

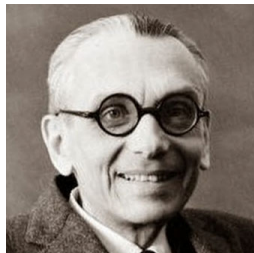
Sequential algorithms as a Dialectica interpretation

Valentin Blot

INRIA - LS2N
Univ. Nantes, France

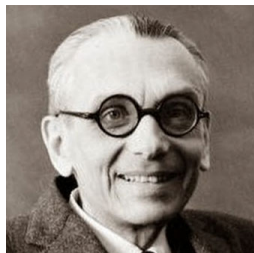
Gödel's Dialectica

- ▶ Incompleteness theorems
 no absolute consistency
 → relative consistency?



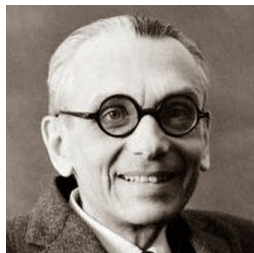
Gödel's Dialectica

- ▶ Incompleteness theorems
no absolute consistency
→ relative consistency?
- ▶ Dialectica (1941):
reduce consistency of arithmetic
to termination of system T
(simply-typed λ -calc. + rec. on nat)



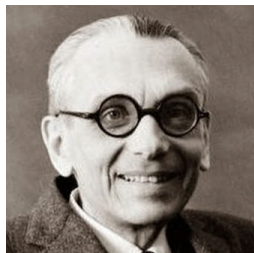
Gödel's Dialectica

- ▶ Incompleteness theorems
no absolute consistency
→ relative consistency?
- ▶ Dialectica (1941):
reduce consistency of arithmetic
to termination of system T
(simply-typed λ -calc. + rec. on `nat`)
- ▶ Why the name?
Name of the journal it was published in (1958)



Gödel's Dialectica

- ▶ Incompleteness theorems
no absolute consistency
→ relative consistency?
- ▶ Dialectica (1941):
reduce consistency of arithmetic
to termination of system T
(simply-typed λ -calc. + rec. on nat)
- ▶ Why the name?
Name of the journal it was published in (1958)

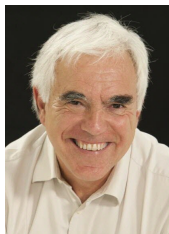


$$\frac{\dots}{\vec{x} \vdash_{HA} A} \quad \rightsquigarrow \quad M : \underline{A} \text{ s.t. } \frac{\dots}{\vec{x} \vdash_T A_D(M, \vec{x})}$$

where $\underline{A}$ is a simple type and A_D is quantifier-free.

Berry-Curien's sequential algorithms

- ▶ Scott domains:
model of languages with general recursion



Berry-Curien's sequential algorithms

- ▶ Scott domains:
model of languages with general recursion

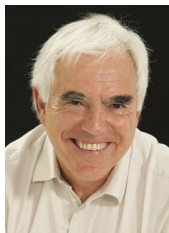
- ▶ But not sequential, the model contains:

$\text{unit} \times \text{unit} \rightarrow \text{unit}$

$\Omega, \quad () \quad \mapsto ()$

$() , \quad \Omega \quad \mapsto ()$

$\Omega, \quad \Omega \quad \mapsto \Omega$



Berry-Curien's sequential algorithms

- ▶ Scott domains:
model of languages with general recursion

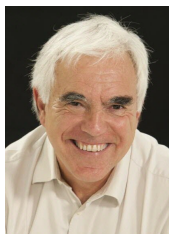
- ▶ But not sequential, the model contains:
 $\text{unit} \times \text{unit} \rightarrow \text{unit}$

$$\Omega, \quad () \quad \mapsto ()$$

$$(), \quad \Omega \quad \mapsto ()$$

$$\Omega, \quad \Omega \quad \mapsto \Omega$$

- ▶ Sequential algorithms (1982)
intensionality
eliminates this program from the model



