Notes on the game semantics of Gödel's T

Masaru Shirahata 白旗優
sirahata@math.hc.keio.ac.jp
Department of Mathematics
Keio University, Hiyoshi Campus[†]

April 30, 2001

1 Introduction

In 1958, K. Gödel published a paper in the journal *Dialectica*, in which he gave an interpretation of the first-order intuitionistic arithmetic HA (Heyting arithmetic) by the higher-order functional calculus T. This result implies the consistency of the first-order classical arithmetic PA (Peano arithmetic) in the following three steps:

1. Gödel's own double-negation translation of PA into HA;

2. the given interpretation of HA by T;

3. the normalization of T, proved by Tait, using the transfinite induction up to ε_0 as expected.

The functional interpretation is now known as the *Dialectica* interpretation, and extended to the various fragments of first-order and second-order arithmetic. The system T of the functional calculus is a variant of the typed lambda (combinatory) calculus, but notably with the recursors for each type as the new constants.

In 1969, D. Scott wrote up a paper, in which he introduced the typed combinatory calculus, strengthened by the fixpoint operators for each type, and provided with a sound semantics. The paper was intended to exhibit the superiority of the typed calculi to their untyped counterparts, which were likely to be only symbol manipulations without any semantic foundation. Right after the paper was finished, Scott himself found the set-theoretic model for the untyped lambda calculus, and the paper had been famous, but unpublished until 1993. The calculus is now known as PCF, and became one of the most intensively studied calculi in the theoretical computer science.

The most long standing problem with PCF was its semantics. The semantics Scott gave is certainly sound, but it is not "fully abstract" in the sense given by Plotkin. In short, the semantics is fully abstract if the same meaning is

[†] 慶應義塾大学商学部

always assigned to the two terms which are indistinguishable with respect to the observable results in all computational contexts. The problem was finally solved by the game semantics of Abramsky-Jagadeesan-Malacaria, and independently, of Hyland-Ong.

Now, PCF can be formulated as the typed-lambda calculus with a number of built-in constants: the constant zero, the successor, the predecessor, the zero test, the conditionals for each type, and the fixpoint operators for each type. T can be similarly formulated as the typed-lambda calculus with the constant zero, the successor and the recursor operators for each type. By the standard construction of the recursor `Rec` through the fixpoint operator `Y` and the predecessor, T can be seen as the subsystem of PCF. Then it is certainly possible to apply the game semantics of PCF to T, and HA as well: This is our motivation in this paper.

We emphasize that the work presented here is in a very much preliminary stage. We hope, however, that the reader will find the idea itself fairly clear and worth exploring. So, we try to present the entire, not yet completely developed, landscape. We first summarize the game semantics of PCF by Abramsky-Jagadeesan-Malacaria, and, secondly, we consider the characterization of the terms of T within the PCF game semantics. Thirdly, we give the informal argument for the game-theoretical reading of the *Dialectica* interpretation. In the last section of the paper, we briefly discuss the prospect of this line of works.

2 The game semantics of PCF

2.1 Some notational conventions

In this section, we summarize the game semantics given by Abramsky-Jagadeesan-Malacaria [1, 2]. First of all, we need to fix the notations. For a given set $X$, we use the following notational conventions.

- $a, b, c, d, m, n \ldots$: the elements of $X$;
- $s, t, u, v \ldots$: the finite sequences of elements of $X$;
- $|s|$ : the length of the sequence $s$;
- s_i : the ith element of the sequence $s$, where $1 \leq i \leq |s|$;
- $st, as, sa, \ldots$: the concatenation of two sequences, where the element $a$ is identified with its corresponding unit sequence;
- $X^\star$: the set of the finite sequences of elements of $X$;
- $S^{\mathbf{even}}, S^{\mathbf{odd}}$: the set of even (odd) length sequences in $S$, where $S \subseteq X^\star$;
- `Pref` $S$: the set of even length sequences in the prefix closure of $S \subseteq X^\star$;
- $f^\star : X^\star \rightarrow Y^\star$: the unique monoid homomorphism extending $f : X \rightarrow Y$;

- $s \upharpoonright Y$: the sequence obtained by deleting all the occurrences of elements not in Y from s, where $s \in X^*$ and $Y \subseteq X$;
- $s \sqsubseteq t$: meaning that s is a prefix of t, *i.e.*, $su = t$ for some u.

Furthermore, let us fix the notations for the coproduct of sets.

- $\Sigma_{i \in I} X_i$: the disjoint union (co-product) of a family of sets, *i.e.* ,

$$\Sigma_{i \in I} X_i = \{(i, x) \mid i \in I, x \in X_i\}$$

- $X_1 + X_2$: the disjoint union of X_1 and X_2;
- $s \upharpoonright i$: the sequence obtained by deleting from $s \in \Sigma_{i \in I} X_i$ all the occurrences of elements not having the form (i, x) and then identifying (i, x) with x.

The first and second projections are denoted by `fst` and `snd`, respectively. We denote the partial functions f from X to Y by $f : X \rightharpoonup Y$.

- $fx \succeq y$: meaning that fx is defined and equal to y;
- $f^* : Y \rightharpoonup X$: the inverse function for the injective f;
- $f \vee g : X \rightharpoonup Y$: the union of $f : X \rightharpoonup Y$ and $g : X \rightharpoonup Y$ with disjoint domains of definition.

2.2 The definition of games

The game A is played between the Player P and the Opponent O, alternately. The moves M_A of the game A is partitioned by the function λ_A into the Player's and Opponent's moves, and further into Questions and Answers. The acceptable plays are specified by the prefix closed set P_A of sequences in M_A^*. Furthermore, in order to cancel the difference due to the coding convention, the equivalence relation $\approx_A$ is imposed on P_A.

Formally, a game A is the structure $(M_A, \lambda_A, P_A, \approx_A)$, where

- M_A : the set of moves;
- $\lambda_A : M_A \rightarrow \{P, O\} \times \{Q, A\}$; we use the notations
  - $\lambda_A^{PO} = \mathtt{fst} \circ \lambda_A$ and $\lambda_A^{QA} = \mathtt{snd} \circ \lambda_A$
  - $M_A^P = \lambda_A^{-1}(\{P\} \times \{Q, A\})$: the Player's moves
  - $M_A^O = \lambda_A^{-1}(\{O\} \times \{Q, A\})$: the Opponent's moves
  - $M_A^Q = \lambda_A^{-1}(\{P, O\} \times \{Q\})$: the questions
  - $M_A^A = \lambda_A^{-1}(\{P, O\} \times \{A\})$: the answers
  - $\overline{P} = O$ and $\overline{O} = P$
  - $\overline{\lambda_A^{PO}}(a) = \overline{\lambda_A^{PO}(a)}$ and $\overline{\lambda_A} = \langle \overline{\lambda_A^{PO}}, \lambda_A^{QA} \rangle$

- P_A : the set of valid positions of the game; this is a non-empty prefix closed subset of the set $M_A^{\circledast}$ of all finite sequences $s \in M_A^{\star}$ satisfying the three conditions

  (p1) the *starting condition*, *i.e.*, the Opponent moves first;

  $$s = at \implies a \in M_A^O$$

  (p2) the *alternating condition*, *i.e.*, the Player and Opponent alternate;

  $$(\forall i : 1 \le i \le |s|) \left[\lambda_A^{PO}(s_{i+1}) = \overline{\lambda_A^{PO}(s_i)}\right]$$

  (p3) the *bracketing condition*, *i.e.*, no answer is given unless a corresponding question is asked;

  $$(\forall t \sqsubseteq s) \left(|t \restriction M_A^A| \le |t \restriction M_A^Q|\right)$$

- $\approx_A$: an equivalence relation on P_A satisfying

  (e1) $s \approx_A t \implies \lambda_A^{\star}(s) = \lambda_A^{\star}(t)$

  (e2) $s \approx_A t,\ s' \sqsubseteq s,\ t' \sqsubseteq t,\ |s'| = |t'| \implies s' \approx_A t'$

  (e3) $s \approx_A t,\ sa \in P_A \implies \exists b.\ sa \approx_A tb.$

2.3 The definition of strategies

A strategy specifies the unique response of the player against a given move by the opponent. Note that a strategy does not need to specify the responses to all the possible moves. This is different from the definition of strategies in the Abramsky-Jagadeesan games for the multiplicative linear logic. To put it differently, a strategy corresponds to a partial function.

