Institute for Language, Logic and Information

AN OREY SENTENCE FOR PREDICATIVE ARITHMETIC

Marianne Kalsbeek

ITLI Prepublication Series
X-89-01

University of Amsterdam

Institute for Language, Logic and Information
Instituut voor Taal, Logica en Informatie

AN OREY SENTENCE FOR PREDICATIVE ARITHMETIC

Marianne Kalsbeek
Department of Mathematics and Computer Science
University of Amsterdam

Received January 1989

Master's Thesis, supervisors A. Visser and D.H.J. de Jongh

Correspondence to:

Faculteit der Wiskunde en Informatica
(Department of Mathematics and Computer Science)
Roetersstraat 15
1018WB Amsterdam

or

Faculteit der Wijsbegeerte
(Department of Philosophy)
Grimburgwal 10
1012GA Amsterdam

Contents

§0 Introduction 1
§1 A formalization of the notion tableau 10
§2 Search for an infinite branch in a systematic tableau 24
§3 A formalization of the model existence lemma for finite theories 37
§4 The model existence lemma for infinite theories 50
§5 A provability priciple for $I\Delta_0+\Omega_1$ 57
§6 An Orey sentence for $I\Delta_0+\Omega_1$ 61
References 65

Acknowledgements

I am grateful to Albert Visser, who provided the ideas for many of the theorems and proofs of this paper. I would like to thank Dick de Jongh, for his stimulating presence, and for introducing me to the subject of bounded arithmetic. I would like to thank also Michiel van Lambalgen, who spent a lot of effort in reading and criticizing the various drafts of this paper.

§0 Introduction

This master's thesis is a study in the field of bounded arithmetic. Systems of bounded arithmetics are relatively small subsystems of PA which contain at least

(1) basic axioms concerning the defining properties of 0, the successor function, addition and multiplication. Hence bounded arithmetics are extensions of Robinson's arihmetic Q.

In addition they may contain

(2) the scheme of induction restricted to Δ_0-formulae, i.e. formulae with only quantifiers of the form $\exists x \leqslant t$, $\forall x \leqslant t$ (with t a term in the language of the theory) or even a subclass of this class of formulae;

(3) defining axioms for $|x|$, which gives the length of the binary representation of x;

(4) defining axioms for the binary 'smash' function $\#$, where $x \# y = 2^{|x| \cdot |y|}$, or an equivalent of this function. (The relation $2^{|x| \cdot |y|} = z$ can be defined by a Δ_0-formula $\psi(x,y,z)$.)

(5) an axiom expressing the totality of exponentiation. (In Pudlák [83b] it is shown that the relation $x^y = z$ can be expressed by a Δ_0-formula.)

If the theory contains Δ_0-induction and axioms expressing the totality of $\#$ resp. exponentiation, then it proves $\Delta_0(\#)$- resp. $\Delta_0(\exp)$-induction.

Bounded arithmetic is interesting for various reasons.

In the first place, there are strong connections, which we shall not explore here, between bounded arithmetic and complexity theory.

Secondly, an interesting part of metamathematics and proof theory can be formalized in bounded arithmetic, for example, the incompleteness theorems are provable. Here the smash function plays an important role: it enables us to execute substitutions.

Moreover, bounded arithmetic is interesting from a philosophical point of view. In this paper we will prove some technical results which shed some light on a philosophical question concerning bounded arithmetic.

A vigorous and rather radical advocate of the philosophical advantages of bounded arithmetic (without exponentiation) over and above PA is Edward Nelson in his book Predicative Arithmetic [86]. His position is finitistic. Thus he does not accept the existence of the set of natural numbers. He considers exponentiation unacceptable, because it is an idealized construction. He rejects the induction principle (for formulae that are not Δ_0), because he considers it to be impredicative: "The induction principle assumes that the natural number system is given" (Nelson [86, p.1] and he goes as far as to doubt the consistency of PA. He proposes to work in theories that are interpretable in Q, which is very weak and does not contain induction. Bounded arithmetic consisting of (1), (2), (3) and (4) is interpretable in Q with methods initiated by Solovay and further developed by Wilkie and Pudlák (see Pudlák [83]). The interpretations involved are of a very simple type: they only involve relativization of quantifiers. Nelson baptizes theories that are interpretable by relativization in Q Predicative Arithmetics:

> "We would like to have have a formula A in the language of Q be a theorem of Predicative Arithmetic if and only if Q[A] is interpretable in Q. Perhaps this is possible, but I do not know the answer to the following *compatibility problem*: if Q[A] and Q[B] are interpretable in Q, then is Q[A,B] interpretable in Q?"

In this paper we will give a strong argument against the robustness of the concept of Predicative Arithmetic. We will show that there exists an Orey sentence for Q, i.e. a sentence G such that both Q+G and Q+¬G are interpretable in Q. We will show this by the following means. A substantial part of this paper will be devoted to the proof of various formalized versions of the model existence lemma for tableau provability in Buss's theory S^1_2. A simple application of two theorems of Paris and Wilkie [87] then provides us with the Orey sentence.

A theory is *tableau consistent* if none of the tableaux for this theory closes. The model existence lemma for tableau provability says: if a theory is tableau consistent then there is a model for this theory. This model can be constructed from a tableau for the theory.

In paragraph 1 we will give an intensional (in the sense of Feferman) formalization of the notion tableau in the theory S^1_2; here we will heavily rely on the arithmetizations developed in Buss [86].
In paragraph 2 we show how to construct initial segments of the leftmost consistent branch of a tableau. These initial segments provide us with an interpretation of the axioms of a finite relational theory A in S^1_2 plus the tableau consistency of A. It will be shown that this interpretation also serves to interpret the theorems of A.
In paragraph 3 we will then prove formalized versions of the results of paragraph 2.
In paragraph 4 it is shown that the results of the paragraphs 2 and 3 also hold in case the theory A is infinitely axiomatized.
In paragraph 5 we will use the results of the paragraph 3 and 4 to derive a provability principle of bounded arithmetic.
In paragraph 6, one of the results of paragraph 4 is used to construct an Orey sentence for bounded arithmetic, and we will dicuss there whether this constitutes a negative solution of the compatibility problem for bounded arithmetic.

The remainder of this introduction is devoted to a description of S^1_2. We will informally describe the predicates and functions defined in Buss [86] (modulo some minor modifications) that we will need for the formalizations.

S^1_2

is the theory containing
(1) the basic axioms concerning the definitions of 0, the successor function, addition, multiplication, $|x|$, $\lfloor\frac{1}{2}x\rfloor$ and $\#$; ($\lfloor\frac{1}{2}x\rfloor$ is the 'shift right' function; it is equal to the entier of $\frac{1}{2}x$. For a complete list of these axioms, see Buss [86, pp. 30,31])
(2) a weak type of induction for a restricted class of functions of the polynomial time hierarchy, viz. PIND for Σ^b_1-formulae.
PIND is the following type of induction:
$\varphi(0) \wedge \forall x\, (\varphi(\lfloor\frac{1}{2}x\rfloor) \rightarrow \varphi(x)) \rightarrow \forall x\, \varphi(x)$.
To determine the class in the polynomial time hierarchy to which a formula belongs the number of alternations of *bounded* quantifiers (i.e. quantifiers of the form $\exists x \leqslant y$, $\forall x \leqslant y$) of the formula is counted.

A formula is Δ_0^b if it contains only *sharply bounded* quantifiers, i.e. quantifiers of the form $\exists x \leqslant |y|$, $\forall x \leqslant |y|$.
A formula φ is Σ_1^b if it is Δ_0^b, or if it is of the form $\exists x < y\ \psi$, $\exists x \leqslant |y|\ \psi$, or $\forall x \leqslant |y|\ \psi$, where ψ is Σ_1^b.
A formula φ is Π_1^b if it is Δ_0^b, or if it is of the form $\forall x < y\ \psi$, $\exists x \leqslant |y|\ \psi$, or $\forall x \leqslant |y|\ \psi$, where ψ is Π_1^b.
And similarly for Σ_i^b and Π_i^b, for $i > 1$.
A formula is Δ_1^b with respect to S_2^1 if it is, provably in S_2^1, equivalent to some Σ_1^b-formula and to some Π_1^b-formula.

The LIND axioms are induction axioms of the following type:
$\varphi(0) \wedge \forall x\,(\varphi(x) \rightarrow \varphi(Sx)) \rightarrow \forall x\ \varphi(|x|)$.
S_2^1 proves LIND for Σ_1^b- and Π_1^b-formulae, and PIND for Π_1^b-formulae.
Moreover, S_2^1 proves the following type of minimalization for Δ_1^b-formulae φ, which we will call *inimization*:
$\exists x \leqslant |y|\ \varphi(x,y) \rightarrow \exists x \leqslant |y|\ (\varphi(x,y) \wedge \forall z < x\ \neg\varphi(z,y))$.
This follows easily with PIND:
Let φ be Δ_1^b.
Then $\psi(y) \equiv \exists x \leqslant |y|\ \varphi(x,y) \rightarrow \exists x \leqslant |y|\ (\varphi(x,y) \wedge \forall z < x\ \neg\varphi(z,y))$ is Δ_1^b, hence also Σ_1^b, so we can apply PIND on y.
Reason in S_2^1. Clearly $\psi(0)$.
Suppose $\psi(\lfloor \frac{1}{2}y \rfloor)$.
Suppose $\exists x \leqslant |y|\ \varphi(x,y)$. From the basic axioms we know that $|y| = |\lfloor \frac{1}{2}y \rfloor| + 1$, hence either $\exists x \leqslant |\lfloor \frac{1}{2}y \rfloor|\ \varphi(x,y)$ or $\varphi(|y|,y) \wedge \forall z < |y|\, \neg\varphi(z,y)$.
In the first case, the induction hypothesis provides us with a minimal x, in the second case |y| is minimal.

At first sight, LIND, PIND and inimization seem rather unwieldy. However, properties of numbers needed for the arithmetization of syntax, can be proven in S_2^1 by considering the binary representation of numbers. It turns out that PIND, which can be conceived of as going from a string of 0's and 1's of length n to a string of length n+1, (instead of going from a number to its successor, as is done normally in induction) is appropriate for proving the needed properties.

A function f is Σ^b_1-*definable* if it is defined by: $f(x)=y \equiv A(x,y)$,
where (1) A Σ^b_1, and

(2) $S^1_2 \vdash \forall x\, \exists y < t\, A(x,y)$ for some term t, and

(3) $S^1_2 \vdash \forall x\, \forall y\, \forall z\, (A(x,y) \wedge A(x,z) \rightarrow y=z)$.

Δ^b_1-predicates containing function symbols for Σ^b_1-definable functions are still Δ^b_1.

Buss shows that it is possible to define Δ^b_1-predicates and Σ^b_1-definable functions in S^1_2 in an inductive manner, by socalled p-inductive definitions. Predicates and functions defined in this way are intensionally correct: S^1_2 proves their properties.

We will now give an informal description of the Δ^b_1-predicates and Σ^b_1-definable functions we will frequently use in this paper. For precise definitions, see Buss [86, pp.37-50, pp.116-118, p.126]

In S^1_2 seqences can be coded, and the predicate Seq(x) expresses that x is a sequence. Addition of an element z to a sequence x is indicated by x*z. We get the sequence containing one element x by taking 0*x. Concatenation of two sequences x and y is written as x**y. Concatenation is more or less multiplication.

The number of elements of a sequence x is given by Len(x). Due to the special features of the coding, $|x|$ is much larger than Len(x).

There is a β-function for sequences: $\beta(0,x)$ gives the number of elements in the sequence x; $\beta(i,x)$ gives the i^{th} element of x, provided that $i \leqslant \mathrm{Len}(x)$; if $i > \mathrm{Len}(x)$ then $\beta(i,x)=x+1$. If we concatenate a new element to a sequence, then this will be the last element of the resulting sequence.

We tacitly assume that all sequences are UniqSeq (see Buss [86, p.49]), in order to have the following property of sequences:
$S^1_2 \vdash \mathrm{Seq}(x) \wedge \mathrm{Seq}(y) \wedge \forall i \leqslant \mathrm{Len}(x)\, (\beta(i,x)=\beta(i,y) \rightarrow x=y)$.
We will frequently use this property when we apply inimization to prove that two sequences which both satisfy a certain Δ^b_1-property, must be equal.

SubSeq(i,j+1,x) gives the subsequence of x which contains the i^{th} until the j^{th} element of x in the order in which they occur in x.
For reasons of readability, we will indicate codes with $\ulcorner\urcorner$ instead of names, for instance, we will write $\ulcorner(\urcorner$ instead of Buss's $\overline{\mathrm{LParen}}$, and even $\ulcorner(\exists\urcorner$ for $\ulcorner(\urcorner*\ulcorner\exists\urcorner$, etc.

If $\varphi(z,x,y)$ is Δ_1^b then we can Σ_1^b-define a function f(x) such that f(x) has as value the number of $z \leqslant |x|$ such that $\varphi(z,x,y)$. f will be denoted as $(\# z \leqslant |x|)\ \varphi(z,x,y)$.

Trees

A *tree* is coded by a sequence with two special symbols [and] which denote the structure of the tree:

```
 a
b  c
d
```

In the tree a[b[d]c], b and c are the direct successors, or *sons* of a, which is the *root* of this tree, and d is the only son of b. a is the *father* of b and c. d and c are the *leaves* of this tree. b is the first son of a, or the *sonposition* of b is 1, and the sonposition of c is 2. a has two sons, or the *valence* of a is 2, whereas the valence of b is 1 and the valence of d is 0. The *depth* of a is 0, the depth of b and c is 1, and the depth of d is 2.
[is coded by 0, and] is coded by 1;
a[b[d]c] is coded by the sequence $x=\langle a+2,\ 0,\ b+2,\ 0,\ d+2,\ 1,\ c+2,\ 1\rangle$.
The addition of the 2's is nessecary to be able to differentiate in the code of the tree between the symbols [and] and the *nodes*, which are by definition coded by a number $\geqslant 2$.
Because x is a sequence, we indicate the elements of x by their position in the sequence x. For example, for the nodes we have:
Father(3,x)=1, Father(7,x)=1, Father(5,x)=3;
Valence(1,x)=2, Valence(5,x)=0;
Depth(5,x)=2, (but also: Depth(6,x)=2);
SonPos(3,1,x)=1;
Further we have:
Leaf(i,x) $\equiv$ Valence(i,x)=0,
Rootβ(x)=a,
Depth(x) is the maximum of Depth(i,x).
If a tree x is non-branching, i.e. if the valences of all nodes is 1, or 0, then x has one leaf, namely, if Depth(x)=n, the $\mathrm{Len}(x) \dot{-} n^{th}$ element of x. For example: in a[b[c[d]]], which has depth 3, the 7th element, d, is the leaf.

Pairing function

In S^1_2 we can define the standard pairing function by

$P(x,y)=z \equiv 2z=(x+y)^2+3x+y$.

This is indeed the bijective pairing function, because

$S^1_2 \vdash \forall xy\, \exists! z\, P(x,y)=z$ and

$S^1_2 \vdash \forall z\, \exists! xy\, P(x,y)=z$.

So if we define

$\pi_1(z)=x \equiv \exists y\, P(x,y)=z$, and

$\pi_2(z)=y \equiv \exists x\, P(x,y)=z$,

then π_1 and π_2 are (Σ^b_1-defined) functions in S^1_2.

Canonical terms

Under the provability predicate (see below) we use canonical terms instead of the standardly used numerals. These are defined inductively by :

$I_0 = 0$

$I_{2k} = SS0 \cdot I_k$

$I_{2k+1} = SS0 \cdot I_k + S0$.

The advantage of these canonical terms over standard numerals $S^{(k)}0$ is that the length of the canonical term I_k is proportional to the length of k, whereas the length of a standard numeral $S^{(k)}0$ is proportional to k. The code of I_x is a Σ^b_1-definable function.

Formal system for predicate logic

In this paper we will use a different formal system from the one Buss uses. We have axiom schemes

$\varphi \to (\psi \to \varphi)$

$(\varphi \to (\psi \to \theta) \to ((\varphi \to \psi) \to (\varphi \to \theta))$

$(\neg\varphi \to \neg\psi) \to ((\neg\varphi \to \psi) \to \varphi)$

$\forall x\varphi(x) \to \varphi(t)$, t a term free for the variable x in $\varphi(x)$

$\forall x(\varphi \to \psi) \to (\varphi \to \forall x\psi)$, x not free in φ

$(\varphi \to \forall x\psi) \to \forall x(\varphi \to \psi)$, x not free in φ

equality axioms.

The derivation rules are Modus Ponens and generalization.

Provability predicates

We will use provability predicates as they are defined in Paris and Wilkie [87]. A proof will be a sequence of formulae. If the theory T is Σ_1^b-axiomatized, the provability predicate $Prov_T$ is Σ_1^b.

With this provability predicate, S_2^1 is Σ_1^b-complete :
if φ is Σ_1^b, then $S_2^1 \vdash \varphi(x) \rightarrow Prov_{S_2^1}(\ulcorner \varphi(I_x) \urcorner)$.
Also, Prov satisfies the Löb-conditions:
if T is Σ_1^b-axiomatized and $T \vdash S_2^1$, then

$T \vdash \varphi \Rightarrow S_2^1 \vdash Prov_T(\ulcorner \varphi \urcorner)$

$T \vdash Prov_T(\ulcorner \varphi \urcorner) \rightarrow Prov_T(\ulcorner Prov_T(\ulcorner \varphi \urcorner) \urcorner)$

$T \vdash Prov_T(\ulcorner \varphi \rightarrow \psi \urcorner) \rightarrow (Prov_T(\ulcorner \varphi \urcorner) \rightarrow Prov_T(\ulcorner \psi \urcorner))$.

Thus we also have, for φ is Σ_1^b,

$S_2^1 \vdash \varphi \rightarrow \psi \Rightarrow S_2^1 \vdash \varphi \rightarrow Prov_{S_2^1}(\ulcorner \psi \urcorner)$.

Cuts, Initials, Inductivity

A formula $\varphi(x)$, with x free in φ, is called *inductive* for the theory T if

$T \vdash \varphi(0) \wedge \forall x(\varphi(x) \rightarrow \varphi(x+1))$.

Note that if T contains induction for the class of formulae to which φ belongs, then the inductivity of φ implies $T \vdash \forall x \varphi(x)$.

A formula $\varphi(x)$, with x free in φ, is a *cut* for the theory T, if φ is inductive for T and closed under $\leqslant$, i.e. if

$T \vdash \varphi(0) \wedge \forall x(\varphi(x) \rightarrow \varphi(x+1)) \wedge \forall x (\varphi(x) \rightarrow \forall z \leqslant x (\varphi(z))$.

We will sometimes write $x \in \varphi$ instead of $\varphi(x)$ if φ is a cut.
If φ is inductive for T and T contains minimalization axioms for the class of formulae to which φ belongs, then φ is also a cut.

A formula $\varphi(x)$, with x free in φ is an *initial* for T if φ is a cut for T and φ is closed under + and ·, i.e. if

$$T \vdash \varphi(0) \wedge \forall x(\varphi(x) \rightarrow \varphi(x+1)) \wedge \forall x (\varphi(x) \rightarrow \forall z \leqslant x (\varphi(z)) \wedge \forall x \forall y (\varphi(x) \wedge \varphi(y) \rightarrow \varphi(x+y) \wedge \varphi(x \cdot y)).$$

With the methods initiated by Solovay (see, for instance, Pudlák [83a], Paris and Wilkie [87] or Nelson [86]), every cut can be closed under +, · and $\#$. There exist however cuts that cannot be closed under exponentiation (see Paris and Dimitracopoulos [82]).
We will indicate cuts and initials with capitals, for instance with I or J.

We will write, if I and J are cuts or initials for T,
$I \subset J$ if $T \vdash \forall x\,(I(x) \rightarrow J(x))$; and if $I \subset J$ we will also say: I is below J.

We will also consider the following theories:

$I\Delta_0 + \Omega_1$

is the theory containing Q, the induction scheme for Δ_0-formulae, and an axiom (indicated as Ω_1) expressing the totality of the function $x^{|x|}$. This function has the same growth rate as $\#$, and the functionality of one of them implies the functionality of the other. Therefore we can identify the two. $I\Delta_0 + \Omega_1$ is the system of Paris and Wilkie [87].
$I\Delta_0 + \Omega_1$ is interpretable in Q. (For proofs see Pudlák [83a], Paris and Wilkie [87].)
Clearly $I\Delta_0 + \Omega_1 \vdash S^1_2$.

$I\Delta_0 + EXP$

Is the theory containing Q, the induction scheme for Δ_0-formulae, and an axiom expressing the totality of exponentiation.
There is a large gap between $I\Delta_0 + \Omega_1$ and $I\Delta_0 + EXP$. $I\Delta_0 + EXP$ is not interpretable in Q, whereas $I\Delta_0 + \Omega_1$ is. Also $I\Delta_0 + EXP$ and $I\Delta_0 + \Omega_1 + Con(I\Delta_0 + \Omega_1)$ are interpretable into each other (see Visser [88]), even though $I\Delta_0 + EXP \nvdash Con(I\Delta_0 + \Omega_1)$ (see paragraph 5).

