Metalogika (6)

Jerzy Pogonowski

Zakład Logiki Stosowanej UAM
www.logic.amu.edu.pl
pogon@amu.edu.pl

Uniwersytet Opolski

Plan wykładu

Omówimy jedną z możliwości matematycznych reprezentacji pojęcia ***obliczalności***, a mianowicie ***funkcje rekurencyjne***.

- Funkcje pierwotnie rekurencyjne.
- Funkcje rekurencyjne.
- Zbiory i relacje: rekurencyjne oraz rekurencyjnie przeliczalne.
- Indeksy.
- Twierdzenia o: postaci normalnej, enumeracji, parametryzacji, diagonalizacji, punkcie stałym.
- Zbiory: produktywne, twórcze, proste.
- Sprowadzalność. Stopnie nierozstrzygalności.
- Informacja o innych matematycznych reprezentacjach pojęcia obliczalności.

Plan wykładu

Uwaga. Niniejsza prezentacja w żadnej mierze nie jest przybliżonym choćby wykładem teorii funkcji rekurencyjnych. Ograniczamy się do definicji wybranych pojęć, podajemy nieco przykładów i formułujemy kilka twierdzeń.

Cele tej prezentacji są zasadniczo dwa:

- pokazanie, że intuicyjne pojęcie obliczalności można reprezentować matematycznie;
- oswojenie słuchaczy z operacjami kodowania, które będą wykorzystywane w arytmetyzacji składni w wykładzie następnym.

Słuchacze zainteresowani teorią funkcji rekurencyjnych zechcą skorzystać z literatury przedmiotu podanej na końcu prezentacji.

Pojęcie algorytmu

Metoda obliczalna (efektywna): w skończonej liczbie prostych, mechanicznych kroków daje odpowiedź dla dowolnych danych ustalonej postaci. Dane mogą mieć strukturę złożoną. Obliczenie za pomocą metody efektywnej nazywa się ***algorytmem***.
Podane wyżej pojęcie obliczalności ma charakter **intuicyjny**. Możliwe są jego różne matematyczne precyzacje.

- Przykład metody efektywnej: algorytm ustalania, czy dana formuła języka Klasycznego Rachunku Zdań jest prawem (tautologią) tego rachunku.
- Przykład problemu, dla którego *nie istnieje* metoda obliczalna: ustalanie, czy dowolna formuła języka Klasycznego Rachunku Predykatów jest prawem (tautologią) tego rachunku.

Przyjmowane oznaczenia

- ω oznacza zbiór wszystkich liczb naturalnych.
- $dom(f)$ oznacza dziedzinę funkcji f, a $rng(f)$ jej przeciwdziedzinę. Podobnie dla relacji.
- Używamy standardowo przyjętych symboli dla operacji i relacji arytmetycznych.
- Zakładamy, że słuchacze pamiętają podstawowe własności operacji i relacji arytmetycznych.
- Zakładamy, że słuchacze znają podstawowe pojęcia rachunku zbiorów i relacji.

W dalszym ciągu, używając terminu „zbiór" będziemy mieli na myśli tylko podzbiory zbioru ω wszystkich liczb naturalnych, zaś ***zbiorami n-tek*** (***ciągów długości n***) będziemy nazywać podzbiory zbioru ω^n ($n \geqslant 1$).

Rekursja = ?

- Na pierwszym polu szachownicy umieszczamy 1 ziarno zboża. Na każdym następnym dwa razy tyle, co na poprzednim. Ile ziaren potrzeba (szachownica ma 64 pola)?
  $1 + 2 + 2^2 + \ldots + 2^{63} = 2^{64} - 1 = 18446744073709551615$
- Mamy pierwszą parę królików (samca i samicę). Chcemy obliczyć, ile par królików otrzymamy po n miesiącach przy założeniu, że każda para królików rodzi co miesiąc nową parę (samca i samicę), która staje się reproduktywna po miesiącu. Nadto, króliki żyją wiecznie, są monogamiczne i kazirodcze (począwszy od drugiej pary tylko brat z siostrą dają potomstwo), oraz nie ustają w rozmnażaniu (pierwsza para też). [Jest również wersja ze śmiertelnymi królikami.]
  - $f(0) = 0$
  - $f(1) = 1$
  - $f(n+2) = f(n) + f(n+1)$

Rekursja = ?

- Łacińskie *recurrere* znaczy m.in.: przybiec na powrót, wrócić do czegoś, pospieszyć na powrót, a *recursare* znaczy: wracać, często powracać.
- Obliczanie wartości pewnych funkcji dla argumentu $n+1$ wymaga znajomości wartości tych funkcji dla argumentów $0, 1, 2, \ldots, n$.

Rozważamy przy tym nie całkiem dowolne funkcje, ale jedynie takie, których wartości otrzymujemy w sposób ***efektywny*** (***obliczalny***). Za funkcje obliczalne uznamy np. funkcje stałe (nic nie trzeba liczyć), funkcję następnika (dodanie jedynki do liczby), funkcje rzutu (wybranie liczby ze skończonej listy). Uznamy, że składając funkcje obliczalne otrzymamy funkcję obliczalną. Uznamy też, że pewne inne operacje (rekursja prosta, minimum efektywne) stosowane do funkcji obliczalnych dają funkcje obliczalne.

Funkcje częściowe i całkowite

Częściowe funkcje liczbowe $f(x_1, \ldots, x_n)$ (dla $n = 1, 2, \ldots$), to funkcje określone na pewnym podzbiorze zbioru ω^n o wartościach będących liczbami naturalnymi.

Dla dowolnych liczb $a_1, \ldots, a_n \in \omega$ oraz funkcji f (k-argumentowej) i g (s-argumentowej) piszemy

$$f(a_{j_1}, \ldots, a_{j_k}) = g(a_{j_1}, \ldots, a_{j_s}),$$

jeśli: albo wartości $(a_{j_1}, \ldots, a_{j_k})$ oraz $g(a_{j_1}, \ldots, a_{j_s})$ są nieokreślone albo są obie określone i identyczne.

n-argumentowa funkcja $f(x_1, \ldots, x_n)$ jest ***całkowita***, jeśli jej dziedziną jest cały zbiór ω^n, czyli gdy $dom(f) = \omega^n$.

Funkcje proste, złożenie i podstawienie

Następujące funkcje całkowite nazywamy ***prostymi***:

- $s(x) = x + 1$ (następnik),
- $o(x) = 0$ (funkcja stała równa 0),
- $I^n_m(x_1, \ldots, x_n) = x_m$ (dla $1 \leqslant m \leqslant n$) (rzut na m-tą współrzędną).

Funkcja $h(x_1, \ldots, x_n) = g(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n))$ otrzymywana jest z funkcji $g, f_1, \ldots, f_m$ przez operację ***złożenia***.

Funkcję $h(x_1, \ldots, x_n) = g(t_1, \ldots, t_m)$ otrzymujemy z pomocą operacji ***podstawienia*** z funkcji $g, f_1, \ldots, f_k$, gdy $t_i = f_j(x_{j_1}, \ldots, x_{j_s})$, gdzie każde x_{j_l} jest jedną ze zmiennych $x_1, \ldots, x_n$ lub t_i jest jedną ze zmiennych $x_1, \ldots, x_n$.

Schemat rekursji prostej

Funkcję $f(x_1, \ldots, x_n, y)$ otrzymujemy z funkcji $g(x_1, \ldots, x_n)$ oraz $h(x_1, \ldots, x_n, y, z)$ za pomocą ***operatora rekursji prostej***, gdy może ona być określona następującym ***schematem rekursji prostej***:

- $f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n)$,
- $f(x_1, \ldots, x_n, y + 1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y))$.

Dla $n = 0$ schemat rekursji prostej przyjmuje następującą postać:

- $f(0) = a$,
- $f(y + 1) = g(y, f(y))$,

gdzie a jest jednoargumentową funkcją stałą o wartości a.

Operacja minimum

Funkcję $f(x_1, \ldots, x_n)$ otrzymujemy z funkcji $g(x_1, \ldots, x_n, y)$ za pomocą operacji ***minimum*** (za pomocą μ-***operatora***), co zaznaczamy następująco:

$$f(x_1, \ldots, x_n) = \mu y \, [g(x_1, \ldots, x_n, y) = 0],$$

gdy spełniony jest warunek:

$f(x_1, \ldots, x_n)$ jest określona i równa y wtedy i tylko wtedy, gdy $g(x_1, \ldots, x_n, 0), \ldots, g(x_1, \ldots, x_n, y-1)$ są wszystkie określone i różne od 0, zaś $g(x_1, \ldots, x_n, y) = 0$.

Funkcje: pierwotnie, częściowo i ogólnie rekurencyjne

- Funkcja $f(x_1, \ldots, x_n)$ jest ***pierwotnie rekurencyjna***, jeśli może być otrzymana z funkcji prostych za pomocą skończonej liczby zastosowań operacji złożenia oraz rekursji prostej.
- Funkcja $f(x_1, \ldots, x_n)$ jest ***częściowo rekurencyjna***, jeśli może być otrzymana z funkcji prostych za pomocą skończonej liczby zastosowań operacji złożenia, rekursji prostej oraz operacji minimum.
- Funkcja $f(x_1, \ldots, x_n)$ jest ***ogólnie rekurencyjna*** (krótko: ***rekurencyjna***), gdy jest ona całkowitą funkcją częściowo rekurencyjną.

Każda funkcja pierwotnie rekurencyjna jest też ogólnie rekurencyjna (lecz nie na odwrót).

Minimum efektywne

Jeżeli funkcja $g(x_1, \ldots, x_n)$ jest pierwotnie rekurencyjna oraz

- $(*)$ dla wszystkich $x_1, \ldots, x_n$ istnieje y taka, że $g(x_1, \ldots, x_n, y) = 0$,

to mówimy, że funkcja $f(x_1, \ldots, x_n)$ określona warunkiem:

$$f(x_1, \ldots, x_n) = \mu y\ [g(x_1, \ldots, x_n, y) = 0]$$

(czyli $f(x_1, \ldots, x_n) =$ najmniejsza y taka, że $g(x_1, \ldots, x_n, y) = 0$) jest zdefiniowana przez ***minimum efektywne***. Warunek $(*)$ nazywamy ***warunkiem efektywności***.

Funkcje określone za pomocą minimum efektywnego stosowanego do funkcji całkowitych są całkowite.

Ograniczony μ-operator

Funkcję $f(x_1, \ldots, x_n)$ otrzymujemy z funkcji

$$g(x_1, \ldots, x_n, y) \text{ oraz } h(x_1, \ldots, x_n)$$

za pomocą ***ograniczonego μ-operatora***, jeśli dla wszystkich $x_1, \ldots, x_n$:

$$\mu y[g(x_1, \ldots, x_n, y) = 0]$$

jest określona i nie większa od $h(x_1, \ldots, x_n)$ oraz

$$f(x_1, \ldots, x_n) = \mu y\ [g(x_1, \ldots, x_n, y) = 0].$$

W szczególności, jeśli $h(x_1, \ldots, x_n) = a$ jest funkcją stałą, to piszemy: $f(x_1, \ldots, x_n) = \mu y \leqslant a\ [g(x_1, \ldots, x_n, y) = 0]$.

Zbiory i relacje rekurencyjne

Niech P będzie dowolną n-argumentową relacją na zbiorze ω. Funkcję $\chi_P(x_1, \ldots, x_n)$ nazywamy ***funkcją charakterystyczną relacji P***, jeśli funkcja ta spełnia warunek

$$\chi_P(x_1, \ldots, x_n) = \begin{cases} 0, & \text{gdy} \quad P(x_1, \ldots, x_n) \text{ zachodzi,} \\ 1, & \text{gdy} \quad P(x_1, \ldots, x_n) \text{ nie zachodzi.} \end{cases}$$

W szczególności, funkcje charakterystyczne zbiorów to funkcje charakterystyczne relacji jednoargumentowych.

Uwaga. W niektórych podręcznikach przez funkcję charakterystyczną relacji P rozumie się funkcję $1 - \chi_P(x_1, \ldots, x_n)$. To kwestia umowy, przyjmowanej dla wygody obliczeń.

Zbiory i relacje rekurencyjne

Relacja P jest ***rekurencyjna*** (***pierwotnie rekurencyjna***), jeśli jej funkcja charakterystyczna jest ogólnie rekurencyjna (pierwotnie rekurencyjna).

Zbiór n-tek M nazywamy ***rekurencyjnym*** (***pierwotnie rekurencyjnym***), jeśli relacja

$$P(x_1, \ldots, x_n) \text{ zachodzi} \Leftrightarrow (x_1, \ldots, x_n) \in M$$

jest rekurencyjna (pierwotnie rekurencyjna).

W szczególności, zbiór $M \subseteq \omega$ jest rekurencyjny, gdy jego funkcja charakterystyczna χ_M jest rekurencyjna.

Funkcje rekurencyjne: inna definicja

W myśl innej definicji ***zbiór funkcji rekurencyjnych*** to najmniejszy zbiór funkcji:

- zawierający funkcje rzutu I^n_m, dodawania $+$, mnożenia $\cdot$ i funkcję charakterystyczną $\chi_<$ relacji mniejszości $<$,
- domknięty na operacje złożenia funkcji oraz definiowanie przez minimum efektywne.

Obie te definicji określają tę samą klasę funkcji:

- można pokazać, że $+$, $\cdot$ i $\chi_<$ są pierwotnie rekurencyjne (zobacz niżej);
- można pokazać, że schemat rekursji prostej daje się wyrazić za pomocą minimum efektywnego (zobacz następny wykład).

Inne schematy rekursji

Funkcję f otrzymujemy z $g, h, t_1, \ldots, t_s$ z pomocą schematu ***rekursji zwrotnej***, gdy może ona być określona schematem:

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n),$$
$$f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, t_1(y+1)), \ldots$$
$$\ldots, f(x_1, \ldots, x_n, t_s(y+1))),$$

gdzie $t_1(y+1) \leqslant y, \ldots, t_s(y+1) \leqslant y$.

Jeśli funkcje $g, h, t_1, \ldots t_s$ są pierwotnie rekurencyjne, to funkcja f jest pierwotnie rekurencyjna.

Inne schematy rekursji

Niech $f_1, \ldots, f_k$ będą zdefiniowane przez ***rekursję jednoczesną***, tzn. za pomocą następującego schematu:

$$\begin{cases} f_i(x_1, \ldots, x_n, 0) = g_i(x_1, \ldots, x_n), \\ f_i(x_1, \ldots, x_n, y+1) = \\ = h_i(x_1, \ldots, x_n, y, f_1(x_1, \ldots, x_n, y), \ldots, f_k(x_1, \ldots, x_n, y)) \end{cases}$$

dla wszystkich $1 \leqslant i \leqslant k$.

Można udowodnić, że jeśli funkcje $g_1, \ldots, g_k, h_1, \ldots, h_k$ są pierwotnie rekurencyjne, to funkcje $f_1, \ldots, f_k$ są pierwotnie rekurencyjne.

Inne schematy rekursji

Schemat ***rekursji ograniczonej*** ma postać następującą:

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n)$$

$$f(x_1, \ldots, x_n, x+1) = h(x_1, \ldots, x_n, x, f(x_1, \ldots, x_n, x))$$

$$f(x_1, \ldots, x_n, x) \leqslant j(x_1, \ldots, x_n, x).$$

Możliwe są różne dalsze schematy rekursji.
Definiowanie przez rekursję to ważne narzędzie w językach programowania.

Oznaczenia

Stosujemy oznaczenie:

$$f(x_1, \ldots, x_n) = \mu y[g(x_1, \ldots, x_n, y) = h(x_1, \ldots, x_n, y)],$$

gdy spełniony jest warunek: $f(x_1, \ldots, x_n)$ jest określone i równe y wtedy i tylko wtedy, gdy $g(x_1, \ldots, x_n, i)$ oraz $h(x_1, \ldots, x_n, i)$ są określone dla $i = 0, 1, \ldots, y$, $g(x_1, \ldots, x_n, i) \neq h(x_1, \ldots, x_n, i)$ dla $i < y$ oraz $g(x_1, \ldots, x_n, y) = h(x_1, \ldots, x_n, y)$.

W podobny sposób rozumiemy oznaczenia:

- $\mu y[g(x_1, \ldots, x_n, y) \neq h(x_1, \ldots, x_n, y)]$,
- $\mu y[g(x_1, \ldots, x_n, y) \leqslant h(x_1, \ldots, x_n, y)]$,
- $\mu y[g(x_1, \ldots, x_n, y) < h(x_1, \ldots, x_n, y)]$, itd.