Formally, a *strategy* for the Player is a non-empty subset σ of P_A^{even} such that $\overline{\sigma} = \sigma \cup \mathrm{dom}(\sigma)$ with

$$\mathrm{dom}(\sigma) = \{sa \in P_A^{\mathrm{odd}} \mid \exists b.\ sab \in \sigma\}$$

is prefix closed. In particular, a strategy is *history-free* if it satisfies

- $sab,\ tac \in \sigma \implies b = c$
- $sab, t \in \sigma,\ ta \in P_A \implies tab \in \sigma.$

The equivalence relation $\approx_A$ can be naturally extended to a relation on strategies. Let $\sigma \lessapprox \tau$ if and only if

$$sab \in \sigma,\ s' \in \tau,\ sa \approx_A s'a' \implies \exists b.\ [s'a'b' \in \tau \wedge sab \approx_A s'a'b']$$

and, let $\sigma \approx_A \tau$ if and only if $\sigma \lessapprox \tau \wedge \tau \lessapprox \sigma$. The relation $\approx_A$ on strategies then becomes a partial equivalence relation, and we write $[\sigma] = \{\tau \mid \sigma \approx_A \tau\}$ when

2.4 The multiplicatives and exponential

The tensor product and linear implications of games A and B are defined as follows.

Tensor The game $A \otimes B$ is defined as follows.

- $M_{A\otimes B} = M_A + M_B$
- $\lambda_{A\otimes B} = [\lambda_A, \lambda_B]$, *i.e.* given by cases
- $P_{A\otimes B}$ is the set of all $s \in M^{\circledast}_{A\otimes B}$ such that
  1. *Projection condition:* The restriction of s to the moves in M_A (resp. M_B) is in P_A (resp. P_B)
  2. *Stack discipline:* Every answer in s must be in the same component game (*i.e.* A or B) as the corresponding question.
- $s \approx_{A\otimes B} t$ iff $s \upharpoonright A \approx_A t \upharpoonright A \wedge s \upharpoonright B \approx_A t \upharpoonright B \wedge \mathtt{fst}^\star(s) = \mathtt{fst}^\star(t)$.

The tensor unit is given by

$$I = (\emptyset, \emptyset, \{\epsilon\}, \{(\epsilon, \epsilon)\}).$$

Linear implication The game $A \multimap B$ is defined as follows.

- $M_{A\multimap B} = M_A + M_B$
- $\lambda_{A\multimap B} = [\overline{\lambda_A}, \lambda_B]$
- $P_{A\multimap B}$ is the set of all $s \in M^{\circledast}_{A\multimap B}$ such that
  1. *Projection condition:* The restriction of s to the moves in M_A (resp. M_B) is in P_A (resp. P_B)
  2. *Stack discipline:* Every answer in s must be in the same component game (*i.e.* A or B) as the corresponding question.
- $s \approx_{A\multimap B} t$ iff $s \upharpoonright A \approx_A t \upharpoonright A \wedge s \upharpoonright B \approx_A t \upharpoonright B \wedge \mathtt{fst}^\star(s) = \mathtt{fst}^\star(t)$.

Exponential The game $!A$ is defined as the "infinite tensor power" of A. Formally, it is given as follows.

- $M_{!A} = \omega \times M_A = \Sigma_{i\in\omega} M_A$
- $\lambda_{!A}(i, a) = \lambda_A(a)$
- $P_{!A}$ is the set of all $s \in M^{\circledast}_{!A}$ such that
  1. *Projection condition:* $(\forall i)[s \upharpoonright i \in P_A]$
  2. *Stack discipline:* Every answer in s is in the same index as the corresponding question;
- Let $S(\omega)$ be the set of permutations on ω; then

  $s \approx_{!A} t \implies (\exists \pi \in S(\omega))[(\forall i \in \omega.\ s \upharpoonright i \approx_A t \upharpoonright \pi(i) \wedge (\pi \circ \mathtt{fst})^\star = \mathtt{fst}^\star(t)]$

2.5 The category of games

Games and strategies form a category in which

- the objects are games,
- the morphisms from A to B are the equivalence classes $[\sigma]$ of strategies σ on the game $A \multimap B$ with $\sigma \approx_{A \multimap B} \sigma$.

The composition of $[\sigma] : A \to B$ and $[\tau] : B \to C$ is defined as follows. The strategies σ and τ can be identified with the partial functions $\sigma_f : M_A^P + M_B^O \rightharpoonup M_A^O + M_B^P$ and and $\tau_g : M_B^P + M_C^O \to M_B^O + M_C^P$, respectively. The composition $\tau \circ \sigma$ is then the strategy on the game $A \multimap C$, *i.e.* the partial function $(\tau \circ \sigma)_h : M_A^P + M_C^O \rightharpoonup M_A^O + M_C^P$ given by the instruction:

- Case: $a \in M_A^P$; apply σ_f; if the result is in M_A^O, return it; else apply the following loop to it;
  - Case: $b \in M_B^P$; apply τ_g; if the result is in M_C^P, return it; else continue the loop with the result;
  - Case: $b \in M_B^O$; apply σ_f; if the result is in M_A^O, return it; else continue the loop with the result;
- Case: $a \in M_C^O$; apply τ_g; if the result is in M_C^P, return it; else apply the above loop to it.

Identity The identity morphism $\mathtt{id}_A : A \to A$ is given by the "copycat strategy." Let us distinguish two copies of A, as A_0 and A_1. Then the identity is the function from $M_{A_0}^P + M_{A_0}^O \to M_{A_1}^O + M_{A_1}^P$ such that $(0, a) \in M_{A_0}^P$ and $(0, a') \in M_{A_0}^O$ are copied as $(1, a) \in M_{A_1}^P$ and $(1, a') \in M_{A_1}^O$, respectively.

Linear application The linear application $\mathtt{LAPP}_{A,B}$ from $A_0 \otimes (A_1 \multimap B_0)$ to B_1 is the copycat strategy between A_0 and A_1, and between B_0 and B_1.

Dereliction The dereliction $\mathtt{der}_A : \,!A \to A$ is obtained by choosing one component A_i in $!A$ and doing the copycat between A_i and the target A.

Promotion Given $[\sigma] : \,!A \to B$, we can obtain the morphism $[\sigma^\dagger] : \,!A \to \,!B$ by taking the disjoint union of ω many copies of σ, and identifying $\omega \times \omega \times M_A$ and $\omega \times M_A$ *via* a bijection between $\omega \times \omega$ and ω.

Contraction The contraction $\mathtt{cont}_A : \,!A \to \,!A \otimes \,!A$ is given by taking two disjoint infinitary subsets κ_0 and κ_1 of ω, and doing the copycat between $\kappa_0 \times M_A$ (resp. $\kappa_1 \times M_A$) and $\{0\} \times \omega \times M_A$ (resp. $\{1\} \times \omega \times M_A$) in the target.

Weakening The weakening $\mathtt{weak}_A : \,!A \to I$ is given by the empty strategy. Note that I is indeed the terminal object in this category.

2.6 The model of PCF

The model of PCF is obtained as the co-Kleisli category of the category of games, where

- the objects are games,
- the morphisms from A to B are morphisms $f : {!A} \to B$ in the category of games.

We use the notation $f : A \Rightarrow B$ for the morphism f in the co-Kleisli category. For $f : A \Rightarrow B$ and $g : B \Rightarrow C$, we write $g \circ f$ for the composition in the co-Kleisli category defined as the composition in the category of games:

$$g \circ f : {!A} \xrightarrow{f^\dagger} {!B} \xrightarrow{g} C$$

The notation $f; g$ may be used solely for the category of games.

Definition 2.1. *We define the game $A\&B$ as follows:*

- $M_{A\&B} = M_A + M_B$
- $\lambda_{A\&B} = [\lambda_A, \lambda_B]$, *i.e. defined by cases*
- $P_{A\&B} = P_A + P_B$
- $\approx_{A\&B} = \approx_A + \approx_B$.

