Annals of Pure and Applied Logic 41 (1989) 153–178
North-Holland

ON THE NUMBER OF STEPS IN PROOFS

Jan KRAJÍČEK*
Mathematical Institute ČSAV, Žitná 25, Praha 1, 115 67 Czechoslovakia

Communicated by D. van Dalen
Received 15 October 1985; revised 2 July 1986

In this paper we prove some results about the complexity of proofs. We consider proofs in Hilbert-style formal systems such as in [17]. Thus a proof is a sequence of formulas satisfying certain conditions. We can view the formulas as being strings of symbols; hence the whole proof is a string too.

We consider the following measures of complexity of proofs: *length* (= the number of symbols in the proof), *depth* (= the maximal depth of a formula in the proof) and *number of steps* (= the number of formulas in the proof).

For a particular formal system and a given formula A we consider the shortest length of a proof of A, the minimal depth of a proof of A and the minimal number of steps in a proof of A. The main results are the following: (1) a bound on the depth in terms of the number of steps: Theorem 2.2, (2) a bound on the depth in terms of the length: Theorem 2.3, (3) a bound on the length in terms of the number of steps for restricted systems: Theorem 3.1. These results are applied to obtain several corollaries. In particular we show: (1) a bound on the number of steps in a cut-free proof, (2) some speed-up results, (3) bounds on the number of steps in proofs of Paris–Harrington sentences.

Some papers related to our research are listed in the references. We were especially influenced by Parikh's paper [17] on the famous conjecture of Kreisel (cf. [3], problem 34]).

Many important problems in this field remain open. We hope that our paper will contribute to progress in this area.

It should be noted that some results of this paper can be equally well obtained using unification theory (cf. [5]), by translating a complexity-of-proof problem into a unification problem. A unification algorithm solving the unification problem can then be constructed and the complexity of the unification algorithm analyzed.

As pointed out by the referee this method can be used to solve the problem stated in Section 3 for some non-simple (defined below) schematic systems.

0. Preliminaries

A general notion of a formal logical system about which we will prove our results is the *schematic system.* Roughly speaking: a schematic system is specified by its *language, a finite number of axiom schemes* (including single axioms) and *a finite number of schematic rules of inference.* The most important thing is, however, to set down the conditions that determine the set of formulas, terms or other syntactic objects which can be substituted in a particular scheme or a schematic rule. E.g., if no restrictions on these conditions are assumed, then any

* I would like to thank P. Pudlák for many valuable discussions. I acknowledge important suggestions of G. Kreisel concerning this research (communicated via P. Pudlák). The suggestions of the referee greatly helped to improve the presentation of the paper.

theory can be formalized as a schematic system: take a scheme consisting of a single formula variable R together with a substitutability condition that says that a formula A can be substituted for R iff A is a theorem of the theory.

We shall consider the following situation. A *language* L is determined by specifying its:

(i) variables: $x, y, \ldots$ possibly of higher orders: $X, Y, \ldots, \mathscr{X}, \mathscr{Y}, \ldots$ but with finitely many orders;

(ii) first-order function symbols $f, g, \ldots$;

(iii) first-order predicate symbols: $p, q, \ldots$ and possibly the binary symbol $\in$ if the system is of a higher order;

(iv) logical symbols, $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\exists$, $\forall \ldots$ and $=$; and

(v) other symbols: brackets $(,), [,], \ldots$.

We do not assume that all above mentioned symbols actually occur in L. Terms and formulas are built according to usual rules. The variables of $(n+1)$-th order are intended to range over the sets of objects of the n-th order; in particular, variables have no arguments and and the only atomic expression in which $(n+1)$-th order variable $\mathscr{X}$ can occur is $X \in \mathscr{X}$ where X is the n-th order.

We have included the higher order variables for making the applications of the results of Section 4 more straight-forward. However, the results for systems allowing higher-order variables are trivial generalizations of the results for first-order systems (since higher-order variables do not have arguments).

The simplest way to define a scheme or a schematic rule is to extend L by adding metasymbols for variables $\tilde{x}, \tilde{y}, \ldots, \tilde{X}, \tilde{Y}, \ldots$, terms $\tilde{t}, \tilde{s}, \ldots$, and formulas $\tilde{A}, \tilde{B}, \ldots$; let us denote this extension L^*. Metavariables and term variables are assumed not to have arguments. Thus terms of L^* have form $t(x/s, y/r, \ldots)$, where $t(x, y, \ldots)$ is an L-term and $s, r, \ldots$ are metavariables or term variables. A formula variable may contain as arguments any L^*-terms, in particular: variables, metavariables, terms, term variables. More properly: for any n we have in L^* infinitely many formula variables of arity n, $\tilde{A}(x_1, \ldots, x_n)$. Substitutions of L-expressions for metasymbols in L^*-expressions must fulfil the usual conditions (cf. [17]). In particular, if an L-formula $A(x_1, \ldots, x_n)$ is substituted for $\tilde{A}(x_1, \ldots, x_n)$ and L-terms, $s', r', \ldots$ for L^*-terms $s, r, \ldots$, then for an L^*-expression $\tilde{A}(x_1/s, x_2/r, \ldots)$ we substitute the L-expression $A(x_1/s', y_1/r', \ldots)$, where all free occurrences of the $x_1, x_2, \ldots$ are substituted for.

A schematic rule (w.r.t. L) is determined by:

(i) *a* $(k+1)$*-tuple* $A_0, A_1, \ldots, A_k$ of L^*-formulas written usually as:

$$\frac{A_1, \ldots, A_k}{A_0} \quad \text{and}$$

(ii) *a domain* which is given in the following way: if in the formulas $A_0, \ldots, A_k$ altogether a metavariables, b term variables and c formula variables occur, then the domain is some set of $(a+b+c)$-tuples where the first a elements are variables of L, the second b elements are terms of L and the last c elements are formulas of L.

An instance of the schematic rule is obtained by taking an $(a+b+c)$-tuples from its domain and substituting its elements into a $(k+1)$-tuple $A_0, \ldots, A_k$ (according to the comment above, cf. [17, p. 31]).

Although the subject is familiar, an example may be useful: let L be the usual language of the first order predicate calculus. Then the schematic rule

$$\frac{\tilde{C}\to\tilde{A}(\tilde{x})}{\tilde{C}\to\forall\tilde{x}\,\tilde{A}(\tilde{x})}$$

with the domain $\{(x, A, C) \mid x \text{ is not free in } C\}$ defines the generalization rule.

The special case of the schematic rule with $k=0$ is usually called *a scheme*. E.g. the schematic rule

$$\frac{}{\neg\neg\tilde{A}\to\tilde{A}}$$

with the full domain defines one of the usual axiom schemes of the classical propositional calculus.

Since schematic systems are designed to represent systems with simple syntactic properties and which are easily manageable, we will now pose conditions on the domains of schematic rules.

Thus the *schematic system* $\mathbb{A}$ is given by:

(i) a language L;

(ii) a finite number of schematic rules whose domains are defined by some conjunctions of the conditions (on $(a+b+c)$-tuples) of one of the following types:

1. $\tilde{x}\neq\tilde{y}, \ldots, \tilde{X}\neq\tilde{Y}, \ldots$
2. $\tilde{x}$ does not occur in $\tilde{A}$, resp. in $\tilde{t}$ resp. $\tilde{X}, \ldots$ does not $\ldots$,
3. $\tilde{x}$ is free for $\tilde{t}$ in $\tilde{A}$,
4. $\tilde{x}$ resp. $\tilde{X}, \ldots$ is not free in $\tilde{A}$,
5. variable $\tilde{x}$ resp. $\tilde{X}, \ldots$ is of order k.

(Although we confined ourselves to this list, it is by no means an exhaustive one. E.g. a condition of type "$\tilde{A}$ has depth at most n", or of type "$\tilde{A}$ does not contain a bounded variable of order $\geqslant 2$" would help to define various fragments of arithmetic. Generally we can allow, without damaging our results, any conditions on syntactic objects which are preserved by subobjects, i.e. subformulas and subterms—e.g. the condition "t is closed".) *From now on* we assume that the schematic rules are defined according to the above restrictions.

A list of restrictions, called 'Parikh's restrictions', based on 2., 3. and 4. is found in [5].

A proof in the schematic system $\mathbb{A}$, called an $\mathbb{A}$-proof, is any finite sequence $d = A_1, \ldots, A_k$ of L-formulas such that the obvious condition is satisfied: for each $i \leqslant k$ there exist $i_1, \quad , i_j < i$ such that

$$\begin{array}{c} A_{i_1}, \ldots, A_{i_j} \\ A_i \end{array}$$

is an instance of some schematic rule of $\mathbb{A}$. Such a proof d is an $\mathbb{A}$-*proof* of A_k and has k *steps*: $A_1, \ldots, A_k$.

For a formula A (resp. a proof d) let us denote by $|A|$ (resp. $|d|$) *the length* of A (resp. d), i.e. the total number of symbols in it.

