Recursion Theorem and Kleene's *s–m–n* Theorem

Martin Čermák, Jiří Koutný and Alexander Meduna

Deparment of Information Systems
Faculty of Informatin Technology
Brno University of Technology, Faculty of Information Technology
Božetěchova 2, Brno 612 00, Czech Republic

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Advanced Topics of Theoretical Computer Science

FRVŠ MŠMT FR2581/2010/G1

Part I

Recursion Theorem and Kleene's *s*–*m*–*n* Theorem

- Consider any total computable function γ over $\mathbb{N}$ and apply γ to the indices of Turing Machines in ζ.
- There necessarily exists $n \in \mathbb{N}$, customarily referred to as a *fixed point* of γ, such that ${}_nM$ and ${}_{\gamma(n)}M$ compute the same function.
- That is, in terms of ξ, ${}_nM\text{–}f =_{\gamma(n)} M\text{–}f$.

Theorem

For every total computable function γ over $\mathbb{N}$, there is $n \in \mathbb{N}$ such that ${}_nM\text{–}f =_{\gamma(n)} M\text{–}f$ in ξ.

The recursion theorem is a powerful tool frequently applied in the theory of computation.

Generalization to the m-argument function $M\text{–}f^{\underline{m}}$ computed by $M \in_{TM} \Psi$.

Definition

Let $M \in_{TM} \Psi$. The *m-argument function computed* by M is denoted by $M\text{–}f^{\underline{m}}$ and defined as

$$M\text{–}f^{\underline{m}} = \{(x, y) | x \in \triangle^*, occur(x, \#) = m - 1, y \in (\triangle - \{\#\})^*, \\ \triangleright\blacktriangleright x\triangleleft \Rightarrow^* \triangleright\blacksquare yu\triangleleft \text{ in } M, u \in \{\square\}^*\}$$

That is

- $f^{\underline{m}}(x_1, x_2, \ldots, x_m) = y$ iff $\triangleright\blacktriangleright x_1\#x_2\#\ldots\#x_m\triangleleft \Rightarrow^* \triangleright\blacksquare yu\triangleleft$ in M with $u \in \{\square\}^*$,
- $f^{\underline{m}}(x_1, x_2, \ldots, x_m)$ is undefined iff M loops on $x_1\#x_2\#\ldots\#x_m$ or rejects $x_1\#x_2\#\ldots\#x_m$,
- Notice that $M\text{–}f^{\underline{1}}$ coincides with $M\text{–}f$.

Definition

Let $m \in \mathbb{N}$. A function $f^{\underline{m}}$ is a computable function if there exists $M \in_{TM} \Psi$ such that $f^{\underline{m}} = M\text{–}f^{\underline{m}}$; otherwise, $f^{\underline{m}}$ is incomputable.

To use Turing Machines as computers of m-argument integer functions, we assume these machines work with the *unary*-based representation of integers by analogy with one-argument integer functions computed by these machines.

Definition

Let $M \in_{TM} \Psi$, $m \in \mathbb{N}$, and $f^{\underline{m}}$ be an m-argument function from $A_1 \times \cdots \times A_m$ to $\mathbb{N}$, where $A_i = \mathbb{N}$, for all $1 \leq i \leq m$. M computes $f^{\underline{m}}$ iff this equivalence holds

$$f^{\underline{m}}(x_1, \ldots, x_m) = y \text{ iff } (unary(x_1)\#\ldots\#unary(x_m), unary(y)) \in M\text{–}f^{\underline{m}}$$

Kleene's $s-m-n$ theorem says that for all $m, n \in \mathbb{N}$, there is a total computable function s of $m+1$ arguments such that ${}_iM\text{–}f^{\underline{m+n}}(x_1, \ldots, x_m, y_1, \ldots, y_n) =_{s(i,x_1,\ldots,x_m)} M\text{–}f^{\underline{n}}(y_1, \ldots, y_n)$ for all $i, x_1, \ldots, x_m, y_1, \ldots, y_n$.

- That is, the number of arguments is lowered, yet the same function is computed.

Theorem

For all $i, m, n \in \mathbb{N}$, there is a total computable $(m+1)$-argument function $s^{\underline{m+1}}$ such that ${}_iM\text{–}f^{\underline{m+n}}(x_1, \ldots, x_m, y_1, \ldots, y_n) =_{s^{\underline{m+1}}(i,x_1,\ldots,x_m)} M\text{–}f^{\underline{n}}(y_1, \ldots, y_n)$.

Theorem represents a powerful tool for demonstrating closure properties concerning computable functions.

