

Nexus Authorization Logic (NAL): Logical Results

Andrew K. Hirsch Michael R. Clarkson
Department of Computer Science
George Washington University
{akhirsch, clarkson}@gwu.edu

November 15, 2012

Abstract

Nexus Authorization Logic (NAL) [Schneider et al. 2011] is a logic for reasoning about authorization in distributed systems. A revised version of NAL is given here, including revised syntax, a revised proof theory using localized hypotheses, and a new Kripke semantics. The proof theory is proved sound with respect to the semantics, and that proof is formalized in Coq.

1 Introduction

Authorization logics are epistemic logics used to reason about whether principles are permitted to take actions in a distributed computer system. Nexus Authorization Logic (NAL), invented by Schneider et al. [8], is notable for enabling rich reasoning about axiomatic, synthetic, and analytic bases for authorization of actions. NAL extends a well-known authorization logic, cut-down dependency core calculus (CDD) [1]. Among other features, NAL upgrades CDD from having only propositional variables to having functions and predicates on system state.

The NAL rationale [8] gives a natural-deduction proof system for the logic and sketches the intuition for a semantics based on the idea of a *worldview*, which is the set of statements that a principle believes, or would be prepared to support. However, neither a formal semantics nor a proof of soundness is given in the rationale.

Here, we initiate the formal study of the metatheory of NAL by developing a formal semantics and a proof of soundness. Along the way, we streamline NAL in various ways, particularly in the syntax (by eliminating second-order quantification) and in the proof system (by localizing hypothetical judgments). We also fix a bug in the original proof system, which allowed derivation of a formula that arguably should be considered invalid.

Since our formalization of NAL differs from that of the NAL rationale, it will be convenient to have names to distinguish these two formal systems. Henceforth, we write “NAL₀”

to refer to the original formalization of the NAL rationale [8], and “NAL₁” to refer to the new formalization in this paper.

Our proof of soundness, including the syntax, proof system, and semantics of NAL₁, is formalized in the Coq proof assistant.¹ The formalization contains about 3,000 lines of code.

This short paper describes our formal syntax, proof system, and semantics for NAL. Familiarity with epistemic logics, constructive logics, and their Kripke semantics is assumed. Readers who seek background in these areas can consult standard references [3, 10].

2 Syntax

NAL₁ is a constructive, first order, multimodal logic. It has two syntactic classes, terms τ and formulas ϕ . Metavariable x ranges over first-order variables, f over first-order functions, and r over first-order relations. Logical formulas ϕ are described by the following grammar:

$\phi ::=$	true	
	false	
	$r(\tau, \dots, \tau)$	first-order relation
	$\tau_1 = \tau_2$	term equality
	$\phi_1 \wedge \phi_2$	conjunction
	$\phi_1 \vee \phi_2$	disjunction
	$\phi_1 \Rightarrow \phi_2$	implication
	$\neg\phi$	negation
	$(\forall x : \phi)$	first-order universal quantification
	$(\exists x : \phi)$	first-order existential quantification
	$\tau \text{ says } \phi$	affirmation
	$\tau_1 \Rightarrow \tau_2$	delegation
	$\tau_1 \Rightarrow \tau_2 \text{ on } (x : \phi)$	restricted delegation

Unlike NAL₀, formulas of NAL₁ do not permit monadic second-order universal quantification. In NAL₀, that quantifier was used only to define certain connectives, particularly delegation, as syntactic sugar. NAL₁ instead adds delegation as a primitive connective to the logic. This simplifies the logic from second-order down to first-order, at the small cost of adding a few extra axioms to the proof system to handle the delegation primitive.

Logical terms are described by the following grammar:

¹<http://coq.inria.fr>

$\tau ::= x$	first-order variable
$f(\tau, \dots, \tau)$	first-order function
$\tau_1.\tau_2$	subprincipal
$\{x : \phi\}$	group principal

There are some small, unimportant syntactic differences between NAL_0 and NAL_1 . The biggest of these is the notation for delegation: NAL_0 uses \rightarrow , whereas NAL_1 uses \Rightarrow to avoid any potential confusion with implication.

