Chapter 4

Normal Proofs - Normal Terms

We observe that in natural deduction there are many derivations for a given proposition, in particular the reasoning system admits detours. Consider the following two proofs for $A \supset B \supset A$. The proof on the left is what we consider a *normal proof* while the one one the right is not; it contains a detour, i.e. $\wedge I$ followed by $\wedge E_l$, which can be eliminated by using a local reduction.

$$\dfrac{\dfrac{\overline{A}^{\,v}}{B \supset A}\supset I^{v}}{A \supset B \supset A}\supset I^{u} \qquad\qquad \dfrac{\dfrac{\dfrac{\dfrac{\overline{A}^{\,u} \qquad \overline{B}^{\,v}}{A \wedge B}\wedge I}{A}\wedge E_l}{B \supset A}\supset I^{v}}{A \supset B \supset A}\supset I^{u}$$

The fact that the natural deduction system admits such detours and admits many proofs for the same proposition is problematic for two main reasons: First, it makes it difficult to argue when no proof exists. Even worse, we cannot even easily show that we cannot derive falsehood from no assumptions. As a consequence, we do not know whether the given system is consistent. Second, the system is not suitable for finding proofs, as there are just so many possibilities when we are constructing a proof.

As a technical device, we introduce a natural deduction calculus where we restrict derivations to *normal derivations*, i.e. derivations which do not admit detours. As we have seen when we considered proof terms for the natural deduction system, a detour meant we had a redex in our proof term, i.e. a subterm which can be reduced; our proof term is in *normal form*. In fact, it might be helpful to consider the subset of terms consisting of lambda-terms, applications, pairs and projections to characterize normal forms more precisely. We will subsequently add the other proof terms as well.

$$\begin{array}{lll} \text{Normal Terms} & M, N & ::= \lambda x{:}A.M \mid \langle M,\ N\rangle \mid () \mid R \\ \text{Neutral Terms} & R & ::= x \mid \mathsf{fst}\ R \mid \mathsf{snd}\ R \mid R\ N \end{array}$$

As we can see, a term $\lambda x{:}A.(\lambda y{:}A.y)\ x$ is not valid normal term, because we have a redex $(\lambda y{:}A.y)\ x$ which reduces to x. According to our grammar of normal and neutral terms $(\lambda y{:}A.y)\ x$ is ill-formed.

The normal natural deduction calculus, the proof calculus which only admits normal terms, captures also a very intuitive simple strategy which we already used informally when constructing proofs. When proving a proposition, we use introduction rules reasoning bottom-up (from the proposed theorem towards the hypothesis) and elimination rules top-down (from the assumptions towards the proposed theorem) meeting the result of the intro-rules.

We will introduce two new judgements to describe this proof strategy:

$M : A \uparrow$ Proposition A has a normal deduction described by the normal term M
$R\ : A \downarrow$ Proposition A is extracted from a hypothesis described the the neutral term R

We immediately give the judgements annotated with proof terms; this highlights that we are only constructing normal terms. However, we will often simply write $A \uparrow$ and $A \downarrow$, if the proof term itself is not of interest to us.

All assumptions will be written as $x : A \downarrow$, and hence the localized form, i.e. the form where explicitly list our assumptions, can be described as

$$\begin{array}{lll} u_1{:}A_1 \downarrow, \ldots, u_n{:}A_n \downarrow & \vdash & M : A \uparrow \\ u_1{:}A_1 \downarrow, \ldots, u_n{:}A_n \downarrow & \vdash & R\ : A \downarrow \end{array}$$

We write $\Gamma^{\downarrow}$ for a context $u_1{:}A_1 \downarrow, \ldots, u_n{:}A_n \downarrow$.

Let us know revisit the natural deduction rules.

Hypothesis The general hypothesis rule simply reflects the fact that from a list of assumptions, we can extract one.

$$\frac{}{\Gamma_1^{\downarrow},\ u{:}A \downarrow,\ \Gamma_2^{\downarrow} \vdash u : A \downarrow}\ u$$

Coercion The introduction and elimination rules must be able to meet and we need to be able to switch to extracting information from the assumptions. From the proof term point of view, every neutral term is also a normal term. This is achieved by a coercion.

$$\frac{\Gamma^{\downarrow} \vdash R : A \downarrow}{\Gamma^{\downarrow} \vdash R : A \uparrow} \downarrow\uparrow$$

Note that the opposite direction is not allowed; not all normal terms are neutral terms. It would also contradict our intended strategy.

