Metalogicon (2007) XX, 2

Hyperincursive Proof Theory

Arturo Graziano Grappone

1. Summary

This paper supplies an automatic procedure to decide whether any formula of first order predicative calculus is an axiom or a theorem by using Dubois's hyperincursive algorithms. The given procedure is also useful to decide whether any formula is an axiom or a theorem in Robinson's formal number theory.

2. Hyperincursive Proof Theory

2.1 To Build Proofs Automatically: the Anticipatory Approach

Actually, a standard theorem proof develops itself from some premises (axioms or proven theorems) to the formula to prove. Gentzen's natural deduction is a classical example of this fact.[1] A limit of this approach is the non-automatism of proof building because many distinct consequences can be deduced from the same premises. The problem "given a formula determinate whether it is a theorem" has not an automatic solution in the first order predicative calculus too.

Now, consider the time development of a standard proof. We can put its premises as "past" and its conclusion as "future". Thus the "past" does not determine the "future" because, in general, many distinct consequences can be deduced from the same premises. Instead, consider the possibility to revert a proof from its conclusion to its premises as in Aristotle's approach.[2] If we use an axiom only and one-to-one inferences (one conclusion from

[1] See 1. and 2. in the references.

[2] See 3. and 4, in the references.

one premise) only, then we can decide if a given formula is an axiom or a theorem in a deterministic way by its univocal reverting to the one axiom: the "future" determines the "past" as Dubois has observed to happen in anticipatory systems.

Thus to build proofs automatically we use an anticipatory approach and, in particular, Dubois's concepts of incursivity and hyperincursivity. First their definitions according to D.M. Dubois' words are reported in the following, and then the general concept of hyperincursive procedure is discussed to decide whether a formula is an axiom or a theorem in a formal structure.

It is clear that we can always identify the procedures to decide if a formula is an axiom or a theorem and to build a proof that it is a theorem.

2.2 Definition of Incursion and Hyperincursion

To study the anticipatory systems D. M. Dubois has introduced the concepts of incursion and hyperincursion. He writes: "*The recursion consists of the computation of the future value of the variable vector* X(t+1) *at time* t+1 *from the values of these variables at present and/or past times,* t*,* t-1*,*t-2*, ... by a recursive function:* X(t+1) = *f*(X(t), X(t-1),…, p) *where* p *is a command parameter vector. So, the past always determines the future, the present being the separation line between the past and the future. ... Starting from cellular automata the concept of fractal machines was proposed in which composition rules where propagated along paths in the machine frame. The computation is based on what I called 'INclusive reCURSION', i. e. INCURSION ... An incursive relation is defined by:* X(t+1) = *f*(…, X(t+1), X(t), X(t-1), …, p) *which consists in the computation of the values of the vector* X(t+1) *at time* t+1 *from the values* X(t-i) *at time* t-i*,* i = 1,2,… *as a function of a command vector* p*. This incursive relation is not trivial because future values of the variable vector at time steps* t+1, t+2,… *must be known to compute them at the time step* t+1*. ... In a similar way to that in which we define hyper recursion when each recursive step generates multiple solutions, I define HYPERINCURSION. ... I have decided to do this for three reasons. First, in relativity theory space and time are considered* *as a four-vector where time plays a role similar to space. If time* t *is replaced by space* s *in the above definition of incursion, we obtain* $X(s+1) = f(\ldots, X(s+1), X(s), X(s-1), \ldots, p)$ *and nobody is astonished – a laplacean operator looks like this. Second, in control theory, the engineers control engineering systems by defining goals in the future to compute their present state, similarly to our human anticipative behaviour. ... Third, I wanted to try to do a generalisation of the recursive and sequential Turing machine in looking at space-time cellular automata where the order in which the computations are made is taken into account with an inclusive recursion*".[3]

2.3 Definition of Hyperincursive Building of a Proof

D.M. Dubois has given a concrete approach to build incursive and hyperincursive algorithms.[4]

An incursive algorithm is applied to a $n \times m$ cell matrix and it is developed in four stages:

- STAGE 1: Initialization of all the cells.
- STAGE 2: Saving of the $n \times m$ cell matrix and calculation of any cell in row i and column j from a given set of cells with row numbers $\leq i$ and column numbers $\leq j$.
- STAGE 3: Copy the last row on the first row and the last column on the first column
- STAGE 4: Testing if the obtained $n \times m$ cell matrix is identical to the saved $n \times m$ cell matrix: if yes than end, otherwise return to stage 2.

A hyperincursive algorithm is a branched chain (that can have got cycles too) of incursive algorithms on the same cell matrix whose succession is regulated by parameter or conditional instructions.

Thus, define *hyperincusive procedure* to decide whether a given formula is an axiom or a theorem in a formal structure as a

[3] See 5. and 6. in the references.

[4] See 6. in the references.

hyperincursive algorithm on a 1×1 cell matrix. Remember from paragraph 2.1 that we use only an axiom and only one-to-one inferences. The starting stage 1 is to put in the one cell the given formula. So, any incursive algorithm is a reverted one-to-one inference in a way that, for every stage 2, the formula in the cell is replaced, by the considered reverted one-to-one inference, with its immediate premise (stage 3) with repetition till changes happen (stage 4). The stage 4 of an algorithm is the stage 1 of its successive algorithm. The incursive algorithm succession is the structure of the hyperincursive algorithm of hyperincursive procedure. This one ends when no changes are more possible (final stage 4): then the given formula is a theorem if and only if the last obtained formula is the one axiom that we accept.

3. Useful Statements on First Order Predicative Calculus

Assume Mendelson's formal theory $\mathbb{K}$ for first order predicative calculus.[5]

3.1 Let "…" be generic $\mathbb{K}$ formula series empty too. Two occurrences of "…" denote necessarily the same thing only if are in corresponding positions in the same statement.

3.2 Let (), i.e. <u>no character</u> inside round brackets, be $(\mathcal{A} \vee \sim\mathcal{A})$ and accept () as one axiom for our hyperincursive proofs.[6] Thus 4.8.1.