3 Proof System

The NAL_1 proof system is a natural-deduction proof system, like the NAL_0 proof system. But unlike the NAL_0 proof system, which uses hypothetical judgments for proving implication introduction, the NAL_1 proof system uses localized hypotheses.

In NAL_1 , the derivability judgment is written

$$\Gamma \vdash \phi$$

where Γ is a set of formulas. If $\Gamma \vdash \phi$, then ϕ is derivable from Γ according to the rules of the proof system. Rules for formulas are given in figure 1. Rules for terms are given in figure 2. In those figures, $\phi[\tau/x]$ denotes capture-avoiding substitution of τ for x in ϕ .

Most of the proof system is routine. The rules for says use notation $p \text{ says } \Gamma$, which intuitively means that p says all the formulas in set Γ . Formally, $p \text{ says } \Gamma$ is the set $\{p \text{ says } \phi \mid \phi \in \Gamma\}$. The says rules necessarily differ from the corresponding rules found in NAL_0 because of the use of localized hypotheses Γ . Nonetheless, the NAL_1 rules are essentially standard—for example, two of the three rules correspond to standard natural deduction rules for a necessity modality [6], and the third rule is symmetric to the second.

There is one important, deliberate change in the NAL_1 proof system that makes its theory differ from the NAL_0 system, which we now discuss. There are two standard ways of importing beliefs into a principal’s worldview. The first is a rule known as Necessitation: “if $\vdash p$ then $\vdash p \text{ says } \phi$.” The second is an axiom known as Unit: $\vdash p \Rightarrow (p \text{ says } \phi)$. Though superficially similar, Necessitation and Unit lead to different theories.

Example 1. *Machines M_1 and M_2 execute processes P_1 and P_2 , respectively. M_1 has a register R . Let Z be a proposition representing “register R is currently set to zero.” According to Unit, $\vdash Z \Rightarrow (P_1 \text{ says } Z)$ and $\vdash Z \Rightarrow (P_2 \text{ says } Z)$. The former means that a process on a machine knows the current contents of a register on that machine; the latter means that a process on a different machine must also know the current contents of the register. But according to Necessitation, if $\vdash Z$ then $\vdash P_1 \text{ says } Z$ and $\vdash P_2 \text{ says } Z$. Only if R is always zero must the two processes say so.*

Unit, therefore, is better used when propositions (or relations or functions) represent global state upon which all principals are guaranteed to agree. Necessitation is better used when propositions represent local state that could be unknown to some principals.

NAL was designed to reason about state in distributed systems, where principals (such as machines) may have local state, and where global state does not necessarily exist—the reading at a clock, for example, is not agreed upon by all principals in NAL. So Unit would be an overly strong restriction on NAL principals; Necessitation is the appropriate choice. Fortunately, NAL_0 does include Necessitation as an inference rule and does not include Unit as an axiom.

Unfortunately, NAL_0 [8] permits Unit to be derived as a theorem² because of an interaction between Necessitation and the NAL_0 introduction rule for implication. NAL_1 fixes this bug and does not permit derivation of Unit.

4 Semantics

The semantics of NAL_1 is combination of three standard semantic models: first-order models, constructive models, and modal models. This combination is probably not completely novel (see, e.g., [4, 11]), though we are not aware of any authorization logic semantics that is identical to or that subsumes our semantics. Our presentation mostly follows the Kripke semantics of intuitionistic predicate calculus given by Troelstra and van Dalen [10].

Below, we give a moderately pedagogic description of the definition of a semantic model for NAL, by building up progressively more complicated models.

