Modele Obliczeń

Wykład 3 - Maszyny RAM i funkcje rekurencyjne

Marcin Szczuka

Instytut Matematyki, Uniwersytet Warszawski

Wykład fakultatywny w semestrze zimowym 2008/2009

Plan wykładu

Plan wykładu

Random Access Machine (RAM)

Maszyna o dostępie swobodnym (Random Access Machine – RAM) jest jednym z najstarszych sposobów opisu formalnego procesu obliczania. Została zaproponowana w połowie lat czterdziestych XX wieku przez Johna von Neumanna we współpracy z Johnem Williamem Mauchly i J. Presper Eckertem. Miała pomóc w zrozumieniu zasad działania pierwszych maszyn elektronowych (EDVAC, ENIAC).

Maszyna RAM

Random Access Machine (RAM) składa się z:

- Taśmy wejściowej, z której maszyna wczytuje dane wejściowe;
- Taśmy wyjściowej, na której maszyna zapisuje wyniki działania;
- Nieskończenie wielu rejestrów $R_0, R_1, \ldots$, z których każdy może przechowywać liczbę całkowitą;
- Listy instrukcji (programu) $\Pi = (\pi_1, \ldots, \pi_m)$.

Przez r_i oznaczamy wartość przechowywaną w rejestrze R_i.

Instrukcje w modelu RAM

W skład ciągu instrukcji Π mogą wchodzić następujące operacje:

LOAD **op**	READ **op**	STORE **op**	WRITE **op**
ADD **op**	SUB **op**	MULT **op**	DIV **op**
JUMP **et**	JGTZ **et**	JZERO **et**	HALT

op może przyjmować postać:

- liczby całkowitej – ADD 3;
- wartości rejestru – MULT r_3;
- wskaźnika do wartości rejestru r_{r_i} – STORE r_{r_7}.

et etykieta (numer) instrukcji, np. JZERO 7, JGTZ **początek**.

Akumulator i licznik

Rejestr r_0 jest nazywany **akumulatorem** i ma specjalne znaczenie. Służy jako przestrzeń robocza do przechowywania danych dla bieżącej operacji. Kolejność wykonywania instrukcji jest kontrolowana przez **licznik programu** κ, który przechowuje numer aktualnie przetwarzanej instrukcji.

Plan wykładu

Przykład obliczenia w RAM

Gdybyśmy chcieli napisać program, który liczy funkcję

$$f(n) = \begin{cases} n^n & n > 0 \\ 0 & w\ p.p. \end{cases}$$

to wyglądałby on mniej więcej tak:

```
read n to r1;
if(r1 = 0) then return(0)
else {
  r2 := r1;
  r3 := r1 - 1;
   while r3 > 0 {
   r2 := r2 * r1;
   r3 := r3 - 1;
   };
   return(r2);
};
```

Program dla RAM

Etykieta	Instrukcja	Działanie
	READ r_1	Wczytanie n do r_1
	LOAD r_1	Wczytanie n do akumulatora
	JGTZ `pos`	`if(`$r_1 = 0$ `)`
	WRITE 0	`then return(0)`
	JUMP endif	Skok na koniec programu
`pos:`	LOAD r_1	`else` – $r_1 = n$ do akumulatora
	STORE r_2	$r_2 := r_1$
	SUB 1	$r_0 := r_0 - 1$
	STORE r_3	$r_3 := r_1 - 1$
`while:`	LOAD r_3	r_3 do akumulatora
	JGTZ `continue`	`while` $r_3 > 0$
	JUMP `endwhile`	gdy $r_3 = 0$ - zakończ
`continue:`	LOAD r_2	r_2 do akumulatora
	MULT r_1	$r_0 := r_2 * r_1$
	STORE r_2	$r_2 := r_2 * r_1$
	LOAD r_3	$r_0 := r_2$
	SUB 1	$r_0 := r_0 - 1$
	STORE r_3	$r_3 := r_3 - 1$
`endwhile:`	WRITE r_2	`return(`r_2`)`
`endif:`	HALT	Koniec programu

Plan wykładu

Złożoność obliczeniowa a model RAM

Istotna uwaga

Model maszyny o dostępie swobodnym jest nierealistyczny, jeżeli założymy, że nieskończenie (przeliczalnie) wiele rejestrów jest w stanie przechowywać dowolnie duże liczby całkowite bez dodatkowych kosztów związanych z odczytem, zapisem i przetwarzaniem.

