

Light Monotone Dialectica

Extraction of moduli of uniform continuity for closed terms from
Goedel's **T** of type $(\mathbb{N} \Rightarrow \mathbb{N}) \Rightarrow (\mathbb{N} \Rightarrow \mathbb{N})$

Mircea-Dan Hernest

Project LogiCal – Paris, France and GKLI – Munich, Germany

LFMTP'06 Talk in Seattle, 16 August 2006

Outline of this Talk

1 Introduction

- A seemingly simple game-problem of discrete mathematics
- Generalization and set-up for the Proof-theoretic machinery

2 Hereditarily Extensional Equality in computer system MinLog

- Term system, majorizability and Hereditarily Extensional Equality
- Weakly extensional monotonic Arithmetic for Gödel functionals

3 From Gödel's Dialectica to Light Monotone Dialectica

- The pure Gödel's functional "Dialectica" interpretation
- The Contraction Problem → Achilles' heel for Dialectica !
- The Light Monotone Dialectica majorant extraction

4 Conclusions and Future work

- The human/computer outcome for our general game-problem
- Computer really necessary? Implementing Monotone Dialectica
- Work to be done → the real "Light" Monotone Dialectica

A game-problem of discrete mathematics

① Let $f, g, h : \mathbb{N} \mapsto \mathbb{N}$ s.t. h fixed & $\forall n \in \mathbb{N}. f(n) \leq h(n) \wedge g(n) \leq h(n)$

Given $k \in \mathbb{N}$ find $L \in \mathbb{N}$ s.t. $[\forall i \leq L. f(i) = g(i)] \Rightarrow [\forall j \leq k. f(j) = g(j)]$

② Dummy answer $L := k$. Hence try to complicate the demand:

$$\forall f, g \leq_{\mathbb{N} \rightarrow \mathbb{N}} h. [\forall i \leq L. f(i) = g(i)] \Rightarrow [\forall j \leq k. f(f(j)) = g(g(j))]$$

③ Simple optimal answer $\max\{k, h(0), \dots, h(k)\}$. But what about:

$$\forall f, g \leq h. [\forall i \leq L. f(i) = g(i)] \Rightarrow [\forall j \leq k. f(f(f(j))) = g(g(g(j)))]$$

④ Temptation $\max\{k, h(0), \dots, h(k), h(h(0)), \dots, h(h(k))\}$. **False**, since $f(j) \leq h(j) \not\Rightarrow f(f(j)) \leq h(h(j))$, hence $f(f(0)) > h(h(j))$ possible.

⑤ **How to solve this?** And what about the fully general case?

$$\forall f, g \leq_{\mathbb{N} \rightarrow \mathbb{N}} h. [\forall i \leq L. f(i) = g(i)] \Rightarrow [\forall j \leq k. f^{(m)}(j) = g^{(m)}(j)]$$

Set-up for the Proof-theoretic machinery (1/2)

- 1 Extract moduli of uniform continuity for *closed* terms \mathbf{t}_m of Goedel's \mathbf{T} of type $(\mathbb{N} \Rightarrow \mathbb{N}) \Rightarrow (\mathbb{N} \Rightarrow \mathbb{N})$ where

$$\mathbf{t}_m \equiv \lambda h^{\mathbb{N} \Rightarrow \mathbb{N}}, k^{\mathbb{N}}. \mathbf{R}_{\mathbb{N} \Rightarrow \mathbb{N}}(0)[\lambda p, q. h^{(m)}(p) + q](k + 1)$$

- 2 Hence $\mathbf{t}_m(h, k) \equiv h^{(m)}(0) + h^{(m)}(1) + \dots + h^{(m)}(k)$. How???

$$\text{Let } \mathcal{P} \equiv \vdash \forall f, g. [\forall i. f(i) =_{\mathbb{N}} g(i)] \Rightarrow [\forall j. \mathbf{t}_m(f, j) =_{\mathbb{N}} \mathbf{t}_m(g, j)]$$

