Decorating natural deduction

Helmut Schwichtenberg
(j.w.w. Diana Ratiu)

Mathematisches Institut, LMU, München

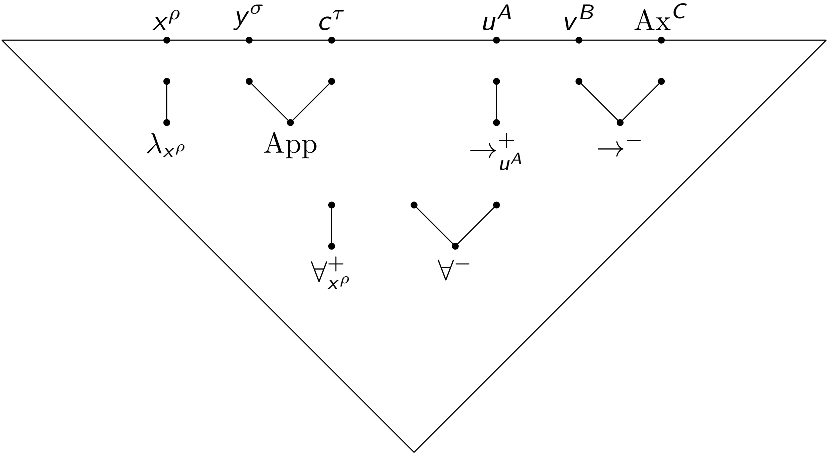
General Proof Theory, Tübingen, 27. - 29. November 2015

- Proofs may have computational content, which can be extracted (via realizability).
- Proofs (as opposed to programs) can easily be checked for correctness.

Issues:

- Why proofs in natural deduction?
- Complexity.

Proof terms in natural deduction



The realizability interpretation transforms such a proof term directly into an object term.

Logic

- Use $\to$, $\forall$ only, defined by introduction and elimination rules.
- View $\exists_x A$, $A \vee B$, $A \wedge B$ as inductively defined predicates (with parameters A, B).
- In addition, define classical existence and disjunction by

$$\tilde{\exists}_x A := \neg \forall_x \neg A,$$
$$A \mathbin{\tilde{\vee}} B := \neg(\neg A \wedge \neg B)$$

where $\neg A := (A \to \mathbf{F})$ and $\mathbf{F} := (0 = 1)$.

Example: disjunction

$A \vee B$ is inductively defined by the clauses (introduction axioms)

$$A \to A \vee B, \qquad B \to A \vee B$$

with least-fixed-point (elimination) axiom

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

Decoration

- Goal: fine tune the computational content of a proof.
- Tool: distinguish $\to^{\mathrm{c}}$, $\forall^{\mathrm{c}}$ (computational) and $\to^{\mathrm{nc}}$, $\forall^{\mathrm{nc}}$ (non-computational).

The rules for $(\to^{\mathrm{nc}})^+$, $(\forall^{\mathrm{nc}})^+$ are restricted: the abstracted (object or assumption) variable must not be "used computationally".

Remark: Coq uses Set and Prop instead (but this is less flexible).

Example: computational variants of disjunction

We have four possibilities to decorate the two clauses for $\vee$:

$$\begin{cases} A \to^{\mathrm{c}} A \vee^{\mathrm{d}} B \\ B \to^{\mathrm{c}} A \vee^{\mathrm{d}} B \end{cases} \quad \begin{cases} A \to^{\mathrm{c}} A \vee^{\mathrm{l}} B \\ B \to^{\mathrm{nc}} A \vee^{\mathrm{l}} B \end{cases} \quad \begin{cases} A \to^{\mathrm{nc}} A \vee^{\mathrm{r}} B \\ B \to^{\mathrm{c}} A \vee^{\mathrm{r}} B \end{cases} \quad \begin{cases} A \to^{\mathrm{nc}} A \vee^{\mathrm{u}} B \\ B \to^{\mathrm{nc}} A \vee^{\mathrm{u}} B \end{cases}$$

Elimination axioms:

$$A \vee^{\mathrm{d}} B \to^{\mathrm{c}} (A \to^{\mathrm{c}} C) \to^{\mathrm{c}} (B \to^{\mathrm{c}} C) \to^{\mathrm{c}} C,$$
$$A \vee^{\mathrm{l}} B \to^{\mathrm{c}} (A \to^{\mathrm{c}} C) \to^{\mathrm{c}} (B \to^{\mathrm{nc}} C) \to^{\mathrm{c}} C,$$
$$A \vee^{\mathrm{r}} B \to^{\mathrm{c}} (A \to^{\mathrm{nc}} C) \to^{\mathrm{c}} (B \to^{\mathrm{c}} C) \to^{\mathrm{c}} C,$$
$$A \vee^{\mathrm{u}} B \to^{\mathrm{c}} (A \to^{\mathrm{nc}} C) \to^{\mathrm{c}} (B \to^{\mathrm{nc}} C) \to^{\mathrm{c}} C.$$

