Notre Dame Journal of Formal Logic
Volume V, Number 1, January 1964

DIRECT CONSISTENCY PROOF OF GENTZEN'S SYSTEM OF NATURAL DEDUCTION

ANDRÉS R. RAGGIO

Gentzen proves[1] the equivalence of his system of natural deduction with an axiomatization of propositional logic due to Hilbert. This last system is consistent, therefore Gentzen's system also. We avoid this roundabout way and give a direct proof that all provable formulas of Gentzen's system of natural deduction are tautologies. From this we can easily infer that the system is consistent.

In the natural deduction we start with assumptions and apply rules of inference. These are the following:

Simple transformation rules

$\dfrac{A \text{ et } B}{A}$ $\qquad$ $\dfrac{A \text{ et } B}{B}$ $\qquad$ "et" elimination

$\dfrac{A}{A \text{ vel } B}$ $\qquad$ $\dfrac{A}{B \text{ vel } A}$ $\qquad$ "vel" introduction

$\dfrac{\text{abs}}{A}$ $\qquad$ "abs" elimination

$\dfrac{\text{non non } A}{A}$ $\qquad$ "non non" elimination

Compound transformation rules

$\dfrac{A, B}{A \text{ et } B}$ $\qquad$ "et" introduction

$\dfrac{A, A \text{ seq } B}{B}$ $\qquad$ "seq" elimination

$\dfrac{A, \text{ non } A}{\text{abs}}$ $\qquad$ "abs" introduction

---

1) *Cf*. Gentzen, Untersuchungen über das logische Schliessen, *Mathematische Zeitschrift*, vol. 39, p. 417.

Received October 23, 1962

Assumption discharging rules

$$\frac{\begin{matrix}(A)\\ B\end{matrix}}{A \text{ seq } B}$$ "seq" introduction (discharge of A)

$$\frac{\begin{matrix}(A)\\ \text{abs}\end{matrix}}{\text{non } A}$$ "non" introduction (discharge of A)

$$\frac{A \text{ vel } B \quad \begin{matrix}(A)\\ C\end{matrix} \quad \begin{matrix}(B)\\ C\end{matrix}}{C}$$ "vel" elimination (discharge of A and B)

We build a tree $\mathfrak{B}$ indicating all the assumptions and all the deduction steps. If all the assumptions are discharged, $\mathfrak{B}$ is a deductive tree, i.e., a proof of its endformula. If the endformula of $\mathfrak{B}$ depends on some assumptions, $\mathfrak{B}$ is a derivation tree.

A branch of $\mathfrak{B}$ is a finite sequence of formulas[2] $A_0, A_1, A_2, \ldots A_p$, such that A_0 is an assumption and if A_n is a premiss of one of the rules of inference A_{n+1} is its conclusion. A formula C is above another formula D in a given branch if C precedes D in the corresponding sequence.

A formula C is above another formula D in $\mathfrak{B}$ if there is a branch in which C is above D.

A subtree $\mathfrak{B}^*$ of $\mathfrak{B}$ is the result of deleting all formulas of $\mathfrak{B}$ which are above one or more formulas of $\mathfrak{B}$.

The upper ends of a subtree $\mathfrak{B}^*$ are either original assumptions of $\mathfrak{B}$ or stumps.

$\mathfrak{B}$ and all its subtrees have the same endformula.

A subtree of a subtree of $\mathfrak{B}$ is a subtree of $\mathfrak{B}$ (transitivity of the subtree relation).

The depth of a formula A in a given branch is the number of formulas which are above A in the branch or zero in case the given branch does not contain A.

The depth of the formula A in the tree $\mathfrak{B}$ is the greatest depth of A in all branches of $\mathfrak{B}$.

A valuation of $\mathfrak{B}$(Val ($\mathfrak{B}$)) is a value-assignment to all its atomic formulas which gives to "*abs*" the value **F**.

An evaluation of a formula A of $\mathfrak{B}$ for a given Val($\mathfrak{B}$) (Ev(A)) is its evaluation based in the value-assignment corresponding to Val($\mathfrak{B}$).

The degree of a tree $\mathfrak{B}$ for a Val($\mathfrak{B}$) is the number of assumptions evaluated with **F**'s.

The degree of a subtree $\mathfrak{B}^*$ of $\mathfrak{B}$ for a Val($\mathfrak{B}$) is the number of its assumptions evaluated with **F**'s plus the number of assumptions of $\mathfrak{B}$ evaluated with **F**'s on which the stumps of $\mathfrak{B}^*$ depend.

We prove first some lemmas:

2) We should say instead of formulas, occurrences of formulas; but there is no danger of confusion.

Lemma I: *If the depth of the formula A in $\mathfrak{B}$ is $n + 1$ and A is the conclusion of a rule of inference, then there is at least one premiss of the rule of inference whose depth is n.*

We simply choose a branch of $\mathfrak{B}$ in which A has the greatest depth; the preceding formula in this branch must have the depth n and is one of the premisses of the corresponding rule of inference.

Lemma II: *If $\mathfrak{B}$ is a tree not using* "vel" *elimination and a* Val($\mathfrak{B}$) *evaluates all its assumptions with* **T***'s, then all formulas of $\mathfrak{B}$ are evaluated with* **T***'s.*

In this case "*abs*" elimination, "*abs*" introduction and "non" introduction cannot be used. These three rules require that one of its premisses should be evaluated with **F**, but the assumptions are all evaluated with **T**'s and the other rules of inference yield from premisses and assumptions evaluated with **T**'s conclusions evaluated with **T**'s. Applying this last remark and lemma I (if the depths of the premisses of a rule of inference are different) we can prove by a course of values induction that all formulas of $\mathfrak{B}$ are evaluated with **T**'s for the given Val($\mathfrak{B}$).

