The Basic Feasible Functionals in Bounded Arithmetic

Diplomarbeit

der Philosophisch-naturwissenschaftlichen

Fakultät der Universität Bern

vorgelegt von

Simone Liniger

2008

Leiter der Arbeit: Prof. Dr. Thomas Strahm,

Institut für Informatik und

Angewandte Mathematik der Universität Bern

Abstract

The aim of this work is to elaborate the proof of the following main theorem, which was introduced by Aleksandar Ignjatovic and Arun Sharma in their paper "Some Applications of Logic to Feasibility in Higher Types" from 2004:

> The class BFF of the *Basic Feasible Functionals* is exactly the class of Σ_1^b provably total functions of $\mathbf{S}_2^1$.

That sounds interesting at first glance, but doesn't have any meaning for us, if we are not familiar with contained notions such as "*Basic Feasible Functionals*", "Σ_1^b provably total" or "$\mathbf{S}_2^1$". Well, better so, since this work would be nearly boring for us, if we already knew about those mentioned terms:
Because the first section inter alia contains an explicit introduction of the *Basic Feasible Functionals* and their features.
In the second section we introduce four second order theories, among other things $\mathbf{S}_2^1$, which build the main tools for the proof of the above main theorem.
The third section finally contains the main theorem and its proof.
Now, before rushing into the details, we want to know about the main aspects of this work to keep a rough overview. In this connection, let us firstly say some words about feasibility, which is an important facet, and secondly about our implementation of logic:
For something to be feasibly computable, it must be computable in practice in the real world, not merely effectively computable in the sense of being recursively computable. We already know, that functions are considered to be feasibly computable, if they are computable on a Turing machine in polynomial time. While this formalisation of the notion of a feasibly computable function is commonly accepted, there is no similar agreement in relation to the feasibility of functionals*. Kurt Mehlhorn introduced one possible paradigm for the feasibility of *type*-2 functionals** by extending Alan Cobham's definition of feasible functions from 1964. Thereafter, Mehlhorn's class of functionals was studied by Mike Townsend, who calls the same class *Poly*. This name clearly refers to the fact, that the functionals contained in *Poly* are computable in polynomial time.
About the same time Stephen Cook and Bruce Kapron gave an oracle Tur-

ing machine model characterization of Mehlhorn's class and named it *Basic Feasible Functionals*, abbreviated *BFF*.
Now, let us say some words about our implementation of logic. A special aspect here is, that our proofs are based on essential applications of logic. We introduce a weak fragment of second order arithmetic with second order variables ranging over *type*-1 functionals which suitably characterizes the *Basic Feasible Functionals*. This characterization is a second order extension of Samuel Buss's characterization of the polynomial time computable functions.

Last but not least we want to mention that the two lectures "Introduction into Theoretical Computer Science" and "Logic and Computer Science" are assumed here.

* Functionals are functions which in turn take functions as arguments.

** A *type*-2 functional takes tupels, containing total functions from $\mathbb{N}$ to $\mathbb{N}$ and natural numbers, as arguments.

Acknowledgements.
I am very grateful to Prof. Dr. Thomas Strahm, and also to Dr. Dieter Probst, for their guidance, suggestions and useful discussions. I also want to thank to my boyfriend Samuel Gutmann, to my friends Nora Nussbaum, Diana Fagherazzi, Dr. Lolin Cuenca and to my family, especially to my parents Katharina and Leo Liniger-Blaser for their mental support.

Contents

1 Introduction

1.1 Feasible Functions

As already mentioned in the Abstract, we say that a function is feasibly computable, if it is computable on a Turing machine in polynomial time. Now, Cobham established a fundamental result, which is formulated in the following theorem 1.1.1 and builds the headstone of Townsend's definition 1.2.3 of the $Basic\ Feasible\ Functionals$ more below. We exclusively operate on natural numbers, which are represented in the unbounded case by the letters u, v and w and in the bounded case by x, y and z respectively. We as well receive natural numbers by operating on them with terms, functions or functionals.

1.1.1 Theorem (Cobham)

Functions which are computable on a Turing machine in polynomial time are exactly functions which can be obtained from the basic functions:

$$\begin{aligned} O(u) &= 0, \\ S_0(u) &= 2u, \\ S_1(u) &= 2u+1, \\ Pr_i^n(u_1,\ldots,u_n) &= u_i, \\ Smash(u,v) &= u\sharp v = 2^{|u|\cdot|v|}, \end{aligned}$$

by using *composition* (see definition 1.1.2) and *limited recursion on notation* (see definition 1.1.3).

1.1.2 Definition (composition)

Let f be a function.
Then f is defined from $g_1,\ldots,g_l$ and h by *composition* if for all $\vec{u}$:

$$f(\vec{u}) = h(\vec{u}, g_1(\vec{u}),\ldots,g_l(\vec{u})).$$

1.1.3 Definition (limited recursion on notation)

Let f be a function.
Then f is defined from g, h_0, h_1 and k by *limited recursion on notation*, if for all $\vec{u}, v$:

$$
\begin{aligned}
f(\vec{u}, 0\ ) &= g(\vec{u}), && (1)\\
f(\vec{u}, 2v) &= h_0(\vec{u}, v, f(\vec{u}, v)), \qquad v > 0 && (2)\\
f(\vec{u}, 2v+1) &= h_1(\vec{u}, v, f(\vec{u}, v)), && (3)\\
|f(\vec{u}, v)| &\leq |k(\vec{u}, v)|, && (4)
\end{aligned}
$$

where $|f(\vec{u}, v)| = |w|$ stands for the length of the binary representation of the natural number w, so

$$|w| \quad = \quad \lceil log_2(w+1) \rceil.$$

We replace the above schema 1.1.3 of *limited recursion on notation* by the following schema 1.1.4 of *limited recursion on notation**, where equation (6) combines the above equations (2) and (3), and condition (7) replaces condition (4).

1.1.4 Definition (limited recursion on notation*)

Let f be a function.
Then f is defined from g and h by *limited recursion on notation** if for all $\vec{u}, v$:

$$
\begin{aligned}
f(\vec{u}, 0) &= g(\vec{u}), && (5)\\
f(\vec{u}, v) &= h(\vec{u}, v, f(\vec{u}, \lfloor \frac{1}{2} v \rfloor)), && (6)\\
|f(\vec{u}, v)| &\leq q(|\vec{u}|, |v|), && (7)
\end{aligned}
$$

where q is a polynomial with natural coefficients.

It is easy to see that Cobham's theorem stays the same under application of this new schema 1.1.4 of *limited recursion on notation** in place of schema 1.1.3 of *limited recursion on notation*.

1.2 Basic Feasible Functionals

Before delving into the area of feasible functionals, we have to read some general definitions concerning functionals. In the following definition 1.2.1 we see how the type of a functional, to which we have referred in the Abstract, is defined:

1.2.1 Definition (Functional-types)

The *set of types* is defined inductively as follows:

(i) 0 is a *type*,

(ii) $(\delta \to \tau)$ is a *type*, if δ and τ are *types*.

We denote the $set\ of\ functionals\ of\ type\ \tau$ by $Fn(\tau)$.
$Fn(\tau)$ is defined inductively as follows:

(i) $Fn(0) = \mathbb{N}$,

(ii) $Fn(\delta \to \tau) = \{F \mid F : Fn(\delta) \to Fn(\tau)\}$.

After the above definition 1.2.1 it is easy to show that every *type* has a unique normal form:

$$\delta \quad = \quad \tau_1 \to \cdots \to \tau_k \to 0,$$

where the missing parantheses are put in with association to the right. Hence a functional F of type τ is considered in a natural way as a function of variables $X_1, \ldots, X_k$ with X_i ranging over $Fn(\tau_i)$, and returning a natural number value:

$$F(X_1)(X_2)\ldots(X_k) \quad = \quad F(X_1, \ldots, X_k).$$

Hence

A *type*-0 functional (or function) is a constant $c \in \mathbb{N}$.

A *type*-1 functional (or function) is a total mapping from $\mathbb{N}$ to $\mathbb{N}$. We will denote
the set of all such functionals resp. functions by $\mathbb{N}^{\mathbb{N}}$.

A *type*-2 functional is a total mapping from $(\mathbb{N}^{\mathbb{N}})^k \times \mathbb{N}^l$ to $\mathbb{N}$, for some $k, l \in \mathbb{N}$.

And so on...

We concentrate on *type*-2 functionals in this work, for which the arguments are tupels containing *type*-0 and *type*-1 functionals.

For *type*-≥ 3 functionals there exist several attempts to define polynomial time objects, but there is still no general agreement in the definition of polynomial time computability for them.

1.2.2 Definition (Rank (k,l))

A function $F : (\mathbb{N}^{\mathbb{N}})^k \times \mathbb{N}^l \rightarrow \mathbb{N}$ is called a *type*-2 functional of *rank* (k, l).

We are enough prepared now to read Townsend's definition of the *Basic Feasible Functionals*. He considered them as the least class of functionals, which together with the polynomial time computable functions contains the *Application Functional Ap*, and which is closed under *expansion* (see definition 1.2.6), *functional composition* (see definition 1.2.5), *functional substitution* (see definition 1.2.7) and *limited recursion on notation* (see definition 1.2.8). Townsend has also shown that the scheme of *functional substitution* is redundant and so can be omitted. Thus we introduce the following appropriately modified definition 1.2.3.

1.2.3 Definition (Basic Feasible Functionals BFF)

The class BFF of *Basic Feasible Functionals* is the least class of functionals containing the following initial functions:

$$\begin{aligned} O(u) &= 0, \\ S_0(u) &= 2u, \\ S_1(u) &= 2u+1, \\ Pr_i^n(u_1,\ldots,u_n) &= u_i, \\ Smash(u,v) &= u\sharp v = 2^{|u|\cdot|v|}, \end{aligned}$$

and the *Application Functional*:

$$Ap(f,u) = f(u),$$

which is closed under *functional composition* (see definition 1.2.5), *expansion* (see definition 1.2.6) and *limited recursion on notation* (see definition 1.2.8).

As we see, the class of *Basic Feasible Functionals* BFF extends the class of feasible functions (see theorem 1.1.1) in a minimal way. There are only two differences:

The first one is addition of the *Application Functional* $Ap(f,u)$ to the set of basic functions. And the second difference is addition of *expansion* (see definition 1.2.6) to the set of closure conditions. All the rest coincides completely.
In any reasonable model of feasibility in higher *types*, such a *Basic Feasible Functional* must be considered feasible. The only functions which belong to the class of the BFF's are naturally just the polynomial time computable functions as we see in the following theorem 1.2.4.

1.2.4 Theorem

A *type*-1-functional $F:\mathbb{N}^k \to \mathbb{N}$ is basic feasible iff it is polynomial time computable.

The following definitions for *type*-2 functionals are second order extensions of the corresponding definitions for *type*-1 functionals. The primal difference are certainly the arguments, which now are tupels contaning total functions from $\mathbb{N}$ to $\mathbb{N}$ and natural numbers:

1.2.5 Definition (functional composition)

Let F be a *type*-2 functional.
Then F is defined from $G_1, \ldots, G_l$, H by *functional composition* if:

$$F(\vec{f}, \vec{u}) \quad = \quad H(\vec{f}, \vec{u}, G_1(\vec{f}, \vec{u}), \ldots, G_l(\vec{f}, \vec{u})),$$

for all $\vec{f}, \vec{u}$.

1.2.6 Definition (expansion)

Let F be a *type*-2 functional.
Then F is defined from G by *expansion* if:

$$F(\vec{f}, \vec{g}, \vec{u}, \vec{v}) \quad = \quad G(\vec{f}, \vec{u}),$$

for all $\vec{f}, \vec{g}, \vec{u}, \vec{v}$.

1.2.7 Definition (functional substitution)

Let F be a *type*-2 functional.
Then F is defined from $G_1, \ldots, G_l$, H by *functional substitution*, if:

$$F(\vec{f}, \vec{u}) \quad = \quad H(\vec{f}, \vec{u}, \lambda v.G_1(\vec{f}, \vec{u}, v), \ldots, \lambda v.G_l(\vec{f}, \vec{u}, v)),$$

for all $\vec{f}, \vec{u}, v$.

1.2.8 Definition (limited recursion on notation)

Let F be a *type*-2 functional.
Then F is defined from G, H_0, H_1 and K by *limited recursion on notation*, if:

$$\begin{aligned} F(\vec{f},\vec{u},0) &= G(\vec{f},\vec{u}), && (1)\\ F(\vec{f},\vec{u},2v) &= H_0(\vec{f},\vec{u},v,F(\vec{f},\vec{u},v)), \quad v>0, && (2)\\ F(\vec{f},\vec{u},2v+1) &= H_1(\vec{f},\vec{u},v,F(\vec{f},\vec{u},v)), && (3)\\ |F(\vec{f},\vec{u},v)| &\leq |K(\vec{f},\vec{u},v)|, && (4) \end{aligned}$$

for all $\vec{f},\vec{u},v$.

For our further work we analogously as in the first subsection replace the schema 1.2.8 of *limited recursion on notation* by the following schema 1.2.9 of *limited recursion on notation**, where again equation (6) replaces the above equations (2) and (3), and condition (7) stands in place of condition (4). Condition (4) is equivalent to the condition $F(\vec{f},\vec{u},v) \leq K^*(\vec{f},\vec{u},v)$ for $K^*(\vec{f},\vec{u},v) = 1\sharp K(\vec{f},\vec{u},v) \dot{-} 1 = 2^{|K(\vec{f},\vec{u},v)|\dot{-}1}$.

1.2.9 Definition (limited recursion on notation*)

Let F be a *type*-2 functional.
Then F is defined from G, H, K by *limited recursion on notation**, if for all $\vec{f},\vec{u},v$:

$$\begin{aligned} F(\vec{f},\vec{u},0) &= G(\vec{f},\vec{u}), && (5)\\ F(\vec{f},\vec{u},v) &= H(\vec{f},\vec{u},v,F(\vec{f},\vec{u},\lfloor\frac{v}{2}\rfloor)), \quad v>0, && (6)\\ F(\vec{f},\vec{u},v) &\leq K(\vec{f},\vec{u},v). && (7) \end{aligned}$$

And again we are allowed to replace the schema 1.2.8 of *limited recursion on notation* by the new schema 1.2.9 of *limited recursion on notation** in the definition 1.2.3 of the BFF's.

The growth rate of a *Basic Feasible Functional* clearly cannot be majorised by a first order polynomial. Thus we can't have simple analogues for schemas involving polynomial bounds as for feasible functions. Also the machine models of feasibility in higher *types* differ basically from those of first order feasibility.

And that's why we need the following definitions introduced by Kapron and Cook[1996].

1.2.10 Definition (Norm Functional)

The functional of *type* $\mathbb{N}^{\mathbb{N}} \to \mathbb{N}^{\mathbb{N}}$, defined such that

$$f \mapsto |f|,$$

where

$$|f|(u) = max_{|v| \leq u} |f(v)|,$$

is called the *norm functional*.
The function $|f|$ is called the *norm* of the function f.

Kapron and Cook have shown that the functional of *type* $(1,1)$ such that $\langle f, u \rangle \mapsto |f|(|u|)$ is not basic feasible and that it neighter can be majorised by any *Basic Feasible Functional*.

They also introduced the class of *Second Order Polynomials* (see definition 1.2.11), which limit the growth rate of the *Basic Feasible Functionals*. But surprisingly, as we will see in the following definition 1.2.11, *Second Order Polynomials* themselves are not basic feasible. This fact is a consequence of the above mentioned result of Kapron and Cook.

1.2.11 Definition (Second Order Polynomials)

Let $x_0, x_1, \ldots$ and $f_0, f_1, \ldots$ be sets of first and second order variables respectively; then the set of *Second Order Polynomials* in $|f_0|, |f_1|, \ldots$ and $|x_0|, |x_1|, \ldots$ is defined inductively as the least set of terms of the language L_P containing constants c for each natural number $n \in \mathbb{N}$ and all terms $|x_0|, |x_1|, \ldots$ and which satifies the following closure condition: If P and Q are second order polynomials and f_i is a second order variable, then $P + Q$, $P \cdot Q$ and $|f_i|(P)$ are also second order polynomials.

