Lecture Notes on
Quantifiers and Dependent Types

15-317: Constructive Logic
Frank Pfenning André Platzer

Lecture 13
October 13, 2016

1 Introduction

We are now revisiting the missing connectives of quantifiers to study them in sequent calculus [Gen35] with a structural cut elimination result [Pfe00]. There is no major difference between the original and the restricted sequent calculus for quantifiers, so we handle both $\Gamma \Longrightarrow C$ and $\Gamma \longrightarrow C$ at once.

2 Quantification in Sequent Calculus

In natural deduction for first-order intuitionistic logic, we had two forms of hypotheses: A *true* and $c : \tau$ for parameters c. The latter form was introduced into deductions by the $\forall I$ and $\exists E$ rules. In the sequent calculus we make all assumptions explicit on the left-hand side of sequents. In order to model parameters we therefore need a second kind of judgment on the left that reads $c : \tau$. It is customary to collect all such hypotheses in a different context, denoted Σ for *signature*, that is separated from the remaining antecedent Γ by a semicolon. A sequent then has the form

$$\underbrace{c_1{:}\tau_1, \ldots, c_m{:}\tau_m}_{\Sigma} \; ; \; \underbrace{A_1 \,\mathsf{left}, \ldots, A_n \,\mathsf{left}}_{\Gamma} \Longrightarrow C \;\mathsf{right}$$

We assume that all parameters declared in a signature Σ are distinct. Sometimes this requires us to choose a parameter with a name that has not yet been used. When writing down a sequent $\Sigma; \Gamma \Longrightarrow C$ we presuppose that all parameters in Γ and C are declared in Σ. In the bottom-up construction of a deduction we make sure to maintain this.

The typing judgment for terms, $t : \tau$, can depend on the signature Σ but not on logical assumptions $A\ \mathit{left}$. We therefore write $\Sigma \vdash t : \tau$ to express that term t has type τ in signature Σ.

In all the propositional rules we have so far, the signature Σ is propagated unchanged from the conclusion of the rule to all premises. In order to derive the rules for the quantifiers, we reexamine verifications for guidance, as we did for the propositional rules in Lecture 9.

Universal quantification. We show the verification for universal quantifiers on the left and the corresponding right rule on the right.

$$\dfrac{\begin{array}{c}\overline{c : \tau} \\ \vdots \\ A(c)\!\uparrow\end{array}}{\forall x{:}\tau.\ A(x)\!\uparrow}\ \forall I^c \qquad \dfrac{\Sigma, c{:}\tau; \Gamma \Longrightarrow A(c)}{\Sigma; \Gamma \Longrightarrow \forall x{:}\tau.\ A(x)}\ \forall R$$

Our general assumption that the signature declares every parameter at most once means that c cannot occur in Σ already or the rule would not apply. Also note that Σ declares all parameters occurring in Γ, so c cannot occur there, either, which is critical for soundness. Hence, proving from assumption Γ that $A(x)$ holds for all x of type τ amounts to proving $A(c)$ for a new generic c of that type.

The elimination rule that uses a universally quantified assumption corresponds to a left rule.

$$\dfrac{\forall x{:}\tau.\ A(x)\!\downarrow \quad t : \tau}{A(t)\!\downarrow}\ \forall E \qquad \dfrac{\Sigma \vdash t : \tau \quad \Sigma; \Gamma, \forall x{:}\tau.\ A(x), A(t) \Longrightarrow C}{\Sigma; \Gamma, \forall x{:}\tau.\ A(x) \Longrightarrow C}\ \forall L$$

If we assume that $A(x)$ holds for all x of type τ, we might as well also assume that $A(t)$ also holds for a term t of said type.

Existential quantification. Again, we derive the sequent calculus rules from the introduction and elimination rules.

$$\dfrac{t : \tau \quad A(t)\!\uparrow}{\exists x{:}\tau.\ A(x)\!\uparrow}\ \exists I \qquad \dfrac{\Sigma \vdash t : \tau \quad \Sigma; \Gamma \Longrightarrow A(t)}{\Sigma; \Gamma \Longrightarrow \exists x{:}\tau.\ A(x)}\ \exists R$$

Existence of an x of type τ for which $A(x)$ is proved from assumptions Γ after $A(t)$ has been proved from assumptions Γ provided t indeed as said type τ. The arbitrary term t is called *witness*.