By *a tree* we shall mean a finite, structured tree with a root, i.e. a structure $(T, \leqslant_T, L_T)$ where:

(i) T is a finite set,

(ii) $\leqslant_T$ is a partial ordering of T,

(iii) there is a unique element $\hat{T}$ of T, called *the root* of T, such that for any $x \in T$: $x \leqslant_T \hat{T}$,

(iv) for any $x \in T$ the set $\{y \in T \quad x \leqslant_T y\}$ is linearly ordered by $\leqslant_T$,

(v) the relation L_T, called *the structure* of T, is a subset of $T \times (T \times T)$,

(vi) for each $x \in T$ the set of all *sons* of x:

$$\{y \in T \mid y \leqslant_T x \wedge x \neq y \wedge (\forall z : y \leqslant_T z \rightarrow (y = z \vee x \leqslant_T z))\}$$

is linearly ordered by the relation:

$$L''_T\{x\} = \{[y, z] \mid [x, [y, z]] \in L_T\}.$$

Roughly: "the structure of a tree orders sons of any node x of the tree from left to right ($L''_T\{x\}$ is the ordering)".

Since there will be no danger of confusion, we shall use T also to denote the whole structure $(T, \leqslant_T, L_T)$.

With a tree T we associate *the rank function* $r_T : T \rightarrow \omega$ such that

(i) $r_T(\hat{T}) = 0$,

(ii) $r_T(y) = 1 + r_T(x)$ for y a son of x.

Hence $x \leqslant_T y$ implies $r_T(x) \geqslant r_T(y)$.

Let us define $r(T) = \max\{r_T(x) \mid x \in T\}$; $r(T)$ is called *the height* of T. It is the maximal length of a path in T minus 1.

If $T_1, \ldots, T_k$ are mutually disjoint trees, then the structure

$$(T_1, \ldots, T_k) = \left(\bigcup_{i \leqslant k} T_i, \bigcup_{i \leqslant k} \leqslant_{T_i}, \bigcup_{i \leqslant k} L_{T_i}\right)$$

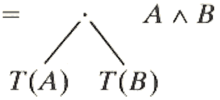
is called *a forest*. It is clear that the definitions of rank function and height apply to forests as well. It should be stressed explicitly that if $x \in (T_1, \ldots, T_k)$, then there is a unique $i \leqslant k$ such that $x \in T_i$.

A labelled tree (resp. forest) is a tree (resp. forest) with *a label function* $l : T \rightarrow X$, where X is some set (of labels). In our case the labels will be L^*-formulas.

With a formula A we associate a *labelled tree* $T(A)$ defined informallv as follows:

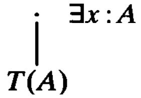
(i) A atomic: $T(A) = \cdot A$,

(ii) $T(A \wedge B) =$



and similarly for other connectives,

(iii) $T(\exists x : A) =$



and similarly for $\forall$.

Thus $T(A)$ is a tree of subformulas of A whose root corresponds to A and sons of the node u correspond to the immediate subformulas of $l(u)$.

We define *the depth of a formula A*, $\mathrm{dp}(A)$, to be the height of its tree $T(A)$. Thus $\mathrm{dp}(A) = 0$ iff A is atomic.

The depth of proof d, $\mathrm{dp}(d)$, is the maximal depth of its steps, i.e. in the example above: $\mathrm{dp}(d) = \max\{\mathrm{dp}(A_i) \mid i \leqslant k\}$.

If $d = A_1, \ldots, A_k$ is a proof, then $T(d)$ will be its *forest* defined as the disjoint union of $T(A_i)$'s, $i \leqslant k$.

Now we are coming to the first technical result about schematic systems. The reader can go through the proofs with some favourite example in mind, e.g. PC (predicate calculus), PA (Peano's arithmetic), A_n (n-th order arithmetic), ZF, GB, etc.

1. Technical result

Let T be a forest. A set $S \subseteq T$ satisfying:

$$u \geqslant_T v \in S \rightarrow u \in S$$

is called *a filter* on T. Since any filter on T is a forest the notions defined above can be applied to filters as well.

Lastly, given a forest T and $u \in T$ define a *tree* $T[u]$ by:

$$T[u] = \{v \in T \mid v \leqslant_T u\},$$

with the ordering and the structure inherited from T.

Throughout this paper an isomorphism of trees is assumed to preserve the roots, the structure, the relation 'to be a son', the ordering and the rank but not necessarily the label even if the trees are labelled. Observe that there is at most one isomorphism between any two trees.

Technical Lemma. *Let T be a forest and S a filter on T. Then there exists another filter C on T such that*:

(i) $C \supseteq S$,

(ii) $C[u] \simeq C[v]$, *whenever* $u, v \in S$ *and* $T[u] \simeq T[v]$,

(iii) $r(C) \leqslant |S|$, *where* $|S|$ *is the cardinality of the filter* S,

(iv) *for any* $u \in C$, u *is either an end-node of* C *or* C *contains all* T-*sons of* u.

Proof. Define the set

$$C_0 = \{x \in T \mid \exists y \in S\colon T[x] \simeq T[y]\}$$

and let C_1 be the maximal filter such that $C_1 \subseteq C_0$.

We claim that C_1 fulfils the conditions (i), (ii) and (iii) of the lemma.

Clearly $C_1 \supseteq S$ since $C_0 \supseteq S$ and S is a filter. Assume that $u, v \in S$, $T[u] \simeq T[v]$ and let $x \in T[u]$ correspond to $y \in T[v]$ in the isomorphism $T[u] \simeq T[v]$. Assume also $x \in C_1$.

Since $T[x] \simeq T[y]$ we have $y \in C_0$. To prove that $y \in C_1$ it is enough to prove that for all $z \geqslant_T y$: $z \in C_0$.

Let $z \geqslant_T y$. If $z \geqslant_T v$, then $z \in S$ and so $z \in C_0$ too. If $v \geqslant_T z$ and $z \geqslant_T y$, then take $t \in T[u]$ corresponding to z in $T[u] \simeq T[v]$. $x \in C_1$ implies $t \in C_0$, hence $z \in C_0$ too (since $T[t] \simeq T[z]$). This proves (ii).

To prove (iii) let $x_1, \ldots, x_n \in C_1$ be a sequence of nodes such that x_{i+1} is a son of x_i. Let $y_1, \ldots, y_n \in S$ be the y's corresponding to the x's by the definition of C_0. In particular, $T[x_i] \simeq T[y_i]$ and so all y_i's are mutually different (since $T[x_i] \not\simeq T[x_j]$ for $i \neq j$). Evidently then $n \leqslant |S|$. Thus in C_1 it is possible to find a path of length at most $|S|$, i.e. $r(C_1) \leqslant S$. This proves (iii).

Now define:

$$C = C_1 \cup \{x \in T \quad \exists y, z \in C_1\colon \text{“}x, y \text{ are sons of } z\text{”}\}.$$

C is clearly a filter satisfying (i) and (ii) since C_1 satisfies them. Also $r(C) \leqslant |S|$ since $r(C) = r(C_1)$. C satisfies (iv) by its definition. We are done. □

2. The estimate of the depth from the number of steps

Let us fix some schematic system $\mathbb{A}$ and some $\mathbb{A}$-proof $d = A_1, \ldots, A_k$ of a formula $B = A_k$ together with information I which determines the schematic rules of $\mathbb{A}$ and A_j's which were used to infer A_i, for all $i \leqslant k$.

If d should be an $\mathbb{A}$-proof, then it must satisfy some substitutability conditions given by the information I above. E.g., if d contains an application of the rule modus ponens (cut-rule) such that A_i is derived from A_j and $A_r = A_j \rightarrow A_i$, then the left immediate subformula of A_r must be identical with A_j and the right one with A_i. In the language of trees and forests: if x_j and x_i are the left and the right sons of $\hat{T}(A_r)$ (i.e. of the root of $T(A_r)$) resp., then:

$$T(d)[\hat{T}(A_i)] \simeq T(d)[x_i] \quad \text{and} \quad T(d)[\hat{T}(A_j)] \simeq T(d)[x_j].$$

On the other hand: if we have a forest fulfilling these conditions, it is possible to label it in such a way that modus ponens is applicable to these formulas.

Let us now be more general and give more detail. If Z is an L^*-formula and A is an instance of Z, then it is possible to inject the tree $T(Z)$ into $T(A)$ without necessarily preserving labels. Call the image of $T(Z)$ in $T(A)$ under this injection the *Z-part of $T(A)$* (since the root is assumed to be preserved, such a part is unique). Thus any Z-part is a filter on $T(A)$.

For our fixed proof d and its fixed information I we define the *basic part of* $T(d)$ as the least filter S on $T(d)$ satisfying: for $i_0 \leq k$, if A_{i_0} is derived using an instance

$$\frac{A_{i_1}, \quad\_\_\_\_ , A_{i_j}}{A_{i_0}}$$

of the schematic rule

$$\frac{Z_{i_1}, \quad\_\_\_\_ , Z_{i_j}}{Z_{i_0}}$$

then S contains all Z_{i_u}-parts of $T(A_{i_u})$'s, $u \leq j$.

The basic part of a proof is, roughly speaking, the 'image' of all schematic rules used in it. Let us denote by S_d the basic part of d. (A better notation would be $S_d(I)$, referring also to information I, but we shall omit this.)

Let us have any filter $C \supseteq S_d$ on $T(d)$ for which the following holds:

(a) any $x \in C$ is either an end-node of C or C contains all $T(d)$-sons of x,

(b) if $x, y \in S_d$ and $T(d)[x] \simeq T(d)[y]$, then $C[x] \simeq C[y]$.

