η Logic: An Authorization Logic with Explicit Time

Henry DeYoung

DRAFT of March 21, 2008

Contents

Chapter 1

Introduction

The tension between protecting an object and allowing it to be used or displayed is a fundamental one, even for objects that are not digital. For example, how can intruders be prevented from reading a classified document while still allowing the members of that document's security compartment to read and edit it? Or, how can the public be prevented from using a departmental photocopier, while still allowing members of the department to use it?

Because of this fundamental tension, organizations usually establish policies that delineate the conditions under which an object can be accessed. These policies, along with a mechanism for their enforcement, constitute an access control system. But, an access control system is valuable only if it can be trusted to be correct: the policies must allow only what is desired by the organization and the system must correctly enforce all of the policies.

As access control systems become more widespread and more complex, it is increasingly clear that ad hoc methods can no longer guarantee a sufficient level of trust in the system's correctness: a formal approach to access control is needed. One promising avenue is the use of logic for specifying policies. Given an appropriately defined logic, policies can be encoded as concrete logical structures, rather than relying on abstract policy descriptions.

But, why is logic a solid foundation for access control? The specification of policies in a logic provides three important benefits. First, once written in a formal logic, policies have precisely specified meanings. The ambiguity inherent in a natural language formulation no longer exists. Instead, the semantics of the logic define the meaning of a policy exactly.

Second, by expressing them in a logic, access control policies can be enforced by proof-carrying authorization (PCA) [3, 4]. In a PCA-based access control system, each resource is guarded by a resource monitor. A user requesting access to a resource must present the corresponding resource monitor with a formal proof of why she is authorized, under the system's policies, to access that resource. The monitor then checks this proof for correctness. If the proof is correct, access is granted; if the proof is incorrect, access is denied.

In PCA, then, the logical model of access control coincides with access control in the real world: access is granted in practice if, and only if, it is granted formally by the logical forms of the policies. In this way, a PCA-based implementation of an access control system is guaranteed to correctly adhere to that system's policies, whatever they may be.

Third, policies written in a logic can be subjected to extensive meta-analysis. For example, non-interference properties of the logic can be proven and used in this analysis, as demonstrated by Garg and Pfenning [13]. Potentially unintended consequences of the policies can then be discovered by automated policy analysis tools based on these properties. This and other meta-analyses increase confidence in policies' correctness.

To take advantage of these benefits, it is crucial that the underlying authorization logic be able to model as many policy motifs as possible. Of course, if the logic cannot express a critical feature of some policy, that feature could be enforced by extra-logical methods. But, by abandoning the use of logic and reverting to ad hoc methods, the above benefits will no longer apply to that feature. Specifically, although a PCA proof may be correct according to the logic's rules, access may still be denied due to the failure of the extra-logical checks. This destroys the correspondence between the logical model of access and access in practice. Even worse, meta-analysis of the formal policies cannot be used to *guarantee* their correctness with respect to an informal specification because the logic does not model a critical feature.

For this reason, when designing an authorization logic, common policy motifs should be considered for inclusion. One such motif is time. It is often desirable to limit the times during which a resource can be accessed or to grant authorizations that expire. For example, students should not be able to view the solutions to a homework assignment until after the due date. Because of the ubiquity of such time-dependent access control policies, one would hope that an authorization logic incorporating time exists.

Surprisingly, of the numerous logics [1–3, 7, 12, 13, 17, 18] and languages [7, 11] proposed in the access control literature, few allow time-dependent policies. Those that do handle time do so using extra-logical mechanisms: we know of no authorization logic that incorporates time internally. This void motivates us to develop an authorization logic with time.

Because time-dependent authorizations typically use explicit times, such as "between 9am and 5pm" or "during the month of May 2008," the logic developed in this thesis incorporates explicit time intervals rather than relative times, such as "at some time in the future." For this reason, the logic is dubbed η logic, where η (spelled "eta") stands for *E*xplicit *T*ime *A*uthorization.

η logic borrows ideas from constructive hybrid logic [8, 9, 20] to model time intervals as possible worlds in which propositions may be true. Accordingly, the @ connective of hybrid logic is used to relativize the truth of a proposition to a time interval, as in $A \mathbin{@} I$. η logic also adopts techniques from constraint-based reasoning [16, 21] to manage an inclusion relation between intervals.

Another common policy motif is that of consumable credentials. One often wants to allow only a finite number of accesses. For example, students might be freely authorized to make 250 photocopies per semester and must purchase the authorization to make additional copies. That is, a finite number of accesses are free of charge.

An authorization logic that can express changes of state would be able to account for such policies. Linear logic [10, 15] is a logic that can model consumable resources. For this reason, logics of authorization that include ideas from linear logic have been proposed [12]. To incorporate linear policies in addition to time-dependent ones, η logic is extended with techniques from linear authorization logics. Thus, η logic is actually a family of logics comprised of a non-linear η logic and a linear η logic.

In summary, this thesis makes two conceptual contributions. First, an authorization logic that directly incorporates time is developed and its applicability to natural time-dependent policies is demonstrated. Second, by the existence of a linear version of η logic, linearity is shown to be orthogonal to explicit time. This peaceful coexistence of the two features was not initially obvious because both linearity and time "consume" objects—linearity by usage and time by expiration.

This thesis also makes a small practical contribution. The natural deduction proof checker presented shows that a full-fledged PCA implementation of the linear η logic should be easily constructible, at least in a centralized system.

Related Work

Coming soon.

Organization of the Thesis

The remainder of the thesis is organized as follows. Chapter 2 reviews a non-linear logic of authorization that does not use time. Examples are given to clarify the use of the logic and demonstrate the need for time-dependent policies. In Chapter 3, we develop non-linear η logic. Examples highlight the increased expressive power of non-linear η logic and indicate the need for linear policies. Meta-theoretic properties of the logic are proven, increasing our confidence in the logic's soundness. Chapter 4 extends the previous logic by adding linearity. As the examples show, linearity increases the expressive power even more. The meta-theoretic properties are also extended to account for linearity. Finally, Chapter 5 presents a natural deduction formulation of linear η logic and briefly describes the corresponding implementation. The soundness and completeness of the natural deduction system with respect to the linear η sequent calculus are also established.

Chapter 2

Preliminaries: Garg-Pfenning Authorization Logic

η logic draws very heavily from a constructive, proof-theoretic authorization logic developed by Garg and Pfenning [13]. Before presenting η logic, it will be useful to review the so-called Garg-Pfenning logic (hereafter GP logic). This review will allow us to introduce concepts from proof-theoretic authorization logics, including the key concept of affirmation, will familiarize the reader with the expression of access control policies in an authorization logic, and will afford us an opportunity to present some meta-theory.

2.1 Logical System

Proof-theoretic logics, as an alternative to axiomatic logics, were first introduced by Gentzen [14]. These logics make the meanings of propositions exact by precisely specifying how each form of proposition may be verified. By coinciding a logic's semantics with its syntactic proofs, proof theory provides a high degree of assurance in that logic's correctness.

Later, Martin-Löf introduced a distinction between judgments and propositions [19]. Under this formulation, a judgment is an object of knowledge and is made evident by a formal proof. Propositions, then, are those things that can be acted on by the logical connectives, such as conjunction and implication.

GP logic adheres to both of these fundamental ideas in an effort to keep the meanings of proofs clean and direct. We begin by reviewing the first-order terms and sorts of the logic. Next, we introduce the truth and affirmation judgments that form the foundation of GP logic. This introduction is carefully separated from the description of the logic's propositions, to emphasize Martin-Löf's distinction. Finally, we present the proof rules of GP logic as a Gentzenstyle sequent calculus.

2.1.1 First-order Terms and Sorts

To account for atomic propositions built from predicates and for universal and existential quantification, GP logic contains terms t which are classified by sorts s. That term t has sort s is denoted by the judgment $t{:}s$.

The particular sorts and terms available in GP logic are left open-ended, with the exception that a sort $\mathsf{principal}$ of principals is specifically assumed. Principals are the entities, typically users or machines, that can make statements of affirmation. The meta-variable K is used to stand for an arbitrary principal.

Because we will want to be able to reason parametrically with terms, GP logic introduces a signature, Σ, to track the parameters in scope and their respective sorts. The syntax of a signature is:

$$\Sigma ::= \cdot \mid \Sigma, x{:}s$$

Thus, a signature is simply a list of sort-parameter ascriptions: a signature may be empty, written as $\cdot$; or, it may be a signature Σ followed by the ascription of a sort s to a parameter x, written as $\Sigma, x{:}s$. To avoid ambiguities, we assume that all parameters declared in Σ are distinct from x; this convention can be maintained by implicitly renaming variables according to α conversion.

Since GP logic now includes parameters, the judgment $t{:}s$ must be extended to account for parameters, in addition to the ground terms it already handles. The new judgment is written $\Sigma \vdash t{:}s$, meaning that term t has sort s in signature Σ. In particular, $\Sigma, x{:}s \vdash x{:}s$ holds. Also, $[t/x]$ stands for the capture-avoiding substitution of term t for all occurrences of the free variable x. In particular, $[t/x]A$ is the proposition A with all free occurrences of x replaced by t.

2.1.2 Judgments

In GP logic, it is necessary to reason about the truth of propositions. That is, statements of the form "Proposition A is true" are objects of knowledge and the subjects of proofs. Following Martin-Löf's philosophy, GP logic therefore includes the judgment form $A\ \mathsf{true}$, which presupposes that A is a well-formed proposition. For syntactic simplicity, the modifier true will often be dropped, so that A will implicitly stand for the judgment $A\ \mathsf{true}$.

However, the truth of propositions is not a sufficiently expressive notion upon which to base an authorization logic. In addition to reasoning about objective truths, it is necessary to reason about principals' policies or intents. The approach taken by GP logic is to add a new judgment form $K\ \mathsf{affirms}\ A$, meaning that "Principal K affirms that proposition A is true." A principal, then, issues a policy by affirming the truth of that policy. In an implementation, the affirmation $K\ \mathsf{affirms}\ A$ will be established by a certificate signed by K containing A.

These judgments of truth and affirmation are the basic judgment forms of GP logic. However, they are of little use in and of themselves; we need to be able to reason from hypotheses. The mechanism that GP logic uses is termed a hypothetical judgment or sequent, an extension of a basic judgment that explicitly lists the allowable assumptions.

Specifically, GP logic uses two hypothetical judgment forms:

$$\Sigma; \Gamma \Longrightarrow A \ \mathsf{true}$$
$$\Sigma; \Gamma \Longrightarrow K \ \mathsf{affirms} \ A$$

where Σ is a signature of the parameters, ascribed with sorts, that appear in the remainder of the judgment, and Γ is an unordered collection of hypotheses of the form $A \ \mathsf{true}$, called a context[1].

The first of the above hypothetical judgments may be read "Under the hypotheses of Γ, proposition A is true, parametrically in the terms of Σ." Similarly, the second hypothetical judgment states "Under the hypotheses of Γ, principal K affirms that proposition A is true, parametrically in the terms of Σ."

