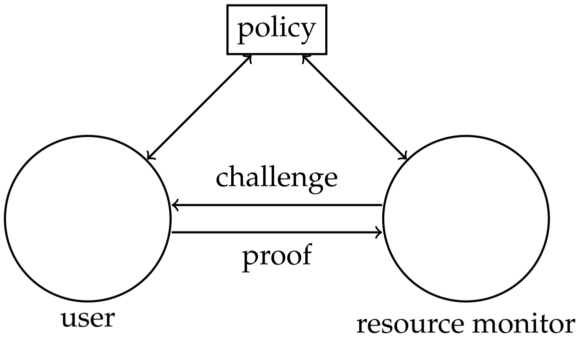
Lecture Notes on
Proof Representation

15-316: Software Foundations of Security & Privacy
Frank Pfenning

Lecture 17
November 5, 2024

1 Introduction

The picture below illustrates the general proof-carrying architecture we have considered in the last two lectures.



In this particular use of authorization logic, the user bears the burden of proof. If the policy is not particularly complex, the resource monitor itself could construct a proof instead, either explicitly or implicitly via an implementation that has been proved correct against the policy.

One advantage of this architecture is that new affirmations can enter the picture dynamically. For example, when *myra* stands in front of my office she might contact me to obtain an affirmation from me that she is my student and can therefore enter it. Such an affirmation would come in the form of a *signed certificate*, which we will discuss in the next lecture.

If we stick to the architecture as depicted, it is the user's responsibility to produce a proof of the challenge formula and the resource monitor's responsibility to check the proof it received. In the last lecture we talked about some strategies for finding proofs; today we'll talk about how to represent them so they can be communicated to the resource monitor and then checked.

The mainstay of the whole idea is that the policy expresses the intended authorization policy in a straightforward and understandable way.

2 Proof Terms for Intuitionistic Propositional Logic

We start with the proof terms for propositional logic. The basic idea is to annotate a sequent

$$P_1, \ldots, P_n \vdash Q$$

with *proof terms* M_i and N such that

$$M_1 : P_1, \ldots, M_n : P_n \vdash N : Q$$

The initial sequent we try to prove has the form

$$c_1 : P_1, \ldots, c_n : P_n \vdash \,?\, : Q$$

where c_i are *signed certificates* that serve as justifications for the antecedents. As we proceed with the proof, we may create additional antecedents $M : P$ and eventually justify the conclusion with a proof term $N : Q$. The term N should have enough information to check that the initial sequent has a proof.

For the remainder of this lecture, we will use Γ to also stand for a sequent where each antecedent has a suitable justification.

Conjunction. We start with conjunction:

$$\frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q}\ \wedge R \qquad\qquad \frac{\Gamma, P, Q \vdash \delta}{\Gamma, P \wedge Q \vdash \delta}\ \wedge L$$

We see that the justification for $P \wedge Q$ should be a pair, consisting of a justification for P and one for Q.

$$\frac{\Gamma \vdash M : P \quad \Gamma \vdash N : Q}{\Gamma \vdash \langle M, N \rangle : P \wedge Q}\ \wedge R$$

To check that $\langle M, N \rangle$ is a proof of $P \wedge Q$ we just need to check that M is a proof of P and N is a proof of Q.

How does the left rule work? Assume that M is a justification for $P \wedge Q$. That means that M represents a pair, and its first component should be a proof of P and its second component a proof of Q.

$$\frac{\Gamma, M.\pi_1 : P, M.\pi_2 : Q \vdash N : \delta}{\Gamma, M : P \wedge Q \vdash N : \delta}\ \wedge L$$

Identity. Let's complete this first analysis with the identity rule. Because all antecedents have a justification, we just use that as our justification of the (identical) succedent.

$$\dfrac{}{\Gamma, P \vdash P}\,\mathsf{id} \qquad\qquad \dfrac{}{\Gamma, M : P \vdash M : P}\,\mathsf{id}$$

At this point we can already write out a small example.