Now we can define labelled trees $T(A_i)^C$, for $i \leq k$, as follows (l is the label function of $T(d)$, l^C is the new label function):

(i) $T(A_i)^C = T(A_i) \cap C$,

(ii) the structure, the ordering and the rank of $T(A_i)^C$ are the same as in $T(A_i)$,

(iii) define l^C by:

(a) if $x \in T(A_i)^C$ is an end-node of $T(A_i)$ put: $l^C(x) = l(x)$,

(b) if $x \in T(A_i)^C$ is an end-node of $T(A_i)^C$ but not of $T(A_i)$ put: $l^C(x) = \tilde{A}_{l(x)}$, 0-ary formula variable (this means that the l^C-labels of such nodes x, y are equal iff their l-labels are equal),

(c) for $x, y, z \in T(A_i)^C$, y resp. z the left resp. the right son of x put: if $l(x) = l(y) \wedge l(z)$ define $l^C(x) = l^C(y) \wedge l^C(z)$, and analogously for other connectives and for quantifiers.

Observe that condition (a) put above on C implies that (iii)(c) is a correct definition.

Let A_i^C be the unique L^*-formula such that $T(A_i)^C = T(A_i^C)$ and let $d^C = A_1^C, \dots, A_k^C$.

Definition. *The relation R* on the set of end-nodes of $T(d^C)$ which are labelled by formula variables is defined by:

$$u\,R\,v \quad \text{iff} \quad T(d)[u] \simeq T(d)[v]$$

(we use u resp. v for an end-node of $T(d^C)$ and also for its image in $T(d)$, but this does not lead to any confusion).

Clearly, R is an equivalence relation.

Definition. $V_{\mathbb{A}}$ is the set of all L-variables occurring in formula variables used in the schematic rules of $\mathbb{A}$, i.e., if $\tilde{A}_i(x_1^i, \dots, x_{k_i}^i)$, $i \leqslant m$, are all of the formula variables used in the schematic rules of $\mathbb{A}$, then:

$$V_{\mathbb{A}} = \bigcup_{i \leqslant m} \{x_1^i, \quad , x_{k_i}^i\}.$$

Let ε be a substitution of L-formulas for formula variables in d^C. We shall call ε a *suitable substitution* iff for any equivalence class P of the equivalence relation R:

either (i) for some L-formula H and for all $u \in P$: $\varepsilon(\tilde{A}_{l(u)}) = H$, and H contain no variables from $V_{\mathbb{A}}$ or variables occurring in d,

or (ii) for all $u \in P$: $\varepsilon(\tilde{A}_{l(u)}) = l(u)$.

Lemma. *Let C be a filter on $T(d)$ such that*:

(i) $C \supseteq S_d$,

(ii) *for $x, y \in S_d$, $T(d)[x] \simeq T(d)[y]$ implies $C[x] \simeq C[y]$,*

(iii) *any $x \in C$ is either an end-node of C or C contains all its T-sons.*

Then any suitable substitution ε of L-formulas for formula variables in d^C transforms d^C into an $\mathbb{A}$-proof, i.e. $\varepsilon(d^C)$ is an $\mathbb{A}$-proof.

Proof. Let $\varepsilon(A_1^C), \dots, \varepsilon(A_k^C)$ be any substitution-instance of d^C given by a suitable substitution ε.

We shall show, for $i_0 \leqslant k$, that if A_{i_0} was derived using an instance

$$(\dagger) \qquad \frac{A_{i_1}, \dots, A_{i_j}}{A_{i_0}}$$

of the schematic rule

$$\frac{Z_{i_1}, \dots, Z_{i_j}}{Z_{i_0}}$$

then

$$(\dagger\dagger) \qquad \frac{(A_{i_1}^C), \dots, \varepsilon(A_{i_j}^C)}{\varepsilon(A_{i_0}^C)}$$

is also an instance of this schematic rule.

It is enough to specify an $(a+b+c)$-tuple $x_1, \dots, x_a, t_1, \dots, t_b, B_1, \dots, B_c$ from the domain of the rule which would determine (††). Let $y_1, \dots, s_1, \dots, C_1, \dots$ be the $(a+b+c)$-tuple determining (†). Define $x_1 = y_1, \dots$ and $t_1 = s_1, \dots$.

Assume that C_j was substituted for $\tilde{D}_j$ in the schematic rule. $\tilde{D}_j$ occurs as the label of some end-nodes of the forest $T(Z_{i_j}) \dot{\cup} \cdots \dot{\cup} T(Z_{i_0})$. All these end-nodes (more precisely their images) are in Z_{i_u}-parts of $T(A_{i_u})$'s, $u \leqslant j$, so in the basic part S_d of d.

Moreover, for any two such nodes x, y we have: $T(d)[x] \simeq T(d)[y\}$, since $T(l(x)) = T(l(y)) = T(C_j)$.

By the assumption about C: $C[x] \simeq C[y]$ for any such x and y.

Now $l^C(x)$ and $l^C(y)$ have forms:

$$l^C(x) = E_j({}^x1/p_1, \dots, {}^xn/p_n, \tilde{A}_{l(u_1)}, \dots)$$

and

$$l^C(y) = E_j({}^x1/r_1, \dots, {}^xn/r_n, \tilde{A}_{l(v_1)}, \dots),$$

where $x_1, \dots, x_n$ are all the variables of $\tilde{D}_j$, $(u_1, v_1), \dots$ are all the pairs of corresponding end-nodes of $C[x]$ resp. $C[y]$ labelled (in d^C, i.e. by l^C) by formula variables and $p_1, \dots, r_1, \dots$ are L-terms obviously determined by the $(a+b)$-tuple $y_1, \dots, s_1, \dots$.

Since ε is suitable there are two possibilities:

either (a) $\varepsilon(\tilde{A}_{l(u_i)}) = \varepsilon(\tilde{A}_{l(v_i)}) = H_i$ and H_i does not, in particular, contain any of $x_1, \dots, x_n$,

or (b) $\varepsilon(\tilde{A}_{l(u_i)}) = l(u_i)$, $\varepsilon(\tilde{A}_{l(v_i)}) = l(v_i)$.

In case (a) we trivially have $H_i(x_1/p_1, \dots) = H_i(x_1/r_1, \dots) = H_i$. In case (b) we have: $l(u_i) = F_i(x_1/p_1, \dots)$ and $l(v_i) = F_i(x_1/r_1, \dots)$, where the formula F_i is determined by the formula C_j.

Thus we define:

$$B_j = E_j(x_1, \dots, x_n, G_1, \dots),$$

where G_i is either H_i or F_i according to which of the above cases occur for the particular i.

To show that the $(a+b+c)$-tuple $x_1, \dots, t_1, \dots, B_1, \dots$ lies in the domain of the schematic rule it is sufficient to go through the conditions defining the domain and to check that they are preserved. This is easy by the discussion of the possibilities above (and using the fact that ε is suitable). □

The intuition behind the next result is that the important information about a proof d is determined by the structure of the upper part of the forest $T(d)$. Compare the following two results with Theorem 2 of [17].

Theorem 2.1. *For any schematic system $\mathbb{A}$ there exists a constant $c > 0$ such that if d is an $\mathbb{A}$-proof with k steps, then there exists a sequence $B_1, \dots, B_k$ of*

L^*-formulas containing no metavariables or term variables and satisfying

(i) $\mathrm{dp}(B_i) \leqslant c \cdot k$, *for* $i \leqslant k$;

(ii) *any instance of* $B_1, \ldots, B_k$ *given by a suitable substitution is an* $\mathbb{A}$*-proof*; *and*

(iii) d *is also an instance of* $B_1, \ldots, B_k$ *given by a suitable substitution.*

Proof. Define $c' = \max\{\mathrm{dp}(A') , A'$ is an L^*-formula occurring in some schematic rule of $\mathbb{A}\}$. Thus c' depends only on $\mathbb{A}$. Then the basic part S_d of d fulfils: $|S_d| \leqslant 2^{c'+1} \cdot k$.

By the Technical Lemma there exists a filter $C \supseteq S_d$ satisfying the hypothesis of the lemma from Section 2 and

$$r(C) \leqslant |S_d| \leqslant 2^{c'+1} \cdot k.$$

By the lemma above d^C is the required sequence $B_1, \ldots, B_k$. Also $\mathrm{dp}(d^C) \leqslant 2^{c'+1} \cdot k$, since $\mathrm{dp}(d^C) = r(C)$.

To obtain d as an instance of d^C define the substitution ε by: $\varepsilon(\tilde{A}_{l(x)}) = l(x)$ (l is the label function of $T(d)$). So if we put $c = 2^{c'+1}$, we are done. □

Theorem 2.2. *For any schematic system* $\mathbb{A}$ *there exists a constant* $c > 0$ *such that for any L-formula B the following holds*:

If B has some $\mathbb{A}$*-proof with k steps, then B has also an* $\mathbb{A}$*-proof d satisfying*:

(i) *d has k steps*;

(ii) $\mathrm{dp}(d) \leqslant c \cdot k + \mathrm{dp}(B)$.

Proof. Let d be an $\mathbb{A}$-proof of B with k steps. Let $d^C = B_1, \ldots, B_k$ be a sequence of L^*-formulas constructed in Theorem 2.1.

Define a suitable substitution ε:

(i) if $\tilde{A}_{l(u)}$ is a label of end-node u (of $T(d^C)$) which is in relation R with some end-node of $T(B_k)$, then put $\varepsilon(\tilde{A}_{l(u)}) = l(u)$,

(ii) and put $\varepsilon(\tilde{A}_{l(u)}) = H$, for all other end-nodes u, where H is the formula $(y = y)$, y a variable occurring neither in d not in $V_{\mathbb{A}}$.

Since d^C does not contain metavaraibles or term variables, the above definition of the substitution is complete.