Second order polynomials play quite the same role which first order polynomials play for feasible functions of *type* $\mathbb{N}^k \to \mathbb{N}$. But to repeat the major difference between them: While first order polynomials are feasibly computable functions themselves, *Second Order Polynomials* are not *Basic Feasible Functionals.*

We can now state the best possible second order analogues of the corresponding first order definition and theorem:

1.2.12 Definition (Polynomially Bounded Recursion On Notation with the Bound Q)

Let $Q(|f|, |\vec{u}|)$ be a second order polynomial and let $G(f, \vec{u})$ and $H(f, \vec{u}, z, v)$ be two *Basic Feasible Functionals.* Assume that the functional $F(f, \vec{u}, v)$ satisfies

$$F(f, \vec{u}, 0) = G(f, \vec{u}), \tag{1}$$

$$F(f, \vec{u}, v) = H(f, \vec{u}, v, F(f, \vec{u}, \lfloor \frac{v}{2} \rfloor)), \qquad v > 0, \tag{2}$$

$$|F(f, \vec{u}, v)| \leq Q(|f|, |\vec{u}|, |v|). \tag{3}$$

Then we say that F is defined from functionals G, H by *Polynomially Bounded Recursion On Notation with the Bound Q.*

1.2.13 Theorem (Ignjatovic and Sharma [2002])

Assume that the functional $F(f, \vec{u}, v)$ is defined from the functionals $G(f, \vec{u})$ and $H(f, \vec{u}, z, v)$ by *Polynomially Bounded Recursion On Notation with the Bound* $Q(|f|, |\vec{u}|, |v|)$. Then $F(f, \vec{u}, v)$ is a *Basic Feasible Functional.*

We need the following definition 1.2.14 for the second order theories Σ_1^b-$PIND$, $n - \Sigma_1^b$-$PIND$ and QF-$PIND$, which will be introduced later in the second section.

1.2.14 Definition (multiple limited recursion on notation)

Let F_i for $1 \leq i \leq n$ be *type*-2 functionals, and let G_i, H_i and K_i for $1 \leq i \leq n$ be *Basic Feasible Functionals.*
Then F_i are defined by *multiple limited recursion on notation* from G_i, H_i and K_i, if:

$$\begin{aligned}
F_i(\vec{f}, \vec{u}, 0) &= G_i(\vec{f}, \vec{u}), \\
F_i(\vec{f}, \vec{u}, v) &= H_i(\vec{f}, \vec{u}, v, F_1(\vec{f}, \vec{u}, \lfloor \frac{1}{2} v \rfloor), \ldots, F_n(\vec{f}, \vec{u}, \lfloor \frac{1}{2} v \rfloor)), \\
F_1(\vec{f}, \vec{u}, v) &\leq K_1(\vec{f}, \vec{u}, v), \\
F_i(\vec{f}, \vec{u}, v) &\leq K_i(\vec{f}, \vec{u}, v, F_1(\vec{f}, \vec{u}, u), \ldots, F_{i-1}(\vec{f}, \vec{u}, v)), \quad \text{for } 2 \leq i \leq n,
\end{aligned}$$

for all $\vec{f}, \vec{u}, v$.

As we see, *limited recursion on notation** (see definition 1.2.9) is a special case of the just introduced *multiple limited recursion on notation*, namely the case $n = 1$.

1.2.15 Theorem (Kapron and Cook [1996])

Basic Feasible Functionals are closed for *multiple limited recursion on notation.*

As in the first subsection of this section relating to feasible functions we analogously turn to the Turing machine model characterization of *Basic Feasible Functionals*. We use the usual model for computability with *Oracle Turing Machines OTM*.
Function inputs are presented using oracles corresponding to the input functions. Such oracles are queried using separate "write-only oracle input tapes" and "read-only oracle output tapes", while the machine is in the "oracle query state".
To query function input f at the value x, x is written in binary notation on the oracle input tape associated with f, and the corresponding oracle query state is entered. After entering the oracle state which corresponds to f, the value $f(x)$ appears on the oracle output tape associated with f, the oracle input tape is then erased and both the write head of the oracle input tape and the read head of the oracle output tape are placed at the corresponding initial cells of the tapes.
Thus, iterations of the form $f(f(\ldots f(x)\ldots))$ cannot be computed without the machine having to copy the intermediate results from the oracle output tape to the oracle input tape.
In general, there are two possible conventions for accounting for the running time of an oracle call:
In Mehlhorn's model (see 1.2.16), an oracle call has unit cost, while in the Kapron and Cook model (see 1.2.17), the oracle call described above has a cost of $|f(x)|$ time steps.
Mehlhorn [1976] and Kapron and Cook [1996] proved the following theorems 1.2.16 and 1.2.17:

1.2.16 Theorem (Mehlhorn [1976])

A *type*-2-functional $F(\vec{f},\vec{u})$, where $\vec{f} \in (\mathbb{N}^{\mathbb{N}})^k$ and $\vec{u} \in \mathbb{N}^l$ is basic feasible if and only if there exists an *Oracle Turing Machine* M with oracles for functions $\vec{f}$ and a *Basic Feasible Functional* $K(\vec{f},\vec{u})$ such that M computes $F(\vec{f},\vec{u})$ and the running time $T(\vec{f},\vec{u})$ with a unit cost for each oracle query, satisfies

$$\forall \vec{f} \forall \vec{x} \ (\ T(\vec{f},\vec{x}) \ \leq \ K(\vec{f},\vec{x})\ ).$$

1.2.17 Theorem (Kapron and Cook [1996])

A *type*-2-functional $F(\vec{f},\vec{u})$, where $\vec{f} \in (\mathbb{N}^{\mathbb{N}})^k$ and $\vec{u} \in \mathbb{N}^l$ is basic feasible if and only if there exists an *Oracle Turing Machines* M with oracles for functions $\vec{f}$ and a *second order polynomial* $P(|\vec{f}|,|\vec{u}|)$ such that M computes $F(\vec{f},\vec{u})$ and the running time $T(\vec{f},\vec{u})$ with $|f_i(z)|$ as the cost for an oracle query of $f_i \in \vec{f}$ at oracle input value z, satisfies

$$\forall \vec{f} \forall \vec{x} \ (\ T(\vec{f},\vec{x}) \ \leq \ P(|\vec{f}|,|\vec{x}|) \ ).$$

And Ignjatovic and Sharma combined the best features of the just introduced theorems 1.2.16 and 1.2.17:

1.2.18 Definition (Computable in Polynomial Time)

A functional $F(\vec{f},\vec{u})$ is *computable in polynomial time* if there exists an *Oracle Turing Machine* M with oracles for functions $\vec{f}$ and a second order polynomial $P(|\vec{f}|,|\vec{u}|)$ such that M computes $F(\vec{f},\vec{u})$ and for all $\vec{f},\vec{u}$ the running time $T(\vec{f},\vec{u})$ obtained by counting each oracle query as a single step regardless of the size of the oracle output, satisfies

$$T(\vec{f},\vec{u}) \quad \leq \quad P(|\vec{f}|,|\vec{u}|).$$

1.2.19 Theorem (Ignjatovic and Sharma [2002])

A functional $F(\vec{f},\vec{u})$ is a *Polynomial Time Computable Functional* if and only if it is a *Basic Feasible Functional*.

1.3 Logic

We have seen in the Abstract, that all our proofs are based on the application of essential tools of logic. That's why we devote to the field of logic in this subsection. The notation of the Hilbert-Calculus is assumed, but we give among other things a short introduction of the Tait-Calculus, which we also already know from the lecture "Logic and Computer Science".
The languages of the theories we are going to work with, will be introduced in the second section. Here we only give a short "reminder" of the axioms and rules of inference of the Tait-Calculus.

1.3.1 Tait-Calculus

In the Tait-Calculus, we define the idea of the provability for finite sets of formulas. We denote these sets by the letters Γ, Δ, where the comma stands for disjunction. Hence $\Gamma, \Delta, \alpha, \delta$ means $\Gamma \cup \Delta \cup \{\alpha\} \cup \{\delta\}$.

1. The **axioms** of TA:

   Every set of formulas, which can be written as

   $$\Gamma, \alpha, \neg\alpha,$$

   is an axiom of TA, where α is an arbitrary literal.

2. The **rules of inference** of TA are:

   (a) "Conjunction":

   $$\frac{\Gamma,\ \alpha_0 \qquad \Gamma,\ \alpha_1}{\Gamma,\ \alpha_0 \wedge \alpha_1} \qquad (\wedge),$$

(b) "Disjunction":

$$\frac{\Gamma,\ \alpha_i}{\Gamma,\ \alpha_0 \vee \alpha_1} \qquad (\vee),$$

where $i \in \{0, 1\}$. In addition the order is not relevant,

(c) "Universal Quantifier":

$$\frac{\Gamma,\ \alpha(u)}{\Gamma,\ \forall x \alpha(x)} \qquad (\forall),$$

where the free variable u must not be contained in $\Gamma,\ \forall x \alpha(x)$,

(d) "Existential Quantifier":

$$\frac{\Gamma,\ \alpha(t)}{\Gamma,\ \exists x \alpha(x)} \qquad (\exists),$$

where t is a term,

(e) "Cut":

$$\frac{\Gamma,\ \alpha \qquad \Gamma,\ \neg\alpha}{\Gamma} \qquad (\Diamond),$$

where the rank of the cut accords to the rank of its cut-formulas α resp. $\neg\alpha$.

We are going to see in the second section, how the rank of a formula is defined.

1.3.2 Preface

Closing this short logic-subsection 1.3, we turn to a notation which will be used frequently in the second and third section and which we should also already know from the lecture "Logic and Computer Science":

Let Th stand for an arbitrary theory and let Γ represent an arbitrary finite set of formulas. Then we say that:

1. $Th \vdash \Gamma$, if Γ can be derived from Th.
2. $Th \overset{n}{\vdash} \Gamma$, if there exists a derivation of Γ in the theory Th, whose length is at most n.
3. $Th \underset{r}{\vdash} \Gamma$, if the rank of every cut in the derivation of Γ in the theory Th is less than r.
4. $Th \overset{n}{\underset{r}{\vdash}} \Gamma$, if there is a derivation of Γ in the theory Th, whose depth is at most n and the rank of every cut in it is less than r.

1.4 Formulas

In this subsection 1.4 we inter alia explain the complexity of those formulas we will frequently need in our proofs. So if the complexity of a formula is not commented cohesively, we can look the definition up here.

To get a clearer representation later in our proofs, we also define the following equivalences concerning bounded formulas:

1. $(\forall x \leq t)\alpha \quad :\equiv \quad \forall x(x \leq t \rightarrow \alpha)$,
2. $(\exists x \leq t)\alpha \quad :\equiv \quad \exists x(x \leq t \wedge \alpha)$,
3. $(\forall x < |t|)\alpha \quad :\equiv \quad \forall x(x < |t| \rightarrow \alpha)$,
4. $(\exists x < |t|)\alpha \quad :\equiv \quad \exists x(x < |t| \wedge \alpha)$,
5. $s < t \quad :\equiv \quad s \leq t \wedge \neg(s = t)$.

The hierarchy of bounded formulas (e.g. $\sum_i^b$, $\prod_i^b$) of the second order arithmetic is obtained from the corresponding hierarchies of bounded formulas of the first order bounded arithmetic. That means by counting the alternations of bounded quantifiers ignoring sharply bounded ones.
After the above introduced equivalences we have:

1. $\Sigma_0^b = \Pi_0^b$ is the set of formulas all of whose quantifiers are sharply bounded.

2. Σ_{k+1}^b is defined inductively by:

   (a) $\Sigma_{k+1}^b \subseteq \Pi_k^b$,

   (b) If α is in Σ_{k+1}^b then so are $(\exists x \leq t)\alpha$ and $(\forall x \leq |t|)\alpha$,

   (c) If $\alpha, \beta \in \Sigma_{k+1}^b$ then $\alpha \wedge \beta$ and $\alpha \vee \beta$ are in Σ_{k+1}^b,

   (d) If $\alpha \in \Sigma_{k+1}^b$ and $\beta \in \Pi_{k+1}^b$ then $\neg\beta$ and $\beta \rightarrow \alpha$ are in Σ_{k+1}^b.

We allow the *Application Functional* $Ap(f, u) = f(u)$ to appear in the atomic formulas.

An open formula θ is a formula without any quantifiers.

A Σ_1^0-formula is a formula consisting of one existential quantifier followed by an open formula.

An $n - \Sigma_1^b$ formula is composed of at most n bounded existential quantifiers (no sharply bounded universal quantifiers), followed by an open formula θ.

1.5 Notation

If there isn't any other declaration, we use the following notation:

1. Constants: c, i, j, k, l, m, n, r,
2. Free variables: a, b, u, v, w,
3. Bounded variables: d, e, q, x, y, z,
4. *Type*-1 Functionals: f, g, h,
5. *Type*-2 Functionals: F, G, H, K, T,
6. Terms: s, t,
7. Formulas: Θ, Ψ, α, $\digamma$, δ, θ, ϕ, ψ,
8. Sets of formulas: Γ, Δ.

2 Fragments of Second Order Arithmetic

In this section we give an introduction of the logic tools we need to prove our main theorem in the third and last section:
In the first subsection 2.1 we introduce the second order theory $\mathbf{S}_2^1$ and the definition of "Σ_1^b-definable in $\mathbf{S}_2^1$" or "provably total in $\mathbf{S}_2^1$" respectively.
In the following subsections 2.2, 2.3 and 2.4 we introduce three second order theories $\Psi - PIND$, where Ψ stands for the complexity of the formula in the induction schema ($\Psi - PIND$). The first theory is $\Sigma_1^b - PIND$, the second one is $n - \Sigma_1^b - PIND$ and the third and last one is $QF - PIND$, where QF is a shortcut of *Quantor-Free*. So the formulas in the induction schema ($QF - PIND$) are open formulas.

2.1 The Second Order Formal Theory $\mathbf{S}_2^1$

The formal theories $\mathbf{S}_2^i$, $i \in \mathbb{N}$, are second order extensions of Buss's S_2^i, $i \in \mathbb{N}$. In this work we concentrate on the theory $\mathbf{S}_2^1$, $i = 1$, where one sort of variables u, v, w range over the set of *type*-0 functionals (constants) and the second sort of variables f, g, h range over the set of *type*-1 functionals (total functions from $\mathbb{N}$ to $\mathbb{N}$). Our aim is to characterize *type*-2 feasible functionals in the same way how Buss characterized polynomial time computable functions with the theory S_2^1 (see Buss[1986]).

2.1.1 The Language L_B^2 of $\mathbf{S}_2^1$

The theories $\mathbf{S}_2^i$ are formulated in the following language L_B^2:

$$L_B^2 \quad = \quad \{\leq,\ 0,\ 1,\ +,\ \cdot,\ |u|,\ \lfloor \frac{u}{2} \rfloor,\ \sharp,\ u \upharpoonright v,\ Ap(f,u)\}.$$

As we see, L_B^2 extends the language of Buss's theories S_2^i only by two symbols, firstly by $u \upharpoonright v$:

$$u \upharpoonright v = \lfloor \frac{u}{2^{|u|-v}} \rfloor,$$

and secondly by the *Application Functional* $Ap(f, u)$:

$$Ap(f, u) = f(u).$$

Besides, the two languages coincide exactly.

$u \upharpoonright v$ produces the number of the first more significant v bits of u in the binary representation of u, for example:

$$134 \upharpoonright 6 \equiv 100001, \text{ where } 134 \equiv 10000110.$$

2.1.2 The Axioms of $\mathbf{S}_2^1$

The set of axioms of the second order theories $\mathbf{S}_2^i$ firstly contains the $BASIC$ axioms, which build the set of axioms of Buss's first order theories S_2^i, and secondly some axioms for $u \upharpoonright v$.

