Abstract Storage Devices*

Robert König[1,**], Ueli Maurer[2], and Stefano Tessaro[2]

[1] California Institute of Technology, Pasadena, CA 91125
rkoenig@caltech.edu
[2] Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland
{maurer,tessaros}@inf.ethz.ch

Abstract. The purpose of this paper is to initiate the study of a combinatorial abstraction, called *abstract storage device* (ASD), which models deterministic storage devices with the property that only partial information about the state can be read, but that there is a degree of freedom as to which partial information should be retrieved.
We study combinatorial problems related to ASD's, including reducibility among ASD's, which is proved to be $\mathcal{NP}$-complete, and the factorization of ASD's. In particular, we prove that the factorization into binary-output ASD's (if it exists) is unique.

1 Introduction

MOTIVATION. The term *storage device* is conventionally used for a physical device with a *write* and a *read* operation which can store data reliably, i.e., with the property that the read operation yields an exact copy of the data previously written into the device. This paper introduces a generalized type of storage devices for which the write operation consists of setting the device's state to some value in the state space, and the subsequent read operation provides some (usually only *partial*) information about the state.

Such a storage device is a relevant special case of a general physical system whose state can not be measured exactly. This may be due to intrinsic physical reasons: For instance, it is inherently impossible to perfectly measure a quantum state, unless it is known to be one of a set of orthogonal states. Furthermore, inaccuracy in retrieving the full state of a system may be due to practical constraints: Slow access to the data medium used in a storage device (consider e.g. a tape-based storage device) may only allow for efficient *partial* access to its contents, as retrieving the entire state of the device would be (at least in certain application contexts) infeasible.

The task of a conventional storage device (e.g., a hard disc) is to store information reliably. For this reason, the design goal of such a system is to define a finite subset of its state space (as large as possible) such that the available read operation allows to distinguish different such states with negligible error probability, and a storage device

* This research was partially supported by the Swiss National Science Foundation (SNF), project no. 200020-113700/1.
** Work done while at ETH Zurich.

is hence traditionally characterized in terms of its *storage capacity*, i.e., the number of bits that can be stored reliably in it. Here, however, we take a more general approach to storage devices, by modeling explicitly the fact that, on one hand, a read operation provides only partial information about the state, but that, on the other hand, many different such read operations can be available.

There are different motivations for considering such a setting. A first motivation is *quantum cryptography* or, more precisely, *privacy amplification*, the last step of a quantum key agreement protocol (see [6]). In simplified terms, an adversary is assumed to have access to a bit string S of length n, shared by the legitimate users, and can store information about S in a 2^k-dimensional quantum device, where $k < n$. Since the (reliable) storage capacity of the device is only k, the adversary cannot store S perfectly. Later, the legitimate users select a hash function h from n bits to t bits (where $t < k$) at random from a class of such functions, and the adversary can now perform a measurement of the quantum state, *depending* on the choice of h. In this context, the goal is to prove that every such measurement yields only a negligible amount of information about $h(S)$. One can naturally generalize the setting of privacy amplification to other types of storage devices.

Subsequent to the on-line publication of a first version of this paper [7], Köpf and Basin [8] have suggested that generalized storage devices (and in particular the concept of ASD's proposed in this paper) can be used to model leakage of information in *side-channel attacks*. An adversary mounting such an attack is given physical access to some cryptographic device (e.g. a smart card) storing a secret value (generally the secret key) in a tamper-proof way, and in a legitimate interaction with the device the adversary gains additional side-channel information (such as the power consumption of the device or spurious electromagnetic radiation) through appropriate physical measurements. This information is generally correlated with the secret and can hence be of substantial help in breaking the given device. As proposed in [8], this setting can be abstracted in terms of a storage device with the secret value as its state and such that each allowed input to the cryptographic device is interpreted as a read operation resulting in the corresponding secret-dependent side information.

As a final motivating example, we look at the abstract general game in which an entity, say Alice, is given access to an n-bit string $s = [s_1, \ldots, s_n]$ about which she stores partial information. Later, she will learn a function f drawn from a given set and will have to guess $f(s)$. For example, this set of functions might consist of all linear predicates $a_1 s_1 + \cdots + a_n s_n \pmod 2$ for some $a_1, \ldots, a_n \in \{0, 1\}$. A natural question is finding the minimal amount of reliable storage required to win this game. More generally, one may be interested in deciding whether keeping information about s in a certain storage device suffices to succeed in the game. It is also interesting to compare such games in the sense of determining whether one game is strictly more difficult than another one.

CONTRIBUTIONS OF THIS PAPER. This paper introduces the notion of an *abstract storage device (ASD)*, a combinatorial abstraction which models the described property that only partial information about the state can be read, but that there is a degree of freedom as to which partial information should be retrieved. Both generalized storage devices as well as the above game can be described as an ASD. Here we only consider

deterministic storage devices, i.e., we analyze the case with no error probability. This is similar in spirit to the investigation of the *zero-error capacity* [10] in communication theory. Like there, the treatments of the zero-error and the negligible-error cases are quite different and deserve separate investigation. (Partial results in the probabilistic case have been given in [11].)

We study the central question of *reducibility* of devices, namely deciding whether a certain device can be implemented by another one. This concept directly yields a notion of *equivalence* of devices, as well as different natural quantities characterizing ASD's: The *storage capacity* provides a measure of the amount of information that can be reliably stored in a device, the *state complexity* characterizes the minimal amount of reliable storage needed to simulate the device, and the *perfectness index* of an ASD is the minimal number of read operations needed to entirely retrieve the state of the device. We show that these quantities yield easily verifiable necessary conditions for reducibility, and we give relations among these quantities. However, we prove the general problem of deciding reducibility of ASD's to be $\mathcal{NP}$-complete, whereas deciding equivalence of ASD's is shown to be at least as difficult as deciding the isomorphism of graphs. Also, equivalence of ASD's is unlikely to be $\mathcal{NP}$-complete, as its $\mathcal{NP}$-completeness would imply a collapse of the polynomial hierarchy.

