System T and the Product of Selection Functions[*]

Martín Escardó[1], Paulo Oliva[2], and Thomas Powell[2]

1 **University of Birmingham**
Department of Computer Science
`m.escardo@cs.bham.ac.uk`

2 **Queen Mary University of London**
School of Electronic Engineering and Computer Science
`{paulo.oliva, tpowell}@eecs.qmul.ac.uk`

Abstract

We show that the finite product of selection functions (for all finite types) is primitive recursively equivalent to Gödel's higher-type recursor (for all finite types). The correspondence is shown to hold for similar restricted fragments of both systems: The recursor for type level $n+1$ is primitive recursively equivalent to the finite product of selection functions of type level n. Whereas the recursor directly interprets induction, we show that other classical arithmetical principles such as bounded collection and finite choice are more naturally interpreted via the product of selection functions.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases Primitive recursion, product of selection functions, finite choice, dialectica interpretation

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

In his 1958 paper published in the journal *dialectica* [7] Gödel introduced a novel interpretation of intuitionistic arithmetic into a quantifier-free calculus of functionals, the so-called system **T**. System **T** is essentially primitive recursive arithmetic PRA with the schema of recursion extended to all finite types $\{\mathbb{N}, \mathbb{N} \to \mathbb{N}, (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}, \ldots\}$. Gödel's aim was to show that quantifier dependencies in arithmetic could be captured by this class of primitive recursive functionals, and therefore that the consistency of arithmetic could be reduced to that of system **T**.

In Gödel's dialectica interpretation, the recursors play a fundamental role in the interpretation of the induction axioms. Parsons [15] studied the precise relationship between the complexity of the type of the recursor and the logical complexity of the induction formula, establishing a correspondence between the well-known fragments of arithmetic based on restricted induction and fragments of system **T** based on restricted recursion.

The dialectica interpretation of arithmetic was quickly extended to classical analysis by Spector [17], via a new form of recursion on well-founded trees known as *bar recursion*. Spector's dialectica interpretation of arithmetical comprehension goes via the classical axiom of countable choice, which in turn is reduced to the double negation shift (see [17] for details)

$$\forall i \neg\neg A(i) \to \neg\neg \forall i A(i).$$

* This work was partially supported by Royal Society (2nd author) grant 516002.K501/RH/kk, and an EPSRC Doctoral Training Account (3rd author).

Hence, the computational interpretation of full classical analysis was reduced to the interpretation of this seemingly harmless (obviously true) principle.

The first two authors have recently shown [3, 2, 5] that Spector's bar recursion is primitive recursively equivalent to an unbounded iteration of the product of *selection functions*, a highly intuitive construction that has appeared over and over again in various guises in game theory, fixed point theory, algorithms, and proof theory. For further details see Section 3, their original paper [3] or a recent survey [6].

In [3] it is observed that just as an unbounded iteration of the product of selection functions provides an intuitive interpretation of the double negation shift, a finite iteration of the product directly interprets the *finite* double negation shift

$$\forall m(\forall i \leqslant m \neg\neg A(i) \to \neg\neg\forall i \leqslant m A(i)),$$

which in turn is closely related to a number of well-known 'set theoretic' principles such as finite choice and bounded collection. In Section 4 we show that just as the computational analogue of induction is Gödel's primitive recursion on all finite type, a natural computational analogue of *finite choice* is given by the product of selection functions. Furthermore, analogous to Parsons result for induction we establish a correspondence between the logical complexity of the choice formula and the complexity of the product.

It is well-known, however, that both bounded collection and finite choice are equivalent to number induction. More specifically, Parsons proved that the hierarchy of bounded collection axioms is strictly interleaving on the hierarchy of induction axioms [14]. Therefore it is natural to ask precisely how the product of selection functions is related to primitive recursion. Our main result (Section 5) is that Gödel's primitive recursor of type level $n+1$ is equivalent, over a weak base theory, to the finite product of selection functions of type level n. In particular, the finite product of selection functions of lowest types already defines the Ackermann function.

There are several advantages of considering this equivalent form of the Gödel primitive recursor R

1. Given the equivalence of the unbounded product of selection functions with Spector's bar recursion [3], the equivalence of the finite product with the recursor R provides for a smooth passage from the functional interpretation of arithmetic (finite fixed number of iterations) to that of analysis (finite but unbounded number of iterations). Therefore, we obtain the correspondence

$$\frac{\text{Finite product of selection functions}}{\text{Arithmetic}} = \frac{\text{Unbounded product of selection functions}}{\text{Analysis}}$$

   In this sense, the product of selection functions allows for a uniform transition from arithmetic to analysis. This is discussed further in Section 6.
2. The product of selection functions has a natural reading in terms of calculations of optimal strategies in sequential games. Hence, witnessing terms involving the product of selection functions (rather than R or Spector's bar recursion) can normally be given a clear intuitive meaning. The connection between games and the product is explained in Section 3.
3. Whereas the recursor directly interprets induction, we show that when interpreting other classical arithmetical principles closely related to bounded collection and finite choice it is the finite product of selection functions which allows for a direct, and hence more illuminating, interpretation. See Section 4 for details.

2 Fragments of Arithmetic

We take as a base theory the standard induction-free fragment of arithmetic Q (cf. [1]), which contains axioms for the non-logical symbols 0, S, $+$, $\cdot$ and $\leqslant$. It is well known that we can construct a hierarchy of strong fragments of arithmetic by adding induction axioms, or alternatively choice or collection axioms, to our base theory. The axiom *scheme of induction* is specified as

$$\mathrm{IND}\;:\;\; A_0 \wedge \forall i < m\,(A_i \to A_{i+1}) \to A_m.$$

The *scheme of finite choice* is specified as

$$\mathrm{FAC}\;:\;\; \forall i \leqslant m\, \exists x A_i(x) \to \exists s \forall i \leqslant m\, A_i(s_i).$$

The *scheme of bounded collection* is specified as

$$\mathrm{BC}\;:\;\; \forall i \leqslant m\, \exists x A_i(x) \to \exists k \forall i \leqslant m\, \exists x \leqslant k\, A_i(x)$$

As usual, if S is one of our schemata, Σ_n-S (Π_n-S) denotes S restricted to Σ_n (Π_n) formulas.

▸ **Definition 1** (Fragments of arithmetic)**.** In this paper we will consider the following fragments of classical arithmetic.

1. The weak theory PA_0 consists of the base theory Q plus induction restricted to quantifier-free formulas.
2. The theories $\mathrm{I}\Sigma_n$, $\mathrm{F}\Sigma_n$, $\mathrm{B}\Sigma_n$ consist of PA_0 plus the Σ_n-IND, Σ_n-FAC, Σ_n-BC schema respectively. The theories $\mathrm{I}\Pi_n$, $\mathrm{F}\Pi_n$ and $\mathrm{B}\Pi_n$ are defined similarly.
3. Full Peano arithmetic PA consists of PA_0 plus induction for all formulas (or, equivalently as we will discuss, finite choice or bounded collection for all formulas).