2.1.3 Propositions

The syntax of propositions in GP logic is:

$$A, B ::= P \mid A \wedge B \mid \top \mid A \vee B \mid A \supset B \mid \forall x{:}s.A \mid \exists x{:}s.A \mid \langle K \rangle A$$

GP logic contains nearly all of the ordinary connectives from first-order logic: atomic propositions, P; conjunction, $A \wedge B$; truth, $\top$; implication, $A \supset B$; universal quantification, $\forall x{:}s.A$; and existential quantification, $\exists x{:}s.A$. However, falsehood, $\perp$, is conspicuously absent. Falsehood is omitted from non-linear η logic for reasons that will discussed in Section 3.1.4, and, for consistency, it is also omitted here.

Despite the close similarity of these propositions to those of first-order logic, there is one form of proposition that is unique to authorization logics: $\langle K \rangle A$, read "K says A". This proposition internalizes the affirmation judgment $K \mathsf{affirms}$ A, meaning that it is semantically equivalent to $K \, \mathsf{affirms} \, A$, but is a proposition rather than a judgment.

Having an affirmation *proposition* allows affirmations to be combined with logical connectives, such as implication. For example, we could not combine the judgment $K \ \mathsf{affirms} \ A$ with the proposition B via implication because this would violate Martin-Löf's distinction between judgments and propositions: only propositions, and not judgments, can be operated on by the logical connectives. But, we can combine the *proposition* $\langle K \rangle A$ with the proposition B via implication as $(\langle K \rangle A) \supset B$.

2.1.4 Inference Rules

Since GP logic possesses a proof-theoretic semantics, its proof rules are critically important. They, and not any other external semantics, establish the meaning

[1] Although this meaning is distinct from its usage in the access control literature, we will continue to use this terminology, as it is common in logic and type theory.

of the truth and affirmation judgments. We must therefore proceed to present the proof rules of GP logic.

In proof-theoretic logics, each inference rule is written in the form:

$$\frac{J_1 \quad J_2 \quad \cdots \quad J_n}{J}\ \mathit{label}$$

This notation means that if the premise judgments $J_1, J_2, \ldots, J_n$ are evident, then the conclusion judgment J is also evident by the rule named *label*. Note that n may be 0. In this case, the rule has the form:

$$\frac{}{J}\ \mathit{label}$$

and the conclusion judgment J is always evident: there are no proof obligations.

With the notation explained, we can now describe the inference rules of GP logic. We begin by examining the meaning of hypotheses through the init rule.

$$\frac{}{\Sigma; \Gamma, P \Longrightarrow P}\ \mathsf{init}$$

We would expect that an assumption $A\,\mathsf{true}$ could be used to immediately conclude $A\,\mathsf{true}$. This is, in fact, the case. However, for technical reasons, we do not adopt this in its full generality as an inference rule, but instead use the above init rule which restricts the direct use of hypotheses to atomic propositions P. We can recover the more general form as a meta-theorem (Theorem 2.1, Section 2.3).

Next, we consider the rules for the affirmation judgment and its internalization as a proposition.

$$\frac{\Sigma; \Gamma \Longrightarrow A}{\Sigma; \Gamma \Longrightarrow K\ \mathsf{affirms}\ A}\ \mathsf{affirms}$$

When is an affirmation judgment evident? That is, when can we conclude that a principal K affirms the truth of proposition A? If A is true, it is made evident by a proof. When this proof is presented to K, K is confronted with irrefutable evidence of the truth of A. K cannot possibly deny the truth of A, for doing so would violate K's rationality. Instead, K must affirm it. Thus, one way of establishing $K\ \mathsf{affirms}\ A$ is to establish $A\,\mathsf{true}$. This is captured by the above affirms rule.

In a sequent calculus, the meaning of each logical connective $\star$ is defined by a set of right rules and a set of left rules. Right rules show how $A \star B\,\mathsf{true}$ may be established, and left rules show how a hypothesis $A \star B\,\mathsf{true}$ may be used. For the says connective, there is one right rule and one left rule:

$$\frac{\Sigma; \Gamma \Longrightarrow K\ \mathsf{affirms}\ A}{\Sigma; \Gamma \Longrightarrow \langle K \rangle A}\ \langle\rangle R \qquad \frac{\Sigma; \Gamma, \langle K \rangle A, A \Longrightarrow K\ \mathsf{affirms}\ B}{\Sigma; \Gamma, \langle K \rangle A \Longrightarrow K\ \mathsf{affirms}\ B}\ \langle\rangle L$$

The right rule $\langle\rangle R$ specifies that $\langle K \rangle A\,\mathsf{true}$ may be established by evidence that $K\ \mathsf{affirms}\ A$ holds. This is consistent with our above claim that the proposition

$\langle K\rangle A$ is the internalization of the judgment K $\mathsf{affirms}$ A. We now know how to verify $\langle K\rangle A\ \mathsf{true}$, but how does one use the hypothesis $\langle K\rangle A\ \mathsf{true}$?

The left rule $\langle\rangle L$ gives instructions for how the hypothesis $\langle K\rangle A\ \mathsf{true}$ may be used. Because $\langle K\rangle A\ \mathsf{true}$ represents the knowledge that K affirms $A\ \mathsf{true}$, from K's perspective, A may as well be true; K will never admit that one of her beliefs is invalid. So, provided that we are reasoning about an affirmation made by K, that is, provided that we are inside K's mind, the hypotheses $\langle K\rangle A\ \mathsf{true}$ and $A\ \mathsf{true}$ are equivalent.

We now review the inference rules for implication and universal quantification. A reader familiar with the sequent calculus presentation of first-order logic may skip this discussion; there is nothing unique to GP logic in the remaining rules.

First, we give the rules for implication.

$$\frac{\Sigma;\Gamma, A \Longrightarrow B}{\Sigma;\Gamma \Longrightarrow A \supset B}\supset\! R \qquad \frac{\Sigma;\Gamma, A\supset B \Longrightarrow A \quad \Sigma;\Gamma, A\supset B, B \Longrightarrow \gamma}{\Sigma;\Gamma, A\supset B \Longrightarrow \gamma}\supset\! L$$

The implication $A \supset B$ may be intuitively thought of as a plan for converting a proof of $A\ \mathsf{true}$ to a proof of $B\ \mathsf{true}$. Such a conversion can be established by assuming that a proof of $A\ \mathsf{true}$ is given and constructing a proof of $B\ \mathsf{true}$ from this assumption. This is captured by the right rule $\supset R$. The conversion intuition also suggests that the hypothesis $A \supset B\ \mathsf{true}$ can be used by executing this plan. Given $A\ \mathsf{true}$, the plan $A \supset B\ \mathsf{true}$ can be carried out to produce $B\ \mathsf{true}$. This intuition is formalized in the left rule $\supset L$.

Next, we give the rules for universal quantification.

$$\frac{\Sigma, x{:}s;\Gamma \Longrightarrow A}{\Sigma;\Gamma \Longrightarrow \forall x{:}s.A}\forall R \qquad \frac{\Sigma \vdash t{:}s \quad \Sigma;\Gamma,\forall x{:}s.A, [t/x]A \Longrightarrow \gamma}{\Sigma;\Gamma,\forall x{:}s.A \Longrightarrow \gamma}\forall L$$

The right rule $\forall R$ states that $\forall x{:}s.A\ \mathsf{true}$ may be verified by establishing $A\ \mathsf{true}$ for all possible terms of sort s. This is done by introducing a new parameter x of sort s and establishing $A\ \mathsf{true}$ parametrically in x. Just as the implication $A \supset B$ can be thought of as a plan for converting a proof of $A\ \mathsf{true}$ to a proof of $B\ \mathsf{true}$, the right rule $\forall R$ suggests that $\forall x{:}s.A$ can be thought of as a plan for creating a proof of $[t/x]A\ \mathsf{true}$ for any term t of sort s. So, assuming such a plan and given a term t of sort s, the plan can be carried out to produce $[t/x]A\ \mathsf{true}$. This intuition is captured by the left rule $\forall L$.

The remaining connectives of GP logic and their rules are taken directly from first-order logic. A summary of all of the inference rules in GP logic is given Figure 2.1.

To illustrate some properties of GP logic, we state a few judgments derivable and a few judgments not derivable in the logic.

1. $\cdot;\cdot \Longrightarrow A \supset \langle K\rangle A\ \mathsf{true}$
2. $\cdot;\cdot \Longrightarrow \langle K\rangle\langle K\rangle A \supset \langle K\rangle A\ \mathsf{true}$
3. $\cdot;\cdot \Longrightarrow \langle K\rangle(A \supset B) \supset \langle K\rangle A \supset \langle K\rangle B\ \mathsf{true}$
4. $\cdot;\cdot \not\Longrightarrow \langle K\rangle A \supset A\ \mathsf{true}$

Initial Rule

$$\frac{}{\Sigma; \Gamma, P \Longrightarrow P}\ \mathsf{init}$$

Affirmation and $\langle K \rangle A$

$$\frac{\Sigma; \Gamma \Longrightarrow A}{\Sigma; \Gamma \Longrightarrow K\ \mathsf{affirms}\ A}\ \mathsf{affirms}$$

$$\frac{\Sigma; \Gamma \Longrightarrow K\ \mathsf{affirms}\ A}{\Sigma; \Gamma \Longrightarrow \langle K \rangle A}\ \langle\rangle R \qquad \frac{\Sigma; \Gamma, \langle K \rangle A, A \Longrightarrow K\ \mathsf{affirms}\ B}{\Sigma; \Gamma, \langle K \rangle A \Longrightarrow K\ \mathsf{affirms}\ B}\ \langle\rangle L$$

Other Connectives

$$\frac{\Sigma; \Gamma \Longrightarrow A \quad \Sigma; \Gamma \Longrightarrow B}{\Sigma; \Gamma \Longrightarrow A \wedge B}\ \wedge R$$

$$\frac{\Sigma; \Gamma, A \wedge B, A \Longrightarrow \gamma}{\Sigma; \Gamma, A \wedge B \Longrightarrow \gamma}\ \wedge L_1 \qquad \frac{\Sigma; \Gamma, A \wedge B, B \Longrightarrow \gamma}{\Sigma; \Gamma, A \wedge B \Longrightarrow \gamma}\ \wedge L_2$$

$$\frac{}{\Sigma; \Gamma \Longrightarrow \top}\ \top R$$

$$\frac{\Sigma; \Gamma \Longrightarrow A}{\Sigma; \Gamma \Longrightarrow A \vee B}\ \vee R_1 \qquad \frac{\Sigma; \Gamma \Longrightarrow B}{\Sigma; \Gamma \Longrightarrow A \vee B}\ \vee R_2$$

$$\frac{\Sigma; \Gamma, A \vee B, A \Longrightarrow \gamma \quad \Sigma; \Gamma, A \vee B, B \Longrightarrow \gamma}{\Sigma; \Gamma, A \vee B \Longrightarrow \gamma}\ \vee L$$

$$\frac{\Sigma; \Gamma, A \Longrightarrow B}{\Sigma; \Gamma \Longrightarrow A \supset B}\ \supset R \qquad \frac{\Sigma; \Gamma, A \supset B \Longrightarrow A \quad \Sigma; \Gamma, A \supset B, B \Longrightarrow \gamma}{\Sigma; \Gamma, A \supset B \Longrightarrow \gamma}\ \supset L$$

$$\frac{\Sigma, x{:}s; \Gamma \Longrightarrow A}{\Sigma; \Gamma \Longrightarrow \forall x{:}s.A}\ \forall R \qquad \frac{\Sigma \vdash t{:}s \quad \Sigma; \Gamma, \forall x{:}s.A, [t/x]A \Longrightarrow \gamma}{\Sigma; \Gamma, \forall x{:}s.A \Longrightarrow \gamma}\ \forall L$$

$$\frac{\Sigma \vdash t{:}s \quad \Sigma; \Gamma \Longrightarrow [t/x]A}{\Sigma; \Gamma \Longrightarrow \exists x{:}s.A}\ \exists R \qquad \frac{\Sigma, x{:}s; \Gamma, \exists x{:}s.A, A \Longrightarrow \gamma}{\Sigma; \Gamma, \exists x{:}s.A \Longrightarrow \gamma}\ \exists L$$

Figure 2.1: The inference rules for Garg-Pfenning logic.