In order to investigate the structural properties of ASD's and to considerably simplify the general question of deciding both reducibility and equivalence, the last part of this paper introduces the concept of factorizations of ASD's as the *direct product* (i.e. the parallel composition) of simpler ASD's. We prove that every device admits a unique factorization in terms of *binary* devices (i.e. with binary output), if such a factorization exists. This result is to be seen as a first step towards answering the general question of the existence of unique factorizations into (prime) ASD's, which we state as an open problem. It also adds an additional contribution to the long line of work investigating product factorizations of discrete structures, such as graphs and finite relational structures (see [4, 5] for respective surveys).

OUTLINE OF THIS PAPER. The remainder of this paper is organized as follows. Section 3.1 introduces ASD's, provides some examples, and defines the basic composition operations for ASD's (such as direct products). Sections 3.2 and 3.3 deal with the central problems of reducibility and equivalence of ASD's, and in Section 3.4 we present and analyze relevant quantities related to ASD's, such as the storage capacity, the state complexity, and the perfectness index. In Section 4, we investigate the complexity of reducibility and equivalence of ASD's, whereas the last section (Section 5) addresses direct product factorizations of ASD's.

Relevant basic facts about set partitions and the partition lattice are briefly reviewed in Section 2. For lack of space, some technical proofs are omitted and can be found in the full version of this paper [7].

2 Preliminaries

Throughout this paper, we make use of capital calligraphic letters to denote sets. An (undirected) *graph* is an ordered pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of *vertices*, and $\mathcal{E} \subseteq \binom{\mathcal{V}}{2}$ is the set of *edges* of $\mathcal{G}$.

A *(set) partition* π of a set $\mathcal{S}$ is a family $\{\mathcal{B}_1, \ldots, \mathcal{B}_k\}$ of disjoint non-empty subsets of $\mathcal{S}$, called *blocks*, with the property that $\bigcup_{i=1}^{k} \mathcal{B}_i = \mathcal{S}$. We write $s \equiv_\pi t$ whenever both elements $s, t \in \mathcal{S}$ are in the same block of π. Moreover, we denote by $\Pi(\mathcal{S})$ the set of partitions of $\mathcal{S}$. We say that $\pi \in \Pi(\mathcal{S})$ *refines* $\pi' \in \Pi(\mathcal{S})$, denoted $\pi \sqsubseteq \pi'$, if for all $\mathcal{B} \in \pi$ there exists a $\mathcal{B}' \in \pi'$ such that $\mathcal{B} \subseteq \mathcal{B}'$. Recall that $(\Pi(\mathcal{S}); \sqsubseteq)$ is a bounded lattice (cf. e.g. [3]), with the minimal element being $id_{\mathcal{S}} = \{\{s\} \mid s \in \mathcal{S}\}$ and the maximal element being $\{\mathcal{S}\}$. The *meet* of $\pi, \pi' \in \Pi(\mathcal{S})$ is the partition $\pi \wedge \pi' = \{\mathcal{B} \cap \mathcal{B}' \mid \mathcal{B} \in \pi, \mathcal{B}' \in \pi', \mathcal{B} \cap \mathcal{B}' \neq \emptyset\}$, whereas their *join* $\pi \vee \pi'$ is such that $x \equiv_{\pi \vee \pi'} y$ if and only if we can find a sequence of elements $x = x_0, x_1, \ldots, x_r = y$ (for some r) such that $x_i \equiv_\pi x_{i+1}$ or $x_i \equiv_{\pi'} x_{i+1}$ holds for all $i = 0, \ldots, r-1$. For a set Π of partitions, we generally write $\bigwedge \Pi = \bigwedge_{\pi \in \Pi} \pi$ and $\bigvee \Pi = \bigvee_{\pi \in \Pi} \pi$. Also, such a set Π is called an *antichain* if $\pi \not\sqsubseteq \pi'$ for all distinct $\pi, \pi' \in \Pi$.

The *direct product* of the partitions $\pi \in \Pi(\mathcal{S})$ and $\pi' \in \Pi(\mathcal{S}')$ is the partition $\pi \times \pi' = \{\mathcal{B} \times \mathcal{B}' \mid \mathcal{B} \in \pi, \mathcal{B}' \in \pi'\} \in \Pi(\mathcal{S} \times \mathcal{S}')$. In particular, we have $(s, s') \equiv_{\pi \times \pi'} (t, t')$ if and only if $s \equiv_\pi s'$ and $t \equiv_{\pi'} t'$ for all $s, t \in \mathcal{S}$, $s', t' \in \mathcal{S}'$. Let now $\pi, \rho \in \Pi(\mathcal{S}), \pi', \rho' \in \Pi(\mathcal{S}')$ be partitions. Then, both equalities $(\pi \wedge \rho) \times (\pi' \wedge \rho') = (\pi \times \pi') \wedge (\rho \times \rho')$ and $(\pi \vee \rho) \times (\pi' \vee \rho') = (\pi \times \pi') \vee (\rho \times \rho')$ hold. Furthermore, $\pi \times \pi' \sqsubseteq \rho \times \rho'$ is satisfied if and only if $\pi \sqsubseteq \pi'$ and $\rho \sqsubseteq \rho'$.

