Intruder Deduction for AC-like Equational Theories with Homomorphisms

Pascal Lafourcade, Denis Lugiez, and Ralf Treinen

19th November 2004

Addresses of the authors:

Pascal Lafourcade
Laboratoire Spécification et Vérification
Ecole Normale Supérieure de Cachan & CNRS UMR 8643
61, avenue du Président Wilson
F- 94235 Cachan, France
Email: `lafourca@lsv.ens-cachan.fr`
`http://www.lsv.ens-cachan.fr/~lafourca`

Denis Lugiez
Laboratoire d'Informatique Fondamentale
Université Aix-Marseille 1 & CNRS UMR 6166
39, rue Joliot-Curie
F-13453 Marseille Cedex 13, France
Email: `lugiez@cmi.univ-mrs.fr`
`http://www.cmi.univ-mrs.fr/~lugiez/`

Ralf Treinen
Laboratoire Spécification et Vérification
Ecole Normale Supérieure de Cachan & CNRS UMR 8643
61, avenue du Président Wilson
F- 94235 Cachan, France
Email: `treinen@lsv.ens-cachan.fr`
`http://www.lsv.ens-cachan.fr/~treinen`

This work was partially supported by the research programs ACI-SI Rossignol, and RNTL PROUVÉ (n° 03 V 360).

Abstract

Cryptographic protocols are small programs which involve a high level of concurrency and that are difficult to analyze by hand. For instance, a flaw in the infamous Needham-Schroeder public key authentication protocol (consisting of three simple message exchanges) was only discovered 17 years after its first publication. Therefore, the need for formal methods to achieve this analysis has been recognized and many approaches to model and analyze cryptographic protocols have been proposed. The most successful methods rely on rewriting techniques and automated deduction in order to implement or mimic the process calculus describing the protocol execution. The first existing tools and results implemented the cryptographic primitives as black-boxes and did not consider the algebraic properties of the operations that are actually used in the real implementations of protocols. For instance, the DES and the AES encryption methods make extensive use of the *exclusive or* operation; other protocols rely on homomorphic properties of modular exponentiation. Therefore, the formal approach has been extended to handle algebraic properties. Several extensions of the formal approach exist, among others, for the *exclusive or*, a restricted notion of modular exponentiation, and for Abelian groups.

The classical approach of Dolev and Yao consists in modeling the capabilities of an intruder by a deduction system. The analysis of this deduction system relies on the notion of a *local theory*, i.e. a theory where a simplest proof that a term t is deducible from a set of terms T can consist only of subterms of T and t. The problem whether an intruder can gain a certain information t from a set of initial knowledge T, that is in this formalization whether there is a proof of t from T, is called the *intruder deduction problem*. The deduction system contains rules describing cryptographic primitives (for instance knowing a key K, a term t that is the encryption of s by K, one can deduce s) and possibly axioms describing the algebraic properties of the relevant operations. A further problem, which lies beyond the intruder deduction problem, is to design an algorithm for constraint satisfaction that relies on the intruder deduction procedure.

In this paper we focus on the intruder deduction problem in presence of several variants of AC-like axioms (from AC to Abelian groups, including the theory of *exclusive or*) and homomorphism that are the most frequent axioms arising in cryptographic protocols. The first issue is to prove the decidability of the intruder deduction problem in these cases, the second issue is to have a good complexity for the decision procedure. Solutions are known for the cases of *exclusive or* and of Abelian groups alone. In this paper, we address the combination of these theories with a law of homomorphism which leads to much more complex decision problems.

We have been able to prove decidability of the intruder deduction problem in all cases considered. Our decision procedure is in EXPTIME, except for a restricted case in which we have been able to get a PTIME decision procedure using a property of one-counter automata.

Contents

1 Introduction **4**

2 Preliminaries **8**
2.1 Terms . . . 8
2.2 Equations and Rewriting Systems . . . 8
2.3 Miscellaneous . . . 9

3 A Dolev-Yao Model for AC-like Equational Theories **10**
3.1 The Dolev-Yao Model Extended by Equational Reasoning . . . 11
3.2 The Dolev-Yao Model Extended by Rewriting . . . 12
3.3 The Equational Theory ACh . . . 14
3.3.1 Equational Theory . . . 14
3.3.2 Rewrite Presentation . . . 14
3.3.3 Local Confluence . . . 14
3.3.4 Termination . . . 14
3.4 The Equational Theory ACUNh . . . 14
3.4.1 Equational Theory . . . 15
3.4.2 Orientation of the Axioms . . . 15
3.4.3 Local Confluence . . . 15
3.4.4 Rewrite Presentation . . . 15
3.4.5 Termination . . . 16
3.5 The Equational Theory AGh . . . 16
3.5.1 Equational Theory . . . 16
3.5.2 Orientation of the Axioms . . . 16
3.5.3 Local Confluence . . . 17
3.5.4 Rewrite Presentation . . . 17
3.5.5 Termination . . . 18

4 Locality and Complexity of the Intruder Deduction Problem **19**
4.1 A Generalization of Locality . . . 19
4.2 One-Step Deducibility for AC-like Theories . . . 20

5 Locality for AC-Like Equational Theories with Homomorphism **24**
5.1 Preliminary Definitions and Properties . . . 24
5.1.1 Vocabulary . . . 24
5.1.2 Different Kinds of Proofs . . . 25
5.1.3 Different Kinds of Subterm . . . 26
5.1.4 Properties and Lemmata . . . 29
5.2 Locality for the Equational Theory ACh . . . 32
5.2.1 Observations and Technical Lemmata for the Theory ACh . . . 32
5.2.2 Lemmata and Locality . . . 33
5.3 Locality for the Equational Theory $ACUNh$ . . . 36

5.3.1 Properties of $\oplus$-*lazy* Proofs in the *ACUNh* Case 36
5.3.2 Locality in the Binary Case 40
5.3.3 Locality in the General Case 45
5.4 Locality for the Equational Theory AGh 46
5.4.1 Some New Definitions 46
5.4.2 Some Lemmata and Properties in the AGh Case 47
5.4.3 Locality in the Binary Case 50
5.4.4 Locality in the General Case 52

6 Locality in Case of a Homomorphic Encryption Operation 54
6.1 Locality in the ACh Case 54
6.2 Locality in the ACUNh Case 56
6.2.1 Properties of $\oplus$-*lazy* Proofs in the ACUNh Case 56
6.2.2 Locality in the Binary Case 56
6.2.3 Locality in the General Case 56
6.3 Locality in the Case of Abelian Groups 57
6.3.1 Some New Definitions 57
6.3.2 Some Lemmata and Properties in the AGh Case 57
6.3.3 Locality in the Binary Case 57
6.3.4 Locality in the General Case 58

7 Conclusion 59

8 Appendix 61
8.1 One-Counter Automata 61
8.2 Pushdown Automata 63

Chapter 1

Introduction

Automatic Verification of Cryptographic Protocols in Presence of Equational Axioms. Cryptographic protocols are ubiquitous in distributed computing applications. They are employed for instance in Internet banking, video on demand services, wireless communication, or secure UNIX services like `ssh` or `scp`. Cryptographic protocols can be described as relatively simply programs which are executed in an untrusted environment. These protocols use cryptographic primitives in order to implement symmetric (shared-key) encryption, and asymmetric (public-key) encryption and signatures. Often both symmetric and asymmetric encryption are used in combination by a single protocol. The `ssh` protocol, for instance, uses in a first phase asymmetric encryption in order to have the two protocol participants agree on a shared key which is then used to encrypt subsequent data flows using symmetric encryption (which is much more efficient than asymmetric encryption).

Contrary to the classical verification problem, the question is here not the correct execution of usually large and complex programs in a trusted environment, but the correct execution of a usually simple program in a hostile environment. Ross Anderson and Roger Needham [AN95] have paraphrased the problem of protocol design as the problem of "programming Satan's Computer". Verifying protocols is notoriously difficult, and even very simple protocols which look completely harmless may have serious security flaws, as it was dramatically demonstrated by the bug of the Needham-Schroeder protocol found by Lowe [Low95] using a model-checking tool. The protocol was published since 17 years and widely believed secure before the flaw, a so-called *man in the middle attack*, was found.

There are many different protocols published in the literature. An overview of *authentication protocols* known a decade ago can be found in [CJ97], a more recent data base of protocols and known flaws is [Jac]. These protocols are often implemented in small variants which differ from the originally proposed protocol, or are used in combination with other protocols. This raises the need of *automatic* tools for protocol verification.

There are different approaches to modeling cryptographic protocols and analyzing their security properties: process calculi like the *spi-calculus* [AG99], cryptographic proofs (see, for instance, [AR00]) which consist in reducing the existence of an attack against a protocol using certain cryptographic primitives to cryptanalytic attacks against the cryptographic primitives, and the so-called approach of Dolev and Yao [DY83] which consists in modeling an attacker by a deduction system. This deduction system specifies how the attacker can obtain new information from previous knowledge, which he has either obtained by silently eavesdropping the communication between honest protocol participants (in case of a *passive* attacker), or by eavesdropping and fraudulently emitting messages, thus provoking honest protocol participants to reply according to the protocol rules (this is the case of a so-called *active* attacker). We call *intruder deduction problem* the question whether a passive eavesdropper can obtain a certain information from knowledge that he observes on the network.

The first two approaches to model cryptographic protocols, process calculi and cryptographic proofs, lead to verification problems which are very hard to automatize. The Dolev-Yao approach, however, lends itself to automatization since the question whether the intruder can obtain a certain information now reduces to the question whether this information can be deduced using a certain deduction system. On the other hand, a drawback of this approach is its limitation to so-called trace properties of protocols, such as the question whether the attacker can gain knowledge of a certain information (the so-called *security problem*).

Classically, the verification of cryptographic protocols was based on a black-box view of cryptographic primitives. That is, verification of cryptographic protocols adopted the so-called *perfect cryptography assumption* which states that it is impossible to obtain any information about an encrypted message without knowing the exact key necessary to decrypt this message. This assumption allowed a separation of verification tasks into proving lower bounds for the cryptanalysis of the cryptographic primitives on the one hand, and verification of a distributed program on the other hand. Unfortunately, this perfect cryptography assumption has proven too idealistic since there are protocols which can be proven secure under the perfect cryptography assumption, but which are in reality insecure since an attacker can use properties of the cryptographic primitives in combination with the protocol rules in order to obtain knowledge of a secret. These properties are typically expressed as equational axioms (so-called algebraic properties), like for instance associativity and commutativity of certain operators. Algebraic properties may be essential for the executability of the protocol, or may just come into play because for some reason cryptographic primitives with these properties are employed. A recent overview of algebraic properties of cryptographic primitives, their use to mount attacks on protocols, and existing results on verification of cryptographic protocols in presence of equational axioms can be found in [CDL04].

A number of results have been obtained, both for the intruder deduction problem and for the preservation of secrecy under active attacks[1]. We here only mention some results which are of particular relevance to the problems studied in this report: the intruder deduction problem is decidable [CLS03] in polynomial time [CKRT03] in case of the equational axioms of *exclusive or*, and decidable [CLS03] in polynomial time [Tur03] in case of the equational axioms of Abelian groups[2]. Likewise, the intruder deduction problem is decidable in polynomial time [CLT03] in the case of the equational theory of an homomorphism. Note that the two equational theories of *exclusive or* and of homomorphism model basic properties of important cryptographic primitives:

- *exclusive or* is a basic building block in many symmetric encryption methods (for instance DES or the more recent AES) or even used directly as an encryption method;
- homomorphisms are ubiquitous in cryptography since many asymmetric encryption methods are based on modular arithmetic. Furthermore, symmetric encryption methods which often work on data blocks of fixed size are in the simplest of cases (the so-called *electronic codebook mode*) homomorphically extended to data streams of arbitrary size.

An Example of an Attack against a Protocol Using the Equational Theory of *Exclusive Or* and of Homomorphism. This example is taken from [CDL04]. The Wired Equivalent Privacy (WEP) protocol, described in [80299], is used to protect data during wireless transmission. To crypt the message M, A applies the `xor` (exclusive or) operator to *RC4(v,Kab)* and *(M,C(M))* where *C(M)* is the *integrity checksum* of the message M and *RC4* is a function modeling the RC4 algorithm which is used to generate a *keystream* (*i.e.* a long sequence of pseudo-random bytes) from the initial vector v and the secret key Kab shared between A and B. To decrypt the received message, B computes *RC4(v,Kab)* and after applying *exclusive or* to it and to the received message, he obtains *(M,C(M))* and can verify that the checksum is correct. We can represent this simple protocol by:

$$A \to B : v, ((M, C(M)) \oplus RC4(v, Kab))$$

[1] These results have been obtained only for a *bounded* number for parallel sessions since this verification problem for an unbounded number of sessions is already undecidable for an empty equational theory, that is under the perfect cryptography assumption.

[2] In fact, the NP-decision procedure in the case of Abelian groups given by [CLS03] can also be improved to *deterministic* polynomial time using the techniques explained in this report.

The checksum function C has the homomorphism property over `xor`, *i.e.* $C(x \oplus y) = C(x) \oplus C(Y)$. When messages have compatible lengths we have also $(x1, y1) \oplus (x2, y2) = (x1 \oplus x2, y1 \oplus y2)$.

We will now describe two attacks among those given in [BGW01].

The first one uses the fact that encrypting two messages *P1*, *P2* with the same initial vector v and with the same key k can reveal some information. Indeed, we have the following equalities between the ciphers *C1*, *C2* and their associated plain texts *P1*, *P2* which allows an intruder who knows a plain text *P1* and its cipher *C1* to decrypt any cipher *C2*. This attack only uses the properties of the `xor` symbol:

$$\begin{aligned} C1 \oplus C2 &= ((P1, C(P1)) \oplus RC4(v,k)) \oplus (P2, C(P2)) \oplus RC4(v,k)) \\ &= (P1, C(P1)) \oplus (P2, C(P2)) \end{aligned}$$

The second attack allows an intruder to make controlled modifications to a cipher text without disrupting the checksum. Assume that the intruder has intercepted $(M, C(M)) \oplus RC4(v, Kab)$ and knows D. He can now obtain the cipher text associated to the message $M \oplus D$ by computing:

$$\begin{aligned} ((M, C(M)) \oplus RC4(v, Kab)) \oplus (D, C(D)) &= RC4(v, Kab) \oplus ((M, C(M)) \oplus (D, C(D))) \\ &= RC4(v, Kab) \oplus (M \oplus D, C(M) \oplus C(D)) \\ &= RC4(v, Kab) \oplus (M \oplus D, C(M \oplus D)) \end{aligned}$$

In this second attack the intruder uses the properties of *exclusive or*, a special property of pairing and *exclusive or*, and the homomorphism property of C over the *exclusive or* operation.

Contribution and Approach of this Paper. In this paper we investigate the intruder deduction problem in presence of several variants of the equational theory of associativity and commutativity (short AC) of a binary operator $\oplus$, plus the homomorphism property of a binary function symbol over the AC operator. The variants of AC which we consider are: pure AC, the theory of *exclusive or* (also called ACUN), and the theory of Abelian groups. We are furthermore interested in the combination of these AC-like theories with a generalization of one homomorphic function to some form of distributivity of the encryption operator over the binary operator $\oplus$. The homomorphism law is now replaced by a law stating that the encryption of the $\oplus$ of two messages is equal to the $\oplus$ of the encryptions of the two messages using the same encryption key. This can be seen as the extension to an infinite family of homomorphisms, one for each possible encoding key.

We are interested both in decidability of the intruder deduction problem in any of these equational theories, and also in obtaining polynomial-time algorithms. Our results can be summarized as follows:

1. The intruder deduction problem is decidable, more precisely it is NP-complete in case of the theory AC plus homomorphism, and we have an EXPTIME upper bound for the equational theory ACUN plus homomorphism and Abelian groups plus homomorphism.

2. The intruder deduction problem is in all three cases decidable in polynomial time if we restrict the class of problems to the so-called *binary* case, that is the case where the set of assumptions and the theorem do not contain $\oplus$'s of more than two elements.

3. The first two sets of results carry over to the generalization which consists in replacing the homomorphic function by an encryption operation which distributes over *exclusive or*.

We follow the approach of [CLS03] and [CLT03] which consists in adapting Mc Allester's *locality* method to the problem at hand. Mc Allester [McA93] introduced *local* proof systems which are deduction systems (expressed as finite sets of Horn clauses) with the property that if there is a proof of a theorem then there also is a so-called *local* proof, that is a proof which only uses syntactic subterms of the set of assumption and the theorem. In case of a finite set of assumptions, existence of a local proof can be decided in polynomial time by the well-known bottom-up algorithm for the construction of the deduction closure of a set of terms by a finite set of *ground* Horn clauses (in this case all instances of the proof rules by syntactic subterms of the assumptions or the theorem).

In general, the difficulty in applying this technique to a given deduction system consists in showing the locality of the proof system. This is achieved by exhibiting a rewrite system which transforms an arbitrary given proof into a local proof of the same theorem, following the idea of *proof transformation* pioneered by Gentzen.

In [CLS03] and [CLT03], this approach was employed in a generalized form which consists in extending the notion of subterms from *syntactic* subterms to a wider notion which is consistent with the equational theory at hand. In case of the axioms of associativity and commutativity we furthermore have to cope with an infinite family of rules since we have to flatten successive application of the construction rules of an AC operator, as it was already observed in [CLS03].

In our case, the combination of for instance the theory of *exclusive or* with the homomorphism law poses a major obstacle when proving locality: In the construction of proofs it may now be necessary to construct large intermediate terms which partially cancel out when combining them by *exclusive or*, as it is explained later in the paper.

It has been shown in [CDL04] that decidability of unification modulo an equational theory E is a necessary condition for the decidability of the security of a protocol for a bounded number of sessions and in presence of this equational theory E. Since unification modulo AC plus homomorphism is undecidable [Nar96], security against active attackers is undecidable at least for this equational theory as well.

Structure of the Paper. After reminding about some basic notions in Chapter 2 we present in Chapter 3 the Dolev-Yao model of intruder capacities extended by equational reasoning, and prove it equivalent to a variant of the Dolev-Yao model which works only with normal forms of terms modulo a rewrite system in case the equational theory can be oriented into a rewrite system modulo a background equational theory. In this chapter we also present the three equational theories studied in this paper and their presentation as rewrite systems. Chapter 4 presents Mc Allester's locality technique and the generalization we are using here. In Chapter 5 we give definitions of the notion of *subterm* which allow us to prove locality in the three cases. Finally, we discuss in Chapter 6 an extension of our results to a variant where instead of one homomorphic function we have a family of homomorphism, each of which is the encryption function by some key. We summarize our results in Chapter 7.

Chapter 2

Preliminaries

We summarize some basic notations used in this paper, see [DJ90, BN98] for an overview of rewriting.

2.1 Terms

Let Σ be a signature. $T(\Sigma, X)$ denotes the set of terms over the signature Σ and the set of variables X, that is the smallest set such that

1. $X \subseteq T(\Sigma, X)$;
2. if $t_1, \ldots, t_n \in T(\Sigma, X)$, and $f \in \Sigma$ has arity $n \geq 0$, then $f(t_1, \ldots, t_n) \in T(\Sigma, X)$.

We abbreviate $T(\Sigma, \emptyset)$ as $T(\Sigma)$; elements of $T(\Sigma)$ are called Σ-*ground terms*. The set of variables occurring in a term t is denoted by $\mathcal{V}(t)$.

The *set of occurrences* of a term t is defined recursively as $\mathcal{O}(f(t_1, \ldots, t_n)) = \{\epsilon\} \cup \bigcup_{i=1\ldots n} i \cdot \mathcal{O}(t_i)$. For instance, $\mathcal{O}(f(a, g(b, x))) = \{\epsilon, 1, 2, 21, 22\}$. The *size* $|t|$ of a term t is defined as its number of occurrences, that is $|t| = \mathit{cardinality}(\mathcal{O}(t))$. If $o \in \mathcal{O}(t)$ then the *subterm of* t *at position* o is defined recursively by

- $t \mid_\epsilon = t$
- $f(t_1, \ldots, t_n) \mid_{j \cdot o} = t_j \mid_o$

$St(t)$ is the set of (not necessarily strict) *subterms* of the term t, that is $St(t) = \{t \mid_o \mid o \in \mathcal{O}(t)\}$. If t and s are terms and $o \in \mathcal{O}(t)$ then the *grafting* of s onto t at position o is defined recursively as

- $t[\epsilon \leftarrow s] = s$
- $f(t_1, \ldots, t_n)[j \cdot o \leftarrow s] = f(t_1, \ldots, t_{j-1}, t_j[o \leftarrow s], t_{j+1}, \ldots, t_n)$

For instance, $f(a, g(b, x))[22 \leftarrow h(c)] = f(a, g(b, h(c)))$.

For a unary function symbol f and a set of terms T we define $f^*(T)$ as the smallest set of terms which contains T and which is closed under application of f.

2.2 Equations and Rewriting Systems

A Σ-*equation* is a pair $(l, r) \in T(\Sigma, X)$, commonly written as $l = r$. The relation $=_E$ generated by a set of Σ equations E is the smallest congruence on $T(\Sigma)$ that contains all ground instances of all equations in E.

A Σ-*rewriting system* R is a finite set of so-called *rewriting rules* $l \to r$ where $l \in T(\Sigma, X)$ and $r \in T(\Sigma, \mathcal{V}(l))$. A term $t \in T(\Sigma, X)$ *rewrites* to s in one step by R if there is a rewriting rule $l \to r$ in R, an occurrence o and a substitution σ such that $t\mid_o = l\sigma$ and $s = t[o \leftarrow r\sigma]$. We write $\to^*$ for the reflexive and transitive closure of $\to$. A term t is in *normal form* if there is no term s with $t \to s$. If $t \to^* s$ and s is a normal form then we say that s is a *normal form of* t, and write $s = t\downarrow$, or $t \to^! s$.

A term rewriting system is called *convergent* if it is

- *strongly terminating*, that is if there is no infinite sequence of the form $t_1 \to t_2 \to t_3 \to \cdots$.
- *locally confluent*, that is if $t \to s_1$ and $t \to s_2$ then there exists a term r with $s_1 \to^* r$ and $s_2 \to^* r$.

By a well known result (see, e.g., [DJ90]), every convergent rewrite system is *confluent*, that is if $t \to^* s_1$ and $t \to^* s_2$ then there exists a term r with $s_1 \to^* r$ and $s_2 \to^* r$. As a consequence, in a convergent rewrite system every term has a unique normal form

By R/S we denote the so-called *class rewrite system* composed of a set $R = \{l_i \to r_i\}$ of rewrite rules and a set $S = \{u_i = v_i\}$ of equations. Generalizing the notion of term rewriting, we say that s rewrites to t *modulo* S, denoted $s \to_{R/S} t$, if $s =_S u[l\sigma]_p$ and $u[r\sigma]_p =_S t$, for some context u, position p in u, rule $l \to r$ in R, and substitution σ.

2.3 Miscellaneous

Definition 1 *We write* $A \subseteq_{fin} B$ *if:*

- $A \subseteq B$
- A *is a finite set*

We write $A \uplus B$ for the union of two *disjoint* sets A and B.

Chapter 3

A Dolev-Yao Model for AC-like Equational Theories

We consider the classic model of deduction rules introduced by Dolev and Yao [DY83] in order to model the deductive capacities of a passive intruder. In this model, an intruder may use any term he has previously observed on the network, and construct new terms by pairing, unpairing, using a free constructor, encryption and decryption, where in the last two cases the intruder also has to know the encryption key. For the sake of simplicity we here only consider symmetric encryption; the results and techniques can be easily transferred to the case of asymmetric encryption. Our aim in this chapter is to extend this model by an equational theory which can be exploited by the intruder to mount new attacks.

We are in particular interested in the case where the equational theory comprises the axioms of *associativity and commutativity* (AC) of a distinguished binary function symbol. For reasons that will become apparent in the next chapter we separate the construction rule for this binary AC operator from the construction rule for the free function symbols, and furthermore generalize this rule into a variadic rule.