It is easy to show (cf. [13, 14]) that PA_0 proves the equivalences

$$\begin{aligned}
\Sigma_n\text{-IND} &\quad\leftrightarrow\quad \Pi_n\text{-IND},\\
\Sigma_{n+1}\text{-FAC} &\quad\leftrightarrow\quad \Pi_n\text{-FAC},\ \text{and}\\
\Sigma_{n+1}\text{-BC} &\quad\leftrightarrow\quad \Pi_n\text{-BC}
\end{aligned}$$

for all n. Hence, the fragments $\mathrm{I}\Sigma_n$ and $\mathrm{I}\Pi_n$ are equivalent, similarly $\mathrm{F}\Sigma_{n+1} = \mathrm{F}\Pi_n$ and $\mathrm{B}\Sigma_{n+1} = \mathrm{B}\Pi_n$.

It has also been shown by Sieg [16] that $\mathrm{F}\Pi_n$ and $\mathrm{B}\Pi_n$ are equivalent. Although the fragment based on bounded collection $\mathrm{B}\Pi_n$ is more widely used, we will see that the dialectica interpretation of finite choice is slightly more natural and direct than that of bounded collection.

The precise relationship between the fragments of arithmetic based on induction and those based on choice and collection principles was established by Parsons [14] and Paris and Kirby [13].

▸ **Theorem 2** ([13, 14])**.** *Let $T \subseteq S$ mean that every theorem of T is a theorem of S. Then*

1. $\mathrm{B}\Pi_n \subseteq \mathrm{I}\Sigma_{n+1}$ *but* $\mathrm{I}\Sigma_{n+1} \nsubseteq \mathrm{B}\Pi_n$
2. $\mathrm{I}\Sigma_n \subseteq \mathrm{B}\Pi_n$ *but* $\mathrm{B}\Pi_n \nsubseteq \mathrm{I}\Sigma_n$.

Proof. These results are collected together in [13]. We remark that $\mathrm{I}\Sigma_{n+1} \nsubseteq \mathrm{B}\Pi_n$ was discovered independently by Lessan [12]. ◂

2.1 Fragments of Gödel's system $\mathbf{T}$

In this section we recall some basic facts about Gödel's dialectica interpretation. As we mentioned in the introduction, Gödel showed that intuitionistic arithmetic can be interpreted in the quantifier-free system $\mathbf{T}$. Combining the dialectica interpretation with the usual negative translation allows us to reduce full classical arithmetic to $\mathbf{T}$. For that we shall normally take Kuroda's negative translation $A \mapsto A^N \equiv \neg\neg A^*$, which places double negations after universal quantifiers and in front of the whole formula [11].

We are interested in fragments of system $\mathbf{T}$ that correspond, under the dialectica interpretation, to the fragments of arithmetic discussed above. We take as a base theory the fragment $\mathbf{T}_0$ in which recursion is restricted to type 0.

▸ **Definition 3** (Fragment $\mathbf{T}_0$)**.** We work in a many-sorted language in which the set of types is defined inductively, containing the type $\mathbb{N}$ of natural numbers, function types $X \to Y$ (also written as Y^X), product types $X \times Y$ and types X^* representing finite sequences of elements of type X. As usual, the degree of each type is defined inductively as

$$\begin{aligned}
\deg(\mathbb{N}) &:= 0\\
\deg(X \to Y) &:= \max\{\deg(X)+1, \deg(Y)\}\\
\deg(X \times Y) &:= \max\{\deg(X), \deg(Y)\}\\
\deg(X^*) &:= \deg(X).
\end{aligned}$$

The set of terms of $\mathbf{T}_0$ are those of the simply typed λ-calculus with finite products and function types, plus constants for all functions definable using primitive recursion of type 0. The axioms of $\mathbf{T}_0$ consist of:

1. standard axioms of classical propositional logic, axioms for (fully extensional) equality, substitution and induction,
2. defining axioms for each constant symbol.

Notation. We will denote the operation of concatenating a finite sequence $s\colon X^*$ with an infinite sequence $\alpha\colon X^{\mathbb{N}}$ as $s * \alpha\colon X^{\mathbb{N}}$. We will use the same notation also when concatenating two finite sequences, or appending an element to a sequence. For $q\colon X^{\mathbb{N}} \to R$ and $s\colon X^*$ we write $q_s\colon X^{\mathbb{N}} \to R$ for the function $q_s(\alpha) = q(s * \alpha)$.

We obtain extensions of $\mathbf{T}_0$ by adding constant symbols for higher type recursion together with their defining axioms. For any type X the recursor R^X of type $X \to (\mathbb{N} \to X \to X) \to \mathbb{N} \to X$ has defining axioms:

$$\begin{aligned}
\mathsf{R}^X_0(y,z) &\overset{X}{=} y\\
\mathsf{R}^X_{n+1}(y,z) &\overset{X}{=} z(n, \mathsf{R}^X_n(y,z))
\end{aligned}$$

where $y\colon X$ and $z\colon \mathbb{N} \to X \to X$.

▸ **Definition 4** (Fragments of system $\mathbf{T}$)**.** We consider the following well-known extensions of $\mathbf{T}_0$.

1. The theory $\mathbf{T}_n$ consists of $\mathbf{T}_0$ plus recursors R^X and their defining axioms for all types X with $\deg(X) \leqslant n$.
2. System $\mathbf{T}$ consists of $\mathbf{T}_0$ plus recursors of all finite types.

Gödel's dialectica interpretation interprets each formula A of Heyting arithmetic as a formula A^D of the form $\exists x \forall y A_D(x,y)$, where A_D is a quantifier-free formula in the language of $\mathbf{T}$ and x, y are tuples of potentially higher type. For details of the translation, the reader is referred to [1].

▸ **Theorem 5** ([7])**.** *Gödel's key results are the following:*

1. *If Heyting arithmetic proves the formula A, then there is a sequence of terms t in system $\mathbf{T}$ such that $\mathbf{T}$ proves $A_D(t,y)$.*
2. *If Peano arithmetic proves the formula A, then there is a sequence of terms t such that $\mathbf{T}$ proves $(A^N)_D(t,y)$, where A^N denotes the negative translation of A.*

The recursors in $\mathbf{T}$ are essentially only required to interpret the *non-logical* axioms of arithmetic, and form a natural functional analogue of induction. There is in fact a precise correspondence under the dialectica interpretation between the arithmetic hierarchy $\mathrm{I}\Sigma_n$ and the functional hierarchy $\mathbf{T}_n$, which is a consequence of the following lemma.