- 3 The above \mathcal{P} is a Minimal Logic proof of (almost) $\mathbf{t}_m \approx \mathbf{t}_m$. We apply on \mathcal{P} a Light Monotone Dialectica extraction in MinLog.
- 4 Gödel's Dialectica would give an exact realizer $\mathbf{t}'[f, g, j]$ for i s.t.

$$\forall f, g \forall j. f(\mathbf{t}'[f, g, j]) =_{\mathbb{N}} g(\mathbf{t}'[f, g, j]) \Rightarrow \mathbf{t}_m(f, j) =_{\mathbb{N}} \mathbf{t}_m(g, j)$$

- 5 If $\tilde{\mathbf{t}} \text{ maj } \lambda f, g, j. \mathbf{t}'$ then for $k \geq j$, $h^* \text{ maj } h$ and $h \geq f, g$ one has $(L \equiv \tilde{\mathbf{t}}(h^*, h^*, k)) \geq \mathbf{t}'[f, g, j]$ and therefore such an L is a **solution**:

$$\forall h \forall f, g \leq_{\mathbb{N} \rightarrow \mathbb{N}} h \forall k. [\forall i \leq L. f(i) = g(i)] \Rightarrow [\forall j \leq k. f^{(m)}(j) = g^{(m)}(j)]$$

Set-up for the Proof-theoretic machinery (2/2)

- Start from proof of *hereditarily extensional equality* of \mathbf{t} to itself.
- Hence a proof of $\mathbf{t} \approx_{(\mathbb{N} \Rightarrow \mathbb{N}) \Rightarrow (\mathbb{N} \Rightarrow \mathbb{N})} \mathbf{t}$ in system Z_0 of Berger-Buchholz-Schwichtenberg, the base logic of machine system MinLog.
- Hence a Minimal Logic proof without use of Extensionality Axiom.
- Two *extreme* approaches:
 - 1 First extract \mathbf{t}' by Gödel's Dialectica and then majorize it via Howard's algorithm (Kohlenbach's PhD thesis, JSL paper '92).
 - 2 Directly extract $\tilde{\mathbf{t}}$ by producing a majorant for the *closed* extracted term at each of the Dialectica recursion step (Kohlenbach '93).
- None of the two efficient on the computer. **Solution:** use an intermediate approach \rightarrow Extract partial majorants which are not necessarily closed terms, only simplify treatment of Contraction.
- Also use a Normalization during Extraction, i.e. NbE-normalize the extracted term of the conclusion of a Modus Ponens. (**NdE**)
- **Huge** impact of such Partial Evaluation. No solution without it!!

The term system – a lambda-variant of Gödel's T

- 1 All finite types generated from \mathbb{N} and \mathbb{B} by the rule $\sigma, \tau \mapsto (\sigma\tau)$
- 2 $\#^{\mathbb{B}}, \#^{\mathbb{B}}$ equality $=^{\mathbb{N}\mathbb{N}\mathbb{B}}$ and inequality $\geq^{\mathbb{N}\mathbb{N}\mathbb{B}}$, maximum $Max^{\mathbb{N}\mathbb{N}\mathbb{N}}$
- 3 $0^{\mathbb{N}}$ (zero), $Suc^{\mathbb{N}\mathbb{N}}$ (successor) and Gödel's recursor $\mathbf{R}_\tau^{\tau(\mathbb{N}\tau\tau)\mathbb{N}\tau}$
 $And^{\mathbb{B}\mathbb{B}\mathbb{B}} := \lambda p, q. \mathbf{If}_{\mathbb{B}} p q \mathit{ff}$ $Imp^{\mathbb{B}\mathbb{B}\mathbb{B}} := \lambda p, q. \mathbf{If}_{\mathbb{B}} p q \mathit{tt}$
- 4 Combinators at all types are *defined* in terms of λ -abstraction:
 $\Sigma := \lambda x, y, z. x z (y z)$ $\Pi := \lambda x, y. x$
- 5 $at^{\mathbb{B}}$ is the *unique* predicate symbol of WeZ_m^{\exists} – one \mathbb{B} argument
- 6 *Extensionally* defined equality and inequality (below $\sigma \in \{\mathbb{B}, \mathbb{N}\}$)