First-order models. A *first-order model with equality* is a tuple $(D, =, R, F)$. The purpose of a first-order model is to interpret the first-order fragment of the logic, specifically first-order quantification, functions, and relations. D is a set, the *domain* of individuals. These individuals are what quantification in the logic ranges over. R is a set $\{r_i \mid i \in I\}$ of relations on D , indexed by set I , with associated arity function m , such that $r_i \subseteq D^{m(i)}$. Likewise, F is a set $\{f_j \mid j \in J\}$ of functions on D , indexed by set J , with associated arity function n , such that $f_j \in D^{n(j)} \rightarrow D$. There is a distinguished equality relation $=$, which is an equivalence relation on D , such that equality is indistinguishable by relations and functions:

- if $\vec{d} = \vec{d}'$ and $\vec{d} \in r_i$, where $|\vec{d}| = |\vec{d}'| = m(i)$, then $\vec{d}' \in r_i$, and
- if $\vec{d} = \vec{d}'$, where $|\vec{d}| = |\vec{d}'| = n(j)$, then $f_j(\vec{d}) = f_j(\vec{d}')$.

Constructive models. A *constructive model* is a tuple (W, \leq, s) . The purpose of a constructive model is to interpret the constructive fragment of the logic, specifically implication and universal quantification, (whose semantics differ from the classical semantics). W is a set, the *possible worlds*. We denote an individual world as w . Intuitively, a world w represents the *state of knowledge* of a constructive reasoner. Relation \leq , called the *constructive*

²From \mathcal{F} infer A says \mathcal{F} by SAYS-I. Then infer $\mathcal{F} \Rightarrow A$ says \mathcal{F} by IMP-I.

accessibility relation, is a partial order on W . If $w \leq w'$, then the constructive reasoner's state of knowledge could grow from w to w' . Function s is called the *interpretation function*. It assigns a first-order model $(D_w, =_w, R_w, F_w)$ to each world w . (Let the individual elements of R_w be notated as $\{r_{i,w} \mid i \in I\}$, and likewise for F_w , as $\{f_{j,w} \mid j \in J\}$.) Thus, s enables a potentially different first-order interpretation at each world. But to help ensure that the constructive reasoner's state of knowledge only grows—hence never invalidates a previously admitted construction—we require s to be monotonic w.r.t. \leq . That is, if $w \leq w'$ then

- $D_w \subseteq D_{w'}$,
- $d =_w d'$ implies $d =_{w'} d'$,
- $r_{i,w} \subseteq r_{i,w'}$, and
- for all \vec{d} such that $|\vec{d}| = n(j)$, it holds that $f_{j,w}(\vec{d}) =_w f_{j,w'}(\vec{d})$.

Constructive modal models. A *constructive modal model* is a tuple (W, \leq, s, P, A) . The purpose of a constructive modal model is to interpret the modal fragment of the logic, specifically the says connective and the delegation connectives. The first part of a constructive modal model, (W, \leq, s) , must itself be a constructive model as above. The next part, P , is a set of *principals*. Note that we treat principals differently than individuals: although individuals can vary from world to world in a model, the set of principals is assumed to be constant across the entire model. This assumption is consistent with other constructive multimodal logics [9, 11], which have a fixed set of modalities (just \Box and \Diamond). However, it would be interesting in future work to explore removing this assumption.

A is a set $\{A_p \mid p \in P\}$ of binary relations on W , called the *principal accessibility relations*. If $(w, w') \in A_p$, then in world w , principal p considers world w' possible. Like \leq in a constructive model, we require s to be monotonic w.r.t. each A_p . This requirement enforces a kind of constructivity on each principal p , such that if p is in a world in which individual d is constructed, then p cannot consider possible any world in which d has not been constructed.

Constructive modal models thus have two kinds of accessibility relations, constructive \leq and principal A_p . These relations cannot be completely orthogonal: for sake of soundness, we need to impose four *frame conditions* that relate constructive accessibility and principal accessibility.

- **F1.** If $w \leq w'$ and $(w, v) \in A_p$, then there exists a v' such that $v \leq v'$ and $(w', v') \in A_p$.
- **F2.** If $(w, v) \in A_p$ and $v \leq v'$, then there exists a w' such that $w \leq w'$ and $(w', v') \in A_p$.
- **IT.** If $(w, v) \in A_p$ and $(v, u) \in A_p$, then there exists a w' such that $w \leq w'$ and $(w', u) \in A_p$.

- **ID.** If $(w, u) \in A_p$, then there exists a w' and v such that $w \leq w'$, and $(w', v) \in A_p$ as well as $(v, u) \in A_p$.

