Basic Proof Theory

A. S. Troelstra and H. Schwichtenberg

September 13, 2020

Contents

1 Introduction

1.1 Preliminaries

Definition 1.1. The notion of **positive**, **negative**, **strictly positive** subformula are defined in a similar style

1. A is a positive and a strictly positive subformula of itself
2. if $B \wedge C$ or $B \vee C$ is a positive [negative, strictly positive] subformula of A, then so are B, C
3. if $\forall x B$ or $\exists x B$ is a positive [negative, strictly positive] subformula of A, then so is $B[x/t]$ for any t free for x in B
4. if $B \to C$ is a positive [negative] subformula of A, then B is a negative [positive] subformula of A, and C is a positive [negative] subformula of A
5. if $B \to C$ is a strictly positive subformula of A then so is C

A strictly positive subformula of A is called a **strictly positive part (s.p.p.)** of A

1.1.1 Contexts and Formula Occurrences

Formula occurrences (f.o.'s) will play an even more important role than the formulas themselves. An f.o. is nothing but a formula with a position in another structure (prooftree, sequent, a larger formula etc.).

A **context** is nothing but a formula with an occurrences of a special propositional variable. Alternatively, a context is sometimes described as a formula with a hole in it.

Definition 1.2. We define **positive** ($\mathcal{P}$) and **negative (formula-)contexts** ($\mathcal{N}$) simultaneously by an induction definition. The symbol "$*$" functions as a special proposition letter, a **placeholder**

1. $* \in \mathcal{P}$

and if $B^+ \in \mathcal{P}, B^- \in \mathcal{N}$ and A is any formula, then

2. $A \wedge B^+, B^+ \wedge A, A \vee B^+, B^+ \vee A, A \to B^+, B^- \to A, \forall x B^+, \exists x B^+ \in \mathcal{P}$
3. $A \wedge B^-, B^- \wedge A, A \vee B^-, B^- \vee A, A \to B^-, B^+ \to A, \forall x B^-, \exists x B^- \in \mathcal{N}$

The set of **formula contexts** is the union of $\mathcal{P}$ and $\mathcal{N}$. Note that a context contains always only a single occurrence of $*$.

For arbitrary contexts we sometimes write $F[*], G[*], \ldots$ Then $F[A], G[A], \ldots$ are the formulas obtained by replacing $*$ by A

The **strictly positive** contexts $\mathcal{SP}$ are defined by

4. $* \in \mathcal{SP}$; and if $B \in \mathcal{SP}$, then
5. $A \wedge B, B \wedge A, A \vee B, B \vee A, A \to B, \forall x B, \exists x B \in \mathcal{SP}$

An alternative definition

$$\mathcal{P} = * \mid A \wedge \mathcal{P} \mid \mathcal{P} \wedge A \mid A \vee \mathcal{P} \mid \mathcal{P} \vee A \mid A \to \mathcal{P} \mid \mathcal{N} \to A \mid \forall x \mathcal{P} \mid \exists x \mathcal{P}$$
$$\mathcal{N} = A \wedge \mathcal{N} \mid \mathcal{N} \wedge A \mid A \vee \mathcal{N} \mid \mathcal{N} \vee A \mid A \to \mathcal{N} \mid \mathcal{P} \to A \mid \forall x \mathcal{N} \mid \exists x \mathcal{N}$$
$$\mathcal{SP} = * \mid A \wedge \mathcal{SP} \mid \mathcal{SP} \wedge A \mid A \vee \mathcal{SP} \mid \mathcal{SP} \vee A \mid A \to \mathcal{SP} \mid \forall x \mathcal{SP} \mid \exists x \mathcal{SP}$$

A **formula occurence** (**f.o.** for short) in a formula B is a literal subformula A together with a context indicating the place where A occurs.

Definition 1.3. The **length** or **size** of a finite tree is the number of nodes in the tree. We write $s(\mathcal{T})$ for the size of $\mathcal{T}$

The **depth** or **height** $|\mathcal{T}|$ of a tree $\mathcal{T}$ is the maximum length of the branches

The **leafszie** of a tree $\mathcal{T}$ is the number of top nodes of the trees

1.2 Simple type theories

Definition 1.4 (the set of simple types)**.** the set of **simple types** $\mathcal{T}_\to$ is constructed from a countable set of **type variables** $P_0, P_1, ...$ by means of a type-forming operation (**function-type constructor**) $\to$

1. type variables belong to $\mathcal{T}_\to$
2. if $A, B \in \mathcal{T}_\to$, then $(A \to B) \in \mathcal{T}_\to$

A type of the form $A \to B$ is called a **function type**

Definition 1.5 (Terms of the simply typed lambda calculus $\lambda_\to$)**.** All terms appear with a type; for terms of type A we use t^A, s^A, r^A. The terms are generated by the following three clauses

1. For each $A \in T_\to$ there is a countably infinite supply of variables of type A; for arbitrary variables of type A we use $u^A, v^A, w^A, x^A, y^A, z^A$
2. if $t^{A\to B}, s^A$ are terms, then $\mathrm{App}(t^{A\to B}, s^A)^B$ is a term of type B
3. if t^B is a term of type B and x^A a variable of type A, then $(\lambda x^A.t^B)^{A\to B}$

For $\mathrm{App}(t^{A\to B}, s^A)^B$ we usually write simply $(t^{A\to B}s^A)^B$

Example 1.1. $\mathbf{k}_\lambda^{A,B} := \lambda x^A y^B.x^A, \mathbf{s}_\lambda^{A,B,C} := \lambda x^{A\to(B\to C)} y^{A\to B} z^A.xz(yz)$

Definition 1.6. The set $\mathrm{FV}(t)$ of variables free in t is specified by

$$\begin{aligned}
\mathrm{FV}(x^A) &:= x^A \\
\mathrm{FV}(ts) &:= \mathrm{FV}(t) \cup \mathrm{FV}(s) \\
\mathrm{FV}(\lambda x.t) &:= \mathrm{FV}(t) \backslash \{x\}
\end{aligned}$$

Definition 1.7 (Substitution)**.** The operation of substitution of a term s for a variable x in a term t (notation $t[x/s]$) may be defined by recursion on the complexity of t, as follows

$$\begin{array}{ll} x[x/s] & := s \\ y[x/s] & := y \text{ for } y \not\equiv x \\ (t_1 t_2)[x/s] & := t_1[x/s]t_2[x/s] \\ (\lambda x.t)[x/s] & := \lambda x.t \\ (\lambda y.t)[x/s] & := \lambda y.t[x/s] \text{ for } y \not\equiv x; \text{ w.l.o.g. } y \notin \mathrm{FV}(s) \end{array}$$