▸ **Lemma 6.** *Given an arbitrary formula A, suppose that $(A^N)^D = \exists x \forall y (A^N)_D(x,y)$. Then for $n > 1$:*

1. *if A is a Π^0_n formula then the tuple x contains variables of degree at most $n-1$ and the tuple y contains variables of degree at most $n-2$;*
2. *if A is a Σ^0_n formula then the tuple x contains variables of degree at most n and the tuple y contains variables of degree at most $n-1$.*

Proof. Simple induction on n. ◂

▸ **Theorem 7** ([15])**.** *The functional interpretation of* Π_n-IND *only requires primitive recursion of level $n-1$. Therefore the fragment of arithmetic $\mathrm{I}\Pi_n$ (or equivalently $\mathrm{I}\Sigma_n$) is interpreted in $\mathbf{T}_{n-1}$, in the sense that if $\mathrm{I}\Pi_n$ proves A then there is a sequence of terms t in $\mathbf{T}_{n-1}$ such that $\mathbf{T}_{n-1}$ proves $(A^N)_D(t,y)$.*

Proof. Follows from Lemma 6. See [15] for details. ◂

3 The Product of Selection Functions

In [5], a *selection function* is defined to be any function of type $(X \to R) \to X$. The intuition behind the name is that we view functions $X \to R$ as predicates over the type X (where R is interpreted as a set of truth values), and the selection function as a choice procedure that for each predicate selects some element of X. Selection functions are closely related to the notion of *generalised quantifiers*, functionals of type $(X \to R) \to R$, in the sense that every selection function ε is associated with a quantifier $\overline{\varepsilon}p := p(\varepsilon p)$.

Notation. We abbreviate the types $(X \to R) \to X$ by $J_R X$, and the types $(X \to R) \to R$ by $K_R X$.

▸ **Example 8.** 1. By the law of excluded middle, for any non-empty type X we have $\forall p \exists y^X (\exists x^X p(x) \Rightarrow p(y))$. Hence, by the axiom of choice there exists a selection function $\varepsilon \colon J_{\mathbb{B}} X$ that satisfies $\exists x\, p(x) \Leftrightarrow p(\varepsilon p)$ for any logical predicate $p \colon X \to \mathbb{B}$ (this is similar to Hilbert's ε operator of the ε-calculus). Similarly, there is also a selection function δ such that $\forall x\, p(x) \Leftrightarrow p(\delta p)$ for all p. These selection functions are associated, respectively, with the usual logical quantifiers $\exists, \forall \colon K_{\mathbb{B}} X$.

2. By the extreme value theorem there exists a selection function $\mathrm{argsup}\colon J_{\mathbb{R}}[0,1]$ that for any continuous function $f\colon [0,1] \to \mathbb{R}$ returns a point at which f attains it supremum i.e. $\sup(f) = f(\mathrm{argsup} f)$. The selection function $\mathrm{arginf}\colon J_{\mathbb{R}}[0,1]$ is defined similarly. These selection functions are associated with the quantifiers $\sup, \inf\colon K_{\mathbb{R}}[0,1]$.

The theory of generalised quantifiers and selection functions is explored in detail in [3, 5, 6], where in particular, a product operation on selection functions is defined. The main achievement of these papers has been to demonstrate that this product of selection functions is an extremely versatile construction that appears naturally in several different areas of mathematics and computer science, such as fixed point theory (Bekič's lemma), game theory (backward induction), algorithms (backtracking), and proof theory (bar recursion).

For the rest of this section we shall define this product of selection functions, and explain how it has an intuitive meaning in terms of optimal plays in sequential games. In Section 4 we then show how the number theoretic principle of finite choice is naturally (dialectica) interpreted by this product.

▸ **Definition 9** (Binary product of selection functions, [5]). Given selection functions $\delta\colon J_R X$ and $\Delta\colon J_R X^{\mathbb{N}}$ and a functional $q\colon X^{\mathbb{N}} \to R$, let

$$\begin{aligned} A(x^X) &\overset{X^{\mathbb{N}}}{:=} \Delta(\lambda \alpha . q_x(\alpha)), \\ a &\overset{X}{:=} \delta(\lambda x . q_x(A(x))), \end{aligned}$$

where $q_x(\alpha)$ abbreviates $q(x * \alpha)$. Then we define the binary product of the selection functions δ and Δ, denoted $\delta \otimes \Delta\colon J_R X^{\mathbb{N}}$, by

$$(\delta \otimes \Delta)(q) := a * A(a).$$

As described in [5], one can iterate the binary product above on a given sequence of selection functions. In this paper we will only consider the finite iteration of the binary product.

▸ **Definition 10** (Finite product of selection functions). We define the *finite product of selection functions* for types (X, R), denoted $\mathsf{P}_i^{X,R}$, by the recursion schema

$$\mathsf{P}_i^{X,R}(\varepsilon)(m) \overset{J_R X^{\mathbb{N}}}{=} \begin{cases} \mathbf{0}^{J_R X^{\mathbb{N}}} & \text{if } i > m \\ \varepsilon_i \otimes \mathsf{P}_{i+1}^{X,R}(\varepsilon)(m) & \text{if } i \leqslant m \end{cases}$$

where $m \in \mathbb{N}$, $\mathbf{0}$ is the constant 0 functional of appropriate type, and ε_i are selection functions of type $J_R X$. Expanding the definition of the binary product (Definition 9) this is equivalent to the schema

$$\mathsf{P}_i^{X,R}(\varepsilon)(m)(q) \overset{X^{\mathbb{N}}}{=} \begin{cases} \mathbf{0}^{X^{\mathbb{N}}} & \text{if } i > m \\ a * \mathsf{P}_{i+1}^{X,R}(\varepsilon)(m)(q_a) & \text{if } i \leqslant m \end{cases}$$

where $a := \varepsilon_i(\lambda x . q_x(\mathsf{P}_{i+1}^{X,R}(\varepsilon)(m)(q_x)))$.

As opposed to in [5] here the finite product is taken over an infinite stream of selection functions. In what follows, where only $\varepsilon_0, \ldots, \varepsilon_m$ are specified it is implicit that the finite product $\mathsf{P}_i^{X,R}(\varepsilon)(m)$ is taken over a canonical extension of this finite sequence.

As an alternative to adding the recursors R to $\mathbf{T}_0$, as in Definition 4, we shall also consider extending $\mathbf{T}_0$ with the finite product operator P instead.

▸ **Definition 11.** 1. The theory $\mathbf{P}_n$ consists of $\mathbf{T}_0$ plus a symbol for the finite product of selection functions $\mathsf{P}^{X,R}$ and its defining axiom for all types X with $\deg(X) \leqslant n$.
2. $\mathbf{T}_0 + \mathsf{P}$ consists of $\mathbf{T}_0$ plus the finite product $\mathsf{P}^{X,R}$ for all finite types.