5. $\cdot;\cdot \not\Longrightarrow A\ \mathsf{true}$

The first three judgments show that $\langle K \rangle$ satisfies the properties of a lax modality. The fourth judgment highlights the difference between truth and affirmation: truth is always affirmed (as shown in the first judgment), but an affirmation by some principal does not entail truth. The fifth judgment states that not every proposition is true a priori in GP logic, implying consistency of the logic.

2.2 Examples

Now that we have presented the judgments and crucial inference rules of GP logic, the reader should be sufficiently prepared to consider a few examples of policies written in GP logic. First, we present an example that will recur throughout the remainder of this paper: controlling access to academic offices. Although this example is relatively small, it will still demonstrate the use of affirmation in GP logic, and, in later chapters, highlight the increased expressive power of η logic. Second, we examine the application of GP logic to chemical laboratory inspections.

2.2.1 Office Entry

In this example, we describe two hypothetical policies for the Grey system [5, 6], an architecture for controlling entry to academic offices that was developed and is currently deployed at Carnegie Mellon University. In the Grey system, each office door is equipped with a processor that controls access to the office through PCA. Following the standard PCA methodology, the office door will unlock only if the principal requesting access presents the doorfront processor with a correct proof that, under the security policies of the system, she is authorized to enter.

For this example, we postulate the existence of an administrating principal, admin, that controls entry to the various faculty, staff, and student offices in his administrative domain. For simplicity, we also assume that the ownership relation from principals to offices is an injective function, so that each office can be named according to its owner.

Only one predicate is used here: $\mathsf{may\_enter}$. $\mathsf{may\_enter}(K_2, K_1)$ means that principal K_2 is allowed to enter K_1's office.

One reasonable policy to include in such a system is the authorization of every principal to enter her own office. Because admin controls each office, this policy is expressed in GP logic as:

$$\mathsf{own} : \langle \mathrm{admin} \rangle (\forall K{:}\mathsf{principal}.\mathsf{may\_enter}(K, K))\ \mathsf{true} \qquad (2.1)$$

This policy may be read as “The administrator says that each principal K may enter her own office.” Although extremely simple, this policy exhibits an important point. Because the certificate corresponding to an affirmation must be an independent object, it cannot contain free variables. Thus, any quantifiers must appear inside the top-level affirmation, as seen in the own policy.

Another reasonable feature to have in an office access control system is the ability of each office owner to decide who may enter her office. To accomplish this, the administrator can agree to trust office owners' access control decisions:

$$\mathsf{trust} : \langle \mathrm{admin} \rangle (\forall K_1{:}\mathsf{principal}.\forall K_2{:}\mathsf{principal}.\langle K_1 \rangle \mathsf{may\_enter}(K_2, K_1) \supset \mathsf{may\_enter}(K_2, K_1))\mathsf{true} \tag{2.2}$$

This policy may be read as "The administrator says that, for all pairs of principals K_1 and K_2, if K_1 says K_2 may enter K_1's office, then K_2 may indeed enter K_1's office." The trust policy expresses a kind of delegation: K_1 now speaks for admin on matters of K_1's office.

To clarify how the trust policy can be used, consider a professor Alice and her graduate student Bob. Suppose that Alice is out of the office on May 7, 2008. But, Bob needs to retrieve a paper from Alice's office that he and Alice are collaborating on. He calls Alice and she agrees to issue the following credential:

$$\mathcal{C}_0 : \langle \mathrm{Alice} \rangle \mathsf{may\_enter}(\mathrm{Bob}, \mathrm{Alice})\ \mathsf{true}$$

Bob then approaches Alice's door and requests entry to her office using his cell phone. Before the door will unlock, Bob must submit a correct proof of

$$\mathrm{Alice}{:}\mathsf{principal}, \mathrm{Bob}{:}\mathsf{principal}; \mathsf{own}, \mathsf{trust}, \mathcal{C}_0 \Longrightarrow \langle \mathrm{admin} \rangle \mathsf{may\_enter}(\mathrm{Bob}, \mathrm{Alice})\mathsf{true}$$

That is, Bob must prove that the administrator allows him to enter Alice's office. Bob's phone constructs the required proof by simply applying the trust hypothesis to the $\mathcal{C}_0$ hypothesis. The doorfront processor checks this proof, and, since it is correct, unlocks the door.

Although this policy serves its purpose, it is a rather coarse approximation to the behavior desired in general. It is likely that Alice wants the credential $\mathcal{C}_0$ to allow Bob access to her office *only* on May 7, 2008. If he needs access at a later time, he should be required to contact Alice first. But, under GP logic, once Alice issues this credential, Bob will be able to enter her office at any time, even months or years after May 7, 2008!

As noted previously, time might be handled in such a system using extralogical checks. But then, the proof does not accurately reflect the true state of the system: access might be denied even though the proof is correct. This inaccuracy, even for such a simple example as office entry, motivates the development of η logic. We will revisit this example in Section 3.2.1 and show that, in η logic, users can restrict access to their offices by time.

2.2.2 Chemical Laboratory Inspections

We now consider a more complicated example. Inspection duties of the United States Occupational Safety and Health Administration (OSHA) include the oversight of chemical laboratories. As a rough approximation, the inspection process can be thought of as a verification that all employees of the laboratory are "safe" in some appropriately defined way. Only if OSHA can be guaranteed of this, will it certify the laboratory.

To model the inspection process in GP logic, we assume the existence of a sort, lab, of chemical laboratories and the existence of a distinguished principal OSHA. The following predicates are required:

$\mathsf{is\_employee}(K, L)$	Principal K is an employee of lab L.
$\mathsf{is\_manager}(K, L)$	Principal K is a manager of lab L.
$\mathsf{is\_technician}(K, L)$	Principal K is a technician of lab L.
$\mathsf{is\_janitor}(K, L)$	Principal K is a janitor of lab L.
$\mathsf{is\_safe}(K, L)$	Principal K is safe in lab L.
$\mathsf{is\_certified}(L)$	Lab L is certified and may continue operating.

It is reasonable to assume that OSHA classifies each employee of a laboratory according to his job description. We assume that the three classes established by OSHA are: manager, technician, and janitor. This classification policy can be expressed as:

$$\mathsf{job} : \langle \mathrm{OSHA} \rangle (\forall L{:}\mathsf{lab}.\forall K{:}\mathsf{principal}.\mathsf{is\_employee}(K, L) \supset \\ (\mathsf{is\_manager}(K, L) \vee \mathsf{is\_technician}(K, L) \vee \mathsf{is\_janitor}(K, L)))\ \mathsf{true} \tag{2.3}$$

This policy provides a method for distinguishing the job that an employee holds. Employees holding different positions may be "safe" under different conditions. For example, janitors may be exposed to chemicals but need not operate lab equipment, while technicians will handle chemicals and operate equipment. For this reason, a janitor might be "safe" if he can access safety procedures for all chemicals in the lab, but he need not (and perhaps should not) access equipment manuals. On the other hand, a technician would need to be able to access both chemical safety procedures and equipment manuals to be "safe."

OSHA's certification policy can then be expressed as:

$$\mathsf{certify} : \langle \mathrm{OSHA} \rangle (\forall L{:}\mathsf{lab}.(\forall K{:}\mathsf{principal}.\mathsf{is\_employee}(K, L) \supset \mathsf{is\_safe}(K, L)) \supset \mathsf{is\_certified}(L))\mathsf{true} \tag{2.4}$$

This can be read as "OSHA says that a lab L is certified if, for all employees K of lab L, K is safe in lab L."

In many policies, credentials are required to establish a result. Note that, in the $\mathsf{certify}$ policy, however, the requirement is a kind of conditional credential: the safety of a principal K in lab L is only needed when K is an employee of L. Because this condition exists, it is possible, using the case analysis induced by the job policy, to take the specific job of K into account when determining K's safety.

2.3 Meta-theory

One of the key advantages of a proof-theoretic logic is its vulnerability to a rigorous meta-theoretic analysis. Meta-theorems are stated as natural and desirable properties of the logic—properties that one would expect to hold. The proofs of these properties serve as a kind of "sanity check" on the design of the logic; if some expected property fails to hold, perhaps the logic's design should be reconsidered.

As a proof-theoretic logic, the meta-theory of GP logic can be explored in this way. There are two reasonable properties for GP logic. First, as alluded to in the discussion of the init rule (see Section 2.1.4), for *any* proposition A, from the assumption that A is true, it should be possible to establish that A is true. For atomic propositions P, this is captured explicitly in the init inference rule. For arbitrary propositions A, this is stated and proved as the following theorem.

Theorem 2.1 (Identity)**.** *For any proposition A,* $\Sigma; \Gamma, A\,\mathsf{true} \Longrightarrow A\,\mathsf{true}$ *.*

Second, the logic should possess a so-called cut elimination property. One cut rule for GP logic states that a proof of $A\,\mathsf{true}$ can be used to replace the hypothesis $A\,\mathsf{true}$ in a proof of γ to yield a direct proof of γ. For this reason, a cut rule might be intuitively thought of as a method for creating and using lemmata; the proof of $A\mathsf{true}$ functions as the lemma and the hypothesis $A\mathsf{true}$ in the proof of γ corresponds to the use of the lemma in proving the main theorem.

Since GP logic also permits conclusions of the form K $\mathsf{affirms}$ A, a rule for cutting affirmation is also needed. $K\,\mathsf{affirms}\,A$ can replace the hypothesis $A\,\mathsf{true}$ in a proof of $K\,\mathsf{affirms}\,B$ since, from K's perspective, truth and K's affirmations are equivalent.

Cut elimination means that an explicit cut rule is not needed in the logic: any uses of the rule are unnecessary. The following theorem states the admissibility of cut. Because cut elimination follows from this by a straightforward induction, often only the admissibility of cut is formally stated and proven.

Theorem 2.2 (Admissibility of Cut)**.**

1. *If* $\Sigma; \Gamma \Longrightarrow A\,\mathsf{true}$ *and* $\Sigma; \Gamma, A\,\mathsf{true} \Longrightarrow \gamma$*, then* $\Sigma; \Gamma \Longrightarrow \gamma$*.*
2. *If* $\Sigma; \Gamma \Longrightarrow K\,\mathsf{affirms}\,A$ *and* $\Sigma; \Gamma, A\,\mathsf{true} \Longrightarrow K\,\mathsf{affirms}\,B$*, then* $\Sigma; \Gamma \Longrightarrow K\,\mathsf{affirms}\,B$*.*

The proofs and associated lemmas for the above meta-theorems are given in [13].