The need for these frame conditions originates from the proof system rules for says, especially the latter two rules. It's well known in modal logic that axioms and rules about modalities correspond to frame conditions on accessibility relations (see, e.g., chapter 3 of [3]). IT and ID are intuitionistic generalizations of transitivity and density of the A_p relations. In the presence of F1 and F2, IT and ID are necessary and sufficient conditions for the soundness of the says rules—a result that follows from work by Plotkin and Stirling [7]. Furthermore, F1 and F2 are arguably the right fundamental frame conditions to impose in a constructive modal logic [9]. In the case of NAL, we could actually remove F1 without suffering any unsoundness or incompleteness. (F1 is needed only to show soundness of a \diamond modality, which does not exist in NAL.) However, the others—F2, IT, and ID—are all necessary to impose in NAL_1 .

NAL models. A *NAL model* is a tuple $(W, \leq, s, P, A, \vee, \perp, SUB)$. The purpose of a NAL model is to interpret NAL formulas. Specifically, it adds machinery to interpret group and subprincipals. The first part of a NAL model, (W, \leq, s, P, A) , must itself be a constructive modal model as above.

The next part of a NAL model, (\vee, \perp) , is used to interpret group principals. Specifically, (P, \vee) must be a join semilattice, with \perp as its bottom element. (Thus, \perp is a principal. Its intended use is as a principal who believes only tautologies. We do not require the existence of a top element in the lattice, because there is no need for such an element in the semantics.) Join operator \vee is used to take disjunctions of principals—intuitively, $p \vee q$ is the principal who believes those statements that either p or q believe, or statements that logically follow from those. Formally, we require that, for all principals p and q , it holds that $A_{p \vee q} \subseteq A_p$.

The *SUB* part of a NAL model is used to interpret subprincipals. Intuitively, it requires the existence of a distinguished first-order function sub_w of type $P \times D_w \rightarrow P$ at each world w . Further, we require that if $sub_w(p, d) = q$, then $A_p \supseteq A_q$, ensuring that superprincipals speak for subprincipals. Since sub_w is a function, it must obey the requirement of monotonicity w.r.t. constructive accessibility relation \leq , just as all other functions $f_{j,w}$ must in a constructive model.

NAL models for Coq. Finally, a *NAL model for Coq* is a NAL model extended with a pair of sets Δ and Π . This is a technical extension that unfortunately seems to be necessary in order to express something that is, in actuality, fairly simple set theory. We'd like to require that set P of principals be a subset of every domain D_w in a NAL model, such that there is one unchanging set of principals throughout the model. Expressing that idea in Coq's type theory turns out to be quite difficult, so we instead stipulate the existence of two sets of coercion functions, Δ and Π , between principals and individuals. Δ is a set $\{\delta_w : P \rightarrow D_w \mid w \in W\}$ of coercion functions that map principals to individuals. Since

every principal should be represented by a unique domain element, we require each δ_w to be injective. Π is a set $\{\pi_w : D_w \rightarrow P \mid w \in W\}$ of coercion functions that map individuals to principals. If individual d does not represent a principal, then $\delta_w(d)$ is \perp .

Given these coercion functions, it is possible to define equality $=_P$ of principals in terms of equality of individuals: $p =_P q$ iff for all w , $\delta_w(p) =_w \delta_w(q)$.

NAL semantics. We give a semantics of NAL_1 in figure 3. The validity judgment is written

$$M, w, v \models \phi$$

where M is a NAL model for Coq and w is a world in that model. Function v is a *valuation* mapping first-order variables to individuals; it is used to interpret first-order quantification. The semantics also relies on an *interpretation* function μ , defined in figure 4, that maps syntactic terms τ to individuals.