§1 A formalization of the notion tableau

In the construction of tableaux for finitely axiomatized relational theories we will proceed as follows:
A *tableau* is a finite tree in which all nodes are sequences coding finite sets of formulae. The root of a tableau is a node which contains axioms of the theory. Successors of a node X are constructed by applying one of the five tableau rules, respectively:

Definition 1.1

τ := for any formula $\neg\neg\varphi$ in X, φ may be added;
α := for any formula $\neg(\varphi \rightarrow \psi)$ in X, one may add φ and $\neg\psi$;
β_1 := for any formula $\varphi \rightarrow \psi$ in X, $\neg\varphi$ may be added;
β_2 := for any formula $\varphi \rightarrow \psi$ in X, ψ may be added;
γ := for any formula $\exists x\varphi$ in X, $\varphi(c[\exists x\varphi x])$ may be added;
δ := for any formula $\neg\exists x\psi$ in X, $\neg\psi(c[0])$ may be added, and for any formula $\exists x\varphi$ in X such that $\varphi(c[\exists x\varphi x])$ is in X, one may add $\neg\psi(c[\exists x\varphi x])$ to X.

If an immediate successor of a node X is the result of applying β_1 for a formula $\varphi \rightarrow \psi$, then X has a second immediate successor, which is the result of applying β_2 for $\varphi \rightarrow \psi$, i.e., the tableau splits in X (and vice versa: if X has a β_2-successor for a formula $\varphi \rightarrow \psi$, than X also has a β_1-successor for the same formula).
It is admissible for a node to be the same as its predecessor, but a node which is *closed*, i.e. contains an atomic formula and its negation, does not have successors.
A *systematic tableau* will be a tableau in which the rules defined above are applied in the following fixed order: τ, α, β_1/β_2, γ, δ. Moreover, these rules are applied to all formulae they can be applied to, except for the β rule which is applied to an appropriately chosen implication of the node to which β is applied, i.e. in a systematic tableau we apply the following set of rules:

Definition 1.2

τ := add, for *all* formulae $\neg\neg\varphi$ in X, φ;

α := add, for *all* formulae $\neg(\varphi \to \psi)$ in X, φ and $\neg\psi$;

β_1:= add, for a systematically chosen formula $\varphi \to \psi$ in X for which neither $\neg\varphi$, nor ψ is in X, $\neg\varphi$;

β_2:= add, for a systematically chosen formula $\varphi \to \psi$ in X for which neither $\neg\varphi$, nor ψ is in X, ψ;

γ :=add, for *all* formulae $\exists x\varphi$ in X, $\varphi(c[\exists x\varphi x])$;

δ := add, for *all* formulae $\neg\exists x\psi$ in X, and add for *all* $\exists x\varphi$ in X such that $\varphi(c[\exists x\varphi x])$ is in X, $\neg\psi(c[\exists x\varphi x])$ to X;
and add for *all* formulae $\neg\exists x\psi$ in X, $\neg\psi(c[0])$ to X.

Clearly, a systematic tableau is a tableau.
A tableau is *closed* if all its end nodes (leaves) are closed. A closed tableau from $A\cup\neg\varphi$ is a *tableau proof* of φ from A; such a tableau is also called a tableau proof of $\bot$ from $A\cup\neg\varphi$.
A theory A is *tableau consistent* if there are no proofs of $\bot$ from A.
For infinite theories we also admit successors resulting from the application of the rule

EX := add a finite number of axioms of A to X.

In a systematic tableau for an infinite theory after each application of a rule from the set τ, α, β_1/β_2, γ, δ, EX is applied in a systematic way. We will make this more precise in paragraph 3, when we discuss theories which have infinitely many axioms.

To prove the formalization of the model existence lemma in S^1_2, we arithmetize the notions of tableau and systematic tableau, using Δ^b_1-predicates and Σ^b_1-definable functions. We use the notations and conventions of Buss [86]. We will also use the following predicates: I(x,y), which expresses that x and y are sequences, and x is an initial subsequence of y; $y\subseteq_p x$, which expresses the sequence y to be a subsequence of the sequence x; $z\subseteq x$, which expresses z to be a subset of x; $z\in x$, which expresses z to be an element of the sequence coded by x; $z\in(i,w)$, which expresses that w is a tree of which the i^{th} element codes a sequence containing z as one of its elements; ORD(x), which

expresses x to be a sequence of which the elements are ordered according to size; Node(i,w), which expresses w to be a tree and $\beta(i,w)$ a node of w.
These predicates are defined as follows:

Definition 1.3

$I(x,y) \equiv \exists t \leqslant Len(y)\ (x=SubSeq(1,\ t+1,\ y))$

$y \subseteq_p x \equiv \exists i,j \leqslant Len(x)\ (y=SubSeq(i+1,\ j+1,\ x))$

$z \in x \equiv Seq(x) \wedge \exists i < Len(x)\ (z=\beta(i+1,x))$

$z \subseteq x \equiv Seq(z) \wedge Seq(x) \wedge \forall t(t \in z \rightarrow t \in x)$

$z \in (i,w) \equiv (Tree(w) \wedge i<Len(w) \wedge z \in (\beta(i+1,w) \dot{-} 2)$

$Node(i,w) \equiv Tree(w) \wedge i \neq 0 \wedge \beta(i,w) \geqslant 2$

$ORD(x) \equiv Seq(x) \wedge \forall i,j \leqslant Len(x)\ (\ 1 \leqslant i < j \leqslant Len(x) \rightarrow \beta(i,x) < \beta(j,x))$

Clearly, $I(x,y)$, $y \subseteq_p x$, $z \in x$ and $z \in (i,w)$ are Δ_1^b-predicates, and so are $\exists z(z \in (i,w) \wedge \varphi(z))$, $\forall z(z \in (i,w) \rightarrow \varphi(z))$, $\exists z(z \in x \wedge \varphi(z))$, and $\forall z(z \in x \rightarrow \varphi(z))$, if φ is Δ_1^b. Also $z \subseteq x$ is Δ_1^b.
We will also write $\forall z \in x\ \varphi(z)$ and $\exists z \in x\ \varphi(z)$ for $\forall z(z \in x \rightarrow \varphi(z))$ and $\exists z\ (z \in x \wedge \varphi(z))$; and we will also write $\exists z \in (i,w)\ \varphi(z)$ for $\exists z(z \in (i,w) \wedge \varphi(z))$ and $\forall z(i,w)\ \varphi(z)$ for $\forall z(z \in (i,w) \rightarrow \varphi(z))$.
We have the following lemma.

Lemma 1.4

$\forall z\ \forall x\ (ORD(x) \rightarrow \exists! y\ (ORD(y) \wedge \forall t(t \in y \leftrightarrow t \in x \vee t=z)))$

Proof

Suppose ORD(x).
If $z \in x$, then take $y=x$.
If $z \notin x$, then $\forall t \in x\ (t \leqslant z) \vee \exists i < Len(x)\ (\beta(i+1,x) > z)$.
In the first case, take $y=x * z$.
In the second case we have, by inimization, a minimal such i. Then take $y=(SubSeq(1,\ i+1,\ x) * z) ** SubSeq(i+1,\ Len(x)+1,\ x)$.
In both cases, y is ordered and unique. ⊠

We also define a function which gives us, in case x is an initial sequence of the sequence y, the part of y which comes after the last element of x:

$$\mathrm{Tail}(y,x)=z \equiv (I(x,y) \wedge z=\mathrm{SubSeq}(\mathrm{Len}(x)+1, \mathrm{Len}(y)+1, y)) \vee (\neg I(x,y) \wedge z=0)$$

Tail(y,x) is Σ_1^b-defined, and from the fact that SubSeq(·,·) is a function follows that Tail is a function.

We assume A to be a relational theory, L to be the language of A, L^+ to be L plus all special constants for existential formulae in L^+.

The special constant for an existential formula coded by x will be defined as (0 * 14) ** x ; we also admit the special constant (0 * 14 * 0) , which we need in case we deal with a branch of a tableau in which no existential formulae occur.

To be able to use special constants in formulae we simultaneously define predicates Term^+, AtForm^+, Form^+, and EForm, using Buss' theorem [86, p.119] on p-inductive definability of Δ_1^b-predicates. Form^+ differs from Buss' predicate Fmla in the following aspects: The only logical symbols occurring in Form^+ are $\exists$, $\neg$ and $\rightarrow$; Form^+ does not admit codes for $((\exists x \leqslant t)\varphi)$ but only codes for $((\exists x)(x \leqslant t \wedge \varphi))$ and for $((\exists x)\varphi)$; the reason for this is that in the construction of tableaux from A we are not interested in deciding which class of the polynomial hierarchy the formulae occurring in the tableaux belong to; Moreover, Term^+, AtForm^+, Form^+, and EForm allow the use of special constants in the construction of formulae and terms.

Because in our set-up we do not distinguish between different sorts of variables, we can use the codes Buss uses for the class of bounded variables to encode the relation symbols of L: we encode the relation symbol R_i $(i \geqslant 1)$ from L by $14+i\cdot 4$. We assume for simplicity that all R_i are unary, and define Rel(x) as $\exists t > 0$ $(x=14+4\cdot t)$. We will write Var(x) instead of FVar(x).

We simultaneously define in S^1_2 the unary Δ_1^b-predicates Term^+, AtForm^+, Form^+ and EForm by the following p-inductive definition:

Definition 1.5

(1) $\neg\mathrm{Term}^+(0)$.

(2) If Seq(x) and Len(x)=1 and Var(β(1,x)) then $\mathrm{Term}^+(x)$.

(3) If Seq(x) and β(1,x)=14 and
$\mathrm{EForm}(\mathrm{SubSeq}(2, \mathrm{Len}(x)+1, x)) \vee (\mathrm{Len}(x)=2 \wedge \beta(2,x)=0)$,
then $\mathrm{Term}^+(x)$.

(4) If x is not required to be Term^+ by the above conditions then x is not Term^+.

(5) $\neg\mathrm{AtForm}^+(0)$.

(6) If Rel(x) and $\mathrm{Term}^+(y)$ then
$\mathrm{AtForm}^+((0 * \ulcorner(\urcorner * x) ** (y * \ulcorner)\urcorner))$

(7) If x is not required to be AtForm^+ by the above conditions, then x is not AtForm^+.

(8) $\neg\mathrm{Form}^+(0)$

(9) If $\mathrm{AtForm}^+(x)$ then $\mathrm{Form}^+(x)$

(10) If $\mathrm{Form}^+(x)$ then $\mathrm{Form}^+((0 * \ulcorner(\neg\urcorner)**(x*\ulcorner)\urcorner))$

(11) If $\mathrm{Form}^+(x)$ and $\mathrm{Form}^+(y)$ then
$\mathrm{Form}^+((0 * \ulcorner(\urcorner) **(x * \ulcorner\rightarrow\urcorner) ** (y * \ulcorner)\urcorner))$

(12) If $\mathrm{Form}^+(x)$ and Var(z) then
$\mathrm{Form}^+(0 * \ulcorner(\,(\exists\urcorner * z * \ulcorner)\urcorner ** (x * \ulcorner)\urcorner))$

(13) If x is not required to be Form^+ by the above conditions then x is not Form^+

(14) If $\mathrm{Form}^+(x)$ and Var(z) then
$\mathrm{EForm}(0 * \ulcorner((\exists\urcorner * z * \ulcorner)\urcorner ** (x * \ulcorner)\urcorner))$

(15) If x is not required to be EForm by the above conditions then x is not EForm.

For legibility's sake, we stated this definition in a rather informal way. An industrious reader, curious for the formal statement of this definition, would readily observe that in fact two more predicates are simultaeously defined, namely one expressing x to be an implication, and one expressing that x is the negation of a $^+$formula, and that $\mathrm{Form}^+(x)$ is defined as: $\mathrm{Form}^+(x)$ iff x is an atomic formula ($\mathrm{AtForm}^+(x)$) or x is a negation of a $^+$formula or x is an implication of $^+$formulae or x is an existential formula. We will now give a list of definitions concerning formulae which we will need in the sequel. Each predicate defined in this list is Δ_1^b; moreover, of this list NOT(x) is equivalent (in S^1_2) to the predicate expressing x to be a negation which was defined in the p-inductive definition of $Form^+$ and IMP is equivalent to the predicate which expresses that its subject is an implication.

Definition 1.6

$NEG(x,y) \equiv Form^+(y) \wedge x=(0 * \ulcorner(\neg\urcorner) ** (y * \ulcorner)\urcorner)$

$NOT(x) \equiv \exists a \subseteq_p x\, (Form^+(a) \wedge NEG(x,a))$

$Pos(x)=y \equiv NEG(x,y) \vee (y=0 \wedge \neg Form^+(x))$

$DNeg(x,y) \equiv \exists a \subseteq_p y\, (NEG(x,a) \wedge NEG(a,y))$

$DN(x) \equiv \exists a \subseteq_p x\, (DNeg(x,a))$

$IMPL(x,y,z) \equiv Form^+(y) \wedge Form^+(z) \wedge x=(0 * \ulcorner(\urcorner) ** (y * \ulcorner\rightarrow\urcorner) ** (z * \ulcorner)\urcorner)$

$IMP(x) \equiv \exists a,b \subseteq_p x\, (IMPL(x,a,b))$

$NEGImp(x,y,z) \equiv \exists a \subseteq_p x\, (NEG(x,a) \wedge IMPL(a,y,z))$

$NIMP(x) \equiv \exists a,b \subseteq_p x\, (NEGImp(x,a,b))$

$NEForm(x) \equiv NOT(x) \wedge EForm(Pos(x))$

The importance of the observations made above is the following. Buss shows in [86], Theorem 2, pp. 123,124, that S^1_2 can prove theorems involving p-inductively defined predicates. Inspection of this theorem and of the formal version of Definition 1.5 shows that we can prove the following lemma:

Lemma 1.7 "Unique Reading Lemma"

1. $S^1_2 \vdash \forall x\, (Form^+(x) \rightarrow AtForm^+(x) \vee IMP(x) \vee NOT(x) \vee EForm(x))$
2. $S^1_2 \vdash \forall x\, (Form^+(x) \wedge AtForm^+(x) \rightarrow \neg IMP(x) \wedge \neg NOT(x) \wedge \neg EForm(x)))$
3. $S^1_2 \vdash \forall x\, (Form^+(x) \rightarrow Seq(x))$
4. $S^1_2 \vdash \forall x\, (Form^+(x) \rightarrow (\beta(1,x)=\ulcorner(\urcorner \wedge \beta(Len(x), x)=\ulcorner)\urcorner)$
5. $S^1_2 \vdash \forall x\, (Form^+(x) \rightarrow (\#t<Len(x))\, (\beta(t+1,x)=\ulcorner(\urcorner) = (\#t<Len(x))\, (\beta(t+1,x)=\ulcorner)\urcorner)$
6. $S^1_2 \vdash \forall x,a,b,y,z\, (IMPL(x,a,b) \wedge IMPL(x,y,z) \rightarrow a=y \wedge b=z)$
7. $S^1_2 \vdash \forall x,y,z\, (NEG(x,y) \wedge NEG(x,z) \rightarrow y=z)$.

It will be clear that 2. of this lemma is just one instance of the disjointness of the predicates AtForm$^+$(x), IMP(x), NOT(x), and EForm(x) that are provable in S^1_2.

We need the two last statements of this lemma for the following reason: for the application of the tableau rules α, β_1, and β_2, we need to be able to get $\neg\varphi$ and ψ from an implication $\varphi\rightarrow\psi$, and φ and $\neg\psi$ from $\neg(\varphi\rightarrow\psi)$. Define the following Δ^b_1-predicates:

Definition 1.8

$$\text{Con}(x)=y \equiv \exists a\subseteq_p x\,(\text{IMPL}(x,a,y)) \vee (\neg\text{IMP}(x) \wedge y=0)$$
$$\text{NAnt}(x)=y \equiv \exists a\subseteq_p x\,\exists b\subseteq_p y\,(\text{IMPL}(x,b,a) \wedge \text{NEG}(y,b)) \vee (\neg\text{IMP}(x) \wedge y=0)$$
$$\text{NConN}(x)=y \equiv \exists a,b\subseteq_p x\,(\text{NEGImp}(x,a,b) \wedge \text{NEG}(y,b)) \vee (\neg\text{NIMP}(x) \wedge y=0)$$
$$\text{AntN}(x)=y \equiv \exists a\subseteq_p x\,(\text{NEGImp}(x,y,a)) \vee (\neg\text{NIMP}(x) \wedge y=0)$$

These predicates readily give us Σ^b_1-definable functions Con(x), NAnt(x), NConN(x), NConN(x): existence is clear, and unicity is provided for by Lemma 1.7.

We can define a Δ^b_1-predicate SC(x) which expresses that its subject is a special constant and a binary Δ^b_1-predicate SpeCon(x)=y which expresses that y is the special constant belonging to the existential formula x if x is EForm, and y is 0 otherwise.

Definition 1.9

$$\text{SC}(x) \equiv \text{Seq}(x) \wedge \beta(1,x)=14 \wedge [(\text{Len}(x)=2 \wedge \beta(2,x)=0) \vee \text{EForm}(\text{SubSeq}(2, \text{Len}(x)+1, x))]$$
$$\text{SpeCon}(x)=y \equiv (\text{EForm}(x) \wedge y=\langle 14,x\rangle) \vee(\neg\text{EForm}(x) \wedge y=0)$$

Since it is easily seen that S^1_2 proves existence and uniqueness of SpeCon(x), we can consider SpeCon as a Σ^b_1-definable function.

We will also need the following Δ^b_1-predicates:

VarOc(y,i,x), which expresses that x is Form$^+$, and the i^{th} element of x is the variable y, and this is not inside a special constant, i.e. VarOc(y,i,x) describes an in some sense 'real' occurrence of y in x.

QVar(y,i,x) expresses that the variable y occurs on the i^{th} place in the formula x directly preceded by the code of an existential quantifier.
BVar(y,i,x) expresses that the variable y occurs bounded by an existential quantifier on the i^{th} place in the formula x.
FVar(x) expresses that the variable y occurs free on the i^{th} place in the formula x.
Deg(x)=y, expresses that either x is $Form^+$ and y is the number of logical symbols occurring in x, or x is not a $^+$formula and y is $|x|+1$.

Definition 1.10

$$\begin{aligned}
\mathrm{VarOc}(y,i,x) &\equiv \mathrm{Var}(y) \wedge \mathrm{Form}^+(x) \wedge i>0 \wedge \beta(i,x)=y \\
&\quad \wedge \neg\exists j,k\,(j<i<k\leqslant \mathrm{Len}(x) \wedge \mathrm{SC}(\mathrm{Subseq}(j+1,k+1,x)))) \\
\mathrm{QVar}(x) &\equiv \mathrm{VarOc}(y,i,x) \wedge \beta(i\dot{-}1,x)=\ulcorner\exists\urcorner \\
\mathrm{BVar}(y,i,x) &\equiv \mathrm{VarOc}(y,i,x) \wedge \neg\mathrm{QVar}(x) \\
&\quad \wedge \exists j,k\leqslant \mathrm{Len}(x)\,(j+3<i<k \wedge \mathrm{Form}^+(\mathrm{SubSeq}(j,k+1,x)) \\
&\qquad\qquad \wedge \beta(j+2,x)=\ulcorner\exists\urcorner \wedge \beta(j+3,x)=y))) \\
\mathrm{FVar}(y,i,x) &\equiv \mathrm{VarOc}(y,i,x) \wedge \neg\mathrm{QVar}(x) \wedge \neg\mathrm{BVar}(y,i,x)
\end{aligned}$$

$$\begin{aligned}
\mathrm{Deg}(x)=y &\equiv (\mathrm{Form}^+(x) \\
&\quad \wedge y=(\#t\leqslant \mathrm{Len}(x)) \\
&\qquad ((\beta(t,x)=\ulcorner\exists\urcorner \vee \beta(t,x)=\ulcorner\neg\urcorner \vee \beta(t,x)=\ulcorner\rightarrow\urcorner) \\
&\quad \wedge \neg\exists j,k\,(j<t<k\leqslant \mathrm{Len}(x) \wedge \mathrm{SC}(\mathrm{Subseq}(j,k,x))))) \\
&\quad \vee (\neg\mathrm{Form}^+(x) \wedge y=|x|+1)
\end{aligned}$$

Theorem 7 of Buss [86, p.46] shows that Deg(x) is a Σ_1^b-definable function which takes as its value the number of logical symbols occurring in x outside the special constants if x is $Form^+$, and takes value $|x|+1$ if x is not $Form^+$. Lemma 1.7 shows that

$$S_2^1 \vdash \forall x\,(\mathrm{Form}^+(x) \rightarrow (\mathrm{Deg}(x)=0 \leftrightarrow \mathrm{AtForm}^+(x)))$$

Now we are ready to define Δ_1^b-predicates expressing that a collection of formulae is the result of applying one of the systematic tableau rules $\tau,\alpha,\beta_1/\beta_2,\gamma,\delta$, to some other collection of formulae. An example: if x is a sequence of $^+$formulae, and y the result of application of τ to x, then y is a sequence of $^+$formulae which can be divided in two parts:

y contains x as an initial subsequence, and the other part (1) contains all ⌜φ⌝ for which (a) ⌜¬¬φ⌝ is in x and (b) ⌜φ⌝ is not in x; and (2) it contains these ⌜φ⌝ in order of size; (3) it contains nothing else.
For convenience we also define $\tau(i,j,w)$ and $\alpha(i,j,w)$, etc., which express that in case w is a tree with nodes i and j, τ resp. α is applied to node i, which results in node j.

