Using Hypergraph Homomorphisms to Guess Three Secrets

Daniele Micciancio [*]
Department of Computer Science
University of California, San Diego
La Jolla, CA 92093
daniele@cs.ucsd.edu

Nathan Segerlind [†]
School of Mathematics
Institute for Advanced Study
Princeton, New Jersey 08540
nsegerli@math.ias.edu

April 16, 2004

Abstract

We present the first polynomial-time strategy for solving the problem of guessing three secrets as introduced by Chung, Graham and Leighton [6]. "Guessing three secrets" is a combinatorial search problem in which an adversary holds three n-bit secret strings, and a seeker attempts to learn as much as possible about these strings by asking the adversary Boolean questions about individual strings. Upon receiving a question, the adversary chooses one of the secrets and answers the question correctly for that string. We give both adaptive and oblivious solutions that run in time polynomial the length of the strings. For strings of length n, the adaptive solution makes $O(n)$ many queries and the oblivious solution makes $O(n^5)$ many queries. The queries of the oblivious strategy come from a kind of universal system that we call a *prefix universal family*. The recovery algorithm is based on properties of the equivalence classes of mutually-homomorphic three-uniform intersecting hypergraphs. We also show a reduction from certain instances of the problem of learning a hidden subgraph (as defined in Alon and Asodi [2]) to the guessing secrets problem.

[*] Supported by NSF Career Award CCR-0093029

[†] Supported by NSF Grant DMS-0303258. Work partially done during graduate study at UC San Diego Department of Computer Science while supported by NSF Grant DMS-9803515 and NSF Grant DMS-0100589.

1 Introduction

The problem of guessing k secrets is a game between two players, an adversary and a seeker. The adversary holds a set of k distinct n-bit strings which are hidden from the seeker. These strings are the secrets. The seeker wishes to find out as much as possible about these strings, and repeatedly asks the adversary yes-or-no questions about strings such as “Is the sum of the bits even?” or “Is 00 a prefix?”. For each question, the adversary chooses one of the k secrets, and correctly answers the question for that string. What makes this problem tricky is that when the seeker asks a question such that for at least one secret the answer is “yes” and for at least one secret the answer is “no”, the adversary may give either answer. This problem was introduced by Chung, Graham and Leighton [6] to model difficulties encountered while routing internet traffic. The problem also has connections with the algorithmic study of separating systems [6, 3, 7] and the problem of learning a hidden subgraph (appendix 9 of this paper). An informal discussion of the guessing secrets problem and its connections to other topics can be found in [17].

The adversary is accorded a great deal of freedom in the guessing secrets game, and the first question that must be resolved is “What can the seeker possibly learn?”. In [6] it was shown that regardless of computational resources, the most that the seeker can possibly learn is an intersecting collection of k-sets of strings that includes the secret k-set, and that the seeker has a strategy that makes $O(n)$ queries and runs in space $2^{O(kn)}$ that finds an intersecting family containing the secret set. As in that paper, when we say that a strategy *solves the problem of guessing k secrets*, we mean that when a seeker uses that strategy in a guessing secrets game with any adversary, the seeker is guaranteed to learn an intersecting family of k-sets of strings that contains the secret k-set.

Further work on the guessing secrets problem has concentrated on efficiently recovering the intersecting family [3, 7, 12]. Previously, polynomial time strategies for guessing k secrets had been known only for $k \leq 2$. We present the first polynomial time algorithms for guessing three secrets. We first present a strategy that is *adaptive*, in which the seeker asks questions that depend on the previous answers of the adversary. Later, the adaptive strategy is made *oblivious*, so that the seeker asks all questions before the adversary provides any answers.

A Comparison with Previous Work on Guessing Secrets

To the best of our knowledge, our solution is the first that solves the problem of guessing three secrets in polynomial time. Solutions that solve the guessing two secrets problem in polynomial time have appeared in [7, 3, 12]. The solution of Chung, Graham, and Lu uses $O(n^3)$ queries, where each query is a dot-product with a vector of Hamming weight three from GF_2^n [7]. The solution of Alon, Kaufmann, Guruswami, and Sudan makes $O(n)$ queries, where each query is the dot-product of the string with a row of the generating matrix for an ϵ-biased, list-decodable code. In the same paper, Alon, Kaufmann, Guruswami, and Sudan give partial solution given that finds a set of size $poly(k)$ that is guaranteed to contain at least one of the secret strings [3]. A quantum solution of the guessing two secrets problem appeared in [12].

Any strategy that makes only linear queries cannot solve the guessing three secrets problem, cf. [7]. The algorithms used for recovering the hypergraph from the answers in [6, 3] make extensive use of the linearity of the queries. Our solution makes nonlinear queries by necessity, and as a result, the process of recovering the intersecting hypergraph from the answers is very different from the processes used previously.

Our Techniques and their Connections to Other Work

There are two fundamental issues in solving the guessing k secrets problem. The first is finding a family of queries that is rich enough that the adversary’s answers must uniquely specify an intersecting family of k-sets, and the second is devising a method to reconstruct this intersecting hypergraph from those answers.

Families of queries that can solve the guessing k secrets problem are exactly the (k, k)-separating systems [6, 3]. A (j, l) separating system is a family of mappings $F_i : \{0, 1\}^n \to \{0, 1\}$ so that for all distinct strings, $\alpha_1, \ldots, \alpha_j$ and $\beta_1, \ldots, \beta_l$, there exists i so that $F_i(\alpha_1) = \cdots = F_i(\alpha_j) \neq F_i(\beta_1) = \cdots = F_i(\beta_l)$. There are well-known constructions of (k, k)-separating systems of size $c_k n$ where c_k is a constant exponential in k, cf. [1], but efficiently recovering the intersecting family of k-sets seems to require that the separating system have further structure that can be algorithmically exploited. For example, the solutions of [7, 3] depend on the linearity of certain $(2, 2)$ separating systems. Our oblivious strategy uses a stronger (and larger) kind of separating system that separates all strings with given prefixes: a *k-prefix universal family*. This is a collection of functions $\{F_i \mid i \in I\}$ from $\{0, 1\}^n$ to $\{0, 1\}$ so that for each collection of k strings $\alpha_1, \ldots, \alpha_k \in \{0, 1\}^{\leq n}$, so that for all $i \neq j$ α_i is not a prefix of α_j, and each $\epsilon_1, \ldots, \epsilon_k \in \{0, 1\}$, there is a function F_i so that for all $j \in [k]$, all $x \in \{0, 1\}^n$, if $\alpha_j \preceq x$ then $F_i(x) = \epsilon_j$. We give a simple construction of a k-prefix universal family of size $2^{O(k)}n^{k-1}$.

Hypergraph homomorphisms are at the heart of our algorithm for recovering the intersecting hypergraph from the adversary's answers. The algorithm maintains a three-uniform hypergraph with vertices from $\{0,1\}^n$ that contains each triple that is consistent with our knowledge so far about the secret triple. A three-uniform hypergraph over $\{0,1\}^n$ can have size as large as $2^{O(n)}$). To save space, we represent hypergraphs G over $\{0,1\}^n$ as a pair (H, λ) where H is a constant size hypergraph, λ is a succinctly defined partial-mapping from $\{0,1\}^n$ to $V(H)$, and $G = \lambda^{-1}(H)$ (almost, see section 3). This is similar to the use of graph homomorphisms for knowledge representation [4, 5, 11].

The combinatorial properties of hypergraph homomorphisms are used to prove termination for our algorithm. Central to this is the preorder on hypergraphs given by the relation "G homomorphically embeds into H". For unrestricted classes of graphs and hypergraphs, the homomorphism preorder is virtually without structure (or so full of structure as to be nearly useless). It is universal – any countable partial order can be embedded into the partial order on equivalence classes of graphs. The proof of this universality result uses somewhat unnatural extremal graphs, and an active area of research is understanding the homomorphism poset for restricted classes of graphs (such as directed paths [16], vertex transitive graphs [19], planar graphs or bounded degree graphs [15]). We prove a simple result along these lines and use this to guarantee the termination of our algorithm: for three-uniform, intersecting hypergraphs, there only finitely many equivalence classes that contain a hypergraph without a central vertex (Theorem 6). For an introduction to graph homomorphisms and this preorder we recommend the survey by Hahn and Tardif [8] and those by Nešetřil [14, 15].

Outline of the Paper

Section 2 contains basic notation and definitions, as well as some simple facts about hypergraphs and their homomorphisms. Section 3 develops the principal data structure used by the recovery algorithm: a succinct representation of hypergraphs over the vertex set $\{0,1\}^n$ by labeling a small hypergraph with sets of strings.

Section 4 introduces the basic sub-routines of the seeker's strategy for recovering the secrets: splitting the nodes of a labeled hypergraph and querying the adversary to delete non-intersecting edges. The algorithm for recovering the hypergraph repeats these two steps, and it turns out that this creates a sequence of three-uniform, intersecting hypergraphs with each homomorphically embedding into the previous.

The pre-order on three-uniform, intersecting hypergraphs is studied in section 5. This will provide a measure of progress for the routine and later be used to guarantee termination. The central result here is that there are only finitely many homomorphism classes for three-uniform intersecting hypergraphs without central vertices (Theorem 4).

The details of the adaptive strategy are given in section 6. At the heart of the strategy is a routine that takes a labeled hypergraph G and returns a labeled hypergraph H that contains the secret triple of strings and either represents an intersecting hypergraph over the vertex set $\{0,1\}^n$ or strictly precedes G in the partial order given by the homomorphism relation (as in section 5). By the results of section 5, this descent can happen only a constant number of times. Therefore, a representation of an intersecting hypergraph over $\{0,1\}^n$ containing the secret will eventually be found.

In section 7 we show how to construct prefix universal families and how to use them to convert the adaptive strategy into an oblivious one. Proofs of combinatorial lemmas used in section 6 are presented in section 8. We discuss the connection with the learning a hidden subgraph problem in section 9. Finally, we discuss some open issues in section 10 and remark on the subsequent work on the guessing secrets problem by Alexander Razborov [18].

2 Background

Throughout this paper we will treat the length of the strings and the secret triple held by the adversary as fixed: all binary strings as having length $\leq n$ and the secret triple of strings held by the adversary will be written as S^*.

Hypergraphs and their homomorphisms A *(finite) hypergraph* is an ordered pair (V, E) where V is a finite set of vertices, and $E \subseteq \mathbb{N}^V$ is a collection of multisets over V called *edges*. The set of vertices of G is denoted $V(G)$ and the set of edges of G is denoted $E(G)$. A hypergraph is called *k-uniform* if all of its edges have size k. A hypergraph G is *intersecting* if $\forall e, f \in E(G)$, $e \cap f \neq \emptyset$. A vertex v is a *central vertex of* G if for every $e \in E(G)$, $v \in e$. For each $v \in V(G)$, and each $e \in E(G)$, the *multiplicity of* v *in* e, $m_e(v)$, is the number of times that v occurs in e; the *maximum multiplicity of* v *in* G, $m_G(v)$, is defined to be $\max_{e \in E(G)} m_e(v)$. A hypergraph is *simple* if all of its vertices have maximum-multiplicity at most one; equivalently, if every edge is a set.

Let H and G be two hypergraphs. We say that H *is a subgraph of* G *(written* $H \subseteq G$*)* if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. Let H be a hypergraph and let V be a subset of $V(H)$. The *sub-hypergraph of* H *induced by* V, is defined to b $H[V] = (V, \{e \in E(H) \mid e \subseteq V\})$. For hypergraph G and function ϕ with domain $V(G)$, define the hypergraph $\phi(G) = (\phi(V(G)), \phi(E(G)))$. Similarly, if ϕ is a function with range $V(G)$, define the hypergraph $\phi^{-1}(G) = (\phi^{-1}(V(G)), \phi^{-1}(E(G)))$. A function $\phi\colon V(G) \to V(H)$ is a homomorphism if $\phi(G) \subseteq H$. We say that ϕ is an *isomorphism* if ϕ is a bijection and $\phi(G_1) = G_2$. An *endomorphism* is a homomorphism from a hypergraph to itself. An *automorphism* is an isomorphism from a hypergraph to itself. Two graphs G_1 and G_2 are *isomorphic* (written $G_1 \cong G_2$) if there is an isomorphism from G_1 to G_2.

Proposition 1 *Let G be a hypergraph, let $V = V(G)$ and let W be a set and let $\phi\colon V \to W$ and $\psi\colon W \to V$ be arbitrary functions. Then $\phi\colon G \to \phi(G)$ and $\psi\colon \psi^{-1}(G) \to G$ are both homomorphisms and $\psi(\psi^{-1}(G)) \subseteq G \subseteq \phi^{-1}(\phi(G))$. Moreover, if G is k-uniform then $\phi(G)$ and $\psi^{-1}(G)$ are both k-uniform. If G is intersecting then $\phi(G)$ is intersecting.*

Proposition 2 *If $\psi \circ \phi\colon H \to G$ is a homomorphism, then $\psi\colon \phi(H) \to G$ is a homomorphism.*

3 Succinct Representation of Hypergraphs Over $\{0,1\}^n$

Our algorithm represents hypergraphs over $\{0,1\}^n$ as homomorphic preimages of constant-size hypergraphs. Hypergraphs over $\{0,1\}^n$ are called *ground* hypergraphs. A ground hypergraph G is represented by a pair (H, λ_H) where H is a constant-size, three-uniform hypergraph and λ_H is a partial function from $\{0,1\}^n \to V$ so that G is the largest simple hypergraph with $G \subseteq \lambda_H^{-1}(H)$ [1]. The partial map λ_H is represented by labeling each vertex $v \in V(H)$ by a succinct representation of $\lambda_H^{-1}(v)$.

Definition 3.1 *Let $S \subseteq \{0,1\}^n$. Let $L \subseteq \{0,1\}^{\leq n}$. We say that L* represents *S if $\forall s \in S\ \exists x \in L,\ x \preceq s$, and $\forall x \in L,\ \forall y \in \{0,1\}^n,\ x \preceq y \Rightarrow y \in S$.*

Lemma 1 *For all $S \subseteq \{0,1\}^n$ there is a unique set L of minimum cardinality that represents S.*

Proof: Let $S \subseteq \{0,1\}^n$ be given. For any string $x \in \{0,1\}^{\leq n}$, let the *continuations of* x, $Cont(x)$, be the set of all $s \in \{0,1\}^n$ such that x is a prefix of s. Set $L = \{x \in \{0,1\}^{\leq n} \mid Cont(x) \subseteq S,\ \forall y \prec x, Cont(y) \not\subseteq S\}$. Let $L' \subseteq \{0,1\}^{\leq n}$ representing S be given. Let $y \in L$ be given. Choose $s \in Cont(y) \subseteq S$, Because L' represents S, we may choose $x \in L'$ so that $x \preceq s$. Therefore, $x \preceq y$ or $y \preceq x$. Because $Cont(x) \subseteq S$ and by the minimality of elements of L, $x \not\prec y$, so $y \preceq x$. This associates each $y \in L$ with $x \in L'$ so that $y \preceq x$; therefore $|L| \leq |L'|$. ■

Definition 3.2 *For each $S \subseteq \{0,1\}^n$, let $\mathsf{label}(S) = L$ where L is the unique minimum-size set of prefixes that represents S.*

In a labeled hypergraph (H, λ_H), the partial function $\lambda\colon \{0,1\}^n \to V$, is represented by associating $\mathsf{label}(\lambda^{-1}(v))$ with each $v \in V(H)$. The size of a label is measured by the *rank function*, defined for any $\{x_1, \ldots, x_k\} \subseteq \{0,1\}^{\leq n}$:

$$\text{rank}(\{x_1, \ldots, x_k\}) = k + \max_i (n - |x_i|).$$

The rank of a nonempty set $S \subseteq \{0,1\}^n$ is defined in the obvious way: $\text{rank}(S) = \text{rank}(\mathsf{label}(S))$. Define $\text{rank}(\emptyset) = 0$. The rank function is subadditive; for all S_1, S_2, $\text{rank}(S_1 \cup S_2) \leq \text{rank}(S_1) + \text{rank}(S_2)$. For any set $S \subseteq \{0,1\}^n$, by consecutively listing the strings in $\mathsf{label}(S)$, S can be represented using $(n+1) \cdot \text{rank}(S)$ bits. Also, for any non-empty S, $\text{rank}(S) \geq 1$, and $\text{rank}(S) = 1$ if and only if $|S| = 1$.

Lemma 2 *Let (H, λ_H) be a labeled hypergraph such that H is intersecting and λ_H is onto. Then, $\lambda_H^{-1}(H)$ is intersecting if and only if for every $e, f \in E(H)$, there exists $v \in e \cap f$ such that rank$(\lambda_H^{-1}(v)) = 1$.*

[1] We use this definition rather than $\lambda_H^{-1}(H)$ because we are searching for a secret set of three strings and $\lambda_H^{-1}(H)$ might not be simple. If we allow for the case when the adversary has fewer than three distinct strings, a routine for guessing two secrets can precede this routine.

Proof: Recall that for any $S \subseteq \{0,1\}^n$, rank$(S) = 1$ if and only if $|S| = 1$.

Suppose that $\forall e, f \in E(H)$, there is $v \in e \cap f$ so that $|\lambda_H^{-1}(v)| = 1$. Let $E, F \in \lambda_H^{-1}(H)$ be given, and set $e = \lambda_H(E)$, $f = \lambda_H(F)$. Choose $v \in e \cap f$ with $|\lambda_H^{-1}(v)| = 1$. Let S be the unique element of $\lambda_H^{-1}(v)$ and observe that $S \in E \cap F$.