Now, if u is in relation R with some end-node v ocurring in $T(B_k)$, then clearly $\mathrm{dp}(l(u)) \leqslant \mathrm{dp}(l(v)) \leqslant \mathrm{dp}(B)$. Thus $\mathrm{dp}(\varepsilon(d^C)) \leqslant c \cdot k + \mathrm{dp}(B)$. □

Remark. In Section 6 we shall prove that the above result cannot be essentially improved, i.e. the bound cannot be less than some linear function.

A result similar to the following was proved independently by Pudlák (cf. [22]).

Theorem 2.3. *Let $\mathbb{A}$ be a schematic system, B an L-formula, d its shortest $\mathbb{A}$-proof, i.e. $|d|$ is minimal. Then:*

$$\mathrm{dp}(d) \leqslant \sqrt{2\,|d|\,(\max(\mathrm{dp}(B), c) + 1)},$$

where the constant c depends on $\mathbb{A}$ only.

Proof. Let A be a step in d and let us write:

$$\mathrm{dp}(A) = t \cdot \sqrt{|d|}, \quad \text{for some } t > 0.$$

Let S denote the forest $S_d \cup T(B)$, where B is the last step of d. Hence for some constant $c > 0$, depending only on $\mathbb{A}$, $r(S) \leqslant \max(c, \mathrm{dp}(B))$.

Let C be the filter on $T(d)$ constructed from S in the proof of the Technical Lemma. Since d is the shortest proof of B, necessarily $C = T(d)$; otherwise there would be a shorter proof of B—some instance of d^C.

It follows that for any $x \in T(A)$ there exists a node $x' \in S$ such that $T(d)[x] \simeq T(d)[x']$, and thus also $r_{T(d)}(x') \leqslant \max(c, \mathrm{dp}(B))$.

Define $c' = \max(c, \mathrm{dp}(B)) + 1$. Assume that $x_s, \ldots, x_0$ is a maximal sequence of nodes from $T(A)$ such that

(i) $x_1 <_{T(d)} x_{i+1}$, and
(ii) $r_{T(d)}(x_i) = (s - i) \cdot c'$.

So we have:

$$\text{(A)} \qquad s = \frac{t\sqrt{|d|}}{c'} - \frac{u}{c'}, \quad \text{for some } 0 \leqslant u < c$$

Let $x'_s, \ldots, x'_0$ be the nodes from S corresponding to $x_s, \ldots, x_0$ as above, i.e. $T(d)[x_i] \simeq T(d)[x'_i]$ and $r_{T(d)}(x'_i) < c'$.

Finally let P_j be a maximal path in $T(d)[x'_j]$ into which $x_j, \ldots, x_0$ are mapped by the isomorphism $T(d)[x_j] \simeq T(d)[x'_j]$, i.e. $P_j = y_0, \ldots, y_r$ is such that each y_{i+1} is a son of y_i and each $y_{(j-k)c'}$ is the image of x_k (so $y_0 = x'_j$). Hence:

$$r = j \cdot c' + 1 + u.$$

Now observe that for $i \neq j$, $x'_i \notin P_j$. Otherwise, since $r_{T(d)}(x'_i) < c'$, one would have $x'_i = y_w$ for some $w < c'$. So for the counterimage z of y_w it would hold that $x_j >_{T(d)} z >_{T(d)} x_{j-1}$. In particular, $z \neq x_i$. But also $T(d)[z] \simeq T(d)[x_i]$ and $z <_{T(d)} x_i$ or $x_i <_{T(d)} z$. This would be a contradiction.

This clearly implies that all $P_0, \ldots, P_s$ are mutually disjoint and thus:

$$|d| \geqslant \sum_{j=0}^{s} |P_j| = \sum_{j=0}^{s} (j \cdot c' + 1 + u).$$

By (A) then

$$
\begin{aligned}
|d| &\geqslant \Big(c' \sum_{j=0}^{s} j\Big) + (s+1)(1+u) \\
&\geqslant \frac{c'}{2}\Big(\frac{t\sqrt{|d|}}{c'} - \frac{u}{c'}\Big)\Big(\frac{t\sqrt{|d|}}{c'} - \frac{u}{c'} + 1\Big) \\
&\quad + \Big(\frac{t\sqrt{|d|}}{c'} - \frac{u}{c'} + 1\Big)(1+u) \\
&\geqslant \Big(\frac{t\sqrt{|d|}}{2} + \frac{u}{2} + 1\Big)\Big(\frac{t\sqrt{|d|}}{c'} - \frac{u}{c'} + 1\Big)
\end{aligned}
$$

Since $0 \leqslant u < c'$:

$$|d| \geqslant \frac{t^2\,|d|}{2c'}, \quad \text{i.e} \quad t \leqslant \sqrt{2c'}$$

We have

$$\mathrm{dp}(A) \leqslant \sqrt{2\,|d|(\max(c, \mathrm{dp}(B)) + 1} \qquad \sqcap$$

Remark. From the results of Pudlák [23] it follows that the best upper bound which can be proved in the above proposition cannot be better (i.e. smaller) than $c\sqrt{|d|}/\log|d|$ (for B of some bounded depth).

We sketch the argument (for the necessary details see [23]). We should note that systems used in [23] contain schematic rules defined using the condition "t is closed". We can allow this condition too—see the discussion in Section 0—since any subterm of a closed term is also closed.

Let Prf($\boldsymbol{n}$, $\ulcorner A \urcorner$) formalize the sentence: "A has an $\mathbb{A}$-proof of length $\leqslant n$". The symbol $\boldsymbol{n}$ denotes, in this remark only (cf. Section 4), the 'dyadic numeral' of n. Thus $|\boldsymbol{n}| = O(\log(n))$.

Let Con($\boldsymbol{n}$) be the formula $\neg$Prf($\boldsymbol{n}$, $\ulcorner x \neq x \urcorner$). Pudlák proved that for a suitable system $\mathbb{A}$ the shortest proof d_n of Con($\boldsymbol{n}$) (all Con($\boldsymbol{n}$)'s are $\mathbb{A}$-provable) has length $|d_n|$ at least $c_1 n/(\log n)^2$ (Theorem 2.9 of [23]).

Assume now that $F(x)$ is some upper bound from a proposition analogous to 2.3 (for B of a bounded depth). More precisely, assume that $F(x)$ is a non-decreasing function and that:

($*$) for any formula B such that $\mathrm{dp}(B) \leqslant \mathrm{dp}(\mathrm{Con}(x))$ and any shortest proof d of B it holds that $\mathrm{dp}(d) \leqslant F(|d|)$.

Then we disprove $\neg$Con($\boldsymbol{n}$) in $\mathbb{A}$ by the concatenation of $\mathbb{A}$-proofs of the following propositions (Sat$_k$ is the partial-truth definition for formulas of depth $\leqslant k$):

(i) Prf($\boldsymbol{n}$, $\ulcorner x \neq x \urcorner$) $\rightarrow$ "$x \neq x$ has a proof of depth $\leqslant F(n)$".

(ii) "$x \neq x$ has a proof of depth $\leqslant F(n)$" $\rightarrow$ $\mathrm{Sat}_{F(n)}(\ulcorner x \neq x \urcorner)$.

(iii) $\mathrm{Sat}_{F(n)}(\ulcorner x \neq x \urcorner) \rightarrow x \neq x$.

We assume $\mathbb{A}$ to be sufficiently strong to prove proposition (*) above; thus in particular (i) has an $\mathbb{A}$-proof of length $O(\log n)$.

According to Lemmas 3.4 and 3.5 of [23], the parts (ii) and (iii) have $\mathbb{A}$-proofs of length $O(F(n)^2)$.

Thus $\neg\mathrm{Con}(\boldsymbol{n})$ can be disproved in $\mathbb{A}$ by a proof of length $c_2(\log n + F(n)^2)$. So, by Pudlák's theorem:

$$c_1 \frac{n}{(\log n)^2} \leqslant c_2(\log n + F(n)^2).$$

From this it follows, for some $0 < c_3 < c_1/c_2$ and n sufficiently large, that:

$$c_3 \frac{n}{(\log n)^2} < F(n)^2.$$

We are done □

The following two results are easy observations which, however, are not without some interest.

Corollary 2.4. *Let $\mathbb{A}$ be some schematic system of first-order arithmetic in the language $\{0, 1, =, \leqslant, +, \cdot\}$. Let an L-formula B have an $\mathbb{A}$-proof with k steps which are all Σ_0-formulas (i.e. bounded). Then B has another $\mathbb{A}$-proof d with k steps which are all Σ_0-formulas and with depth $\mathrm{dp}(d) \leqslant c \cdot k + \mathrm{dp}(B)$, where the constant c depends only on $\mathbb{A}$.*

Proof. The same as the proof of 2.2. It is enough to observe that the whole construction of d^C preserves the arithmetical complexity of formulas, i.e. if some quantifier is bounded. □

Before we state the next proposition let us recall the result about lengthening of proof after cut-elimination (cf. [25]).

Let G be the first-order predicate calculus formulated in Gentzen style, i.e. 'more rules, less axioms' (cf. [8] or [29]). A G-proof is a tree labelled by sequents (we shall use $\mapsto$ as a 'sequent arrow') and its *cut-rank* is the maximal depth of a formula occurring in it as a cut-formula.

Define a useful function $2^y_x : 2^y_0 = y$ and $2^y_{x+1} = 2^{(2^y_x)}$.