Lemma 1.8 (Substitution lemma)**.** *If $x \not\equiv y, x \notin FV(t_2)$, then*

$$t[x/t_1][y/t_2] \equiv t[y/t_2][x/t_1[y/t_2]]$$

Definition 1.9 (Conversion, reduction, normal form)**.** Let T be a set of terms, and let conv be a binary relation on T, written in infix notation: t conv s. If t conv s, we say that t **converts to** s; t is called a **redex** or **convertible** term and s the **conversum** of t. The replacement of a redex by its conversum is called a **conversion**. We write $t \succ_1 s$ (t **reduces in one step to** s) if s is obtained from t by replacement of a redex t' of t by a conversum t'' of t'. The relation $\succ$ (**properly reduces to**) is the transitive closure of $\succ_1$ and $\succeq$ (**reduces to**) is the reflexive and transitive closure of $\succ_1$. The relation $\succeq$ is said to be the notion of reduction **generated** by cont.

With the notion of reduction generated by cony we associate a relation on T called **conversion equality**: $t =_{\mathrm{conv}} s$ (t is equal by conversion to s) if there is a sequence $t_0, \ldots, t_n$ with $t_0 \equiv t, t_n \equiv s$, and $t_i \preceq t_{i+1}$ or $t_i \succeq t_{i+1}$ for each $i, 0 \leq i < n$. The subscript "conv" is usually omitted when clear from the context

A term t is in **normal form**, or t is **normal**, if t does not contain a redex. t **has a normal form** if there is a normal s such that $t \succeq s$.

A **reduction sequence** is a (finite or infinite) sequence of pairs $(t_0, \delta_0), (t_1, \delta_1), \ldots$ with δ_i an (occurrence of a) redex in t_i and $t_i \succ t_{i+1}$ by conversion of δ_i, for all i. This may be written as

$$t_0 \overset{\delta_0}{\succ_1} t_1 \overset{\delta_1}{\succ_1} t_2 \overset{\delta_2}{\succ_1} \ldots$$

We often omit the δ_i, simply writing $t_0 \succ_1 t_1 \succ_1 t_2$

Finite reduction sequences are partially ordered under the initial part relation ("sequence σ is an initial part of sequence τ "); the collection of finite reduction sequences starting from a term g forms a tree, the **reduction**

tree of t. The branches of this tree may be identified with the collection of all infinite and all terminating finite reduction sequences.

A term is **strongly normalizing** (is SN) if its reduction tree is finite

β-conversion:

$$(\lambda x^A.t^B)s^A \ \ \mathrm{cont}_\beta \ \ t^B[x^A/s^A]$$

η-conversion:

$$\lambda x^A.tx \ \ \mathrm{cont}_\eta \ \ t \quad (x \notin \mathrm{FV}(t))$$

$\beta\eta$-conversion $\mathrm{cont}_{\beta\eta}$ is $\mathrm{cont}_\beta \cup \mathrm{cont}_\eta$

No free variables may become bound when executing the substitution in a β-conversion

Definition 1.10. A relation R is said to be **confluent**, or to have the **Church-Rosser property** (CR), if whenever t_0Rt_1 and t_0Rt_2, then there is a t_3 s.t. t_1Rt_3 and t_2Rt_3. A relation R is said to be **weakly confluent** or to have the **weak Church-Rosser property** if whenever t_0Rt_1, t_0Rt_2 there is a t_3 s.t. $t_1R^*t_3$ and $t_2R^*t_3$ where R^* is the reflexive and transitive closure of R

Theorem 1.11. *For a confluent reduction relation $\succeq$ the normal forms of terms are unique. Furthermore, if $\succeq$ is a confluent reduction relation we have $t = t'$ iff there is a term t'' s.t. $t \succ t''$ and $t' \succ t''$*

Proof. If $t = t'$ (for the equality induced by $\succeq$), then by definition there is a chain $t \equiv t_0, t_1, \dots, t_n \equiv t'$ s.t. for all $i < n$, $t_i \succeq t_{i+1}$ or $t_{i+1} \preceq t_i$. The existence of the required t'' is now established by induction on n. Consider the step from n to $n+1$. By induction hypothesis there is an s s.t. $t_0 \succeq s, t_n \succeq s$. If $t_{n+1} \succeq t_n$, take $s'' = s$. If $t_n \succeq t_{n+1}$, using the confluence to find a t'' s.t. $s \succeq t''$ and $t_{n+1} \succeq t''$ □

Theorem 1.12 (Newman's lemma)**.** *Let $\succeq$ be the transitive and reflexive closure of $\succ_1$, and let $\succ_1$ be weakly confluent. Then the normal form w.r.t. $\succ_1$ of a strongly normalizing t is unique. Moreover, if all terms are strongly normalizing w.r.t. $\succ_1$ then the relation $\succeq$ is confluent.*

Proof. Assume WCR, and let write $s \in UN$ to indicate that s has a unique normal form. If a term is strongly normalizing, then so are all the terms occuring in its reduction tree

Assume $t \in SN, t \notin UN$. Then there are two reduction sequences $t \succ_1 t_1' \cdots \succ_1 t'$ and $t \succ_1 t_1'' \succ_1 \cdots \succ_1 t''$ with $t' \not\equiv t''$. Then either $t_1' = t_1''$ or $t_1' \neq t_1''$

In the first case we can take $t_1 := t_1' = t_1''$. In the second case, by WCR we can find a t^* s.t. $t^* \prec t_1', t_1''$; $t \in SN$ hence $t^* \succ t'''$ for some normal t'''.

Since $t' \neq t'''$ or $t'' \neq t'''$, either $t_1' \notin UN$ or $t_1'' \notin UN$; so take $t_1 := t_1'$ if $t' \neq t'''$, $t_1 := t_1''$ otherwise.