Furthermore, the projections $A \xleftarrow{\mathtt{first}} A\&B \xrightarrow{\mathtt{second}} B$ are defined as the strategies induced by the partial identities, respectively,

$$\begin{array}{ll} (M_A^P + M_B^P) + M_A^O \rightharpoonup (M_A^O + M_B^O) + M_A^P & \textit{undefined on } M_B^P \\ (M_A^P + M_B^P) + M_B^O \rightharpoonup (M_A^O + M_B^O) + M_B^P & \textit{undefined on } M_A^P. \end{array}$$

Proposition 2.2. *In the category of games,*

1. *there are natural isomorphisms* $\mathsf{e}_{A,B} : {!(A\&B)} \cong {!A} \otimes {!B}$,
2. $!I = I$.

Proposition 2.3. *I is terminal in the co-Kleisli category.*

Proposition 2.4. *$A\&B$ is a Cartesian product in the co-Kleisli category with the projections $\pi_1 : A\&B \Rightarrow A$ and $\pi_2 : A\&B \Rightarrow B$ defined as*

$$\begin{array}{l} \pi_1 : {!(A\&B)} \xrightarrow{\mathtt{der}} A\&B \xrightarrow{\mathtt{first}} A \\ \pi_2 : {!(A\&B)} \xrightarrow{\mathtt{der}} A\&B \xrightarrow{\mathtt{second}} B. \end{array}$$

Proposition 2.5. *The co-Kleisli category has a countable product.*

Proof. It is just a countable disjoint union of games. □

Note that the application morphism $\mathtt{Ap}_{A,B} : (A \Rightarrow B)\&A \Rightarrow B$ is given by

$$\begin{array}{l} \mathtt{Ap}_{A,B} : {!((!A \to B)\&A)} \xrightarrow{e} {!(!A \to B)} \otimes {!A} \\ \qquad\qquad\qquad\qquad \xrightarrow{\mathtt{der} \otimes \mathtt{id}} (!A \to B) \otimes {!A} \xrightarrow{\mathtt{LAPP}} B. \end{array}$$

Ground types The ground type nat is interpreted as the game N where

- $M_N = \{*\} \cup \omega$
- $\lambda_N = \{(*, OQ)\} \cup \{(n, PA) \mid n \in \omega\}$
- $P_N = \{*\} \cup \{(*, n) \mid n \in \omega\}$
- $\approx_N$ is the identity.

The ground type `bool` is interpreted by the game B defined similarly with the values `true` and `false`.

The constants The constant zero is the strategy $\{\epsilon, *0\}$ on N. Truth values are defined similarly.

Successor and predecessor The successor is the strategy

$$\{\epsilon, *_1 *_i\} \cup \{*_1 *_i n_i (n+1)_1 \mid n \in \omega\}$$

on $!N_0 \multimap N_1$ where we write $*_i$ for $(i, (0, *))$ and so on. The predecessor is defined similarly.

Zero test and conditionals The zero test is the strategy

$$\{\epsilon,\ *_B *_{N_i},\ *_B *_{N_i} 0\,\mathtt{false}\} \cup \{*_B *_{N_i} n\,\mathtt{true} \mid n \in \omega \wedge 0 < n\}$$

on $!N \multimap B$ where $*_B$ is initial question by the opponent in B and so on. For the conditional of the type $B \to A_0 \to A_1 \to A_2$, we use the strategy

$$\begin{aligned}&\{\epsilon,\ *_2 *_B,\ *_2 *_B\ \mathtt{true} *_0,\ *_2 *_B\ \mathtt{false} *_1\}\\ &\qquad \cup \{*_2 *_B\ \mathtt{true}\ *_0\, a_0 a_2,\ *_2 *_B\ \mathtt{false}\ *_1\, b_1 b_2 \mid a \in M^O_{A_0}, b \in M^O_{A_1}\}\end{aligned}$$

where a_0 and a_2 are copies of the same object, and similarly for b_1 and b_2.

Fixpoint operators The homset of our co-Kleisli category is the set of strategies and forms a pointed poset (*i.e.* poset with a least element) under the ordering $\lessapprox$ on strategies. It is now known whether this poset is a CPO or not, but it has a certain property sufficient for the standard construction of fixpont operators. The property is called *rationality* and defined as below.

A cartesian closed category C is ppo-enriched if:

- Every homset has a pointed poset structure;
- Composition, pairing and currying are monotone;
- Composition is left-strict, *i.e.* for all $f : A \to B$,

$$\perp_{B,C} \circ f = \perp_{A,C}$$

where $\perp_{B,C}$ and $\perp_{A,C}$ are the least elements of the homsets.

$\mathcal{C}$ is *rational* if it is ppo-enriched, and for all $f : A \times B \to B$:

- The chain $(f^{(k)} \mid k \in \omega)$ in the homset $\mathcal{C}(A, B)$ defined inductively by
$$f^{(0)} = \bot_{A,B}, \qquad f^{(k+1)} = f \circ \langle \mathtt{id}_A, f^{(k)} \rangle$$
has a least upper bound f^{∇};
- For all $g : C \to A$ and $h : B \to D$,
$$h \circ f^{\nabla} \circ g = \sup_{k \in \omega} h \circ f^{(k)} \circ g.$$

Now let Θ_A from $I \times (A \to A) \to A$ to $(A \to A) \to A$ be
$$[\![F : (A \to A) \to A \vdash \lambda f^{A \to A}.\ f(Ff) : (A \to A) \to A]\!]$$
and define the fixpoint operator $\mathtt{Y}_A = \Theta_A^{\nabla}$. This is indeed nothing but the least upper bound of
$$[\![\mathtt{Y}_A^{(k)}]\!] = [\![\lambda f^{A \to A}.\ \underbrace{f(f(f \ldots (f}_{k} \bot_A) \ldots))]\!].$$

2.7 The decomposition lemma

Definition 2.6. *We define the strategy $\chi : N \& N^{\omega} \Rightarrow N$ and $\chi_{\alpha} : N \otimes N \to N$, where N^{ω} is the infinitary product of N, by*
$$\chi_{\alpha} = \mathtt{Pref}\ \{*_0 *_1 n *_{2,n} m_{2,n} m_0 \mid n, m \in \omega\} : N_1 \otimes N_2^{\omega} \multimap N_0$$
and
$$\chi : {!}(N \& N^{\omega}) \xrightarrow{\mathsf{e}} {!}N \otimes {!}(N^{\omega}) \xrightarrow{\mathsf{der} \otimes \mathsf{der}} N \otimes N^{\omega} \xrightarrow{\chi_{\alpha}} N$$

Let us consider the game $(A_1 \& \cdots \& A_k) \Rightarrow N$ where
$$A_i = (B_{i,1} \Rightarrow \cdots \Rightarrow B_{i,l_i}) \Rightarrow N.$$

We write $\tilde{A}$ for $A_1 \& \cdots \& A_k$. We define the strategies $\bot_{\tilde{A}} : \tilde{A} \Rightarrow N$ and $\mathtt{K}_{\tilde{A}} n : \tilde{A} \Rightarrow N$ by
$$\bot_{\tilde{A}} = \{\epsilon\}, \qquad \mathtt{K}_{\tilde{A}} n = \{\epsilon, *n\}.$$
and, if for some $1 \leq i \leq k$ and for each $1 \leq j \leq l_i$, we have
$$\sigma_j : \tilde{A} \Rightarrow B_{i,j}$$
and for each $n \in \omega$,
$$\tau_n : \tilde{A} \Rightarrow N$$
then we define the strategy
$$\mathtt{C}_i(\sigma_i, \cdots, \sigma_{l_1}, (\tau_n \mid n \in \omega)) : \tilde{A} \Rightarrow N$$
by
$$\mathtt{C}_i(\sigma_i, \cdots, \sigma_{l_1}, (\tau_n \mid n \in \omega)) = \chi \circ \langle \mathsf{Ap} \circ \langle \cdots \mathsf{Ap} \circ \langle \pi_i, \sigma_1 \rangle, \cdots, \sigma_{l_i} \rangle, \langle \tau_n \mid n \in \omega \rangle \rangle.$$

Theorem 2.7 (The decomposition lemma). *Let*

$$\sigma : (A_1 \& \cdots \& A_p) \Rightarrow (A_{p+1} \Rightarrow \cdots \Rightarrow A_q \Rightarrow N)$$

be any strategy, where

$$A_i = B_{i,1} \Rightarrow \cdots \Rightarrow B_{i,l_i} \Rightarrow N, \quad 1 \le i \le q.$$

We write $\tilde{C}$ for $A_1, \cdots, A_p$, and $\tilde{D}$ for $A_{p+1}, \cdots, A_q$. If $\tau : \tilde{C}, \tilde{D} \Rightarrow N$, then we write $\Lambda_{\tilde{D}}(\tau) : \tilde{C} \Rightarrow (A_{p+1} \Rightarrow \cdots \Rightarrow A_q \Rightarrow N)$ for the currying of τ with respect to $\tilde{D}$. Then exactly one of the following three cases applies.