Conjunction The rules for conjunction are straightforward.

$$\frac{\Gamma^{\downarrow} \vdash M : A \uparrow \qquad \Gamma^{\downarrow} \vdash N : B \uparrow}{\Gamma^{\downarrow} \vdash \langle M,\ N\rangle : A \wedge B \uparrow} \wedge I$$

$$\frac{\Gamma^{\downarrow} \vdash R : A \wedge B \downarrow}{\Gamma^{\downarrow} \vdash \mathsf{fst}\ R : A \downarrow} \wedge E_l \qquad \frac{\Gamma^{\downarrow} \vdash R : A \wedge B \downarrow}{\Gamma^{\downarrow} \vdash \mathsf{snd}\ M : B \downarrow} \wedge E_r$$

Truth The rule for truth, is also straightforward. As it is an introduction rule, it constructs a normal derivation.

$$\frac{}{\Gamma^{\downarrow} \vdash () : \top \uparrow} \top I$$

Implication The rule for implication follows a similar recipe as before. In the introduction rule, we add a new assumption labelled as neutral. In the elimination rule, we extract from the assumptions in $\Gamma^{\downarrow}$ a proof R for $A \supset B$; we now can verify that A has a normal derivation described by M and are able to extract a proof for B described by the neutral term $R\ M$.

$$\frac{\Gamma^{\downarrow}, u{:}A \downarrow \vdash M : B \uparrow}{\Gamma^{\downarrow} \vdash \lambda x{:}A.M : A \supset B \uparrow} \supset I^{u} \qquad \frac{\Gamma^{\downarrow} \vdash R : A \supset B \downarrow \qquad \Gamma^{\downarrow} \vdash M : A \uparrow}{\Gamma^{\downarrow} \vdash R\ M : B \downarrow} \supset E$$

Disjunction The introduction rules can easily be annotated with $norm$. For the elimination rule we again extract the premise $A \vee B$, hence annotating it with $\downarrow$. We choose to identify the main conclusion C with a normal term annotating it with $\uparrow$.

$$\frac{\Gamma^{\downarrow} \vdash M : A \uparrow}{\Gamma^{\downarrow} \vdash \mathsf{inl}^{A}\ M : A \lor B \uparrow} \lor I^{l} \qquad \frac{\Gamma^{\downarrow} \vdash N : B \uparrow}{\Gamma^{\downarrow} \vdash \mathsf{inr}^{B}\ N : A \lor B \uparrow} \lor I^{r}$$

$$\frac{\Gamma^{\downarrow} \vdash R : A \lor B \downarrow \qquad \Gamma^{\downarrow}, x{:}A \downarrow \vdash M_l : C \uparrow \qquad \Gamma^{\downarrow}, y{:}B \downarrow \vdash M_r : C \uparrow}{\Gamma^{\downarrow} \vdash \mathsf{case}\ R\ \mathsf{of}\ \mathsf{inl}^{A}\ x \to M_l \mid \mathsf{inr}^{B}\ y \to M_r : C \uparrow} \lor E^{x,y}$$

It would also be consistent to allow the derivations of C to be extractions, but it is not necessary to obtain a complete search procedure and complicates the relation to the sequent calculus. It also complicates our computational reading of the disjunction rule, since it would mean we extract a proposition C_l in the first branch and a (possibly another) proposition C_r in the second branch, and we then need to check that they are the same.

Falsehood If we can synthesize a contradiction, then we have constructed a proof for C. It would not make sense to have C being synthesized, i.e. being annotated with $\downarrow$, since it would be completely unrestricted.

$$\frac{\Gamma^{\downarrow} \vdash R : \bot \downarrow}{\mathsf{abort}^{C}\ M : C \uparrow} \bot E$$

Exercise 4.0.1. Annotate the rules for universal and existential quantifiers

Exercise 4.0.2. Annotate the rules for negation

$$\frac{\Gamma, u : A \vdash p}{\Gamma \vdash \neg A} \neg I^{p} \qquad \frac{\Gamma \vdash \neg A \qquad \Gamma \vdash A}{\Gamma \vdash C} \neg E$$

Theoretical properties It is quite easy to see that normal and neutral derivations are sound with respect to natural deduction. In order to state and prove this theorem, we introduce some conventions, namely we can obtain $\Gamma = u_1{:}A_1, \dots, u_n{:}A_n$ from $\Gamma^{\downarrow}$ simply by dropping the $\downarrow$ annotation from each assumption. In the opposite direction, we simply add the $\downarrow$ annotation to each assumption to obtain a $\Gamma^{\downarrow}$ from Γ.