Definition 1.11

$$\begin{aligned}\tau(x,y) \equiv\ & \mathrm{FormSeq}(x) \wedge \mathrm{FormSeq}(y) \wedge I(x,y) \\ & \wedge \mathrm{ORD}(\mathrm{Tail}(y,x)) \\ & \wedge \forall t \in x\, (t \in \mathrm{Tail}(y,x) \leftrightarrow \exists z \in x\, (\mathrm{DNeg}(z,t) \wedge t \notin x))\end{aligned}$$

$$\tau(i,j,w) \equiv \tau(\beta(i,w) \dot{-} 2, \beta(j,w) \dot{-} 2)$$

$$\begin{aligned}\alpha(x,y) \equiv\ & \mathrm{FormSeq}(x) \wedge \mathrm{FormSeq}(y) \wedge I(x,y) \\ & \wedge \mathrm{ORD}(\mathrm{Tail}(y,x)) \wedge \forall t \in \mathrm{Tail}(y,x)\, (t \notin x) \\ & \wedge \forall t \in x\, (\mathrm{NIMP}(t) \rightarrow \\ & \qquad \exists a,z \in y\, \exists b \subseteq_p t\ (\mathrm{NEGImp}(t,a,b)) \wedge \mathrm{NEG}(z,b)))^1 \\ & \wedge \forall z \in \mathrm{Tail}(y,x)\, \exists t \in x\, \exists a,b \subseteq_p t\, (\mathrm{NIMP}(t) \\ & \qquad \wedge [\mathrm{NEGImp}(t,z,b)^2 \vee (\mathrm{NEGImp}(t,a,b) \wedge \mathrm{NEG}(z,b))]))^1\end{aligned}$$

$$\alpha(i,j,w) \equiv \alpha(\beta(i,w) \dot{-} 2, \beta(j,w) \dot{-} 2)$$

Comment: ad 1. $t = \neg(a \rightarrow b)$ and $z = \neg b$;
ad 2. $t = \neg(z \rightarrow b)$;

In paragraph 2 we will show that $S^1_2 \vdash \forall x\, \exists! y\, \tau(x,y)$, and the like for the tableau rule α and the other tableau rules which we will define in this paragraph.
We define ternary Δ^b_1-predicates $\beta_1(x,y,t)$ and $\beta_2(x,y,t)$, in which t, if it is an implication occurring in x, is split up under the condition that neither the negation of its antecedent, nor its consequent occurs in x.

Definition 1.12

$$\begin{aligned}\beta_1(x,y,t) \equiv\ & \mathrm{FormSeq}(x) \wedge \mathrm{FormSeq}(y) \wedge I(x,y) \\ & \wedge [\{t \in x \wedge \mathrm{IMP}(t) \wedge [((\mathrm{NAnt}(t) \in x \vee \mathrm{Con}(t) \in x) \wedge x = y) \\ & \qquad \vee (\neg(\mathrm{NAnt}(t) \in x \vee \mathrm{Con}(t) \in x) \wedge \mathrm{NAnt}(t) \in y)]\} \\ & \vee \{(t \notin x \vee \neg \mathrm{IMP}(t)) \wedge x = y\}]\end{aligned}$$

$\beta_2(x,y,t) \equiv \mathrm{FormSeq}(x) \wedge \mathrm{FormSeq}(y) \wedge I(x,y)$
$\wedge\ [\{t\in x \wedge \mathrm{IMP}(t)\ \wedge [((\mathrm{NAnt}(t)\in x \vee \mathrm{Con}(t)\in x) \wedge x=y)$
$\vee\ (\neg(\mathrm{NAnt}(t)\in x \vee \mathrm{Con}(t)\in x) \wedge \mathrm{Con}(t)\in y)]\}$
$\vee\ \{(t\notin x \vee \neg\mathrm{IMP}(t)) \wedge x=y\}]$

We will use these definitions to define predicates $\beta_1(i,j,w)$ and $\beta_2(i,j,w)$ which express that the node j of a tree w is the systematic β_1- respectively β_2-successor of the node i of w, in the following sense. If w is a tree, consider the depth k of node i in w. There are unique a and z such that $k=a+5\cdot z$ and $a<5$. There is a unique t such that t is $\pi_1(z)$. Then $\beta_1(i,j,w)$ c.q. $\beta_2(i,j,w)$ is true iff β_1 c.q. β_2 is applied to $(\beta(i,w)\dot{-}2, \beta(j,w)\dot{-}2, t)$.
The reason that we define the β-rules in such a cumbersome way will be explained in paragraph 2.

Definition 1.13

$\beta_1(i,j,w) \equiv \mathrm{Tree}(w) \wedge \exists k \leqslant |w|\ \exists t \leqslant k\ (\ k=\mathrm{Depth}(i,w) \wedge t=\pi_1(\lfloor k\dot{-}2/5 \rfloor)$
$\wedge\ \beta_1(\beta(i,w)\dot{-}2, \beta(j,w)\dot{-}2, t))$

$\beta_2(i,j,w) \equiv \mathrm{Tree}(w) \wedge \exists k \leqslant |w|\ \exists t \leqslant k\ (\ k=\mathrm{Depth}(i,w) \wedge t=\pi_1(\lfloor k\dot{-}2/5 \rfloor)$
$\wedge\ \beta_2(\beta(i,w)\dot{-}2, \beta(j,w)\dot{-}2, t))$

To define Δ_1^b-predicates expressing the application of the rules γ and δ we need to be able to talk about the variable which is bound by the outermost existential quantifier in an existential formula. We define such variables by the Σ_1^b-definable function EVar :
$\mathrm{EVar}(x)=y\ \equiv (\mathrm{EForm}(x) \wedge y=\beta(4,x)) \vee (\neg\mathrm{EForm}(x)\ \wedge y=0)$.
Moreover, we need a function which gives us φ in the formula $((\exists x)\varphi)$.
If we define the binary Δ_1^b-predicate
$B(x)=y \qquad \equiv y=\mathrm{SubSeq}(6, \mathrm{Len}(x), x)$,
then an easy verification shows that B(x) is the S_2^1-definable function we looked for.
We also need a S_2^1-defined substitution function Sub (slightly different from the substitution function Buss defines in [86, p.130]) which replaces all free occurrences of a variable in a formula by a sequence, for instance by a special constant. We take for Sub(v,x,z) the function that satisfies the following predicate:

Definition 1.14

$$
\begin{aligned}
Sub(v,x,z)=y \equiv \{&Seq(v) \wedge Var(x) \wedge Form^{+}(z)\\
&\wedge [([\exists i<Len(z)\,(FVar(x,i,z)\\
&\quad \wedge Len(y) = Len(z) +\\
&\qquad + (Len(v)\dot{-}1)\cdot(\#i<Len(z))(FVar(x,i,z)\\
&\quad \wedge \forall j<Len(z)\,\exists k<Len(y)\,(k=j+\\
&\qquad + (Len(v)\dot{-}1)\cdot(\#i<j)(FVar(x,i,z)\\
&\qquad \wedge (\beta(j+1,z)\neq x \vee BVar(x,j+1,z) \vee QVar(x,j+1,z)\\
&\qquad\qquad \rightarrow \beta(k+1,y)=\beta(j+1,z))\\
&\qquad \wedge (\beta(j+1,z)=x \wedge FVar(x,j+1,z)\\
&\qquad\qquad \rightarrow \forall t<Len(v)\,(\beta(t+k+1,y)=\beta(t+1,v))))]\\
&\vee (\neg\exists i<Len(z)\,(FVar(x,i,z) \wedge y=z)]\}\\
&\vee \{(\neg Seq(v) \vee \neg Var(x) \vee \neg Form^{+}(z)) \wedge y=0\}
\end{aligned}
$$

S^1_2 proves uniqueness and existence of Sub(v,x,z); moreover,

$$S^1_2 \vdash (Var(v) \vee SC(v)) \wedge Var(x) \wedge Form^{+}(z) \rightarrow Form^{+}(Sub(v,x,z))$$

Now we are able to describe the γ and δ rule:

Definition 1.15

$$
\begin{aligned}
\gamma(x,y) \equiv\ & FormSeq(x) \wedge FormSeq(y) \wedge I(x,y)\\
&\wedge ORD(Tail(y,x)) \wedge \forall t\in Tail(y,x)\,(t\notin x)\\
&\wedge \forall z\in x\,(EForm(z) \rightarrow \exists t\in y\,(t=Sub(SpeCon(z), EVar(z), B(z))))\\
&\wedge \forall t\in Tail(y,x)\,\exists z\in x\,(EForm(z)\\
&\qquad \wedge t=Sub(SpeCon(z), EVar(z), B(z))))
\end{aligned}
$$

$$
\begin{aligned}
\delta(x,y) \equiv\ & FormSeq(x) \wedge FormSeq(y) \wedge I(x,y)\\
&\wedge ORD(Tail(y,x)) \wedge \forall t\in Tail(y,x)\,(t\notin x)\\
&\wedge \forall z\in x\,\forall v\in x\,(NEForm(z) \wedge EForm(v)\\
&\qquad \rightarrow [\exists w,z\in x\,(w=Sub(SpeCon(v), EVar(v), B(v))\\
&\qquad \rightarrow \exists t\in y\,\exists s\subseteq_p t\,(s=Sub(SpeCon(v), EVar(Pos(z)), B(Pos(z)))\\
&\qquad\qquad \wedge NEG(t,s))))])\\
&\wedge \forall z\in x\,[NEForm(z)\\
&\qquad \rightarrow \exists t\in y\,\exists s\subseteq_p t\,(s=Sub(SpeCon(0), EVar(Pos(z)), B(Pos(z)))
\end{aligned}
$$

$$\land \text{NEG}(t,s))]\}$$

$$\land \{\forall t \in \text{Tail}(y,x)$$
$$\{\exists s \subseteq_p t\ ([\exists z \in x\ \exists v \in x\ \exists w \in x\ (\text{NEForm}(z) \land \text{EForm}(v)$$
$$\land\ w = \text{Sub}(\text{SpeCon}(v), \text{EVar}(v), B(v))$$
$$\land\ s = \text{Sub}(\text{SpeCon}(v), \text{EVar}(\text{Pos}(z)), B(\text{Pos}(z)))$$
$$\land\ \text{NEG}(t,s))]$$
$$\lor\ [\exists z \in x\ (\text{NEForm}(z) \land s = \text{Sub}(\text{SpeCon}(0), \text{EVar}(z), B(z))$$
$$\land\ \text{NEG}(t,s)])\})\}$$

$$\delta(i,j,w) \equiv \delta(\beta(i,w) \dot{-} 2, \beta(j,w) \dot{-} 2)$$

Furthermore we define when a sequence of formulae is closed or open:

Definition 1.16

$$\text{Closed}(x) \equiv \text{FormSeq}(x) \land \exists z,u \in x\ (\text{AtForm}^{+}(z) \land \text{NEG}(u,z))$$
$$\text{Closed}(i,w) \equiv \text{Tree}(w) \land \text{Closed}(\beta(i,w) \dot{-} 2)$$
$$\text{Open}(x) \equiv \text{FormSeq}(x) \land \neg\text{Closed}(x)$$
$$\text{Open}(i,w) \equiv \text{Tree}(w) \land \text{Open}(\beta(i,w) \dot{-} 2)$$

Using these definitions we now are able to construct a predicate adequately expressing x to be a systematic tableau from a theory A.
We assume A to be axiomatized by finitely many closed formulae. Let $t_1, \ldots, t_k$ be the codes for the axioms of A. We take the Δ_1^b-predicate A(x) as follows:

Definition 1.17

$$A(x) \equiv \text{Seq}(x) \land \text{Len}(x) = k \land \forall i \leq k\ (i \neq 0 \rightarrow \beta(i,x) = t_i).$$

I.e., A(x) expresses x to be a sequence which contains the axioms of A in a certain fixed order and contains nothing else. Of course, there is a standard number N such that A(N) and S^1_2 proves that there is exactly one x such that A(x).
A systematic tableau is a tree of which the root codes x such that A(x), i.e. the root is x+2, and which satisfies some further requirements.
We assume $\mathcal{A}(x)$ to be a Δ_1^b-formula adequately expressing x to be an axiom of A, i.e. we take $\mathcal{A}(x)$ provably equivalent (in S^1_2) to $x = t_1 \lor \ldots \lor x = t_k$.

In a systematic tableau the (systematic) rules are applied in fixed order: τ, α, β_1/β_2, γ, δ. So τ is applied to all nodes which occur on a depth in the tree that is 0 mod(5), etc. The β-rules are applied as follows: if x is a node with depth $2+5\cdot P(t,v)$, then split t if t is an implication in x.

Definition 1.18

$$\begin{aligned}
STab_A(x) \equiv\ & Tree(x) \\
& \wedge \forall i \leqslant Len(x)\ (Node(i,x) \rightarrow (Leaf(i,x) \vee Valence(i,x) \leqslant 2) \\
& \qquad \wedge FormSeq(\beta(i,x) \dot{-} 2) \\
& \qquad \wedge Depth(i,x) = 0 \rightarrow A(\beta(i,x) \dot{-} 2) \\
& \qquad \wedge Closed(i,x) \rightarrow Leaf(i,x)) \\
& \wedge \forall i,j \leqslant Len(x)\ ((Open(i,x) \wedge \neg Leaf(i,x) \wedge Father(j,x) = i) \rightarrow \\
& \quad [(Depth(i,x) = 0 \bmod(5) \wedge Valence\ (i,x) = 1 \wedge \tau(i,j,x)) \\
& \quad \vee (Depth(i,x) = 1 \bmod(5) \wedge Valence\ (i,x) = 1 \wedge \alpha(i,j,x)) \\
& \quad \vee (Depth(i,x) = 2 \bmod(5) \wedge Valence\ (i,x) = 2 \wedge SonPos(j,i,x) = 1 \\
& \qquad \wedge \exists k \leqslant Len(x)\ (k > j \wedge Father(k,x) = i \\
& \qquad\qquad \wedge \beta_1(i,j,x) \wedge \beta_2(i,k,x))) \\
& \quad \vee (Depth(i,x) = 3 \bmod(5) \wedge Valence\ (i,x) = 1 \wedge \gamma(i,j,x)) \\
& \quad \vee (Depth(i,x) = 4 \bmod(5) \wedge Valence\ (i,x) = 1 \wedge \delta(i,j,x))]
\end{aligned}$$

Clearly the definition of the predicate expressing its subject to be a (possibly non-systematic) tableau from the theory A will very much resemble the definition of $STab_A$. There are three differences: (1) the root of a non-systematic tableau need not contain all axioms of A, (2) the tableau rules need not be applied in fixed order, and (3) the tableau rules need not be applied to all formulae they can usefully be applied to, or, in the case of the β-rules, need not be applied to the smallest implication. We will not bother to exactly define the predicates needed to express application of a non-systematic tableau rule, but we will indicate them in bold face.

Definition 1.19

$$
\begin{aligned}
Tab_A(x) \equiv\ & Tree(x) \\
& \wedge \forall i \leqslant Len(x)\ (Node(i,x) \rightarrow [(Leaf(i,x) \vee Valence(i,x) \leqslant 2) \\
& \qquad \wedge\ FormSeq(\beta(i,x) \dot{-} 2) \\
& \qquad \wedge\ Depth(i,x) = 0 \rightarrow \forall z \in (i,x)\ (\mathcal{A}(z)) \\
& \qquad \wedge\ (Closed(i,x) \rightarrow Leaf(i,x))]) \\
& \wedge \forall i,j \leqslant Len(x)\ (Open(i,x) \wedge \neg Leaf(i,x) \wedge Father(i,j,x) \rightarrow \\
& \quad [Valence\ (i,x)=1 \wedge (\tau(i,j,x) \vee \alpha(i,j,x) \vee \gamma(i,j,x) \vee \delta(i,j,x))] \\
& \vee [Valence\ (i,x)=2 \wedge SonPos(j,i,x)=1 \\
& \wedge \exists k \leqslant Len(x)\ (k > i \wedge Father(k,x)=i \wedge (\beta_1(i,j,x) \wedge \beta_2(i,k,x))])
\end{aligned}
$$

We also define:

$$
\begin{aligned}
ClTab_A(x) &\equiv Tab_A(x) \wedge \forall i \leqslant Len(x)\ (Leaf(i,x) \rightarrow Closed(i,x)) \\
\nabla A &\equiv \neg \exists x\ ClTab_A(x)
\end{aligned}
$$

In the sequel we will sometimes omit the subscript A if it is clear that the theory concerned is A.

We will also use the predicate $Tab(x,y)$ for which we use the following modification of the definition of $Tab_A(x)$: substitute in the definition of $Tab_A(x)$ the clause $Depth(i,x)=0 \rightarrow \forall z \in (i,x)\ (\mathcal{A}(z))$ by the clause $Depth(i,x) = 0 \rightarrow \forall z \in (i,x)\ (z \in y)$, and add a clause expressing that y is a sequence of closed $^+$formulae. Accordingly, the predicate $ClTab(x,y)$ can be defined, and we will use the notation ∇y for $\neg \exists x\ ClTab(x,y)$.

§ 2 Search for an infinite branch in a systematic tableau

An infinite branch in the systematic tableau for a tableau-consistent finite theory can be found in the following way: Start with the root of the systematic tableau, which is tableau-consistent by hypothesis; if n nodes are already chosen, take for the n+1st node a tableau-consistent successor of the nth node. It is not difficult to see that if a tableau rule is applied to a tableau-consistent node, then at least one of its direct successors is tableau-consistent. If we also demand that in every step the leftmost tableau-consistent successor is chosen, then this procedure gives us the leftmost infinite branch with only tableau consistent nodes from the tableau.
We could try to execute this procedure in the following way:
For every n, take the fully developed tableau up to level n, this is the systematic tableau in which every node that is not closed or on depth n has a successor; then take the leftmost branch in this tableau which has only tableau-consistent nodes. These branches will fit into each other, thus giving us the leftmost infinite branch. The fully developed systematic tableau can be defined as follows:

$$\mathrm{FullTab}(x,n) \equiv \mathrm{STab}(x) \wedge \forall i \leqslant \mathrm{Len}(x)\,(\mathrm{Depth}(i,x) \leqslant n \wedge (\mathrm{Leaf}(i,x) \rightarrow \mathrm{Closed}(i,x) \vee \mathrm{Depth}(i,x)=n)).$$

We will follow a different procedure, in which we construct initial segments of the infinite leftmost tableau-consistent branch, without reference to the fully developed tableaux. The first step consists in taking the root of the systematic tableau. If a branch of depth n has been constructed, then to get the branch of depth n+1 take the leftmost systematic tableau-consistent successor of the leaf of this branch, that is, apply to this leaf the systematic tableau rule which 'belongs to' n mod(5), add the result of this application to the branch of depth n, on the understanding that, if n mod(5)=2, we add the β_1-successor if it is tableau consistent, otherwise we add the β_2-successor.
In the definition of these initial segments of the leftmost tableau-consistent branch we use the folowing Δ_1^b-predicate, which expresses its subject to be a branch of the systematic tableau from A:

Definition 2.1

$$
\begin{aligned}
Bra(x) \equiv\ & Tree(x) \\
& \wedge \forall i \leqslant Len(x)\ (Node(\beta(i,x) \rightarrow \\
& \qquad [FormSeq(\beta(i,x) \dot{-} 2) \\
& \qquad \wedge (Depth(i,x) = 0 \rightarrow A(\beta(i,x) \dot{-} 2)) \\
& \qquad \wedge (\neg Leaf(i,x) \rightarrow Valence(i,x) = 1) \\
& \qquad \wedge (Closed(i,x) \rightarrow Leaf(i,x))]) \\
& \wedge \forall i,j \leqslant Len(x)\ (Node(i,x) \wedge Father(j,x) = i \rightarrow \\
& \qquad [(Depth(i,x) = 0 \bmod(5) \wedge \tau(i,j,x)) \\
& \qquad \vee (Depth(i,x) = 1 \bmod(5) \wedge \alpha(i,j,x)) \\
& \qquad \vee (Depth(i,x) = 2 \bmod(5) \wedge (\beta_1(i,j,x) \vee \beta_2(i,j,x))) \\
& \qquad \vee (Depth(i,x) = 3 \bmod(5) \wedge \gamma(i,j,x)) \\
& \qquad \vee (Depth(i,x) = 4 \bmod(5) \wedge \delta(i,j,x))])
\end{aligned}
$$

We can now define, using this definition, the leftmost branch of the systematic tableau from A with only tableau-consistent nodes:

Definition 2.2

$$
\begin{aligned}
Br(n) = x \equiv\ & Bra(x) \wedge Depth(x) = n \\
& \wedge \forall i \leqslant Len(x)\ (Node(i,x) \rightarrow \nabla(\beta(i,x) \dot{-} 2)) \\
& \wedge \forall i,j \leqslant Len(x)\ \{Node(i,x) \wedge Depth(i,x) = 2 \bmod(5) \\
& \qquad \wedge Father(j,x) = i \rightarrow \exists z \leqslant Len(x)\ ((Depth(i,x) = 2 + 5 \cdot z \\
& \qquad\qquad \wedge \neg\beta_1(i,j,x)) \rightarrow \neg\nabla(x * NAnt(\pi_1(z))))\}
\end{aligned}
$$

Because of the occurrence of tableau consistency in several clauses of the definition of $Br(n)=x$, this is *not* a Σ_1^b-predicate, so that we cannot hope to be able to define $Br(n)$ as a Σ_1^b-defined function in S_2^1.