As for disjunction elimination, the natural deduction rule already has somewhat of the flavor of the sequent calculs.

$$\dfrac{\exists x{:}\tau.\ A(x){\downarrow} \qquad \begin{array}{c}\overline{c:\tau} \quad \overline{A(c){\downarrow}}\ u \\ \vdots \\ C{\uparrow}\end{array}}{C{\uparrow}}\ \exists E^{c,u} \qquad\qquad \dfrac{\Sigma, c{:}\tau; \Gamma, \exists x{:}\tau.\ A(x), A(c) \Longrightarrow C}{\Sigma; \Gamma, \exists x{:}\tau.\ A(x) \Longrightarrow C}\ \exists L$$

From the assumption that $A(x)$ holds for some x of type τ, C follows, if C follows from the additional assumption that $A(c)$ holds for a new generic c of that type. Of course, by the well-formedness assumptions on sequents, c will be new (so not in Σ, Γ or C), which is important, because we could hardly assume $A(c)$ to hold for our specific favorite c if all we assume is that $A(x)$ holds for some x, which does not have to be our favorite c if c is not actually new.

As an example, we prove that if there is a single x for all y such that $P(x, y)$ then it is also the case that for every y there is an x such that $P(x, y)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{a{:}\tau, b{:}\tau \vdash b{:}\tau} \qquad \dfrac{\overline{a{:}\tau, b{:}\tau \vdash a{:}\tau} \qquad \overline{a{:}\tau, b{:}\tau;\ \forall y{:}\tau.\ P(b, y), P(b, a) \Longrightarrow P(b, a)}\ init}{a{:}\tau, b{:}\tau;\ \forall y{:}\tau.\ P(b, y) \Longrightarrow P(b, a)}\ \forall L}{a{:}\tau, b{:}\tau;\ \forall y{:}\tau.\ P(b, y) \Longrightarrow \exists x{:}\tau.\ P(x, a)}\ \exists R}{a{:}\tau;\ \exists x{:}\tau.\ \forall y{:}\tau.\ P(x, y) \Longrightarrow \exists x{:}\tau.\ P(x, a)}\ \exists L}{;\ \exists x{:}\tau.\ \forall y{:}\tau.\ P(x, y) \Longrightarrow \forall y{:}\tau.\ \exists x{:}\tau.\ P(x, y)}\ \forall R}{;\ \Longrightarrow (\exists x{:}\tau.\ \forall y{:}\tau.\ P(x, y)) \supset (\forall y{:}\tau.\ \exists x{:}\tau.\ P(x, y))}\ {\supset}R$$

Note how the init rule for initial sequents applies to atomic propositions, even if they have the same parameters. So $P(b, a) \Longrightarrow P(b, a)$ closes by init, but, of course, $P(b, a) \Longrightarrow P(c, a)$ does not, because $P(b, a)$ and $P(c, a)$ are different atomic propositions since their parameters differ.

It is critically important in both $\forall R$ and $\exists L$ that the newly introduced parameter c is indeed new and has not been used in the sequent yet. Otherwise the rules would allow us to derive incorrect statements:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{a{:}\tau;\ P(a) \Longrightarrow P(a)}\ \mathit{init}}{a{:}\tau;\ P(a) \Longrightarrow \forall x{:}\tau.\ P(x)}\ \forall R??}{;\ \exists x{:}\tau.\ P(x) \Longrightarrow \forall x{:}\tau.\ P(x)}\ \exists L}{;\ \Longrightarrow (\exists x{:}\tau.\ P(x)) \supset (\forall x{:}\tau.\ P(x))}\ {\supset}R$$

3 Cut Elimination with Quantification

Continuing the cut theorem from a previous lecture, we generalize its statement to obtain the cut theorem (and cut elimination) for first-order intuitionistic logic.