▸ Remark. The complexity of the type R has no effect on the recursive strength of $\mathsf{P}^{X,R}$, as one can show that $\mathsf{P}^{X,R}$, for arbitrary type R, is definable over $\mathbf{T}_0$ from $\mathsf{P}^{X,X^{\mathbb{N}}}$. Formally, given $\varepsilon_i \colon J_R X$ and $q \colon X^{\mathbb{N}} \to R$ define a new selection function $\varepsilon^q \colon J_{X^{\mathbb{N}}} X$ as

$$\varepsilon_i^q(P^{X \to X^{\mathbb{N}}}) \stackrel{X}{=} \varepsilon(\lambda x^X . q(P(x))).$$

We have that $\mathsf{P}(\varepsilon)(m)(q) = \mathsf{P}(\varepsilon^q)(m)(\mathrm{id})$, where $\mathrm{id} \colon X^{\mathbb{N}} \to X^{\mathbb{N}}$ is the identity functional.

The main result of this article is that $\mathbf{T}_0 + \mathsf{P}$ is equivalent to $\mathbf{T}_0 + \mathsf{R}$ (and hence to Gödel's system $\mathbf{T}$), and that more specifically there is a direct correspondence between the restricted fragments of both systems. But first we explain how selection functions and their finite product are fundamental in the study of sequential games.

3.1 Finite sequential games

One of the most interesting aspects of the product of selection functions is that it computes *optimal strategies* for a general class of sequential games. This concrete setting offers the most insight into how the product works, so we explain it briefly here.

▸ **Definition 12** (Finite sequential games)**.** An m-round sequential game is defined by a tuple (R, X, ε, q) where R and X are arbitrary types.

- X is the set of possible moves for any round. A play is a sequence $\alpha \colon X^m$.
- R is the set of possible outcomes.
- $q \colon X^m \to R$ is the outcome function that maps a play to its outcome.
- $\varepsilon_i \colon J_R X$ is the selection function for round $i \leqslant m$.

These kind of games have been defined in full generality in [3, 6], where in particular the set of moves may vary from one round to the other. Moreover, the games defined there allows for arbitrary *quantifiers* to describe the goal of each round. When these quantifiers have associated selection functions an optimal strategy for the game can be computed. Here, for simplicity, we will assume the selection functions are explicitly given in the definition of the game.

Also in [3, 6], the notions of optimal strategy and optimal play are defined. The intuition is as follows. We think of the selection functions ε_i as specifying at round i what the optimal move at that round would be if we knew the final outcome corresponding to each of the candidate moves, i.e. $p \colon X \to R$. The selection function takes this mapping of moves to outcomes and tells us what the "best" move would be in that particular case $\varepsilon_i(p) \colon X$. Now, a play is considered optimal if at all rounds the best move has indeed been played. That is to say that, there are functions p_i which compute the real outcome from the move being played, i.e. $p_i(\alpha(i)) = q\alpha$, and that $\alpha(i)$ is exactly what the selection function at round i would choose, i.e. $\alpha(i) = \varepsilon_i(p_i)$.

The main theorem of [6] is that the product of the given selection functions for each round, when applied to the outcome function, computes an optimal play α in the game.

▸ **Theorem 13** ([6])**.** *Given a sequential game as above, let*

$$\alpha := \mathsf{P}_0(\varepsilon)(m)(q).$$

Then, α is an optimal play in the sense that setting

$$p_i := \lambda x.q_{[\alpha](i)*x}\left(\mathsf{P}_{i+1}(\varepsilon)(m)(q_{[\alpha](i)*x})\right),$$

where $[\alpha](i)$ is the finite initial segment of α of length i, we have

$$\begin{aligned}\alpha(i) &\overset{X}{=} \varepsilon_i(p_i) \\ p_i(\alpha(i)) &\overset{R}{=} q\alpha\end{aligned} \tag{1}$$

for all $i \leqslant m$.

In more general terms, the equations (1) characterise the product as an operation that generates a state of equilibrium from a finite sequence of selection functions. An optimal strategy is one instance of such an equilibrium. The significance of the product of selection functions lies in the fact that these governing equations appear repeatedly in a variety of different contexts.

4 Interpreting the Principle of Finite Choice

In this section we show how the finite product of selection functions allows for a more direct interpretation of (the classical) finite choice and bounded collection principles. As observed by Spector [17], the interpretation of the negative translation of choice follows intuitionistically from choice itself given the double negation shift

$$\forall i \neg\neg A(i) \to \neg\neg\forall i A(i).$$

The same applies to finite choice, where the negative translation of finite choice follows from finite choice plus the finite double negation shift

$$\forall m(\forall i{\leqslant}m \neg\neg A(i) \to \neg\neg\forall i{\leqslant}m A(i)).$$

Contrary to the double negation shift, the *finite* double negation shift is provable in Heyting arithmetic, by induction on m. The proof, however, is rather intricate, and when interpreted (via the dialectica) leads to witnesses based on the recursor R which are difficult to grasp computationally. This is in stark contrast with the proof of the following theorem:

▸ **Theorem 14.** *The finite product of selection functions interprets (via the dialectica interpretation) the finite double negation shift.*

Proof. Assume $A(i)$ has dialectica interpretation $\exists x \forall y A_i(x, y)$. The (partial) dialectica interpretation of the finite double negation shift is equivalent to

$$\forall m(\exists \varepsilon \forall p \forall i{\leqslant}m A_i(\varepsilon_i p, p(\varepsilon_i p)) \to \forall q \exists \alpha \forall i{\leqslant}m A_i(\alpha(i), q\alpha)).$$

Given m, ε and q then taking α, p_i as in Theorem 13 we clearly have

$$A_i(\varepsilon_i p_i, p_i(\varepsilon_i p_i)) \to A_i(\alpha(i), q\alpha)$$

for all $i \leqslant m$, so the α, p_i computed directly via the product of selection function witness the interpretation of the double negation shift. ◂

It should be observed that when using modified realizability (instead of the dialectica interpretation) it is the so-called J-shift which is directly interpreted by the product of selection functions (cf. [4]). It is, therefore, rather interesting that when applying the dialectica interpretation the same product of selection functions allows for a direct interpretation of the double negation shift instead.

A closer look at the dialectica interpretation of the double negation shift sheds some light on why an operation that computes optimal plays in sequential games crops up in proof theory in this manner. The selection functions ε_i above act as realisers for the premise of the double negation shift, and as such can be seen as a collection of strategies $(\varepsilon_i)_{i<m}$ that for each i refute any counterexample functions p (in the sense of Kreisel [9, 10]) attempting to disprove the predicate A_i.

The functional interpretation of the double negation shift calls for a procedure that takes this collection of 'point-wise' strategies and produces a co-operative strategy in which the ε_i work together to refute a *global* counterexample function q attempting to disprove the predicate $\forall i < m\, A_i$. Such a procedure is provided naturally by the product of selection functions.

