# 1 Notes

## 1.1 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 $\phi_1$ and $\phi_2$ 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 $\forall$, 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 $R$.

## 1.2 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 $R(x, f(x)$

### 1.3 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. [...?]

# 2 Skeletons in Geometric Logic

## 2.1 Signature

- Sorts:
  Strand $A$, $B$, $C$, $D$
  Node $n$, $m$, $p$
  Term $s$, $t$, $g$, $h$, $k$

- Functions:
  msg() : Node → Term

$k^{-1}$ : Term → Term
strand() : Node → Strand
$gh$ : Term, Term → Term
$\{g\}_k$ : Term, Term → Term
sk() : Strand → Term
pk() : Strand → Term
lkt() : Strand, Strand → Term

- Given Relations:
  Node $\Rightarrow^*$ Node
  Node $\leq$ Node
  +Node
  −Node
  non Term
  Term uniq-orig Node
  (atomic Term)

- Defined Relations:
  Node out-pred Node
  Term $\sqsubseteq$ Term (within)
  Term $\sqsubseteq_v$ Term (visibly within)
  avail Term
  (Term occ Node)
  Term sent-before Node
  Term expl Node

The predecessor relation forms a total order, and strand predecessors form a partial order.

(The rules for keys were taken from *Shapes: Surveying Crypto Protocol Runs*, section 2.1.)

## 2.2 Axioms

**Atomic**

$$\begin{aligned} &\vdash \text{ atomic } t \\ &\vee \exists g\ \exists h\ t = gh \\ &\vee \exists g\ \exists k\ t = \{g\}_k \\ \text{atomic } gh &\vdash \\ \text{atomic } \{g\}_h &\vdash \end{aligned}$$

### Predecessors

$$\begin{aligned} n \leq m, m \leq p &\vdash n \leq p \\ n \leq m, m \leq n &\vdash n = m \\ &\vdash n \leq m \vee m \leq n \end{aligned}$$

$$\begin{aligned} n \Rightarrow^* m, m \Rightarrow^* p &\vdash n \Rightarrow^* p \\ n \Rightarrow^* m, m \Rightarrow^* n &\vdash n = m \end{aligned}$$

$$n \Rightarrow^* m \vdash n \leq m$$

### Direction

$$\begin{aligned} &\vdash -n \vee +n \\ -n, +n &\vdash \end{aligned}$$

### Keys

$$\begin{aligned} &\vdash k^{-1-1} = k \\ k^{-1} = k &\vdash \\ \mathrm{sk}(A) = \mathrm{sk}(B) &\vdash A = B \\ \mathrm{pk}(A) = \mathrm{pk}(B) &\vdash A = B \\ \mathrm{pk}(A)^{-1} = \mathrm{pk}(B)^{-1} &\vdash A = B \\ \mathrm{lkt}(A, B) = \mathrm{lkt}(C, D) &\vdash A = C, B = D \\ \mathrm{sk}(A) = pkB &\vdash \\ \mathrm{sk}(A) = pkB^{-1} &\vdash \\ \mathrm{pk}(A) = \mathrm{pk}(B)^{-1} &\vdash \end{aligned}$$

## 2.3 Defined Relations

### Outbound Predecessor

$$n \text{ out-pred } m \equiv +n, n \leq m$$

$$+n, n \leq m ...$$

$$+n, n \leq m \vdash n \text{ out-pred } m$$

**Visibly Within** One term is visibly within another if it is accessible via deconcatenation alone.

$$\begin{aligned} s \sqsubseteq_v t &\equiv s = t \\ &\vee \exists g \, \exists h \; t = gh, s \sqsubseteq_v g \\ &\vee \exists g \, \exists h \; t = gh, s \sqsubseteq_v h \end{aligned}$$

**Within** One term is within another if it is accessible by deconcatenation and decryption. So the plaintext of an encryption is within it, but the encryption key is not.

$$\begin{aligned} s \sqsubseteq t &\equiv \exists g \; g \sqsubseteq_v t, s \sqsubseteq g \\ &\vee \exists g \, \exists k \; t = \{g\}_k, s \sqsubseteq g \end{aligned}$$

**Occurrence**

$$t \text{ occ } n \equiv t \sqsubseteq \text{msg}(n)$$

**Availability** Unavailability is inductive, but availability is not.

$$\begin{aligned} \text{unavail } t &\equiv \text{non } t \\ &\vee \exists n \; t \text{ uniq-orig } n \end{aligned}$$

$$\text{avail } t \equiv \not\text{unavail } t$$

**Sent Before** [Is there a better name?]

$$t \text{ sent-before } n \equiv \exists n' \;\; +n', n' \leq n, t \text{ occ } n'$$

**Explained In** A term is explained in a node if a penetrator could construct it from the messages sent by outbound predecessors of the node. 'explained in' appears to be inductive only over the 'available' relation.

$$\begin{aligned} t \text{ expl } n &\equiv \text{avail } t \\ &\vee \exists g \, \exists h \; t = gh, g \text{ expl } n, h \text{ expl } n \\ &\vee \exists g \, \exists s \; g = ts, g \text{ expl } n \\ &\vee \exists g \, \exists s \; g = st, g \text{ expl } n \\ &\vee \exists g \, \exists k \; t = \{g\}_k, g \text{ expl } n, k \text{ expl } n \\ &\vee \exists g \, \exists k \; g = \{t\}_k, g \text{ expl } n, k^{-1} \text{ expl } n \\ &\vee \exists n' + n', -n, n' \leq n, t \text{ occ } n' \end{aligned}$$

**Compromised** What if instead of adding the *unavailable* relation as the complement of *available*, we add a *compromised* relation as a partial complement? A term is *compromised* if it is not only available to an adversary, but actually used by an adversary to construct an inbound message.

By adding *compromised*, we no longer deal with shapes, but instead with shapes augmented with a set of compromised messages. There isn't always a minimum augmented shape.

$$\vdash\ t \text{ expl } n$$

$$\begin{array}{rl}
t \text{ expl } n & \vdash\ t \text{ sent-before } n \\
& \lor \text{ compromised } t \\
& \lor \exists g\ \exists h\ t = gh, g \text{ expl } n, h \text{ expl } n \\
& \lor \exists g\ \exists s\ g = st, g \text{ expl } n \\
& \lor \exists g\ \exists s\ g = ts, g \text{ expl } n \\
& \lor \exists g\ \exists k\ t = \{g\}_k, g \text{ expl } n, k \text{ expl } n \\
& \lor \exists g\ \exists k\ g = \{t\}_k, g \text{ expl } n, k^{-1} \text{ expl } n
\end{array}$$

$$\text{compromised } t, \text{unavail } t\ \vdash$$

## 3 Annotations

**Soundness** A protocol is *sound* if "in every execution, whenever a message is received and its formula is relied upon, there were corresponding [previous] message transmissions with guaranteed formulas that allow it to be deduced."

Ideally, only guarenteed formulas would be written down, and reliances would be inferred. In this case, a protocol is *sound* if, in every execution, whenever a message is sent with a guarenteed formula, there were previous message transmissions with guarenteed formulas from which it could be deduced.