Berry-Curien's sequential algorithms

- ▶ Scott domains:
model of languages with general recursion

- ▶ But not sequential, the model contains:

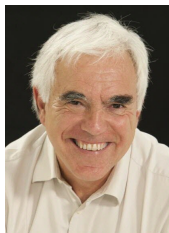
$\text{unit} \times \text{unit} \rightarrow \text{unit}$

$\Omega, \quad () \quad \mapsto ()$

$() , \quad \Omega \quad \mapsto ()$

$\Omega, \quad \Omega \quad \mapsto \Omega$

- ▶ Sequential algorithms (1982)
intensionality
eliminates this program from the model
- ▶ precursor of game semantics



A striking similarity

Interpretation of function types

► Dialectica:

$$\underline{A \rightarrow B} \equiv (\underline{A} \rightarrow \underline{B}) \times (\underline{A} \times \overline{B} \rightarrow \overline{A})$$

$\underline{A}$ = witnesses of A $\overline{A}$ = counter-witnesses of A

A striking similarity

Interpretation of function types

- Dialectica:

$$\underline{A \rightarrow B} \equiv (\underline{A} \rightarrow \underline{B}) \times (\underline{A} \times \overline{B} \rightarrow \overline{A})$$

$\underline{A}$ = witnesses of A $\overline{A}$ = counter-witnesses of A

- Sequential algorithms:

$$[A \rightarrow B] \simeq ([A] \rightarrow [B]) \times ([A] \times C_B \rightarrow C_A)$$

$[A]$ = "strategies" on A C_A = "opponent moves" on A

The reverse component in Dialectica

- Brouwer interpretation of $A \Rightarrow B$:

$f : \underline{A} \rightarrow \underline{B}$ such that

The reverse component in Dialectica

- Brouwer interpretation of $A \Rightarrow B$:

$f : \underline{A} \rightarrow \underline{B}$ such that

if x is a “universal” proof of A

then $f(x)$ is a “universal” proof of B

The reverse component in Dialectica

- Brouwer interpretation of $A \Rightarrow B$:

$$f : \underline{A} \rightarrow \underline{B} \quad \text{such that}$$

if x is a “universal” proof of A

then $f(x)$ is a “universal” proof of B

- Dialectica interpretation of $A \Rightarrow B$:

$$f : \underline{A} \rightarrow \underline{B} \quad \text{and} \quad \varphi : \underline{A} \times \overline{B} \rightarrow \overline{A} \quad \text{such that}$$

The reverse component in Dialectica

- Brouwer interpretation of $A \Rightarrow B$:

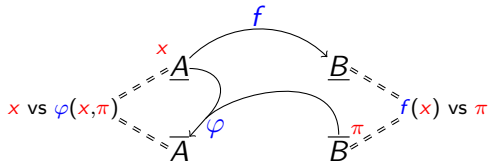
$$f : \underline{A} \rightarrow \underline{B} \quad \text{such that}$$

if x is a “universal” proof of A

then $f(x)$ is a “universal” proof of B

- Dialectica interpretation of $A \Rightarrow B$:

$$f : \underline{A} \rightarrow \underline{B} \quad \text{and} \quad \varphi : \underline{A} \times \overline{B} \rightarrow \overline{A} \quad \text{such that}$$



if x “wins” against $\varphi(x, \pi)$ then $f(x)$ “wins” against π

The reverse component in Dialectica

- Brouwer interpretation of $A \Rightarrow B$:

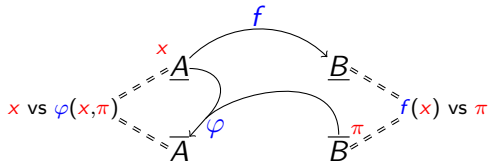
$$f : \underline{A} \rightarrow \underline{B} \quad \text{such that}$$

if x is a “universal” proof of A

then $f(x)$ is a “universal” proof of B

- Dialectica interpretation of $A \Rightarrow B$:

$$f : \underline{A} \rightarrow \underline{B} \quad \text{and} \quad \varphi : \underline{A} \times \overline{B} \rightarrow \overline{A} \quad \text{such that}$$



if x “wins” against $\varphi(x, \pi)$ then $f(x)$ “wins” against π

x is a “relative” proof of A w.r.t. $\varphi(x, \pi)$

$f(x)$ is a “relative” proof of B w.r.t. π

The reverse component in sequential algorithms

- ▶ Scott interpretation of $M : A \rightarrow B$:

$$[M] : [A] \rightarrow [B] \quad \text{such that}$$

The reverse component in sequential algorithms

- Scott interpretation of $M : A \rightarrow B$:

$$[M] : [A] \rightarrow [B] \quad \text{such that}$$

if we have information x on A

then we have information $[M](x)$ on B

The reverse component in sequential algorithms

- ▶ Scott interpretation of $M : A \rightarrow B$:

$$[M] : [A] \rightarrow [B] \quad \text{such that}$$

if we have information x on A

then we have information $[M](x)$ on B

- ▶ Sequential algorithms interpretation of $M : A \rightarrow B$:

$$[M]_1 : [A] \rightarrow [B] \quad \text{and} \quad [M]_2 : [A] \times C_B \rightarrow C_A \quad \text{such that}$$

The reverse component in sequential algorithms

- Scott interpretation of $M : A \rightarrow B$:

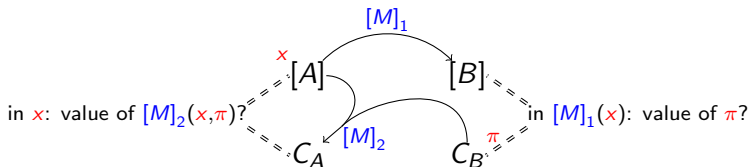
$$[M] : [A] \rightarrow [B] \quad \text{such that}$$

if we have information x on A

then we have information $[M](x)$ on B

- Sequential algorithms interpretation of $M : A \rightarrow B$:

$$[M]_1 : [A] \rightarrow [B] \quad \text{and} \quad [M]_2 : [A] \times C_B \rightarrow C_A \quad \text{such that}$$



The reverse component in sequential algorithms

- Scott interpretation of $M : A \rightarrow B$:

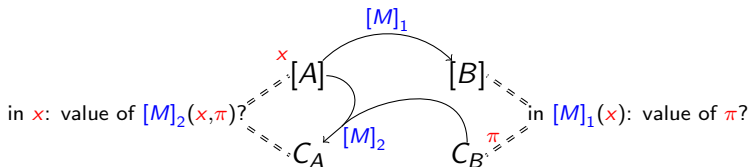
$$[M] : [A] \rightarrow [B] \quad \text{such that}$$

if we have information x on A

then we have information $[M](x)$ on B

- Sequential algorithms interpretation of $M : A \rightarrow B$:

$$[M]_1 : [A] \rightarrow [B] \quad \text{and} \quad [M]_2 : [A] \times C_B \rightarrow C_A \quad \text{such that}$$



if π is asked on current output $[M]_1(x)$

then we must ask $[M]_2(x, \pi)$ on current input x

The reverse component in sequential algorithms

- Scott interpretation of $M : A \rightarrow B$:

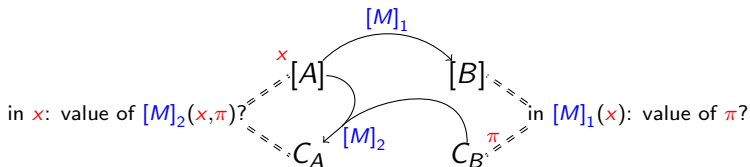
$$[M] : [A] \rightarrow [B] \quad \text{such that}$$

if we have information x on A

then we have information $[M](x)$ on B

- Sequential algorithms interpretation of $M : A \rightarrow B$:

$$[M]_1 : [A] \rightarrow [B] \quad \text{and} \quad [M]_2 : [A] \times C_B \rightarrow C_A \quad \text{such that}$$



if π is asked on current output $[M]_1(x)$

then we must ask $[M]_2(x, \pi)$ on current input x

what is computed **and** *how* it is computed

But there are differences

- ▶ Sequential algorithms model partial computations:
general fixpoint combinator $Y : (A \rightarrow A) \rightarrow A$
represented via the order of Scott domains
- ▶ Dialectica is only concerned with logical validity:
only total elements