Iteracja, odwrócenie, funkcja uniwersalna

Będziemy mówić, że funkcję $f(x)$ otrzymujemy z funkcji $g(x)$ przez ***iterację*** i oznaczać ten fakt przez $f(x) = \mathrm{i}g(x)$, gdy

- $f(0) = 0$,
- $f(x+1) = g(f(x))$.

Będziemy mówić, że funkcję $f(x)$ otrzymujemy z funkcji $g(x)$ przez ***odwrócenie*** i zaznaczać ten fakt przez $f(x) = g^{-1}(x)$, gdy

$$f(x) = \mu y[g(y) = x].$$

Niech $\boldsymbol{G}$ będzie pewną rodziną n-argumentowych funkcji częściowych. Funkcję $n+1$-argumentową F nazwiemy ***funkcją uniwersalną*** dla $\boldsymbol{G}$, jeśli

$$\boldsymbol{G} = \{F(0, x_1, \ldots, x_n), F(1, x_1, \ldots, x_n), F(2, x_1, \ldots, x_n), \ldots\}.$$

Przykłady funkcji pierwotnie rekurencyjnych

Następujące funkcje są pierwotnie rekurencyjne:

- (1) $f(x) = x + n$;
- (2) $f(x) = n$;
- (3) $f(x, y) = x + y$;
- (4) $f(x, y) = x \cdot y$;
- (5) $f(x, y) = x^y$ (przyjmujemy $0^0 = 1$);
- (6) $f(x) = x!$ (przyjmujemy $0! = 1$).

Przykłady funkcji pierwotnie rekurencyjnych

Dowód (pierwotnej rekurencyjności tych funkcji).

- (1) $f(x) = \underbrace{s(s(\ldots s}_{n\ razy}(x)\ldots))$.
- (2) $f(x) = \underbrace{s(s(\ldots s}_{n\ razy}(o(x))\ldots))$.
- (3) $f(x,y)$ otrzymujemy przez rekursję prostą z funkcji $g(x) = I^1_1(x)$ oraz $h(x,y,z) = s(I^3_3(x,y,z))$.
- (4) $f(x,y)$ otrzymujemy przez rekursję prostą z $g(x) = o(x)$ i $h(x,y,z) = I^3_1(x,y,z) + I^3_3(x,y,z)$.
- (5) $f(x,0) = 1,\ f(x,y+1) = x \cdot f(x,y)$.
- (6) $f(0) = 1,\ f(x+1) = s(x) \cdot f(x)$.

Przykłady funkcji pierwotnie rekurencyjnych

Następujące funkcje są pierwotnie rekurencyjne:

- (7) $sg(x) = \begin{cases} 0, & \text{gdy} \quad x = 0 \\ 1, & \text{gdy} \quad x > 0; \end{cases}$
- (8) $\overline{sg}(x) = \begin{cases} 0, & \text{gdy} \quad x > 0 \\ 1, & \text{gdy} \quad x = 0; \end{cases}$
- (9) $x \dot{-} 1 = \begin{cases} 0, & \text{gdy} \quad x = 0 \\ x - 1, & \text{gdy} \quad x > 0; \end{cases}$
- (10) $x \dot{-} y = \begin{cases} 0, & \text{gdy} \quad x \leqslant y \\ x - y, & \text{gdy} \quad x > y; \end{cases}$
- (11) $|x - y|$;
- (12) $\max(x, y)$;
- (13) $\min(x, y)$.

Przykłady funkcji pierwotnie rekurencyjnych

Dowód.

- (7) $\mathrm{sg}\, 0 = 0$; $\mathrm{sg}(x+1) = s(o(x))$.
- (8) $\overline{\mathrm{sg}} 0 = 1$, $\overline{\mathrm{sg}}(x+1) = o(x)$.
- (9) $0 \dot{-} 1 = 0$, $(x+1) \dot{-} 1 = x$.
- (10) $x \dot{-} 0 = x$, $x \dot{-} (y+1) = (x \dot{-} y) \dot{-} 1$.
- (11) $|x - y| = (x \dot{-} y) + (y \dot{-} x)$.
- (12) $\max(x, y) = x \cdot \mathrm{sg}(x \dot{-} y) + y \cdot \overline{\mathrm{sg}}(x \dot{-} y)$.
- (13) $\min(x, y) = x \cdot \mathrm{sg}(y \dot{-} x) + y \cdot \overline{\mathrm{sg}}(y \dot{-} x)$.

Przykłady funkcji pierwotnie rekurencyjnych

Niech funkcje $f_0, f_1, \ldots, f_s$ mają następującą własność: dla dowolnych argumentów $x_1, \ldots, x_n$ jedna i tylko jedna z tych funkcji równa jest 0. Powiemy, że funkcja g jest ***określona warunkowo***, gdy

$$g(x_1, \ldots, x_n) = \begin{cases} h_0(x_1, \ldots, x_n), & \text{gdy} \quad f_0(x_1, \ldots, x_n) = 0, \\ \ldots & \ldots \quad \ldots \\ h_s(x_1, \ldots, x_n), & \text{gdy} \quad f_s(x_1, \ldots, x_n) = 0. \end{cases}$$

Jeśli funkcje $h_0, \ldots, h_s, f_0, \ldots, f_s$ są pierwotnie rekurencyjne, to g jest pierwotnie rekurencyjna.

Dowód. $g(x_1, \ldots, x_n) = h_0(x_1, \ldots, x_n) \cdot \overline{\mathrm{sg}}\, f_0(x_1, \ldots, x_n) + \ldots$
$$\ldots + h_s(x_1, \ldots, x_n) \cdot \overline{\mathrm{sg}}\, f_s(x_1, \ldots, x_n).$$

Przykłady funkcji pierwotnie rekurencyjnych

Niech g, α, β będą funkcjami pierwotnie rekurencyjnymi. Wtedy następujące funkcje są pierwotnie rekurencyjne:

(14) $f(x_1, \ldots, x_n, x_{n+1}) = \sum\limits_{i=0}^{x_{n+1}} g(x_1, \ldots, x_n, i)$;

(15) $f(x_1, \ldots, x_n, y, z) = \begin{cases} \sum\limits_{i=y}^{z} g(x_1, \ldots, x_n, i), & \text{gdy} \quad y \leqslant z, \\ 0, & \text{gdy} \quad y > z; \end{cases}$

(16) $f(x_1, \ldots, x_n, y_1, \ldots, y_m) =$

$$= \begin{cases} \sum\limits_{i=\alpha(y_1,\ldots,y_m)}^{\beta(y_1,\ldots,y_m)} g(x_1, \ldots, x_n, i), & \text{gdy } \alpha(y_1, \ldots, y_m) \leqslant \beta(y_1, \ldots, y_m), \\ 0, & \text{w pozostałych przypadkach}; \end{cases}$$

Przykłady funkcji pierwotnie rekurencyjnych

(17) $f(x_1,\ldots,x_n,x_{n+1}) = \prod_{i=0}^{x_{n+1}} g(x_1,\ldots,x_n,i)$;

(18) $f(x_1,\ldots,x_n,y,z) = \begin{cases} \prod_{i=y}^{z} g(x_1,\ldots,x_n,i), & \text{gdy} \quad y \leqslant z \\ 0, & \text{gdy} \quad y > z; \end{cases}$

(19) $f(x_1,\ldots,x_n,y_1,\ldots,y_m) =$

$$= \begin{cases} \prod_{i=\alpha(y_1,\ldots,y_m)}^{\beta(y_1,\ldots,y_m)} g(x_1,\ldots,x_n,i), & \text{gdy } \alpha(y_1,\ldots,y_m) \leqslant \beta(y_1,\ldots,y_m), \\ 0, & \text{w pozostałych przypadkach.} \end{cases}$$

Jeśli funkcję f otrzymujemy z funkcji pierwotnie rekurencyjnych g i h za pomocą ograniczonego μ-operatora, to f jest pierwotnie rekurencyjna.

Dowód. $f(x_1,\ldots,x_n) = \sum_{i=0}^{h(x_1,\ldots,x_n)} \mathrm{sg}(\prod_{j=0}^{i} g(x_1,\ldots,x_n,j)).$

Przykłady funkcji pierwotnie rekurencyjnych

Następujące funkcje są pierwotnie rekurencyjne:

- (20) $[\frac{x}{y}]$ — część całkowita z dzielenia x przez y (przyjmujemy, że $[\frac{x}{0}] = x$);
- (21) $\mathrm{rest}(x, y)$ — reszta z dzielenia x przez y (przyjmujemy, że $\mathrm{rest}(x, 0) = x$);
- (22) $\tau(x)$ — liczba dzielników liczby x, gdzie $\tau(0) = 0$;
- (23) $\sigma(x)$ — suma dzielników liczby x, gdzie $\sigma(0) = 0$;
- (24) $\mathrm{lh}(x)$ — liczba dzielników liczby x, które są liczbami pierwszymi (przyjmujemy $\mathrm{lh}(0) = 0$);
- (25) $\pi(x)$ — liczba liczb pierwszych nie większych niż x;
- (26) $k(x, y)$ — najmniejsza wspólna wielokrotność liczb x i y, gdzie $k(x, 0) = k(0, y) = 0$;
- (27) $d(x, y)$ — największy wspólny dzielnik liczb x i y, gdzie $d(0, 0) = 0$.

Przykłady funkcji pierwotnie rekurencyjnych

(20) $[\frac{x}{y}] = \sum_{i=1}^{x} \overline{\mathrm{sg}}\,(iy \dot{-} x)$.

(21) $\mathrm{rest}\,(x, y) = x \dot{-} y[\frac{x}{y}]$.

(22) $\tau(x) = \sum_{i=1}^{x} \overline{\mathrm{sg}}\,(\mathrm{rest}\,(x, i))$.

(23) $\sigma(x) = \sum_{i=1}^{x} i\overline{\mathrm{sg}}\,(\mathrm{rest}\,(x, i))$.

(24) x jest liczbą pierwszą $\Leftrightarrow \tau(x) = 2$ (zob. (22));

$\mathrm{lh}\,(x) = \sum_{i=1}^{x} \overline{\mathrm{sg}}\,(|\tau(i) - 2| + \mathrm{rest}\,(x, i))$.

(25) $\pi(x) = \sum_{i=1}^{x} \overline{\mathrm{sg}}\,(|\tau(i) - 2|)$ (zob. (22)).

(26) $k(x, y) = \mu z[z \cdot \overline{\mathrm{sg}}\,(x \cdot y) + \mathrm{sg}(x \cdot y)(\overline{\mathrm{sg}}\,z +$
$+\mathrm{rest}\,(z, x) + \mathrm{rest}\,(z, y)) = 0] \leqslant x \cdot y$.

(27) $d(x, y) = [\frac{xy}{k(x,y)}] + x \cdot \overline{\mathrm{sg}}\,y + y \cdot \overline{\mathrm{sg}}\,x$ (zob. (26)).

Przykłady funkcji pierwotnie rekurencyjnych

Następujące funkcje są pierwotnie rekurencyjne:

- (28) p_x — x-ta liczba pierwsza ($p_0 = 2, p_1 = 3, p_2 = 5, \ldots$);
- (29) $\mathrm{long}(x)$ – numer największego dzielnika liczby x, będącego liczbą pierwszą;
- (30) $\mathrm{ex}(x, y)$ — wykładnik potęgi x-tej liczby pierwszej p_x w kanonicznym rozkładzie liczby y na czynniki pierwsze; przyjmujemy, że $\mathrm{ex}(x, 0) = 0$;

Dowód.

- (28) $p_x = \mu y[|\pi(y) - (x+1)| = 0] \leqslant 2^{2^x}$ (zob. (25)).
- (29) $\mathrm{long}\,(x) = \mu y[\sum\limits_{i=y+1}^{x} \overline{\mathrm{sg}}\,(\mathrm{rest}\,(x, p_i)) = 0] \leqslant x$.
- (30) $\mathrm{ex}\,(x, y) = \mu z[(\overline{\mathrm{sg}}\,\mathrm{rest}\,(y, (p_x)^{z+1})) \cdot \mathrm{sg}\, y = 0] \leqslant x$ (zob. (28)).

Przykłady funkcji pierwotnie rekurencyjnych

- Jest nieskończenie wiele funkcji pierwotnie (częściowo, ogólnie) rekurencyjnych.
- Każda z tych klas zawiera jednak tylko $\aleph_0$ funkcji.
- Ponieważ ***wszystkich*** funkcji z ω^n w ω jest ***kontinuum***, więc ***prawie wszystkie*** funkcje są poza klasą funkcji ogólnie rekurencyjnych.

- Przykłady funkcji, które:
  - są rekurencyjne, ale nie sa pierwotnie rekurencyjne;,
  - nie są rekurencyjne,

  zostaną podane niżej. Czy spotkałaś już funkcję, która nie jest rekurencyjna? Na czym miałoby polegać ***obliczanie*** jej wartości?

Własności zbiorów i relacji rekurencyjnych

Następujące relacje są pierwotnie rekurencyjne:

- (a) $x = y$;
- (b) $x + y = z$;
- (c) $x \cdot y = z$;
- (d) x dzieli y;
- (e) x jest parzyste;
- (f) x oraz y są względnie pierwsze;
- (g) $\exists n(x = 1^2 + 2^2 + \ldots + n^2)$;
- (h) $\exists n(x = 1 + 2 + \ldots + n)$.

Własności zbiorów i relacji rekurencyjnych

Jeżeli relacje $P(x_1, \ldots, x_n)$ oraz $Q(x_1, \ldots, x_n)$ są rekurencyjne (pierwotnie rekurencyjne), to następujące relacje są również rekurencyjne (pierwotnie rekurencyjne):

- (a) $(P(x_1, \ldots, x_n) \wedge Q(x_1, \ldots, x_n))$;
- (b) $(P(x_1, \ldots, x_n) \vee Q(x_1, \ldots, x_n))$;
- (c) $\neg P(x_1, \ldots, x_n)$;
- (d) $(P(x_1, \ldots, x_n) \rightarrow Q(x_1, \ldots, x_n))$;
- (e) $P(x_1, x_1, x_3, \ldots, x_n)$;
- (f) $P(f(x_1, \ldots, x_m), x_{m+1}, \ldots, x_{m+n-1})$, jeśli $f(x_1, \ldots, x_m)$ jest rekurencyjna (pierwotnie rekurencyjna).

Własności zbiorów i relacji rekurencyjnych

- Jeżeli relacja $R(x_1, \ldots, x_n, y)$ jest rekurencyjna (pierwotnie rekurencyjna), to relacje $\exists y(y \leqslant z \wedge R(x_1, \ldots, x_n, y))$ oraz $\forall y(y \leqslant z \rightarrow R(x_1, \ldots, x_n, y))$ również są rekurencyjne (pierwotnie rekurencyjne).
- Dowolny skończony zbiór liczb naturalnych jest pierwotnie rekurencyjny.
- Jeżeli f jest funkcją ogólnie rekurencyjną (pierwotnie rekurencyjną), zaś a jest ustaloną liczbą, to zbiór rozwiązań równania $f(x_1, \ldots, x_n) = a$ jest rekurencyjny (pierwotnie rekurencyjny).
- Niech f będzie funkcją częściowo, ale nie ogólnie rekurencyjną. Wtedy dziedzina funkcji f^{-1} jest zbiorem pierwotnie rekurencyjnym.
- Jeżeli zbiory A oraz B są rekurencyjne (pierwotnie rekurencyjne), to również zbiory $A \cap B$, $A \cup B$, $\omega - A$ są rekurencyjne (pierwotnie rekurencyjne).

Funkcja numerująca Cantora

$$
\begin{array}{ccccccccc}
 & & & & (0,0) & & & & \\
 & & & (1,0) & & (0,1) & & & \\
 & & (2,0) & & (1,1) & & (0,2) & & \\
 & (3,0) & & (2,1) & & (1,2) & & (0,3) & \\
(4,0) & & (3,1) & & (2,2) & & (1,3) & & (0,4) \\
 & & & & \vdots & & & &
\end{array}
$$

Ustawimy te wszystkie pary w ***jeden*** ciąg nieskończony, wedle porządku pięter tej nieskończonej piramidy (z góry w dół), a w ramach każdego pietra z lewej do prawej.

Funkcja numerująca Cantora

Funkcja numerująca Cantora

$$c(x, y) = \frac{(x + y)^2 + 3x + y}{2}$$

ustanawia wzajemnie jednoznaczną odpowiedniość między $\omega \times \omega$ a ω (koduje pary liczb naturalnych wedle wspomnianego wyżej porządku).

Funkcja c jest pierwotnie rekurencyjna. Takie są też funkcje:

$l(x) = \mu z \leqslant x\ (\exists t \leqslant x\ c(z, t) = x)$
$r(x) = \mu z \leqslant x\ (\exists t \leqslant x\ c(t, z) = x)$.