$$\dfrac{\dfrac{\dfrac{}{P, Q \vdash Q}\,\mathsf{id} \quad \dfrac{}{P, Q \vdash P}\,\mathsf{id}}{P, Q \vdash Q \wedge P}\,\wedge R}{P \wedge Q \vdash Q \wedge P}\,\wedge L$$

We now annotate this in several steps, leaving question marks where information is still to be filled in

$$\dfrac{\dfrac{\dfrac{}{? : P, ? : Q \vdash ? : Q}\,\mathsf{id} \quad \dfrac{}{? : P, ? : Q \vdash ? : P}\,\mathsf{id}}{? : P, ? : Q \vdash ? : Q \wedge P}\,\wedge R}{x : P \wedge Q \vdash ? : Q \wedge P}\,\wedge L$$

Here, x is the initial justification for the antecedent $P \wedge Q$ which could be a signed certificate, or perhaps a variable that can come from somewhere else. From it, we can construct the justifications for P and Q by projection. These then also flow upward in the derivation.

$$\dfrac{\dfrac{\dfrac{}{x.\pi_1 : P, x.\pi_2 : Q \vdash ? : Q}\,\mathsf{id} \quad \dfrac{}{x.\pi_1 : P, x.\pi_2 : Q \vdash ? : P}\,\mathsf{id}}{x.\pi_1 : P, x.\pi_2 : Q \vdash ? : Q \wedge P}\,\wedge R}{x : P \wedge Q \vdash ? : Q \wedge P}\,\wedge L$$

Now we can copy over the justifications for P and Q in the two applications of identity.

$$\dfrac{\dfrac{\dfrac{}{x.\pi_1 : P, x.\pi_2 : Q \vdash x.\pi_2 : Q}\,\mathsf{id} \quad \dfrac{}{x.\pi_1 : P, x.\pi_2 : Q \vdash x.\pi_1 : P}\,\mathsf{id}}{x.\pi_1 : P, x.\pi_2 : Q \vdash ? : Q \wedge P}\,\wedge R}{x : P \wedge Q \vdash ? : Q \wedge P}\,\wedge L$$

Now we can fill in the proof term for $Q \wedge P$ and propagate it down the derivation.

$$\dfrac{\dfrac{\dfrac{}{x.\pi_1 : P, x.\pi_2 : Q \vdash x.\pi_2 : Q}\,\mathsf{id} \quad \dfrac{}{x.\pi_1 : P, x.\pi_2 : Q \vdash x.\pi_1 : P}\,\mathsf{id}}{x.\pi_1 : P, x.\pi_2 : Q \vdash \langle x.\pi_2, x.\pi_1 \rangle : Q \wedge P}\,\wedge R}{x : P \wedge Q \vdash \langle x.\pi_2, x.\pi_1 \rangle : Q \wedge P}\,\wedge L$$

Already here we might anticipate something that holds on a larger scale. Namely, the proof looks like a piece of code that reverses the elements of a pair. So the term representing a proof is like a program, and the proposition is like its type. We can only scratch the surface of that connection, common called the *Curry-Howard Isomorphism* [Curry, 1934, Howard, 1969]. The CMU course on Constructive Logic investigates this connection in depth.

Implication. The intuitionistic reading of $P \to Q$ is as a function from proofs of P to proofs of Q.

$$\frac{\Gamma, P \vdash Q}{\Gamma \vdash P \to Q}\ {\to}R \qquad\qquad \frac{\Gamma \vdash P \quad \Gamma, Q \vdash \delta}{\Gamma, P \to Q \vdash \delta}\ {\to}L$$

Then the proof term for a right rule is a function.

$$\frac{\Gamma, x : P \vdash N : Q}{\Gamma \vdash (\lambda x.\, N) : P \to Q}\ {\to}R$$

This notation goes back to Church's λ-calculus [Church and Rosser, 1936] and may be written in concrete syntax as `fn x => N` (Standard ML) or `fun x -> N` (OCaml) or `\x -> N` (Haskell).