In Section 3.1 we present a first variant of the Dolev-Yao model extended by equational reasoning. The extension consists of a rule for passing from one term to a term which is equivalent in the equational theory. Then, in Section 3.2, we present a more effective variant of the extended Dolev-Yao model for the case where the equational theory can be presented by a convergent term rewriting system modulo a background equational theory (which usually is AC). In this case we can work with normal forms of terms modulo the background theory, instead of allowing for unrestricted equational reasoning modulo the equational theory. Finally, in Sections 3.3 to 3.5, we show that the three equational theories we are interested in can be presented by a convergent term rewriting system modulo AC, and hence can be handled in the framework introduced in this chapter. These equational theories are:

- Associativity and commutativity of the distinguished binary operator $\oplus$, plus homomorphism of a unary function symbol f with respect to $\oplus$ (Section 3.3), we denote this equational theory by ACh;

- Associativity, commutativity, and nilpotence of the distinguished binary operator $\oplus$ and existence of a zero element for $\oplus$ (that is the theory of *exclusive or*), plus homomorphism of a unary function symbol f with respect to $\oplus$ (Section 3.4), we denote this equational theory by ACUNh;

- Abelian groups, plus homomorphism of a unary function symbol with respect to the group operator (Section 3.5), we denote this equational theory by AGh.

The material in Sections 3.1 and 3.2 follows the presentation of [CLT03], but is here extended to the case of an additional background equational theory.

(A) $\dfrac{u \in T}{T \vdash_E u}$

(P) $\dfrac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \langle u, v\rangle}$

(UL) $\dfrac{T \vdash_E \langle u, v\rangle}{T \vdash_E u}$

(UR) $\dfrac{T \vdash_E \langle u, v\rangle}{T \vdash_E v}$

(C) $\dfrac{T \vdash_E u \qquad T \vdash_E v}{T \vdash_E \{u\}_v}$

(D) $\dfrac{T \vdash_E \{u\}_v \qquad T \vdash_E v}{T \vdash_E u}$

(F) $\dfrac{T \vdash_E u_1 \quad \cdots \quad T \vdash_E u_n}{T \vdash_E \phi(u_1, \ldots, u_n)} \quad \phi \in \Sigma^-$

(Eq(E)) $\dfrac{T \vdash_E u \qquad u =_E v}{T \vdash_E v}$

(GX) $\dfrac{T \vdash_E u_1 \quad \cdots \quad T \vdash_E u_n}{T \vdash_E u_1 \oplus \ldots \oplus u_n}$

Figure 3.1: Dolev-Yao System extended equations theory E

3.1 The Dolev-Yao Model Extended by Equational Reasoning

Let Σ be a finite signature which can be partitioned as

$$\Sigma = \{\langle\cdot,\cdot\rangle, \{\cdot\}_\cdot, \oplus\} \uplus \Sigma^-$$

that is Σ consists of pairing $\langle\cdot,\cdot\rangle$, encryption $\{\cdot\}_\cdot$, a distinguished binary operator $\oplus$, and some set Σ^- of so-called free function symbols.

Let E be an equational theory over the signature Σ.

We use sequents of the form $T \vdash_E t$, where $T \subseteq_{\text{fin}} \mathcal{T}(\Sigma)$ and $t \in \mathcal{T}(\Sigma)$. The intended meaning of such a sequent $T \vdash_E t$ is that an intruder with a certain set of deduction capabilities can deduce the term t from his knowledge T and using the equational theory E. In the context of cryptographic protocols, T is typically a set of messages that an intruder has previously observed on a network. Different deduction capabilities can be defined by different deduction systems for these sequents.

The classic Dolev-Yao model [DY83] defines the deduction capacities of an intruder assuming perfect cryptography. This deduction system is composed of the following rules: (A) the intruder knows any term that he has previously observed, (P) the intruder can build a pair of two messages, (UL, UR) he can extract each member of a pair, (C) he can crypt a message m with a key k, (D) if he knows a key k he can decrypt a message encrypted by the same key, (F) he can construct a new term using a free function symbol $\phi \in \Sigma^-$.

Since we distinguish a special binary operator $\oplus$ we here furthermore add a rule (GX) which allows the intruder to build a new term from an arbitrary number of already known terms by using the (associative) $\oplus$ operator.

Finally, we model the weakening of the *perfect cryptography assumption* by giving the intruder the power to use equational reasoning modulo a given set E of equational axioms. The resulting set of deduction rules is given in Figure 3.1.

Definition 2 (Proof) *A Σ, E-*sequent *is an expression of the form $T \vdash_E u$ where E is an equational theory over Σ, $T \subseteq_{fin} \mathcal{T}(\Sigma)$, and $u \in \mathcal{T}(\Sigma)$.*

A proof *of a Σ, E-sequent $T \vdash_E u$ is a tree whose nodes are labeled by either Σ, E-sequents or expressions of the form "$v \in T$", such that:*

- *Each leaf is labeled by an expression of the form $v \in T$, and each non-leaf node is labeled by an Σ, E-sequent.*
- *Each node labeled by a sequent $T \vdash_E v$ has n children labeled by $T \vdash_E s_1, \ldots, T \vdash_E s_n$ such that there is an instance of an inference rule with conclusion $T \vdash_E v$ and hypotheses $T \vdash_E s_1, \ldots, T \vdash_E s_n$.*
- *The root of the tree is labeled by $T \vdash_E u$.*

3.2 The Dolev-Yao Model Extended by Rewriting

The above model is not appropriate for automated proof search since the Eq(E) rule allows equational reasoning at any moment of a proof. In order to define a more effective model we split the equational theory E into a *background theory* S and a rewrite system R.

Definition 3 (Rewrite Presentation of an Equational Theory) *Let E and S be equational theories over a signature Σ, and R a Σ-term rewriting system. (R, S) is a* rewrite presentation *of E iff*

- *R is locally confluent modulo S*
- *R is terminating modulo S*
- *For all closed Σ-terms u, v : $u =_E v$ iff $u \downarrow_{R/S} =_S v \downarrow_{R/S}$.*

In this case we can consider a variant of the extended model of Dolev-Yao defined in the preceding section which works on the normal form modulo S of the term at each step of the proof. The idea is that equivalence modulo S is easy to decide, such that we may omit the Eq(S) rule and just work with equivalence classes modulo S.

For the purpose of this paper the background theory S will always consist of AC($\oplus$), that is the axioms stating associativity and commutativity of the operator $\oplus$.

In the next part we will explain the equivalence of the two models. In the three final sections of this chapter we will give rewrite presentations of the three equational theories that we are interested in, *i.e.* AC and homomorphism, ACUN and homomorphism, and Abelian groups and homomorphism. We will verify that the rewriting systems associated to each of these equational theories are confluent and terminating modulo AC (to show these points we use the tool C*i*ME [CM96]).

To define the right notion of a normal form we need that the term rewriting system modulo the background equational theory is convergent. Notice that local confluence and termination modulo an equational theory of the term rewriting system imply its convergence.

We now define normal forms for a such system.

Definition 4 (Term in Normal Form) *Let (R, S) be a rewrite presentation of some equational theory E. A term t is in* normal form *if there is no term s such that $t \rightarrow_{R/S} s$ (t reduces to s by a rule of the rewriting system of R modulo S). If $t \rightarrow^* s$ and s is in normal form then we call s the normal form of t, denoted $s = t\downarrow$.*

Note that normal forms are unique only up to S-equivalence.

Example 1 *If we have the rewriting rule $f(x \oplus y) \rightarrow_R f(x) \oplus f(y)$ and S is the theory of associativity and commutativity of $\oplus$, then the normal form of $f((a \oplus b) \oplus c)$ is $f(a) \oplus f(b) \oplus f(c)$.*

Normal forms have the following properties:

- $\forall u, v : u =_E v \Rightarrow u\downarrow =_S v\downarrow$
- $\forall u : u =_E u\downarrow$

$$
\begin{array}{llll}
\text{(A)} & \dfrac{u \in T}{T \vdash u{\downarrow}} & \text{(P)} & \dfrac{T \vdash u \qquad T \vdash v}{T \vdash \langle u, v\rangle{\downarrow}} \\[3ex]
\text{(UL)} & \dfrac{T \vdash r}{T \vdash u{\downarrow}} \quad if \langle u, v\rangle \to^! r & \text{(UR)} & \dfrac{T \vdash r}{T \vdash v{\downarrow}} \quad if \langle u, v\rangle \to^! r \\[3ex]
\text{(C)} & \dfrac{T \vdash u \qquad T \vdash v}{T \vdash \{u\}_v{\downarrow}} & \text{(D)} & \dfrac{T \vdash r \qquad T \vdash v}{T \vdash u{\downarrow}} \quad if \{u\}_v \to^! r \\[3ex]
\text{(F)} & \dfrac{T \vdash u_1 \quad \cdots \quad T \vdash u_n}{T \vdash \phi(u_1, \ldots, u_n){\downarrow}} \quad \phi \in \Sigma^- & \text{(GX)} & \dfrac{T \vdash u_1 \quad \cdots \quad T \vdash u_n}{T \vdash (u_1 \oplus \ldots \oplus u_n){\downarrow}}
\end{array}
$$

Figure 3.2: A Dolev-Yao proof system working on normal forms modulo a background equational theory.

An immediate remark is: if $t[.]$ is a context and u a ground term then $t[u{\downarrow}]{\downarrow} =_S t[u]{\downarrow}$. In particular $(u_1 \oplus \ldots \oplus u_n){\downarrow} =_S (u_1{\downarrow} \oplus \ldots \oplus u_n{\downarrow}){\downarrow}$ We omit the rule (Eq(E)) and consider a new system $\vdash$ presented in Figure 3.2 which only works on normal forms.

Theorem 1 *Let (R, S) be a rewrite presentation of the equational theory E, $T \subseteq_{fin} \mathcal{T}(\Sigma)$, and $T \in \mathcal{T}(\Sigma)$. We have that*

$$T \vdash_E u \qquad \textit{iff} \qquad T \vdash u{\downarrow}$$

Proof: Given a proof of $T \vdash u{\downarrow}$ we can easily find a proof of $T \vdash_E u$ by inserting Eq(E)-steps.

For the other direction of the the theorem we transform a proof in $\vdash_E$ into a proof in $\vdash$ by the transformation of Figure 3.3. This transformation does not change the leaves of a proof tree. We show by induction that if there is a proof of $T \vdash_E u$ then the transformation yields a proof of $T \vdash u{\downarrow}$.

$$
\text{(Eq(E))}\ \dfrac{\text{(R)}\ \dfrac{\psi_1}{T \vdash_E u} \qquad u =_E v}{T \vdash_E v} \quad \Longrightarrow \quad \text{(R)}\ \dfrac{\psi_1}{T \vdash v{\downarrow}}
$$

$$
\text{(R)}\ \dfrac{\text{(R}_1\text{)}\ \dfrac{\psi_1}{T \vdash_E u_1} \quad \cdots \quad \text{(R}_n\text{)}\ \dfrac{\psi_n}{T \vdash_E u_n}}{T \vdash_E v} \quad \Longrightarrow \quad \text{(R)}\ \dfrac{\text{(R}_1\text{)}\ \dfrac{\psi_1}{T \vdash_E u_1} \quad \cdots \quad \text{(R}_n\text{)}\ \dfrac{\psi_n}{T \vdash_E u_n}}{T \vdash v{\downarrow}}
$$

Figure 3.3: Transformation of a proof of $T \vdash_E u$ into a proof of $T \vdash u{\downarrow}$.

We proceed by case distinction on the last deduction rule:

- (A): obvious.
- (Eq(E)): Since (R, S) is a rewrite presentation of E we get $u{\downarrow} = v{\downarrow}$ (modulo S) so we obtain a proof of $T \vdash u{\downarrow}$
- (P), (C) or (F): by induction hypothesis on all the hypotheses of the rule and with the fact $f(u_1, \ldots, u_n){\downarrow} = f(u_1{\downarrow}, \ldots, u_n{\downarrow}){\downarrow}$ we get the result.
- (GX) we know that $(u_1 \oplus \ldots \oplus u_n){\downarrow} = (u_1{\downarrow} \oplus \ldots \oplus u_n{\downarrow}){\downarrow}$ and by induction $T \vdash u_1{\downarrow}, \ldots, T \vdash u_n{\downarrow}$. We apply (GX) and conclude.

- (D), by induction $T \vdash \{u\}_v\downarrow$ and $T \vdash v\downarrow$, with $\{u\}_{v\downarrow} = \{u\}_{v\downarrow}\downarrow$ and the rule (D) we get $T \vdash u\downarrow$, hence a proof in $\vdash$.
- (UL) or (UR) by induction we obtain the result. $\square$

In the following we will always work with the system $\vdash$ which works on normal forms modulo a background equational theory.

3.3 The Equational Theory ACh

The equational theory denoted by ACh consists of the axioms of associativity (A) and commutativity (C) of an operator $\oplus$, plus the law of homomorphism of a function f over $\oplus$.

3.3.1 Equational Theory

We consider the symbols $\oplus$ and f with the following axioms:

1. $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ (Associativity)
2. $x \oplus y = y \oplus x$ (Commutativity)
3. $f(x \oplus y) = f(x) \oplus f(y)$ (Homomorphism)

3.3.2 Rewrite Presentation

We do not orient the two rules of associativity and commutativity and consider the new system modulo AC and transform the homomorphism equation into the following rewriting rule:

- $f(x \oplus y) \to f(x) \oplus f(y)$

We show that this system is locally confluent and terminating.

3.3.3 Local Confluence

Showing that the system is locally confluent is equivalent to showing that all critical pairs are joinable since the system is terminating (see below).

We see that there are no critical pairs.

3.3.4 Termination

We show termination by giving a polynomial interpretation of the function symbols:

- constant: $[0] = 0$
- variable: $[x] = x$
- function f: $[f(x)] = 2x$
- xor: $[x \oplus y] = x + y + 1$

We show that all the rules of the rewriting system modulo AC are decreasing:

- $f(x \oplus y) \to f(x) \oplus f(y)$ gives: $[f(x \oplus y)] = 2x + 2y + 2 > 2x + 2y + 1 = [f(x) \oplus f(y)]$

Hence, the term rewriting system is terminating .

3.4 The Equational Theory ACUNh

The equational theory denoted by ACUNh consists of associativity (A), commutative (C), unity (U) and nilpotence (N) of an operator $\oplus$, plus the law of homomorphism of a function f over $\oplus$.

3.4.1 Equational Theory

We consider the symbols $\oplus, 0, f$ with the following axioms:

1. $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ (Associativity)
2. $x \oplus y = y \oplus x$ (Commutativity)
3. $0 \oplus x = x$ (Unit)
4. $x \oplus x = 0$ (Nilpotence)
5. $f(x \oplus y) = f(x) \oplus f(y)$ (Homomorphism)

3.4.2 Orientation of the Axioms

Consider the following term rewriting system modulo AC:

1. $0 \oplus x \to x$
2. $x \oplus x \to 0$
3. $f(x \oplus y) \to f(x) \oplus f(y)$

We do not orient the two rules of associativity and commutativity. We now show that this system can be extended to an equivalent and convergent system.

3.4.3 Local Confluence

Showing that a term rewrite system is locally confluent is equivalent to showing that all its critical pairs are joinable since the system is terminating (see below).

We see that there are two critical pairs:

$$f(x) \oplus f(0) \leftarrow f(x \oplus 0) \to f(x)$$

$$f(x) \oplus f(x) \leftarrow f(x \oplus x) \to f(0)$$

With the homomorphic property $f(x \oplus y) = f(x) \oplus f(y)$ and the properties of $\oplus$ we deduce that $f(0) = f(x \oplus x) = f(x) \oplus f(x) = 0$ so $f(0) = 0$. We add the rewrite rule $f(0) \to 0$ and obtain a completed rewrite system in which all critical pairs are joinable. This system is hence locally confluent.

3.4.4 Rewrite Presentation

Consider the following confluent term rewriting system modulo AC:

1. $0 \oplus x \to x$
2. $x \oplus x \to 0$
3. $f(x \oplus y) \to f(x) \oplus f(y)$
4. $f(0) \to 0$

3.4.5 Termination

We show termination of the confluent term rewriting system modulo AC by giving a polynomial interpretation:

- constant: $[0] = 0$
- variable: $[x] = x$
- function f: $[f(x)] = 2x + 1$
- xor: $[x \oplus y] = x + y + 1$

We show that all the rules of the rewriting system modulo AC are decreasing:

- $0 \oplus x \to x$ gives: $[0 \oplus x] = x + 1 > x = [x]$
- $x \oplus x \to 0$ gives: $[x \oplus x] = 2x + 1 > 0 = [0]$
- $f(x \oplus y) \to f(x) \oplus f(y)$ gives: $[f(x \oplus y)] = 2x + 2y + 3 > 2x + 2y + 2 = [f(x) \oplus f(y)]$
- $f(0) \to 0$ gives: $[f(0)] = 1 > 0 = [0]$

Hence, the term rewriting system is terminating .

3.5 The Equational Theory AGh

The equational theory denoted by AGh consists of the laws of Abelian groups with operator $\oplus$, plus the law of homomorphism of a function f over $\oplus$.

3.5.1 Equational Theory

1. $(x \oplus y) \oplus z = x \oplus (y \oplus z)$ (Associativity)
2. $x \oplus y = y \oplus x$ (Commutativity)
3. $0 \oplus x = x$ (Unit)
4. $x \oplus I(x) = 0$ (Nilpotence)
5. $f(x \oplus y) = f(x) \oplus f(y)$ (Homomorphism)

3.5.2 Orientation of the Axioms

Associate the following term rewrite system modulo AC:

1. $0 \oplus x \to x$
2. $x \oplus I(x) \to 0$
3. $f(x \oplus y) \to f(x) \oplus f(y)$
4. $f(0) \to 0$

The last rewriting rule comes from the study of the ACUNh case (see Section 3.4).

3.5.3 Local Confluence

We complete the system into a locally confluent system. We compute the critical pairs and add each time a rule to solve the problem. We see that there are four critical pairs:

1. $I(0) \leftarrow I(0) \oplus 0 \rightarrow 0$
2. $I((I(x) \oplus x)) \oplus y \leftarrow I((I(x) \oplus x)) \oplus I(x) \oplus x \oplus y \rightarrow 0 \oplus y$
3. $0 \oplus x \leftarrow I((I(x) \oplus x)) \oplus I(x) \oplus x \oplus x \rightarrow I(I(x)) \oplus x$
4. $f(0) \leftarrow f(I(x) \oplus x) \rightarrow f(I(x)) \oplus f(x)$

For solve each critical pairs we add the following rules :

- $I(0) \rightarrow 0$
- $I(x \oplus y) \rightarrow I(x) \oplus I(y)$
- $I(I(x)) \rightarrow x$
- $f(I(x)) \rightarrow I(f(x))$

We can add these rewrite rules without problem because from the initial equations we can deduce the following properties:

Property 1 *The following equations are consequences of the axioms:*

1. $I(0) = 0$
2. $I(x \oplus y) = I(x) \oplus I(y)$
3. $I(I(x)) = x$
4. $f(I(x)) = I(f(x))$

Proof:

1. Axiom 4 applied on 0 gives $0 \oplus I(0) = 0$, hence $I(0) = 0$.
2. Axiom 4 yields $x \oplus I(x) = 0$, hence (replacing x by $x \oplus y$) we also obtain $x \oplus y \oplus I(x \oplus y) = 0$. Add on each side $I(x) \oplus I(y)$ to obtain $I(x \oplus y) = I(x) \oplus I(y)$.
3. Double application of Axiom 4 yields $I(I(x)) \oplus I(x) = 0$. We add on each side x and use Axiom 4 to obtain $I(I(x)) = x$.
4. We have already proved that $f(0) = 0$ then $0 = f(0) = f(I(x) \oplus x) = f(I(x)) \oplus f(x)$ then $f(I(x))$ is the inverse of $f(x)$ so $I(f(x)) = f(I(x))$. □

All the critical pairs are joinable by the new added rewriting rules, hence the completed system is locally confluent.

3.5.4 Rewrite Presentation

Consider the following confluent term rewriting system modulo AC:

1. $0 \oplus x \rightarrow x$
2. $x \oplus x \rightarrow 0$
3. $f(x \oplus y) \rightarrow f(x) \oplus f(y)$
4. $f(0) \rightarrow 0$

5. $I(0) \to 0$
6. $I(x \oplus y) \to I(x) \oplus I(y)$
7. $I(I(x)) \to x$
8. $f(I(x)) \to I(f(x))$

3.5.5 Termination

For the termination we consider the following polynomial interpretation :

- For a constant : $[0] = 1$
- For a variable : $[x] = x$
- For a function f : $[f(x)] = 2x$
- For the function inverse I : $[I(x)] = 2x + 1$
- For the *xor* symbol : $[x \oplus y] = x + y + 2$

We show that all the rules modulo AC are decreasing.

- $0 \oplus x \to x$: $[0 \oplus x] = x + 3 > x = [x]$
- $x \oplus I(x) \to 0$: $[x \oplus I(x)] = 3x + 3 > 1 = [0]$
- $I(I(x)) \to x$: $[I(I(x))] = 4x + 3 > x = [x]$
- $I(x \oplus y) \to I(x) \oplus I(y)$: $[I(x \oplus y)] = 2x + 2y + 5 > 2x + 2y + 4 = [I(x) \oplus I(y)]$
- $I(0) \to 0$: $[I(0)] = 3 > 1 = [0]$
- $f(0) \to 0$: $[f(0)] = 2 > 1 = [0]$
- $f(I(x)) \to I(f(x))$: $[f(I(x))] = 4x + 2 > 4x + 1 = [I(f(x))]$
- $f(x \oplus y) \to f(x) \oplus f(y)$: $[f(x \oplus y)] = 2x + 2y + 4 > 2x + 2y + 2 = [f(x) \oplus f(y)]$

The system is terminating modulo AC.

Chapter 4

Locality and Complexity of the Intruder Deduction Problem

4.1 A Generalization of Locality

Our starting point is the locality technique introduced by David McAllester [McA93]. He considers deduction systems which are represented by finite sets of Horn clauses and he proves that there exists a polynomial-time algorithm to decide the deducibility of a term w from a finite set of terms T_0 if the deduction system has the so-called *locality property*. A deduction system has the *locality property* if any proof can be transformed into a local proof where a *local proof* is a proof where all the nodes are syntactic subterms of T_0 and w.

The idea of the proof is as follows: the existence of a local proof can be checked in polynomial time since there is only a polynomial number of relevant instances of the deduction rules (The relevant instances are the instances by subterms of the given problem). To check the existence of a local proof amounts to computing the intersection of the deduction closure of the initial knowledge with the set of relevant terms. Algorithm 1 computes the restriction of the deduction closure of T_0 to the set of relevant terms.

We say that w is *one-step deducible* from T, if we can obtain w from T with only one application of a rule of the proof system. We denote in the following algorithm the one-step deduction relation by $\vdash^{\leq 1}$.

Input: T_0, w
$T \leftarrow T_0$;
while $\exists s \in \mathit{SyntacticSubterms}(T_0, w)$ *such that* $(T \vdash^{\leq 1} s$ *and* $s \notin T)$ **do**
| $T \leftarrow T \cup \{s\}$;
end
return $w \in T$;

Algorithm 1: McAllester's Algorithm to check the existence of a local proof.

There are two main restrictions in this approach :

- The deduction system must be finite.
- The notion of locality is restricted to syntactic subterms.

These restrictions raise a serious problem when we are working modulo AC, as it is already pointed out in [CLS03]. If we used the binary rule (GX) we would have to consider all possible subterms modulo AC. Unfortunately, there is in general an exponential number of subterms modulo AC of a given term. The solution proposed in [CLS03], and which we also adopt here, is to use the rule (GX) with an arbitrary number of hypotheses. In this way, we can avoid the exponential number of subterms. However, we are now stuck with an infinite number of rules. Fortunately, we can still obtain an polynomial algorithm by implementing the test used in the loop in Algorithm 1 in a clever way.