Nadto, $l(c(x, y)) = x$ oraz $r(c(x, y)) = y$, a także $c(l(z), r(z)) = z$. Funkcje l oraz r również są wzajemnie jednoznaczne.

Funkcja numerująca Cantora

Analogicznie kodujemy trójki liczb naturalnych: $c^3(x, y, z) = c(x, c(y, z))$ oraz dowolne n-tki takich liczb:

- $c^1(x) = x$
- $c^2(x_1, x_2) = c(x_1, x_2)$
- $c^{n+1}(x_1, x_2, \ldots, x_n, x_{n+1}) = c^n(x_1, x_2, \ldots, c(x_n, x_{n+1}))$.

- $c_1^1(x) = x$,
- $c_1^2(x) = l(x)$,
- $c_2^2(x) = r(x)$,
- $c_1^{n+1} = c_n^1, \;\; c_2^{n+1} = c_2^n, \; \ldots, \; c_{n-1}^{n+1} = c_{n-1}^n$
- $c_n^{n+1} = c_1^2 \circ c_n^n, \quad c_{n+1}^{n+1} = c_2^2 \circ c_n^n$, gdzie $\circ$ jest operacją złożenia funkcji.

Funkcja numerująca Cantora

- Funkcje c^n oraz c^n_m są pierwotnie rekurencyjne.
- Funkcje $c^n(x_1, \ldots, x_n)$ ustanawiają wzajemnie jednoznaczne odpowiedniości między ω^n oraz ω (kodują ciągi liczb naturalnych długości n).
- Z jednoargumentowych funkcji częściowo rekurencyjnych oraz z funkcji $c^n(x_1, \ldots, x_n)$ otrzymać można wszystkie funkcje częściowo rekurencyjne.
- Istnieje wzajemnie jednoznaczna funkcja pierwotnie rekurencyjna ze zbioru ***wszystkich ciągów skończonych*** liczb naturalnych w zbiór ω.

Jedną z zalet kodowania z użyciem funkcji Cantora jest to, że jej zbiór wartości jest równy całemu zbiorowi ω.

Funkcje uniwersalne

Dla każdej liczby naturalnej n istnieją funkcje uniwersalne dla klas wszystkich n-argumentowych funkcji:

- pierwotnie rekurencyjnych;
- ogólnie rekurencyjnych.

Można udowodnić, że dla każdej liczby naturalnej n istnieje ***ogólnie rekurencyjna*** funkcja uniwersalna dla klasy wszystkich n-argumentowych funkcji pierwotnie rekurencyjnych. W dowodzie tego twierdzenia wykorzystuje się możliwość kodowania liczb naturalnych. W szczególności, wykazuje się, że zakodować można: definiowanie przez schemat rekursji prostej, definiowanie przez złożenie oraz definiowanie przez operację minimum efektywnego (zob. też niżej, w punkcie o indeksach funkcji).

Funkcje uniwersalne

Twierdzenie A. Nie istnieje funkcja pierwotnie rekurencyjna uniwersalna dla rodziny wszystkich *n*-argumentowych funkcji pierwotnie rekurencyjnych.

Twierdzenie B. Nie istnieje funkcja częściowo rekurencyjna uniwersalna dla rodziny wszystkich *n*-argumentowych funkcji ogólnie rekurencyjnych.

Dowód A. Niech $F(t, x_1, \ldots, x_n)$ będzie funkcją uniwersalną dla rodziny wszystkich *n*-argumentowych funkcji pierwotnie rekurencyjnych i ***przypuśćmy***, że jest ona pierwotnie rekurencyjna. Wtedy $f(x_1, \ldots, x_n) = 1 + F(x_1, x_1, \ldots, x_n) = F(t_0, x_1, \ldots, x_n)$ dla pewnego t_0. Stąd $1 + F(t_0, t_0, \ldots, t_0) = F(t_0, t_0, \ldots, t_0)$. Dochodzimy do sprzeczności.

Dowód B. Zauważmy, że funkcja uniwersalna powinna być wszędzie określona, tzn. całkowita. Dalej, zobacz dowód Twierdzenia A.

Funkcje uniwersalne

Tak więc, choć można skonstruować funkcje uniwersalne dla rodziny wszystkich n-argumentowych funkcji:

- pierwotnie rekurencyjnych;
- ogólnie rekurencyjnych,

to z powyższych twierdzeń A i B otrzymujemy przykłady $(n+1)$-argumentowych funkcji, które ***nie*** są:

- pierwotnie rekurencyjne;
- ogólnie rekurencyjne.

Inna metoda pokazywania, iż jakaś funkcja nie należy do określonej klasy funkcji to dowód, że funkcja ta „rośnie szybciej" niż każda z funkcji tej klasy. W ten sposób pokazuje się np., że funkcja Ackermanna nie jest funkcją pierwotnie rekurencyjną (zob. niżej).

Funkcje elementarnie rekurencyjne

Klasa funkcji ***elementarnie rekurencyjnych*** to najmniejsza klasa funkcji zawierająca funkcje:

- odejmowania $\dot{-}$,
- funkcję wykładniczą,
- funkcję następnika,

oraz zamknięta ze względu na operacje:

- złożenia,
- minimum ograniczonego.

Można udowodnić, że klasa wszystkich funkcji elementarnie rekurencyjnych jest zawarta w klasie wszystkich funkcji pierwotnie rekurencyjnych (i ta inkluzja jest właściwa).

Hierarchia Grzegorczyka

Hierarchia Grzegorczyka. Niech: $f_0(x,y) = y+1$, $f_1(x,y) = x+y$, $f_2(x,y) = (x+1)\cdot(y+1)$, i dla $n \geqslant 2$:
$f_{n+1}(0,y) = f_n(x+1,y+1)$
$f_{n+1}(x+1,y) = f_{n+1}(x, f_{n+1}(x,y))$.

Dla dowolnego n niech E_n będzie najmniejszą klasą funkcji zawierającą funkcje: I_1^2, I_2^2, funkcję następnika oraz funkcję f_n i zamkniętą ze względu na złożenie i schemat rekursji ograniczonej. Wtedy:

- E_3 jest równa klasie funkcji elementarnie rekurencyjnych;
- dla każdego n mamy: $E_n \subset E_{n+1}$ (wszystkie inkluzje właściwe);
- $\bigcup\limits_n E_n$ jest równy klasie wszystkich funkcji pierwotnie rekurencyjnych;
- dla każdego n funkcje f_n są ściśle rosnące względem każdego z argumentów;
- dla każdego n funkcja $f_{n+1}(x,x)$ należy do $E_{n+1} - E_n$.

Funkcja Ackermanna

Mówimy, że funkcja $f(x)$ ***majoryzuje*** funkcję $g(x_1, \ldots, x_n)$, gdy istnieje liczba a taka, że dla wszystkich $x_1, \ldots, x_n$:
$g(x_1, \ldots, x_n) \leqslant f(\max(x_1, \ldots, x_n, a))$.
W szczególności, jeśli f jest ściśle rosnąca, to $f(x)$ majoryzuje $g(x_1, \ldots, x_n)$ dokładnie wtedy, gdy $g(x_1, \ldots, x_n) \leqslant f(\max(x_1, \ldots, x_n))$ dla wszystkich, oprócz skończonej liczby, ciągów $(x_1, \ldots, x_n)$.

- Podamy przykład funkcji, która majoryzuje wszystkie funkcje pierwotnie rekurencyjne: ***funkcji Ackermanna***. Dowodzi się, że (zobacz dowody w dodatkach na stronie wykładów):
- Funkcja Ackermanna nie jest pierwotnie rekurencyjna (właśnie dlatego, że majoryzuje wszystkie funkcje pierwotnie rekurencyjne).
- Funkcja Ackermanna jest rekurencyjna.

Funkcja Ackermanna

Dla dowolnych $m > 0$ oraz $n > 0$ niech:

- $Ack(0, n) = n + 1$
- $Ack(m, 0) = Ack(m - 1, 1)$
- $Ack(m, n) = Ack(m - 1, Ack(m, n - 1))$.

Wprowadźmy też oznaczenia:

- $A_m(n) = A(m, n)$
- $A(n) = A_n(n)$ (czyli $A(n) = Ack(n, n)$).

Funkcję $A(n)$ nazywamy ***funkcją Ackermanna***.

Wspomnienia ze szkoły

Pisząc dalej a^{b^c} mamy na myśli funkcję $a^{(b^c)}$ (a nie funkcję $(a^b)^c$ — sprawdź, że to różne funkcje!). Podobnie, $a^{b^{c^d}}$ to $a^{(b^{(c^d)})}$, itd.
Zdefiniujemy teraz funkcję ${}^b a$ przez warunki:

- ${}^0 a = 1$
- ${}^{(b+1)} a = a^{({}^b a)}$.

Wtedy:

- ${}^1 a = a^{({}^0 a)} = a^1 = a$
- ${}^2 a = a^{({}^1 a)} = a^a$
- ${}^3 a = a^{({}^2 a)} = a^{a^a}$
- ${}^n a = a^{a^{\cdot^{\cdot^{\cdot^{a}}}}}$ (a występuje n razy).

Notacja strzałkowa Knutha

- $a \cdot b = a + a + \ldots + a$ (a występuje b razy)
- $a^b = a \cdot a \cdot \ldots \cdot a$ (a występuje b razy)
- ${}^b a = a^{a^{\cdot^{\cdot^{\cdot^{a}}}}}$ (a występuje b razy).

W literaturze wprowadza się oznaczenia:

- $a \uparrow b = a \cdot a \cdot \ldots \cdot a$ (a występuje b razy)
- $a \uparrow\uparrow b = a \uparrow a \uparrow \ldots \uparrow a$ (a występuje b razy)
- a zatem $a \uparrow\uparrow b = {}^b a = a^{a^{\cdot^{\cdot^{\cdot^{a}}}}}$ (a występuje b razy)
- $a \uparrow\uparrow\uparrow b = a \uparrow\uparrow (a \uparrow\uparrow (\ldots (a \uparrow\uparrow a) \ldots))$ (a występuje b razy)
- $a \uparrow^m b = a \uparrow^{m-1} (a \uparrow^{m-1} \ldots (a \uparrow^{m-1} a) \ldots)$ (a występuje b razy).

Notacja strzałkowa Knutha

Formalnie ciąg funkcji $\uparrow^n$ określamy w następujący sposób:

- $a \uparrow^n b = a^b$, gdy $n = 1$,
- $a \uparrow^n b = 1$, gdy $b = 0$,
- $a \uparrow^n b = a \uparrow^{n-1} (a \uparrow^n (b-1))$, gdy $n > 1$.

Łatwo można pokazać, że np.:

- $1 \uparrow^n b = 1$
- $a \uparrow^n 1 = a$
- $2 \uparrow^n 2 = 4$.

Uwaga. Możesz pisać $\uparrow^n (a, b)$ zamiast $a \uparrow^n b$, wtedy bodaj łatwiej dokonywać rachunków.

Notacja strzałkowa Knutha

Nietrudno też wykazać rachunkiem, że np.:

- $2 \uparrow^2 3 = 2 \uparrow (2 \uparrow 2) = 2 \uparrow 2^2 = 2 \uparrow 4 = 2^4 = 16$
- $2 \uparrow^3 3 = 2 \uparrow^2 (2 \uparrow^2 2) = 2 \uparrow^2 4 = 2 \uparrow (2 \uparrow (2 \uparrow 2)) =$
  $= 2 \uparrow (2 \uparrow 4) = 2 \uparrow (2^4) = 2 \uparrow 16 = 2^{16} = 65536$
- $3 \uparrow^2 2 = 3 \uparrow (3 \uparrow^2 1) = 3 \uparrow (3 \uparrow (3 \uparrow^2 0)) = 3 \uparrow (3 \uparrow 1) = 3 \uparrow 3 =$
  $= 3^3 = 27$
- $3 \uparrow^3 2 = 3 \uparrow^2 (3 \uparrow^3 1) = 3 \uparrow^2 (3 \uparrow^2 (3 \uparrow^3 0)) = 3 \uparrow^2 (3 \uparrow^2 1) =$
  $= 3 \uparrow^2 (3 \uparrow (3 \uparrow^2 0)) = 3 \uparrow^2 (3 \uparrow 1) = 3 \uparrow^2 3 = 3 \uparrow (3 \uparrow^2 2) =$
  $= 3 \uparrow 27 = 3^{27} = 7625597484987.$

Funkcja Ackermanna

Mamy też np.:

- $3 \uparrow^3 3 = 3 \uparrow^2 (3 \uparrow^3 2) = 3 \uparrow^2 (3^{27}) = 3 \uparrow^2 7625597484987 = 3^{3^{\cdot^{\cdot^{\cdot^{3}}}}}$ (liczba 3 występuje tu 7625597484987 razy).

Czytelnik zechce uzasadnić, że:

- $2 \uparrow^3 4 = 2 \uparrow (2 \uparrow \ldots (2 \uparrow 2) \ldots)$ (gdzie 2 występuje 65536 razy)
- $3 \uparrow^2 4 = 3^{3^{27}} = 3^{7625597484987}$
- $3 \uparrow^3 3 = 3 \uparrow (3 \uparrow \ldots (3 \uparrow 3) \ldots)$ (gdzie 3 występuje 3^{27} razy)
- $3 \uparrow^3 4 = 3 \uparrow (3 \uparrow \ldots (3 \uparrow 3) \ldots)$ (gdzie 3 występuje $3 \uparrow^2 3^{27}$ razy).

Funkcja Ackermanna

Dla wszystkich n:

- $A_0(n) = n + 1$
- $A_1(n) = n + 2 = 2 + (n + 3) - 3$
- $A_2(n) = 2n + 3 = 2 \cdot (n + 3) - 3$
- $A_3(n) = 2^{n+3} - 3 = 2 \uparrow (n + 3) - 3$
- $A_4(n) = 2 \uparrow^2 (n + 3) - 3$
- $A_5(n) = 2 \uparrow^3 (n + 3) - 3$
- $A_m(n) = 2 \uparrow^{m-2} (n + 3) - 3$, dla $m > 2$.

Ponadto dla wszystkich $n \geqslant 1$:

- $A_4(n) = 2^{A_4(n-1)+3} - 3$.

Funkcja Ackermanna

Dla wszystkich $m > 0$ wartość $A_m(n)$ jest równa kolejno:

- $A_m(n) = A_{m-1}(A_m(n-1)) =$
- $= A_{m-1}(A_{m-1}(A_m(n-2))) =$
- $= A_{m-1}(A_{m-1}(A_{m-1}(A_m(n-3)))) =$
- $\ldots$
- $= A_{m-1}(A_{m-1}(\ldots A_{m-1}(A_m(0))\ldots)) =$
- $= A_{m-1}(A_{m-1}(\ldots A_{m-1}(A_{m-1}(1))\ldots))$ (tu mamy $n+1$ iteracji funkcji A_{m-1}).

Mamy ponadto: $A_1(1) = 2$, $A_2(1) = 3$, $A_3(1) = 13$, $A_4(1) = 65533$.

Funkcja Ackermanna

Inny wariant notacyjny otrzymujemy określając ciąg funkcji B_m:

- $B_0(n) = n + 1$
- $B_{m+1}(0) = B_m(1)$
- $B_{m+1}(n + 1) = B_m(B_{m+1}(n))$.

Wtedy:

- $B_1(n) = 2 + (n + 3) - 3$
- $B_2(n) = 2 \cdot (n + 3) - 3$
- $B_3(n) = 2 \uparrow (n + 3) - 3$
- $B_4(n) = 2 \uparrow^2 (n + 3) - 3$
- $B_m(n) = 2 \uparrow^{m-2} (n + 3) - 3$.

Funkcja Ackermanna

Zauważmy jeszcze, że jeśli określimy funkcję trójargumentową $Ac(k, m, n)$ następująco:

- $Ac(1, m, n) = m + n$
- $Ac(k + 1, m, 1) = m$
- $Ac(k + 1, m, n + 1) = Ac(k, m, Ac(k + 1, m, n))$,

to zachodzą następujące równości:

- $Ac(0, m, n) = m + n$
- $Ac(1, m, n) = m \cdot n$
- $Ac(2, m, n) = m^n = m \uparrow n$
- $Ac(3, m, n) = {}^n m = m \uparrow^2 n = m^{m^{\cdot^{\cdot^{\cdot^{m}}}}}$ (m występuje tu n razy)
- $Ac(4, m, n) = m \uparrow^3 n$.

Funkcja Ackermanna

Funkcja $Ack(m, n)$ „rośnie bardzo szybko.” Wartość $Ack(4, 2)$ ma w zapisie dziesiętnym 19729 cyfr. Dla dowolnej n wartość $A(4, n)$ możemy zapisać też następująco:

- $Ack(4, n) = {}^{(n+3)}2 - 3$
- $Ack(4, n) = 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} - 3$ (liczba 2 występuje tu $n + 3$ razy).