Formulas as computational problems

- Kolmogorov (1932) proposed to view a formula A as a computational problem, of type $\tau(A)$, the type of a potential solution or “realizer” of A.
- Example: $\forall^{\mathrm{c}}_{n}\exists_{m>n}\mathrm{Prime}(m)$ has type $\mathbf{N} \to \mathbf{N}$.
- $A \mapsto \tau(A)$, a type or the “nulltype” symbol $\circ$.
- In case $\tau(A) = \circ$ proofs of A have no computational content; such formulas A are called non-computational (n.c.) or Harrop formulas; the others computationally relevant (c.r.).

Decoration can simplify extracts

- Suppose that a proof M uses a lemma $L^{\mathrm{d}}\colon A \vee^{\mathrm{d}} B$.
- Then the extract $\mathrm{et}(M)$ will contain the extract $\mathrm{et}(L^{\mathrm{d}})$.
- Suppose that the only computationally relevant use of L^{d} in M was which one of the two alternatives holds true, A or B.
- Express this by using a weakened lemma $L\colon A \vee^{\mathrm{u}} B$.
- Since $\mathrm{et}(L)$ is a boolean, the extract of the modified proof is "purified": the (possibly large) extract $\mathrm{et}(L^{\mathrm{d}})$ has disappeared.

Decoration algorithm

Goal: Insert as few as possible decorations $\forall^{c}, \to^{c}$ into a proof.

- $\mathrm{Seq}(M)$ of a proof M consists of its context and end formula.
- The uniform proof pattern $\mathrm{P}(M)$ of a proof M is the result of changing in c.r. formulas of M (i.e., not above a n.c. formula) all $\to^{c}$, $\forall^{c}$ into $\to^{nc}$, $\forall^{nc}$ (some restrictions apply on axioms and theorems).
- A formula D extends C if D is obtained from C by changing some $\to^{nc}$, $\forall^{nc}$ into $\to^{c}$, $\forall^{c}$.
- A proof N extends M if (i) N and M are the same up to variants of $\to$, $\forall$ in their formulas, and (ii) every c.r. formula in M is extended by the corresponding one in N.

Decoration algorithm (ctd.)

- Assumption: For every axiom or theorem A and every decoration variant C of A we have another axiom or theorem whose formula D extends C, and D is the least among those extensions.
- Example: Induction

$$A'(0) \to^{c/nc} \forall_n^{c/nc}(A''(n) \to^{c/nc} A'''(n{+}1))) \to^{c/nc} \forall_n^{c/nc} A''''(n).$$

Let A be the lub (w.r.t. deco) of $A', \dots, A''''$. Extended axiom:

$$A(0) \to^{c} \forall_n^{c}(A(n) \to^{c} A(n+1))) \to^{c} \forall_n^{c} A(n).$$

Decoration algorithm (ctd.)

Theorem (Ratiu & S., 2010)

Under the assumption above, for every uniform proof pattern U and every extension of its sequent $\mathrm{Seq}(U)$ we can find a decoration M_∞ of U such that

(a) $\mathrm{Seq}(M_\infty)$ *extends the given extension of* $\mathrm{Seq}(U)$, *and*

(b) M_∞ *is optimal in the sense that any other decoration M of U whose sequent $\mathrm{Seq}(M)$ extends the given extension of $\mathrm{Seq}(U)$ has the property that M also extends M_∞.*

Case $(\to^{\mathrm{nc}})^-$. Consider a proof pattern

$$\dfrac{\begin{matrix}\Phi, \Gamma \\ \mid U \\ A \to^{\mathrm{nc}} B\end{matrix} \qquad \begin{matrix}\Gamma, \Psi \\ \mid V \\ A\end{matrix}}{B}\,(\to^{\mathrm{nc}})^-$$

Given: extension $\Pi, \Delta, \Sigma \Rightarrow D$ of $\Phi, \Gamma, \Psi \Rightarrow B$. Alternating steps:

