

art.1 Derivations in Natural Deduction

inc:art:pnd:
sec

In order to arithmetize **derivations**, we must represent **derivations** as numbers. Since **derivations** are trees of **formulas** where each inference carries one or two labels, a recursive representation is the most obvious approach: we represent a **derivation** as a tuple, the components of which are the end-**formula**, the labels, and the representations of the sub-**derivations** leading to the premises of the last inference.

explanation

Definition art.1. If δ is a **derivation** in natural deduction, then $\# \delta \#$ is

1. $\langle 0, \# \varphi \#, n \rangle$ if δ consists only of the assumption φ . The number n is 0 if it is an **undischarged** assumption, and the numerical label otherwise.
2. $\langle 1, \# \varphi \#, n, k, \# \delta_1 \# \rangle$ if δ ends in an inference with one premise, k is given by the following table according to which rule was used in the last inference, and δ_1 is the immediate subproof ending in the premise of the last inference. n is the label of the inference, or 0 if the inference does not **discharge** any assumptions.

Rule:	\wedge Elim	\vee Intro	\rightarrow Intro	\neg Intro	\perp_I
k:	1	2	3	4	5

Rule:	\perp_C	\forall Intro	\forall Elim	\exists Intro	$=$ Intro
k:	6	7	8	9	10

3. $\langle 2, \# \varphi \#, n, k, \# \delta_1 \#, \# \delta_2 \# \rangle$ if δ ends in an inference with two premises, k is given by the following table according to which rule was used in the last inference, and δ_1, δ_2 are the immediate sub**derivations** ending in the left and right premise of the last inference, respectively. n is the label of the inference, or 0 if the inference does not **discharge** any assumptions.

Rule:	\wedge Intro	\rightarrow Elim	\neg Elim
k:	1	2	3

4. $\langle 3, \# \varphi \#, n, \# \delta_1 \#, \# \delta_2 \#, \# \delta_3 \# \rangle$ if δ ends in an \vee Elim inference. $\delta_1, \delta_2, \delta_3$ are the immediate sub**derivations** ending in the left, middle, and right premise of the last inference, respectively, and n is the label of the inference.

Example art.2. Consider the very simple derivation

$$1 \frac{\frac{[(\varphi \wedge \psi)]^1}{\varphi} \wedge \text{Elim}}{(\varphi \rightarrow \psi)} \rightarrow \text{Intro}$$

The Gödel number of the assumption would be $d_0 = \langle 0, \#(\varphi \wedge \psi)\#, 1 \rangle$. The Gödel number of the derivation ending in the conclusion of \wedge Elim would be

$d_1 = \langle 1, \# \varphi^\#, 0, 1, d_0 \rangle$ (1 since \wedge Elim has one premise, Gödel number of conclusion φ , 0 because no assumption is discharged, 1 is the number coding \wedge Elim). The Gödel number of the entire derivation then is $\langle 1, \#(\varphi \rightarrow \psi)^\#, 1, 3, d_1 \rangle$, i.e.,

$$2^2 \cdot 3^{\#(\varphi \rightarrow \psi)^\# + 1} \cdot 5^2 \cdot 7^4 \cdot 11^{(2^2 \cdot 3^{\# \varphi^\# + 1} \cdot 5^1 \cdot 7^2 \cdot 11^{(2^1 \cdot 3^{\#(\varphi \wedge \psi)^\# + 1} \cdot 5^2)})}$$

explanation

Having settled on a representation of **derivations**, we must also show that we can manipulate such **derivations** primitive recursively, and express their essential properties and relations so. Some operations are simple: e.g., given a Gödel number d of a **derivation**, $(d)_1$ gives us the Gödel number of its end-**formula**. Some are much harder. We'll at least sketch how to do this. The goal is to show that the relation “ δ is a **derivation** of φ from Γ ” is primitive recursive on the Gödel numbers of δ and φ .

Proposition art.3. *The following relations are primitive recursive:*

inc:art:pnd:
prop:followsby

1. φ occurs as an assumption in δ with label n .
2. All assumption in δ with label n are of the form φ (i.e., we can **discharge** the assumption φ using label n in δ).
3. φ is an **undischarged** assumption of δ .
4. An inference with conclusion φ , upper **derivations** δ_1 (and δ_2, δ_3), and discharge label n is correct.
5. δ is a correct natural deduction **derivation**.

Proof. We have to show that the corresponding relations between Gödel numbers of **formulas**, sequences of Gödel numbers of **formulas** (which code sets of **formulas**), and Gödel numbers of **derivations** are primitive recursive.