2.4 Conclusion

In hopes of adequately preparing the reader for the following discussion of η logic, this chapter has reviewed a proof-theoretic authorization logic developed by Garg and Pfenning [13]. We have also seen the application of GP logic to two disparate systems: office access control and chemical laboratory inspections. Finally, we have presented the meta-theory of GP logic and explained its importance as an expression of the logic's soundness. We now proceed to Chapter 3 where we develop non-linear η logic, which is heavily based on principles from GP logic reviewed in this chapter.

Chapter 3

Non-linear η Logic

Introduction coming soon.

3.1 Logical System

Introduction coming soon.

3.1.1 First-order Terms and Sorts

The basic system for first-order terms and sorts remains as it was in GP logic. Σ is still a signature listing the parameters in scope and their respective sorts. We continue to write $\Sigma \vdash t{:}s$ for the judgment that term t has sort s and $[t/x]A$ for the substitution of term t for the free variable x in proposition A.

The sort $\mathsf{principal}$ of principals is carried over from GP logic. Non-linear η logic includes two additional sorts: the sort time of times and the sort $\mathsf{interval}$ of time intervals. The remaining sorts in the logic are left open-ended so that application-specific sorts can be added as needed.

Times are the components that comprise the time intervals about which non-linear η logic reasons. Because the logic does not depend on it, a concrete structure for times is not given, but instead left to be specified by individual applications. However, intuitively, one may think of times as points on the real line. Times are usually represented by t; it should be clear from the context whether a given occurrence of t indicates an arbitrary term or a time.

Intervals, represented with the meta-variable I, are sets of time about which reasoning occurs. Despite the use of the terminology "interval," these sets of time need not be intervals in the mathematical sense; that is, they need not have the form $[t_1, t_2] = \{x \mid t_1 \leq x \leq t_2\}$ or the related open interval forms. Non-linear η logic is flexible enough to permit the use of arbitrary sets of time (provided they possess a preorder of inclusion). However, we overlook the slight abuse of terminology since structures that are strictly intervals appear naturally in many applications.

3.1.2 Constraints

As will be seen in Section 3.1.5, the rules of non-linear η logic will require a preorder of inclusion for intervals. Because interval parameters are permitted in the logic, it is not sufficient to simply adopt a mathematical definition of interval inclusion. Instead, a constraint domain is incorporated in the logic. The superset constraint form $I \supseteq I'$ is required, but the remainder of this domain is left open-ended: other constraint forms may be freely added for application-specific purposes.

The meta-variable C denotes an arbitrary constraint form. Because it will be necessary to assume that certain constraints hold during reasoning, a constraint context is introduced, with the following syntax:

$$\Psi ::= \cdot \mid \Psi, C$$

Thus, each constraint context Ψ is a (possibly empty) list of constraints. Reordering of the members of Ψ is freely permitted.

We will use the constraint entailment judgment

$$\Sigma; \Psi \models C$$

to mean "Under the constraints of Ψ, constraint C holds, parametrically in the members of signature Σ." Note that the signature Σ is required because Ψ and C may contain parameters from Σ.

Because the structure of intervals is left abstract, even the particular decision procedure used to solve superset constraints remains relatively unspecified: any system satisfying the following six basic properties can be used as the constraint domain. These properties are required for the meta-theory that will be presented in Section 3.3.

(Hypothesis) $\Sigma; \Psi, C \models C$.
(Weakening) If $\Sigma; \Psi \models C$, then $\Sigma, \Sigma'; \Psi, \Psi' \models C$.
(Cut) If $\Sigma; \Psi \models C$ and $\Sigma; \Psi, C \models C'$, then $\Sigma; \Psi \models C'$.
(Substitution) If $\Sigma \vdash t{:}s$ and $\Sigma, x{:}s; \Psi \models C$, then $\Sigma; [t/x]\Psi \models [t/x]C$.
(Reflexivity) $\Sigma; \Psi \models I \supseteq I$.
(Transitivity) If $\Sigma; \Psi \models I \supseteq I'$ and $\Sigma; \Psi \models I' \supseteq I''$, then $\Sigma; \Psi \models I \supseteq I''$.

3.1.3 Judgments

Our goal in designing non-linear η logic is to allow reasoning about explicit time within an authorization logic. Instead of reasoning about the truth of propositions, as was done in GP logic, it is necessary to reason about the truth of propositions *during* explicit time intervals. Therefore, the objects of knowledge in non-linear η logic are not statements of the form "Proposition A is true," but rather statements of the form "Proposition A is true during interval I." Then, according to Martin-Löf's philosophy, the logic should include a judgment form that relativizes truth to a time interval. We choose to write $A[I]$ for the judgment meaning "Proposition A is true during interval I."

In addition to its truth judgment form, $A\,\mathsf{true}$, GP logic included an affirmation judgment form, $K\,\mathsf{affirms}\,A$, to model principals' intents and policies. It is therefore natural to include affirmation in non-linear η logic, since it is still necessary to model policies. But, how should affirmation interact with explicit time intervals?

By adopting the reasonable notion that everything can be relativized to a time interval, it can be concluded that each affirmation made by a principal occurs on some time interval. Moreover, a principal cannot affirm a proposition, but must instead affirm a judgment. Combining these two ideas naturally leads to statements of the form "During interval I, principal K affirms the truth of proposition A on interval I'" as objects of knowledge. Using the @ connective described in the next two sections, the previous statement will be equivalent to "During interval I, principal K affirms the truth of proposition $A \mathbin{@} I'$ on interval I." As a result, it is sufficient to consider only statements of the latter form; if the interval of truth is different than the interval of affirmation, it can be embedded in the proposition.

We therefore arrive at the judgment form $(K\ \mathsf{affirms}\ A)\ \mathsf{at}\ I$ meaning that "During interval I, principal K affirms the truth of proposition A on I." Since, as mentioned previously, principals do not affirm propositions, but instead judgments, it would be more precise to write the affirmation judgment form as $(K\ \mathsf{affirms}\ A[I])\ \mathsf{at}\ I$. But because the two intervals are the same, we elide the unnecessary first interval.

Because reasoning from assumptions is needed, non-linear η logic extends the basic judgment forms $A[I]$ and $(K\,\mathsf{affirms}\,A)\,\mathsf{at}\,I$ to permit hypotheses. The hypothetical judgment forms are:

$$\Sigma;\Psi;\Gamma \Longrightarrow A[I]$$
$$\Sigma;\Psi;\Gamma \Longrightarrow (K\ \mathsf{affirms}\ A)\ \mathsf{at}\ I$$

where Σ is a signature ascribing sorts to the parameters that may appear in the remainder of the judgment, Ψ is a constraint context containing the constraints assumed to hold, and Γ is a context of hypotheses of the form $A[I]$.

The first hypothetical judgment form means "Assuming that the constraints in Ψ hold and under the assumptions in Γ, proposition A is true during interval I, parametrically in the terms of Σ." Similarly, the second hypothetical judgment form means "Assuming that the constraints in Ψ hold and under the assumptions in Γ, during interval I, principal K affirms that proposition A is true on I, parametrically in the terms of Σ."

3.1.4 Propositions

The propositions in non-linear η logic are given by the following grammar:

$$\begin{aligned} A, B ::= {} & P \mid A \wedge B \mid \top \mid A \vee B \mid A \supset B \mid \forall x{:}s.A \mid \exists x{:}s.A \mid \langle K \rangle A \\ & \mid A \mathbin{@} I \mid C \mathbin{\dot{\supset}} A \mid C \mathbin{\dot{\wedge}} A \end{aligned}$$

These propositions include those of GP logic. Just as $\langle K \rangle A$ internalized the judgment $K \,\mathsf{affirms}\, A$ in GP logic, $\langle K \rangle A$ now internalizes the judgment $(K\,\mathsf{affirms}\ A)\ \mathsf{at}\ I$. Although the formal meanings of the connectives must shift with the change from time-independent basic judgments to time-dependent ones, the connectives still retain their intuitive meanings. For example, $A \wedge B$ still behaves like a pair of A and B. This is made precise by the meta-theory in Section 3.3.2 that establishes a formal correspondence between GP logic and a fragment of non-linear η logic.

Despite the inclusion of the propositions of GP logic, there are three new proposition forms in non-linear η logic: $A @ I$, $C \dot{\supset} A$, and $C \dot{\wedge} A$. The proposition $A\ @\ I$ internalizes the new judgment $A[I]$, allowing us to legitimately combine it with the other logical connectives. For example, although $(A[I]) \supset B$ would violate the distinction between judgments and propositions, $(A\ @\ I) \supset B$ is a well-formed proposition.

$C \dot{\supset} A$ and $C \dot{\wedge} A$ are constraint implication and constraint conjuction propositions, respectively, adapted from Saranlı and Pfenning's Constrained Intuitionistic Linear Logic [21]. They permit the constraint domain to interact with the rest of the logic.

It should be noted that falsehood, $\bot$, is not included in the logic, stemming from the need to avoid security risks. If falsehood was included and, by some accident of policy management, a contradiction existed for any interval I, even an arbitrarily small one, then the judgment $\bot[I]$ would be derivable. From this judgment, any user would be able to give a valid proof of any judgment, including those allowing him to access protected resources. We therefore exclude falsehood from the logic to prevent this nightmare scenario from ever arising.

One consequence of the absence of falsehood is that policies to explicitly deny a group of users access cannot be written; only policies that explicitly *allow* a group of users access can be written. Stated differently, only whitelists, and not blacklists, can be created.

3.1.5 Inference Rules

Following the presentation of GP logic, we now state a few key proof rules and attempt to provide some intuition for them. As noted previously, we postpone the inference rules for the well-formedness of propositions, judgments, and contexts to Section 5.1 to avoid obscuring the key proof rules.

We begin by presenting the init rule that defines the nature of hypotheses:

$$\frac{\Sigma; \Psi \models I \supseteq I'}{\Sigma; \Psi; \Gamma, P[I] \Longrightarrow P[I']}\ \mathsf{init}$$

We would expect that, from the assumption that proposition A is true on interval I, it should be possible to prove that A is true on I. More generally, since truth on an interval refers to truth over the whole of that interval, it should be possible to prove that A is true on any subinterval I' of I from this assumption. The init rule captures this intuition, though, as in GP logic, it is restricted to atomic propositions P for technical reasons. The init rule in its full generality is proven admissible in Theorem 3.1 (see Section 3.3.1).

Next, consider the new connective: @.

$$\dfrac{\Sigma;\Psi;\Gamma \Longrightarrow A[I]}{\Sigma;\Psi;\Gamma \Longrightarrow A \mathbin{@} I[I']}\;@R \qquad \dfrac{\Sigma;\Psi;\Gamma, A \mathbin{@} I[I'], A[I] \Longrightarrow \gamma}{\Sigma;\Psi;\Gamma, A \mathbin{@} I[I'] \Longrightarrow \gamma}\;@L$$

The right rule $@R$ shows that establishing $A[I]$ is sufficient evidence for $A@I[I']$, for any interval I'. The left rule $@L$ allows the hypothesis $A \mathbin{@} I[I']$ to be used as $A[I]$.

Taken together, the rules for @ imply an equivalence between $A[I]$ and $A \mathbin{@} I[I']$ for any I', and also show that $A \mathbin{@} I$ internalizes the hybrid judgment $A[I]$. For example, establishing that "In 2008, it is true that 'During 1815–1821, Napoleon Bonaparte is in exile'" is equivalent to establishing that "During 1815–1821, Napoleon Bonaparte is in exile." In other words, whether it is true *now* that Napoleon was in exile depends only on whether it was true *then* that Napoleon was in exile.