Just as was the case with fully developed tableaux, we cannot hope to prove in S_2^1 that $Br(n)$ exists for all n. This is because consecutive application of the systematic tableau rules also requires existence of exponentiation. For example, apply the rule δ to a sequence of formulae, say x, and let the resulting sequence be y. Then $|y|$ is bounded by $|x|^4$: To get y from x we must add to x all the instantiations of negative existential formulae in x with the special constants in x. The number of negative existential formulae in x is bounded by Len(x), i.e. by $|x|$. The number of special constants occurring in x is also bounded by $|x|$. Hence the number of formulae that we must add to x to get y is bounded by $|x|^2$. Let $z \in x$ be $\neg\exists t\varphi$, let s be a special constant in x, and let v be the result of instantiation of $\neg\varphi$ with s . Both z and s are bounded by x, hence $|v|$ is bounded by $|x|^2$. There are $|x|^2$ such v, so $|y|$ is bounded by $|x|^4$ (in fact it is bounded by $|x|+C\cdot|x|^2+|x|^4$ for some constant C, we left out x itself and the $|x|^2$ extra comma's and codes for brackets and $\neg$). The same reasoning shows that if $\tau(x,y)$, then $|y| \leqslant 2\cdot|x|$; in general, if one of the systematic taleau rules is applied to x, then the length of the resulting y is bounded by $|x|+B\cdot|x|^2+C\cdot|x|^3+|x|^4$. Consider the family of functions f_k: $z \mapsto z^k$. An n times repeated application of such f_k to z results in $z^{(k^n)}$, i.e. a proof of the existence of $f_k^{(n)}(z)$ for all n and standard z requires existence of k^n for all n. Now if $Br(n)=y$, then $|y|$ is roughly equal to the value of $f_k^{(n)}(z)$ for a $k \geqslant 2$. This shows that, as we have shown to be the case with FullTab(n), we can not hope to prove existence of Br(n) for all n in S^1_2. However, using a theorem of Pudlák, we can define Br(n) as a function on a cut, i.e. for every element n in this cut there does exist a unique branch with depth n.

Induction Theorem 2.3 (Pudlák [83a])

For any formula $\varphi(x)$ there exists a formula J, which is inductive in S^1_2, such that for all cuts I such that $I \subset J$

$$S^1_2 \vdash (\varphi(0) \land \forall n \in I\, (\varphi(n) \to \varphi(n+1))) \to \forall n \in I\, \varphi(n).$$

Proof

Define J as follows:

$$J(x) \equiv \varphi(0) \land \forall k < x\, (\varphi(k) \to \varphi(k+1)) \to \forall k \leqslant x\, \varphi(k).$$

We will check that J is inductive in S^1_2:

$S^1_2 \vdash \varphi(0) \to \varphi(0)$, so $S^1_2 \vdash J(0)$.

Reason in S^1_2. Suppose J(x), and suppose $\varphi(0) \land \forall k < x+1\, (\varphi(k) \to \varphi(k+1))$. By hypothesis we have $\varphi(0) \land \forall k < x\, (\varphi(k) \to \varphi(k+1))$, and J(x), so by definition of J we have $\forall k \leqslant x\, \varphi(k)$. Hence $\varphi(x)$; by hypothesis $\varphi(x) \to \varphi(x+1)$, hence $\forall k \leqslant x+1\, \varphi(k)$. Hence J(x+1).

(It is however not necessarily true that J is provably closed under $\leqslant$, i.e. J need not be a cut.)

Let I be a cut in S^1_2 such that $I \subset J$. Suppose $\varphi(0) \land \forall x \in I\, (\varphi(x) \to \varphi(x+1))$. Let $x \in I$, we will prove $\varphi(x)$. Because $I \subset J$: $x \in J$, so (by definition of J) $\varphi(0) \wedge \forall k<x\ (\varphi(k) \rightarrow \varphi(k+1)) \rightarrow \forall k \leqslant x\ \varphi(k)$. By hypothesis $\varphi(0)$. Let $k<x$. I is a cut, so it is closed under $\leqslant$, and $x \in I$, hence $k \in I$. By hypothesis $\varphi(k) \rightarrow \varphi(k+1)$. Hence $\forall k \leqslant x\ \varphi(k)$, i.e., $\varphi(x)$.

⊠

We apply this theorem to the formula $\exists! y\ Br(n)=y$. This gives us an inductive formula J such that for all cuts I such that $I \subset J$

$$S^1_2 \vdash \exists! y\ Br(0)=y \wedge \forall n \in I\ (\exists! y\ Br(n)=y \rightarrow \exists! y\ Br(n+1)=y) \rightarrow \forall n \in I\ \exists! y\ Br(n)=y.$$

So after closing J under $\leqslant$, S^1_2 proves that if $\exists! y\ Br(n)=y$ is inductive on J, then Br(n) is a function on J. The following two lemmata show that $\exists! y\ Br(n)=y$ is an inductive formula of S^1_2.

Lemma 2.4

$S^1_2 \vdash \nabla X \wedge (\alpha(X,Y) \vee \gamma(X,Y) \vee \delta(X,Y) \vee \tau(X,Y)) \rightarrow \nabla Y$

$S^1_2 \vdash \nabla X \wedge \beta_1(X,Y_1,t) \wedge \beta_2(X,Y_2,t) \rightarrow \nabla Y_1 \vee \nabla Y_2$

Proof

Reason in S^1_2. Suppose ∇X, $\alpha(X,Y)$ and $\neg\nabla Y$. Then we can construct a closed tableau from X in the following way: There is a closed tableau, say p, from Y. (p might be non systematic.) The sequence coded by the root of p, say Y', is a subset of Y. From p we will construct a closed tableau q which has root Y, and in which the rules will be applied to the nodes in the same order and to the same formulas as they are applied to in p. q will have the same form as p, and if a node n' from q corresponds to node n from p, then the sequence coded in n is a subset from the sequence coded in n'. It is clear how to proceed: Add, to all nodes in p, the formulae of Y which do not occur in Y'. Clearly, for q so constructed we have $|q| \leqslant |p| \cdot (1+|Y|)$: to p we add at most Len(p) times a set of length at most $|Y|$. (It is essential here that we do not demand q to be a systematic tableau, or a tableau in which the rules for construction of a systematical tableau are applied: in that case we could not hope to be able to prove the existence of q in S^1_2.)

The proof of existence of such q, given p, is by PIND on p in:

$$x \leqslant p \wedge Tab(x,y) \rightarrow \exists q \leqslant 2^{|x|} \cdot 2^{|x| \cdot |y|} (Tab(q) \wedge (\beta(1,q) \dot{-} 2) = y \wedge Len(q) = Len(x) \wedge \forall i \leqslant Len(q) (Node(i,q) \leftrightarrow Node(i,x) \wedge Node(i,q) \rightarrow (\beta(i,x) \dot{-} 2) \subseteq (\beta(i,q) \dot{-} 2))).$$

Having constructed q, let r be the tree (0 * X+2 * ⌜[⌝) ** (q * ⌜]⌝). r is a closed tableau from X: q is a tableau and $\alpha(X,Y)$ guarantees r to be a tableau, and if s is a leaf in q then it is a subset of a leaf from p, which is closed by hypothesis, so s is closed.
The same reasoning works for γ, δ, τ.

Suppose ∇X, $\beta_1(X,Y_1,t)$, $\beta_2(X,Y_2,t)$ and $(\neg\nabla Y_1 \wedge \neg\nabla Y_2)$, then again we can construct a closed tableau from X: Suppose p_1 and p_2 are closed tableaus from respectively Y_1 and Y_2, with roots U and V. Now "fill in" p_1 and p_2 the same way we proceeded above, to get closed tableaux q_1 and q_2 with roots Y_1 and Y_2. Then (0 * X+2 * ⌜[⌝) ** q_1 ** (q_2 * ⌜]⌝) is a closed tableau from X. ⊠

Lemma 2.5
There exist Σ_1^b-definable functions $\tau(\cdot)$, $\alpha(\cdot)$, $\beta_1(\cdot,\cdot)$, $\beta_2(\cdot,\cdot)$, $\gamma(\cdot)$, $\delta(\cdot)$, such that $\tau(x)=y$ iff $\tau(x,y)$ and FormSeq(x), y=0 otherwise, and $\beta_i(x,t)=y$ iff $\beta_i(x,y,t)$ and FormSeq(x), y=0 otherwise, etc.

Proof
We will prove the existence part of the functionality of τ by Σ_1^b-PIND on

$t \leqslant x \rightarrow \exists y\, (|y| \leqslant 2 \cdot |t| \wedge (\tau(t,y) \vee (\neg FormSeq(t) \wedge y=o)))$

For x=0 we take y=0.
Assume $t \leqslant \lfloor \frac{1}{2}x \rfloor \rightarrow \exists y\, (|y| \leqslant 2 \cdot |t| \wedge (\tau(t,y) \vee (\neg FormSeq(t) \wedge y=0)))$.
Let $\lfloor \frac{1}{2}x \rfloor < s \leqslant x$ and FormSeq(s). Consider v=SubSeq(1, Len(s), s), i.e. v is s minus the last element of s. We have FormSeq(v) and $v \leqslant \lfloor \frac{1}{2}x \rfloor$. By assumption there is a w such that $|w| \leqslant 2 \cdot |v| \wedge \tau(v,w)$. So v is an initial subsequence of w.

Suppose $\neg DN(\beta(Len(s),s)$, i.e. the last element of s is not a double negation. Let y be s ** SubSeq(Len(v)+1, Len(w)+1, w); then $\tau(s,y)$ and $|y| \leqslant |2 \cdot |s|$.
If, on the other hand, $DN(\beta(Len(s),s))$, say $b=\beta(Len(s),s)$ then let a be such that DNeg(b,a). Now consider SubSeq(Len(v)+1, Len(w)+1, w) * a. This sequence is not necessarily ordered. However, by Lemma 1.4, there is an ordered sequence z such that $\forall k(k \in z \leftrightarrow k \in$ SubSeq(Len(v)+1, Len(w)+1, w) * a).
Now $\tau(s, s ** z)$ and $|s ** z| \leqslant 2 \cdot |s|$.
Unicity follows from the fact that $\tau(x,y)$ is Δ^b_1, with inimization. For suppose $\tau(x,y)$ and $\tau(x,z)$ and $y \neq z$, then there is an i such that $\beta(i,y) \neq \beta(i,z)$
For the other tableau rules the proofs resemble the proof for τ. In the β_i functions the first variable indicates the formula sequence x to be considered, the second the implication to be considered . ⊠

Using the two preceding lemmata we can now prove that $\exists! y\ Br(x)=y$ is an inductive formula of S^1_2:

Lemma 2.6
$S^1_2 + \nabla A \vdash \exists! y\ Br(0)=y \wedge \forall x\ (\exists! y\ Br(x)=y \rightarrow \exists! y\ Br(x+1)=y)$

Proof
First we check that $S^1_2 + \nabla A \vdash \exists! y\ Br(0)=y$:
As we mentioned in paragraph 1, $S^1_2 \vdash \exists! n\ A(n)$. Moreover, $S^1_2 \vdash$ Tree(n+2) $\wedge$ Depth(n+2)=0, and ∇A implies ∇n. This shows that $S^1_2 + \nabla A$ proves Br(0)=n+2, and that $S^1_2 + \nabla A$ proves uniqueness of Br(0).
Now reason in S^1_2. Suppose there is a unique y such that y is Br(x). Let u=Leaf(y). Then u is tableau consistent. We can construct Br(x+1) from y, by taking the appropriate tableau successor v of u and putting [u+2] in the tree y directly after u. To get v, we apply to u the tableau rule which matches the depth of u in y (i.e. x). That is, take $v=\tau(u)$ if x=0 mod(5). If $x=2+5 \cdot z$, take for v the result of the application of β_1 or β_2 to u and $\pi_1(z)$; take for v the outcome of β_1 if it is tableau consistent, otherwise take the outcome of β_2. Lemma 2.4 shows that either the outcome of application of β_1 to u is tableau consistent, or the outcome of β_2 is, because v is tableau consistent; and Lemma 2.5 shows that if

we take for v the outcome of application of one of the other tableau rules to u, then also v is tableau consistent, because u is. Lemma 2.6 shows that v is unique. Now by putting [v+2] in the tree y directly after the node that contains u we get Br(x+1). ⊠

It will now be clear that S^1_2 proves that Br(·) is a function on the cut J. In the sequel we will have to restrict ourselves to an initial below J. Close J under +, · and Ω.

The following lemma shows, among other things, that Br(·) defines a unique branch along the initial J.

Lemma 2.7

$S^1_2 \vdash Br(n)=y \wedge z\in(i,y) \rightarrow z\in Leaf(y)$

$S^1_2 \vdash J(n) \wedge J(m) \wedge n<m \wedge Br(n)=y \wedge Br(m)=z \wedge t\in(i,y) \rightarrow t\in Leaf(z)$

Proof

By inspection of the definitions of Tree and Br(n)=y it is easy to verify that

$S^1_2 \vdash n<m \wedge Br(n)=y \wedge Br(m)=z \rightarrow$

$\quad Br(m)=SubSeq(1, Len(y)\dot{-}n+1, y) ** SubSeq(Len(y)\dot{-}n+1, Len(z)+1, z).$

Moreover, Br(n) is a function on J, so that

$S^1_2 \vdash J(n) \wedge J(m) \wedge n<m \wedge Br(n)=y \wedge Br(m)=z \rightarrow$

$\quad z=SubSeq(1, Len(y)\dot{-}n+1, y) ** SubSeq(Len(y)\dot{-}n+1, Len(z)+1, z).$

(Note that we need J(n) and J(m) here: we cannot apply inimization because Br(m)=z is not a Δ^b_1-predicate.)

This immediately gives us

(1) $S^1_2 \vdash J(n) \wedge J(m) \wedge n<m \wedge Br(n)=y \wedge Br(m)=z \rightarrow$

$\quad (x\in(i,y) \rightarrow x\in(i,z)).$

Using the fact that Bra (Definition 2.1) is Δ^b_1 we can show that S^1_2 proves that if a formula occurs in a node on a branch of the systematic tableau, it will occur in all its successors:

(2) $S^1_2 \vdash Bra(y) \wedge z\in(i,y) \rightarrow \forall j \leqslant Len(y)\ (j>i \wedge Node(j,y) \rightarrow z\in(j,y)$.

Reason in S^1_2: Suppose Bra(y) and $z\in(i,y)$ and suppose there is a $j>i$ such that Node(j,y) and $z\notin(j,y)$. Then by inimization there is a minimal such j. Let $k \geqslant i$ be the father of j in y. Then by minimality of j, $z\in(k,y)$,

and because fathers and sons in y are linked by the tableau rules, we have $I(\beta(k,y)\dot{-}2), \beta(j,y)\dot{-}2)$, which implies $z\in(j,y)$. Contradiction.
Combination of (1), (2) and the fact that, by definition of $Br(\cdot)$, S^1_2 proves $Br(n)=y \rightarrow Bra(y)$, we get $S^1_2 \vdash Bra(y) \wedge z\in(i,y) \rightarrow z\in Leaf(y)$
$S^1_2 \vdash Br(n)=y \wedge z\in(i,y) \rightarrow z\in Leaf(y)$
$S^1_2 \vdash J(n) \wedge J(m) \wedge n<m \wedge Br(n)=y \wedge Br(m)=z \wedge t\in(i,y) \rightarrow t\in Leaf(z).$

⊠

Now that we have a unique branch along the initial J we are almost ready to define an interpretation. We first define a predicate K(x) which expresses that x occurs somewhere on this branch, and a predicate D(x) which expresses that x is EForm and occurs somewhere on the branch, or x is 0 and is used as a special constant somewhere on the branch. K will be used as the basis of the translation, and D will be the domain of the interpretation.

Definition 2.8

$K(x) \equiv J(x) \wedge \exists n\, (J(n) \wedge x\in Leaf(Br(n)))$
$D(x) \equiv (EForm(x) \wedge K(x)) \vee x=0$

We will need the following property of the set of formulae occurring in the infinite tableau consistent branch on J, which was called the Hintikka-property by Smullyan [68]:

Lemma 2.9

$S^1_2 \vdash DNeg(x,y) \wedge K(x) \rightarrow K(y);$
$S^1_2 \vdash IMP(x) \wedge K(x) \rightarrow (K(NAnt(x)) \vee K(Cons(x)));$
$S^1_2 \vdash NIMP(x) \wedge K(x) \rightarrow (K(AntN(x)) \wedge K(NConN(x)));$
$S^1_2 \vdash EForm(x) \wedge K(x) \rightarrow K(Sub(SpeCon(x), EVar(x), B(x));$
$S^1_2 \vdash NEForm(x) \wedge K(x) \rightarrow$
$$\forall z\, [D(z) \rightarrow \exists y\, \exists t\, (t=Sub(SpeCon(z), EVar(x), B(x)) \wedge NEG(y,t) \wedge K(y)].$$

Proof

Reason in S^1_2.

Suppose K(x) and DNeg(x,y). By definition of K we have J(x), there is an n such that J(n), there is a unique z such that z=Br(n) and for this z we have $x\in Leaf(z)$. If DNeg(x,y), then $y<x$; J is closed under $\leqslant$, so we

have J(y). Remember that in branches of a systematic tableau as defined in Definition 2.1 the rule τ is applied to nodes occurring on depth 0 mod(5). There is an m such that $n \leqslant m \leqslant n+4$ and $m=0 \bmod(5)$. J is closed under successor, so J(m) and J(m+1), which imply $\exists!v$ $v=Br(m)$ and $\exists!w$ $w=Br(m+1)$. By Lemma 2.8, $x \in Leaf(v)$. From the definition of Br(·) it follows that w is the result of applying the systematic τ rule to the leaf of v and adding the result to v. Hence if x is the double negation of y, y will be in the leaf of w.
Suppose IMP(x) and K(x). Then J(x) and there is an $n \in J$ such that $x \in Leaf(Br(n)$. NAnt(x) and Cons(x) are smaller than x, so both are in J. We need some stage m in J in which, according to the definition of Br, the implication x is dealt with. Take $m=2+5\cdot P(x,n)$, where P is the pairing function. From the definition of P, and from the fact that J is closed under + and ·, it follows that $m \in J$.

Suppose EForm(x) and K(x). Let n be such that J(n), let z be Br(n) with $x \in Leaf(z)$. From the definition of K we have J(x). Let m be such that $m=3 \bmod(5)$ and $n \leqslant m \leqslant n+4$. Then J(m) and J(m+1). Let v be Br(m), then $x \in Leaf(v)$. Now let w be Br(m+1). Leaf(w) is the result of applying the systematic δ rule to the leaf of v, hence Sub(SpeCon(x), EVar(x), B(x)) is in Leaf(w). From the fact that $x \in J$ and that J is closed under Ω, we see that Sub(SpeCon(x), EVar(x), B(x))$\in$J.
We leave the verification of the other clauses of this lemma to the reader. ⊠

Another fact that we will need is that every axiom of A provably occurs on the branch along J:

Lemma 2.10
If φ is an axiom of A, then $S^1_2+\nabla A \vdash K(\ulcorner\varphi\urcorner)$.

Proof
Let φ be an axiom of A. We will show that $S^1_2+\nabla A \vdash J(\ulcorner\varphi\urcorner) \wedge \exists n\, (J(n) \wedge \ulcorner\varphi\urcorner \in Leaf(Br(n)))$.
Let m be the standard number such that A(m), i.e. m is the sequence which contains the codes of the axioms of A. A(m) and $\ulcorner\varphi\urcorner \in m$ are true Δ^b_1-formulae, so S^1_2 proves A(m) and $\ulcorner\varphi\urcorner \in m$. S^1_2 proves that m+2 is the code of a tree whose depth is 0, and ∇A implies ∇m. Hence $S^1_2+\nabla A$

proves $Br(0)=m+2 \wedge \ulcorner\varphi\urcorner \epsilon Leaf(m+2)$. Because J is a cut in $S^1_2+\nabla A$, $S^1_2+\nabla A$ proves $J(0)$. $\ulcorner\varphi\urcorner$ is a standard number, so $S^1_2+\nabla A$ proves $J(\ulcorner\varphi\urcorner)$. ⊠

We will now, in three consecutive steps, define an interpretation based on K and D.

Definition 2.11

For every unary predicate R in the language L define the valuation $(\cdot)^{K'}$ as follows:

$$R(x)^{K'} \equiv \begin{cases} K(\ulcorner R(c[\psi])\urcorner) & \text{if } \psi \text{ is an existential formula from } L^+ \text{ and } x \text{ codes } \psi, \\ K(\ulcorner R(c[0])\urcorner) & \text{if } x=0, \\ 0=S0 & \text{if not EForm}(x) \text{ and } x\neq 0. \end{cases}$$

We introduce for convenience the following notation:

We will write $R[x]$ for $R(\cdot)$ instantiated with the special constant for x if x is EForm or 0; and

$\ulcorner R[x]\urcorner$ for $\ulcorner(Rc\urcorner * x * \ulcorner)\urcorner$.

$\varphi(\vec{x})$ indicates that $\vec{x}=x_1, \ldots, x_n$ are the free variables occurring in φ

$\varphi[\vec{x}]$ will be the obvious generalization of $R[x]$ to formulae, i.e. replace in φ all occurrences of Rx, if x is free in φ, by $R[x]$.