Theorem 1 (Cut) *If* $\Sigma; \Gamma \Longrightarrow A$ *and* $\Sigma; \Gamma, A \Longrightarrow C$ *then* $\Sigma; \Gamma \Longrightarrow C$.

The cases for propositional connectives are as in our cut proof for intuitionistic propositional logic, just with the typing context Σ carried around.

The proof of the cut theorem extends to the case where we add quantifiers. A crucial property we need is substitution for parameters, which corresponds to a similar substitution principle on natural deductions:

Lemma 2 (Parameter substitution) *If* $\Sigma \vdash t : \tau$ *and* $\Sigma, c : \tau;\ \Gamma \vdash A$ *then also* $\Sigma;\ [t/c]\Gamma \vdash [t/c]A$.

This lemma is proved by a straightforward induction over the structure of the second deduction, appealing to some elementary properties such as weakening where necessary.

We show only two cases of the extended proof of cut, where an existential (or universal) formula is cut and was just introduced on the right and left, respectively.

Subcase:

$$\mathcal{D} = \dfrac{\begin{array}{c}\mathcal{T}\\ \Sigma \vdash t : \tau\end{array} \quad \begin{array}{c}\mathcal{D}_1\\ \Sigma; \Gamma \Longrightarrow A_1(t)\end{array}}{\Sigma; \Gamma \Longrightarrow \exists x{:}\tau.\ A_1(x)}\ \exists R$$

and $$\mathcal{E} = \dfrac{\begin{array}{c}\mathcal{E}_1\\ \Sigma, c{:}\tau; \Gamma, \exists x{:}\tau.\ A_1(x), A_1(c) \Longrightarrow C\end{array}}{\Sigma; \Gamma, \exists x{:}\tau.\ A_1(x) \Longrightarrow C}\ \exists L$$

$\Sigma; \Gamma, \exists x{:}\tau.\ A_1(x), A_1(t) \Longrightarrow C$	By substitution $[t/c]\mathcal{E}_1$ using $\mathcal{T}$
$\Sigma; \Gamma, A_1(t) \Longrightarrow C$	By i.h. on $\exists x.\ A_1(x)$, $\mathcal{D}$, and $[t/c]\mathcal{E}_1$
$\Sigma; \Gamma \Longrightarrow C$	By i.h. on $A_1(t)$, $\mathcal{D}_1$, and above

The induction requires that $A_1(t)$ is considered smaller than $\exists x.\ A_1(x)$. Formally, this can be justified by counting the number of quantifiers and logical connectives in a proposition and noting that the term t does not contain any, so quantifiers are considered bigger than terms. A similar remark applies to check that the proof $[t/c]\mathcal{E}_1$ is smaller than $\mathcal{E}$. Unlike a use of the induction hypothesis to perform a cut, a use of the parameter substitution lemma does not improve the size of the deduction (considering parameters bigger than terms of any size). Also note how the side condition that c must be a new parameter in the $\exists L$ rule is required in the substitution step to ensure that it leaves the rest of the sequent unchanged: $[t/c]\Gamma = \Gamma$, $[t/c]A_1(c) = A(t) = [t/x]A_1(x)$, as well as $[t/c]\exists x{:}\tau.\ A_1(x) = \exists x{:}\tau.\ A_1(x)$ and $[t/c]C = C$.

Subcase:

$$\mathcal{D} = \dfrac{\begin{array}{c}\mathcal{D}_1\\ \Sigma, c{:}\tau; \Gamma \Longrightarrow A_1(c)\end{array}}{\Sigma; \Gamma \Longrightarrow \forall x{:}\tau.\ A_1(x)}\ \forall R$$

$$\text{and}\quad \mathcal{E} = \dfrac{\begin{array}{c}\mathcal{T}\\ \Sigma \vdash t : \tau\end{array} \quad \begin{array}{c}\mathcal{E}_1\\ \Sigma; \Gamma, \forall x{:}\tau.\ A_1(x), A_1(t) \Longrightarrow C\end{array}}{\Sigma; \Gamma, \forall x{:}\tau.\ A_1(x) \Longrightarrow C}\ \forall L$$