Proposition (cf. [25]). *Let D be a G-proof of a sequent $\mapsto C$, k its height and r its cut-rank. Then $\mapsto C$ has another G-proof D' such that:*

(i) *D' has cut-free,*

(ii) *D' has height at most 2^{k+1}_{r+1}.*

Now we can state our observation. Let H be a Hilbert-style formulation of predicate calculus—say as in [26].

Corollary 2.5. *Let a formulae B have an H-proof with k steps. Then the sequent $\mapsto B$ has a G-proof D such that*:

(i) *D is cut-free,*

(ii) *D has the height at most 2^{ck}_{ck}, where the constant c depends on H and G onlv.*

Proof (sketch). Let B have an H-proof with k steps and let $d^C = B_1, \ldots, B_k$ be a sequence of L^*-formulas assured by Theorem 2.1. In particular, $\mathrm{dp}(B_i) \leqslant c_1 \cdot k$, c_1 some constant depending on H.

It is a straightforward task to transform d^C into a 'G-proof' of $\mapsto B_k$, i.e. to construct a tree D' labelled by a sequents of L^*-formulas (in D' would occur only subformulas of B_i's) such that any substitution which transforms d^C into an H-proof transforms D' into a G-proof.

Now we apply the cut-elimination procedure to D' to obtain a tree D labelled by sequents of L^*-formulas with the root labelled by an (end-) sequent $\mapsto B_k$ such that any substitution making D' a G-proof makes D a cut-free G-proof.

Tree D' has height at most $c_2 \cdot k$, c_2 a constant depending on H, G, and 'cut-rank' at most $c_1 \cdot k$. Thus by the above proposition the height of D is at most 2^{ck}_{ck} taking $c := 1 + \max(c_1, c_2)$. □

Let us finish this section noting that by using essentially the same ideas, results similar to 2.1, 2.2 and 2.3 can be proved also for some other formal systems which are not covered by our definition of schematic systems, e.g. Gentzen's sequential calculus. However, the new estimate obtained is generally not linear.

3. From the number of steps to the length

In this section we prove an upper bound to the length of a proof in terms of the number of steps in the proof.

A related result has been proved by Orevkov [15]. He has shown that there exists a proof-analysis (see below) such that the set of formulas provable using instances of this proof-analysis is *not* recursive (cf. also [10]). Hence the length of the proofs cannot be recursively bounded using the *size* of the proof-analysis and the *size* of the proved formula.

We prove our result only for special schematic systems, which we call *simple* (since they contain only simple terms). For general systems no bounds are known. In fact, the following seems to be open.

Problem. Given a schematic system, is there a recursive function $f(x, y)$ such that for any formula A, if A has a proof with $\leqslant k$ steps, then A has a proof of length $\leqslant f(k, |A|)$? (For related problems and some results see [11] and [10].)

Let us call any sequence $Z_1, \dots, Z_k$ of L^*-formulas together with information I which, for any $i \leqslant k$, gives $i_1, \dots, i_n < i$ and

$$\text{an } \mathbb{A}\text{-rule} \quad \frac{U_1, \dots, U_n}{U_0} \quad (n \geqslant 0)$$

a *proof-analysis.*

A proof $d = B_1, \dots, B_k$ is an instance of the proof-analysis iff there exists a substitution ε such that $B_i = \varepsilon(Z_i)$, $i \leqslant k$, and each B_i was inferred according to the information I, i.e.

$$\frac{B_{i_1}, \dots, B_{i_n}}{B_i} \quad \text{is an instance of the } \mathbb{A}\text{-rule} \quad \frac{U_1, \quad , U_n}{U_0}.$$

Let $\mathbb{A}$ be any schematic system with a finite language. $\mathbb{A}$ is *simple* iff it does not contain n-ary function symbols with $n > 1$ and contains at most one unary function symbol.

Define $t(B)$ to be the *maximal depth of a term in B*, i.e. the maximal length of a term in B minus 1 if the system is simple.

Theorem 3.1. *Let $\mathbb{A}$ be any simple schematic system and B any formula having $\mathbb{A}$-proof with k steps.*

Then B has an $\mathbb{A}$-proof d with k steps and length:

$$|d| \leqslant 2^{2^{c\,k+2\,\mathrm{dp}(B)}} \cdot t(B),$$

where the constant $c > 0$ depends only on $\mathbb{A}$.

Proof. Since the proof is rather long we shall divide it into several parts.

Part 1. Define two formulas A, B to be *similar, $A \sim B$* in symbols, iff they are identical when all terms and variables are omitted (the inductive definition is obvious). Similarly define two proofs d, d' to be *similar, $d \sim d'$*, iff they have the same number of steps and the corresponding pairs of formulas are similar.

A proof d from a particular $\sim$-class has a fixed number of quantifiers and maximal terms. Since all functions are at most unary, all terms contain at most one variable (this is not the sole essential use of the assumption that functions are at most unary). Thus the total number of variables in d is also fixed.

Assume that together with the $\sim$-class of d we have also a proof-analysis. Then, in order to specify a proof d, it is sufficient to specify a finite number of terms and variables. Since there are also only a finite number of orders of variables and a finite number of possible conditions on variables when defining domains of schematic rules, it is sufficient to specify which variables occur in which quantifiers and terms. There are, of course, infinitely many of such possibilities but only finitely many essentially different ones (e.g. formulas $x = y$ and $u = v$ are essentially the same).

Thus let us define: a *type of a proof* is given by:

(i) specifying its $\sim$-class,

(ii) proof-analysis,

(iii) for all quantifiers (resp. term) specifying the variable (resp. the variable or the constant) occurring in it.

We *assume* that all types considered are specified correctly, i.e., if a step is assumed to be an axiom, it has the particular form, if x is assumed to be of order i in a given scheme, then the variable x specified for it is of order i, etc. (Put otherwise, a type is correct iff there exists at least one proof of this type.)

Part 2. Without loss of generality assume that the language of $\mathbb{A}$ contains a unary function symbol f. By an *individual* we shall mean a variable (not necessarily of order 1) or a constant of L. $F = \{f\}^*$ is the set of all words over the alphabet $\{f\}$, including the empty word Λ and $^\frown$ denotes concatenation.

In this terminology we can represent any L-term t as an ordered pair $[\alpha, a]$, $\alpha \in F$ and a an individual. Obviously $[f^{(n)}, a]$ corresponds to $f(f(\cdots f(a)\cdots))$, where f appears n times.

In order to obtain a particular proof from its type, it remains to determine its (maximal) terms. Since all individuals are already given by the type, it is enough to specify the 'prefixes' of function symbols applied to them. In other words, we have to specify a finite number of words from F such that certain conditions will be fulfilled. (This is one important use of the assumption that L-functions are at most unary. Otherwise we would be faced with labelled trees instead of words.)

Now let us see which ones. Consider an example of the scheme:

$$\forall\bar{x}: \tilde{B}(f(\bar{x}))\rightarrow\tilde{B}(f(\bar{t})).$$

Then the corresponding maximal terms in the instances of $\tilde{B}(f(\bar{x}))$ and $\tilde{B}(f(\bar{t}))$ must have the forms $[\beta_0{}^\frown f, x]$ and $[\beta_0{}^\frown f{}^\frown\beta_1, a]$, where β_0 is given by the formula substituted for $\tilde{B}$ and $[\beta_1, a]$ is the term substituted for $\bar{t}$.

More generally, an occurence of a term-variable $\bar{t}$ in a scheme is always within an L^*-term of the form $[f^{(n)}, \bar{t}]$ and may additionally be within an argument of a formula-variable. (Here we use the assumption (given in Section 0) that term variables have no arguments.) Thus the general description of a maximal L-term occurring in an instance of a scheme is $[\beta_0{}^\frown u{}^\frown\beta_1, a]$, where β_0 originates from a formula variable in the scheme, $u \in F$ originates from a word in the scheme, and $[\beta_1, a]$ originates from a term variable in the scheme. (Note: each of β_0, u and β_1 may be empty.)

For a given type T of a proof (say $T = (X, I, A)$, where X is its $\sim$-class, I its proof-analysis and A its assignment of individuals) we shall construct the set of conditions as follows. Variables $\alpha_0, \alpha_1, \ldots, \beta_0, \beta_1, \ldots$ and constants $u, v, \ldots$ stand for words from F.

(i) For each occurrence of a maximal term in T (i.e. for a 'place' for it) introduce one variable $\alpha_0, \alpha_1, \ldots$

(ii) Each application of some schematic rule in T given by I determines a condition of the general form

$$\exists \beta_0, \beta_1, \qquad \bigwedge_j \alpha_{i_j} = \beta_{k_j} {}^\frown u_{l_j} {}^\frown \beta_{m_j}$$

as explained above.

In the example above introduce α_0, α_1, variables for maximal terms in the instance $A(f(x))$ and $A(f(t))$ resp., and form the condition

$$\exists \beta_0, \beta_1: \quad \alpha_0 = \beta_0 {}^\frown f \wedge \alpha_1 = \beta_0 {}^\frown f {}^\frown \beta_1 .$$

We shall state this as a lemma.