Definition 5 (S-Local Proof) *Let S be a function which maps a set of terms to a set of terms. A proof P of $T \vdash w$ is* S-local *if all nodes are labeled by some $T \vdash v$, with $v \in S(T \cup \{w\})$.*

Definition 6 (S-Locality) *A proof system is* S-local *if whenever there is a proof of $T \vdash w$ then there also is a S-local proof of $T \vdash w$.*

In the following theorem, $\mathcal{K}_1$ and $\mathcal{K}_2$ are complexity classes which are closed under multiplication (for instance, PTIME, NPTIME, EXPTIME, etc.).

Theorem 2 (S-Locality) *Let S be a function mapping a set of terms to a set of terms, and P a proof system. If:*

- *the set $S(T)$ can be constructed in time $\mathcal{K}_1$,*
- *P is S-local,*
- *one-step deductibility in P is decidable in time $\mathcal{K}_2$,*

then provability in the proof system P is decidable in time $\max(\mathcal{K}_1, \mathcal{K}_2)$.

Proof: Let n be the sum of the size of the set T_0 of hypotheses and of the size of the term w for which we want to check whether $T \vdash w$ holds. The number of iterations of the loop is bounded by the number of instances of the conclusions of the rules of the proof system by terms in $S(T_0 \cup \{w\})$. Hence, by hypothesis, the number of iterations of the loop is bounded by $\mathcal{K}_1(n)$. As a consequence, execution of the algorithm takes time at most $\mathcal{K}_1(n) * \mathcal{K}_2(n)$. By S-locality of the proof system, existence of a proof is equivalent to existence of a local proof. $\square$

This theorem generalizes McAllester's result because in his case:

- The size of the set of syntactic subterms of the set T is polynomial in the size of T.
- One-step deducibility is decidable in polynomial time for a finite proof system.

Hence, in McAllester's case, it remained only the (S-)locality to show. In our more general setting we have to do the following steps in order to obtain a decidability result, for instance in polynomial time, for the Dolev-Yao deduction system modulo a AC-rewrite system:

- define the notion of subterms which can be computed in polynomial time ,
- show locality with respect to this notion of subterms,
- show that one-step deductibility can be tested in polynomial time.

4.2 One-Step Deducibility for AC-like Theories

Definition 7 (Headed with $\oplus$) *Let u be a term in normal form, u is* headed with $\oplus$ *if u is of the form $u_1 \oplus \ldots \oplus u_n$ with $n > 1$. Otherwise u is not headed with $\oplus$.*

Example 2 *$u \oplus \langle v, w\rangle$ is headed with $\oplus$, but $\langle u, v \oplus w\rangle$ is not.*

Definition 8 (Atom) *Let u be a term, we define the function atoms(u) as following :*

- *If $u = u_1 \oplus \ldots \oplus u_n$, where each of the u_i is not headed with $\oplus$, then atoms$(u) = \{u_1, \ldots, u_n\}$. The u_i's are called the* atoms *of u.*
- *If u is not headed with $\oplus$ then atoms$(u) = u$.*

The definition of atoms(T) generalizes in a natural way to sets of terms T in normal form by $\text{atoms}(T) := \bigcup_{t \in T} \text{atoms}(t)$.

Definition 9 (Idempotent Mapping over Sets of Terms) *Let T be a set of terms. The mapping $S : T \to T$ is idempotent if for every $X \subseteq T$: $S(S(X)) = S(X)$.*

Definition 10 (Monotone Mapping over Set of Terms) *Let T be a set of terms. The mapping $S : T \to T$ is monotone if for all $X, Y \subseteq T$: if $X \subseteq Y$ then $S(X) \subseteq S(Y)$.*

Lemma 1 *The function atoms is monotone and idempotent.*

Proof: Straightforward from the definition. □

Definition 11 (Binary Proof) *A proof P of $T \vdash w$ is binary if in all nodes labeled $T \vdash v$ the term v has at most two atoms.*

If the equational theory is AC and if we consider binary proofs only, then one-step deducibility is in PTIME.

We now investigate the complexity of one-step deducibility for the three equational theories defined in Sections 3.3 to 3.5.

Our method is inspired by the method used in [Nar96] to solve a unification problem modulo AC-like theories. We only show how to decide one-step deducibility for the family of rules (GX), since checking one-step deducibility for the remaining deduction rules is straightforward. We transform the problem of testing one-step deducibility into the solvability of a system of linear Diophantine equations. The domain over which this system of equations is solved depends on the equational theory considered, as discussed below.

- Input :
  - a finite set of terms $T = \{t_0, \ldots, t_n\}$
  - a term s
- Output
  - A system $D(T, s)$ of linear Diophantine equations over the variables $X = \{x_0, \ldots, x_n\}$ such that $T \vdash_E s$ if only if $D(T, s)$ is solvable in a domain which depends on E.
- Algorithm
  - To each t_i we associate the variable x_i for $i = 0, \ldots, n$.
  - If u is an atom of t, let $\delta(u, t)$ denote the number of occurrences of the atom u in t.
  - Let $A = \{a_1, \ldots, a_m\}$ be the set of atoms of $T \cup \{s\}$.
  - For each a_i atom of s, we introduce the equation:

    $$\delta(a_i, s) = \sum_{j=0}^{n} \delta(a_i, t_j) * x_j$$

    which states that the number of occurrences of a_i in s is equal to the sum of the number of occurrences of a_i in (a sum of) t_j's. The system $D(T, s)$ is the conjunction of these equations:

    $$D(T, s) := \bigwedge_{i=1}^{m} \sum_{j=0}^{n} \delta(a_i, t_j) * x_j = \delta(a_i, s)$$

Example 3 *Let $T = \{a_1 \oplus a_2 \oplus a_3, a_1 \oplus a_4, a_2 \oplus a_4\}$ and $s = a_1 \oplus a_2$, where all the a_i are not headed with $\oplus$. We introduce numerical variables x_0, x_1, x_2, that is one numerical variable for each element of T :*

$$\begin{array}{lll} x_0 & \textit{for} & a_1 \oplus a_2 \oplus a_3 \\ x_1 & \textit{for} & a_1 \oplus a_4 \\ x_2 & \textit{for} & a_2 \oplus a_4 \end{array}$$

For every atom a_i we create an equation. This yields the following equation system:

$$\left\{ \begin{array}{lll} a_1 & : & x_0 + x_1 = 1 \\ a_2 & : & x_0 + x_2 = 1 \\ a_3 & : & x_0 = 0 \\ a_4 & : & x_1 + x_2 = 0 \end{array} \right.$$

Lemma 2 *Let S be the following system of equations:*

$$\left\{ \begin{array}{ccc} c_{1,1}x_1 + \ldots + c_{1,n}x_n & = & d_1 \\ \vdots & \vdots & \vdots \\ c_{m,1}x_1 + \ldots + c_{m,n}x_n & = & d_m \end{array} \right.$$

$$w_S = d_1 * A_1 \oplus \ldots \oplus d_m * A_m$$

$$\Sigma = \{A_1, \ldots, A_m\}, T_S = \{t_1, \ldots, t_m\}$$

Where $\forall i, 1 \leq i \leq n, t_i = c_{1,i} * A_1 \oplus \ldots \oplus c_{m,i} * A_m$ $\forall i, 1 \leq i \leq m, A_i$ are atoms of T_S.

The system of equations (S) is satisfiable if and only if $T_S \vdash w_S$ using only (A) and (GX).

Proof:

- If (S) is satisfiable then there exists a solution of (S) α such that:

$$\left\{ \begin{array}{ccc} c_{1,1}\alpha(x_1) + \ldots + c_{1,n}\alpha(x_n) & = & d_1 \\ \vdots & \vdots & \vdots \\ c_{m,1}\alpha(x_1) + \ldots + c_{m,n}\alpha(x_n) & = & d_m \end{array} \right.$$

  Hence, we compute w_S from T_S and α:

$$\begin{array}{ll} & \alpha(x_1) * t_1 \oplus \ldots \oplus \alpha(x_n) * t_n \\ = & \alpha(x_1) * (c_{1,1} * A_1 \oplus \ldots \oplus c_{m,1} * A_m) \oplus \ldots \oplus \alpha(x_n) * (c_{1,n} * A_1 \oplus \ldots \oplus c_{m,n} * A_m) \\ = & c_{1,1} * \alpha(x_1) * A_1 \oplus \ldots \oplus c_{1,n} * \alpha(x_n) * A_1 \oplus \ldots \oplus c_{m,1} * \alpha(x_1) * A_m \oplus \ldots \oplus c_{m,n} * \alpha(x_n) * A_m \\ = & d_1 * A_1 \oplus \ldots \oplus d_m * A_m \\ = & w_S \end{array}$$

- Let P be a proof of $T_S \vdash w_S$, using only (A)(GX). We construct the system (S) from T_S and w_S.

$$T_S = \{t_1, \ldots, t_m\}, \Sigma = \{A_1, \ldots, A_m\} = \text{atoms}(T_S)$$

$$w_S = d_1 * A_1 \oplus \ldots \oplus d_m A_m$$

  $\forall i, 1 \leq i \leq n, \exists c_{j,i} 1 \leq j \leq m, t_i = c_{1,i} * A_1 \oplus \ldots \oplus c_{m,i} * A_m$

  We can deduce w_S from T_S, there exist a decomposition of d_i into $c_{i,j}$, hence we obtain the following system:

$$\left\{ \begin{array}{ccc} c_{1,1}x_1 + \ldots + c_{1,n}x_n & = & d_1 \\ \vdots & \vdots & \vdots \\ c_{m,1}x_1 + \ldots + c_{m,n}x_n & = & d_m \end{array} \right.$$

□

We now explain in which domain this linear Diophantine equation system is solved according to the equational theory that is considered.

ACh Case

We obtain a system of linear Diophantine equations over $\mathbb{N}$. This system has a solution iff s is deducible from T in one step. Since solvability of a system of linear equations over $\mathbb{N}$ is a NP-complete problem [Pap94] we get an algorithm in NP for checking one-step deducibility.

ACUNh Case

In this case we solve a system of linear Diophantine equations over $\mathbb{Z}/2\mathbb{Z}$. This system has a solution iff s is deducible from T in one step. Since the former problem is in PTIME [KKS87], we get a PTIME algorithm for checking one-step deducibility.

AGh Case

Now we consider the case where $\oplus$ is the operator of Abelian groups. We can assume w.l.o.g. that no term headed with $\oplus$ contains an atom and its inverse because we are only working on terms in normal form. Now we redefine the function δ:

- if u is an atom of t which is not headed with I then we note $\delta(u,t)$ for the number of occurrences of the atom u in t,
- otherwise if $I(u)$ is an atom of t then we note $\delta(u,t)$ for the inverse of the number of occurrences of $I(u)$ in t.

We get a system of linear Diophantine equations to solve over $\mathbb{Z}$. This system has a solution iff s is deducible from T in one step. Since the solvability of a system of linear Diophantine equations over $\mathbb{Z}$ is in PTIME [Sch86], we get a PTIME algorithm for checking one-step deducibility.

We denote by $n * a := a \oplus a \oplus \ldots \oplus a$ n times.

Example 4 *Let $T = \{a_1 \oplus a_2 \oplus I(a_3) \oplus I(a_3), a_1 \oplus a_4, a_2 \oplus I(a_4)\}$ and $s = a_1 \oplus a_2$, where all the a_i are not headed with $\oplus$. To each element of T we associate one numerical variable:*

$$\begin{array}{lll} x_0 & \textit{for} & a_1 \oplus a_2 \oplus I(a_3) \oplus I(a_3) \\ x_1 & \textit{for} & a_1 \oplus a_4 \\ x_2 & \textit{for} & a_2 \oplus I(a_4) \end{array}$$

For every atom a_i we create an equation, yielding the following equation system:

$$\left\{ \begin{array}{lcl} a_1 & : & x_0 + x_1 = 1 \\ a_2 & : & x_0 + x_2 = 1 \\ a_3 & : & -2x_0 = 0 \\ a_4 & : & x_1 - x_2 = 0 \end{array} \right.$$

Chapter 5

Locality for AC-Like Equational Theories with Homomorphism

In the first part of this chapter we list all the definitions and properties we need to decide the intruder deduction problem modulo AC-like equational theories. Then we show, for our three equational theories introduced in Chapter 3, that we can define an operator S which in some cases is polynomial-size, and for which we can show locality.

5.1 Preliminary Definitions and Properties

5.1.1 Vocabulary

We define some notions.

Definition 12 (Headed with f) *A term u in normal form is called* headed with f *if u is of the form $u = f(t)$ for some term t. Otherwise u is* not headed with f.

We need to know the number of applications of f applied at the head of a term.

Definition 13 ($\#_f(t)$) *We define the number of applications of f at the head of a term t by:*

- *if t is not headed with f then $\#_f(t) = 0$*
- *otherwise $t = f(t')$ and $\#_f(t) = \#_f(t') + 1$*

Example 5 *Let $t = f^{13}(a) \oplus f^5(\langle b, f^{10}(c)\rangle)$ be a term in normal form then $\#_f(t) = 0$. If we have $t = f^5(\langle b, f^{10}(c)\rangle)$ then we get $\#_f(t) = 5$.*

Definition 14 (*Strip*$_f$) *Let t be a term. We define $Strip_f(t)$ by:*

- *if t is not headed with f then $Strip_f(t) = t$*
- *otherwise $t = f(t')$ and $Strip_f(t) = Strip_f(t')$*

Definition 15 (Size of a Term) *Let be t a term in normal form, we define recursively the* size $|t|$ *of t by:*

- $|t| = 1$ *if t is a constant.*
- $|t| = |\langle u, v\rangle| = 1 + |u| + |v|$
- $|t| = |\{u\}_v| = 1 + |u| + |v|$

- $|t| = |u_1 \oplus \ldots \oplus u_n| = 1 + \sum_{i=1}^{n} |t_i|$
- $|t| = |f(t_1, \ldots, t_n)| = 1 + \sum_{i=1}^{n} |t_i|$

We extend this notion to the size of a set of terms by $|T| = \sum_{t \in T} |t|$.

Example 6 $|f(\langle a, \{b\}_k \rangle)| = 6$

5.1.2 Different Kinds of Proofs

Now we want to characterize different kinds of proofs.

Definition 16 (Subproof) *Let P be a proof of $T \vdash u$. A proof P' is a* subproof *of P if P' is a sub-tree of the tree P.*

Definition 17 (Size of a Proof) *The* size of a proof *P is the number of nodes in P, denoted by $|P|$.*

Definition 18 (Minimal Proof) *A proof P of $T \vdash u$ is* minimal *if there is no proof P' of $T \vdash u$ such that $|P'| < |P|$.*

Definition 19 (Simple Proof) *A* simple proof *is a proof where each node $T \vdash v$ occurs at most once on each branch.*

Property 2 (P Minimal $\Rightarrow$ P Simple) *If P is a minimal proof of $T \vdash u$ then P is a simple proof of $T \vdash u$.*

Proof: Let us assume to the contrary that P is a non-simple proof of $T \vdash u$. Then there is a branch of P in which $T \vdash v$ occurs twice. We can cut the derivation between these two occurrences and so get a smaller proof P', which is in contradiction to the minimality of P. □

Definition 20 (Flat Proof) *Let P be a proof of $T \vdash w$, P is a* flat proof *if there is no (GX) rule immediately above another (GX) rule.*

Since two successive (GX) rules can be merged into a single (GX) rule a minimal proof is a flat proof.

Definition 21 ($\oplus$-*lazy* Proof) *Let P be a proof of $T \vdash w$, P is a $\oplus$-lazy* proof *if P is flat and there is no (GX) rule immediately above an (F) rule in P.*

Definition 22 ($\oplus$-*eager* Proof) *Let P be a proof of $T \vdash w$, P is a $\oplus$-eager proof if P is flat and if there is at most one (F) rule immediately above a (GX) rule in P.*

Intuitively, in a $\oplus$-*lazy* proof the (GX) rule is applied as late as possible, and in a $\oplus$-*eager* proof the (GX) rule is applied as early as possible.

Example 7 (A simple proof which is not minimal) *Consider the equational theory of* exclusive or *with homomorphism. The following $\oplus$-eager proof with $T = \{u, v, k\}$ is simple:*

$$\textit{(GX)}\ \dfrac{\textit{(F)}\ \dfrac{\textit{(GX)}\ \dfrac{\textit{(A)}\ \dfrac{u \in T}{T \vdash u} \qquad \textit{(A)}\ \dfrac{v \in T}{T \vdash v}}{T \vdash u \oplus v}}{T \vdash f(u \oplus v) = f(u) \oplus f(v)} \qquad \textit{(A)}\ \dfrac{f(v) \in T}{T \vdash f(v)}}{T \vdash f(u)}$$

We transform it into a $\oplus$-lazy simple proof

$$\text{(GX)}\ \dfrac{\text{(F)}\ \dfrac{\text{(A)}\ \dfrac{u \in T}{T \vdash u}}{T \vdash f(u)} \qquad \text{(F)}\ \dfrac{\text{(A)}\ \dfrac{v \in T}{T \vdash v}}{T \vdash f(v)} \qquad \text{(A)}\ \dfrac{f(v) \in T}{T \vdash f(v)}}{T \vdash f(u)}$$

However, these proofs are not minimal since there is a smaller proof:

$$\text{(F)}\ \dfrac{\text{(A)}\ \dfrac{u \in T}{T \vdash u}}{T \vdash f(u)}$$

Example 8 (An $\oplus$-*eager* Proof and a $\oplus$-*lazy* Proof) *The following proof of $f(b) \oplus f(c)$ with $T = \{a \oplus b, d \oplus c, f(d) \oplus f(a)\}$ is minimal and $\oplus$-eager but it is not $\oplus$-lazy.*

$$\text{(GX)}\ \dfrac{\text{(F)}\ \dfrac{\text{(GX)}\ \dfrac{\text{(A)}\ \dfrac{a \oplus b \in T}{T \vdash a \oplus b} \qquad \text{(A)}\ \dfrac{c \oplus d \in T}{T \vdash c \oplus d}}{T \vdash a \oplus b \oplus c \oplus d}}{T \vdash f(a) \oplus f(b) \oplus f(c) \oplus f(d)} \qquad \text{(A)}\ \dfrac{f(a) \oplus f(d) \in T}{T \vdash f(a) \oplus f(d)}}{T \vdash f(b) \oplus f(c)}$$

We can also construct a $\oplus$-lazy proof of $T \vdash f(b) \oplus f(c)$:

$$\text{(GX)}\ \dfrac{\text{(F)}\ \dfrac{\text{(A)}\ \dfrac{a \oplus b \in T}{T \vdash a \oplus b}}{T \vdash f(a) \oplus f(b)} \qquad \text{(F)}\ \dfrac{\text{(A)}\ \dfrac{c \oplus d \in T}{T \vdash c \oplus d}}{T \vdash f(c) \oplus f(d)} \qquad \text{(A)}\ \dfrac{f(a) \oplus f(d) \in T}{T \vdash f(a) \oplus f(d)}}{T \vdash f(b) \oplus f(c)}$$

5.1.3 Different Kinds of Subterm

We now list the different definitions of subterms that we will use to prove a locality theorem in each case.

Definitions of Subterm

Syntactic subterms are defined as usual:

Definition 23 ($S(t)$) *Define the set of* syntactic subterms *of a term t as the smallest set $S(t)$ such that:*

- $t \in S(t)$
- *if* $\langle u, v\rangle \in S(t)$ *then* $u, v \in S(t)$
- *if* $\{u\}_k \in S(t)$ *then* $u, k \in S(t)$
- *if* $u = u_1 \oplus \ldots \oplus u_n \in S(t)$ *then* $atoms(u) \subseteq S(t)$.
- *if* $f(u) \in S(t)$ *then* $u \in S(t)$

The definition of S is extended to a set T of terms in normal form by setting $S(T) := \bigcup_{t\in T} S(t)$.

Example 9 *Let* $T = \{\langle a \oplus f(c), \{d\}_k\rangle, f(a) \oplus f(b) \oplus f^2(c)\}$ *be a set of terms, we compute* $S(T) = T \cup \{a \oplus f(c), \{d\}_k, a, f(c), c, d, k, f(a), f(b), b, f^2(c)\}$.

Let T be a set of terms, we denote $\#(T)$ the number of terms in the set T.

Property 3 *Let* T *be a set of terms then* $\#(S(T)) \leq |T|$.

Proof: It is enough to show it on a term t since

$$\#(S(T)) = \#(\bigcup_{t\in T} S(t)) \leq \sum_{t\in T} \#(S(t)) \leq \sum_{t\in T} |t| = |T|$$

By induction on the size of the term t:

- The base case: t is a constant $\#(S(t)) = 1 = |t|$.

We analyze all possible cases to construct $S(T \cup \{u\})$ according the definition:

- $\#(S(\langle u, v\rangle)) \leq 1 + \#(S(u)) + \#(S(v)) \leq 1 + |u| + |v| = |\langle u, v\rangle|$
- $\#(S(\{u\}_v)) \leq 1 + \#(S(u)) + \#(S(v)) \leq 1 + |u| + |v| = |\{u\}_v|$
- $\#(S(f(u_1, \ldots, u_n)) \leq 1 + \#(S(u_1)) + \ldots + \#(S(u_n)) \leq 1 + |u_1| + \ldots + |u_n| = |f(u_1, \ldots, u_n)|$
- $\#(S(u_1 \oplus \ldots \oplus u_n)) \leq 1 + \#(S(u_1)) + \ldots + \#(S(u_n)) \leq 1 + |u_1| + \ldots + |u_n| = |u_1 \oplus \ldots \oplus u_n|$

□

Property 4 $atoms(M) \subseteq S(M)$ *for any set of terms* $M \subseteq T_\Sigma$.

Proof: Obvious from the definition. □

In the definition of $S(t)$ we do not take care of the homomorphism rule. Because we work only on normal forms the notion of a syntactic subterm ignores the fact that the term $f(a) \oplus f(b) \oplus f(c)$ is equal to $f(a \oplus b \oplus c)$, and that $a \oplus b \oplus c$ should be considered to be a subterm of $f(a) \oplus f(b) \oplus f(c)$. This is accounted for in the next definition.

Definition 24 ($S_T(t)$) *For any term* t*, the set* $S_T(t)$ *is the smallest set such that:*

- $S(t) \subseteq S_T(t)$
- *If* $n \geq 1$ *and* $f(u_1) \oplus \ldots \oplus f(u_n) \in S_T(t)$ *then* $u_1 \oplus \ldots \oplus u_n \in S_T(t)$.

By definition $S(T) \subseteq S_T(T)$. The definition is extended to a set T of terms in normal form by setting $S_T(T) := \bigcup_{t\in T} S_T(t)$.

Example 10 *Let* T *be as in Example 9. Then*

$$S_T(T) = S(T) \cup \{a \oplus b \oplus f(c)\}$$

Property 5 $atoms(S_T(M)) \subseteq S_T(M)$ *for any set of terms* $M \subseteq T_\Sigma$.

Proof: Obvious from Property 4 and the definition of S_T. □

Definition 25 ($S_\oplus(T)$) *Define $S_\oplus$ as all the combinations of terms of $S_T(T)$ by the symbol $\oplus$:*

$$S_\oplus(T) := \Big\{ \bigoplus_{s\in M} s \mid M \subseteq S_T(T) \Big\}$$

Property 6 *For all sets $M \subseteq T(\Sigma)$ we have that*

$$S_\oplus(T) = \Big\{ \bigoplus_{s\in M} s \mid M \subseteq atoms(S_T(T)) \Big\}$$

Proof: For all sets $M \subseteq T(\Sigma)$ we have that:

- According to Property 5 we have $\text{atoms}(S_T(M)) \subseteq S_T(S_T(M))$ with the idempotence of S_T we conclude that
$$\Big\{ \bigoplus_{s\in M} s \mid M \subseteq \text{atoms}(S_T(T)) \Big\} \subseteq S_\oplus(T)$$
- By definition of atoms and S_T, we have that every element of $S_T(T)$ is a sum of some elements of $\text{atoms}(S_T(M))$. Hence, we conclude that
$$S_\oplus(T) \subseteq \Big\{ \bigoplus_{s\in M} s \mid M \subseteq \text{atoms}(S_T(T)) \Big\}$$

□

Note that the size of $S_\oplus$ is exponential in the size of T and that $S_T(T) \subseteq S_\oplus(T)$

Example 11 *Let $T = \{\langle a, b\rangle\}$. Then we get:*

$$S_T(T) = \{\langle a, b\rangle, a, b\}$$

$$S_\oplus(T) = S_T(T) \cup \{a \oplus b, b \oplus \langle a, b\rangle, a \oplus \langle a, b\rangle\}$$

Definition 26 ($S_f(t, h_{max})$) *Let t be a term, and let h_{max} be an upper bound on the number of successive applications of f, to be computed later. We define S_f as:*

$$S_f(t, h_{max}) = \bigcup_{i=0}^{h_{max}} f^i(S_T(t))$$

The definition of $S_f(T, h_{max})$ is extended to a set of terms T in normal form by setting $S_f(T, h_{max}) := \bigcup_{t\in T} S_f(t, h_{max})$.