Next, we examine the constraint connectives. First, the rules for constraint implication:

$$\dfrac{\Sigma;\Psi,C;\Gamma \Longrightarrow A[I]}{\Sigma;\Psi;\Gamma \Longrightarrow C \mathbin{\dot{\supset}} A[I]}\;\dot{\supset}R \qquad \dfrac{\Sigma;\Psi \models C \quad \Sigma;\Psi;\Gamma, C \mathbin{\dot{\supset}} A[I], A[I] \Longrightarrow \gamma}{\Sigma;\Psi;\Gamma, C \mathbin{\dot{\supset}} A[I] \Longrightarrow \gamma}\;\dot{\supset}L$$

$C \mathbin{\dot{\supset}} A$ represents the proposition A with the constraint precondition C. Thus, as formalized in the right rule $\dot{\supset}R$, verifying $C \mathbin{\dot{\supset}} A[I]$ involves verifying that A is true during interval I under the assumption that constraint C holds. The left rule $\dot{\supset}L$ states that to extract $A[I]$ from $C \mathbin{\dot{\supset}} A[I]$, one must simply establish the constraint precondition C.

The other constraint connective is constraint conjunction.

$$\dfrac{\Sigma;\Psi \models C \quad \Sigma;\Psi;\Gamma \Longrightarrow A[I]}{\Sigma;\Psi;\Gamma \Longrightarrow C \mathbin{\dot{\wedge}} A[I]}\;\dot{\wedge}R \qquad \dfrac{\Sigma;\Psi,C;\Gamma, C \mathbin{\dot{\wedge}} A[I], A[I] \Longrightarrow \gamma}{\Sigma;\Psi;\Gamma, C \mathbin{\dot{\wedge}} A[I] \Longrightarrow \gamma}\;\dot{\wedge}L$$

The right rule $\dot{\wedge}R$ requires that the constraint C holds and that A is true during interval I, reminiscent of the right rule for ordinary conjunction. The left rule $\dot{\wedge}L$ allows the hypothesis $C \mathbin{\dot{\wedge}} A[I]$ to be used by projecting out the two component hypotheses: C and $A[I]$.

Next, we consider the rules for the affirmation judgment $(K\ \mathsf{affirms}\ A)\ \mathsf{at}\ I$ and its internalization as $\langle K\rangle A$.

$$\dfrac{\Sigma;\Psi;\Gamma \Longrightarrow A[I]}{\Sigma;\Psi;\Gamma \Longrightarrow (K\ \mathsf{affirms}\ A)\ \mathsf{at}\ I}\;\mathsf{affirms} \qquad \dfrac{\Sigma;\Psi;\Gamma \Longrightarrow (K\ \mathsf{affirms}\ A)\ \mathsf{at}\ I}{\Sigma;\Psi;\Gamma \Longrightarrow \langle K\rangle A[I]}\;\langle\rangle R$$

$$\dfrac{\Sigma;\Psi;\Gamma, \langle K\rangle A[I], A[I] \Longrightarrow (K\ \mathsf{affirms}\ B)\ \mathsf{at}\ I' \quad \Sigma;\Psi \models I \supseteq I'}{\Sigma;\Psi;\Gamma, \langle K\rangle A[I] \Longrightarrow (K\ \mathsf{affirms}\ B)\ \mathsf{at}\ I'}\;\langle\rangle L$$

The $\mathsf{affirms}$ rule indicates that, during interval I, every principal K is prepared to affirm the truth of A on I if confronted with incontrovertible evidence of it: K cannot possibly ignore the evidence and must therefore affirm $A[I]$.

The right rule $\langle\rangle R$ shows that $\langle K\rangle A$ internalizes the affirmation judgment $(K\ \mathsf{affirms}\ A)\ \mathsf{at}\ I$. That is, by establishing $(K\ \mathsf{affirms}\ A)\ \mathsf{at}\ I$, one may conclude that the proposition $\langle K\rangle A$ is true on interval I.

The left rule $\langle\rangle L$ shows how to use an affirmation made by K during interval I. As in GP logic, the distinction between K's affirmations and truth disappears when trying to prove an affirmation made by K. However, with time-dependent affirmations, the disappearance of this distinction is only valid for affirmations made by K during a superinterval I of the interval I' for the affirmation made by K that is being established. Without the interval constraint, this rule would be incorrect. If I is not a superinterval of I', one cannot be assured that K still affirms A during all of interval I'.

Next, we examine implication. Implication interacts very strongly with time, as evidenced by the combination of parameters, constraints, and hybrid worlds in its right rule:

$$\frac{\Sigma, i{:}\mathsf{interval}; \Psi, I \supseteq i; \Gamma, A[i] \Longrightarrow B[i]}{\Sigma; \Psi; \Gamma \Longrightarrow A \supset B[I]}\ \supset R$$

$$\frac{\Sigma; \Psi; \Gamma, A \supset B[I] \Longrightarrow A[I'] \quad \Sigma; \Psi \models I \supseteq I' \quad \Sigma; \Psi; \Gamma, A \supset B[I], B[I'] \Longrightarrow \gamma}{\Sigma; \Psi; \Gamma, A \supset B[I] \Longrightarrow \gamma}\ \supset L$$

The judgment $A \supset B[I]$ may be intuitively thought of as a plan for converting A to B that is available during any subinterval of I. Such a conversion can be established by deriving $B[i]$ under the assumption $A[i]$, parametrically in the arbitrary subinterval i of I. The parameter and corresponding constraint ensure that the conversion is valid at every time in I. This intuition is formalized in the right rule $\supset R$.

The conversion intuition also appears in the left rule $\supset L$. The plan $A \supset B[I]$ for converting A to B can be carried out to produce $B[I']$ from $A[I']$, provided I' is a subinterval of I. The rule is incorrect without the subinterval proviso because the plan would not be available at an arbitrary I'.

Finally, the remaining connectives do not interact extensively with time. One such connective is conjunction:

$$\frac{\Sigma; \Psi; \Gamma \Longrightarrow A[I] \quad \Sigma; \Psi; \Gamma \Longrightarrow B[I]}{\Sigma; \Psi; \Gamma \Longrightarrow A \wedge B[I]}\ \wedge R$$

$$\frac{\Sigma; \Psi; \Gamma, A \wedge B[I], A[I] \Longrightarrow \gamma}{\Sigma; \Psi; \Gamma, A \wedge B[I] \Longrightarrow \gamma}\ \wedge L_1 \qquad \frac{\Sigma; \Psi; \Gamma, A \wedge B[I], B[I] \Longrightarrow \gamma}{\Sigma; \Psi; \Gamma, A \wedge B[I] \Longrightarrow \gamma}\ \wedge L_2$$

To show that $A \wedge B$ is true on interval I, it is sufficient to show both that A is true on I and that B is true on I; this is captured by the right rule $\wedge R$. The left rules $\wedge L_1$ and $\wedge L_2$ show that both A and B are true on I if $A \wedge B$ is true on I. These right and left rules do not manipulate the interval annotations; they are the same as the rules in first-order logic for conjunction, but are tagged with intervals.

The remaining proof rules are given in Figure 3.1.

$\boxed{\text{Initial Rule}}$

$$\frac{\Sigma; \Psi \models I \supseteq I'}{\Sigma; \Psi; \Gamma, P[I] \Longrightarrow P[I']}\ \mathsf{init}$$

$\boxed{A \mathbin{@} I}$

$$\frac{\Sigma; \Psi; \Gamma \Longrightarrow A[I]}{\Sigma; \Psi; \Gamma \Longrightarrow A \mathbin{@} I[I']}\ @R \qquad\qquad \frac{\Sigma; \Psi; \Gamma, A \mathbin{@} I[I'], A[I] \Longrightarrow \gamma}{\Sigma; \Psi; \Gamma, A \mathbin{@} I[I'] \Longrightarrow \gamma}\ @L$$

$\boxed{\text{Constraints}}$

$$\frac{\Sigma; \Psi, C; \Gamma \Longrightarrow A[I]}{\Sigma; \Psi; \Gamma \Longrightarrow C \mathbin{\dot{\supset}} A[I]}\ \dot{\supset}R \qquad\qquad \frac{\Sigma; \Psi \models C \quad \Sigma; \Psi; \Gamma, C \mathbin{\dot{\supset}} A[I], A[I] \Longrightarrow \gamma}{\Sigma; \Psi; \Gamma, C \mathbin{\dot{\supset}} A[I] \Longrightarrow \gamma}\ \dot{\supset}L$$

$$\frac{\Sigma; \Psi \models C \quad \Sigma; \Psi; \Gamma \Longrightarrow A[I]}{\Sigma; \Psi; \Gamma \Longrightarrow C \mathbin{\dot{\wedge}} A[I]}\ \dot{\wedge}R \qquad\qquad \frac{\Sigma; \Psi, C; \Gamma, C \mathbin{\dot{\wedge}} A[I], A[I] \Longrightarrow \gamma}{\Sigma; \Psi; \Gamma, C \mathbin{\dot{\wedge}} A[I] \Longrightarrow \gamma}\ \dot{\wedge}L$$

$\boxed{\text{Affirmation and } \langle K \rangle A}$

$$\frac{\Sigma; \Psi; \Gamma \Longrightarrow A[I]}{\Sigma; \Psi; \Gamma \Longrightarrow (K \mathsf{\ affirms\ } A) \mathsf{\ at\ } I}\ \mathsf{affirms} \qquad\qquad \frac{\Sigma; \Psi; \Gamma \Longrightarrow (K \mathsf{\ affirms\ } A) \mathsf{\ at\ } I}{\Sigma; \Psi; \Gamma \Longrightarrow \langle K \rangle A[I]}\ \langle\rangle R$$

$$\frac{\Sigma; \Psi; \Gamma, \langle K \rangle A[I], A[I] \Longrightarrow (K \mathsf{\ affirms\ } B) \mathsf{\ at\ } I' \quad \Sigma; \Psi \models I \supseteq I'}{\Sigma; \Psi; \Gamma, \langle K \rangle A[I] \Longrightarrow (K \mathsf{\ affirms\ } B) \mathsf{\ at\ } I'}\ \langle\rangle L$$