Suppose that there are $e, f \in E(H)$ with $\forall v \in e \cap f$, $|\lambda_H^{-1}(v)| \geq 2$. For each $v \in \cap f$, choose $s_v, t_v \in \lambda_H^{-1}(v)$ so that $t_v \neq s_v$. Choose any $E, F \in \{0,1\}^n$ so that $\lambda_H(E) = e$ and $\lambda_H(F) = e$. Let $E' = (E \setminus \lambda_H^{-1}(e \cap f)) \cup \{s_v \mid v \in e \cap f\}$ and $F' = (F \setminus \lambda_H^{-1}(e \cap f)) \cup \{t_v \mid v \in e \cap f\}$. Clearly, $\lambda_H(E') = e$ and $\lambda_H(F') = f$, yet $E' \cap F' = \emptyset$. ■

Definition 3.3 *Let (G, λ_G) and (H, λ_H) be to labeled hypergraphs. We say that (G, λ_G) is contained in (H, λ_H) (written $(G, \lambda_G) \subseteq (H, \lambda_H)$) if $G \subseteq H$, and for every $v \in V(G)$, $\lambda_G^{-1}(v) \subseteq \lambda_H^{-1}(v)$. For any function ϕ with domain $V(G)$, define the image of (G, λ_G) under ϕ as the labeled hypergraph $(\phi(G), \phi \circ \lambda_G)$. We say that a function ϕ: $V(G) \to V(H)$ is a homomorphism from (G, λ_G) to (H, λ_H), if $\phi(G, \lambda_G) \subseteq (H, \lambda_H)$.*

Definition 3.4 *We say that (H, λ_H) is* valid *if H is a three-uniform, intersecting hypergraph, $S^* \in \lambda^{-1}(H)$, λ_H is onto, and for each $v \in V(H)$, $|\lambda_H^{-1}(v)| \geq m_H(v)$.*

Proposition 3 *Let (G, λ_G) be valid, and let $\phi : V(G) \to W$. The hypergraph $\phi(G, \lambda_G)$ is also valid.*

Proposition 4 *There is a polynomial-time algorithm that on input (G, λ_G) and ϕ outputs $\phi(G, \lambda_G)$.*

Throughout the routine we split labels into two pieces and remerge them. It is easy to see that there efficient routines that do this.

Proposition 5 *There exists a polynomial-time algorithm that for any $S_1, S_2 \subseteq \{0,1\}^n$, on input* label$(S_1)$ *and* label(S_2)*, outputs* label$(S_1 \cup S_2)$*. There exists a polynomial-time algorithm Split that on input* label(S)*, where $S \subseteq \{0,1\}^n$ is of size > 1, outputs* $($label$(S_1),$ label$(S_2))$ *so that (S_1, S_2) is a a partition of (S_1, S_2) of S and rank$(S_1) \leq$ rank$(S_2) <$ rank(S).*

4 Splitting and Querying a Hypergraph

The algorithm maintains a valid labeled hypergraph (G, λ_G). At any time when $\lambda_G^{-1}(G)$ is not intersecting, by lemma 2, there exist $e, f \in E(G)$ so that for all $v \in e \cap f$, rank$(\lambda_G^{-1}(v)) > 1$. The algorithm splits each vertex v with rank$(\lambda_G^{-1}(v)) > 1$ into two nodes, v_0 and v_1, with each node receiving roughly one-half of the strings in $\lambda_G^{-1}(v)$. Call this new labeled hypergraph (H, λ_H); it is easy to see that H is three-uniform and $\lambda_H^{-1}(H)$ contains the secret triple, but H is not intersecting. To obtain a valid labeled hypergraph, the seeker repeatedly queries the adversary to delete non-intersecting edges.

Definition 4.1 *For any set V and subset $A \subseteq V$, let Split$_A(V)$ be the set $(V \setminus A) \cup \{v_i : v \in A, i = 0, 1\}$, where v_0, v_1 are assumed to be new nodes different from all elements of V. For any hypergraph G and set $A \subseteq V(G)$, let Split$_A(G)$ be the hypergraph $\pi^{-1}(G)$ where π_A is the function defined by $\pi_A(v_0) = \pi_A(v_1) = v$ for all $v \in A$, and $\pi_A(v) = v$ for all $v \in V \setminus A$.*

For any labeled hypergraph (G, λ_G) and set $A \subseteq V(G)$ such that rank$(\lambda^{-1}(a)) > 1$ for all $a \in A$, let Split$_A(G, \lambda_G)$ be the labeled hypergraph (H, λ_H) where $\lambda_G^{-1}(v) = \lambda_H^{-1}(v)$ for all $v \in V \backslash A$, and $(\lambda_H^{-1}(v_0), \lambda_H^{-1}(v_1)) =$ Split$(\lambda_G^{-1}(v))$ for all $v \in A$, and H is the following sub-hypergraph of Split$_A(G)$: $E(H) = \{e \in$ Split$_A(H) \mid \forall v \in e,\ m_e(v) \leq |\lambda_H^{-1}(v)|\}$, $V(H) = \bigcup E(H)$.

Proposition 6 *For any valid hypergraph (G, λ_G) and $A \subseteq V(G)$ so that $\forall a \in A$, rank$(\lambda_G^{-1}(a)) > 1$, the labeled hypergraph $(H, \lambda_H) =$ Split$_A(G, \lambda_G)$ can be computed in polynomial-time, and it satisfies the following properties: (i) $\lambda_H^{-1}(H) = \lambda_G^{-1}(G)$, (ii) π_A: $(H, \lambda_H) \to (G, \lambda_G)$ is a homomorphism, (iii) For every $v \in \pi_A^{-1}(A)$, rank$(\lambda_H^{-1}(v)) <$ rank$(\lambda_G^{-1}(\pi_A(v)))$ (iv) For every $v \in V \setminus A$, rank$(\lambda_H^{-1}(v)) =$ rank$(\lambda_G^{-1}(\pi_A(v)))$ (v) H is three-uniform (vi) λ_H is onto (vii) for each $e \in E(H)$, each $v \in e$, $m_e(v) \leq |\lambda_H^{-1}(v)|$.*

Definition 4.2 *The routine* Query *takes as input (H, λ_H) and outputs the following subgraph of (H, λ_H) in which edges are deleted in the following manner: For for every edge $e \in E(H)$, make the query "Does the secret have a prefix belonging to $\bigcup_{v \in e} \mathsf{label}(\lambda_H)(v)$?". If the answer to the query is no, then remove edge e from H. Otherwise, delete all edges disjoint with e. If this process creates any vertices that are not contained in an edge, those vertices are removed from the vertex set, and the strings mapping to those vertices are mapped removed from the domain of λ_H.*

Notice that the procedure Query is *adaptive* because the queries made depends on the sets $\lambda_H^{-1}(v)$, $v \in V(H)$. It makes a total of $|E(H)|$ queries, and each query can be represented by a list of prefixes of bit-size $O(nr)$ where r is maximum rank of a set $\lambda_H^{-1}(v)$. The queries are made oblivious in section 7.

Proposition 7 *For any labeled hypergraph (G, λ_G), the labeled hypergraph $(H, \lambda_H) = \mathsf{Query}(G, \lambda_G)$ can be computed in polynomial-time, and it satisfies the following properties: (i)$(H, \lambda_H) \subseteq (G, \lambda_G)$ (ii)H is intersecting (iii) If (G, λ_G) contains the secret then (H, λ_H) contains the secret. (iv) If λ_G is onto then λ_H is onto*

Proposition 8 *Let (G, λ_G) be a valid labeled hypergraph and let $A \subseteq V(G)$ be such that $\forall a \in A$, rank$(\lambda_G^{-1}(a)) > 1$. Let $(H, \lambda_H) = \mathsf{Query}(Split_A(G, \lambda_G))$. The hypergraph (H, λ_H) is valid, and the mapping π_A is a homomorphism from (H, λ_H) to (G, λ_G). Furthermore, for every $v \in \pi_A^{-1}(A)$, rank$(\lambda_H^{-1}(v)) <$ rank$(\lambda_G^{-1}(\pi_A(v)))$, and for every $v \in V \setminus A$, rank$(\lambda_H^{-1}(v)) =$ rank$(\lambda_G^{-1}(\pi_A(v)))$*

5 The hypergraph poset and hypergraph cores

At the end of the last section we saw that the process of splitting nodes and querying the adversary to delete disjoint edges creates a sequence of hypergraphs, each homomorphically mapping into the previous. The homomorphism relation induces a partial order, and we use this partial order to guarantee the termination of the algorithm.

We say that hypergraph G embeds in hypergraph H (written $G \leq H$) if there is a homomorphism $\phi\colon G \to H$. It is easy to verify that the embedding relation is is reflexive and transitive, i.e. it is a preorder. Let $\sim$ be the equivalence relation associated to this preorder: $G \sim H$ if the two hypergraphs are *mutually homomorphic*, i.e., there exists homomorphisms $\phi\colon G \to H$ and $\psi\colon H \to G$. Notice that mutually homomorphic hypergraphs $G_1 \sim G_2$ are not necessarily isomorphic. The equivalence class of hypergraph G, i.e., the set of hypergraphs mutually homomorphic to G, is denoted $\langle G \rangle$.

Definition 5.1 *We say that a hypergraph G is a* core *if every endomorphism of G is surjective. When G and H are hypergraphs, we say that G* is a core of *H if $G \subseteq H$, G is a core and $G \sim H$.*

The following facts about cores are used extensively in this paper. These statements hold because the hypergraphs we consider are finite; they fail for infinite hypergraphs.

Proposition 9 *If G is a core, then every homomorphism from G to G is an automorphism of G. If G_1 and G_2 are mutually homomorphic cores, then G_1 and G_2 are isomorphic. Every hypergraph H has a (not necessarily unique) vertex-induced subgraph $G = H[V]$ such that G is a core of H. All cores of a hypergraph are isomorphic to one another. Cores are unique (up to isomorphism) representatives for the equivalence classes of mutually homomorphic hypergraphs. The relation $\leq$ is a partial order on (the isomorphism types of) hypergraph cores.*

Lemma 3 *Let H be a hypergraph with core G. For any homomorphism $\phi : H \to G$, there exists a homomorphism $\psi : G \to H$ such that $\phi \circ \psi = Id_G$. For any homomorphism $\psi : G \to H$, there exists a homomorphism $\phi : H \to G$ such that $\phi \circ \psi = Id_G$. For any homomorphism $\psi : G \to H$ and $\phi : H \to G$ there is a homomorphism $\eta : G \to H$ with $\mathrm{Rng}\eta = \mathrm{Rng}\psi$ and $\phi \circ \eta = Id_G$.*

Proof:

1. $\phi \restriction_G$ is an endomorphism of G, and because G is a core, it is an automorphism of G. Let $\psi = (\phi \restriction_G)^{-1}$.

2. Because G is a core of H, we may choose $\theta : H \to G$. $\theta \circ \psi$ is a homomorphism from G to G, and because G is a core, $\theta \circ \psi$ is an automorphism of G. Let $\phi = (\theta \circ \psi)^{-1} \circ \theta$. Clearly $\phi \circ \psi = (\theta \circ \psi)^{-1} \circ \theta \circ \psi = \mathrm{Id}_G$.

3. Because $\phi \circ \psi$ is an endomorphism of G and G is a core, $\phi \circ \psi$ is invertible. Let $\eta = \psi \circ (\phi \circ \psi)^{-1}$. Clearly, $\phi \circ \eta = \phi \circ \psi \circ (\phi \circ \psi)^{-1} = \mathrm{Id}_G$.

■

At times, the secret recovery algorithm must compute the core of a hypergraph. In general this is a very hard problem, as recognizing the core of a graph is co-NP complete [9]. Because we work with labelings of constant-size hypergraphs, it is acceptable to compute the cores in a brute-force manner. This can be done in time $O(m^m)$ where m is the number of vertices. Let Core denote a routine that takes a labeled hypergraph (H, λ_H), computes a core G of H and $\eta : H \to G$ and returns $\eta(H, \lambda_H)$. Notice that if (H, λ_H) is valid then $\mathsf{Core}(H, \lambda_H)$ is valid.

We use the homomorphism poset to provide a measure of progress for the algorithm and guarantee termination. The basic idea is to bound the number of times the hypergraph can descend in the partial order. Unfortunately, there are infinite descending chains of three-uniform, intersecting hypergraphs (appendix Lemma 38). What we do is distinguish two kinds of three-uniform, intersecting hypergraphs. The the first class contains those three-uniform, intersecting hypergraphs that are mutually homomorphic with a three-uniform intersecting hypergraph without a unique central vertex. We show that there are only have only finitely many homomorphism-equivalence classes for these hypergraphs. The other class of three-uniform, intersecting hypergraphs are those H such that for every three-uniform, intersecting $H_0 \sim H$, H_0 has a unique central vertex. It turns out that this class of hypergraphs is easy for our algorithm to handle.

Definition 5.2 *Let $\mathcal{C}$ be the set of all equivalence classes $\langle H \rangle$ of three-uniform, intersecting hypergraphs H that do not have a unique central vertex, ie. $|\bigcap E(H)| \neq 1$.*

Theorem 4 *Up to isomorphism, there are only finitely many three-uniform, intersecting cores that are mutually homomorphic with a three-uniform, intersecting hypergraph without a central vertex. In other words, the class $\mathcal{C}$ is finite.*

A remark on Theorem 4: Maximal three-uniform intersecting hypergraphs without central vertices have an easy characterization into a small set of classes of hypergraphs, cf. [6]. However, this does not immediately imply Theorem 4 because distinct non-isomorphic cores can belong to the same equivalence class in this characterization.

The only properties of $\mathcal{C}$ used by our algorithm are that $\mathcal{C}$ is finite, and that if H is a three-uniform intersecting hypergraph without a unique central vertex, then its core belongs to $\mathcal{C}$. In particular, an implementation of our algorithm would not require the set $\mathcal{C}$ to be identified and enumerated.

With a short sequence of lemmas, we prove Theorem 4 and show that the set $\mathcal{C}$ is finite. First we need some elementary definitions.

Covering Numbers Let G be a hypergraph. A *covering set* or *hitting set* is a set $S \subseteq V(G)$ so that $\forall e \in E(G)$, $e \cap S \neq \emptyset$. The *covering number of G, $\tau(G)$,* is the minimum size of a covering set for G. A set S is said to be a *strong-cover of G* if it is a cover of G and for every $e \in E$, $|e \setminus S| \leq 1$. The *strong-covering number of G, $\tau^*(G)$,* is the cardinality of the smallest strong-cover of G.

Lemma 5 *If G is a 3-uniform, intersecting hypergraph without a unique central vertex, then $\tau^*(G) \leq 6$.*

Lemma 5 is implicit in the characterization of maximal, three-uniform, intersecting hypergraphs given without proof in [6]. For completeness, we offer the following simple argument:

Proof: Let $V = V(G)$ and $E = E(G)$. If $|\bigcap E| > 1$, then clearly $[\bigcap E]$ is a strong cover of size at most 3. Now, assume that G has no central vertex, and that $\tau^*(G) > 5$. Choose $e_1 \in E$. Because e_1 is not a strong-cover of G, we may find $e_2 \in E$ so that $|e_2 \setminus e_1| \geq 2$. Since e_1 and e_2 intersect, it must be $|e_1 \cap e_2| = 1$. Let $\{v\} = [e_1 \cap e_2]$. Since G has no central vertices, there exists an edge $e_3 \in E$ so that $v \notin e_3$. Because e_3 intersects both e_1 and e_2 without containing v, it must be that $|e_1 \cup e_2 \cup e_3| \leq 6$. We will now show that $S = [e_1 \cup e_2 \cup e_3]$ is a strong cover of G. Note that $v \in S$. Let $f \in E$ be given. If f contains v, because f meets e_3, then $|f \cap S| \geq 2$ and we are done. If $v \notin f$, then because f intersects both e_1 and e_2 we must have that $|f \cap S| \geq 2$. ■

The bound of lemma 5 is optimal, because the following 3-uniform intersecting hypergraph has no central vertex and it has strong cover number 6:

$$\{1,2,3\},\{1,4,5\},\{1,6,7\},\{2,4,6\},\{3,5,6\},\{2,5,7\},\{3,4,7\}$$

Definition 5.3 *Let G be a hypergraph, and let $v \in V(G)$ be given. The type of v in G is the set $tp_G(v) = \{e \setminus \{v\} \colon v \in e, e \in E(G)\}$.*

Proposition 10 *Let G be a uniform hypergraph. For any two $v, w \in V(G)$, the substitution function $[w \to v]$ (where $[w \to v](v) = w$ and $[w \to v](z) = z$ for all $z \neq v$) is an endomorphism of G if and only if $tp_G(v) \subseteq tp_G(w)$. Therefore, if G is a uniform core then $\{tp_G(v) \mid v \in V(G)\}$ is an antichain with respect to inclusion.*

Theorem 6 *There exists a constant c so that every three-uniform intersecting hypergraph without a unique central vertex has a core of size at most c.*

Proof: Let H be a 3-uniform, intersecting hypergraph with no unique central vertex, and let G be a core of H. Because G is three-bounded and intersecting, we may choose a strong-cover of G, S, so that $|S| \leq 6$. For $v \in V(G) \setminus S$, let $\mathrm{tp}(v) = \{e \setminus \{v\} \mid e \in E(G),\ v \in e\}$. By proposition 10, for each $u, v \in V(G) \setminus S$, if $\mathrm{tp}(u) \subseteq \mathrm{tp}(v)$ then there is a endomorphism ϕ of G so that $\phi(u) = v$. Therefore, $\{\mathrm{tp}(u) \mid u \in V(G) \setminus S\}$ is an antichain. On the other hand, for all $u \in V(G) \setminus S$, $\mathrm{tp}(u) \subseteq [S]^{\leq 2}$ and the set $[S]^{\leq 2}$ has size at most $\binom{6}{1} + \binom{6}{2} = 21$. So by Sperner's lemma, $|V(G) \setminus S| \leq \binom{21}{10}$ Therefore, the number of vertices in G is at most $6 + \binom{21}{10}$. ■

The actual maximum number of vertices for a three-uniform, intersecting hypergraph core without a unique central vertex is 7. Even though an improvement from $\binom{21}{10}$ to 1 is dramatic, the proof is just a more careful analysis with the same techniques as in the proof of theorem 6; we omit the more detailed proof for space reasons. Regardless, because there is a constant bound on the size of the core for a three-uniform, intersecting hypergraph without a unique central vertex, there are only finitely many equivalence classes for such hypergraphs.