3.3 Let $\mathcal{A}\mathcal{B}\mathcal{C}\ldots$ be $\mathcal{A}\wedge\mathcal{B}\wedge\mathcal{C}\wedge\ldots$.

3.4 Let $[\mathcal{A}\mathcal{B}\mathcal{C}\ldots]$ be $\sim(\mathcal{A}\wedge\mathcal{B}\wedge\mathcal{C}\wedge\ldots)$. Thus [] becomes [()], i.e., $[\mathcal{A}\vee\sim\mathcal{A}]$, i.e., $\sim(\mathcal{A}\vee\sim\mathcal{A})$, i.e., $(\mathcal{A}\wedge\sim\mathcal{A})$.

3.5 $[\mathcal{A}] \mapsto \sim\mathcal{A}$ is valid by 3.3. Thus 4.2.1.

3.6 $\mathcal{A}\mathcal{B} \mapsto \mathcal{A}\wedge\mathcal{B}$ is valid by 3.2. Thus 4.2.2.

---

[5] See 7. in the references.

[6] Consider the bottom of the first column (premise column) of a proof in Gentzen's natural deduction.

3.7 $[[\mathcal{A}][\mathcal{B}]] \mapsto \sim(\sim\mathcal{A}\wedge\sim\mathcal{B}) \mapsto \mathcal{A}\vee\mathcal{B}$. Thus 4.2.3.

3.8 $[\mathcal{A}[\mathcal{B}]] \mapsto \sim(\mathcal{A}\wedge\sim\mathcal{B}) \mapsto \mathcal{A}\supset\mathcal{B}$. Thus 4.2.4.

3.9 $[\mathcal{B}[\mathcal{C}]][[\mathcal{B}]\mathcal{C}] \mapsto \sim(\mathcal{A}\wedge\sim\mathcal{B})\ \sim(\sim\mathcal{A}\wedge\mathcal{B}) \mapsto \mathcal{A}\equiv\mathcal{B}$. Thus 4.2.5.

3.10 $(\ldots) \mapsto [[\ldots]]$ because $\mathcal{A} \mapsto \sim\sim\mathcal{A}$.

3.11 $[\ldots[\mathcal{A}]\ldots][\ldots[\mathcal{B}]\ldots][\ldots[\mathcal{C}]\ldots]\ldots \mapsto [\ldots[\mathcal{A}\mathcal{B}\mathcal{C}\ldots]\ldots]$ because $\sim(\ldots\wedge\sim\mathcal{A}\wedge\ldots)\wedge\sim(\ldots\wedge\sim\mathcal{B}\wedge\ldots)\wedge\sim(\ldots\wedge\sim\mathcal{C}\wedge\ldots)\wedge\ldots$ $\mapsto \sim(\ldots\wedge\sim(\mathcal{A}\wedge\mathcal{B}\wedge\mathcal{C}\wedge\ldots)\wedge\ldots)$. Thus 4.3.1 from 3.10 and 3.11.

3.12 $[\ldots\mathcal{A}\ldots[\ldots\ldots]\ldots] \mapsto [\ldots\mathcal{A}\ldots[\ldots\mathcal{A}\ldots]\ldots]$ because $\sim(\ldots\wedge\mathcal{A}\wedge\ldots\wedge\sim(\ldots\wedge\ldots)\wedge\ldots) \mapsto \sim(\ldots\wedge\mathcal{A}\wedge\ldots\wedge\sim(\ldots\wedge\mathcal{A}\wedge\ldots)\wedge\ldots)$.

3.13 $[\ldots\ [\ldots\ldots]\ldots\mathcal{A}\ldots] \mapsto [\ldots\ [\ldots\mathcal{A}\ldots]\ldots\mathcal{A}\ldots]$ because $\sim(\ldots\wedge\sim(\ldots\wedge\ldots)\wedge\ldots\wedge\mathcal{A}\wedge\ldots) \mapsto \sim(\ldots\wedge\sim(\ldots\wedge\mathcal{A}\wedge\ldots)\wedge\ldots\wedge\mathcal{A}\wedge\ldots)$. Thus 4.3.2 from 3.12 and 3.13.

3.14 $[] \mapsto (\ldots[]\ldots)$ because $\mathcal{A}\wedge\sim\mathcal{A} \mapsto (\ldots\wedge\mathcal{A}\wedge\sim\mathcal{A}\wedge\ldots)$. Thus 4.3.3.

3.15 $(x_i)\mathcal{A}(x_i) \mapsto \mathcal{A}(\tau)$ with τ free term for x_i in $\mathcal{A}(x_i)$ because $(x_i)\mathcal{A}(x_i)\supset\mathcal{A}(\tau)$ with τ free term for x_i in $\mathcal{A}(x_i)$. Thus 4.4.1.

3.16 Given a formula $\mathcal{A}$ where all the connectives are represented by 3.3 and 3.4, let us observe that any quantifier universal (x_i) (particular (Ex_i)) keeps universal (particular) in the prenexe normal form of $\mathcal{A}$ if and only if the same quantifier is nested in an even number[7] of [...], otherwise it becomes particular (universal).

3.17 $((x_j)(x_k)(x_l)\ldots\mathcal{A}(\ldots x_j\ldots x_k\ldots x_l\ldots)) \mapsto (x_j)\mathcal{A}(\ldots x_j\ldots x_j\ldots x_j\ldots)$. Thus 4.5.1 and 4.5.4 from 3.16.

3.18 $(((Ex_j)(x_k)(x_l)\ldots\mathcal{A}(\ldots x_j\ldots x_k\ldots x_l\ldots))\vee((Ex_k)(x_j)(x_l)\ldots\mathcal{A}(\ldots x_j\ldots x_k\ldots x_l\ldots))\vee((Ex_l)(x_k)(x_j)\ldots\mathcal{A}(\ldots x_j\ldots x_k\ldots x_l\ldots))\vee\ldots).\mapsto.(Ex_j)\mathcal{A}(\ldots x_j\ldots x_j\ldots x_j\ldots)$. Thus 4.5.2 and 4.5.3 from 3.16.

---

[7] Consider zero *even* once for all.

