System T and the Product of Selection Functions

Thomas Powell

(joint work with Martín Escardó and Paulo Oliva)
Queen Mary, University of London

CSL, Bergen
12 September, 2011

Gödel's functional interpretation

$$\textbf{Classical theory} \quad \mapsto \quad \textbf{Higher type functionals}$$

$$\text{Quantifier dependencies} \quad \rightsquigarrow \quad \text{Functional dependencies}$$

Interprets formulas A by quantifier-free formulas $A^I(x, y)$.

$$A \Leftrightarrow \exists x \forall y \, A^I(x, y)$$

We say that $\mathcal{T}$ has a functional interpretation in F if

$$\mathcal{T} \vdash A \;\Rightarrow\; \mathsf{F} \vdash A^I(t, y) \text{ for some } t \in \mathsf{F}$$

Applications

- Consistency proofs: $\mathcal{T}$ interpreted in F then

$$\text{cons}(\text{F}) \Rightarrow \text{cons}(\mathcal{T}).$$

- Bridge between logic and computation: e.g. characterizing provably recursive functions.

- Proof mining: extracting computational content from proofs.

$$\mathcal{T} \vdash \forall y \exists x A(x, y) \Rightarrow A(ty, y) \text{ for } t \in \text{F}$$

Interpreting subsystems of mathematics

Subsystem of mathematics	$\mapsto$	**Higher-type functionals**
Predicate logic	$\mapsto$	Simply typed λ-calculus
$+$		$+$
Axiom schemata	$\mapsto$	Recursion schemata

Interpreting subsystems of mathematics

Classical analysis *Countable choice*	$\mapsto$	Spector's bar recursion
Classical arithmetic *Induction all formulas*	$\mapsto$	Gödel's system **T** *Primitive recursion all types*
Restricted induction	$\mapsto$	*Primitive recursion restricted type*
Predicate logic	$\mapsto$	Simply-typed λ-calculus

Interpreting subsystems of mathematics

Classical analysis *Countable choice*	$\mapsto$	Spector's bar recursion / Unbounded product of selection functions
Classical arithmetic *Induction all formulas*	$\mapsto$	Gödel's system **T** *Primitive recursion all types*
Restricted induction	$\mapsto$	*Primitive recursion restricted type*
Predicate logic	$\mapsto$	Simply-typed λ-calculus

Interpreting subsystems of mathematics

Classical analysis *Countable choice*	$\mapsto$	Spector's bar recursion / Unbounded product of selection functions
Classical arithmetic *Induction all formulas* $\Updownarrow$ *Finite choice all formulas*	$\mapsto$	Gödel's system **T** *Primitive recursion all types* *Finite product all types*
Restricted induction	$\mapsto$	*Primitive recursion restricted type*
Restricted finite choice	$\mapsto$	*Finite product restricted type*
Predicate logic	$\mapsto$	Simply-typed λ-calculus

Interpreting subsystems of mathematics

Classical analysis *Countable choice*	$\mapsto$	Spector's bar recursion / Unbounded product of selection functions
Classical arithmetic *Induction all formulas* $\Updownarrow$ *Finite choice all formulas*	$\mapsto$	Gödel's system **T** *Primitive recursion all types* $\Updownarrow$ *Finite product all types*
Restricted induction	$\mapsto$	*Primitive recursion restricted type* $\Updownarrow$
Restricted finite choice	$\mapsto$	*Finite product restricted type*
Predicate logic	$\mapsto$	Simply-typed λ-calculus

Interpreting subsystems of mathematics

Classical analysis *Countable choice*	$\mapsto$	Spector's bar recursion/ Unbounded product of selection functions
Classical arithmetic *Induction all formulas* $\Updownarrow$ *Finite choice all formulas*	$\mapsto$	Gödel's system **T** **Primitive recursion all types** $\Updownarrow$ **Finite product all types**
Restricted induction	$\mapsto$	**Primitive recursion restricted type** $\Updownarrow$
Restricted finite choice	$\mapsto$	**Finite product restricted type**
Predicate logic	$\mapsto$	Simply-typed λ-calculus

Main results

1. Introduce finite product of selection functions, a novel recursion schema based on the computational of optimal strategies in finite games.
2. Show that axiom of finite choice has a **natural** functional interpretation by finite product.
3. Prove that finite product is equivalent to primitive recursion, and that equivalence holds for restricted fragments of both systems.