Widzimy też, że przy rozważaniu tego typu funkcji warto rozważać inne jeszcze (oprócz dodawania, mnożenia i „zwykłego” potęgowania) operacje arytmetyczne na liczbach.

Liczby Ackermanna

Wartości $n \uparrow^n n$ nazywa się ***liczbami Ackermana***. Mamy zatem:

- $1 \uparrow 1 = 1$
- $2 \uparrow^2 2 = {}^2 2 = 4$
- $3 \uparrow^3 3 = {}^{^3 3}3$ (tutaj ${}^{^3 3}3$ rozumiemy jako ${}^{(^3 3)}3$). Ta liczba równa jest 3^{27}, czyli $3^{3^{\cdot^{\cdot^{\cdot^{3}}}}}$ (tu 3 występuje 7625597484987 razy).

Opis arytmetyczny czwartej liczby Ackermanna $4 \uparrow^4 4$ jest dość złożony:

- $4 \uparrow^4 4 = 4 \uparrow^3 (4 \uparrow^3 (4 \uparrow^3 4)) = 4 \uparrow^2 (4 \uparrow^2 \ldots (4 \uparrow^2) \ldots)$, gdzie liczba 4 występuje a razy, gdzie $a = 4 \uparrow^2 (4 \uparrow^2 \ldots (4 \uparrow^2) \ldots)$, gdzie w a liczba 4 występuje $4 \uparrow^2 (4 \uparrow^2 (4 \uparrow^2 (4 \uparrow^2 4)))$ razy.

Zbiory i relacje rekurencyjnie przeliczalne

Zbiór n-tek M nazywamy ***rekurencyjnie przeliczalnym*** (***zbiorem r.e.***), jeśli istnieje $(n+1)$-argumentowa relacja pierwotnie rekurencyjna

$$R_M(x_1, \ldots, x_n, y)$$

spełniająca dla każdych $x_1, \ldots, x_n$ warunek:

$$(x_1, \ldots, x_n) \in M \Leftrightarrow \exists y R_M(x_1, \ldots, x_n, y).$$

Zbiory i relacje rekurencyjnie przeliczalne stanowią matematyczne odpowiedniki pojęć ***pozytywnie obliczalnych***.

Zbiory i relacje rekurencyjnie przeliczalne

Mówimy, że relacja $R \subseteq \omega^n$ jest ***pozytywnie obliczalna*** wtedy i tylko wtedy, gdy dla dowolnego układu liczb naturalnych $a_1, \ldots, a_n$, jeżeli zachodzi $R(a_1, \ldots, a_n)$, to otrzymamy w skończonej liczbie z góry określonych kroków odpowiedź na pytanie: „Czy zachodzi $R(a_1, \ldots, a_n)$?". Jeżeli natomiast ***nie*** zachodzi $R(a_1, \ldots, a_n)$, to możemy nie uzyskać żadnej odpowiedzi na to pytanie.

Przykład. Klasyczny Rachunek Predykatów jest nierozstrzygalny. Nie istnieje efektywna metoda rozstrzygania, czy jakaś formuła języka tego rachunku jest jego tautologią. Klasyczny Rachunek Predykatów jest jednak ***półrozstrzygalny***: własność bycia tautologią tego rachunku jest pozytywnie obliczalna. Np. metoda ***tablic analitycznych*** pozwala, gdy jakaś formuła jest tautologią tego rachunku, dowieść tego w sposób efektywny.

Zbiory i relacje rekurencyjnie przeliczalne

Dla dowolnego zbioru n-tek M zdefiniujemy ***częściową funkcję charakterystyczną*** $\chi^*_M(x_1, \ldots, x_n)$ w sposób następujący:

$$\chi^*_M(x_1, \ldots, x_n) = \begin{cases} 0, & \text{gdy} \quad (x_1, \ldots, x_n) \in M, \\ \text{nie określona}, & \text{gdy} \quad (x_1, \ldots, x_n) \notin M. \end{cases}$$

Jeżeli f jest n-argumentową funkcją częściową, to zbiór

$$\Gamma_f = \{(x_1, \ldots, x_n, f(x_1, \ldots, x_n)) : (x_1, \ldots, x_n) \in dom(f)\}$$

nazywamy ***wykresem*** funkcji f.

Funkcję $f(x_1, \ldots, x_n)$ nazywamy ***dookreśleniem*** funkcji $g(x_1, \ldots, x_n)$, jeśli $\Gamma_g \subseteq \Gamma_f$ oraz $dom(f) = \omega$.

Własności zbiorów i relacji rekurencyjnie przeliczalnych

Podajemy (bez dowodów) niektóre wybrane własności zbiorów i relacji rekurencyjnie przeliczalnych:

- Jeżeli relacja $R(x_1, \ldots, x_n, y, z)$ jest pierwotnie rekurencyjna, to $M = \{(x_1, \ldots, x_n) : \exists y \exists z R(x_1, \ldots, x_n, y, z)\}$ jest zbiorem rekurencyjnie przeliczalnym.
- Jeżeli zbiory A i B są rekurencyjnie przeliczalne, to zbiory $A \cap B$ i $A \cup B$ też są rekurencyjnie przeliczalne.
- Każdy zbiór pierwotnie rekurencyjny jest rekurencyjnie przeliczalny.
- Niech zbiory A i B różnią się skończoną liczbą elementów. Wtedy:
  - (a) jeśli A jest rekurencyjny, to B jest rekurencyjny;
  - (b) jeśli A jest rekurencyjnie przeliczalny, to B jest rekurencyjnie przeliczalny.

Własności zbiorów i relacji rekurencyjnie przeliczalnych

- **Twierdzenie Posta**. Jeżeli zbiór A oraz jego dopełnienie $\omega - A$ są rekurencyjnie przeliczalne, to A jest rekurencyjny.
- Niech $M \subseteq \omega^n$. Przyjmijmy:

$$c^n(M) = \{c^n(x_1, \ldots, x_n) : (x_1, \ldots, x_n) \in M\},$$

  gdzie c^n jest funkcją Cantora kodującą ciągi, zdefiniowaną wcześniej. Wtedy:

  - (a) M jest pierwotnie rekurencyjny wtedy i tylko wtedy, gdy $c^n(M)$ jest pierwotnie rekurencyjny;
  - (b) M jest rekurencyjny wtedy i tylko wtedy, gdy $c^n(M)$ jest rekurencyjny;
  - (c) M jest rekurencyjnie przeliczalny wtedy i tylko wtedy, gdy $c^n(M)$ jest rekurencyjnie przeliczalny.

Własności zbiorów i relacji rekurencyjnie przeliczalnych

- Niech $M \subseteq \omega$ będzie zbiorem niepustym. M jest rekurencyjnie przeliczalny wtedy i tylko wtedy, gdy istnieje funkcja pierwotnie rekurencyjna $f(x)$ taka, że $M = \{f(x) : x \in \omega\}$.
- Niech M będzie niepustym zbiorem n-tek. Wtedy zbiór M jest rekurencyjnie przeliczalny wtedy i tylko wtedy, gdy istnieją jednoargumentowe funkcje pierwotnie rekurencyjne $f_1, \ldots, f_n$ takie, że:

$$M = \{(f_1(x), \ldots, f_n(x)) : x \in \omega\}.$$

- Niech funkcja ogólnie rekurencyjna $f(x)$ spełnia warunek: $f(x) \geqslant x$ dla wszystkich $x \in \omega$. Wtedy zbiór wartości $rng(f)$ funkcji f jest rekurencyjny.

Własności zbiorów i relacji rekurencyjnie przeliczalnych

- Zbiór nieskończony A jest rekurencyjny wtedy i tylko wtedy, gdy A jest zbiorem wartości ściśle rosnącej funkcji ogólnie rekurencyjnej.
- Niepusty zbiór A jest rekurencyjny wtedy i tylko wtedy, gdy A jest zbiorem wartości rosnącej (niekoniecznie ściśle) funkcji ogólnie rekurencyjnej.
- Każdy nieskończony zbiór rekurencyjnie przeliczalny zawiera nieskończony zbiór rekurencyjny.
- Każdy nieskończony zbiór rekurencyjnie przeliczalny daje się przedstawić w postaci $A = rng(f)$, dla pewnej wzajemnie jednoznacznej funkcji ogólnie rekurencyjnej f.

Własności zbiorów i relacji rekurencyjnie przeliczalnych

- Wykres funkcji ogólnie rekurencyjnej jest zbiorem rekurencyjnym.
- Jeśli wykres Γ_f funkcji f jest rekurencyjnie przeliczalny, to funkcja f jest częściowo rekurencyjna.
- Przeciwobraz zbioru rekurencyjnego względem funkcji ogólnie rekurencyjnej jest rekurencyjny.
- Niech A będzie zbiorem rekurencyjnym, f funkcją ogólnie rekurencyjną i przy tym niech $rng(f) = \omega$, $f(A) \cap f(\omega - A) = \emptyset$. Wtedy $f(A)$ jest rekurencyjny.
- Niech A, B będą zbiorami rekurencyjnie przeliczalnymi, zaś C zbiorem rekurencyjnym takim, że $A \cap B = \emptyset$, $A \subseteq C \subseteq A \cup B$. Wtedy A jest rekurencyjny.

Własności zbiorów i relacji rekurencyjnie przeliczalnych

- Niech A, B będą zbiorami rekurencyjnie przeliczalnymi. Wtedy istnieją zbiory rekurencyjnie przeliczalne $A_1 \subseteq A$, $B_1 \subseteq B$ takie, że $A_1 \cap B_1 = \emptyset$, $A_1 \cup B_1 = A \cup B$.
- Można udowodnić, że:
  - (a) funkcja otrzymana za pomocą złożenia z funkcji o wykresie rekurencyjnie przeliczalnym ma wykres rekurencyjnie przeliczalny;
  - (b) funkcja utworzona za pomocą schematu rekursji prostej z funkcji o wykresie rekurencyjnie przeliczalnym ma wykres rekurencyjnie przeliczalny;
  - (c) funkcja utworzona z pomocą μ-operatora z funkcji o wykresie rekurencyjnie przeliczalnym ma wykres rekurencyjnie przeliczalny;
  - (d) wykres dowolnej funkcji częściowo rekurencyjnej jest rekurencyjnie przeliczalny.

Własności zbiorów i relacji rekurencyjnie przeliczalnych

- Funkcja jest częściowo rekurencyjna wtedy i tylko wtedy, gdy jej wykres jest rekurencyjnie przeliczalny.
- Dziedzina funkcji częściowo rekurencyjnej jest zbiorem rekurencyjnie przeliczalnym.
- Zbiór wartości funkcji częściowo rekurencyjnej jest zbiorem rekurencyjnie przeliczalnym.
- Każdy zbiór rekurencyjny jest rekurencyjnie przeliczalny.
- Zbiór n-tek jest rekurencyjnie przeliczalny wtedy i tylko wtedy, gdy jego funkcja charakterystyczna jest częściowo rekurencyjna.

Własności zbiorów i relacji rekurencyjnie przeliczalnych

- Można udowodnić, że:
  - (a) obraz zbioru rekurencyjnie przeliczalnego względem funkcji częściowo rekurencyjnej jest rekurencyjnie przeliczalny;
  - (b) przeciwobraz zbioru rekurencyjnie przeliczalnego względem funkcji częściowo rekurencyjnej jest rekurencyjnie przeliczalny.
- Zbiór A rozwiązań równania

$$f(x_1, \ldots, x_n) = a$$

  jest rekurencyjnie przeliczalny, jeśli f jest częściowo rekurencyjną funkcją n-argumentową.
- Jeśli f jest funkcją częściowo rekurencyjną, to zbiór $M = \{(x_1, \ldots, x_n) : \exists y\ f(x_1, \ldots, x_n, y) = 0\}$ jest rekurencyjnie przeliczalny.

Własności zbiorów i relacji rekurencyjnie przeliczalnych

Niech $M_1, \ldots, M_n$ będą parami rozłącznymi rekurencyjnie przeliczalnymi zbiorami n-tek, a $f_1, \ldots, f_k$ częściowo rekurencyjnymi funkcjami n-argumentowymi. Wtedy funkcja g zdefiniowana następująco:

$$g(x_1, \ldots, x_n) = \begin{cases} f_1(x_1, \ldots, x_n), & \text{gdy} \quad (x_1, \ldots, x_n) \in M_1, \\ \ldots & \ldots \quad \ldots \\ f_k(x_1, \ldots, x_n), & \text{gdy} \quad (x_1, \ldots, x_n) \in M_k, \\ \text{nie określona} & \quad\quad \text{w pozostałych przypadkach,} \end{cases}$$

jest częściowo rekurencyjna.

- Funkcję częściową $f(x_1, \ldots, x_n)$ można przedstawić w postaci $f(x_1, \ldots, x_n) = \mu t[g(x_1, \ldots, x_n, t) = 0]$ dla stosownej funkcji pierwotnie rekurencyjnej $g(x_1, \ldots, x_n, t)$ wtedy i tylko wtedy, gdy wykres funkcji $f(x_1, \ldots, x_n)$ jest pierwotnie rekurencyjny.

Własności zbiorów i relacji rekurencyjnie przeliczalnych

- Niech $F(x,y)$ będzie funkcją zdefiniowaną z pomocą ***schematu rekursji względem dwu zmiennych***:

$$\begin{cases} F(0,y) = \varphi(y), \\ F(x+1,0) = \psi(x, F(x,\alpha(x)), F(x, F(x,\gamma(x)))), \\ F(x+1,y+1) = \tau(x,y,F(x,F(x+1,y))). \end{cases}$$

- Wtedy, jeśli funkcje $\varphi, \psi, \alpha, \gamma, \tau$ są ogólnie rekurencyjne, to funkcja F jest ogólnie rekurencyjna.

- Jeśli dziedzina częściowo rekurencyjnej funkcji f jest zbiorem rekurencyjnym, to f ma rekurencyjne dookreślenie.
- Istnieje funkcja częściowo rekurencyjna $f(x)$, która nie ma ogólnie rekurencyjnego dookreślenia.

Własności zbiorów i relacji rekurencyjnie przeliczalnych

- Istnieje funkcja częściowo rekurencyjna $f(x)$, która nie daje się przedstawić w postaci $f(x) = \mu y[g(x, y) = 0]$ dla żadnej ogólnie rekurencyjnej funkcji g.
- Jeśli $V(n, x)$ jest częściowo rekurencyjną funkcją uniwersalną dla klasy wszystkich jednoargumentowych funkcji częściowo rekurencyjnych, to zbiór $M = \{x : V(x, x) = 0\}$ jest rekurencyjnie przeliczalny, ale nie jest rekurencyjny.
- Jeśli $V(n, x)$ jest częściowo rekurencyjną funkcją uniwersalną dla klasy wszystkich jednoargumentowych funkcji pierwotnie rekurencyjnych, to zbiór

$$\boldsymbol{G} = \{n : V(n, x) \text{ jest ogólnie rekurencyjna}\}$$

nie jest rekurencyjnie przeliczalny.

Indeksy

Pokażemy teraz, że funkcje rekurencyjne można kodować, przypisując im liczby naturalne jako kody. W konsekwencji otrzymujemy szereg ważnych twierdzeń elementarnej teorii rekursji.
Indeksowanie funkcji rekurencyjnych pozwala również na sformułowanie pewnych ważnych twierdzeń metalogicznych (dotyczących niezupełności, nierozstrzygalności, itd.) w terminach dotyczących różnych rodzajów zbiorów (produktywnych, twórczych i innych).

- Podstawą prezentacji w tym punkcie są monografie: Hinman, P. 2005. *Fundamentals of Mathematical Logic*, A K Peters, Ltd., Wellesley oraz Odifreddi, P.G. 1989. *Classical recursion theory.* North-Holland Publishing Company, Amsterdam.
- Jest wiele metod przyporządkowywania funkcjom indeksów, zależnych od używanych funkcji kodujących.

Systemy indukcyjne

Systemem indukcyjnym nazywamy każdy układ postaci $\mathcal{X} = (X, X_0, \mathcal{H})$ taki, że:

- $X \neq \emptyset$
- $X_0 \subseteq X$
- $\mathcal{H}$ jest rodziną skończenie argumentowych funkcji, tj. dla każdego $H \in \mathcal{H}$ istnieje k_H taka, że $H : X^{k_H} \to X$.

