Outline:

1. Formally define strand spaces / infiltrated skeletons
2. Write as a geometric theory
3. Partition the rules into soundness and completeness rules
4. Give inference rules (normalisation and efficiency and chaining -vs.- cuts)
5. Prove their equivalence to the completeness rules; show that each is redundant with respect to the other.
6. Find circumstances under which the chase fully terminates.

Works so far:

1. Geometric Logic
2. The Chase
3. Normalization, Efficiency, Chaining

Work to do:

1. CPSA algorithm
2. Translating protocol specs into geometric logic
3. Extensions

1 Motivation

What are the advantages of geometric logic; why bother performing cryptographic protocol analysis in this new domain?

Foremost, cryptographic protocol analysis is traditionally implemented in imperitive programming languages. Switching to a declarative style has the advantage of making correctness easier to verify, and the system easier to reason about in general. First order logic is also a natural way to express the definitions of the formalism used.

Second, a geometric theory is both flexible and extensible. Any extension expressible in geometric logic can be added trivially; the only implementation concern is the priority of the rules.

Third, the chase can perhaps shed some light on the working of existing algorithms. It seems to mimic them in a way; there is perhaps a close relation between the chase's search for minimal theories and the algorithms' search for representative shapes.

2 Geometric Logic

2.1 Definitions

Signature A set of constant symbols c, function symbols f with arity, and relation symbols R with arity.

Model (over a signature) Contains a non-empty *universe* from which a relation is assigned to each relation symbol, a function to each function symbol, and an element to each constant symbol.

Term a (over a signature) The closure of function symbols over constants and variables.

Homomorphism h (over a signature, from one model A to another B) A function from the universe of A to that of B, obeying:

$$\begin{array}{rcl} h(c_A) & = & c_B \\ h(f_A(a_1, ...a_n)) & = & f_B(h(a_1), ...h(a_n)) \\ R_A(a_1, ...a_n) & \Rightarrow & R_B(h(a_1), ...h(a_n)) \end{array}$$

Positive Existential Formula (over a signature) The closure of $\exists$, $\bigwedge$, and infinitary $\bigvee$ over terms, $\bot$, and $\top$.

Geometric Sequent $\phi_1 \Rightarrow \phi_2$, where ϕ_1 and ϕ_2 are positive existential formulas. Written $\phi_1 \vdash \phi_2$, with $\forall$ suppressed and ',' for 'and'.

Geometric Theory A finite conjunction of geometric sequents. Without loss of generality, they can be written in the form

$$\forall \vec{x} \; (D(\vec{x}) \;\vdash\; \bigvee_i (\exists \vec{y_i} \; . \; E_i(\vec{x}, \vec{y_i})))$$

where D and E_i are conjunctions of atomic formulas. [There must be better terminology than "strictly geom" and "weakly geom".]

A relation is *geometric*, or *strictly geometric*, with respect to a geometric theory if it can be expressed by adding sequents to the theory. It is *weakly geometric* if it can be expressed by expanding the signature as well as adding new sequents.

Inductive Definition An inductive definition for a relation is a positive existential formula, lacking ∀, and with only finite disjunctions, which may include the relation. It may always be written in the form [check]:

$$R(\vec{x}) \equiv \bigvee_i \exists \vec{y} \; \bigwedge_j \phi_{ij}(\vec{x}, \vec{y})$$

where $\phi_i j$ is atomic and may contain '=' or R.

2.2 Some Tricks in GL

Negating a Sequent To negate a sequent,

$$\phi(\vec{x}) \;\vdash\; \bigvee_i \exists \vec{y} \; \psi_i(\vec{x}, \vec{y}, \vec{z})$$

introduce constants $\vec{x_0}$ and $\vec{z_0}$. Force the LHS to be true over $\vec{x_0}$,

$$\vdash \; \phi(\vec{x_0})$$

Then force each disjunct to be false over $\vec{x_0}$ and $\vec{z_0}$,

$$\psi_i(\vec{x_0}, \vec{y}, \vec{z_0}) \;\vdash$$

Replacing Functions with Relations and Equations Create a relation symbol F for each function symbol f. Add a sequent $F(\vec{(x)}, a), F(\vec{(x)}, b) \;\vdash$ $a = b$. Then replace each sequent

$$\phi(f(\vec{x})) \;\vdash\; \bigvee_i \exists \vec{y} \; \psi(f(\vec{x}, \vec{y}, \vec{z}))$$

with

$$F(\vec{x}, a), \phi(a) \ \vdash \ \bigvee_i \exists \vec{y}\, \exists b\, F(\vec{x}, \vec{y}, \vec{z}, b), \psi(b)$$

Replacing Constants with Functions and Equality For each constant symbol c, create a function symbol C, add the sequent $C(x), C(y) \ \vdash \ x = y$, and replace each occurrence of c with C.