3.19 $(x_i)(\mathcal{A}\wedge\mathcal{B}(x_i)\wedge\mathcal{C})$ with $\mathcal{A}$ and $\mathcal{C}$ without free occurrences of x_i is equivalent to $(\mathcal{A}\wedge(x_i)\mathcal{B}(x_i)\wedge\mathcal{C})$. Thus 4.6.1.

3.20 $(x_i)\sim(\mathcal{A}\wedge\mathcal{B}(x_i)\wedge\mathcal{C})$ with $\mathcal{A}$ and $\mathcal{C}$ without free occurrences of x_i is equivalent to $\sim(\mathcal{A}\wedge(Ex_i)\mathcal{B}(x_i)\wedge\mathcal{C})$. Thus 4.6.2.

3.21 $(Ex_i)(\mathcal{A}\wedge\mathcal{B}(x_i)\wedge\mathcal{C})$ with $\mathcal{A}$ and $\mathcal{C}$ without free occurrences of x_i is equivalent to $(\mathcal{A}\wedge(Ex_i)\mathcal{B}(x_i)\wedge\mathcal{C})$. Thus 4.6.3.

3.22 $(Ex_i)\sim(\mathcal{A}\wedge\mathcal{B}(x_i)\wedge\mathcal{C})$ with $\mathcal{A}$ and $\mathcal{C}$ without free occurrences of x_i is equivalent to $\sim(\mathcal{A}\wedge(x_i)\mathcal{B}(x_i)\wedge\mathcal{C})$. Thus 4.6.4.

3.23 The free exchange of linked variables justify 4.6.6.

3.24 The hierarchy of the distinct closure by quantifiers of a sentence $\mathcal{A}(x_1\ldots x_n)$ can be build only by two laws:[8]

- *First quantifier particularization law*: $(x_i)\mathcal{A}\supset(Ex_i)\mathcal{A}$
- *Regression law*: $(\ldots(Ex_i)(x_j)\ldots\mathcal{A})\supset(\ldots(x_j)(Ex_i)\ldots\mathcal{A})$

Thus 4.7.1.

4. Hyperincursive Procedure to Decide a $\mathbb{K}$ Wff

4.1 Starting Stage 1

4.1.1 Let $\mathcal{A}$ be the $\mathbb{K}$ wff to decide.

4.2 Translation of Connectives in Terms of […]

4.2.1 If $\mathcal{A}$=$\mathcal{A}(\sim\mathcal{B})$ then put $\mathcal{A}$ =$\mathcal{A}([\mathcal{B}])$ till changes happen.

4.2.2 If $\mathcal{A}$=$\mathcal{A}(\mathcal{B}\wedge\mathcal{C})$ then put $\mathcal{A}$ =$\mathcal{A}(\mathcal{B}\mathcal{C})$ till changes happen.

4.2.3 If $\mathcal{A}$=$\mathcal{A}(\mathcal{B}\vee\mathcal{C})$ then put $\mathcal{A}$=$\mathcal{A}([[\mathcal{B}][\mathcal{C}]])$ till changes happen.

4.2.4 If $\mathcal{A}$=$\mathcal{A}(\mathcal{B}\supset\mathcal{C})$ then put $\mathcal{A}$ =$\mathcal{A}([\mathcal{B}[\mathcal{C}]])$ till changes happen.

---

[8] M. Malatesta found this meta-theorem in so general form during a stage in hospital after an infarct in 1996.

4.2.5 If $\mathcal{A}=\mathcal{A}(\mathcal{B}\equiv\mathcal{C})$ then put $\mathcal{A}=\mathcal{A}([\mathcal{B}[\mathcal{C}]][[\mathcal{B}]\mathcal{C}])$ till changes happen.

4.3 Simplification of [...]

4.3.1 If $\mathcal{A}=\mathcal{A}([\ldots[\ldots[\mathcal{B}]\ldots]\ldots])$ and eventual free term variables of $[\ldots[\mathcal{B}]\ldots]$ and $[\ldots[\ldots[\mathcal{B}]\ldots]\ldots]$ are linked by the same quantifiers then put $\mathcal{A}=\mathcal{A}([\ldots[\ldots]\ldots][\ldots\mathcal{B}\ldots][\ldots[\ldots]\ldots])$ till changes happen.[9]

4.3.2 $\mathcal{A}=\mathcal{A}([\ldots\mathcal{B}\ldots[\ldots\mathcal{B}\ldots]\ldots])$ ($\mathcal{A}=\mathcal{A}([\ldots[\ldots\mathcal{B}\ldots]\ldots\mathcal{B}\ldots])$ respectively) and eventual free term variables of the two occurrences of $\mathcal{B}$ are linked by the same quantifiers then put $\mathcal{A}=\mathcal{A}([\ldots[\ldots\ldots]\ldots\mathcal{B}\ldots])$ ($\mathcal{A}=\mathcal{A}([\ldots[\ldots\ldots]\ldots\mathcal{B}\ldots])$ respectively) till changes happen.

4.3.3 If $\mathcal{A}=\mathcal{A}(\ldots[]\ldots)$ then put $\mathcal{A}=\mathcal{A}([])$ till changes happen.

4.3.4 Return to 4.3.1 till changes happen

4.4 Elimination of Individual Constants and Functional Letters

4.4.1 If $\mathcal{A}=\mathcal{A}(\mathcal{B}(\tau))$ with τ free term for x_i in $\mathcal{B}(x_i)$, individual constant or term function of variables only then put $\mathcal{A}=(x_i)(\mathcal{A}(\mathcal{B}(x_i)))$ till changes happen.

4.5 Elimination of Quantifiers that Links More Scope Variables

4.5.1 If $\mathcal{A}=\mathcal{A}((x_i)\mathcal{B}(\ldots x_i\ldots x_i\ldots x_i\ldots))$ with $(x_i)\mathcal{B}(\ldots x_i\ldots x_i\ldots x_i\ldots)$ nested in even [...] and x_j, x_k, x_l, ... not occurring in it, then put $\mathcal{A}=\mathcal{A}((x_j)(x_k)(x_l)\ldots\mathcal{B}(\ldots x_j\ldots x_k\ldots x_l\ldots))$ till changes happen.