$$s =_{\mathbb{N}} t := at(= s t) \quad s =_{\mathbb{B}} t := at(s) \leftrightarrow at(t)$$

$$s \geq_{\mathbb{N}} t := at(\geq s t) \quad s \geq_{\mathbb{B}} t := at(t) \rightarrow at(s)$$

$$s =_{\sigma_1 \dots \sigma_n \rightarrow \sigma} t := \forall x_1^{\sigma_1} \dots x_n^{\sigma_n} (s x_1 \dots x_n =_{\sigma} t x_1 \dots x_n)$$

$$s \geq_{\sigma_1 \dots \sigma_n \rightarrow \sigma} t := \forall x_1^{\sigma_1} \dots x_n^{\sigma_n} (s x_1 \dots x_n \geq_{\sigma} t x_1 \dots x_n)$$

Majorizability and Hereditarily Extensional Equality (1)

$$x \text{ maj}_{\mathbf{N}} y \quad :\equiv \quad x \geq_{\mathbf{N}} y \quad :\equiv \quad at(\geq x^{\mathbf{N}} y^{\mathbf{N}})$$

$$x \geq_{\sigma\tau} y \quad \equiv \quad \forall z^{\sigma} (xz \geq_{\tau} yz)$$

$$x \text{ maj}_{\sigma\tau} y \quad :\equiv \quad \forall z_1^{\sigma}, z_2^{\sigma} (z_1 \text{ maj}_{\sigma} z_2 \rightarrow xz_1 \text{ maj}_{\tau} yz_2)$$

$$0 \text{ maj}_{\mathbf{N}} 0, \text{ Suc } \text{ maj}_{\mathbf{NN}} \text{ Suc}, \quad \boxed{\Sigma \text{ maj } \Sigma, \Pi \text{ maj } \Pi \text{ and } \mathbf{R}^M \text{ maj } \mathbf{R}}$$

$$\text{WeZ}_m \vdash t^* \text{ maj}_{\sigma\tau} t \wedge s^* \text{ maj}_{\sigma} s \implies t^* s^* \text{ maj}_{\tau} ts$$

$$x \approx_{\mathbf{N}} y \quad :\equiv \quad x =_{\mathbf{N}} y \quad :\equiv \quad at(= x^{\mathbf{N}} y^{\mathbf{N}})$$

$$x =_{\sigma\tau} y \quad \equiv \quad \forall z^{\sigma} (xz =_{\tau} yz)$$

$$x \approx_{\sigma\tau} y \quad :\equiv \quad \forall z_1^{\sigma}, z_2^{\sigma} (z_1 \approx_{\sigma} z_2 \rightarrow xz_1 \approx_{\tau} yz_2)$$

$$0 \approx_{\mathbf{N}} 0, \text{ Suc } \approx_{\mathbf{NN}} \text{ Suc}, \quad \boxed{\Sigma \approx \Sigma, \Pi \approx \Pi \text{ and } \mathbf{R} \approx \mathbf{R}}$$

$$\text{WeZ}_m \vdash t^* \approx_{\sigma\tau} t \wedge s^* \approx_{\sigma} s \implies t^* s^* \approx_{\tau} ts$$

System $WeZ_m \rightarrow$ Implic. Introd. with Contraction

① WeZ_m - Weakly extensional Minimal Arithmetic with \geq and Max

② Minimal Arith. \Leftrightarrow Heyting Arith. in all finite types $HA^\omega \setminus \perp \rightarrow A$

③ WeZ_m - underlying Logic is Natural Deduction, not Hilbert-style!