We define a translation $(\cdot)^{K'}$ of formulae in L^+ inductively as follows: on atomic formulae $(\cdot)^{K'}$ is as defined above;

$(\neg\varphi)^{K'}=\neg\varphi^{K'}$;

$(\varphi\rightarrow\psi)^{K'}=\varphi^{K'}\rightarrow\psi^{K'}$;

$((\exists x)\varphi)^{K'}=((\exists x)((D(x) \wedge \varphi^{K'}))$.

Define the translation $(\cdot)^K$ as follows:

if $x_1, \ldots, x_n$ are the free variables occurring in φ, then (writing $D(\vec{x})$ for $D(x_1) \wedge \ldots \wedge D(x_n)$)

$\varphi(\vec{x})^K = D(\vec{x}) \rightarrow \varphi(\vec{x})^{K'}$.

(This completes Definition 2.11)

Clearly the domain of this translation is not empty, because 0 is in it (Definition 2.8).
The translation $(\cdot)^K$ provides us with an interpretation of the axioms of the finite theory A in $S^1_2+\nabla A$, that is the following theorem holds:

Theorem 2.12
If A is a finitely axiomatized relational theory, axiomatized by closed axioms, then there is an interpretation $(\cdot)^K$ such that for every axiom φ of A, $S^1_2+\nabla A \vdash \varphi^K$.

We need the following lemma.

Lemma 2.13
$S^1_2 \vdash \forall\vec{x}\ (K(\ulcorner\varphi[\vec{x}]\urcorner)) \rightarrow \varphi^K))$ for all formulae $\varphi(\vec{x})$.

Proof
We prove this by induction on the degree of φ.

Let $Deg(\varphi)=0$. Then φ is $R(x)$ for some variable x. Suppose $D(x)$ and $K(\ulcorner R[x]\urcorner)$. By definition $D(x) \wedge K(\ulcorner R[x]\urcorner)$ implies $R(x)^{K'}$, which implies by definition $R(x)^K$.

We consider separately the case that $\varphi(x)$ is $\neg R(x)$. Suppose $D(x)$ and $K(\ulcorner\neg R[x]\urcorner)$. We will show $(\neg R(x))^K$, i.e. $D(x) \rightarrow (\neg R(x))^{K'}$, i.e. $\neg(D(x) \wedge R(x)^{K'})$. Suppose $D(x) \wedge R(x)^{K'}$. Then we get by definition $D(x) \wedge K(\ulcorner R[x]\urcorner)$; but by hypothesis $D(x)$ implies $K(\ulcorner\neg R[x]\urcorner)$. So we have $K(\ulcorner R[x]\urcorner) \wedge K(\ulcorner\neg R[x]\urcorner)$, which implies, by definition of K and by Lemma 2.7, $\exists n\ (J(n) \wedge \exists z\ (Br(n)=z \wedge \ulcorner R[x]\urcorner \in Leaf(z) \wedge \ulcorner\neg R[x]\urcorner \in Leaf(z)))$. But then $Leaf(z)$ is not tableau consistent, as it should be by the definition of $Br(n)$. Hence $\neg(D(x) \wedge R(x)^{K'})$, i.e. $(\neg R(x))^K$.

Suppose that for all formulae $\varphi(\vec{x})$ with $Deg(\varphi)<n$, $S^1_2 \vdash \forall x\ (D(\vec{x}) \rightarrow (K(\ulcorner\varphi[\vec{x}]\urcorner)) \rightarrow \varphi^K))$. Let $Deg(\varphi(\vec{x}))=n$.

Let $\varphi(\vec{x})=\neg\neg\psi(\vec{x})$. Suppose $D(\vec{x})$ and suppose $K(\ulcorner\neg\neg\psi[\vec{x}]\urcorner)$. By Lemma 2.9 this gives us $D(\vec{x}) \rightarrow K(\ulcorner\psi[\vec{x}]\urcorner)$. $Deg(\psi(\vec{x}))<n$, so by hypothesis we now have $\psi(\vec{x})^K$, i.e. $D(\vec{x}) \rightarrow \psi(\vec{x})^{K'}$. By definition of $(\cdot)^{K'}$ this gives us $D(\vec{x}) \rightarrow (\neg\neg\psi(\vec{x}))^{K'}$, i.e. $(\neg\neg\psi(\vec{x}))^K$.

We will leave the cases $\varphi = \neg(\psi \to \chi)$ and $\varphi = (\psi \to \chi)$ to the reader.

Let $\varphi(\vec{x}) = \exists z\, \psi(z, \vec{x})$. Suppose $D(\vec{x})$ and suppose $K(\ulcorner \varphi[\vec{x}] \urcorner)$. Then, by Lemma 2.9, stating the Hintikka property for formulae in K, it follows from $K(\ulcorner \varphi[\vec{x}] \urcorner)$ that $\exists z\ (D(z) \wedge K(\ulcorner \psi[z, \vec{x}] \urcorner))$. By the induction hypothesis $D(\vec{x}) \wedge D(z) \wedge K(\ulcorner \psi[z, \vec{x}] \urcorner)$ implies $\psi(z, \vec{x})^K$, i.e. $D(\vec{x}) \wedge D(z) \to \psi(z, \vec{x})^{K'}$. This gives us, because we have $D(\vec{x})$ as hypothesis, $\exists z\ (D(z) \to \psi(z, \vec{x})^{K'})$. Hence $D(\vec{x}) \to \exists z\ (D(z) \to \psi(z, \vec{x})^{K'})$, which is $\varphi(\vec{x})^K$. Hence $K(\ulcorner \varphi[\vec{x}] \urcorner) \to \varphi(\vec{x})^K$.

Let $\varphi(\vec{x}) = \neg\exists z\, \psi(z, \vec{x})$ and suppose $D(\vec{x})$ and $K(\ulcorner \varphi[\vec{x}] \urcorner)$. With Lemma 2.9 we get $\forall u\ (D(u) \to K(\ulcorner \neg\psi[u, \vec{x}] \urcorner))$ from $K(\ulcorner \varphi[\vec{x}] \urcorner)$. Now suppose $D(u)$. Then we have $D(\vec{x}) \wedge D(u) \wedge K(\ulcorner \neg\psi[u, \vec{x}] \urcorner)$, which by induction hypothesis gives us $D(\vec{x}) \wedge D(u) \to (\neg\psi(u, \vec{x}))^{K'}$. Because we supposed $D(\vec{x})$, an application of generalization gives us $\forall u\ (D(u) \to (\neg\psi(u, \vec{x}))^{K'})$, i.e. $\neg\exists u\ (D(u) \wedge \psi(u, \vec{x})^{K'})$. Then clearly we have $D(\vec{x}) \to \neg\exists u\ (D(u) \wedge \psi(u, \vec{x})^{K'})$, which is by definition $\varphi(\vec{x})^K$. Hence $K(\ulcorner \varphi[\vec{x}] \urcorner) \to \varphi(\vec{x})^K$. ⊠

To complete the proof of Theorem 2.12, we only have to observe that the axioms of A are sentences. Hence the form of Lemma 2.14 we apply to φ if φ is an axiom of A, is $S^1_2 \vdash K(\ulcorner \varphi \urcorner) \to \varphi^K$. Lemma 2.10 shows that for such φ $S^1_2 + \nabla A \vdash K(\ulcorner \varphi \urcorner)$. This completes the proof of Theorem 2.12.

But we have more: $S^1_2 + \nabla A$ also proves the *theorems* of A under the interpretation $(\cdot)^K$. Let φ be a theorem of A, i.e. $A \vdash \varphi$. Let $\psi_1, \psi_2, \ldots, \psi_n$ be axioms of A such that $\psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n \vdash \varphi$. Call this proof π. Replace every formula in π by its $(\cdot)^K$-translation. Theorem 2.12 shows that there are proofs of $\psi_1{}^K, \psi_2{}^K, \ldots,$ and $\psi_n{}^K$ from $S^1_2 + \nabla A$. If we take these proofs together and add π^K, the result is almost a proof of φ^K from $S^1_2 + \nabla A$. It might not be a proof yet for the following reason.
Suppose that in π we have applied generalization to the free variable x in $\varphi(x,y)$. This gives $\forall x\ \varphi(x,y)$. The respective $(\cdot)^K$-translations of these formulae are: $D(x) \wedge D(y) \to \varphi(x,y)^{K'}$ and $D(y) \to \forall x\ (D(x) \to \varphi(x,y)^{K'})$. This second formula does not follow directly from the first with generalization: generalization of $D(x) \wedge D(y) \to \varphi(x,y)^{K'}$ gives $\forall x\ (D(x) \wedge D(y) \to \varphi(x,y)^{K'})$. Hence we have to add $\forall x\ (D(x) \wedge D(y) \to \varphi(x,y)^{K'}) \to (D(y) \to \forall x\ (D(x) \to \varphi(x,y)^{K'}))$. This proves:

Corollary 2.14

If A is a finitely axiomatized relational theory, axiomatized by closed axioms, then there is an interpretation $(\cdot)^K$ such that for every theorem φ of A, $S^1_2 + \nabla A \vdash \varphi^K$, i.e., if $A \vdash \varphi$, then $S^1_2 + \nabla A \vdash \varphi^K$.

§ 3 A formalization of the model existence lemma for finite theories

In this paragraph we will prove a formalized version of Theorem 2.12, that is, we will prove

Theorem 3.1
Let A be a finitely axiomatized relational theory, axiomatized by closed formulae.
Let $\mathcal{A}$ be some Δ_1^b-predicate with one free variable, such that $\mathcal{A}(x)$ expresses adequately that x is the code of an axiom of A.
Then there is an S^1_2-function $(\cdot)^K$, which associates with every code of a formula in the language of A, a code of a formula in the language of S^1_2, and which behaves like an interpretation, such that

$$S^1_2 \vdash \forall x\, (\mathcal{A}(x) \rightarrow \mathrm{Prov}_{S^1_2 + \nabla A}(x^K)).$$

The proof of Theorem 3.1 will consist in showing that the interpretation of A in $S^1_2 + \nabla A$ which we constructed in paragraph 2 can be executed in S^1_2.
The function $(\cdot)^K$ of which Theorem 3.1 claims existence, will be the formalized version of the translation $(\cdot)^K$ defined in paragraph 2 (Definition 2.11). Just like the translation $(\cdot)^K$, the function $(\cdot)^K$ will be constructed from a function $(\cdot)^{K'}$, which is the formalized version of the translation $(\cdot)^{K'}$ defined in Definition 2.11.
We will define $(\cdot)^{K'}$ as a Σ_1^b-function which associates to every x, with $\mathrm{Form}^+(x)$, the *code* of the K'-translation of x. For example, if x is the code of the atomic formula R(y), $x^{K'}$ will be the code of $K(\ulcorner R[y] \urcorner)$, where $K(\cdot)$ is the (S^1_2-) formula which we defined in Definition 2.8.
First we will define (Definition 3.2) for all formulae x a unique parsing tree, which contains x in its root and parses x to all its subformulae. Existence and uniqueness of these parsing trees will be proved in Lemma 3.3. By working upwards from the leaves to the root in the parsing tree, we will construct, step by step, a unique tree, which has the same form as the parsing tree, and which contains the $(\cdot)^{K'}$ translations of all subformulae of x (Definition 3.5). Such a tree will contain $x^{K'}$ in its root. Analogous to the definition of K' in paragraph 2,

the code of the (S^1_2-) formula $D(\cdot)$, defined in Definition 2.8, will play the role of the domain of the translation. Again, existence and uniqueness of the tree which has in its root the K' translation of x has to be proved (Lemma 3.6).
We will show that $(\cdot)^{K'}$ commutes, provably in S^1_2, with the logical symbols (Lemma 3.9).
The function x^K will be defined from the function $x^{K'}$, analogous to the definition of $(\cdot)^K$ from $(\cdot)^{K'}$ in paragraph 2, by 'adding' the code for a formula which expresses that all free variables occurring in x are in the domain D of the translation (Definition 2.10).

First we define a binary Δ^b_1-predicate PT(w,x) which expresses that, provided x is Form$^+$, w is a tree which constitutes a parsing tree for x.

Definition 3.2

$$
\begin{aligned}
PT(w,x) \equiv\ & (w=0 \wedge \neg Form^+(x)) \\
& \vee (Tree(w) \wedge Form^+(x) \wedge Root\beta(w)=x \\
& \wedge \forall i \leqslant Len(w)\ (Node(i,w) \\
& \qquad \rightarrow (AtForm^+(\beta(i,w)\dot{-}2) \rightarrow Leaf(i,w))) \\
& \wedge \forall i \leqslant Len(w)\ (Node(i,w) \rightarrow \\
& \quad ([IMP(\beta(i,w)\dot{-}2) \rightarrow [Valence(i,w)=2 \\
& \quad \wedge \exists j,k \leqslant Len(w)\ (SonPos(i,j,w)=1 \wedge SonPos(i,k,w)=2 \\
& \qquad \wedge IMPL(\beta(i,w)\dot{-}2, \beta(j,w)\dot{-}2, \beta(k,w)\dot{-}2))]] \\
& \quad \wedge [NOT(\beta(i,w)\dot{-}2) \rightarrow [Valence(i,w)=1 \\
& \qquad \wedge \exists k \leqslant Len(w)\ (Son(i,w)=k \\
& \qquad\quad \wedge NEG(\beta(i,w)\dot{-}2, \beta(k,w)\dot{-}2))]] \\
& \quad \wedge [EForm(\beta(i,w)\dot{-}2) \rightarrow [Valence(i,w)=1 \\
& \qquad \wedge \exists k \leqslant Len(w)\ (Son(i,w)=k \\
& \qquad\quad \wedge \beta(k,w)\dot{-}2=B(\beta(i,w)\dot{-}2))]]))
\end{aligned}
$$

To prove existence of a unique parsing tree w for every x, we will use the fact that we can bound w in the formula $\exists w\, PT(w,x)$ by an S^1_2-term in x. Hence $\exists w\, PT(w,x)$ is equivalent to a Σ^b_1-formula, and we can use Σ^b_1-PIND to prove existence.
That there is a term r(x) such that for all x there is a $w \leqslant r(x)$ such that PT(w,x) can be seen as follows. A parsing tree w of a formula x contains all subformulae of x. There are at most |x| subformulae of x,

and each of them is bounded in length by $|x|$, i.e. w will have at most $|x|$ nodes, and these will have length $\leqslant 2\cdot(|x|+2)$. w will then contain at most $2(|x|\dot{-}1)$ left and right brackets, and we have to add enough commas. This gives us the upper bound $2\cdot|x|^2+14\cdot|x|$ for $|w|$.

Lemma 3.3

$S^1_2 \vdash \forall x\, \exists! w\, (|w| \leqslant 2\cdot|x|^2+14\cdot|x| \wedge PT(w,x))$

Proof

We prove existence by Σ^b_1-PIND.

If x=0 then $\neg Form^+(x)$, so then we take w=0; clearly $S^1_2 \vdash PT(0,0)$. Reason in S^1_2.

Suppose that for all $y \leqslant \lfloor \frac{1}{2}x \rfloor$ there is a parsing tree which is bounded in length by $2\cdot|y|^2+14\cdot|y|$.

If $\neg Form^+(x)$, then take w=0.

Let x be a formula.

- If x is atomic, then the parsing tree of x is 0 * x+2.
- Let x be an implication, say, IMPL(x,a,b). Then $a < \lfloor \frac{1}{2}x \rfloor$ and $b < \lfloor \frac{1}{2}x \rfloor$. By the induction hypothesis there are parsing trees p and q for a, resp. b, with $|p| \leqslant 2\cdot|a|^2+14\cdot|a|$ and $|q| \leqslant 2\cdot|b|^2+14\cdot|b|$. Consider w=(0 * x+2 * ⌜[⌝) ** p ** (q * ⌜]⌝). w is a parsing tree for x. Some counting will show that $|w| \leqslant 2\cdot|x|^2+14\cdot|x|$.
- We will not spell out the cases NOT(x) and EForm(x).

This shows that $S^1_2 \vdash \forall x\, \exists w\, (|w| \leqslant 2\cdot|x|^2+14\cdot|x| \wedge PT(w,x))$.

Unicity is proven as follows. Suppose $w \neq v$. Then $\exists i \leqslant Len(w)$ $\beta(i,v) \neq \beta(i,w)$, so either $\exists i \leqslant Len(w)$ $\beta(i+1,w) \neq \beta(i+1,v)$ or $\forall i \leqslant Len(w)$ $(\beta(i+1,w) = \beta(i+1,v) \wedge \beta(0,w) \neq \beta(0,v)$.

In the first case, we get a minimal such i by inimization. Inspection of the definition of PT will show such an i can not exist.

In the second case, we have Len(v) > Len(w), and I(w,v). Suppose Len(w)=1. Then x is atomic, so that also Len(v) =1. If Len(w) > 1, then consider Tail(v,w). Either Tail(v,w) has no nodes, which implies that v is not a tree, and thus not a parsing tree, or it has a node. Inimization then gives us a minimal i such that Node(i,Tail(v,w)). Let j be such that $\beta(j,v) = \beta(i,Tail(v,w))$. If Depth(j,v)=0, then we have a contradiction with the fact that v does not have more than one root. If Depth(j,v) > 0, then there is some $k < Len(w)$, such that $Father(j,v)=k$. Then we get a contradiction with the fact that all possible sons of k are already in w.

⊠

In the definition of the translation tree for a formula x we will use the following predicate which expresses that its subject is a parsing tree:

Definition 3.4
$PT(w) \equiv PT(w, Root\beta(w))$.

We will now define the Δ_1^b-predicate PreKT(z,w,c) which expresses that
(1) w is a parsing tree,
(2) z is a tree with the same form as the tree w,
(3) until $\beta(Len(w) \dot{-} c, z)$, z is identical to w,
(4) z contains the $(\cdot)^{K'}$ translations of the formulae in w from $\beta(Len(w) \dot{-} c, z)$ onward.

Ad (2). The following formula expresses that w and z are two trees of the same form:
$$Tree(z) \wedge Tree(w) \wedge Len(z)=Len(w)$$
$$\wedge\ \forall i \leqslant Len(z)\ (\neg Node(i,z) \rightarrow \beta(i,z)=\beta(i,w)).$$

Ad (4). The translation tree will contain (if c is large enough) in its leaves the intended $(\cdot)^{K'}$-translations of atomic subformulae, i.e. codes of S_2^1-formulae of the form $K(\ulcorner R[x] \urcorner)$.
This implies that, in order to define PreKT, we need a Σ_1^b-definable function $\mathcal{S}$, which enables us to define the translation of atomic formulae.

Definition 3.5
$\mathcal{S}(\ulcorner (Rx) \urcorner)=\ulcorner (0 * \ulcorner (\urcorner * \ulcorner R \urcorner * 14) ** (x * \ulcorner) \urcorner) \urcorner$.

Later on we will generalize $\mathcal{S}$ to get a function which replaces all free variables of a formula by their special constants. For the moment however, we will define $\mathcal{S}(x)$ as the identical function for x with $\neg AtForm^{+}(x)$.

Moreover, for the definition of PreKT we need a substitution function which can substitute ⌜R[x]⌝ for y in the code of K(y). In paragraph 1 we have already defined a substitution function (Definition 1.19). However, this substitution function only admits substitution in formulae in the language L^+, whereas we now need substitution in K(·), which is a formula in the language of S^1_2. We will just assume that we have a suitable predicate expressing formulahood in S^1_2, which we can use in our previously defined substitution function instead of $Form^+$.
We write ⌜K(x)⌝ and ⌜D(x)⌝ for the codes for the S^1_2-formulae K(x) resp. D(x), which we defined in the previous paragraph. Renaming of variables will sometimes be necessary. We will not spell out how this should be done and just mention the fact that S^1_2 can handle this.