The left rule represents function application, written as juxtaposition $N\,M$.

$$\frac{\Gamma \vdash M : P \quad \Gamma, N\,M : Q \vdash O : \delta}{\Gamma, N : P \to Q \vdash O : \delta}\ {\to}L$$

Just like for $\wedge L$, the succedent does not change in this rule.

To resume our example: we can finish with an $\to R$ rule.

$$\cfrac{\cfrac{\cfrac{\cfrac{}{x.\pi_1 : P, x.\pi_2 : Q \vdash x.\pi_2 : Q}\ \mathsf{id} \quad \cfrac{}{x.\pi_1 : P, x.\pi_2 : Q \vdash x.\pi_1 : P}\ \mathsf{id}}{x.\pi_1 : P, x.\pi_2 : Q \vdash \langle x.\pi_2, x.\pi_1 \rangle : Q \wedge P}\ {\wedge}R}{x : P \wedge Q \vdash \langle x.\pi_2, x.\pi_1 \rangle : Q \wedge P}\ {\wedge}L}{\vdash \lambda x.\, \langle x.\pi_2, x.\pi_1 \rangle : P \wedge Q \to Q \wedge P}\ {\to}R$$

Recall that the left rule for implication is *not invertible*. This means we actually may need the antecedent $P \to Q$ (or, annotated, $N : P \to Q$) again in the proof. We don't explicitly reflect this in $\to L$, or in left rules in general. Instead, by convention, the antecedent we apply a left rule to may be kept or discarded in the premises. If one wants to build a theorem prover that is complete, a detailed analysis of whether antecedents may be discarded would be indicated.

Universal Quantification. Intuitionistically, a proof of $\forall x.\, P(x)$ also represents a function. It takes as argument an element c from the domain of quantification and returns a proof of $P(c)$. Inside a proof, this element c could also be a variable denoting a constant.

$$\frac{\Gamma \vdash P(y) \quad y \not\in \Gamma, P(x)}{\Gamma \vdash \forall x.\, P(x)}\ \forall R^y \qquad \frac{\Gamma, P(c) \vdash \delta}{\Gamma, \forall x.\, P(x) \vdash \delta}\ \forall L$$

Since it also is a function, we reuse the same notation as for implication. The context of use will provide enough information to disambiguate.

$$\frac{\Gamma \vdash M(y) : P(y) \quad y \not\in \Gamma, P(x)}{\Gamma \vdash (\lambda x.\, M(x)) : \forall x.\, P(x)}\ \forall R^y \qquad \frac{\Gamma, M\, c : P(c) \vdash N : \delta}{\Gamma, M : \forall x.\, P(x) \vdash N : \delta}\ \forall L$$

Cut. The cut rule introduces a lemma into a proof. With terms, this means we introduce a name for a possibly complex term.

$$\frac{\Gamma \vdash P \quad \Gamma, P \vdash \delta}{\Gamma \vdash \delta}\ \mathsf{cut} \qquad \frac{\Gamma \vdash M : P \quad \Gamma, x : P \vdash N : Q}{\Gamma \vdash \mathbf{let}\ x = M\ \mathbf{in}\ N : Q}\ \mathsf{cut}$$

There is a potential issue with checking applications of cut since the conclusion (and the term **let** $x = M$ **in** N) does not contain P. So we might need to add this to the term. Note that there potentially is another rule for a succedent A **aff** Q.

Disjunction. We omit the proof terms for disjunction since our application ultimately does not use disjunction. In brief, the right rule tags members of a sum while the left rule represents a program that distinguishes cases.