Hence we can always find a $t_1 \prec t$ with $t_1 \notin UN$ and get an infinite sequence contradicting the SN of t □

Definition 1.13. The **simple typed lambda calculus** $\boldsymbol{\lambda}_{\rightarrow}$ is the calculus of β-reduction and β-equality on the set of terms of $\lambda_{\rightarrow}$. $\lambda_{\rightarrow}$ has the term system as described with the following axioms and rules for $\prec$ ($\prec_\beta$) and $=$ (is $=_\beta$)

$$t \succeq t \quad (\lambda x^A.t^B)s^A \succeq t^B[x^A/s^A]$$
$$\frac{t \succeq s}{rt \succeq rs} \quad \frac{t \succ s}{tr \succ sr} \quad \frac{t \succeq s}{\lambda x.t \succeq \lambda x.s} \quad \frac{t \succeq s \quad s \succeq r}{t \succeq r}$$
$$\frac{t \succeq s}{t = s} \quad \frac{t = s}{s = t} \quad \frac{t = s \quad s = r}{t = r}$$

The **extensional simple typed lambda calculus** $\lambda\eta_{\rightarrow}$ is the calculus of $\beta\eta$-reduction and $\beta\eta$-equality and the ser of terms of $\lambda_{\rightarrow}$; in addition there is the axiom

$$\lambda x.tx \succeq t \quad (x \notin \mathrm{FV}(t))$$

Lemma 1.14 (Substitutivity of $\succ_\beta$ and $\succ_{\beta\eta}$)**.** *For $\succeq$ either $\succeq_\beta$ or $\succ_{\beta\eta}$ we have*

$$\textit{if } s \succeq s' \textit{ then } s[y/s''] \succeq s'[y/s'']$$

Proof. By induction on the depth of a proof of $s \succeq s'$. It suffices to check the crucial basis step, where s is $(\lambda x.t)t'$ and s' is $t[x/t']$.

$$(\lambda x.t)t'[y/s''] = (\lambda x.(t[y/s''])t'[y/s'']) = t[y/s''][x/t'[y/s'']] = t[x/t'][y/s'']$$

□

Proposition 1.15. *$\succ_{\beta,1}$ and $\succ_{\beta\eta,1}$ are weakly confluent*

Proof. If the conversions leading from t to t' and t to t'' concern disjoint redexes, then t''' is simply obtained by converting both redexes

If $t \equiv \ldots(\lambda x.s)s' \ldots$, $t' \equiv \ldots s[x/s'] \ldots$ and $t'' \equiv \ldots(\lambda x.s)s'' \ldots$, $s' \succ_1 s''$, then $t''' \equiv \ldots s[x/s''] \ldots$ and $t' \succeq t'''$ in as many steps as there are occurrences of x in s, hence *weak*

If $t \equiv \ldots(\lambda x.s)s' \ldots$, $t' \equiv \ldots s[x/s'] \ldots$ and $t'' \equiv \ldots(\lambda x.s'')s' \ldots$, $s \succ_1 s''$, then $t''' \equiv \ldots s''[x/s'] \ldots$

If $t \equiv \ldots(\lambda x.sx)s'$, $t' = \ldots(sx)[x/s'] \ldots$, $t'' = \ldots ss' \ldots$ □

Theorem 1.16. *The terms of $\lambda_{\rightarrow}, \lambda\eta_{\rightarrow}$ are SN for $\succeq_\beta$ and $\succeq_{\beta\eta}$ respectively, then hence the β- and $\beta\eta$-normal forms are unique*

From the preceding theorem it follows that the reduction relations are confluent. This can also be proved directly, without relying on strong normalization, by the following method, due to W. W. Tait and P. Martin-Löf (see Barendregt [1984, 3.2]) which also applies to the untyped lambda calculus. The idea is to prove confluence for a relation $\succeq_p$ which intuitively corresponds to conversion of a finite set of redexes such that in case of nesting the inner redexes are converted before the outer ones.

Definition 1.17. $\succeq_p$ on $\lambda_\to$ is generated by the axiom and rules

$$(\mathrm{id}) x \succeq_p x$$

$$(\lambda\mathrm{mon})\frac{t \succeq_p t'}{\lambda x.t \succeq_p \lambda x.t'} \qquad (\mathrm{appmon})\frac{t \succeq_p t' \quad s \succeq_p s'}{ts \succeq_p t's'}$$

$$(\beta\mathrm{par})\frac{t \succeq_p t' \quad s \succeq_p s'}{(\lambda x.t)s \succeq_p t'[x/s']} (\eta\mathrm{par})\frac{t \succeq_p t'}{\lambda x.tx \succeq_p t'} (x \notin \mathrm{FV}(t))$$

Lemma 1.18 (Substitutivity of $\succ_p$)**.** *If $t \succ_p t', s \succ_p s'$, then $t[x/s] \succ_p t'[x/s']$*

Proof. By induction on t. Assume, w.l.o.g., $x \notin \mathrm{FV}(s)$

1. $t \equiv (\lambda y.t_1)t_2$, then

$$t \succeq_p t_1'[y/t_2']$$
$$t[x/s] \equiv (\lambda y.t_1[x/s])t_2[x/s] \succeq_p t_1'[x/s'][y/t_2'[x/s']] \equiv t_1'[y/t_2'][x/s']$$

2. $t \equiv \lambda x.t_1 x$

□

Lemma 1.19. *$\succeq_p$ is confluent*

Proof. Induction on t □

Theorem 1.20. *β- and $\beta\eta$-reduction are confluent*

Proof. The reflexive closure of $\succ_1$ for $\beta\eta$-reduction is contained in $\succeq_p$, and $\succeq$ is therefore the transitive closure of $\succeq_p$. Write $t \succeq_{p,n} t'$ if there is a chain $t \equiv t_0 \succeq_p t_1 \succeq_p \cdots \succeq_p t_n \equiv t'$. Then we show by induction on $n+m$ using the preceding lemma, that if $t \succeq_{p,n} t', t \succeq_{p,m} t''$ then there is a t''' s.t. $t' \succeq_{p,m} t''', t'' \succeq_{p,n} t'''$

$$\begin{tikzcd}
t \arrow[r, "\alpha-1"] \arrow[dr, "n+m+1-\alpha"'] & t_0' \arrow[r, "1"] \arrow[dr] & t' \arrow[dr, "n+m+1-\alpha"'] & \\
& t'' \arrow[r, "\alpha-1"] & t_0''' \arrow[r] & t'''
\end{tikzcd}$$

□

Definition 1.21 (Terms of typed combinatory logic $\mathbf{CL}_{\to}$)**.** The terms are inductive defined as in the case of $\lambda_{\to}$, but now with the clauses

1. For each $A \in \mathcal{T}_{\to}$ there is a countably infinite supply of variables of type A; for arbitrary variables of type A we use $u^A, v^A, w^A, x^A, y^A, z^A$
2. for each $A, B, C \in \mathcal{T}$ there are constant terms

$$\mathbf{k}^{A,B} \in A \to (B \to A)$$
$$\mathbf{s}^{A,B,C} \in (A \to (B \to C)) \to ((A \to B) \to (A \to C))$$