Replacing Equality with a Relation Create a relation E. Make it symmetric and transitive,

$$\begin{aligned} E(x, y) &\ \vdash \ E(y, x) \\ E(x, y), E(y, z) &\ \vdash \ E(x, z) \end{aligned}$$

Then add sequents of the form

$$R(x), E(x, y) \ \vdash \ R(y)$$

[Is this all that's neccessary?]

Replacing Existential Quantifiers with Functions (Skolemization) $\exists x\ R(x, y)$ becomes $R(x, f(x)$

2.3 Theorems

Theorem 1. *Inductive definitions are strictly geometric.*

Expand the inductive definition as an infinite disjunction (first the base cases, then the cases with one inductive step, etc.). Express this as the right half of a geometric sequent.

Theorem 2. *A geometric theory is satisfiable iff there is a run of the chase which does not fail.*

First, see that if a geometric theory is satisfiable then there is a run of the chase which does not fail. Let M be one of the models satisfying the theory. Begin with an empty chase structure and a map from its variables to variables in the model. As long as the chase has not terminated, pick an unsatisfied sequent. One of the disjuncts of the sequent, interpreted using the variable map, must be true in the model. Expand the chase structure to make this disjunct true, adding to the variable map as necessary. (But this doesn't prove there is a *computable* run of the chase which does not fail!)

Next, if a geometric theory is unsatisfiable then every run of the chase fails. Clearly no run of the chase could succeed, because then it would produce a model satisfying the theory. Nor could the chase run forever, because then the theory would be satisfied by the model formed by taking the union of the structures formed by the chase at each iteration (more detail?).

Theorem 3. *Let T be geometric. For any model M of T there is a run of the chase that yields a model N of T such that there is a homomorphism from N to M.*

See above. The variable map is the homomorphism.

Theorem 4 (?)**.** *Suppose T is geometric and can be expressed with no branching on the right hand side of sequents. If T is satisfiable then no run of the chase fails, and any result is a universal model.*

As before, pick a model and use it as an oracle for the chase. Notice that this time, there are no choices between disjuncts; only between sequents. [...?]

3 Homomorphisms

A *graph* is a set of *vertices* along with a set of unordered pairs of distinct vertices called *edges*. A *digraph* is like a graph, but it's edges are ordered pairs. A *relational structure*, or just *structure*, has a set of *vertices*, a set of *relations* with natural *arity*, and a set of n-tuples of vertices for each relation of arity n.

Unless otherwise mentioned, the following definitions and theorems should apply equally well to all three kinds of objects: graphs, digraphs, and structures.

3.1 Cores

Homomorphism A *homomorphism from G to H* is a function ϕ from the vertices of G to the vertices of H that preserves edges. That is, if e is an edge of G, then the edge formed by applying ϕ to each component of e is an edge of H.

Retract A *retract*, or *folding*, of G is an endomorphism ϕ onto a subgraph H of G such that $x \in H$ implies $\phi(x) = x$.

Core An object for which every endomorphism is also an automorphism.

Antichain A set of objects unrelated by homomorphisms.

3.2 Results from [1]

1. A homomorphism equivalence class has at most one core.

2. A core is uniquely represented as an antichain of connected cores.

3. A graph G is uniquely represented as the infinite sequence $|Hom(F_i, G)|$ for any enumeration of all finite graphs F_i.

3.3 Proofs

1. If an equivalence class has two cores, then there are homomorphisms from each to the other, ϕ and ϕ'. Consider the compositions $\phi \circ \phi'$ and $\phi' \circ \phi$. The first is an endomorphism from the first object to itself, and hence an automorphism, and the second is an endomorphism from second object to itself, hence an automorphism. Since both $\phi \circ \phi'$ and $\phi' \circ \phi$ are bijections, so are ϕ and ϕ'. Now we can show that ϕ is an isomorphism. We already know that it is a bijective homomorphism, so we need only show that it's inverse ϕ^{-1} is a homomorphism. ϕ^{-1} is equal to $(\phi' \circ \phi)^{-1} \circ \phi'$, which is the composition of an automorphism and a homomorphism, which is a homomorphism. Thus ϕ is an isomorphism and the equivalence class's cores are isomorphic.

2. Every core is the disjoint union of some connected components. Each component must be a core, or else it would have an endomorphism which is not an automorphism and so would the whole object. Likewise, there can be no homomorphism between components, since it could be used to construct an endomorphism which is not an automorphism by mapping one component to the other, and every other component to itself. Thus the components of any core (which are themselves connected cores) form an antichain.