Jeśli $\mathcal{X} = (X, X_0, \mathcal{H})$ jest systemem indukcyjnym, to niech:

$$X_{n+1} = X_n \cup \{H(x_0, \ldots, x_{k_H-1}) : \ H \in \mathcal{H} \wedge x_0, \ldots, x_{k_H-1} \in X_n\}$$

$$\overline{\mathcal{X}} = \bigcup_{n \in \omega} X_n.$$

Systemy indukcyjne

Niech $\mathcal{X} = (X, X_0, \mathcal{H})$ będzie systemem indukcyjnym. Zbiór $Y \subseteq X$ nazywamy ***$\mathcal{X}$-domkniętym***, gdy:

- $X_0 \subseteq Y$ oraz dla wszystkich $H \in \mathcal{H}$ oraz $x_0, \ldots, x_{k_H} \in Y$ mamy: $H(x_0, \ldots, x_{k_H}) \in Y$.

Dla dowolnego systemu indukcyjnego $\mathcal{X} = (X, X_0, \mathcal{H})$:

- $\overline{\mathcal{X}}$ jest $\subseteq$-najmniejszym zbiorem $\mathcal{X}$-domkniętym.
- $\mathcal{X} = \bigcap\{Y \subseteq X : Y \text{ jest } \mathcal{X}\text{-domknięty}\}$.

Derywacje

Przez ***$\mathcal{X}$-derywację*** rozumiemy każdy niepusty skończony ciąg $(x_0, \ldots, x_n)$ elementów X taki, że dla dowolnych $i \leqslant n$ zachodzi alternatywa:

- $x_i \in X_0$ lub
- dla pewnej $H \in \mathcal{H}$ oraz pewnych $j_0, \ldots, j_{k_H - 1} < i$ mamy: $x_i = H(j_0, \ldots, j_{k_H - 1})$.

$\mathcal{X}$-derywacja $(x_0, \ldots, x_n)$ jest ***$\mathcal{X}$-derywacją*** elementu x_n.

Definicje przez rekursję

Klasyczne definicje przez rekursję wykorzystują następujące dobrze znane twierdzenie:
Dla dowolnego zbioru Z, dowolnego $z_0 \in Z$ oraz dowolnej funkcji $G : Z \times \omega \to Z$ istnieje dokładnie jedna funkcja $F : \omega \to Z$ taka, że $F(0) = z_0$ oraz dla wszystkich $n \in \omega$ mamy: $F(n+1) = G(F(n), n)$.

Mówimy, że system indukcyjny $\mathcal{X} = (X, X_0, \mathcal{H})$ jest ***jednoznacznie czytelny***, gdy:

- zbiory wartości wszystkich funkcji $H \in \mathcal{H}$ są parami rozłączne oraz rozłączne ze zbiorem X_0
- wszystkie funkcje $H \in \mathcal{H}$ są injekcjami.

Definicje przez rekursję

Twierdzenie. ***O definiowaniu przez rekursję.***

Niech $\mathcal{X} = (X, X_0, \mathcal{H})$ będzie jednoznacznie czytelnym systemem indukcyjnym. Dla dowolnego zbioru Z, dowolnej funkcji $F_0 : X_0 \to Z$ oraz dowolnej rodziny funkcji $(G_H : H \in \mathcal{H})$ takiej, że $G_H : Z^{k_H} \times X^{k_H} \to Z$ dla wszystkich $H \in \mathcal{H}$ istnieje dokładnie jedna funkcja $F : \overline{\mathcal{X}} \to Z$, która rozszerza F_0 i taka, że dla każdej $H \in \mathcal{H}$ oraz wszystkich $x_0, \ldots, x_{k_H - 1} \in \overline{\mathcal{X}}$:

$$F(H(x_0, \ldots, x_{k_H-1})) = G_H(F(x_0), \ldots, F(x_{k_H-1}), x_0, \ldots, x_{k_H-1}).$$

Dla ustalonego systemu indukcyjnego $\mathcal{X}$ zastosowania tego twierdzenia nazywamy definicjami przez ***$\mathcal{X}$-rekursję***.

Kodowanie

Umówmy się, że w tym punkcie stosujemy (pierwotnie rekurencyjne) kodowanie określone przez rekursję prostą:

- $\langle\rangle^0 = 1$
- $\langle m_0, \ldots, m_k\rangle^{k+1} = p_0^{m_0+1} \cdot \ldots \cdot p_k^{m_k+1}$

(potem można opuszczać indeks k i pisać po prostu $\langle m_0, \ldots, m_k\rangle$). „Odkodowanie" daje funkcja: $(s)_i = \mu m < s[p_i^{m+2} \nmid s]$. Tu: p_i jest i-tą liczbą pierwszą, symbol „$|$" oznacza „dzieli bez reszty", a „$\nmid$" dopełnienie tej relacji. W zwykły sposób definiujemy też:

- $lg(s)$ funkcję ***długości*** $lg(s) = \mu s < k\ (p_k \nmid s)$
- $Sq(s)$ zbiór ***numerów*** $Sq(s) \equiv \forall i < s\ (p_i | s \rightarrow i < lg(s))$
- operację $*$ ***konkatenacji***: $s * t = s \times p_{lg(s)}^{(t)_0+1} \times \ldots \times p_{lg(s)+lg(t)-1}^{(t)_{lg(t)-1}+1}$.

Systemy indukcyjne

Rozważmy dowolny system indukcyjny $\mathcal{X} = (\omega, X_0, \mathcal{H})$. Wtedy:

- (a) Jeśli X_0 jest (pierwotnie) rekurencyjny, a $\mathcal{H}$ jest skończonym zbiorem funkcji (pierwotnie) rekurencyjnych, to $\overline{\mathcal{X}}$ jest rekurencyjnie przeliczalny.
- (b) Jeśli w dodatku każda $H \in \mathcal{H}$ jest rosnąca (względem każdego argumentu), to $\overline{\mathcal{X}}$ jest (pierwotnie) rekurencyjny.

- $Der_{\mathcal{X}} = \{s : ((s)_0, \ldots, (s)_{lg(s)-1})$ jest $\mathcal{X}$-derywacją$\}$
- $Derof_{\mathcal{X}} = \{(s, m) : Der_{\mathcal{X}}(s) \wedge (s)_{lg(s)-1} = m\}$
- $m \in \overline{\mathcal{X}} \equiv \exists s\ Derof_{\mathcal{X}}(s, m)$.

Indeksy pierwotnie rekurencyjne

Zbiór *PRI* ***indeksów pierwotnie rekurencyjnych*** jest najmniejszym zbiorem liczb naturalnych takim, że dla wszystkich k:

- dla wszystkich n oraz $i < k$: $\langle 0, k, n\rangle, \langle 1, k, i\rangle, \langle 2, k, i\rangle \in PRI$
- dla wszystkich l, jeśli $b, c_0, \ldots, c_{l-1} \in PRI$, to $\langle 3, k, b, c_0, \ldots, c_{l-1}\rangle \in PRI$
- jeśli $b, c \in PRI$, to $\langle 4, k+1, b, c\rangle \in PRI$.

Z kolei, przyporządkujemy indeksom pierwotnie rekurencyjnym funkcje pierwotnie rekurencyjne. Funkcjami prostymi są: funkcje stałe równe n, rzuty oraz funkcje następnika względem i-tego argumentu. Operacje na funkcjach to: złożenie i schemat rekursji prostej.

Dygresja. Indeksy pierwotnie rekurencyjne można traktować jako kody obliczeń. Obliczenia mają, jak wiadomo, strukturę drzew.

Indeksy pierwotnie rekurencyjne

Funkcja $a \mapsto [a]$ ze zbioru PRI w zbiór wszystkich funkcji pierwotnie rekurencyjnych jest zdefiniowana warunkami:

- $[\langle 0, k, n\rangle]$ jest (k-argumentową) funkcją stałą równą n;
- $[\langle 1, k, i\rangle]$ jest (k-argumentową) funkcją rzutu na i-tą współrzędną;
- $[\langle 2, k, i\rangle]$ jest (k-argumentową) funkcją następnika względem i-tego argumentu;
- $[\langle 3, k, b, c_0, \ldots, c_{l-1}\rangle]$ jest (k-argumentową) funkcją otrzymaną przez złożenie funkcji $[b]$ i funkcji $[c_0]$, …, $[c_{l-1}]$;
- $[\langle 4, k+1, b, c\rangle]$ jest (k-argumentową) funkcją otrzymaną przez rekursję prostą z funkcji $[b]$ oraz $[c]$.

Mówimy, że $a \in PRI$ jest ***indeksem pierwotnie rekurencyjnym*** funkcji pierwotnie rekurencyjnej $[a]$.

Indeksy pierwotnie rekurencyjne

Niech $PRI^k = \{a \in PRI : (a)_1 = k\}$. Wtedy PRI^k jest zbiorem indeksów pierwotnie rekurencyjnych k-argumentowych funkcji pierwotnie rekurencyjnych. Każda funkcja pierwotnie rekurencyjna ma nieskończenie wiele pierwotnie rekurencyjnych indeksów.

Dla każdej k i wszystkich $a, m_0, \ldots, m_{k-1}$ niech:

$$U^k_{PRI}(a, m_0, \ldots, m_{k-1}) = \begin{cases} [a](m_0, \ldots, m_{k-1}), & \text{gdy} \quad a \in PRI^k \\ 0, & \text{gdy} \quad a \notin PRI^k. \end{cases}$$

Wtedy U^k_{PRI} jest ***funkcją uniwersalną*** dla klasy wszystkich funkcji pierwotnie rekurencyjnych k-argumentowych. Jest ona rekurencyjna, ale nie jest pierwotnie rekurencyjna.

Indeksy pierwotnie rekurencyjne

Dla dowolnej 2-argumentowej funkcji J na ω oraz dowolnych liczb k i l, niech $J^{*k,l}$ będzie funkcją k-argumentową:

$$J^{*k,l}(m_0, \ldots, m_{k-1}) = J(\langle m_0, \ldots, m_{k-1} \rangle, k),$$

a jeśli $\mathcal{I}$ jest zbiorem funkcji 2-argumentowych, to niech:

$$\mathcal{I}^* = \{J^{*k,l} : \ J \in \mathcal{I} \wedge k, l \in \omega\}.$$

Powiemy, że J jest ***porządna***, gdy dla wszystkich k, l oraz $m_0, \ldots, m_{k-1}$:

$$k, l < J^{*k,l}(m_0, \ldots, m_{k-1}).$$

Indeksy pierwotnie rekurencyjne

Powiemy, że $\mathcal{X} = (\omega, X_0, \mathcal{H} \cup \mathcal{I})$ jest ***(pierwotnie) rekurencyjnym systemem indukcyjnym***, gdy:

- $\mathcal{X}$ jest systemem indukcyjnym,
- X_0 jest (pierwotnie) rekurencyjny,
- $\mathcal{I}$ jest skończonym zbiorem porządnych 2-argumentowych funkcji (pierwotnie) rekurencyjnych.

Dla dowolnego (pierwotnie) rekurencyjnego systemu indukcyjnego $\mathcal{X} = (\omega, X_0, \mathcal{H} \cup \mathcal{I})$:

- $\overline{\mathcal{X}}$ jest rekurencyjnie przeliczalny.
- Jeśli wszystkie $H \in \mathcal{H}$ oraz $J^{*k,l} \in \mathcal{I}^*$ są rosnące (względem wszystkich argumentów), to $\overline{\mathcal{X}}$ jest (pierwotnie) rekurencyjny.

W konsekwencji, zbiór *PRI* ***jest pierwotnie rekurencyjny.***

Obliczenia pierwotnie rekurencyjne

Definiujemy zbiór PRC (kodów) ***obliczeń pierwotnie rekurencyjnych*** jako najmniejszy zbiór taki, że:

1. Dla wszystkich n oraz $i < k$:
   - $\langle\langle 0, k, n\rangle, m_0, \ldots, m_{k-1}, n\rangle \in PRC$
   - $\langle\langle 1, k, i\rangle, m_0, \ldots, m_{k-1}, m_i\rangle \in PRC$
   - $\langle\langle 2, k, i\rangle, m_0, \ldots, m_{k-1}, m_i + 1\rangle \in PRC$

2. Dla wszystkich $l, p_0, \ldots, p_{l-1}, q$:
   - jeśli dla wszystkich $i < l$: $\langle c_i, m_0, \ldots, m_{k-1}, p_i\rangle \in PRC$ oraz $\langle b, p_0, \ldots, p_{l-1}, q\rangle \in PRC$, to $\langle\langle 3, k, b, c_0, \ldots, c_{l-1}\rangle, m_0, \ldots, m_{k-1}, q\rangle \in PRC$

Obliczenia pierwotnie rekurencyjne

3. Dla wszystkich q:
   - jeśli $\langle b, m_0, \ldots, m_{k-1}, q\rangle \in PRC$, to $\langle\langle 4, k+1, b, c\rangle, m_0, \ldots, m_{k-1}, 0, q\rangle \in PRC$
   - dla wszystkich n, jeśli $\langle\langle 4, k+1, b, c\rangle, m_0, \ldots, m_{k-1}, n, p\rangle \in PRC$ oraz $\langle c, p, m_0, \ldots, m_{k-1}, n, q\rangle \in PRC$, to $\langle\langle 4, k+1, b, c\rangle, m_0, \ldots, m_{k-1}, n+1, q\rangle \in PRC$.

Mamy, dla każdej k i dowolnych $a \in PRI^k$, $m_0, \ldots, m_{k-1}$:
$[a](m_0, \ldots, m_{k-1}) =$ jedyne q takie, że $\langle a, m_0, \ldots, m_{k-1}, q\rangle \in PRC$.
Zbiór** PRC **jest rekurencyjnie przeliczalny.
Uwaga. Dla funkcji 1-argumentowych (czyli gdy $(a)_1 = 1$) mamy:
$[a](m) = q \;\equiv\; \exists s\,(Derof^{\mathcal{X}}(s, \langle a, m, q\rangle))$, gdzie $\overline{\mathcal{X}}$ jest systemem indukcyjnym definiującym PRC.

Częściowe obliczenia rekurencyjne

Zdefiniujemy zbiór COR (kodów) ***częściowych obliczeń rekurencyjnych*** jako najmniejszy zbiór taki, że:

- Zachodzą warunki 1.–3. z definicji PRC (gdzie oczywiście zamieniamy PRC na COR)
- 4. Dla wszystkich b oraz n, jeśli $\langle b, m_0, \ldots, m_{k-1}, n, 0\rangle \in COR$ oraz dla każdego $p < n$ istnieje $q_p > 0$ taka, że $\langle b, m_0, \ldots, m_{k-1}, p, q_p\rangle \in COR$, to $\langle\langle 5, k, b\rangle m_0, \ldots, m_{k-1}, n\rangle \in COR$.

Zbiór** COR **jest rekurencyjnie przeliczalny.

Przypomnijmy, że k-argumentowa funkcja jest ***częściowa***, gdy jej dziedzina jest podzbiorem ω^k.

Funkcje częściowe: kilka oznaczeń

Piszemy $F(x_0, \ldots, x_{k-1}) \downarrow$, gdy $(x_0, \ldots, x_{k-1}) \in dom(F)$.
Jeśli nie zachodzi $F(x_0, \ldots, x_{k-1}) \downarrow$, to piszemy $F(x_0, \ldots, x_{k-1}) \uparrow$.
Klasa wszystkich ***częściowych funkcji rekurencyjnych*** to najmniejsza klasa zawierająca funkcje proste i domknięta na złożenia, rekursję prostą oraz operację ***minimum***, rozumianą następująco:
Jeśli $G(m_0, \ldots, m_{k-1}, n)$ jest częściową funkcją rekurencyjną oraz dla wszystkich $m_0, \ldots, m_{k-1}$ spełnione są warunki:

- $\mu y \, (G(m_0, \ldots, m_{k-1}, y) = 0) \downarrow$
- $\forall z \leqslant y \, (G(m_0, \ldots, m_{k-1}, z)) \downarrow$,

to $F(m_0, \ldots, m_{k-1}) = \mu y \, (G(m_0, \ldots, m_{k-1}, y) = 0)$ jest częściową funkcją rekurencyjną.
Dla funkcji częściowych F i G piszemy:
$F(m_0, \ldots, m_{k-1}) \simeq G(n_0, \ldots, n_{l-1})$, gdy wartości po obu stronach $\simeq$ są określone i równe, lub gdy obie są nieokreślone.

Indeksy (dla funkcji częściowych)

Dla każdej k i wszystkich a takich, że $k = (a)_1$ oraz dowolnych $m_0, \ldots, m_{k-1}$ i n niech:
$\{a\}(m_0, \ldots, m_{k-1}) \simeq n \;\equiv\; \langle a, m_0, \ldots, m_{k-1}, n\rangle \in COR$
$U^k(a, m_0, \ldots, m_{k-1}) \simeq \{a\}(m_0, \ldots, m_{k-1}) \simeq \mu n\,(\langle a, m_0, \ldots, m_{k-1}, n\rangle \in COR)$.
U^k jest (częściową) ***funkcją uniwersalną*** dla k-argumentowych funkcji rekurencyjnych. Dla każdej k, funkcja U^k jest częściową funkcją rekurencyjną.