Given sets $\mathcal{S}, \mathcal{S}'$, a partition $\pi \in \Pi(\mathcal{S}')$, and some function $\phi : \mathcal{S} \to \mathcal{S}'$, we define $\pi \circ \phi \in \Pi(\mathcal{S})$ as the partition such that $x \equiv_{\pi \circ \phi} y$ if and only if $\phi(x) \equiv_\pi \phi(y)$ for all $x, y \in \mathcal{S}$. Notice that $(\pi \circ \phi) \wedge (\pi' \circ \phi) = (\pi \wedge \pi') \circ \phi$, and $(\pi \circ \phi) \vee (\pi' \circ \phi) = (\pi \vee \pi') \circ \phi$. Moreover, the *kernel (partition)* of a function $f : \mathcal{X} \to \mathcal{Y}$ is $\ker(f) = \{f^{-1}(\{y\}) \mid y \in \text{range}(f)\}$. Given a further function $\phi : \mathcal{S} \to \mathcal{X}$, we have $\ker(f \circ \phi) = \ker(f) \circ \phi$.

Finally, recall that a k-variate *lattice polynomial* p in the variables $x_1, \ldots, x_k$ is a formal expression of the form either (i) x_i for $i = 1, \ldots, k$, or (ii) one of $q(x_1, \ldots, x_k) \wedge q'(x_1, \ldots, x_k)$ and $q(x_1, \ldots, x_k) \vee q(x_1, \ldots, x_k)$ for k-variate lattice polynomials q, q'. Given partitions $\pi_1, \ldots, \pi_k$, $\rho_1, \ldots, \rho_k$ such that $\pi_i \sqsubseteq \rho_i$ for $i = 1, \ldots, k$, then for every k-variate lattice polynomial p we have $p(\pi_1, \ldots, \pi_k) \sqsubseteq p(\rho_1, \ldots, \rho_k)$.

3 Abstract Storage Devices

3.1 Definition

In the following, we look at storage devices used by two entities, called the *writer* and the *reader*, respectively.[3] The writer writes to such a device by selecting a state s from the *state space* of the device. The reader subsequently chooses a (possibly randomized) function g mapping states to output symbols from a set of possible such mappings, and obtains the output $g(s)$. Note, however, that the actual labeling of the outputs is irrelevant, as long as the reader knows a complete description of the function to be read out. In particular, as we only focus on devices whose behavior is entirely *deterministic*, we abstract from the notion of an output domain and we solely describe the kernel partitions of the functions of the storage device. This allows us to formulate the following combinatorial abstraction of deterministic devices.

[3] These entities are not necessarily distinct in a physical sense.

Definition 1. *An* abstract storage device (ASD) D *is a pair* $D = \left(\mathcal{S}^D, \Pi^D\right)$*, where* $\mathcal{S}^D$ *is a set called the* state space of D*, and* Π^D *is a family of partitions of* $\mathcal{S}^D$*, called the* partition set of D.

For an ASD D, a *write operation* of the writer consists in selecting a state $s \in \mathcal{S}^D$, and in a subsequent *read operation* the reader selects a partition $\pi \in \Pi^D$ and learns the (unique) block $\mathcal{B} \in \pi$ such that $s \in \mathcal{B}$. We assume that a single read operation is performed. Furthermore, in the following, we are going to focus on ASD's with finite state space and partition set.

Whenever $id_{\mathcal{S}^D} \in \Pi^D$, the reader can distinguish any pair of states with a single read operation. In this case, D is called *perfect*, and it is called *non-perfect* otherwise. If the partition set contains only the trivial partition $\{\mathcal{S}^D\}$, the ASD is called *trivial*. Moreover, it is called r*-regular* if $|\pi| = r$ for all $\pi \in \Pi^D$. In particular, 2-regular ASD's are also called *binary*.

The following are examples of ASD's.

- For a given set $\mathcal{X}$, the *perfect device* $C_{\mathcal{X}}$ has state space $\mathcal{X}$ and its state can be retrieved perfectly, that is, $\Pi^D = \{id_{\mathcal{X}}\}$. The special case where $\mathcal{X} = \{1, \ldots, m\}$ for $m \in \mathbb{N}$ is denoted as C_m.
- For $i \in \{1, \ldots, n\}$, let $p_i : \{0,1\}^n \to \{0,1\}$ be such that $p_i(x_1, \ldots, x_n) = x_i$ for all $(x_1, \ldots, x_n) \in \{0,1\}^n$. The *projective device* P_n has state space $\mathcal{S}^{P_n} = \{0,1\}^n$ and its partition set is $\Pi^{P_n} = \{\ker(p_i) \,|\, i = 1, \ldots, n\}$. This device is similar to the 1*-out-of-*n *oblivious transfer (OT)* primitive considered in cryptography (introduced in [9]). One may also extend this device to allow for retrieving any $k < n$ consecutive bits of the state. Such a device could be used to model a tape-based storage device.
- The *linear device* $L_{n,k}$ where $n \geq k$ is the ASD having state space $\mathcal{S}^{L_{n,k}} = \{0,1\}^n$, and the partition set is the set of the kernel partitions of all linear maps $\{0,1\}^n \to \{0,1\}^k$. We denote by L_n the binary ASD $L_{n,1}$.

One way of constructing a complex device from simpler devices is the parallel composition of two ASD's to obtain a new ASD modeling a setting where the reader and the writer use both devices in a *non-adaptive* fashion. That is, if D has state s and D' has state s', the reader first selects *both* partitions $\pi \in \Pi^D$ and $\pi' \in \Pi^{D'}$, and only subsequently learns the unique blocks $\mathcal{B} \in \pi$, $\mathcal{B}' \in \pi'$ such that $s \in \mathcal{B}$ and $s' \in \mathcal{B}'$.

Definition 2. *The* direct product $D \times D'$ *of the ASD's* D, D' *is the ASD with* $\mathcal{S}^{D \times D'} = \mathcal{S}^D \times \mathcal{S}^{D'}$ *and* $\Pi^{D \times D'} = \{\pi \times \pi' \,|\, \pi \in \Pi^D, \pi' \in \Pi^{D'}\}$.

For example, since $id_{\mathcal{S}^D \times \mathcal{S}^{D'}} = \pi \times \pi'$ holds if and only if $\pi = id_{\mathcal{S}^D}$ and $\pi' = id_{\mathcal{S}^{D'}}$, we immediately see that $D \times D'$ is perfect if and only if both D and D' are perfect.

