Proof Theory and Logical Complexity

Jean-Yves Girard

January 30, 2020

Contents

1 Preliminaries

1.1 Languages

A (first-order) language is defined as follows: **L** is built up from the following atmoic symbols:

1. for all integers n, **predicate letters** p_j^n (p_j^n is *n-ary*)
2. for all integers n, **function letters** f_j^n (f_j^n is *n-ary*). A *0-ary* function letter is a **constant**
3. the **connectives** $\wedge, \vee, \neg, \rightarrow$
4. variables x_j ($j \in \mathbb{N}$)
5. the quantifiers $\forall$ and $\exists$

We shall always assume that our languages are **denumerable**; this means that the set of function and predicate letters is denumerable.

The **terms** of **L** are inductively defined as follows:

1. a variable x_j is a term
2. if $t_1, \ldots, t_n$ are terms and f_j^n is an n-ary function letter, then $f_j^n t_1 \ldots t_n$ is a term
3. the only terms of **L** are given by (1) and (2)

The **formulas** of **L** are inductively defined as follows:

1. if $t_1, \ldots, t_n$ are terms and p_j^n is an n-ary function letter, then $p_j^n t_1 \ldots t_n$ is a formula (**atomic formula**)
2. if A and B are formulas, so are $\wedge AB, \vee AB, \rightarrow AB, \neg A$
3. if A is a formula and x_j is a variable, $\forall x_j A$ and $\exists x_j A$ are formulas
4. the only formulas of **L** are given by (1)-(3)

1.2 occurrences

1. in $\forall x(px \rightarrow pfx)$ there are
   - three occurrences of x
   - two occurrences of p
   - one occurrence of $\forall$
   - one occurrence of $\rightarrow$
   - one occurrence of f
2. in $A, A \rightarrow A, A \vdash A$ there are
   - three occurrences of A
   - one occurrence of $A \rightarrow A$
   - one occurrence of $\vdash$

3. in the proof

$$\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A, A \vdash A \wedge A}\wedge\mathrm{I} \qquad A \vdash A}{A, A, A \vdash (A \wedge A) \wedge A}\wedge\mathrm{I}$$

- the sequent $A \vdash A$ occurs three times
- $A, A \vdash A \wedge A$ occurs once
- $A, A, A \vdash (A \wedge A) \wedge A$ occurs once
- $\wedge I$ occurs twice

If one wants to distinguish between various occurrences of sequents and rules, one can add indices, say:

$$\dfrac{\dfrac{A \vdash^1 A \qquad A \vdash^2 A}{A, A \vdash^1 A \wedge A}\wedge^1\mathrm{I} \qquad A \vdash^3 A}{A, A, A \vdash^1 (A \wedge A) \wedge A}\wedge^2\mathrm{I}$$

1.3 free and bound variables

We shall use square brackets to denote all free occurrences of variables of one or several variables in an expression; if A is $A[x_1, \ldots, x_n]$, then $A[x_1, \ldots, x_n]$ denotes $A[t_1, \ldots, t_n / x_1, \ldots, x_n]$

A bound variable has no individuality

2 The Fall of Hilbert Program

2.1 Hilbert's Program

2.1.1 the formalist philosophy

For the **formalist**, mathematical activity is *mechanical*: a machine could as well form sequences of strings of symbols, according to fixed laws.

2.1.2 Hilbert's ontology

The idea of Hilbert was to use this formal aspect of mathematics (which has the consequence that a mathematical proof can be viewed as a mathematical object itself) in order to *prove* some general facts concerning mathematical activity. Hilbert's ontology of mathematics distinguished between:

1. *Real* (or *elementray*, *finitist*) objects which *do* exists
2. *Abstract* objects which do not actually exist

2.1.3 Hilbert's program

purity of methods

2.1.4 consistency proofs

2.1.5 the fall

2.2 Recursive Functions

Definition 2.1. A function is **recursive** iff it maps $\mathbb{N}^k$ into $\mathbb{N}$ ($k \geq 0$), and is obtained by means of the following schemes:
($R1$) $I_i^n(a_1, \ldots, a_n) = a_i$, $a_1 + a_2, a_1 \cdot a_2, \chi_<(a_1, a_2)$
($R2$) composition
($R3$) μ**-operator**