[9] Consider the particular case: $\mathcal{A}=\mathcal{A}((\ldots(\mathcal{B}\ldots)\ldots))$ from $\mathcal{A}=\mathcal{A}((\ldots[[\mathcal{B}\ldots]]\ldots))$.

4.5.2 If $\mathcal{A}=\mathcal{A}((x_i)\mathcal{B}(\ldots x_i\ldots x_i\ldots x_i\ldots))$ with $(x_i)\mathcal{B}(\ldots x_i\ldots x_i\ldots x_i\ldots)$ nested in odd $[\ldots]$ and x_j, x_k, x_l, … not occurring in it, then put. $\mathcal{A}=\mathcal{A}([[(Ex_j)(x_k)(x_l)\ldots\mathcal{B}(\ldots x_j\ldots x_k\ldots x_l\ldots)][(Ex_k)(x_j)(x_l)\ldots \mathcal{B}(\ldots x_j\ldots x_k\ldots x_l\ldots)][(Ex_l)(x_k)(x_j)\ldots\mathcal{B}(\ldots x_j\ldots x_k\ldots x_l\ldots)]\ldots]])$ till changes happen.

4.5.3 If $\mathcal{A}=\mathcal{A}((Ex_i)\mathcal{B}(\ldots x_i\ldots x_i\ldots x_i\ldots))$ with $(Ex_i)\mathcal{B}(\ldots x_i\ldots x_i\ldots x_i\ldots)$ nested in even $[\ldots]$ and x_j, x_k, x_l, … not occurring in it, then put. $\mathcal{A}=\mathcal{A}([[(Ex_j)(x_k)(x_l)\ldots\mathcal{B}(\ldots x_j\ldots x_k\ldots x_l\ldots)][(Ex_k)(x_j)(x_l)\ldots \mathcal{B}(\ldots x_j\ldots x_k\ldots x_l\ldots)][(Ex_l)(x_k)(x_j)\ldots\mathcal{B}(\ldots x_j\ldots x_k\ldots x_l\ldots)]\ldots]])$ till changes happen.

4.5.4 If $\mathcal{A}=\mathcal{A}((Ex_i)\mathcal{B}(\ldots x_i\ldots x_i\ldots x_i\ldots))$ with $(Ex_i)\mathcal{B}(\ldots x_i\ldots x_i\ldots x_i\ldots)$ nested in odd $[\ldots]$ and x_j, x_k, x_l, … not occurring in it, then put $\mathcal{A}=\mathcal{A}((x_j)(x_k)(x_l)\ldots\mathcal{B}(\ldots x_j\ldots x_k\ldots x_l\ldots))$ till changes happen.

4.6 Quantifier Regression to Its Linked Atomic Sentence

4.6.1 If $\mathcal{A}=\mathcal{A}((x_i)(\ldots\mathcal{B}(x_i)\ldots))$ then put $\mathcal{A}=\mathcal{A}((\ldots(x_i)\mathcal{B}(x_i)\ldots))$ till changes happen.

4.6.2 If $\mathcal{A}=\mathcal{A}((x_i)[\ldots\mathcal{B}(x_i)\ldots])$ then put $\mathcal{A}=\mathcal{A}([\ldots(Ex_i)\mathcal{B}(x_i)\ldots])$ till changes happen.

4.6.3 If $\mathcal{A}=\mathcal{A}((Ex_i)(\ldots\mathcal{B}(x_i)\ldots))$ then put $\mathcal{A}=\mathcal{A}((\ldots(Ex_i)\mathcal{B}(x_i)\ldots))$ till changes happen.

4.6.4 If $\mathcal{A}=\mathcal{A}((Ex_i)[\ldots\mathcal{B}(x_i)\ldots])$ then put $\mathcal{A}=\mathcal{A}([\ldots(x_i)\mathcal{B}(x_i)\ldots])$ till changes happen.

4.6.5 Return to 4.6.1 till changes happen.

4.6.6 If $\mathcal{A}=\mathcal{A}(\mathcal{B}(x_ix_jx_k\ldots))$ with $\mathcal{B}(x_ix_jx_k\ldots)$ close then put $\mathcal{A}=\mathcal{A}(\mathcal{B}(x_1x_2x_3\ldots))$ and replace every non-quantified $\mathcal{C}(x_1x_2x_3\ldots)$ in $\mathcal{B}(x_1x_2x_3\ldots)$ with $\mathcal{C}$ till changes happen.

4.6.7 Return to 4.3.1 till changes happen.

4.7 Quantifier Equalization

4.7.1 If $\mathcal{A}$=$\mathcal{A}$($\mathcal{B}$…[…$\mathcal{C}$…]) ($\mathcal{A}$=$\mathcal{A}$([…$\mathcal{C}$…]…$\mathcal{B}$) respectively) and $\mathcal{C}$ is nested in even […] and it can be obtained from $\mathcal{B}$ by application to it of a sequence of these operations:

- $\Omega_1((Ex)\mathcal{D})=(x)\mathcal{D}$
- $\Omega_2((x)(y)\mathcal{D})=(y)(x)\mathcal{D}$
- $\Omega_3((x)(Ey)\mathcal{D})=(Ey)(x)\mathcal{D}$
- $\Omega_4((Ex)(Ey)\mathcal{D})=(Ey)(Ex)\mathcal{D}$

then put. $\mathcal{A}$=$\mathcal{A}$($\mathcal{B}$…[…$\mathcal{B}$…]) ($\mathcal{A}$=$\mathcal{A}$([…$\mathcal{B}$…]…$\mathcal{B}$) respectively).

4.7.2 Return to 4.7.1 till changes happen.

4.7.3 Return to 4.3.1 till changes happen.

4.8 Decision if A_1 is a $\mathbb{K}$ theorem

4.8.1 $\mathcal{A}$ is a $\mathbb{K}$ axiom or theorem if and only if $\mathcal{A}$ is ().

5. Practical Examples of Automatic Decisions in $\mathbb{K}$

It is clear that we are interested more in the "automatism" of the decisions than in their "elegancy". This approach can produce very onerous calculations for relatively simple formulas. But a computer solves this problem easily if the automatism of decision building is completely warranted.