3 Proof Terms for Affirmations

The complication for the rules of affirmation is that the left rules for principle A are unlocked for a limited section of the proof. This section needs to be represented explicitly and we use $\{M\}_A$ to represent this scope.

$$\frac{\Gamma \vdash A\ \mathbf{aff}\ P}{\Gamma \vdash (A\ \mathbf{says}\ P)\ \mathsf{true}}\ \mathbf{says}R \qquad \frac{\Gamma \vdash M : A\ \mathbf{aff}\ P}{\Gamma \vdash \{M\}_A : (A\ \mathbf{says}\ P)\ \mathsf{true}}\ \mathbf{says}R$$

The left rule "strips off" the scoping that may be present in the term M and binds a fresh variable x within the current scope.

$$\frac{\Gamma, P \vdash A\ \mathbf{aff}\ Q}{\Gamma, A\ \mathbf{says}\ P \vdash A\ \mathbf{aff}\ Q}\ \mathbf{says}L \qquad \frac{\Gamma, x : P \vdash N : A\ \mathbf{aff}\ Q}{\Gamma, M : A\ \mathbf{says}\ P \vdash \mathbf{let}\ \{x\}_A = M\ \mathbf{in}\ N : A\ \mathbf{aff}\ Q}\ \mathbf{says}L$$

Finally, the rule of affirmation is a judgmental transition (rather than being connected to a particular proposition), so the proof term remains the same.

$$\dfrac{\Gamma \vdash M : P\ \mathsf{true}}{\Gamma \vdash M : A\ \textbf{aff}\ P}\ \textbf{aff}$$

Before we go to our motivating example, we work out a relatively simple one.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P \vdash P}\ \mathsf{id} \quad \dfrac{}{Q \vdash Q}\ \mathsf{id}}{P \to Q, P \vdash Q}\ {\to}L}{P \to Q, P \vdash A\ \textbf{aff}\ Q}\ \textbf{aff}}{A\ \textbf{says}\ (P \to Q), A\ \textbf{says}\ P \vdash A\ \textbf{aff}\ Q}\ \textbf{says}L \times 2}{A\ \textbf{says}\ (P \to Q), A\ \textbf{says}\ P \vdash A\ \textbf{says}\ Q}\ \textbf{says}R}{\vdash A\ \textbf{says}\ (P \to Q) \to (A\ \textbf{says}\ P \to A\ \textbf{says}\ Q)}\ {\to}R \times 2$$

We start by annotating bottom-up, leaving "unknowns" as question marks.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P \vdash ? : P}\ \mathsf{id} \quad \dfrac{}{Q \vdash ? : Q}\ \mathsf{id}}{P \to Q, P \vdash ? : Q}\ {\to}L}{P \to Q, P \vdash ? : A\ \textbf{aff}\ Q}\ \textbf{aff}}{x : A\ \textbf{says}\ (P \to Q), y : A\ \textbf{says}\ P \vdash ? : A\ \textbf{aff}\ Q}\ \textbf{says}L \times 2}{x : A\ \textbf{says}\ (P \to Q), y : A\ \textbf{says}\ P \vdash ? : A\ \textbf{says}\ Q}\ \textbf{says}R}{\vdash ? : A\ \textbf{says}\ (P \to Q) \to (A\ \textbf{says}\ P \to A\ \textbf{says}\ Q)}\ {\to}R \times 2$$

Now we apply the rule of affirmation, which introduces two new antecedents derived from x and y, which we call x' and y'.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x' : P \vdash ? : P}\ \mathsf{id} \quad \dfrac{}{y' : Q \vdash ? : Q}\ \mathsf{id}}{x' : P \to Q, y' : P \vdash ? : Q}\ {\to}L}{x' : P \to Q, y' : P \vdash ? : A\ \textbf{aff}\ Q}\ \textbf{aff}}{x : A\ \textbf{says}\ (P \to Q), y : A\ \textbf{says}\ P \vdash ? : A\ \textbf{aff}\ Q}\ \textbf{says}L \times 2}{x : A\ \textbf{says}\ (P \to Q), y : A\ \textbf{says}\ P \vdash ? : A\ \textbf{says}\ Q}\ \textbf{says}R}{\vdash ? : A\ \textbf{says}\ (P \to Q) \to (A\ \textbf{says}\ P \to A\ \textbf{says}\ Q)}\ {\to}R \times 2$$