④ $\frac{[u : A] \dots / B}{A \rightarrow B} \rightarrow^+$, particular set of instances of A in the same

parcel (assumption variable) u get discharged; If at least two A get discharged then one has *logical Contraction*; If moreover A contains at least one positive universal or a negative existential quantifier then one has a *computationally relevant Contraction*

⑤ Comp. Relevance relative to both Gödel and Monotone Dialectica

$$\{A_D(z; T_i(\underline{z}, \underline{x}, y))\}_{i=1}^{n+1}, \{C_D^i(x_i; T_i(\underline{z}, \underline{x}, y))\}_{i=n+2}^m \vdash B_D(T(\underline{z}, \underline{x}); y)$$

Same tuple z produced by $2 \leq n+1 \leq m$ discharged instances of A

If $\{T_i\}_{i=1}^{n+1}$ non-null (A is Dialectica relevant) \Rightarrow *Equalization* is a must!

Gödel's functional "Dialectica" interpretation

- 1 A translation of proofs which includes a translation of formulas.
- 2 $A(\underline{a}) \mapsto A^D \equiv \exists \underline{x} \forall \underline{y} A_D(\underline{x}; \underline{y}; \underline{a})$ with \underline{a} all free vars of formula A
- 3 A_D is quantifier-free for Gödel's Dialectica, since decidability needed \rightarrow this no longer for Monotone setup \Rightarrow *Bounded Dialectica*
- 4 Recursive syntactic translation from proofs in *Constructive Arithmetic* (or *Classical Arithmetic*, modulo the double-negation translation) to proofs in *Intuitionistic Arithmetic* such that positive occurrences of \exists and negative occurrences of \forall in the proof's conclusion get actually realized by terms in Gödel's \mathbf{T} .
- 5 Contraction Problem: \rightarrow choose between a number of realizers according to a boolean term associated to the contraction formula;
Diller-Nahm: \rightarrow postpone all choices to the very end by collecting all candidates and making a single final global choice;
Monotone Dialectica: \rightarrow use a simple common upper bound (maximum majorant) of the candidates \implies extract *majorants*

The Light Dialectica interpretation of formulas

$$A^D \quad ::= \quad (A_D \quad ::= \quad A) \text{ for prime formulas } A$$

$$(A \wedge B)^D \quad ::= \quad \exists \underline{x}, \underline{u} \forall \underline{y}, \underline{v} [(A \wedge B)_D \quad ::= \quad A_D(\underline{x}; \underline{y}; \underline{a}) \wedge B_D(\underline{u}; \underline{v}; \underline{b})]$$

$$(A \rightarrow B)^D \quad ::= \quad \exists \underline{Y}, \underline{U} \forall \underline{x}, \underline{v} [(A \rightarrow B)_D \quad ::= \quad A_D(\underline{x}; \underline{Y}(\underline{x}, \underline{v})) \rightarrow B_D(\underline{U}(\underline{x}); \underline{v})]$$

$$(\exists z A(z, \underline{a}))^D \quad ::= \quad \exists z^\dagger, \underline{x} \forall \underline{y} [(\exists z A(z, \underline{a}))_D(z^\dagger, \underline{x}; \underline{y}; \underline{a}) \quad ::= \quad A_D(\underline{x}; \underline{y}; z^\dagger, \underline{a})]$$

$$(\exists z A(z, \underline{a}))^D \quad ::= \quad \exists \underline{x} \forall \underline{y} [(\exists z A(z, \underline{a}))_D(\underline{x}; \underline{y}; \underline{a}) \quad ::= \quad \exists z A_D(\underline{x}; \underline{y}; z, \underline{a})]$$

$$(\forall z A(z, \underline{a}))^D \quad ::= \quad \exists \underline{X} \forall z^\dagger, \underline{y} [(\forall z A(z, \underline{a}))_D(\underline{X}; z^\dagger, \underline{y}; \underline{a}) \quad ::= \quad A_D(\underline{X}(z^\dagger); \underline{y}; z^\dagger, \underline{a})]$$

$$(\forall z A(z, \underline{a}))^D \quad ::= \quad \exists \underline{x} \forall \underline{y} [(\forall z A(z, \underline{a}))_D(\underline{x}; \underline{y}; \underline{a}) \quad ::= \quad \forall z A_D(\underline{x}; \underline{y}; z, \underline{a})]$$

Here $\cdot \mapsto \cdot^\dagger$ is a mapping which assigns to every given variable z a completely new variable z^\dagger which has the same type of z .