Lemma 7 *Let G be a three-uniform, intersecting hypergraph. If G has two distinct central vertices, then the core of G is isomorphic to one of $\{abc\}$, $\{abb\}$ or $\{abb, aab\}$.*

Proof: We give a homomorphism from G to a subgraph of the form $\{abc\}$, $\{abb\}$ or $\{abb, aab\}$. Let u, v be two distinct central vertices of G. We distinguish two cases:

- If G is not simple, then it contains either $\{uuv\}$ or $\{uvv\}$ (or both). Assume without loss of generality that $\{uuv\}$ is in $E(G)$. The function $\phi \colon V(G) \to \{u, v\}$, defined by $\phi(v) = v$ and $\phi(x) = u$ for $x \neq v$, is a homomorphism from G to its core $G[\{u, v\}] = \{uuv\}$.
- If G is simple, then pick and edge $uvw \in E(G)$, The function $\phi \colon V(G) \to \{u, v, w\}$ defined by $\phi(u) = u$, $\phi(v) = v$ and $\phi(x) = w$ for all $x \in V(G) \setminus \{u, v\}$, is a homomorphism from G to its core $G[\{u, v, w\}] = \{uvw\}$.

■

6 The Adaptive Strategy

The adaptive strategy for guessing three secrets is based on two routines: the routine Reduce, which takes as input a valid labeled core(G, λ_G) and outputs a valid labeled hypergraph (H, λ_H) so that $(H, \lambda_H) \leq (G, \lambda_G)$ and either $\lambda_H^{-1}(H)$ is intersecting or $H < G$, and the routine Central, which takes as input (G, λ_G), a valid labeled core such that G has a unique central vertex and outputs a valid labeled hypergraph $(H, \lambda_H) \leq (G, \lambda_G)$ such that either $\lambda_H^{-1}(H)$ is intersecting or $\langle H \rangle \in \mathcal{C}$.

The strategy for guessing three secrets proceeds as follows:

1. Set $(G, \lambda_G) = (\{aaa\}, f_a)$, where f_a is the constant map from $\{0,1\}^n$ to a, and repeat the following steps until an intersecting $\lambda_H^{-1}(H)$ is found:

   (a) If G has a unique central vertex, then let $(H, \lambda_H) = \mathsf{Central}(G, \lambda_G)$. If $\lambda_H^{-1}(H)$ is intersecting, then exit. Otherwise, set $(G, \lambda_G) = \mathsf{Core}(H, \lambda_H)$ and proceed to step 1b.

   (b) Let $(H, \lambda_H) = \mathsf{Reduce}(G, \lambda_G)$. If $\lambda_H^{-1}(H)$ is intersecting, then exit. Otherwise, set $(G, \lambda_G) = \mathsf{Core}(H, \lambda_H)$ and continue the main loop.

The correctness of the algorithm upon termination is clear: Because (G, λ_G) contains the secret, and Reduce, Central and Core all return hypergraphs that contain the secret. Therefore, upon termination, the hypergraph $\lambda_H^{-1}(H)$ contains the secret. Moreover, the algorithm terminates only when $\lambda_H^{-1}(H)$ is intersecting.

We claim that the main loop terminates after a constant number of iterations. Consider all iterations except the last one. Notice that step 1b is executed at each iteration. Let (G_i, λ_{G_i}) be the labeled hypergraph right before the ith execution of step 1b. We observe that $G_0 > G_1 > \cdots > G_k$ form a strictly descending chain. In particular, hypergraphs G_i are all distinct. We claim that $\langle G_i \rangle \in \mathcal{C}$ for all i. It follows that the number of iterations is at most $|\mathcal{C}|$. Notice that if $\langle G \rangle$ at the beginning of step 1a does not belong to $\mathcal{C}$, then G must have a unique central vertex, so step 1a is executed. So, either $\langle G \rangle \in \mathcal{C}$ and $\langle G_i \rangle = \langle G \rangle \in \mathcal{C}$, or $\langle G \rangle \notin \mathcal{C}$ and $(G_i, \lambda_{G_i}) = \mathsf{Core}(\mathsf{Central}(G, \lambda_G))$, so, also in this case $\langle G_i \rangle \in \mathcal{C}$. This proves that the number of iterations is at most $|\mathcal{C}|$.

It is shown in the appendix that for each call to Reduce and Central on input (G, λ_G) the time and space complexity, and the number of queries to the adversary, are proportional to the quantity $R(G, \lambda_G) = \sum_{v \in V(G)} \mathsf{rank}(\lambda_G^{-1}(v))$. Note that $R(\mathsf{Core}(H, \lambda_H)) \leq R(H, \lambda_H)$. In the analysis of these routines it is shown that when a routine returns (H, λ_H), $R(H, \lambda_H) \leq R(G, \lambda_G) + O(n)$. Because $R(\{aaa\}, f_a) = n$ and the main loop makes $O(1)$ iterations, the entire procedure makes $O(n)$ queries and runs in time polynomial in n.

In this extended abstract we have space only to define and analyze the routine Reduce for the general case when G is a core not isomorphic to $\{abc\}$ or $\{abb\}$. The routine Central, and the special cases of Reduce are given in the appendix.

6.1 Central

In this subsection we describe a routine Central that when given a valid (G, λ_G) such that G is a core with a unique central vertex, outputs a valid $(H, \lambda_H) \leq (G, \lambda_G)$ such that either $\lambda_H^{-1}(H)$ is intersecting or $\langle H \rangle \in \mathcal{C}$. Let (G, λ_G) be the input to Central, $V = V(G)$ be the vertices of G, and $u \in V$ the unique central vertex in G. The routine works iteratively, and maintains, at every iteration, a secret-containing labeled hypergraph (H, λ_H) with vertices $V(H) \subseteq V \cup \{\bar{u}\}$ satisfying the following properties:

- u is a central vertex in H.

- There exists a homomorphism $\chi \colon (H, \lambda_H) \to (G, \lambda_G)$ such that $\chi(\bar{u}) = u$ and $\chi(v) = v$ for all $v \neq \bar{u}$.

Initially, $(H, \lambda_H) = (G, \lambda_H)$, which clearly satisfies the properties with homomorphism $\chi = \mathrm{Id}_V$. We now describe the operations performed at every iteration. Notice that if $|\lambda_H^{-1}(u)| = 1$ then $\lambda_H^{-1}(H)$ is intersecting and Central can immediately terminate with output (H, λ_H). Otherwise $|\lambda_H^{-1}(u)| > 1$ and vertex u can be split. Let $(H_0, \lambda_0) = \mathsf{query}(\mathsf{Split}_u(H, \lambda_H))$, and $\pi \colon (H_0, \lambda_0) \to (H, \lambda_H)$ the associated homomorphism. Since π is a homomorphism, $\pi \circ \lambda_0 \subseteq \lambda_H$. If $\pi \circ \lambda_0 \neq \lambda_H$, then we have found a smaller secret-containing labeling for hypergraph H, so we can set λ_H to $\pi \circ \lambda_0$ and move to the next iteration. So, assume $\pi \circ \lambda_0 = \lambda_H$. In particular, $(\pi \circ \lambda_0)^{-1}(u) = \lambda_H^{-1}(u)$ and so it must be that $\pi^{-1}(u) = \{u_0, u_1\}$. If H_0 has zero or multiple central vertices, then $\langle H_0 \rangle \in \mathcal{C}$ and we can terminate with output (H_0, λ_0); note that $(H_0, \lambda_0) \leq (H, \lambda_H) \leq (G, \lambda_G)$. So, assume H_0 has a unique central vertex w. Notice that if $w \notin \{u_0, u_1\}$, then $\pi(w) \neq u$ and $\pi(H_0) \subseteq H$ has two distinct central vertices (namely, u and $\pi(w)$), and the hypergraph $\pi(H_0, \lambda_0) \subseteq (H, \lambda_H) \leq (G, \lambda_G)$ has a core that belongs to $\mathcal{C}$, so we can output (H_0, λ_0). So, consider the case with $w \in \{u_0, u_1\}$ and assume without loss of generality that $w = u_0$. Let $\pi_{u_1,\bar{u}}$ be the function such that $\pi_{u_1,\bar{u}}(u_1) = \bar{u}$ and $\pi_{u_1,\bar{u}}(v) = \pi(v)$ for all $v \in V(H_0) \setminus \{u_1\}$. Notice that $\pi_{u_1,\bar{u}}(u_0) = \pi(u_0) = u$ is the unique central vertex in $\pi_{u_1,\bar{u}}(H_0)$. Moreover, because $\chi \circ \pi_{u_1,\bar{u}} = \chi \circ \pi$ is a homomorphism from (H_0, λ_0) to (G, λ_G), χ

is a homomorphism from $\pi_{u_1,\bar{u}}(H_0,\lambda_0)$ to (G,λ_G). So, we can set (H,λ_H) to $\pi_{u_1,\bar{u}}(H_0,\lambda_0)$ and move to the next iteration.

Clearly, $\mathrm{rank}(\lambda_H^{-1}(u))$ decreases at every iteration, and therefore the routine makes at most $\sum_{v\in V(G)} \mathrm{rank}(\lambda_G^{-1}(v))$ queries. A simple induction argument shows that after every iteration, $\sum_{v\in V(H)} \mathrm{rank}(\lambda_H^{-1}(v)) \leq \cdot \sum_{v\in V(G)} \mathrm{rank}(\lambda_G^{-1}(v)) + O(n)$.

6.2 Reduce: the Case of Hypergraphs not Isomorphic to $\{abb\}$ or $\{abc\}$

In this subsection we describe the routine $\mathsf{Reduce}(G,\lambda_G)$ for the case when G is not isomorphic to $\{abb\}$ or $\{abc\}$. On the input (G,λ_G), Reduce returns a valid labeled hypergraph (H,λ_H) such that either $H < G$ or $\lambda_H^{-1}(H)$ is intersecting.

Let $V = V(G)$ and let $\bar{V} = \{\bar{v} \colon v \in V\}$ be a disjoint copy of V. Reduce works iteratively, maintaining a valid labeled hypergraph (H,λ_H), with $V(H) \subseteq V \cup \bar{V}$, that satisfies the following properties: (i) The function $\chi \colon (V\cup\bar{V}) \to V$ such that $\chi(\bar{v}) = \chi(v) = v$ for all $v \in V$, is a homomorphism from H to G. (ii) $G \subseteq H$ (iii) For every homomorphism $\phi \colon G \to H$, $\phi(V) = V$

If H satisfies properties (i) and (ii), we say that H is an *extension* of G. If it also satisfies property (iii) property, we say that H is *unambiguous*[2]. We guarantee termination by showing that after each iteration the sum $\sum_{v\in V} \mathrm{rank}(\lambda_H(v))$ decreases. It is when we bound this quantity after each iteration that we use that H is an unambiguous extension of G.

Initially, $(H,\lambda_H) = (G,\lambda_G)$. The procedure repeats the following loop:

1. Let $A = \{a \in V \colon \mathrm{rank}(\lambda_H^{-1}(a)) > 1\}$

2. Let $(H_0,\lambda_0) = \mathsf{query}(\mathrm{Split}_A(H,\lambda_H))$, and let π be the homomorphism $\pi_A \colon (H_0,\lambda_0) \to (H,\lambda_H)$ associated with the splitting operation

3. If for all $e,f \in E(H_0)$ there exists $v \in e \cap f$ with $\mathrm{rank}(\lambda_0^{-1}(v)) = 1$, then terminate with output (H_0,λ_0). Otherwise, choose $h_0,h_1 \in E(H_0)$ such that for all $v \in h_0 \cap h_1$, $\mathrm{rank}(\lambda_0^{-1}(v)) > 1$, and continue.

4. If $H_0 \not\to G$, then terminate with output (H_0,λ_0). Otherwise, compute a homomorphism $\psi \colon G \to H_0$ such that $\pi \circ \psi = \mathrm{Id}_G$ and continue.

5. If $\pi \circ \lambda_0 \neq \lambda_H$, set λ_H to $\pi \circ \lambda_0$ and move to the next iteration. Otherwise, for every $a \in A$, let π_a be the function
$$\pi_a(x) = \begin{cases} \bar{a} & \text{if } \pi(x) = a \text{ and } x \neq \psi(a) \\ \pi(x) & \text{otherwise} \end{cases}$$
and continue.

6. Select $a \in A$ such that $\pi_a(H_0)$ is an unambiguous extension of G and set (H,λ_H) to $\pi_a(H_0,\lambda_0)$ and move to the next iteration.

Theorem 8 *For any valid (G,λ_G) such that G is a core not isomorphic to $\{abb\}$ or $\{abc\}$, if $\mathsf{Reduce}(G,\lambda_G)$ terminates with output (H,λ_H), then (H,λ_H) is valid and either $H < G$ or $\lambda_H^{-1}(H)$ is intersecting.*

Proof: The correctness is based on the invariant that, at the beginning of each iteration, (H,λ_H) is valid and H is an unambiguous extension of G. This is clearly true before the first iteration when $(H,\lambda_H) = (G,\lambda_G)$. We need to prove that at every iteration, either the algorithm terminates at step 3 or 4 with a labeled hypergraph (H,λ_H) with $\lambda_H^{-1}(H)$ intersecting or $H < G$, or the invariant is preserved when moving to the next iteration at step 5 or 6.

Let (H,λ_H) be a valid labeled hypergraph so that H is an unambiguous extension of G, and let A be the set of vertices defined in step 1. By proposition 8, (H_0,λ_0) is a valid labeled hypergraph, and $\pi \colon (H_0,\lambda_0) \to (H,\lambda_H)$ is a homomorphism.

[2] In the graph homomorphism literature, it would be said that *G is uniquely H colorable*.

If the algorithm terminates at step 3, then for every $e, f \in E(H_0)$ there exists $v \in e \cap f$ with rank$(\lambda_0^{-1}(v)) = 1$. Therefore, $\lambda_0^{-1}(H_0)$ is intersecting (and valid). Otherwise, if there exist $h_0, h_1 \in E(H_0)$ with $\forall v \in h_0 \cap h_1$ rank$(\lambda_0^{-1}(v)) > 1$.

If the algorithm terminates at step 4, the output is correct because (H_0, λ_0) is secret-containing and $H_0 < G$. Otherwise, $G \sim H_0$ and G is a core. Moreover, $\chi \circ \pi \colon H_0 \to G$ is a homomorphism, so by Lemma 3 there exists a homomorphism $\psi \colon G \to H_0$ such that $\chi \circ \pi \circ \psi = \mathrm{Id}_G$. Because $\pi \circ \psi$ is a homomorphism from G to H and H is an unambiguous extension of G, $\pi \circ \psi = \mathrm{Id}_G$.

If the test in step 5 is satisfied, then the algorithm moves to the following iteration with labeled hypergraph $(H, \pi \circ \lambda_0)$ instead of (H, λ_H). By the induction hypothesis, (H, λ_H) is valid and H is an unambiguous extension of G. We need to prove that $S^* \in (\pi \circ \lambda_0)^{-1}(H)$. Because (H_0, λ_0) is valid, $\pi(H_0, \lambda_0) = (\pi(H_0), \pi \circ \lambda)$ is also valid; in particular it contains the secret. Because $\pi \colon (H_0, \lambda_0) \to (H, \lambda_H)$ is a homomorphism, $\pi(H_0, \lambda_0) = (\pi(H_0), \pi \circ \lambda_0) \subseteq (H, \pi \circ \lambda_0)$, so $(H, \pi \circ \lambda_0)$ contains the secret.

Now assume that the tests at steps 3, 4 and 5 failed. If this is the case, then step 6 is executed and we need to prove that there exists an $a \in A$ such that $\pi_a(H_0)$ is an unambiguous extension of G. Notice that for any $a \in A$, $\pi_a(H_0, \lambda_0)$ is valid because (H_0, λ_0) is valid.

Lemma 9 *For all $a \in A$, $\chi \circ \pi_a = \chi \circ \pi$.*

Proof: By the definition of π_a, for each $x \in V(H_0)$, if $\pi(x) = a$ and $x \neq \psi(a)$, then $\chi(\pi_a(x)) = \chi(\bar{a}) = a = \chi(a) = \chi(\pi(x))$, otherwise, $\pi_a(x) = \pi(x)$ so $\chi(\pi_a(x)) = \chi(\pi(x))$. ■

Lemma 10 *For all $a \in A$, $\pi_a(H_0)$ is an extension of G.*

Proof: First we show that $G \subseteq \pi_a(H_0)$. This is just because $\psi(G) \subseteq H_0$ and $\pi_a(\psi(G)) = \pi(\psi(G)) = G$. Now we show that χ is a homomorphism from $\pi_a(H_0)$ to G. By the composition of homomorphisms, $\chi \circ \pi$ is a homomorphism from H_0 to G. Therefore $\chi \circ \pi_a = \chi \circ \pi$ is a homomorphism from H_0 to G (we apply Lemma 9 here). Therefore, χ is a homomorphism from $\pi_a(H_0)$ to G. ■

Lemma 11 *For all $a \in A$, if $\pi_a(H_0)$ is not an unambiguous extension of G then for every edge $g \in E(G)$ there exists an edge $h \in H_0$ such that $\chi \circ \pi(h) = g$ and $\psi(a) \notin h$.*

Proof: Suppose that $\pi_a(H_0)$ is not an unambiguous extension of G. By Lemma 10, $\pi_a(H_0)$ is an extension of G, so there exists a homomorphism $\theta : G \to \pi_a(H_0)$ so that $\theta(V) \neq V$. By Lemma 3, we may choose θ so that both $\theta(V) \neq V$ and $\chi \circ \theta = \mathrm{Id}_G$. Let $g \in G$ be given, and choose $h \in H_0$ so that $\pi_a(h) = \theta(g)$. Note that $\chi(\pi(h)) = \chi(\pi_a(h)) = \chi(\theta(g)) = g$. In the case when $a \notin g$, $a \notin g = \chi(\theta(g)) = \chi(\pi_a(h)) = \chi(\pi(h))$; therefore $\psi(a) \notin h$ (since by choice of ψ in step 4, $\chi(\pi(\psi(a))) = a$). In the next paragraph we show that $\theta(a) \neq a$. This suffices to prove the lemma for the case when $a \in g$ by the following argument: Suppose for the sake of contradiction that $\psi(a) \in h$. Then $a = \pi_a(\psi(a)) \in \pi_a(h) = \theta(g)$. Choose $x \in g$, $\theta(x) = a$. Because $x = \chi(\theta(x)) = \chi(a) = a$ we have that $\theta(a) = a$; contradiction.