5.1	$(x)(\mathcal{A}\supset\mathcal{B}(x))\supset(\mathcal{A}\supset(x)\mathcal{B}(x))$ with x not free in $\mathcal{A}$.	
	$[(x)[\mathcal{A}[\mathcal{B}(x)]][[\mathcal{A}[(x)\mathcal{B}(x)]]]]$ with x not free in $\mathcal{A}$.	4.2.4
	$[(x)[\mathcal{A}[\mathcal{B}(x)]]\mathcal{A}[(x)\mathcal{B}(x)]]$ with x not free in $\mathcal{A}$.	4.3.1
	$[(x)[[\mathcal{B}(x)]]\mathcal{A}[(x)\mathcal{B}(x)]]$ with x not free in $\mathcal{A}$.	4.3.2
	$[(x)\mathcal{B}(x)\mathcal{A}[(x)\mathcal{B}(x)]]$ with x not free in $\mathcal{A}$.	4.3.1
	$[(x)\mathcal{B}(x)\mathcal{A}[]]$ with x not free in $\mathcal{A}$.	4.3.2
	[[]]	4.3.3
	()	4.3.1
	The given formula is a $\mathbb{K}$ axiom or theorem	4.8.1

5.2 $(x)\mathcal{A}(x)\supset\mathcal{A}(\tau)$ with τ term free for x in $\mathcal{A}(x)$.

$[(x)\mathcal{A}(x)[\mathcal{A}(\tau)]]$ with τ term free for x in $\mathcal{A}(x)$. 4.2.4

$(x)[(x)\mathcal{A}(x)[\mathcal{A}(x)]]$ 4.4.1

$[(x)\mathcal{A}(x)(Ex)[\mathcal{A}(x)]]$ 4.6.2

$[(x)\mathcal{A}(x)[(x)\mathcal{A}(x)]]$ 4.6.4

$[(x_1)\mathcal{A}[(x_1)\mathcal{A}]]$ 4.6.6

$[(x_1)\mathcal{A}[]]$ 4.3.2

$[[]]$ 4.3.3

$()$ 4.3.1

The given formula is a $\mathbb{K}$ axiom or theorem 4.8.1

5.3 $(x)(Ey)\mathcal{A}(xy)\supset(Ey)(x)\mathcal{A}(xy)$

$[(x)(Ey)\mathcal{A}(xy)[(Ey)(x)\mathcal{A}(xy)]]$ 4.2.4

$[(x_1)(Ex_2)\mathcal{A}[(Ex_2)(x_1)\mathcal{A}]]$ 4.6.6

The given formula is not a $\mathbb{K}$ axiom or theorem 4.8.1

5.4 $((x)\mathcal{A}(x)\supset(x)\mathcal{B}(x))\supset(x)(\mathcal{A}(x)\supset\mathcal{B}(x))$

$[[(x)\mathcal{A}(x)[(x)\mathcal{B}(x)]][(x)[\mathcal{A}(x)[\mathcal{B}(x)]]]]$ 4.2.4

$[[(x)\mathcal{A}(x)][(x)[\mathcal{A}(x)[\mathcal{B}(x)]]]][(x)\mathcal{B}(x)[(x)[\mathcal{A}(x)[\mathcal{B}(x)]]]]$ 4.3.1

$[[(x)\mathcal{A}(x)][(x)(y)[\mathcal{A}(x)[\mathcal{B}(y)]]]][(x)\mathcal{B}(x)[(x)(y)[\mathcal{A}(x)[\mathcal{B}(y)]]]]$ 4.5.1

$[[(x)\mathcal{A}(x)][[(Ex)\mathcal{A}(x)(Ey)[\mathcal{B}(y)]]]][(x)\mathcal{B}(x)[[(Ex)\mathcal{A}(x)(Ey)[\mathcal{B}(y)]]]]$ 4.6.2

$[[(x)\mathcal{A}(x)][[(Ex)\mathcal{A}(x)[(y)\mathcal{B}(y)]]]][(x)\mathcal{B}(x)[[(Ex)\mathcal{A}(x)[(y)\mathcal{B}(y)]]]]$ 4.6.4

$[[(x_1)\mathcal{A}][[(Ex_1)\mathcal{A}[(x_1)\mathcal{B}]]]][(x_1)\mathcal{B}[[(Ex_1)\mathcal{A}[(x_1)\mathcal{B}]]]]$ 4.6.6

$[[(x_1)\mathcal{A}](Ex_1)\mathcal{A}[(x_1)\mathcal{B}]][(x_1)\mathcal{B}(Ex_1)\mathcal{A}[(x_1)\mathcal{B}]]$ 4.3.1

$[[(x_1)\mathcal{A}](Ex_1)\mathcal{A}[(x_1)\mathcal{B}]][(x_1)\mathcal{B}(Ex_1)\mathcal{A}[]]$ 4.3.2