$\boxed{\text{Other Connectives}}$

$$\frac{\Sigma; \Psi; \Gamma \Longrightarrow A[I] \quad \Sigma; \Psi; \Gamma \Longrightarrow B[I]}{\Sigma; \Psi; \Gamma \Longrightarrow A \wedge B[I]}\ \wedge R$$

$$\frac{\Sigma; \Psi; \Gamma, A \wedge B[I], A[I] \Longrightarrow \gamma}{\Sigma; \Psi; \Gamma, A \wedge B[I] \Longrightarrow \gamma}\ \wedge L_1 \qquad\qquad \frac{\Sigma; \Psi; \Gamma, A \wedge B[I], B[I] \Longrightarrow \gamma}{\Sigma; \Psi; \Gamma, A \wedge B[I] \Longrightarrow \gamma}\ \wedge L_2$$

$$\frac{}{\Sigma; \Psi; \Gamma \Longrightarrow \top[I]}\ \top R$$

$$\frac{\Sigma; \Psi; \Gamma \Longrightarrow A[I]}{\Sigma; \Psi; \Gamma \Longrightarrow A \vee B[I]}\ \vee R_1 \qquad\qquad \frac{\Sigma; \Psi; \Gamma \Longrightarrow B[I]}{\Sigma; \Psi; \Gamma \Longrightarrow A \vee B[I]}\ \vee R_2$$

$$\frac{\Sigma; \Psi; \Gamma, A \vee B[I], A[I] \Longrightarrow \gamma \quad \Sigma; \Psi; \Gamma, A \vee B[I], B[I] \Longrightarrow \gamma}{\Sigma; \Psi; \Gamma, A \vee B[I] \Longrightarrow \gamma}\ \vee L$$

$$\frac{\Sigma, i{:}\mathsf{interval}; \Psi, I \supseteq i; \Gamma, A[i] \Longrightarrow B[i]}{\Sigma; \Psi; \Gamma \Longrightarrow A \supset B[I]}\ \supset R$$

$$\frac{\Sigma; \Psi; \Gamma, A \supset B[I] \Longrightarrow A[I'] \quad \Sigma; \Psi \models I \supseteq I' \quad \Sigma; \Psi; \Gamma, A \supset B[I], B[I'] \Longrightarrow \gamma}{\Sigma; \Psi; \Gamma, A \supset B[I] \Longrightarrow \gamma}\ \supset L$$

$$\frac{\Sigma, x{:}s; \Psi; \Gamma \Longrightarrow A[I]}{\Sigma; \Psi; \Gamma \Longrightarrow \forall x{:}s.A[I]}\ \forall R \qquad\qquad \frac{\Sigma \vdash t{:}s \quad \Sigma; \Psi; \Gamma, \forall x{:}s.A[I], [t/x]A[I] \Longrightarrow \gamma}{\Sigma; \Psi; \Gamma, \forall x{:}s.A[I] \Longrightarrow \gamma}\ \forall L$$

$$\frac{\Sigma \vdash t{:}s \quad \Sigma; \Psi; \Gamma \Longrightarrow [t/x]A[I]}{\Sigma; \Psi; \Gamma \Longrightarrow \exists x{:}s.A[I]}\ \exists R \qquad\qquad \frac{\Sigma, x{:}s; \Psi; \Gamma, \exists x{:}s.A[I], A[I] \Longrightarrow \gamma}{\Sigma; \Psi; \Gamma, \exists x{:}s.A[I] \Longrightarrow \gamma}\ \exists L$$

Figure 3.1: The inference rules for non-linear η logic.

Before concluding this section, we give a few judgments derivable and a few judgments not derivable in non-linear η logic to illustrate the logic's properties.

1. $\cdot;\cdot;\cdot \not\Longrightarrow (A \mathbin{@} I) \supset (A \mathbin{@} I')[I'']$
2. $\cdot;\cdot;\cdot \Longrightarrow (A \mathbin{@} I) \supset (A \mathbin{@} I')[I'']$ if $\cdot;\cdot \models I \supseteq I'$
3. $\cdot;\cdot;\cdot \Longrightarrow (A \mathbin{@} I) \supset (A \mathbin{@} I \mathbin{@} I')[I'']$
4. $\cdot;\cdot;\cdot \Longrightarrow (A \mathbin{@} I \mathbin{@} I') \supset (A \mathbin{@} I)[I'']$
5. $\cdot;\cdot;\cdot \Longrightarrow ((A \wedge B) \mathbin{@} I) \supset ((A \mathbin{@} I) \wedge (B \mathbin{@} I))[I'']$
6. $\cdot;\cdot;\cdot \Longrightarrow ((A \mathbin{@} I) \wedge (B \mathbin{@} I)) \supset ((A \wedge B) \mathbin{@} I)[I'']$
7. $\cdot;\cdot;\cdot \Longrightarrow A \supset \langle K\rangle A[I]$
8. $\cdot;\cdot;\cdot \Longrightarrow \langle K\rangle\langle K\rangle A \supset \langle K\rangle A[I]$
9. $\cdot;\cdot;\cdot \Longrightarrow \langle K\rangle(A \supset B) \supset \langle K\rangle A \supset \langle K\rangle B[I]$
10. $\cdot;\cdot;\cdot \not\Longrightarrow \langle K\rangle A \supset A[I]$
11. $\cdot;\cdot;\cdot \not\Longrightarrow A[I]$

Judgment 1 shows that truth on one interval does not entail truth on another interval, in general; the intervals may be unrelated. When the intervals are related by inclusion, the entailment does hold, as in judgment 2. Judgments 3 and 4 imply that only the innermost interval matters. This relates to the previously mentioned equivalence between $A[I]$ and $A \mathbin{@} I[I']$. Judgments 5 and 6 show the distributivity of @ over $\wedge$, an intuitive and desirable property. Judgments 7–9 indicate that $\langle K\rangle$ remains a lax modality, as in GP logic. Judgment 10 illustrates the difference between truth and affirmation: truth entails affirmation, as seen in judgment 7, but affirmation does not entail truth. By demonstrating that arbitrary propositions are not a priori true at arbitrary intervals, judgment 11 establishes the consistency of non-linear η logic.

3.2 Examples

Introduction coming soon.

3.2.1 Office Entry

Recall, from Section 2.2.1, the office entry example that was based on the Grey system. This example assumed an administrating principal, admin, that controlled entry to the offices, named each office according to its owner, and used the predicate may_enter, where may_enter(K_2, K_1) meant that K_2 may enter K_1's office. The two policies ((2.1) and (2.2)) proposed for a GP-logic based PCA architecture were:

$$\mathsf{own} : \langle \mathrm{admin}\rangle(\forall K{:}\mathsf{principal}.\mathsf{may\_enter}(K, K))\ \mathsf{true}$$

$$\mathsf{trust} : \langle \mathrm{admin}\rangle(\forall K_1{:}\mathsf{principal}.\forall K_2{:}\mathsf{principal}.\langle K_1\rangle\mathsf{may\_enter}(K_2, K_1) \supset \mathsf{may\_enter}(K_2, K_1))\ \mathsf{true}$$

The first of these policies allowed every office owner to enter her own office. The second policy allowed an office owner to make decisions about who may enter her office, decisions which the administrator trusted.

The above GP logic policies were sufficient for controlling who could enter an office, but not for controlling *when* that person could enter. This deficiency resulted from the inability of GP logic to reason with time internally. Now that we have developed non-linear η logic as an authorization logic with time, it is natural to check that the new logic is expressive enough to handle time-based office entry policies.

First, consider creating a non-linear η logic analogue of the own policy. Because non-linear η logic includes all connectives from GP logic and because these connectives retain their intuitive meanings, a natural attempt uses the same proposition as own:

$$\langle \text{admin} \rangle (\forall K{:}\mathsf{principal}.\mathsf{may\_enter}(K, K))[?]$$

At the moment, this judgment is incomplete: the time interval over which the proposition is true has not yet been specified (indicated by '?').

What interval should be used? It must be the same as the interval over which the policy will be valid. If the administrator wants to allow each office owner to enter her own office only during interval I, then the interval for this policy should be I. In the setting of academic offices, it would seem unusual for an office owner to be prevented from entering her office at any time. So, in this specific instance, the interval is $(-\infty, \infty)$. The non-linear η logic analogue of own is then:

$$\mathsf{own}' : \langle \text{admin} \rangle (\forall K{:}\mathsf{principal}.\mathsf{may\_enter}(K, K))[(-\infty, \infty)] \tag{3.1}$$

This policy means that "At all times, the administrator says that each principal K may enter her own office at any time."

Note that the administrator need not commit to a policy for an extended period of time. For example, suppose that the administrator only wants to commit to allowing an office owner to enter her own office during 2008. The administrator would issue the policy with 2008 as its validity interval. If the administrator later chooses to extend the policy through 2009, he can reissue the same policy with the new interval 2009. If, instead, the administrator chooses not to renew the policy, he simply does nothing: the 2008 version will no longer be valid in 2009.

Next, consider creating an analogue of the trust policy. Again, we use the same proposition as in trust. For concreteness, we choose $(-\infty, \infty)$ as the validity interval, but it should be noted that any desired interval could be used. The policy is then:

$$\mathsf{trust}' : \langle \text{admin} \rangle (\forall K_1{:}\mathsf{principal}.\forall K_2{:}\mathsf{principal}.\langle K_1 \rangle \mathsf{may\_enter}(K_2, K_1) \supset \mathsf{may\_enter}(K_2, K_1))[(-\infty, \infty)] \tag{3.2}$$

This policy means that "At all times, the administrator says that, for all pairs of principals K_1 and K_2, if K_1 says K_2 may enter K_1's office at some time, then K_2 may indeed enter K_1's office at that time."

With this policy, we can now reconsider the situation of the professor Alice and her graduate student Bob. Recall that Alice is out of the office on May 7, 2008 and that Bob needs to retrieve a paper from Alice's office. Alice agrees to authorize Bob to enter her office, but only for that day. So, she issues the following credential:

$$\mathcal{C}_0 : \langle \text{Alice} \rangle \mathsf{may\_enter}(\text{Bob}, \text{Alice})[5/7/08]$$

At some time t, Bob will approach Alice's office door and request access using his cell phone. Before the door will unlock, he must present a correct proof of:

$$\text{Alice:}\mathsf{principal}, \text{Bob:}\mathsf{principal}; \cdot; \mathsf{own}', \mathsf{trust}', \mathcal{C}_0 \Longrightarrow \langle \text{admin} \rangle \mathsf{may\_enter}(\text{Bob}, \text{Alice})[[t, t]]$$

Provided that t is some time during May 7, 2008 (formally, $\models 5/7/08 \supseteq [t, t]$) Bob's phone can construct a correct proof by applying the trust' policy to the credential $\mathcal{C}_0$ that Alice supplied, and Bob will be granted access. If t is not during May 7, 2008 (formally $\not\models 5/7/08 \supseteq [t, t]$), that method cannot be followed to construct a correct proof, and Bob will not be granted access.

As is evident from this example, non-linear η logic permits the expression of a richer set of policies than is possible in GP logic. However, the logic is still not sufficiently expressive. The deficiency occurs even in this small office entry example. Alice can now restrict the times during which Bob may access her office, but it is not possible to restrict the *number* of times Bob may enter. Specifically, because the credential $\mathcal{C}_0$ is never consumed during use, Bob may enter the office as many times as he wants during ???

Because non-linear η logic models the expiration of, but not the consumption of, credentials, it is natural to consider extending the logic with linearity, just as Garg and Pfenning cleanly added linearity to an authorization logic without time [12]. This effort toward a linear η logic that can model consumable credentials is the focus of the following chapter.

3.2.2 Journal Publication

To further demonstrate the increased expressiveness of non-linear η logic, consider a peer-review publication system as employed by academic journals. This example uses time in a more complex way than the previous example and also illustrates the use of time-based constraints and constraint implication.

We postulate the existence of two application-specific sorts: the sort $\mathsf{journal}$ of academic journals and the sort $\mathsf{article}$ of journal articles. To ease the notation, we also use the constraint form $t \in I$ as an abbreviation for $I \supseteq [t, t]$. The following predicates are required:

$\mathsf{is\_approved}(A, K, J)$	Article A is approved by principal K for publication in journal J.
$\mathsf{is\_reviewer}(R, A, J)$	Principal R is the reviewer for article A submitted to journal J.
$\mathsf{is\_editor}(E, J)$	Principal E is an editor for journal J.
$\mathsf{is\_published}(A, J)$	Article A is published in journal J.