▸ **Corollary 15.** *The finite product of selection functions interprets the principle of finite choice.*

Proof. The negative translation of finite choice, assuming that A_i is Π^0_n, is equivalent to

$$\forall i \leqslant m \neg\neg\exists x A_i^*(x) \to \neg\neg\exists\alpha\forall i \leqslant m A_i^*(\alpha_i),$$

where A^* is obtained by placing double negations after each universal quantifier in A. This follows directly from the double negation shift applied to the formula $\exists x A_i^*(x)$, and its dialectica interpretation is precisely that of this double negation shift. Hence, the product of selection functions realises the dialectica interpretation of (the negative translation of) finite choice. ◂

Analogous to Theorem 7, we can extend Corollary 15 to fragments of arithmetic based on choice.

▸ **Theorem 16.** *The fragment of arithmetic* $\mathrm{F}\Pi_n$ *is interpreted in* $\mathbf{P}_{n-1}$.

Proof. This is clear for $n = 1$. For $n > 1$ by Lemma 6 if A_i is a Π^0_n formula the functional interpretation of $\exists x A_i^*(x)$ is of the form $\exists x, \tilde{x}\forall y (A_i^*)_D(x, \tilde{x}, y)$ where the variables of the tuple $\langle x^0, \tilde{x}\rangle$ have degree at most $n-1$ (note that $A^D \leftrightarrow A^*$ for Π^0_n formulas). Hence, by inspecting the proof of Theorem 14 we see that an instance of Π_n-FAC is interpreted in $\mathbf{P}_{n-1}$. ◂

Finally we remark that bounded collection and its consequences, such as the infinite pigeonhole principle (cf. [8], p. 173), are naturally interpreted by the product of selection functions in a similar manner, given that finite choice straightforwardly implies bounded collection. The realiser for the negative translation of bounded collection, namely

$$\forall i \leqslant m \neg\neg\exists x A_i^*(x) \to \neg\neg\exists k \forall i \leqslant m \neg\neg\exists x \leqslant k\, A_i^*(x)$$

based on the product of selection functions is obtained from that for finite choice by essentially applying the maximum operator to the first m elements of the sequence α (see [6] for more details).

5 The Recursor and the Product of Selection Functions

Although the recursor R directly interprets the induction schema, we have seen in Section 4 that it is the finite product of selection functions which directly interprets finite choice and bounded collection. As discussed in Section 2, Parson showed that the hierarchy of bounded collection axioms is strictly interleaving on the hierarchy of induction axioms. Therefore, one might conjecture that the hierarchy of finite products of selection functions would be also strictly interleaved in the hierarchy of Gödel's primitive recursors. In this section we show that this is not the case, and in fact the recursor of type level $n+1$ is primitive recursively equivalent to the finite product of selection functions of type level n.

▸ **Definition 17.** It will be convenient to make use of the functional $\mathsf{B}(\varepsilon)(m)(q) \colon X^* \to X^{\mathbb{N}}$ defined as

$$\mathsf{B}(\varepsilon)(m)(q)(s) := \mathsf{P}_{|s|}(\varepsilon)(m)(q_s).$$

By using the expanded definition of P (cf. Definition 10), it is easy to see that B, for fixed ε, m and q, satisfies the recursion schema

$$\mathsf{B}(s) := \begin{cases} \lambda n.\mathbf{0}^{\mathbf{X}} & \text{if } |s| > m \\ a_s * \mathsf{B}(s * a_s) & \text{if } |s| \leqslant m \end{cases}$$

where $a_s = \varepsilon_{|s|}(\lambda x.q(s * x * \mathsf{B}(s * x)))$. The intuitive reading of B is an operation that takes a partial play s and returns $s * \alpha$ where α is a continuation of s that is optimal up to round m. In particular, an easy induction argument proves that for all $i \leqslant m$

$$\mathsf{P}_0(\varepsilon)(m)(q)_i \equiv \mathsf{B}(\langle\,\rangle)_i = \mathsf{B}(x_0, \ldots, x_{i-1})_0 \tag{2}$$

where $x_j := \mathsf{P}_0(\varepsilon)(m)(q)_j$ for $j < i$. In what follows the parameters of B will always be clear from the context, so we will omit them for simplicity.

Notation. Given two fragments of $\mathbf{T}$, say $\mathbf{T}'$ and $\mathbf{T}''$, we write $\mathbf{T}' \Rightarrow \mathbf{T}''$ if all functionals definable in $\mathbf{T}''$ can be already defined in $\mathbf{T}'$.

▸ **Theorem 18.** *$\mathsf{P}^{X,R}$ is definable in $\mathbf{T}_0 + \mathsf{R}^{X^* \to X^{\mathbb{N}}}$, so in particular $\mathbf{T}_{n+1} \Rightarrow \mathbf{P}_n$.*

Proof. Looking at the definition of the finite product, namely

$$\mathsf{P}^{X,R}_i(\varepsilon)(m)(q) \overset{X^{\mathbb{N}}}{=} \begin{cases} \lambda n.\mathbf{0}^X & \text{if } i > m \\ a * \mathsf{P}^{X,R}_{i+1}(\varepsilon)(m)(q_a) & \text{if } i \leqslant m \end{cases}$$

where $a := \varepsilon_i(\lambda x.q_x(\mathsf{P}^{X,R}_{i+1}(\varepsilon)(m)(q_x)))$, it is clear that the schema is just a standard recursion of type $X^* \to X^{\mathbb{N}}$ in which the quantity $m + 1 - i$ decreases along the recursion until it reaches 0. Formally, define the functionals $y^{\varepsilon,q,m} \colon X^* \to X^{\mathbb{N}}$ and $z^{\varepsilon,q,m} \colon \mathbb{N} \times (X^* \to X^{\mathbb{N}}) \to (X^* \to X^{\mathbb{N}})$ parametrised by ε, q and m as

$$y^{\varepsilon,q,m}(s) := \lambda n.\mathbf{0}^X$$
$$z^{\varepsilon,q,m}(i, F^{X^* \to X^{\mathbb{N}}})(s) := a_s * F(s * a_s)$$

where $a_s := \varepsilon_{m \dot{-} i}(\lambda x.q_{s*x}(F(s * x)))$ and $m \dot{-} i$ denotes truncated subtraction. We claim that for all s

$$\mathsf{B}(s) = \mathsf{R}^{X^* \to X^{\mathbb{N}}}_{m+1 \dot{-} |s|}(y^{\varepsilon,q,m}, z^{\varepsilon,q,m})(s),$$

where B is as in Definition 17. In particular, it would follow that

$$\mathsf{P}_0^{X,R}(\varepsilon)(m)(q) = \mathsf{B}(\langle\,\rangle) = \mathsf{R}_m^{X^*\to X^{\mathbb{N}}}(y^{\varepsilon,q,m}, z^{\varepsilon,q,m})(\langle\,\rangle).$$

The claim is proved by induction on $m + 1 \dot{-} |s|$. For $|s| \geqslant m + 1$ we have

$$\mathsf{R}_{m+1\dot{-}|s|}(s) = \mathsf{R}_0(s) = y^{\varepsilon,q,m}(s) = \lambda n.\mathbf{0}^X = \mathsf{B}(s).$$