In general, we may want to look at more than a single read operation. For an integer $k \geq 1$ and an ASD D, we denote as $D^{(k)}$ the ASD with $\mathcal{S}^{D^{(k)}} = \mathcal{S}^D$ and $\Pi^{D^{(k)}} = \left\{ \bigwedge_{i=1}^k \pi_i \,\middle|\, \pi_i \in \Pi^D, i = 1, \ldots, k \right\}$. It models the scenario where the reader is allowed to perform (at most) k non-adaptive read operations, i.e. given state $s \in \mathcal{S}^D$, it first

chooses k partitions $\pi_1, \ldots, \pi_k \in \Pi^D$ to be retrieved, and only subsequently learns the corresponding blocks $\mathcal{B}_1 \in \pi_1, \ldots, \mathcal{B}_k \in \pi_k$ such that $s \in \bigcap_{i=1}^{k} \mathcal{B}_i$.

Note that both the direct product and the device $D^{(k)}$ can be extended to allow for adaptive read operations, as it essentially suffices to consider all partitions induced by every possible (deterministic) retrieval strategy. However, we do not address this case in this paper.

3.2 Reducibility and Equivalence

In the problem of reducibility of ASD's, we want to decide whether an ASD D can be implemented by a second ASD D'. This is formalized by the following definition.

Definition 3. *We say that an ASD D is* reducible *to an ASD D', denoted $D \leq D'$, if there exist functions $\phi : \mathcal{S}^D \to \mathcal{S}^{D'}$ and $\alpha : \Pi^D \to \Pi^{D'}$ such that $\alpha(\pi) \circ \phi \sqsubseteq \pi$ for all $\pi \in \Pi^D$. Such a pair of functions (ϕ, α) is called a* reduction of D to D'.

In order to clarify this concept, consider the following abstraction in terms of ASD's of the game introduced in Section 1. The writer and the reader are given an ASD D' as well as the description of a further ASD D. The writer is told an arbitrary state $s \in \mathcal{S}^D$ and selects the state $\phi(s) \in \mathcal{S}^{D'}$ for D'. Later, an arbitrary partition $\pi \in \Pi^D$ is revealed to the reader, and it performs a read operation for a partition $\alpha(\pi) \in \Pi^{D'}$. The goal is to find appropriate functions $\phi : \mathcal{S}^D \to \mathcal{S}^{D'}$ and $\alpha : \Pi^D \to \Pi^{D'}$ such the reader can *perfectly* guess the unique block $\mathcal{B} \in \pi$ such that $s \in \mathcal{B}$ from the result of retrieving $\alpha(\pi)$ from D'. If such functions exist, the writer and the reader can simulate D using D'. Note that the ASD D itself can alternatively be seen as the specification of a particular game the writer and the reader try to win by using the ASD D'.

It is easy to see that the condition $\alpha(\pi) \circ \phi \sqsubseteq \pi$ must hold. Otherwise, there would be $s, s' \in \mathcal{S}^D$ such that $s \not\equiv_\pi s'$, but $\phi(s) \equiv_{\alpha(\pi)} \phi(s')$, and hence s and s' could not be distinguished. Conversely, if $\alpha(\pi) \circ \phi \sqsubseteq \pi$, then given state $s \in \mathcal{S}^D$ and $\mathcal{B}' \in \alpha(\pi)$ such that $\phi(s) \in \mathcal{B}'$, there exists a unique block $\mathcal{B} \in \pi$ such that $s \in \mathcal{B}$. Hence, Definition 3 expresses the precise condition in order for ϕ and α to be a winning strategy in the game.

Reducibility is a reflexive and transitive relation. However, it is not antisymmetric, and thus it is only a *quasi-order* on the set of ASD's. In this respect, we say that two ASD's D, D' are *equivalent*, denoted $D \equiv D'$, if both $D \leq D'$ and $D' \leq D$ hold. The relation $\equiv$ is an equivalence relation and reducibility implicitly defines a partial order on its equivalence classes.

The following proposition relates reducibility to direct products and multiple read operations.

Proposition 1. *Let D, D', E, E' be ASD's.*

(i) If $D \leq D'$ and $E \leq E'$, then $D \times E \leq D' \times E'$.
(ii) If $D \leq D'$, then $D^{(k)} \leq D'^{(k)}$.

Proof. The first claim is obvious. For the second one, let (ϕ, α) be a reduction of D to D'. Define $\tilde{\alpha} : \Pi^{D^{(k)}} \to \Pi^{D'^{(k)}}$ such that $\tilde{\alpha}(\bigwedge_{i=1}^{k} \pi_i) = \bigwedge_{i=1}^{k} \alpha(\pi_i)$. Then, $(\phi, \tilde{\alpha})$

reduces $D^{(k)}$ to $D^{(k')}$, since $\tilde{\alpha}(\bigwedge_{i=1}^{k} \pi_i) \circ \phi = \left(\bigwedge_{i=1}^{k} \alpha(\pi_i)\right) \circ \phi = \bigwedge_{i=1}^{k} (\alpha(\pi_i) \circ \phi) \sqsubseteq \bigwedge_{i=1}^{k} \pi_i$. □

The perhaps most natural question related to storage devices is to determine how many bits of information can be reliably stored in it with the guarantee of no errors at read out. This quantity can be expressed in terms of the largest perfect device that can be reduced to the considered device.

Definition 4. *The* storage capacity *of an ASD D is* $C(D) = \max\{\log m \,|\, C_m \leq D, m \in \mathbb{N}\}$.