We now show that $\theta(a) \neq a$. Because $\theta(V) \neq V$ we may choose $e \in E(G)$ so that $\theta(e) \notin E(G)$. Choose $f \in E(H_0)$ so that $\pi_a(f) = \theta(e)$. Because π is a homomorphism from H_0 to H, we have that $\pi(f) \in E(H)$ and therefore $\pi_a(f) \neq \pi(f)$. However, by the definition of π_a, this can happen only when there is $a' \in f$, $a' \neq \psi(a)$ and $\pi(a') = a$. In particular, $\bar{a} \in \pi_a(f) = \theta(e)$, so $a \in \chi(\theta(e))$ and therefore $a \in e$. Suppose for the sake of contradiction that $\theta(a) = a$. Then, because $\pi_a^{-1}(\{a\}) = \{\psi(a)\}$ and $\pi_a(f) = \theta(e)$, $\psi(a) \in f$. Therefore, $\{a, \bar{a}\} \subseteq \pi_a(f) = \theta(e)$. Choose $y, z \in e \subseteq V(G)$ so that $\theta(y) = a$ and $\theta(z) = \bar{a}$. Note that $\chi(\theta(z)) = \chi(\bar{a}) = a = \chi(a) = \chi(\theta(y))$. Because $\chi \circ \theta = \mathrm{Id}_G$, this means that $z = y$, contradiction. ■

Because ψ is an isomorphism from G to $\psi(G)$ we get the following corollary:

Corollary 12 *For all $a \in A$, if $\pi_a(H_0)$ is ambiguous then for every edge $g \in E(\psi(G))$ there exists an edge $h \in E(H_0)$ so that $\psi \circ \chi \circ \pi(h) = g$ and $\psi(a) \notin h$. Moreover, if $\psi(a) \in g$ then $h \notin E(\psi(G))$.*

The following theorem is proved in section 8. It is the key to showing that there exists $a \in A$ so that $\pi_a(H_0)$ is an unambiguous extension of G.

Theorem 13 *(proof in Section 8) Let H_1 be a three-uniform, intersecting hypergraph core that is not isomorphic to $\{abc\}$ or $\{abb\}$. Let H_2 be a three-uniform, intersecting hypergraph so that H_1 is a core of H_2, and let $\phi : H_2 \to H_1$ be a homomorphism so that $\phi \restriction_{H_1} = Id_{H_1}$. There exists a non-empty set $S \subseteq V(H_1)$ so that (i) For each $v \in S$, there exists $e \in E(G)$ so that $v \in e$ and for each $f \in E(H)$ if $\phi(f) = e$ then $v \in f$ (ii) Either there exists $v \in S$ with $m_G(v) > 1$, or, for each $f_0, f_1 \in E(H) \setminus E(G)$, $f_0 \cap f_1 \cap S \neq \emptyset$.*

By hypothesis, the hypergraph G is an intersecting, three-uniform core that is isomorphic to neither $\{abc\}$ nor $\{abb\}$. Because $\chi \circ \pi \circ \psi = \mathrm{Id}_G$, for any $y = \psi(x)$, $\psi(\chi(\pi(y))) = \psi(\chi(\pi(\psi(x)))) = \psi(x) = y$. That is, $\psi \circ \chi \circ \pi \restriction_{\psi(G)} = \mathrm{Id}_{\psi(G)}$. Therefore, the hypotheses of Theorem 13 are met with $H_1 = \psi(G)$, $H_2 = H_0$, and $\phi = \psi \circ \chi \circ \pi$. Choose a set $S \subseteq V(\psi(G))$ as guaranteed by that theorem.

In the case when there is $v \in S$ so that $m_{\psi(G)}(v) > 1$, choose $a \in V(G)$ so that $\psi(a) = v$, note that $\pi(v) = a$. By property (i) and the contrapositive of corollary 12, $\pi_a(H_0)$ is an unambiguous extension of G. Because $m_{\psi(G)}(v) \geq 2$ and $\pi(v) = a$, $m_H(a) \geq 2$, and therefore by validity $|\lambda_H(a)| \geq 2$, so $a \in A$.

Now consider the case when $\forall v \in S$, $m_{\psi(G)}(v) = 1$. In this situation, for each $f_0, f_1 \in E(H) \setminus E(G)$, $f_0 \cap f_1 \cap S \neq \emptyset$. Recall the edges $h_0, h_1 \in E(H_0)$ found in Step 3 so that $\forall u \in h_0 \cap h_1$, $|\lambda_0^{-1}(u)| \geq 2$. For each $u \in h_0 \cap h_1$, because π: $(H_0, \lambda_0) \to (H, \lambda_H)$ is a homomorphism, $|\lambda_H^{-1}(\pi(u))| \geq |\lambda_0^{-1}(u)| > 1$. Therefore, for each $u \in h_0 \cap h_1 \cap \psi(V)$, $\pi(u) \in A$ (here we use that $\pi(\psi(V)) = V$). Choose $u \in h_0 \cap h_1 \cap \psi(V)$, and choose $a \in V$, $\psi(a) = u$. If $\pi_a(H_0)$ is unambiguous, we are finished. Suppose, however, that $\pi_a(H_0)$ is ambiguous. For each $i \in \{0, 1\}$, if $h_i \in E(\psi(G))$ then choose $f_i \in E(H_0) \setminus E(\psi(G))$ so that $\phi(f_i) = h_i$, otherwise set $f_i = h_i$. By the guarantee of Theorem 13, we may choose $w \in f_0 \cap f_1 \cap S$. Note that since $w \in S$, $\phi(w) = w$. Therefore, $w = \phi(w) \in \phi(f_0 \cap f_1) = h_0 \cap h_1$. Choose $a \in V$, so that $\psi(a) = w$. Suppose for the sake of contradiction that $\pi_a(H_0)$ is an unambiguous extension of G. By Corollary 12, for each $g \in E(G)$ there is $h \in E(H_0)$ with $\phi(h) = \psi(\chi(\pi(h))) = \psi(g)$ and $w = \psi(a) \notin h$; this contradicts property (i) of Theorem 13.

■

Where unambiguity comes into play is when we bound the number iterations and the sum of the ranks of the vertex label: Because $\pi \circ \psi = \mathrm{Id}_G$, for all $v \in V$, $\psi(v) \in \pi^{-1}(v)$. Recall that, by proposition 8, for each $v \in A$, $\mathrm{rank}(\lambda_0^{-1}(v_0)) < \mathrm{rank}(\lambda_H^{-1}(v))$ and $\mathrm{rank}(\lambda_0^{-1}(v_1)) < \mathrm{rank}(\lambda_H^{-1}(v))$, and for each $v \in V \setminus A$, $\mathrm{rank}(\lambda_0^{-1}(v)) = \mathrm{rank}(\lambda_H^{-1}(v)) = 1$. Therefore, $\mathrm{rank}((\pi_a \circ \lambda_H)^{-1}(a)) < \mathrm{rank}(\lambda_H^{-1}(a))$, and for all other $v \in V$, $\mathrm{rank}((\pi_a \circ \lambda_H)^{-1}(v)) = \mathrm{rank}(\lambda_H^{-1}(v))$. Applying this reasoning shows that:

Lemma 14 *For any secret-containing labeled hypergraph core (G, λ_G) such that G has at most one central vertex, $\mathsf{Reduce}(G, \lambda_G)$ terminates within at most $\sum_{v \in V}$ rank$(\lambda_G^{-1}(v))$ iterations.*

The proof of Lemma 14:

Proof: For any (H, λ_H) such that H is an extension of G, let $t(H, \lambda_H) = \sum_{v \in V} \mathrm{rank}(\lambda_H^{-1}(v))$. Notice that for $(H, \lambda_H) = (G, \lambda_G)$, $t(H, \lambda_H) = \sum_{v \in V} \mathrm{rank}(\lambda_G^{-1}(v))$. We show that $t(H, \lambda_H)$ decreases at every iteration.

Now is when we use the hypothesis that H is an unambiguous extension of G. We will show that for each $v \in V$, $\psi(v) \in \pi^{-1}(v)$. Because $\chi \circ \pi \circ \psi = \mathrm{Id}_G$, for each $v \in V$, $\psi(v) \in \{\bar{v}\} \cup \pi^{-1}(v)$. Because $\pi \circ \psi$ is a homomorphism from G to H and H is an unambiguous extension of G, $\pi(\psi(G)) = H[V] = G$. Since $\pi(\bar{v}) = \bar{v}$, this implies that $\psi(v) \in \pi^{-1}(v)$ for each $v \in V$. Notice that, by the definition of $(H_0, \lambda_0) \subseteq \mathrm{Split}_A(H, \lambda_H)$: for each $v \in A$, $\mathrm{rank}(\lambda_0^{-1}(v_0)) < \mathrm{rank}(\lambda_H^{-1}(v))$ and $\mathrm{rank}(\lambda_0^{-1}(v_1)) < \mathrm{rank}(\lambda_H^{-1}(v))$, and for each $v \in V \setminus A$, $\mathrm{rank}(\lambda_0^{-1}(v)) = \mathrm{rank}(\lambda_H^{-1}(v)) = 1$.

Consider the case when $\pi \circ \lambda_0 \subsetneq \lambda_H$ and step 5 is applied. We may choose some $v \in A$ with $|\pi^{-1}(v)| \leq 1$. Now, $(\pi \circ \lambda_0)^{-1}(v) = \lambda_0^{-1}(v)$ and for each $u \in V \setminus \{v\}$, $(\pi \circ \lambda_0)^{-1}(v) \subseteq \lambda_H^{-1}(v)$. Therefore, $t(H, \pi \circ \lambda_0) = \sum_{w \in V} \mathrm{rank}((\pi \circ \lambda_0)^{-1}(w)) < \sum_{w \in V} \mathrm{rank}(\lambda_H^{-1}(w)) = t(H, \lambda_H)$.

Now consider the case when $\pi \circ \lambda_0 = \lambda_H$. Choose $a \in A$ so that $\pi_a(H_0)$ is an unambiguous extension, as is done in step 6. Choose $i \in \{0, 1\}$ so that $\psi(a) = a_i$. Note that $(\pi_a \circ \lambda_0)^{-1}(a) = \lambda_0^{-1}(a_i)$, so that $\mathrm{rank}((\pi \circ \lambda_0)^{-1})(a) < \mathrm{rank}(\lambda_H^{-1}(a))$. For every $w \in V \setminus A$, $(\pi_a \circ \lambda_0)^{-1}(w) = \lambda_H^{-1}(w)$, so $\mathrm{rank}((\pi_a \circ \lambda_0)^{-1}(w)) = \mathrm{rank}(\lambda_H(w))$. Therefore $t(\pi_a(H_0), \pi_a \circ \lambda_0) = \sum_{w \in V} \mathrm{rank}((\pi \circ \lambda_0)^{-1}(w)) < \sum_{w \in V} \mathrm{rank}(\lambda_H^{-1}(w)) = t(H, \lambda_H)$.

■

A similar argument shows that when the routine begins with (G, λ_G), the sum of the ranks on all vertices of $V(H)$ never grows too large: $\sum_{v \in V(H)} \text{rank}(\lambda_H^{-1}(v)) \leq \sum_{v \in V(G)} \text{rank}(\lambda_G^{-1}(v)) + O(n)$

6.3 Reduce: the case for $\{abb\}$

Let G denote the hypergraph $\{abb\}$. Let $B = \{b, b^*, \bar{b}_1, \bar{b}_2, \bar{b}_3\}$. Let χ be the the mapping from $\{a\} \cup B$ to $\{a, b\}$ given by $\chi(a) = a$ and $\chi(x) = b$ for $x \in B$, An *extension* of G is a hypergraph H with vertices $V(H) \subseteq \{a, b, b^*, \bar{b}_1, \bar{b}_2, \bar{b}_3\}$, so that $\chi : H \to G$ is a homomorphism. We say that H is an *unambiguous* extension of G if for every homomorphism $\phi : G \to H$, $\phi(b) \in \{b, b^*\}$; H is *ambiguous* if it is not unambiguous.

Proposition 11 *Let H be an extension of G. If H is an ambiguous extension of G, then $\{a, \bar{b}_i, \bar{b}_i\} \in E(H)$ for some $i \in \{1, 2, 3\}$.*

As in the routine Reduce, we maintain an unambiguous extension $H \supseteq G$. Initially, set $(H, \lambda_H) = (G, \lambda_G)$; clearly this is an unambiguous extension of (G, λ_G). The routine repeats the following loop:

1. Let $A = \{x \in \{a, b, b^*\} \colon |\lambda_H^{-1}(x)| > 1\}$
2. Let $(H_0, \lambda_0) = \text{query}(\text{Split}_A(H, \lambda_H))$, and $\pi \colon (H_0, \lambda_0) \to (H, \lambda_H)$ the associated homomorphism.
3. If $\lambda_0^{-1}(H_0)$ is intersecting or $H_0 \not\to G$, then terminate with output (H_0, λ_0). Otherwise, let $\psi \colon G \to H_0$ be a homomorphism such that $\chi \circ \pi \circ \psi = \text{Id}_G$ and continue.
4. Choose a mapping $\rho : V(H_0) \to \{a, b, b^*, \bar{b}_1, \bar{b}_2, \bar{b}_3\}$ so that:
   (a) $\rho(H_0)$ is an unambiguous extension of G
   (b) $\rho^{-1}(\{a\}) \subseteq \text{Split}_A(\{a\})$ and $\rho^{-1}(\{b, b^*\}) \subseteq \text{Split}_A(\{b, b^*\})$
   (c) Either $|\rho^{-1}(\{a\})| = 1$, or $|\rho^{-1}(\{b\})| = 1$ and $|\rho^{-1}(\{b^*\})| \leq 1$
   (d) For each $x \in \{a, b, b^*\}$, if $|\rho^{-1}(x)| > 1$ then $\rho^{-1}(x) = \pi^{-1}(x)$

   Set $(H, \lambda_H) = \rho(H_0, \lambda_0)$, and proceed to the next iteration.

Theorem 15 *For any secret-containing labeled hypergraph (G, λ_G) with $G = \{abb\}$, if* Reduce_abb(G, λ_G) *terminates with output (H, λ_H), then (H, λ_H) is secret-containing and either $H < G$ or $\lambda_H^{-1}(H)$ is intersecting.*

The correctness if based on the invariant that, at the beginning of every iteration, (H, λ_H) is secret-containing and H is an unambiguous extension of G. As already observed, this is true for $(H, \lambda_H) = (G, \lambda_G)$ before the first iteration. We need to prove that at every iteration, either the algorithm terminates at step 3 with a labeled hypergraph (H, λ_H) satisfying the desired properties, or the invariant is preserved when moving to the next iteration at step 4.

Let (H, λ_H) be a secret-containing labeled hypergraph such that H is an unambiguous extension of G, and let A the set of all splittable vertices as defined in step 1. Note that A is nonempty because b has multiplicity 2, so by definition, $|\lambda^{-1}(b)| \geq 2$. Moreover, $a \in A$ because if $|\lambda^{-1}(a)| = 1$ then (H, λ_H) is intersecting. Let also $\pi \colon (H_0, \lambda_0) \to (H, \lambda_H)$ be as defined in step 2. By construction, (H_0, λ_0) is a secret-containing labeled hypergraph, H_0 is intersecting, and $\pi \colon (H_0, \lambda_0) \to (H, \lambda_H)$ is a homomorphism. In particular, $H_0 \leq H \leq G$.

If the algorithm terminates at step 3, the output is correct because (H_0, λ_0) is secret-containing and either $\lambda_0^{-1}(H_0)$ is intersecting or $H_0 < G$. Otherwise, G is the core of H by Lemma 3 there exists a homomorphism $\psi \colon G \to H$ such that $\chi \circ \pi \circ \psi = \text{Id}_V$ because $\chi \circ \pi \colon H \to G$ is a homomorphism.

In step 4, suppose that we have found a mapping ρ satisfying properties 4a,4b, 4c, and 4d. After the execution of step 4, the invariant is easily seen to hold: $\rho(H_0, \lambda_0)$ contains the secret because (H_0, λ_0) contains the secret, and by property 4a of the mapping ρ, $\rho(H_0, \lambda_0)$ is an unambiguous extension of G.

Now we will demonstrate that if step 4 is reached, there exists a mapping ρ satisfying the properties 4a-4d. For brevity, call a mapping satisfying these four properties *good*.

Lemma 16 *If $\psi(a)$ is a central vertex of H_0 then π is a good mapping.*

Proof: Clearly, $G = \pi(\psi(G)) \subseteq \pi(H_0) \subseteq H$, so $\pi(H_0)$ is an unambiguous extension of G, so property 4a holds. By definition, $\pi^{-1}(a) \subseteq \text{Split}_A(\{a\})$ and $\pi^{-1}(\{b, b^*\}) \subseteq \text{Split}_A(\{b, b^*\})$, so property 4b holds. Because $\psi(a)$ is a central vertex of H_0, $\pi^{-1}(\{a\}) = \{\psi(a)\}$, so property 4c holds. Finally, property 4d holds vacuously for π. ■

Lemma 17 *If for all $x \in \text{Split}_A(\{b, b^*\}) \setminus \{\psi(b)\}$, $m_{H_0}(x) = 1$ and there is at most one $i \in \{1, 2, 3\}$ with $\bar{b}_i \sim x$, then there is a good mapping $\rho : V(H_0) \to \{a, b, b^*, \bar{b}_1, \bar{b}_2, \bar{b}_3\}$.*

Proof: Consider the case when $|V(H_0) \cap \text{Split}_A(\{b, b^*\})| = 4$; the cases when $|V(H_0) \cap \text{Split}_A(\{b, b^*\})| \leq 3$ are handled similarly.