Now for $|s| = i < m + 1$ we have

$$\mathsf{R}_{m+1\dot{-}|s|}(s) = \mathsf{R}_{(m\dot{-}i)+1}(s) = z^{\varepsilon,q,m}(m \dot{-} i, \mathsf{R}_{m\dot{-}i})(s) = a_s * \mathsf{R}_{m\dot{-}i}(s * a_s)$$

where $a_s := \varepsilon_i(\lambda x.q_{s*x}(\mathsf{R}_{m\dot{-}i}(s * x)))$. But by induction hypothesis we have that

$$\mathsf{R}_{m\dot{-}i}(s * x) = \mathsf{B}(s * x)$$

for all $|s| = i$. Therefore $\mathsf{R}_{m+1\dot{-}i}(s) = \mathsf{B}(s)$. This completes the induction. ◂

We will show that the the types above are optimal, in the sense that we also have the converse $\mathbf{P}_n \Rightarrow \mathbf{T}_{n+1}$. But before showing that, lets us first show how one easily has $\mathbf{P}_n \Rightarrow \mathbf{T}_n$. The stronger result will require a more involved argument.

▸ **Theorem 19.** *R^X is definable in $\mathbf{T}_0 + \mathsf{P}^{X,X^{\mathbb{N}}}$, so in particular $\mathbf{P}_n \Rightarrow \mathbf{T}_n$.*

Proof. Given arbitrary functionals $y \colon X$ and $z \colon \mathbb{N} \to (X \to X)$, define selection functions $\varepsilon^{y,z} \colon \mathbb{N} \to J_{X^{\mathbb{N}}} X$ parametrised by y and z as

$$\varepsilon_i^{y,z}(p^{X\to X^{\mathbb{N}}}) \overset{X}{:=} \begin{cases} y & \text{if } i = 0 \\ z(i-1, p(\mathbf{0}^X)_{i-1}) & \text{if } i > 0. \end{cases}$$

Clearly $\lambda y, z.\varepsilon^{y,z}$ can be constructed in $\mathbf{T}_0$. We prove by induction on m that

$$\mathsf{R}_m(y, z) = (\mathsf{B}(\varepsilon^{y,z})(m)(\mathrm{id})(\langle\,\rangle))_m,$$

where $\mathrm{id} \colon X^{\mathbb{N}} \to X^{\mathbb{N}}$ is the identity λ-term. We shall actually think of m as fixed and show

$$\mathsf{R}_i(y, z) = (\mathsf{B}(\varepsilon^{y,z})(m)(\mathrm{id})(\langle\,\rangle))_i,$$

for $i \leqslant m$, by induction on i. When $i = 0$ we have (abbreviating $\mathsf{B}(s) \equiv \mathsf{B}(\varepsilon^{y,z})(m)(\mathrm{id})(s)$)

$$\mathsf{B}(\langle\,\rangle)_0 = \varepsilon_0^{y,z}(\lambda x \ldots) = y = \mathsf{R}_0(y, z).$$

Assuming that $x_j = \mathsf{R}_j(y, z) = \mathsf{B}(\langle\,\rangle)_j$, for $j < i$. We have

$$\begin{aligned} \mathsf{B}(\langle\,\rangle)_i &\overset{(2)}{=} \mathsf{B}(x_0, \ldots, x_{i-1})_0 \\ &= \varepsilon_i(\lambda x.\langle x_0, \ldots, x_{i-1}, x\rangle * \mathsf{B}(x_0 \ldots, x_{i-1}, x)) = z(i-1, x_{i-1}) = \mathsf{R}_i(y, z). \end{aligned}$$

◂

Already, we obtain the following key result.

▸ **Corollary 20.** *Gödel's system $\mathbf{T}$, i.e. $\mathbf{T}_0 + \mathsf{R}$, can be equivalently defined as $\mathbf{T}_0 + \mathsf{P}$.*

The intuition behind the proof of Theorem 19 is that the type $R = X^{\mathbb{N}}$ represents a register of elements of type X on which we perform a computation. The selection function ε_i sets the entry x_i at position i on the register to be $z(i-1, x_{i-1})$, where x_{i-1} is the entry at position $i-1$. The product $\mathsf{P}_0(m)(\varepsilon_i)(\mathrm{id})$ carries out the first m steps of this computation, returning $\mathsf{R}_m(y,z)$ at position m.

Of course the finite product of selection functions is able to perform a variety of computations on a register $X^{\mathbb{N}}$ in this manner - in which the ε_i determine the entry in the ith position of the register. However, in general the selection functions are capable of making this decision based not only on the previous entries but depending on the effect that potential choices have on *subsequent* entries. This suggests that the finite product of type X is a more powerful construction than the recursor of type X.

▸ **Theorem 21.** $\mathsf{R}^{X\to X}$ *is definable in* $\mathbf{T}_0 + \mathsf{P}^{X,X^{\mathbb{N}}}$.

Proof. Given arbitrary functionals $y \colon X^X$ and $z \colon \mathbb{N} \to (X^X \to X^X)$, define selection functions $\varepsilon_i^{y,z,n,a} \colon J_{X^{\mathbb{N}}}X$ parametrised by $y \colon X^X$ and $z \colon \mathbb{N} \to (X^X \to X^X)$ and $n \colon \mathbb{N}$ and $a \colon X$ as

$$\varepsilon_i^{y,z,n,a}(p^{X\to X^{\mathbb{N}}}) \overset{X}{:=} \begin{cases} z(n-1, \lambda x.p(x)_1)(a) & \text{if } i = 0 \\ z(n-i-1, \lambda x.p(x)_{i+1})(p(\mathbf{0}^X)_{i-1}) & \text{if } 0 < i < n \\ y(p(\mathbf{0}^X)_{i-1}) & \text{if } i = n \\ \mathbf{0}^X & \text{otherwise} \end{cases}$$

for $n > 0$, and

$$\varepsilon_i^{y,z,0,a}(p^{X\to X^{\mathbb{N}}}) \overset{X}{:=} \begin{cases} y(p(a)_0) & \text{if } i = 0 \\ \mathbf{0}^X & \text{otherwise.} \end{cases}$$

The functional $\lambda y, z, n, a.\varepsilon^{y,z,n,a}$ can be constructed in $\mathbf{T}_0$ since we only make use of combinatory completeness and definition by cases (which is primitive recursive of level 0 and allowed in $\mathbf{T}_0$). We prove that R can be defined as

$$\mathsf{R}_n(y,z) = \lambda a.(\mathsf{P}_0(\varepsilon^{y,z,n,a})(n)(\mathrm{id}))_0.$$