Fragments of arithmetic

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

$$\text{FAC} : \; \forall i \leq m \exists x A_i(x) \rightarrow \exists \alpha \forall i \leq m A_i(\alpha_i).$$

A Σ_n-formula is one of the form $\exists x_0 \forall x_1 \ldots \exists/\forall x_{n-1}\, A_0$.
A Π_n-formula is one of the form $\forall x_0 \exists x_1 \ldots \exists/\forall x_{n-1}\, A_0$.

$$\begin{array}{ccccccc}
\Sigma_n\text{-IND} & \longleftarrow & \Pi_n\text{-FAC} & \longleftarrow & \Sigma_{n+1}\text{-IND} & \leftarrow\cdots & \text{Peano arithmetic} \\
\Updownarrow & & \Updownarrow & & & & \\
\Pi_n\text{-IND} & & \Sigma_{n+1}\text{-FAC} & & & &
\end{array}$$

Gödel's primitive recursive functionals

$$\begin{aligned} \mathrm{R}_0^X(y, z) &= y \\ \mathrm{R}_m^X(y, z) &= z_{m-1}(\mathrm{R}_{m-1}^X(y, z)). \end{aligned}$$

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

The primitive recursive functionals of level n consist of those definable using R^X for $\deg(X) \leq n$.

Interpreting fragments of arithmetic

Peano arithmetic *Induction all formulas*	$\mapsto$	Primitive recursion all levels
$\downarrow$		
Σ_{n+1}-Induction $\downarrow$ Π_n-Finite choice	$\mapsto$	Primitive recursion level n
$\downarrow$		
Weak arithmetic *Quantifier-free induction*	$\mapsto$	λ-calculus + basic recursive functionals

The product of selection functions

- m round sequential game, moves of type X, play of type X^m.
- $q \colon X^m \to R$ maps each play to an *outcome* in of type R.
- To each round we associate a *selection function*

  $$\varepsilon_i \colon (X \to R) \to X$$

  which implements a strategy for that round.

In any such game there exists an optimal strategy (Nash equilibrium)

$$S(\varepsilon, q) \equiv x_0, \ldots, x_{m-1}.$$

The product of selection functions

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

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

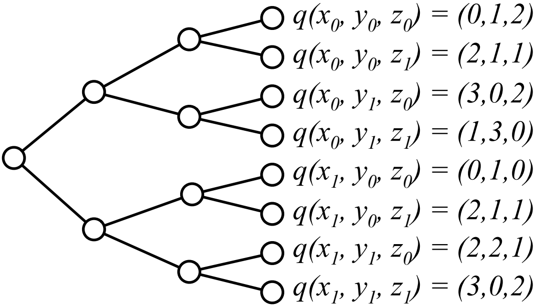
Backward recursion along finite tree

$$\mathrm{P}_0(\varepsilon, m, q) \equiv \langle x_0, x_1, \ldots, x_m, 0, 0, \ldots \rangle$$

where $x_0, \ldots, x_m$ the optimal strategy in $m+1$ round game defined by $\varepsilon_0, \ldots, \varepsilon_m$ and q.

The product of selection functions

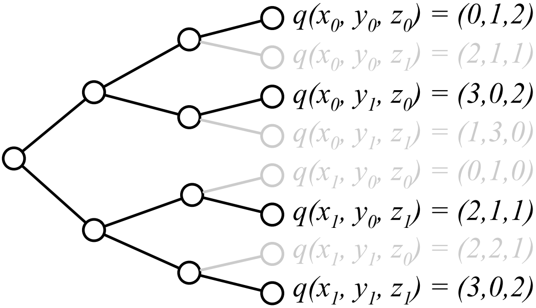
ε_3 - irrelevant



$$\mathsf{P}_3(\varepsilon)(2)(q_{x_1,y_0,z_1}) = \langle 0, 0, \ldots \rangle$$

The product of selection functions

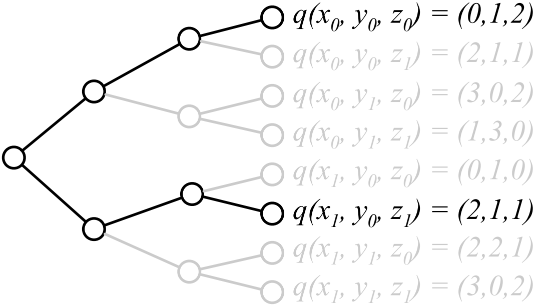
ε_2 - maximise third column



$$\mathsf{P}_2(\varepsilon)(2)(q_{x_1,y_0}) = \langle z_1, 0, 0, \ldots \rangle$$