Journal J appoints E as an editor for term I by issuing the credential $\langle J \rangle \mathsf{is\_editor}(E, J)[I]$. One of an editor's duties is to assign reviewers to articles submitted to the journal. Editor E assigns principal R as the reviewer for article A from time t onward by issuing the credential $\langle E \rangle \mathsf{is\_reviewer}(R, A, J)[[t, \infty)]$.

For simplicity, we assume that each article has at most one reviewer, justifing the reference to a reviewer of an article as *the* reviewer.

Another one of an editor's duties is to process reviews as they come back from reviewers. Editor E accomplishes this by signing the following credential:

$$\begin{array}{ll}\mathsf{approve} : \langle E\rangle(\forall R{:}\mathsf{principal}.\forall t_a{:}\mathsf{time}. & \\ & \langle R\rangle\mathsf{is\_approved}(A, R, J) \mathbin{@} [t_a, t_a] \supset \\ & \mathsf{is\_reviewer}(R, A, J) \mathbin{@} [t_a, t_a] \supset \\ & (t_a \in I_E) \mathbin{\dot{\supset}} \\ & \quad \mathsf{is\_approved}(A, E, J) \mathbin{@} [t_a, \infty))[(-\infty, \infty)]\end{array}$$

If principal R decides to approve article A for publication in journal J, he can submit a positive review at time t_a by issuing the certificate $\langle R\rangle\mathsf{is\_approved}(A, R, J)[[t_a, t_a]]$. Provided that editor E agrees that R is the reviewer of article A at time t_a and that $t_a \in I_E$, E will accept R's review and approve the article for publication from time t_a onward. If R is not the reviewer of article A or if $t_a \notin I_E$, then the review will not be accepted.

Note that, unlike the policies we have previously seen, $\mathsf{approve}$ is not a fixed policy, but rather a template. When E signs the credential, he must instantiate I_E with the interval over which he will accept reviews.

In a similar way, each journal must specify the conditions under which it accepts articles approved by editors. This is done by issuing the following credential:

$$\begin{array}{ll}\mathsf{publish} : \langle J\rangle(\forall E{:}\mathsf{principal}.\forall t_a{:}\mathsf{time}. & \\ & \langle E\rangle\mathsf{is\_approved}(A, E, J) \mathbin{@} [t_a, t_a] \supset \\ & \mathsf{is\_editor}(E, J) \mathbin{@} [t_a, t_a] \supset \\ & (t_a \in I_J) \mathbin{\dot{\supset}} \\ & \quad \mathsf{is\_published}(A, J) \mathbin{@} [t_a, \infty))[(-\infty, \infty)]\end{array}$$

If principal E approves article A for publication in journal J, he issues the credential $\langle E\rangle\mathsf{is\_approved}(A, E, J)[[t_a, t_a]]$. If journal J has appointed E as editor during time t_a and if $t_a \in I_J$, J will accept editor E's approval and publish the article from t_a onward. Again, this policy is a template: J must instantiate I_J with the interval during which it will accept articles for publication.

3.3 Meta-theory and Correspondence to GP Logic

Introduction coming soon.

3.3.1 Meta-theory

The meta-theory for non-linear η logic is slightly more complicated than that of GP logic because of the addition of time. But, it still serves to increase confidence in the soundness of the logic by providing a kind of "sanity check."

As in GP logic, we are still interested in verifying the identity principle. However, with the shift in underlying judgments to hybrid, time-dependent forms, the statement of the theorem must change. For any proposition A, it should be possible to conclude from the hypothesis $A[I]$ that $A[I']$, provided I' is a subinterval of I. This is formalized in the following theorem.

Theorem 3.1 (Identity)**.** *For all propositions A, if $\Sigma; \Psi \models I \supseteq I'$, then $\Sigma; \Psi; \Gamma, A[I] \Longrightarrow A[I']$.*

Proof. By structural induction on A. □

A natural time-dependent property to expect of non-linear η logic is the notion of subsumption. For example, whenever one can prove that A is true on interval I, it should be possible, for any subinterval I' of I, to construct a similar proof that A is true on I'. This can be easily generalized to affirmations. Because this type of subsumption occurs in proof conclusions and not assumptions, it appears to the right of the $\Longrightarrow$ symbol in a hypothetical judgment. It is therefore termed right subsumption.

Theorem 3.2 (Right Subsumption)**.**

1. *If $\Sigma; \Psi; \Gamma \Longrightarrow A[I]$ and $\Sigma; \Psi \models I \supseteq I'$, then $\Sigma; \Psi; \Gamma \Longrightarrow A[I']$.*
2. *If $\Sigma; \Psi; \Gamma \Longrightarrow (K \ \mathsf{affirms}\ A) \ \mathsf{at}\ I$ and $\Sigma; \Psi \models I \supseteq I'$, then $\Sigma; \Psi; \Gamma \Longrightarrow (K \ \mathsf{affirms}\ A) \ \mathsf{at}\ I'$.*

Proof. By simultaneous structural induction on the first given derivation. □

Subsumption can also occur for hypotheses. If interval I is a superinterval of I', the assumption that A is true on interval I' is at least as powerful as assuming that A is true on I': the former assumption contains as much (and possibly more) information as the latter. Because hypotheses appear on the left side of the $\Longrightarrow$ symbol in a hypothetical judgment, this kind of subsumption is termed left subsumption.

Theorem 3.3 (Left Subsumption)**.** *If $\Sigma; \Psi; \Gamma, A[I'] \Longrightarrow \gamma$ and $\Sigma; \Psi \models I \supseteq I'$, then $\Sigma; \Psi; \Gamma, A[I] \Longrightarrow \gamma$.*

Proof. By nested induction on the structures of A and the first given derivation. □

Finally, we can reconsider cut elimination in the context of non-linear η logic. The admissibility of cut for the truth judgment remains relatively unchanged: a proof of $A[I]$ can replace the assumption $A[I]$ of any other proof. However, the admissibility of cut for the affirmation judgment changes in a significant way. As argued in the description of the $\langle\rangle L$ rule, an affirmation made by K during interval I is equivalent to truth, but only if we are currently reasoning about the beliefs that K holds during a subinterval I'. Thus, a proof of $(K \ \mathsf{affirms}\ A) \ \mathsf{at}\ I$ can replace the assumption $A[I]$ in a proof of $(K \ \mathsf{affirms}\ B) \ \mathsf{at}\ I'$, provided that I is a superinterval of I'.

Theorem 3.4 (Admissibility of Cut)**.**

1. *If $\Sigma; \Psi; \Gamma \Longrightarrow A[I]$ and $\Sigma; \Psi; \Gamma, A[I] \Longrightarrow \gamma$, then $\Sigma; \Psi; \Gamma \Longrightarrow \gamma$.*

2. If $\Sigma; \Psi; \Gamma \Longrightarrow (K \ \mathsf{affirms}\ A)\ \mathsf{at}\ I$*,* $\Sigma; \Psi; \Gamma, A[I] \Longrightarrow (K \ \mathsf{affirms}\ B)\ \mathsf{at}\ I'$*, and* $\Sigma; \Psi \models I \supseteq I'$*, then* $\Sigma; \Psi; \Gamma \Longrightarrow (K \ \mathsf{affirms}\ B)\ \mathsf{at}\ I'$*.*

Proof. By simultaneous nested induction on the structures of A and the first two given derivations. □

The above meta-theorems have been mechanically verified using the Twelf logical framework []. The Twelf proofs are available at `http://www.andrew.cmu.edu/user/hdeyoung/etalogic/???`.

3.3.2 Correspondence to GP Logic

Upon careful comparison of the rules of GP logic and the rules of non-linear η logic, a correspondence becomes evident. For example, consider the $\wedge R$ and $\langle\rangle L$ rules:

GP Logic	**Non-linear η Logic**
$\dfrac{\Sigma; \Gamma \Longrightarrow A \quad \Sigma; \Gamma \Longrightarrow B}{\Sigma; \Gamma \Longrightarrow A \wedge B}\ \wedge R$	$\dfrac{\Sigma; \Psi; \Gamma \Longrightarrow A[I] \quad \Sigma; \Psi; \Gamma \Longrightarrow B[I]}{\Sigma; \Psi; \Gamma \Longrightarrow A \wedge B[I]}\ \wedge R$
$\dfrac{\Sigma; \Gamma, \langle K\rangle A, A \Longrightarrow K\ \mathsf{affirms}\ B}{\Sigma; \Gamma, \langle K\rangle A \Longrightarrow K\ \mathsf{affirms}\ B}\ \langle\rangle L$	$\dfrac{\Sigma; \Psi \models I \supseteq I' \quad \Sigma; \Psi; \Gamma, \langle K\rangle A[I], A[I] \Longrightarrow (K\ \mathsf{affirms}\ B)\ \mathsf{at}\ I'}{\Sigma; \Psi; \Gamma, \langle K\rangle A[I] \Longrightarrow (K\ \mathsf{affirms}\ B)\ \mathsf{at}\ I'}\ \langle\rangle L$

Because there are no notions of time or constraints in GP logic, the correspondence does not extend to these constructs.

The above intuition suggests that GP logic can be encoded into non-linear η logic. Let $\vec{I}$ denote a list of time intervals. Also, if $\Gamma = A_1, \ldots, A_n$ is a GP logic context and $\vec{I} = I_1, \ldots, I_n$, let $\Gamma[\vec{I}]$ be the non-linear η logic context $A_1[I_1], \ldots, A_n[I_n]$.[1] Finally, define a translation for signatures such that $\overline{\Sigma}$ is Σ with $\mathsf{interval}$ and time parameters removed.

This permits us to state the following theorem.

Theorem 3.5.
1. If $\overline{\Sigma}; \Gamma \Longrightarrow A$ *and* $\Sigma; \Psi \models I \supseteq I'$ *for all* $I \in \vec{I}$*, then* $\Sigma; \Psi; \Gamma[\vec{I}] \Longrightarrow A[I']$*.*
2. If $\overline{\Sigma}; \Gamma \Longrightarrow K \mathsf{affirms} A$ *and* $\Sigma; \Psi \models I \supseteq I'$ *for all* $I \in \vec{I}$*, then* $\Sigma; \Psi; \Gamma[\vec{I}] \Longrightarrow (K\ \mathsf{affirms}\ A)\ \mathsf{at}\ I'$*.*

Proof. By simultaneous structural induction on the first given derivation. □

Theorem 3.6.
1. If $\Sigma; \Psi; \Gamma[\vec{I}] \Longrightarrow A[I']$*, then* $\overline{\Sigma}; \Gamma \Longrightarrow A$*.*
2. If $\Sigma; \Psi; \Gamma[\vec{I}] \Longrightarrow (K\ \mathsf{affirms}\ A)\ \mathsf{at}\ I'$*, then* $\overline{\Sigma}; \Gamma \Longrightarrow K\ \mathsf{affirms}\ A$*.*

Proof. By simultaneous structural induction on the given derivation. □

[1] There is an implicit identity translation from GP logic propositions to non-linear η logic propositions here.

3.4 Conclusion

Coming soon.

Chapter 4

Linear η Logic

Coming soon.