Następujące warunki są równoważne (dla dowolnej funkcji częściowej F):

- F jest częściowo rekurencyjna.
- $F = \{a\}$ dla pewnego $a \in \omega$.

Indeksy (dla funkcji częściowych)

Mamy zatem wyliczenie $(\{a\} : \ a \in \omega)$ wszystkich funkcji częściowo rekurencyjnych, w tym sensie, że:
$\{a\}(x_0, \ldots, x_{k-1}) \simeq y \ \equiv \ \langle a, x_0, \ldots, x_{k-1}, n\rangle \in COR$.
Liczbę a nazywamy ***indeksem*** funkcji $\{a\}$.

Jeśli nie zachodzi $\{a\}(x_0, \ldots, x_{k-1}) \downarrow$, to piszemy $\{a\}(x_0, \ldots, x_{k-1}) \uparrow$.
Uwaga. Formalnie, każda liczba a jest indeksem, ale może oczywiście być indeksem funkcji pustej. Jeśli $\{a\} \neq \emptyset$, to a musi być postaci $\langle i, k, \ldots\rangle$; wtedy $\{a\}$ jest funkcją k-arg., a więc: jeśli $\{a\}(m_0, \ldots, m_{k-1}) \downarrow$, to $(a)_1 = k$.

Zamiast $\{a\}$ używa się często oznaczeń: φ_a lub ϕ_a (co bywa wygodne ze względów typograficznych).

Indeksy (dla funkcji częściowych)

Dla $a \in \omega$ niech $W_a = \{x : \{a\}(x) \downarrow\}$.
Wtedy: zbiór A jest rekurencyjnie przeliczalny wtedy i tylko wtedy, gdy $A = W_a$ dla pewnego $a \in \omega$ (czyli: A jest r.e. wtedy i tylko wtedy, gdy A jest dziedziną częściowej funkcji rekurencyjnej). Powyższe może więc służyć za ***definicję*** zbiorów rekurencyjnie przeliczalnych.

$\{W_a : a \in \omega\}$ jest zatem wyliczeniem wszystkich zbiorów rekurencyjnie przeliczalnych. Nazywa się je ***wyliczeniem standardowym***.

- $\{\{a\} : a \in \omega\}$ — standardowe wyliczenie wszystkich częściowych funkcji rekurencyjnych (w innych oznaczeniach: $\{\varphi_a : a \in \omega\}$)
- $\{W_a : a \in \omega\}$ — standardowe wyliczenie wszystkich zbiorów rekurencyjnie przeliczalnych.

Twierdzenie o postaci normalnej

Twierdzenie Kleene'go o Postaci Normalnej.

Istnieją:

- *funkcja pierwotnie rekurencyjna $\mathcal{U}$ oraz*
- *dla każdej $n \geqslant 1$ relacja pierwotnie rekurencyjna $\mathcal{T}_n$*

takie, że dla każdej n-argumentowej funkcji pierwotnie rekurencyjnej f istnieje liczba e taka, że:

- $\forall x_1 \ldots \forall x_n \exists y\ \mathcal{T}_n(e, x_1, \ldots, x_n, y)$
- $f(x_1, \ldots, x_n) = \mathcal{U}(\mu y\ \mathcal{T}_n(e, x_1, \ldots, x_n, y))$

Zarys dowodu. Dla każdej funkcji pierwotnie rekurencyjnej f istnieje liczba e taka, że $f = [e]$ oraz $\forall x_1 \ldots \forall x_n \exists!\ y\ \langle e, x_1, \ldots, x_n, y\rangle \in PRC$ (przy tym $f(x_1, \ldots, x_n) = y$), co wyznacza (pierwotnie rekurencyjną) relację $\mathcal{T}_n$. Minimum jest efektywne. Funkcja (rzutu) $\mathcal{U}$ daje wartość y z $\mu y\ \mathcal{T}_n(e, x_1, \ldots, x_n, y)$.

Twierdzenie o enumeracji

Twierdzenie o enumeracji.
Istnieje funkcja rekurencyjna dwuargumentowa ψ taka, że:

- *dla wszystkich liczb e: $\psi(e, x)$ jest jednoargumentową funkcją pierwotnie rekurencyjną,*
- *każda jednoargumentowa funkcja pierwotnie rekurencyjna jest postaci $\psi(e, x)$ dla pewnej e.*

Zarys dowodu. Niech $\psi(e, x) = \mathcal{U}(\mu y\, \mathcal{T}_1(e, x, y))$, gdy $e \in PRI$, a $\psi(e, x) = 0$ w przeciwnym przypadku.

Mamy także wersję twierdzenia o enumeracji dla funkcji n-argumentowych. Twierdzenia o postaci normalnej i enumeracji mają też stosowne wersje dla funkcji częściowych, które sformułujemy bez zarysu dowodu:

Postać normalna i enumeracja: funkcje częściowe

Postać normalna. *Istnieje funkcja pierwotnie rekurencyjna $\mathcal{U}$ oraz, dla każdej $n \geqslant 1$ relacja pierwotnie rekurencyjna $\mathcal{T}_n$ takie, że dla każdej n-argumentowej funkcji częściowej f istnieje liczba e taka, że:*

- $f(x_1, \ldots, x_n) \downarrow$ *dokładnie wtedy, gdy* $\exists y\ \mathcal{T}_n(e, x_1, \ldots, x_n, y)$
- $f(x_1, \ldots, x_n) \simeq \mathcal{U}(\mu y\ \mathcal{T}_n(e, x_1, \ldots, x_n, y))$.

Enumeracja. *Dla każdej n piszmy: $\{e\}^n \simeq \mathcal{U}(\mu y\ \mathcal{T}_n(e, x_1, \ldots, x_n, y))$. Wtedy ciąg $(\{e\}^n)_{e\in\omega}$ jest częściowo rekurencyjną enumeracją wszystkich n-argumentowych funkcji częściowo rekurencyjnych, czyli:*

- *dla każdej e, $\{e\}^n$ jest n-argumentową funkcją częściowo rekurencyjną,*
- *jeśli f jest n-argumentową funkcją częściowo rekurencyjną, to istnieje e taka, że $f \simeq \{e\}^n$*
- *istnieje $(n+1)$-argumentowa częściowa funkcja rekurencyjna ψ taka, że $\psi(e, x_1, \ldots, x_n) \simeq \{e\}^n(x_1, \ldots, x_n)$.*

Twierdzenie o parametryzacji

Twierdzenie o parametryzacji (S^m_n-twierdzenie).
Dla dowolnych $m, n \in N$ istnieje pierwotnie rekurencyjna funkcja S^m_n taka, że dla wszystkich e, $x_0, \ldots, x_{m-1}$ oraz $y_0, \ldots, y_{n-1}$:

$$\{S^m_n(e, x_0, \ldots, x_{m-1})\}(y_0, \ldots, y_{n-1}) \simeq \{e\}(x_0, \ldots, x_{m-1}, y_0, \ldots, y_{n-1}).$$

W szczególności, dla $m = 1$, $n = 1$ mamy: $\{S^1_1(e, x)\}(y) \simeq \{e\}(x, y)$.

Zarys dowodu. Korzystamy ze schematów rekursji prostej:

- $S^1_n(e, x) = \langle 3, n, e, \langle 0, n, x\rangle, \langle 1, n, 0\rangle, \ldots, \langle 1, n, n-1\rangle\rangle$
- $S^{m+1}_n(e, x_0, \ldots, x_m) = S^1_n(S^m_{n+1}(e, x_0, \ldots, x_{m-1}), x_m)$.

O diagonalizacji

Dla całkiem dowolnego zbioru S oraz funkcji $d : S \rightarrow S$, która nie jest identycznością (czyli $d(a) \neq a$ dla $a \in S$) rozważmy (być może) nieskończoną macierz elementów S:

$$\begin{array}{cccc} a_{0,0} & a_{0,1} & a_{0,2} & \ldots \\ a_{1,0} & a_{1,1} & a_{1,2} & \ldots \\ a_{2,0} & a_{2,1} & a_{2,2} & \ldots \\ \ldots & \ldots & \ldots & \ldots \end{array}$$

- ***Ciągiem diagonalnym*** (dla (S, d)) nazwiemy ciąg: $d(a_{0,0}), d(a_{1,1}), d(a_{2,2}), \ldots$.
- Wtedy ciąg diagonalny jest różny od każdego wiersza powyższej macierzy (ponieważ różni się od n-tego wiersza n-tym elementem, na mocy definicji d).

O diagonalizacji

- Szczególnym przypadkiem zastosowania tej konstrukcji jest np. dowód Cantora nieprzeliczalności zbioru wszystkich podzbiorów zbioru ω.

Mamy też rekurencyjną wersję twierdzenia Cantora:

- *Nie istnieje funkcja rekurencyjna, która oblicza (co najmniej jeden) indeks dla każdej (jednoargumentowej) funkcji rekurencyjnej o wartościach w zbiorze $\{0, 1\}$.*

Zarys dowodu (nie wprost). Niech f będzie funkcją rekurencyjną taką, że $\{f(x)\}$ jest całkowita dla każdej x. Zdefiniujmy: $g(x) \simeq 1 - \{f(x)\}(x)$. Wtedy g jest funkcją częściowo rekurencyjną (na mocy twierdzenia o enumeracji) o wartościach w zbiorze $\{0, 1\}$ i całkowitą (bo f całkowita). Nadto, g różni się (na argumencie x) od każdej funkcji $\{f(x)\}$, czyli żaden indeks funkcji g nie należy do $rng(f)$.

O diagonalizacji

Twierdzenie (Post, Gödel, Kleene)

Istnieje zbiór rekurencyjnie przeliczalny, który nie jest rekurencyjny.

Zarys dowodu.

- Niech $\mathbb{K} = \{x : x \in W_x\}$, czyli $x \in \mathbb{K}$ dokładnie wtedy, gdy $\{x\}(x) \downarrow$.
- Na mocy twierdzenia o enumeracji istnieje funkcja częściowo rekurencyjna f taka, że: $f(x) \simeq \{x\}(x)$.
- Stąd $\mathbb{K}$ jest rekurencyjnie przeliczalny, gdyż: $x \in \mathbb{K}$ dokładnie wtedy, gdy $\{x\}(x) \downarrow$.
- Gdyby $\mathbb{K}$ był rekurencyjny, to $\omega - \mathbb{K}$ byłby rekurencyjnie przeliczalny (z twierdzenia Posta). Ale $x \in (\omega - \mathbb{K})$ dokładnie wtedy, gdy $x \notin W_x$, czyli $\omega - \mathbb{K}$ jest różny od wszystkich zbiorów rekurencyjnie przeliczalnych.

O diagonalizacji

Problem stopu.
Zbiór $\mathbb{K}_0$ zdefiniowany warunkiem: $\langle x, e\rangle \in \mathbb{K}_0$ jest rekurencyjnie przeliczalny, lecz nie jest rekurencyjny.

Zarys dowodu. Zauważmy, że $\langle x, e\rangle \in \mathbb{K}_0$ dokładnie wtedy, gdy $\{e\}(x)\downarrow$.

- Rekurencyjnej przeliczalności $\mathbb{K}_0$ dowodzimy tak samo, jak dla $\mathbb{K}$.
- Gdyby $\mathbb{K}_0$ był rekurencyjny, to również $\mathbb{K}$ byłby rekurencyjny, ponieważ: $x \in \mathbb{K}$ dokładnie wtedy, gdy $\langle x, x\rangle \in \mathbb{K}_0$. A zatem $\mathbb{K}_0$ nie jest rekurencyjny.

O diagonalizacji

Mówimy, że A jest ***zbiorem indeksów*** zbioru $\mathcal{A}$ częściowych funkcji rekurencyjnych, gdy $A = \{x : \varphi_x \in \mathcal{A}\}$.
Uwaga. Tu warto pisać φ_x zamiast $\{x\}$ dla uniknięcia nieporozumień.

Nazwiemy zbiór $\mathcal{A}$ funkcji częściowo rekurencyjnych ***zupełnie rekurencyjnym***, gdy jego zbiór indeksów jest rekurencyjny.

Twierdzenie Rice'a.
Zbiór $\mathcal{A}$ funkcji częściowo rekurencyjnych jest zupełnie rekurencyjny dokładnie wtedy, gdy albo jest pusty, albo zawiera wszystkie funkcje częściowo rekurencyjne.

O diagonalizacji

Zarys dowodu. Jeśli $\mathcal{A} = \emptyset$, to zbiorem indeksów zbioru $\mathcal{A}$ jest $\emptyset$, a jeśli $\mathcal{A}$ zawiera wszystkie funkcje częściowo rekurencyjne, to jego zbiorem indeksów jest ω. Przypuśćmy teraz, że $\mathcal{A}$ jest różny od $\emptyset$ i różny od zbioru wszystkich funkcji częściowo rekurencyjnych. Wtedy istnieją a i b takie, że $\varphi_a \in \mathcal{A}$ oraz $\varphi_b \notin \mathcal{A}$.
Jeśli funkcja nigdzie nie określona nie należy do $\mathcal{A}$, to niech f będzie funkcją rekurencyjną taką, że:

- $\varphi_{f(x)} = \varphi_a$, gdy $x \in \mathbb{K}$
- $\varphi_{f(x)}$ nie jest określona w przeciwnym przypadku.

Wtedy: $x \in \mathbb{K}$ dokładnie wtedy, gdy $\varphi_{f(x)} \in \mathcal{A}$, czyli dokładnie wtedy, gdy $f(x) \in A$, gdzie A jest zbiorem indeksów zbioru $\mathcal{A}$. Zbiór A nie jest zatem rekurencyjny, gdyż $\mathbb{K}$ nie jest rekurencyjny.
Jeśli funkcja nigdzie nie określona należy do $\mathcal{A}$, to używamy φ_b, pokazując, iż $\omega - A$ nie jest rekurencyjny.

Twierdzenie o punkcie stałym

Twierdzenie o punkcie stałym (twierdzenie o rekursji).
Dla dowolnej funkcji rekurencyjnej jednoargumentowej f istnieje e taka, że $\varphi_e \simeq \varphi_{f(e)}$.

Zarys dowodu. Skorzystamy z procedury opisanej w punkcie dotyczącym diagonalizacji. Niech f będzie dowolną funkcją rekurencyjną. Niech S będzie zbiorem wszystkich funkcji rekurencyjnych i zdefiniujmy: $d(\varphi_a) = \varphi_{f(a)}$. Szukamy punktu stałego funkcji d, czyli e takiej, iż: $d(\varphi_e) = \varphi_e$. Ponieważ $f \simeq \varphi_a$ dla pewnej a, więc wartości funkcji d są postaci $\varphi_{\varphi_a(b)}$. Rozważmy macierz $(a_{ij})_{i,j\in\omega}$, gdzie $a_{ij} = \varphi_{\varphi_i(j)}$. Jeśli $\varphi_{\varphi_i(j)} \uparrow$, to a_{ij} jest funkcją nigdzie nie określoną. Ciągiem diagonalnym dla (S, d) jest: $\varphi_{f(\varphi_0(0))}, \varphi_{f(\varphi_1(1))}, \varphi_{f(\varphi_2(2))}, \ldots$. Jest to rekurencyjny ciąg funkcji częściowo rekurencyjnych, a więc jest jednym z wierszy macierzy $(a_{ij})_{i,j\in\omega}$. Tak więc, istnieje punkt stały funkcji d.

Twierdzenie o punkcie stałym

Punkt stały, o którym mowa w powyższym twierdzeniu, można wyznaczyć:

- Zauważmy, że punkt stały funkcji d musi być tym elementem ciągu diagonalnego, który leży na przekątnej macierzy.
- Niech więc g będzie (istniejącą na mocy twierdzenia o enumeracji oraz S^m_n-twierdzenia) funkcją taką, że:

$$\varphi_{g(a)} = \varphi_{f(\varphi_a(a))}.$$

- Jeśli b jest dowolnym indeksem funkcji g, to $g(b) = \varphi_b(b)$ jest punktem stałym dla f, ponieważ:

$$\varphi_{g(b)} \simeq \varphi_{f(\varphi_b(b))} \simeq \varphi_{f(g(b))}.$$

W teorii rekursji dowodzi się wielu dalszych twierdzeń o punktach stałych.

Czy można całkiem dowolnie indeksować?

