The power and limitations of Self-interpretation

H. Geuvers

Institute for Computing and Information Sciences
Radboud University Nijmegen

Version: fall 2015

Lambda-terms themselves are "black boxes"

- There is no λ-term F such that

$$F(M\,N) = M \qquad \text{for all } M, N$$

- There is no λ-term F such that

$$F(M\,N) = N \qquad \text{for all } M, N$$

On the other hand, we can compute with the codes of λ-terms

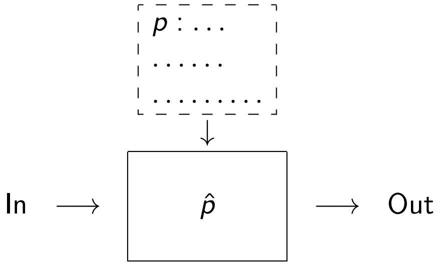
- There is a λ-term F such that $F\ulcorner M\,N\urcorner = \ulcorner M\urcorner$ for all M, N.
- There is a λ-term F such that $F\ulcorner M\,N\urcorner = \ulcorner N\urcorner$ for all M, N.
- We also have an evaluator E:

$$E\ulcorner M\urcorner = M$$

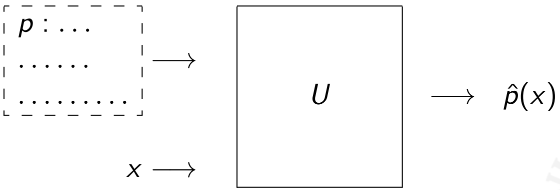
program text p $\xrightarrow{\textit{Compiler}}$ executable $\hat{p}$

Text
- inspect,
- edit,
- ...

Black box
- input/output ("call")
- pass around



Specification of a universal (programmable) machine U:



Reflection over the programs / programming language:
What functions (programs) can we write on program text?

Programs about programs

EXAMPLES of programs one can write

$p : \ldots$ / $\ldots$ / $\ldots$ $\longrightarrow$ M_1 $\longrightarrow$

`true`	if p contains a "while"
`false`	if p contains no "while"

$p : \ldots$ / $\ldots$ / $\ldots$ $\longrightarrow$ M_2 $\longrightarrow$

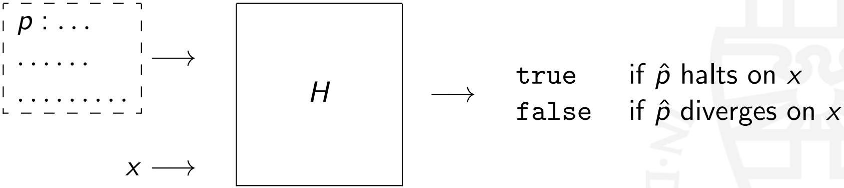
`true`	if $\|p\| \leq 1000$
`false`	if $\|p\| > 1000$

What about this??

$p : \ldots$ / $\ldots$ / $\ldots$ $\longrightarrow$ M_3 $\longrightarrow$

`true`	if $\hat{p}$ halts on all inputs
`false`	if $\hat{p}$ diverges on some input

THEOREM It is impossible to write a program H with the following specification



The undecidability of the Halting Problem was first proven by Turing, for his Turing machines.
We will prove it for the λ-calculus.

Data type for coding lambda terms

We define the constructors Var, App, Abs:

$$\begin{array}{lcl} \texttt{Var} & := & \lambda x\, e.e\, \mathbf{U}^3_1\, x\, e \\ \texttt{App} & := & \lambda x\, y\, e.e\, \mathbf{U}^3_2\, x\, y\, e \\ \texttt{Abs} & := & \lambda x\, e.e\, \mathbf{U}^3_3\, x\, e \end{array}$$

Recursion for the λ-terms data type

THEOREM I. Given $A_1, A_2, A_3 \in \Lambda$ there is an $H \in \Lambda$ such that

$$\begin{array}{rcl} H(\texttt{Var}\, x) & = & A_1\, x\, H \\ H(\texttt{App}\, x\, y) & = & A_2\, x\, y\, H \\ H(\texttt{Abs}\, x) & = & A_3\, x\, H \end{array}$$

PROOF. Take $H = \langle\langle B_1, B_2, B_3\rangle\rangle$ with

$$\begin{array}{lcl} B_1 & := & \lambda x\, z.A_1 x \langle z\rangle \\ B_2 & := & \lambda x\, y\, z.A_2 xy \langle z\rangle \\ B_3 & := & \lambda x\, z.A_3 x \langle z\rangle. \end{array}$$

Then the equations hold indeed. ☻.

Coding of lambda terms

Coding lambda terms $M \leadsto \ulcorner M \urcorner$

Definition (Mogensen) The coding of λ-terms inside the λ-calculus is defined as follows.

$$\begin{array}{rcl} \ulcorner x \urcorner & := & \mathtt{Var}\, x \\ \ulcorner MN \urcorner & := & \mathtt{App} \ulcorner M \urcorner \ulcorner N \urcorner \\ \ulcorner \lambda x.M \urcorner & := & \mathtt{Abs}\, (\lambda x. \ulcorner M \urcorner) \end{array}$$

- coding is not λ-definable:
  There is no term C such that $C\, M = \ulcorner M \urcorner$ for all M.
- The reverse operation, evaluation, is λ-definable:
  There exists a λ-term **E** (evaluator) such that for all $M \in \Lambda$

$$\mathbf{E} \ulcorner M \urcorner = M$$

E can be found by recursion.