$\Sigma; \Gamma, A_1(t) \Longrightarrow C$	By i.h. on $\forall x.\ A_1(x)$, $\mathcal{D}$, $\mathcal{E}_1$
$\Sigma; \Gamma \Longrightarrow A_1(t)$	By substitution $[t/c]\mathcal{D}_1$ using $\mathcal{T}$
$\Sigma; \Gamma \Longrightarrow C$	By i.h. on $A_1(t)$ and the above two

It is again important that $A_1(t)$ is considered smaller than $\forall x.\ A_1(x)$. The side condition that c must be a new parameter in $\forall R$ is needed to ensure that the substitution leaves Γ and C unchanged and that $[t/c]A_1(c) = A_1(t)$.

The restricted sequent calculus rule are the same except that the repetition of formula $\exists x{:}\tau.\ A(x)$ in the antecedent of the premise of rule $\exists L$

is unnecessary, since it will not help to give the parameter for the witness that exists by said assumption yet another name in addition to the name c. That is to be contrasted with rule $\forall L$ where it may very well be helpful to apply the universal knowledge that $A(x)$ holds for all x of type τ to many different terms t of said type. These sequent calculus rules for quantifiers are summarized in Figure 1.

$$\frac{\Sigma, c{:}\tau; \Gamma \longrightarrow A(c)}{\Sigma; \Gamma \longrightarrow \forall x.\ A(x)}\ \forall R \qquad \frac{\Sigma \vdash t : \tau \quad \Sigma; \Gamma, \forall x{:}\tau.\ A(x), A(t) \longrightarrow C}{\Sigma; \Gamma, \forall x{:}\tau.\ A(x) \longrightarrow C}\ \forall L$$

$$\frac{\Sigma \vdash t : \tau \quad \Sigma; \Gamma \longrightarrow A(t)}{\Sigma; \Gamma \longrightarrow \exists x{:}\tau.\ A(x)}\ \exists R \qquad \frac{\Sigma, c{:}\tau; \Gamma, A(c) \longrightarrow C}{\Sigma; \Gamma, \exists x{:}\tau.\ A(x) \longrightarrow C}\ \exists L$$

Figure 1: Sequent calculus rules for quantifiers

4 KeYmaera I: Uni-typed First-order Intuitionistic Logic

The KeYmaera I theorem prover you are working with in this class is forked off of the theorem prover KeYmaera X for hybrid systems combining discrete program dynamics and continuous differential equation dynamics [FMQ+15]. In that hybrid systems setting, the reals are the canonical type of interest, such that KeYmaera I only has a single object type. Since there is only one type, we then write quantifiers as $\forall x\, A(x)$ and $\exists x\, A(x)$ instead.[1]

Since KeYmaera X needs no explicit typing judgments in the sequents, KeYmaera I does not have any either (it type checks terms internally obviously). So let us drop the typing context Γ and consider what happens when we also drop all typing judgments.

$$\frac{\Gamma \longrightarrow A(c)}{\Gamma \longrightarrow \forall x\, A(x)}\ \forall R \qquad \frac{\Gamma, \forall x\, A(x), A(t) \longrightarrow C}{\Gamma, \forall x\, A(x) \longrightarrow C}\ \forall L$$

$$\frac{\Gamma \longrightarrow A(t)}{\Gamma \longrightarrow \exists x\, A(x)}\ \exists R \qquad \frac{\Gamma, A(c) \longrightarrow C}{\Gamma, \exists x\, A(x) \longrightarrow C}\ \exists L$$

Are these rules sound?

[1] By leaving out the dot from the quantifier notation, we also indicate that the scope of KeYmaera I quantifiers is short. So $\forall x\, A(x) \land B$ is $(\forall x\, A(x)) \land B$ instead of $\forall x\, (A(x) \land B)$.