Exact realizer synthesis by Dialectica Interpretation

Extraction and Soundness Theorem: There exists an algorithm which, given at input a $WeZ^{\exists+}$ proof $\mathcal{P} : \{C^i\}_{i=1}^n \vdash A$ [hence of the conclusion formula A , from the *undischarged* assumption formulas $\{C^i\}_{i=1}^n$] will produce at output **1)** the tuples of terms T and $\{T_i\}_{i=1}^n$ **2)** the tuples of variables $\{x_i\}_{i=1}^n$ and y **3)** the verifying proof

$$\mathcal{P}_D : \{C_D^i(x_i; T_i(\underline{x}, y))\}_{i=1}^n \vdash A_D(T(\underline{x}); y)$$

– where $\underline{x} := x_1, \dots, x_n$. Moreover,

- 1) variables \underline{x} and y are all completely new (not occur in \mathcal{P})
- 2) the free variables of T and $\{T_i\}_{i=1}^n$ are among the free variables of A and $\{C^i\}_{i=1}^n$ (this one names “the *free variable condition (FVC)* for programs extracted by the Dialectica Interpretation”)

[$\Rightarrow \underline{x}, y$ not occur free in the *extracted* terms $\{T_i\}_{i=1}^n$ and T]

Notice that: Terms T and $\{T_i\}_{i=1}^n$ are not necessarily closed !!!

Problem \rightarrow Implication Introduction with Contraction

$$\boxed{\frac{[u : A] \dots / B}{A \rightarrow B} \rightarrow^+} \quad n \geq 1, \quad \underline{z} \equiv \overbrace{z, \dots, z}^{n+1} \text{ and } \underline{x} \equiv x_{n+2}, \dots, x_m :$$

$$\{A_{\mathbf{D}}(z; T_i(\underline{z}, \underline{x}, y))\}_{i=1}^{n+1}, \{C_{\mathbf{D}}^i(x_i; T_i(\underline{z}, \underline{x}, y))\}_{i=n+2}^m \vdash B_{\mathbf{D}}(T(\underline{z}, \underline{x}); y)$$

- 1) Same tuple z produced by $n + 1 \leq m$ discharged instances of A
- 2) Case: tuples $\{T_i\}_{i=1}^{n+1}$ are non-null! Recall that $A_{\mathbf{D}}$ is quantifier-free
- 3) Since $\{T_i\}_{i=1}^{n+1}$ non-null \implies their *equalization* is a *must*:

$$\mathbf{S} := \lambda \underline{x}, z, y. \mathbf{If}_\tau^n(\iota_A^{\mathbf{D}}[z; T^1], \dots, \iota_A^{\mathbf{D}}[z; T^n], T_{n+1}(\underline{z}, \underline{x}, y), T^n, \dots, T^1)$$

one can now cancell all $\{A_{\mathbf{D}}\}_{i=1}^{n+1}$ by a single \rightarrow^+ in the verifying proof

$$\{A_{\mathbf{D}}(z; \mathbf{S}(\underline{x}, z, y))\}_{i=1}^{n+1}, \{C_{\mathbf{D}}^i(x_i; S_i(\underline{x}, z, y))\}_{i=n+2}^m \vdash B_{\mathbf{D}}(S(\underline{x}, z); y)$$

$$\{C_{\mathbf{D}}^i(x_i; S_i(\underline{x}, z, y))\}_{i=n+2}^m \vdash A_{\mathbf{D}}(z; \mathbf{S}(\underline{x}, z, y)) \rightarrow B_{\mathbf{D}}(S(\underline{x}, z); y)$$