Church's Thesis: *every computable function is recursive.*

Theorem 2.2. *1. The set of recursive functions is denumerable*

2. the set of recursive functions cannot be enumerated by a recursive function

Proof. (2) means that if $(f_n)_{n\in\mathbb{N}}$ is an enumeration of all recursive functions, then the function $F(n,m) = f_n(m)$ is not recursive: one easily sees that the function $g(n) = F(n,m) + 1$ would otherwise be recursive, but if $g = f_k$, one would have $g(k) = g(k) - 1$ □

Theorem 2.3. *($R4$) Constant functions are recursive*

($R5$) Let F and G be recursive functions with respectively n and $n+2$ arguments; then one can define a recursive function H, with $n+1$ arguments and such that

$$H(a_1, \ldots, a_n, 0) = F(a_1, \ldots, a_n)$$
$$H(a_1, \ldots, a_n, k+1) = G(a_1, \ldots, a_n, k, H(a_1, \ldots, a_n, k))$$

Definition 2.4. F is **primitive recursive** if F can be obtained by means of $(R1), (R2), (R4), (R5)$

Theorem 2.5. *1. the set of primitive recursive functions is denumerable*

2. the set of unary primitive recursive functions can be enumerated by means of a recursive binary function, called the ***Ackermann function***

3. The Ackermann function is not primitive recursive

Definition 2.6. 1. A predicate P is **recursive** iff its characteristic function χ_p is recursive

2. A problem is **decidable** iff the predicate which represents the problem is recursive

Example 2.1. 1. Predicate calculus is undecidable: if one encodes formulas by integers, then the set of integers which are codes of theorems of predicate calculus is not recursive

2. The **word problem** is undecidable: take the free group G generated by a finite number of points, and let $g_1, \ldots, g_k$ be elements of this group; let H be the normal subgroup generated by $g_1, \ldots, g_k$; then the equivalence relation $st^{-1} \in H$ is undecidable for a suitable choice of G and $g_1, \ldots, g_k$

Definition 2.7. $\mathbf{L}_o$ is the language of arithmetic: constant: $\overline{o}$, one unary function letter S, two binary function letters $+$ and $\cdot$, and two binary predicate letters $=$ and $<$

1. $\sum$ is the smallest class formulas of $\mathbf{L}_0$ s.t.
   (a) atomic formulas and their negation belong to $\sum$
   (b) if $A, B \in \sum$, then $A \wedge B, A \vee B \in \sum$
   (c) if $A \in \Sigma$, x is not free in term t, then $\forall x < t\ A \in \sum$
   (d) if $A \in \sum$ and x is a variable, then $\exists x A \in \sum$
2. Δ with the following differences:
   (d) if $A \in \Delta$ and x is not free in term t, then $\exists x < t\ A \in \Delta$
   (e) if $A \in \Delta$, then $\neg A \in \Delta$
3. A formula is $\sum_n^0$ (resp. $\prod_n^0$) iff it can be written $Q_0 x_0 \ldots Q_{n-1} x_{n-1} A$ where A is $\prod$ and the quantifiers Q_i are alternating, and $Q_0 = \exists$ (resp. $Q_0 = \forall$). For instance, Fermat's last theorem for a given n is $\prod_n^0$:

$$\forall z \forall a < z \forall b < z \forall c < z (abc \neq \overline{o} \rightarrow a^n + b^n \neq c^n)$$

Proposition 2.8. *Any $\sum$ formula is equivalent to a $\sum_1^0$-formula*

Proof. If $A \in \sum$, form A^x by replacing all existential quantifiers $\exists z$ of A by bounded quantifiers $(\exists z < x : A^x) \in \Delta$. And A is equivalent to the $\sum_1^0$-formula $\exists x A^x$ □

Theorem 2.9. *The properties $F(x_1, \ldots, x_n) = y$ and $P(x_1, \ldots, x_n)$ when F is a partial recursive function and P and r.e. predicate, can be expressed by means of $\sum$ formulas*

Definition 2.10. 1. Given integers a_0, a_{n-1}, one defines $\langle a_0, \ldots, a_{n-1} \rangle = 2^{a_0+1} 3^{a_1+1} \ldots p_{n-1}^{a_{n-1}+1}$, $Seq(x)$ is the predicate: for some $x_0, \ldots, x_{n-1}$, $x = \langle x_0, \ldots, x_{n-1} \rangle$