Equivalence of ASD's captures that two ASD's D and D' such that $D \equiv D'$ have the same behavior. As an example, it is clear that $D \times D' \equiv D' \times D$, and that $D \times (D' \times D'') \equiv (D \times D') \times D''$, that is, the direct product is commutative and associative with respect to equivalence. The direct product of $D_1, \ldots, D_n$ is thus simply written as $\times_{i=1}^{n} D_i$, and $D^k = \times_{i=1}^{k} D$ for any device D. Finally, notice that $D \times E \equiv D$ holds for any trivial device E.

3.3 Minimality

In this section, we have a closer look at the equivalence relation $\equiv$ and at the inner structure of its equivalence classes. In particular, we are interested in the minimal number of states and partitions needed in order to implement the functionality of a certain ASD.

Definition 5. *An ASD D is* state-minimal *if there is no equivalent device D' with $|\mathcal{S}^{D'}| < |\mathcal{S}^{D}|$. Furthermore, D is* partition-minimal *if there is no equivalent device D' with $|\Pi^{D'}| < |\Pi^{D}|$. Finally, we say that D is* minimal *if D is both state and partition-minimal.*

For every ASD D there exist by definition equivalent ASD's D' and D'' such that D' is state-minimal and D'' is partition minimal. However, it is not clear whether an equivalent ASD exists that satisfies both, i.e., which is minimal. This is shown in the following theorem, which also provides an equivalent characterization of state and partition-minimality. The proof is given in [7].

Theorem 1. *For an ASD D we have the following.*

(i) D is state-minimal if and only if for all pairs of distinct states $s, s' \in \mathcal{S}^D$ there exists a set partition $\pi \in \Pi^D$ such that $s \not\equiv_\pi s'$. In particular, this holds if and only if $\bigwedge \Pi^D = id_{\mathcal{S}^D}$.
(ii) D is partition-minimal if and only if Π^D is an antichain (with respect to $\sqsubseteq$).

Furthermore, for every ASD D, there exists a minimal ASD $D' \equiv D$.

As an example, observe that the projective device P_n is state minimal. Indeed, given distinct $x, x' \in \{0,1\}^n$, there exists a component i such that $x_i \neq x'_i$, and thus $x \not\equiv_{\ker(p_i)} x'$. This also implies that the linear device L_n is state-minimal. Furthermore,

every r-regular device (for some r) is necessarily partition-minimal, since any two partitions with the same number of blocks are either equal or incomparable (with respect to $\sqsubseteq$).

The following lemma provides some properties of minimal devices with respect to device reducibility and it is proved in the full version.

Lemma 1. *(i) If D, D' are state-minimal and (ϕ, α) reduces D to D', then ϕ is injective. In particular, $|\mathcal{S}^D| \leq |\mathcal{S}^{D'}|$.*
(ii) If D, D' are both r-regular for some r (and hence partition minimal) and (ϕ, α) reduces D to D', then α is injective. In particular, $|\Pi^D| \leq |\Pi^{D'}|$.
(iii) If D, D' are both state-minimal (partition-minimal), then the direct product $D \times D'$ is state-minimal (partition-minimal).

It also turns out that equivalence of devices is easier to characterize in the minimal case, and the following proposition can be shown.

Proposition 2. *Let D, D' be minimal ASD's. Then $D \equiv D'$ if and only if there exist bijections $\phi : \mathcal{S}^D \to \mathcal{S}^{D'}$ and $\alpha : \Pi^D \to \Pi^{D'}$ such that $\pi = \alpha(\pi) \circ \phi$ for all $\pi \in \Pi^D$, or, equivalently, $\pi' = \alpha^{-1}(\pi') \circ \phi^{-1}$ for all $\pi' \in \Pi^{D'}$.*

For example, given ASD's D, D', where $\Pi^D = \{\pi_1, \ldots, \pi_k\}$, as well as a k-variate lattice polynomial p, Proposition 2 implies $p(\alpha(\pi_1) \circ \phi, \ldots, \alpha(\pi_k) \circ \phi) = p(\alpha(\pi_1), \ldots, \alpha(\pi_k)) \circ \phi = p(\pi_1, \ldots, \pi_k)$. As ϕ is a bijection, in order to prove that $D \not\equiv D'$ it is sufficient to find a k-variate lattice polynomial p such that $|p(\pi_1, \ldots, \pi_k)| \neq |p(\alpha(\pi_1), \ldots, \alpha(\pi_k)|$.

3.4 Necessary Conditions for Reducibility

In this section, we discuss easily characterizable necessary conditions for reducibility. Let $\mathcal{D}$ be a set of ASD's and let $f : \mathcal{D} \to \mathbb{R}$ be a function. We say that f is *order-preserving on* $\mathcal{D}$ if $D \leq D'$ implies $f(D) \leq f(D')$ for all ASD's $D, D' \in \mathcal{D}$. In particular, note that $f(D) = f(D')$ whenever $D \equiv D'$. Such a function yields a necessary condition for reducibility. In the following paragraphs, we discuss three order-preserving functions.

STORAGE CAPACITY. The storage capacity (cf. Section 3.2) is order-preserving on the set of all ASD's: Given D, D' such that $D \leq D'$, let m be maximal such that $C_m \leq D$. By transitivity we have $C_m \leq D'$, and hence $\log m = C(D) \leq C(D')$. The storage capacity is easy to compute, as stated in the following proposition, which also provides properties with respect to direct products and multiple read operations.

Proposition 3. *(i) $C(D) = \max_{\pi \in \Pi^D} \log |\pi|$ for all ASD's D.*
(ii) $C(D \times D') = C(D) + C(D')$ for all ASD's D, D'.
(iii) For all $k \geq 1$, we have $C(D^{(k)}) \leq k \cdot C(D)$ for all ASD's D.

The first claim follows from the simple observation that $C_m \leq D$ holds if and only if there exists $\pi \in \Pi^D$ such that $|\pi| \geq m$. The simple proofs of (ii) and (iii) are omitted.