In [2], Bauslaugh points out that cores ought be defined as graphs for which every endomorphism is an automorphism, and *not* as a vertex-minimal member of a graph homomorphism equivalence class as suggested in [1]. For finite graphs, these definitions are equivalent, but for infinite graphs only the latter results in cores being unique. Consider, for instance, the (countably) infinite graph with vertices $0, 1, 2, ...$ and edges $(x, y)|x < y$. Under the vertex-minimal core definition, this graph has an infinite number of cores, given by the subgraphs induced by $n, n+1, n+2, ...$ for any $n \geq 1$. These are in the same homomorphism equivalence class – a forward homomorphism maps x to $x + n$, and a reverse homomorphism maps x to x. And each core is indeed vertex minimal – they each have infinitely many vertices, and there is no homomorphism to any finite graph, since that graph would have to include a clique of every order.

A jointly universal set of relational structures may have more than one core. Take as an example the set consisting of a triangle and a Grotzsch Graph.

1. Determining whether G is H-colorable is NP complete for fixed H and varying G.

2. If there is a homomorphism from G to H, what can you say about the existence of cores of G and H?

3. Is the image of every endomorphism isomorphic to a retract?

[1]: Peter J. Cameron. Graph homomorphisms (class notes). September 2006. http://www.maths.qmul.ac.uk/ pjc/csgnotes/hom1.pdf

[2]: Bruce Lloyd Bauslaugh. Homomorphisms of infinite directed graphs. December 1994. Simon Fraser University.

4 The Chase

A run of the chase *fully terminates* when it returns a finite number of finite models and then halts.

4.0.1 Parallelization

It can be unsafe to coerce two rules in parallel. Consider the theory

```
-> A & B & S A -> (P & S) | Q B -> P | (Q & S)
```

The model A, B, S, P, Q could be generated if instances of the second and third rules were coerced in parallel, even though it is not generated by any run of the chase!

Likewise, it can be dangerous to coerce two instances of a rule in parallel. Consider the theory

```
-> Exists x, y: A(x, y) & A(y, x) A(x, y) -> P(x) | P(y)
```

Considering both instances of the second rule in parallel could produce the model $A(0,1), P(0), P(1)$, which again is not generated by any run of the chase.

A sufficient condition for it to be safe to consider two rules in parallel is when their right hand sides share no relations. In this case, coercing the right hand side of

5 Implementation

5.1 Structure

The geometric theory is partitioned into three sections:

1. Protocol Specification

2. Strand Space Axioms

3. Inference Rules

6 Target Theorems

1. Show that every model generated by the chase is, in fact, an infiltrated skeleton.

2. If A is a realized infiltrated skeleton, then there exists a homomorphism from some model returned by the chase to A.

3. Show that the chase fully terminates under certain conditions.

[Possible proof technique: assume the theory ends with “completeness rules” – the rest of the axioms of the strand space formalisms – and then show them to be unnecessary.]

7 Strand Spaces

The *strand space formalism* was developed as a method for formally reasoning about cryptographic protocols. It distinguishes between two different kinds of participants: regular participants and an adversary. A single physical entity can be represented as multiple regular strands if they play more than one role in the protocol.

Participants in a protocol run are represented by *strands*, and communicate with each other by sending and receiving messages. A regular participant is represented by a regular strand and must follow the protocol. The adversary is represented by zero or more adversary strands, and can manipulate the messages that regular strands send and receive.

The actions of both the regular participants and the advesary are abstracted into message passing; this is assumed to be able to capture all relevant information. For instance, if a private key is assumed insecure and the advesary may be able to learn it, this could only be represented by an advesary strand receiving the key. As such, every strand consists of a (non-empty) sequence of message passing events called *nodes*. Each node either sends a message or receives a message. A *term* is any possible message.

In our representation of this formalism, nodes and terms are the first-class objects. Strands are represented only indirectly through nodes.

```
Forall x. Term(x) | Node(x) Term(x) & Node(x) -> False

Send(n, t) -> Node(n) & Term(t) Recv(n, t) -> Node(n) & Term(t)

Node(n) -> (Exists t. Send(n, t)) | (Exists t. Recv(n, t))

Send(n, s) & Send(n, t) -> s = t Recv(n, s) & Recv(n, t) -> s = t

Send(n, s) & Recv(n, t) -> False
```

7.1 Messaging

Terms are defined inductively over *basic terms* as follows:

- Any basic term is a term.
- The ciphertext $\{|\tau_1|\}_{\tau_2}$ is a term if the plaintext τ_1 and the key τ_2 are terms.

- The pair (τ_1, τ_2) is a term if τ_1 and τ_2 are terms.