1. $\sigma = \Lambda_{\tilde{D}}(\bot_{\tilde{C},\tilde{D}})$
2. $\sigma = \Lambda_{\tilde{D}}(\mathtt{K}_{\tilde{C},\tilde{D}} n)$ *for some* $n \in \omega$
3. $\sigma = \Lambda_{\tilde{D}}(\mathtt{C}_i(\sigma_i, \cdots, \sigma_{l_1}, (\tau_n \mid n \in \omega)))$ *where* $1 \le i \le q$, *and*

$$\begin{array}{lll} \sigma_j : & \tilde{C}, \tilde{D} \Rightarrow B_{i,j}, & 1 \le j \le l_i \\ \tau_n : & \tilde{C}, \tilde{D} \Rightarrow N, & n \in \omega. \end{array}$$

Theorem 2.8 (The unicity of decomposition).

1. *If* $\sigma \approx \bot_{\tilde{C},\tilde{D}}$, *then* $\sigma = \bot_{\tilde{C},\tilde{D}}$.
2. *If* $\sigma \approx \mathtt{K}_{\tilde{C},\tilde{D}} n$, *then* $\sigma = \mathtt{K}_{\tilde{C},\tilde{D}} n$.
3. *If* $\mathtt{C}_i(\sigma_i, \cdots, \sigma_{l_1}, (\tau_n \mid n \in \omega)) \lessapprox \mathtt{C}_i(\sigma'_i, \cdots, \sigma'_{l_1}, (\tau'_n \mid n \in \omega))$, *then*

$$\sigma_j \lessapprox \sigma'_j, \ 1 \le j \le l_i, \quad \text{and} \quad \tau_n \lessapprox \tau'_n, \ n \in \omega.$$

3 The game semantics of T

3.1 The functional calculus T

Gödel used an extension of the type lambda calculus, which he called T, to interpret formulas of HA [3, 4]. The calculus T is a standard typed lambda calculus enhanced with the new constants, the constant zero, the successor and the recursion operator for each type. The functions computed by T are called the *primitive recursive functionals of finite type.*

Types Types are defined inductively from the single ground type **nat** by the binary constructor $\rightarrow$.

Terms In addition to the standard machinery of the simply typed lambda calculus, T has the constants:

- the constant zero $\mathtt{0}$ with type **nat**
- the successor $\mathtt{succ}$ with type **nat** $\rightarrow$ **nat**
- the recursor $\mathtt{Rec}_t$ with type $t \rightarrow (\textbf{nat} \rightarrow t \rightarrow t) \rightarrow \textbf{nat} \rightarrow t$ for each type t

Reduction We write $\mathtt{m}$ for the term obtained by applying succ to 0 m times. T has the reductions for the new constants:

- $\mathtt{Rec}_t MN0 \mapsto M$
- $\mathtt{Rec}_t MN(\mathtt{succ}\,\mathtt{m}) \mapsto N\mathtt{m}(\mathtt{Rec}_t MN\mathtt{m})$.

where the terms M and N are of the types t and $\mathbf{nat} \to t \to t$, respectively. Note that $\mathtt{Rec}_t$ can be defined in PCF by the fixpoint operator as follows.

$$\mathtt{Y}(\lambda f.\ \lambda x.\ \lambda y.\ \lambda z.\ \mathtt{if}\ \mathtt{zero?}(z)\ \mathtt{then}\ x\ \mathtt{else}\ y(\mathtt{pred}(z))(fxy(\mathtt{pred}(z))))$$

Hence, it is certainly possible to give the game semantical interpretation of T through PCF. We will give, however, more direct interpretation.

3.2 The strategy χ^A

The key idea is to use the translation into the infinitary calculus, as given by Tait in the normalization proof of T.

In Tait's infinitary calculus, we may form the term $\langle M_i \rangle$ with type $\mathbf{nat} \to t$ from the countably many M_i of type t. This term is required to satisfy the reduction:

- $\langle M_i \rangle \mathtt{n} \mapsto M_n$
- $\langle M_i \rangle \mathtt{n} M' \mapsto \langle M_i M' \rangle \mathtt{n}$

where M_i and M' has the types $s \to t$ and s, respectively.

The strategy χ used in the decomposition lemma gives the semantical version of the above construction for the basic type N. We generalize χ to χ^A for the game $A = A_1 \Rightarrow \cdots A_p \Rightarrow N$ and show that the construction given by χ^A indeed satisfies the second equation (reduction relation) as well.

Definition 3.1. *Let $A = A \to \cdots \Rightarrow N$. We define $\chi^A_\alpha : N \otimes A^\omega \to A_1$ by*

$$\chi^A_\alpha = \mathtt{Pref}\ \{*_1 *_N n_2 *_{n_2} s \mid *_1 *_{n_2} s \in \mathtt{id}_A\}.$$

Proposition 3.2. *χ^A_α is a strategy.*

Proof. Immediate from the fact that $\mathtt{id}_A$ is a strategy. □

Definition 3.3. *We define $\chi^A : N\&A^\omega \Rightarrow A$ by the composition:*

$$\chi^A : !(N\&A^\omega) \xrightarrow{\mathtt{e}} !N \otimes !(A^\omega) \xrightarrow{\mathtt{der}\otimes\mathtt{der}} N \otimes A^\omega \xrightarrow{\chi^A_\alpha} A.$$

Lemma 3.4. *Let $A = A_1 \Rightarrow A_2 \Rightarrow \cdots \Rightarrow A_p \Rightarrow N$ and $A' = A_2 \Rightarrow \cdots \Rightarrow A_p \Rightarrow N$. For $\tau_n : C \Rightarrow A$, $n \in \omega$, define $\tau'_n : C\&A_1 \Rightarrow A'$ by*

$$\tau'_n : !(C\&A_1) \xrightarrow{\mathtt{e}} !C \otimes !A_1 \xrightarrow{\tau_n \otimes \mathtt{id}} A \otimes !A_1 \xrightarrow{\mathtt{LAPP}} A'.$$

$$\sigma : N \otimes {!}(C \& A_1) \xrightarrow{\mathtt{id} \otimes e} N \otimes ({!}C \otimes {!}A_1) \xrightarrow{\alpha} (N \otimes {!}C) \otimes A_1$$
$$\xrightarrow{(\mathtt{id} \otimes \langle \tau_n | n \in \omega \rangle) \otimes \mathtt{id}} (N \otimes A^{\omega}) \otimes {!}A_1 \xrightarrow{\chi_\alpha^A \otimes \mathtt{id}} A \otimes {!}A_1 \xrightarrow{\mathtt{LAPP}} A'$$

and

$$\sigma' : N \otimes {!}(C \& A_1) \xrightarrow{\mathtt{id} \otimes \langle \tau'_n | n \in \omega \rangle} N \otimes A'^{\omega} \xrightarrow{\chi_\alpha^{A'}} A'.$$

Then $\sigma \approx \sigma'$.