We remark that if h_{max} is polynomial in the size of T then the size of the set $S_f(T, h_{max})$ is polynomial in the size of T.

Example 12 *Let $T = \{a, b\}$ be a set of terms and $w = f^3(a) \oplus f(b)$. We assume just for the example that $h_{max} = 2$. We compute the set $S_f(T \cup \{w\}, h_{max})$. It is the union of:*

- $S_T(T \cup \{w\}) = \{a, b, f^3(a) \oplus f(b), f^3(a), f^2(a), f(a), f(b), f^2(a) \oplus b\}$
- $f^1(S_T(T \cup \{u\})) = \{f(a), f(b), f^4(a) \oplus f^2(b), f^4(a), f^3(a), f^2(a), f^2(b), f^3(a) \oplus f(b)\}$
- $f^2(S_T(T \cup \{u\})) = \{f^2(a), f^2(b), f^5(a) \oplus f^3(b), f^5(a), f^4(a), f^3(a), f^3(b), f^4(a) \oplus f^2(b)\}$

Properties of Subterm

Lemma 3 (Idempotence of Subterms) *The mappings S, S_T, S_f and $S_\oplus$ are idempotent.*

Proof: This is a consequence of the fact that $S(T)$ and $S_T(T)$ are defined to be the smallest extensions of T satisfying certain closure properties in Definition 24 and 25. We obtain immediately the idempotence of S_f and $S_\oplus$ according to these definitions. □

Lemma 4 (Monotonicity of Subterms) *The mappings S, S_T and $S_\oplus$ are monotone.*

Proof: Straightforward from the definitions of $S(T)$, $S_T(T)$ and $S_\oplus(T)$. □

Lemma 5 (Union of Subterms) *Let A and B be two sets of terms, the mappings S, S_T and $S_\oplus$ have the property:*

- $S(A \cup B) = S(A) \cup S(B)$
- $S_T(A \cup B) = S_T(A) \cup S_T(B)$
- $S_\oplus(A \cup B) = S_\oplus(A) \cup S_\oplus(B)$

Proof: Obvious from the way how the definitions of $S(T)$, $S_T(T)$ and $S_\oplus(T)$ are extended to sets. □

Definition 27 (Transitive Mapping over Sets of Terms) *Let T be a set of terms. The mapping $S : T \to T$ is transitive if for all $X, Y, Z \subseteq T$, $X \subseteq S(Y)$ and $Y \subseteq S(Z)$ implies $X \subseteq S(Z)$.*

Property 7 (Idempotence and Monotonicity $\Rightarrow$ Transitivity) *Let be S a mapping from sets of terms to sets of term. If the mapping S is idempotent and monotone then it is transitive.*

Proof: Let be S a idempotent mapping from a set of terms to a set of term. Let X, Y, Z and T be sets of terms. If $X \subseteq S(Y)$ and $Y \subseteq S(Z)$ by monotonicity and idempotence we get $S(Y) \subseteq S(S(Z) = S(Z)$, hence $X \subseteq S(Z)$. □

Corollary 6 (Transitivity of Subterms) *The mappings S, S_T and $S_\oplus$ are transitive.*

Proof: By Lemma 3 and Lemma 10, and by Proposition 7. □

5.1.4 Properties and Lemmata

Proof Transformations

In this section we show that it is always possible to transform a proof into a $\oplus$-*lazy* proof or an $\oplus$-*eager* proof.

- We can always flatten successive applications of (GX) rules into a single one, as shown in Figure 5.1.
- We can always switch a rule (GX) with a rule (F) if (GX) is immediately above (F), as shown in Figure 5.2.
- The reverse operation is possible only if all immediate predecessors in the proof tree are labeled with (F), as shown in Figure 5.3.

Lemma 7 *If there is a proof of $T \vdash w$ then there is also a $\oplus$-lazy proof and a $\oplus$-eager proof of $T \vdash w$*

Proof:

$$\text{(GX)}\ \dfrac{\text{(GX)}\ \dfrac{T \vdash x_1 \ldots T \vdash x_n}{T \vdash x_1 \oplus \ldots \oplus x_n} \qquad T \vdash y_1 \qquad \ldots \qquad T \vdash y_m}{T \vdash x_1 \oplus \ldots \oplus x_n \oplus y_1 \oplus \ldots \oplus y_m}$$

$$\Downarrow$$

$$\text{(GX)}\ \dfrac{T \vdash x_1 \quad \ldots \quad T \vdash x_n \quad T \vdash y_1 \quad \ldots \quad T \vdash y_m}{T \vdash x_1 \oplus \ldots \oplus x_n \oplus y_1 \oplus \ldots \oplus y_m}$$

Figure 5.1: Transformation of (GX)-(GX) into (GX)

$$\text{(F)}\ \dfrac{\text{(GX)}\ \dfrac{T \vdash x_1 \ldots T \vdash x_n}{T \vdash x_1 \oplus \ldots \oplus x_n}}{T \vdash f(x_1) \oplus \ldots \oplus f(x_n)} \quad \Longrightarrow \quad \text{(GX)}\ \dfrac{\text{(F)}\ \dfrac{T \vdash x_1}{T \vdash f(x_1)} \quad \cdots \quad \text{(F)}\ \dfrac{T \vdash x_n}{T \vdash f(x_n)}}{T \vdash f(x_1) \oplus \ldots \oplus f(x_n)}$$

Figure 5.2: Transformation of (GX)-(F) into (F)-(GX)

$$\text{(GX)}\ \dfrac{\text{(F)}\ \dfrac{T \vdash x_1}{T \vdash f(x)} \quad \cdots \quad \text{(F)}\ \dfrac{T \vdash x_n}{T \vdash f(x_n)} \quad (G_1)\ \dfrac{T \vdash y_1}{T \vdash z_1} \quad \cdots \quad (G_m)\ \dfrac{T \vdash y_m}{T \vdash z_m}}{T \vdash f(x_1) \oplus \ldots f(x_n) \oplus z_1 \oplus \ldots \oplus z_m}$$

$$\Downarrow$$

$$\text{(GX)}\ \dfrac{\text{(F)}\ \dfrac{\text{(GX)}\ \dfrac{T \vdash x_1 \ldots T \vdash x_n}{T \vdash x_1 \oplus \ldots \oplus x_n}}{T \vdash f(x_1) \oplus \ldots \oplus f(x_n)} \quad (G_1)\ \dfrac{T \vdash y_1}{T \vdash z_1} \quad \cdots \quad (G_m)\ \dfrac{T \vdash y_m}{T \vdash z_m}}{T \vdash f(x_1) \oplus \ldots \oplus f(x_n) \oplus z_1 \oplus \ldots \oplus z_m}$$

Figure 5.3: Transformation of (F)-(GX) into (GX)-(F) where rules (G_i) are all different from (F)

- To obtain a $\oplus$-*lazy* proof we apply the proof transformation rules given in Figures 5.2 and 5.1.
- To obtain an $\oplus$-*eager* proof we apply the proof transformation rules given in Figures 5.3 and 5.1.

$\square$

Definition 28 (Minimal-$\oplus$-*lazy* Proof) *P is a* minimal-$\oplus$-*lazy proof of $T \vdash w$ if only if P is a $\oplus$-lazy proof of $T \vdash w$ which is minimal among all $\oplus$-lazy proofs.*

If there is a proof of $T \vdash w$ then there is also a minimal-$\oplus$-*lazy* proof and a $\oplus$-*eager* proof of $T \vdash w$.

Lemma 8 *Let P be a $\oplus$-lazy simple proof of $T \vdash w$. If all terms in $T \cup \{w\}$ have at most two atoms then the proof P is binary $\oplus$-lazy and simple.*

Proof: By Corollary 20, in a $\oplus$-*lazy* proof all nodes are in $S_T(T, w)$ except those produced by the rule (F) and which lead by a sequence of (F) rules to (GX). These nodes are binary, too, since application of (F) preserves the number of atoms of a term. Hence, the proof P is binary $\oplus$-*lazy* and simple. $\square$

A Lemma about Destruction Rules

Lemma 10 and Lemma 9 are two extensions of a lemma shown in [CLS03] for the $ACUN$ theory. We show that these lemmata hold in the case of Abelian groups with homomorphism, the demonstration can easily be adapted to the case of AC and $ACUN$ plus homomorphism.

Lemma 9 *Let P' be a simple proof of the form:*

$$P' = \left\{ \quad \frac{P'_1 \dots P'_n}{T \vdash w} \right.$$

1. *If $T \vdash u$ does not occur in any of $P'_1, \dots, P'_n$ and $\langle u, v\rangle\!\downarrow \in S(w)$ then there is at least one P'_i and there exists w' such that $\langle u, v\rangle\!\downarrow \in S(w')$ and either the root of P'_i is $T \vdash w'$ or $w' \in T$.*
2. *If $T \vdash u$ does not occur in any of $P'_1, \dots, P'_n$ and $\{u\}_v\!\downarrow \in S(w)$ then there is at least one P'_i and there exists w' such that $\{u\}_v\!\downarrow \in S(w')$ and either the root of P'_i is $T \vdash w'$ or $w' \in T$.*

Proof: We only give the proof for the first case (pairing), the second case is similar. We consider all possible rules for the root of P':

- The last rule is (A) it is obvious.
- The last rule is (UL) or (UR): $\langle u, v\rangle\!\downarrow \in S(w)$ by hypothesis, and by construction $w \in S(\langle u_1, u_2\rangle)$. We deduce by transitivity of the subterm relation that $\langle u, v\rangle\!\downarrow \in S(\langle u_1, u_2\rangle)$.
- The last rule is (D): as in the above case but with $\{u\}_v$ instead of $\langle u, v\rangle$.
- The last rule is (F): $\langle u, v\rangle$ is a subterm of $w = f(w_1)\!\downarrow$, it hence is a subterm of w_1.
- The last rule is (GX): $\langle u, v\rangle$ is a subterm of $(u_1 \oplus \dots \oplus u_n)\!\downarrow$, it is hence a subterm of one of some u_i, because $\langle u, v\rangle$ is not headed with $\oplus$.
- The last rule is (P): since $T \vdash u$ can not be in P then $w = \langle w_1, w_2\rangle \neq \langle u, v\rangle$. But $\langle u, v\rangle$ is a subterm of w so it is a subterm of w_1 or of w_2.
- The last rule is (C): since $T \vdash u$ can not be in P then $w = \{w_1\}_{w_2} \neq \{u\}_v$. But $\langle u, v\rangle$ is a subterm of w so it is a subterm of w_1 or of w_2. $\square$

$$
(\mathrm{UL})\ \dfrac{\dfrac{\vdots}{T \vdash \langle u,v\rangle\!\downarrow = r}}{T \vdash u}
\qquad
(\mathrm{UR})\ \dfrac{\dfrac{\vdots}{T \vdash \langle u,v\rangle\!\downarrow = r}}{T \vdash v}
\qquad
(\mathrm{D})\ \dfrac{\dfrac{\vdots}{T \vdash \{u\}_v\!\downarrow = r} \qquad \dfrac{\vdots}{T \vdash v\!\downarrow}}{T \vdash u}
$$

Figure 5.4: Destruction Rules.

Lemma 10 *Let P be a simple proof of one of the forms described in Figure 5.4.*
Then $\langle u,v\rangle\!\downarrow \in S(T)$ (resp. $\{u\}_v\!\downarrow \in S(T)$).

Proof: Assume that the last rule is (UL), the case (UR) is similar.

$$
P = \left\{ \dfrac{\dfrac{P_1 \ldots P_n}{T \vdash \langle u,v\rangle\!\downarrow = r}}{T \vdash u} \right.
$$

P is minimal so $T \vdash u$ does not occur in any of $P_1, \ldots, P_n$. Hence, we can apply the Lemma 9. The result follows by induction from Lemma 9.

Assume that the last rule is (D):

$$
P = \left\{ (\mathrm{D})\ \dfrac{\dfrac{P_1 \ldots P_n}{T \vdash \{u\}_v} \qquad \dfrac{\vdots}{T \vdash v}}{T \vdash u} \right.
$$

P is minimal so $T \vdash u$ does not occur in any of $P_1, \ldots, P_n$. Hence, we can apply the Lemma 9. The result follows again by induction from Lemma 9. □

5.2 Locality for the Equational Theory ACh

5.2.1 Observations and Technical Lemmata for the Theory ACh

We remark that:

- If $t \in S_T(u) \setminus u$ then $t \in S_T(\text{atoms}(u))$
- $\forall u, S_T(\text{atoms}(u)) \subseteq S_T(u)$

The following property holds in the equational theory ACh: $\forall u, v, \text{atoms}(u) \subseteq \text{atoms}(u \oplus v)$ since in this case no term is deleted by the application of $\oplus$. This property is used to state:

Lemma 11 *Let u, v and t be terms in normal form. If $t \in S_T(u) \setminus u$ then $t \in S_T(u \oplus v)$.*

Proof: Let u, v and t be terms in normal form, if $t \in S_T(u) \setminus u$ then $t \in S_T(\text{atoms}(u))$. By monotonicity of S_T and the previous observation $\forall u, S_T(\text{atoms}(u)) \subseteq S_T(u \oplus v)$ we obtain that $t \in S_T(u \oplus v)$. □

Lemma 12 *In case of the equational theory ACh we have for all terms t, u, v in normal form: if*

- *$t \neq u$ or t is not headed with $\oplus$*

- *and* $t \in S_T(u)$

then $t \in S_T(u \oplus v)$.

Note that this lemma is no longer true in case of the equational theories ACUNh or AGh.
Proof: We consider two cases:

- If $t \neq u$ then $t \in S_T(u) \setminus u$. By Lemma 12 we conclude that $t \in S_T(u \oplus v)$.
- If $t = u$ then t is not headed with $\oplus$. Hence $t \in S_T(u) \setminus u$. By Lemma 12 we conclude that $t \in S_T(u \oplus v)$. □

5.2.2 Lemmata and Locality

Lemma 13 *Let P be a $\oplus$-lazy simple proof of $T \vdash w$ and let P' be a subproof of P with root label $T \vdash N$. If the last rule applied in P' is (GX) then $N \in S_T(w) \cup S_T(T)$.*

Proof: By induction on the size of the proof P. The base case is trivial. Let us assume that P is of the form

$$P = \left\{ \begin{array}{c} P_1 = \left\{ \begin{array}{c} \vdots \\ \hline T \vdash R_1 \end{array} \right. \qquad \ldots \qquad P_n = \left\{ \begin{array}{c} \vdots \\ \hline T \vdash R_n \end{array} \right. \\ \hline T \vdash w \end{array} \right.$$

Assume that all immediate subproofs P_i of P with root label $T \vdash R_i$ satisfy the property *i.e.* if there is the rule (GX) in P_i then the node resulting from this rule application is labeled by $T \vdash N_i$ where $N_i \in S_T(R_i) \cup S_T(T)$.

By induction hypothesis, any immediate subproof P_i of P with root label $T \vdash R_i$ satisfies the property *i.e.* if there is the rule (GX) in P_i then the node resulting from this rule application is labeled by $T \vdash N_i$ with $N_i \in S_T(R_i) \cup S_T(T)$.

We proceed by case analysis on the last rule applied in P:

1. (P), (C), (F): In this case any application of rule (GX) occurs in some subproof P_i. For any of these rules, $S_T(R_i) \subseteq S_T(w)$ for all i. The result follows from the induction hypothesis.
2. (UL) (UR) or (D): In this case any application of rule (GX) occurs in some subproof P_i. By Lemma 10 we have that $S_T(R_i) \subseteq S_T(T)$ for all i. The result follows from the induction hypothesis.
3. (GX): Let P_N be a subproof of P which has $T \vdash N$ as label of the root and which terminates with an application of the rule (GX).
   (a) If P_N is P itself then $N = w$ and the claim obviously holds.
   (b) If P_N is a proper subproof of P then P_N is a subproof of some P_i.
   By induction hypothesis we have that $N \in S_T(T) \cup S_T(R_i)$.
   If $N \in S_T(T)$ we are done, otherwise let P_i' be the subproof of P_i such that P_i is obtained from P_i' by a sequence (possibly empty) of applications of (F) and such that the last rule applied in P_i' is not (F).
   Let $T \vdash R_i'$ be the label of the root of P_i'.
   By induction, it is enough to show that $R_i' \in S_T(T \cup \{w\})$.
   By definition of P_i', the last rule application in P_i' cannot be (F), and since the proof P is $\oplus$-*lazy* it cannot be (GX) neither.
   There are two possible cases: R_i' is headed with $\oplus$ or not.

i. If R'_i is headed with $\oplus$: the last rule application in P'_i cannot be (P) or (C), and it cannot be (GX) nor (F) as seen above. Hence, the last rule application of P'_i is one of (A), (D), (UL) or (UR).
By Lemma 10, we get $R'_i \in S_T(T)$.

ii. If R'_i is not headed with $\oplus$ then R_i is not headed by $\oplus$ neither. By Lemma 12, we get $N \in S_T(T \cup \{w\})$. $\square$

Lemma 14 *Let P be a $\oplus$-lazy simple proof of $T \vdash w$ and let P' be a subproof of P with root label $T \vdash N$. If the last rule applied in P' is (P) or (C), then $N \in S_T(w) \cup S_T(T)$.*

Proof: By induction on the size of the proof P. The base case is trivial. Let us assume that P is of the form

$$P = \left\{ \frac{P_1 = \left\{ \frac{\vdots}{T \vdash R_1} \right. \qquad \dots \qquad P_n = \left\{ \frac{\vdots}{T \vdash R_n} \right.}{T \vdash w} \right.$$

Assume that all immediate subproofs P_i of P satisfy the property. By definition, if $T \vdash R_i$ is the label of the root of P_i and if $T \vdash N_i$ is the label of a node of P_i obtained by the application of the rule (P) or (C), we have $N_i \in S_T(R_i) \cup S_T(T)$.

We proceed by case analysis on the last rule applied in the proof P:

1. (P), (C): Let P_N be a subproof of P which terminates with an application of the rule (P) or (C) and let $T \vdash N$ be the label of the root of P_N.
   (a) If P_N is P then $N = w$ and the claim obviously holds.
   (b) If P_N is a proper subproof of P then P_N is a subproof of some P_i.
   By induction hypothesis we have that $N \in S_T(T) \cup S_T(R_i) \subseteq S_T(T) \cup S_T(w)$.

2. (F): In this case any application of the rule (P) or (C) occurs in some subproof P_i.
   By definition of (F) and S_T, we have $S_T(R_i) \subseteq S_T(u)$ for all i.
   Then the result follows from the induction hypothesis.

3. (UL) (UR) or (D): In this case any application of rule (P) or (C) occurs in some subproof P_i.
   By Lemma 10 we have that $S_T(R_i) \subseteq S_T(T)$ for all i.
   Then the result follows from the induction hypothesis.

4. (GX): In this case any application of rule (P) or (C) occurs in a subproof P_i.
   By induction hypothesis we have that $N \in S_T(T) \cup \subseteq S_T(R_i)$.
   Since N is not headed with $\oplus$, the result follows from Lemma 12. $\square$

Lemma 15 *Let P be a $\oplus$-lazy simple proof of $T \vdash w$ and let P' a subproof of P with root label $T \vdash N$. If the last rule applied in P' is (F) then $N \in S_T(w) \cup S_T(T)$.*

Proof: By induction on the size of the proof P. The base case is trivial. Let us assume that P is of the form

$$P = \left\{ \frac{P_1 = \left\{ \frac{\vdots}{T \vdash R_1} \right. \qquad \dots \qquad P_n = \left\{ \frac{\vdots}{T \vdash R_n} \right.}{T \vdash w} \right.$$

Assume that all immediate subproofs P_i of P with root label $T \vdash R_i$ satisfy the property, *i.e.* if $T \vdash N_i$ is the label of a node of P_i resulting from the application of the rule (F), then $N_i \in S_T(R_i) \cup S_T(T)$.

We proceed by case analysis on the last rule applied in P:

1. (P), (C): In this case any application of rule (F) occurs in some subproof P_i.

   For these rules $S_T(R_i) \subseteq S_T(u)$ for all i.

   Then the result holds by induction hypothesis.

2. (F): Let P_N be a subproof P_N of P which terminates with an application of the rule (F) and let $T \vdash N$ the label of P_N.

   (a) If P_N is P then $N = w$ and the claim holds.

   (b) If P_N is a proper subproof of P then P_N is a subproof of some P_i.
   By induction hypothesis we have that $N \in S_T(T) \cup S_T(R_i) \subseteq S_T(T) \cup S_T(w)$.

3. (UL) (UR) or (D): In this case any application of rule (F) occurs in one subproof P_i.

   By Lemma 10 we have that $S_T(R_i) \subseteq S_T(T)$ for all i, and the result holds by induction hypothesis.

4. (GX): In this case any application of rule (F) occurs in one subproofP_i, hence we have that $N \in S_T(T) \cup S_T(R_i)$ by induction hypothesis.

   If $N \in S_T(T)$ we are done, otherwise let P'_i be the subproof of P_i such that P'_i does not terminate on (F), and P_i is obtained from P'_i by a sequence of applications of (F).

   Let $T \vdash R'_i$ be the label of the root of P'_i.

   We show that $R'_i \in S_T(T \cup \{w\})$, which yields the result by induction hypothesis.

   By definition the last rule application of P'_i cannot be (F), and since the proof P is $\oplus$-*lazy* it also cannot be (GX) neither.

   There are two possible cases: R'_i are headed with $\oplus$ or not.

   (a) If R'_i is headed with $\oplus$: the last rule application in P'_i cannot be (P) or (C), and it cannot be (GX) or (F) as seen above.
   Therefore the last rule application of P'_i is (A), (D), (UL) or (UR).
   By Lemma 10 we get $R'_i \in S_T(T)$.

   (b) If R'_i not headed with $\oplus$ then then R_i is not headed by $\oplus$ either, and we conclude by Lemma 12 that $N \in S_T(T \cup \{w\})$. □

Theorem 3 (Locality) *The proof system for $\vdash$ in the case of the equational theory ACh is S_T-local.*

Proof: If there is a proof of $T \vdash w$ then, by Lemma 7 there also is a $\oplus$-*lazy* and simple proof of $T \vdash w$. By Lemmata 10, 13, 14 and 15 this proof is S_T-local. □

Lemma 16 *Let $T \in T(\Sigma)$ and $w \in T(\Sigma)$ be terms such that atoms(T, w) are constants. If P is a proof of $T \vdash w$ then there exist a minimal proof of $T \vdash w$ using only the rules (A) and (GX).*

Proof: Let P be a minimal-$\oplus$-*lazy* proof of $T \vdash w$. □

5.3 Locality for the Equational Theory *ACUNh*

5.3.1 Properties of $\oplus$-*lazy* Proofs in the *ACUNh* Case

Lemma 17 *Let P be a minimal-$\oplus$-lazy proof of $T \vdash w$ and let P' be a subproof of P with root label $T \vdash N$. If the last rule application is (GX) then $N \in S_T(w) \cup S_T(T)$.*

Note that, in contrast to the analogous lemma in the case *ACh* (Lemma 13), we now assume a *minimal-$\oplus$-lazy* proof instead of merely assuming a *simple-$\oplus$-lazy* proof.