Well then, the $BASIC$-Axioms are a finite set of 32 open axioms, defining simple properties of the function and relation symbols:

1. $v \leq u \quad \rightarrow \quad S_0(v) \leq S_0(u)$,
2. $u \neq S_0(u)$,
3. $0 \leq u$,
4. $(u \leq v \wedge u \neq v) \quad \leftrightarrow \quad (S_0(u) \leq v)$,
5. $u \neq 0 \quad \rightarrow \quad 2 \cdot u \neq 0$,
6. $u \leq v \vee v \leq u$,
7. $(u \leq v \wedge v \leq u) \quad \rightarrow \quad u = v$,
8. $(u \leq v \wedge v \leq w) \quad \rightarrow \quad u \leq w$,

9. $|0| = 0$,

10. $u \neq 0 \quad \rightarrow \quad (\ |2 \cdot u| = S_0(|u|) \quad \wedge \quad |S_0(2 \cdot u)| = S_0(|u|) \ )$,

11. $|S_0(0)| = S_0(0)$,

12. $u \leq v \quad \rightarrow \quad |u| \leq |v|$,

13. $|u \sharp v| = S_0(|u| \cdot |v|)$,

14. $0 \sharp u = S_0(0)$,

15. $u \neq 0 \quad \rightarrow \quad [1 \sharp (2 \cdot u) = 2 \cdot (1 \sharp u) \quad \wedge \quad 1 \sharp (S_0(2 \cdot u)) = 2 \cdot (1 \sharp u)]$,

16. $u \sharp v = v \sharp u$,

17. $|u| = |v| \quad \rightarrow \quad u \sharp w = v \sharp w$,

18. $|u| = |u| + |v| \quad \rightarrow \quad u \sharp v = (u \sharp v) \cdot (v \sharp v)$,

19. $u \leq u + v$,

20. $u \leq v \wedge u \neq v \quad \rightarrow \quad S_0(2 \cdot u) \leq 2 \cdot v \quad \wedge \quad S_0(2 \cdot u) \neq 2 \cdot v$,

21. $u + v = v + u$,

22. $u + 0 = u$,

23. $u + S_0(v) = S_0(u + v)$,

24. $(u + v) + w = u + (v + w)$,

25. $u + v \leq u + w \quad \leftrightarrow \quad v \leq w$,

26. $u \cdot 0 = 0$,

27. $u \cdot S_0(v) = (u \cdot v) + u$,

28. $u \cdot v = v \cdot u$,

29. $u \cdot (v + w) = (u \cdot v) + (u \cdot w)$,

30. $u \geq S_0(0) \quad \rightarrow \quad (u \cdot v \leq u \cdot w \quad \leftrightarrow \quad v \leq w)$,

31. $u \neq 0 \quad \rightarrow \quad (|u| = S_0(|\lfloor \frac{u}{2} \rfloor|)$,

32. $u = \lfloor \frac{v}{2} \rfloor \quad \leftrightarrow \quad (2 \cdot u = v \quad \vee \quad S_0(2 \cdot u) = v)$.

And finally, relating to $u \upharpoonright v$, we add the following four axioms:

1. $u \upharpoonright 0 = 0$,
2. $u \geq 1 \quad \rightarrow \quad u \upharpoonright 1 = 1$,
3. $v < |u| \quad \rightarrow \quad u \upharpoonright v = \lfloor \frac{u \upharpoonright (v+1)}{2} \rfloor$,
4. $v \geq |u| \quad \rightarrow \quad u \upharpoonright v = u$.

2.1.3 The Induction-Schema of $\mathbf{S}_2^1$

Theories $\mathbf{S}_2^i$ are finally obtained from the extended $BASIC$ by adding eihter one of the following two induction schemas for $\sum_i^b$ formulas:

$$(\Sigma_i^b\text{–}PIND) : (A(\vec{f}, \vec{u}, 0) \wedge (\forall x(A(\vec{f}, \vec{u}, \lfloor \tfrac{x}{2} \rfloor) \rightarrow A(\vec{f}, \vec{u}, x)))) \;\rightarrow\; \forall x A(\vec{f}, \vec{u}, x),$$

$$(\Sigma_i^b\text{–}LIND) : (A(\vec{f}, \vec{u}, 0) \wedge (\forall x(A(\vec{f}, \vec{u}, x) \rightarrow A(\vec{f}, \vec{u}, x{+}1)))) \;\rightarrow\; \forall x A(\vec{f}, \vec{u}, |x|).$$

We say “either one of”, because the two theories $(BASIC + \sum_i^b -PIND)$ and $(BASIC + \sum_i^b -LIND)$ are equivalent, so we have

$$(BASIC + \textstyle\sum_i^b\text{-}PIND) \quad \equiv \quad (BASIC + \textstyle\sum_i^b -LIND).$$

(For the proof see Buss[1986]. It goes equally easily by means of the set of axioms in 2.1.2 concerning $u \upharpoonright v$.)

To make reading more comfortable, from now on we set:

$(i) \quad \vec{\varphi} \quad = \quad \vec{f}, \vec{u}$,

$(ii) \quad \vec{\xi} \quad = \quad \vec{f}, \vec{x}$,

where $\vec{\varphi}$ represents any unbounded, $\vec{\xi}$ any bounded tupel of function and number variables.

Consequently we adjust the above introduced induction schemas on our new notation:

($\sum_i^b$-*PIND*): $\quad (A(\vec{\varphi},0) \ \wedge \ (\forall x(A(\vec{\varphi},\lfloor \frac{x}{2} \rfloor) \rightarrow A(\vec{\varphi},x)))) \ \rightarrow \ \forall x A(\vec{\varphi},x),$

($\sum_i^b$-*LIND*): $\quad (A(\vec{\varphi},0) \wedge (\forall x(A(\vec{\varphi},x) \rightarrow A(\vec{\varphi},x{+}1)))) \ \rightarrow \ \forall x A(\vec{\varphi},|x|).$

In the end of this subsection we turn to the following definition 2.1.4 of the Σ_1^b-definability of a functional F in the theory $\mathbf{S}_2^1$. As alluded in the beginning of this section we also use the term "provably total in $\mathbf{S}_2^1$" as a supposably rather known synonyme of "Σ_1^b-definable in $\mathbf{S}_2^1$".
The definition 2.1.4 is the second order extension of the corresponding first order definition in Buss[1986]:

2.1.4 Definition (Σ_1^b-definability in $\mathbf{S}_2^1$)

A functional F is Σ_1^b-definable in the theory $\mathbf{S}_2^1$, if there exists a Σ_1^b formula $\Psi_F(\vec{\varphi}, v)$, such that

$$\mathbf{S}_2^1 \quad \vdash \quad \forall \vec{\xi} \ \exists! y \ \Psi_F(\vec{\xi}, y),$$

where

$$\langle \mathbb{N}^k, (\mathbb{N}^{\mathbb{N}})^m \rangle \quad \models \quad \forall \vec{\xi} \ \Psi_F(\vec{\xi}, F(\vec{\xi})).$$

2.2 The second order theory $\Sigma_1^b - PIND$

The second order theory $\Sigma_1^b - PIND$ is obtained from the just introduced theory $\mathbf{S}_2^1$ by expanding the language L_B^2 of $\mathbf{S}_2^1$ with new functional symbols and by adding recursion equations to the set of axioms of $\mathbf{S}_2^1$, so that the resulting set is closed under *multiple limited recursion on notation* (see definition 1.2.14). Moreover the theory $\Sigma_1^b - PIND$ expands $\mathbf{S}_2^1$ by allowing functional symbols to appear in the induction schema.

2.2.1 The Language B of $\Sigma_1^b - PIND$

The language **B** of the theory $\Sigma_1^b - PIND$ expands the language L_B^2 of $\mathbf{S}_2^1$ with new functional symbols.

Hence **B** firstly contains the in the meantime well-known symbols:

$$\leq,\ 0,\ 1,\ +,\ \cdot,\ |u|,\ \lfloor \tfrac{u}{2} \rfloor,\ \sharp,\ u \restriction v,$$

which come from the language L_B^2 of $\mathbf{S}_2^1$, and secondly the following inductively defined functional symbols:

1. Every functional symbol of L_B^2 is a functional symbol of **B**.
2. O is a functional symbol of *rank* $(0, 1)$ of **B**.
3. S_0 is a functional symbol of *rank* $(0, 1)$ of **B**.
4. S_1 is a functional symbol of *rank* $(0, 1)$ of **B**.
5. Sg is a functional symbol of *rank* $(0, 1)$ of **B**.
6. $Nadi$ is a functional symbol of *rank* $(0, 2)$ of **B**.
7. $Smash$ is a functional symbol of *rank* $(0, 2)$ of **B**.
8. Min is a functional symbol of *rank* $(0, 2)$ of **B**.
9. Max is a functional symbol of *rank* $(0, 2)$ of **B**.
10. Pr_i^n is a functional symbol of *rank* $(0, n)$ of **B**.
11. Ap is a functional symbol of *rank* $(1, 0)$ of **B**.
12. If G is a functional symbol of *rank* (k, l) of **B**, then so is the functional symbol $Exp^{m,n}(G)$ of *rank* $(k + m, l + n)$.
13. If H is a functional symbol of *rank* $(k, l + n)$ of **B**, and $G_1, \ldots, G_l$ are functional symbols each of *rank* (k, l) of **B**, then so is the functional symbol $Comp(H, G_1, \ldots, G_l)$ of *rank* (k, l).
14. If, for $1 \leq i \leq n$, G_i are functional symbols of *rank* (k, l) of **B**, and H_i are functional symbols of *rank* $(k, l+n+1)$ of **B**, and K_i are functional symbols of *rank* $(k, l + i)$ of **B**, then so is
$Rec_i(G_1, \ldots, G_n, H_1, \ldots, H_n, K_1, \ldots, K_n)$ of *rank* $(k, l + 1)$.

Accessory to 12.:
To keep a clear notation, we write Exp in place of $Exp^{m,n}$.

2.2.2 The Axioms of $\Sigma_1^b - PIND$

First of all there are the $BASIC$-Axioms (see 2.1.2), secondly the following $Equality$-Axioms:

(a) $s = s$,

(b) $s = t \quad \rightarrow \quad (\alpha(s) \rightarrow \alpha(t))$,

and third of all the $BASIC_2$-Axioms, which we define as follows for all $Basic$ $Feasible$ $Functionals$:

1. $O(u) = 0$,
2. $S_0(u) = 2u$,
3. $S_1(u) = 2u + 1$,
4. $Sg(u) = u \dot{-} (u \dot{-} 1)$
5. $Nadi(u, v) = u \dot{-} v$,
6. $S_1(u, v) = u\sharp y = 2^{|u|\cdot|v|}$,
7. $Min(u, v) = Sg(u \dot{-} v) \cdot v + Sg(v \dot{-} u) \cdot u$
8. $Max(u, v) = Sg(u \dot{-} v) \cdot u + Sg(v \dot{-} u) \cdot v$
9. $Pr_i^n(u_1, \ldots, u_n) = u_i$,
10. $Ap(f, u) = f(u)$,
11. $Exp(G)(\vec{f}, \vec{g}, \vec{u}, \vec{y}) = g(\vec{\varphi})$,
12. $Comp(H, G_1, \ldots, G_l)(\vec{\vec{\varphi}}) = H(\vec{\varphi}, G_1(\vec{\varphi}), \ldots, G_l(\vec{\varphi}))$,
13. $Rec_i(\vec{G}, \vec{H}, \vec{K})(\vec{\varphi}, 0) = G_i(\vec{\varphi})$,
    $(u \geq 1) \rightarrow Rec_i(\vec{G}, \vec{H}, \vec{K})(\vec{\varphi}, u) = Min(s, t)$, where
    $s \equiv H_i(\vec{\varphi}, u, Rec_1(\vec{G}, \vec{H}, \vec{K})(\vec{\varphi}, \lfloor \frac{u}{2} \rfloor), \ldots, Rec_n(\vec{G}, \vec{H}, \vec{K})(\vec{\varphi}, \lfloor \frac{u}{2} \rfloor))$,
    $t \equiv K_i(\vec{\varphi}, u, Rec_1(\vec{G}, \vec{H}, \vec{K})(\vec{\varphi}, u), \ldots, Rec_{i-1}(\vec{G}, \vec{H}, \vec{K})(\vec{\varphi}, u))$,
    where $\vec{G} = G_1, \ldots, G_n$, $\vec{H} = H_1, \ldots, H_n$ and $\vec{K} = K_1, \ldots, K_n$.

Because of the fact that all polytime functions belong to the *Basic Feasible Functionals* (see theorem 1.2.4), it is obvious that the *type*-1-functional *Nadi* (natural difference) also belongs to the *BFF*'s.

2.2.3 The Induction-Schema of $\Sigma_1^b - PIND$

Finally, we receive the theory $\Sigma_1^b - PIND$ by adding the following induction schema to the above system of axioms. The induction schema ($\sum_1^b - PIND$) newly allows functionals to appear in it:

$$(\textstyle\sum_1^b - PIND): \quad (F(\vec{\varphi}, 0)) \ \wedge \ \forall x(F(\vec{\varphi}), \lfloor \tfrac{x}{2} \rfloor) \rightarrow F(\vec{\varphi}), x)) \ \rightarrow \ \forall x F(\vec{\varphi}, x).$$

Now we are going to introduce the two second order theories $n - \Sigma_1^b - PIND$ and $QF - PIND$. In opposition to the previous theories $\mathbf{S}_2^1$ and $\Sigma_1^b - PIND$, these two won't be written in the *Hilbert-Calculus*, but in the *Tait-Calculus* (see 1.3.1). The new notation concerns in each case the axioms and the induction schema, whereas the latter now rather becomes an induction rule than an axiom.

2.3 The second order theory $n - \Sigma_1^b - PIND$

As the name of this theory $n - \Sigma_1^b - PIND$ betrays, it only differs in the complexity of the formula in the induction rule from the above introduced theory $\Sigma_1^b - PIND$.

2.3.1 The Language B of $n - \Sigma_1^b - PIND$

(See 2.2.1).

2.3.2 The Axioms of $n - \Sigma_1^b - PIND$

Firstly there are the *Equality*-Axioms:

(a) $\Gamma,\, s = s,$

(b) $\Gamma,\, s \neq t,\, \neg\alpha(s),\, \alpha(t),$

and secondly the $BASIC$- respective $BASIC_2$-Axioms:

$$\Delta,\ \alpha,$$

where Δ is a set of formulas and α is one of the $BASIC$- or one of the $BASIC_2$-Axioms.

2.3.3 The Induction-Rule of $n - \Sigma_1^b - PIND$

$$(n{-}\Sigma_1^b{-}PIND) : \qquad \frac{\Gamma(\vec{\varphi}),\ \psi(\vec{\varphi},0) \qquad \Gamma(\vec{\varphi}),\ \neg\psi(\vec{\varphi},\lfloor \frac{v}{2} \rfloor),\ \psi(\vec{\varphi},v))}{\Gamma(\vec{\varphi}),\ \psi(\vec{\varphi},t(\vec{\varphi}))},$$

where ψ is an $n - \Sigma_1^b$-formula and the free variable v must not be contained in $\Gamma(\vec{\varphi}),\ \psi(\vec{\varphi},t(\vec{\varphi}))$.

Before introducing the cut elimination theorem for $n - \Sigma_1^b - PIND$ in 2.3.6, we firstly

turn to the definition of the rank $\mathbf{rn}_{n-\Sigma_1^b-PIND}(\alpha)$ of a **B**-formula α in 2.3.4 and

secondly to the reduction Lemma for $n - \Sigma_1^b - PIND$ in 2.3.5.

2.3.4 The rank $\mathbf{rn}_{n-\Sigma_1^b-PIND}(\alpha)$ of a B-formula α

We define the rank $\mathbf{rn}_{n-\Sigma_1^b-PIND}(\alpha)$ of a **B**-formula α inductively as follows:

1. $\mathbf{rn}_{n-\Sigma_1^b-PIND}(\alpha) = 0$,

   if α has at most n bounded existential quantifiers and no sharply bounded

   universal quantifiers, else

2. $\mathbf{rn}_{n-\Sigma_1^b-PIND}(\alpha \wedge \delta) =$

   $\mathbf{rn}_{n-\Sigma_1^b-PIND}(\alpha \vee \delta) =$

   $sup(\mathbf{rn}_{n-\Sigma_1^b-PIND}(\alpha), \mathbf{rn}_{n-\Sigma_1^b-PIND}(\delta)) + 1$.