But there are differences

- ▶ Sequential algorithms model partial computations:
general fixpoint combinator $Y : (A \rightarrow A) \rightarrow A$
represented via the order of Scott domains
- ▶ Dialectica is only concerned with logical validity:
only total elements

Bridging the gap

- ▶ Diller-Nahm variant of Dialectica:

$$f : \underline{A} \rightarrow \underline{B} \quad \text{and} \quad \varphi : \underline{A} \times \overline{B} \rightarrow \mathcal{P}_0(\overline{A})$$

where $\mathcal{P}_0(\overline{A})$: finite subsets of $\overline{A}$

But there are differences

- ▶ Sequential algorithms model partial computations:
general fixpoint combinator $Y : (A \rightarrow A) \rightarrow A$
represented via the order of Scott domains
- ▶ Dialectica is only concerned with logical validity:
only total elements

Bridging the gap

- ▶ Diller-Nahm variant of Dialectica:

$$f : \underline{A} \rightarrow \underline{B} \quad \text{and} \quad \varphi : \underline{A} \times \overline{B} \rightarrow \mathcal{P}_0(\overline{A})$$

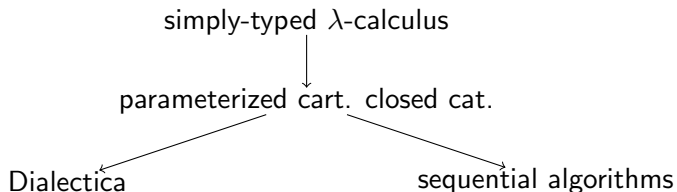
where $\mathcal{P}_0(\overline{A})$: finite subsets of $\overline{A}$

- ▶ Encoding of partiality in sequential algorithms:

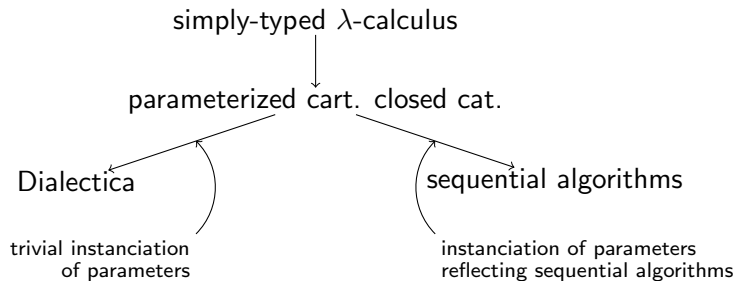
$$[M]_1 : [A] \rightarrow [B] \quad \text{and} \quad [M]_2 : [A] \times C_B \rightarrow \mathcal{P}_0(C_A)$$