The Light Monotone Dialectica program extraction

Majorant realizer synthesis by Light Monotone Dialectica

Theorem: There ex. an algorithm which, given at input a $WeZ_m^{\exists+}$ proof $\mathcal{P} : \{C^i(a_i)\}_{i=1}^n \vdash A(a')$ [hence of the conclusion formula A , whose free variables form the *tuple* a , from the *undischarged* assumption formulas $\{C^i\}_{i=1}^n$] it will produce at output the following ($\underline{a} := a_1, \dots, a_n, a'$):

- 1 tuples of terms $\{T_i[\underline{a}]\}_{i=1}^n$ and $T[\underline{a}]$, with free variables among \underline{a}
- 2 the tuples of variables $\{x_i\}_{i=1}^n$ and y , all together with
- 3 the following verifying proof in WeZ_m^{\exists} (below let $\underline{x} := x_1, \dots, x_n$):

$$\vdash \exists Y_1, \dots, Y_n, X [\bigwedge_{i=1}^n (\lambda \underline{a}. T_i) \text{ maj } Y_i \wedge (\lambda \underline{a}. T) \text{ maj } X \wedge \\ \forall \underline{a}, \underline{x}, y (\{ \bigwedge_{i=1}^n C^i_{\mathbf{D}}(x_i; Y_i(\underline{a}, \underline{x}, y); a_i) \} \rightarrow A_{\mathbf{D}}(X(\underline{a}, \underline{x}); y; a))]$$

Variables \underline{x} and y do not occur in \mathcal{P} (they are all completely new)

$\implies \underline{x}$ and y do not occur free in the *extracted* terms $\{T_i\}_{i=1}^n$ and T .

Majorizability and Hereditarily Extensional Equality (2)

$$x \text{ maj}_{\mathbb{N}} y \quad :\equiv \quad x \geq_{\mathbb{N}} y \quad :\equiv \quad at(\geq x^{\mathbb{N}} y^{\mathbb{N}})$$

$$x \geq_{\sigma\tau} y \quad \equiv \quad \forall z^{\sigma} (xz \geq_{\tau} yz)$$

$$x \text{ maj}_{\sigma\tau} y \quad :\equiv \quad \forall z_1^{\sigma}, z_2^{\sigma} (z_1 \text{ maj}_{\sigma} z_2 \rightarrow xz_1 \text{ maj}_{\tau} yz_2)$$

$$0 \text{ maj}_{\mathbb{N}} 0, \text{ Suc } \text{ maj}_{\mathbb{N}\mathbb{N}} \text{ Suc}, \quad \boxed{\Sigma \text{ maj } \Sigma, \Pi \text{ maj } \Pi \text{ and } \mathbf{R}^M \text{ maj } \mathbf{R}}$$

$$\text{WeZ}_m \vdash t^* \text{ maj}_{\sigma\tau} t \wedge s^* \text{ maj}_{\sigma} s \implies t^* s^* \text{ maj}_{\tau} ts$$

$$x \approx_{\mathbb{N}} y \quad :\equiv \quad x =_{\mathbb{N}} y \quad :\equiv \quad at(= x^{\mathbb{N}} y^{\mathbb{N}})$$

$$x =_{\sigma\tau} y \quad \equiv \quad \forall z^{\sigma} (xz =_{\tau} yz)$$

$$x \approx_{\sigma\tau} y \quad :\equiv \quad \forall z_1^{\sigma}, z_2^{\sigma} (z_1 \approx_{\sigma} z_2 \rightarrow xz_1 \approx_{\tau} yz_2)$$

$$0 \approx_{\mathbb{N}} 0, \text{ Suc } \approx_{\mathbb{N}\mathbb{N}} \text{ Suc}, \quad \boxed{\Sigma \approx \Sigma, \Pi \approx \Pi \text{ and } \mathbf{R} \approx \mathbf{R}}$$

$$\text{WeZ}_m \vdash t^* \approx_{\sigma\tau} t \wedge s^* \approx_{\sigma} s \implies t^* s^* \approx_{\tau} ts$$

The WeZ_m proof at input & post-extraction ops.