For instance, $C(D) = \log r$ for every r-regular ASD D. Furthermore, the storage capacity allows us to easily see that $L_2 \times L_2 \times L_2 \not\leq L_3 \times L_3$, since $C(L_2 \times L_2 \times L_2) = 3 \cdot C(L_2) = 3$, but $C(L_3 \times L_3) = 2 \cdot C(L_3) = 2$.

STATE COMPLEXITY. The *state complexity* $\sigma(D)$ of an ASD D provides the minimal number of states that are necessary in order to reproduce the behavior of D, that is, $\sigma(D) = \min_{E \equiv D} \log |\mathcal{S}^E|$. The state complexity is order-preserving: Given devices D, D' with $D \leq D'$, let E, E' be state-minimal such that $D \equiv E$ and $D' \equiv E'$. We have $E \leq E'$ by transitivity, and by Lemma 1 this implies $\sigma(D) = \log |\mathcal{S}^E| \leq \log |\mathcal{S}^{E'}| = \sigma(D')$. Furthermore, $\sigma(D \times D') = \sigma(D) + \sigma(D')$ by Lemma 1.

Note that $D \leq C_{2^{\sigma(D)}}$, whereas Lemma 1 yields $D \not\leq C_{m'}$ for all $m' < 2^{\sigma(D)}$. For this reason, we obtain $\sigma(D) = \min\{\log m \,|\, m \in \mathbb{N}, D \leq C_m\}$. Therefore, the state complexity $\sigma(D)$ provides the minimal amount of reliable storage in terms of bits needed to win the game (in the sense of Section 3.2) described by the ASD D.

PERFECTNESS INDEX. The *perfectness index* $i(D)$ of a device D is the minimal integer k such that $D^{(k)}$ is perfect, if such k exists. Otherwise, $i(D) = \infty$. Thus, $i(D)$ provides the minimal number of (non-adaptive) read operations needed to retrieve the state perfectly. If $i(D)$ is finite, then in particular $i(D) \leq |\Pi^D|$, and by Theorem 1 $i(D)$ is bounded if and only if D is state-minimal. In the following, for an integer m, consider the set of ASD's $\mathcal{D}_m$ such that for all $D \in \mathcal{D}_m$ we have $|\mathcal{S}^D| = m$.

Proposition 4. *Let $D, D' \in \mathcal{D}_m$ for some m be such that $D \leq D'$. Then, $i(D) \geq i(D')$. That is, $D \mapsto -i(D)$ is an order-preserving function on $\mathcal{D}_m$.*

Proof. If $i(D) = \infty$ holds, the claim is trivially satisfied. Therefore, assume that $i(D)$ is finite, and, towards a contradiction, that $D \leq D'$, but $i(D) < i(D')$. There is an integer $k \geq 1$ such that $D^{(k)}$ is perfect, but $D'^{(k)}$ is not. Thus, $id_{\mathcal{S}^D} \in \Pi^{D^{(k)}}$, but $id_{\mathcal{S}^{D'}} \notin \Pi^{D'^{(k)}}$. Since $|\mathcal{S}^D| = |\mathcal{S}^{D'}| = m$, for all possible $\phi : \mathcal{S}^D \to \mathcal{S}^{D'}$ there is no partition $\pi' \in \Pi^{D'^{(k)}}$ such that $\pi' \circ \phi \sqsubseteq id_{\mathcal{S}^D}$. Hence $D^{(k)} \not\leq D'^{(k)}$, which contradicts $D \leq D'$ according to Proposition 1. □

One can easily verify that $i(\bigtimes_{i=1}^n D_i) = \max_{1 \leq i \leq n} i(D_i)$ for ASD's $D_1, \ldots, D_n$. Furthermore, $i(L_n) = n$, since exactly n distinct, linearly independent, linear predicates have to be read out to learn the state. As an example, consider the ASD's $L_4 \times L_2$ and $L_3 \times L_3$. By the above, we have $i(L_4 \times L_2) = 4$, and $i(L_3 \times L_3) = 3$. Therefore, $L_3 \times L_3 \not\leq L_4 \times L_2$ by Proposition 4.

The presented quantities are related by the following proposition.

Proposition 5. *For all ASD's D, we have $\sigma(D) \leq i(D) \cdot C(D)$.*

Proof. The claim is trivially true if $i(D) = \infty$. Otherwise, we just combine the facts that $\sigma(D) \leq C(D^{(i(D))}) = \log |\mathcal{S}^D|$ and that $C(D^{(i(D))}) \leq i(D) \cdot C(D)$. □

4 Complexity of Reducibility and Equivalence

We investigate the computational complexity of deciding reducibility and equivalence of ASD's. Both problems are obviously in $\mathcal{NP}$, since given a reduction (ϕ, α) reducibility can be verified in polynomial-time (in the numbers of states and partitions),[4] and

[4] We assume some canonical encoding of ASD's.

hence also equivalence (by giving two corresponding reductions). In this section, we prove the following theorem.

Theorem 2. *Reducibility of ASD's is $\mathcal{NP}$-complete. Furthermore, deciding equivalence of ASD's is at least as hard as deciding graph isomorphism.*

First, we briefly recall some graph-theoretic notions. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is *isomorphic* to $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, denoted $\mathcal{G} \cong \mathcal{G}'$, if there exists a bijection $\phi : \mathcal{V} \to \mathcal{V}'$ such that $\{v, w\} \in \mathcal{E}$ if and only if $\{\phi(v), \phi(w)\} \in \mathcal{E}'$. Furthermore, $\mathcal{G}$ is a *subgraph* of $\mathcal{G}'$ if $\mathcal{V} \subseteq \mathcal{V}'$ and $\mathcal{E} \subseteq \mathcal{E}'$. Finally, $\mathcal{G}$ is *contained* in $\mathcal{G}'$, denoted $\mathcal{G} \preceq \mathcal{G}'$, if there exists a subgraph $\mathcal{H}$ of $\mathcal{G}'$ such that $\mathcal{G} \cong \mathcal{H}$. Let $\mathcal{K}_k$ be the complete graph on k vertices. The *k-clique problem* consists in deciding, given a graph $\mathcal{G}$, whether $\mathcal{K}_k \preceq \mathcal{G}$. For arbitrary k, this is a well-known $\mathcal{NP}$-complete problem.