- $\mathrm{IH}_a(U)$ for extension $\Pi, \Delta \Rightarrow A \to^{\mathrm{nc}} D \mapsto$ decoration M_1 of U whose sequent $\Pi_1, \Delta_1 \Rightarrow C_1 \to D_1$ extends $\Pi, \Delta \Rightarrow A \to^{\mathrm{nc}} D$ $(\to \in \{\to^{\mathrm{nc}}, \to^{\mathrm{c}}\})$. Suffices if A is n.c.: extension $\Delta_1, \Sigma \Rightarrow C_1$ of V is a proof (in n.c. parts of a proof $\to^{\mathrm{nc}}$, $\forall^{\mathrm{nc}}$ and $\to^{\mathrm{c}}$, $\forall^{\mathrm{c}}$ are identified). For A c.r:
- $\mathrm{IH}_a(V)$ for the extension $\Delta_1, \Sigma \Rightarrow C_1 \mapsto$ decoration N_2 of V whose sequent $\Delta_2, \Sigma_2 \Rightarrow C_2$ extends $\Delta_1, \Sigma \Rightarrow C_1$.
- $\mathrm{IH}_a(U)$ for $\Pi_1, \Delta_2 \Rightarrow C_2 \to D_1 \mapsto$ decoration M_3 of U whose sequent $\Pi_3, \Delta_3 \Rightarrow C_3 \to D_3$ extends $\Pi_1, \Delta_2 \Rightarrow C_2 \to D_1$.
- $\mathrm{IH}_a(V)$ for the extension $\Delta_3, \Sigma_2 \Rightarrow C_3 \mapsto$ decoration N_4 of V whose sequent $\Delta_4, \Sigma_4 \Rightarrow C_4$ extends $\Delta_3, \Sigma_2 \Rightarrow C_3$. ...

Example: Euler's φ, or avoiding factorization

Let $P(n)$ mean "n is prime". Consider

$$\begin{array}{lll} \texttt{Fact}\colon & \forall^{\mathrm{c}}_{n}(P(n) \vee^{\mathrm{r}} \exists_{m,k>1}(n = mk)) & \text{factorization,} \\ \texttt{PTest}\colon & \forall^{\mathrm{c}}_{n}(P(n) \vee^{\mathrm{u}} \exists_{m,k>1}(n = mk)) & \text{prime number test.} \end{array}$$

Euler's φ has the properties

$$\begin{cases} \varphi(n) = n - 1 & \text{if } P(n), \\ \varphi(n) < n - 1 & \text{if } n \text{ is composed.} \end{cases}$$

Using factorization and these properties we obtain a proof of

$$\forall^{\mathrm{c}}_{n}(\varphi(n) = n - 1 \vee^{\mathrm{u}} \varphi(n) < n - 1).$$

Goal: get rid of the expensive factorization algorithm in the computational content, via decoration.

Example: Euler's φ, or avoiding factorization (ctd.)

How could the better proof be found? Recall that we assumed

$$\texttt{Fact}\colon \forall^{\mathrm{c}}_n (P(n) \vee^{\mathrm{r}} \exists_{m,k>1}(n = mk)),$$
$$\texttt{PTest}\colon \forall^{\mathrm{c}}_n (P(n) \vee^{\mathrm{u}} \exists_{m,k>1}(n = mk))$$

and have a proof of $\forall^{\mathrm{c}}_n (\varphi(n) = n - 1 \vee^{\mathrm{u}} \varphi(n) < n - 1)$ from `Fact`.

- The decoration algorithm arrives at `Fact` with goal

$$P(n) \vee^{\mathrm{u}} \exists_{m,k>1}(n = mk).$$

- `PTest` fits as well, and it has $\vee^{\mathrm{u}}$ rather than $\vee^{\mathrm{r}}$, hence is preferred.

```
(define decnproof (fully-decorate nproof "Fact" "PTest"))
(proof-to-expr-with-formulas decnproof) =>
Elim: allnc n((C n -> F) oru C n ->
 ((C n -> F) -> phi n=n--1 oru phi n<n--1) ->
 (C n --> phi n=n--1 oru phi n<n--1) ->
 phi n=n--1 oru phi n<n--1)
PTest: all n((C n -> F) oru C n)
Intro: allnc n(phi n=n--1 -> phi n=n--1 oru phi n<n--1)
EulerPrime: allnc n((C n -> F) -> phi n=n--1)
Intro: allnc n(phi n<n--1 -> phi n=n--1 oru phi n<n--1)
EulerComp: allnc n(C n -> phi n<n--1)

(lambda (n)
  ((((Elim n) (PTest n))
     (lambda (u1542) ((Intro n) ((EulerPrime n) u1542))))
    (lambda (u1544) ((Intro n) ((EulerComp n) u1544)))))

(pp (nt (proof-to-extracted-term decnproof))) => cPTest
```

Example: Maximal Scoring Segment (MSS)