This is trivial for $n = 0$, so in the following we assume that $n > 0$. Once again, for convenience we set $\mathsf{B}(s) := \mathsf{P}_{|s|}(\varepsilon^{y,z,n,a})(n)(\mathrm{id}_s)$. First, we claim that

$$(\mathsf{B}(x_0, \dots, x_{i-1}))_0 = \mathsf{R}_{n-i}(y,z)(x_{i-1})$$

for all $0 < i \leqslant n$. We proceed by induction on $n - i$. For $i = n$ we have

$$(\mathsf{B}(x_0, \dots, x_{n-1}))_0 = \varepsilon_n(\lambda x.\langle x_0, \dots, x_{n-1}, x\rangle * \lambda n.\mathbf{0}^X) = y(x_{n-1}) = \mathsf{R}_0(y,z)(x_{n-1}).$$

For $0 < i < n$,

$$\begin{aligned} (\mathsf{B}(x_0, \dots, x_{i-1}))_0 &= \varepsilon_i(\lambda x.\langle x_0, \dots, x_{i-1}, x\rangle * \mathsf{B}(x_0, \dots, x_{i-1}, x)) \\ &= z(n-i-1, \lambda x.\mathsf{B}(x_0, \dots, x_{i-1}, x)_0)(x_{i-1}) \\ &= z(n-i-1, \lambda x.\mathsf{R}_{n-i-1}(y,z)(x))(x_{i-1}) \\ &= \mathsf{R}_{n-i}(y,z)(x_{i-1}), \end{aligned}$$

assuming, by hypothesis, that $\mathsf{B}(x_0, \ldots, x_{i-1}, x) = \mathsf{R}_{n-(i+1)}(y, z)(x)$. This proves the claim, and the theorem follows directly:

$$\begin{aligned} \mathsf{P}_0(\varepsilon)(n)(\mathrm{id})_0 &= \mathsf{B}(\langle\rangle)_0 \\ &= \varepsilon_0(\lambda x.x * \mathsf{B}(x)) \\ &= z(n-1, \lambda x.\mathsf{B}(x)_0)(a) \\ &= z(n-1, \lambda x.\mathsf{R}_{n-1}(y, z)(x))(a) = \mathsf{R}_n(y, z)(a). \end{aligned}$$

◂

▸ **Corollary 22.** $\mathbf{P}_n \Leftrightarrow \mathbf{T}_{n+1}$.

Proof. One direction is given by Theorem 18. The other follows from Theorem 21: It can be shown that any type of level n is isomorphic to the pure type of that level, and consequently any two recursors of the same type level are inter-definable over $\mathbf{T}_0$. If $\deg(X) = n$ then $\deg(X \to X) = n + 1$, therefore by Theorem 21, $\mathbf{P}_n \Rightarrow \mathbf{T}_{n+1}$. ◂

Theorem 22 tells us that, in particular, the product of selection functions over the type X is strictly stronger than primitive recursion of type X. For the case $X = \mathbb{N}$ we can illustrate this directly by constructing the Ackermann function in $\mathbf{P}_0$.

▸ **Example 23** (Ackermann function)**.** Define the selection functions $\varepsilon_i^{n,a} \colon J_{\mathbb{N}^\mathbb{N}}\mathbb{N}$ parametrised by natural numbers n and a as

$$\varepsilon_i^{n,a}(p^{\mathbb{N}\to\mathbb{N}^\mathbb{N}}) \overset{\mathbb{N}}{:=} \begin{cases} (\lambda x.p(x)_1)^{(a+1)}(1) & \text{if } i = 0 \\ (\lambda x.p(x)_{i+1})^{(p(\mathbf{0}^X)_{i-1}+1)}(1) & \text{if } 0 < i < n \\ p(\mathbf{0}^X)_{i-1} + 1 & \text{if } i = n \\ 0 & \text{otherwise} \end{cases}$$

for $n > 0$, and

$$\varepsilon_i^{0,a}(p^{\mathbb{N}\to\mathbb{N}^\mathbb{N}}) \overset{\mathbb{N}}{:=} \begin{cases} p(a)_0 + 1 & \text{if } i = 0 \\ 0 & \text{otherwise} \end{cases}$$

where $f^{(j)}$ is defined in $\mathbf{T}_0$ by $f^{(0)}(x) = x$ and $f^{(j+1)}(x) = f(f^{(j)}(x))$. We claim that

$$A(n, a) = (\mathsf{P}_0(\varepsilon^{n,a})(n)(\mathrm{id}))_0 = (\mathsf{B}(\langle\rangle))_0$$

where A is the Ackermann function. For instance, $A(3, a)$ is the first entry in an optimal play of a sequential game with selection functions $\varepsilon_i^{3,a}$ and id as the outcome function. We sketch its derivation below.

$$\mathsf{B}(x_0, x_1, x_2) = \varepsilon_3^{3,a}(\lambda x.\langle x_0, x_1, x_2, x, 0, 0, \ldots\rangle), 0, 0, \ldots = x_2 + 1, 0, \ldots.$$

$$\mathsf{B}(x_0, x_1) = \varepsilon_2^{3,a}(\lambda x.\langle x_0, x_1, x, x + 1, 0, \ldots\rangle), \ldots = (\lambda x.x + 1)^{(x_1+1)}(1), \ldots = x_1 + 2, \ldots$$

$$\mathsf{B}(x_0) = \varepsilon_1^{3,a}(\lambda x.\langle x_0, x, x + 2, \ldots\rangle), \ldots = (\lambda x.x + 2)^{(x_0+1)}(1), \ldots = 2x_0 + 3, \ldots$$

$$\mathsf{B}(\langle\rangle) = \varepsilon_0^{3,a}(\lambda x.\langle x, 2x + 3, \ldots\rangle), \ldots = (\lambda x.2x + 3)^{(a+1)}(1), \ldots = 2^{(a+3)} - 3, \ldots.$$

Similarly, the first entry in the 5-round game with selection functions $\varepsilon_i^{4,a}$ is $(2 \uparrow^2 n + 3) - 3$ and so on. Thus the product of selection functions over $\mathbb{N}$ allows a much higher rate of growth than primitive recursion over $\mathbb{N}$.

$$\begin{tikzcd}
\mathrm{PA}_0 \arrow[d] & \mathrm{I}\Sigma_n \arrow[l, dotted] \arrow[d] & \mathrm{F}\Pi_n \arrow[l] \arrow[d] & \mathrm{I}\Sigma_{n+1} \arrow[l] \arrow[d] & \mathrm{PA} \arrow[l, dotted] \arrow[d] \\
\mathbf{T}_{\mathsf{b}} & \mathbf{T}_{n-1} \arrow[l, dotted] & \mathbf{P}_{n-1} \arrow[l, "\text{Thm.19}"'] \arrow[r, Leftrightarrow, "\text{Thms.18,21}"'] & \mathbf{T}_n & \mathbf{T} \arrow[l, dotted]
\end{tikzcd}$$

Figure 1 Fragments of Peano arithmetic and corresponding fragments of system $\mathbf{T}$.