4.1 An Overview of Linear Logic

Menu Item	Linear Logical Formula
Choice of Soup du Jour (M–W: Vegetable; R–F: Chicken Noodle) or Salad	$((\mathrm{V} \oplus \mathrm{CN}) \, \& \, \mathrm{S})$
and Roast Beef Sandwich	$\otimes \, \mathrm{RBS}$

4.2 Logical System

4.2.1 Judgments

4.2.2 Propositions

4.2.3 Inference Rules

4.3 Examples

4.3.1 Office Entry

4.3.2 Filling Painkiller Prescriptions

4.3.3 A Homework Assignment Administration System

4.4 Meta-theory

4.4.1 Internal Meta-theory

4.5 Conclusion

$\boxed{\text{Basic Rules}}$

$$\frac{\Sigma;\Psi \models I \supseteq I'}{\Sigma;\Psi;\Gamma;P[I] \Longrightarrow P[I']}\ \mathsf{init} \qquad \frac{\Sigma;\Psi;\Gamma,A\llbracket I\rrbracket;\Delta,A[I] \Longrightarrow \gamma}{\Sigma;\Psi;\Gamma,A\llbracket I\rrbracket;\Delta \Longrightarrow \gamma}\ \mathsf{copy}$$

$\boxed{A \mathbin{@} I}$

$$\frac{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow A[I]}{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow A \mathbin{@} I[I']}\ @R \qquad \frac{\Sigma;\Psi;\Gamma;\Delta,A[I] \Longrightarrow \gamma}{\Sigma;\Psi;\Gamma;\Delta,A \mathbin{@} I[I'] \Longrightarrow \gamma}\ @L$$

$\boxed{\text{Constraints}}$

$$\frac{\Sigma;\Psi,C;\Gamma;\Delta \Longrightarrow A[I]}{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow C \mathbin{\dot\supset} A[I]}\ \dot\supset R \qquad \frac{\Sigma;\Psi \models C \quad \Sigma;\Psi;\Gamma;\Delta,A[I] \Longrightarrow \gamma}{\Sigma;\Psi;\Gamma;\Delta,C \mathbin{\dot\supset} A[I] \Longrightarrow \gamma}\ \dot\supset L$$

$$\frac{\Sigma;\Psi \models C \quad \Sigma;\Psi;\Gamma;\Delta \Longrightarrow A[I]}{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow C \mathbin{\dot\wedge} A[I]}\ \dot\wedge R \qquad \frac{\Sigma;\Psi,C;\Gamma;\Delta,A[I] \Longrightarrow \gamma}{\Sigma;\Psi;\Gamma;\Delta,C \mathbin{\dot\wedge} A[I] \Longrightarrow \gamma}\ \dot\wedge L$$

$\boxed{\text{Affirmation and } \langle K\rangle A}$

$$\frac{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow A[I]}{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow (K \mathsf{\ affirms\ } A) \mathsf{\ at\ } I}\ \mathsf{affirms}$$

$$\frac{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow (K \mathsf{\ affirms\ } A) \mathsf{\ at\ } I}{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow \langle K\rangle A[I]}\ \langle\rangle R$$

$$\frac{\Sigma;\Psi;\Gamma;\Delta,A[I] \Longrightarrow (K \mathsf{\ affirms\ } B) \mathsf{\ at\ } I' \quad \Sigma;\Psi \models I \supseteq I'}{\Sigma;\Psi;\Gamma;\Delta,\langle K\rangle A[I] \Longrightarrow (K \mathsf{\ affirms\ } B) \mathsf{\ at\ } I'}\ \langle\rangle L$$

$\boxed{\text{Other Connectives}}$

$$\frac{}{\Sigma;\Psi;\Gamma;\cdot \Longrightarrow \mathbf{1}[I]}\ \mathbf{1}R \qquad \frac{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow \gamma}{\Sigma;\Psi;\Gamma;\Delta,\mathbf{1}[I] \Longrightarrow \gamma}\ \mathbf{1}L$$

$$\frac{\Sigma;\Psi;\Gamma;\Delta_1 \Longrightarrow A[I] \quad \Sigma;\Psi;\Gamma;\Delta_2 \Longrightarrow B[I]}{\Sigma;\Psi;\Gamma;\Delta_1,\Delta_2 \Longrightarrow A \otimes B[I]}\ \otimes R$$

$$\frac{\Sigma;\Psi;\Gamma;\Delta,A[I],B[I] \Longrightarrow \gamma}{\Sigma;\Psi;\Gamma;\Delta,A \otimes B[I] \Longrightarrow \gamma}\ \otimes L \qquad \frac{}{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow \top[I]}\ \top R$$

$$\frac{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow A[I] \quad \Sigma;\Psi;\Gamma;\Delta \Longrightarrow B[I]}{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow A \mathbin{\&} B[I]}\ \&R$$

$$\frac{\Sigma;\Psi;\Gamma;\Delta,A[I] \Longrightarrow \gamma}{\Sigma;\Psi;\Gamma;\Delta,A \mathbin{\&} B[I] \Longrightarrow \gamma}\ \&L_1 \qquad \frac{\Sigma;\Psi;\Gamma;\Delta,B[I] \Longrightarrow \gamma}{\Sigma;\Psi;\Gamma;\Delta,A \mathbin{\&} B[I] \Longrightarrow \gamma}\ \&L_2$$

$$\frac{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow A[I]}{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow A \oplus B[I]}\ \oplus R_1 \qquad \frac{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow B[I]}{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow A \oplus B[I]}\ \oplus R_2$$

$$\frac{\Sigma;\Psi;\Gamma;\Delta,A[I] \Longrightarrow \gamma \quad \Sigma;\Psi;\Gamma;\Delta,B[I] \Longrightarrow \gamma}{\Sigma;\Psi;\Gamma;\Delta,A \oplus B[I] \Longrightarrow \gamma}\ \oplus L$$

$$\frac{\Sigma,i{:}\mathsf{interval};\Psi,I \supseteq i;\Gamma;\Delta,A[i] \Longrightarrow B[i]}{\Sigma;\Psi;\Gamma;\Delta \Longrightarrow A \multimap B[I]}\ \multimap R$$

$$\frac{\Sigma;\Psi;\Gamma;\Delta_1 \Longrightarrow A[I'] \quad \Sigma;\Psi \models I \supseteq I' \quad \Sigma;\Psi;\Gamma;\Delta_2,B[I'] \Longrightarrow \gamma}{\Sigma;\Psi;\Gamma;\Delta_1,\Delta_2,A \multimap B[I] \Longrightarrow \gamma}\ \multimap L$$

$$\frac{\Sigma;\Psi;\Gamma;\cdot \Longrightarrow A[I]}{}\ !R \qquad \frac{\Sigma;\Psi;\Gamma,A\llbracket I\rrbracket;\Delta \Longrightarrow \gamma}{}\ !L$$

Chapter 5

Implementing a Proof Checker for Linear η Logic

Coming soon.

5.1 Proof Checker Inference Rules

Chapter 6

Conclusion

Coming soon.

Bibliography

[1] Martín Abadi. Access control in a core calculus of dependency. In *Proceedings of the 2006 ACM International Conference on Functional Programming (ICFP '06)*, pages 263–273, New York, New York, 2006. ACM Press.

[2] Martín Abadi, Michael Burrows, Butler Lampson, and Gordon Plotkin. A calculus for access control in distributed systems. *ACM Transactions on Programming Languages and Systems*, 15(4):706–734, September 1993.

[3] Andrew W. Appel and Edward W. Felten. Proof-carrying authentication. In Gene Tsudik, editor, *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pages 52–62, Singapore, November 1999. ACM Press.

[4] Lujo Bauer. *Access Control for the Web via Proof-Carrying Authorization.* PhD thesis, Princeton University, November 2003.

[5] Lujo Bauer, Scott Garriss, Jonathan M. McCune, Michael K. Reiter, Jason Rouse, and Peter Rutenbar. Device-enabled authorization in the Grey system. In *Information Security: 8th International Conference, ISC 2005*, volume 3650 of *Lecture Notes in Computer Science*, pages 431–445, September 2005.

[6] Lujo Bauer, Scott Garriss, and Michael K. Reiter. Distributed proving in access-control systems. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 81–95, May 2005.

[7] Moritz Y. Becker, Cédric Fournet, and Andrew D. Gordon. Design and semantics of a decentralized authorization language. In *Proceedings of the 20th IEEE Computer Security Foundations Symposium (CSF '07)*, pages 3–15, Venice, Italy, July 2007. IEEE Computer Society Press.

[8] Torben Braüner and Valeria de Paiva. Towards constructive hybrid logic. In *Electronic Proceedings of Methods for Modalities*, September 2003.

[9] Rohit Chadha, Damiano Macedonio, and Vladimiro Sassone. A hybrid intuitionistic logic: Semantics and decidability. *Journal of Logic and Computation*, 16(1):27–59, 2006.

[10] Bor-Yuh Evan Chang, Kaustuv Chaudhuri, and Frank Pfenning. A judgmental analysis of linear logic. Technical Report CMU-CS-03-131R, Carnegie Mellon University, December 2003.

[11] John DeTreville. Binder, a logic-based security language. In Martín Abadi and Steven Bellovin, editors, *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 105–113, Berkeley, California, May 2002. IEEE Computer Society Press.

[12] Deepak Garg, Lujo Bauer, Kevin D. Bowers, Frank Pfenning, and Michael K. Reiter. A linear logic of affirmation and knowledge. In Dieter Gollman, Jan Meier, and Andrei Sabelfeld, editors, *Proceedings of the 11th European Symposium on Research in Computer Security (ESORICS '06)*, pages 297–312, Hamburg, Germany, September 2006.

[13] Deepak Garg and Frank Pfenning. Non-interference in constructive authorization logic. In Joshua Guttman, editor, *Proceedings of the 19th Computer Security Foundations Workshop (CSFW '06)*, pages 283–293, Venice, Italy, July 2006. IEEE Computer Society Press.

[14] Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935. English translation in M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131, North-Holland, 1969.

[15] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

[16] Limin Jia. *Linear Logic and Imperative Programming.* PhD thesis, Department of Computer Science, Princeton University, November 2007.

[17] Butler Lampson, Martín Abadi, Michael Burrows, and Edward Wobber. Authentication in distributed systems: Theory and practice. *ACM Transactions on Computer Systems*, 10(4):265–310, November 1992.

[18] Chris Lesniewski-Laas, Bryan Ford, Jacob Strauss, Robert Morris, and M. Frans Kaashoek. Alpaca: Extensible authorization for distributed services. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS-2007)*, Alexandria, VA, October 2007.

[19] Per Martin-Löf. On the meanings of the logical constants and the justifications of the logical laws. *Nordic Journal of Philosophical Logic*, 1(1):11–60, 1996.

[20] Jason Reed. Hybridizing a logical framework. In *Proceedings of the International Workshop on Hybrid Logic (HyLo 2006)*, volume 174 of *Electronic Notes in Theoretical Computer Science*, pages 135–148, 2007.

[21] Uluç Saranlı and Frank Pfenning. Using constrained intuitionistic linear logic for hybrid robotic planning problems. In *Proceedings of International Conference on Robotics and Automation (ICRA '07)*, Rome, Italy, April 2007. IEEE Computer Society Press.