Theorem 4.0.1 (Soundness). *1. If* $\Gamma^{\downarrow} \vdash M : A \uparrow$ *then* $\Gamma \vdash M : A$ *and*

2. If $\Gamma^{\downarrow} \vdash R : A \downarrow$ *then* $\Gamma \vdash R : A$.

Proof. By induction on the structure of the given derivation. We show only three cases, since the proof is straightforward.

Case $\mathcal{D} = \dfrac{}{\Gamma_1^{\downarrow},\ x{:}A \downarrow,\ \Gamma_2^{\downarrow} \vdash x : A \downarrow}\, x$

Then we can construct directly $\Gamma_1,\ x{:}A,\ \Gamma_2 \vdash A$.

Case $\mathcal{D} = \dfrac{\begin{matrix}\mathcal{D}'\\ \Gamma^{\downarrow} \vdash R : A \downarrow\end{matrix}}{\Gamma^{\downarrow} \vdash R : A \uparrow}\, \uparrow\downarrow$

$\Gamma \vdash R : A$ by i.h. on $\mathcal{D}'$

Case $\mathcal{D} = \dfrac{\begin{matrix}\mathcal{D}'\\ \Gamma^{\downarrow}, x{:}A \downarrow \vdash M : B \uparrow\end{matrix}}{\Gamma^{\downarrow} \vdash \lambda x{:}A.M : A \supset B \uparrow}\, \supset^{x}$

$\Gamma,\ x{:}A \vdash M : B$ by i.h. on $\mathcal{D}'$
$\Gamma \vdash \lambda x{:}A.B : A \supset B$ by $\supset I^{x}$
□

However, we note that we restricted what derivations we allow; so clearly, it is not obvious that we can translate every natural deduction derivation into a normal natural deduction proof. For example, the derivation we have given at the beginning of the chapter cannot be annotated with our rules.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \downarrow}\,u \qquad \dfrac{}{B \downarrow}\,v}{A \wedge B}\,??}{A \downarrow}\,\uparrow\downarrow}{A \uparrow}\,\wedge E_l}{B \supset A \uparrow}\,\supset I^{v}}{A \supset B \supset A \uparrow}\,\supset I^{u}$$

The problem is that while from $A \wedge B \downarrow$ we can extract $A \downarrow$, we cannot construct $A \wedge B \downarrow$. Given our assumptions $A \downarrow$ we can turn it into $A \uparrow$; similarly, we can turn $B \downarrow$ into $B \uparrow$, and conclude $A \wedge B \uparrow$. But there is not way to go from $A \wedge B \uparrow$ to $A \wedge B \downarrow$ - only the opposite direction is allowed!

To resurrect completeness, we temporarily allow the conversion

$$\frac{\Gamma^{\downarrow} \vdash M : A \uparrow}{\Gamma^{\downarrow} \vdash (M{:}A) : A \downarrow} \downarrow\uparrow$$

Computationally, we can read this rule as follows: we can synthesize a type A for the expression $M : A$, if M checks against A. We keep the proposition we check M against as part of the proof term we construct as evidence in the conclusion. Hence, this rule allows explicit type annotations where we transition.

With this rule $\downarrow\uparrow$ we can now of course translate also the non-normal derivation above. We will distinguish between the bi-directional natural deduction system *with* $\downarrow\uparrow$ rule, written as $\Gamma^{\downarrow} \vdash^{+} J$, and the bi-direction natural deduction system *without* $\downarrow\uparrow$, written as $\Gamma^{\downarrow} \vdash J$. In other words $\Gamma^{\downarrow} \vdash^{+} J$ contains all the bi-directional rules from $\Gamma^{\downarrow} \vdash J$, but in addition we can mark and identify the transitions where our derivation is not normal. These places are justified by the $\downarrow\uparrow$ rule.

$\Gamma^{\downarrow} \vdash M : A \uparrow$ $\Gamma^{\downarrow} \vdash R : A \downarrow$	Characterize only normal derivations
$\Gamma^{\downarrow} \vdash^{+} M : A \uparrow$ $\Gamma^{\downarrow} \vdash^{+} R : A \downarrow$	Characterize all normal derivations and identify non-normal derivations via the rule $\downarrow\uparrow$.