To summarise, we have shown that the finite product of selections is equivalent to the higher type recursor in the sense that $\mathbf{P}_n \Leftrightarrow \mathbf{T}_{n+1}$ over $\mathbf{T}_0$. We conclude by pointing out that these equivalences actually hold over a much weaker base theory, and that in particular the finite product of lowest type defines the primitive recursors of lowest type over a weak fragment of $\mathbf{T}_0$. This follows from the observation that in establishing the equivalences we only make use of a very restricted class of primitive recursive functions, namely concatenation of sequences and definition by cases.

▸ **Definition 24.** **1.** The fragment $\mathbf{T}_{\mathsf{b}} \subset \mathbf{T}_0$ of system $\mathbf{T}$ is defined to be the $\mathbf{T}_0$ but with constants for primitive recursion eliminated, save for definition by cases and concatenation $*$ for all types.

2. The binary product of selection functions can be defined in the language of $\mathbf{T}_{\mathsf{b}}$. Therefore, we define the theory $\tilde{\mathbf{P}}_n$ to be $\mathbf{T}_{\mathsf{b}}$ plus the finite product of selection functions $\mathsf{P}^{X,R}$ for all types X with $\deg(X) \leqslant n$.

It is easy to see that in the proof of Theorem 19 we only need to assume $\mathbf{T}_{\mathsf{b}}$ (rather than $\mathbf{T}_0$). Therefore we obtain the following:

▸ **Theorem 25.** $\tilde{\mathbf{P}}_0 \Rightarrow \mathbf{T}_0$*, so in particular* $\tilde{\mathbf{P}}_n$ *can be identified with* $\mathbf{P}_n$*, for all* n*.*

Hence all uses of $\mathbf{T}_0$ above can be replaced by $\mathbf{T}_{\mathsf{b}}$, and as such the finite product of selection functions $\mathsf{P}^{X,R}$ is *truly* interchangeable with Gödel's primitive recursor R^X (for all types X).

6 Final Remarks

The first two authors have studied in [2] an *unbounded product of selection functions*

$$\mathsf{P}_i^{X,R}(\varepsilon)(\psi)(q) \overset{X^{\mathbb{N}}}{=} \begin{cases} \mathbf{0}^{X^{\mathbb{N}}} & \text{if } i > \psi(q(\mathbf{0})) \\ \left(\varepsilon_i \otimes \mathsf{P}_{i+1}^{X,R}(\varepsilon)(\psi)\right)(q) & \text{if } i \leqslant \psi(q(\mathbf{0})) \end{cases}$$

where the fixed bound m (cf. Definition 10) is replaced by a bounding function ψ on the canonical outcome $q(\mathbf{0})$. When ψ is the constant functional m we obtain the finite product as a particular case of this. They have shown that this unbounded product is primitive recursively equivalent to Spector's bar recursion [17], and hence it is precisely what is needed to (dialectica) interpret full classical analysis. Combining this with our results above show that the iterated product of selection functions P provides a uniform link between Gödel's primitive recursor R and Spector's bar recursion, and hence, a uniform way to interpret arithmetic and analysis.

Figure 1 shows the different subsystems of Peano arithmetic we have considered, and the corresponding fragments of system $\mathbf{T}$ needed for a dialectica interpretation. Parsons has shown that the fragment of PA based on Π_n finite choice is strictly weaker than the fragment based on Σ_{n+1} induction. Nevertheless, we have shown that the product of selection functions of type level $n-1$ (which directly interprets Π_n-FAC) is equivalent (over a weak theory) to the recursor of type n (which directly interprets Σ_{n+1}-IND).

Given that $\mathrm{F}\Pi_n$ is strictly weaker than $\mathrm{I}\Sigma_{n+1}$, we conclude with the question of whether $\mathrm{F}\Pi_n$ can be (dialectica) interpreted in a fragment of $\mathbf{T}$ that is strictly weaker than $\mathbf{P}_n$, or whether it is in fact equivalent to $\mathrm{I}\Sigma_{n+1}$ on a computational level, despite being logically weaker?

Acknowledgements. The authors are grateful to Ulrich Kohlenbach for pointing out the proof in [16] of the equivalence between Π_n-FAC and Π_n-BC.

References

1 S. R. Buss, editor. *Handbook of Proof Theory*, volume 137. Elsevier, Amsterdam, 1998.

2 M. H. Escardó and P. Oliva. Bar recursion and products of selection functions. Submitted for publication, 2010.

3 M. H. Escardó and P. Oliva. Computational interpretations of analysis via products of selection functions. In F. Ferreira, B. Lowe, E. Mayordomo, and L. M. Gomes, editors, *Computability in Europe 2010, LNCS 6158*, pages 141–150. Springer, 2010.

4 M. H. Escardó and P. Oliva. The Peirce translation and the double negation shift. In F. Ferreira, B. Löwe, E. Mayordomo, and L. M. Gomes, editors, *Programs, Proofs, Processes - CiE 2010, LNCS 6158*, pages 151–161. Springer, 2010.

5 M. H. Escardó and P. Oliva. Selection functions, bar recursion, and backward induction. *Mathematical Structures in Computer Science*, 20(2):127–168, 2010.

6 M. H. Escardó and P. Oliva. Sequential games and optimal strategies. *Royal Society Proceedings A*, 467:1519–1545, 2011.

7 K. Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica*, 12:280–287, 1958.

8 U. Kohlenbach. *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*. Monographs in Mathematics. Springer, 2008.

9 G. Kreisel. On the interpretation of non-finitist proofs, part I. *The Journal of Symbolic Logic*, 16:241–267, 1951.

10 G. Kreisel. On the interpretation of non-finitist proofs, part II: Interpretation of number theory. *The Journal of Symbolic Logic*, 17:43–58, 1952.

11 S. Kuroda. Intuitionistische Untersuchungen der formalistischen Logik. *Nagoya Mathematical Journal*, 3:35–47, 1951.

12 H. Lessan. *Models of Arithmetic.* PhD thesis, Manchester University, 1978.

13 J. B. Paris and L. A. S. Kirby. Σ_n-collection schemas in arithmetic. In *Logic Colloquium '77*, pages 199–210. North Holland, Amserdam, 1978.

14 C. Parsons. On a number theoretic choice schema and its relation to induction. In A. Kino, J. Myhill, and R. E. Vesley, editors, *Intuitionism and Proof Theory: Proceedings of the Summer Conference at Buffalo, N.Y. 1968*, pages 459–473. North Holland, Amserdam, 1970.

15 C. Parsons. On n-quantifier induction. *The Journal of Symbolic Logic*, 37:466–482, 1972.

16 W. Sieg. Fragments of arithmetic. *Annals of Pure and Applied Logic*, 28:33–71, 1985.

17 C. Spector. Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles in current intuitionistic mathematics. In F. D. E. Dekker, editor, *Recursive Function Theory: Proc. Symposia in Pure Mathematics*, volume 5, pages 1–27. American Mathematical Society, Providence, Rhode Island, 1962.