Proof. Any sequence in σ is identical to some sequence in σ' modulo reindexing, and vice versa. □

Proposition 3.5. *For* $\tau_n : C \Rightarrow A$, $n \in \omega$, *let* $\tau'_n = \mathtt{Ap} \circ (\tau_n \& \mathtt{id}_A)$. *Define* $\sigma : N\&(C\&A_1) \Rightarrow A'$ *and* $\sigma' : N\&(C\&A_1) \Rightarrow A'$ *by*

$$\sigma = \mathtt{Ap} \circ (\chi^A \& \mathtt{id}_{A_1}) \circ ((\mathtt{id}_N \& \langle \tau_n \mid n \in \omega \rangle)) \circ \alpha'$$
$$\sigma' = \chi^{A'} \circ (\mathtt{id}_N \& \langle \tau'_n \mid n \in \omega \rangle)$$

where

$$\alpha' : {!}(N\&(C\&A_1)) \xrightarrow{\mathtt{e}} {!}N \otimes {!}(C\&A_1) \xrightarrow{\mathtt{id} \otimes \mathtt{e}} {!}N \otimes ({!}C \otimes {!}A_1)$$
$$\xrightarrow{\alpha} ({!}N \otimes {!}C) \otimes {!}A_1 \xrightarrow{\mathtt{e}^{-1} \otimes \mathtt{id}} {!}(N\&C) \otimes {!}A_1 \xrightarrow{\mathtt{e}^{-1}} {!}((N\&C)\&A_1)$$
$$\xrightarrow{\mathtt{der}} (N\&C)\&A_1.$$

Then $\tau'_n \approx \mathtt{e}; \tau_n \otimes \mathtt{id}_{!A_1}; \mathtt{LAPP}$ *and* $\sigma \approx \sigma'$.

Proof. Use the previous lemma and the properties of $\mathtt{der}$ and $(\_)^{\dagger}$. □

Corollary 3.6. *Let* $\tau_n : C\&A_1 \Rightarrow A'$, $n \in \omega$. *Then*

$$\Lambda_{A_1}(\chi^{A'} \circ (\mathtt{id}_N \& \langle \tau_n \mid n \in \omega \rangle) \circ \alpha'^{-1}) \approx \chi^A \circ (\mathtt{id}_N \& \langle \Lambda_{A_1}(\tau_n) \mid n \in \omega \rangle).$$

Fact 3.7.

1. *Let* $f : D \to C$ *and* $g : B\&A \to D$. *Then,*

$$\mathtt{Ap} \circ (\Lambda_A(f \circ g) \& \mathtt{id}_A) \approx g \circ \mathtt{Ap} \circ (\Lambda_A(g) \& \mathtt{id}_A).$$

2. *Let* $f_i : B\&A \to C_i$, *for* $i \in I \subseteq \omega$. *Then,*

$$\langle \mathtt{Ap} \circ (\Lambda(f_i) \& \mathtt{id}_A) \mid i \in I \rangle \approx \mathtt{Ap} \circ (\Lambda_A(\langle f_i \mid i \in I \rangle) \& \mathtt{id}_A).$$

Lemma 3.8. *Let* $\tau_n : {!C} \to N$, $n \in \omega$, *and* $\tau'_m : {!D} \to N$, $m \in \omega$. *Define* $\sigma, \sigma' : (N \otimes {!C}) \otimes {!D} \to N$ *by*

$$\sigma : (N \otimes {!C}) \otimes {!D} \xrightarrow{(\mathtt{id} \otimes \langle \tau_n | n \in \omega\rangle) \otimes \mathtt{id}} (N \otimes N^\omega) \otimes {!D} \xrightarrow{\chi_\alpha \otimes \mathtt{id}} \longrightarrow N \otimes {!D}$$
$$\xrightarrow{\mathtt{id} \otimes \langle \tau'_m | m \in \omega\rangle} N \otimes N^\omega \xrightarrow{\chi_\alpha} N$$

$$\sigma' : (N \otimes {!C}) \otimes {!D} \xrightarrow{\alpha^{-1}} N \otimes ({!C} \otimes {!D}) \xrightarrow{\mathtt{id} \otimes \tau''} N \otimes N^\omega \xrightarrow{\chi_\alpha} N$$

where $\tau'' : ({!C} \otimes {!D}) \xrightarrow{\langle \tau_n \otimes \langle \tau'_m | m \in \omega\rangle ; \chi_\alpha | n \in \omega\rangle} N^\omega$. *Then* $\sigma \approx \sigma'$.

Proof. Any sequence in σ is identical to some sequence in σ' modulo reindexing, and vice versa. □

Proposition 3.9. *Let* $\sigma : A \Rightarrow N$, $\tau_n : C \Rightarrow N$ *and* $\tau'_m : D \Rightarrow N$, *for* $n, m \in \omega$. *Then,*

$$\chi \circ ((\chi \circ (\sigma \& \langle \tau_n \mid n \in \omega\rangle)) \& \langle \tau'_m \mid m \in \omega\rangle)$$
$$\approx \chi \circ (\sigma \& \langle \chi \circ (\tau_n \& \langle \tau_m \mid m \in \omega\rangle) \mid n \in \omega\rangle) \circ \alpha^{-1}.$$

Proof. Use the previous lemma. □

Proposition 3.10. $\chi \circ (\mathtt{K}_{!D} n \& \langle \tau_n \mid n \in \omega\rangle) \approx \tau_n \circ \pi_2$.

Proof. This reduces to $\mathtt{K}_I n \otimes \langle \tau_n \mid n \in \omega\rangle ; \chi_\alpha \approx \mathtt{unit}; \tau_n$, where $\mathtt{unit}$ is the empty strategy. □

3.3 The interpretation of T

The interpretation of types is the same as PCF. For terms, we give more direct interpretations by the Tait translation. Furthermore, we present the expanded version of the interpretations which already conform to the normal form given by the decomposition lemma. Hence we have reduced χ^A to χ from the beginning. The soundness of such reduction is exactly what is guaranteed by the properties of χ^A, particularly by Proposition 3.5.

To enhance the readability, we use the notation which is similar to natural deduction. We write the morphism $\mathtt{Ap} \circ \langle \sigma, \tau\rangle : C \Rightarrow B$ as

$$\dfrac{\begin{array}{cc} C & C \\ \vdots\ \sigma & \vdots\ \tau \\ A \Rightarrow B & A \end{array}}{B}\ \mathtt{Ap}$$

and $\chi \circ \langle \sigma, \langle \tau_n \mid n \in \omega\rangle\rangle : C \Rightarrow N$ as

$$\dfrac{\begin{array}{cc} C & C \\ \vdots\ \sigma & \vdots\ \tau_n \\ N & N \end{array}}{N}\ \chi$$

with the index n quantified over ω. Furthermore, we simply write n for $\mathtt{K}_A n$.

$[\![H \rhd x_i : \mathbf{nat}]\!] =$

$$\dfrac{\begin{matrix}[\![H]\!]\\ \vdots\;\pi_i\\ N\end{matrix} \qquad \begin{matrix}[\![H]\!]\\ \vdots\; n\\ N\end{matrix}}{N}\;\chi$$

$[\![H \rhd x_i : A_{p+1} \rightarrow \cdots \rightarrow A_q \rightarrow \mathbf{nat}]\!] =$

$$\begin{array}{c} \dfrac{\dfrac{\begin{matrix}\dfrac{\begin{matrix}[\![H']\!]\\ \vdots\;\pi_i\\ A_{p+1} \Rightarrow \cdots \Rightarrow B\end{matrix} \qquad \begin{matrix}[\![H']\!]\\ \vdots\;[\![H' \rhd x_{p+1} : A_{p+1}]\!]\\ A_{p+1}\end{matrix}}{A_{p+2} \Rightarrow \cdots \Rightarrow N}\;\mathsf{Ap}\\ \vdots\\ A_q \Rightarrow N\end{matrix} \qquad \begin{matrix}[\![H']\!]\\ \vdots\;[\![H' \rhd x_q : A_q]\!]\\ A_q\end{matrix}}{N}\;\mathsf{Ap} \qquad \begin{matrix}[\![H']\!]\\ \vdots\; n\\ N\end{matrix}}{N}\;\chi \\ \hline\hline A_{p+1} \Rightarrow \cdots \Rightarrow A_q \Rightarrow N \end{array}\;\Lambda_{A_{p+1},\cdots,A_q}$$

with $H' = H, x_{p+1} : A_{p+1}, \cdots, x_q : A_q$. Here and in the following, A's are used for the names of types as well as their interpretations.