Now we can copy over x' and y' in applications of the identity and then work our way down, annotating the succedent.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x' : P \vdash x' : P}\,\textsf{id} \qquad \dfrac{}{y' : Q \vdash y' : Q}\,\textsf{id}}{x' : P \to Q, y' : P \vdash x'\,y' : Q}\,{\to}L}{x' : P \to Q, y' : P \vdash x'\,y' : A\ \textbf{aff}\ Q}\,\textbf{aff}}{x : A\ \textbf{says}\ (P \to Q), y : A\ \textbf{says}\ P \vdash ? : A\ \textbf{aff}\ Q}\,\textbf{says}L \times 2}{x : A\ \textbf{says}\ (P \to Q), y : A\ \textbf{says}\ P \vdash ? : A\ \textbf{says}\ Q}\,\textbf{says}R}{\vdash ? : A\ \textbf{says}\ (P \to Q) \to (A\ \textbf{says}\ P \to A\ \textbf{says}\ Q)}\,{\to}R \times 2
$$

The $\textbf{says}L$ rules wrap lets around the proof term for the affirmation judgment.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x' : P \vdash x' : P}\,\textsf{id} \qquad \dfrac{}{y' : Q \vdash y' : Q}\,\textsf{id}}{x' : P \to Q, y' : P \vdash x'\,y' : Q}\,{\to}L}{x' : P \to Q, y' : P \vdash x'\,y' : A\ \textbf{aff}\ Q}\,\textbf{aff}}{x : A\ \textbf{says}\ (P \to Q), y : A\ \textbf{says}\ P \vdash \textbf{let}\ \{x'\}_A = x\ \textbf{in}\ \textbf{let}\ \{y'\}_A = y\ \textbf{in}\ x'\,y' : A\ \textbf{aff}\ Q}\,\textbf{says}L \times 2}{x : A\ \textbf{says}\ (P \to Q), y : A\ \textbf{says}\ P \vdash ? : A\ \textbf{says}\ Q}\,\textbf{says}R}{\vdash ? : A\ \textbf{says}\ (P \to Q) \to (A\ \textbf{says}\ P \to A\ \textbf{says}\ Q)}\,{\to}R \times 2
$$

It remains to wrap the proof to indicate A's perspective ($\textbf{says}R$) and then introduce two λ-abstractions.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x' : P \vdash x' : P}\,\textsf{id} \qquad \dfrac{}{y' : Q \vdash y' : Q}\,\textsf{id}}{x' : P \to Q, y' : P \vdash x'\,y' : Q}\,{\to}L}{x' : P \to Q, y' : P \vdash x'\,y' : A\ \textbf{aff}\ Q}\,\textbf{aff}}{x : A\ \textbf{says}\ (P \to Q), y : A\ \textbf{says}\ P \vdash \textbf{let}\ \{x'\}_A = x\ \textbf{in}\ \textbf{let}\ \{y'\}_A = y\ \textbf{in}\ x'\,y' : A\ \textbf{aff}\ Q}\,\textbf{says}L \times 2}{x : A\ \textbf{says}\ (P \to Q), y : A\ \textbf{says}\ P \vdash \{\textbf{let}\ \{x'\}_A = x\ \textbf{in}\ \textbf{let}\ \{y'\}_A = y\ \textbf{in}\ x'\,y'\}_A : A\ \textbf{says}\ Q}\,\textbf{says}R}{\vdash \lambda x.\,\lambda y.\,\{\textbf{let}\ \{x'\}_A = x\ \textbf{in}\ \textbf{let}\ \{y'\}_A = y\ \textbf{in}\ x'\,y'\}_A : A\ \textbf{says}\ (P \to Q) \to (A\ \textbf{says}\ P \to A\ \textbf{says}\ Q)}\,{\to}R \times 2
$$

4 Example Revisited

Recall the motivating example, where we have labeled the antecedents with c_i. We imagine that in an implementation, they would be signed certificates.