In order to prove Theorem 2, we introduce a class of ASD's representing graphs. For a given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we define its *graph device* $D(\mathcal{G})$ as the 3-regular ASD such that $\mathcal{S}^{D(\mathcal{G})} = \mathcal{V}$ and $\Pi^{D(\mathcal{G})} = \{\pi_e \mid e \in \mathcal{E}\}$, where for $e = \{u, v\} \in \mathcal{E}$, we have $\pi_e = \big\{\{u\}, \{v\}, V - \{u, v\}\big\}$. Note that graph devices are only meaningful if $|\mathcal{V}| \geq 4$, since in the case where $|\mathcal{V}| = 3$, all edges define the same partition.

For instance, if one takes the complete graph $\mathcal{K}_k$ (for $k \geq 4$), the resulting graph device $D(\mathcal{K}_k)$ has state space $\{1, \ldots, k\}$ and its partition set contains all partitions of the form $\{\{i\}, \{j\}, \{1, \ldots, k\} - \{i, j\}\}$ for all $i < j$, $i, j \in \{1, \ldots, k\}$.

The following result can easily be verified using Theorem 1.

Lemma 2. *The ASD $D(\mathcal{G})$ is minimal for all graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| \geq 4$ and no isolated[5] vertices.*

The following lemma is the central point in the proof of Theorem 2.

Lemma 3. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ be graphs with no isolated vertices such that $\min\{|\mathcal{V}|, |\mathcal{V}'|\} \geq 4$. Then, $\mathcal{G} \preceq \mathcal{G}'$ if and only if $D(\mathcal{G}) \leq D(\mathcal{G}')$.*

Proof. For notational convenience, let $\Pi^{D(\mathcal{G})} = \{\pi_e \mid e \in \mathcal{E}\}$ and $\Pi^{D(\mathcal{G}')} = \{\pi'_{e'} \mid e' \in \mathcal{E}'\}$. If $\mathcal{G} \preceq \mathcal{G}'$, then there is an injective map $\phi : \mathcal{V} \to \mathcal{V}'$ such that, for all $u, v \in \mathcal{V}$, $\{u, v\} \in \mathcal{E}$ implies $\{\phi(u), \phi(v)\} \in \mathcal{E}'$. That is, for all $e \in \mathcal{E}$, we have $\pi'_{\phi(e)} \in \Pi^{D(\mathcal{G}')}$. Construct a map $\alpha : \Pi^{D(\mathcal{G})} \to \Pi^{D(\mathcal{G}')}$ such that for all $e \in \mathcal{E}$, we set $\alpha(\pi_e) = \pi'_{\phi(e)}$. One can now easily see that for all $e \in \mathcal{E}$, we have $\pi_e = \pi'_{\phi(e)} \circ \phi$, and thus (ϕ, α) reduces $D(\mathcal{G})$ to $D(\mathcal{G}')$.

For the converse, assume that $D(\mathcal{G}) \leq D(\mathcal{G}')$, and let (ϕ, α) be a reduction of $D(\mathcal{G})$ to $D(\mathcal{G}')$. Since both graphs have at least four vertices, $D(\mathcal{G})$ and $D(\mathcal{G}')$ are both state-minimal by Lemma 2, and therefore the function ϕ is injective by Lemma 1. For all $e \in \mathcal{E}$, there is $e' \in \mathcal{E}'$ such that $\alpha(\pi_e) = \pi_{e'}$ and such that $\pi_e = \pi'_{e'} \circ \phi$. For all $e = \{v, w\}$, this means that $\phi(v) \not\equiv_{\pi'_{e'}} \phi(w)$, and that the remaining block of $\pi'_{e'}$ contains at least two elements. Thus, $e' = \{\phi(v), \phi(w)\}$, and since $e' \in \mathcal{E}'$, we have $\mathcal{G} \preceq \mathcal{G}'$. □

[5] A vertex $v \in \mathcal{V}$ is *isolated* if there exists no $e \in \mathcal{E}$ such that $v \in e$.

Given a graph $\mathcal{G}$ with at least four vertices, none of which is isolated, as well as an integer $k \geq 4$, in order to decide whether $\mathcal{G}$ contains a k-clique, one simply constructs the ASD's $D(\mathcal{K}_k)$ and $D(\mathcal{G})$, and checks whether $D(\mathcal{K}_k) \leq D(\mathcal{G})$. It is easy to see that the reduction is polynomial-time, and this implies $\mathcal{NP}$-completeness.[6] Lemma 3 also implies that $D(\mathcal{G}) \equiv D(\mathcal{G}')$ if and only if $\mathcal{G} \cong \mathcal{G}'$ for any two graphs $\mathcal{G}, \mathcal{G}'$ as in the statement of the lemma. Hence, deciding equivalence of ASD's is at least as difficult as deciding graph isomorphism, since deciding isomorphism is clearly not (computationally) easier when restricted to such graphs. This completes the proof of Theorem 2.

We conclude this section by noting that one can provide a simple two-round interactive proof for the problem of deciding non-equivalence of ASD's (cf. [7]). This means that deciding non-equivalence is in the complexity class $\mathcal{IP}(2)$, and hence also in $\mathcal{AM}$ [2]. For this reason, if the problem of deciding equivalence of ASD's were $\mathcal{NP}$-complete, we would have $\mathcal{NP} \subseteq \mathsf{co}\text{-}\mathcal{AM}$, and it is well-known [1] that this implies a collapse of the polynomial hierarchy $\mathcal{PH}$ to its second level. Therefore, it is very unlikely that deciding device equivalence is $\mathcal{NP}$-complete.