Recursion for lambda terms using the encoding

We can state the recursion theorem for the encoded lambda terms slightly differently, as follows.

THEOREM II. Given $A_1, A_2, A_3 \in \Lambda$ there is an $H \in \Lambda$ such that

$$\begin{array}{rcl} H\ulcorner x\urcorner & = & A_1\, x\, H \\ H\ulcorner M\,N\urcorner & = & A_2\ulcorner M\urcorner\ulcorner N\urcorner H \\ H\ulcorner \lambda x.M\urcorner & = & A_3\,(\lambda x.\ulcorner M\urcorner)\, H \end{array}$$

EXAMPLE

There is a λ-term C satisfying

$$\begin{array}{rcl} C\ulcorner x\urcorner & = & c_0 \\ C\ulcorner M\,N\urcorner & = & c_1 \\ C\ulcorner \lambda x.M\urcorner & = & c_2 \end{array}$$

PROOF Just take $A_1 := \lambda xy.c_0$, $A_2 := \lambda xyz.c_1$ and $A_3 := \lambda xy.c_2$.

☻

Exercise: Write a function R that checks whether a code of a term is a redex. (So: $R\ulcorner M\urcorner = \mathbf{T}$ if M is a redex and $R\ulcorner M\urcorner = \mathbf{F}$ if M is not a redex.)

We need to compute with codes

Recall that $\mathbf{T} = \lambda x\, y.x$ and $\mathbf{F} = \lambda x\, y.y$.
The following are impossible to define with λ-terms themselves.
There is no term G satisfying

$$\begin{array}{rcl} G\, M & = & \mathbf{T} \text{ if } M \text{ has a normal form} \\ G\, M & = & \mathbf{F} \text{ if } M \text{ has no normal form} \end{array}$$

There is no term H (compare the Halting problem) satisfying

$$\begin{array}{rcl} H\, M\, N & = & \mathbf{T} \text{ if } M\, N \text{ has a normal form} \\ H\, M\, N & = & \mathbf{F} \text{ if } M\, N \text{ has no normal form} \end{array}$$

That these are impossible is not surprising: we can't "look inside a black box".
What if we recast these question with coded λ-terms?

The Halting problem is undecidable (λ-calculus version):
THEOREM There is no term H satisfying

$$\begin{array}{rcl} H \ulcorner M \urcorner N & = & \mathbf{T} \text{ if } M\,N \text{ has a normal form} \\ H \ulcorner M \urcorner N & = & \mathbf{F} \text{ if } M\,N \text{ has no normal form} \end{array}$$

PROOF Suppose H exists, Consider

$$Q := \lambda x.H\,x\,x\,\Omega\,\mathbf{T}$$

Then

$$\begin{array}{rcl} Q \ulcorner Q \urcorner & = & H \ulcorner Q \urcorner \ulcorner Q \urcorner \Omega\,\mathbf{T} \\ & = & \mathbf{T}\Omega\,\mathbf{T} = \Omega \text{ if } Q \ulcorner Q \urcorner \text{ has a normal form} \\ & = & \mathbf{F}\Omega\,\mathbf{T} = \mathbf{T} \text{ if } Q \ulcorner Q \urcorner \text{ has no normal form} \end{array}$$

Contradiction. So H doesn't exist. ☻.

The "Blank tape" problem for λ-calculus

THEOREM There is no term B satisfying

$$\begin{aligned} B\,\ulcorner M \urcorner &= \mathbf{T} \text{ if } M \text{ has a normal form} \\ B\,\ulcorner M \urcorner &= \mathbf{F} \text{ if } M \text{ has no normal form} \end{aligned}$$

PROOF Suppose B exists, Consider

$$Q := \lambda x.B\,x\,\Omega\,\mathbf{T}$$

By the second fixed point theorem, there is a R such that $Q\,\ulcorner R \urcorner = R$, that is

$$B\,\ulcorner R \urcorner\,\Omega\,\mathbf{T} = R.$$

Now we have

$$\begin{aligned} R &= B\,\ulcorner R \urcorner\,\Omega\,\mathbf{T} \\ &= \mathbf{T}\,\Omega\,\mathbf{T} = \Omega \text{ if } R \text{ has a normal form} \\ &= \mathbf{F}\,\Omega\,\mathbf{T} = \mathbf{T} \text{ if } R \text{ has no normal form} \end{aligned}$$

Contradiction. So B doesn't exist.

(There may be a proof by reducing H to B; I didn't see it.)

Some other terms one can(not) write

EXAMPLE There is a term L satisfying

$$L \ulcorner M \urcorner \ = \ c_n \text{ if } n \text{ is the number of } \lambda\text{'s inside } M$$

EXAMPLE There is a term V satisfying

$$\begin{aligned} V \ulcorner M \urcorner &= \mathbf{T} \text{ if } M \text{ is of the shape } x\, P_1 \ldots P_n \text{ for some } n, \vec{P}, \\ V \ulcorner M \urcorner &= \mathbf{F} \text{ if } M \text{ is not of the shape } x\, P_1 \ldots P_n. \end{aligned}$$

EXAMPLE There is no term H' satisfying

$$\begin{aligned} H' \ulcorner M \urcorner \ulcorner N \urcorner &= \mathbf{T} \text{ if } M\, N \text{ has a normal form} \\ H' \ulcorner M \urcorner \ulcorner N \urcorner &= \mathbf{F} \text{ if } M\, N \text{ has no normal form} \end{aligned}$$

PROOF Reduce B to H'. (Show that, if H' exists, then we can also define B.)