Definition 3.6.a

$$
\begin{aligned}
\mathrm{PreKT}(z,w,c) \equiv\ & \mathrm{PT}(w) \\
& \wedge \{[w=0 \wedge z=0] \\
& \vee [w>0 \wedge \mathrm{Tree}(z) \wedge \mathrm{Len}(z)=\mathrm{Len}(w) \\
& \wedge \forall i \leqslant \mathrm{Len}(z)\, (\neg \mathrm{Node}(i,z) \rightarrow \beta(i,z))=\beta(i,w) \\
& \wedge \forall i \leqslant \mathrm{Len}(z)\, (\mathrm{Node}(i,z) \wedge i \leqslant \mathrm{Len}(z) \dot- c \rightarrow \beta(i,z)=\beta(i,w)) \\
& \wedge \forall i \leqslant \mathrm{Len}(z)\, (\mathrm{Node}(i,z) \wedge i > \mathrm{Len}(z) \dot- c \rightarrow \\
& ([\mathrm{Leaf}(i,z) \rightarrow \beta(i,z) \dot- 2 = \mathrm{Sub}(\mathcal{S}(\beta(i,w) \dot- 2), \ulcorner x \urcorner, \ulcorner K(x) \urcorner))] \\
& \wedge [\forall j,k \leqslant \mathrm{Len}(w) \\
& ([\mathrm{IMP}(\beta(i,w) \dot- 2) \wedge \mathrm{SonPos}(i,j,w)=1 \wedge \mathrm{SonPos}(i,k,w)=2 \rightarrow \\
& \quad \beta(i,z) \dot- 2 = (0 * \ulcorner (\urcorner * (\beta(j,z) \dot- 2) * \ulcorner \rightarrow \urcorner) ** \\
& \quad\quad ((\beta(k,z) \dot- 2) * \ulcorner) \urcorner] \\
& \wedge [\mathrm{NOT}(\beta(i,w) \dot- 2) \wedge \mathrm{Father}(k,w)=i \rightarrow \\
& \quad (\mathrm{NEG}(\beta(i,w) \dot- 2, \beta(k,w) \dot- 2) \wedge \mathrm{NEG}(\beta(i,z) \dot- 2, \beta(k,z) \dot- 2))] \\
& \wedge [\mathrm{EForm}(\beta(i,w) \dot- 2) \wedge \mathrm{Father}(k,w)=i \rightarrow \\
& \quad \beta(i,z) \dot- 2 = (\ulcorner ((\exists \urcorner * \mathrm{EVar}(\beta(i,w) \dot- 2) * \ulcorner)(\urcorner) ** \\
& (\mathrm{Sub}(\mathrm{EVar}(\beta(i,w) \dot- 2), \ulcorner x \urcorner, \ulcorner D(x) \urcorner) * \ulcorner \wedge \urcorner) ** ((\beta(k,z) \dot- 2) * \ulcorner)) \urcorner))])]\}
\end{aligned}
$$

Definition 3.6.b

$$\mathrm{KT}(z,w) \equiv \mathrm{PreKP}(z,w,|w|)$$

In the formula ∃z PreKT(z,w,c) z can be bounded by a term in **w** and c. This can be seen as follows.

Because K is a fixed formula, the result of substitution of ⌜R[x]⌝ in ⌜K(·)⌝ is linear in ⌜R[x]⌝; and the code of R[x] is linear in the code of R(x). Hence ⌜K(⌜R[x]⌝)⌝, viz. Sub($\mathcal{S}$(⌜(Rx)⌝), ⌜y⌝, ⌜K(y)⌝), is linear in ⌜R(x)⌝. Also the result of substituting ⌜x⌝ in ⌜D(·)⌝ is linear in ⌜x⌝. The K' translation of a formula y consists in replacing every atomic subformula R(x) by the formula expressing it is in K, i.e. by K(⌜R[x]⌝) and relativising all quantifiers to D. Hence to get the $(\cdot)^{K'}$-translation of y, we replace the element t of the sequence y by something that is at most linear in t, so this $(\cdot)^{K'}$-translation of y is linear in y, say it is bounded by h(y).

Now suppose PreKT(z,w,c). Then, if w is a parsing tree, we have replaced at most c formulae occurring in nodes of w by their K'-translations. All those formulae are smaller than w, so z can be bounded by $w+c\cdot h(w)$.

This shows that we can use LIND on the Σ^b_1-formula $\exists z\ (z \leqslant w+c\cdot h(w) \wedge \text{PreKT}(z,w,c))$ to prove $\forall w\ \forall c\ \exists z\ \text{PreKT}(z,w,|c|)$.

Lemma 3.7

$S^1_2 \vdash \forall w\ \exists! z\ \ \text{KT}(z,w)$

Proof

We prove existence by LIND on $\exists z$ PreKT(z,w,c).

Reason in S^1_2.

If $\neg$PT(w) or w=0 take z=0.

Suppose PT(w). Suppose PreKT(z,w,c) and $z \leqslant w+c\cdot h(w)$, with h the linear function described above.

Then w and z correspond with each other until the $\text{Len}(w)\dot{-}c^{th}$ element of w, i.e. SubSeq(1, Len(w)$\dot{-}$c+1, w)=SubSeq(1, Len(w)$\dot{-}$c+1, z).

Let d=Len(w)$\dot{-}$c.

If the last element of this subsequence, i.e β(d,w), is not a node, then PreKT(z,w,c+1), and $z \leqslant w+(c+1)\cdot h(w)$

Suppose β(d,w) is a node, say β(d,w)=f+2. Then f must be Form^+ (it is provable in S^1_2 that all nodes in a parsing tree code a formula).

- If f is an atomic formula, let g=$\mathcal{S}$(f).

If f codes a negation or an existential formula, then by definition of PT there is a k, with $d < k \leqslant$ Len(w), such that Son(d,w)=k.

- In case f codes a negation take g such that NEG(g, β(k,z)$\dot{-}$2).

- If f codes an existential formula, say $\exists x\varphi$, take $g=\ulcorner((\exists x)(Dx \wedge\urcorner ** \ulcorner\varphi\urcorner*\ulcorner)\urcorner$.
- If f codes an implication, then there are k, l such that $d<k<l\leqslant \mathrm{Len}(w)$ and $\mathrm{SonPos}(k,d,w)=1 \wedge \mathrm{SonPos}(l,d,w)=2$. Then take g such that $\mathrm{IMPL}(g, \beta(k,w)\dot{-}2, \beta(l,w)\dot{-}2)$.

Now let $s=(\mathrm{SubSeq}(1,d,w) * g+2) ** \mathrm{SubSeq}(d+2,\mathrm{Len}(w)+1,z)$, then $\mathrm{PreKT}(s,w,c+1)$. Moreover, $\beta(d,s)\dot{-}2\leqslant h(\beta(d,w)\dot{-}2)$, and we can safely assume $h(\beta(d,w)\dot{-}2)\leqslant h(w)$; hence $s\leqslant w+(c+1)\cdot h(w)$.

Unicity of z in PreKT(z,w,c) is proved in the same manner as unicity of w in PT(w,x) was proved in Lemma 3.3).
This gives us $S^1_2 \vdash \forall w,c\ \exists! z\ \mathrm{PreKT}(z,w,|c|)$.
Hence we have $S^1_2 \vdash \forall w\ \exists! z\ \mathrm{PreKT}(z,w)$. ⊠

The next scheme gives an impression of how PT and KT act on formulae:

$\varphi \rightarrow \psi$	$\varphi^{K'}\rightarrow\psi^{K'}$
$\varphi \qquad \psi$	$\varphi^{K'} \qquad \psi^{K'}$
$\neg\varphi$	$\neg(\varphi^{K'})$
φ	$\varphi^{K'}$
$\exists x\varphi$	$\exists x\,(D(x) \wedge \varphi^{K'})$
φ	$\varphi^{K'}$
$R(x)$	$K(\ulcorner R[x]\urcorner)$

We can now define the basic translation $(\cdot)^{K'}$ as a function on codes of formulae:

Definition 3.8

$$x^{K'}=y \quad \equiv (\mathrm{Form}^+(x) \wedge \exists w\,\exists z\,(\mathrm{PT}(w,x) \wedge \mathrm{KT}(z,w) \wedge y=\mathrm{Root}\beta(z))) \vee (\neg\mathrm{Form}^+(x) \wedge y=0)$$

Lemma 3.3 and 3.7 show that $(\cdot)^{K'}$ is a function, i.e.

Lemma 3.9
$S^1_2 \vdash \forall x\, \exists! y\, (x^{K'}=y)$.

Moreover, inspection of the definitions of PT and KT makes clear that S^1_2 proves that $(\cdot)^{K'}$ acts on atomic formulae in the desired way and that $(\cdot)^{K'}$ commutes with $\neg, \rightarrow$ and $\exists$:
Reason in S^1_2. Suppose x is a formula, and NEG(x,t). The parsing tree w for x readily gives us a unique parsing tree w' for t: take for w' the subtree of w with root t, this is the SubSeq(3, Len(w)+1, w). By Lemma 3.2 there is a z' such that KT(z',w'). Let z be such that KT(z,w), then by unicity of KT we have z'=SubSeq(3, Len(w)+1, w). We conclude:
$\ulcorner(\neg\urcorner t^{K'}\ulcorner)\urcorner = \ulcorner(\neg\urcorner \mathrm{Root}\beta(z')\ulcorner)\urcorner = x^{K'}$.
Likewise for $\rightarrow$ and $\exists$.
This shows:

Lemma 3.10
$S^1_2 \vdash (\ulcorner R(x)\urcorner)^{K'} = \mathrm{Sub}(\mathcal{S}(R(x), \ulcorner y\urcorner, \ulcorner K(y)\urcorner)$
$S^1_2 \vdash \mathrm{DNeg}(x,y) \rightarrow \mathrm{DNeg}(x^{K'}, y^{K'})$
$S^1_2 \vdash \mathrm{IMPL}(x,y,z) \rightarrow \mathrm{IMPL}(x^{K'}, y^{K'}, z^{K'})$
$S^1_2 \vdash \mathrm{EForm}(x) \wedge \mathrm{EVar}(x)=y \wedge B(x)=z \rightarrow x^{K'} = (0 * \ulcorner((\exists\urcorner) ** (y * \ulcorner)(\urcorner) ** (\ulcorner D(y)\urcorner * \ulcorner\wedge\urcorner) ** z^{K'})$.

As in paragraph 2, the translation $(\cdot)^K$ will be constructed from the translation $(\cdot)^{K'}$ by adding to $(\cdot)^{K'}$ a hypothesis which expresses that all free variables in the formula in question are in the domain.
It is not difficult to see that we can do the following in S^1_2.
For every formula x we can construct a unique ordered sequence, say y, which contains exactly the free variables v occurring in x. From this sequence y we can construct a unique ordered sequence z which contains for every element v in y, the code for D(v), i.e. for this sequence z we demand:
$\mathrm{Len}(z)=\mathrm{Len}(y) \wedge \forall i \leqslant \mathrm{Len}(z)\, (\beta(i+1,z)=\mathrm{Sub}(\beta(i+1,y), t, \ulcorner D(t)\urcorner))$.
From z we can construct a unique sequence w such that w is a formula which expresses the conjunction of all elements of z. For example, if

$z=\langle z_1,z_2,z_3,z_4\rangle$ then $w=0*\ulcorner(((\urcorner*z_1*\ulcorner\wedge\urcorner*z_2*\ulcorner)\wedge\urcorner*z_3*\ulcorner)\wedge\urcorner*z_4*\ulcorner)\urcorner$. We write this unique sequence z as $\mathcal{D}_x$. In the case x is a formula without free variables, or not a formula at all, we define $\mathcal{D}_x$ as $\ulcorner 0=0\urcorner$. With $\mathcal{D}_x$ we will now Σ_1^b-define the translation $(\cdot)^K$:

Definition 3.11

$$x^K=y \equiv (\mathrm{Form}^+(x) \wedge y=(0*\ulcorner(\urcorner)**(\mathcal{D}_x*\ulcorner\rightarrow\urcorner)**(x^{K'}*\ulcorner)\urcorner)$$
$$\vee\ (\neg\mathrm{Form}^+(x) \wedge y=0)$$

The relation between this function and the translation defined in paragraph 2 is now as we announced it would be: if φ is a formula then the function $(\cdot)^K$ defined just now gives us the code of φ^K.
This function $(\cdot)^K$ is the function that will fulfil Theorem 3.1. To prove this theorem we will prove the formalized analogues of the lemmata 2.10 and 2.13.
The following lemma is the analogue of Lemma 2.10.

Lemma 3.12

Let A and $\mathcal{A}(x)$ be as in the statement of Theorem 3.1, and let K(x) be the formula defined in Definition 2.8. Then
$S_2^1 \vdash \forall x\,(\mathcal{A}(x) \rightarrow \mathrm{Prov}_{S_2^1+\nabla A}(\ulcorner K(I_x)\urcorner))$.

Proof

The proof of this theorem consists in showing that what we can execute in S_2^1 all steps of the proof of Lemma 2.10.
For all standard numbers n we can construct a proof of J(n), with J(·) as in paragraph 2, from $S_2^1+\nabla A$. Hence
(1) $S_2^1 \vdash \mathrm{Prov}_{S_2^1+\nabla A}(\ulcorner J(I_n)\urcorner)$ for all standard n.
In particular $S_2^1 \vdash \mathrm{Prov}_{S_2^1+\nabla A}(\ulcorner J(I_0)\urcorner)$.
Remember that S_2^1 proves $\mathcal{A}(x) \leftrightarrow x=t_1 \vee \ldots \vee x=t_n$, for some finite set of standard numbers t_i. Hence
(2) $S_2^1 \vdash \forall x\,(\mathcal{A}(x) \rightarrow \mathrm{Prov}_{S_2^1+\nabla A}(\ulcorner J(I_x)\urcorner)$.
From the proof of Lemma 2.10 it follows that
$S_2^1 \vdash \mathcal{A}(x) \rightarrow (\nabla A \rightarrow x\in\mathrm{Leaf}(\mathrm{Br}(0)))$. With Σ_1^b-completeness applied to $\mathcal{A}(x)$ it follows that $S_2^1 \vdash \mathcal{A}(x) \rightarrow \mathrm{Prov}_{S_2^1}(\ulcorner\nabla A \rightarrow I_x\in\mathrm{Leaf}(\mathrm{Br}(I_0))\urcorner)$, i.e.
(3) $S_2^1 \vdash \mathcal{A}(x) \rightarrow \mathrm{Prov}_{S_2^1+\nabla A}(\ulcorner I_x\in\mathrm{Leaf}(\mathrm{Br}(I_0))\urcorner)$.

From (1), (2) and (3), and the definition of K(x) as J(x) $\wedge$ $\exists$n (J(n) $\wedge$ x$\in$Leaf(Br(n)) we see that

$S^1_2 \vdash \mathcal{A}(x) \rightarrow \mathrm{Prov}_{S^1_2 + \nabla A}(\ulcorner K(I_x) \urcorner)$. ⊠

The next lemma is the formal version of Lemma 2.13. To state this lemma, we need the generalization of the function $\mathcal{S}$, which we defined in Definition 3.5, to all formulae.

I.e., we need a (Σ^b_1-definable) function $\mathcal{S}$, which replaces every free variable in a formula of L^+ by its special constant. For atomic formulae we take $\mathcal{S}$ like in Definition 3.5. The generalization of this $\mathcal{S}$ for atomic formulae will be: replace in the $^+$formula x all occurrences of atomic formulae of which the variable is free in x, by their $\mathcal{S}$-translation. We can Σ^b_1-define $\mathcal{S}$ for a formula x with the same method we used for the definition the $(\cdot)^{K'}$-translation:

Construct from the parsing tree for x, from bottom to top, a tree that contains in its leaves

(1) the $\mathcal{S}$ of the atomic subformulae of x if the variable in such an atomic subformula is free in x,

(2) it contains the atomic subformula itself, if its free variable is not free in x.

The tree so constructed will contain $\mathcal{S}(x)$ in its root. We will not spell out the definitions and the existence and unicity proofs, because they very much resemble the definition of the translation tree for $(\cdot)^{K'}$, and the proofs use the same tricks as the proof of Lemma 3.6. The fact that $\mathcal{S}(x)$ is linear in x (see the discussion preceding Lemma 3.7) and that the tree will not consist of more than |x| nodes, provides us with a bound for the tree. We define $\mathcal{S}(x) = \ulcorner I_x \urcorner$, if $\neg$Form$^+$(x), to get $\mathcal{S}(x)$ as a function.

We define a Σ^b_1-definable function

Definition 3.13

$K(x) \equiv \mathrm{Sub}(\mathcal{S}(x), \ulcorner y \urcorner, \ulcorner K(y) \urcorner)$.

The following lemma is a formalized version of Lemma 2.14.

Lemma 3.14

$S^1_2 \vdash \forall x\,(\mathrm{Form}^+(x) \rightarrow \mathrm{Prov}_{S^1_2}((0*\ulcorner(\urcorner)**(K(I_x)*\ulcorner\rightarrow\urcorner)**(x^K*\ulcorner)\urcorner)))$

Proof

We will prove this lemma more or less in the manner we proved Lemma 2.14; we will apply length-induction to n in the formula

$\mathrm{Form}^+(x) \wedge \mathrm{Deg}(x)=n \rightarrow \mathrm{Prov}_{S^1_2}((0*\ulcorner(\urcorner)**(K(I_x)*\ulcorner\rightarrow\urcorner)**(x^K*\ulcorner)\urcorner))$.

L-induction can be applied for the following reason: In the construction of the S^1_2-proofs of $(K(I_x)*\ulcorner\rightarrow\urcorner*x^K)$ for the different cases, only a fixed set of S^1_2-proofs is used. Instantiation of these proofs with x is an operation which requires only substitution, so that we can bound the length of the resulting proofs with a polynomial in the length of x.

We will in this proof sometimes omit codes for brackets (and) for reasons of readability.

First we treat the case Deg(x)=0.

From the definition of the translation $(\cdot)^K$ (Definition 3.10), we immediately see that

$S^1_2 \vdash \mathrm{Form}(x) \wedge \mathrm{Deg}(x)=0 \rightarrow x^K=\mathcal{D}_x*\ulcorner\rightarrow\urcorner**K(I_x)$.

We also know that

$S^1_2 \vdash \mathrm{Form}(x) \wedge \mathrm{Deg}(x)=0 \rightarrow \exists y\subseteq_p x\,(\mathrm{Var}(y) \wedge x=\ulcorner(R\urcorner*y*\ulcorner)\urcorner)$.

This shows that

$S^1_2 \vdash \mathrm{Form}(x) \wedge \mathrm{Deg}(x)=0 \rightarrow$

$$\exists y\subseteq_p x\,(\mathrm{Var}(y) \wedge x^K=(0*\ulcorner(\urcorner)**(D(y)*\ulcorner\rightarrow\urcorner)**(K(I_x)*\ulcorner)\urcorner)$$
$$=\ulcorner R(y)\urcorner^K.$$

From Lemma 2.14 we know that

$S^1_2 \vdash \mathrm{Var}(y) \rightarrow (D(y) \rightarrow (K(\ulcorner R[y]\urcorner) \rightarrow R(y)^K))$.

Hence, by application of Σ^b_1-completeness of S^1_2, and the properties of the provability predicate, we get

$S^1_2 \vdash \mathrm{Form}(x) \wedge \mathrm{Deg}(x)=0 \rightarrow \mathrm{Prov}_{S^1_2}(K(I_x)*\ulcorner\rightarrow\urcorner*x^K))$.

This proves the case Deg(x)=0.

From the p-inductive definition of Form^+ we know:

$$S^1_2 \vdash \mathrm{Deg}(x)>0 \rightarrow \mathrm{DN}(x) \vee \mathrm{NIMP}(x) \vee \mathrm{NEForm}(x)$$
$$\vee\ \mathrm{IMP}(x) \vee \mathrm{EForm}(x)$$
$$\vee\ \exists y\subseteq_p x\,(\mathrm{NEG}(x,y) \wedge \mathrm{AtForm}^+(y)).$$

Reason in S^1_2 to get the induction step.

Suppose $\mathrm{Form}(x) \wedge \mathrm{Deg}(x)<n \rightarrow \mathrm{Prov}_{S^1_2}(K(I_x)*\ulcorner\rightarrow\urcorner*x^K)))$.

We will prove the induction step for the cases of double negations and existential formulae, and leave the other cases to the reader.

- Let Form(z) $\wedge$ Deg(z)=n $\wedge$ DNeg(z,x). Then Deg(x)<n and Form(x), so by hypothesis $Prov_{S^1_2}(K(I_x)*^\ulcorner \rightarrow {}^\urcorner * x^K))$. Also, by Σ^b_1 -completeness applied to Lemma 2.9,

DNeg(z,x) $\rightarrow$ $Prov_{S^1_2}(K(I_z)*^\ulcorner \rightarrow {}^\urcorner * K(I_x)))$. Because $(\cdot)^{K'}$ commutes with the logical connectives

$z^K = \mathcal{D}_z *^\ulcorner \rightarrow {}^\urcorner *(\neg\neg x)^{K'} = \mathcal{D}_z *^\ulcorner \rightarrow {}^\urcorner ** {}^\ulcorner(\neg\neg {}^\urcorner * x^{K'} *^\ulcorner)^\urcorner)$.

By definition of $\mathcal{D}$ $\mathcal{D}_z = \mathcal{D}_x$. Hence $Prov_{S^1_2}(x^{K'} ** {}^\ulcorner \rightarrow (\neg\neg {}^\urcorner * x^{K'} *^\ulcorner)^\urcorner)$.

Several applications of formalized modus ponens give us

$S^1_2 \vdash$ Form(z) $\wedge$ Deg(z)$\leqslant$n $\wedge$ DN(z)$\rightarrow$ $Prov_{S^1_2}(K(I_z)*^\ulcorner \rightarrow {}^\urcorner * z^K)$.

Let Form(z) $\wedge$ Deg(z)=n $\wedge$ EForm(z). Let x=Sub(SpeCon(z), EVar(z), B(z)). Then Deg(x)<n. Let EVar(z)=t. Then by Lemma 2.9 we have $Prov_{S^1_2}(K(I_z)*^\ulcorner \rightarrow \exists t\, (D(t) \wedge {}^\urcorner * K(I_x)))$.

By hypothesis we have $Prov_{S^1_2}(K(I_x)*^\ulcorner \rightarrow {}^\urcorner * x^K)$. By applying, under the provability predicate of S^1_2, the reasoning which we used in the proof of Lemma 2.14 for this case we get $Prov_{S^1_2}(K(I_z)*^\ulcorner \rightarrow {}^\urcorner * z^K)$.