$[[(x_1)\mathcal{A}](Ex_1)\mathcal{A}[(x_1)\mathcal{B}]][[]]$ 4.3.3

$[[(x_1)\mathcal{A}](Ex_1)\mathcal{A}[(x_1)\mathcal{B}]]$ 4.3.2

The given formula is not a $\mathbb{K}$ axiom or theorem 4.8.1

5.5	$(x)(\mathcal{A}(x)\lor\mathcal{B}(x))\supset((x)\mathcal{A}(x)\lor(x)\mathcal{B}(x))$	
	$(x)[[\mathcal{A}(x)][\mathcal{B}(x)]]\supset[[(x)\mathcal{A}(x)][(x)\mathcal{B}(x)]]$	4.2.3
	$[(x)[[\mathcal{A}(x)][\mathcal{B}(x)]][[[(x)\mathcal{A}(x)][(x)\mathcal{B}(x)]]]]$	4.2.4
	$[(x)[[\mathcal{A}(x)][\mathcal{B}(x)]][(x)\mathcal{A}(x)][(x)\mathcal{B}(x)]]$	4.3.1
	$[[[(Ex)(y)[[\mathcal{A}(x)][\mathcal{B}(y)]][(x)\mathcal{A}(x)][(x)\mathcal{B}(x)]]][[(Ey)(x)[[\mathcal{A}(x)][\mathcal{B}(y)]][(x)\mathcal{A}(x)][(x)\mathcal{B}(x)]]]]$	4.5.2
	$[[[(Ex)[[\mathcal{A}(x)](Ey)[\mathcal{B}(y)]][(x)\mathcal{A}(x)][(x)\mathcal{B}(x)]]][[(Ey)[(Ex)[\mathcal{A}(x)][\mathcal{B}(y)]][(x)\mathcal{A}(x)][(x)\mathcal{B}(x)]]]]$	4.6.2
	$[[[[(x)[\mathcal{A}(x)][(y)\mathcal{B}(y)]][(x)\mathcal{A}(x)][(x)\mathcal{B}(x)]]][[[[(x)\mathcal{A}(x)](y)[\mathcal{B}(y)]][(x)\mathcal{A}(x)][(x)\mathcal{B}(x)]]]]$	4.6.4
	$[[[[[(Ex)\mathcal{A}(x)][(y)\mathcal{B}(y)]][(x)\mathcal{A}(x)][(x)\mathcal{B}(x)]]][[[[(x)\mathcal{A}(x)][(Ey)\mathcal{B}(y)]][(x)\mathcal{A}(x)][(x)\mathcal{B}(x)]]]]$	4.6.2
	$[[[[[(Ex_1)\mathcal{A}][(x_1)\mathcal{B}]][(x_1)\mathcal{A}][(x_1)\mathcal{B}]]][[[[(x_1)\mathcal{A}][(Ex_1)\mathcal{B}]][(x_1)\mathcal{A}][(x_1)\mathcal{B}]]]]$	4.6.6
	$[(Ex_1)\mathcal{A}[(x_1)\mathcal{A}][(x_1)\mathcal{B}](x_1)\mathcal{A}[(x_1)\mathcal{A}][(x_1)\mathcal{B}]][(Ex_1)\mathcal{A}[(x_1)\mathcal{A}][(x_1)\mathcal{B}](Ex_1)\mathcal{B}[(x_1)\mathcal{A}][(x_1)\mathcal{B}]][(x_1)\mathcal{B}[(x_1)\mathcal{A}][(x_1)\mathcal{B}](x_1)\mathcal{A}[(x_1)\mathcal{A}][(x_1)\mathcal{B}]][(x_1)\mathcal{B}[(x_1)\mathcal{A}][(x_1)\mathcal{B}](Ex_1)\mathcal{B}[(x_1)\mathcal{A}][(x_1)\mathcal{B}]]$	4.3.1
	$[(Ex_1)\mathcal{A}([][(x_1)\mathcal{B}](x_1)\mathcal{A}[][(x_1)\mathcal{B}]][(Ex_1)\mathcal{A}[(x_1)\mathcal{A}][(x_1)\mathcal{B}](Ex_1)\mathcal{B}[(x_1)\mathcal{A}][(x_1)\mathcal{B}]][(x_1)\mathcal{B}[][](x_1)\mathcal{A}[][]][(x_1)\mathcal{B}[(x_1)\mathcal{A}][](Ex_1)\mathcal{B}[(x_1)\mathcal{A}][]]$	4.3.2
	$[[]][(Ex_1)\mathcal{A}[(x_1)\mathcal{A}][(x_1)\mathcal{B}](Ex_1)\mathcal{B}[(x_1)\mathcal{A}][(x_1)\mathcal{B}]][[]][[]]$	4.3.3
	$[(Ex_1)\mathcal{A}[(x_1)\mathcal{A}][(x_1)\mathcal{B}](Ex_1)\mathcal{B}[(x_1)\mathcal{A}][(x_1)\mathcal{B}]]$	4.3.1
	The given formula is not a $\mathbb{K}$ axiom or theorem	4.8.1
5.6	$\mathcal{A}(\tau)\supset(Ex)\mathcal{A}(x)$ with τ term free for x in $\mathcal{A}(x)$.	
	$[\mathcal{A}(\tau)[(Ex)\mathcal{A}(x)]]$ with τ term free for x in $\mathcal{A}(x)$.	4.2.4
	$(x)[\mathcal{A}(x)[(Ex)\mathcal{A}(x)]]$	4.4.1
	$[(Ex)\mathcal{A}(x)[(Ex)\mathcal{A}(x)]]$	4.6.2
	$[(Ex_1)\mathcal{A}[(Ex_1)\mathcal{A}]]$	4.6.6
	$[(Ex_1)\mathcal{A}[]]$	4.3.2
	$[[]]$	4.3.3
	$()$	4.3.1
	The given formula is a $\mathbb{K}$ axiom or theorem	4.8.1

