XOR unification, and an efficient syntactic unification algorithm for uninterpreted function symbols and then to apply a standard combination algorithm to combine the theories[1]. The second technique is to create a convergent equational theory and apply Narrowing to solve the unification problem[5]. Both of these methods are highly nondeterministic. They are not very efficient in practice, but worse they build a highly redundant complete set of unifiers.

In this paper, we try to overcome these problems by devising a set of inference rules that is simple, easy to implement, very deterministic in practice, and produces a small complete set of unifiers. We can compare our work to [10]. That work is based on the combination method, and also has the goal of an efficient unification algorithm for XOR unification. We think our inference rules are simpler and easily extended to other equational theories.

We have developed a sound, complete and terminating set of inference rules for XOR with homomorphism, along with uninterpreted function symbols. We have implemented our inference rules in Maude[3], and they are being incorporated into the NRL protocol analyzer[6]. These inference rules also apply to XOR without homomorphism. We have designed them in such a way that they can be extended to Abelian groups.