- Let X be linearly ordered by $\preceq$. Given $\mathrm{seg}\colon \mathbf{N} \to \mathbf{N} \to X$.
  Want: maximal segment

$$\forall^{\mathrm{c}}_{n} \exists_{i \le k \le n} \forall_{i' \le k' \le n} (\mathrm{seg}(i', k') \preceq \mathrm{seg}(i, k)).$$

- Example: Regions with high G, C content in DNA.

$$\begin{aligned}
&X := \{G, C, A, T\},\\
&g\colon \mathbf{N} \to X \quad \text{(gene)},\\
&f\colon \mathbf{N} \to \mathbf{Z}, \quad f(i) := \begin{cases} 1 & \text{if } g(i) \in \{G, C\},\\ -1 & \text{if } g(i) \in \{A, T\}, \end{cases}\\
&\mathrm{seg}(i, k) = f(i) + \cdots + f(k).
\end{aligned}$$

Example: MSS (ctd.)

Prove the existence of a maximal segment by induction on n, simultaneously with the existence of a maximal end segment.

$$\forall^{\mathrm{c}}_{n}(\exists_{i \le k \le n} \forall_{i' \le k' \le n}(\mathrm{seg}(i', k') \preceq \mathrm{seg}(i, k)) \land \exists_{j \le n} \forall_{j' \le n}(\mathrm{seg}(j', n) \preceq \mathrm{seg}(j, n)))$$

In the step:

- Compare the maximal segment i, k for n with the maximal end segment $j, n+1$ proved separately.
- If $\preceq$, take the new i, k to be $j, n+1$. Else take the old i, k.

Depending on how the existence of a maximal end segment was proved, we obtain a quadratic or a linear algorithm.

Example: MSS (ctd.)

Two proofs of the existence of a maximal end segment for $n+1$: $\forall^{\mathrm{c}}_n \exists_{j \le n+1} \forall_{j' \le n+1} (\mathrm{seg}(j', n+1) \preceq \mathrm{seg}(j, n+1))$.

- Introduce an auxiliary parameter m; prove by induction on m

$$\forall^{\mathrm{c}}_n \forall^{\mathrm{c}}_{m \le n+1} \exists_{j \le n+1} \forall_{j' \le m} (\mathrm{seg}(j', n+1) \preceq \mathrm{seg}(j, n+1)).$$

- Use $\mathtt{ES}_n \colon \exists_{j \le n} \forall_{j' \le n} (\mathrm{seg}(j', n) \preceq \mathrm{seg}(j, n))$ and the additional assumption of monotonicity

$$\forall_{i,j,n} (\mathrm{seg}(i, n) \preceq \mathrm{seg}(j, n) \to \mathrm{seg}(i, n+1) \preceq \mathrm{seg}(j, n+1)).$$

  Proceed by cases on $\mathrm{seg}(j, n+1) \preceq \mathrm{seg}(n+1, n+1)$.
  If $\preceq$, take $n+1$, else the previous j.

Example: MSS (ctd.)

Could decoration help to find the better proof? Have lemmas L:

$$\forall^{\mathrm{c}}_{n}\forall^{\mathrm{c}}_{m\leq n+1}\exists_{j\leq n+1}\forall_{j'\leq m}(\mathrm{seg}(j', n+1) \preceq \mathrm{seg}(j, n+1))$$

and LMon:

$$\mathtt{Mon} \to \forall^{\mathrm{c}}_{n}(\mathtt{ES}_n \to^{\mathrm{c}} \forall^{\mathrm{nc}}_{m\leq n+1}\exists_{j\leq n+1}\forall_{j'\leq m}(\mathrm{seg}(j', n+1) \preceq \mathrm{seg}(j, n+1))).$$

- The decoration algorithm arrives at L with goal

$$\forall^{\mathrm{nc}}_{m\leq n+1}\exists_{j\leq n+1}\forall_{j'\leq m}(\mathrm{seg}(j', n+1) \preceq \mathrm{seg}(j, n+1)).$$

- LMon fits as well, its assumptions Mon and $\mathtt{ES}_n$ are in the context, and it is less extended ($\forall^{\mathrm{nc}}_{m\leq n+1}$ rather than $\forall^{\mathrm{c}}_{m\leq n+1}$), hence is preferred.

References

- U. Berger, Uniform Heyting arithmetic. APAL 133 (2005).
- D. Ratiu and H.S., Decorating proofs. Proofs, Categories and Computations (S. Feferman and W. Sieg, eds.), 2010.
- H.S. and S.S. Wainer, Proofs and Computations. Perspectives in Mathematical Logic, ASL & Cambridge UP, 2012.