3. $\mathbf{rn}_{n-\Sigma_1^b-PIND}(\exists x \alpha(x)) =$

   $\mathbf{rn}_{n-\Sigma_1^b-PIND}(\forall x \alpha(x)) =$

   $\mathbf{rn}_{n-\Sigma_1^b-PIND}(\alpha(u)) + 1$.

2.3.5 The reduction Lemma for $n - \Sigma_1^b - PIND$

For all natural numbers m, n and r, for all formulas α with $\mathbf{rn}_{n-\Sigma_1^b-PIND}(\alpha) \leq r$, we have:

$$n - \Sigma_1^b - PIND \mid\frac{m}{r}\ \Gamma, \alpha \quad \text{and} \quad n - \Sigma_1^b - PIND \mid\frac{n}{r}\ \Delta, \neg\alpha$$

$$\Longrightarrow$$

$$n - \Sigma_1^b - PIND \mid\frac{m+n}{r}\ \Gamma, \Delta$$

So, if we receive a set of formulas after having applied the cut-rule in an $n - \Sigma_1^b - PIND$ derivation d, it is possible to get the same set of formulas without applying the cut-rule in another $n - \Sigma_1^b - PIND$ derivation d', whose depth is greater than the depth of d.

So, now we are prepared to read the cut elimination theorem 2.3.6 for $n - \Sigma_1^b - PIND$:

2.3.6 The cut elimination theorem for $n - \Sigma_1^b - PIND$

For all natural numbers n and r we have:

$$n - \Sigma_1^b - PIND \vdash_r^n \Gamma \quad \Longrightarrow \quad n - \Sigma_1^b - PIND \vdash_1^{2_r(n)} \Gamma$$

as a consequence of the fact that:

$$n - \Sigma_1^b - PIND \vdash_{r+1}^n \Gamma \quad \Longrightarrow \quad n - \Sigma_1^b - PIND \vdash_r^{2^n} \Gamma,$$

where $2_r(n)$ is defined inductively as follows:

$$2_0(n) := n,$$

$$2_{r+1}(n) := 2^{2_r(n)}.$$

As we see, the just introduced "cut elimination" is effectively a "partial cut elimination":
We only eliminate those cuts, where the rank of the cut formula is greater than zero. Hence we don't apply the cut elimination to induction formulas, which means to $n - \Sigma_1^b$ formulas.
Therefore the proof of this theorem goes equal to the proof of the corresponding result which we know from the lecture "Logic and Computer Science".

2.4 The second order theory $QF - PIND$

The theory $QF - PIND$ again only differs in the complexity of the formula in the induction rule from the theory $\Sigma_1^b - PIND$.

2.4.1 The Language B of $QF - PIND$

(See 2.2.1).

2.4.2 The Axioms of $QF - PIND$

(See 2.3.2).

2.4.3 The Induction-Rule of $QF - PIND$

$$(QF{-}PIND): \qquad \frac{\Gamma(\vec{\varphi}),\ \theta(\vec{\varphi},0) \qquad \Gamma(\vec{\varphi}),\ \neg\theta(\vec{\varphi},\lfloor \frac{v}{2} \rfloor),\ \theta(\vec{\varphi}),v)}{\Gamma(\vec{\varphi}),\ \theta(\vec{\varphi},t(\vec{\varphi}))},$$

where θ is a QF-formula, respectively an open formula and the free variable v must not be contained in $\Gamma(\vec{\varphi}),\ \theta(\vec{\varphi},t(\vec{\varphi}))$.

The following definition 2.4.4 of the rank $\mathbf{rn}_{QF-PIND}(\alpha)$ of a **B**-formula α, the reduction Lemma for $QF - PIND$ in 2.4.5 and the cut elimination theorem for $QF - PIND$ in 2.4.6 correspond to the analogue definition, Lemma and theorem before in 2.3.4, 2.3.5 and 2.3.6 relating to the theory $n - \Sigma_1^b - PIND$:

2.4.4 The rank $\mathbf{rn}_{QF-PIND}(\alpha)$ of a B-formula α

We define the rank $\mathbf{rn}_{QF-PIND}(\alpha)$ of a **B**-formula α inductively as follows:

1. $\mathbf{rn}_{QF-PIND}(\alpha) = 0$,
   if α is an open formula, else:

2. $\mathbf{rn}_{QF-PIND}(\alpha \wedge \delta) =$
   $\mathbf{rn}_{QF-PIND}(\alpha \vee \delta) =$
   $sup(\mathbf{rn}_{QF-PIND}(\alpha), \mathbf{rn}_{QF-PIND}(\delta)) + 1.$

3. $\mathbf{rn}_{QF-PIND}(\exists x \alpha(x)) =$
   $\mathbf{rn}_{QF-PIND}(\forall x \alpha(x)) =$
   $\mathbf{rn}_{QF-PIND}(\alpha(u)) + 1.$

2.4.5 The reduction Lemma for $QF-PIND$

For all natural numbers m, n and r, for all formulas α with $\mathbf{rn}_{QF-PIND}(\alpha) \leq r$, we have:

$$QF-PIND \mid\frac{m}{r}\; \Gamma, \alpha \quad \text{and} \quad QF-PIND \mid\frac{n}{r}\; \Delta, \neg\alpha$$

$$\Longrightarrow$$

$$QF-PIND \mid\frac{m+n}{r}\; \Gamma, \Delta$$

2.4.6 The cut elimination theorem for $QF-PIND$

For all natural numbers n and r we have:

$$QF-PIND \mid\frac{n}{r}\; \Gamma \quad \Longrightarrow \quad QF-PIND \mid\frac{2_r(n)}{1}\; \Gamma$$

as a consequence of the fact that:

$$QF - PIND \mathop{\vdash}\limits^{n}_{r+1} \Gamma \quad \Longrightarrow \quad QF - PIND \mathop{\vdash}\limits^{2^n}_{r} \Gamma,$$

where $2_r(n)$ is defined inductively as follows:

$$2_0(n) := n,$$

$$2_{r+1}(n) := 2^{2_r(n)}.$$

In the next Lemma 2.4.7 we show that for any open formula θ there exists a characteristical functional F_θ in $\mathbf{B}$ which builds the analogue of the characteristical function we know from the lecture "Introduction into Theoretical Computer Science":

2.4.7 Lemma

Let θ be any open formula of $rank$ (k, l).

Then there exists a functional $F_\theta(\vec{\varphi}) \in \mathbf{B}$, so that

$$(a) \quad QF - PIND \quad \vdash \quad \forall \vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \vee \ F_\theta(\vec{\varphi}) = 1)$$

and

$$(b) \quad QF - PIND \quad \vdash \quad \forall \vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \leftrightarrow \ \theta(\vec{\varphi})).$$

Proof:

Case 1:

θ is of the form $s = t$, where s and t are terms.
We define $F_\theta(\vec{\varphi})$ for all $\vec{\varphi}$ as:

$$F_\theta(\vec{\varphi}) = Sg(s(\vec{\varphi}) \dot{-} t(\vec{\varphi}) + t(\vec{\varphi}) \dot{-} s(\vec{\varphi})),$$

which directly implies that F_θ is an element of $\mathbf{B}$ (see 2.3.1).

(a):

The formula

$$\forall \vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \vee \ F_\theta(\vec{\varphi}) = 1)$$

is valid. Hence

$$QF - PIND \quad \vdash \quad \forall \vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \vee \ F_\theta(\vec{\varphi}) = 1).$$

(b):

The following equivalences can be derived in $QF - PIND$ (we renounce the proof of this fact):

$$\theta(\vec{\varphi})$$

$$\leftrightarrow$$

$$s(\vec{\varphi}) = t(\vec{\varphi})$$

$$\leftrightarrow$$

$$s(\vec{\varphi}) \dot{-} t(\vec{\varphi}) + t(\vec{\varphi}) \dot{-} s(\vec{\varphi}) = 0$$

$$\leftrightarrow$$

$$Sg(s(\vec{\varphi}) \dot{-} t(\vec{\varphi}) + t(\vec{\varphi}) \dot{-} s(\vec{\varphi})) = 0$$

$$\leftrightarrow$$

$$F_\theta(\vec{\varphi}) = 0.$$

Hence

$$QF - PIND \quad \vdash \quad \forall \vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \leftrightarrow \ \theta(\vec{\varphi})).$$

Case 2:

θ is of the form $\theta_0 \wedge \theta_1$.
We define $F_\theta(\vec{\varphi})$ for all $\vec{\varphi}$ as:

$$F_\theta(\vec{\varphi}) = Max(F_{\theta_0}(\vec{\varphi}), F_{\theta_1}(\vec{\varphi})),$$

which directly implies that F_θ is an element of **B**.

(a):

Now

$$\forall\vec{\varphi}(Max(F_{\theta_0}(\vec{\varphi}), F_{\theta_1}(\vec{\varphi})) = 0 \ \vee \ Max(F_{\theta_0}(\vec{\varphi}), F_{\theta_1}(\vec{\varphi})) = 1),$$

hence

$$QF-PIND \quad \vdash \quad \forall\vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \vee \ F_\theta(\vec{\varphi}) = 1).$$

(b):

The following equivalences can again be derived in $QF-PIND$:

$$\theta(\vec{\varphi})$$

$$\leftrightarrow$$

$$\theta_0(\vec{\varphi}) \ \wedge \ \theta_1(\vec{\varphi})$$

$$\leftrightarrow$$

$$F_{\theta_0}(\vec{\varphi}) = 0 \ \wedge \ F_{\theta_1}(\vec{\varphi}) = 0$$

$$\leftrightarrow$$

$$Max(F_{\theta_0}(\vec{\varphi}), F_{\theta_1}(\vec{\varphi})) = 0$$

$$\leftrightarrow$$

$$F_\theta(\vec{\varphi}) = 0.$$

Hence

$$QF-PIND \quad \vdash \quad \forall\vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \leftrightarrow \ \theta(\vec{\varphi})).$$

Case 3:

θ is of the form $\theta_0 \vee \theta_1$.
We define $F_\theta(\vec{\varphi})$ for all $\vec{\varphi}$ as:

$$F_\theta(\vec{\varphi}) = Min(F_{\theta_0}(\vec{\varphi}), F_{\theta_1}(\vec{\varphi})),$$

which directly implies that F_θ is an element of **B**.

(a):

Now

$$\forall\vec{\varphi}(Min(F_{\theta_0}(\vec{\varphi}), F_{\theta_1}(\vec{\varphi})) = 0 \ \vee \ Min(F_{\theta_0}(\vec{\varphi}), F_{\theta_1}(\vec{\varphi})) = 1),$$

hence

$$QF - PIND \quad \vdash \quad \forall\vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \vee \ F_\theta(\vec{\varphi}) = 1).$$

(b):

And again the following equivalences can be derived in $QF - PIND$:

$$\theta(\vec{\varphi})$$

$$\leftrightarrow$$

$$\theta_0(\vec{\varphi}) \ \vee \ \theta_1(\vec{\varphi})$$

$$\leftrightarrow$$

$$F_{\theta_0}(\vec{\varphi}) = 0 \ \vee \ F_{\theta_1}(\vec{\varphi}) = 0$$

$$\leftrightarrow$$

$$Min(F_{\theta_0}(\vec{\varphi}), F_{\theta_1}(\vec{\varphi})) = 0$$

$$\leftrightarrow$$

$$F_\theta(\vec{\varphi}) = 0.$$

Hence

$$QF - PIND \quad \vdash \quad \forall\vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \leftrightarrow \ \theta(\vec{\varphi})).$$

Case 4:

θ is of the form $\neg\theta_0$.
We define $F_\theta(\vec{\varphi})$ for all $\vec{\varphi}$ as:

$$F_\theta(\vec{\varphi}) = 1 \dot{-} F_{\theta_0}(\vec{\varphi}),$$

which directly implies that F_θ is an element of $\mathbf{B}$.

(a):

Now

$$\forall\vec{\varphi}(1 \dot{-} F_{\theta_0}(\vec{\varphi}) = 0 \ \vee \ 1 \dot{-} F_{\theta_0}(\vec{\varphi}) = 1),$$

hence

$$QF-PIND \quad \vdash \quad \forall\vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \vee \ F_\theta(\vec{\varphi}) = 1).$$

(b):

The following equivalences can again be derived in $QF-PIND$:

$$\theta(\vec{\varphi})$$

$$\leftrightarrow$$

$$\neg\theta_0(\vec{\varphi})$$

$$\leftrightarrow$$

$$F_{\theta_0}(\vec{\varphi}) = 1$$

$$\leftrightarrow$$

$$1 \dot{-} F_{\theta_0}(\vec{\varphi}) = 0$$

$$\leftrightarrow$$

$$F_\theta(\vec{\varphi}) = 0.$$

Hence

$$QF-PIND \quad \vdash \quad \forall\vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \leftrightarrow \ \theta(\vec{\varphi})).$$

♣

The following theorem 2.4.8 corresponds to the theorem "Definition by Cases" from the lecture "Introduction into Theoretical Computer Science":

2.4.8 Theorem

Let θ be an open formula, for which

$$QF-PIND \quad \vdash \quad \theta(t_1(\vec{\varphi})), \theta(t_2(\vec{\varphi})),$$

where $t_1(\vec{\varphi})$, $t_2(\vec{\varphi})$ are terms.
Then there exists a functional $G(\vec{\varphi}) \in \mathbf{B}$, so that

$$QF-PIND \quad \vdash \quad \theta(G(\vec{\varphi})).$$

Proof:

After Lemma 2.4.7 there exists a functional $F_\theta \in \mathbf{B}$, so that for all $\vec{\varphi}$:

$$QF-PIND \quad \vdash \quad \forall\vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \vee \ F_\theta(\vec{\varphi}) = 1)$$

and

$$QF-PIND \quad \vdash \quad \forall\vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \leftrightarrow \ \theta(\vec{\varphi})).$$

Hence we can define $G(\vec{\varphi})$ as follows:

$$G(\vec{\varphi}) \ = \ (1 \dot{-} F_\theta(\vec{\varphi})) \cdot t_1(\vec{\varphi}) \ + \ F(\vec{\varphi}) \cdot t_2(\vec{\varphi}),$$

which directly implies that $G(\vec{\varphi})$ is an element of $\mathbf{B}$.

Hence the fact that

$$QF-PIND \quad \vdash \quad \theta(t_1(\vec{\varphi})), \theta(t_2(\vec{\varphi})),$$

directly implies that

$$QF-PIND \quad \vdash \quad \theta(G(\vec{\varphi})).$$

♣

2.4.9 Theorem

Let θ be an open formula. Then there exists a functional $H \in \mathbf{B}$, so that

$$QF-PIND \quad \vdash \quad (\exists y < |x|)\theta(y) \quad \leftrightarrow \quad \theta(H(x)).$$

Proof:

We define the functional H as follows:

$$H(u) \quad = \quad min_{v<|u|}(\theta(v)).$$

After Lemma 2.4.7 there exists a functional $F_\theta \in \mathbf{B}$, so that:

$$QF-PIND \quad \vdash \quad \forall\vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \vee \ F_\theta(\vec{\varphi}) = 1)$$

and

$$QF-PIND \quad \vdash \quad \forall\vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \ \leftrightarrow \ \theta(\vec{\varphi})).$$

Hence we can define the functional H as an element of the language $\mathbf{B}$ as follows:

$$H(0) = 0,$$

$$H(u) = (1 \dot{-} Sg(H(\lfloor \tfrac{u}{2} \rfloor) \dot{-} |\lfloor \tfrac{u}{2} \rfloor|)) \cdot ((1 \dot{-} F_\theta(|\lfloor \tfrac{u}{2} \rfloor|)) \cdot |\lfloor \tfrac{u}{2} \rfloor| + F_\theta(|\lfloor \tfrac{u}{2} \rfloor|) \cdot |u|),$$

$$H(u) \leq S_0(u) \quad \forall u.$$

We recognize at first sight that H only contains symbols of the language $\mathbf{B}$ and is defined by *limited recursion on notation*, which directly implies that $H \in \mathbf{B}$.
But because of the fact that it is not easy to deduce what the functional H does step by step with its argument u to finally deliver, if it exists, the minimal natural number $v < |u|$, for which $\theta(v)$ holds, we explain it via the corresponding commented algorithm:

Case 1: $u = 0$,

$H(u) = H(0) = 0$. Trivial.