The first-order constructive fragment of the semantics is routine. The semantics of says follows from the semantics of a \Box modality in constructive modal logic [9, 11]. The semantics of delegation \Rightarrow follows from a standard definition in authorization logics [2]. The semantics of restricted delegation is similar to one presented by Howell [5], and it is a generalization of the semantics of unrestricted delegation. (To see this, take w''' in the semantics of restricted delegation to be the w'' from the semantics of unrestricted delegation. Then w''' equals w'' , hence is in the same equivalence class.) Restricted delegation uses an equivalence relation $\equiv_{x:\phi}^w$ on worlds. Intuitively, this relation is used to partition worlds into equivalence classes that agree on the validity of formula ϕ in all valuations, assuming the existence of individuals D_w . Formally, define $w' \equiv_{x:\phi}^w w''$ to hold iff

$$\begin{aligned} \forall d \in D_w : \quad & (\forall v : M, w', v[d/x] \models \phi) \\ \iff & (\forall v : M, w'', v[d/x] \models \phi). \end{aligned}$$

The interpretation function is also routine, except for the interpretation of group principals. That interpretation is similar to the algebra of principals defined in the ABLP logic [2].

5 Soundness

The soundness theorem for NAL_1 states that if ϕ is provable from assumptions Γ , and that if a model validates all the formulas in Γ , then that model must also validate ϕ . Therefore, any provable formula is semantically valid.

Theorem 1 (Soundness). *If $\Gamma \vdash \phi$ and for all $\psi \in \Gamma$, it holds that $M, w, v \models \psi$, then $M, w, v \models \phi$.*

A Coq mechanization of the proof of Soundness is in progress. Currently, it contains about 3,000 lines of code and implements all of the proof except for the cases of delegation and restricted delegation.

The current proof also requires adding an additional assumption as an axiom: for all w and w' , if $w \leq w'$, or if there a exists p such that $(w, w') \in A_p$, then it must hold

that $\mu(M, w, v)(\tau) = \mu(M, w', v)(\tau)$. This axiom is actually provable as a theorem for all terms τ except for group principals. Discharging this assumption for group principals remains an open problem.

Acknowledgments

Fred B. Schneider consulted on the design of the proof system and the Kripke semantics. We thank him, Martín Abadi, Deepak Garg, and Colin Stirling for discussions related to this work. This work was supported in part by AFOSR grants F9550-06-0019, FA9550-11-1-0137, and FA9550-12-1-0334, National Science Foundation grants 0430161, 0964409, and CCF-0424422 (TRUST), ONR grants N00014-01-1-0968 and N00014-09-1-0652, and a grant from Microsoft.

References

- [1] Martín Abadi. Access control in a core calculus of dependency. *Electronic Notes in Theoretical Computer Science*, 172:5–31, April 2007.
- [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] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. MIT Press, Cambridge, Massachusetts, 1995.
- [4] Valerio Genovese, Deepak Garg, and Daniele Rispoli. Labeled sequent calculi for access control logics: Countermodels, saturation and abduction. In *Proc. IEEE Computer Security Foundations Symposium (CSF)*, pages 139–153, 2012.
- [5] Jonathan Howell. *Naming and Sharing Resources across Administrative Domains*. PhD thesis, Dartmouth College, 2000.
- [6] George Edward Hughes and Max J. Cresswell. *A New Introduction to Modal Logic*. Routledge, London, 1996.
- [7] Gordon Plotkin and Colin Stirling. A framework for intuitionistic modal logics. In *Proc. Conference on Theoretical Aspects of Reasoning about Knowledge (TARK)*, pages 399–406, 1986.
- [8] Fred B. Schneider, Kevin Walsh, and Emin Gün Sirer. Nexus authorization logic (NAL): Design rationale and applications. *ACM Transactions on Information and System Security*, 14(1):8:1–28, June 2011.
- [9] Alex K. Simpson. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. PhD thesis, University of Edinburgh, 1994.