3. if $t^{A,B}, s^A$ are terms, then so is $t^{A,B}s$

FV($\mathbf{k}$) = FV($\mathbf{s}$) = $\emptyset$

Definition 1.22. The **weak reduction** relation $\succeq_w$ on the terms of $\mathbf{CL}_{\to}$ is generated by a conversion relation cont_w consisting of the following pairs

$$\mathbf{k}^{A,B}\,x^A y^B \;\;\mathrm{cont}_w\;\; x, \quad \mathbf{s}^{A,B,C}\,x^{A\to(B\to C)}y^{A\to B}z^A \;\;\mathrm{cont}_w\;\; xz(yz)$$

In otherwords, $\mathbf{CL}_{\to}$ is the term system defined above with the following axioms and rules for $\succeq_w$ and $=_w$

$$\begin{array}{lll} t \succeq t & \mathbf{k}\,xy \succeq x & \mathbf{s}\,xyz \succeq xz(yz) \\ \dfrac{t \succeq s}{rt \succeq rs} & \dfrac{t \succeq s}{tr \succeq sr} & \dfrac{t \succeq s \quad s \succeq r}{t \succeq r} \\ \dfrac{t \succeq s}{t = s} & \dfrac{t = s}{s = t} & \dfrac{t = s \quad s = r}{t = r} \end{array}$$

Theorem 1.23. *The weak reduction relation in $\mathbf{CL}_{\to}$, is confluent and strongly normalizing, so normal forms are unique.*

Theorem 1.24. *To each term t in $\mathbf{CL}_{\to}$, there is another term $\lambda^* x^A.t$ such that*

1. $x^A \notin FV(\lambda^* x^A.t)$
2. $(\lambda^* x^A.t)s^A \succ_w t[x^A/s^A]$

Proof.

$$\lambda^* x^A.x \;:= \mathbf{s}^{A,A\to A,A}\,\mathbf{k}^{A,A\to A}\,\mathbf{k}^{A,A}$$
$$\lambda^* x^A.y^B \;:= \mathbf{k}^{B,A}\,y^B \text{ for } y \not\equiv x$$
$$\lambda^* x^A.t_1^{B\to C}t_2^B \;:= \mathbf{s}^{A,B,C}(\lambda^* x.t_1)(\lambda^* x.t_2)$$

□

Corollary 1.25. *$\boldsymbol{CL}_\to$ is **combinatorially complete**, i.e. for every applicative combination t of $\boldsymbol{k}, \boldsymbol{s}$ and variables $x_1, x_2, \dots x_n$ there is a closed term s s.t. in $\boldsymbol{CL}_\to \vdash sx_1 \dots x_n =_w t$, in fact even $\boldsymbol{CL}_\to \vdash sx_1 \dots x_n \succeq_w t$*

Remark. Note that: it's not true that if $t = t'$ then $\lambda^* x.t = \lambda^* x.t'$. $\mathbf{k}\, x\, \mathbf{k} = x$ but $\lambda^* x.\, \mathbf{k}\, x\, \mathbf{k} = \mathbf{s}(\mathbf{s}(\mathbf{k}\,\mathbf{k})(\mathbf{s}\,\mathbf{k}\,\mathbf{k}))(\mathbf{k}\,\mathbf{k})$, $\lambda^* x.x = \mathbf{s}\,\mathbf{k}\,\mathbf{k}$

Definition 1.26. The **Church numerals** of type A are β-normal terms $\bar{n}_A$ of type $(A \to A) \to (A \to A), n \in \mathbb{N}$, defined by

$$\bar{n}_A := \lambda f^{A\to A} \lambda x^A . f^n(x)$$

where $f^0(x) := x, f^{n+1}(x) := f(f^n(x))$. $N_A = \{\bar{n}_A\}$

N.B. If we want to use $\beta\eta$-normal terms, we must use $\lambda f^{A\to A}.f$ instead of $\lambda f x.f x$ for $\bar{1}_A$

Definition 1.27. A function ﬀ$f : \mathbb{N}^k \to \mathbb{N}$ is said to be **A-representable** if there is a term F of $\lambda_\to$ s.t. (abbreviating $\bar{n}_A$ as $\bar{n}$)

$$F\bar{n}_1 \dots \bar{n}_k = f(n_1, \bar{\dots}, n_k)$$

for all $n_1, \dots, n_k \in \mathbb{N}, \bar{n}_i = (\bar{n}_i)_A$

Definition 1.28. Polynomials, extended polynomials

1. The n-argument **projections** $\mathbf{p}_i^n$ are given by $\mathbf{p}_i^n(x_1, \dots, x_n) = x_i$, the unary constant functions $\mathbf{c}_m$ by $\mathbf{c}_m(x) = m$, and sg, $\overline{\text{sg}}$ are unary functions which satisfy $\text{sg}(S_n) = 1, \text{sg}(0) = 0$, where S is the successor function.
2. The n-argument function f is the **composition** of m-argument g, n-argument $h_1, \dots, h_m$ if f satisfies $f(\bar{x}) = g(h_1(\bar{x}), \dots, h_m(\bar{x}))$
3. The **polynomials** in n variables are generated from $\mathbf{p}_i^n, \mathbf{c}_m$, addition and multiplication by closure under composition. The **extended polynomials** are generated from $\mathbf{p}_i^n, \mathbf{c}_m$, sg, $s\bar{g}$, addition and multiplication by closure under proposition

Exercise 1.2.1. Show that all terms in β-normal form of type $(P \to P) \to (P \to P)$, P a propositional variable, are either of the form $\bar{n}_P$ or of the form $\lambda f^{P\to P}.f$

Proof.
1. $\lambda f^{P\to P} \lambda x^P .t^P$ and t is in β-normal form.
2. $\lambda f^{P\to P}.f$

□

Theorem 1.29. *All extended polynomials are representable in $\lambda_\to$*

Proof. Abbreviate $\mathbb{N}_A$ as N.

$$
\begin{aligned}
F_+ &:= \lambda x^N y^N f^{A\to A} z^A . x f (y f z)\\
F_\times &:= \lambda x^N y^N f^{A\to A} . x(yf)\\
F_{\mathbf{p}_i^k} &:= \lambda x_1^N \dots x_k^N . x_i\\
F_{\mathbf{c}_n} &:= \lambda x^N . \overline{n}\\
F_{\mathrm{sg}} &:= \lambda x^N f^{A\to A} z^A . x(\lambda u^A . f z) z\\
F_{\overline{\mathrm{sg}}} &:= \lambda x^N f^{A\to A} z^A . x(\lambda u^A . z)(f z)
\end{aligned}
$$