- 1 Let \mathbf{t}_ρ be a closed term of Gödel's \mathbf{T} . Then $WeZ_m \vdash \mathbf{t} \approx_\rho \mathbf{t}$.
- 2 Let $\mathbf{t}_{(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow (\mathbb{N} \rightarrow \mathbb{N})}$ be a closed \mathbf{T} -term. Since

$$WeZ_m \vdash \forall x^{\mathbb{N} \rightarrow \mathbb{N}}, y^{\mathbb{N} \rightarrow \mathbb{N}} [x =_{\mathbb{N} \rightarrow \mathbb{N}} y \leftrightarrow x \approx_{\mathbb{N} \rightarrow \mathbb{N}} y]$$

(due to weak extensionality + reflexivity) it immediately follows that

$$WeZ_m \vdash \forall x^{\mathbb{N} \rightarrow \mathbb{N}}, y^{\mathbb{N} \rightarrow \mathbb{N}} [x =_{\mathbb{N} \rightarrow \mathbb{N}} y \rightarrow \mathbf{t}(x) =_{\mathbb{N} \rightarrow \mathbb{N}} \mathbf{t}(y)]$$

- 3 Let $\mathbf{t}[\underline{a}]$ be a \mathbf{T} -term with free vars \underline{a} . There exists a corresponding \mathbf{T} -term $\mathbf{t}^*[\underline{a}]$ such that $WeZ_m \vdash \lambda \underline{a}. \mathbf{t}^* \text{ maj } \lambda \underline{a}. \mathbf{t}$. Very simple \mathbf{t}^* construction: just replace each \mathbf{R} in \mathbf{t} with the corresponding \mathbf{R}^M .
- 4 If the type of a is $\mathbb{N} \rightarrow \rho$ then $a^M \text{ maj } a$, hence $\mathbf{t}^*[\underline{a}^M] \text{ maj } \mathbf{t}[\underline{a}]$.
- 5 For a of type $\mathbb{N} \rightarrow \rho$ define $a^M(k) \equiv \text{Max}_\rho(a(0), \dots, a(k))$

MinLog computer output for our game-problem, $m = 3$

1 For $\mathbf{t}_3 := \lambda h^{\mathbb{N} \Rightarrow \mathbb{N}}, k^{\mathbb{N}}. \mathbf{R}_{\mathbb{N} \Rightarrow \mathbb{N}}(0)[\lambda p, q. h^{(3)}(p) + q](k + 1)$ want

$$\forall f, g \leq h. [\forall i \leq L. f(i) = g(i)] \Rightarrow [\forall j \leq k. f(f(f(j))) = g(g(g(j)))]$$

2 The MinLog machine outputs in less than one minute:

$$\lambda h, k. \max\{k, h(0) \dots, h(k), \max\{h(0) \dots h(\max\{h(0) \dots h(k)\})\}\}$$

which immediately rewrites more humanly readable as

$$L^3 := \lambda h, k. \max\{k, h(0), h(1), \dots, h(\max\{k, h(0), h(1), \dots, h(k)\})\}$$

3 Recall that for $m = 2$ and $m = 1$ the (human) outcomes were

$$L^2 := \lambda h, k. \max\{k, h(0), h(1), \dots, h(k)\}$$

$$L^1 := \lambda h, k. k$$

Final human solution for our general game-problem

Pattern can be noticed (by the human!) in the solution of our problem for terms

$$\mathbf{t}_m \equiv \lambda h, k. h^{(m)}(0) + \dots + h^{(m)}(k),$$

with $h^{(m)}(i) \equiv h(h \dots (h(i)))$ s.t. h appears m times on the right side.

$$\tilde{t}_1(h, k) \equiv k$$

$$\tilde{t}_2(h, k) \equiv \max\{k, h(0), \dots, h(\tilde{t}_1(h, k))\}$$

$$\tilde{t}_3(h, k) \equiv \max\{k, h(0), \dots, h(\tilde{t}_2(h, k))\}$$

.....