Case 2: $u > 0$,

$H(u) =$

if $H(\lfloor \frac{u}{2} \rfloor) = |\lfloor \frac{u}{2} \rfloor|$, then
  if $F_\theta(|\lfloor \frac{u}{2} \rfloor|) = 0$, then
    $|\lfloor \frac{u}{2} \rfloor|$,
  else $|u|$,
else $H(\lfloor \frac{u}{2} \rfloor)$.

We firstly check for the argument $u > 0$, if $H(\lfloor \frac{u}{2} \rfloor) = |\lfloor \frac{u}{2} \rfloor|$ to see if we have already found the minimal number $v < |u|$ until $|\lfloor \frac{u}{2} \rfloor|$, $|u| - 1$ respectively. If yes, we also check if $F_\theta(|\lfloor \frac{u}{2} \rfloor|)$ is equal to zero, so that both directions "$\rightarrow$" and "$\leftarrow$" of our formula are fulfilled.
If $\neg(H(\lfloor \frac{u}{2} \rfloor) = |\lfloor \frac{u}{2} \rfloor|)$, we repeat the whole thing and check if $H(\lfloor \frac{u}{4} \rfloor) = |\lfloor \frac{u}{4} \rfloor|$, and so on...
In the case that we don't find any number $v < |u|$, so that $F_\theta(v) = 0$, we return $|u|$.

We suppose that our formula is valid in $QF - PIND$ after the upper definition of the functional H and abdicate the detailed inductive proof of:

$$QF - PIND \quad \vdash \quad \forall \vec{\varphi}(F_\theta(\vec{\varphi}) = 0 \; \leftrightarrow \; \theta(\vec{\varphi})).$$

♣

The result, which is formulated in the next theorem 2.4.10, is relevant for the proof of Lemma 3.2 in the third section.

2.4.10 Theorem (The $\exists$-Inversion for QF-PIND)

Let Γ be a set of open formulas or Σ_1^0-formulas.

Then the following implication holds:

$$QF - PIND \ \vert\frac{n}{1} \ \Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi}, z)$$

$$\Longrightarrow$$

There exists a term $t(\vec{\varphi}) \in \mathbf{B}$, so that

$$QF - PIND \ \vdash \ \Gamma(\vec{\varphi}), \theta(\vec{\varphi}, t(\vec{\varphi})).$$

Proof:

We prove by induction on the length of the proof.

Case 1:

If the derivation ends with an axiom, then the "length" of the derivation is equal to zero.
Hence the formula $\exists z\theta(\vec{\varphi}, z)$ can't be the main formula, but is contained in the set of side-formulas $\Gamma(\vec{\varphi})$. We deduce that it is an axiom itself. So it is sure, that we can find any term t, for which it is valid in $QF - PIND$.

Case 2:

The derivation ends with an $(\wedge)$-conclusion.
So $\Gamma(\vec{\varphi})$ is of the form $\Gamma(\vec{\varphi}), \theta_0 \wedge \theta_1$, where $\theta_0 \wedge \theta_1$ can go into $\Gamma(\vec{\varphi})$.
In this second case the last conclusion looks like:

$$\frac{\Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi}, z),\ \theta_0 \qquad \Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi}, z),\ \theta_1}{\Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi}, z),\ \theta_0 \wedge \theta_1} \quad (\wedge).$$

Under the induction hypothesis there exist terms $t_0(\vec{\varphi})$, $t_1(\vec{\varphi}) \in \mathbf{B}$, so that:

$$QF-PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_0(\vec{\varphi})),\ \theta_0,$$

and

$$QF-PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_1(\vec{\varphi})),\ \theta_1.$$

After an $(\vee)$-conclusion in the above case with $\theta(\vec{\varphi}, t_1(\vec{\varphi}))$ and in the lower case with $\theta(\vec{\varphi}, t_0(\vec{\varphi}))$, we receive:

$$QF-PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_0(\vec{\varphi})),\ \theta(\vec{\varphi}, t_1(\vec{\varphi})),\ \theta_0,$$

and

$$QF-PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_0(\vec{\varphi})),\ \theta(\vec{\varphi}, t_1(\vec{\varphi})),\ \theta_1.$$

Now we apply the $(\wedge)$-conclusion on these two received sets of formulas, so:

$$QF-PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_0(\vec{\varphi})),\ \theta(\vec{\varphi}, t_1(\vec{\varphi})),\ \theta_0 \wedge \theta_1,$$

which accords to

$$QF-PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_0(\vec{\varphi})),\ \theta(\vec{\varphi}, t_1(\vec{\varphi})).$$

After Lemma 2.4.8 there exists a functional $F \in \mathbf{B}$, so that

$$F(\vec{\varphi}) \quad = \quad \begin{cases} t_0(\vec{\varphi}), & \text{if } \theta(\vec{\varphi}, t_0(\vec{\varphi})), \\ t_1(\vec{\varphi}), & \text{else.} \end{cases}$$

Hence $F(\vec{\varphi})$ accords to the term $t(\vec{\varphi})$ we have been looking for:

$$QF - PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t(\vec{\varphi})).$$

Case 3:

The derivation ends with an $(\vee)$-conclusion.
So $\Gamma(\vec{\varphi})$ is of the form $\Gamma(\vec{\varphi}), \theta_0 \vee \theta_1$, where $\theta_0 \vee \theta_1$ can go into $\Gamma(\vec{\varphi})$.
In this third case the last conclusion looks like:

$$\frac{\Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi}, z),\ \theta_0}{\Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi}, z),\ \theta_0 \vee \theta_1} \quad (\vee).$$

Under the induction hypothesis there exists a term $t_0(\vec{\varphi}) \in \mathbf{B}$, so that:

$$QF - PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_0(\vec{\varphi})),\ \theta_0.$$

Now we apply an $(\vee)$-conclusion to the just received formula:

$$QF - PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_0(\vec{\varphi})),\ \theta_0 \vee \theta_1,$$

which accords to

$$QF - PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_0(\vec{\varphi})),$$

where $t_0(\vec{\varphi})$ is our searched term $t(\vec{\varphi})$.

Case 4:

The derivation ends with an $(\exists)$-conclusion.

Case 4.1.:

The last conclusion looks like:

$$\frac{\Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi}, z),\ \theta(\vec{\varphi}, t_0(\vec{\varphi}))}{\Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi}, z)} \quad (\exists).$$

Under the induction hypothesis there exists a term $t_1(\vec{\varphi}) \in \mathbf{B}$, so that:

$$QF{-}PIND \ \vdash \ \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_1(\vec{\varphi})),\ \theta(\vec{\varphi}, t_0(\vec{\varphi})).$$

After Lemma 2.4.8 there exists a functional $F \in \mathbf{B}$, so that

$$F(\vec{\varphi}) \quad = \quad \begin{cases} t_0(\vec{\varphi}), & \text{if } \theta(\vec{\varphi}, t_0(\vec{\varphi})), \\ t_1(\vec{\varphi}), & \text{else.} \end{cases}$$

And again, $F(\vec{\varphi})$ accords to the term $t(\vec{\varphi})$ we have been looking for:

$$QF - PIND \ \vdash \ \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t(\vec{\varphi})).$$

Case 4.2.:

$\Gamma(\vec{\varphi})$ is of the form $\Gamma(\vec{\varphi}),\ \exists z_0\theta_0(\vec{\varphi}, z_0)$, where $\exists z_0\theta_0(\vec{\varphi}, z_0)$ can go into $\Gamma(\vec{\varphi})$. The last conclusion looks like:

$$\frac{\Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi}, z),\ \theta_0(\vec{\varphi}, u_0)}{\Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi}, z),\ \exists z_0\theta_0(\vec{\varphi}, z_0)} \quad (\exists).$$

Under the induction hypothesis there exists a term $t_0(\vec{\varphi}) \in \mathbf{B}$, so that:

$$QF-PIND \vdash \Gamma(\vec{\varphi}),\ \theta(t_0(\vec{\varphi}),\vec{\varphi}),\ \theta_0(\vec{\varphi},u_0).$$

Now we apply an ($\exists$)-conclusion to the above received formula:

$$QF-PIND \vdash \Gamma(\vec{\varphi}),\ \theta(t_0(\vec{\varphi}),\vec{\varphi}),\ \exists z_0\theta_0(\vec{\varphi},z_0),$$

which accords to

$$QF-PIND \vdash \Gamma(\vec{\varphi}),\ \theta(t_0(\vec{\varphi}),\vec{\varphi}),$$

where $t_0(\vec{\varphi})$ is our term $t(\vec{\varphi})$.

Case 5:

The derivation ends with a ($\diamondsuit$)-conclusion (cut).
In this fifth case the last conclusion looks like:

$$\frac{\Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi},z),\ \theta_0 \qquad \Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi},z),\ \neg\theta_0}{\Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi},z)} \quad (\diamondsuit),$$

where θ_0 is an open formula.

Under the induction hypothesis there exist terms $t_0(\vec{\varphi}), t_1(\vec{\varphi}) \in$ $\mathbf{B}$,
so that:

$$QF-PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi},t_0(\vec{\varphi})),\ \theta_0,$$

and

$$QF-PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi},t_1(\vec{\varphi})),\ \neg\theta_0.$$

After an $(\vee)$-conclusion in the above case with $\theta(\vec{\varphi}, t_1(\vec{\varphi}))$ and in the lower case with $\theta(\vec{\varphi}, t_0(\vec{\varphi}))$, we receive:

$$QF - PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_0(\vec{\varphi})),\ \theta(\vec{\varphi}, t_1(\vec{\varphi})),\ \theta_0,$$

and

$$QF - PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_0(\vec{\varphi}),\ \theta(\vec{\varphi}, t_1(\vec{\varphi})),\ \neg\theta_0.$$

Now we apply the $(\diamondsuit)$-conclusion to these two received formulas, so:

$$QF - PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_0(\vec{\varphi})),\ \theta(\vec{\varphi}, t_1(\vec{\varphi})).$$

After Lemma 2.4.8 there exists a functional $F \in \mathbf{B}$, so that

$$F(\vec{\varphi}) \quad = \quad \begin{cases} t_0(\vec{\varphi}), & \text{if } \theta(\vec{\varphi}, t_0(\vec{\varphi})), \\ t_1(\vec{\varphi}), & \text{else.} \end{cases}$$

And again, $F(\vec{\varphi})$ accords to the term $t(\vec{\varphi})$ we have been looking for:

$$QF - PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t(\vec{\varphi})).$$

Case 6:

The derivation ends with a $(QF - PIND)$-conclusion.
So $\Gamma(\vec{\varphi})$ is of the form $\Gamma(\vec{\varphi}),\ \theta_0(\vec{\varphi}, t_0(\vec{\varphi}))$, where $\theta_0(\vec{\varphi}, t_0(\vec{\varphi}))$ can go into $\Gamma(\vec{\varphi})$.

In this sixth case the last conclusion looks like:

$$\frac{\Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi},z),\ \theta_0(\vec{\varphi},0) \qquad \Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi},z),\ \neg\theta_0(\vec{\varphi},\lfloor\frac{b}{2}\rfloor),\ \theta_0(\vec{\varphi},b)}{\Gamma(\vec{\varphi}),\ \exists z\theta(\vec{\varphi},z),\ \theta_0(\vec{\varphi},t_0(\vec{\varphi}))},$$

where θ_0 is an open formula.

Under the induction hypothesis there exist terms $t_1(\vec{\varphi}), t_2(\vec{\varphi}) \in$ **B**,
so that firstly

$$QF-PIND \ \vdash\ \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi},t_1(\vec{\varphi})),\ \theta_0(\vec{\varphi},0), \qquad (1)$$

and secondly

$$QF-PIND \ \vdash\ \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi},t_2(\vec{\varphi})),\ \neg\theta_0(\vec{\varphi},\lfloor\frac{b}{2}\rfloor),\ \theta_0(\vec{\varphi},b). \quad (2)$$

We apply the $(QF-PIND)$-conclusion to the just received sets of formulas in (1) and (2). Hence the following formula is also valid in $QF-PIND$:

$$[\theta_0(\vec{\varphi},0)\ \wedge\ \forall i<|t_0(\vec{\varphi})|\ (\theta_0(\vec{\varphi},t_0(\vec{\varphi})\restriction i)\rightarrow\theta_0(\vec{\varphi}),t_0(\vec{\varphi}\restriction(i{+}1)))]$$
$$\rightarrow$$
$$\theta_0(\vec{\varphi},t_0(\vec{\varphi}),).$$

Again as a consequence of the availability of the above formula in
$QF-PIND$, the next formula also holds in $QF-PIND$:

$$(\theta_0(\vec{\varphi},0)\ \rightarrow\ \theta_0(\vec{\varphi},t_0(\vec{\varphi})))$$
$$\vee$$
$$[\exists i<|t_0(\vec{\varphi})|\ (\theta_0(\vec{\varphi},t_0(\vec{\varphi})\restriction i)\wedge\neg\theta_0(\vec{\varphi},t_0(\vec{\varphi})\restriction(i+1)))].$$

After theorem 2.4.9 there exists a functional $F \in \mathbf{B}$, so that

$$\exists i < |t_0(\vec{\varphi})| \; (\theta_0(\vec{\varphi}, t_0(\vec{\varphi}) \upharpoonright i) \wedge \neg\theta_0(\vec{\varphi}, t_0(\vec{\varphi}) \upharpoonright (i+1)))$$

$$\leftrightarrow$$

$$(\theta_0(\vec{\varphi}, \lfloor \frac{F(t_0(\vec{\varphi})}{2})\rfloor) \wedge \neg\theta_0(\vec{\varphi}, F(t_0(\vec{\varphi}))).$$

Hence the formula

$$(\theta_0(\vec{\varphi}, 0) \;\rightarrow\; \theta_0(\vec{\varphi}, t_0(\vec{\varphi})))$$

$$\vee$$

$$(\theta_0(\vec{\varphi}, \lfloor \frac{F(t_0(\vec{\varphi})}{2})\rfloor) \wedge \neg\theta_0(\vec{\varphi}, F(t_0(\vec{\varphi})))$$

is valid in $QF - PIND$, which we can also write as:

$QF - PIND \vdash$

$$\neg\theta_0(\vec{\varphi}, 0), \;\theta_0(\vec{\varphi}, t_0(\vec{\varphi})), \;\theta_0(\vec{\varphi}, \lfloor \frac{F(|t_0(\vec{\varphi}|)\rfloor}{2}), \tag{3}$$

and

$QF - PIND \vdash$

$$\neg\theta_0(\vec{\varphi}, 0), \;\theta_0(\vec{\varphi}, t_0(\vec{\varphi})), \;\neg\theta_0(\vec{\varphi}, F(|t_0(\vec{\varphi}|)) \tag{4}$$

respectively.