Proof: By induction on the size of the proof P. The base case is trivial. Let us assume that P is of the form

$$P = \left\{ \begin{array}{c} P_1 = \left\{ \begin{array}{c} \vdots \\ \hline T \vdash R_1 \end{array} \right. \qquad \dots \qquad P_n = \left\{ \begin{array}{c} \vdots \\ \hline T \vdash R_n \end{array} \right. \\ \hline T \vdash w \end{array} \right.$$

Assume that all immediate subproofs P_i of P with root label $T \vdash R_i$ verify the property *i.e.* if the rule (GX) occurs in P_i and if $T \vdash N_i$ is the label of the node resulting from this rule application, then $N_i \in S_T(R_i) \cup S_T(T)$.

We proceed by case analysis on the last rule applied in the proof P:

1. (P), (C), (F): In this case any application of rule (GX) occurs in a subproof P_i. For these rules $S_T(R_i) \subseteq S_T(w)$ for all i.

   Then the result follows from the induction hypothesis.

2. (UL) (UR) or (D): In this case any application of rule (GX) occurs in a subproof P_i.

   By Lemma 10 we have that $S_T(R_i) \subseteq S_T(T)$ for all i.

   Then the result follows from the induction hypothesis.

3. (GX): Consider a subproof P_N of P which terminates with an application of the rule (GX) and whose root is labeled by $T \vdash N$.

   (a) If P_N is P itself then $N = w$ and the claim obviously holds.

   (b) If P_N is a proper subproof of P then P_N is a subproof of one of the P_i.
   By induction hypothesis we have that $N \in S_T(T) \cup S_T(R_i)$.
   If $N \in S_T(T)$ we are done, otherwise let P'_i be the subproof of P_i such that P'_i does not terminate on (F), and P_i is obtained from P'_i by a sequence of n applications of (F), illustrated in Figure 5.5.
   Let $T \vdash R'_i$ be the label of the root of P'_i.
   It is enough to show that $R'_i \in S_T(T \cup \{w\})$ since the result will follow by induction hypothesis.
   By definition of P'_i, the last rule application in P'_i is not (F), and since the proof P is $\oplus$-*lazy* it also cannot be (GX) neither.
   There are two possible cases: R'_i is headed with $\oplus$ or not. Note that R'_i is headed with $\oplus$ iff R_i is headed with $\oplus$.

      i. If R'_i is headed with $\oplus$: the last rule application in P'_i cannot be (P) or (C), and it cannot be (GX) or (F) as seen above.
      Hence, the last rule application of P'_i is one of (A), (D), (UL) or (UR).
      By Lemma 10, we get $R'_i \in S_T(T)$.

$$
\text{(GX)}\;\dfrac{\cdots \qquad \text{(F)}\;\dfrac{\text{(F)}\;\dfrac{\vdots}{\text{(F)}\;\dfrac{\text{(}\neq\text{F,GX)}\;\dfrac{\vdots}{T \vdash R'_i}}{\vdots}}}{T \vdash R_i = f^n(R'_i)} \qquad \cdots \qquad \text{(F)}\;\dfrac{\vdots}{\text{(F)}\;\dfrac{\text{(}\neq\text{F,GX)}\;\dfrac{\vdots}{T \vdash R'_j = R'_i \oplus A'_i}}{\vdots}}\;\Big/\; T \vdash R_j = R_i \oplus A_i \qquad \cdots}{T \vdash R_1 \oplus \ldots \oplus R_n = w}
$$

Figure 5.5: Illustration of the decomposition used in the proofs of Lemmata 17 and 18.

ii. If R'_i is not headed with $\oplus$: Either $R_i \in S_T(w)$ and we are done, or $R_i \notin S_T(w)$ and R_i is canceled out by another term in the (GX) rule.
Since the proof P is minimal, R_i must be canceled out by some term R_j which is headed with $\oplus$ and which is obtained by a subproof P_j: otherwise the same term would appear twice in the sum, and a smaller proof could be obtained just by omitting these two terms, which contradicts the minimality of P.
Therefore we can write $R_j = R_i \oplus A_i$.
Since P is a $\oplus$-*lazy* proof and since the last rule of P'_i cannot be (GX) or (F) by definition, Lemma 10 yields the result if the last rule of P'_i is one of (UL), (UR), (D) or (A).
The remaining case is when this rule is (P) or (C).
In this case we observe that R'_i is not headed by f and that $\#_f(R_i) = n$.
We now go up from $R_j = R_i \oplus A_i$ as long as possible along a sequence of (F) rules: let P'_j be the subproof of P_j such that P'_j does not terminate on (F), and such that P_j is obtained from P'_j by a sequence of m applications of (F).
Since $R_i \in \text{atoms}(R_j)$, we have $m \leq \#_f(R_j) \leq \#_f(R_i) = n$.
Let $R''_i \oplus A''_i$ be the label of the root of P'_j.
Since P is a $\oplus$-*lazy* proof, the last rule of P'_j cannot be (GX) or (F) by construction. It also cannot be (P) nor (C) because $R''_i \oplus A''_i$ is headed with $\oplus$.
Hence, the last rule of P'_j must be one of (A), (UR), (UL), (D). We have that $S_T(R''_i \oplus A_i) \subseteq S_T(T)$ by Lemma 10.
It follows that $R'_i \in S_T(R''_i \oplus A''_i) \subseteq S_T(T)$ because $m \leq n$, and we conclude by applying the induction hypothesis. □

Lemma 18 *Let P be a minimal-$\oplus$-lazy proof of $T \vdash w$ and let P' be a subproof of P with root label $T \vdash N$. If the last rule applied in P' is (P) or (C), then $N \in S_T(w) \cup S_T(T)$.*

Proof: By induction on the size of the proof P. The base case is trivial. Let us assume that P is of the form

$$
P = \left\{ \dfrac{P_1 = \left\{ \dfrac{\vdots}{T \vdash R_1} \right. \qquad \ldots \qquad P_n = \left\{ \dfrac{\vdots}{T \vdash R_n} \right.}{T \vdash w} \right.
$$

Assume that all immediate subproofs P_i of P with root label $T \vdash R_i$ verify the property *i.e.* if there is the rule (P) or (C) in P_i then the node resulting from this rule application and labeled by $T \vdash N_i$ satisfies that $N_i \in S_T(R_i) \cup S_T(T)$.

We proceed by case analysis on the last rule applied in the proof P:

1. (P), (C): Let P_N be a subproof of P which terminates with an application of the rule (P) or (C) and let $T \vdash N$ be the label of the root of P_N.

(a) If P_N is P then $N = w$ and the claim obviously holds.

(b) If P_N is a proper subproof of P then P_N is a subproof of one of the P_i.
By induction hypothesis we have that $N \in S_T(T) \cup S_T(R_i) \subseteq ST(T) \cup S_T(w)$.

2. (F): In this case any application of rule (P) or (C) occurs in some subproof P_i.

Since $S_T(R_i) \subseteq S_T(u)$ for all i for this rule, the result follows from the induction hypothesis.

3. (UL), (UR) or (D): In this case any application of rule (P) or (C) occurs in some subproof P_i.

By Lemma 10, $S_T(R_i) \subseteq S_T(T)$ for all i, and the result follows from the induction hypothesis.

4. (GX): In this case any application of rule (P) or (C) occurs in some subproof P_i.

By induction hypothesis we have that $N \in S_T(T) \cup S_T(R_i)$.

If $N \in S_T(T)$ we are done, otherwise let P'_i be the subproof of P_i such that P'_i does not terminate on (F), and P_i is obtained from P'_i by a sequence of n applications of (F), as illustrated in Figure 5.5.

Let $T \vdash R'_i$ be the label of the root of P'_i.

It is enough to show that $R'_i \in S_T(T \cup \{w\})$ since the result will follow from the induction hypothesis.

By definition the last rule application of P'_i cannot be (F) and it and cannot be (GX) since the proof P is $\oplus$-*lazy*.

There are two possible cases: R'_i is headed with $\oplus$ or not. Note that R'_i is headed with $\oplus$ iff R_i is headed with $\oplus$.

(a) If R'_i is headed with $\oplus$: the last rule application in P'_i cannot be (P) or (C), and it cannot be (GX) or (F) as seen above. Hence, the last rule application of P'_i must be one of (A), (D), (UL) or (UR).
By Lemma 10 we get $R'_i \in S_T(T)$.

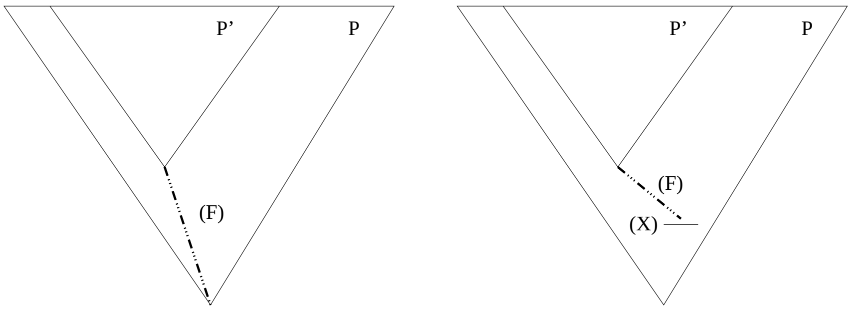
(b) If R'_i is not headed with $\oplus$: Either $R_i \in S_T(w)$ and we are done, or $R_i \notin S_T(w)$ and R_i is canceled out by another term in the (GX) rule.
Since P is minimal, R_i must be canceled out by some term R_j which is headed with $\oplus$ and which is obtained by a subproof P_j: otherwise the same term would appear twice in the sum, and we could obtain a smaller proof just by omitting these two terms, which contradicts the minimality of P.
Therefore $R_j = R_i \oplus A_i$.
Since P is a $\oplus$-*lazy* proof and since the last rule of P'_i cannot be (GX) or (F), the result holds by Lemma 10 if it is one of (UL), (UR), (D) or (A).
The remaining case is when the last rule of P'_i is one of the rules (P) or (C)
In this case we observe that R'_i is not headed by f, and $\#f(R_i) = n$.
We now go up from $R_j = R_i \oplus A_i$ as long as possible along a sequence of (F) rules : let P'_j be the subproof of P_j such that P'_j does not terminate on (F), and such that P_j is obtained from P'_j by a sequence of m applications of (F).
Since $R_i \in \text{atoms}(R_j)$ we have $m \leq \#f(R_j) \leq \#f(R_i) = n$.
Let $R''_i \oplus A''_i$ be the label of the root of P'_j.
Since P is a $\oplus$-*lazy* proof and by construction, the last rule of P'_j cannot be (GX) or (F). It also cannot be (P) or (C) because $R''_i \oplus A''_i$ is headed with $\oplus$.
Hence, the last rule of P'_j must be one of (A), (UR), (UL), (D). By the Lemma 10 we have that $S_T(R''_i \oplus A_i) \subseteq S_T(T)$.
If follows that $R'_i \in S_T(R''_i \oplus A''_i) \subseteq S_T(T)$ because $m \leq n$, and we conclude by applying the induction hypothesis. □

Lemma 19 *Let P be a proof which is minimal-$\oplus$-lazy proofs of $T \vdash w$, and let P' be a subproof of P with root label $T \vdash N$ such that the last rule applied in P' is (F). If either*

1. *all nodes from the root of P' to the root of P are (F),*
2. *or if the first successor not labeled by (F) of the root of P' in P is labeled by a rule different from (GX),*

then $N \in S_T(T \cup \{w\})$.

This lemma states that (F) nodes are harmless as long as they do not produce an hypothesis of a (GX) rule with a sequence of successive (F) nodes. The two cases of the lemma can be illustrated like this:



In the right picture, (X) denotes a rule different from (F) and from (GX).
Proof: If all nodes from the root of P' to the root of P are obtained by (F) rules, then we get $N \in S_T(w)$ by induction hypothesis: otherwise let (Z) (different from (GX)) be the label of the first successor in P of the root of P' which is not labeled by (F):

- (Z) cannot be (A).
- (Z) cannot be (UL), (UR), (D) since the hypothesis of (Z) is headed by f.
- Therefore (Z) must be (P) or (C), and in this case we conclude by Lemma 18. □

Corollary 20 *Let P be a minimal-$\oplus$-lazy proof of $T \vdash w$. All nodes of P are labeled by terms in $S_T(T, w)$ except the nodes which are constructed by (F) and which lead to a hypothesis of a (GX) rule by a sequence of (F) rules.*

Proof: By Lemmata 17, 18, and 19. □

Since the application of rule (F) to a binary term yields a binary term we obtain immediately:

Corollary 21 *If T and w are binary then every minimal-$\oplus$-lazy proof of $T \vdash w$ is binary.*

Proof: According to Lemma 20 all nodes are in $S_T(T, w)$ except the nodes which are constructed by (F) and which yield an hypothesis of a (GX) rule by a sequence of (F) rules. If T and w are binary then the construction of $S_T(T, w)$ does not generate any term headed with $\oplus$ which is not binary. Since (F) cannot create binary terms, every minimal-$\oplus$-*lazy* proof of $T \vdash w$ is binary. □

Example 13 *We present a minimal proof of $T \vdash w$, where $T = \{u \oplus v, f(v)\}, w = f(u)$.*

$$\textit{(GX)}\;\dfrac{\textit{(F)}\;\dfrac{\textit{(A)}\;\dfrac{u \oplus v \in T}{T \vdash u \oplus v}}{T \vdash f(u) \oplus f(v)} \qquad \textit{(A)}\;\dfrac{f(v) \in T}{T \vdash f(v)}}{T \vdash f(u)}$$

We compute $S_T(T \cup \{w\}) = \{u, v, u \oplus v, f(u), f(v)\}$. This proof is not S_T-local since $f(u) \oplus f(v) \notin S_T(T \cup \{w\})$

As seen in this example, the problem in defining S-locality for a polynomial-size S is to bound the number of applications of the (F) proof rule when constructing hypotheses to a (GX) rule.

5.3.2 Locality in the Binary Case

In the *binary* case, that is when all terms in $S_T(T, w)$ have at most two atoms, we can actually find an upper bound for the number of applications of (F). The key to this upper bound is a technical lemma about counter automata which is proved in the appendix.

In the binary case we associate a one-counter automaton to the set $S_T(T, w)$. The states of the automaton are atoms of $S_T(T, w)$ without top-level f-symbols, and the counter represents the number of applications of f to a term. We first give an informal explanation of the construction. The formal definition of the automaton is given below (Definition 29).

- First Step: For every term $t \in S_T(T, w)$:
  - If t is headed with $\oplus$ then for each atom a of t the term $\mathit{Strip}_f(a)$ is a state of the automaton. As we can see in Example 14, the term $a \oplus f^2(b)$ gives raise to two states a and b. We add oriented transitions between the two states. If the transition starts from a which stems from $f^n(a)$ and goes to state b which stems from $f^m(b)$, the transition has condition $c \geq n$ and action $c := c + m - n$. We add the symmetric transition from b to a.
  - If t is not headed with $\oplus$ we create a state $\mathit{Strip}_f(t)$ decorated with a prime, and a transition from this state to itself, without condition and with action $c := c + 1$.
- Second Step: Merge all states labeled with the same term.
- Third Step: Connect every primed state a' stemming from some term $f^n(t)$ to its unprimed equivalent a by a transition oriented from the primed state to the unprimed state with condition $c \geq n$ and with action $c := c$.
- Forth Step: Create a new initial state, and a transition with (vacuous) condition $c \geq 0$ and with action $c := c$ from the initial state to any primed state.

We now give the formal definition of the automaton:

Definition 29 *Let T be a set of terms such that every term in T has at most two atoms. The* automaton associated with T, *abbreviated A_T, is a one-counter automaton without input defined as follows:*

We partition $T = T_1 \uplus T_2$ where T_1 is the set of terms not headed with $\oplus$, and T_2 is the set of terms headed with $\oplus$.

- *The set of states Q_T of A_T is defined as*

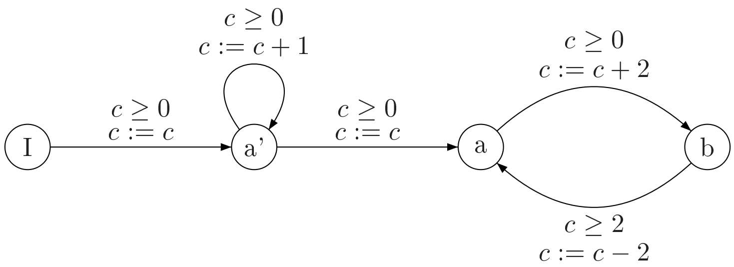
$$\begin{aligned} Q_T &= P_T \cup R_T \cup \{\text{INIT}\} \\ P_T &= \{p' \mid p \in \mathit{Strip}_f(T_1)\} \\ R_T &= \{r \mid r \in \mathit{Strip}_f(T_1) \cup \mathit{Strip}_f(\mathit{atoms}(T_2))\} \end{aligned}$$

- INIT *is the initial state of A_T.*
- *The set of transitions is:*

	From	*To*	*Condition*	*Action*
$\forall t \in T_1 :$	INIT	$(\mathit{Strip}_f(t))'$	$c \geq 0$	$c := c$
$\forall t \in T_1 :$	$(\mathit{Strip}_f(t))'$	$(\mathit{Strip}_f(t))'$	$c \geq 0$	$c := c + 1$
$\forall t \in T_1 :$	$(\mathit{Strip}_f(t))'$	$\mathit{Strip}_f(t)$	$c \geq \#_f(t)$	$c := c$
$\forall t \oplus s \in T_2 :$	$\mathit{Strip}_f(t)$	$\mathit{Strip}_f(s)$	$c \geq \#_f(t)$	$c := c - \#_f(t) + \#_f(s)$

In the definition of the automaton, the primed states in P_T are only intermediate states which do not correspond to a proof; only the states in R_T correspond to proofs in a sense to be made precise below. Note that in the last line of the above transition table the statement "$t \oplus s \in T_2$" is to be understood modulo AC, such that we obtain from a binary clause a back and a forth transition, as already explained in the informal definition of the automaton.

Example 14 *The automaton A_T for $T = \{a \oplus f^2(b), a\}$ is as follows, where I denotes the initial state:*



Lemma 22 *Let T be a set of binary terms. For all terms $t_0, t_1, \ldots, t_n \in Strip_f(atoms(T))$ and all natural numbers $c_0, c_1, \ldots, c_n$ we have that:*

$$A_T \models (\text{INIT}, 0) \to (t'_0, 0) \to \ldots \to (t'_0, c_0) \to (t_0, c_0) \to (t_1, c_1) \to \ldots \to (t_n, c_n)$$

iff there are terms $s_0, s_1, \ldots, s_n \in T$ and natural numbers $d_0, d_1, \ldots, d_n$ such that:

1. *s_0 is not headed with $\oplus$, and for $1 \leq i \leq n$ the term s_i is headed with $\oplus$ and has exactly two atoms, that is $s_i = s_i^1 \oplus s_i^2$*
2. *$f^{d_0}(s_0) = f^{c_0}(t_0)$*
3. *$\forall 1 \leq i \leq n : f^{d_i}(s_i^1) = f^{c_{i-1}}(t_{i-1})$*
4. *$\forall 1 \leq i \leq n$: $f^{d_i}(s_i^2) = f^{c_i}(t_i)$*

As a consequence, we obtain that

$$\bigoplus_{i=0}^{n} f^{d_i}(s_i)\downarrow = f^{d_n}(s_n^2) = f^{c_n}(t_n)$$

Example 15 *We continue Example 14. With this automaton, we have a run*

$$(\text{INIT}, 0) \to (a', 0) \to (a', 1) \to (a, 1) \to (b, 3)$$

In the formalism of Lemma 22 this means

$$\begin{array}{rclcrcl} t_0 & = & a & \quad & c_0 & = & 1 \\ t_1 & = & b & & c_1 & = & 3 \end{array}$$

The claim of Lemma 22 is illustrated by the choice

$$\begin{array}{rclcrcl} s_0 & = & a & & & & \\ s_1 & = & a \oplus f^2(b) & \quad & d_0 & = & 1 \\ s_1^1 & = & a & & c_1 & = & 1 \\ s_1^2 & = & f^2(b) & & & & \end{array}$$

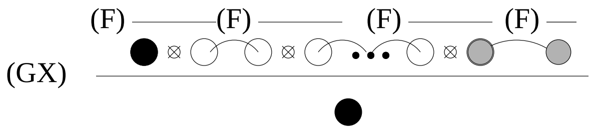
Lemma 23 *Let T be a set of binary terms. For all $t_0, \ldots, t_n \in Strip_f(atoms(T))$ and all natural numbers $c_0, \ldots, c_n$ we have that*

$$A_T \models (t_0, c_0) \to (t_1, c_1) \to \ldots \to (t_n, c_n)$$

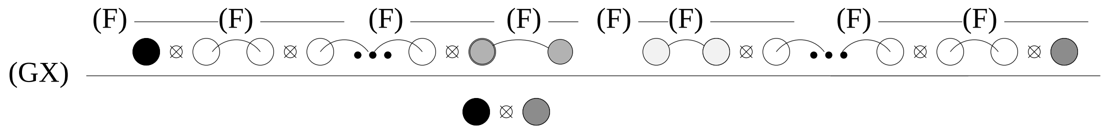
iff there are terms $s_1, \ldots, s_n \in T$ and natural numbers $d_1, \ldots, d_n$ such that:

$$
\text{(GX)}\,\dfrac{
\text{(F)}\,\dfrac{\dfrac{\text{(F)}\,\dfrac{T \vdash s_0}{\vdots}}{\ }}{T \vdash \underbrace{f^{d_0}(s_0) = f^{c_0}(t_0)}}
\qquad
\text{(F)}\,\dfrac{\dfrac{\text{(F)}\,\dfrac{T \vdash s_1 = s_1^1 \oplus s_1^2}{\vdots}}{\ }}{T \vdash \underbrace{f^{d_1}(s_1^1)} \oplus (f^{d_1}(s_1^2) = f^{c_1}(t_1))}
\qquad \cdots \qquad
\text{(F)}\,\dfrac{\dfrac{\text{(F)}\,\dfrac{T \vdash s_n = s_n^1 \oplus s_n^2}{\vdots}}{\ }}{T \vdash f^{d_n}(s_n^1) \oplus f^{d_n}(s_n^2)}
}{T \vdash f^{c_n}(t_n) = f^{d_n}(s_n^2)}
$$

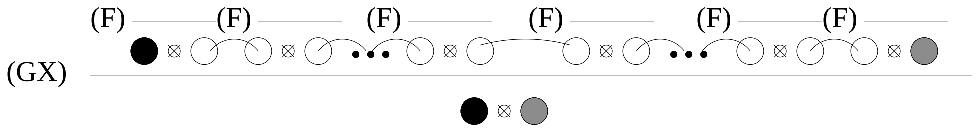
Figure 5.6: Illustration of Lemma 22



(a) A singleton term obtained from a singleton term and a chain of binary terms



(b) A sum of two singleton terms, each of which is obtained from a singleton term and a chain of binary terms



(c) A sum of two terms which are linked by a chain of binary terms

Figure 5.7: Illustration of Lemma 22

1. *for* $1 \leq i \leq n$ *the term* s_i *is headed with* $\oplus$ *and has exactly two atoms, that is* $s_i = s_i^1 \oplus s_i^2$
2. $\forall 1 \leq i \leq n \, : \, f^{d_i}(s_i^1) = f^{c_{i-1}}(t_{i-1})$
3. $\forall 1 \leq i \leq n \, : \, f^{d_i}(s_i^2) = f^{c_i}(t_i)$

As a consequence, we obtain that

$$\bigoplus_{i=1}^{n} f^{d_i}(s_i)\!\downarrow = f^{d_1}(s_1^1) \oplus f^{d_n}(s_n^2) = f^{c_0}(t_0) \oplus f^{c_n}(t_n)$$

We can now give the definition of h_{max}: Given T a set of terms and w a term, we define $h_{max} = n'^2 + n'^3 + g_{max}$ where:

$$\begin{array}{rcl} n' & = & |A_{S_T(T,w)}| * c_{max} \\ c_{max} & = & \mathit{maxdiff}(S_T(T,w)) \\ \mathit{maxdiff}(T) & = & \max\{|\#_f(u) - \#_f(v)| \mid u \oplus v \in T\} \\ g_{max} & = & \max\{\#_f(t) \mid t \in S_T(T,w)\} \end{array}$$

Lemma 24 *Let* P *be a minimal-*$\oplus$*-lazy proof of* $T \vdash w$. *All nodes of* P *are in* $S_f(T, w, h_{max})$.