It is easy to show that the extended bi-directional natural deduction system is sound and complete with respect to the original natural natural deduction system.

Theorem 4.0.2 (Soundness).

1. *If* $\Gamma^{\downarrow} \vdash^{+} M : A \uparrow$ *then* $\Gamma \vdash M : A$.

2. *If* $\Gamma^{\downarrow} \vdash^{+} R : A \downarrow$ *then* $\Gamma \vdash R : A$.

Proof. By simultaneous structural induction over the structure of the given derivations. □

Since adding proof terms complicates the completeness theorem and the proof slightly, we omit the proof terms in the statement and proof below. In essence, the proof terms we have in the original system are not exactly the same as the ones in the bi-directional system, because we have type annotations at normal terms which are embedded within neutral terms.

Theorem 4.0.3 (Completeness of annotated natural deduction).

1. *If* $\Gamma \vdash A$ *then* $\Gamma^{\downarrow} \vdash^{+} A \uparrow$ *and* $\Gamma^{\downarrow} \vdash^{+} A \downarrow$.

Proof. By induction over the structure of the given derivation. We show only two cases. We often refer to $\Gamma^{\downarrow} \vdash^{+} A \uparrow$ as (1) and $\Gamma^{\downarrow} \vdash^{+} A \downarrow$ as (2).

Case $\mathcal{D} = \dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma \vdash A \supset B\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ \Gamma \vdash A\end{array}}{\Gamma \vdash B} \supset E$

$\Gamma^{\downarrow} \vdash A \supset B \downarrow$ by i.h. (2)

$\Gamma^{\downarrow} \vdash A \uparrow$ by i.h. (1)

$\Gamma^{\downarrow} \vdash B \downarrow$ by $\supset E$ proving (2)

$\Gamma^{\downarrow} \vdash B \uparrow$ by $\uparrow\downarrow$, proving (1)

Case $\mathcal{D} = \dfrac{\begin{array}{c}\mathcal{D}_1\\ \Gamma, x{:}A \vdash B\end{array}}{\Gamma \vdash A \supset B} \supset I^{x}$

$\Gamma^{\downarrow},\ x{:}A \downarrow \vdash B \uparrow$ by i.h. (1)

$\Gamma^{\downarrow} \vdash A \supset B \uparrow$ by $\supset I^{x}$ proving (1)

$\Gamma^{\downarrow} \vdash A \supset B \downarrow$ by $\downarrow\uparrow$ proving (2) □

Note that although natural deduction and bi-directional natural deduction extended with $\downarrow\uparrow$ rule are very similar, they are not in a bijective correspondence. In the bi-directional natural deduction system we can simply alternate the two coercions an arbitrary number of times and they are identified explicitly, while in the natural deduction system they are invisible.

Finally, we state some substitution properties for normal natural deductions. They take the following form.

Lemma 4.0.4 (Substitution property for normal natural deductions).

1. *If* $\Gamma_1^{\downarrow}, u{:}A \downarrow, \Gamma_2^{\downarrow} \vdash C \uparrow$ *and* $\Gamma_1^{\downarrow} \vdash A \downarrow$ *then* $\Gamma_1^{\downarrow}, \Gamma_2^{\downarrow} \vdash C \uparrow$.
2. *If* $\Gamma_1^{\downarrow}, u{:}A \downarrow, \Gamma_2^{\downarrow} \vdash C \downarrow$ *and* $\Gamma_1^{\downarrow} \vdash A \downarrow$ *then* $\Gamma_1^{\downarrow}, \Gamma_2^{\downarrow} \vdash C \downarrow$.

Proof. By induction on the structure of the given derivation of $C \uparrow$ and $C \downarrow$ using weakening. □

Chapter 5

First-order Logic - An Extended Discussion

So far, we have considered propositional logic and the programming language arising from it is very basic. It does not allow us to reason about data-types such as natural numbers or booleans for example.

In this chapter, we develop first-order logic which allows us to quantify over data. This will allow us to reason about data and from a proof about a given property we are able to extract a programs manipulating data. The resulting program is correct-by-construction. In practice, we rarely formally prove our programs to be correct - for real programs with mutable state or concurrency the specification of what a program is supposed to do may be challenging. Moreover, we cannot mechanically and automatically establish that a program satisfies a given invariant. However, partial invariants are useful in practical programming.