Let $\text{Split}_A(\{b, b^*\}) = \{\psi(b), x, y, z\}$. Choose $i \in \{1, 2, 3\}$ so that $y \not\sim \bar{b}_i$, and choose $j \in \{1, 2, 3\} \setminus \{i\}$ so that $z \not\sim \bar{b}_j$ (this may be done because y and z each share an edge with at most one of $\bar{b}_1$, $\bar{b}_2$, $\bar{b}_3$).

Let ρ be the following mapping: $\rho(a_0) = \rho(a_1) = a$, $\rho(\psi(b)) = b$, $\rho(x) = b^*$, $\rho(y) = \bar{b}_i$, and $\rho(z) = \bar{b}_j$, for each $k \in \{1, 2, 3\}$, $\rho(\bar{b}_k) = \rho(\bar{b}_k)$, and for $z \in \text{Split}_A(\{b, b^*\}) \setminus \{\psi(b), x\}$, $\rho(z) = \bar{b}_1$.

Clearly, ρ satisfies properties 4b, 4c and 4d. What needs to be demonstrated is property 4a, that $\rho(H_0)$ is an unambiguous extension of G. Note that $\rho(\psi(G)) = G$, so $G = \rho(\psi(G)) \subseteq \rho(H_0)$. Also, $\chi \circ \rho = \chi \circ \pi$, so χ is a homomorphism from $\rho(H_0)$ to G. Therefore, $\rho(H_0)$ is an extension of G.

Suppose that there is $e \in E(H_0)$ with $\rho(e) = \{a, \bar{b}_i, \bar{b}_i\}$. Since $\rho^{-1}(\bar{b}_i) = \{y, \bar{b}_i\}$, $\psi(\{a, b, b\}) \in E(H_0)$, and H_0 is intersecting, $\psi(a) \in e$. Because $y \not\sim_{H_0} \bar{b}_i$, $e \neq \{\psi(a), y, \bar{b}_i\}$. We cannot have that $e = \{\psi(a), \bar{b}_i, \bar{b}_i\}$ because then then $\pi(e) = \{a, \bar{b}_i, \bar{b}_i\} \in \pi(H_0) \subseteq (H)$, contradiction to H being unambiguous. Finally, we cannot have that $e = \{\psi(a), y, y\}$ because y has maximum multiplicity 1 by hypothesis. The possibilities of $\rho(e) = \{a, \bar{b}_k \bar{b}_k\}$ for each $k \in \{1, 2, 3\} \setminus \{i\}$ are ruled out in a similar fashion. ■

Lemma 18 *If $\psi(a)$ is not a central vertex of H_0 and there is $x \in \text{Split}_A(\{b, b^*\}) \setminus \psi(b)$ so that $m_{H_0}(x) = 2$ or there are distinct $i, j \in \{1, 2, 3\}$ so that $x \sim \bar{b}_i$ and $x \sim \bar{b}_j$, then there is a good mapping $\rho : V(H_0) \to \{a, b, b^*, \bar{b}_1, \bar{b}_2, \bar{b}_3\}$.*

Proof: We proceed in three steps. First, let $e_0 = \{\psi(a), \psi(b), \psi(b)\}$, and let a' be the unique element of $\text{Split}_A(\{a\}) \setminus \{\psi(a)\}$. We will show that, regardless of whether the case is that $m_{H_0}(x) = 2$ or there are distinct $\bar{b}_i$, $\bar{b}_j$ with $x \sim \bar{b}_i$ and $x \sim \bar{b}_j$, the edge $f = \{a', \psi(b), x\}$ is an edge of H_0. Then we will show that for every $e \in E(H_0)$, $e \cap \{x, \psi(b)\} \neq \emptyset$. Finally, we use this observation to construct ρ as desired.

Consider the case when there is $x \in \text{Split}_A(\{b, b^*\}) \setminus \psi(b)$ so that $m_{H_0}(x) = 2$. Choose e_1 so that x occurs in e_1 with multiplicity two; because H_0 is intersecting $e_1 = \{\psi(a), x, x\}$. Because $\psi(a)$ is not a central vertex of H_0, we may choose $f \in E(H_0)$ with $\psi(a) \notin f$. Note that $f = \{a', \psi(b), x\}$ because f contains a vertex of $(\chi \circ \pi)^{-1}(a)$ and f intersects both e_0 and e_1.

Now consider the case that there is $x \in \text{Split}_A(\{b, b^*\}) \setminus \psi(b)$ so that there are distinct $i, j \in \{1, 2, 3\}$ so that $x \sim b_i$ and $x \sim b_j$. Let $e_0 = \{\psi(a), \psi(b), \psi(b)\}$, and choose e_1, e_2 so that $x, b_i \in e_1$ and $x, b_j \in e_3$. Because H_0 is intersecting and each edge contains an element of $(\chi \circ \pi)^{-1}(a)$, $e_1 = \{\psi(a), x, b_i\}$ and $e_2 = \{\psi(a), x, b_i\}$. Because $\psi(a)$ is not a central vertex of H_0, we may choose $f \in E(H_0)$ with $\psi(a) \notin f$. Note that $f = \{a', \psi(b), x\}$ because f contains a vertex of $(\chi \circ \pi)^{-1}(a)$ and f intersects each of e_0,e_1, and e_2.

Let $e \in E(H_0)$ be given, and suppose for the sake of contradiction that $e \cap \{\psi(b), x\} = \emptyset$. Because $e \cap f \neq \emptyset$, $a' \in e$. Because $\chi(\pi(e)) = \{a, b, b\}$ and $a' \in e$, $\psi(a) \notin e$. Therefore, $e \cap e_0 = e \cap \{\psi(a), \psi(b), x\} = \emptyset$; contradiction to H_0 being intersecting.

Let ρ be the following mapping: $\rho(a') = \rho(a) = a$, $\rho(\psi(b)) = b$, $\rho(x) = b^*$, for $z \in \text{Split}_A(\{b, b^*\}) \setminus \{\psi(b), x\}$, $\rho(z) = \bar{b}_1$, and for each $k \in \{1, 2, 3\}$, $\rho(\bar{b}_k) = \rho(\bar{b}_k)$. Clearly, properties 4b, 4c, and 4d hold. Observe that $\psi(G) = G$ so $G = \rho(\psi(G)) \subseteq \rho(H_0)$. Also, $\chi \circ \rho = \chi \circ \pi$ so χ is a homomorphism from $\rho(H_0)$ to G. What needs to be shown is that $\rho(H_0)$ is an unambiguous extension of H.

Let $e \in E(H_0)$ be given; we will show that $\rho(e) \neq \{a, \bar{b}_i, \bar{b}_i\}$, for any $i \in \{1, 2, 3\}$. If $\psi(b) \in e$, the $b \in \rho(e)$ and $\rho(e)$ is neither $\{a, \bar{b}_1, \bar{b}_1\}$, $\{a, \bar{b}_2, \bar{b}_2\}$, nor $\{a, \bar{b}_3, \bar{b}_3\}$. Similarly, if $x \in e$, then $b^* \in \rho(e)$ and $\rho(e)$ is neither $\{a, \bar{b}_1, \bar{b}_1\}$, $\{a, \bar{b}_2, \bar{b}_2\}$, nor $\{a, \bar{b}_3, \bar{b}_3\}$. ■

Theorem 19 *Let (G, λ_G) be a secret-containing labeld hypergraph with $G = \{abb\}$. The routine* Reduce_abc(G, λ_G) *terminates within at most rank$(\lambda_G^{-1}(a))$ + rank$(\lambda_G^{-1}(b))$ many iterations.*

Proof: We bound the number of iterations by showing that at each iteration, either the rank of $\lambda_H^{-1}(a)$ decreases, or $\max(\text{rank}(\lambda_H^{-1}(b)), \text{rank}(\lambda_H^{-1}(b^*)))$ decreases.

Consider the cases guaranteed by property 4c: $|\rho^{-1}(\{a\}) = 1$ or $|\rho^{-1}(\{b\})| = 1$ and $|\rho^{-1}(\{b\})| \leq 1$.

Consider the case when $|\rho^{-1}(a)| = 1$. Because $\rho^{-1}(a) \subseteq \text{Split}_A(\{a\})$, property 4b, there is a unique $i \in \{0, 1\}$ so that $\rho(a_i) = a$. Note that rank$((\rho \circ \lambda_0)^{-1}(a))$ = rank$(\lambda_{H_0}^{-1}(a_i))$ < rank$(\lambda_H^{-1}(a))$. Because $|\rho^{-1}(b)| \leq 1$ or $\rho^{-1}(b) = \pi^{-1}(b)$, rank$((\rho \circ \lambda_0)^{-1}(b)) \leq$ rank$((\pi \circ \lambda_0)^{-1}(b))$ = rank$(\lambda_H^{-1}(b))$. Similarly, rank$((\rho \circ \lambda_0)^{-1}(b^*)) \leq$ rank$(\lambda_H^{-1}(b^*))$.

Consider the case when $|\rho^{-1}(\{b\})| = 1$ and $|\rho^{-1}(\{b^*\})| \leq 1$. Because $\rho^{-1}(\{b, b^*\}) \subseteq \text{Split}_A(\{b, b^*\})$, there is exactly one $x \in \text{Split}_A(\{b, b^*\})$ with $\rho(x) = b$ and at most one $y \in \text{Split}_A(\{b, b^*\})$ so that $\rho(y) = b^*$. Note that rank$(\lambda_0^{-1}(x)) < \max(\text{rank}(\lambda_H^{-1}(b)), \text{rank}(\lambda_H^{-1}(b^*)))$ and rank$(\lambda_0^{-1}(y)) < \max(\text{rank}(\lambda_H^{-1}(b)), \text{rank}(\lambda_H^{-1}(b^*)))$. Therefore, $\max(\text{rank}((\rho \circ \lambda_0)^{-1}(b), (\rho \circ \lambda_0)^{-1}(b^*))) < \max(\text{rank}(\lambda_H^{-1}(b)), \text{rank}(\lambda_H^{-1}(b^*)))$. Finally, rank$((\rho \circ \lambda_0)^{-1}(a)) \leq$ rank$(\lambda_H^{-1}(a))$ because either $|\rho^{-1}(a)| = 1$ or $\rho^{-1}(\{a\}) = \pi^{-1}(\{a\})$.

■

Proposition 12 $\sum_{v \in V(H)}$ *rank*$(\lambda_H^{-1}(v)) \leq \sum_{v \in V(G)}$ *rank*$(\lambda_G^{-1}(v)) + O(n)$

6.4 Reduce: the case for $\{abc\}$

There is no meaningful notion of "unambiguous" extension for the hypergraph $\{abc\}$: its one edge can get mapped into any edge of any non-empty three-uniform hypergraph. For this reason, the routine Reduce_abc differs from the other routines. It is based on a case-analysis of what hypergraphs embed into $\{abc\}$.

There are three kinds of three-uniform intersecting hypergraphs that embed into $\{abc\}$ (corollary 32 of section 8). These are: hypergraphs with a central vertex, hypergraphs with no central vertex and a strong-cover of size three, and hypergraphs isomorphic with a particular hypergraph on six vertices that we call K.

Definition 6.1 *Throughout this subsection, let G denote the hypergraph $\{abc\}$. Let K denote the hypergraph $\{abc, ab'c', a'bc', a'b'c\}$. Let T denote the hypergraph $\{abc, ab\bar{c}, a\bar{b}c, \bar{a}bc\}$. Let χ be the mapping from $\{a, a', \bar{a}, b, b', \bar{b}, c, c', \bar{c}\}$ to $\{a, b, c\}$ given by $\chi(\bar{a}) = \chi(a') = \chi(a) = a$, $\chi(\bar{b}) = \chi(b') = \chi(b) = b$, and $\chi(\bar{c}) = \chi(c') = \chi(c) = c$.*

Proposition 13 *Let H be one of G, T, or K. The mapping χ is a homomorphism from H to G.*

The routine is a case analysis of the different transitions between the hypergraph H and the new hypergraph H_0 obtained by splitting the vertices and deleting edges. We use two sub-routines called Reduce_T and Reduce_K to handle the hypergraphs that arise. These routines are defined in subsubsection 6.4.1) and 6.4.2, respectively.

The routine Reduce_abc takes labeled hypergraph (G, λ_G) (where $G = \{abc\}$), and repeats the following loop until an exit condition is met:

1. Let $A = \{v \in \{a, b, c\}\colon \text{rank}(\lambda_G^{-1}(v)) > 1\}$

2. Set $(H_0, \lambda_0) = \text{query}(\text{Split}_A(G, \lambda_G))$

3. If ${\lambda_0}^{-1}(H_0)$ is intersecting, then output (H_0, λ_0)

4. (a) If H_0 has a central vertex, then let $\lambda_G = \pi \circ \lambda_0$ and proceed to the next iteration

(b) If $\tau(H_0) > 1$ and $\tau^*(H_0) = 3$ then choose $a^* \in \mathsf{Split}_A(\{a\})$, $b^* \in \mathsf{Split}_A(\{b\})$, and $c^* \in \mathsf{Split}_A(\{c\})$ so that $\{a^*, b^*, c^*\}$ is a strong cover of H_0, and define ρ as follows:

$$\rho(x) = \begin{cases} \bar{a} & \text{if } \pi(x) = a \text{ and } x \neq a^* \\ \bar{b} & \text{if } \pi(x) = b \text{ and } x \neq b^* \\ \bar{c} & \text{if } \pi(x) = c \text{ and } x \neq c^* \\ \chi(\pi(x)) & \text{otherwise} \end{cases}$$

Let $(H, \lambda_H) = \mathsf{Reduce\_T}(\rho(H_0, \lambda_0))$ If $\lambda_H^{-1}(H)$ is intersecting, return (H, λ_H), otherwise set $\lambda_G = \lambda_H$ and proceed to the next iteration of the loop.

(c) If $\tau(H_0) > 1$ and $\tau^*(H_0) > 3$ then let ρ be an isomorphism from H_0 to K so that $\chi \circ \rho = \chi \circ \pi$, and let $(H, \lambda_H) = \mathsf{Reduce\_K}(\rho(H_0, \lambda_0), \lambda_G)$. If $\lambda_H^{-1}(H)$ is intersecting then return (H, λ_H) otherwise set $\lambda = \lambda_H$ and proceed to the next iteration.

This is the correctness invariant of Reduce_abc is that at each stage, (G, λ_G) contains the secret.

Clearly the invariant holds at the beginning of the first iteration. Suppose that the invariant holds for λ_G.

It is clear that if $\lambda_G^{-1}(G)$ is not intersecting, then there is $v \in V(H)$ with $\text{rank}(\lambda_H^{-1}(v)) > 1$. By the properties of splitting vertices and querying the adversary to remove edges, (H_0, λ_0) contains the secret, H_0 is intersecting and three-uniform, and π is a homomorphism from (H_0, λ_0) to (G, λ_G).

Consider the case when $\tau(H_0) = 1$ and branch 4a is taken. Because (H_0, λ_0) contains the secret, $\pi(H_0, \lambda_0)$ contains the secret. Observe that $\pi(H_0, \lambda_0) = (G, \pi \circ \lambda_0)$.

Consider the case when $\tau(H_0) > 1$ and $\tau^*(H_0) = 3$. We first show that we can choose $a^* \in \mathsf{Split}_A(\{a\})$, $b^* \in \mathsf{Split}_A(\{b\})$, and $c^* \in \mathsf{Split}_A(\{c\})$ so that $\{a^*, b^*, c^*\}$ is a strong-cover of H_0.

Lemma 20 *Suppose that $\tau(H_0) > 1$ and $\tau^*(H_0) = 3$. Let Δ be a strong-cover of H of size 3. For each $w \in \{a, b, c\}$, $|(\chi \circ \pi)^{-1}(w) \cap \Delta| = 1$.*

Proof: Let $\{x, y, z\} = \Delta$. Because H_0 has no central vertex, there is $e \in E(H_0)$ so that $z \notin e$. Because $\{x, y, z\}$ is a strong-cover of H_0, $x \in e$ and $y \in e$. Because $\chi \circ \pi$ is a homomorphism from H_0 to $\{a, b, c\}$, $\chi \circ \pi(e) = \{a, b, c\}$, so $\chi \circ \pi(x) \neq \chi \circ \pi(y)$. Similarly, by considering $f, g \in E(H_0)$ so that $x \notin f$ and $y \notin g$, we learn that $\chi(\pi(x)) \neq \chi(\pi(z))$ and that $\chi(\pi(y)) \neq \chi(\pi(z))$. Therefore, $\chi(\pi(\Delta)) = \chi(\pi(\{x, y, z\})) = \{a, b, c\}$, so for each $w \in \{a, b, c\}$, $|(\chi \circ \pi)^{-1}(w) \cap \Delta| = 1$. ■

Let ρ be the function defined in step 4b. We now show that the preconditions of the routine Reduce_T are met (see subsubsection 6.4.1): Because (H_0, λ_0) contains the secret, $\rho(H_0, \lambda_0)$ contains the secret. Clearly, the vertex set of $\rho(H)$ is $\{a, b, c, \bar{a}, \bar{b}, \bar{c}\}$. Because $\chi \circ \rho = \chi \circ \pi$, χ is a homomorphism from $\rho(H_0)$ to G. Finally, because $\{a', b', c'\}$ is a strong-cover of H_0, $\rho(\{a', b', c'\}) = \{a, b, c\}$ is a strong-cover of $\rho(H_0)$, and $\rho(H_0)$ has no central vertex because...

Because the preconditions of the routine Reduce_T are met, $\mathsf{Reduce\_T}(\rho(H_0, \lambda_0), \lambda_G)$ returns (H, λ_H) so that either $\lambda_H^{-1}(H)$ is intersecting or $(H, \lambda_H) \subseteq (G, \lambda_G)$.