Now we replace the free variable b in (2) by $F(|t_0(\vec{\varphi}|)$:

$QF - PIND \vdash$

$$\Gamma(\vec{\varphi}), \;\theta(\vec{\varphi}, t_2(\vec{\varphi})), \;\neg\theta_0(\vec{\varphi}, \lfloor \frac{F(|t_0(\vec{\varphi}|)}{2}\rfloor), \;\theta_0(\vec{\varphi}, F(|t_0(\vec{\varphi}|)) \tag{5}$$

After an application of the $(\Diamond)$-rule to (3) and (5) we receive:

$QF - PIND \vdash$

$$\Gamma(\vec{\varphi}), \;\theta(\vec{\varphi}, t_2(\vec{\varphi})), \;\theta_0(\vec{\varphi}, t_0(\vec{\varphi})), \;\neg\theta_0(\vec{\varphi}, 0), \;\theta_0(\vec{\varphi}, F(|t_0(\vec{\varphi}|)). \tag{6}$$

We apply the ($\diamondsuit$)-rule once more to (4) and (6):

$$QF - PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_2(\vec{\varphi}),\ \theta_0(\vec{\varphi}, t_0(\vec{\varphi})),\ \neg\theta_0(\vec{\varphi}, 0). \quad (7)$$

After a final application of the ($\diamondsuit$)-rule to (1) and (7) we receive:

$$QF{-}PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_1(\vec{\varphi})),\ \theta(\vec{\varphi}, t_2(\vec{\varphi})),\ \theta_0(\vec{\varphi}, t_0(\vec{\varphi})),$$

which accords to

$$QF - PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t_1(\vec{\varphi})),\ \theta(\vec{\varphi}, t_2(\vec{\varphi})).$$

And again after Lemma 2.4.8 there exists a functional $G \in \mathbf{B}$, so that

$$G(\vec{\varphi}) \quad = \quad \begin{cases} t_1(\vec{\varphi}), & \text{if } \theta(\vec{\varphi}, t_1(\vec{\varphi})), \\ t_2(\vec{\varphi}), & \text{else.} \end{cases}$$

The functional $G(\vec{\varphi})$ accords to the term $t(\vec{\varphi})$ we have been looking for:

$$QF - PIND \vdash \Gamma(\vec{\varphi}),\ \theta(\vec{\varphi}, t(\vec{\varphi})).$$

♣

We don't have to prove the case of the ($\forall$)-conclusion, because: According to the assumption, $\Gamma(\vec{\varphi})$ only contains open formulas or Σ_1^0-formulas. Furthermore our proof is partial cut-free which implies that there can't have been any universal quantifiers in the whole proof.

3 Main Theorem

In this third section we finally want to prove our main theorem 3.3. Hence we are going to show that for any *type*-2 functional F the following is valid:

$$F \in BFF$$
$$\Longleftrightarrow$$
$$F \text{ is } \Sigma_1^b\text{-definable in the theory } \mathbf{S}_2^1.$$

We start with the easier direction "$\Longrightarrow$" in the theorem 3.1.
The proof of the converse "$\Longleftarrow$" is identical to the Buss proof of the corresponding result for S_2^1 and the polynomial time computable functions (see Buss[1986]), with one single exception:
We are going to use Sieg's method of Herbrand Analyses from Sieg[1985].
Lemma 3.2 in the second part of this section, replaces Lemma 1.3.4 from Sieg[1991]. The essential step of the whole proof, which we already mentioned in the Abstract, is contained in the proof of this Lemma 3.2:
Because of the fact that the *Basic Feasible Functionals* are closed under *multiple limited recursion on notation* and by the closure of the language **B**, we are allowed to substitute a chain of interdependent existential quantifiers by *Basic Feasible Functionals*, which are closed under *mlrn*.
In the proof of the main theorem 3.3 we finally inter alia combine both proofs of theorem 3.1 and Lemma 3.2.

3.1 Theorem

Every *Basic Feasible Functional* F is Σ_1^b-definable in the theory $\mathbf{S}_2^1$.

Proof:

As we know, the class BFF of the *Basic Feasible Functionals* (see definition 1.2.3) is closed under *expansion*, *functional composition* and *limited recursion on notation**.

Consequently we have exactly these three possibilities for our functional F:

1. F is defined by *expansion*.

2. F is defined by *functional composition*.

3. F is defined by *limited recursion on notation**.

To avoid having very large formulas in this proof, we apply the following abbreviations (here in relation to F, respectively Θ_F):

- $\vec{\mathbf{t}}^F$ for $\vec{t}_1, \ldots, \vec{t}_k$,

- $\vec{\mathbf{z}}^F$ for $\vec{z}_1, \ldots, \vec{z}_k$,

- $\vec{\mathbf{z}}^F \leq \vec{\mathbf{t}}^F$ for $\vec{z}_1 \leq \vec{t}_1(\vec{\xi}) \wedge \ldots \wedge \vec{z}_k \leq \vec{t}_k(\vec{\xi}, \vec{z}_1, \ldots, \vec{z}_{k-1})$,

- $\exists \vec{\mathbf{z}}^F \leq \vec{\mathbf{t}}^F$ for $\exists \vec{z}_1 \leq \vec{t}_1(\vec{\xi}) \ldots \exists \vec{z}_k \leq \vec{t}_k(\vec{\xi}, \vec{z}_1, \ldots, \vec{z}_{k-1})$.

We prove by induction on the respective complexity of definition of F that there exists a Σ_1^b formula $\Theta_F(\vec{\varphi}, \vec{\mathbf{w}}^F, v)$ and a sequence of terms $\vec{\mathbf{t}}^F, t_{k+1}$, such that

$$\mathbf{S}_2^1 \quad \vdash \quad \forall \vec{\xi} \, [\, \exists \vec{\mathbf{z}}^F \leq \vec{\mathbf{t}}^F \, \exists! y \leq t_{k+1}(\vec{\xi}, \vec{\mathbf{z}}^F) \; \Theta_F(\vec{\xi}, \vec{\mathbf{z}}^F, y) \,],$$

where

$$\langle \mathbb{N}^k, (\mathbb{N}^{\mathbb{N}})^m \rangle \quad \models \quad \forall \vec{\xi} \, (\exists \vec{\mathbf{z}}^F \leq \vec{\mathbf{t}}^F \,) \; \Theta_F(\vec{\xi}, \vec{\mathbf{z}}^F, F(\vec{\xi})),$$

because this directly implies the Σ_1^b-definability of F in $\mathbf{S}_2^1$ (see definition 2.1.4).

From the fact that:

$$\mathbf{S}_2^1 \quad \vdash \quad \forall \vec{\xi} \, [\, \exists \vec{\mathbf{z}}^F \leq \vec{\mathbf{t}}^F \, \exists! y \leq t_{k+1}(\vec{\xi}, \vec{\mathbf{z}}^F) \; \Theta_F(\vec{\xi}, \vec{\mathbf{z}}^F, y) \,],$$

it follows that

$$\mathbf{S}^1_2 \quad \vdash \quad \forall \vec{\xi}\ \exists ! y\ (\exists \vec{\mathbf{z}}^F \leq\ \vec{\mathbf{t}}^F)\ [\ y \leq t_{k+1}(\vec{\xi}, \vec{\mathbf{z}}^F)\ \wedge\ \Theta_F(\vec{\xi}, \vec{\mathbf{z}}^F, y)\ ].$$

By the informations on page 19 the formula $\Psi_F(\vec{\varphi}, v)$, defined as

$$\Psi_F(\vec{\varphi}, v) \quad \equiv \quad (\exists \vec{\mathbf{z}}^F \leq\ \vec{\mathbf{t}}^F)\ [\ v \leq t_{k+1}(\vec{\varphi}, \vec{\mathbf{z}}^F)\ \wedge\ \Theta_F(\vec{\varphi}, \vec{\mathbf{z}}^F, v)\ ].$$

is a Σ^b_1 formula.

So we have

$$\mathbf{S}^1_2 \quad \vdash \quad \forall \vec{\xi}\ \exists ! y\ \Psi_F(\vec{\xi}, \vec{\mathbf{z}}^F, y),$$

which accords to the wanted form.

Because of the reason that there exist three different cases for the definition of F, our proof is subdevided into the corresponding three parts. The first and second part will be easy to understand. But because of the recursivity, the third part will demand more from us to get the idea.

1. F is defined by expansion

F is defined by *expansion* from $G(\vec{\varphi})$ (see definition 1.2.6):

$$F(\vec{f}, \vec{g}, \vec{u}, \vec{v}) \quad = \quad G(\vec{\varphi}).$$

By induction hypothesis there exists a formula Θ_G and a sequence of terms $\vec{\mathbf{t}}^G, t^G_{k+1}$, such that

$$\mathbf{S}^1_2 \quad \vdash \quad \forall \vec{\xi}\ [\ (\exists \vec{\mathbf{z}}^G \leq \vec{\mathbf{t}}^G)\ \exists ! y \leq t^G_{k+1}(\vec{\xi}, \vec{\mathbf{z}}^G)\ \Theta_G(\vec{\xi}, \vec{\mathbf{z}}^G, y)\ ].$$

We define Θ_F as follows:

$$\Theta_F \quad \equiv \quad \Theta_G,$$

and correspondingly:

$$\vec{\mathbf{z}}^F \equiv \vec{\mathbf{z}}^G, \quad \text{as well as} \quad \vec{\mathbf{t}}^F \equiv \vec{\mathbf{t}}^G.$$

Hence the induction hypothesis directly implies that

$$\mathbf{S}^1_2 \quad \vdash \quad \forall \vec{\xi} \, [\, (\exists \vec{\mathbf{z}}^F \le \vec{\mathbf{t}}^F) \; \exists! y \le t^F_{k+1}(\vec{\xi}, \vec{\mathbf{z}}^F) \; \Theta_F(\vec{\xi}, \vec{\mathbf{z}}^F, y) \,].$$

2. F is defined by functional composition

F is defined by *functional composition* from $G_1(\vec{\varphi}), \ldots, G_l(\vec{\varphi})$, $H(\vec{f}, \vec{v}, \vec{u})$ (see definition 1.2.5):

$$F(\vec{u}, \vec{v}) \quad = \quad H(\vec{f}, G_1(\vec{\varphi}), \ldots, G_l(\vec{\varphi}), \vec{u}).$$

By the induction hypothesis there exist formulas $\Theta_{G_1}, \ldots, \Theta_{G_l}$, Θ_H and sequences of terms $\vec{\mathbf{t}}^{G_1}, t^{G_1}_{k+1}, \ldots, \vec{\mathbf{t}}^{G_l}, t^{G_l}_{k+1}$ and $\vec{\mathbf{t}}^H, t^H_{k+1}$, such that firstly for all $1 \le j \le l$

$$\mathbf{S}^1_2 \quad \vdash \quad \forall \vec{\xi} \, [\, (\exists \vec{\mathbf{z}}^{G_j} \le \vec{\mathbf{t}}^{G_j}) \; \exists! g_j \le t^{G_j}_{k+1}(\vec{\xi}, \mathbf{z}^{G_j}) \; \Theta_{G_j}(\vec{\xi}, \vec{\mathbf{z}}^{G_j}, g_j) \,],$$

and secondly

$$\mathbf{S}^1_2 \quad \vdash \quad \forall \vec{\xi} \, \forall \vec{g} \, [\, (\exists \vec{\mathbf{z}}^H \le \vec{\mathbf{t}}^H) \; \exists! h \le t^H_{k+1}(\vec{\xi}, \vec{g}, \vec{\mathbf{z}}^H) \; \Theta_H(\vec{\xi}, \vec{g}, \vec{\mathbf{z}}^H, h) \,],$$

at which $\vec{g}$ is an abbreviation of $g_1, \ldots, g_l$.

Correspondingly to the definition of F, we define the formula Θ_F as follows:

$$\Theta_F(\vec{\varphi}, \vec{\mathbf{w}}^{G_1}, \ldots, \vec{\mathbf{w}}^{G_l}, g_1, \ldots, g_l, h)$$

$$\equiv$$

$$\Theta_{G_1}(\vec{\varphi}, \vec{\mathbf{w}}^{G_1}, g_1) \; \wedge \; \ldots \; \wedge \; \Theta_{G_l}(\vec{\varphi}, \vec{\mathbf{w}}^{G_l}, g_l) \; \wedge \; \Theta_H(\vec{\varphi}, \vec{g}, \vec{\mathbf{w}}^H, h).$$

Hence the induction hypothesis directly implies that

$\mathbf{S}^1_2 \vdash$

$\forall\vec{\xi}\ [\ (\exists\vec{\mathbf{z}}^{G_1} \leq \vec{\mathbf{t}}^{G_1})\ \exists g_1 \leq t^{G_1}_{k+1}(\vec{\xi}, \mathbf{z}^{G_1}) \ldots (\exists\vec{\mathbf{z}}^{G_l} \leq \vec{\mathbf{t}}^{G_l})\ \exists g_l \leq t^{G_l}_{k+1}(\vec{\xi}, \mathbf{z}^{G_l})$

$\exists\vec{\mathbf{z}}^{H} \leq \vec{\mathbf{t}}^{H} \exists! h \leq t^{H}_{k+1}(\vec{\xi}, \vec{g}, \vec{\mathbf{z}}^{H})\ \Theta_F(\vec{\xi}, \vec{\mathbf{z}}^{G_1}, \ldots, \vec{\mathbf{z}}^{G_l}, g_1, \ldots, g_l, h)\ ].$

3. F is defined by limited recursion on notation*

F is defined by limited recursion on notation* from G, H and K (see definition 1.2.9):

$$\begin{aligned} F(\vec{\varphi}, 0) &= G(\vec{\varphi}), && \text{("initial value")} \\ F(\vec{\varphi}, v) &= H(\vec{\varphi}, v, F(\vec{\varphi}, \lfloor \tfrac{v}{2} \rfloor)), \quad v > 0, && \text{("recursion on notation")} \\ |F(\vec{\varphi}, v)| &\leq |K(\vec{\varphi}, v)|, && \text{("bounding")} \end{aligned}$$

for all $\vec{\varphi}$.

By induction hypothesis there are formulas Θ_G, Θ_H, Θ_K and terms $\vec{\mathbf{t}}^{G}, t^{G}_{k+1}$, $\vec{\mathbf{t}}^{H}, t^{H}_{k+1}$, $\vec{\mathbf{t}}^{K}, t^{K}_{k+1}$, such that:

$(a)\quad \mathbf{S}^1_2 \vdash\ \forall\vec{\xi}\ [(\exists\vec{\mathbf{z}}^{G} \leq \vec{\mathbf{t}}^{G})\ \exists! y_G \leq t^{G}_{k+1}(\vec{\xi}, \vec{\mathbf{z}}^{G})\ \Theta_G(\vec{\xi}, \vec{\mathbf{z}}^{G}, y_G)],$

$(b)\quad \mathbf{S}^1_2 \vdash\ \forall\vec{\xi}\forall y \forall z\ [(\exists\vec{\mathbf{z}}^{H} \leq \vec{\mathbf{t}}^{H})\ \exists! y_H \leq t^{H}_{k+1}(\vec{\xi}, y, z, \vec{\mathbf{z}}^{H})\ \Theta_H(\vec{\xi}, y, z, \vec{\mathbf{z}}^{H}, y_H)],$

$(c)\quad \mathbf{S}^1_2 \vdash\ \forall\vec{\xi}\forall y\ [(\exists\vec{\mathbf{z}}^{K} \leq \vec{\mathbf{t}}^{K})\ \exists! y_K \leq t^{K}_{k+1}(\vec{\xi}, y, \vec{\mathbf{z}}^{K})\ \Theta_K(\vec{\xi}, y, \vec{\mathbf{z}}^{K}, y_K)].$

To reach our aim in this third part as well, we have to bring the "initial value"-, the "recursion on notation"- and the "bounding"-part together in one Σ^b_1-formula Θ_F and its corresponding prefix.

According to this we are going to prove by induction on y, that

$$\mathbf{S}^1_2 \quad \vdash \quad \forall\vec{\xi}\forall y \ \Psi_F(\vec{\xi}, y, t, e, \vec{\mathbf{z}}^K, d),$$

where

$$\Psi_F(\vec{\xi}, y, t, e, \vec{\mathbf{z}}^K, d)$$
$$\equiv$$
$$\exists\vec{\mathbf{z}}^K \leq \vec{\mathbf{t}}^K \ \exists t \leq y \ \exists e \leq SqBd(t^K_{k+1}(\vec{\xi}, t, \vec{\mathbf{z}}^K), y) \ \exists! d \leq e \ \Theta_F(\vec{\xi}, y, t, e, \vec{\mathbf{z}}^K, d).$$

The Σ^b_1-formula Θ_F is the conjunction of the following two formulas Θ^1_F under (a) and Θ^2_F under (b), whereas the "bounding part" is included in both formulas:

(a) "initial value" and "bounding":

$$\Theta^1_F \quad \equiv \quad (\exists\vec{\mathbf{z}}^G \leq \vec{\mathbf{t}}^G) \ \exists! y_G \leq t^G_{k+1}(\vec{\xi}, \vec{\mathbf{z}}^G) \ [\ A \quad \wedge \quad B \quad \wedge \quad C\ ];$$

$A \ \equiv \ \Theta_G(\vec{\xi}, \vec{\mathbf{z}}^G, y_G)$

corresponds to the functional value $F(\vec{\xi}, 0)$,

$B \ \equiv \ \exists q_0 \leq t^K_{k+1}(\vec{\xi}, t, \vec{\mathbf{z}}^K) \ \Theta_K(\vec{\xi}, 0, \vec{\mathbf{z}}^K, q_0)$

corresponds to the bounding value q_0 of $F(\vec{\xi}, 0)$,

$C \ \equiv \ (e)_0 = Min(y_G, q_0)$

corresponds to the first instance of the sequence of e: $(e)_0 \ = \ F(\vec{\xi}, 0) \ = \ Min(y_G, q_0) \ = \ y_G$. The functional value y_G has to be less or equal to its bounding value q_0.