Proof: Let P be a minimal-$\oplus$-*lazy* proof of $T \vdash w$. First note that, by Corollary 20, all nodes except those that are obtained by rule (F) and which yield by a sequence of successive applications of (F) a hypothesis of rule (GX), are labeled by terms in $S_T(T, w) \subseteq S_f(T, w, h_{max})$.

Let P' be a maximal subtree of the proof P such that its root is obtained by rule (GX) and all non-root nodes are obtained by rule (F). Let the root of P' be labeled with $T \vdash u$. Again by Corollary 20, both u and the labels at all the leaves of P' are in $S_T(T, w)$. There are two cases:

1. u is not headed with $\oplus$.

   By minimality of the proof P, there is no non-empty subset of the hypotheses to the rule (GX) at the root of P' for which their combination by $\oplus$ yields 0. Hence, the leaves of P' are marked with exactly one term s which is not headed with $\oplus$, and a set of terms $s_1, \ldots, s_n$ which are headed with $\oplus$ and have exactly two atoms. This is illustrated in Figure 5.7(a).

   In this case, there are terms $s_0, s_1, \ldots, s_n \in T$ and natural numbers $d_0, d_1, \ldots, d_n$ such that

   - s_0 is not headed with $\oplus$, and for $1 \leq i \leq n$ the term s_i is headed with $\oplus$ and has exactly two atoms, that is $s_i = s_i^1 \oplus s_i^2$
   - $f^{d_0}(s_0) = f^{d_1}(s_1^1)$
   - $\forall 1 \leq i \leq n-1$: $f^{d_i}(s_i^2) = f^{d_{i+1}}(s_{i+1}^1)$

   The d_i are the respective lengths of the chains of (F) rules. By Lemma 22, Corollary 25, and the minimality of the proof we obtain that all the d_i are smaller than or equal to h_{max}.

2. u is headed with $\oplus$, that is $u = u_1 \oplus u_2$.

   By minimality of the proof P, there is no non-empty subset of the hypotheses to the rule (GX) at the root of P' for which their combination by $\oplus$ yields 0. There are two possibilities:

   (a) there are terms $s_0, s_1, \ldots, s_n \in T$ and natural numbers $d_0, d_1, \ldots, d_n$ such that

   - s_0 is not headed with $\oplus$, and for $1 \leq i \leq n$ the term s_i is headed with $\oplus$ and has exactly two atoms, that is $s_i = s_i^1 \oplus s_i^2$
   - $f^{d_0}(s_0) = f^{d_1}(s_1^1)$
   - $\forall 1 \leq i \leq n-1$: $f^{d_i}(s_i^2) = f^{d_{i+1}}(s_{i+1}^1)$

   and there are terms $r_0, r_1, \ldots, r_m \in T$ and natural numbers $e_0, e_1, \ldots, e_m$ such that

   - r_0 is not headed with $\oplus$, and for $1 \leq i \leq m$ the term r_i is headed with $\oplus$ and has exactly two atoms, that is $r_i = r_i^1 \oplus r_i^2$
   - $f^{e_0}(r_0) = f^{e_1}(r_1^1)$
   - $\forall 1 \leq i \leq n-1$: $f^{e_i}(r_i^2) = f^{e_{i+1}}(r_{i+1}^1)$

   This is illustrated in Figure 5.7(b). In this case we conclude as in the first case.

   (b) there are terms $s_1, \ldots, s_n \in T$ and natural numbers $d_1, \ldots, d_n$ such that

   - for $1 \leq i \leq n$ the term s_i is headed with $\oplus$ and has exactly two atoms, that is $s_i = s_i^1 \oplus s_i^2$
   - $\forall 1 \leq i \leq n-1$: $f^{d_i}(s_i^2) = f^{d_{i+1}}(s_{i+1}^1)$

   This is illustrated in Figure 5.7(c).

   The d_i are the respective lengths of the chains of (F) rules. By Lemma 22, Corollary 25, and the minimality of the proof we obtain that all the d_i are smaller than or equal to h_{max}. □

Corollary 25 *Let A be a one-counter automaton and $\pi : (q, c_q) \to^\star (r, r_q)$ a path between the state q with the counter $c_q \geq 0$ and the state r with the counter $c_r \geq 0$. Let $n' = max(n, \lceil\sqrt{c_q}\rceil, \lceil\sqrt{c_r}\rceil)$. There exists a path $\pi' : (q, c_q) \to^\star (r, c_r)$ between the state q with the counter $c_q \geq 0$ and the state r with the counter $c_r \geq 0$ such that the counter is always smaller than $n'^3 + n'^2 + g_{max}$.*

The proof of this lemma is given in the appendix.

5.3.3 Locality in the General Case

In the general case we obtain a locality result for a notion of subterm which may yield an exponential blow-up.

Lemma 26 *Let $M \subseteq T(\Sigma)$, $t_0 \in T(\Sigma)$, and $t_1, \ldots, t_n \in S_T(M)$.*
If $(t_0 \oplus t_1 \oplus \ldots \oplus t_n){\downarrow} \in S_\oplus(M)$ then $t_0 \in S_\oplus(M)$.

Proof: By Property 6, $t_0 \in S_\oplus(M)$ if every atom of t_0 is an element of $S_T(M)$. This is obvious since every atom of t_0 either appears in the final sum, or is canceled out by an atom of one of the terms $t_1, \ldots, t_n$. □

Theorem 4 (Locality) *If there is a proof P of $T \vdash u$ then there is a $S_\oplus$-local proof of $T \vdash u$.*

Proof: If there is a proof of $T \vdash w$ then there also is a $\oplus$-*lazy* and minimal proof of $T \vdash w$. By Corollary 20, all nodes except those that are obtained by rule (F) and which lead to a hypothesis of rule (GX) by a sequence of successive applications of (F), are labeled by terms in $S_T(T, w) \subseteq S_\oplus(T, w)$.

In particular, all labels at the nodes obtained by (GX) are in $S_\oplus(T, w)$.

We now transform this proof into an $\oplus$-*eager* proof by pushing all applications of (GX) as far as possible to the leaves. Note that if a label in some node is in $S_\oplus(T, w)$ then its predecessor by rule (F) is also in $S_\oplus(T, w)$.

Hence we obtain that the labels of all nodes are in $S_\oplus(T, w)$ by applying inductively Lemma 26 on this $\oplus$-*eager* proof. □

The construction is illustrated in the following example:

Example 16 (Transformation of a $\oplus$-*lazy* Proof into $S_\oplus$-Local Proof) *Let T be the set of terms $T = \{a \oplus d, b \oplus c, f(b)\}$. The following proof of $T \vdash f^2(a) \oplus f^2(d) \oplus f(b) \oplus f(c)$ is $\oplus$-lazy:*

$$\text{(GX)}\ \dfrac{\text{(F)}\ \dfrac{\text{(F)}\ \dfrac{\text{(A)}\ \dfrac{a \oplus d \in T}{T \vdash a \oplus d}}{T \vdash f(a) \oplus f(d)}}{T \vdash f^2(a) \oplus f^2(d)} \qquad \text{(F)}\ \dfrac{\text{(A)}\ \dfrac{b \oplus c \in T}{T \vdash b \oplus c}}{T \vdash f(b) \oplus f(c)} \qquad \text{(A)}\ \dfrac{f(b) \in T}{T \vdash f(b)}}{T \vdash f^2(a) \oplus f^2(d) \oplus f(c)}$$

We transform it into a $S_\oplus$-local proof.

$$\text{(GX)}\ \dfrac{\text{(F)}\ \dfrac{\text{(GX)}\ \dfrac{\text{(F)}\ \dfrac{\text{(A)}\ \dfrac{a \oplus d \in T}{T \vdash a \oplus d}}{T \vdash f(a) \oplus f(d)} \qquad \text{(A)}\ \dfrac{b \oplus c \in T}{T \vdash b \oplus c}}{T \vdash f(a) \oplus f(d) \oplus b \oplus c}}{T \vdash f^2(a) \oplus f^2(d) \oplus f(b) \oplus f(c)} \qquad \text{(A)}\ \dfrac{f(b) \in T}{T \vdash f(b)}}{T \vdash f^2(a) \oplus f^2(d) \oplus f(c)}$$

5.4 Locality for the Equational Theory AGh

5.4.1 Some New Definitions

We have to adapt some definitions in the presence of the inverse operator I. First note that, in the definition of subterm we can now also decompose an I operator. The definition 23 of syntactic subterms now becomes:

Definition 30 (Syntactic Subterm S) *We define the set of syntactic subterms $S(t)$ of a term t as the smallest set $S(t)$ such that:*

- $t \in S(t)$
- $\langle u, v\rangle \in S(t)$ *then* $u, v \in S(t)$
- $\{u\}_k \in S(t)$ *then* $u, k \in S(t)$
- *if* $t \in S(t)$ *then* $atoms(t) \subseteq S(t)$
- $f(u) \in S(t)$ *then* $u \in S(t)$
- $I(u) \in S(t)$ *then* $u \in S(t)$

Definition 31 ($I(T)$) *Let T be a set of terms, $I(T)$ is the set of the normal forms of all terms of T where the operator I is applied once, that is*

$$I(T) := \{I(t)\downarrow \mid t \in T\}$$

Example 17 *If $T = \{I(a), b \oplus f(c), I(f(c)) \oplus a\}$ then $I(T) = \{a, I(b) \oplus I(f(c)), f(c) \oplus I(a)\}$.*

Definition 32 (S_{T_I}) $S_{T_I} = S_T(T \cup I(T)) = S_T(T) \cup S_T(I(T))$

Definition 33 ($SI_\oplus(T)$) $SI_\oplus(T) = S_\oplus(S_{T_I}(T))$

Definition 34 (I-$\oplus$-*lazy* Proof) *Let P a proof of $T \vdash w$, P is a I-$\oplus$-lazy proof if P is $\oplus$-lazy, there is no (GX) immediately above (I) in P and there is no (I) immediately above (F) in P.*

Definition 35 (I-Minimal-$\oplus$-*lazy* Proof) *P is a* I-minimal-$\oplus$-*lazy proof of $T \vdash w$ if only if P is a I-$\oplus$-lazy proof of $T \vdash w$ which is minimal among all I-$\oplus$-lazy proofs.*

Notice that there are never two successive applications of (I) in a minimal proof. We obtain again that

Lemma 27 *If there is a proof of $P \vdash w$ then there is also an I-$\oplus$-lazy proof of $P \vdash w$.*

Proof: To obtain a I-$\oplus$-*lazy* proof we apply the proof transformation rules given in Figures 5.2, 5.1, 5.8 and 5.9.

$$\dfrac{\dfrac{T \vdash x}{T \vdash I(x)}\,(\mathrm{I})}{T \vdash f(I(x))\downarrow = I(f(x))}\,(\mathrm{F}) \quad \Longleftrightarrow \quad \dfrac{\dfrac{T \vdash x}{T \vdash f(x)}\,(\mathrm{F})}{T \vdash I(f(x))}\,(\mathrm{I})$$

Figure 5.8: Equivalence between (I)-(F) and (F)-(I)

□

Obviously, if there is a proof of $T \vdash w$ then there is also an I-minimal-$\oplus$-*lazy* proof of $T \vdash w$.

$$
\begin{array}{ll}
\text{(GX)} & \dfrac{T \vdash x_1 \ldots T \vdash x_n}{T \vdash x_1 \oplus \ldots \oplus x_n} \\
\text{(I)} & \dfrac{}{T \vdash I(x_1) \oplus \ldots \oplus I(x_n)}
\end{array}
\quad \Longrightarrow \quad
\text{(GX)} \; \dfrac{\text{(I)} \; \dfrac{T \vdash x_1}{T \vdash I(x_1)} \quad \cdots \quad \text{(I)} \; \dfrac{T \vdash x_n}{T \vdash I(x_n)}}{T \vdash I(x_1) \oplus \ldots \oplus I(x_n)}
$$

Figure 5.9: Transformation of (GX)-(I) into (I)-(GX)

5.4.2 Some Lemmata and Properties in the AGh Case

The following lemma is similar to Lemma 17:

Lemma 28 *Let P be a I-minimal-$\oplus$-lazy proof of $T \vdash w$ and let P' be a subproof P with root label $T \vdash N$. If the last rule applied in P' is (GX) then $N \in S_{T_I}(w) \cup S_{T_I}(T)$.*

Proof: By induction on the size of the proof P. The base case is trivial. Let us assume that P is of the form

$$
P = \left\{ \dfrac{P_1 = \left\{ \dfrac{\vdots}{T \vdash R_1} \right. \qquad \ldots \qquad P_n = \left\{ \dfrac{\vdots}{T \vdash R_n} \right.}{T \vdash w} \right.
$$

Assume that all immediate subproofs P_i of P with root label $T \vdash R_i$ verify the property *i.e.* if there is the rule (GX) in P_i then the node resulting from this rule application and labeled by $T \vdash N_i$ satisfies that $N_i \in S_{T_I}(R_i) \cup S_{T_I}(T)$.

We proceed by case analysis on the last rule applied in the proof P:

1. (P), (C), (F), (I): In this case any application of rule (GX) occurs in a subproof P_i. For these rules $S_{T_I}(R_i) \subseteq S_{T_I}(w)$ for all i and the result follows from the induction hypothesis.
2. (UL), (UR) or (D): In this case any application of rule (GX) occurs in a subproof P_i. By Lemma 10 we have that $S_{T_I}(R_i) \subseteq S_{T_I}(T)$ for all i, we hence conclude by induction hypothesis.
3. (GX): Let P_N be a subproof of P with root label $T \vdash N$ which terminates with an application of the rule (GX).
   (a) If P_N is P itself then $N = w$ and the claim holds.
   (b) If P_N is a proper subproof of P then P_N is a subproof of one of the P_i's.
   By induction hypothesis we have that $N \in S_{T_I}(T) \cup S_{T_I}(R_i)$.
   If $N \in S_{T_I}(T)$ we are done, otherwise let P'_i be the subproof of P_i such that P'_i does not terminate on (F) or (I), and P_i is obtained from P'_i by a sequence of n applications of (F) or (I).
   Let $T \vdash R'_i$ be the label of the root of P'_i.
   It is enough to show that $R'_i \in S_{T_I}(T \cup \{w\})$ since the result then follows by induction hypothesis.
   By construction the last rule application of P'_i cannot be (F), and since the proof P is $\oplus$-*lazy* it also cannot be (GX) or (I).
   There are two possible cases: R'_i is headed with $\oplus$ or not. Note that R'_i is headed with $\oplus$ iff R_i is headed with $\oplus$.
      i. If R'_i is headed with $\oplus$: the last rule application in P'_i cannot be (P) or (C), and it cannot be (GX) or (F) or (I) as seen above.
      Hence, the last rule application of P'_i must be one of (A), (D), (UL) or (UR).
      By Lemma 10, we get $R'_i \in S_{T_I}(T)$.

ii. If R'_i is not headed with $\oplus$: Either $R_i \in S_{T_I}(w)$ and we are done, or $R_i \notin S_{T_I}(w)$ and R_i is canceled out by another term in the (GX) rule.
Since P is minimal, R_i must be canceled out by some term R_j which is headed with $\oplus$ and which is obtained by a subproof P_j: otherwise the same term would appear twice in the sum, and we could obtain a smaller proof just by omitting these two terms, which contradicts the minimality of P.
Therefore we can write $R_j = R_i \oplus A_i$.
Since P is a $\oplus$-*lazy* proof the last rule of P'_i cannot be (GX) nor (F) nor (I) by definition.
If it was one of (UL), (UR), (D) or (A) then we are done by applying as before Lemma 10.
If it is (P) or (C), we observe that R'_i is not headed by f and that $\#_f(R_i) = n$.
We now go up from $R_j = R_i \oplus A_i$ as long as possible along a sequence of (F) or (I) rules: let P'_j be the subproof of P_j such that P'_j does not terminate on (F) or (I), and such that P_j is obtained from P'_j by a sequence of m applications of (F) or (I).
Since $R_i \in \text{atoms}(R_j)$ we get $m \leq \#_f(R_j) \leq \#_f(R_i) = n$.
Let $R''_i \oplus A''_i$ be the label of the root of P'_j. Since the proof is I-$\oplus$-*lazy* proof and by construction, the last rule of P'_j cannot be (GX) or (F) or (I).
It also cannot be (P) nor (C) because $R''_i \oplus A''_i$ is headed with $\oplus$.
Hence, the last rule of P'_j must be one of (A), (UR), (UL), (D). By the Lemma 10 we have that $S_{T_I}(R''_i \oplus A_i) \subseteq S_{T_I}(T)$.
It follows that $R'_i \in S_{T_I}(R''_i \oplus A''_i) \subseteq S_{T_I}(T)$ because $m \leq n$, and we conclude by applying the induction hypothesis. □

The next lemma is similar to Lemma 18:

Lemma 29 *Let P be a I-minimal-$\oplus$-lazy proof of $T \vdash w$. For every subproof P' of P with root label $T \vdash N$, and where the last rule application is (P) or (C), we have that $N \in S_{T_I}(w) \cup S_{T_I}(T)$.*

Proof: By induction on the size of the proof P. The base case is trivial. Let us assume that P is of the form

$$P = \left\{ \begin{array}{c} P_1 = \left\{ \begin{array}{c} \vdots \\ \hline T \vdash R_1 \end{array} \right. \quad \ldots \quad P_n = \left\{ \begin{array}{c} \vdots \\ \hline T \vdash R_n \end{array} \right. \\ \hline T \vdash w \end{array} \right.$$

Assume that all immediate subproofs P_i of P with root label $T \vdash R_i$ verify the property *i.e.* if there is the rule (P) or (C) in P_i then the node resulting from this rule application and labeled by $T \vdash N_i$ satisfies that $N_i \in S_{T_I}(R_i) \cup S_{T_I}(T)$.

We proceed by case distinction on the last rule applied in the proof P:

1. (P), (C): Consider a subproof P_N of P which terminates with an application of the rule (P) or (C) and whose root is labeled by $T \vdash N$.
   (a) If P_N is P itself then $N = w$ and the claim obviously holds.
   (b) If P_N is a proper subproof of P then P_N must be a subproof of one of the P_i, hence by induction hypothesis we have that $N \in S_{T_I}(T) \cup S_{T_I}(R_i) \subseteq ST(T) \cup S_{T_I}(w)$.
2. (F): In this case any application of rule (P) or (C) must be in one of the the subproofs P_i. We conclude immediately by induction hypothesis and by the fact that for this rule $S_{T_I}(R_i) \subseteq S_{T_I}(u)$ for all i.
3. (UL) (UR) or (D): In this case any application of rule (P) or (C) must be in one of the subproofs P_i. By Lemma 10 we have that $S_{T_I}(R_i) \subseteq S_{T_I}(T)$ for all i, we hence conclude by induction hypothesis.

4. (GX): In this case any application of rule (P) or (C) must be in one of the the subproofs P_i, hence by induction hypothesis we have that $N \in S_{T_I}(T) \cup S_{T_I}(R_i)$. If $N \in S_{T_I}(T)$ we are done, otherwise let P_i' be the subproof of P_i such that P_i' does not terminate on (F) or (I), and P_i is obtained from P_i' by a sequence of n applications of (F) or (I). Let the root of P_i' be labeled by $T \vdash R_i'$. By induction, it is enough to show that $R_i' \in S_{T_I}(T \cup \{w\})$.

   By construction the last rule application of P_i' cannot be (F), and since the proof P is $\oplus$-*lazy* it also cannot be (GX) or (I). There are two possible cases: R_i' is headed with $\oplus$ or not. Note that R_i' is headed with $\oplus$ iff R_i is headed with $\oplus$.

   (a) If R_i' is headed with $\oplus$: the last rule application in P_i' cannot be (P) or (C), and it cannot be (GX), (F) or (I) as seen above. Hence, the last rule application of P_i' must be one of (A), (D), (UL) or (UR). As a consequence, $R_i' \in S_{T_I}(T)$ by Lemma 10.

   (b) If R_i' is not headed with $\oplus$: Either $R_i \in S_{T_I}(w)$ and we are done, or $R_i \notin S_{T_I}(w)$ and R_i is canceled out by another term in the (GX) rule. By minimality of the proof P, R_i must be canceled out by some term R_j, which is obtained by a subproof P_j, and which is headed with $\oplus$ (otherwise we would have twice the same term in the sum, and we could obtain a smaller proof just by omitting these two terms, which is contradiction to the minimality of the proof). We can hence write $R_j = R_i \oplus A_i$.

   Since we have a $\oplus$-*lazy* proof and by construction, the last rule of P_i' cannot be (GX), (F) or (I) and if it was one of (UL), (UR), (D) or (A) then we are done by applying as before Lemma 10. It remains the case that the last rule of P_i' is one of the rules (P) or (C). In this case we observe that R_i' is not headed by f, and that by consequence $\#f(R_i) = n$.

   We now mount up from $R_j = R_i \oplus A_i$ as long as possible over a sequence of (F) or (I) rules : let P_j' be the subproof of P_j such that P_j' does not terminate on (F) or (I), and such that P_j is obtained from P_j' by a sequence of m applications of (F) or (I). Obviously, $m \leq \#f(R_j) \leq \#f(R_i) = n$, since $R_i \in \text{atoms}(R_j)$. Let the root of P_j' be labeled by $R_i'' \oplus A_i''$.

   Since we have a $\oplus$-*lazy* proof and by construction, the last rule of P_j' cannot be (GX), (F) or (I). It also cannot be (P) or (C) because $R_i'' \oplus A_i''$ is headed with $\oplus$. Finally, if it is one of (A), (UR), (UL), (D) then by the Lemma 10 we have that $S_{T_I}(R_i'' \oplus A_i) \subseteq S_{T_I}(T)$. In this case, $R_i' \in S_{T_I}(R_i'' \oplus A_i'') \subseteq S_{T_I}(T)$ because $m \leq n$, and we conclude by applying the induction hypothesis. □

The next lemma is similar to Lemma 19.

Lemma 30 *Let P be an I-minimal-$\oplus$-lazy proof of $T \vdash w$ and consider a subproof P' with root label $T \vdash N$, such that the last rule of P' is (F) or (I). If either*

1. *all nodes from the root of P' to the root of P are (F) or (I),*

2. *or if the first successor not labeled by (F) or (I) of the root of P' in P is labeled by a rule different from (GX),*

then $N \in S_{T_I}(T \cup \{w\})$.

This lemma states that (F) or (I) nodes are harmless as long as they do not yield via a sequence of successive (F) or (I) nodes to a hypothesis of a (GX) rule.

Proof: If all nodes from the root of P' to the root of P are (F) or (I), then obviously by induction $N \in S_{T_I}(w)$. Otherwise, if the first successor of the root of P' in P which is not labeled by (F) is labeled by a rule (Z) different from (GX) we consider the different possibilities for (Z):

- (Z) cannot be (A)

- (Z) cannot be (UL), (UR), (D) since the hypothesis to (Z) is headed by f.

- (Z) can only be (P) or (C), and in this case we conclude by Lemma 29. □

The following corollary summarizes the lemmata of this subsection analogously to Corollary 20:

Corollary 31 *Let P be a I-minimal-$\oplus$-lazy proof of $T \vdash w$. All nodes of P are labeled by terms in $S_{T_I}(T, w)$ except the nodes which are constructed by (F) or (I) and which yield by a sequence of (F) or (I) steps to a hypothesis of a (GX) rule.*

5.4.3 Locality in the Binary Case

In the *binary* case, that is when all terms in $S_T(T, w)$ have at most two atoms, we can in fact find an upper bound for the number of applications of (F). The key to this upper bound is a technical lemma about counter automata which is proven in the appendix. The construction is similar to the one used in the binary case for the equational theory $ACUNh$.

Definition 36 *Let T be a set of terms such that every term in T has at most two atoms. The* automaton associated with T*, abbreviated A_T, is a one-counter automaton without input defined as follows:*

We partition $T = T_1 \uplus T_2$ where T_1 is the set of terms not headed with $\oplus$, and T_2 is the set of terms headed with $\oplus$.