Consider the case when $\tau(H_0) > 1$ and $\tau^*(H_0) > 3$. By corollary 32, $H_0 \cong K$. The following lemma will show that there is an isomorphism $\rho : H_0 \to K$ so that $\chi \circ \rho = \chi \circ \pi$; for now suppose that we have such an isomorphism. Because (H_0, λ_0) contains the secret, $\rho(H_0, \lambda_0)$ contains the secret as well. Because $\chi \circ \rho = \chi \circ \pi$ is a homomorphism from (H_0, λ_0) to (G, λ_G), χ is a homomorphism from $\rho(H_0, \lambda_0)$ to (G, λ_G). Thus, the preconditions of the routine Reduce_K are met (see subsubsection 6.4.2), and $\mathsf{Reduce\_K}(\rho(H_0, \lambda_0), \lambda_G)$ returns (H, λ_H) such that either $\lambda_H^{-1}(H)$ is intersecting or $(H, \lambda_H) \subseteq (G, \lambda_G)$.

Lemma 21 *If H_0 is isomorphic to K then there is an isomorphism $\rho : H_0 \to K$ so that $\chi \circ \rho = \chi \circ \pi$.*

Proof: Let ρ be an isomorphism from H_0 to K; we will find an automorphism ν of K so that $\nu \circ \rho$ has the desired property. Choose a permutation η of $\{a, b, c\}$ so that $\chi \circ \pi = \eta \circ \chi \circ \rho$. Let ν be the permutation of $\{a, a', b, b', c, c'\}$ defined by $\nu(d) = \eta(a)'$, $\nu(b') = \eta(b)'$, $\nu(c') = \eta(c)'$, and $\nu \restriction_{\{a,b,c\}} = \eta$. Note that ν is an automorphism of K. Moreover, $\eta \circ \chi = \chi \circ \nu$. Therefore, $\chi \circ \pi = \eta \circ \chi \circ \rho = (\eta \circ \chi) \circ \rho = (\chi \circ \nu) \circ \rho = \chi \circ (\nu \circ \rho)$. ■

Complexity Considerations: In subsections 6.4.1 and 6.4.2, it is shown that if the routines Reduce_T and Reduce_K return $(J, \lambda_J) \subseteq (G, \lambda_G)$ within $O(s)$ steps, then $\sum_{v \in V(J)} \text{rank}(\lambda_J^{-1}(v)) \leq \sum_{v \in V(J)} \text{rank}(\lambda_J^{-1}(v)) - s$. Notice that when branch 4a is taken, $\sum_{v \in V(G)} \text{rank}(\lambda_G^{-1}(v))$ decreases by 1. Therefore, the routine terminates within $\sum_{v \in V(G)} \text{rank}(\lambda^{-1}(v))$ many steps.

6.4.1 The Routine Reduce_T

The input is a secret-containing labeled hypergraph (H, λ_H) and a labeling λ_G so that so that: χ is a homomorphism from (H, λ_H) to (G, λ_G), $V(H) = \{a, b, c, \bar{a}, \bar{b}, \bar{c}\}$, $\{a, b, c\}$ is a strong-cover of H and H has no central vertex. The output will be a secret containing labeled hypergraph (J, λ_J) so that either $\lambda_J^{-1}(J)$ is intersecting or $(J, \lambda_J) \subseteq (G, \lambda_G)$.

The following loop is iterated until an exit condition is met.

1. Let $A = \{x \in \{a, b, c\} \colon |\lambda_H^{-1}(x)| > 1\}$
2. Let $(H_0, \lambda_0) = \text{query}(\text{Split}_A(H, \lambda_H))$, and $\pi \colon (H_0, \lambda_0) \to (H, \lambda_H)$ the associated homomorphism.
3. If $\lambda_0^{-1}(H_0)$ is intersecting then terminate with output (H_0, λ_0).
4. (a) If $\{\bar{a}, \bar{b}, \bar{c}\} \not\subseteq V(H_0)$ then

   $$\lambda(s) = \begin{cases} z & \text{if } \lambda_G(s) = z \text{ and } \lambda_0(s) = \hat{z} \\ \lambda_G(s) & \text{if } \lambda_G(s) \neq z \\ \perp & \text{Otherwise} \end{cases}$$

   return (G, λ)

   (b) Choose $a^* \in \text{Split}_A(\{a\})$, $b^* \in \text{Split}_A(\{b\})$, and $c^* \in \text{Split}_A(\{c\})$ so that $\{a^*, b^*, c^*\}$ is a strong cover of H_0, and define ρ as follows:

   $$\rho(x) = \begin{cases} \bar{a} & \text{if } \pi(x) = a \text{ and } x \neq a^* \\ \bar{b} & \text{if } \pi(x) = b \text{ and } x \neq b^* \\ \bar{c} & \text{if } \pi(x) = c \text{ and } x \neq c^* \\ \pi(x) & \text{otherwise} \end{cases}$$

   Set $(H, \lambda_H) = \rho(H_0, \lambda_0)$ and repeat the loop.

The invariant for the routine Reduce_T is that: (H, λ_H) contains the secret, $V(H) = \{a, b, c, \bar{a}, \bar{b}, \bar{c}\}$, χ is a homomorphism from (H, λ_H) to (G, λ_G), $\{a, b, c\}$ is a strong cover of H, and H has no central vertex.

Clearly this invariant holds at the beginning of the first iteration. Consider a subsequent iteration and assume that the invariant holds at the beginning of the loop for the labeled hypergraph (H, λ_H). By the simple properties of splitting a hypergraph and querying the adversary to delete edges, H_0 is secret-containing and π is a homomorphism from H_0 to H.

In step 3, if $\lambda_0^{-1}(H_0)$ is intersecting, we return (H_0, λ_0) and are finished.

Consider the case when branch 4a is taken. Because $\chi \circ \pi$ is a homomorphism from (H_0, λ_0) to (G, λ_G), $\lambda^{-1}(z) = \lambda_0^{-1}(\hat{z}) \subseteq \lambda_G^{-1}(z)$. For $x \in \{a, b, c\} \setminus \{z\}$, $\lambda^{-1}(x) = \lambda_G^{-1}(x)$.

Consider the case when step 4b is taken. First we will show that because $\{\bar{a}, \bar{b}, \bar{c}\} \subseteq V(H_0)$, we can find $a^* \in \text{Split}_A(\{a\})$, $b^* \in \text{Split}_A(\{b\})$, and $c^* \in \text{Split}_A(\{c\})$, so that $\{a^*, b^*, c^*\}$ is a strong cover of H_0. The we will show that $\rho(H_0, \lambda_0)$ satisfies the invariant.

Because π is a homomorphism from H_0 to H, $\{a, b, c\}$ is a strong-cover of H, and χ is a homomorphism from H to G, for each $e \in E(H_0)$, there are $x, y \in \{a, b, c\}$ so that $|\text{Split}_A(\{x\}) \cap e| = 1$ and $|\text{Split}_A(\{y\}) \cap e| = 1$.

Choose $e, f, g \in E(H_0)$ so that $\bar{a} \in e$, $\bar{b} \in f$, and $\bar{c} \in g$. Choose $a^{**}, a^{***} \in \text{Split}_A(\{a\})$, $b^*, b^{***} \in \text{Split}_A(\{b\})$, and $c^*, c^{**} \in \text{Split}_A(\{c\})$ so that $e = \{\bar{a}, b^*, c^*\}$, $f = \{a^{**}, \bar{b}, c^{**}\}$, and $g = \{a^{***}, b^{***}, \bar{c}\}$. Because $e \cap f \neq \emptyset$, $c^* = c^{**}$, because $e\cap \neq \emptyset$, $b^* = b^{***}$, and because $f \cap g \neq \emptyset$, $a^{**} = a^{***}$. Therefore, $\{\bar{a}, b^*, c^*\}$, $\{a^*, \bar{b}, c^*\}$, $\{a^*, b^*, \bar{c}\} \in E(H_0)$, and therefore $\{a^*, b^*, c^*\}$ is a strong cover of H_0.

We now show that the loop invariant is preserved: $\rho(H_0, \lambda_0)$ contains the secret because (H_0, λ_0) contains the secret It is trivial to check that $V(\rho(H_0)) = \{a, b, c, \bar{a}, \bar{b}, \bar{c}\}$. χ is a homomorphism from $\rho(H_0, \lambda_0)$ to (G, λ_G) simply because $\chi \circ \rho = \chi \circ \pi$ and $\chi \circ \pi$ is a homomorphismfrom (H_0, λ_0) to (G, λ_G). Because $\{a^*, b^*, c^*\}$ is a strong-cover of H_0, $\rho(\{a^*, b^*, c^*\}) = \{a, b, c\}$ is a strong-cover of $\rho(H_0)$. Finally, because χ is a homomorphism from $\rho(H_0)$ to G, no edge of $\rho(H_0)$ contains both a and $\bar{a}$, nor does one contain both b and $\bar{b}$, nor does one contain both c and $\bar{c}$. Because $\{a, b, c, \bar{a}, \bar{b}, \bar{c}\} \subseteq V(\rho(H_0))$, $\rho(H_0)$ has no central vertex.

Proposition 14 *Consider an iteration of the loop that begins with (H, λ_H).*

1. *If the branch 4a is taken, then rank$(\lambda^{-1}(z)) =$ rank$(\lambda_0^{-1}(\hat{z}))$ for some $z \in \{a, b, c\}$ and some $\hat{z} \in$ Split$_A(\{z\})$. For $x \in \{a, b, c\} \setminus \{z\}$, rank$(\lambda^{-1}(x)) =$ rank$(\lambda_G^{-1}(x))$.*
2. *If the branch 4b is taken, then for each $z \in \{a, b, c\}$, either rank$((\rho \circ \lambda_H)^{-1}(z)) = 1$ or rank$((\rho \circ \lambda_H)^{-1}(z)) <$ rank$(\lambda_H^{-1}(z))$*

Proposition 15 *The routine* Reduce_T$((H, \lambda_H), \lambda_G)$ *terminates within rank$(\lambda_H^{-1}(a))+$rank$(\lambda_T^{-1}(b))+$rank$(\lambda_T^{-1}(c))$ steps. Moreover, if, for each $z \in \{a, b, c\}$, rank$(\lambda_H^{-1}(z)) \leq$ rank$(\lambda_G^{-1}(z))$, and the routine exits thru branch 4a after s iterations, then rank$(\lambda^{-1}(a))+$ rank$(\lambda^{-1}(b))+$ rank$(\lambda^{-1}(c)) \leq$ rank$(\lambda_G^{-1}(a))+$ rank$(\lambda_G^{-1}(b))+$ rank$(\lambda_G^{-1}(c)) - s$.*

Proposition 16 $\sum_{v \in V(H)}$ *rank*$(\lambda_H^{-1}(v)) \leq \sum_{v \in V(G)}$ *rank*$(\lambda_G^{-1}(v)) + O(n)$

6.4.2 The Routine Reduce_K

The input is a secret-containing labeled hypergraph (H, λ_H), and a labeling function λ_G where H is the hypergraph K and χ is a homomorphism from (H, λ_H) to (G, λ_G). The output will be a secret containing labeled hypergraph (J, λ_J) so that either $\lambda_J^{-1}(J)$ is intersecting or $(J, \lambda_J) \subseteq (G, \lambda_G)$.

1. Let $A = \{x \in \{a, b, c, a', b', c'\} \colon |\lambda_H^{-1}(x)| > 1\}$
2. Let $(H_0, \lambda_0) =$ query(Split$_A(H, \lambda_H)$), and $\pi \colon (H_0, \lambda_0) \to (H, \lambda_H)$ the associated homomorphism.
3. If $\lambda_0^{-1}(H_0)$ is intersecting then terminate with output (H_0, λ_0).
4. (a) If $\tau(H_0) > 1$ and $\tau^*(H_0) > 3$ then let ρ be an isomorphism from H_0 to K so that $\chi \circ \rho = \chi \circ \pi$, let $(H, \lambda_H) = \rho(H_0, \lambda_0)$, and proceed to the next iteration

   (b) If H_0 has a central vertex then let $\hat{z}$ be a central vertex of H_0, and let $z = \chi(\pi(\hat{z}))$. Let $\lambda : \{0,1\}^n \to \{a, b, c\}$ be the following partial function

$$\lambda(s) = \begin{cases} z & \text{if } \lambda_G(s) = z \text{ and } \lambda_0(s) = \hat{z} \\ \lambda_G(s) & \text{if } \lambda_G(s) \neq z \\ \perp & \text{Otherwise} \end{cases}$$

   return (G, λ)

   (c) Otherwise, choose $a^* \in$ Split$_A(\{a, a'\})$, $b^* \in$ Split$_A(\{b, b'\})$, and $c^* \in$ Split$_A(\{c, c'\})$ so that $\{a^*, b^*, c^*\}$ is a strong cover of H_0, and define ρ as follows:

$$\rho(x) = \begin{cases} \bar{a} & \text{if } \chi(\pi(x)) = a \text{ and } x \neq a^* \\ \bar{b} & \text{if } \chi(\pi(x)) = b \text{ and } x \neq b^* \\ \bar{c} & \text{if } \chi(\pi(x)) = c \text{ and } x \neq c^* \\ \chi(\pi(x)) & \text{otherwise} \end{cases}$$

   Return Reduce_T$(\rho(H_0, \lambda_0), \lambda_G)$

The correctness of the routine Reduce_K is based on the following invariant: (H, λ_H) contains the secret, H is isomorphic to K, and χ is a homomorphism from (H, λ_H) to (G, λ_G).

The labeled hypergraph (H_0, λ_0) contains the secret by the elementary properties of the splitting and querying routines.

Consider the case when branch 4a is taken. By lemma 21, there is an isomorphism $\rho : H_0 \to K$ so that $\chi\circ\rho = \chi\circ\pi$. Note that once this is known, χ is a homomorphism from $\rho(H_0)$ to G because $\chi \circ \rho = \chi \circ \pi$ is a homomorphism from (H_0, λ_0) to (G, λ_G).

Consider the case when branch 4b is taken. Because $\chi\circ\pi$ is a homomorphism from (H_0, λ_0) to (G, λ_G), $\lambda^{-1}(z) = \lambda_0^{-1}(\hat{z}) \subseteq \lambda_G^{-1}(z)$. For $x \in \{a, b, c\} \setminus \{z\}$, $\lambda^{-1}(x) = \lambda_G^{-1}(x)$.

Consider the case when branch 4c is taken. Note that $\chi \circ \rho = \chi \circ \pi$, so χ is a homomorphism from $\rho(H_0, \lambda_0)$ to (G, λ_G). By lemma 20 we can choose $a^* \in \mathrm{Split}_A(\{a, a'\})$, $b^* \in \mathrm{Split}_A(\{b, b'\})$, and $c^* \in \mathrm{Split}_A(\{c, c'\})$ so that $\{a^*, b^*, c^*\}$ is a strong cover of H_0. Because $\{a^*, b^*, c^*\}$ is a strong-cover of H_0, $\rho(\{a^*, b^*, c^*\})$ is a strong-cover of $\rho(H_0)$. Finally, because χ is a homomorphism from $\rho(H_0)$ to G, no edge of $\rho(H_0)$ contains both a and $\bar{a}$, nor does one contain both b and $\bar{b}$, nor does one contain both c and $\bar{c}$. Because $\{a, b, c, \bar{a}, \bar{b}, \bar{c}\} \subseteq V(\rho(H_0))$, $\rho(H_0)$ has no central vertex.

Proposition 17 *Consider an iteration of the loop that begins with* (H, λ_H).

1. *If the branch 4a is taken, then for some* $z \in \{a, b, c\}$ *and some* $\hat{z} \in \mathit{Split}_A(\{z\})$, *rank*$(\lambda^{-1}(z)) =$ *rank*$(\lambda_0^{-1}(\hat{z}))$. *For* $x \in \{a, b, c\} \setminus \{z\}$, *rank*$(\lambda^{-1}(x)) =$ *rank*$(\lambda_G^{-1}(x))$.

2. *If the branch 4a is taken, then for each* $z \in \{a, b, c, a', b', c'\}$, *either rank*$((\rho \circ \lambda_H)^{-1}(z)) = 1$ *or rank*$((\rho \circ \lambda_H)^{-1}(z)) <$ *rank*$(\lambda_H^{-1}(z))$.

Proposition 18 *The routine* Reduce_K$((H, \lambda_H), \lambda_G)$ *terminates within* $\sum_{v\in V(H)}$ *rank*$(\lambda_H^{-1}(v))$ *steps.*

Moreover, if, for each $z \in \{a, b, c\}$, *rank*$(\lambda_H^{-1}(z)) \leq$ *rank*$(\lambda_G^{-1}(z))$ *and rank*$(\lambda_H^{-1}(z')) \leq$ *rank*$(\lambda_G^{-1}(z))$, *and the routine exits through branch 4a after* s *iterations, then rank*$(\lambda^{-1}(a))+$*rank*$(\lambda^{-1}(b))+$*rank*$(\lambda^{-1}(c)) \leq$ *rank*$(\lambda_G^{-1}(a))+$ *rank*$(\lambda_G^{-1}(b)) +$ *rank*$(\lambda_G^{-1}(c)) - s$.

Proposition 19 $\sum_{v\in V(H)}$ *rank*$(\lambda_H^{-1}(v)) \leq \sum_{v\in V(G)}$ *rank*$(\lambda_G^{-1}(v)) + O(n)$

7 Oblivious Strategies

We convert the adaptive solution into an oblivious one by using a tool that we call a "prefix universal family". These are variants of universal families that seem to be useful for making oblivious many variations of adaptive binary searches.

Definition 7.1 *Let* $l \geq 1$ *be an integer. A* l-prefix universal family of mappings *is a family mappings,* $F_i : \{0, 1\}^n \to \{0, 1\}$, *with* $i \in I$, *so that for any collection of* $\alpha_1, \ldots, \alpha_l \in \{0, 1\}^{\leq n}$ *so that for all* $j \neq k$, $\alpha_j \not\preceq \alpha_k$, *and for all* $\vec{\epsilon} \in \{0, 1\}^l$, *there exists* $i \in I$ *so that for all* $x \in \{0, 1\}^n$, *for* $j \in \{1, \ldots, l\}$, $\alpha_j \preceq x$ *implies* $F_i(x) = \epsilon_i$.