5.1 Universal and Existential Quantification

In this section, we introduce logical quantifiers. We extend our grammar for logical formulae with universal quantification, written as $\forall x{:}\tau.A(x)$, and existential quantification $\exists x{:}\tau.A(x)$.

$$\begin{array}{lll}
\text{Terms} & t & ::= \; x \mid f\ (t_1, \ldots, t_n) \\
\text{Type} & \tau & \\
\text{Propositions} & A, B, C & ::= \; \ldots \mid P(t) \mid \forall x{:}\tau.A(x) \mid \exists x{:}\tau.A(x)
\end{array}$$

We can read $\forall x{:}\tau.A(x)$ as "for all elements, the proposition $A(x)$ holds". We hence quantify over terms of type τ. $P(t)$ describes some basic predicate which depends on terms. We keep the grammar of terms abstract and simply state that terms are formed with variables and via some predefined function symbols f. First-order logic abstracts over the concrete data we are reasoning about, but it may nevertheless be useful to see specific instances where we choose τ to be nat. In this instance, our terms contain variables, 0 (nullary function symbol or constant), and $\mathsf{suc}\ t$ where suc is a unary function symbol. We can then state some simple facts about even and odd numbers using two predicates even and odd.

$$\begin{array}{l}
\forall x{:}\mathsf{nat}.\mathsf{even}\ x \supset \mathsf{odd}\ (\mathsf{suc}\ x)\\
\mathsf{even}\ 0\\
\forall x{:}\mathsf{nat}.\mathsf{even}\ x \supset \mathsf{even}\ (\mathsf{suc}\ (\mathsf{suc}\ x))
\end{array}$$

The meaning of logical quantifiers is however independent of the concrete domain τ we are reasoning about. We will come back and introduce concrete domains when we extend our logic with induction.

For now, we may ask: what does $\forall x{:}\tau.A(x)\ \mathsf{true}$ mean? Intuitively, we must require that $A(x)$ be valid for arbitrary x, since we do not choose a specific domain τ. We note that we now introduce an assumption about the new parameter x. Hence, we have two kinds of assumptions in proofs now: *hypothetical assumptions* of the form $A\ \mathsf{true}$ as for example introduced by the rules $\supset I$ or $\vee E$ and *parametric assumptions* of the form $x{:}\tau$.

$$\dfrac{\begin{array}{c}\overline{a{:}\tau}\\ \vdots\\ A(a)\ \mathsf{true}\end{array}}{\forall x{:}\tau.A(x)\ \mathsf{true}}\ \forall I^{a}
\qquad\qquad
\dfrac{\forall x{:}\tau.A(x)\ \mathsf{true}\qquad t{:}\tau}{A(t)\ \mathsf{true}}\ \forall E$$

If our domain τ is finite, we might hope to check for each element t_i in τ that $A(t_i)\ \mathsf{true}$. However, our domain may be infinite which makes this approach infeasible. Instead, we make no commitment as to what shape or property the element might have and pick one representative a. Note that a is arbitrary and fresh, i.e. it cannot have been used before in the same derivation. If we are able to establish $A(a)\ \mathsf{true}$ then it is indeed the case that $\forall x{:}\tau.A(x)\ \mathsf{true}$, because we have proven A generically for an arbitrary a.

If we have a proof of $\forall x{:}\tau.A(x)\ \mathsf{true}$, then we know that $A(x)$ is true for arbitrary x. Hence, we should be able to obtain specific instances by instantiating x with a concrete term of type τ. In the rule $\forall E$, we explicitly establish the fact that t has type τ using the judgment $t{:}\tau$. We keep the definition of $t{:}\tau$ abstract at this point, but keep in mind for every concrete domain τ we can define terms belonging to it. For example, for the domain nat, we might define

$$\frac{}{0 : \mathsf{nat}}\,N_0 \qquad \frac{t : \mathsf{nat}}{\mathsf{suc}\ t : \mathsf{nat}}\,N_{\mathsf{suc}}$$

We return to other domains shortly.

