Fundamentele Informatica 3

voorjaar 2015

`http://www.liacs.leidenuniv.nl/~vlietrvan1/fi3/`

Rudy van Vliet
kamer 124 Snellius, tel. 071-527 5777
rvvliet(at)liacs(dot)nl

college 14, 18 mei 2015

10. Computable Functions
10.3. Gödel Numbering
10.4. All Computable Functions are μ-Recursive

A slide from lecture 13

Definition 10.11. Bounded Minimalization

For an $(n+1)$-place predicate P, the *bounded minimalization* of P is the function $m_P : \mathbb{N}^{n+1} \to \mathbb{N}$ defined by

$$m_P(X,k) = \begin{cases} \text{min}\{y \mid \ 0 \leq y \leq k \text{ and } P(X,y)\} & \text{if this set is not empty} \\ k+1 & \text{otherwise} \end{cases}$$

The symbol μ is often used for the minimalization operator, and we sometimes write

$$m_P(X,k) = \overset{k}{\mu}\, y[P(X,y)]$$

An important special case is that in which $P(X,y)$ is $(f(X,y) = 0)$, for some $f : \mathbb{N}^{n+1} \to \mathbb{N}$. In this case m_P is written m_f and referred to as the bounded minimalization of f.

A slide from lecture 13

Theorem 10.12.

If P is a primitive recursive $(n+1)$-place predicate,
its bounded minimalization m_P is a primitive recursive function.

Proof. . .

A slide from lecture 13

Example 10.13. The nth Prime Number

$PrNo(0) = 2$
$PrNo(1) = 3$
$PrNo(2) = 5$

$$\begin{aligned} Prime(n) \;=\; & (n \geq 2) \wedge \neg(\text{there exists } y \text{ such that} \\ & \quad y \geq 2 \wedge y \leq n-1 \wedge Mod(n,y) = 0) \end{aligned}$$

A slide from lecture 13

Example 10.13. The nth Prime Number

Let

$$P(x, y) = (y > x \wedge \mathit{Prime}(y))$$

Then $m_P(x, k)$ . . .
and

$$\begin{aligned} \mathit{PrNo}(0) &= 2 \\ \mathit{PrNo}(k+1) &= m_P(\mathit{PrNo}(k), (\mathit{PrNo}(k))! + 1) \end{aligned}$$

is primitive recursive, with $h(x_1, x_2) = \ldots$

A slide from lecture 13

Definition 10.14. Unbounded Minimalization

If P is an $(n+1)$-place predicate, the *unbounded minimalization* of P is the partial function $M_P : \mathbb{N}^n \to \mathbb{N}$ defined by

$$M_P(X) = \text{min}\{y \mid \ P(X, y) \text{ is true}\}$$

$M_P(X)$ is undefined at any $X \in \mathbb{N}^n$ for which there is no y satisfying $P(X, y)$.

The notation $\mu\, y[P(X, y)]$ is also used for $M_P(X)$.
In the special case in which $P(X, y) = (f(X, y) = 0)$, we write $M_P = M_f$ and refer to this function as the unbounded minimalization of f.

A slide from lecture 13

Definition 10.15. μ-Recursive Functions

The set $\mathcal{M}$ of μ-recursive, or simply *recursive*, partial functions is defined as follows.

1. Every initial function is an element of $\mathcal{M}$.

2. Every function obtained from elements of $\mathcal{M}$ by composition or primitive recursion is an element of $\mathcal{M}$.

3. For every $n \geq 0$ and every total function $f : \mathbb{N}^{n+1} \to \mathbb{N}$ in $\mathcal{M}$, the function $M_f : \mathbb{N}^n \to \mathbb{N}$ defined by
$$M_f(X) = \mu\, y[f(X, y) = 0]$$
is an element of $\mathcal{M}$.

A slide from lecture 13

Theorem 10.16.

All μ-recursive partial functions are computable.

Proof. . .

A slide from lecture 13

Definition 10.17.
The Gödel Number of a Sequence of Natural Numbers

For every $n \geq 1$ and every finite sequence $x_0, x_1, \ldots, x_{n-1}$ of n natural numbers, the *Gödel number* of the sequence is the number

$$gn(x_0, x_1, \ldots, x_{n-1}) = 2^{x_0} 3^{x_1} 5^{x_2} \ldots (PrNo(n-1))^{x_{n-1}}$$

where $PrNo(i)$ is the ith prime (Example 10.13).

Example 10.18.

The Power to Which a Prime is Raised in the Factorization of x

Function *Exponent* $: \mathbb{N}^2 \to \mathbb{N}$ defined as follows:

$$\mathit{Exponent}(i, x) = \begin{cases} \text{the exp. of } \mathit{PrNo}(i) \text{ in } x\text{'s prime fact.} & \text{if } x > 0 \\ 0 & \text{if } x = 0 \end{cases}$$

A slide from lecture 11

Definition 10.2. The Operations of Composition and Primitive Recursion (continued)

2. Suppose $n \geq 0$ and g and h are functions of n and $n+2$ variables, respectively. (By "a function of 0 variables," we mean simply a constant.)
   The function obtained from g and h by the operation of *primitive recursion* is the function $f : \mathbb{N}^{n+1} \to \mathbb{N}$ defined by the formulas

$$\begin{aligned} f(X,0) &= g(X) \\ f(X,k+1) &= h(X,k,f(X,k)) \end{aligned}$$

   for every $X \in \mathbb{N}^n$ and every $k \geq 0$.