(b) "recursion on notation" and "bounding":

$$\Theta^2_F \quad \equiv \quad (\forall i < |y|)\ [\ L \quad \wedge \quad M \quad \wedge \quad N \quad \wedge \quad P \quad \wedge \quad Q\ ];$$

$L \ \equiv\ \exists q_{i+1} \leq t^K_{k+1}(\vec{\xi}, t, \vec{\mathbf{z}}^K)\ \Theta_K(\vec{\xi}, y \restriction (i+1), \vec{\mathbf{z}}^K, q_{i+1})$

corresponds to the bounding values $q_1, \ldots, q_{|y|}$ of the functional values:
$F(\vec{\xi}, 1) = F(\vec{\xi}, y \restriction 1)$,
$\ldots$
$F(\vec{\xi}, \lfloor \frac{y}{4} \rfloor) = F(\vec{\xi}, y \restriction (|y| - 2))$,
$F(\vec{\xi}, \lfloor \frac{y}{2} \rfloor) = F(\vec{\xi}, y \restriction (|y| - 1))$,
$F(\vec{\xi}, y) = F(\vec{\xi}, y \restriction |y|)$,

$M \ \equiv\ (e)_i \leq t^K_{k+1}(\vec{\xi}, t, \vec{\mathbf{z}}^K)$

corresponds to the instances
$e_0 \ = \ F(\vec{\xi}, 0)$,
$e_1 \ = \ F(\vec{\xi}, 1)$,
$\ldots$
$e_{|y|-2} \ = \ F(\vec{\xi}, \lfloor \frac{y}{4} \rfloor)$,
$e_{|y|-1} \ = \ F(\vec{\xi}, F(\vec{\xi}, \lfloor \frac{y}{2} \rfloor))$
of the sequence of e. These are smaller or equal to their corresponding bounding values, which are otherwise smaller or equal to $t^K_{k+1}(\vec{\xi}, t, \vec{\mathbf{z}}^K)$,

$N \ \equiv\ (\exists \vec{\mathbf{z}}^H \leq \vec{\mathbf{t}}^H) \exists! y_H \leq t^H_{k+1}(\vec{\xi}, y \restriction i, (e)_i, \vec{\mathbf{z}}^H) \Theta_H(\vec{\xi}, y \restriction i, (e)_i, \vec{z}^H, y_H)$

corresponds to the functional values:
$F(\vec{\xi}, 0) \ = \ F(\vec{\xi}, y \restriction 0)$,
$F(\vec{\xi}, 1) \ = \ F(\vec{\xi}, y \restriction 1)$,

$$\ldots$$

$$F(\vec{\xi}, \lfloor \tfrac{y}{4} \rfloor) \;=\; F(\vec{\xi}, y \upharpoonright (|y|-2)),$$

$$F(\vec{\xi}, \lfloor \tfrac{y}{2} \rfloor) \;=\; F(\vec{\xi}, y \upharpoonright (|y|-1))).$$

Logically these are exactly the values we need, after the rules of the "recursion on notation*", to finally calculate $F(\vec{\xi}, y)$.

$P \;\equiv\; (e)_{i+1} = Min(y_H, q_{i+1})$

$Min(y_H, q_{i+1}) = y_H$ for all $i < |y|$, because the functional values are smaller or equal to their corresponding bounding values.

$Q \;\equiv\; (e)_{|y|} = d$

corresponds to the functional value $F(\vec{\xi}, y)$.

After this precise explanation of the Σ_1^b-formula $\Theta_F \equiv \Theta_F^1 \wedge \Theta_F^2$, we firstly turn to Buss's "Sequence Bound"-functional $SqBd(u, v)$ and afterwards to the buildup of the sequence of e:

$$SqBd(u, v) \;=\; (2v+1)\#(4(2u+1)^2) \;=\; 2^{(2v+1)\cdot(4(2u+1)^2)}.$$

The $SqBd$-Functional puts an upper bound on codes of sequences of length at most $|v|+1$, consisting of numbers less or equal than u.

Here in the formula Ψ_F we have

$$SqBd(t^K_{k+1}(\vec{\varphi}, t, \vec{\mathbf{w}}^K), v)$$

$$=$$

$$(2v + 1) \ \# \ (\ 4(2(t^K_{k+1}(\vec{\varphi}, t, \vec{\mathbf{w}}^K)) + 1)^2 \ )$$

$$=$$

$$(2v + 1) \ \# \ (\ 4(2(t^K_{k+1}(\vec{\varphi}, t, \vec{\mathbf{w}}^K)) + 1) \cdot (2(t^K_{k+1}(\vec{\varphi}, t, \vec{\mathbf{w}}^K)) + 1) \ )$$

$$=$$

$$[\ (S(S(0)) \cdot v + 1)$$

$$\#$$

$$(S(S(S(S(0)))) \cdot (S(S(0)) \cdot (t^K_{k+1}(\vec{\varphi}, t, \vec{\mathbf{w}}^K)) + 1) \cdot$$

$$\cdot (S(S(0))(t^K_{k+1}(\vec{\varphi}, t, \vec{\mathbf{w}}^K)) + 1))],$$

which is a term in $\mathbf{S}^1_2$. (see Buss's "Bounded Arithmetic", Chapter 4.1).

This fact implies that

$$\Psi_F(\vec{\xi}, y, t, e, \vec{\mathbf{w}}^K, d)$$

$$\equiv$$

$$\exists \vec{\mathbf{z}}^K \leq \vec{\mathbf{t}}^K \ \exists t \leq y \ \exists e \leq SqBd(t^K_{k+1}(\vec{\xi}, t, \vec{\mathbf{z}}^K), y) \ \exists ! d \leq e \ \Theta_F(\vec{\xi}, y, t, e, \vec{\mathbf{z}}^K, d)$$

is a Σ^b_1-formula in $\mathbf{S}^1_2$.

Hence we are allowed to apply the induction schema $(\Sigma^b_1 - PIND)$ (see 2.1.3) in this third part of our proof. But before doing this, let us have a look at the structure of e:

After the definition of $SqBd(u, v)$, every instance of the sequence of e has to be less or equal to $t^K_{k+1}(\vec{\xi}, t, \vec{\mathbf{z}}^K)$ and the length of the sequence of e must be less or equal to $|y| + 1$. Here

$$e \ = \ \langle \ F(\vec{\varphi}, 0), \ldots, F(\vec{\varphi}, \lfloor \frac{v}{8} \rfloor), F(\vec{\varphi}, \lfloor \frac{v}{4} \rfloor), F(\vec{\varphi}, \lfloor \frac{v}{2} \rfloor), F(\vec{\varphi}, v) \ \rangle,$$

with what all the conditions are fulfilled after the induction hypothesis.
Now then the sequence e contains those values we need to finally calculate $F(\vec{\varphi}, v)$ after the "recursion on notation". And at the last position $e_{|v|}$ there is the result $F(\vec{\varphi}, v) = d$:

$$
\begin{aligned}
(e)_0 &= F(\vec{\varphi},0),\\
&\vdots\\
(e)_{|v|-1} &= F(\vec{\varphi},\lfloor\frac{v}{2}\rfloor),\\
(e)_{|v|} &= F(\vec{\varphi},v).
\end{aligned}
$$

For example:

$v = 33 = 100001$ (in binary notation).
So $|v| = \lfloor \log_2(33) \rfloor = 6$, which implies that $|e| \leq 7$:

$$e \ = \ \langle\ F(\vec{\varphi},0), F(\vec{\varphi},1), F(\vec{\varphi},2), F(\vec{\varphi},4), F(\vec{\varphi},8), F(\vec{\varphi},16), F(\vec{\varphi},33)\ \rangle,$$

where

$$
\begin{aligned}
(e)_0 &= F(\vec{\varphi},0) \ = \ G(\vec{\varphi}),\\
(e)_1 &= F(\vec{\varphi},1) \ = \ H(\vec{\varphi},1,F(\vec{\varphi},\lfloor\frac{1}{2}\rfloor)) = \ H(\vec{\varphi},y,F(\vec{\varphi},0)),\\
&\vdots\\
(e)_6 &= F(\vec{\varphi},33) \ = \ H(\vec{\varphi},33,F(\vec{\varphi},\lfloor\frac{33}{2}\rfloor)) = \ H(\vec{\varphi},y,F(\vec{\varphi},16)).
\end{aligned}
$$

Now let us finally prove that

$$\mathbf{S}_2^1 \quad \vdash \quad \forall\vec{\xi}\forall y \ \Psi_F(\vec{\xi},y,t,e,\vec{\mathbf{z}}^K,d)$$

by induction on y with the induction schema $(\Sigma_1^b - PIND)$:

$$(\Sigma_1^b - PIND): \ [\ A(\vec{\varphi},0) \ \wedge \ \forall y(A(\vec{\varphi},\lfloor\frac{y}{2}\rfloor) \to A(\vec{\varphi}),y)\ ] \ \to \ \forall y A(\vec{\varphi},y),$$

where in our case the Σ_1^b-formula $A(\vec{\varphi},v)$ naturally corresponds to the formula Ψ_F.

1. Basis: $\Psi_F(\vec{\varphi}, 0, t, e, \vec{\mathbf{w}}^K, d)$

$\Psi_F(\vec{\varphi}, 0, t, e, \vec{\mathbf{w}}^K, d)$ accords to the "initial value", which is calculated in the formula Θ_F^1 under (a). $y = 0$ implies that also $t = 0$ and $i = 0$. So we only have one cycle through the formula (b). In this only cycle we do not more than calculating another time what we calculated before in the formula (a).

The fact that

$$\mathbf{S}_2^1 \quad \vdash \quad \Psi_F(\vec{\varphi}, 0, t, e, \vec{\mathbf{w}}^K, d)$$

follows directly from the induction hypothesis.

2. Inductive step: $\forall y(\Psi_F(\vec{\varphi}, \lfloor \frac{y}{2} \rfloor, t, e, \vec{\mathbf{w}}^K, d)) \quad \rightarrow \quad \Psi_F(\vec{\varphi}, y, t, e, \vec{\mathbf{w}}^K, d))$

accords to the formula Θ_F^2 under (b), the "recursion on notation", because for all y we need the value of $F(\vec{\varphi}, \lfloor \frac{y}{2} \rfloor)$ to calculate $F(\vec{\varphi, y})$.

We assume, that

$$\Psi_F(\vec{\varphi}, \lfloor \frac{v}{2} \rfloor, t, e, \vec{\mathbf{w}}^K, d)$$

holds in $\mathbf{S}_2^1$.

So we have a sequence $e = \langle\ F(\vec{\varphi}, 0), F(\vec{\varphi, 1}), \ldots, F(\vec{\varphi, \lfloor \frac{v}{2} \rfloor})\ \rangle$, where $d = F(\vec{\varphi, \lfloor \frac{v}{2} \rfloor})$ is the instance at the last position.
Every instance of e is less or equal to $t_{k+1}^K(\vec{\varphi}, t, \vec{\mathbf{w}}^K)$ for a given term t which is otherwise less or equal to $\lfloor \frac{v}{2} \rfloor$.
For these given t, e and d, the formula Ψ_F holds in the theory $\mathbf{S}_2^1$.

As a result of these facts, Ψ_F also holds in $\mathbf{S}_2^1$ for d', e' and t', which are defined as follows:

$$d' = F(\vec{\varphi, v}),$$

$$e' = \langle\ F(\vec{\varphi}, 0), F(\vec{\varphi, 1}), \ldots, F(\vec{\varphi, \lfloor \frac{v}{2} \rfloor}, F(\vec{\varphi, v}))\ \rangle,$$

$$t' = \begin{cases} t & \text{if } F(\vec{\varphi}, v) \leq t_{k+1}^K(\vec{\varphi}, t, \vec{\mathbf{w}}^K), \\ v & \text{else.} \end{cases}$$

The fact that $F(\vec{\varphi}, v) \leq t^K_{k+1}(\vec{\varphi}, v, \vec{\mathbf{w}}^K)$, follows directly from the induction hypothesis. Therefore and after the induction hypothesis every instance of e' is less or equal to t'.

♣

3.2 Lemma

Now we turn to the Lemma 3.2, which as mentioned in the beginning of this section, replaces Lemma 1.3.4 from Sieg[1991], and which contains the main part of the "$\Longleftarrow$"-proof of our theorem 3.3.

> Let Δ be a set containing only existential formulas (with the existential quantifier bounded or unbounded). Then
>
> $$n - \Sigma_1^b - PIND \quad \vdash \quad \Delta$$
> $$\Longrightarrow$$
> $$QF - PIND \quad \vdash \quad \Delta.$$

Proof:

Because of the fact that the two theories $n - \Sigma_1^b - PIND$ and $QF - PIND$ coincide with the exception of the complexity of the formula in the induction rule, we only have to prove this case.
We proceed by induction on the number of applications of the induction rule applied to $n - \Sigma_1^b$-formulas.
To remind: $n - \Sigma_1^b$-formulas contain at most n existential quantifiers followed by an open formula.

Let us suppose that the claim of the Lemma 3.2 holds for derivations with k applications of the induction rule.

Now we assume that

$$(n - \Sigma_1^b - PIND) \quad \vdash \quad \Delta$$

with a derivation d with $k + 1$ such applications of the induction rule.
After the cut elimination theorem for $n - \Sigma_1^b - PIND$ in 2.3.6 we can also assume, that all cuts in the derivation d are on induction or atomic formulas only.

So let us now examine this derivation d:

Consider a top-most instance of the induction rule applied to an $n - \Sigma_1^b$-formula ψ. By "top-most" we mean, that it concerns an application of the induction rule so that all other applications of the induction rule appearing above it in d are on open formulas. We surely ignore these applications of open induction, because our interest in this proof lies in reducing $(n - \Sigma_1^b - PIND)$-applications to $(QF - PIND)$-applications.

Now, let us have a look at the above mentioned "top-most" instance of the induction-rule applied to an $n - \Sigma_1^b$-formula ψ:

$$\frac{\Gamma(\vec{\varphi}), \quad \psi(\vec{\varphi}, 0) \qquad\qquad \Gamma(\vec{\varphi}), \quad \neg\psi(\vec{\varphi}, \lfloor \frac{b}{2} \rfloor), \ \psi(\vec{\varphi}, b)}{\Gamma(\vec{\varphi}), \quad \psi(\vec{\varphi}, t(\vec{\varphi}))}.$$

As denoted above we want to prove now that we can reduce this application of $(n - \Sigma_1^b - PIND)$ to an application of open induction.

Concerning the $n - \Sigma_1^b$-formula ψ, we prove this Lemma 3.2 explicitly for the case $n = 2$. Thus we keep the overview. For $n \geq 3$ we would have large formulas which generally are hard to read. Besides it is easy to conclude the proof of the general case $n \in \mathbb{N}$ from this proof of the case $n = 2$.

So, if $n = 2$, then $\psi(\vec{\varphi}, b)$ is of the form

$$\psi(\vec{\varphi}, b) \quad \equiv \quad \exists z_1 \leq t_0(\vec{\varphi}, b) \ \exists z_2 \leq t_1(\vec{\varphi}, z_1, b) \ \theta(\vec{\varphi}, z_1, z_2, b),$$

where b does not appear in the sequence of variables $\vec{\varphi}$, and θ is open.