Lemma 1. *Let R be an $\mathbb{A}$-rule and*

$$\frac{A_1, \qquad , A_k}{A_0}$$

be an instance of it; let $t_1, \ldots, t_m$ be all maximal terms in $A_0, \ldots, A_k$. Then there exists a system S of finitely many conditions (the number is bounded in terms of R, k and m) of the form

$$\alpha_{i_j} = \beta_{k_j} {}^\frown u_{l_j} {}^\frown \beta_{m_j},$$

such that:

(i) *$i_j \leq m$ and the u_{l_j}'s are constant words given by $A_0, \ldots, A_k$,*

(ii) *the system S together with the conjunction $\alpha_1 = t_1 \wedge \cdots \wedge \alpha_m = t_m$ has a solution for the β_i's,*

(iii) *if S together with the conjunction $\alpha_1 = t'_1 \wedge \cdots \wedge \alpha_m = t'_m$, for any terms (i.e. their prefixes) $t'_1, \ldots, t'_m$ has a solution then replacing t_1 by $t'_1, \ldots, t'_m$ by t'_m in $A_0, \ldots, A_k$ we obtain an instance of the rule R.*

The proof proceeds by detailed inspection as outlined above and we leave it to the reader.

In fact, the bound will be correct for any simple system for which Lemma 1 holds true, not only of the systems whose rules are defined using conditions of type 1, . . . , 5 on p. 5 (e.g. the condition "t is closed", already mentioned, can be allowed).

Part 3. Let $\Omega_0(\boldsymbol{\alpha}, \boldsymbol{\beta})$ be the conjunction of all conditions above obtained for each application of some schematic rule determined by the proof-analysis I of the type T. Since we have assigned the variables α_i to the maximal terms in the whole proof globally, we may conclude that all instances of the schematic rules are properly linked together and not only individually correct.

Assume that a formula B has an $\mathbb{A}$-proof of type T. Let $\Omega(\boldsymbol{\alpha}, \boldsymbol{\beta})$ be the conjunction of $\Omega_0(\boldsymbol{\alpha}, \boldsymbol{\beta})$ with conditions of the form $\alpha_{r_i} = u_{s_i}$, where the α_{r_i}'s are all variables assigned to the maximal terms in the last formula of T and the u_{s_j}'s are the corresponding maximal terms from B.

Thus there exists an $\mathbb{A}$-proof of B of type T iff $\exists \boldsymbol{\alpha}, \boldsymbol{\beta}$: $\Omega(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

Now, for $u, v, w \in F$ evidently $u^\frown v = w$ iff $|u| + |v| = |w|$, where $|u|$ is the length of the word u. (This is another essential use of the assumption that there is at most one unary function symbol.) Hence the conditions above can be written in the form:

$$\exists b_0, b_1, \qquad \bigwedge_j a_{i_j} = b_{k_j} + |u_{l_j}| + b_{m_j}, \quad \bigwedge_j a_{r_j} = |u_{s_j}|$$

which we shall abbreviate also by $\exists \boldsymbol{a}, \boldsymbol{b}$: $\Omega(\boldsymbol{a}, \boldsymbol{b})$.

Part 4. Assume now that the condition $\Omega(\boldsymbol{a}, \boldsymbol{b})$ corresponds to the type T. If we were able to deduce some upper bound on the smallest solution $\boldsymbol{a}, \boldsymbol{b}$ of $\Omega(\boldsymbol{a}, \boldsymbol{b})$, we would also be able to deduce some upper bound on the length of the shortest proof of type T. This is done as follows.

Let c_1 *be the depth* of proofs in type T and c_2 *be the maximum arity* of the predicates of L. A formula of depth $\leqslant c_1$ has length $\leqslant 2^{c_1+1}c_2 h$, where h is the maximum length of a term in the formula.

If b is an upper-bound on some solution of $\Omega(\boldsymbol{a}, \boldsymbol{b})$ then, clearly, $h \leqslant b + 1$.

Part 5. Now we shall search for an upper bound b on some solution $\boldsymbol{a}, \boldsymbol{b}$ of $\Omega(\boldsymbol{a}, \boldsymbol{b})$.

We can rewrite Ω in a more standard form:

$$M \cdot \boldsymbol{x} = \boldsymbol{c},$$

where M is some integer matrix, $\boldsymbol{x}$ is the vector of a_i's and b_i's, and $\mathbf{c}$ is the vector of constants $|u_i|$'s.

This is a pleasant situation, since we can use a result proved by Papadimitriou [19, p. 320, Theorem 13.4].

Lemma 2 (Papadimitriou). *Let M be $m \times n$ integer matrix, $\boldsymbol{c}$ an m-vector, $p_1 = \max_{i,j} |M_{ij}|$ and $p_2 = \max_i |c_i|$.*

Then: if the equation $M \cdot \boldsymbol{x} = \boldsymbol{c}$ has a nonnegative integer solution $\boldsymbol{x}$, then there exists a nonnegative integer solution $\boldsymbol{y}$ such that $y_i \leqslant b$, $i = 1, \ldots, n$, where:

$$b = n(mp_1)^{2m+3}(1 + p_2).$$

For the particular case that we consider, $p_1 = 1$ and p_2 is $\max(c, t(B))$ where c is some constant given by $\mathbb{A}$ only. Thus we need only to estimate n (the number of a_i's and b_i's) and m (the number of equations in Ω). This is done as follows. We assume that the proofs of type T have k steps.

By Theorem 2.2 we may assume that a proof of type T has depth $\leqslant c\,k + \mathrm{dp}(B)$, hence it has $\leqslant k \cdot 2^{c\,k+\mathrm{dp}(B)}$ atomic formulas and so $\leqslant c_2 k \cdot 2^{c\,k+\mathrm{dp}(B)}$

maximal terms, i.e. the variables a_i. Evidently there are at most K times more variables b_i then a_i, for a constant K.

Thus for c possibly bigger,

$$n \leqslant 2^{c\,k+\mathrm{dp}(B)}.$$

The number of equations in Ω is estimated analogously:

$$m \leqslant 2^{c\,k+\mathrm{dp}(B)}$$

(again for c possibly slightly bigger).

Using Lemma 2 we can estimate:

$$b \leqslant 2^{c\,k+\mathrm{dp}(B)}(2^{c\,k+\mathrm{dp}(B)})^{2\cdot 2^{c\,k+\mathrm{dp}(B)}+3} \cdot (1+\max(c, t(B)))$$

By slightly increasing c this can be simplified to

$$b \leqslant 2^{2^{c\,k+2\,\mathrm{dp}(B)}}$$

Part 6. If B has a proof of type T (i.e. having k steps) it has a proof of the length

$$\leqslant 2^{c\,k+\mathrm{dp}(B)+} \quad c_2(b+1)k \qquad \text{(Part 4)}$$

By Part 5

$$b \leqslant 2^{2^{c\,k+2\,\mathrm{dp}(}}$$

Thus again slightly increasing c we obtain the required bound. □

Remark. The assumption that the term variables have no arguments affects the size of $\Omega(\boldsymbol{a}, \boldsymbol{b})$. If one allows term variables to have arguments, then essentially the same proof works but the bound is greater—there would be more variables and equations in $\Omega(\boldsymbol{a}, \boldsymbol{b})$ and the equations would have more members.

Corollary 3.2. *Under the same assumption as in Theorem* 3.1 *B has an* $\mathbb{A}$*-proof d of length*:

$$|d| \leqslant 2^{2^{c\,k+|B|}}$$

where $c > 0$ *depends only on* $\mathbb{A}$.

Notice that since the 'shorter proof d' constructed in Theorem 3.1 has the same proof-analysis as the original one, we cannot expect a similar result for a general case of schematic systems. This would contradict Orevkov's result.

Let us also stress the relation of our proof with

Kreisel's conjecture. This conjecture says: "Assume that there exists a positive integer k such that for any nonnegative integer n, $A(\boldsymbol{n})$ has a PA-proof with k

steps. Then $\forall x\colon A(x)$ is PA-provable”. (Here $\boldsymbol{n}$ is the numeral for n, i.e. the term $s(s(\cdots(0)\cdots))$, where s occurs n times.)

Using the same method as in the proof above (and, in fact, the same as in [17]) we can prove this conjecture for all *simple arithmetical* (cf. next section) schematic systems. We omit the details since they are essentially those of [17].

4. Application to finitistic consistency statements

In this section we shall use Theorem 3.1 for proving some lower bound on the number of steps in the proofs of finitistic consistency statements introduced by Friedman [4] and by Pudlák [22].

This will allow us to construct (in Section 5) for a system a formula $A(x)$ such that all its instances $A(\boldsymbol{n})$ are provable in the system but there is no common upper bound to the number of steps in the proofs.

The idea is this. For certain provable formulas lower bounds are known for the minimum length of their proofs (cf. [22]). For a simple system we can use Theorem 3.1 to compute from these bounds lower bounds on the number of steps in the proofs. This idea works for ‘simple arithmetical system’ (defined below).

For general schematic systems it seems that no lower bounds on the number of steps are known (cf. Problem in Section 3). An affirmative solution of Kreisel’s conjecture (cf. Section 3) would offer an elegant method: if $\forall x\colon A(x)$ is not provable, then either one of the $A(\boldsymbol{n})$’s is not provable or there is no common upper bound on the number of steps in proofs of the $A(\boldsymbol{n})$’s.

Let us call a schematic system $\mathbb{A}$ *simple arithmetical* iff:

(i) $\mathbb{A}$ is simple.

(ii) The language of $\mathbb{A}$ contains 0 and s (thus the successor is the only unary function symbol of $\mathbb{A}$).

(iii) $\mathbb{A}$ contains Robinson’s arithmetic Q (cf. [30]); that is there are three formulas $N(x)$, $U(x, y, z)$ and $V(x, y, z)$ such that U and V define addition and multiplication on the domain N and $\mathbb{A}$ proves (the translation of) all axioms of Q (i.e. $s(x)=s(y)\rightarrow x=y$, $s(x)\neq 0$, $x\neq 0\rightarrow \exists y\colon s(y)=x$, $x+0=x$, $x+s(y)=s(x+y)$, $x\cdot 0=0$ and $x\cdot s(y)=x\cdot y+x$) together with “U and V define total functions on ”.