Lemma III: *If $\mathfrak{B}^*$ is a subtree of $\mathfrak{B}$ not using* "vel" *elimination, and for a given* Val($\mathfrak{B}$) *$\mathfrak{B}^*$ is of degree zero and all its stumps are evaluated with* **T***'s, then all formulas of $\mathfrak{B}^*$ are evaluated with* **T***'s for the given* Val($\mathfrak{B}$).

The only difference with lemma II is that the stumps of $\mathfrak{B}^*$ are not assumptions; but they are evaluated with **T**'s and depend in $\mathfrak{B}$ on assumptions evaluated with **T**'s. The discharge of this assumptions can only be performed by "seq" introductions and the corresponding conclusions must be evaluated with **T**'s.

Lemma IV: *If $\mathfrak{B}$ is a deductive tree with the endformula D not using* "vel" *elimination and a* Val($\mathfrak{B}$) *evaluates some of its assumptions with* **F***'s, then* Ev(D) = **T** *for the given* Val($\mathfrak{B}$).

Because $\mathfrak{B}$ is a deductive tree all its assumptions evaluated with **F**'s must be discharged. We choose one of the "seq" introductions or "non" introductions discharging an assumption evaluated with **F**. The corresponding conclusion, say C, is such that Ev(C) = **T**. We delete in $\mathfrak{B}$ all formulas standing above C and obtain a subtree $\mathfrak{B}'$ whose degree is less than that of $\mathfrak{B}$. Applying this method at most as many times as assumptions evaluated with **F**'s exist in $\mathfrak{B}$, we arrive at a subtree $\mathfrak{B}''$ of degree zero all whose stumps are evaluated with **T**'s. Then by lemma III we can infer that Ev(D) = **T**.

Theorem I: *If $\mathfrak{B}$ is a deductive tree with the endformula D not using* "vel" *elimination, then for all* Val($\mathfrak{B}$) Ev(D) = **T**.

Lemmas II and IV exhaust all possible cases: a Val($\mathfrak{B}$) either evaluates all assumptions of $\mathfrak{B}$ with **T**'s or at least one assumption is evaluated with **F**.

Lemma V: *If* $\mathfrak{B}$ *is a tree with the endformula* D *not using* "vel" *elimination and for a given* Val($\mathfrak{B}$) Ev(D) = **F**, *then* $\mathfrak{B}$ *is a derivation tree and* D *depends in* $\mathfrak{B}$ *on at least one assumption evaluated with* **F**.

The given Val($\mathfrak{B}$) must evaluate some assumptions with **F**'s; this follows from lemma II. By the contraposition of lemma IV we can infer that $\mathfrak{B}$ is a derivation tree. If D depends in $\mathfrak{B}$ only on assumptions evaluated with **T**'s we could discharge all of them by "seq" eliminations and the new endformula D' would be evaluated with **F**. But this is impossible according to theorem I. Thus there must be an assumption evaluated with **F**.

Theorem II: *If* $\mathfrak{B}$ *is a deductive tree with the endformula* D, *then for all* Val($\mathfrak{B}$) Ev(D) = **T**. *Therefore* D *is a tantology.*

In this case $\mathfrak{B}$ may contain "vel" eliminations. We choose a "vel" elimination

$$\frac{A \text{ vel } B \qquad \begin{array}{c}(A)\\ C+\end{array} \qquad \begin{array}{c}(B)\\ C-\end{array}}{C}$$

which is not preceded by another one in all branches going through C. Let us take a Val($\mathfrak{B}$). If Ev(C) = **T** we delete all formulas which are above C and obtain a subtree $\mathfrak{B}'$ with one "vel" elimination less than $\mathfrak{B}$ and a stump evaluated with **T**.

If Ev(C) = **F**, then from lemma V applied to the two trees whose endformulas are $C+$ and $C-$ we can infer that a) $C+$ or $C-$ depends on an assumption evaluated with **F** different from A or B or b) both $C+$ and $C-$ depend respectively only on A and B and these two formulas are evaluated with **F**'s. In the first case we choose one of the assumptions evaluated with **F** different from A or B on which $C+$ or $C-$ depends and delete in $\mathfrak{B}$ all formulas standing above the conclusion of the discharging rule by which this assumption is discharged. We obtain a subtree $\mathfrak{B}''$ with at least one "vel" elimination less than $\mathfrak{B}$ and with a stump evaluated with **T**.

In the second case both A and B are evaluated with **F**'s; thus A vel B must also be evaluated with **F**. Applying lemma V to the tree with the endformula A vel B we infer that C depends in $\mathfrak{B}$ on at least one assumption evaluated with **F**. We delete in $\mathfrak{B}$ all formulas standing above the conclusion of the discharging rule by which this assumption is discharged and obtain a subtree $\mathfrak{B}'''$ which has at least one "vel" elimination less than $\mathfrak{B}$ and a stump evaluated with **T**.

Applying this method at most as many times as they are "vel" eliminations in $\mathfrak{B}$, we obtain a subtree $\mathfrak{B}^*$ with no "vel" elimination and with stumps evaluated with **T**'s. Now we can use the same reasoning as in proving theorem I. We build a subtree of $\mathfrak{B}^*$ of degree zero with stumps evaluated with **T**'s.

But then by lemma III Ev(D) = **T**.

Extending the concepts of valuation and evaluation to cover predicate and individual variables and considering the rules of introduction and elimination of quantifiers, it can be also easily proved that the functional calculus is semantically consistent.

Universidad Nacional de Córdoba
Córdoba, Argentina