with the property that $\# |[M]_2(x, \pi)| \leq 1$

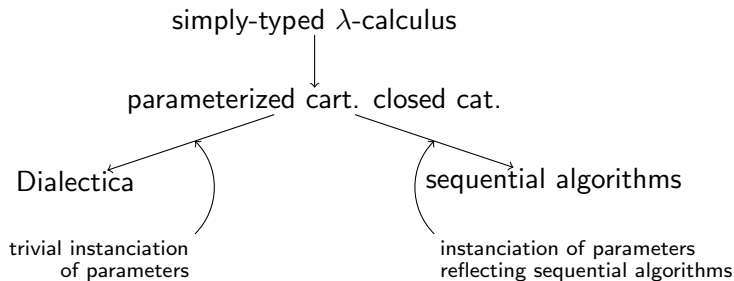
A parameterized interpretation



A parameterized interpretation



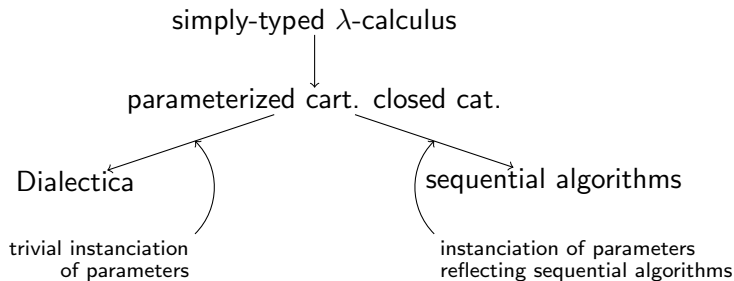
A parameterized interpretation



Parameters allow handling specificities of sequential algorithms:

- Order on $[A]$ (Scott ordering)

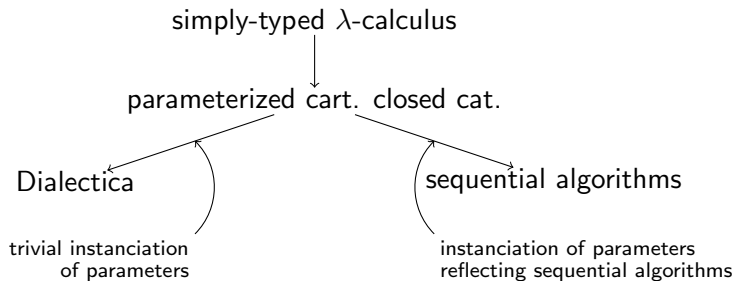
A parameterized interpretation



Parameters allow handling specificities of sequential algorithms:

- ▶ Order on $[A]$ (Scott ordering)
- ▶ Enabling, filling, accessibility of cells $\pi \in C_A$ in states $x \in [A]$

A parameterized interpretation



Parameters allow handling specificities of sequential algorithms:

- ▶ Order on $[A]$ (Scott ordering)
- ▶ Enabling, filling, accessibility of cells $\pi \in C_A$ in states $x \in [A]$
- ▶ If $[M]_2(x, \pi) = \{\rho\}$ then:
 - ▶ π is accessible in $[M]_1(x)$
 - ▶ ρ is accessible in x
- ▶ A technical difference in the definition of uncurrying

A parameterized interpretation

Parameterized category

- ▶ objects are of the form $(W, C, \leq, \dots)$
 - W : witnesses / strategies
 - C : counter-witnesses / opponent moves
 - $\leq$: Scott ordering, trivial in the Dialectica case
 - $\dots$: technical requirements, trivial in the Dialectica case

A parameterized interpretation

Parameterized category

- ▶ objects are of the form $(W, C, \leq, \dots)$
 - W : witnesses / strategies
 - C : counter-witnesses / opponent moves
 - $\leq$: Scott ordering, trivial in the Dialectica case
 - $\dots$: technical requirements, trivial in the Dialectica case

Parameterized exponential

$$\begin{aligned} & (W', C', \leq', \dots)^{(W, C, \leq, \dots)} \\ &= (\{(f, \varphi) \in (W \rightarrow W') \times (W \times C' \rightarrow C) \mid P(f, \varphi)\}, \\ & \quad \{(x, \pi) \in W \times C' \mid Q(x, \pi)\}, \dots) \end{aligned}$$

A parameterized interpretation

Parameterized category

- ▶ objects are of the form $(W, C, \leq, \dots)$
 - W : witnesses / strategies
 - C : counter-witnesses / opponent moves
 - $\leq$: Scott ordering, trivial in the Dialectica case
 - $\dots$: technical requirements, trivial in the Dialectica case