We can now prove some simple statements which are true for any domain τ such as the following:

$$\frac{\dfrac{\dfrac{\dfrac{\dfrac{}{\forall x{:}\tau.P(x) \wedge Q(x)\ \mathsf{true}}\,u \qquad \dfrac{}{a : \tau}}{P(a) \wedge Q(a)\ \mathsf{true}}\,\forall E}{P(a)\ \mathsf{true}}\,\wedge E_l}{\forall x{:}\tau.P(x)\ \mathsf{true}}\,\forall I^a \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\forall x{:}\tau.P(x) \wedge Q(x)\ \mathsf{true}}\,u \qquad \dfrac{}{b : \tau}}{P(b) \wedge Q(b)\ \mathsf{true}}\,\forall E}{Q(b)\ \mathsf{true}}\,\wedge E_r}{\forall x{:}\tau.Q(x)\ \mathsf{true}}\,\forall I^b}{\dfrac{(\forall x{:}\tau.P(x)) \wedge (\forall x{:}\tau.Q(x))\ \mathsf{true}}{(\forall x{:}\tau.P(x) \wedge Q(x)) \supset (\forall x{:}\tau.P(x)) \wedge (\forall x{:}\tau.Q(x))\ \mathsf{true}}\,\supset I^u}\,\wedge I$$

We note that the parameter a in the left branch is different from the parameter b in the right branch; we chose different names for clarity, however note since their scope is different, choosing the same name in both branches would still be fine and they would still refer be distinct.

To check that our introduction and elimination rules are harmonic, we give local soundness and completeness proofs.

$$\frac{\dfrac{\begin{array}{c}\overline{x{:}\tau}\\ \mathcal{D}\\ A(x)\ \mathsf{true}\end{array}}{\forall x{:}\tau.A(x)\ \mathsf{true}}\,\forall I^u \qquad \begin{array}{c}\mathcal{E}\\ t{:}\tau\end{array}}{A(t)\ \mathsf{true}}\,\forall E \quad \Longrightarrow \quad \begin{array}{c}\mathcal{E}\\ \hline t : \tau\\ {[t/x]}\mathcal{D}\\ A(t)\ \mathsf{true}\end{array}$$

Since the derivation $\mathcal{D}$ is parametric in x, we can simply replace all instances of x with concrete terms t of the same type.

We now check whether our elimination rules are strong enough to get all the information out they contain, i.e. can we reconstitute $\forall x{:}\tau.A(x)\ \mathsf{true}$ given a proof for it?

$$\begin{array}{c}\mathcal{D}\\ \forall x{:}\tau.A\ \mathsf{true}\end{array} \implies \dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ \forall x{:}\tau.A(x)\ \mathsf{true}\end{array} \qquad \dfrac{}{x:\tau}}{A(x)\ \mathsf{true}}\,\forall E}{\forall x{:}\tau.A(x)\ \mathsf{true}}\,\forall I^{x}$$

Let us now define the meaning of existential quantification. (Finite) universal quantification corresponds to conjunction; dually, (finite) existential quantification corresponds to disjunction. To prove $\exists x{:}\tau.A(x)$, we pick a term t from τ and show $A(t)$ true. Note that we require that t actually exists. This is an important distinction in reasoning constructively. It means we require that the type τ is in fact inhabited, i.e. elements exist and it is not empty. Classically, we are not required to provide an element explicitly. As a consequence, one might argue that constructive reasoning allows us to make more fine-grained distinction between when a domain is empty and when it is not. In fact, constructively, we can interpret the empty type as false which is often exploited in practice.

Our existential introduction rule is similar to disjunction in that we need to pick a t belonging to τ. It involves a choice.

$$\dfrac{A(t)\ \mathsf{true} \qquad t:\tau}{\exists x{:}\tau.A(x)\ \mathsf{true}}\,\exists I \qquad \dfrac{\exists x{:}\tau.A(x) \qquad \begin{array}{c}\dfrac{}{a{:}\tau}\quad \dfrac{}{A(a)\ \mathsf{true}}\,u\\ \vdots\\ C\ \mathsf{true}\end{array}}{C\ \mathsf{true}}\,\exists E^{a,u}$$

What can we deduce given a proof for $\exists x{:}\tau.A(x)$? - Although we know that there exists some element a in τ such that $A(a)$ true, we don't know which one. Recall again the elimination rule for disjunction where we had a similar dilemma. Although we have $A \vee B$ true, we do not know whether A true or B true. We therefore split the proof into two cases: Assuming A true we can prove C true and assuming B true we can prove C true. We will define existential elimination similarly; if the domain τ were finite, we would have n cases to consider: assuming $A(t_i)$ true we prove C true for all $1 \leq i \leq n$. However, we do not make any assumptions about our domain. Hence, we hence pick a fresh arbitrary parameter a and assuming $A(a)$ true we establish C true. Since a was arbitrary and we have a proof for $\exists x{:}\tau.A(x)$, we have established C true.