Przy założeniu stałego kosztu (czasu) wszystkich operacji w maszynie RAM moglibyśmy policzyć wartość wyrażenia 2^{2^k} w k krokach, a co za tym idzie koszt obliczeniowy dla tej funkcji byłby liniowy. Tymczasem w maszynie Turinga samo zapisanie wyniku (reprezentacja binarna) na taśmie wymaga co najmniej 2^k kroków.

Złożoność operacji RAM

Dla otrzymania bardziej sensownych oszacowań złożoności obliczeniowej procedur realizowanych przez model RAM wprowadza się koszty wykonywania poszczególnych operacji. Koszty operacji są uzależnione od rozmiaru argumentu.

Będziemy przyjmować, że koszt przetworzenia liczby całkowitej n jest równy jej długości w zapisie pozycyjnym (binarnym) – $\log n$.

Koszt niektórych operacji RAM

ADD/SUB r_1 – $\log(r_0) + \log(r_1)$
MULT 2 – $\log(r_0) + 1$
LOAD r_{r_3} – $\log(r_3) + \log(r_{r_3})$
STORE r_7 – $\log(r_0) + 3$

Złożoność obliczenia w RAM

Złożoność obliczeniową RAM możemy teraz przedefiniować jako sumę kosztów wszystkich wykonanych w algorytmie kroków.
Analogicznie, możemy zdefiniować złożoność pamięciową obliczenia w RAM jako największy obszar jaki zajmowały rejestry (zapisane w nich liczby) w trakcie wykonywania poszczególnych kroków programu, tj.

$$\max_{\kappa} \sum_{i} \log(|r_i|),$$

gdzie κ jest licznikiem programu (pilnuje numeru wykonywanej instrukcji).

Powrót do przykładu

W przedstawionym przykładzie obliczania funkcji n^n złożoności policzone z uwzględnieniem kosztu operacji to odpowiednio:

- $O(n^2 \log n)$ – złożoność czasowa;
- $O(n \log n)$ – złożoność pamięciowa.

Model RAM vs. maszyny Turinga

Symulacja TM w modelu RAM

Twierdzenie: Jeżeli zadanie rozpoznania języka L przez deterministyczną maszynę Turinga M ma złożoność czasową $f(n)$ (n – rozmiar danych wejściowych) to istnieje program RAM, który oblicza funkcję charakterystyczną χ_L języka L w czasie $O(f(n))$.

Symulacja programu RAM w modelu TM

Twierdzenie: Jeżeli program Π maszyny RAM wylicza funkcję całkowitoliczbową ϕ w czasie $f(n)$ to istnieje deterministyczna maszyna Turinga M o siedmiu taśmach taka, że M oblicza ϕ w czasie $O(f(n)^3)$.

Dowody (stosunkowo elementarne) obu tych twierdzeń można znaleźć w rozdziale 2.6 książki:

C.H. Papadimitriou
Złożoność obliczeniowa.
WNT, Warszawa, 2002.

Plan wykładu

Motywacja

Stwierdziliśmy już wcześniej, że jednym ze sposobów reprezentacji obliczenia może być utożsamienie go z funkcją całkowitoliczbową (a nawet naturalną) która przyporządkowuje wyjście wejściu.

Oczywiście natychmiast pojawia się pytanie, o to czy można jakoś dobrze wyróżnić klasy funkcji odpowiadających sensownym rodzajom obliczeń.

Inną kwestią jest jak powiązać klasy funkcji całkowitoliczbowych z poznanymi wcześniej modelami obliczeń.

Podstawowe wymagania

Jest jak najbardziej na miejscu wymagać od zbiorów funkcji które uznamy za "obliczalne", aby zawierały funkcje intuicyjnie możliwe do wyliczenia.

Plan wykładu

Funkcje elementarne

Klasa funkcji elementarnie rekurencyjnych jest intuicyjnie najprostszą "sensowną" klasą funkcji, dla których jesteśmy skłonni przyznać, że mogą być obliczalne.