Parameterized exponential

$$\begin{aligned} (W', C', \leq', \dots)^{(W, C, \leq, \dots)} \\ = (\{(f, \varphi) \in (W \rightarrow W') \times (W \times C' \rightarrow C) \mid P(f, \varphi)\}, \\ \{(x, \pi) \in W \times C' \mid Q(x, \pi)\}, \dots) \end{aligned}$$

- ▶ Dialectica:

$$P(f, \varphi) \equiv Q(x, \pi) \equiv \top$$

- ▶ Sequential algorithms:

$$P(f, \varphi) \text{ and } Q(x, \pi) \text{ encode specific requirements}$$

The main theorems

Theorem

The Dialectica instantiation is equivalent to the De Paiva Dialectica category built from the synactic model of simply-typed λ -calculus

The main theorems

Theorem

The Dialectica instantiation is equivalent to the De Paiva Dialectica category built from the syntactic model of simply-typed λ -calculus

Theorem

The sequential algorithms instantiation is equivalent to the category of concrete data structures and sequential algorithms between them

Summary

- ▶ Gödel's functional interpretation is an intensional refinement of Kreisel's realizability

Summary

- ▶ Gödel's functional interpretation is an intensional refinement of Kreisel's realizability
- ▶ Sequential algorithms are an extension of Dialectica that handles partiality

Summary

- ▶ Gödel's functional interpretation is an intensional refinement of Kreisel's realizability
- ▶ Sequential algorithms are an extension of Dialectica that handles partiality
- ▶ Once again, the logical and programmatic approaches lead to the same concepts

Summary

- ▶ Gödel's functional interpretation is an intensional refinement of Kreisel's realizability
- ▶ Sequential algorithms are an extension of Dialectica that handles partiality
- ▶ Once again, the logical and programmatic approaches lead to the same concepts
- ▶ Duality is pervasive in both interpretations:
 - witnesses vs. counter-witnesses
 - strategies vs. opponent moves

A lot of exciting perspectives

- ▶ Very preliminary work: more structure to be found

A lot of exciting perspectives

- ▶ Very preliminary work: more structure to be found
- ▶ Linear decomposition:
 - Dialectica: De Paiva, Pédrot
 - Sequential algorithms: Lamarche

A lot of exciting perspectives

- ▶ Very preliminary work: more structure to be found
- ▶ Linear decomposition:
 - Dialectica: De Paiva, Pédrot
 - Sequential algorithms: Lamarche
- ▶ Dependent types

A lot of exciting perspectives

- ▶ Very preliminary work: more structure to be found
- ▶ Linear decomposition:
 - Dialectica: De Paiva, Pédrot
 - Sequential algorithms: Lamarche
- ▶ Dependent types
- ▶ Many variants of Dialectica:
 - I use the Diller-Nahm variant
 - many other variants: bounded interpretation
 - Herbrand interpretation, etc.

A lot of exciting perspectives

- ▶ Very preliminary work: more structure to be found
- ▶ Linear decomposition:
 - Dialectica: De Paiva, Pédrot
 - Sequential algorithms: Lamarche
- ▶ Dependent types
- ▶ Many variants of Dialectica:
 - I use the Diller-Nahm variant
 - many other variants: bounded interpretation
 - Herbrand interpretation, etc.
- ▶ Game semantics conceptually similar to sequential algorithms

A lot of exciting perspectives

- ▶ Very preliminary work: more structure to be found
- ▶ Linear decomposition:
 - Dialectica: De Paiva, Pédrot
 - Sequential algorithms: Lamarche
- ▶ Dependent types
- ▶ Many variants of Dialectica:
 - I use the Diller-Nahm variant
 - many other variants: bounded interpretation
 - Herbrand interpretation, etc.
- ▶ Game semantics conceptually similar to sequential algorithms
- ▶ At a more general level:
 - connecting functional and intensional approaches

Sequential algorithms as a Dialectica interpretation

questions?

preprint should be available soon...