We say that such a family is computable in time $t = t(n)$ *if given* $\alpha_1, \ldots, \alpha_l$ *and* $\epsilon_1, \ldots, \epsilon_l$, *the index* i *can be computed in time* t, *and from* $i \in I$ *and* $x \in \{0, 1\}^n$, *the value of* $F_i(x)$ *can be computed in time* t.

We say that the mappings of the family have bit-size s *if for each* $i \in I$, *the function* F_i *can be represented by* s *many bits.*

It is easy to check that the family of all decision trees with l leaves form an l-prefix universal family. The proof correctness appears in the appendix, and is based on a variant of Bondy's theorem (cf. [10]).

Theorem 22 *(proof in appendix) For positive integers* n *and* k. *The family of* k-*leaf decision trees over* $x_1, \ldots, x_n$ *with leaves labeled by* 0 *and* 1 *is a* k-*prefix universal family of size* $2^{O(k)}n^{k-1}$.

Definition 7.2 *A* decision tree *is a rooted binary tree in which every internal node is labeled with an index, i, and from each internal node, there is an outgoing arc labeled 0 and or 1, and the leaves are labeled with a value from a set V. A decision tree computes a function from $\{0,1\}^n$ to V as follows. On input x, at a node labeled i, if $x_i = 0$, we move to the subtree underneath the arc labeled 0 and compute the function of that subtree. Similarly, if $x_i = 1$, we move to the subtree underneath the arc labeled 1 and compute the function of that subtree. The value returned at leaf is the value of its label.*

Let $\alpha_1, \dots, \alpha_k \in \{0,1\}^{\leq n}$ be given. We say that a decision tree T separates *$\alpha_1, \dots, \alpha_k$ if there is unique leaf v_i associated with each α_i, so that for any $x \in \{0,1\}^n$ with $\alpha_i \preceq x$, T reaches the leaf v_i on input x.*

Theorem 23 *Let n and l be positive integers with $l \leq n$. For every set of strings $\alpha_1, \dots, \alpha_l \in \{0,1\}^{\leq n}$ so that for all $i \neq j$, $\alpha_i \not\preceq \alpha_j$, there exists a decision tree T with $l - 1$ many internal nodes that separates $\alpha_1, \dots, \alpha_l$. Moreover, the tree T can be calculated from $\alpha_1, \dots, \alpha_l$ in time $O(nl)$.*

Proof: The proof follows that of Bondy's theorem and we proceed by induction on l; the induction proof is clearly constructive and yields a recursive procedure to build the tree. When $l = 1$, a decision tree consisting of a single node suffices. Let $l > 1$ be given, and assume that the theorem holds for all $k < l$. Sort the strings $\alpha_1, \dots, \alpha_l$ by length so that α_1 is of minimum length. Choose an index i so that $(\alpha_1)_i \neq (\alpha_2)_i$. Set $A = \{\alpha_j \mid (\alpha_j)_i = (\alpha_1)_i\}$, $a = |A|$, $B = \{\alpha_j \mid (\alpha_j)_i = (\alpha_2)_i\}$ and $b = |B|$. Notice that $a \geq 1$, $b \geq 1$ and $a + b = l$. Apply the induction hypothesis for A and B to obtain trees T_A and T_B separating the sets A and B, respectively, so that T_A has $a - 1$ many internal nodes and T_B has $b - 1$ many internal nodes. Let T be a decision tree that queries x_i at the root and underneath the branch labeled $(\alpha_1)_i$, place a copy of T_A, and underneath the branch labeled $(\alpha_2)_i$, place a copy of T_B. Clearly, T separates the prefixes $\alpha_1, \dots, \alpha_l$, and T has at most $1 + a - 1 + b - 1 = l - 1$ many internal nodes. ■

Definition 7.3 *Let n and k be positive integers. Let $\mathcal{F}_{n,k}$ be the family of functions obtained by applying every possible $0/1$ labeling to the leaves of every decision tree over $x_1, \dots, x_n$ with $k - 1$ internal nodes.*

As a corollary of theorem 23, the family $\mathcal{F}_{n,k}$ is a k-prefix universal family, and by the well-known bound on the number of binary trees with $k - 1$ internal nodes via the Catalan numbers, we can bound its size.

Proposition 20 *For positive integers n and k, $\mathcal{F}_{n,k}$ is a k prefix universal family, and $|\mathcal{F}_{n,k}| = \Theta(2^{2k} n^{k-1})$.*

The proof of Theorem 22 gives a mapping from $(\vec{\alpha}, \vec{\epsilon})$ to an $F \in \mathcal{F}_{n,k}$ so that for all $x \in \{0,1\}^n$ can be computed in time $O(kn)$.

Our construction of a prefix universal family is substantially larger than the best constructions of a universal family; for constant k, the difference is between $\Theta(n^{k-1})$ and $\Theta(n)$ (with the big-O notation hiding constant factors dependent on k). The following lemma shows that this is necessary, that prefix universal families necessarily have size $n^{\Omega(k)}$. However, this lower bound does not match our upper bound, and it is an interesting question whether or not the lower bound can be improved to $\Omega(n^{k-1})$.

Lemma 24 *Let $\mathcal{F}$ be a k-prefix universal family for strings of length n, with $n > \lceil \log_2 k \rceil$. $|\mathcal{F}| = \Omega(2^{\lfloor k/2 \rfloor} n^{\lfloor k/2 \rfloor})$.*

Proof: Set $k_0 = \lfloor k/2 \rfloor$. Let $\gamma_1, \dots, \gamma_k$ be distinct strings of length $\lceil \log_2 k \rceil$. Let $\vec{i}, \vec{l} \in \{1, \dots n - \lceil \log_2 k \rceil - 1\}^{k_0}$, and $\vec{\epsilon}, \vec{\delta} \in \{0,1\}^{k_0}$ be given so that $\vec{i} \neq \vec{l}$ or $\vec{\epsilon} \neq \vec{\delta}$. We will show that for any $F \in \mathcal{F}$ it is impossible that $F(x) = \epsilon_j$, for $x \in \{0,1\}^n$ so that $\gamma_j 0^{i_j} 0 \preceq x$, $F(x) = 1 - \epsilon_j$, for $x \in \{0,1\}^n$ so that $\gamma_j 0^{i_j} 1 \preceq x$, $F(x) = \delta_j$, for $x \in \{0,1\}^n$ so that $\gamma_j 0^{l_j} 0 \preceq x$, and $F(x) = 1 - \delta_j$, for $x \in \{0,1\}^n$ so that $\gamma_j 0^{l_j} 1 \preceq x$.

Suppose that $i_j < l_j$ for some $j \in [k_0]$. Choose $x, y \in \{0,1\}^n$ with $\gamma_j 0^{l_j} 0 \preceq x$ and $\gamma_j 0^{l_j} 1 \preceq y$. Because $\gamma_j 0^{i_j} 0 \preceq \gamma_j 0^{l_j} 0 \preceq x$ and $\gamma_j 0^{i_j} 0 \preceq \gamma_j 0^{l_j} 1 \preceq y$, we have that $\delta_j = F(x) = \epsilon_j = F(y) = 1 - \delta_j$; contradiction. The case when $i_j > l_j$ for some $j \in [k_0]$ is similar. Suppose that $\vec{i} = \vec{l}$ but for some $j \in [k_0]$, $\epsilon_j \neq \delta_j$. Clearly, F cannot map every x with $\gamma_j 0^{i_j} 0 \preceq x$ to both δ_j and ϵ_j.

■

The proof of Theorem 22 is constructive: It gives a polynomial-time mapping from $(\vec{\alpha}, \vec{\epsilon})$ to a k-leaf decision tree F so that for all $x \in \{0,1\}^n$, $\alpha_i \preceq x \Rightarrow F(x) = \epsilon_i$. For each $\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3 \in \{0,1\}^{\leq n}$, let

$Q(\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3)$ denote the function computed by the decision tree which gives $F(x) = 1$ whenever α_1, α_2, or α_3 is a prefix of x, $F(x) = 0$ whenever β_1, β_2, or β_3 is a prefix of x.

Lemma 25 *Let (G, λ_G) be a labeled hypergraph so that $S^* \in \lambda_G^{-1}(G)$.*

Let $e \in E(G)$ be so that $S^* \in \lambda_G^{-1}(e)$. There exists $\alpha_1, \alpha_2, \alpha_3 \in \bigcup_{v \in e} \mathsf{label}(\lambda_G)(v)$ so that for all $f \in E(G)$, if $e \cap f = \emptyset$ then for all $\beta_1, \beta_2, \beta_3 \in \bigcup_{v \in e} \mathsf{label}(\lambda_G)(v)$, the adversary's answer to $Q(\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3)$ is 1.

Let $f \in E(G)$ be so that there exists $e \in E(G)$ with $e \cap f = \emptyset$ and $S^* \in \lambda_G^{-1}(e)$. Then for all $\alpha_1, \alpha_2, \alpha_3 \in \bigcup_{v \in f} \mathsf{label}(\lambda_G)(v)$ there exists $\beta_1, \beta_2, \beta_3 \in \bigcup_{v \in e} \mathsf{label}(\lambda_G)(v)$ so that the adversary's answer to $Q(\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3)$ is 0.

Proof: Let $S^* = \{s_1, s_2, s_3\}$ and let $e = \lambda_G(S^*)$. Choose $\alpha_1, \alpha_2, \alpha_3 \in \bigcup_{v \in e} \mathsf{label}(\lambda_G)(v)$ so that $\alpha_1 \preceq s_1$, $\alpha_2 \preceq s_2$, and $\alpha_3 \preceq s_3$. Consider an edge f with $e \cap f = \emptyset$. Let $\beta_1, \beta_2, \beta_3 \in \bigcup_{v \in f} \mathsf{label}(\lambda_G)(v)$ be given. Note that the collection of strings $\{\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3\}$ is prefix-free. Choose $F \in \mathcal{F}$ so that $F(x) = 1$ for all $x \in \{0,1\}^n$ with $\alpha_1 \preceq x$, $\alpha_2 \preceq x$, or $\alpha_3 \preceq x$, and $F(x) = 0$ for all $x \in \{0,1\}^n$ with $\beta_1 \preceq x$, $\beta_2 \preceq x$, or $\beta_3 \preceq x$. Therefore, $F(s_1) = F(s_2) = F(s_3) = 1$ and the adversary must answer "yes".

The second statement follows from the first, using the inference if $\exists x \in e\ \forall y \in f\ \psi(x, y)$ then $\forall y \in f\ \exists x \in e\ \psi(x, y)$. ■

The guessing secrets routine is made oblivious by initially making queries in the family of 6-leaf decision trees and then simulating the adaptive strategy, but using the following routine instead of the routine query: For each $e \in E(G)$, if there exists $\alpha_1, \alpha_2, \alpha_3 \in \bigcup_{v \in e} \mathsf{label}(\lambda_G(v))$ so that for all $f \in E(G)$, with $e \cap f = \emptyset$, for all $\beta_1, \ldots, \beta_3 \in \bigcup_{u \in f} \mathsf{label}(\lambda_G(u))$, $Q(\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3) = 1$, then delete all $f \in E(G)$ so that $e \cap f = \emptyset$, otherwise, delete e.

Clearly, a call to this subroutine creates an intersecting hypergraph. Moreover, Lemma 25 guarantees that the edge containing the secret triple survives. Each call to this subroutine requires time $O(poly(n)s^6)$ where s is the maximum size of a label.

8 Structural Results for Three-uniform, Intersecting Hypergraphs

8.1 Simple Multi-edge Cores

Theorem 26 *Let H_2 be a three-uniform, simple intersecting hypergraph with core H_1. Then, either $H_1 \cong \{abc\}$, or for any $e, f \in E(H_2)$, $e \cap f \cap V(H_1) \neq \emptyset$.*

Proof: Let H_2 be a three-uniform, simple intersecting hypergraph with core H_1. Let $V = V(H_1)$. We prove that if there exist hyperedges $e, f \in E(H_2)$ such that $e \cap f \cap V = \emptyset$, then H_1 has two central vertices. It then follows (by simplicity and Lemma 7 in the appendix) that $H_1 \cong \{abc\}$.

Let $e, f \in E(H_2)$ be such that $e \cap f \cap V = \emptyset$, and let $\phi \colon H_2 \to H_1$ be a homomorphism such that $\phi \restriction_V = Id_V$. Since ϕ is a homomorphism, we have $\phi(e), \phi(f) \in E(H_1) \subseteq E(H_2)$. From the intersecting property of H_2, we know that $e \cap f$, $\phi(e) \cap f$ and $e \cap \phi(f)$ are all nonempty. Choose $x \in e \cap f$, $y \in \phi(e) \cap f$, and $z \in \phi(f) \cap e$. Choose $y_0 \in e$, $z_0 \in f$ so that $\phi(y_0) = y$ and $\phi(z_0) = z$. Note that $x \notin V$ by the hypothesis that $e \cap f \cap V = \emptyset$. Also, $y_0 \notin V$ because if $y_0 \in V$ then $y_0 = \phi(y_0) = y \in f$, and therefore $y \in e \cap f \cap V = \emptyset$. Similarly, $z_0 \notin V$. Now we show that $x \neq y_0$: if $x = y_0$ then $y_0, y \in e$ so y appears in $\phi(e)$ with multiplicity at least two, in contradiction to simplicity. Similarly $x \neq z_0$.

Since H_2 is three-uniform, this implies that $e = \{x, y_0, z\}$ and $f = \{x, y, z_0\}$. Because $H_1 \subseteq H_2$ and H_2 is intersecting, y and z are central vertices of H_1. Finally, $y \neq z$ because then $y \in e \cap f \cap V$. Therefore, H_1 has two distinct central vertices.

■

Definition 8.1 *Let $H_1 \subseteq H_2$ be two hypergraphs. The* residual family of H_1 and H_2 *is the graph* $res(H_1, H_2) = \{e \cap V(H_1) \mid e \in E(H_2) \setminus E(H_1)\}$.

Because each $X \in \text{res}(H_1, H_2)$ has size 1 or 2, we have the following corollary of Theorem 26:

Corollary 27 *For any 3-uniform simple intersecting hypergraph H_2 with core $H_1 = H_2[V] \subsetneq H_2$, either res$(H_1, H_2)$ has a central vertex, or res(H_1, H_2) is a triangle, i.e., res$(H_1, H_2) \cong \{ab, bc, ca\}$.*

As a corollary to theorem 26, we obtain a short proof of a somewhat novel and non-obvious fact. Of course, this result could be also be shown by exhaustive search because by well-known bound of Erdös and Lovasz there are at most 27 edges in a three-uniform, intersecting hypergraph H with $\tau(H) = 3$.

Corollary 28 *Let H be a three-uniform, intersecting hypergraph. If $\tau(H) = 3$ then H is a core.*

Proof: Let $G \subseteq H$ be a core of H. Because $\tau(H) = 3$, H is simple, and by corollary 33, the core of H contains at least two edges. Therefore, by theorem 26, each set in res(G, H) is a hitting set for H and has size at most two. However, $\tau(H) = 3$ so we must have that res$(G, H) = \emptyset$. In other words, $G = H$ and H is a core. ■

We now use Theorem 26 to prove Theorem 13.

Proof: First, consider the case when res(H_1, H_2) has a central vertex. Let w be the central vertex of res(H_1, H_2), and let $S = \{w\}$. Clearly for all $h_0, h_1 \in E(H_2) \setminus E(H_1)$, $w \in h_0 \cap h_1$, so property (ii) holds. We now show that property (i) holds: Take any $e \in E(H_1)$ with $w \in e$, let f be any $f \in E(H_2)$ so that $\phi(f) = e$. Suppose for the sake of contradiction that $w \notin f$. Because $\phi \restriction_{H_1} = \text{Id}_{H_1}$, $f \notin E(H_1)$, and therefore $f \cap V(H_1) \in \text{res}(H_1, H_2)$, so $w \in f$, contradiction.

Now consider the case when res(H_1, H_2) does not have a central vertex. Choose $u, v, w \in V(H_1)$ so that res$(H_1, H_2) = \{\{u, v\}, \{u, w\}, \{v, w\}\}$. Let $S = \{u, v, w\}$. For $h_0, h_1 \in E(H_2) \setminus E(H_1)$, $|h_0 \cap S| = |h_1 \cap S| = 2$, so $h_0 \cap h_1 \cap S \neq \emptyset$ and (ii) holds. We now prove property (i): Suppose for the sake of contradiction that for every $e \in E(H_1)$ there is $f \in E(H_2)$ with $\phi(f) = e$ and $u \notin e$ (the cases for v and w are the same up to renaming). Let $e \in E(H_1)$ with $u \in e$ be given. Choose $f \in E(H_2)$ with $\phi(f) = e$ and $u \notin e$. Because $f \cap V(H_1) \in \{\{u, v\}, \{u, w\}, \{v, w\}\}$, $\{v, w\} \subseteq f$, and therefore $e = \{u, v, w\}$. That is, $\forall e \in E(H_1)$ with $u \in e$, $e = \{u, v, w\}$. Because each $f \in E(H_1)$ with $u \notin f$ must intersect both $\{u, v\}$ and $\{u, w\}$, every $f \in E(H_1)$ has the form $\{v, w, x\}$. Therefore H_1 homomorphically embeds into $\{u, v, w\}$ via the mapping $v \mapsto v$, $w \mapsto w$, and $x \mapsto u$ if $x \neq v, w$. This contradicts the hypothesis that H_1 is not isomorphic to $\{abc\}$, so property (i) must hold.

Consider the case when G is not simple and res(G, H) has no central vertex. There are two cases to consider, when G is isomorphic to $\{abb, aab\}$ and when G is not isomorphic to $\{abb, aab\}$.

