A puzzle about Gödel’s numbering

James Avery\textsuperscript{1} \quad Jean-Yves Moyen\textsuperscript{1} \quad Jakob Grue Simonsen\textsuperscript{1}

Jean-Yves.Moyen@lipn.univ-paris13.fr

\textsuperscript{1}Datalogisk Institut
University of Copenhagen

Supported by the VILLUM FONDEN network for Experimental Mathematics in Number Theory, Operator Algebras, and Topology; the Marie Curie action “Walgo” program H2020-MSCA-IF-2014 number 655222; and the Danish Council for Independent Research Sapere Aude grant “Complexity via Logic and Algebra” (COLA).

October 6-7 2016
Part 2: some answers
The puzzle

Can you choose:
- a programming language, $\text{Pgms}$;
- a Gödel’s numbering, $\varepsilon$, for it;
- a binary operator, $F$, on it;

such that the induced operator, $F$, on numbers is “as simple as possible”?

**Sequential composition**: $\mathcal{S}$ (on programs), $\mathcal{S}$ (on numbers).
**Parallel composition**: $\mathcal{P}$, $\mathcal{P}$. 
Compositions

What is a “sequential composition” operator on programs?
Something that behaves as expected with respect to semantics!

\[ S(p, q) = r \text{ with } \lfloor r \rfloor = \lfloor p \rfloor \circ \lfloor q \rfloor \]

Same goes with (non-deterministic, no communication) parallel composition:

\[ \lfloor P(p, q) \rfloor = \lfloor p \rfloor \parallel \lfloor q \rfloor \]
A deceptively simple answer
Commutativity

If $S$ (on number) is commutative, then so must be $S$ (on programs). (because $\varepsilon$ is a morphism)

$$\varepsilon(S(p, q)) = S(\varepsilon(p), \varepsilon(q)) = S(\varepsilon(q), \varepsilon(p)) = \varepsilon(S(q, p))$$

By injectivity: $S(p, q)) = S(q, p)$.

Addition is commutative. Sequential composition is not commutative (because $\circ$ is not). Therefore, there is no Gödel encoding and sequential composition such that $S$ is addition.
Associativity

Sequential composition may be not associative: we’re on the syntactical level, so \( \{\{x++;y++\};z++\} \) and \( \{x++;\{y++;z++\}\} \) are two different commands (strings).

But since \( \circ \) is associative, associative sequential composition operator do exists.

**Theorem** (Bell, 1936): the only associative polynomials with 2 variables are the projections and 
\[
P(X, Y) = a + b \cdot (X + Y) + c \cdot XY.
\]

Sequential composition cannot be a projection. The other solution is commutative.

There is no Gödel encoding and associative sequential composition such that \( S \) is a polynomial.
Going further
Other functions

**Theorem** (Aczél, 1948): a function on the real numbers is continuous, strictly increasing and associative iff it has the shape

$$M(x, y) = f^{-1}(f(x) + f(y))$$

Especially, it is then commutative.

There is no Gödel encoding and sequential composition such that S can be extended as a continuous, strictly increasing and associative function.

One extension with the property is enough!

Thus, we need infinitely many discontinuities (or decreases) in **all** the possible extensions to the reals.
Concatenation

It is possible to design a language and an encoding such that concatenation (of the programs, or the binary encodings) is a sequential composition.

Idea: assembly like language, one designed input-output register (must reset all other to 0 before ending), only relative jumps, encoding with leading ’1’ everywhere.

Concatenation is \( x, y \mapsto x \times 2^{\lfloor \log y \rfloor + 1} + y \), roughly equal to \((2x + 1) \cdot y\). Simple, and polynomially bounded!
Parallel composition
Distributivity

Parallel composition can be commutative, so we cannot rule out addition so easily.

On functions, sequential composition is distributive over parallel composition:

\[ f \circ (g \parallel h) = (f \circ g) \parallel (f \circ h) \]

Thus, there exist sequential composition operators which are distributive over a parallel composition operator.

There is no operation on the natural number that is distributive over multiplication.

Therefore, **parallel composition cannot be multiplication** (if there is a sequential composition distributing over it).
It’s all about functions

$$(\text{Pgms, } \mathbb{F}) \xrightarrow{\varepsilon} (N, F)$$

$$\xrightarrow{[\bullet]}$$

$$(C, \hat{F}) \xrightarrow{\hat{\varepsilon}} (C, \hat{F})$$

Sequential composition: $\hat{S} = \circ$, parallel composition: $\hat{P} = ||$.
$\hat{P}$ is compatible with $[\bullet]$: 

$$[p] = [p'] \Rightarrow [\hat{P}(p, q)] = [\hat{P}(p', q)]$$
A deceptively simple answer
Going further
Parallel composition

Computable functions

\[(\text{Pgm}s, \mathbb{P}) \stackrel{\varepsilon}{\longrightarrow} (N, P)\]

\[\downarrow \quad \downarrow \]

\[\bullet \quad \bullet\]

\[(C, \hat{\mathbb{P}}) \stackrel{\hat{\varepsilon}}{\longrightarrow} (C, \hat{P})\]

\[\bullet\] is the extensional equivalence, it must have the same structure as \([\bullet]\), especially \(P\) must be compatible with \([\bullet]\).

Equivalences on the natural numbers compatible with addition have finitely many non-singleton classes. \([\bullet]\) has infinitely many infinite classes.

Therefore, there is no Gödel encoding and parallel composition such that \(P\) is addition.