Systemem indeksów nazywamy dowolną rodzinę odwzorowań $\Psi = \{\psi^n : n \in \omega\}$ zbioru ω na zbiór wszystkich n-argumentowych funkcji częściowo rekurencyjnych. Przez ψ^n_e oznaczać będziemy n-argumentową funkcję częściowo rekurencyjną o indeksie e. Mówimy, że Ψ spełnia warunek:

- *enumeracji*, gdy dla każdej n istnieje a taka, że $\psi^{n+1}_a(e, x_1, \ldots, x_n) \simeq \psi^n_e(x_1, \ldots, x_n)$,
- *parametryzacji*, gdy dla wszystkich m, n istnieje całkowita funkcja rekurencyjna s taka, że $\psi_{s(e,x_1,\ldots,x_n)}(y_1, \ldots, y_m) \simeq \psi^{m+n}_e(x_1, \ldots, x_n, y_1, \ldots, y_m)$.

Czy można całkiem dowolnie indeksować?

System indeksów Ψ nazywamy ***akceptowalnym***, gdy dla wszystkich n istnieją całkowite funkcje rekurencyjne f oraz g takie, że:

- $\psi_e^n \simeq \varphi_{f(e)}^n$
- $\varphi_e^n \simeq \psi_{g(e)}^n$

(przypominamy, że $\{\varphi_e : e \in \omega\}$ to standardowe wyliczenie wszystkich funkcji częściowo rekurencyjnych i niech $\{\varphi_e^n : e \in \omega\}$ będzie standardowym wyliczeniem wszystkich n-argumentowych funkcji częściowo rekurencyjnych).

System indeksów Ψ jest akceptowalny wtedy i tylko wtedy, gdy spełnia warunki enumeracji oraz parametryzacji.

Sprowadzalność

Dla dowolnych $A, B \subseteq \omega$ mówimy, że A jest ***m-redukowalny*** do B (piszemy wtedy $A \leqslant_m B$) wtedy i tylko wtedy, gdy istnieje funkcja rekurencyjna f taka, że dla wszystkich $n \in \omega$: $n \in A \equiv f(n) \in B$ (co oznacza, że $A = f^{-1}[B]$).
Jeśli $A \leqslant_m B$, to A jest „co najwyżej tak samo złożony" jak B, jeśli chodzi o procedury ustalania, co jest elementem danego zbioru.
Zauważmy, że jeśli B jest rekurencyjny i $A \leqslant_m B$, to A jest rekurencyjny.

Relacja $\leqslant_m$ jest częściowym porządkiem. W standardowy sposób definiujemy:

- $A <_m B \equiv A \leqslant_m B \wedge \neg B \leqslant_m A$
- $A \equiv_m B \equiv A \leqslant_m B \wedge B \leqslant_m A$ (A i B są ***m-równoważne***)

Stopnie

- $dg_m(A) = \{B : \ A \equiv_m B\}$ (m-***stopień*** zbioru A)
- $dg_m(A) \leqslant_m dg_m(B) \ \equiv \ A \leqslant_m B$
- $dg_m(A) <_m dg_m(B) \ \equiv \ A <_m B$
- $Dg_m = \{dg_m(A) : \ A \subseteq \omega\}$ (zbiór wszystkich m-stopni)
- $dg_m(\emptyset) = \{\emptyset\}$ oraz $dg_m(\omega) = \{\omega\}$ są minimalnymi m-stopniami.
- Niech $\mathbf{0} = \{A : \ \emptyset \neq A \neq \omega$ i A rekurencyjny$\}$. Wtedy $\mathbf{0} \in Dg_m$ oraz:
  - $dg_m(\emptyset) <_m \mathbf{0}$
  - $dg_m(\omega) <_m \mathbf{0}$
  - $\mathbf{0} <_m dg_m(A)$ dla każdego nierekurencyjnego zbioru A
- Dla każdej pary m-stopni istnieje ich l.u.b. Nie jest prawdą, że dla pary dowolnych m-stopni istnieje ich g.l.b.
- m-stopnie różne od $dg_m(\emptyset)$ oraz $dg_m(\omega)$ nie są uporządkowane liniowo.
- Nie istnieje maksymalny m-stopień.

Zbiory m-zupełne

Dla dowolnego zbioru C mówimy, że:

- C jest ***m-zupełny***, gdy C jest r.e. i $A \leqslant_m C$ dla każdego r.e. zbioru A.
- $dg_m(C)$ jest ***r.e. m-stopniem***, gdy istnieje r.e. zbiór A taki, że $A \equiv_m C$.

Oto przykłady ważnych zbiorów m-zupełnych:

- $\mathbb{K} = \{a : a \in W_a\}$.
- $\mathbb{K}_0 = \{\langle x, y\rangle : x \in W_y\}$.
- $\mathbb{K}_1 = \{x : W_x \neq \emptyset\}$.

Żaden z powyższych zbiorów nie jest rekurencyjny. Zbiór $\mathbb{K}$ nazywany jest czasem ***zbiorem przekątniowym***. Jego dopełnienie oznaczmy przez $\overline{\mathbb{K}}$.

Zbiory *m*-zupełne

Niech $\mathbf{1} = \{C : \ C \text{ jest m-zupełny}\}$.

Wtedy $\mathbf{1}$ jest jedynym, największym r.e. *m*-stopniem.
Ponadto, $\mathbf{0} < \mathbf{1}$.

Nie każdy r.e. zbiór nierekurencyjny jest zupełny.

Istnieją r.e. zbiory A takie, że: $\mathbf{0} < dg_m(A) < \mathbf{1}$, jak za chwilę zobaczymy.

Zbiory produktywne

Mówimy, że zbiór P jest ***produktywny***, jeśli istnieje funkcja częściowo rekurencyjna f taka, że dla każdego indeksu a: jeśli $W_a \subseteq P$, to $f(a)$ jest określona oraz $f(a) \in P - W_a$.
Każdy zbiór produktywny P jest zatem, w pewnym sensie, „efektywnie nie r.e.": jeśli $W_a \subseteq P$, to $f(a)$ dostarcza przykładu, że $P \neq W_a$.

- Zbiór $\overline{\mathbb{K}}$ jest produktywny.
- Dla dowolnych P oraz Q, jeśli P jest produktywny i $P \leqslant_m Q$, to Q jest produktywny.
- Dla dowolnego A, jeśli A jest m-zupełny, to $\omega - A$ jest produktywny.
- Każdy zbiór produktywny zawiera nieskończony r.e. podzbiór. Każdy zbiór produktywny zawiera nieskończony podzbiór rekurencyjny.
- Jeśli A jest produktywny, to A nie jest r.e.
- Istnieje $2^{\aleph_0}$ zbiorów produktywnych.

Zbiory twórcze. Zbiory proste.

Mówimy, że zbiór A jest:

- ***prosty***, jeśli A jest r.e., a jego dopełnienie $\omega - A$ jest nieskończone i nie zawiera żadnego nieskończonego r.e. podzbioru.
- ***twórczy***, jeśli A jest r.e., a $\omega - A$ jest produktywny.

Jeśli A jest zbiorem prostym, to A jest zatem zbiorem r.e. o nieskończonym dopełnieniu oraz A ma niepusty przekrój ze wszystkimi zbiorami r.e. W_a. Jeśli A jest zbiorem prostym, to:

- A nie jest rekurencyjny.
- A nie jest twórczy.
- A nie jest m-zupełny ($\mathbb{K} \nleqslant_m A$).

Dowolny zbiór prosty jest nierekurencyjnym r.e. zbiorem, który nie jest m-zupełny.

Zbiory twórcze. Zbiory proste.

Zbiór $\mathbb{K}$ jest twórczy. Każdy zbiór twórczy A jest „efektywnie nierekurencyjny", w tym sensie, że istnieje funkcja częściowo rekurencyjna f taka, że dla każdego r.e. „kandydata" W_a do bycia zbiorem $\omega - A$ wartość $f(a)$ należy do $(\omega - A) - W_a$.
Zbiór A jest twórczy wtedy i tylko wtedy, gdy A jest m-zupełny.
Niech A będzie r.e. zbiorem różnym od ω. Wtedy A jest twórczy wtedy i tylko wtedy, gdy dla każdego r.e. zbioru B, jeśli $A \cap B = \emptyset$, to $A \equiv_m A \cup B$.

Istnieje zbiór prosty, a więc istnieje zbiór r.e., który nie jest ani rekurencyjny, ani m-zupełny.
Istnienia zbioru prostego (zbiorów prostych) dowieść można na różne sposoby. Podajemy jeden z nich (inny sposób istotnie wykorzystuje tzw. ***metodę priorytetu***, której w tym wykładzie nie omawiamy.)

Dowód istnienia zbioru prostego

Niech P będzie relacją zdefiniowaną warunkiem:
$P(a,x) \equiv x \in W_a \wedge x > 2a$. Wtedy P jest semi-rekurencyjna. Istnieje zatem selektor, tj. funkcja częściowo rekurencyjna F o dziedzinie $\{a : \exists x\, P(a,x)\}$ taka, że dla każdego a z dziedziny F zachodzi $P(a, F(a))$.
Niech A będzie dziedziną F. Pokażemy, że A jest zbiorem prostym.
Ponieważ $A = rng(F)$, więc A jest r.e. Ponieważ $F(a) > 2a$ dla $a \in dom(F)$, co najwyżej a elementów zbioru A jest elementami każdego odcinka początkowego $\{0, 1, \ldots, 2a\}$. Tak więc, odcinek taki zawiera co najmniej $a + 1$ elementów zbioru $\omega - A$. W konsekwencji, zbiór $\omega - A$ jest nieskończony.
Przypuśćmy, że W_a jest nieskończony. Wtedy istnieją elementy $x \in W_a$ takie, że $x > 2a$, a więc $F(a)$ jest określona, a stąd $F(a)$ jest elementem $A \cap W_a$. Ostatecznie, $W_a \nsubseteq (\omega - A)$. Pokazaliśmy, że A jest zbiorem prostym.

Porządek stopni

Istnieje r.e. m-stopień $\mathbf{d}$ taki, że: $\mathbf{0} < \mathbf{d} < \mathbf{1}$. Nadto:

- (T1) Istnieją r.e. zbiory A oraz B takie, że A jest prosty oraz $B \nleqslant_m A$.
- (T2) Istnieją r.e. zbiory A i B takie, że $A \nleqslant_m B$ oraz $B \nleqslant_m A$.
- (T3) Istnieje m-niezależna r.e. rodzina (zobacz niżej).
- (T4) Każdy rekurencyjny porządek częściowy zbioru ω można włożyć w uporządkowanie m-stopni, tj. dla dowolnego rekurencyjnego porządku częściowego $\preceq$ zbioru ω istnieje rodzina $(C_i : \ i \in \omega)$ taka, że dla wszystkich i oraz j: $i \preceq j \ \equiv \ C_i \leqslant_m C_j$.
- (T5) Każdy przeliczalny porządek częściowy można włożyć w uporządkowanie m-stopni, tj. dla dowolnego przeliczalnego porządku częściowego $\preceq$ zbioru ω istnieje rodzina $(C_i : \ i \in \omega)$ taka, że dla wszystkich i oraz j: $i \preceq j \ \equiv \ C_i \leqslant_m C_j$.

Porządek stopni

W dowodzie (T5) korzysta się z faktu, że istnieje rekurencyjny porządek częściowy zbioru ω, w który można włożyć każdy przeliczalny porządek częściowy.

Dla dowolnego $I \subseteq \omega$ i dowolnej rodziny zbiorów $(A^n: \ n \in \omega)$:

- $\bigoplus(A^n: \ n \in \omega) = \{\langle n, x\rangle : \ x \in A^n \wedge n \in I\}$
- $(A^n: \ n \in \omega)$ jest *m-**niezależna*** wtedy i tylko wtedy, gdy dla wszystkich $n \in \omega$: $A^n \nleqslant_m \bigoplus(A^n: \ n \in \omega)$.
- $(A^n: \ n \in \omega)$ jest ***r.e.-rodziną*** wtedy i tylko wtedy, gdy $\bigoplus(A^n: \ n \in \omega)$ jest zbiorem r.e.

Struktura rodziny wszystkich stopni jest niezwykle skomplikowana. Tu podaliśmy jedynie kilka dość prostych faktów.

Maszyny Turinga

Maszyna Turinga T jest określona przez:

- (a) ***alfabet zewnętrzny*** $\mathcal{A} = \{a_0, a_1, \ldots, a_n\}$ (gdzie $a_0 = 0,\ a_1 = 1$);
- (b) ***alfabet stanów wewnętrznych*** $Q = \{q_0, q_1, \ldots, q_m\}$;
- (c) ***program***, tj. zbiór wyrażeń $T(i,j)$ ($i = 1, \ldots, m$; $j = 0, \ldots, n$), z których każde może mieć jedną z następujących postaci:
  - $q_i a_j \to q_k a_l$,
  - $q_i a_j \to q_k a_l R$
  - $q_i a_j \to q_k a_l L$

  gdzie $0 \leqslant k \leqslant m,\ 0 \leqslant l \leqslant n$.

Stan q_0 jest wyróżniony — jest stanem ***końcowym***. Stan q_1 też jest wyróżniony — jest stanem ***początkowym***. Symbole a_0 i a_1 także są wyróżnione — a_0 jest symbolem tzw. ***pustej*** klatki taśmy. Symbole a_0 i a_1 wystarczają do zapisania ***każdej*** informacji. Wyrażenia $T(i,j)$ nazywają się ***rozkazami***.

Maszyny Turinga

Słowem maszynowym lub ***konfiguracją*** nazywamy słowo postaci:

$$Aq_k a_l B$$

gdzie $0 \leqslant k \leqslant m$, $0 \leqslant l \leqslant n$, A oraz B — są słowami (być może pustymi) w alfabecie $\mathcal{A}$.

Piszemy a_i^x jako skrót wyrażenia $\underbrace{a_i a_i \ldots a_i}_{x\ razy}$.

Proszę zauważyć, że maszyna Turinga jest pewnym ***obiektem matematycznym***. Wszelkie określenia w rodzaju: „praca maszyny", „maszyna zatrzymuje się", itd. także mają ściśle zdefiniowany sens ***matematyczny***.

Maszyny Turinga

Niech dane będą maszyna T oraz słowo maszynowe $M = Aq_ia_jB$, gdzie $0 \leqslant i \leqslant m$. Przez M'_T oznaczymy słowo otrzymane z M według następujących reguł:

- (1) dla $i = 0$ niech $M'_T = M$;
- (2) dla $i > 0$:
  - (a) jeśli $T(i, j)$ jest postaci $q_ia_j \to q_ka_l$, to $M'_T = Aq_ka_lB$;
  - (b) jeśli $T(i, j)$ jest postaci $q_ia_j \to q_ka_lR$, to:
    - (B_1) jeśli B nie jest słowem pustym, to $M'_T = Aa_lq_kB$,
    - (B_2) jeśli B jest słowem pustym, to $M'_T = Aa_lq_ka_0$;
  - (c) jeśli $T(i, j)$ jest postaci $q_ia_j \to q_ka_lL$, to:
    - (C_1) jeśli $A = A_1a_s$ dla pewnych A_1 oraz a_s, to $M'_T = A_1q_ka_sa_lB$,
    - (C_2) jeśli A jest słowem pustym, to $M'_T = q_ka_0a_lB$.

Maszyny Turinga

Przyjmijmy $M_T^{(0)} = M$, $M_T^{(n+1)} = (M_T^{(n)})'$.

- Mówimy, że maszyna T ***przetwarza*** słowo maszynowe M w słowo M_1, jeżeli dla pewnego n: $M_T^{(n)} = M_1$. Piszemy wtedy $M \underset{T}{\Rrightarrow} M_1$.
- Piszemy $M \underset{T}{\Rrightarrow} M_1$, jeśli maszyna T przetwarza M w M_1 i nie jest przy tym wykorzystywany warunek (C_2) powyższej definicji.
- Piszemy natomiast $M \underset{T}{\mapsto} M_1$, jeśli maszyna T przetwarza M w M_1, a przy tym nie są wykorzystywane warunki (B_1) oraz (C_2) powyższej definicji.

Maszyny Turinga

Mówimy, że maszyna T ***oblicza*** n-argumentową częściową funkcję liczbową f, gdzie $dom(f) \subseteq \omega^n$, $rng(f) \subseteq \omega$, jeśli spełnione są następujące warunki:

- (a) jeśli $(x_1, x_2, \ldots, x_n) \in dom(f)$, to maszyna T ***zatrzymuje się***, tj. przetwarza słowo $q_1 01^{x_1} 01^{x_2} 0 \ldots 01^{x_n} 0$ w pewne słowo Aq_0B, a przy tym słowo Aq_0B zawiera $f(x_1, x_2, \ldots, x_n)$ wystąpień symbolu 1;

- (b) jeśli $(x_1, x_2, \ldots, x_n) \notin dom(f)$, to maszyna rozpoczynając działanie od słowa $M = q_1 01^{x_1} 01^{x_2} 0 \ldots 01^{x_n} 0$ ***pracuje w nieskończoność***, tj. q_0 nie występuje w $M_T^{(n)}$ dla żadnego n.