It is clear that any simple arithmetical system is able to formalize its own syntax, in particular a provability predicate.

Let $\mathrm{Con}_{\mathbb{A}}(x)$ be a formula saying “there is no $\mathbb{A}$-proof of $0=s(0)$ whose length is at most x”.

In [22] a finitization of Gödel’s second incompleteness theorem is proved: “for any reasonable $\mathbb{A}$ containing Q and reasonable formula $\mathrm{Con}_{\mathbb{A}}(x)$ there exists $\varepsilon>0$ such that for any closed numerical term t the least $\mathbb{A}$-proof of $\mathrm{Con}_{\mathbb{A}}(t)$ has length at least t^{ε}” (for details see [22], cf. also [4]). Observe that this result is trivial for simple arithmetical systems since in that case the only closed numerical terms are numerals and already: $|\mathrm{Con}_{\mathbb{A}}(\boldsymbol{n})| > |\boldsymbol{n}| \geqslant n$.

Define the sequence of functions f_i, $i < \omega$, by: $f_0(x) = s(x)$ and $f_{i+1}(x) = f_i^{(x+2)}(x)$, where $f^{(y)}$ denotes the y-th iterate of f.

Call any system $\mathbb{A}$ *regular,* if it arises from some simple arithmetical system by adding finitely many function symbols for functions from the sequence f_i and natural axioms defining them.

We have chosen the particular f_i's since these are primitive recurisve, any primitive recursive function is majorized by some f_j and the system with terms built up from f_i's is reasonable in the sense of the informal statement above. We shall present, without details or proof, the following result:

Proposition 4.1 (Pudlák). *Let $\mathbb{A}$ be any regular schematic system which contains a simple arithmetical system $\mathbb{B}$. Then there exists a formula* $\mathrm{Con}_{\mathbb{A}}(x)$ *in the langauge of $\mathbb{B}$ (having the above explained sense) and $\varepsilon > 0$ such that for any closed term t of $\mathbb{A}$ and any $\mathbb{A}$-proof d of* $\mathrm{Con}_{\mathbb{A}}(t)$ *it holds that $|d| > t^{\varepsilon}$.*

(In [22] also an upper bound is proved. The results there are proved for first-order systems, but the general proof is essential the same.) Let us stress the obvious fact that any instance $\mathrm{Con}_{\mathbb{A}}(t)$ has an $\mathbb{A}$-proof.

We would like to combine Results 3.1 and 4.1 to obtain a lower bound on the number of steps in $\mathrm{Con}_{\mathbb{A}}(t)$. However, if the language of $\mathbb{A}$ contains at least two unary functions, then we cannot apply 3.1, and if it does not contain s we are not able to use the result for results of the next section since they deal with theories always containing the successor function. But, if the language of $\mathbb{A}$ contains s, then any term t has length $|t| \geqslant t$ ($\mathbb{A}$ is simple) so 4.1 gives no information. We shall overcome this difficulty by the following construction.

Instead of using terms for defining big numbers we shall define these by (short) formulas. So we need a formula $B(x, y)$ which defines some rapidly growing and provably total function. It is easy to observe that this can be done at least for primitive recursive functions (by formalizing their definitions) without any strong assumptions about $\mathbb{A}$. In particular, any primitive recurisive function is definable by a Σ_1-formula in a simple arithmetical system which proves (the translation of) induction axioms for all Σ_1-formulas.

Let $\mathrm{Con}_{\mathbb{A}}^{B}(x)$ be the formula $(\exists y\colon B(x, y) \wedge \mathrm{Con}_{\mathbb{A}}(y))$. Now, the Result 4.1 can also be proved using these 'definitions' of terms instead of terms themselves. Hence we state:

Proposition 4.2. *Let $\mathbb{A}$ be a simple arithmetical schematic system and let $B(x, y)$ define in $\mathbb{A}$ a primitive recursive function from the sequence f_i, say f. Then there exists an $\varepsilon > 0$ such that the least $\mathbb{A}$-proof of* $\mathrm{Con}_{\mathbb{A}}^{B}(\boldsymbol{n})$ *has length at least $(f(n))^{\varepsilon}$.*

Theorem 4.3. *Let $\mathbb{A}$ be a simple arithmetical schematic system which proves (the translation of) induction axiom for each Σ_1-formula. Let f be any primitive recursive function.*

Then there exist a formula $B(x, y)$ such that for each n: (i) *$\mathbb{A}$ does not prove* $\mathrm{Con}^B_{\mathbb{A}}(\boldsymbol{n})$ *in $\leqslant f(n)$ steps, while* (ii) $\mathrm{Con}^B_{\mathbb{A}}(\boldsymbol{n})$ *is $\mathbb{A}$-provable.*

Proof. Using 3.1 and 4.1. The length of $\mathrm{Con}^B_{\mathbb{A}}(\boldsymbol{n})$ is $\mathrm{O}(n)$.

Using the estimate from 3.2, for the given f, choose the definition $B(x, y)$ of some primitive recursive function from the sequence f_i, say f_j, such that

$$2^{2^{c(f(n)+n)}} < (f_j(n))^{\varepsilon}.$$

Σ_1-induction is sufficient for defining any primitive recursive function; c and ε depend only on $\mathbb{A}$. Finally, since $\mathrm{Con}_B\mathbb{A}(\boldsymbol{n})$ is a true Σ_1-sentence, it is $\mathbb{A}$-provable. (Alternative proof: enumerate all proofs of length at most $f_j(n)$ and verify that none of them is an $\mathbb{A}$-proof of $0 = s(0)$.) □

5. Some speed-up results

In this section we shall use the results of Section 4 for proving some unbounded speed-up's, namely between arithmetics of lower and higher order and between ZF and GB.

Let A_n denote the usual schematic system of the n-th order arithmetic where + and · are treated as relations. So the A_n's are simple arithmetical.

More specifically: A_1 is Peano's arithmetic formulated in the language having ternary relations "$x + y = z$" and "$x \cdot y = z$" instead of functions + and ·, A_{n+1} extends A_n by adopting variables of $(n + 1)$-th order and the full comprehension scheme for them. So the A_n's are simple arithmetical. (The exact formulation of the A_n's is not so important for the results below. What is important is that A_{n+1} proves the consistency of A_n.)

In 1936 Gödel announced a result (cf. [7]) that for any recursive function $g(x)$ and any n there exists a natural number k and a formula B such that B has an A_{n+1}-proof of length $\leqslant k$, B has an A_n-proof but the shortest such A_n-proof has length at least $g(k)$. It is not clear from the abstract that Gödel considered the length of the proof as the number of symbols, but Kreisel communicated to us that this was the case.

Later some related speed-up's were proved in [14], [2], [17], [27], We are going to prove a result in the spirit of [7]. The result is, in fact, that of [17] but the proof is new. (Parikh proved his result only for the particular case $n = 1$ but the method easily extends to the general case.)

Theorem 5.1. *For any $n \geqslant 1$ there exists a constant c such that for any m there is a* *formula C_m for which*

(i) *C_m has an A_{n+1}-proof with c steps,*

(ii) *C_m is A_n-provable,*

(iii) *any A_n-proof of C_m has at least m steps.*

Proof. Choose any primitive recursive function which grows faster than $s(x)$, say 2^x. By Theorem 4.3 we can choose a formula B already in the language of A_1 such that $\mathrm{Con}^B_{\mathrm{A}_n}(\boldsymbol{m})$ does not have an A_n-proof with $\leqslant 2^m$ steps.

We finish by observing that A_{n+1} proves the formula $\forall x\colon \mathrm{Con}^B_{\mathrm{A}_n}(\boldsymbol{m})$ is A_n-provable. □

Remark. The idea of another proof of the result is as follows.

Let a formula $\mathrm{Pr}(y, x)$ say: "formula x has an $\mathbb{A}$-proof with length $\leqslant y$" (cf. [22]). Using diagonalization construct a formula $A(x)$ such that for any $n < \omega$, $\mathbb{A}$ proves "$A(\boldsymbol{n}) \leftrightarrow \neg \mathrm{Pr}(2^{\boldsymbol{n}}_4, \ulcorner A(\boldsymbol{n})\urcorner)$". Using Theorem 3.1 show that each $A(\boldsymbol{n})$ has no $\mathbb{A}$-proof with n steps. But clearly $\forall x\colon \mathrm{Con}_{\mathbb{A}}(x)$ implies $\forall x\colon A(x)$.

We used the results of Section 4 since they seem to be of some independent interest and they also allow us to prove Proposition 5.3.

We shall finish this section by proving a speed-up result of GB over ZF analogous to 5.1. We shall use the result of Pudlák proved in [22].

Proposition 5.2 (Pudlák). *There exist formulas B_n, $n = 1, 2, \ldots$ and constants ε, $c > 0$ such that*:

(i) *each B_n has a* GB-*proof of length n^c,*

(ii) *each B_n, while* ZF-*provable, does not have a* ZF-*proof of length less than $(2^0_{\boldsymbol{n}})^\varepsilon$.*

(The B_n's are suitable translations of the formulas $\mathrm{Con}_{\mathrm{ZF}}(2^0_{\boldsymbol{n}})$.)