Later in this proof we want to apply the $\exists$-Inversion 2.4.10. Therefore we have to convert the formula $\psi(b, \vec{\varphi})$ to the following defined formula $\psi'(\vec{\varphi}, b)$:

$$\psi'(\vec{\varphi}, b) \ \equiv \ \exists z((z)_1 \leq t_0(\vec{\varphi}, b) \ \wedge \ (z)_2 \leq t_1(\vec{\varphi}, (z)_1, b) \ \wedge \ \theta(\vec{\varphi}, (z)_1, (z)_2, b),$$

which fulfills the preconditioned form and accordingly allows us to apply the $\exists$-Inversion.

After our assumption in the beginning of the proof, the following holds:

(a_1) $QF - PIND \quad \vdash \quad \Gamma(\vec{\varphi}), \ \psi'(\vec{\varphi}, 0)$,

(b_1) $QF - PIND \quad \vdash \quad \Gamma(\vec{\varphi}), \ \neg\psi(\vec{\varphi}, \lfloor \frac{b}{2} \rfloor), \ \psi'(\vec{\varphi}, b)$.

From now on we separate the sets of large formulas by lines to make reading more comfortable. Hence the above sets of formulas, according to the definitions of ψ and ψ' above, tendered look like:

(a_2) $QF - PIND \quad \vdash$

$$\Gamma(\vec{\varphi}),$$

$$\exists z((z)_1 \leq t_0(\vec{\varphi}, 0) \ \wedge \ (z)_2 \leq t_1(\vec{\varphi}, (z)_1, 0) \ \wedge \ \theta(\vec{\varphi}, (z)_1, (z)_2, 0)),$$

(b_2) $QF - PIND \quad \vdash$

$$\Gamma(\vec{\varphi}),$$

$$\forall z_1 \leq t_0(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor) \ \forall z_2 \leq t_1(\vec{\varphi}, z_1, \lfloor \tfrac{b}{2} \rfloor) \ \neg\theta(\vec{\varphi}, z_1, z_2, \lfloor \tfrac{b}{2} \rfloor),$$

$$\exists z((z)_1 \leq t_0(\vec{\varphi}, b) \ \wedge \ (z)_2 \leq t_1(\vec{\varphi}, (z)_1, b) \ \wedge \ \theta(\vec{\varphi}, (z)_1, (z)_2, b)).$$

After applying twice the $(\forall)$-Inversion of the Tait-Calculus and some of the De Morgan's rules to the set of formulas in (b_2), we receive the following set of formulas in (b_3):

(b_3) $QF - PIND \quad \vdash$

$$\Gamma(\vec{\varphi}),$$

$$(u_1 \leq t_0(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor) \ \wedge \ u_2 \leq t_1(\vec{\varphi}, u_1, \lfloor \tfrac{b}{2} \rfloor)) \ \rightarrow \ \neg\theta(\vec{\varphi}, u_1, u_2, \lfloor \tfrac{b}{2} \rfloor),$$

$$\exists z((z)_1 \leq t_0(\vec{\varphi}, b) \ \wedge \ (z)_2 \leq t_1(\vec{\varphi}, (z)_1, b) \ \wedge \ \theta(\vec{\varphi}, (z)_1, (z)_2, b)),$$

where the new free variable u is not contained in the derivation leading to the set in (b_2).

Now we apply the $\exists$-Inversion 2.4.10 to the sets of formulas in (a_2) and in (b_3) and therefore receive the following sets in (a_4) and in (b_4). After the $\exists$-Inversion 2.4.10 there exist terms s_0, d_0, s_1, d_1 in the language **B**, so that:

(a_4) $QF - PIND \quad \vdash$

$$\Gamma(\vec{\varphi}),$$

$$s_0(\vec{\varphi}) \leq t_0(\vec{\varphi}, 0) \wedge d_0(\vec{\varphi}) \leq t_1(\vec{\varphi}, s_0(\vec{\varphi}), 0) \wedge \theta(\vec{\varphi}, s_0(\vec{\varphi}), d_0(\vec{\varphi}), 0),$$

$(b_4)\ QF-PIND \quad \vdash$

$$\Gamma(\vec{\varphi}),$$

$$(u_1 \leq t_0(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor) \ \wedge \ u_2 \leq t_1(\vec{\varphi}, u_1, \lfloor \tfrac{b}{2} \rfloor)) \ \rightarrow \ \neg\theta(\vec{\varphi}, u_1, u_2, \lfloor \tfrac{b}{2} \rfloor),$$

$$s_1(\vec{\varphi}, u_1, u_2, b) \leq t_0(\vec{\varphi}, b) \ \wedge$$

$$d_1(\vec{\varphi}, u_1, u_2, b) \leq t_1(\vec{\varphi}, s_1(\vec{\varphi}, u_1, u_2, b), b) \ \wedge$$

$$\theta(\vec{\varphi}, s_1(\vec{\varphi}, u_1, u_2, b), d_1(\vec{\varphi}, u_1, u_2, b), b).$$

Now then we define two functionals F and G as follows:

$$\begin{array}{lcl} F(\vec{\varphi}, 0) & = & s_0(\vec{\varphi}), \\ F(\vec{\varphi}, b) & = & s_1(\vec{\varphi}, F(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), G(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), b), \\ F(\vec{\varphi}, b) & \leq & t_0(\vec{\varphi}, b); \qquad \text{(see } (a_4) \text{ and } (b_4)\text{)} \end{array}$$

and

$$\begin{array}{lcl} G(\vec{\varphi}, 0) & = & d_0(\vec{\varphi}), \\ G(\vec{\varphi}, b) & = & d_1(\vec{\varphi}, F(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), G(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), b), \\ G(\vec{\varphi}, b) & \leq & t_1(\vec{\varphi}, F(\vec{\varphi}, b), b). \qquad \text{(see } (a_4) \text{ and } (b_4)\text{)} \end{array}$$

We see that both functionals F and G are defined by *multiple limited recursion on notation*. And by the closure of the language **B** such a definition is correct. Hence we are allowed to replace the free variables u_1 and u_2 in (b_4) by $F(\vec{\varphi}, \lfloor \frac{b}{2} \rfloor)$ and $G(\vec{\varphi}, \lfloor \frac{b}{2} \rfloor)$ and receive:

$(b_5)\ QF-PIND \quad \vdash$

$$\Gamma(\vec{\varphi}),$$

$$(F(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor) \leq t_0(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor) \quad \wedge \quad G(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor) \leq t_1(\vec{\varphi}, F(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), \lfloor \tfrac{b}{2} \rfloor)) \quad \rightarrow$$

$$\neg\theta(\vec{\varphi}, F(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), G(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), \lfloor \tfrac{b}{2} \rfloor),$$

$$s_1(\vec{\varphi}, F(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), G(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), b) \leq t_0(\vec{\varphi}, b) \quad \wedge$$

$$d_1(\vec{\varphi}, F(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), G(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), b) \leq t_1(\vec{\varphi}, s_1(\vec{\varphi}, F(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), G(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), b), b) \quad \wedge$$

$$\theta(\vec{\varphi}, s_1(\vec{\varphi}, F(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), G(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), b), d_1(\vec{\varphi}, F(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), G(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), b), b).$$

After the definition of the functionals F and G above we are also allowed to make the following substitutions to the sets of formulas in (a_4) and (b_5):

$s_0(\vec{\varphi})$ by $F(\vec{\varphi}, 0)$,

$d_0(\vec{\varphi})$ by $G(\vec{\varphi}, 0)$,

$s_1(\vec{\varphi}, F(\vec{\varphi}, \lfloor \frac{b}{2} \rfloor), G(\vec{\varphi}, \lfloor \frac{b}{2} \rfloor), b)$ by $F(\vec{\varphi}, b)$,

$d_1(\vec{\varphi}, F(\vec{\varphi}, \lfloor \frac{b}{2} \rfloor), G(\vec{\varphi}, \lfloor \frac{b}{2} \rfloor), b)$ by $G(\vec{\varphi}, b)$:

(a_6) $QF - PIND \quad \vdash$

$$\Gamma(\vec{\varphi}),$$
$$F(\vec{\varphi}, 0) \leq t_0(\vec{\varphi}, 0) \quad \wedge \quad G(\vec{\varphi}, 0) \leq t_1(\vec{\varphi}, F(\vec{\varphi}, 0), 0) \quad \wedge$$
$$\theta(\vec{\varphi}, F(\vec{\varphi}, 0), G(\vec{\varphi}, 0), 0),$$

(b_6) $QF - PIND \quad \vdash$

$$\Gamma(\vec{\varphi}),$$
$$(F(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor) \leq t_0(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor) \quad \wedge \quad G(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor) \leq t_1(\vec{\varphi}, F(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), \lfloor \tfrac{b}{2} \rfloor)) \quad \rightarrow$$
$$\neg\theta(\vec{\varphi}, F(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), G(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), \lfloor \tfrac{b}{2} \rfloor),$$
$$F(\vec{\varphi}, b) \leq t_0(\vec{\varphi}, b) \quad \wedge \quad G(\vec{\varphi}, b) \leq t_1(\vec{\varphi}, F(\vec{\varphi}, b), b) \quad \wedge$$
$$\theta(\vec{\varphi}, F(\vec{\varphi}, b), G(\vec{\varphi}, b), b).$$

Now we define an open formula θ_0 as follows:

$$\theta_0(\vec{\varphi}, v, w, b) \quad \equiv \quad v \leq t_0(\vec{\varphi}, b) \quad \wedge \quad w \leq t_1(\vec{\varphi}, v, b) \quad \wedge \quad \theta(\vec{\varphi}, v, w, b).$$

Then the above sets are of the following form:

(a_7) $QF - PIND \quad \vdash$

$$\Gamma(\vec{\varphi}), \quad \theta_0(\vec{\varphi}, F(\vec{\varphi}, 0), G(\vec{\varphi}, 0), 0),$$

(b_7) $QF - PIND \quad \vdash$

$$\Gamma(\vec{\varphi}), \quad \neg\theta_0(\vec{\varphi}, F(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), G(\vec{\varphi}, \lfloor \tfrac{b}{2} \rfloor), \lfloor \tfrac{b}{2} \rfloor), \ \theta_0(\vec{\varphi}, F(\vec{\varphi}, b), G(\vec{\varphi}, b), b).$$

And after applying the induction rule for open formulas, we receive:

$$QF - PIND \quad \vdash \quad \Gamma(\vec{\varphi}), \quad \theta_0(\vec{\varphi},\ F(\vec{\varphi}, t(\vec{\varphi})),\ G(\vec{\varphi}, t(\vec{\varphi})),\ t(\vec{\varphi})).$$

Hence after the definition of θ_0:

$$QF - PIND \quad \vdash$$

$$\Gamma(\vec{\varphi}),$$

$$F(\vec{\varphi}, t(\vec{\varphi})) \leq t_0(\vec{\varphi}, t(\vec{\varphi})) \quad \wedge$$

$$G(\vec{\varphi}, t(\vec{\varphi})) \leq t_1(\vec{\varphi}, F(\vec{\varphi}, t(\vec{\varphi})), t(\vec{\varphi})) \quad \wedge$$

$$\theta(\vec{\varphi},\ F(\vec{\varphi}, t(\vec{\varphi})),\ G(\vec{\varphi}, t(\vec{\varphi})),\ t(\vec{\varphi})).$$

So:

$$QF{-}PIND \quad \vdash \quad \exists z_1 \leq t_0(\vec{\varphi}, b)\ \exists z_2 \leq t_1(\vec{\varphi}, z_1, b)\ \theta(\vec{\varphi}, z_1, z_2, b).$$

And we have arrived at the end of our proof:

$$QF - PIND \quad \vdash \quad \Gamma(\vec{\varphi}), \quad \psi(\vec{\varphi}, t(\vec{\varphi})).$$

♣

3.3 Main Theorem

The class *BFF* of the *Basic Feasible Functionals* is exactly the class of Σ_1^b provably total functions of $\mathbf{S}_2^1$.

Proof:

"$\Longrightarrow$":

See theorem 3.1.

"$\Longleftarrow$":

After definition 2.1.4, for every Σ_1^b provably total function F of $\mathbf{S}_2^1$, there exists a Σ_1^b-formula Ψ_F, so that

$$\mathbf{S}_2^1 \quad \vdash \quad \forall \vec{\xi} \; \exists ! y \; \Psi_F(\vec{\xi}, y),$$

which implies that there exists an $\mathbf{S}_2^1$-proof d with $depth(d) < \infty$ to prove this formula $\forall \vec{\xi} \; \exists ! y \; \Psi_F(\vec{\xi}, y)$. So we also have a finite number m of applications of the $\mathbf{S}_2^1$ induction schema to Σ_1^b-formulas $\alpha_1, \ldots, \alpha_m$ in this derivation d.
Σ_1^b-formulas without any sharply bounded universal quantifiers are called strict Σ_1^b-formulas. Because of the fact that every Σ_1^b-formula is equal to a strict Σ_1^b-formula (see Buss[1986]), and therewith to an $n - \Sigma_1^b$-formula, we can say that also:

$$n - \Sigma_1^b - PIND \quad \vdash \quad \forall \vec{\xi} \; \exists ! y \; \Psi_F(\vec{\xi}, y)$$

for some $n \in \mathbb{N}$, which we define as follows:
After the above informations, every formula $\alpha_i, 1 \leq i \leq m$ can be written as an $n - \Sigma_1^b$-formula. We now take that formula $\alpha_i, 1 \leq i \leq m$, which has the greatest number of bounded existential quantifiers in its prefix. That greatest number accords to our n.

We can also assume that the formula Ψ_F is a strict Σ_1^b-formula.

So by the previous Lemma 3.2:

$$QF - PIND \quad \vdash \quad \forall\vec{\xi}\,\exists!y\ \Psi_F(\vec{\xi}, y),$$

which directly implies that:

$$QF - PIND \quad \vdash \quad \forall\vec{\xi}\,\exists y\ \Psi_F(\vec{\xi}, y).$$

After applying the $\forall$-Inversion we have:

$$QF - PIND \quad \vdash \quad \exists y\ \Psi_F(\vec{\varphi}, y).$$

And after the $\exists$-Inversion 2.4.10 there exists a term $t(\vec{\varphi}) \in \mathbf{B}$, so that:

$$QF - PIND \quad \vdash \quad \Psi_F(\vec{\varphi}, t(\vec{\varphi})).$$

Because of the fact that all the terms in $\mathbf{B}$ correspond to *Basic Feasible Functionals*, we have shown that the class of Σ_1^b provably total functions of $QF - PIND$, respectively of $\mathbf{S}_2^1$, is exactly the class of the BFF's.

♣

Bibliography

[1]

Aleksandar Ignjatovic and Arun Sharma:
"Some Applications of Logic to Feasibility in Higher Types",
Association for Computing Machinery (ACM),
Transactions on Computational Logic (TOCL),
Volume 5, P. 332-350, April 2004.

[2]

Samuel R. Buss:
"Bounded Arithmetic",
Bibliopolis, Naples, 1986.

[3]

Samuel R. Buss:
"Bounded Arithmetic and Propositional Proof Complexity" in:
"Logic of Computation",
edited by H. Schwichtenberg, P. 67-122,
Springer-Verlag, Berlin, 1997.

[4]

Samuel R. Buss:
"Handbook of Proof Theory",
Elsevier, North-Holland, 1998.

[5]

Stephen A. Cook and Bruce M. Kapron:
"Characterizations of the Basic Feasible Functionals of Finite Type", in:
Feasible Mathematics: A Mathematical Sciences Institute Workshop,
Eds. S. Buss, P. Scott, P. 71-96,
Birkhauser, 1990.

[6]

Stephen A. Cook and Bruce M. Kapron:
"A new Characterization of Type 2 Feasibility"
SIAM J. on Computing, 25(1), P.117-132, 1996.

[7]

P. Clote:
"Computation Models and Function Algebras", in:
Handbook of Computability Theory, ed. E.R. Griffor,
Elsevier Science B.V., P. 589-681, 1999.

[8]

Gerhard Jaeger:
"Einfuehrung in die theoretische Informatik",
Vorlesungsskript, Sommersemester 2004.

[9]

Gerhard Jaeger:
"Logik und Informatik",
Vorlesungsskript, Sommersemester 2003/2004.

[10]

J. Barwise:
"Handbook of Mathematical Logic",
North-Holland Publishing Company, 1977.