- [10] Anne Sjerp Troelstra and Dirk van Dalen. *Constructivism in Mathematics: Volume I*, volume 121 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, Amsterdam, 1988.
- [11] Duminda Wijesekera. Constructive modal logics I. *Annals of Pure and Applied Logic*, 50(3):271–301, December 1990.

$$\begin{array}{c}
\frac{}{\Gamma, \phi \vdash \phi} \\
\frac{}{\Gamma \vdash \text{true}} \\
\frac{\Gamma \vdash \phi \quad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi} \\
\frac{\Gamma \vdash \phi_1 \quad \Gamma \vdash \phi_2}{\Gamma \vdash \phi_1 \vee \phi_2} \\
\frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \Rightarrow \psi} \\
\frac{\Gamma, \phi \vdash \text{false}}{\Gamma \vdash \neg \phi} \\
\frac{\Gamma \vdash \phi \quad x \notin FV(\Gamma)}{\Gamma \vdash (\forall x : \phi)} \\
\frac{\Gamma \vdash \phi[\tau/x]}{\Gamma \vdash (\exists x : \phi)} \\
\frac{\Gamma \vdash \phi}{p \text{ says } \Gamma \vdash p \text{ says } \phi}
\end{array}
\qquad
\begin{array}{c}
\frac{\Gamma \vdash \phi}{\Gamma, \psi \vdash \phi} \\
\frac{\Gamma \vdash \text{false}}{\Gamma \vdash \phi} \\
\frac{\Gamma \vdash \phi \wedge \psi \quad \Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \phi \quad \Gamma \vdash \psi} \\
\frac{\Gamma \vdash \phi_1 \vee \phi_2 \quad \Gamma, \phi_1 \vdash \psi \quad \Gamma, \phi_2 \vdash \psi}{\Gamma \vdash \psi} \\
\frac{\Gamma \vdash \phi \quad \Gamma \vdash \phi \Rightarrow \psi}{\Gamma \vdash \psi} \\
\frac{\Gamma \vdash \phi \quad \Gamma \vdash \neg \phi}{\Gamma \vdash \text{false}} \\
\frac{\Gamma \vdash (\forall x : \phi)}{\Gamma \vdash \phi[\tau/x]} \\
\frac{\Gamma \vdash (\exists x : \phi) \quad \Gamma, \phi \vdash \psi \quad x \notin FV(\Gamma, \psi)}{\Gamma \vdash \psi} \\
\frac{p \text{ says } \Gamma \vdash \phi \quad \Gamma \vdash p \text{ says } \phi}{p \text{ says } \Gamma \vdash p \text{ says } \phi}
\end{array}$$

Figure 1: Derivability judgment for formulas

$$\begin{array}{c}
\frac{}{\Gamma \vdash \tau = \tau} \quad \frac{\Gamma \vdash \tau_1 = \tau_2}{\Gamma \vdash \tau_2 = \tau_1} \quad \frac{\Gamma \vdash \tau_1 = \tau_2 \quad \Gamma \vdash \tau_2 = \tau_3}{\Gamma \vdash \tau_1 = \tau_3} \\
\\
\frac{\Gamma \vdash \tau_1 = \tau'_1 \quad \dots \quad \Gamma \vdash \tau_n = \tau'_n}{\Gamma \vdash f(\tau_1, \dots, \tau_n) = f(\tau'_1, \dots, \tau'_n)} \\
\\
\frac{\Gamma \vdash r(\tau_1, \dots, \tau_n) \quad \Gamma \vdash \tau_1 = \tau'_1 \quad \dots \quad \Gamma \vdash \tau_n = \tau'_n}{\Gamma \vdash r(\tau'_1, \dots, \tau'_n)} \\
\\
\frac{\Gamma \vdash \tau_2 \text{ says } \tau_1 \Rightarrow \tau_2}{\Gamma \vdash \tau_1 \Rightarrow \tau_2} \quad \frac{\Gamma \vdash \tau_2 \text{ says } \tau_1 \Rightarrow \tau_2 \text{ on } (x : \phi)}{\Gamma \vdash \tau_1 \Rightarrow \tau_2 \text{ on } (x : \phi)} \\
\\
\frac{\Gamma \vdash \tau_1 \Rightarrow \tau_2 \quad \Gamma \vdash \tau_1 \text{ says } \phi}{\Gamma \vdash \tau_2 \text{ says } \phi} \quad \frac{\Gamma \vdash \tau_1 \Rightarrow \tau_2 \text{ on } (x : \phi) \quad \Gamma \vdash \tau_1 \text{ says } \phi[\tau/x]}{\Gamma \vdash \tau_2 \text{ says } \phi[\tau/x]} \\
\\
\frac{}{\Gamma \vdash \tau \Rightarrow \tau} \quad \frac{}{\Gamma \vdash \tau \Rightarrow \tau \text{ on } (x : \phi)} \\
\\
\frac{\Gamma \vdash \tau_1 \Rightarrow \tau_2 \quad \Gamma \vdash \tau_2 \Rightarrow \tau_3}{\Gamma \vdash \tau_1 \Rightarrow \tau_3} \quad \frac{\Gamma \vdash \tau_1 \Rightarrow \tau_2 \text{ on } (x : \phi) \quad \Gamma \vdash \tau_2 \Rightarrow \tau_3 \text{ on } (x : \phi)}{\Gamma \vdash \tau_1 \Rightarrow \tau_3 \text{ on } (x : \phi)} \\
\\
\frac{\Gamma \vdash \phi[\tau/x]}{\Gamma \vdash \tau \Rightarrow \{x : \phi\}} \quad \frac{\Gamma, \phi \vdash x \Rightarrow \tau \quad x \notin FV(\tau)}{\Gamma \vdash \{x : \phi\} \Rightarrow \tau} \\
\\
\frac{}{\Gamma \vdash \tau_1 \Rightarrow \tau_1 \cdot \tau_2}
\end{array}$$