$[\![H \rhd \lambda x : s. M : s \rightarrow t]\!] =$

$$\dfrac{\begin{matrix}[\![H, x : s]\!]\\ \vdots\;[\![M]\!]\\ [\![t]\!]\end{matrix}}{[\![s]\!] \Rightarrow [\![t]\!]}\;\Lambda_{[\![s]\!]}$$

$[\![H \rhd 0 : \mathbf{nat}]\!] =$

$$\begin{matrix}[\![H]\!]\\ \vdots\; 0\\ N\end{matrix}$$

$[\![H \rhd \mathbf{succ} : \mathbf{nat} \rightarrow \mathbf{nat}]\!] =$

$$\dfrac{\dfrac{\begin{matrix}[\![H, x : \mathbf{nat}]\!]\\ \vdots\;\pi_{p+1}\\ N\end{matrix} \qquad \begin{matrix}[\![H, x : \mathbf{nat}]\!]\\ \vdots\; n+1\\ N\end{matrix}}{N}\;\chi}{N \Rightarrow N}\;\Lambda_N$$

$[\![H \rhd MN : t]\!] =$

$$
\cfrac{
 \cfrac{
  \begin{array}{c}
   \cfrac{
    \cfrac{
     \begin{array}{c}[\![H']\!]\\ \vdots\ [\![M]\!]\\ [\![s]\!]\Rightarrow[\![t]\!]\end{array}
     \quad
     \begin{array}{c}[\![H']\!]\\ \vdots\ [\![N]\!]\\ [\![s]\!]\end{array}
    }{[\![t]\!]}\,\mathsf{Ap}
    \quad
    \begin{array}{c}[\![H']\!]\\ \vdots\ [\![H'\rhd x_{p+1}:A_{p+1}]\!]\\ A_{p+1}\end{array}
   }{A_{p+2}\Rightarrow\cdots\Rightarrow N}\,\mathsf{Ap}\\
   \vdots\\
   A_q\Rightarrow N
  \end{array}
  \quad
  \begin{array}{c}[\![H']\!]\\ \vdots\ [\![H'\rhd x_q:A_q]\!]\\ A_q\end{array}
 }{N}\,\mathsf{Ap}
 \quad
 \begin{array}{c}[\![H']\!]\\ \vdots\ n\\ N\end{array}
}{
 \cfrac{N}{[\![t]\!]}\,\Lambda_{A_{p+1},\cdots,A_q}
}\,\chi
$$

with $t = A_{p+1} \to \cdots \to A_q \to \mathbf{nat}$ and $H' = H, x_{p+1} : A_{p+1}, \cdots, x_q : A_q$.

$[\![H \rhd \mathtt{Rec}_t : t \to (\mathbf{nat} \to t \to t) \to \mathbf{nat} \to t]\!] =$

$$
\cfrac{
 \cfrac{
  \cfrac{
   \cfrac{
    \cfrac{
     \begin{array}{c}[\![H']\!]\\ \vdots\ \pi_{p+3}\\ N\end{array}
     \quad
     \begin{array}{c}[\![H']\!]\\ \vdots\ \tau_n\\ N\end{array}
    }{N}\,\chi
   }{[\![t]\!]}\,\Lambda_{A_{p+4},\cdots,A_q}
  }{N\Rightarrow[\![t]\!]}\,\Lambda_N
 }{(N\Rightarrow[\![t]\!]\Rightarrow[\![t]\!])\Rightarrow N\Rightarrow[\![t]\!]}\,\Lambda_{N\Rightarrow[\![t]\!]\Rightarrow[\![t]\!]}
}{[\![t]\!]\Rightarrow(N\Rightarrow[\![t]\!]\Rightarrow[\![t]\!])\Rightarrow N\Rightarrow[\![t]\!]}\,\Lambda_{[\![t]\!]}
$$

with $t = A_{p+4} \to \cdots \to A_q \to \mathbf{nat}$ and

$$H' = H, x_{p+1} : t, x_{p+2} : \mathbf{nat} \to t \to t, x_{p+3} : \mathbf{nat}, x_{p+4} : A_{p+4}, \cdots, x_q : A_q$$

where τ_0 is

$$
\cfrac{
 \cfrac{
  \begin{array}{c}
   \cfrac{
    \begin{array}{c}[\![H']\!]\\ \vdots\ \pi_{p+1}\\ [\![t]\!]\end{array}
    \quad
    \begin{array}{c}[\![H']\!]\\ \vdots\ [\![H'\rhd x_{p+4}:A_{p+4}]\!]\\ A_{p+4}\end{array}
   }{A_{p+5}\Rightarrow\cdots\Rightarrow A_q\Rightarrow N}\,\mathsf{Ap}\\
   \vdots\\
   A_q\Rightarrow N
  \end{array}
  \quad
  \begin{array}{c}[\![H']\!]\\ \vdots\ [\![H'\rhd x_q:A_q]\!]\\ A_q\end{array}
 }{N}\,\mathsf{Ap}
 \quad
 \begin{array}{c}[\![H']\!]\\ \vdots\ n\\ N\end{array}
}{N}\,\chi
$$

and τ_{n+1} is

$$
\dfrac{
  \begin{array}{c}
  \dfrac{
    \dfrac{
      \dfrac{
        \begin{array}{c}\llbracket H'\rrbracket\\ \vdots\ \pi_{p+2}\\ N \Rightarrow \llbracket t\rrbracket \Rightarrow \llbracket t\rrbracket\end{array}
        \quad
        \begin{array}{c}\llbracket H'\rrbracket\\ \vdots\ n\\ N\end{array}
      }{\llbracket t\rrbracket \Rightarrow \llbracket t\rrbracket}\,\mathtt{Ap}
      \quad
      \dfrac{\begin{array}{c}\llbracket H'\rrbracket\\ \vdots\ \tau_n\\ N\end{array}}{\llbracket t\rrbracket}\,\Lambda_{A_{p+4},\cdots,A_q}
    }{\llbracket t\rrbracket}\,\mathtt{Ap}
    \quad
    \begin{array}{c}\llbracket H'\rrbracket\\ \vdots\ \llbracket H'\rhd x_{p+4} : A_{p+4}\rrbracket\\ A_{p+4}\end{array}
  }{A_{p+5} \Rightarrow \cdots \Rightarrow A_q \Rightarrow N}\,\mathtt{Ap}
  \\ \vdots \\
  A_q \Rightarrow N
  \end{array}
  \quad
  \begin{array}{c}\llbracket H'\rrbracket\\ \vdots\ \llbracket H'\rhd x_q : A_q\rrbracket\\ A_q\end{array}
}{N}\,\mathtt{Ap}
\quad
\dfrac{}{}
\begin{array}{c}\llbracket H'\rrbracket\\ \vdots\ n\\ N\end{array}
\;\Big/\; N \ \chi
$$

3.4 The characterization of T

In this part, we show that if we recursively apply the decomposition lemma to any interpretation of a term of T, then we obtain the well-founded tree with the rank $< \varepsilon_0$.

Note that the well-foundedness of strategies (seen as trees of moves) does not characterize T at all, since the identity strategy may not terminate depending on how the Opponent plays.

The restriction of the class of strategy-inducing partial functions does not work, either. Consider the strategy $\sigma : N_1 \Rightarrow N_0$ induced by

$$*_0 \mapsto *_{1,0}, \qquad n_{1,i} \mapsto *_{1,i+1}.$$

This function can be encoded by a primitive recursive function (indeed pairing suffices), but σ is a totally undefined function, since it first responds to the initial question in N_0 by the question in the first component of $!N$, and whatever answer may be returned, it keeps asking a question in the next component.

Our result is not surprising nor sufficient for the complete characterization of T, since we can encode any total function from ω to ω by a strategy with the well-founded decomposition tree, but it at least shows how the techniques in the infinitary logic [5] can be applied to the game semantics as well.

For the complete characterization, we seem to need the restriction of the class of strategy-inducing partial functions and the well-foundedness of the decomposition tree, both.