□

1.3 Three Types of Formalism

1.3.1 The BHK-interpretation

Minimal logic and intuitionistic logic differ only in the treatment of negation, or (equivalently) falsehood, and minimal implication logic is the same as intuitionistic implication logic

The informal interpretation underlying intuitionistic logic is the Brouwer-Heyting-Kolmogorov interpretation; this interpretation tells us what it means to prove a compound statement such as $A \to B$ in terms of what it means to prove the components B and A

> A construction p proves $A \to B$ if p transforms any possible proof q of A into a proof $p(q)$ of B

The BHK-interpretation of intuitionistic logic is based on the notion of **proof** instead of truth

A **logical law** of implication logic, according to the BHK-interpretation, is a formula for which we can give a proof, no matter how we interpret the atomic formulas. A **rule** is valid for this interpretation if we know how to construct a proof for the conclusion, given proofs of the premises

The following two rules for $\to$ are obviously valid on the basis of the BHK-interpretation:

1. If, starting from a hypothetical (unspecified) proof u of A, we can find a proof $t(u)$ of B, then we have in fact given a proof of $A \to B$ (without the assumption that u proves A). This proof may be denoted by $\lambda u.t(u)$.

2. Given a proof t of $A \to B$, and a proof s of A, we can apply t to s to obtain a proof of B. For this proof we may write $\mathrm{App}(t, s)$ or ts (t applied to s).

1.3.2 A natural deduction system for minimal implication logic

Characteristic for natural deduction is the use of assumptions which may be **closed** at some later step in the deduction.

The assumptions in a deduction which are occurrences of the same formula with the same marker form together an **assumption class**. The notations

$$\begin{array}{llll} [A]^u & A^u & \mathcal{D}' & \mathcal{D}' \\ \mathcal{D} & \mathcal{D} & [A] & A \\ B & B & \mathcal{D} & \mathcal{D} \\ & & B & B \end{array}$$

have the following meaning, from left to right:

1. a deduction $\mathcal{D}$, with conclusion B and a set $[A]$ of open assumptions, consisting of all occurrences of the formula A at top nodes of the prooftree $\mathcal{D}$ with marker u (note: both B and the $[A]$ are part of $\mathcal{D}$, and we do not talk about the **multiset** $[A]^u$ since we are dealing with formula occurrences);
2. a deduction $\mathcal{D}$, with conclusion B and a single assumption of the form A marked u occurring at some top node;
3. deduction $\mathcal{D}$ with a deduction $\mathcal{D}'$, with conclusion A, substituted for the assumptions $[A]^u$ of $\mathcal{D}$; (4) the same, but now for a single assumption occurrence A in $\mathcal{D}$.
4. the formula A shown is the conclusion of $\mathcal{D}'$ as well as the formula in an assumption class of $\mathcal{D}$.

We now consider a system $\to$**Nm** for the minimal theory of implication.

A single formula occurrence A labelled with a marker is a single-node prooftree, representing a deduction with conclusion A from open assumption A.

$$\dfrac{\begin{array}{c}[A]^u \\ \mathcal{D} \\ B\end{array}}{A \to B}\,{\to}\mathrm{I}, u \qquad\qquad \dfrac{\begin{array}{c}\mathcal{D} \\ A \to B\end{array} \quad \begin{array}{c}\mathcal{D}' \\ A\end{array}}{B}\,{\to}\mathrm{E}$$

By application of the rule →I of **implication introduction**, a new prooftree is formed from $\mathcal{D}$ by adding at the bottom the conclusion $A \to B$ while **closing** the set of open assumptions A marked by u. All other open assumptions remain open in the new prooftree

The rule →E of **implication elimination** (also known as **modus ponens**) constructs from two deductions $\mathcal{D}, \mathcal{D}'$ with conclusions $A \to B, A$ a new combined deduction with conclusion B, which has as open assumptions the open assumptions of $\mathcal{D}$ and $\mathcal{D}'$ combined

Two occurrences α,β of the same formula belong to the same **assumption class** if they bear the same label and either are both open or have both been closed at the same inference.

It should be noted that in the rule →I the "degenerate case", where $[A]^u$ is empty, is permitted; thus for example the following is a correct deduction:

$$\dfrac{\dfrac{A^u}{B \to A}\,v}{A \to (B \to A)}\,u$$

1.3.3 Formulas-as-types

1. To assumptions A correspond variables of type A; more precisely, formulas with the same marker get the same variable.
2. For the rules →I and →E the assignment of terms to the conclusion is shown below

$$\dfrac{\begin{matrix}[u : A] \\ \mathcal{C} \\ t : B\end{matrix}}{\lambda u^A.t^B : A \to B}\,u \qquad \dfrac{\begin{matrix}\mathcal{D} \\ t : A \to B\end{matrix} \qquad \begin{matrix}\mathcal{D}' \\ s : A\end{matrix}}{(t^{A \to B} s^A) : B}$$

Thus there is a very close relationship between $\lambda_{\to}$ and →**Nm**

A β-conversion

$$(\lambda x^A.t^B)s^A \ \ \mathrm{cont}_\beta \ \ t^B[x^A/s^A]$$

corresponds to a transformation on prooftrees:

$$\dfrac{\dfrac{\begin{matrix}[A]^u \\ \mathcal{D} \\ B\end{matrix}}{A \to B}\,u \qquad \begin{matrix}\mathcal{D}' \\ A\end{matrix}}{B} \quad \mapsto \quad \begin{matrix}\mathcal{D}' \\ [A] \\ \mathcal{D} \\ B\end{matrix}$$

A proof without detours is said to be a **normal** proof. In a normal proof the left premise of →E is never the conclusion of →I

1.3.4 Gentzen systems

There are two motivations leading to Gentzen systems, which will be discussed below. The first one views a Gentzen system as a metacalculus for natural deduction; this applies in particular to systems for minimal and intuitionistic logic. The second motivation is semantical: Gentzen systems for classical logic are obtained by analysing truth conditions for formulas. This also applies to intuitionistic and minimal logic if we use Kripke semantics instead of classical semantics.

A **Gentzen system as a metacalculus**. Let us first consider a Gentzen system obtained as a metacalculus for the system $\rightarrow$**Nm**. Consider the following four construction steps for prooftrees.