5.7	$(x)\mathcal{A}(x)\supset(Ex)\mathcal{A}(x)$	
	$[(x)\mathcal{A}(x)[(Ex)\mathcal{A}(x)]]$	4.2.4
	$[(x_1)\mathcal{A}[(Ex_1)\mathcal{A}]]$	4.6.6
	$[(x_1)\mathcal{A}[(x_1)\mathcal{A}]]$	4.7.1
	$[(x_1)\mathcal{A}[]]$	4.3.2
	$[[]]$	4.3.3
	$()$	4.3.1
	The given formula is a $\mathbb{K}$ axiom or theorem	4.8.1
5.8	$(x)(y)\mathcal{A}(xy)\supset(y)(x)\mathcal{A}(xy)$	
	$[(x)(y)\mathcal{A}(xy)[(y)(x)\mathcal{A}(xy)]]$	4.2.4
	$[(x_1)(x_2)\mathcal{A}[(x_2)(x_1)\mathcal{A}]]$	4.6.6
	$[(x_1)(x_2)\mathcal{A}[(x_1)(x_2)\mathcal{A}]]$	4.7.1
	$[(x_1)(x_2)\mathcal{A}[]]$	4.3.2
	$[[]]$	4.3.3
	$()$	4.3.1
	The given formula is a $\mathbb{K}$ axiom or theorem	4.8.1
5.9	$(x)\mathcal{A}(x)\equiv\sim(Ex)\sim\mathcal{A}(x)$	
	$(x)\mathcal{A}(x)\equiv[(Ex)[\mathcal{A}(x)]]$	4.2.1
	$[(x)\mathcal{A}(x)[[(Ex)[\mathcal{A}(x)]]]][[(x)\mathcal{A}(x)][(Ex)[\mathcal{A}(x)]]]$	4.2.5
	$[(x)\mathcal{A}(x)(Ex)[\mathcal{A}(x)]][[(x)\mathcal{A}(x)][(Ex)[\mathcal{A}(x)]]]$	4.3.1
	$[(x)\mathcal{A}(x)[(x)\mathcal{A}(x)]][[(x)\mathcal{A}(x)][[(x)\mathcal{A}(x)]]]$	4.6.4
	$[(x_1)\mathcal{A}[(x_1)\mathcal{A}]][[(x_1)\mathcal{A}][[(x_1)\mathcal{A}]]]$	4.6.6
	$[(x_1)\mathcal{A}[(x_1)\mathcal{A}]][[(x_1)\mathcal{A}](x_1)\mathcal{A}]$	4.3.1
	$[(x_1)\mathcal{A}[]][[](x_1)\mathcal{A}]$	4.3.2
	$[[]][[]]$	4.3.3
	$()$	4.3.1
	The given formula is a $\mathbb{K}$ axiom or theorem	4.8.1
5.10	$(x)(\mathcal{A}(x)\supset\mathcal{B}(x))\supset((x)\mathcal{A}(x)\supset(x)\mathcal{B}(x))$	
	$[(x)[\mathcal{A}(x)[\mathcal{B}(x)]][[(x)\mathcal{A}(x)[(x)\mathcal{B}(x)]]]]$	4.2.4
	$[(x)[\mathcal{A}(x)[\mathcal{B}(x)]](x)\mathcal{A}(x)[(x)\mathcal{B}(x)]]$	4.3.1

$[[[(Ex)(y)[\mathcal{A}(x)[\mathcal{B}(y)]](x)\mathcal{A}(x)[(x)\mathcal{B}(x)]]][[(Ey)(x)[\mathcal{A}(x)[\mathcal{B}(y)]](x)\mathcal{A}(x)[(x)\mathcal{B}(x)]]]]$ 4.5.2

$[[[(Ex)[\mathcal{A}(x)(Ey)[\mathcal{B}(y)]](x)\mathcal{A}(x)[(x)\mathcal{B}(x)]]][[(Ey)[(Ex)\mathcal{A}(x)[\mathcal{B}(y)]](x)\mathcal{A}(x)[(x)\mathcal{B}(x)]]]]$ 4.6.2

$[[[[(x)\mathcal{A}(x)[(y)\mathcal{B}(y)]](x)\mathcal{A}(x)[(x)\mathcal{B}(x)]]][[[(Ex)\mathcal{A}(x)(y)[\mathcal{B}(y)]](x)\mathcal{A}(x)[(x)\mathcal{B}(x)]]]]$ 4.6.4

$[[[[(x)\mathcal{A}(x)[(y)\mathcal{B}(y)]](x)\mathcal{A}(x)[(x)\mathcal{B}(x)]]][[[(Ex)\mathcal{A}(x)[(Ey)\mathcal{B}(y)]](x)\mathcal{A}(x)[(x)\mathcal{B}(x)]]]]$ 4.6.2

$[[[[(x_1)\mathcal{A}[(x_1)\mathcal{B}]](x_1)\mathcal{A}[(x_1)\mathcal{B}]]][[[(Ex_1)\mathcal{A}[(Ex_1)\mathcal{B}]](x_1)\mathcal{A}[(x_1)\mathcal{B}]]]]$ 4.6.6

$[[(x_1)\mathcal{A}](x_1)\mathcal{A}[(x_1)\mathcal{B}][(Ex_1)\mathcal{A}](x_1)\mathcal{A}[(x_1)\mathcal{B}]][[(x_1)\mathcal{A}](x_1)\mathcal{A}[(x_1)\mathcal{B}](Ex_1)\mathcal{B}(x_1)\mathcal{A}[(x_1)\mathcal{B}]][(x_1)\mathcal{B}(x_1)\mathcal{A}[(x_1)\mathcal{B}][(Ex_1)\mathcal{A}](x_1)\mathcal{A}[(x_1)\mathcal{B}]][(x_1)\mathcal{B}(x_1)\mathcal{A}[(x_1)\mathcal{B}](Ex_1)\mathcal{B}(x_1)\mathcal{A}[(x_1)\mathcal{B}]]$ 4.3.1

$[[](x_1)\mathcal{A}[(x_1)\mathcal{B}][(Ex_1)\mathcal{A}](x_1)\mathcal{A}[(x_1)\mathcal{B}]][[](x_1)\mathcal{A}[(x_1)\mathcal{B}](Ex_1)\mathcal{B}(x_1)\mathcal{A}[(x_1)\mathcal{B}]][(x_1)\mathcal{B}(x_1)\mathcal{A}[][(Ex_1)\mathcal{A}](x_1)\mathcal{A}[]][(x_1)\mathcal{B}(x_1)\mathcal{A}[](Ex_1)\mathcal{B}(x_1)\mathcal{A}[]]$ 4.3.2

$[[]][[]][[]][[]]$ 4.3.3

$()$ 4.3.1

The given formula is a $\mathbb{K}$ axiom or theorem 4.8.1

6. Automatic Decisions of Elementary Arithmetic Theorems

R. Robinson has given a representation of formal number theory[10] that can be obtained from $\mathbb{K}$ by adding the following axioms:

6.1 $A^{1}_{=}(xx)$

6.2 $A^{1}_{=}(xy)\supset A^{1}_{=}(yx)$

6.3 $A^{1}_{=}(xy)\supset(A^{1}_{=}(yz)\supset A^{1}_{=}(xz))$

6.4 $A^{1}_{=}(xy)\supset A^{1}_{=}(f^{1}_{s}(x)f^{1}_{s}(y))$

6.5 $A^{1}_{=}(xy)\supset A^{1}_{=}(f^{2}_{+}(xz)f^{2}_{+}(yz))\wedge A^{1}_{=}(f^{2}_{+}(zx)f^{2}_{+}(zy))$

6.6 $A^{1}_{=}(xy)\supset A^{1}_{=}(f^{2}_{\times}(xz)f^{2}_{\times}(yz))\wedge A^{1}_{=}(f^{2}_{\times}(zx)f^{2}_{\times}(zy))$

---

[10] See 8. and 9. in the references.