2. **length**

$$lh(x) = \begin{cases} 0 & x \notin Seq \\ n & x = \langle x_0, \dots, x_{n-1} \rangle \end{cases}$$

3. **projection**

$$(x)_i = \begin{cases} a_i & i < lh(x) \\ 0 & i \geq lh(x) \end{cases}$$

4. **concatenation**
$\langle a_0, \dots, a_{n-1} \rangle * \langle b_0, \dots, b_{m-1} \rangle = \langle a_0, \dots, a_{n-1}, b_0, \dots, b_{m-1} \rangle$
5. $\langle a_0, \dots, a_{n-1} \rangle \restriction i = \langle a_0, \dots, a_{i-1} \rangle$ if $i < n$, otherwise $x \restriction i = 0$

Definition 2.11. 1. A **numeral** is a term of $\mathbf{L}_o$ which canonically represents an integer; $\overline{n}$ is the nth numeral; hence $\overline{0}$ is the constant of $\mathbf{L}_0$, and $\overline{n+1} = S\overline{n}$

2. One defines the following prim. rec. predicates:
   - $Term(a)$: a is a term
   - $Form(a)$: a is a formula
   - $Fr(a, b)$: b is the Gödel number of a variable occurring freely in the expression encoded by a
   - $Cl(a)$: a is the Gödel number of a closed expression $\mathbf{L}_0$
   - $Subst(a, b, c)$: c is the Gödel number of a term substitutable for the variable with Gödel b in the formula with Gödel number a
3. prim. rec. functions
   - $Num(a) = \ulcorner \overline{a} \urcorner$, the Gödel number of the ath numeral
   - $Sub(a, b, c)$ = the Gödel number of $A[t/x_n]$ if $a = \ulcorner A \urcorner, b = \ulcorner x_n \urcorner, c = \ulcorner t \urcorner$

Theorem 2.12. *1. There exists a prim. rec. function Val s.t. if a is the Gödel number of a closed term of $\mathbf{L}_0$, $Val(a)$ is the integer represented by this term; in particular, $Val(Num(a)) = a$*

2. There exists a prim. rec. predicate Tr s.t. if a is the Gödel number of a closed Δ-formula of $\mathbf{L}_0$, then $Tr(a)$ iff the formula is true

Theorem 2.13 (Kleene Normal Form Theorem)**.** *1. For each integer $n \geq 0$, one can define a prim. rec. predicate T_n with $n + 2$ arguments and a prim. rec. function U with the following property: if F is a partial recursive function of n arguments, then there is an integer e (an **index** of F) s.t. for all $x_1, \dots, x_n$*

$$F(x_1, \dots, x_n) \simeq U(\mu y T_n(e, x_1, \dots, x_n, y))$$

2. *If P is an r.e. predicate of n arguments, then there is an integer c (an **index** of P) s.t. for all $x_1, \dots, x_n$*

$$P(x_1, \dots, x_n) \leftrightarrow \exists y T_n(e, x_1, \dots, x_n, y)$$

Proof. 1. We represent $F(x_1) \simeq y$ by a formula $A[x_1, y]$ which is $\sum$; hence there is a Δ-formula $B[z, x_1, y]$ s.t. $F(x_1) \simeq y$ iff $\exists z B[z, x_1, y]$). If one defines $T_1(e, a_1, b)$ by

$$Tr(Sub(Sub(Sub(e, \ulcorner x_0 \urcorner, Num((b)_0)), \ulcorner x_1 \urcorner, Num(a_1)), \ulcorner x_2 \urcorner, Num((b)_1))))$$

and $U(b) = (b)_1$, one sees that the result holds with $e = \ulcorner B \urcorner$

2. $P(x)$ can be written "$F(x)$ is defined" for an appropriate F (if $P(x) \leftrightarrow G(x) \simeq 0$, let $F(x) \simeq \mu y(G(x) \simeq 0)$)

□

Corollary 2.14. 1. *A non-void subset of $\mathbb{N}$ is r.e. iff it is the range of a prim. rec. function*

2. *A set is recursive iff it and its complement are r.e.*
3. *A partial recursive function which is total is a recursive function*

Proof. 1. The range of a partial recursive function is always an r.e. set. Conversely, if $A \subset \mathbb{N}$ is defined by the index e, and $a_0 \in A$ define F by $F(\langle x, b \rangle)) = x$ if $T_1(e, x, b)$, $F(x) = a_0$ otherwise

□