1. The single-node tree with label A, marker u is a prooftree
2. Add at the bottom of a prooftree an application of $\rightarrow$I, discharging an assumption class
3. Given a prooftree $\mathcal{D}$ with open assumption class $[B]^u$ and a prooftree $\mathcal{D}_1$ deriving A, replace all occurrences of B in $[B]^u$ by

$$\dfrac{A \to B^v \qquad \begin{array}{c}\mathcal{D}_1\\ A\end{array}}{B}$$

4. Substitute a deduction of A for the occurrences of an (open) assumption class $[A]^u$ of another deduction

These construction principles suffice to obtain any prooftree of $\rightarrow$**Nm**. The closure under $\rightarrow$E is seen as follows: in order to obtain the tree

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ A \to B\end{array} \qquad \begin{array}{c}\mathcal{D}_2\\ A\end{array}}{B}$$

we first combine the first and third construction principles to obtain

$$\dfrac{A \to B^u \qquad \begin{array}{c}\mathcal{D}_2\\ A\end{array}}{B}$$

and then use the fourth principle to obtain the desired tree

Let $\Gamma \Rightarrow A$ express that A is deducible in $\rightarrow$**Nm** from assumptions in Γ. Then the four construction principles correspond to the following axiom and rules

$$\Gamma \cup \{A\} \Rightarrow A \text{ (Axiom)}$$

$$\frac{\Gamma \cup \{A\} \Rightarrow B}{\Gamma \Rightarrow A \to B}\,\mathrm{R}{\to} \qquad \frac{\Gamma \Rightarrow A \qquad \Delta \cup \{B\} \Rightarrow C}{\Gamma \cup \Delta \cup \{A \to B\} \Rightarrow C}\,\mathrm{L}{\to}$$

$$\frac{\Gamma \Rightarrow A \qquad \Delta \cup \{A\} \Rightarrow B}{\Gamma \cup \Delta \Rightarrow B}\,\mathrm{Cut}$$

Call the resulting system $\mathcal{S}$. Here in the sequents $\Gamma \Rightarrow A$ the Γ is treated as a (finite) set. If we rewrite the system above with multisets, we get the Genzten system $\mathcal{S}'$ described below.

$$A \Rightarrow A \text{ (Axiom)}$$

$$\frac{\Gamma \Rightarrow A \qquad \Delta, B \Rightarrow C}{\Gamma, \Delta, A \to B \Rightarrow C}\,\mathrm{L}{\to} \qquad \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \to B}\,\mathrm{R}{\to}$$

$$\frac{\Gamma \Rightarrow A}{\Gamma, B \Rightarrow A}\,\mathrm{LW} \qquad \frac{\Gamma, B, B, \Rightarrow A}{\Gamma, B \Rightarrow A}\,\mathrm{LC}$$

$$\frac{\Gamma \Rightarrow A \qquad A, \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow B}\,\mathrm{Cut}$$

R→ and L→ are called the logical rules, LW, LC and Cut the structural rules. LC is called the rule of (left-) **contraction**, LW the rule of (left-) **weakening**.

It is not hard to convince oneself that, as long as only the principles 1-3 for the construction of prooftrees are applied, the resulting proof will always be **normal**. Conversely, it may be proved that all normal prooftrees can be obtained using construction principles 1-3 only. Thus we see that normal prooftrees in →**Nm** correspond to deduction in the sequent calculus without Cut;

Deductions in $\mathcal{S}$ without the rule Cut have a very nice property, which is immediately obvious: the **subformula property**: all formulas occurring in a deduction of $\Gamma \Rightarrow A$ are subformulas of the formulas in Γ, A.

Exercise 1.3.1. There are other possible choices for the construction principles for prooftrees. For example, we might replace principle 3 by the following principle 3′:

Given a prooftree $\mathcal{D}$ with open assumption class $[B]^u$, replace all occurrences of B in $[B]^u$ by

$$\frac{A \to B^v \qquad A}{B}$$

1.3.5 Semantical motivation of Gentzen systems

Here we use sequents $\Gamma \Rightarrow \Delta$ with Γ and Δ finite sets; the intuitive interpretation is that $\Gamma \Rightarrow \Delta$ is valid iff $\bigwedge \Gamma \to \bigvee \Delta$ is true. Now suppose we want to find out if there is a valuation making all of Γ true and all of Δ false. We can break down this problem by means of two rules, one for reducing $A \to B$ on the left, another for reducing $A \to B$ on the right:

$$\frac{\Gamma \Rightarrow A, \Delta \qquad \Gamma, B \Rightarrow \Delta}{\Gamma, A \to B \Rightarrow \Delta}\,\mathrm{L}\!\to \qquad\qquad \frac{\Gamma, A, \Rightarrow B, \Delta}{\Gamma \Rightarrow A \to B, \Delta}\,\mathrm{R}\!\to$$

1.3.6 A Hilbert system

The Hilbert system $\to$**Hm** for minimal implication logic has as axioms all formulas of the forms:

$$A \to (B \to A) \quad \text{k-axioms}$$
$$(A \to (B \to C)) \to ((A \to B) \to (A \to C)) \quad (s - axioms)$$

The corresponding term system for $\to$**Hm** is $\mathbf{CL}_{\to}$

2 N-systems and H-systems

2.1 Natural Deduction Systems

Definition 2.1 (*The systems* **Nm**, **Ni**, **Nc**)**.** Assumptions are formula occurrences always appearing at the top of a branch and are supposed to be labelled by markers. The set of assumptions of the same form with the same marker forms an **assumption class**. Distinct formulas must have distinct markers. We permit empty assumption classes.

Assumptions may be closed; assumption classes are always closed "en bloc", that is to say, at each inference, either all assumptions in a class are closed, or they are left open.

Deductions in the system of natural deduction are generated as follows. *Basis*. The single-node tree with label A is a (natural) **deduction** from the open assumption A; there are no closed assumptions
Inductive step. Let $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ be deductions. A (natural) **deduction** $\mathcal{D}$ may be constructed according to one of the rules below. The class $[A]^u, [B]^v$

For $\wedge, \vee, \to, \forall, \exists$ we have **introduction rules** (**I-rules**) and **elimination rules** (**E-rules**)

$$\dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_2\\ A & B\end{array}}{A \wedge B}\wedge\mathrm{I} \qquad \dfrac{\begin{array}{c}\mathcal{D}_1\\ A \wedge B\end{array}}{A}\wedge\mathrm{E}_\mathrm{R} \qquad \dfrac{\begin{array}{c}\mathcal{D}_1\\ A \wedge B\end{array}}{B}\wedge\mathrm{E}_\mathrm{L}$$

$$\dfrac{\begin{array}{c}[A]^u\\ \mathcal{D}_1\\ B\end{array}}{A \to B}\to\mathrm{I}, u \qquad \dfrac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}\\ A \to B & A\end{array}}{B}\to\mathrm{E}$$