Let us consider an example to see how we prove statements involving existential quantification.

$$(\exists x{:}\tau.P(x) \lor Q(x)) \supset (\exists x{:}\tau.P(x)) \lor (\exists x{:}\tau.Q(x))\ \mathsf{true}$$

We show the proof in two stages, since its proof tree is quite large.

$$\dfrac{\dfrac{\dfrac{}{\exists x{:}\tau.P(x) \lor Q(x)\ \mathsf{true}}\,u \qquad \begin{array}{c} \dfrac{}{P(a) \lor Q(a)\ \mathsf{true}}\,v \quad \dfrac{}{a{:}\tau} \\ \mathcal{D}^{avu} \\ (\exists x{:}\tau.P(x)) \lor (\exists x{:}\tau.Q(x))\ \mathsf{true} \end{array}}{(\exists x{:}\tau.P(x)) \lor (\exists x{:}\tau.Q(x))\ \mathsf{true}}\,\exists E^{av}}{(\exists x{:}\tau.P(x) \lor Q(x)) \supset (\exists x{:}\tau.P(x)) \lor (\exists x{:}\tau.Q(x))\ \mathsf{true}}\,\supset I^{u}$$

We now give the derivation $\mathcal{D}^{auv}$; we write the assumptions available as a superscript.

$$\dfrac{\dfrac{}{P(a) \lor Q(a)\ \mathsf{true}}\,v \qquad \dfrac{\dfrac{\dfrac{}{P(a)\ \mathsf{true}}\,w_1 \qquad \dfrac{}{a{:}\tau}}{\exists x{:}\tau.P(x)\ \mathsf{true}}\,\exists I}{(\exists x{:}\tau.P(x)) \lor (\exists x{:}\tau.Q(x))\ \mathsf{true}}\,\lor I_r \qquad \ldots}{(\exists x{:}\tau.P(x)) \lor (\exists x{:}\tau.Q(x))\ \mathsf{true}}\,\lor E^{w_1 w_2}$$

To understand better the interaction between universal and existential quantification, let's consider the following statement.

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

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\exists x{:}\tau.\neg A(x)\ \mathsf{true}}\,u \qquad \dfrac{\dfrac{\dfrac{}{\forall x{:}\tau.A(x)\ \mathsf{true}}\,u \qquad \dfrac{}{a{:}\tau}}{A(a)\ \mathsf{true}}\,\forall E \qquad \dfrac{}{\neg A(a)\ \mathsf{true}}\,v}{\bot\ \mathsf{true}}\,\supset E}{\bot\ \mathsf{true}}\,\exists E^{av}}{\neg\forall x{:}\tau.A(x)\ \mathsf{true}}\,\supset I^{w}}{\exists x{:}\tau.\neg A(x) \supset \neg\forall x{:}\tau.A(x)\ \mathsf{true}}\,\supset I^{u}$$

Let's consider the converse;

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

After using implication introduction, we are in the following situation:

$$\dfrac{\begin{array}{c}\dfrac{}{\neg\forall x{:}\tau.A(x)\ \mathsf{true}}\,u\\ \vdots\\ \exists x{:}\tau.\neg A(x)\ \mathsf{true}\end{array}}{(\neg\forall x{:}\tau.A(x)) \supset \exists x{:}\tau.\neg A(x)\ \mathsf{true}}\supset I^u$$

But now we are stuck; to use the rule $\exists I$ we need a concrete witness for x, which we do not have, since we know nothing about the domain τ. We also cannot do anything with the assumption $\neg\forall x{:}\tau.A(x)$ which is equivalent to $\forall x{:}\tau.A(x) \supset \bot$. To eliminate it, we would need a proof of $\forall x{:}\tau.A(x)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\forall x{:}\tau.A(x)\ \mathsf{true}}\,u \qquad \dfrac{\begin{array}{c}\dfrac{}{\neg\forall x{:}\tau.A(x)\ \mathsf{true}}\,u \quad \dfrac{}{c{:}\tau}\\ \vdots\\ A(c)\ \mathsf{true}\end{array}}{\forall x{:}\tau.A(x)\ \mathsf{true}}\,\forall I^c}{\bot\ \mathsf{true}}\supset E}{\exists x{:}\tau.\neg A(x)\ \mathsf{true}}\bot E}{(\neg\forall x{:}\tau.A(x)) \supset \exists x{:}\tau.\neg A(x)\ \mathsf{true}}\supset I^u$$

But how should we obtain a proof for $A(c)$ given $\neg\forall x{:}\tau.A(x)\ \mathsf{true}$. There is not much we can do; we can attempt again to derive a contradiction using $\supset E$, but this simply leads to a loop. At the moment we do not have the syntactic tools to argue why this statement is not provable, so this argument may seem unsatisfying. We will get back to more syntactic methods of arguing why something is not provable later. It turns out that if a proof exists, it must exist without any detours (i.e. without any combinations of intro-elim rules) and moreover every proof in first-order logic must satisfy the subformula property, i.e. we can concentrate on using only introduction and elimination rules for connectives which occur in the formula we are trying to prove.