Consider the case when G is isomorphic to $\{abb, aab\}$. If there is no $h \in E(H)$ with $\phi(h) = \{a, a, b\}$ and $a \notin h$, then we may take $S = \{a\}$. Otherwise, choose such and h and let $S = \{b\}$. Suppose for the sake of contradiction that there is $f \in E(H)$ with $b \notin f$ and $\phi(f) = \{a, a, b\}$. Because $\{a, a, b\} \cap h \neq \emptyset$, $h \cap V(G) = \{b\}$, and because $\phi(h) = \{a, a, b\}$, for all $x \in h \setminus \{b\}$, $\phi(x) = a$. Similarly, $f \cap V(G) = \{a\}$, and for all $y \in f \setminus \{a\}$, $\phi(y) = b$. Therefore, $h \cap f = \emptyset$, contradction to H being intersecting.

In the case when G is not isomorphic to $\{abb, aab\}$, because G is not isomorphic to $\{abb\}$ by hypothesis, we may conclude that G has at most one central vertex. Choose $a \in V$ with $m_G(a) = 2$, and let $S = \{a\}$. Choose $g \in E(G)$ with $g = \{a, a, b\}$ (up to renaming some element to "b"). Suppose for the sake of contradiction that there is $h \in E(H) \setminus E(G)$ so that $a \notin h$ and $\phi(h) = g$. Because $g \cap h \neq \emptyset$, $b \in h$. Because $\phi(h) = \{a, a, b\}$ and $\phi \restriction_G = \text{Id}_G$, $h \cap V(G) = \{b\}$. Therefore, b is a central vertex of G. Because H does not have a central vertex, we may choose $f \in E(H)$ with $b \notin f$. Because $f \cap g \neq \emptyset$, $a \in f$. Because $\phi(f) \in E(G)$, we may choose $b' \in f$ so that $\phi(b') = b$. Note that $b' \notin h$, so there is $a' \in f \cap h$ with $\phi(a') = a$ and $a' \notin V(G)$. That is, $f = \{a, a', b'\}$. However, because $G \subseteq H$ and H is intersecting, this makes a a central vertex of G. Thus a and b are distinct central vertices of G, contradiction.

■

8.2 Hypergraphs Mutually Homomorphic with abc

In this subsection we classify the three-uniform, intersecting hypergraphs that homomorphically embed into a single edge.

Lemma 29 *Let H be a hypergraph. If $\tau^*(H_0) \leq 2$, then $\tau(H_0) = 1$.*

Proof: Choose S to be a strong-cover of C. For every $e \in E(H_0)$, $S \subseteq e$, and therefore each vertex of S is a central vertex of H. ■

Definition 8.2 *Let K denote the hypergraph $\{abc, ab'c', a'bc', a'b'c\}$.*

Proposition 21 *For any homomorphism $\phi : K \to \{abc\}$, $\phi(a) = \phi(a')$, $\phi(b) = \phi(b')$, and $\phi(c) = \phi(c')$.*

Lemma 30 *Let e be any edge of size three that intersects every edge of K (possibly including vertices not in K). If $e \notin K$ then $K \cup \{e\}$ does not embed into $\{abc\}$.*

Proof: Choose $x \in e \cap \{a, b, c\}$ and rename the vertices so that $x = a$. Choose $y \in e \cap \{a', b, c'\}$ and rename the vertices b,b',c,c' so that $y = b$. Because $e \cap \{a', b', c\} \neq \emptyset$ and $e \neq \{a, b, c\}$, $e = \{a, a', b\}$ or $\{b, b', c\}$. However, this cannot embed into $\{a, b, c\}$ because $\phi(a) = \phi(a')$ and $\phi(b) = \phi(b')$ whenever ϕ is a homomorphism from K to $\{abc\}$. ■

Lemma 31 *Let H be a three-uniform, intersecting hypergraph that embeds into $\{abc\}$. If $\tau(H) > 1$ and $\tau^*(H) > 3$, then H is isomorphic to the hypergraph K.*

Proof: Choose an edge $e \in H$ and rename its vertices so that $e = \{a, b, c\}$. Because $\{a, b, c\}$ is not a strong cover of H, we may choose $f \in H$ so that $e \cap f = \{a\}$ (up to renaming). Rename the vertices of H so that $f = \{a, b', c'\}$. Because a is not a central vertex of H, we may choose an edge g so that $a \notin g$. Because g intersects both e and f, up to renaming, $g = \{a', b, c'\}$. Because $\{a, b, c'\}$ is not a strong-cover of H, there is $h \in H$ with $|h \cap \{a, b, c'\}| \leq 1$. If $a \in h$, then because h must intersect $\{a', b, c'\}$, we would have $a' \in h$, which contradicts the hypothesis that H embeds into $\{a, b, c\}$. Similarly, $b \notin h$ and $c' \notin h$. Therefore, because $h \cap \{a, b', c'\} \neq \emptyset$, $b' \in h$, because $h \cap \{a, b, c\} \neq \emptyset$, $c \in h$, and because $h \cap \{a', b, c'\} \neq \emptyset$, $a' \in h$, and therefore $h = \{a', b', c\}$.

Therefore, up to renaming vertices, $K \subseteq H$. Because H embeds into $\{abc\}$, by lemma 30, H is isomorphic to K. ■

Corollary 32 *Let H be a three-uniform, intersecting hypergraph that embeds into $\{abc\}$. H either has a central vertex, has a strong cover of size three, or is isomorphic with K.*

Corollary 33 *If H is a three-uniform, intersecting hypergraph that embeds into $\{abc\}$ then $\tau(H) \leq 2$.*

9 Connections Between Guessing Secrets and Learning a Hidden Subgraph

The problem of "learning a hidden subgraph" [2] is a search for an unknown graph on N vertices that proceeds as follows: Let $\mathcal{H}$ be a family of (hyper) graphs on the vertex set $V = \{1, \ldots, N\}$ that is closed under isomorphism (isomorphic graphs on distinct vertex sets are considered distinct). An adversary has a graph $H \in \mathcal{H}$ and a seeker would like to learn the graph using queries of the form "does the set S contain an an edge of H?". The adversary answers "yes" if there exists $e \in E(H)$ so that $e \subseteq S$ and "no" if $\forall e \in E(H)$, $e \not\subseteq S$.

Problems of this form first arose in computational biology, and typically, the set $\mathcal{H}$ is a natural class of graphs such as stars, cliques, matchings or Hamiltonian paths. The restriction that the hidden graph come from a limited family of labeled graphs allows for non-trivial algorithms – if the search is among the $2^{\binom{N}{2}}$ labeled graphs on N vertices then clearly $\binom{N}{2}$ queries are necessary! Earlier work has focused on on distinguishing the hidden subgraphs with a minimum number of queries, and the efficiency of recovery was typically the obvious "maintain a graph and delete edges as the queries are processed" method which uses time and space $O(N^2)$. We will show that by using the efficient solutions for guessing k secrets as a black box, some classes of hidden subgraphs can be learned with $\log^{O(1)}(N)$ queries and recovered in $\log^{O(1)}(N)$ time. (We do not claim that these speed-ups are relevant for applications to computational biology, only that this algorithmic improvement for a natural combinatorial problem is theoretically interesting.)

We consider the generalization of this problem to k-uniform hypergraphs, with the specialization that we are trying to learn a hidden maximal intersecting sub-hypergraph. By interpreting the problem of learning a hidden subgraph as a guessing secrets game, there are extremely efficient methods for learning a maximal intersecting subgraph.

Definition 9.1 *Let $\mathcal{I}_{N,k}$ denote the set of all maximal, intersecting, k-uniform hypergraphs on the vertex set $\{1, \ldots, N\}$.*

Theorem 34 *If there is a strategy that solves the guessing k-secrets problem on n bit strings in time $t(n)$ and space $s(n)$ making $q(n)$ queries of size $r(n)$, then there is a strategy that learns a hidden graph of the family $\mathcal{I}_N$ on an N vertex set in time $t(\log N)$, space $s(\log N)$, making $q(\log N)$ queries of size $r(\log N)$.*

Proof: View the vertex set $[N]$ as the set of all $\log N$ bit strings. Simulate the guessing secrets strategy and each time the strategy makes a query of the form "does the string belong to S?", make the query "does the set S induce an edge?". Return the answer of the adversary to the simulated guessing secrets strategy. The seeker's answer is the hypergraph returned by the guessing secrets routine.

Clearly, this routine runs in time $t(\log N)$ and space $s(\log N)$, and makes $q(\log N)$ queries of size $r(\log N)$. What we need to show is that the sequence of answers given to the guessing secrets routine are legal answers for some fixed k-set of strings, and that the intersecting hypergraph produced by the guessing secrets routine is indeed the adversary's hidden subgraph.

Let $H \in \mathcal{I}_{N,k}$ be the adversary's hidden hypergraph. Let $e \in E(H)$ be given. First we show that, the sequence of answers for the questions "does S induce an edge of H?" are valid answers for the questions in the guessing secrets game when e is the secret k-set of edges. This is because if S does induce an edge of H, then $e \cap S \neq \emptyset$ (and "yes" is a legal answer), and if S does not induce an edge, then $e \not\subseteq S$ (and "no" is a legal answer). Therefore, the guessing secrets routine returns a k-uniform intersecting hypergraph H' so that $e \in E(H')$. Because $e \in E(H)$ was arbitrary, $H \subseteq H'$. Because H' is intersecting and H is maximal intersecting, $H = H'$. ■

As a corollary, the methods of [3, 7] give very space efficient oblivious strategies for learning a hidden maximal intersecting subgraph.

Corollary 35 *There is an oblivious strategy for learning a hidden subgraph from $\mathcal{I}_{N,2}$ that makes $O(\log N)$ queries, each of size $O(\log N)$, and the hidden graph can be reconstructed using time and space $O(\log^{O(1)} N)$.*

There is an oblivious strategy for learning a hidden subgraph from $\mathcal{I}_{N,2}$ that makes $O(\log^3 N)$ queries, each of size $O(\log \log N)$, and the hidden graph can be reconstructed using time and space $O(\log^{O(1)} N)$.

Our results on guessing three secrets show similar results for learning a hidden maximal intersecting sub-hypergraph for the three-uniform case.

Corollary 36 *There is an adaptive strategy for learning a hidden sub-hypergraph from $\mathcal{I}_{N,3}$ that makes $O(\log N)$ queries, each of size $O(\log \log N)$, and the hidden graph can be reconstructed using time and space $O(\log^{O(1)} N)$.*

There is an oblivious strategy for learning a hidden sub-hypergraph from $\mathcal{I}_{N,3}$ that makes $O(\log^5 N)$ queries, each of size $O(\log \log N)$, and the hidden graph can be reconstructed using time and space $O(\log^{O(1)} N)$.

10 Conclusion

We have presented the first polynomial time algorithm for guessing three secrets. This is the first solution of the guessing k secrets problem for $k > 2$, and it overcomes the difficulty of recovering the secrets after asking non-linear queries, which presented a barrier to earlier solutions.

A few weeks after this work was presented at the Discrete Mathematics and Computer Science Seminar of the Institute for Advanced Study, Alexander Razborov found a solution for the guessing k-secrets problem which makes $2^{O(k)}n$ oblivious queries and recovers the secrets in time $2^{O(k^2)}n^2$ [18].

In light of this sudden improvement of our central result, we consider the central contribution of our work to be the techniques developed. Two ideas that seem promising are the use of the homomorphism partial order to guarantee termination for a search procedure, and the use of prefix universal families to convert adaptive queries into oblivious ones. Some of the properties of three-uniform intersecting hypergraph cores may also be of interest for combinatorial purposes.

An open problem from this work is the asymptotic size of k-prefix universal families for n bit strings. The upper bound of Theorem 22 gives a family of size $2^{O(k)}n^{k-1})$, and an easy lower bound shows that a k-prefix universal family must have size at least $2^{\lfloor k/2 \rfloor}n^{\lfloor k/2 \rfloor}$ (Lemma 24 of the appendix). Do there exist such families of size $O(n^{k/2})$ or do all k-prefix universal families for n bit strings require size $\Omega(n^{k-1})$?

We would like to thank Ron Graham and Fan Chung for introducing us to this problem. Nate would like to thank Ranjita Bhagwan, Matt Devos, and Russell Impagliazzo for helpful conversations. In particular, after learning of our complicated first construction of prefix universal families, Russell suggested that we consider "decision trees of height k", which is very similar to what we ended up using (decision trees with k leaves). Subhash Khot provided comments on an early draft of this paper.

References

[1] N. Alon. Explicit construction of exponential sized families of k-independent sets. *Discrete Mathematics*, 58:191–193, 1986.

[2] N. Alon and V. Asodi. Learning a hidden subgraph. Submitted, 2003.

[3] N. Alon, T. Kaufman, V. Guruswami, and M. Sudan. Guessing secrets efficiently via list decoding. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2002.

[4] M. Chein and M. L. Mugnier. Conceptual graphs are also graphs. Technical report, LIRMM, 1995.

[5] M. Chein, M. L. Mugnier, and G. Simonet. Nested graphs: a graph-based knowledge representation model with FOL semantics. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning*, pages 524–535, 1998.

[6] F. Chung, R. Graham, and T. Leighton. Guessing secrets. *The Electronic Journal of Combinatorics*, 8, 2001.

[7] F. Chung, R. Graham, and L. Lu. Guessing secrets with inner product questions. In *Proceedings of the Thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 247–253, 2002.

[8] G. Hahn and C. Tardif. *Graph homomorphisms: structure and symmetry*, pages 107 – 166. Kluwer, 1997.

[9] P. Hell and J. Nešetřil. The core of a graph. *Discrete Mathematics*, 109:117–126, 1992.

[10] S. Jukna. *Extremal Combinatorics: with applications to computer science*. Springer-Verlag, 2001.

[11] M. L. Mugnier. *Knowledge Representation and Reasonings Based on Graph Homomorphism*, volume 1867 of *Lecture Notes in Artificial Intelligence*, pages 172–192. Springer, 2000.

[12] M. Nathanson. Quantum guessing via Deutsch-Jozsa. Manuscript available at http://arxiv.org/abs/quant-ph/0301025, 2003.

[13] J. Nešetřil. The homomorphism structure of classes of graphs. *Combinatorics, Probability and Computing*, 8:177–184, 1999.

[14] J. Nešetřil. Structural combinatorics (graph homomorphisms and their use). *Taiwanese Journal of Mathematics*, 3:381–423, 1999.

[15] J. Nešetřil. Combinatorics of mappings (graph homomorphisms and their use). Available at http://citeseer.nj.nec.com/407260.html, 2000.

[16] J. Nešetřil and X. Zhu. Path homomorphisms. *Proceedings of the Cambridge Philosophical Society*, 120:207–220, 1996.

[17] I. Peterson. Guessing secrets. *Science News*, 161(14), April 2002.

[18] A. Razborov. Guessing more secrets via list decoding. Unpublished manuscript, 2004.

[19] C. Tardif. Fractional multiples of graphs and the density of vertex transitive graphs. *Journal of Algebraic Combinatorics*, 10:61–68, 1999.

[20] E. Welzl. Symmetric graphs and interpretations. *Journal of Combinatorial Theory Series B*, 37:235–244, 1982.

A The Hypergraph Poset and Hypergraph Cores

An easy application of the density result for graph homomorphisms provides an infinite descending sequence of three-uniform, intersecting hypergraph cores.

Theorem 37 *([20, 13], c.f. [14]) For any two* graphs *G_1 and G_2 so that $G_1 < G_2$ and G_1 contains at least two vertices, there is a third graph G so that $G_1 < G < G_2$.*

Lemma 38 *There exists an infinite sequence $H_1, H_2 \ldots$ of three-uniform, intersecting hypergraph cores so that for every i, $H_i > H_{i+1}$.*

Proof: Apply theorem 37 and choose an infinite sequence $G_1, G_2, \ldots$ of graph cores so that for every i, $G_i > G_{i+1}$ and each G_i contains multiple edges.

For a (two-uniform) graph G, let $\widehat{G}$ be the three-uniform hypergraph obtained by taking a vertex $v \notin V(G)$, and setting $V(\widehat{G}) = V(G) \cup \{v\}$ and $E(\widehat{G}) = \{\{v, x, y\} \mid \{x, y\} \in E(G)\}$.

In the next paragraph we show that there is a homomorphism from G to H if and only if there is a homomorphism from $\widehat{G}$ to $\widehat{H}$. This will prove our lemma when we take the cores of the hypergraphs $\widehat{G_1}, \widehat{G_2}, \ldots$

Clearly, if there is a homomorphism from G to H then there is a homomorphism $\widehat{G}$ to $\widehat{H}$. Suppose that ϕ is a homomorphism from $\widehat{G}$ to $\widehat{H}$. Choose $v \in V(\widehat{G}) \setminus V(G)$ and $u \in V(\widehat{H}) \setminus V(H)$. If $\phi(v) = u$, then for every $\{x, y\} \in E(G)$, $\phi(\{v, x, y\}) = \{u, a, b\}$ with $\{a, b\} \in E(H)$, so there is a homomorphism from G to H. On the other hand, if $\phi(v) \neq u$ then we have that for every $x \in V(G)$, $\phi(x) \neq \phi(v)$ and for every $\{v, x, y\} \in E(\widehat{G})$, $\phi(x) \neq \phi(y)$ and either $\phi(x) = u$ or $\phi(y) = u$. Consider the mapping ψ with $\psi(v) = u$, and for $x \in V(G)$, $\psi(x) = \phi(v)$ if $\phi(x) = u$ and $\psi(x) = \phi(x)$ otherwise. For $\{v, x, y\} \in E(\widehat{G})$, we have that if $\phi(x) = u$ then $\psi(\{v, x, y\}) = \{u, \phi(v), \phi(y)\} \in E(\widehat{H})$ and if $\phi(y) = u$ then $\psi(\{v, x, y\}) = \{u, \phi(x), \phi(v)\} \in E(\widehat{H})$. Therefore, ψ is a homomorphism from $\widehat{G}$ to $\widehat{H}$ with $\psi(v) = u$. Therefore G homomorphically embeds into H. ■