- *The set of states Q_T of A_T is defined as*

$$\begin{aligned} Q_T &= P_T \cup R_T \cup \{\textsc{init}\} \\ P_T &= \{p', I(p') \mid p \in \mathit{Strip}_f(T_1)\} \\ R_T &= \{r, I(r') \mid r \in \mathit{Strip}_f(T_1) \cup \mathit{Strip}_f(\mathit{atoms}(T_2))\} \end{aligned}$$

- INIT *is the initial state of A_T.*
- *The set of transitions is:*

	From	*To*	*Condition*	*Action*
$\forall t \in T_1$:	INIT	$(\mathit{Strip}_f(t))'$	$c \geq 0$	$c := c$
$\forall t \in T_1$:	$(\mathit{Strip}_f(t))'$	$(\mathit{Strip}_f(t))'$	$c \geq 0$	$c := c + 1$
$\forall t \in T_1$:	$(\mathit{Strip}_f(t))'$	$(I(\mathit{Strip}_f(t)))'$	$c \geq 0$	$c := c$
$\forall t \in T_1$:	$(\mathit{Strip}_f(t))'$	$\mathit{Strip}_f(t)$	$c \geq \#_f(t)$	$c := c$
$\forall t \in T_1$:	$(I(\mathit{Strip}_f(t)))'$	$I(\mathit{Strip}_f(t))$	$c \geq \#_f(t)$	$c := c$
$\forall t \oplus s \in T_2$:	$I(\mathit{Strip}_f(t))$	$\mathit{Strip}_f(s)$	$c \geq \#_f(t)$	$c := c - \#_f(t) + \#_f(s)$

In the definition of the automaton, the primed states in P_T are only intermediate states which do not correspond to a proof; only the states in R_T correspond to proofs in a sense to be made precise below. Note that in the last line of the above transition table the statement "$t \oplus s \in T_2$" is to be understood modulo AC, such that we obtain from a binary clause a back and a forth transition, as already explained in the informal definition of the automaton given in Section 5.3.2.

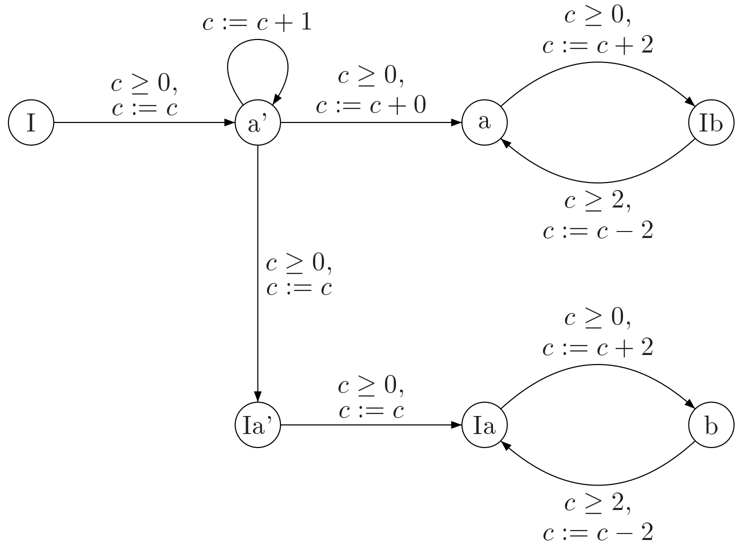
Example 18 *The automaton A_T for $T = \{a \oplus f^2(b), a\}$ is as follows, where I denotes the initial state:*

Lemma 32 *Let T be a set of binary terms. For all terms $t_0, t_1, \ldots, t_n \in \mathit{Strip}_f(\mathit{atoms}(T)) \cup I(\mathit{Strip}_f(\mathit{atoms}(T)))$ and all natural numbers $c_0, c_1, \ldots, c_n$ we have that*

$$A_T \models (\textsc{init}, 0) \to (t'_0, 0) \to \ldots \to (t'_0, c_0) \to (t_0, c_0) \to (t_1, c_1) \to \ldots \to (t_n, c_n)$$

iff there are terms $s_0, s_1, \ldots, s_n \in T \cup I(T)$ and natural numbers $d_0, d_1, \ldots, d_n$ such that:

1. *s_0 is not headed with $\oplus$, and for $1 \leq i \leq n$ the term s_i is headed with $\oplus$ and has exactly two atoms, that is $s_i = s_i^1 \oplus s_i^2$*



2. $f^{d_0}(s_0) = f^{c_0}(t_0)$
3. $\forall 1 \leq i \leq n$: $f^{d_i}(s_i^1) = I(f^{c_{i-1}}(t_{i-1}))$
4. $\forall 1 \leq i \leq n$: $f^{d_i}(s_i^2) = f^{c_i}(t_i)$

As a consequence, we obtain that

$$\bigoplus_{i=0}^{n} f^{d_i}(s_i)\!\downarrow = f^{d_n}(s_n^2) = f^{c_n}(t_n)$$

Lemma 33 *Let T be a set of binary terms. For all terms $t_0, \ldots, t_n \in Strip_f(atoms(T)) \cup I(Strip_f(atoms(T)))$ and all natural numbers $c_0, \ldots, c_n$ we have that*

$$A_T \models (t_1, c_1) \to \ldots \to (t_n, c_n)$$

iff there are terms $s_1, \ldots, s_n \in T \cup I(T)$ and natural numbers $d_0, d_1, \ldots, d_n$ such that:

1. *for $1 \leq i \leq n$ the term s_i is headed with $\oplus$ and has exactly two atoms, that is $s_i = s_i^1 \oplus s_i^2$*
2. $\forall 1 \leq i \leq n$: $f^{d_i}(s_i^1) = I(f^{c_{i-1}}(t_{i-1}))$
3. $\forall 1 \leq i \leq n$: $f^{d_i}(s_i^2) = f^{c_i}(t_i)$

As a consequence, we obtain that

$$\bigoplus_{i=1}^{n} f^{d_i}(s_i)\!\downarrow = f^{d_1}(s_1^1) \oplus f^{d_n}(s_n^2) = f^{c_0}(t_0) \oplus f^{c_n}(t_n)$$

We can now give the definition of h_{max}: Given T and w, we define $h_{max} = n'^2 + n'^3 + g_{max}$ where:

$$\begin{array}{rcl} n' & = & |A_{S_T(T,w)}| * c_{max} \\ c_{max} & = & maxdiff(S_T(T,w)) \\ maxdiff(T) & = & \max\{|\#_f(u) - \#_f(v)| \mid u \oplus v \in T\} \\ g_{max} & = & \max\{\#_f(t) \mid t \in S_T(T,w)\} \end{array}$$

Lemma 34 *Let be P an I-minimal-$\oplus$-lazy proof of $T \vdash w$. All nodes of P are in $S_f(T, w, h_{max})$.*

Proof: Let be P an I-minimal-$\oplus$-*lazy* proof of $T \vdash w$. First note that, by Corollary 31, all nodes except those that are obtained by rule (F) and which yield by a sequence of successive applications of (F) or (I) a hypothesis of rule (GX), are labeled by terms in $S_T(T, w) \subseteq S_f(T, w, h_{max})$.

Let P' be a maximal subtree of the proof P such that its root is obtained by rule (GX) and all non-root nodes are obtained by rule (F) or (I). Let the root of P' be labeled with $T \vdash u$. Again by Corollary 20, both u and the labels at all the leaves of P' are in $S_T(T, w)$. There are two cases:

1. u is not headed with $\oplus$.

   By minimality of the proof P, there is no non-empty subset of the hypotheses to the rule (GX) at the root of P' for which their combination by $\oplus$ yields 0. Hence, the leaves of P' are marked with exactly one term s which is not headed with $\oplus$, and a set of terms $s_1, \dots, s_n$ which are headed with $\oplus$ and have exactly two atoms.

   In this case, there are terms $s_0, s_1, \dots, s_n \in T$ and natural numbers $d_0, d_1, \dots, d_n$ such that

   - s_0 is not headed with $\oplus$, and for $1 \leq i \leq n$ the term s_i is headed with $\oplus$ and has exactly two atoms, that is $s_i = s_i^1 \oplus s_i^2$
   - $f^{d_0}(s_0) = f^{d_1}(s_1^1)$
   - $\forall 1 \leq i \leq n-1$: $f^{d_i}(s_i^2) = f^{d_{i+1}}(s_{i+1}^1)$

   The d_i are the respective lengths of the chains of (F) rules. By Lemma 32, Corollary 25, and the minimality of the proof we obtain that all the d_i are smaller than or equal to h_{max}.

2. u is headed with $\oplus$, that is $u = u_1 \oplus u_2$.

   By minimality of the proof P, there is no non-empty subset of the hypotheses to the rule (GX) at the root of P' for which their combination by $\oplus$ yields 0. There are two possibilities:

   (a) there are terms $s_0, s_1, \dots, s_n \in T \cup I(T)$ and natural numbers $d_0, d_1, \dots, d_n$ such that
   - s_0 is not headed with $\oplus$, and for $1 \leq i \leq n$ the term s_i is headed with $\oplus$ and has exactly two atoms, that is $s_i = s_i^1 \oplus s_i^2$
   - $f^{d_0}(s_0) = f^{d_1}(s_1^1)$
   - $\forall 1 \leq i \leq n-1$: $f^{d_i}(s_i^2) = f^{d_{i+1}}(s_{i+1}^1)$

   and there are terms $r_0, r_1, \dots, r_m \in T$ and natural numbers $e_0, e_1, \dots, e_m$ such that
   - r_0 is not headed with $\oplus$, and for $1 \leq i \leq m$ the term r_i is headed with $\oplus$ and has exactly two atoms, that is $r_i = r_i^1 \oplus r_i^2$
   - $f^{e_0}(r_0) = f^{e_1}(r_1^1)$
   - $\forall 1 \leq i \leq n-1$: $f^{e_i}(r_i^2) = f^{e_{i+1}}(r_{i+1}^1)$

   In this case we conclude as in the first case.

   (b) there are terms $s_1, \dots, s_n \in T \cup I(T)$ and natural numbers $d_1, \dots, d_n$ such that
   - for $1 \leq i \leq n$ the term s_i is headed with $\oplus$ and has exactly two atoms, that is $s_i = s_i^1 \oplus s_i^2$
   - $\forall 1 \leq i \leq n-1$: $f^{d_i}(s_i^2) = f^{d_{i+1}}(s_{i+1}^1)$

   The d_i are the respective lengths of the chains of (F) rules. By Lemma 32, Corollary 25, and the minimality of the proof we obtain that all the d_i are smaller than or equal to h_{max}. $\square$

5.4.4 Locality in the General Case

In the general case we obtain a locality result for a notation of subterm which is of exponential size.

Lemma 35 *Let $M \subseteq T(\Sigma)$, $t_0 \in T(\Sigma)$, and $t_1, \dots, t_n \in S_{T_I}(M)$.*
If $(t_0 \oplus t_1 \oplus \dots \oplus t_n)\!\downarrow \in SI_\oplus(M)$ then $t_0 \in SI_\oplus(M)$.

Proof: In order to show that $t_0 \in S_\oplus(M)$ it is by Property 6 sufficient to show that every atom of t_0 is an element of $S_{T_I}(M)$, which is obvious since every atom of t_0 either appears in the final sum, or is canceled out by an atom of one of the terms $t_1, \ldots, t_n$. □

Theorem 5 (Locality) *If there is a proof P of $T \vdash u$ then there is a $SI_\oplus$-local proof of $T \vdash u$.*

Proof: If there is a proof of $T \vdash w$ then there also is an I-minimal-$\oplus$-*lazy* proof of $T \vdash w$. Since the proof has a minimal number of nodes among all I-$\oplus$-*lazy* proofs of $T \vdash w$ there cannot be two (I)-steps which are connected by a sequence of (F)-steps (otherwise we could obtain a smaller and still I-$\oplus$-*lazy* proof by omitting the two (I)-steps). We can now commute the (I) steps with the (F) steps according to Figure 5.8 and push the (I) steps as far as possible towards the leaves. The resulting proof is still I-minimal-$\oplus$-*lazy*.

By Corollary 31, all nodes except those that are obtained by rule (F) and which yield by a sequence of successive applications of (F) a hypothesis of rule (GX), are labeled by terms in $S_T(T, w) \subseteq S_\oplus(T, w)$. In particular all labels at the nodes obtained by (GX) are in $SI_\oplus(T, w)$. We now transform this proof into an $\oplus$-*eager* proof by pushing all application of (GX) as far as possible to the leaves. Note that if a label in some node is in $SI_\oplus(T, w)$ then its predecessor by rule (F) is also in $SI_\oplus(T, w)$. Hence we obtain on this $\oplus$-*eager* proof, by applying inductively Lemma 35, that the labels of all nodes are in $SI_\oplus(T, w)$. □

Chapter 6

Locality in Case of a Homomorphic Encryption Operation

In this section we generalize the problem and replace the homomorphism symbol f by the encryption operation $\{\ \}_k$. More precisely, we replace the law of homomorphism for one function symbol f by the new equational axiom

$$\{x \oplus y\}_z = \{x\}_z \oplus \{y\}_z$$

In some sense we now get an infinite family of homomorphism functions, one for each possible key. Note, however, that the encryption key is a first-class value, and that now the homomorphic operation can also be decomposed by virtue of the (D) rule.

We have to push up the application of the rule (GX) over the rule (D). This transformation is possible as it is shown by Figure 6.2

The Lemmata 10 and 9 are easily adapted to the case where the homomorphism symbol f is $\{\ \}_k$.

6.1 Locality in the ACh Case

Lemma 12 and Lemma 11 still hold in this case.

Lemma 36 *Let P be a $\oplus$-lazy simple proof of $T \vdash w$ and let P' be a subproof P' of P with root label $T \vdash N$. If the last rule applied in P' is (GX) then $N \in S_T(w) \cup S_T(T)$.*

$$\text{(D)}\ \dfrac{\text{(I)}\ \dfrac{\text{(X)}\ \dfrac{\vdots}{T \vdash \{x\}_k}}{T \vdash I(\{x\}_k)} \quad \text{(A)}\ \dfrac{k \in T}{T \vdash k}}{T \vdash I(x)} \quad \Longleftrightarrow \quad \text{(I)}\ \dfrac{\text{(D)}\ \dfrac{\text{(X)}\ \dfrac{\vdots}{T \vdash \{x\}_k} \quad \text{(A)}\ \dfrac{k \in T}{T \vdash k}}{T \vdash x}}{T \vdash I(x)}$$

Figure 6.1: Commutation of (D) and (I)

$$
\text{(GX)}\ \dfrac{\text{(D)}\ \dfrac{T \vdash \{x_1\}_k \qquad T \vdash k}{T \vdash x_1} \quad \cdots \quad \text{(D)}\ \dfrac{T \vdash \{x_n\}_k \qquad T \vdash k}{T \vdash x_n}}{T \vdash x_1 \oplus \ldots \oplus x_n}
$$

$$\Downarrow$$

$$
\text{(D)}\ \dfrac{\text{(GX)}\ \dfrac{T \vdash \{x_1\}_k \ldots T \vdash \{x_n\}_k}{T \vdash \{x_1\}_k \oplus \ldots \oplus \{x_n\}_k} \qquad T \vdash k}{T \vdash x_1 \oplus \ldots \oplus x_n}
$$

Figure 6.2: Transformation of (D)-(GX) into (GX)-(D) in case that all hypotheses of (GX) are obtained by (D) with the same key

Proof: By induction on the size of the proof P. The base case is trivial. Let us assume that P is of the form

$$
P = \left\{ \dfrac{P_1 = \left\{ \dfrac{\vdots}{T \vdash R_1} \right. \qquad \cdots \qquad P_n = \left\{ \dfrac{\vdots}{T \vdash R_n} \right.}{T \vdash w} \right.
$$

Assume that all immediate subproofs P_i of P with root label $T \vdash R_i$ satisfy the property *i.e.* if there is the rule (GX) in P_i then the node resulting from this rule application and labeled by $T \vdash N_i$ satisfies that $N_i \in S_T(R_i) \cup S_T(T)$.

We proceed by case analysis on the last rule applied in the proof P:

1. (P), (C): In this case any application of rule (GX) must be in one of the the subproofs P_i. We conclude immediately by induction hypothesis and by the fact that for these rules $S_T(R_i) \subseteq S_T(w)$ for all i.

2. (UL), (UR) or (D): In this case any application of rule (GX) must be in one of the subproofs P_i. By Lemma 10 we have that $S_T(R_i) \subseteq S_T(T)$ for all i, we hence conclude by induction hypothesis.

3. (GX): Consider a subproof P_N of P which terminates with an application of the rule (GX) and whose root is labeled by $T \vdash N$.

   (a) If P_N is P itself then $N = w$ and the claim obviously holds.

   (b) If P_N is a proper subproof of P then P_N must be a subproof of one of the P_i, hence by induction hypothesis we have that $N \in S_T(T) \cup S_T(R_i)$.
   If $N \in S_T(T)$ we are done, otherwise let P_i' be the subproof of P_i such that P_i' does not terminate on (C), and P_i is obtained from P_i' by a sequence of applications of (C). Let $T \vdash R_i'$ be the label of the root of P_i'. It is enough to show that $R_i' \in S_T(T \cup \{w\})$ since the result then follows by induction hypothesis.
   By construction the last rule application of P_i' cannot be (C), and since the proof P is $\oplus$-*lazy* it also cannot be (GX).
   There are two possible cases: R_i' is headed with $\oplus$ or not.

      i. If R_i is headed with $\oplus$: the last rule application in P_i' cannot be (P), and it cannot be (GX) or (C) as seen above. Hence, the last rule application of P_i' must be one of (A), (D), (UL) or (UR). By Lemma 10 we get $R_i' \in S_T(T)$.

ii. If R'_i is not headed with $\oplus$ then R_i is not headed by $\oplus$ either, and we conclude by Lemma 12 that $N \in S_T(T \cup \{w\})$. □

The proof of the Lemma 14 for the case (P) and the proof of the Lemma 15 can be adapted to the general setting of this chapter following the same way than for the Lemma 36.

Hence, we obtain the locality result.

6.2 Locality in the ACUNh Case

6.2.1 Properties of $\oplus$-*lazy* Proofs in the ACUNh Case

Corollary 37 *Let P be a $\oplus$-lazy proof of $T \vdash w$. All the keys used in the proof P are in $S_T(T, W)$.*

Proof: It is an immediate consequence of Corollary 20 because in a $\oplus$-*lazy* proof all nodes are in $S_T(T, w)$ except those which are produced by the rule (F) and ended on (GX), so all the keys used in the proof P are in $S_T(T, W)$. □

The consequence of this lemma is that in a minimal and $\oplus$-*lazy* proof, only a finite number of different encryption keys is used. We can hence generalize the automata technique of Section 5.3.2 from a one-counter automaton to a pushdown automaton, where now every encryption key corresponds to a stack symbol. As a consequence of the above corollary we only get a finite stack alphabet.

Using the same technique than in the previous section, the Lemmata 17 and 18 in the case of (P) can be adapted to the more general setting of this chapter.. Lemma 19 can be easily adapted in the ACUNh case. We deduce again Corollary 20.

6.2.2 Locality in the Binary Case

In the *binary* case, that is when all terms in $S_T(T, w)$ have at most two atoms, we can in fact find an upper bound for the number of applications of (C). The key to this upper bound is a technical lemma about push down automata which is proven in the appendix.

In the binary case we associate to the set $S_T(T, w)$ a push down automaton. The states of the automaton are atoms of $S_T(T, w)$ without top-level f-symbols, and the stack represents the number of applications of C to a term. We first give an informal explanation of the construction. The formal definition of the automaton is the same that in Definition 29 replacing one-counter automaton by push down automata. We obtain all the lemmata, in particular:

Lemma 38 *Let be P a minimal-$\oplus$-lazy proof of $T \vdash w$. All nodes of P are in $S_f(T, w, h_{max})$.*

6.2.3 Locality in the General Case

In the general case we obtain a locality result for a notation of subterm which is of exponential size.

We have to define a new notion of a $\oplus$-*eager* proof, because now there are many homomorphic function symbols.

Definition 37 (C-$\oplus$-*eager* Proof) *Let P be a proof of $T \vdash w$, P is a C-$\oplus$-eager proof if P is flat and if for every k there is at most one (F) with encryption key k immediately above (GX) in P.*

Example 19 ($\oplus$-*lazy* Proof Transforms into a C-$\oplus$-*eager* Proof) *Here is a $\oplus$-lazy proof of $T \vdash w \oplus \{v\}_k$ where $T = \{u, v, w \oplus \{u\}_k, k\}$.*

$$\text{(GX)}\ \dfrac{\text{(C)}\ \dfrac{\text{(A)}\ \dfrac{u \in T}{T \vdash u} \quad \text{(A)}\ \dfrac{k \in T}{T \vdash k}}{T \vdash \{u\}_k} \qquad \text{(C)}\ \dfrac{\text{(A)}\ \dfrac{v \in T}{T \vdash u} \quad \text{(A)}\ \dfrac{k \in T}{T \vdash k}}{T \vdash \{v\}_k} \qquad \text{(A)}\ \dfrac{w \oplus \{u\}_k \in T}{T \vdash w \oplus \{u\}_k}}{T \vdash w \oplus \{v\}_k}$$

We transform it into a C-$\oplus$-eager proof

$$\text{(GX)}\ \dfrac{\text{(C)}\ \dfrac{\text{(GX)}\ \dfrac{\text{(A)}\ \dfrac{u \in T}{T \vdash u} \quad \text{(A)}\ \dfrac{v \in T}{T \vdash v}}{T \vdash u \oplus v} \quad \text{(A)}\ \dfrac{k \in T}{T \vdash k}}{T \vdash \{u \oplus v\}_k = \{u\}_k \oplus \{v\}_k} \qquad \text{(A)}\ \dfrac{w \oplus \{u\}_k \in T}{T \vdash w \oplus \{u\}_k}}{T \vdash w \oplus \{v\}_k}$$

We have to generalize Lemma 26 in order to cope with an infinite number of homomorphic functions. This yields

Lemma 39 *Let $M \subseteq T(\Sigma)$, $t_{0_1}, \ldots t_{0_k} \in T(\Sigma)$, and $t_1, \ldots, t_n \in S_T(M)$ and for all i, j if $i \neq j$ then $t_{0_i} \neq t_{0_j}$. If $(t_{0_1} \oplus \ldots \oplus t_{0_k} \oplus t_1 \oplus \ldots \oplus t_n){\downarrow} \in S_\oplus(M)$ then $\forall i \in \{1, 2, \ldots, k\}, t_{0_i} \in S_\oplus(M)$.*

Proof: In order to show that $\forall i \in \{1, 2, \ldots, k\}, t_{0_i} \in S_\oplus(M)$ it is by Property 6 sufficient to show that every atom of t_{0_i} is an element of $S_T(M)$, which is obvious since every atom of t_{0_i} either appears in the final sum, or is canceled out by an atom of one of the terms $t_1, \ldots, t_n$. □

Theorem 6 (Locality) *If there is a proof P of $T \vdash u$ then there is a $S_\oplus$-local proof of $T \vdash u$.*

Proof: If there is a proof of $T \vdash w$ then there also is a minimal-$\oplus$-*lazy* proof of $T \vdash w$. By Corollary 20, all nodes except those that are obtained by rule (C) and which yield by a sequence of successive applications of (C) a hypothesis of rule (GX), are labeled by terms in $S_T(T, w) \subseteq S_\oplus(T, w)$. In particular, all labels at the nodes obtained by (GX) are in $S_\oplus(T, w)$. We now transform this proof into an C-$\oplus$-*eager* proof by pushing all application of (GX) as far as possible to the leaves. Note that if a label in some node is in $S_\oplus(T, w)$ then its predecessor by rule (C) is also in $S_\oplus(T, w)$. Hence we obtain on this C-$\oplus$-*eager* proof (there is at most one application of (C) with the same key, hence we get that for all i, j if $i \neq j$ then $t_{0_i} \neq t_{0_j}$), by applying inductively Lemma 39, that the labels of all nodes are in $S_\oplus(T, w)$. □

6.3 Locality in the Case of Abelian Groups

6.3.1 Some New Definitions

The rule (I) and (C) commute, because (F) and (I) commute. However (I) and (D) commute like it is shown in Figure 6.1.