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ A\end{array}}{A \vee B}\vee\mathrm{I}_\mathrm{R} \qquad \dfrac{\begin{array}{c}\mathcal{D}_1\\ B\end{array}}{A \vee B}\vee\mathrm{I}_\mathrm{L} \qquad \dfrac{\begin{array}{ccc} & [A]^u & [B]^u\\ \mathcal{D}_1 & \mathcal{D}_2 & \mathcal{D}_3\\ A \vee B & C & C\end{array}}{C}\vee\mathrm{E}, u, v$$

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ A[x/y]\end{array}}{\forall x A}\forall\mathrm{I} \qquad \dfrac{\begin{array}{c}\mathcal{D}_1\\ \forall x A\end{array}}{A[x/t]}\forall\mathrm{E}$$

In $\forall$I, $y \equiv x$ or $y \notin \mathrm{FV}(A)$, and y is not free in any assumption open in $\mathcal{D}_1$

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ A[x/t]\end{array}}{\exists x A}\exists\mathrm{I} \qquad \dfrac{\begin{array}{cc} & [A[x/y]]^u\\ \mathcal{D}_1 & \mathcal{D}_2\\ \exists x A & C\end{array}}{C}\exists\mathrm{E}, u$$

In $\exists$I, $y \equiv x$ or $y \notin \mathrm{FV}(A)$, and y is not free in C nor in any assumption open in $\mathcal{D}_2$ except in $[A[x/y]]^u$

This completes the description of the rules for the minimal logic **Nm**. Note that $\perp$ has not been mentioned in any of the above rules, and therefore it behaves in minimal logic as an arbitrary unprovable propositional constant.

To obtain the intuitionistic and classical system **Ni** and **Nc** we add the **intuitionistic absurdity rule** $\perp_\mathrm{i}$ and the more general **classical absurdity rule** $\perp_\mathrm{c}$ respectively:

$$\dfrac{\begin{array}{c}\mathcal{D}_1\\ \perp\end{array}}{A}\perp_\mathrm{i} \qquad \dfrac{\begin{array}{c}[\neg A]^u\\ \mathcal{D}_1\\ \perp\end{array}}{A}\perp_\mathrm{c}, u$$

($\perp_\mathrm{c}$ is more general than $\perp_\mathrm{i}$ since $[\neg A]^u$ may be empty). In an E-rule application, the premise containing the occurence of the logical operator being eliminated is called the **major** premise. The other premise(s) are called the **minor** premise(s). As a standard convention in displaying prooftrees, we place the major premises of elimination rule applications in leftmost position.

As to individual variables which are considered to be free in deduction, we stipulate

- The deduction consisting of assumption A only has $\mathrm{FV}(A)$ as free variables;
- at each rule application, the free individual variables are inherited from the immediate subdeduction, except that
- in an application of $\exists$E the occurrences of the free variable y in $\mathcal{D}_2$ become bound, and in an application of $\forall$I the occurrences of variable y in $\mathcal{D}_1$ become bound, and
- in $\to$I the vairables in $\mathrm{FV}(A)$ have to be added in case $[A]^u$ is empty, in $\vee\mathrm{I}_\mathrm{R}$ those in $\mathrm{FV}(B)$ have to be added, and in $\vee\mathrm{I}_\mathrm{L}$ those in $\mathrm{FV}(A)$ have to be added

The individual variable becoming bound in an application α of $\forall$I or $\exists$E is said to be the **proper** variable of α.

If A is among the open assumptions of a dduction $\mathcal{D}$ with conclusion B, then conclusion B in $\mathcal{D}$ is said to **depend** on A in $\mathcal{D}$. From now on we regard "assumption of $\mathcal{D}$" and "open assumption of $\mathcal{D}$" as synonymous.

Definition 2.2. A convenient global assumption in the presentation of a deduction is the **variable convention**. A deduction is said to satisfy the variable convention if the proper variables of the application of $\exists$E and $\forall$I are kept distinct.

If moreover the bound and free variables are kept distinct, the deduction is said to be a **pure-variable** deduction.

Remark. Since in out notation for prooftrees, $[A]^u$ refers to all assumptions A labelled u, it is tacitly understood that in $\vee$E the label u occurs in $\mathcal{D}_2$ only, and v in $\mathcal{D}_3$ only.

Definition 2.3. The theories generated by **Nm**, **Ni** and **Nc** are denoted by **M** (minimal logic), **I** (intuitionistic logic) and **C** (classical logic) respectively.

$\Gamma \vdash_{\mathbf{S}} A$ for $\mathbf{S} = \mathbf{M}, \mathbf{I}, \mathbf{C}$ iff A is derivable from the set of assumptions Γ in the N-systems for **S**

Remark. 1. Sometimes it is more natural to write $\forall$E and $\exists$I as two-premise rules, with the individual term as second premise

$$\frac{\forall x A \qquad t}{A[x/t]} \qquad\qquad \frac{A[x/t] \qquad t}{\exists x A}$$

2. The statement of the rules $\forall$I and $\exists$E may be simplified somewhat if we rely on convention that formulas differing only in the naming of

bound variables are equal. These rules may then be written as:

$$\dfrac{\begin{matrix}\mathcal{D}_1\\ A\end{matrix}}{\forall x A}\,\forall\mathrm{I} \qquad\qquad \dfrac{\begin{matrix}\mathcal{D}_1\\ \exists x A\end{matrix} \qquad \begin{matrix}[A]^u\\ \mathcal{D}_2\\ C\end{matrix}}{C}\,\exists\mathrm{E}, u$$

where in $\forall$I x is not free in any assumption open in $\mathcal{D}_1$ and and $\exists$E x is not free in C nor in any assumption open in $\mathcal{D}_2$ except in $[A]^u$

2.1.1 Natural deductions in sequent style

We call the set of open assumptions at a node the **context**. A context is a set

$$u_1 : A_1, u_2 : A_2, \dots, u_n : A_n$$

where u_i are pairwise distinct; the A_i need not be distinct. The deductions now become trees where each node is labelled with a sequent of the form $\Gamma \Rightarrow B$, Γ is a context. Below, when writing a union of contexts such as $\Gamma\Delta$ (short for $\Gamma \cup \Delta$), it will always be assumed that the union is **consistent**