We will work with the expressions for morphisms rather than morphisms themselves. The same symbols are used for the expressions as for morphisms. All the proofs are rather straightforward and we only give their sketches.

Definition 3.11 (Degree). *We define the degree of of the expression A, denoted* $\deg A$*, by*

- $\deg N = 0$
- $\deg(A \Rightarrow B) = \max(\deg A, \deg B) + 1.$

If the expression σ has the form given by

$$\sigma ::= \pi_i \mid \mathsf{K}_A n \mid \Lambda_A(\sigma) \mid \mathsf{Ap}_{A,B} \circ \langle \sigma, \sigma \rangle \mid \chi \circ \langle \sigma \mid n \in \omega \rangle$$

then σ will be called a T-expression.

Definition 3.12 (Rank). *We define the rank of the T-expression σ, denoted $|\sigma|$, by*

- $|\pi_i| = 1$ *and* $|\mathsf{K}n| = 1$
- $|\mathsf{Ap} \circ \langle \sigma, \tau \rangle| = \max(|\sigma|, |\tau|) + 1$
- $|\Lambda_A(\sigma)| = |\sigma| + 1$
- $|\chi \circ \langle \sigma, \langle \tau_n \mid n \in \omega \rangle \rangle| = \sup(\{|\sigma + 1|\} \cup \{|\tau_n| + 1 \mid n \in \omega\})$.

Definition 3.13. *For T-expressions $\sigma : [\![H, x : s]\!] \to A$ and $\tau : [\![H]\!] \to [\![s]\!]$, we define the T-expression $\tau + \sigma : [\![H]\!] \to A$ by*

- $\tau + \pi_i = \tau$
- $\tau + \mathsf{K}n = \mathsf{K}n$
- $\tau + (\mathsf{Ap} \circ \langle \sigma, \nu \rangle) = \mathsf{Ap} \circ \langle \tau + \sigma, \tau + \nu \rangle$
- $\tau + \Lambda_A(\sigma) = \Lambda_A(\tau + \sigma)$
- $\tau + (\chi \circ \langle \sigma, \langle \nu_n \mid n \in \omega \rangle \rangle) = \chi \circ \langle \tau + \sigma, \langle \tau + \nu_n \mid n \in \omega \rangle \rangle$.

Definition 3.14 (Ap-reduction).

$$\dfrac{\dfrac{\begin{matrix}[\![H,x:s]\!]\\ \vdots\,\sigma\\ [\![t]\!]\end{matrix}}{[\![s]\!]\Rightarrow[\![t]\!]}\Lambda_{[\![s]\!]} \qquad \begin{matrix}[\![s]\!]\\ \vdots\\ \tau\ [\![H]\!]\end{matrix}}{[\![t]\!]}\mathsf{Ap} \qquad \mapsto \qquad \tau+\sigma$$

Definition 3.15 (R-normal expressions). *The expression σ is r-normal if one of the following conditions holds:*

- *for each occurrence of $\chi \circ \langle \sigma', \langle \tau_n \mid n \in \omega \rangle \rangle$ in σ, the expressions τ_n end with χ or $\tau_n \equiv \mathsf{K}n$*
- *for each occurrence of $\mathsf{Ap} \circ \langle \tau, \nu \rangle$ in σ, the expression τ has the form $\Lambda_{\bar{D}}(\chi \circ \langle \nu', \langle \tau'_n \mid n \in \omega \rangle \rangle)$ or $\tau \equiv \mathsf{K}n$*
- *σ itself has the form $\Lambda_{\bar{D}}(\chi \circ \langle \nu, \langle \tau_n \mid n \in \omega \rangle \rangle)$ or $\tau \equiv \mathsf{K}n$*

Lemma 3.16. *Any r-normal expression reduces to an r-normal expression by a single application of the* Ap*-reduction.*

Proof. Immediate from the definitions of $\tau + \sigma$ and r-expressions. □

Definition 3.17 (χ-reduction).

$$\dfrac{\begin{array}{c}[\![H]\!]\\ \vdots\, n\\ N\end{array} \quad \begin{array}{c}[\![H]\!]\\ \vdots\, \tau_m\\ N\end{array}}{N}\,\chi \quad \mapsto \quad \begin{array}{c}[\![H]\!]\\ \vdots\, \tau_n\\ N\end{array}$$

$$\dfrac{\dfrac{\begin{array}{c}[\![H]\!]\\ \vdots\, \sigma\\ N\end{array} \quad \begin{array}{c}[\![H]\!]\\ \vdots\, \tau_n\\ N\end{array}}{N}\,\chi \quad \begin{array}{c}[\![H]\!]\\ \vdots\, \nu_m\\ N\end{array}}{N}\,\chi \quad \mapsto \quad \dfrac{\begin{array}{c}[\![H]\!]\\ \vdots\, \sigma\\ N\end{array} \quad \dfrac{\begin{array}{c}[\![H]\!]\\ \vdots\, \tau_n\\ N\end{array} \quad \begin{array}{c}[\![H]\!]\\ \vdots\, \nu_m\\ N\end{array}}{N}\,\chi}{N}\,\chi\,.$$

The soundness of χ-reductions are guaranteed by Proposition 3.9 and 3.10.

Lemma 3.18. *Any r-normal expression reduces to an r-normal expression by a single application of the χ-reduction without creating any* Ap*-redexes.*

Proof. Immediate from the definitions of r-expressions and χ-reduction. □

Lemma 3.19. *Let σ be the expression $[\![H \rhd M : t]\!]$, then $|\sigma| \leq \omega + n$ for some $n \in \omega$.*

Proof. By induction. □

Lemma 3.20. *Let* $\mathtt{Ap} \circ \langle \sigma, \tau \rangle \mapsto \sigma'$ *by a single application of the* Ap*-reduction. Then $|\sigma'| \leq |\tau| + |\sigma|$.*

Proof. By induction, using the continuity of $\lambda\gamma.\, \alpha + \gamma$. □

Lemma 3.21. *The expression $[\![H \rhd M : t]\!]$ is r-normal.*

Proof. By inspection and induction. □

Theorem 3.22. *Let the expression $[\![H \rhd M : t]\!]$ reduce to σ by the successive* Ap*-reductions and σ has no* Ap*-redex. Then $|\sigma| < \varepsilon_0$ and σ is r-normal.*

Proof. By the standard argument, using the induction on the degrees and ranks. □

Definition 3.23. *The expressions without any χ-redexes are called χ-normal.*

Lemma 3.24. *Let α, ν_m $(m \in \omega)$ be χ-normal, and the expression*

$$\dfrac{\begin{array}{c}[\![H]\!]\\ \vdots\, \alpha\\ N\end{array} \quad \begin{array}{c}[\![H]\!]\\ \vdots\, \nu_m\\ N\end{array}}{N}\,\chi$$

be a χ-redex. Then this expression can be reduced to a χ-normal expression with the rank $\leq |\alpha| \,\sharp\, \sup_m |\nu_m|$, where $\sharp$ is the natural sum.

Proof. By induction on $|\alpha|$. $\square$

Theorem 3.25. *Any expression σ can be reduced to a χ-normal expression with the rank $\leq 2^{|\sigma|+1}$.*

Proof. By induction on $|\sigma|$, using the previous lemma. $\square$

Theorem 3.26. *Any expression $[\![H \rhd M : t]\!]$ can be reduced to the expression which is r-normal and χ-normal with the rank $< \varepsilon_0$.*

Proof. Immediate. $\square$

Corollary 3.27. *The decomposition of $[\![H \rhd M : t]\!]$ ends up with a well-founded tree of the rank $< \varepsilon_0$.*

Proof. The r-normal and χ-normal expression has the same form as the decomposition tree. The result then follows from the unicity of decomposition. $\square$

4 The games and the *Dialectica* interpretation

4.1 The *Dialectica* interpretation

In the *Dialectica* interpretation, each formula A in HA is interpreted as a $\exists\forall$-sentence A^G quantifying over the primitive recursive functionals of finite type.

The sentence has A^G the form

$$\exists\vec{x}\forall\vec{y}[A^g(\vec{x},\vec{y},\vec{u}) \mapsto 0]$$

where:

- $\vec{x}$ stands for a finite sequence of variables;
- $\vec{u}$ is the list of all free variables of type **nat**;
- $A^g(\vec{x},\vec{y},\vec{u})$ is a term of T of type **nat**; it is understood that 0 means the truth and others the falsity.