6.7 $A^{1}_{=}(f^{1}_{s}(x)f^{1}_{s}(y))\supset A^{1}_{=}(xy)$
6.8 $\sim A^{1}_{=}(a_0 f^{1}_{s}(x))$
6.9 $\sim A^{1}_{=}(xa_0)\supset(Ey)A^{1}_{=}(xf^{1}_{s}(y))$
6.10 $A^{1}_{=}(f^{2}_{+}(xa_0)x)$
6.11 $A^{1}_{=}(f^{2}_{+}(xf^{1}_{s}(y))f^{1}_{s}(f^{2}_{+}(xy)))$
6.12 $A^{1}_{=}(f^{2}_{\times}(xa_0)a_0)$
6.13 $A^{1}_{=}(f^{2}_{\times}(xf^{1}_{s}(y))f^{2}_{+}(f^{2}_{\times}(xy)x))$

Abbreviate the axioms 6.1, …, 6.13 with (*A1*), …, (*A13*) respectively and Robinson's formal number theory with $\mathbb{R}$.

Let (*NF*) be any formula of formal number theory to decide. Now, consider (*A1*)⊃(…⊃((*A13*)⊃(*NF*))…), i.e. [(*A1*), …, (*A13*)[(*NF*)]], we can always decide if [(*A1*), …, (*A13*)[(*NF*)]] is a $\mathbb{K}$ theorem by procedure in §4. But $\mathbb{R}$ is obtained from $\mathbb{K}$ formula by adding (*A1*), …, (*A13*) as axioms. Thus if (*A1*)⊃(…⊃((*A13*)⊃(*NF*))…) is a $\mathbb{K}$ theorem then (*NF*) is an $\mathbb{R}$ theorem. Vice versa, (*NF*) is an $\mathbb{R}$ theorem only if (*A1*)⊃(…⊃((*A13*)⊃(*NF*))…) is an $\mathbb{R}$ theorem because truth is implied by any premise; but, obviously, (*A1*)⊃(…⊃((*A13*)⊃(*NF*))…) does not require the axioms (*A1*), …, (*A13*) in its proof, thus it is a $\mathbb{K}$ theorem too that can be decided by procedure in §4. Finally, (*NF*) is an $\mathbb{R}$ theorem if and only if (*A1*)⊃(…⊃((*A13*)⊃(*NF*))…) is a $\mathbb{K}$ theorem and so procedure in §4 can decide indirectly if an $\mathbb{R}$ formula is an $\mathbb{R}$ axiom or theorem.

Now, let (*NF*) be an $\mathbb{R}$ logically valid formula that is not an $\mathbb{R}$ theorem. Thus (*A1*)⊃(…⊃((*A13*)⊃(*NF*))…) has to be $\mathbb{R}$ logically valid non-theorem. As the proof of (*A1*)⊃(…⊃((*A13*)⊃(*NF*))…) does not require the axioms (*A1*), …, (*A13*), it can not be $\mathbb{K}$ theorem too. So, Gödel's completeness theorem for first order predicative calculus[11] assures us that some interpretations where

---

[11] See 10. in the References.

(*A1*), …, (*A13*) are true and (*NF*) are false have to exist. But the truth of (*A1*), …, (*A13*) allows these interpretations to be applied to $\mathbb{R}$ too. Thus (*NF*) is false in $\mathbb{R}$ for some interpretations and it cannot be logically valid in $\mathbb{R}$ in contradiction with the starting hypothesis. Hence,

6.14 **Meta-theorem:** *The hyperincursive procedure in §4 makes complete Robinson's formal number theory too*, *i.e. it makes the elementary arithmetic decidable*.[12]

References

1. G. Gentzen (1935) *Untersuchungen über das logische Schliessen*, Mathematische Zeitschrift, 39, pp. 176-210, 405-431.
2. Malatesta M. (1997) *The Primary Logic*, Leominster: Gracewing Fowler Wright Books, pp. 159-164.
3. Aristotelis, *Analytica priora* in *Analytica Priora et Posteriora*, rec. W. D. Ross, aux. L. minio-Paluello, (1964) Oxonii: E Typographeo Clarendoniano.
4. Malatesta M. (2000), *La logica delle funzioni*, Roma: Millennium Romæ, Vol. I, pp. 46-50.
5. Dubois D. M. (1997) *Generation od fractals from incursive automata, digital diffusion and wave equation systems*, BioSystems, 43 pp. 97-114.
6. Dubois D. M., and Resconi G. (1992) *Hyperincursivity - A new mathematical theory*. Liège: Presses Universitaires de Liège.
7. Mendelson E. (1964) *Introduction to Mathematical Logic*. Princeton: Van Nostrand, chapt. 2.
8. Robinson R. M. (1950) *An Essentially Undecidable Axiom System*, Proc. int. Congr. Math. Cambridge, Vol 1, pp. 729-730
9. Mendelson E. (1964) *Introduction to Mathematical Logic*. Princeton: Van Nostrand, chapt. 3. par. 6.
10. K. Gödel (1930), *Die Vollständigkeit der Axiome des logischen Funktionenkalkülus*, Mh. Math. Phys., 37, pp. 349-360.
11. Dubois D. M. (1999) *Hyperincursive Mc Culloch and Pitts Neurons for Designing a Computing Flip-Flop Memory* in *Computing Anticipatory System: CASYS '98 - Second International Conference*, ed. by D. M. Dubois, American Institute of Physics, AIP Conference Proceedings 465, pp. 3-21.
12. Grappone A. G. (2000) *Decidability of Formal Theories and Hyperincursivity Theory: CASYS '99 - Third International Conference* ed.

[12] Cf. 11. and 12. in the References.

by D. M. Dubois, American Institute of Physics, AIP Conference Proceedings 517, pp. 141-148.