1. We want to show that $\text{Assum}(x, d, n)$, which holds if x is the Gödel number of an assumption of the **derivation** with Gödel number d labelled n , is primitive recursive. For this we need a helper relation $\text{hAssum}(x, d, n, i)$ which holds if the **formula** φ with Gödel number x occurs as an initial **formula** with label n in the **derivation** with Gödel number d within i inferences up from the end-**formula**.

$$\begin{aligned} \text{hAssum}(x, d, n, 0) &\Leftrightarrow 1 \\ \text{hAssum}(x, d, n, i + 1) &\Leftrightarrow \\ &\text{Sent}(x) \wedge (d = \langle 0, x, n \rangle \vee \\ &((d)_0 = 1 \wedge \text{hAssum}(x, (d)_4, n, i)) \vee \\ &((d)_0 = 2 \wedge (\text{hAssum}(x, (d)_4, n, i) \vee \\ &\quad \text{hAssum}(x, (d)_5, n, i))) \vee \\ &((d)_0 = 3 \wedge (\text{hAssum}(x, (d)_3, n, i) \vee \\ &\quad \text{hAssum}(x, (d)_2, n, i)) \vee \text{hAssum}(x, (d)_3, n, i)) \end{aligned}$$

If the number i is large enough, e.g., larger than the maximum number of inferences between an initial formula and the end-formula of δ , it holds of x , d , n , and i iff φ is an initial formula in δ labelled n . The number d itself is larger than that maximum number of inferences. So we can define

$$\text{Assum}(x, d, n) = \text{hAssum}(x, d, n, d).$$

2. We want to show that $\text{Discharge}(x, d, n)$, which holds if all assumptions with label n in the derivation with Gödel number d all are the formula with Gödel number x . But this relation holds iff $(\forall y < d) (\text{Assum}(y, d, n) \rightarrow y = x)$.
3. An occurrence of an assumption is not open if it occurs with label n in a subderivation that ends in a rule with discharge label n . Define the helper relation $\text{hNotOpen}(x, d, n, i)$ as

$$\begin{aligned} \text{hNotOpen}(x, d, n, 0) &\Leftrightarrow 1 \\ \text{hNotOpen}(x, d, n, i + 1) &\Leftrightarrow \\ & (d)_2 = n \vee \\ & ((d)_0 = 1 \wedge \text{hNotOpen}(x, (d)_4, n, i)) \vee \\ & ((d)_0 = 2 \wedge \text{hNotOpen}(x, (d)_4, n, i) \wedge \\ & \quad \text{hNotOpen}(x, (d)_5, n, i)) \vee \\ & ((d)_0 = 3 \wedge \text{hNotOpen}(x, (d)_3, n, i) \wedge \\ & \quad \text{hNotOpen}(x, (d)_4, n, i) \wedge \text{hNotOpen}(x, (d)_5, n, i)) \end{aligned}$$

Note that all assumptions of the form φ labelled n are discharged in δ iff either the last inference of δ discharges them (i.e., the last inference has label n), or if it is discharged in all of the immediate subderivations.

A formula φ is an open assumption of δ iff it is an initial formula of δ (with label n) and is not discharged in δ (by a rule with label n). We can then define $\text{OpenAssum}(x, d)$ as

$$(\exists n < d) (\text{Assum}(x, d, n, d) \wedge \neg \text{hNotOpen}(x, d, n, d)).$$

4. Here we have to show that for each rule of inference R the relation $\text{FollowsBy}_R(x, d_1, n)$ which holds if x is the Gödel number of the conclusion and d_1 is the Gödel number of a derivation ending in the premise of a correct application of R with label n is primitive recursive, and similarly for rules with two or three premises.

The simplest case is that of the =Intro rule. Here there is no premise, i.e., $d_1 = 0$. However, φ must be of the form $t = t$, for a closed term t . Here, a primitive recursive definition is

$$(\exists t < x) (\text{CITerm}(t) \wedge x = (\# = (\# \frown t \frown \# , \# \frown t \frown \#) \#)) \wedge d_1 = 0).$$

For a more complicated example, $\text{FollowsBy}_{\rightarrow\text{Intro}}(x, d_1, n)$ holds iff φ is of the form $(\psi \rightarrow \chi)$, the end-formula of δ is χ , and any initial formula in δ labelled n is of the form ψ . We can express this primitive recursively by

$$\begin{aligned} & (\exists y < x) (\text{Sent}(y) \wedge \text{Discharge}(y, d_1) \wedge \\ & \quad (\exists z < x) (\text{Sent}(y) \wedge (d)_1 = z) \wedge \\ & \quad \quad x = (\#(\# \frown y \frown \# \rightarrow \# \frown z \frown \#)\#)) \end{aligned}$$

(Think of y as the Gödel number of ψ and z as that of χ .)