Figure 2: Derivability judgment for terms

$M, w, v \models \text{true}$	always
$M, w, v \models \text{false}$	never
$M, w, v \models r_i(\vec{\tau})$	iff $\mu(M, w, v)(\vec{\tau}) \in r_{i,w}$
$M, w, v \models \tau = \tau'$	iff $\mu(M, w, v)(\tau) =_w \mu(M, w, v)(\tau')$
$M, w, v \models \phi_1 \wedge \phi_2$	iff $M, w, v \models \phi_1$ and $M, w, v \models \phi_2$
$M, w, v \models \phi_1 \vee \phi_2$	iff $M, w, v \models \phi_1$ or $M, w, v \models \phi_2$
$M, w, v \models \phi_1 \Rightarrow \phi_2$	iff for all $w' \geq w : M, w', v \models \phi_1$ implies $M, w', v \models \phi_2$
$M, w, v \models \neg\phi$	iff for all $w' \geq w : M, w', v \not\models \phi$
$M, w, v \models (\forall x : \phi)$	iff for all $w' \geq w, d \in D_{w'} : M, w', v[d/x] \models \phi$
$M, w, v \models (\exists x : \phi)$	iff there exists $d \in D_w : M, w, v[d/x] \models \phi$
$M, w, v \models \tau \text{ says } \phi$	iff for all $w', w'' : w \leq w'$ and $(w', w'') \in A_{\mu(M, w, v)(\tau)}$ implies $M, w'', v \models \phi$
$M, w, v \models \tau_1 \Rightarrow \tau_2$	iff for all $w', w'' : (w', w'') \in A_{\mu(M, w, v)(\tau_2)}$ implies $(w', w'') \in A_{\mu(M, w, v)(\tau_1)}$
$M, w, v \models \tau_1 \Rightarrow \tau_2 \text{ on } (x : \phi)$	iff for all $w', w'' : (w', w'') \in A_{\mu(M, w, v)(\tau_2)}$ there exists $w''' : w'' \equiv_{x:\phi}^{w'} w'''$ and $(w', w''') \in A_{\mu(M, w, v)(\tau_1)}$

Figure 3: Validity judgment

$$\begin{aligned}
\mu(M, w, v)(x) &= v(x) \\
\mu(M, w, v)(f_j(\vec{\tau})) &= f_{j,w}(\mu(M, w, v)(\vec{\tau})) \\
\mu(M, w, v)(\tau_1.\tau_2) &= \text{sub}_w(\mu(M, w, v)(\tau_1), \mu(M, w, v)(\tau_2)) \\
\mu(M, w, v)(\{x : \phi\}) &= (\bigvee p : M, w, v[p/x] \models \phi : p)
\end{aligned}$$

Figure 4: Interpretation function