5 Binary ASD's and Unique Factorizations

We say that an ASD D has *direct product factorization* $\times_{i=1}^{m} D_i$ if this product is equivalent to D. Furthermore, an ASD D is *prime* if, whenever $D \equiv E \times E'$, then either E or E' is trivial. For example, if D is minimal with a partition $\pi \in \Pi^D$ such that $|\pi| = p$ for a prime number p, then D is prime. Clearly, every ASD D has a prime factorization with at most $\log |\mathcal{S}^D|$ factors.

In the following, we look at the class $\mathcal{D}_2^{\times}$ of ASD's having (at least one) prime factorization consisting uniquely of binary ASD's. Note that this class is closed under taking direct products. The following lemma provides a strong necessary and sufficient condition for deciding reducibility among members of the class $\mathcal{D}_2^{\times}$ with the same number of states, and such that no perfect factor appears in their binary factorization. The reader is referred to the full version for a proof.

Lemma 4. *Let $D_1, \ldots, D_m$, $D'_1, \ldots, D'_n$ be non-perfect state-minimal binary ASD's such that $\prod_{i=1}^{m} |\mathcal{S}^{D_i}| = \prod_{j=1}^{n} |\mathcal{S}^{D'_j}|$. Then $\times_{i=1}^{m} D_i \leq \times_{j=1}^{n} D'_j$ holds if and only if there exists a partition $\{J_1, \ldots, J_m\}$ of $\{1, \ldots, n\}$ such that $D_i \leq \times_{j \in J_i} D'_j$ for all $i \in \{1, \ldots, m\}$.*

As a corollary of this fact, for given linear devices $L_{k_1}, \ldots, L_{k_m}$, $L_{r_1}, \ldots, L_{r_n}$, where $\sum_{i=1}^{m} k_i = \sum_{j=1}^{n} r_j$, we have $\times_{i=1}^{m} L_{k_i} \leq \times_{j=1}^{n} L_{r_j}$ if and only if $m \leq n$ and there exists a partition $\{J_1, \ldots, J_m\}$ of $\{1, \ldots, n\}$ such that $k_i = \sum_{j \in J_i} r_j$. For instance, one can see that $L_3 \times L_3 \not\leq L_2 \times L_2 \times L_2$. Otherwise, the above would imply that $L_3 \leq L_2$, which is obviously false.

The following theorem makes use of Lemma 4 to show that the factorization in terms of *binary* ASD's in unique.

[6] Of course, the k-clique problem is still $\mathcal{NP}$-complete even when imposing $k \geq 4$ and when looking at graphs with no isolated vertices.

Theorem 3. *Let D be an ASD, and assume that $\times_{i=1}^{m} D_i$ is a factorization of D where $D_1, \ldots, D_m$ are* binary. *Then, this factorization is unique (with respect to the set of all factorizations into binary devices), up to order and equivalence of the factors.*

An immediate corollary of the theorem is the following.

Corollary 1. *Two products of binary linear devices are equivalent if and only if they consist of exactly the same devices.*

For instance, the corollary immediately yields $L_4 \times L_3 \times L_3 \not\equiv L_4 \times L_4 \times L_2$. Note that this non-equivalence could not be proved using simpler arguments based on order-preserving functions.

We stress that Theorem 3 does not rule out the fact that there might be additional factorizations in terms of non-binary prime ASD's. Indeed, the general question of deciding whether prime factorizations of ASD's are unique appears to be challenging. For instance, it is easy to see that every perfect ASD C_m where $m = \prod_{i=1}^{r} p_i^{\alpha_i}$ for distinct primes $p_1, \ldots, p_r$, and positive integers $\alpha_1, \ldots, \alpha_r$ can be uniquely factorized as $\times_{i=1}^{r} C_{p_i}^{\alpha_i}$. We leave the more general question as an open problem. Note that the problem is related to a line of research investigating unique factorizations of *relational structures* (cf. e.g. [5] for a survey). Even though ASD's are related to relational structures, known results only apply to a weaker form of direct product.

Acknowledgments. We would like to thank Thomas Holenstein and Renato Renner for helpful discussions.

References

1. R. B. Boppana, J. Håstad, and S. Zachos, "Does co-NP have short interactive proofs?," *Inf. Process. Lett.*, vol. 25, no. 2, pp. 127–132, 1987.
2. S. Goldwasser and M. Sipser, "Private coins versus public coins in interactive proof systems," in *STOC '86*, pp. 59–68, 1986.
3. G. Grätzer, *General Lattice Theory*. Birkhäuser, 1998.
4. W. Imrich and S. Klavžar, *Product Graphs: Structure and Recognition*. Wiley, 2000.
5. B. Jónsson, "The unique factorization problem for finite relational structures," *Colloq. Math.*, vol. 14, pp. 1–32, 1966.
6. R. König, U. Maurer, and R. Renner, "On the power of quantum memory," *IEEE Transactions on Information Theory*, vol. 51, no. 7, pp. 2391–2401, 2005.
7. R. König, U. Maurer, and S. Tessaro, "Abstract storage devices." Available at http://www.arxiv.org/abs/0706.2746, June 2007.
8. B. Köpf and D. Basin, "An information-theoretic model for adaptive side-channel attacks," in *ACM CCS 2007*, pp. 286–296, 2007.
9. M. O. Rabin, "How to exchange secrets with oblivious transfer." Technical Memo TR-81, Aiken Computation Laboratory, Harvard University, 1981.
10. C. E. Shannon, "The zero-error capacity of a noisy channel," *IEEE Transactions on Information Theory*, vol. 2, pp. 8–19, 1956.
11. S. Tessaro, "An abstract model for storage devices," Sept. 2005. Master Thesis, ETH Zurich.