For $\vec{X} = (X_0, \dots, X_n)$, let $\vec{X}\vec{u}$ stands for the list $(X_0\vec{u}, \dots, X_n\vec{u})$. We say that A^G is *true* if and only if

- if $\vec{X}$ is non-empty, there exist closed terms $\vec{X}$ of T such that for all closed $\vec{Y}$ and $\vec{U}$,
$$[\vec{U}/\vec{u}][\vec{X}\vec{u}/\vec{x}][\vec{Y}/\vec{y}]A^g(\vec{x},\vec{y},\vec{u}) \mapsto 0,$$
- otherwise; for all closed $\vec{Y}$ and $\vec{U}$,
$$[\vec{U}/\vec{u}][\vec{Y}/\vec{y}]A^g(\vec{x},\vec{y},\vec{u}) \mapsto 0.$$

The truth functions, the addition, the multiplication, the characteristic function of equality on natural numbers and the case constructor ("if ... then ... else ... ") are all definable in T. Using them, A^G is defined inductively as follows. We suppress all the free variables in this definition.

- $x^G \equiv x^{\mathbf{nat}}$,
- $0^G \equiv 0$,
- $(s + t)^G \equiv s^G + t^G$,
- $(s \times t)^G \equiv s^G \times t^G$,
- $(s = t)^G \equiv [s^G = t^G \mapsto 0]$.

Let $A^G = \exists \vec{x} \forall \vec{y} [A^g(\vec{x}, \vec{y}) \mapsto 0]$ and $B^G = \exists \vec{u} \forall \vec{v} [B^g(\vec{u}, \vec{v}) \mapsto 0]$.

- $(A \wedge B)^G \equiv \exists \vec{x} \exists \vec{u} \forall \vec{y} \forall \vec{v} [A^g(\vec{x}, \vec{y}) \wedge B^g(\vec{u}, \vec{v}) \mapsto 0]$
- $(A \vee B)^G \equiv \exists \vec{x} \exists \vec{u} \exists d \forall \vec{y} \forall \vec{v} [(A^g(\vec{x}, \vec{y}) \wedge d = 0) \vee (B^g(\vec{u}, \vec{v}) \wedge d = 1) \mapsto 0]$
- $(\neg A)^G \equiv \exists \vec{Y} \forall \vec{x} [\neg A^g(\vec{x}, \vec{Y} \vec{x}) \mapsto 0]$
- $(A \supset B)^G \equiv \exists \vec{U} \exists \vec{Y} \forall \vec{x} \forall \vec{v} [A^g(\vec{x}, \vec{Y} \vec{x} \vec{v}) \supset B^g(\vec{U} \vec{x}, \vec{v}) \mapsto 0]$
- $(\exists z\, A)^G \equiv \exists z \exists \vec{x} \forall \vec{y} [A^g(\vec{x}, \vec{y}) \mapsto 0]$
- $(\forall z\, A)^G \equiv \exists \vec{X} \forall z \forall \vec{y} [A^g(\vec{X} z, \vec{y}) \mapsto 0]$

Gödel then proved that for any formula A in HA, if it is provable in HA, then A^G is true in the above sense.

4.2 Gödel's heuristics

In this part, we follow the exposition of S. Feferman [4]. After his second incompleteness theorem, Gödel's concern was to give the constructive foundation to mathematics. In the Yale lecture in 1941, he presented what would later amounts to the *Dialectica* interpretation, and he then gave three criteria for constructivity:

1. All primitive (undefined) functions ... must be calculable for any given arguments and all primitive relations must be decidable for any given arguments.
2. Existential assertions must have a meaning only as abbreviations for actual construction ...
3. Universal propositions can be negated in the sense that a counter example exists in the sense just described ... Therefore, leaving out abbreviations, universal propositions can't be negated at all ...

He then proceeded to explain the heuristics to arrive at his functional interpretation, in particular, the most important one, *i.e.* the interpretation of implication. Let us consider the formula

$$\exists x \forall y A(x,y) \supset \exists u \forall v B(u,v), \tag{1}$$

where $A(x,y)$ and $B(u,v)$ are quantifier-free. In Gödel's view of constructivity, this sentence can only mean

$$\forall x \exists u [\forall y A(x,y) \supset \forall v B(u,v)], \tag{2}$$

although this equivalence is not accepted in intuitionistic logic. The second sentence is then converted into

$$\exists U \forall x [\forall y A(x,y) \supset \forall v B(Ux,v)]. \tag{3}$$

The formula inside the square bracket is then interpreted as a claim about counterexamples. For this, we first consider its contrapositive

$$\neg \forall v B(Ux,v) \supset \neg \forall y A(x,y) \tag{4}$$

This is understood as the claim that one can convert any counterexample to $\forall v B(Ux,v)$ to a counterexample of $\forall y A(x,y)$. Hence

$$\exists Y' \forall v [\neg B(Ux,v) \supset \neg A(x,Y'v)] \tag{5}$$

Since the formulas $B(Ux,v)$ and $A(x,Y'v)$ are taken to be decidable, this is equivalent to

$$\exists Y \forall v [A(x,Y'v) \supset B(Ux,v)] \tag{6}$$

Hence, (1) becomes

$$\exists U \forall x \exists Y' \forall v [A(x,Y'v) \supset B(Ux,v)] \tag{7}$$

and the choice principle again yields

$$\exists U \exists Y \forall x \forall v [A(x,Yxv) \supset B(Ux,v)]. \tag{8}$$

4.3 The game-theoretical reading of *Dialectica*

From Gödel's above argument, we can intuitively understand that $\vec{y}$ in

$$A^G \equiv \exists \vec{x} \forall \vec{y} [A^g(\vec{x},\vec{y},\vec{u}) \mapsto 0]$$

are supposed to be filled in with functions trying to falsify A^g, and $\vec{x}$ with functions trying to verify it, i.e.,

- the existential quantification corresponds to the Player's strategy trying to verify A,

- the universal quantification corresponds to the Opponent's counter strategy trying to falsify A.

Then what the *Dialectica* interpretation claims is that if A is provable, then the Player has the winning strategy in the verification game for A against any counter strategy.

Such a game for the first-order classical logic is known as the Hintikka game. Hence we seem to be able to understand the game semantics of T as the constructive version of Hintikka game for arithmetic. We have been working on this account by restating the game semantics for T in terms of cut-elimination.

5 HA and beyond

Given the game-theoretical interpretation of HA, we can expect the extension of game semantics to various systems of logic. The *Dialectica* interpretation itself is extended to the various fragments of second order logic, and we expect their game-theoretical counterparts.

From the other direction, there are works on the realizability interpretation of the excluded middle by Berardi, Coquant and others, which can be stated in terms of games. Is there any connection with our game semantics of T? Furthermore, the axiomatic set theory is formalized in the first-order logic. Can we extend the game semantics to, say, ZF?

In general, the game semantics has a strong proof-theoretic flavor. This may make one wonder if it is worth the name "semantics." On the other hand, we believe that more and more people are beginning to feel the frustration with the denotational (or model-theoretic) semantics, which have been developed in the tradition of Tarski.

It seems that the game semantics points to a certain middle ground between proof theory and model theory, and it may be only in this middle ground that the notion of "meaning" can be truly studied.

References

[1] S. Abramsky, R. Jagadeesan and P Malacaria. " Full abstraction for PCF." (extended abstract) *TACS'94*, LNCS 789, 1-15, Springer.

[2] S. Abramsky, R. Jagadeesan and P Malacaria. " Full abstraction for PCF." (manuscript) December, 1995.

[3] J.Avigad and S. Feferman. "Gödel's Functional ("Dialectica") Interpretation." *Handbook of Proof Theory*, ed. by S. Buss, North Holland, 1998.

[4] "Gödel's Dialectica interpretation in two-way stretch." *Computational Logic and Proof Theory*, LNCS 713, 23-40, Springer.

[5] W. Pohlers. *Proof Theory, An Introduction.* LNM 1407, Springer, 1989.