An alternative is to give a counter example by choosing a specific domain and specific predicate instantiation for A.

5.2 Encoding proofs in Tutch

We can encode first-order logic proofs in Tuch by using the following notation:

`!x:t. A(x)`	$\forall x{:}\tau.A(x)$
`?x:t. A(x)`	$\exists x{:}\tau.A(x)$

We give here the representation of the proof for previous statement

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

in Tutch:

```
proof ExNotImpNotAll : (?x:t. ~A(x)) => ~!x:t. A(x)=
begin
[ ?x:t. ~A(x);
  [ !x:t. A(x);
    [ c: t, ~A(c);
      A(c);
      F ];
    F ];
  ~!x:t. A(x) ];
(?x:t. ~A(x)) => ~!x:t. A(x);
end;
```

In order to distinguish between inferred and assumed judgments, we separate new, multiple assumptions by comas. For example the rule ∃E introduces two assumptions: `c:t` and `~A(c)`.

Existential Introduction

```
c:t;
A(c);
?x:t.A(x)
```

Existential Elimination

```
?x:t.A(x);
[c:t, A(c);


B];
B;
```

Universal Introduction

```
[c:t;



 A(c)];
!x:t.A(x)
```

Universal Elimination

```
!x:t.A(x);
c:t;
A(c);
```

We use c as a new parameter to distinguish it more clearly from bound variables.

5.3 Proof terms

Similar to proof terms for propositional logic, we can introduce proof terms for quantifiers. The proof term for introducing an existential quantifier, encapsulates the witness t together with the actual proof M. It is hence similar to a pair and we write it as $\langle M,\ t\rangle$ overloading the pair notation. The elimination rule for existentials is modeled by $\mathsf{let}\ \langle u, a,\ =\rangle M\ \mathsf{in}\ N$ where M is the proof for $\exists x{:}\tau.A(x)$ and N is the proof depending on the assumption $u : A(a)$ and $a : \tau$.

The proof term for introducing a universal quantifier is modeled by lambda-abstraction. Elimination is modeled by application. We again overload abstraction and application.

$$\text{Terms}\quad M, N \quad ::= \quad \lambda a{:}\tau.M \mid M\ t \mid \langle M,\ t\rangle \mid \mathsf{let}\ \langle u,\ a\rangle = M\ \mathsf{in}\ N$$

$$\frac{\Gamma, a{:}\tau \vdash M : A(a)}{\Gamma \vdash \lambda a : \tau.M : \forall x{:}\tau.A(x)}\ \forall I^{a} \qquad \frac{\Gamma \vdash M : \forall x{:}\tau.A(x) \quad \Gamma \vdash t{:}\tau}{\Gamma \vdash M\ t : A(t)}\ \forall E$$

$$\frac{\Gamma \vdash M : A(t) \quad \Gamma \vdash t{:}\tau}{\Gamma \vdash \langle M,\ t\rangle : \exists x{:}\tau.A(x)}\ \exists I \qquad \frac{\Gamma \vdash M : \exists x{:}\tau.A(x) \quad \Gamma, a{:}\tau, u{:}A(a) \vdash N : C}{\Gamma \vdash \mathsf{let}\ \langle u,\ a\rangle = M\ \mathsf{in}\ N : C}\ \exists E^{au}$$

We obtain two additional reduction rules.

$$\begin{array}{lcl} (\lambda a{:}\tau.M)\ t & \Longrightarrow & [t/a]M \\ \mathsf{let}\ \langle u,\ a\rangle = \langle M,\ t\rangle\ \mathsf{in}\ N & \Longrightarrow & [M/u][t/a]M \end{array}$$

Note that we also overload our substitution operation writing $[M/u]$ to replace a proof assumption u with the proof term M and writing $[t/a]$ to replace a parameter a with the term t from our reasoning domain. We assume that substitution is capture-avoiding.