No these rules are not sound unless we make up for what we lost from the assumptions that were enforced via the typing contexts. Recall that sequents $\Sigma; \Gamma \implies C$ were only allowed to mention parameters in Γ and C that are also declared in the typing context Σ. Since we just lost the typing context Σ, all parameters are allowed in the sequent. The bigger deal is that the way we enforced that rules $\forall R$ and $\exists L$ introduced fresh parameters c is via c being new in the the typing context Σ. That indirection no longer works, so we, instead, directly require c to be new, so not occurring anywhere in the sequent in the $\forall R$ and $\exists L$ rules.

With that sharpened understanding, the rules are sound as well. But it is insightful to understand one nuance that has become more subtle after having removed the explicit typing judgments from the quantifier rules. The $\forall L$ rule and $\exists R$ rule lost their typing premise $\Sigma \vdash t : \tau$ which checks that the term t is indeed of type τ in typing context Σ. When there is only a single type, this check is redundant and hence the premise dropped. But the remaining premise still needs the term t written down. How can such a term be written down at all in a proof?

Especially if there is only one type, a decision needs to be made about whether that type comes with appropriate term constructors or not. If it does provide constructors (such as $\mathtt{0} : \mathtt{nat}$), then that directly implies that something exists that satisfies no particular conditions, so $\exists x \top$ is true. Otherwise, if the single type does not afford any constructors, then it may denote the empty type that is not inhabited because nothing of this type exists. In the latter case $\neg \exists x \top$ would be a distinct possibility and explicit assumptions about the existence of at least something ($\exists x \top$) are frequently needed if one wants to talk about nonempty domains.

The possibility of types being uninhabited is an interesting one if there are multiple types. But if there is only one type and that single type could even be empty, then one is in the realm of free logic. KeYmaera I decides to have constructors for its single type in order to make sure it does not talk about an empty void. KeYmaera I is said to make the *existence presupposition*, because it assumes that something exists.

In order to understand the implications of existence presupposition (marked $\exists\exists$ in the following proof for clarity), the following sequent is provable only for inhabited types, so it is provable under existence presupposition:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A(0) \longrightarrow A(0)}\ \mathsf{init}}{A(0) \longrightarrow \exists x{:}\tau.\ A(x)}\ \exists R}{\forall x{:}\tau.\ A(x) \longrightarrow \exists x{:}\tau.\ A(x)}\ \exists L + \exists\exists}{\longrightarrow \forall x{:}\tau.\ A(x) \supset \exists x{:}\tau.\ A(x)}\ {\supset}R$$

In classical logic, existence presupposition is necessary to imply the equivalence of $\forall x{:}\tau.\ A(x)$ and $\neg\exists x{:}\tau.\ \neg A(x)$. Let us investigate whether the existence presupposition also makes them equivalent in constructive logic. One direction of the implication always works easily:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{c{:}\tau;\ \forall x{:}\tau.\ A(x), A(c), \neg A(c) \longrightarrow A(c)}\ \mathsf{init}}{c{:}\tau;\ \forall x{:}\tau.\ A(x), A(c), \neg A(c) \longrightarrow \bot}\ {\supset}L \quad \dfrac{}{c{:}\tau;\ \forall x{:}\tau.\ A(x), A(c), \bot \longrightarrow \bot}\ \bot L}{c{:}\tau;\ \forall x{:}\tau.\ A(x), \neg A(c) \longrightarrow \bot}\ \forall L}{;\ \forall x{:}\tau.\ A(x), \exists x{:}\tau.\ \neg A(x) \longrightarrow \bot}\ \exists L}{;\ \forall x{:}\tau.\ A(x) \longrightarrow \neg\exists x{:}\tau.\ \neg A(x)}\ {\supset}R}{;\ \longrightarrow (\forall x{:}\tau.\ A(x)) \supset (\neg\exists x{:}\tau.\ \neg A(x))}\ {\supset}R$$

The other direction is rather more difficult. The first steps are deterministic following what we saw in inversion and the contraction-free calculus:

$$\dfrac{\dfrac{\dfrac{c{:}\tau;\ \neg\exists x{:}\tau.\ \neg A(x) \longrightarrow \exists x{:}\tau.\ \neg A(x) \quad \dfrac{}{c{:}\tau;\ \bot \longrightarrow A(c)}\ \bot L}{c{:}\tau;\ \neg\exists x{:}\tau.\ \neg A(x) \longrightarrow A(c)}\ {\supset}L}{;\ \neg\exists x{:}\tau.\ \neg A(x) \longrightarrow \forall x{:}\tau.\ A(x)}\ \forall R}{;\ \longrightarrow (\neg\exists x{:}\tau.\ \neg A(x)) \supset (\forall x{:}\tau.\ A(x))}\ {\supset}R$$