The product of selection functions

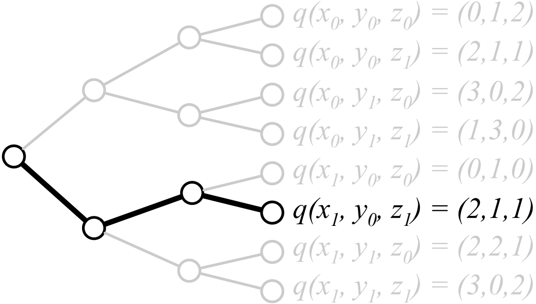
ε_1 - maximise second column



$$\mathsf{P}_1(\varepsilon)(2)(q_{x_1}) = \langle y_0, z_1, 0, 0, \ldots \rangle$$

The product of selection functions

ε_0 - maximise first column



$$\mathsf{P}_0(\varepsilon)(2)(q) = \langle x_1, y_0, z_1, 0, 0, \ldots \rangle$$

The product of selection functions

Primitive recursion

Recursion along natural numbers.

A natural computational analogue of induction.

Finite product of selection functions

Backward recursion along finite tree.

Optimal strategies in sequential games, backtracking algorithms etc...

A natural computational analogue of finite choice.

Interpreting finite choice

What is the functional interpretation of $\exists x \forall y A(x, y)$?

$$\exists x \forall y A(x, y) \mapsto \neg\neg\exists x \forall y A(x, y)$$

$$\mapsto \neg\exists p \forall x \neg A(x, px) \qquad p \text{ counterexample function}$$

$$\mapsto \exists \varepsilon \forall p A(\varepsilon p, p(\varepsilon p)) \qquad \varepsilon \text{ selection function.}$$

There exists a selection function $\varepsilon \colon (X \to Y) \to X$ that for any counterexample function $p \colon X \to Y$ selects a point at which it fails i.e. $A(\varepsilon p, p(\varepsilon p))$ holds.

Interpreting finite choice

$$\forall i \leq m \exists x \forall y A_i(x,y) \rightarrow \exists \alpha \forall i \leq m \forall y A_i(\alpha_i, y)$$
$$\Downarrow$$
$$\exists \varepsilon \forall i \leq m \forall p A_i(\varepsilon_i p, p(\varepsilon_i p)) \rightarrow \forall q \exists \alpha \forall i \leq m A_i(\alpha_i, q\alpha)$$

Premise: there exists a collection (ε_i) of strategies refuting **pointwise** counterexample functions p for A_i.

Conclusion: there exists a co-operative strategy α_q refuting a **global** counterexample function q for $\forall i \leq m A_i$.

Interpreting fragments of arithmetic

Peano arithmetic *Finite choice all formulas*	$\mapsto$	Finite product all levels
$\downarrow$		
Σ_{n+1}-Induction $\downarrow$ Π_n-Finite choice	$\mapsto$	Finite product level $n-1$
$\downarrow$		
Weak arithmetic *Quantifier-free induction*	$\mapsto$	λ-calculus $+$ basic recursive functionals

Interpreting fragments of arithmetic

Peano arithmetic *Induction/Choice all formulas*	$\mapsto$	Primitive recursion all levels Finite product all levels
$\downarrow$		
Σ_{n+1}-Induction	$\mapsto$	Primitive recursion level n
$\downarrow$		
Π_n-Finite choice	$\mapsto$	Finite product level $n-1$
$\downarrow$		
Weak arithmetic *Quantifier-free induction*	$\mapsto$	λ-calculus + basic recursive functionals

Interpreting fragments of arithmetic

Peano arithmetic *Induction/Choice all formulas*	$\mapsto$	Primitive recursion all levels $\Updownarrow$ Finite product all levels
$\downarrow$		
Σ_{n+1}-Induction	$\mapsto$	Primitive recursion level n
$\downarrow$		$\Updownarrow$
Π_n-Finite choice	$\mapsto$	Finite product level $n-1$
$\downarrow$		
Weak arithmetic *Quantifier-free induction*	$\mapsto$	λ-calculus + basic recursive functionals

Advantages of the finite product

1. Has a natural reading in terms of optimal strategies in sequential games.
2. Offers a more direct interpretation of theorems based on finite choice.
3. Provides a uniform transition from funtional interpretation of arithmetic to that of analysis:

$$\begin{array}{ccc} \text{Arithmetic} & \mapsto & \text{Finite product} \\ \text{Analysis} & \mapsto & \text{Unbounded product} \end{array}$$