$u : A \Rightarrow A$ (Axiom)

$$\dfrac{\Gamma[u : A] \Rightarrow B}{\Gamma \Rightarrow A \to B}\,\to\mathrm{I} \qquad \dfrac{\Gamma \Rightarrow A \to B \qquad \Delta \Rightarrow A}{\Gamma\Delta \Rightarrow B}\,\to\mathrm{E}$$

$$\dfrac{\Gamma \Rightarrow A \qquad \Delta \Rightarrow B}{\Gamma\Delta \Rightarrow A \wedge B}\,\wedge\mathrm{I} \qquad \dfrac{\Gamma \Rightarrow A_0 \wedge A_1}{\Gamma \Rightarrow A_i}\,\wedge\mathrm{E}$$

$$\dfrac{\Gamma \Rightarrow A_i}{\Gamma \Rightarrow A_0 \vee A_1}\,\vee\mathrm{I} \qquad \dfrac{\Gamma \Rightarrow A \vee B \quad \Delta[u : A] \Rightarrow C \quad \Delta'[v : B] \Rightarrow C}{\Gamma\Delta\Delta' \Rightarrow C}\,\vee\mathrm{E}$$

$$\dfrac{\Gamma[x : \neg A] \Rightarrow \bot}{\Gamma \Rightarrow A}\,\bot_{\mathrm{c}} \qquad \dfrac{\Gamma \Rightarrow \bot}{\Gamma \Rightarrow A}\,\bot_{\mathrm{i}}$$

$$\dfrac{\Gamma \Rightarrow [x/y]}{\Gamma \Rightarrow \forall x A}\,\forall\mathrm{I} \qquad \dfrac{\Gamma \Rightarrow \forall x A}{\Gamma \Rightarrow A[x/t]}\,\forall\mathrm{E}$$

$$\dfrac{\Gamma \Rightarrow A[x/t]}{\Gamma \Rightarrow \exists x A}\,\exists\mathrm{I} \qquad \dfrac{\Gamma \Rightarrow \exists y A[x/y] \qquad \Delta[u : A] \Rightarrow C}{\Gamma\delta \Rightarrow C}\,\exists\mathrm{E}$$

Here $[u : C]$ means that the assumption $u : C$ in the context may be present or absent.

Exercise 2.1.1. Give proofs in **Nm** or in **Ni** of

$$\begin{aligned}
&A \to (B \to A)\\
&(A \to A \vee B), B \to (A \vee B)\\
&(A \to C) \to [(B \to C) \to (A \vee B \to C)]\\
&A \wedge B \to A, A \wedge B \to B, A \to (B \to A \wedge B)\\
&\perp \to A\\
&\forall x A \to A[x/t]; \quad A[x/t] \to \exists x A\\
&\forall x(B \to A) \leftrightarrow (B \to \forall y A[x/y]) \;\; (x \notin \mathrm{FV}(B), y \equiv x \text{ or } y \notin \mathrm{FV}(A))\\
&\forall x(A \to B) \leftrightarrow (\exists y A[x/y] \to B) \;\; (x \notin \mathrm{FV}(B), y \equiv x \text{ or } y \notin \mathrm{FV}(A))
\end{aligned}$$

Proof.

$$\dfrac{\dfrac{A^u}{B \to A}\,{\to}\mathrm{I}, v}{A \to (B \to A)}\,{\to}\mathrm{I}, u \qquad \dfrac{\dfrac{A^u}{A \vee B}\,\forall\mathrm{I}_\mathrm{R}}{A \to (A \vee B)}\,{\to}\mathrm{I}, u$$

□

Exercise 2.1.2. Give proofs in **Nm** of

$$\begin{aligned}
&A \to \neg\neg A\\
&\neg\neg\neg A \leftrightarrow \neg A\\
&\neg\neg(A \to B) \to (\neg\neg A \to \neg\neg B)\\
&\neg\neg(A \wedge B) \leftrightarrow (\neg\neg A \wedge \neg\neg B)\\
&\neg(A \vee B) \leftrightarrow (\neg A \wedge \neg B)\\
&\neg\neg\forall x A \to \forall x \neg\neg A
\end{aligned}$$

Proof.

$$\dfrac{\dfrac{\dfrac{A \to \perp^u \qquad A^v}{\perp}\,{\to}\mathrm{E}}{(A \to \perp) \to \perp}\,{\to}, u}{A \to ((A \to \perp) \to \perp)}\,{\to}, v \qquad \dfrac{\neg\neg\neg A^u \qquad \begin{matrix} A^v \\ \mathcal{D} \\ \neg\neg A \end{matrix}}{\perp}$$

□

Exercise 2.1.3. Give proofs in **Nm** of

1. $(B \to C) \to (A \to B) \to A \to C$ (**b**-axioms)
2. $(A \to B \to C) \to B \to A \to C$ (**c**-axioms)

3. $(A \to A \to B) \to A \to B$ (**w**-axioms)

Exercise 2.1.4 ($\star$). Prove in **Ni** $(\neg\neg A \to \neg\neg B) \to \neg\neg(A \to B)$

Proof.

$$\dfrac{\dfrac{(A \to B) \to \bot^{v} \qquad \dfrac{\begin{matrix}\mathcal{D}\\ B \to (A \to B)\end{matrix} \qquad B^{u}}{A \to B}}{\bot}}{B \to \bot}\ \to, u$$

□

Exercise 2.1.5. Prove in **Nc**

1. $A \vee B \leftrightarrow \neg(\neg A \wedge \neg B)$
2. $\exists x A \leftrightarrow \neg\forall x \neg A$
3. $((A \to B) \to A) \to A$ (Peirce's law)

Proof.

$$\dfrac{\dfrac{(\neg A \wedge \neg B) \to \bot^{u} \qquad \begin{matrix}(\neg(A \vee B))^{v}\\ \mathcal{D}\\ (\neg A \wedge \neg B)\end{matrix}}{\bot}}{A \vee B}$$

□

Exercise 2.1.6. Construct in $\to$**Nm** a proof of

$$((A \to B) \to C) \to (A \to C) \to C$$

from two instances of Peirce's law as assumptions: $((A \to B) \to A) \to A$ and $((C \to A) \to C) \to C$

Exercise 2.1.7. Derive in $\to$**Nm** $P_{A,B\wedge C}$ from $P_{A,B}$ and $P_{A,C}$ where $P_{X,Y}$ is $((X \to Y) \to X) \to X$

Exercise 2.1.8. Let $F[*], G[*]$ be a positive and negative context respectively. Prove in **Nm** that

$$\vdash \forall\vec{x}(A \to B) \to (F[A] \to F[B])$$
$$\vdash \forall\vec{x}(A \to B) \to (G[A] \to G[B])$$

where $\vec{x}$ consists of the variables in $A \to B$ becoming bound by substitution of A and B into $F[*]$ in the first line, and into $G[*]$ in the second line

2.1.2 The Complete Discharge Convention

wef