We use all definitions of the previous section.

6.3.2 Some Lemmata and Properties in the AGh Case

Lemmata 28, 29, 19 still hold, and we obtain an analogon of Corollary 31.

6.3.3 Locality in the Binary Case

The same construction is still available considering now a push down automaton. Hence, we obtain the same result as in the AGh case (lemmata about push down automata are given in Appendix).

6.3.4 Locality in the General Case

In the general case we obtain a locality result for a notation of subterm which is of exponential size.

Lemma 40 *Let $M \subseteq T(\Sigma)$, $t_{0_1}, \ldots t_{0_k} \in T(\Sigma)$, and $t_1, \ldots, t_n \in S_{T_I}(M)$ and for all i, j if $i \neq j$ then $t_{0_i} \neq t_{0_j}$. If $(t_{0_1} \oplus \ldots \oplus t_{0_k} \oplus t_1 \oplus \ldots \oplus t_n)\downarrow \in SI_{\oplus}(M)$ then $\forall i \in \{1, 2, \ldots, k\}, t_{0_i} \in SI_{\oplus}(M)$.*

Proof: In order to show that $\forall i \in \{1, 2, \ldots, k\}, t_{0_i} \in S_{\oplus}(M)$ it is by Property 6 sufficient to show that every atom of t_{0_i} is an element of $S_{T_I}(M)$, which is obvious since every atom of t_{0_i} either appears in the final sum, or is canceled out by an atom of one of the terms $t_1, \ldots, t_n$. $\square$

Theorem 7 (Locality) *If there is a proof P of $T \vdash u$ then there is a $SI_{\oplus}$-local proof of $T \vdash u$.*

Proof: If there is a proof of $T \vdash w$ then there also is a I-minimal-$\oplus$-*lazy* proof of $T \vdash w$. Since the proof has a minimal number of nodes among all I-$\oplus$-*lazy* proofs of $T \vdash w$ there cannot be two (I)-steps which are connected by a sequence of (C)-steps (otherwise we could obtain a smaller and still I-$\oplus$-*lazy* proof by omitting the two (I)-steps). We can now commute the (I) steps with the (C) steps like in Figure 5.8 and push the (I) steps as far as towards the leaves. The resulting proof is still I-minimal-$\oplus$-*lazy*.

By Corollary 20, all nodes except those that are obtained by rule (C) and which yield by a sequence of successive applications of (C) a hypothesis of rule (GX), are labeled by terms in $S_T(T, w) \subseteq S_{\oplus}(T, w)$. In particular all labels at the nodes obtained by (GX) are in $SI_{\oplus}(T, w)$. We now transform this proof into an $\oplus$-*eager* proof by pushing all application of (GX) as far as possible to the leaves. Note that if a label in some node is in $SI_{\oplus}(T, w)$ then its predecessor by rule (C) is also in $SI_{\oplus}(T, w)$. Hence we obtain on this $\oplus$-*eager* proof, by applying inductively Lemma 40, that the labels of all nodes are in $SI_{\oplus}(T, w)$. $\square$

Chapter 7

Conclusion

A summary of the results obtained on the complexity of the intruder deduction system modulo AC-like equational theories with homomorphism is given in the following table. The results for homomorphism only (without AC axioms) have been shown in a different paper [CLT03] and are here cited only for completeness.

	Intruder deduction problem ground case	
	Binary case	**General case**
Homomorphism	*PTIME [CLT03]*	
AC and Homomorphism	*PTIME*	*NP-Complete*
ACUN and Homomorphism	*PTIME*	*EXPTIME*
Abelian Groups and Homomorphism	*PTIME*	*EXPTIME*

The reason for the high complexity in the general case is a different one for the different equational theories considered, as shown in the following table:-

	Complexity in general case		
	Computation of sub-terms	**One step deductibility**	**General deductibility**
Homomorphism	*PTIME*	*PTIME*	*PTIME*
AC and Homomorphism	*PTIME*	*NP-Complete*	*NP-Complete*
ACUN and Homomorphism	*EXPTIME*	*PTIME*	*EXPTIME*
Abelian Groups and Homomorphism	*EXPTIME*	*PTIME*	*EXPTIME*

As future work, we plan to investigate the case of an active intruder. We can yet observe that it has been shown in [CDL04] that decidability of unification modulo an equational theory E is a necessary condition for the decidability of the security of a protocol for a bounded number of sessions and in presence of this equational theory E. Since unification modulo AC plus homomorphism is known undecidable [Nar96], the security against active attackers is undecidable at least for this equational theory as well.

Acknowledgments

The authors have profited enormously from the research environment provided by LSV. We received many useful remarks and pointers to existing results from our colleagues. It is now impossible to make a complete list of all the people who helped us with their comments, but the list comprises at least (in alphabetic order) Mathieu Baudet, Hubert Comon-Lundh, Stéphanie Delaune, Jean Goubault-Larrecq, Florent Jacquemard, Claudine Picaronny, and Philippe Schnoebelen. In particular, Philippe provided us with a first proof of Lemma 42.

Chapter 8

Appendix

8.1 One-Counter Automata

In this report we consider one-counter automata, which are non-deterministic finite-state automata operating on a single counter variable which takes values into $\mathbb{N}$. We denote by $|A|$ the number of states of the automaton A.

Definition 38 (One-Counter Automata) *A one-counter automaton A is a tuple (Q, Δ, q_0) where:*

- *Q is a finite set of control states.*
- *$q_0 \in Q$ is a distinguished initial control state.*
- *$\Delta \subseteq_{fin} Q \times Q \times \mathbb{N} \times \mathbb{Z}$ is a finite transition relation.*

A *configuration* of a one-counter automaton is a pair (q, c) composed of a state $q \in Q$ and of a natural number c which represents the current value of the counter variable. The transition relation between configurations is defined by $(q_1, c_1) \to (q_2, c_2)$ if there is some $(q_1, q_2, m, c) \in \Delta$ such that $c_1 \geq m$ and $c_2 = c_1 + c$. The reflexive transitive closure of $\to$ is denoted by $\to^\star$. A sequence of transitions leading from a configuration (q_1, c_1) to a configuration (q_2, c_2) is called a *path*. We denote by $\pi : (q_1, c_1) \to^\star (q_2, c_2)$ the fact that π is a path from (q_1, c_1) to (q_2, c_2). A path π' is a subpath of π iff $\pi = \pi_1 \pi' \pi_2$.

Given a one-counter automaton $A = (Q, \Delta, q_0)$ we define:

- $c_{max} = max\{c \mid \exists q_1, q_2, m : (q_1, q_2, m, c) \in \Delta\}$
- $g_{max} = max\{m \mid \exists q_1, q_2, c : (q_1, q_2, m, c) \in \Delta\}$
- $n = |A| * c_{max}$

Lemma 41 *Let A be a one-counter automaton and let $\pi : (q, c) \to^\star (q', c + n)$ be a path between the configuration (q, c) and the configuration $(q', c + n)$. Then there exists a subpath π' of π such that $\pi' : (q'', c'') \to^\star (q'', c'' + d)$ where q'' is some state and c'', d are natural numbers such that $0 < d \leq n$.*

Proof: We have a path between the state q with the counter c and the state q' with the counter $q + n$, where $n = |A| * c_{max}$. A transition increments or decrements the counter by at most c_{max}. Therefore there is a sequence $(q, c) = (q_1, c_1) \to^\star (q_2, c_2) \to^\star \ldots \to^\star (q_p, c_p) = (q', c + n)$ such that $c_1 < c_2 < \ldots < c_p$, $c_i - c_{i-1} \leq c_{max}$ for $i = 2, \ldots, p$ and $p > |A|$. Therefore there exists a state q'' that appears twice in the sequence yielding a subpath $\pi' : (q'', c'') \to^\star (q'', c'' + d)$ with $0 < d \leq n$. $\square$

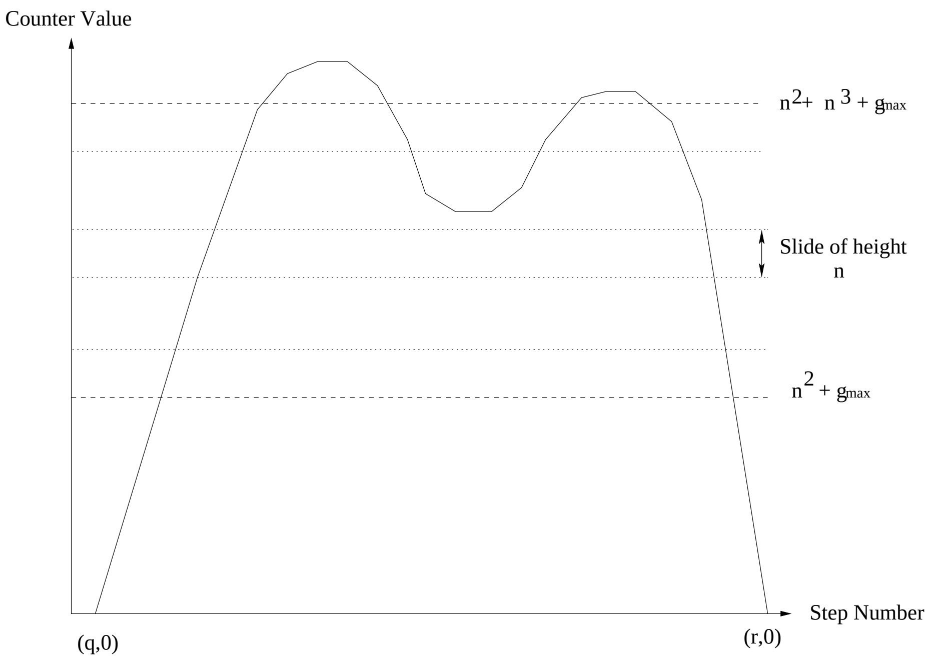


Figure 8.1: Illustration of Lemma 42

Lemma 42 *Let A be a one-counter automaton and let π be a path such that $\pi : (q,0) \rightarrow^\star (r,0)$. Then there exists a path $\pi' : (q,0) \rightarrow^\star (r,0)$ such that for any configuration (s,m) in π' the value m' of the counter is smaller than $n^3 + n^2 + g_{max}$.*

Proof:

Assume that $\pi : (q,0) \rightarrow^\star (r,0)$ is a path and that there exists a part of this path where the counter is greater than or equal to $n^3 + n^2 + g_{max}$. We may assume that nowhere in this path the counter has the value zero, otherwise we cut this path into two and analyze each part separately.

We aim at constructing a new path from π where the counter is always smaller than $n^3 + n^2 + g_{max}$ and with the same starting and ending configurations.

A peak is a subpath of π where the value of the counter is greater or equal to $n^3 + n^2 + g_{max}$ and the idea of the proof is to decrease a peak by removing parts of the path.

If between two peaks there is a "valley", *i.e.* the part between two peaks, which has a counter smaller than $n^2 + g_{\max}$, then we divide the problem into two parts: the first part between q and the lowest point of the valley, the second between the lowest point of the valley and the rest. We divide thus the path into several parts on which we can now perform some cuts.

We are going to delete some parts of the path. Everywhere on each deleted part the counter is between $n^2 + g_{max}$ and $n^2 + n^3 + g_{max}$. We distinguish the "ascent", *i.e.* the subpath between the last point where the counter is smaller than $n^2 + g_{max}$ and the peak, and the descent which is the subpath from the peak to the first point where the counter is smaller than $n^2 + g_{max}$.

We partition the ascent into slices where on each slice the counter increases by n; analogously we partition the descent into slices where on each slice the counter decreases by n, we can see this partition on Figure 8.1. There are at least n^2 slices between $n^2 + g_{max}$ and $n^2 + n^3 + g_{max}$. Using Lemma 41, we find in each slice a value d in the ascent and a value d' in the descent. Since there are n possible values for d, n possible values ford d' and n^2 slices there is one value for d which occurs at least n times, and one value for d' which occurs at least n times.

We remove $d' \leq n$ parts of size d from the ascent and $d \leq n$ parts of size d' from the descent. Since we cut only parts above $n^2 + g_{max}$, there are at most n "cuts" and each cut reduces the value of the counter by at most n, the counter cannot fall below 0, and all conditions remain valid. We have the same initial and final state and the height of the peak has decreased. By repeating this process we get a path such that the counter is always smaller than $n^3 + n^2 + g_{max}$. □

Corollary 43 *Let A be a one-counter automaton and $\pi : (q, c_q) \to^\star (r, c_r)$ a path between the state q with the counter $c_q \geq 0$ and the state r with the counter $c_r \geq 0$. Let $n' = max(n, \lceil\sqrt{c_q}\rceil, \lceil\sqrt{c_r}\rceil)$. There exists a path $\pi' : (q, c_q) \to^\star (r, c_r)$ between the state q with the counter $c_q \geq 0$ and the state r with the counter $c_r \geq 0$ such that the counter is always smaller than $n'^3 + n'^2 + g_{max}$.*

Proof: In the proof of Lemma 42 we have used the fact that the counter is initially and finally 0 only to obtain that the start and the end of the path are in a valley, that is below $n^2 + g_{max}$. Hence, we can reuse exactly the same proof where we have exchanged n by n'. □

8.2 Pushdown Automata

One-counter automata are a special case of pushdown automata when the stack vocabulary consists of a single letter (plus the special symbol denoting the bottom of the stack). This leads us to generalizing the previous lemma to pushdown automata. First, we recall the definitions of pushdown automata, then we prove the corresponding lemmata. Several definitions can be given for pushdown automata. Since we use pushdown automata as computing devices instead of accepting devices, we use the following definition.

Definition 39 (Pushdown Automaton) *A pushdown automaton M is defined as a system $(Q, \Gamma, \Delta, q_0, Z_0)$ where:*

- *Q is a finite set of states, including q_0.*
- *Γ is a finite alphabet called the stack alphabet including Z_0.*
- *$\Delta \subseteq_{fin} Q \times Q \times \Gamma^\star \times \Gamma^\star$ is a transition relation.*
- *q_0 is the initial state, an element of Q.*
- *Z_0 is the start symbol of the stack, an element of Γ.*

The additional power of pushdown automata (with respect to finite automata) comes from the stack which can be used as an infinite memory. A configuration is a pair $(q, Z_0 w)$ with $q \in Q, w \in (\Gamma - \{Z_0\})^*$. The initial configuration is (q_0, Z_0). The transition relation is extended to configurations by setting $(q, w) \to (q', w')$ iff $w = mv, w' = cv'$ and $(q, q', m, c) \in \Delta$. As in the case of one-counter automata, we denote by $\to^\star$ the reflexive transitive closure of $\to$ and a path π between two configurations is a sequence of transitions that allows to go from the first one to the second one.

Given a one-counter automaton $A = (Q, \Gamma, \Delta, q_0, Z_0)$ we define:

- $c_{max} = max\{|c| \mid \exists q_1, q_2, m : (q_1, q_2, m, c) \in \Delta\}$
- $g_{max} = max\{|g| \mid \exists q_1, q_2, c : (q_1, q_2, g, c) \in \Delta\}$
- $n = |A| * c_{max}$

Lemma 44 *Let A be a pushdown automaton and let $\pi : (q, w) \to^\star (q', w' = wu)$ be a path between the configuration (q, w) and the configuration (q', w') such that $|w'| = |w| + n$. Then there exists a subpath π' of of π such that $\pi' : (q'', w'') \to^\star (q'', w''t)$ with $0 < |t| \leq n$.*

Proof: We have a path between the state q with word w into the stack and the state q' with the word $w' = wt$ into the stack, where $|t| = n$ which is globally increasing.

We can push and pop by at most c_{max} symbols into the stack, so there is a sequence of at least $n + 1$ configurations $(q_0, w_0) \to^\star \ldots \to^\star (q_n, w_n)$ such that the word w_i is a prefix of the word w_{i+1} for $i = 0, \ldots, n - 1$, $|w| \leq |w_0|$, and w_n is a prefix of w'.

Therefore there is a state q'' that appears twice in the sequence.

One occurrence is in a configuration $(q'', w'' = w_i)$, where w'' is a prefix of w'.

Another occurrence is in a configuration $(q'', w''t = w_j)$ $(j > i)$ such that $|t| = d$, $0 < d \leq n$, and $w''t$ is prefix a of w'. $\square$

Lemma 45 *Let A be a pushdown automaton and $\pi : (s, \epsilon) \rightarrow^\star (s', \epsilon)$ a path between the state s with the word ϵ into the stack and the state s' with the word ϵ into the stack. There exists a path $\pi' : (s, \epsilon) \rightarrow^\star (s', \epsilon)$ between the state s with the word ϵ into the stack and the state s' with the word ϵ into the stack such that the size of the stack is always smaller than $n^3 + n^2 + g_{max}$.*

Proof:

Assume that $\pi : (s, \epsilon) \rightarrow^\star (s', \epsilon)$ is a path of A a pushdown automaton, with n states, and that there exists a part of this path where the size of the stack is greater than $n^3 + n^2 + g_{max}$. We assume that nowhere in this path the stack has the value ϵ, otherwise we cut this path into two and analyze each part separately. We are going to construct a new path from π where the size of the stack is always smaller than $n^3 + n^2 + g_{max}$. We use the same method that in the proof of Lemma 42 using now Lemma 44.

To this end, we try to lower a peak by removing parts of the path. We first find the peaks *i.e.* the parts of the path where the size of the stack is greater or equal to $n^3 + n^2 + g_{max}$. If between two peaks there is a "valley", *i.e.* the part between two peaks, which has a size of the stack smaller than $n^2 + g_{max}$, then we divide the problem into two parts: the first part between q and the lowest point of the valley, the second between the lowest point of the valley and the rest. We divide thus the path into several parts on which we can now perform some cuts.

We are going to delete some parts of the path. Everywhere on each deleted part the size of the stack is between $n^2 + g_{max}$ and $n^2 + n^3 + g_{max}$. We distinguish the "ascent", *i.e.* the subpath between the last point where the size of the stack is smaller than $n^2 + g_{max}$ and the peak, and the descent which is the subpath from the peak to the first point where the size of the stack is smaller than $n^2 + g_{max}$.

We partition the ascent into slices where on each slice the size of the stack increases by n; analogously we partition the descent into slices where on each slice the size of the stack decreases by n, we can see this partition on Figure 8.1. There are at least n^2 slices between $n^2 + g_{max}$ and $n^2 + n^3 + g_{max}$. Using Lemma 44, we find in each slice a value d in the ascent and a value d' in the descent. Since there are n possible values for d, n possible values ford d' and n^2 slices there is one value for d which occurs at least n times, and one value for d' which occurs at least n times.

We remove $d' \leq n$ parts of size d from the ascent and $d \leq n$ parts of size d' from the descent. Since we cut only parts above $n^2 + g_{max}$, there are at most n "cuts" and each cut reduces the value of the counter by at most n, the size of the stack cannot fall below ϵ, and all conditions remain valid. We have the same initial and final state and the height of the peak has decreased. By repeating this process we get a path such that the counter is always smaller than $n^3 + n^2 + g_{max}$. $\square$

Corollary 46 *Let A be pushdown automaton and $\pi : (q, w_q) \rightarrow^\star (r, w_r)$ a path between the state q with the word w_q with $|w_q| \geq 0$ into the stack and the state r with the word w_r with $|w_r| \geq 0$ into the stack. Let $n' = max(n, \lceil\sqrt{|w_q|}\rceil, \lceil\sqrt{|w_r|}\rceil)$ there exists a path $\pi' : (q, w_q) \rightarrow^\star (r, w_r)$ between the state q with the word w_q with $|w_q| \geq 0$ into the stack and the state r with the word w_r with $|w_r| \geq 0$ into the stack such that the size of the stack is always smaller than $n'^3 + n'^2 + g_{max}$.*

Proof: In the proof of Lemma 45 we have used the fact that the size of the stack is initially and finally ϵ only to obtain that the start and the end of the path are in a valley, that the size of the stack is below $n^2 + g_{max}$. Hence, we can reuse exactly the same proof where we have exchanged n by n'. $\square$

Bibliography

[80299] IEEE 802.11 Local and Metropolitan Area Networks: Wireless LAN Medium Acess Control (MAC) and Physical (PHY) Specifications, 1999.

[AG99] Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 148(1):1–70, January 1999.

[AN95] R. Anderson and R. Needham. Programming Satan's computer. In *Computer Science Today: Recent Trends and Developments*, volume 1000 of *Lecture Notes in Computer Science*, pages 426–440. Springer-Verlag, 1995.

[AR00] Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). In *Proc. 1st IFIP International Conference on Theoretical Computer Science (IFIP–TCS)*, volume 1872 of *Lecture Notes in Computer Science*, pages 3–22. Springer-Verlag, 2000.

[BGW01] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: The insecurity of 802.11. In *Proc. 7th Annual International Conference on Mobile Computing and Networking (MOBICOM'01)*, pages 180–188, Rome (Italy), 2001. ACM Press.

[BN98] Franz Baader and Tobias Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[CDL04] Véronique Cortier, Stéphanie Delaune, and Pascal Lafourcade. A survey of algebraic properties used in cryptographic protocols. Reseach Report LSV-04-15, LSV, ENS de Cachan, September 2004. 36 pages.

[CJ97] J. Clark and J. Jacob. A survey of authentication protocol literature. `http://www.cs.york.ac.uk/~jac/papers/drareviewps.ps`, 1997.

[CKRT03] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP decision procedure for protocol insecurity with XOR. In *Proc. of 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 261–270, Ottawa (Canada), 2003. IEEE Comp. Soc. Press.

[CLS03] H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In *Proc. of 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 271–280, Ottawa (Canada), 2003. IEEE Comp. Soc. Press.

[CLT03] Hubert Comon-Lundh and Ralf Treinen. Easy intruder deductions. In Nachum Dershowitz, editor, *Verification: Theory & Practice, Essays Dedicated to Zohar Manna on the Occasion of His 64th Birthday*, volume 2772 of *LNCS*, pages 225–242. Springer-Verlag, 2003.

[CM96] Evelyne Contejean and Claude Marché. CiME: Completion Modulo E. In Harald Ganzinger, editor, *7th International Conference on Rewriting Techniques and Applications*, volume 1103 of *Lecture Notes in Computer Science*, pages 416–419, New Brunswick, NJ, USA, July 1996. Springer-Verlag.

[DJ90] Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B - Formal Models and Semantics, chapter 6, pages 243–320. Elsevier Science Publishers and The MIT Press, 1990.

[DY83] D. Dolev and A.C. Yao. On the security of public-key protocols. In *Transactions on Information Theory*, volume 29, pages 198–208. IEEE Computer Society Press, March 1983.

[HU79] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, MA, 1979.

[Jac] Florent Jacquemard. Security protocols open repository. Available at `http://www.lsv.ens-cachan.fr/spore/index.html`.

[JKMS02] Petr Jančar, Antonín Kucera, Faron Moller, and Zdenek Sawa. Equivalence-checking with one-counter automata: A generic method for proving lower bounds. In *Proceedings of the 5th International Conference on Foundations of Software Science and Computation Structures*, pages 172–186. Springer-Verlag, 2002.

[KKS87] E. Kaltofen, M. S. Krishnamoorthy, and B. D. Saunders. Fast parallel computation of hermite and smith forms of polynomial matrices. *SIAM J. Algebraic Discrete Methods*, 8(4):683–690, 1987.

[Low95] G. Lowe. An attack on the Needham-Schroeder public key authentication protocol. *Information Processing Letters*, 56(3):131–133, 1995.

[McA93] David A. McAllester. Automatic recognition of tractability in inference relations. *JACM*, 40(2):284–303, April 1993.

[Nar96] Paliath Narendran. Solving linear equations over polynomial semirings. In *Proc. of 11th Annual Symposium on Logic in Computer Science (LICS)*, pages 466–472, July 1996.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Sch86] Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1986.

[Tur03] Mathieu Turuani. Personal communication, 2003.