Maszyny Turinga

Mówimy, że maszyna T ***prawidłowo oblicza*** funkcję f, jeśli spełnione są warunki:

- (a) jeśli $(x_1, x_2, \ldots, x_n) \in dom(f)$, to

$$q_1 0 1^{x_1} 0 1^{x_2} 0 \ldots 0 1^{x_n} 0 \underset{T}{\Rrightarrow} q_0 0 1^{f(x_1, x_2, \ldots, x_n)} 0 0 \ldots 0;$$

- (b) jeśli $(x_1, x_2, \ldots, x_n) \notin dom(f)$, to maszyna, rozpoczynając działanie od słowa $q_1 0 1^{x_1} 0 1^{x_2} 0 \ldots 0 1^{x_n} 0$ pracuje w nieskończoność.

Funkcję f nazywamy ***obliczalną*** [w sensie Turinga] (***prawidłowo obliczalną***), jeśli istnieje maszyna Turinga, która oblicza (prawidłowo oblicza) funkcję f.

Dwa przykłady operacji na maszynach Turinga

Niech T_1, T_2, T_3 będą maszynami Turinga z tym samym alfabetem zewnętrznym $\mathcal{A} = \{a_0, a_1, \ldots, a_n\}$, z alfabetami stanów wewnętrznych:

$$Q_1 = \{q_0, q_1, \ldots, q_r\}, \quad Q_2 = \{q_0, q_1, \ldots, q_s\}, \quad Q_3 = \{q_0, q_1, \ldots, q_t\}$$

oraz programami Π_1, Π_2, Π_3, odpowiednio.

Złożeniem $T_1 \cdot T_2$ maszyn T_1 i T_2 nazywamy maszynę T, której program jest sumą zbiorów:

$$S^{q_0}_{q_{r+1}}\Pi_1 \cup S^{q_1 \ldots q_s}_{q_{r+1} \ldots q_{r+s}}\Pi_2,$$

gdzie $S^{q_j}_{q_i}\Pi$ oznacza zbiór rozkazów otrzymanych z Π poprzez zamianę wszystkich wystąpień q_j na q_i.

Rozwidleniem maszyn T_1, T_2, T_3 względem (q_i, q_j), (symbolicznie $T_1 \left\{ \begin{array}{l} q_i = T_2 \\ q_j = T_3 \end{array} \right.$), gdzie $q_i, q_j \in Q_1$,

nazywamy maszynę T, której program otrzymujemy w sposób następujący:

- z Π_1 usuwamy rozkazy $T_1(i,k)$ oraz $T_1(j,k)$ dla $k = 0, 1, \ldots, n$, otrzymany w ten sposób zbiór oznaczamy przez Π_1';
- wtedy

$$\Pi = \Pi_1' \cup S^{q_1 q_2 \ldots q_s}_{q_i q_{r+1} \ldots q_{r+s+1}} \Pi_2 \cup S^{q_1 q_2 \ldots q_t}_{q_j q_{r+s} \ldots q_{r+s+t-2}} \Pi_3.$$

Złożenia i rozwidlenia maszyn Turinga to zatem pewne operacje na tych maszynach. Przy obliczaniu różnych funkcji czasem wygodnie jest pracować z maszynami Turinga będącymi wynikami operacji na innych, prostszych maszynach Turinga. Dla przykładu, funkcje definiowane przez ***schemat rekursji prostej*** z funkcji prawidłowo obliczalnych w sensie Turinga określać można przez złożenie i rozwidlenie stosownych maszyn Turinga.

Maszyny Turinga

Pokażemy teraz, że maszyny Turinga mogą być kodowane przez liczby naturalne. Niech p_n oznacza n-tą liczbę pierwszą.

Niech $A = a_{s_0} \ldots a_{s_k}$ będzie słowem w alfabecie $\{a_0, a_1, a_2 \ldots\}$. Przyjmijmy:

$$k_l(A) = \prod_{t=0}^{k} p_t^{s_{k-t}}, \; k_r(A) = \prod_{t=0}^{k} p_t^{s_t}.$$

Jeśli $M = Aq_i a_j B$ jest słowem maszynowym, to przyjmijmy:

$$\nu(M) = 2^{k_l(A)} \cdot 3^i \cdot 5^j \cdot 7^{k_r(B)}.$$

Maszyny Turinga

Numerem rozkazu $T(i,j)$ nazwiemy liczbę:

$$\mu(T(i,j)) = p_{c(i,j)}^{p_0^k \cdot p_1^l \cdot p_2^s},$$

gdzie $c(x,y) = \frac{(x+y)^2+3x+y}{2}$ jest ***funkcją numerującą Cantora*** oraz:

$$\begin{array}{lll} s = 0, & \text{jeśli } T(i,j) \text{ jest postaci} & q_i a_j \rightarrow q_k a_l, \\ s = 1, & \text{jeśli } T(i,j) \text{ jest postaci} & q_i a_j \rightarrow q_k a_l L, \\ s = 2, & \text{jeśli } T(i,j) \text{ jest postaci} & q_i a_j \rightarrow q_k a_l R. \end{array}$$

Numerem $\lambda(T)$ ***maszyny*** T nazwiemy iloczyn wszystkich numerów rozkazów $T(i,j)$ maszyny T.

Tak więc, ***maszyny*** oraz ich ***programy*** (a także ich ***dane***) możemy traktować tak, jak liczby naturalne!

Maszyny Turinga: przykłady

1. Jaką funkcję $f(x)$ oblicza maszyna T o następującym programie:

- $q_1 0 \to q_2 0 R$,
- $q_1 1 \to q_0 1$,
- $q_2 0 \to q_0 1$,
- $q_2 1 \to q_2 1 R$?

Odpowiedź. $f(x) = x + 1$.
Aby tę odpowiedź uzasadnić, należy ***udowodnić***, że maszyna Turinga T przetwarza słowo $q_1 0 1^x 0$ w słowo $A q_0 B$ takie, że w $A q_0 B$ występuje $x + 1$ razy symbol 1.

Maszyny Turinga: przykłady

2. Skonstruować maszynę Turinga T, która prawidłowo oblicza funkcję $o(x) = 0$.

Odpowiedź. Na przykład:

$$
\begin{array}{ll}
q_1 0 \longrightarrow q_2 0 R, & \\
q_2 0 \longrightarrow q_3 0 L, & q_2 1 \longrightarrow q_2 1 R, \\
q_3 0 \longrightarrow q_0 0, & q_3 1 \longrightarrow q_3 0 L.
\end{array}
$$

Aby tę odpowiedź uzasadnić, należy ***udowodnić***, że dla dowolnego słowa A mamy:

$$q_1 0 A 0 \underset{T}{\Rightarrow} q_0 000 \ldots 0.$$

Maszyny Turinga: przykłady

3. Skonstruować maszynę Turinga T prawidłowo obliczającą funkcję $f(x, y) = x + y$.

Odpowiedź. Na przykład:

$$
\begin{array}{ll}
q_1 0 \longrightarrow q_2 0 R, & \\
q_2 0 \longrightarrow q_3 1 R, & q_2 1 \longrightarrow q_2 1 R, \\
q_3 0 \longrightarrow q_4 0 L, & q_3 1 \longrightarrow q_3 1 R, \\
 & q_4 1 \longrightarrow q_5 0 L, \\
q_5 0 \longrightarrow q_0 0, & q_5 1 \longrightarrow q_5 1 L.
\end{array}
$$

Aby tę odpowiedź uzasadnić, należy ***udowodnić***, że dla wszystkich x oraz y:

$$q_1 0 1^x 0 1^y 0 \underset{T}{\Rightarrow} q_0 0 1^{x+y} 0 0 \ldots 0.$$

Maszyny Turinga: przykłady

4. Skonstruować maszynę Turinga T obliczającą funkcję $f(x) = x \dot{-} 1$, gdzie $x \dot{-} 1 = x - 1$ dla $x > 0$ oraz $0 \dot{-} 1 = 0$.

Odpowiedź. Na przykład:

$$\begin{array}{ll}
q_1 0 \longrightarrow q_2 0 R, & \\
q_2 0 \longrightarrow q_0 0 L, & q_2 1 \longrightarrow q_3 1 R, \\
q_3 0 \longrightarrow q_4 0 L, & q_3 1 \longrightarrow q_3 1 R, \\
 & q_4 1 \longrightarrow q_5 0 L, \\
q_5 0 \longrightarrow q_0 0, & q_5 1 \longrightarrow q_5 1 L.
\end{array}$$

Aby tę odpowiedź uzasadnić, należy ***udowodnić***, że dla każdego $x > 0$:

$$q_1 0 1^x 0 \underset{T}{\Rightarrow} A q_0 B$$

dla pewnych A, B takich, że symbol 1 występuje $x - 1$ razy w $A q_0 B$ oraz że $q_1 000 \underset{T}{\Rightarrow} C q_0 D$, dla pewnych C, D nie zawierających symbolu 1.

Maszyny Turinga: przykłady

5. Skonstruować maszynę Turinga T prawidłowo obliczającą funkcję $sg(x)$; gdzie $sg(x) = 1$ dla $x > 0$ oraz $sg(0) = 0$.

Odpowiedź. Na przykład:

$$
\begin{array}{ll}
q_1 0 \longrightarrow q_2 0 R, & \\
q_2 0 \longrightarrow q_0 0 L, & q_2 1 \longrightarrow q_3 1 R, \\
q_3 0 \longrightarrow q_4 0 L, & q_3 1 \longrightarrow q_3 0 R, \\
q_4 0 \longrightarrow q_4 0 L, & q_4 1 \longrightarrow q_0 0 L.
\end{array}
$$

Ćwiczenie. Co należy ***udowodnić***, aby uzasadnić tę odpowiedź?

Maszyny Turinga

Jak „mocne" są maszyny Turinga? Języki przez nie rozpoznawane to języki ***rekurencyjnie przeliczalne***. Jest jasne, że nie każdy język może być rozpoznany przez maszynę Turinga: wszystkich języków nad ustalonym skończonym alfabetem jest ***kontinuum***, a maszyn Turinga z tym alfabetem jest tylko ***przeliczalnie*** wiele.

Problem stopu. Problem, czy dowolna maszyna Turinga dla dowolnych danych zakończy obliczenie, jest problemem nierozstrzygalnym.

Maszyny Turinga są dość trudnym narzędziem w ***praktycznych*** obliczeniach. Są jednak przydatne w rozważaniach ***metateoretycznych***. A bez takich rozważań, jak wiadomo, byłabyś ***dzieckiem we mgle***. Bo skąd możesz wiedzieć, że coś robisz ***poprawnie***, jeśli nie wiesz, ***co*** właściwie robisz?

Busy Beaver

Busy Beaver. Jest skończenie wiele zatrzymujących się maszyn Turinga o n stanach nad alfabetem binarnym. Funkcja przyporządkowująca liczbie n największą liczbę symboli 1, która pozostaje na taśmie po zakończeniu pracy maszyny Turinga o n stanach i zaczynającej pracę z taśmą, na której są tylko symbole 0, jest dobrze określona. ***Nie*** jest ona jednak obliczalna przez żadną maszynę Turinga.

Gdybyśmy mieli jakiś systematyczny sposób na otrzymywanie wartości tej funkcji, to można byłoby rozstrzygać prawie każdy problem matematyczny.

Busy Beaver dla $n = 6$ wyprodukował 10^{865} symboli 1, zatrzymując się po 10^{1730} krokach.

Test Turinga

Test Turinga. To procedura zaproponowana przez Turinga. Masz dwóch (niewidocznych dla ciebie) interlokutorów: człowieka oraz komputer. Jeśli nie potrafisz wskazać, który z nich jest komputerem, to należy uznać, iż maszyna przeszła test na bycie podmiotem ***inteligentnym***.

Test Turinga nie jest problemem matematycznym. Wzbudza natomiast, z oczywistych względów, wiele emocji.

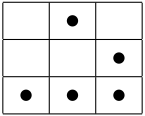
Chiński pokój

Chinese Room. To problem zaproponowany przez ***Johna Searle'a***. Nie znasz jezyka chińskiego. Masz do dyspozycji kompletne reguły przekładu z chińskiego na twój język oraz z twojego języka na chiński. Zamknięto cię w pokoju, do którego dostarczane są pytania po chińsku. Posługując się tylko wspomnianymi regułami, udzielasz odpowiedzi. Czy oznacza to, że ***rozumiesz*** język chiński?

Ten problem wiąże się zatem z pytaniem o to, jaka część naszej wiedzy ma charakter algorytmiczny.

Gra Conwaya

Gra Conwaya. Popularna „gra w życie” ma taką samą moc obliczeniową, jak uniwersalna maszyna Turinga. A oto ***glider***, jeden z bohaterów tej gry (a także symbol subkultury hakerów):



Inne matematyczne modele obliczalności

Jest wiele takich modeli; pokazuje się, że są one ***równoważne*** (określają tę samą klasę funkcji).

- Algorytmy Markowa
- Rachunek lambda
- Systemy równań Herbranda-Gödla
- Rachunek kombinatorów
- Universal Register Machines
- Systemy Posta
- Obliczalność wedle Kołmogorowa
- Abstract State Machines

Teza Churcha-Turinga

Teza Churcha-Turinga. Każda funkcja, która jest (w intuicyjnym sensie) obliczalna, jest obliczalna przez pewną maszynę Turinga. To oczywiście **nie** jest twierdzenie matematyczne. Także twierdzenie do niego odwrotne, głoszące, iż każda funkcja obliczalna przez pewną maszynę Turinga jest (w intuicyjnym sensie) obliczalna, ***nie*** jest twierdzeniem matematycznym. To, że te dwie klasy funkcji są identyczne (a także pokrywają się z klasami wyznaczonymi przez inne matematyczne reprezentacje obliczalności) nazywa się ***Tezą Churcha-Turinga***.

Czasami rozdzielamy: Teza Churcha dotyczy obliczeń na liczbach naturalnych, Teza Turinga „obliczeń" na ciągach wyrażeń ustalonego alfabetu.

Wykorzystywana literatura polska

- Grzegorczyk, A. 1973. *Zarys logiki matematycznej.* PWN, Warszawa.
- Ławrow, I.A., Maksimowa, L.L. 2004. *Zadania z teorii mnogości, logiki matematycznej i teorii algorytmów.* Wydawnictwo Naukowe PWN, Warszawa.
- Moczurad, M. 2002. *Wybrane zagadnienia z teorii rekursji.* Wydawnictwo Uniwersytetu Jagiellońskiego, Kraków.
- Murawski, R. 2000^3. *Funkcje rekurencyjne i elementy metamatematyki. Problemy zupełności, rozstrzygalności, twierdzenia Gödla.* Wydawnictwo Naukowe UAM, Poznań.

Wykorzystywana literatura obcojęzyczna

- Barwise, J. (ed.) 1977. *Handbook of mathematical logic.* North-Holland Publishing Company, Amsterdam, New York Oxford.
- Boolos, G.S., Burgess, J.P., Jeffrey, R.C. 2002. *Computability and logic.* Cambridge University Press, Cambridge.
- Cori, R., Lascar, D. 2001. *Mathematical Logic. A Course with Exercises.* Oxford University Press, Oxford.
- Cutland, N. 1980. *Computability. An introduction to recursive function theory.* Cambridge University Press, Cambridge.
- Hinman, P.G. 2005. *Fundamentals of Mathematical Logic.* A K Peters, Wellesley.
- Kleene, S.C. 1952. *Introduction to Metamathematics.* Wolters-Noordhoff Publishing — Groningen, North-Holland Publishing Company — Amsterdam Oxford, American-Elsevier Publishing Company, Inc. — New York.

Wykorzystywana literatura obcojęzyczna

- Mendelson, E. 1997. *Introduction to mathematical logic.* Chapman & Hall, London.
- Odifreddi, P.G. 1989. *Classical recursion theory.* North-Holland Publishing Company, Amsterdam.
- Odifreddi, P.G. 1999. *Classical recursion theory, Vol. II.* North-Holland Publishing Company, Amsterdam.
- Rogers, H. 1972. *Theory of Recursive Functions and Effective Computability.* MIT Press, Cambridge.
- Shoenfield, J.R. 1967. *Mathematical logic.* Addison-Wesley, Reading, MA.
- Smullyan, R. 1993. *Recursion theory for metamathematics.* Oxford University Press.
- Soare, R.I. 1987. *Recursively enumerable sets and degrees.* Springer-Verlag, Berlin.