Proposition 5.3. *There exist* ZF-*formulas B_n, $n = 1, 2, \ldots$, and constants c_1, $c_2 > 0$ such that*:

(i) *B_n has a* GB-*proof with $n^{(c_1)}$ steps,*

(ii) *B_n is* ZF-*provable,*

(iii) *B_n does not have a* ZF-*proof with less than $2^0_{(n-c_2)}$ steps.*

Proof. Take the B_n's from 5.2; then (i) and (ii) are trivial.

(iii) follows from the preceding result 5.2 and from Theorem 3.1 □

6. One more application

Let us briefly recall the Paris–Harrington modification of the finite Ramsey theorem (for details see [20]). We fix some notation: $[x, y]$ denotes the set

$\{x, x+1, \ldots, y\}$, the number v is identified with the set $[0, v-1]$ and $X^{(n)}$ denotes the set of all n-element subsets of X.

The symbol $[x, y] \underset{*}{\to} (u)^w_v$ stands for the proposition: "for any function $f\colon [x, y]^{(w)} \to v$ there exists a set $H \subseteq [x, y]$ such that $u \leqslant |H|$, $\min(H) \leqslant |H|$ and f is constant on $H^{(n)}$".

Write PH(w) for $\forall xuv\, \exists y\colon [x, y] \underset{*}{\to} (u)^{w+1}_v$. The Paris–Harrington statement is the formula $\forall w$: PH(w). It is now widely known that this formula is not provable in PA (cf. [20]) but all of its instances PH($\boldsymbol{k}$) are provable. We shall provide an upper and a lower bound on the number of steps in the PA-proofs of PH($\boldsymbol{k}$)'s.

Theorem 6.1. *There exist constants $c_1, c_2 > 0$ such that for any natural number k:*

(i) PH($\boldsymbol{k}$) *does not have a* PA-*proof with $c_1 \cdot k$ steps,*

(ii) PH($\boldsymbol{k}$) *has a* PA-*proof with $c_2 \cdot k$ steps.*

Thus both upper and lower bounds are linear.

Proof. First we recall the result of Paris (cf. [18]) that $\mathrm{I}\Sigma_k$ does not prove PH($\boldsymbol{k}$), where $\mathrm{I}\Sigma_k$ is the fragment of PA with induction axioms only for Σ_k-formulas. Let d_k denote the shortest proof (in PA) of PH($\boldsymbol{k}$) and s_k the number of its steps.

The result above clearly implies $\mathrm{dp}(d_k) > k$. By Theorem 2.2 there exists a constant $c_0 > 0$ such that $\mathrm{dp}(d_k) \leqslant c_0 \cdot s_k$, hence $c_0^{-1} \cdot k < s_k$. This proves part (i).

Part (ii) is proved by a detailed inspection of the proof of PH($\boldsymbol{k}$) obtained by formalization of an infinite Ramsey theorem for k-tuples (an instance of it) and of König's lemma (an instance of it) (cf. [1]).

In some detail, $\neg$PH($\boldsymbol{k}$) implies (using a particular instance of König's lemma) the falsity of a particular instance of the infinite Ramsey theorem for $(k+1)$-tuples. Also there is a uniform method to prove a particular instance of the infinite Ramsey theorem for $(k+1)$-tuples from a particular instance of the infinite Ramsey theorem for k-tuples using a particular instance of König's lemma.

The uniformity of these proofs assures that they have a constant number of steps. Thus repeating the whole procedure k times we prove, from $\neg$PG($\boldsymbol{k}$), the falsity of a particular instance of the infinite Ramsey theorem for 1-tuples — this takes $c \cdot k$ steps. But any such instance is easily — and uniformly — provable in a constant number of steps. So, for some c, a contradiction can be derived from $\neg$PH($\boldsymbol{k}$) in $c \cdot k$ steps.

The details are left to the reader (cf. [20] and [1]). □

Now we use this result for proving that the Result 2.2 is in a sense optimal

Corollary 6.2. *Let K be a natural number such that $K \geqslant \mathrm{dp}(\mathrm{PH}(x))$. Let $F(x)$ be* *the function*:

$$F(k) := \min\{m \mid \textit{any PA-formula of depth} \leqslant K \textit{ which is PA-provable in } k \textit{ steps has a PA-proof with } k \textit{ steps and depth} \leqslant m\}.$$

Then there exists $c > 0$ *such that* $F(k) \geqslant c \cdot k$.

Proof. Use the d_k's from the preceding proof. We have $s_k \leqslant c_2 \cdot k$, i.e. there exist PA-proofs d'_k, $k = 1, 2, \ldots$ of PH(k) such that $\mathrm{dp}(d'_k) \leqslant F(c_2 \cdot k)$.

Also $k < \mathrm{dp}(d'_k)$, so $k < F(c_2 \cdot k)$; hence, $c_2^{-1} \cdot k < F(k)$. □

Notice that by the same argument Theorem 2.1 can also be proved to be optimal.

References

[1] P. Clote, Application of the low-basis theorem in arithmetic, Proc. of Recursion Theory Week at Oberwolfach (Springer, Berlin, 1985).

[2] A. Ehrenfeucht and J. Mycielski, Abbreviating proofs by adding new axioms, Bull. A.M.S. 77 (1971) 366–367.

[3] H. Friedman, One hundred and two problems in mathematical logic, J. Symbolic Logic 40 (1975) 113–129.

[4] H. Friedman, On the consistency, completeness and correctness problems, Manuscript, 1979.

[5] W.F. Farmer, Length of proofs and unification theory, Ph.D. thesis, Univ. of Wisconsin–Madison, 1984.

[6] R. Gandy, Limitations to mathematical knowledge, in: D. van Dalen, D. Lascar and J. Smiley, eds., Logic Colloquium, 80, (North-Holland, Amsterdam, 1982).

[7] K. Gödel, Ueber die Länge der Beweise, Ergenbnisse eines Mathematischen Kolloquiums, 1936.

[8] S.C. Kleene, Introduction to Metamatemathics (North-Holland, Amsterdam, 1952).

[9] H. Kotlarski, S. Krajewski and A.H. Lachlan, Construction of satisfaction classes for nonstandard models, Canad. Math. Bull. 24 (3) (1981).

[10] J. Krajíček and P. Pudlák, The number of proof lines and the size of proofs in first order logic, Arch. Math. Logic (1988) 69–84.

[11] J. Krajíček, Generalizations of proofs, Proc. 5th Easter Conf. on Model Theory, Wendisch-Rietz 1987, in: Seminarberichte, Humboldt-Univ., Berlin, (1987) 82–99.

[12] G. Kreisel and H. Wang, Some applications of formalized consistency proofs, Fund. Math. 42 (1955) 101–110.

[13] T. Miyatake, On the length of proofs in formal systems, Tsukuba J. Math. 4 (1980) 115–125.

[14] A. Mostowski, Sentences undecidable in formalized arithmetic (North-Holland, Amsterdam, 1952).

[15] V.P. Orevkov, Reconstruction of the proof from its scheme (in Russian), 7-th Conf. Math. Logic (1984) 133.

[16] R. Parikh, Existence and feasibility in arithmetic, J. Symbolic Logic 36 (1971) 494–508.

[17] R. Parikh, Some results on the length of proofs, Trans. A.M.S. 177 (1973) 29–36.

[18] J. Paris, A hierarchy of cuts in models of arithmetic, Lecture Notes in Math. (Springer, Berlin, 1980) 312–337.

[19] Ch. P. Papadimitriou and K. Steinglitz, Combinatorial Optimization (Prentice Hall, Englewood Cliffs, NJ 1982).

[20] J. Paris and L.A. Harrington, A mathematical incompleteness in Peano arithmetic, in: J. Barwise, ed., Handbook of Mathematical Logic, (North-Holland, Amsterdam, 1978) 1133–1142.

[21] P. Pudlák, Cuts, consistency statements and interpretations, J. Symbolic Logic 50 (1985) 423–441.

[22] P. Pudlák, On the length of proofs of finitistic consistency statements in first order theories, in: J.B. Paris et al., eds., Logic Colloquium, 84 (North-Holland, Amsterdam, 1984) 165–196.

[23] P. Pudlák, Improved bounds to the lengths of proofs of finitistic consistency statements, Proc. Appl. of Math. Logic to Finite Combinatorics, Arcata, 1985.

[24] K. Schüte, Proof Theory (Springer, Berlin, 1977).

[25] H. Schwichtenberg, Proof theory: Some applications of cut-elimination, in: J. Barwise, ed., Handbook of Mathematical Logic, (North-Holland, Amsterdam, 1978) 867–895.

[26] J.R. Shoenfield, Mathematical Logic (Addison-Wesley, Reading, MA, 1967).

[27] R. Statman, Speed-up by theories with infinite models, Proc. A.M.S. 81 (1981) 465–469.

[28] R. Statman, Proof-search and speed-up in the predicate calculus, Ann. Math. Logic 15 (1978) 225–287.

[29] G. Takeuti, Proof Theory (North-Holland, Amsterdam, 1975).

[30] A. Tarski, A. Mostowski and A. Robinson, Undecidable theories (North-Holland, Amsterdam, 1953).

[31] T. Yukami, A theorem on the formalized arithmetic with function symbols ′ and +, Tsukuba J. Math. 1 (1977) 195–211.

[32] T. Yukami, a note on formalized arithmetic with function symbols ′ and +, Tsukuba J. Math. 2 (1978) 69–73.

[33] T. Yukami, some results on speed-up, Ann. Japan Ass. Phil. Sci. 6 (4) (1984) 195–205.