We conclude, after application of LIND:

$S^1_2 \vdash \forall x\, (Form(x) \rightarrow Prov_{S^1_2}(K(I_x)*^\ulcorner \rightarrow {}^\urcorner * x^K))$. ⊠

We complete the proof of Theorem 3.1 by remarking that in S^1_2, $\mathcal{A}(x)$ implies $Form^+(x)$ and $\mathcal{D}_x = {}^\ulcorner 0=0^\urcorner$ and $\mathcal{S}(x) = {}^\ulcorner I_x{}^\urcorner$. Then from Lemma 3.12 we get $S^1_2 \vdash \forall x\, (\mathcal{A}(x) \rightarrow Prov_{S^1_2}(K(I_x)*^\ulcorner \rightarrow {}^\urcorner * x^K))$. This, together with Lemma 3.12 proves Theorem 3.1.

We also have theorem interpretability, i.e. the formalized version of Corrollary 2.14.

Corollary 3.15

Let A be a finitely axiomatized relational theory, axiomatized by closed formulae.

Let $\mathcal{A}$ be some Δ^b_1-predicate with one free variable, such that $\mathcal{A}(x)$ expresses adequately that x is the code of an axiom of A.

Then there is an S^1_2-function $(\cdot)^K$, which associates with every code of a formula in the language of A, a code of a formula in the language of S^1_2, and which behaves like an interpretation, such that

$S^1_2 \vdash \forall x\, (Prov_A(I_x) \rightarrow Prov_{S^1_2 + \nabla A}(x^K))$.

Proof

The proof is an analogon of the proof of Corollary 2.14. Suppose $Proof_A(p,x)$. p is a sequence of formulae, say $p=\langle p_1,p_2,...,p_n\rangle$. Let p^K be the unique sequence that contains the K-translations of the formulae in p in the same order in which they occur in p, i.e., $p^K=\langle p_1{}^K, p_2{}^K, \dots, p_n{}^K\rangle$. Following the same reasoning which we used to get the desired bounds for the translation tree of Lemma 3.7, we know that p^K is linear in p. Let $a=\langle a_1,a_2,...,a_m\rangle$ be the unique ordered subsequence of p such that $(i<j\leqslant m \rightarrow a_i \neq a_j) \wedge (z \epsilon a \leftrightarrow z \epsilon p \wedge \mathcal{A}(z))$. Theorem 3.1 states the existence in S^1_2 of $S^1_2+\nabla A$ proofs q_i for $a_i{}^K$, for every a_i from a. Now consecutively replace the a_i by q_i, to get the sequence q_1 ** q_2 ** ... ** q_m. As we saw in paragraph 2, the result of the concatenation of q_1 ** q_2 ** ... ** q_m with p^K is almost a proof in $S^1_2+\nabla A$ for x^K. We only have to add to this sequence some elements which are instantiations of the rule $(\varphi \rightarrow \forall x\ \psi) \rightarrow \forall x\ (\varphi \rightarrow \psi)$, where x is not free in φ, to get a sequence which indeed is a proof of x^K from $S^1_2+\nabla A$. Because we add at most Len(p) such elements, all of which are bounded in length by $2 \cdot |p|$, we can safely assume existence of the result of this procedure. S^1_2 verifies that this new sequence indeed is a proof $S^1_2+\nabla A$ for x^K. ⊠

§ 4 The model existence lemma for infinite theories

In this paragraph we will show that the two versions of the model existence lemma proven above, (Theorem 2.12 and Theorem 3.1) and their respective corollaries (Corollary 2.15 and Corollary 3.14), also hold for infinite theories.
We assume in this section that the axioms of the theory A are described by a unary predicate $\mathcal{A}$ such that $\mathbb{N} \models \mathcal{A}(n)$ iff n is the code of an axiom of A. It will be shown that formalized model existence is provable for an infinite theory A if $\mathcal{A}$ is Σ^b_1. This is not a serious constraint, because Δ^b_1-predicates are already suitable for the description of the set of axioms of most theories. Predicates of this class can describe the *form* of a formula, i.e. are suitable to characterize theories that have a limited number of axiom schemes.

We start by redefining the notions of tableau and systematic tableau.
As we announced in paragraph 1, we have to admit addition of axioms to the nodes of a tableau for an infinite theory A, and we do so by adding an extra tableau rule. We call this rule **EX**, and it says: add to the node X a finite number of axioms of A, i.e.:
$EX(x,y) \equiv FormSeq(x) \wedge FormSeq(y) \wedge I(x,y) \wedge \forall z \in y\, (z \notin x \rightarrow \mathcal{A}(z))$.
We formalize the systematic version of the extra rule in two stages.

Definition 4.1.a

$$\mathit{EX}(x,y,t) \equiv FormSeq(x) \wedge FormSeq(y) \wedge I(x,y) \wedge [(\mathcal{A}(t) \wedge y=x * t) \vee (\neg\mathcal{A}(t) \wedge y=x\,)]$$

In a systematic tableau w for an infinite theory A we apply *EX* to all nodes whose depth in w is odd, and we do this in the following way: if the depth of node x is $1+2\cdot t$ the successor of x will be x concatenated with t, provided t is an axiom of A, otherwise the successor of x is identical to x, i.e. if y is the successor of node x, and x is on depth $1+2\cdot t$ then we have *EX*(x,y,t). Hence in systematic tableaux we apply the following rule:

Definition 4.1.b

$EX(i,j,w) \equiv Depth(i,w)=1+2\cdot t \wedge \mathit{EX}(\beta(i,w)\dot{-}2, \beta(j,w)\dot{-}2, t)$

We modify the definition of systematic tableaux as follows:

- We fix standard number N, sufficiently large, such that $S^1_2 \vdash \exists x\,(x \in N) \wedge \forall x\,(x \in N \rightarrow \mathcal{A}(x))$. We replace the clause for the root of the tableau x by the following clause:
  $Depth(i,x)=0 \rightarrow \beta(i,x)=N+2;$
- τ is applied to nodes on depth 0 mod(10);
- α is applied to nodes on depth 2 mod(10);
- β is applied to nodes on depth 4 mod(10),
  that is, we apply the following β_i-rules:
  $$\beta_i(i,j,w) \equiv Tree(w) \wedge \exists k \leqslant w\,(k=Depth(i,w) \wedge t=\pi_1(\lfloor {}^{k \dot{-} 4}/{}_{10} \rfloor) \wedge \beta_i(\beta(i,w)\dot{-}2, \beta(j,w)\dot{-}2, t)$$
- γ is applied to nodes on depth 6 mod(10);
- δ is applied to nodes on depth 8 mod(10);
- add the clause that EX is applied to nodes on depth 1 mod(2)

Redefining the predicate $Tab_A(\cdot)$ only requires the following change:

- In the clause for open nodes with valence 1 add as an alternative possibility the application of the non-systematic extra rule, i.e. add the following clause:
  $\vee\ EX(\beta(i,x)\dot{-}2, \beta(j,x)\dot{-}2))$.

Note that if A is not Δ^b_1-axiomatized, then Tab_A is, because of the occurrence of $\mathcal{A}$ in the extra rule, no longer a Δ^b_1-predicate.

We will prove the following analogue of Theorem 2.12.

Theorem 4.2

If A is a relational theory, axiomatized by closed axioms, whose set of axioms can be described by a unary arithmetic formula $\mathcal{A}$, then there is an interpretation $(\cdot)^K$ such that for all axioms φ of A, $S^1_2+\nabla A \vdash \varphi^K$.

We will prove Theorem 4.2 by suitably adjusting the proof of Theorem 2.12.

It is not dificult to see how we have to adjust the definitions of the predicates Bra(x) and Br(n)=y for infinite theories. We need not bother about the complexity of these definitions, cf. the discussion following Definition 2.2.

Just as in the case of the other tableau rules we can define a (binary) function Ex, which gives the tableau successor of a sequence of formulae under application of the extra rule (cf. Lemma 2.6):

$Ex(x,t)=y \equiv (FormSeq(x) \wedge \mathit{EX}(x,y,t)) \vee (\neg FormSeq(x) \wedge y=0)$

Note that if A is not Δ_1^b-axiomatized, then Ex is not a Σ_1^b-defined function. It is however not difficult to see that Ex is a function in S^1_2, and that is enough for our purposes.

Again, Pudlák's induction theorem provides us with a $S^1_2+\nabla A$-cut J on which Br is a total function, provided $\exists! y\ Br(x)=y$ is inductive in $S^1_2+\nabla A$.

To show that $\exists! y\ Br(n)=y$ is an inductive formula also in the case that A is an infinite theory, we need the following lemma to complete Lemma 2.2.

Lemma 4.3

$S^1_2 \vdash \nabla x \wedge \mathit{EX}(x,y,t) \rightarrow \nabla y$

Proof

$\neg\nabla y$ gives us a closed tableau t from y; concatenate x+2 with [t] : this gives a closed tableau from x. ⊠

The definitions of K and D and the translation $(\cdot)^K$ are the same as in the finite case; Lemma 2.9, 2.13 and 2.14 are also true in the infinite case, and have the same proofs (with some numerical changes in the case of the proof of Lemma 2.9).

The missing link in the proof of the model existence lemma for infinite theories is the proof of Lemma 2.10:

If φ is an axiom of A, then $S^1_2+\nabla A \vdash K(\ulcorner\varphi\urcorner)$.

Suppose φ is an axiom of A. Because $\ulcorner\varphi\urcorner$ is a standard number and $\exists! y\, Br(n)=y$ is inductive in $S^1_2+\nabla A$, we get a unique $Br(1+2\cdot\ulcorner\varphi\urcorner+1)$ in $S^1_2+\nabla A$. From the definition of the extra rule it follows directly that

$\ulcorner\varphi\urcorner$ is in the leaf $Br(1+2\cdot\ulcorner\varphi\urcorner+1)$. Also, because $\ulcorner\varphi\urcorner$ is standard and J is an initial, $S^1_2+\nabla A$ proves $J(\ulcorner\varphi\urcorner)$ and $J(1+2\cdot\ulcorner\varphi\urcorner+1)$.
We may conclude that if φ is an axiom of A, $S^1_2+\nabla A$ proves $K(\ulcorner\varphi\urcorner)$.
This completes the proof of Theorem 4.2.

As in the case of finite theories, we also have theorem interpretability for infinite theories, i.e. Corrolary 2.14 is also true for infinite theories. This is proven in exactly the same way as it was done for finite theories A in paragraph 2.

Formalized version for infinite theories

We will prove Theorem 3.1 for infinite theories, i.e.

Theorem 4.4
Let A be relational theory with infinitely many axioms, all of which are closed. Let $\mathcal{A}(\cdot)$ adequately describe the set of axioms of A, and let $\mathcal{A}$ be equivalent in S^1_2 to some Σ^b_1 formula . Then there is an S^1_2-function $(\cdot)^K$ from codes of formulae in the language of A to codes of formulae in the language of S^1_2, which constitutes an interpretation, such that
$S^1_2 \vdash \forall x\, (\mathcal{A}(x) \rightarrow Prov_{S^1_2+\nabla A}(x^K))$.

Note that, in contrast to the unformalized version of this theorem in the case A is an infinite theory, we need for this theorem that A can be Σ^b_1 axiomatized.

We need the lemmata 3.12 and 3.13 for infinite theories A.
Lemma 3.9 is also true if A is infinite.
The proof of Lemma 3.12 does not give any difficulties: just take a version of the predicate Bra that expresses branchhood of the systematic tableau of the *infinite* theory A.

To prove Lemma 3.13 however, which reads $S^1_2 \vdash \forall x\, (\mathcal{A}(x) \rightarrow Prov_{S^1_2+\nabla A}(\ulcorner K(I_x)\urcorner)$, we need the following property of initials:

Lemma 4.5
Let T be a Σ_1^b-axiomatized theory extending S^1_2. Let J be an initial in T, i.e T proves J to be closed under S, + and $\cdot$, then

$S^1_2 \vdash \forall x\, Prov_T(\ulcorner J(I_x)\urcorner)$.

Proof
Because J is an initial in T, we have fixed numbers p_0, p_+, and $p_\cdot$, such that

$S^1_2 \vdash Proof_T(p_0, \ulcorner J(0)\urcorner)$
$S^1_2 \vdash Proof_T(p_+, \ulcorner \forall y\, (J(y) \rightarrow J(SS0\cdot y+S0))\urcorner)$
$S^1_2 \vdash Proof_T(p_\cdot, \ulcorner \forall y\, (J(y) \rightarrow J(SS0\cdot y))\urcorner)$.

To get a proof π_k such that $S^1_2 \vdash Proof_T(\pi_k, \ulcorner J(I_k)\urcorner)$, consider the construction of I_k. It takes $|k|$ steps to get from I_0 to I_k; in each of those steps we go from I_t to I_{2t} or I_{2t+1}. To each step corresponds one of the proofs p_0, p_+, and $p_\cdot$. Hence the proof π_k of $J(I_k)$ will be the concatenation of
(1) the proofs corresponding to the steps we have to take to get I_k,
(2) the necessary instantiations of the conclusions of these proofs, and
(3) necessary applications of modus ponens.
The instantiations (2) are instantiations of $\forall y\, (J(y) \rightarrow J(SS0\cdot y+S0))$ or $\forall y\, (J(y) \rightarrow J(SS0\cdot y))$ with a canonical term I_t with $t<k$, hence we have a bound $|k|\cdot B$, with B some fixed constant, for the length of each of them. Modus ponens is then applied to these instantiations.
Hence we can bound $|\pi_k|$ with $|k|\cdot(C\cdot|k|+D)$ for some fixed constants C and D, where $D= 2\cdot Max\{|p_0|, |p_+|, |p_\cdot|\}+2$, and C is $2\cdot(B+2)$.This shows that in $\exists\pi\, (Proof_T(\pi, \ulcorner J(I_k)\urcorner)$ we can bound $|\pi|$ by $|k|\cdot(C\cdot|k|+D)$, i.e. we can use PIND to prove the lemma.

-If $x=0$, we have $S^1_2 \vdash Proof_T(p_0, \ulcorner J(x)\urcorner)$.

-Suppose that for $z=\lfloor\frac{1}{2}x\rfloor$ we have proof π such that

$S^1_2 \vdash |\pi|\leqslant|z|\cdot(C\cdot|z|+D) \wedge Proof_T(\pi, \ulcorner J(I_z)\urcorner)$.

By definition of $\lfloor\frac{1}{2}x\rfloor$ we have $x=SS0\cdot z$ or $x=SS0\cdot z+S0$, hence $I_x=SS0\cdot I_z$, or $I_x=SS0\cdot I_z+S0$.
Let $I_x=SS0\cdot I_z$. Concatenate π with $p_\cdot$ (if $p_\cdot$ is not yet a subsequence of π), and with the instantiation of the conclusion of $p_\cdot$ with I_z, and with $\ulcorner J(SS0\cdot I_z)\urcorner$. The result is a T-proof π' of $J(I_x)$.
Because $z=\lfloor\frac{1}{2}x\rfloor$, we have $|z|=|x|\dot{-}1$. C and D were chosen in such a manner that also $|\pi'|\leqslant|x|\cdot(C\cdot|x|+D)$. ⊠

In case A is a finite theory, all axioms of A appear in the root of the systematic tableau from A, so that the only thing we had to do was prove that the axioms of A were in the leaf of Br(0). If A is infinite and x is the code of an axiom of A, x will in any case be in the leaf of $Br(1+2\cdot x+1)$. We will prove:

(*) $S^1_2 \vdash \forall x\,(\mathcal{A}(x) \to Prov_{S^1_2+\nabla A}(\ulcorner K(I_x)\urcorner)$.

To this end, it suffices to prove

(0) $S^1_2 \vdash \forall x\,(\mathcal{A}(x) \to Prov_{S^1_2+\nabla A}(\ulcorner J(I_x) \wedge J(I_{1+2\cdot x+1})$
$\wedge\ \exists! y\,(Br(I_{1+2\cdot x+1})=y \wedge I_x \in Leaf(y))\urcorner))$.

Because J is an initial in $S^1_2+\nabla A$ we have, by Lemma 4.5,

(1) $S^1_2 \vdash \forall x\ Prov_{S^1_2+\nabla A}(\ulcorner J(I_x) \wedge J(I_{1+2\cdot x+1})\urcorner)$.

By the unicity of $Br(\cdot)$ on J we have

(2) $S^1_2 \vdash \forall x\ Prov_{S^1_2+\nabla A}(\ulcorner J(I_{1+2\cdot x+1}) \to \exists! y\, Br(I_{1+2\cdot x+1}=y\urcorner)$.

Br is defined in such a way that

$S^1_2 \vdash \mathcal{A}(x) \to \forall y(Br(1+2\cdot x+1)=y \to x \in Leaf(y))$.

If $\mathcal{A}$ is Σ^b_1, we get, by verifiable Σ^b_1-completeness of S^1_2,

(3) $S^1_2 \vdash \mathcal{A}(x) \to Prov_{S^1_2}(\ulcorner \forall y\,(Br(I_{1+2\cdot x+1})=y \to I_x \in Leaf(y))\urcorner)$.

Combination of (2) and (3) gives us (0), which proves (*).

This proves Theorem 4.4.

To prove theorem interpretability, i.e. Corollary 3.15, we have to do some additional work. We need a theorem of Parikh [71], in the following form:

Theorem 4.6 Buss [87,p.83]

Let A(x,y) be a Δ_0-formula, and $S^1_2 \vdash \forall x\, \exists y\, A(x,y)$.
Then there is a term r(x) in the language of S^1_2 such that
$S^1_2 \vdash \forall x\, \exists y \leqslant r(x)\, A(x,y)$.

Theorem 4.7

Under the same conditions as in Theorem 4.4,
$S^1_2 \vdash \forall x\,(Prov_A(x) \to Prov_{S^1_2+\nabla A}(x^K)$.

Proof

Reason in S^1_2. Let p be such that $Proof_A(p,x)$. We follow the reasoning of the proof of Corollary 3.15. There we showed that a $(\cdot)^K$-translation p^K

of p exists in S^1_2, and that there is a unique ordered sequence a which contains exactly the axioms of A that occur in p.
From Theorem 4.4 we know that $S^1_2 \vdash y \in a \rightarrow Prov_{S^1_2 + \nabla A}(y^K))$.
With Parikh's theorem this gives us
$S^1_2 \vdash y \in a \rightarrow \exists q \leqslant r(y)\ Proof_{S^1_2 + \nabla A}(q, y^K)$.
Now reason as we did in the proof of Lemma 3.7 and substitute, one by one, all elements y of a by the proof of y^K in $S^1_2 + \nabla A$. As in Lemma 3.7, because we have a bound for all these proofs, we can show existence of the final result of these subsequent substitutions.
Concatenate the result with p^K, and, as in the proof of Corollary 3.15, add necessary instantiations of the axiom $(\varphi \rightarrow \forall x \psi) \rightarrow \forall x (\varphi \rightarrow \psi)$.

⊠

§5 A provability principle for $I\Delta_0+\Omega_1$

Theorem 4.4 readily gives us an interesting provability principle for $I\Delta_0+\Omega_1$ which is a sharpened form of the second Löb condition.

$\Box$ stands here for a standard provability predicate (cf. the introduction). I.e. if T is a theory and ψ is a formula in the language of T, we write

$\Box_T\psi$ for $Prov_T(\ulcorner\psi\urcorner)$.

We write

$\Diamond_T\psi$ for $\neg\Box_T\neg\psi$,

i.e. $\Diamond_T\psi$ expresses that ψ is consistent with T.

We write, if T is a relational theory and ψ is a sentence in the language of T,

$\Delta_T\psi$ for $\exists x\, ClTab_{T+\neg\psi}(x)$,

i.e. $\Delta_T\psi$ expresses that ψ is tableau-provable from A.

Under the same conditions we write

$\nabla_T\psi$ for $\nabla(T+\psi)$,

i.e. for $\neg\exists x\, ClTab_{T+\psi}(x)$. So $\nabla_T\psi$ expresses the tableau-consistency of the theory $T+\psi$. Clearly

$\nabla_T\psi \leftrightarrow \neg\Delta_T(\neg\psi)$.

It is folklore that in $I\Delta_0+\Omega_1$, tableau provability implies standard provability, i.e. that every tableau proof can be transformed into a stadard proof.

It is however not true that standard provability implies tableau provability in $I\Delta_0+\Omega_1$. Consider the transformation of an ordinary proof x into a tableau proof y (see Schwichtenberg [77, Corrolary 2.7.1]). y will be of the order $supexp(|x|, \varrho(x))$, where $\varrho(x)$ is the supremum of the length of the cut-formulae occurring in x, and supexp is defined inductively as follows: $supexp(x,0)=x$, $supexp(x,y+1)=2^{supexp(x,y)}$. In $I\Delta_0+\Omega_1$ superexponentiation is not a total function in $I\Delta_0+\Omega_1$, because exponentiation is not total in $I\Delta_0+\Omega_1$.

The following two lemmata show that in $I\Delta_0+\Omega_1$, standard provability does not imply tableau provability.

Lemma 5.1 (Pudlák [85, Theorem 2.1])
Let T contain Q, be consistent and finitely axiomatizable. Let J be a cut in T. Then
$T \nvdash (\Diamond T)^J$.

Proof
For the proof, see Pudlák [85].

Lemma 5.2
Let T be a relational theory containing Q. Then
$I\Delta_0+\Omega_1 \nvdash \Box_T\varphi \to \Delta_T\varphi$.