Proof idea

- Construct a Turing machine $S \in {}_{TM}\Psi$ and demonstrate that $S\text{–}f^{m+1}_{\underline{}}$ satisfies the properties of $s^{m+1}_{\underline{}}$ stated in previous Theorem.
- Thus, we just take $s^{m+1}_{\underline{}} = S\text{–}f^{m+1}_{\underline{}}$ to complete the proof.

Construction of S

- Let $i, m, n \in \mathbb{N}$.
- Construct a Turing machine $S \in {}_{TM}\Psi$ so S itself constructs another machine in ${}_{TM}\Psi$ and produces its index in ζ as the resulting output value.

Properties of $S\text{–}f^{\underline{(m+1)}}$

- Consider the $(m+1)$-argument function $S\text{–}f^{\underline{(m+1)}}$ computed by S constructed above.
- $S\text{–}f^{\underline{(m+1)}}$ maps $(i, x_1, \ldots, x_m)$ to the resulting output value equal to the index of $M[i, x_1, \ldots, x_m]$ in ζ.
- $M[i, x_1, \ldots, x_m]$ computes ${}_iM\text{–}f^{\underline{m+n}}(x_1, \ldots, x_m, y_1, \ldots, y_n)$ on every input $(y_1, \ldots, y_n)$, where ${}_iM\text{–}f^{\underline{m+n}}$ denotes the $(m+n)$-argument computable function.
- Thus, ${}_iM\text{–}f^{\underline{m+n}}(x_1, \ldots, x_m, y_1, \ldots, y_n) = {}_jM\text{–}f^{\underline{n}}(y_1, \ldots, y_n) = {}_{S-f^{\underline{m+1}}(i,x_1,\ldots,x_m)}M\text{–}f^{\underline{n}}(y_1, \ldots, y_n)$.
- Therefore, to obtain the total computable $(m+1)$-argument function $s^{\underline{m+1}}$ satisfying previous Theorem, set $s^{\underline{m+1}} = S\text{–}f^{\underline{m+1}}$.

This theorem represents a powerful tool for demonstrating closure properties concerning computable functions.

Kleene's s–m–n theorem

Example

- Consider a total computable 2-argument function $\underline{g^2}$ such that ${}_iM\text{–}f({}_jM\text{–}f(x)) =_{g^2(i,j)} M\text{–}f(x)$ for all $i, j, x \in \mathbb{N}$.
- Define the 3-argument function $\underline{h^3}$ as $h^3(i, j, x) = {}_iM\text{–}f({}_jM\text{–}f(x))$ for all $i, j, x \in \mathbb{N}$.
- Introduce a Turing Machine H that computes $\underline{h^3}$ so it works on every input x as follows:
  1. H runs ${}_jM$ on x,
  2. if ${}_jM\text{–}f(x)$ is defined and produced by H in (1), H runs ${}_iM$ on ${}_jM\text{–}f(x)$,
  3. if ${}_iM\text{–}f({}_jM\text{–}f(x))$ is defined, H produces ${}_iM\text{–}f({}_jM\text{–}f(x))$, so H computes ${}_iM\text{–}f({}_jM\text{–}f(x))$.

  Thus, h^3 is computable.

Example (cont.)

- Let $h^{\underline{3}}$ be computed by ${}_kM$ in ζ. That is, ${}_kM\text{–}f^{\underline{3}} = h$.
- There is a total computable function s such that ${}_{s^{\underline{3}}(k,i,j)}M\text{–}f(x) = {}_kM\text{–}f^{\underline{3}}(i,j,x)$ for all $i, j, x \in \mathbb{N}$.
- Set $g^{\underline{2}}(i,j) = s^{\underline{3}}(k,i,j)$ for all $i, j \in \mathbb{N}$.
- Thus, ${}_iM\text{–}f({}_jM\text{–}f(x)) =_{s^3(k,i,j)} M\text{–}f(x) =_{g^2(i,j)} M\text{–}f(x)$, for all $i, j, x \in \mathbb{N}$.

So, the composition of two computable functions is again computable, so the set of computable one-argument functions is closed with respect to composition.

- Wayne Goddard.
  Introducing the Theory of Computation.
  Jones Bartlett Publishers, 2008.

- Jeffrey D. Ullman John E. Hopcroft, Rajeev Motwani.
  Introduction to Automata Theory, Languages, and Computation.
  Addison Wesley, 2006.

- Dexter C. Kozen.
  Automata and Computability.
  Springer, 2007.

- Dexter C. Kozen.
  Theory of Computation.
  Springer, 2010.

- John C. Martin.
  Introduction to Languages and the Theory of Computation.
  McGraw-Hill Science/Engineering/Math, 2002.

Thank you for your attention!

End