$$\begin{array}{l}
c_1 : \mathit{admin}\ \textbf{says}\ (\forall A.\, \forall R.\, \mathsf{owns}(A, R) \to \mathsf{mayOpen}(A, R)),\\
c_2 : \mathit{admin}\ \textbf{says}\ (\forall A.\, \forall B.\, \forall R.\, \mathsf{owns}(A, R) \land \mathit{fp}\ \textbf{says}\ \mathsf{studentOf}(B, A) \to \mathsf{mayOpen}(B, R)),\\
c_3 : \mathit{admin}\ \textbf{says}\ \mathsf{owns}(\mathit{fp}, \mathit{ghc6017}),\\
c_4 : \mathit{fp}\ \textbf{says}\ \mathsf{studentOf}(\mathit{hemant}, \mathit{fp})\\
\vdash\\
?Q_0 : \mathit{admin}\ \textbf{says}\ \mathsf{mayOpen}(\mathit{hemant}, \mathit{ghc6017})
\end{array}$$

We have also named the resulting (as yet to be determined) proof term so we can reference it. First, we apply $\textbf{says}R$ and then unlock c_1, c_2, and c_3. We arrive at the sequent

$$\begin{array}{l}
x_1 : \forall A.\, \forall R.\, \mathsf{owns}(A, R) \to \mathsf{mayOpen}(A, R),\\
x_2 : \forall A.\, \forall B.\, \forall R.\, \mathsf{owns}(A, R) \land \mathit{fp}\ \textbf{says}\ \mathsf{studentOf}(B, A) \to \mathsf{mayOpen}(B, R),\\
x_3 : \mathsf{owns}(\mathit{fp}, \mathit{ghc6017}),\\
c_4 : \mathit{fp}\ \textbf{says}\ \mathsf{studentOf}(\mathit{hemant}, \mathit{fp})\\
\vdash\\
?Q_1 : \mathit{admin}\ \textbf{aff}\ \mathsf{mayOpen}(\mathit{hemant}, \mathit{ghc6017})
\end{array}$$

and

$$\begin{array}{rl}
?Q_0 = \{ & \textbf{let}\ \{x_1\}_{\mathsf{admin}} = c_1\ \textbf{in}\\
& \textbf{let}\ \{x_2\}_{\mathsf{admin}} = c_2\ \textbf{in}\\
& \textbf{let}\ \{x_3\}_{\mathsf{admin}} = c_3\ \textbf{in}\\
& ?Q_1\ \}_{\mathsf{admin}}
\end{array}$$

Now x_2 together with the pair $\langle x_3, c_4\rangle$ should complete the proof. But we also need to instantiate A, B, and R, with fp, hemant, and $\mathit{ghc6017}$, respectively, which is a form of function application.

$$?Q_1 = x_2\ \mathit{fp}\ \mathit{hemant}\ \mathit{ghc6017}\ \langle x_3, c_4\rangle$$

Substituting this out in the original sequent, we get

$$\begin{array}{rl}
?Q_0 = \{ & \textbf{let}\ \{x_1\}_{\mathsf{admin}} = c_1\ \textbf{in}\\
& \textbf{let}\ \{x_2\}_{\mathsf{admin}} = c_2\ \textbf{in}\\
& \textbf{let}\ \{x_3\}_{\mathsf{admin}} = c_3\ \textbf{in}\\
& x_2\ \mathit{fp}\ \mathit{hemant}\ \mathit{ghc6017}\ \langle x_3, c_4\rangle\ \}_{\mathsf{admin}}
\end{array}$$

This proof term can now be communicated to and checked by the resource monitor.

References

Alonzo Church and J.B. Rosser. Some properties of conversion. *Transactions of the American Mathematical Society*, 39(3):472–482, May 1936.

H. B. Curry. Functionality in combinatory logic. *Proceedings of the National Academy of Sciences, U.S.A.*, 20:584–590, 1934.

W. A. Howard. The formulae-as-types notion of construction. Unpublished note. An annotated version appeared in: *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, 479–490, Academic Press (1980), 1969.