Theorem 10.19.

Suppose that $g : \mathbb{N}^n \to \mathbb{N}$ and $h : \mathbb{N}^{n+2} \to \mathbb{N}$ are primitive recursive functions, and $f : \mathbb{N}^{n+1} \to \mathbb{N}$ is obtained from g and h by course-of-values recursion; that is

$$\begin{aligned} f(X, 0) &= g(X) \\ f(X, k+1) &= h(X, k, \mathsf{gn}(f(X, 0), \ldots, f(X, k))) \end{aligned}$$

Then f is primitive recursive.

Proof. . .

Example.

Fibonacci

$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ f(n-1) + f(n-2) & \text{if } n \geq 2 \end{cases}$$

Configuration of Turing machine determined by

- state
- position on tape
- tape contents

A slide from lecture 4:

Assumptions:

1. Names of the states are irrelevant.

2. Tape alphabet Γ of every Turing machine T is subset of infinite set $\mathcal{S} = \{a_1, a_2, a_3, \ldots\}$, where $a_1 = \Delta$.

A slide from lecture 4:

Definition 7.33. An Encoding Function

Assign numbers to each state:
$n(h_a) = 1$, $n(h_r) = 2$, $n(q_0) = 3$, $n(q) \geq 4$ for other $q \in Q$.

Assign numbers to each tape symbol:
$n(a_i) = i$.

Assign numbers to each tape head direction:
$n(R) = 1$, $n(L) = 2$, $n(S) = 3$.

Now different numbering

Let $T = (Q, \Sigma, \Gamma, q_0, \delta)$ be Turing machine

States:

h_a	h_r	q_0	$\ldots$	.
0	1	2	$\ldots$	s_T

with $s_T = \ldots$

Tape symbols:

Δ	$\ldots$	.
0	$\ldots$	ts_T

with $ts_T = \ldots$

Now different numbering

Let $T = (Q, \Sigma, \Gamma, q_0, \delta)$ be Turing machine

States:

h_a	h_r	q_0	$\ldots$	$.$
0	1	2	$\ldots$	s_T

with $s_T = |Q| + 1$

Tape symbols:

Δ	$\ldots$	$.$
0	$\ldots$	ts_T

with $ts_T = |\Gamma|$

$$\begin{aligned} \textit{tapenumber}(\Delta 1 a \Delta b 1 \Delta) &= 2^0 3^1 5^2 7^0 11^3 13^1 17^0 \ldots \\ \textit{confignumber} &= 2^q 3^P 5^{\textit{tapenumber}} \end{aligned}$$

10.4. All Computable Functions are μ-Recursive

We must show that $f : \mathbb{N}^n \to \mathbb{N}$ defined by

$$f(X) = \mathit{Result}_T(f_T(\mathit{InitConfig}^{(n)}(X)))$$

is μ-recursive.

Step 1

The function *InitConfig*$^{(n)} : \mathbb{N}^n \to \mathbb{N}$

Exercise 10.34.

Show using mathematical induction that if $tn^{(n)}(x_1, \ldots, x_n)$ is the tape number containing the string

$$\Delta 1^{x_1} \Delta 1^{x_2} \Delta \ldots \Delta 1^{x_n}$$

then $tn^{(n)} : \mathbb{N}^n \to \mathbb{N}$ is primitive recursive.

Use $nr(\Delta) = 0$ and $nr(1) = 1$.

A slide from lecture 11

Definition 10.2. The Operations of Composition and Primitive Recursion (continued)

2. Suppose $n \geq 0$ and g and h are functions of n and $n+2$ variables, respectively. (By "a function of 0 variables," we mean simply a constant.)
   The function obtained from g and h by the operation of *primitive recursion* is the function $f : \mathbb{N}^{n+1} \to \mathbb{N}$ defined by the formulas

$$\begin{aligned} f(X, 0) &= g(X) \\ f(X, k+1) &= h(X, k, f(X, k)) \end{aligned}$$

   for every $X \in \mathbb{N}^n$ and every $k \geq 0$.

Exercise 10.34.

Show using mathematical induction that if $tn^{(n)}(x_1, \dots, x_n)$ is the tape number containing the string

$$\triangle 1^{x_1} \triangle 1^{x_2} \triangle \dots \triangle 1^{x_n}$$

then $tn^{(n)} : \mathbb{N}^n \to \mathbb{N}$ is primitive recursive.
Suggestion: In the induction step, show that

$$tn^{(m+1)}(X, x_{m+1}) = tn^{(m)}(X) * \prod_{j=1}^{x_{m+1}} PrNo(m + \textstyle\sum_{i=1}^{m} x_i + j)$$

Use $nr(\triangle) = 0$ and $nr(1) = 1$.