At this point there is a choice. We could apply ${\supset}L$ again but this would loop giving the same sequent again, which, as we saw in previous lectures, cannot lead to a proof that we do not also find in shorter form without applying that redundant rule. The only other option is to use $\exists R$ to continue the first premise using some term 0 from existence presupposition:

$$\dfrac{\dfrac{\dfrac{c{:}\tau;\ \neg\exists x{:}\tau.\ \neg A(x), A(\mathsf{0}) \longrightarrow \exists x{:}\tau.\ \neg A(x) \qquad \dfrac{}{c{:}\tau;\ \bot, A(\mathsf{0}) \longrightarrow \bot}\,\bot L}{c{:}\tau;\ \neg\exists x{:}\tau.\ \neg A(x), A(\mathsf{0}) \longrightarrow \bot}\,\supset L}{c{:}\tau;\ \neg\exists x{:}\tau.\ \neg A(x) \longrightarrow \neg A(\mathsf{0})}\,\supset R}{c{:}\tau;\ \neg\exists x{:}\tau.\ \neg A(x) \longrightarrow \exists x{:}\tau.\ \neg A(x)}\,\exists R$$

At this point there is another choice to apply either $\supset L$ or $\exists R$. The former would again lead to a loop, so we do not need to pursue it since, if there is a proof at all, there is a shorter proof. The remaining premise thus has to continue with $\exists R$ either with the same suspected witness $\mathsf{0}$:

$$\dfrac{\dfrac{c{:}\tau;\ \neg\exists x{:}\tau.\ \neg A(x), A(\mathsf{0}), A(\mathsf{0}) \longrightarrow \bot}{c{:}\tau;\ \neg\exists x{:}\tau.\ \neg A(x), A(\mathsf{0}) \longrightarrow \neg A(\mathsf{0})}\,\supset R}{c{:}\tau;\ \neg\exists x{:}\tau.\ \neg A(x), A(\mathsf{0}) \longrightarrow \exists x{:}\tau.\ \neg A(x)}\,\exists R$$

at which point there is nothing left to try but loop with $\supset L$. Or to instead continue the above proof with a different witness, say t:

$$\dfrac{\dfrac{c{:}\tau;\ \neg\exists x{:}\tau.\ \neg A(x), A(\mathsf{0}), A(t) \longrightarrow \bot}{c{:}\tau;\ \neg\exists x{:}\tau.\ \neg A(x), A(\mathsf{0}) \longrightarrow \neg A(t)}\,\supset R}{c{:}\tau;\ \neg\exists x{:}\tau.\ \neg A(x), A(t) \longrightarrow \exists x{:}\tau.\ \neg A(x)}\,\exists R$$

at which point there is again nothing left to try but loop with $\supset L$. Since we have exhausted all options in the proof, we conclude that the following direction is not provable so not true in constructive logic

$$(\neg\exists x{:}\tau.\ \neg A(x)) \supset (\forall x{:}\tau.\ A(x))$$

As another illustration of the ideas from previous lectures, observe how useful it was for this proof of nonprovability to have a solid understanding of what proof search not to try. Generally, however, the terms that need to be instantiated in quantifiers make proof search matters significantly more challenging but also significantly more interesting than propositional logic.

References

[FMQ+15] Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völp, and André Platzer. KeYmaera X: An axiomatic tactical theorem prover for hybrid systems. In Amy Felty and Aart Middeldorp, editors, *CADE*, volume 9195 of *LNCS*, pages 527–538. Springer, 2015.

[Gen35] 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.

[Pfe00] Frank Pfenning. Structural cut elimination I. Intuitionistic and classical logic. *Information and Computation*, 157(1/2):84–141, March 2000.