Proof
Reason by contraposition, and suppose $I\Delta_0+\Omega_1 \vdash \Box_T\varphi \to \Delta_T\varphi$. Then $I\Delta_0+\Omega_1 \vdash \nabla_T\varphi \to \Diamond_T\varphi$, and in particular
(1) $I\Delta_0+\Omega_1+\nabla(I\Delta_0+\Omega_1) \vdash \Diamond(I\Delta_0+\Omega_1)$.
From a theorem of Paris and Wilkie [87, Lemma 8.10] we know that $I\Delta_0$+EXP proves the tableau consistency of $I\Delta_0+\Omega_1$:
(2) $I\Delta_0+EXP \vdash \nabla(I\Delta_0+\Omega_1)$.
With (1) we then get
(3) $I\Delta_0+EXP \vdash \Diamond(I\Delta_0+\Omega_1)$.
From another result of Paris and Wilkie [87, Corrolary 8.8] we know that for Π^0_1-formulae φ, $I\Delta_0+EXP \vdash \varphi$ iff there is an $I\Delta_0+\Omega_1$-cut J such that $I\Delta_0+\Omega_1 \vdash \varphi^J$.
$\Diamond(I\Delta_0+\Omega_1)$ is a Π^0_1-formula, so we can aply this result to (4) to get an $I\Delta_0+\Omega_1$-cut J such that $I\Delta_0+\Omega_1 \vdash (\Diamond(I\Delta_0+\Omega_1))^J$.
Let V_0 be a finite fragment of $I\Delta_0+\Omega_1$ such that J is a cut in V_0. Let V_1 be a finite fragment of $I\Delta_0+\Omega_1$ such that $V_1 \vdash (\Diamond(I\Delta_0+\Omega_1))^J$. Take $V=V_0+V_1+Q$. Then
(4) $V \vdash (\Diamond(I\Delta_0+\Omega_1))^J$.
By the definition of the provability predicate it is clear that $\Diamond(I\Delta_0+\Omega_1)$ logically implies $\Diamond V$. Thus
(5) $\Diamond(I\Delta_0+\Omega_1) \vdash \Diamond V$,
From (4) and (5) however, we can derive a contradiction with Theorem 5.1:

Let π be a proof of $\Diamond V$ from $\Diamond(I\Delta_0+\Omega_1)$. We can associate with π a proof π^J of $(\Diamond V)^J$ from $(\Diamond(I\Delta_0+\Omega_1))^J$, by replacing every formula in π by its $(\cdot)^J$-translation (confer the proof of Corollary 2.14). Combine π^J with a proof of $(\Diamond(I\Delta_0+\Omega_1))^J$ from V.
Thus we get
(8) $V \vdash (\Diamond V)^J$. ⊠

This proof also shows that $I\Delta_0+EXP \nvdash \Diamond(I\Delta_0+\Omega_1)$.

The following theorem shows that although tableau provability is not equivalent to standard provability in $I\Delta_0+\Omega_1$, standard provability implies the provability of tableau provability.
The theorem can also be seen as a sharper form of the second Löb condition. From the discussion preceding Lemma 5.1 it follows that in general tableau proofs are considerably larger than standard proofs. Still, as the following theorem shows, standard provability does not only imply the provable existence of a standard proof, but also the provable existence of a tableau proof.

Theorem 5.3 (Visser [88b])
Let T be a relational theory, of which the set of axioms, all of which are sentences, can be described by a Σ_1^b-formula. Let φ be a formula in the language of T.
Then we have the following provability principle for $I\Delta_0+\Omega_1$:

$$I\Delta_0+\Omega_1 \vdash \Box_T\varphi \rightarrow \Box_{I\Delta_0+\Omega_1}\Delta_T\varphi.$$

Proof
Reason in $I\Delta_0+\Omega_1$.
Suppose (1) $\Box_T\varphi$.
Then $\Box_T(\neg\varphi \rightarrow \bot)$, so by the Deduction Theorem (2) $\Box_{T+\neg\varphi}\bot$. I.e. $I\Delta_0+\Omega_1$ verifies the existence of a proof of $\bot$ (i.e. a proof of $\psi \wedge \neg\psi$ for some formula ψ in the language of T) from $T+\neg\varphi$.
Clearly Theorem 4.4 is also true for $I\Delta_0+\Omega_1$, i.e. there is an $I\Delta_0+\Omega_1$-function $(\cdot)^K$ which constitutes an interpretation such that
(3) $I\Delta_0+\Omega_1 \vdash \Box_{T+\neg\varphi}(\psi) \rightarrow \Box_{I\Delta_0+\Omega_1+\nabla(T+\neg\varphi)}(\psi^K)$ for all ψ in the language of T.

Note that, by commutation of $(\cdot)^K$ with $\neg$, the $(\cdot)^K$-translation of $\bot$ is just $\bot$.
(3) showes that from (2) we get (4) $\Box_{I\Delta_0+\Omega_1+\nabla(T+\neg\varphi)}(\bot)$. By the properties of the provability predicate it is trivial that from (4) we get (5) $\Box_{I\Delta_0+\Omega_1}(\nabla(T+\neg\varphi) \rightarrow \bot)$, i.e.
(6) $\Box_{I\Delta_0+\Omega_1}(\neg\nabla(T+\neg\varphi))$, which gives us $\Box_{I\Delta_0+\Omega_1}(\neg\nabla_T\neg\varphi)$, which is by definition of ∇ and Δ,
(7) $\Box_{I\Delta_0+\Omega_1}(\Delta_T\varphi)$. ⊠

For another proof of this theorem, see Visser [88b].

§6 An Orey sentence for $I\Delta_0+\Omega_1$

The motivation for the material in this section was given in the introduction.
We will use the notations developed in paragraph 5 with the following modification: we will write $\Delta_{I\Delta_0+\Omega_1}(\ulcorner\varphi\urcorner)$ and $\nabla_{I\Delta_0+\Omega_1}(\ulcorner\varphi\urcorner)$ for $\Delta_{I\Delta_0+\Omega_1}(\varphi)$ resp. $\nabla_{I\Delta_0+\Omega_1}(\varphi)$.

If I is a cut in the theory T, then we can use I to define a special kind of interpretation, which is called relativized interpretation:
$\varphi^I \equiv \varphi$ if φ is atomic,
$(\exists x\varphi)^I \equiv \exists x\,(I(x) \wedge \varphi^I)$
$(\neg\varphi)^I \equiv \neg\varphi^I$,
$(\varphi \to \psi)^I \equiv \varphi^I \to \psi^I$.
The interpretation thus defined only relativizes quantifiers.

If I is an interpretation which interprets all axioms of a theory T in a theory D, we will write $D \vdash T^I$.

Theorem 6.1(Visser [88a])**
There exists an Orey sentence for $I\Delta_0+\Omega_1$, in the following sense.
There is a sentence G, an $I\Delta_0+\Omega_1$-cut I and an interpretation C, such that
$I\Delta_0+\Omega_1 \vdash (I\Delta_0+\Omega_1+G)^I$
$I\Delta_0+\Omega_1 \vdash (I\Delta_0+\Omega_1+\neg G)^C$.

** When this thesis was completed, I learned about a letter from Solovay to Nelson, dated May 12, 1986, in which Solovay gives a Orey sentence for $I\Delta_0+\Omega_1$. This Orey sentence is constructed in a very elegant way, using a function Log*(x), which approximates the inverse of supexp(1,y) (i.e. it has as its value approximately the number of times the ^{2}Log of x should be taken to arrive at 1). Using Log*(x) Solovay also constructs two sentences φ and ψ, both interpretable on a cut in $I\Delta_0+\Omega_1$, such that the sentence $\varphi\wedge\psi$ (which is equivelent to EXP) cannot be interpreted in $I\Delta_0+\Omega_1$.

The theorem is a consequence of a series of lemmas.
We have Gödel's diagonalization lemma in $I\Delta_0+\Omega_1$, i.e.

Lemma 6.2
Let $\psi(x)$ be a formula with only one free variable x. Then there is a sentence φ such that $I\Delta_0+\Omega_1 \vdash \varphi \leftrightarrow \psi(\ulcorner\varphi\urcorner)$.

A proof of this lemma can be found in Buss [87, p.124].

In the sequel we take G to be the sentence such that
$$I\Delta_0+\Omega_1 \vdash G \leftrightarrow \neg\Delta_{I\Delta_0+\Omega_1}(\ulcorner G\urcorner).$$
Note that in the expression $\Delta_{I\Delta_0+\Omega_1}$ we take the relational version of $I\Delta_0+\Omega_1$.

Lemma 6.3
$I\Delta_0+EXP \vdash G$

Proof
(1) By definition of G we have $I\Delta_0+\Omega_1 \vdash \Delta_{I\Delta_0+\Omega_1}(\ulcorner G\urcorner)\rightarrow \neg G$.
$I\Delta_0+EXP$ is stronger than $I\Delta_0+\Omega_1$, so
$I\Delta_0+EXP \vdash \Delta_{I\Delta_0+\Omega_1}(\ulcorner G\urcorner)\rightarrow \neg G$.
(2) On the other hand, G is a Π^0_1-sentence. From a result of Paris and Wilkie [87, Lemma 8.10] we know that $I\Delta_0+EXP$ proves tableau reflection for Σ^0_2-sentences: $I\Delta_0+EXP \vdash \Delta_{I\Delta_0+\Omega_1}(\ulcorner\sigma\urcorner)\rightarrow \sigma$, if σ is Σ^0_2.
Hence $I\Delta_0+EXP \vdash \Delta_{I\Delta_0+\Omega_1}(\ulcorner G\urcorner)\rightarrow G$.
(1) and (2) show that $I\Delta_0+EXP$ proves $\neg\Delta_{I\Delta_0+\Omega_1}(\ulcorner G\urcorner)$. This proves the lemma. ⊠

Lemma 6.4
There exists an $I\Delta_0+\Omega_1$-initial I such that $I\Delta_0+\Omega_1 \vdash (I\Delta_0+\Omega_1+G)^I$.

Proof
From Corollary 8.8 in Paris and Wilkie [87] we know that for Π^0_1-formulae φ, $I\Delta_0+EXP \vdash \varphi$ iff there is an $I\Delta_0+\Omega_1$-cut I such that $I\Delta_0+\Omega_1 \vdash \varphi^I$.

This result gives us the existence of an $I\Delta_0+\Omega_1$-cut I such that $I\Delta_0+\Omega_1 \vdash G^I$. Construct a cut J below I which is closed under $+,\cdot$ and Ω_1, and on which we have induction for Δ_0-formulae. Then
$I\Delta_0+\Omega_1 \vdash (I\Delta_0+\Omega_1+G)^I$. ⊠

Lemma 6.5
There is an interpretation C such that $I\Delta_0+\Omega_1 \vdash (I\Delta_0+\Omega_1+\neg G)^C$.

Proof
From Lemma 6.4 we have, by definition of G:
$I\Delta_0+\Omega_1 \vdash (I\Delta_0+\Omega_1+\nabla_{I\Delta_0+\Omega_1}(\ulcorner\neg G\urcorner))^I$.
Theorem 4.2 tells us that there is an interpretation K such that
$I\Delta_0+\Omega_1+\nabla_{I\Delta_0+\Omega_1}(\ulcorner\neg G\urcorner) \vdash (I\Delta_0+\Omega_1+\neg G)^K$.
Define an interpretation C in the following way: $\varphi^C \equiv (\varphi^K)^I$.
Then
$I\Delta_0+\Omega_1 \vdash (I\Delta_0+\Omega_1+\neg G)^C$. ⊠

This completes the proof of Theorem 6.1.

This Orey sentence for $I\Delta_0+\Omega_1$ is a strong indication that the concept predicative arithmetic, as the collection of sentences interpretable in Q, is not robust. The interpretability of both $I\Delta_0+\Omega_1+G$ and $I\Delta_0+\Omega_1+\neg G$ in $I\Delta_0+\Omega_1$ seems to show that the putative definition of predicative arithmetic is incoherent. One might, however, raise some objections to this view.
(1) The interpretability of $I\Delta_0+\Omega_1+G$ in $I\Delta_0+\Omega_1$ is proven (in the lemmata 6.2-4), by impredicative means.
(2) Lemma 6.5 proves the interpretation of the *relational* version of $I\Delta_0+\Omega_1+\neg G$ in $I\Delta_0+\Omega_1$.
(3) The interpretation of $I\Delta_0+\Omega_1+\neg G$ in $I\Delta_0+\Omega_1$ is not a relativization defined by a cut.
We shall discuss these objections each in turn.

Ad (1). The interpretability of $I\Delta_0+\Omega_1+G$ in $I\Delta_0+\Omega_1$, proven by the lemmata 6.2 and 6.3 and 6.4, is not established predicatively. It is proven via provability in $I\Delta_0+EXP$, a theory which is not interpretable in Q, and thus is not a predicative theory. Moreover, the proofs of Lemma 8.10 and Corollary 8.8 of Paris and Wilkie [87] use modeltheoretic arguments. We conjecture however that is possible to prove in $I\Delta_0+\Omega_1$, on a cut, tableau reflection for Π_1-sentences, by using a restricted truth predicate. This will establish Lemma 6.4 in an entirely predicative way.

Ad (2). By Lemma 6.5 we get an interpretation of the *relational* version of $I\Delta_0+\Omega_1$. However, we conjecture that is is possible to prove Theorem 4.7 also for *functional* theories.

Ad (3). A third objection is that $I\Delta_0+\Omega_1+\neg G$ is interpreted in $I\Delta_0+\Omega_1$ by an interpretation that also translates relation symbols, so that it definitely is not a relativized interpretation defined by a cut.

We should note here that if φ is a sentence in the language of arithmetic and φ is interpretable in $I\Delta_0+\Omega_1$ by relativization on a cut, then φ is true on the standard model $\mathbb{N}$: $\mathbb{N}$ is a model for $I\Delta_0+\Omega_1$, and on this model a relativization defined by a cut does not do anything, because the cut *is* $\mathbb{N}$. This implies that Orey sentences φ for $I\Delta_0+\Omega_1$ such that both φ and $\neg\varphi$ are interpretable in $I\Delta_0+\Omega_1$ by relativization on a cut do not exist.

If one accepts model theoretic arguments, a distinction between general interpretations and relativizations defined by a cut can clearly be made. On nonstandard models of arithmetic, relativization defined by a cut constitutes an initial segment of the model. An interpretation on the other hand, constitutes a model possibly different from $\mathbb{N}$ or the non standard model. One could argue however, that from a finitistic point of view there is no reason to prefer interpretations that only relativize quantifiers to interpretations that also translate relation and function symbols. That is, if one does not accept model theoretic arguments, then the difference between interpretations and relativizations is entirely formal, and it is not at all clear that this formal distinction can motivate the predilection of relativizations defined by a cut above general interpretations.

References

Buss, S., 1985, *Bounded Arithmetic,* Thesis, Princeton University, Princeton. Reprinted: 1986, Bibliopolis, Napoli.

Feferman, S., 1960, *Arithmetization of metamathematics in a general setting,* Fund. Math. 49, 33-92.

Nelson E., 1986, *Predicative Arithmetic,* Math Notes 32, Princeton University Press, Princeton.

Orey, S., 1961, *Relative Interpretations,* Zeitschrift für Mathematische Logik und Grundlagen der Mathematik 7, 146-153.

Parikh, R., 1971, *Existence and feasibility in arithmetic,* JSL 36, 494-508.

Paris, J.B., Dimitracopoulos, C., 1982, *Truth definitions for Δ_0-formulae,* in: *Logic and Algorithmic,* Monography 30 de l'Enseignement Mathematique Genève, 319-329.

Paris, J.B., Dimitracopoulos, C., 1983, *A note on the undefinability of cuts,* JSL 48, 564-569.

Paris, J., Wilkie, A., 1987, *On the scheme of induction for bounded arithmetic formulas,* Annals for Pure and Applied Logic 35, 261-302.

Pudlák, P., 1983a, *Some prime elements in the lattice of interpretability types,* Transactions of the AMS 280, 255-275.

Pudlák, P., 1983b, *A definition of exponentiation by a bounded arithmetical formula,* Commentationes Mathematicae Universitatis Carolinae 24, 667-671.

Pudlák, P., 1985, *Cuts, consistency statements and interpretability,* JSL 50, 423-441.

Pudlák, P., 1988, *Review of Edward Nelson's "Predicative Arithmetic",* JSL 53, 987-989.

Schwichtenberg, H., 1977, *Proof Theory: some applications of cut-elimination,* Handbook of Mathematical Logic, ed. J. Barwise, 867-895.

Smullyan, R.M., 1968, *First Order Logic,* Springer, New York.

Solovay, R., 1986, *Letter to E. Nelson.*

Visser, A., 1988a, *Manuscript.*

Visser, A., 1988b, *Interpretability Logic,* Logic Group Preprint Series nr 40, Dept. of Philosophy, University of Utrecht, Heidelberglaan 2, 3584CS Utrecht.

The ITLI Prepublication Series

1986

86-01	The Institute of Language, Logic and Information
86-02 Peter van Emde Boas	A Semantical Model for Integration and Modularization of Rules
86-03 Johan van Benthem	Categorial Grammar and Lambda Calculus
86-04 Reinhard Muskens	A Relational Formulation of the Theory of Types
86-05 Kenneth A. Bowen, Dick de Jongh	Some Complete Logics for Branched Time, Part I Well-founded Time, Forward looking Operators
86-06 Johan van Benthem	Logical Syntax

1987

87-01 Jeroen Groenendijk, Martin Stokhof	Type shifting Rules and the Semantics of Interrogatives
87-02 Renate Bartsch	Frame Representations and Discourse Representations
87-03 Jan Willem Klop, Roel de Vrijer	Unique Normal Forms for Lambda Calculus with Surjective Pairing
87-04 Johan van Benthem	Polyadic quantifiers
87-05 Víctor Sánchez Valencia	Traditional Logicians and de Morgan's Example
87-06 Eleonore Oversteegen	Temporal Adverbials in the Two Track Theory of Time
87-07 Johan van Benthem	Categorial Grammar and Type Theory
87-08 Renate Bartsch	The Construction of Properties under Perspectives
87-09 Herman Hendriks	Type Change in Semantics: The Scope of Quantification and Coordination

1988

Logic, Semantics and Philosophy of Language:

LP-88-01 Michiel van Lambalgen	Algorithmic Information Theory
LP-88-02 Yde Venema	Expressiveness and Completeness of an Interval Tense Logic
LP-88-03	Year Report 1987
LP-88-04 Reinhard Muskens	Going partial in Montague Grammar
LP-88-05 Johan van Benthem	Logical Constants across Varying Types
LP-88-06 Johan van Benthem	Semantic Parallels in Natural Language and Computation
LP-88-07 Renate Bartsch	Tenses, Aspects, and their Scopes in Discourse
LP-88-08 Jeroen Groenendijk, Martin Stokhof	Context and Information in Dynamic Semantics
LP-88-09 Theo M.V. Janssen	A mathematical model for the CAT framework of Eurotra
LP-88-10 Anneke Kleppe	A Blissymbolics Translation Program

Mathematical Logic and Foundations:

ML-88-01 Jaap van Oosten	Lifschitz' Realizabiility
ML-88-02 M.D.G. Swaen	The Arithmetical Fragment of Martin Löf's Type Theories with weak Σ-elimination
ML-88-03 Dick de Jongh, Frank Veltman	Provability Logics for Relative Interpretability
ML-88-04 A.S. Troelstra	On the Early History of Intuitionistic Logic
ML-88-05 A.S. Troelstra	Remarks on Intuitionism and the Philosophy of Mathematics

Computation and Complexity Theory:

CT-88-01 Ming Li, Paul M.B.Vitanyi	Two Decades of Applied Kolmogorov Complexity
CT-88-02 Michiel H.M. Smid	General Lower Bounds for the Partitioning of Range Trees
CT-88-03 Michiel H.M. Smid, Mark H. Overmars Leen Torenvliet, Peter van Emde Boas	Maintaining Multiple Representations of Dynamic Data Structures
CT-88-04 Dick de Jongh, Lex Hendriks Gerard R. Renardel de Lavalette	Computations in Fragments of Intuitionistic Propositional Logic
CT-88-05 Peter van Emde Boas	Machine Models and Simulations (revised version)
CT-88-06 Michiel H.M. Smid	A Data Structure for the Union-find Problem having good Single-Operation Complexity
CT-88-07 Johan van Benthem	Time, Logic and Computation
CT-88-08 Michiel H.M. Smid, Mark H. Overmars Leen Torenvliet, Peter van Emde Boas	Multiple Representations of Dynamic Data Structures
CT-88-09 Theo M.V. Janssen	Towards a Universal Parsing Algorithm for Functional Grammar
CT-88-10 Edith Spaan, Leen Torenvliet Peter van Emde Boas	Nondeterminism, Fairness and a Fundamental Analogy
CT-88-11 Sieger van Denneheuvel Peter van Emde Boas	Towards implementing RL

Other prepublications:

X-88-01 Marc Jumelet	On Solovay's Completeness Theorem

1989

Logic, Semantics and Philosophy of Language:

LP-89-01 Johan van Benthem	The Fine-Structure of Categorial Semantics

Computation and Complexity Theory:

CT-89-01 Michiel H.M. Smid	Dynamic Deferred Data Structures

Other prepublications:

X-89-01 Marianne Kalsbeek	An Orey Sentence for Predicative Arithmetic