For another example, consider $\exists\text{Intro}$. Here, φ is the conclusion of a correct inference with one upper derivation iff there is a formula ψ , a closed term t and a variable x such that $\psi[t/x]$ is the end-formula of the upper derivation and $\exists x \psi$ is the conclusion φ , i.e., the formula with Gödel number x . So $\text{FollowsBy}_{\exists\text{Intro}}(x, d_1, n)$ holds iff

$$\begin{aligned} & \text{Sent}(x) \wedge (\exists y < x) (\exists v < x) (\exists t < d) (\text{Frm}(y) \wedge \text{Term}(t) \wedge \text{Var}(v) \wedge \\ & \quad \text{FreeFor}(y, t, v) \wedge \text{Subst}(y, t, v) = (d)_1 \wedge x = (\#\exists\# \frown v \frown z)) \end{aligned}$$

5. We first define a helper relation $\text{hDeriv}(d, i)$ which holds if d codes a correct derivation at least to i inferences up from the end sequent. $\text{hDeriv}(d, 0)$ holds always. Otherwise, $\text{hDeriv}(d, i + 1)$ iff either d just codes an assumption or d ends in a correct inference and the codes of the immediate sub-derivations satisfy $\text{hDeriv}(d', i)$.

$$\begin{aligned} & \text{hDeriv}(d, 0) \Leftrightarrow 1 \\ & \text{hDeriv}(d, i + 1) \Leftrightarrow \\ & \quad (\exists x < d) (\exists n < d) (\text{Sent}(x) \wedge d = \langle 0, x, n \rangle) \vee \\ & \quad ((d)_0 = 1 \wedge \\ & \quad \quad ((d)_3 = 1 \wedge \text{FollowsBy}_{\wedge\text{Elim}}((d)_1, (d)_4, (d)_2) \vee \\ & \quad \quad \quad \vdots \\ & \quad \quad ((d)_3 = 10 \wedge \text{FollowsBy}_{=\text{Intro}}((d)_1, (d)_4, (d)_2)) \wedge \\ & \quad \quad \quad \text{nDeriv}((d)_4, i)) \vee \\ & \quad ((d)_0 = 2 \wedge \\ & \quad \quad ((d)_3 = 1 \wedge \text{FollowsBy}_{\wedge\text{Intro}}((d)_1, (d)_4, (d)_5, (d)_2)) \vee \\ & \quad \quad \quad \vdots \\ & \quad \quad ((d)_3 = 3 \wedge \text{FollowsBy}_{\rightarrow\text{Elim}}((d)_1, (d)_4, (d)_5, (d)_2)) \wedge \\ & \quad \quad \quad \text{hDeriv}((d)_4, i) \wedge \text{hDeriv}((d)_5, i)) \vee \\ & \quad ((d)_0 = 3 \wedge \\ & \quad \quad \text{FollowsBy}_{\vee\text{Elim}}((d)_1, (d)_3, (d)_4, (d)_5, (d)_2) \wedge \\ & \quad \quad \text{hDeriv}((d)_3, i) \wedge \text{hDeriv}((d)_4, i) \wedge \text{hDeriv}((d)_5, i) \end{aligned}$$

This is a primitive recursive definition. Again we can define $\text{Deriv}(d)$ as $\text{hDeriv}(d, d)$.

□

Problem art.1. Define the following relations as in [Proposition art.3](#):

1. $\text{FollowsBy}_{\rightarrow\text{Elim}}(x, d_1, d_2, n)$,
2. $\text{FollowsBy}_{=\text{Elim}}(x, d_1, d_2, n)$,
3. $\text{FollowsBy}_{\vee\text{Elim}}(x, d_1, d_2, d_3, n)$,
4. $\text{FollowsBy}_{\vee\text{Intro}}(x, d_1, n)$.

For the last one, you will have to also show that you can test primitive recursively if the formula with Gödel number x and the **derivation** with Gödel number d satisfy the eigenvariable condition, i.e., the eigenvariable a of the $\vee\text{Intro}$ inference occurs neither in x nor in an open assumption of d .

Proposition art.4. *Suppose Γ is a primitive recursive set of **sentences**. Then the relation $\text{Prf}_\Gamma(x, y)$ expressing “ x is the code of a **derivation** δ of φ from **undischarged** assumptions in Γ and y is the Gödel number of φ ” is primitive recursive.*

Proof. Suppose “ $y \in \Gamma$ ” is given by the primitive recursive predicate $R_\Gamma(y)$. We have to show that $\text{Prf}_\Gamma(x, y)$ which holds iff y is the Gödel number of a sentence φ and x is the code of a natural deduction **derivation** with end **formula** φ and all **undischarged** assumptions in Γ is primitive recursive.

By the previous proposition, the property $\text{Deriv}(x)$ which holds iff x is the code of a correct derivation δ in natural deduction is primitive recursive. If x is such a code, then $(x)_1$ is the code of the end-**formula** of δ . Thus we can define $\text{Prf}_\Gamma(x, y)$ by

$$\begin{aligned} \text{Prf}_\Gamma(x, y) \Leftrightarrow & \text{Deriv}(x) \wedge (x)_1 = y \wedge \\ & (\forall z < x) (\text{OpenAssum}(z, x) \rightarrow R_\Gamma(z)) \end{aligned}$$

□

Photo Credits

Bibliography