Immediate inference of the generic (recursive) solution for $m \in \mathbb{N}$:

$$\widetilde{t}_{m+1}(h, k) \equiv \max\{k, h(0), \dots, h(\widetilde{t}_m(h, k))\}$$

Verify that $\widetilde{\mathbf{t}}_m$ is the optimal modulus of uniform continuity for \mathbf{t}_m !

Now an easy exercise for the human ! \implies See my PhD thesis :)

Was the Computer really necessary?

- 1 Maybe not, but what if the problem were more complex/tedious ?
- 2 Certainly helpful for preventing the human error ! **Effectively !**

Implementing Monotone Dialectica

- 1 The “light” variant of Monotone Dialectica is the result of our implementation effort ! Many operations which are “easy” for the human (mathematician) are not really that easy for the machine !
- 2 On the computer, the **Goal** is to produce programs in **normal form** !
Hence improve the Nbe-normalization by its own Partial Evaluation \implies Normalization during Extraction (NdE) \iff NbE-normalize the term extracted for the conclusion of each Modus Ponens .
- 3 Only majorize at Contraction \implies produce a partial majorant which is transformed at the end by replacing each **R** with its corresp. **R^M** .
- 4 Why? Well, some of the **R** may be eliminated during the partial NbE-normalization ... Also use the more clever **R^M**, with just 1 **R** .

A lot of Work to be done . . .

- 1 Completely formalize and explore the limits of *Normalization during Extraction* (NdE) \Rightarrow generic optimization for $(\mathbf{t}_n..(\mathbf{t}_2(\mathbf{t}_1\mathbf{t}_0))..)$
- 2 Completely formalize these ad-hoc optimizations of the computer implementation of Monotone Dialectica and combine with the “light” optimization brought by the use of quantifiers without comp. content
- 3 We suspect that the use of these *ncm* quantifiers may eliminate some of the comput. contractions in the Hered. Ext. Eq. extraction ! This game-problem is already solved for a very particular case only !
- 4 Find other more interesting **T**-terms \mathbf{t}_m , for which the modulus of uniform continuity is far more difficult to find !
- 5 Find other more interesting examples for the *Proof Mining* by the Light (monotone) Dialectica on the Computer !
- 6 Improve the human-interaction side of our Dialectica extraction modules in MinLog, in order to render “MinLog for Dialectica” as an indispensable computer tool even for the more pure mathematically oriented Proof Mining !

Short List of related Papers I



U. Kohlenbach.

Proof Interpretations and the Computational Content of Proofs.
Lecture Course, latest version in the author's web page.



U. Kohlenbach and P. Oliva.

Proof Mining: a systematic way of analysing proofs in
Mathematics.

Proc. of the Steklov Inst. of Mathem., 242:136–164, 2003.



U. Kohlenbach.

Pointwise hereditary majorization and some applications.

Arch. Math. Logic, 31:227–241, 1992.



U. Kohlenbach.

Analysing proofs in Analysis.

In *Logic: from Foundations to Applications*, Keele, 1993, European
Logic Colloquium, pages 225–260. Oxford University Press, 1996.

Short List of related Papers II



M.-D. Hernest.

Light Dialectica program extraction from a classical Fibonacci proof Proceedings of DCM@ICALP'06, ENTCS (2007), 10pp.



M.-D. Hernest. Light Functional Interpretation.

CSL 2005 - In LNCS 3634 pp. 477 – 492, July 2005.



M.-D. Hernest and U. Kohlenbach.

A complexity analysis of functional interpretations.

Theoretical Computer Science, 338(1-3):200–246, 2005.



U. Berger.

Uniform Heyting Arithmetic.

Annals of Pure and Applied Logic, 133(1-3):125–148, 2005.



U. Berger, W. Buchholz, and H. Schwichtenberg.

Refined program extraction from classical proofs.

Annals of Pure and Applied Logic, 114:3–25, 2002.