Definicja: Klasą **funkcji elementarnie rekurencyjnych**, lub krócej **elementarnych EL** nazwiemy najmniejszy zbiór funkcji z $\mathbb{N}$ w $\mathbb{N}$ taki że:

1. **EL** zawiera następujące funkcje specjalne:
   1. Dwuargumentową funkcję dodawania $+ : \mathbb{N}^2 \to \mathbb{N}$;
   2. Dwuargumentową funkcję mnożenia $* : \mathbb{N}^2 \to \mathbb{N}$;
   3. Dwuargumentową funkcję $f : \mathbb{N}^2 \to \mathbb{N}$ taką, że $f(m,n) = |m-n|$;
   4. Dwuargumentową funkcję $g : \mathbb{N}^2 \to \mathbb{N}$ taką, że $g(m,n) = \lfloor \frac{m}{n} \rfloor$;
   5. n-argumentowe funkcje rzutu na i-tą współrzędną dla wszystkich $n, i \in \mathbb{N}$ i $0 < i \leq n$. Przy oznaczeniu U_i^n mamy $U_i^n(x_1, \ldots, x_n) = x_i$.
2. **EL** jest zamknięta ze względu na: ***złożenie, sumowanie, produkt***.

Zamkniętość funkcji elementarnych

1. Jeżeli f jest funkcją m argumentową, a $g_1, \ldots, g_m$ funkcjami n argumentowymi to złożeniem f z $g_1, \ldots, g_m$ nazywamy taką funkcję n-argumentową $K(f; g_1, \ldots, g_m)$, że dla $x_1, \ldots, x_n \in \mathbb{N}$

   $$K(f; g_1, \ldots, g_m)(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n))$$

2. Jeżeli f jest funkcją m argumentową to powiemy, że m argumentowa funkcja g została otrzymana z f przez podsumowanie, co oznaczamy przez $g = \sum f$, jeśli dla $x_1, \ldots, x_m \in \mathbb{N}$

   $$g(x_1, \ldots, x_m) = \sum_{y=0}^{x_m} g(x_1, \ldots, x_{m-1}, y)$$

3. Jeżeli f jest funkcją m argumentową to powiemy, że m argumentowa funkcja g została otrzymana z f przez wymnożenie, co oznaczamy przez $g = \prod f$, jeśli dla $x_1, \ldots, x_m \in \mathbb{N}$

   $$g(x_1, \ldots, x_m) = \prod_{y=0}^{x_m} g(x_1, \ldots, x_{m-1}, y)$$

Przykłady funkcji elementarnych

Funkcje elementarne mają różne pożyteczne własności. W szczególności wiele intuicyjnie elementarnych funkcji należy do tej klasy.

Twierdzenie

Następujące funkcje działające na liczbach naturalnych są elementarne:

- n-argumentowa funkcja stała $C_m^n(x_1, \ldots, x_n) = m$, dla $n > 0, m \in \mathbb{N}$;
- Funkcja następnika $s(n) = n + 1$;
- Funkcje $sg(n) = \begin{cases} 0 & \text{gdy} \quad n = 0 \\ 1 & \text{gdy} \quad n \neq 0 \end{cases}$ i $\overline{sg}(n) = \begin{cases} 1 & \text{gdy} \quad n = 0 \\ 0 & \text{gdy} \quad n \neq 0 \end{cases}$;
- Funkcja wykładnicza $f(x, y) = x^y$;
- Silnia $n!$;
- Funkcja poprzednika $p(n) = \begin{cases} 0 & \text{gdy} \quad n = 0 \\ n - 1 & \text{gdy} \quad n \neq 0 \end{cases}$.

Dowód

1. C_0^1 jest elementarna, bo $C_0^1(x) = |x - x|$ dla $x \in \mathbb{N}$;
2. $\overline{sg}$ jest elementarna, bo $\overline{sg}(x) = \prod_{y<x} C_0^1(y)$;
3. sg jest elementarna, bo $sg(x) = \overline{sg}(\overline{sg}(x))$;
4. C_1^1 jest elementarna, bo $C_1^1(x) = \overline{sg}(C_0^1(x))$;
5. C_m^1 jest elementarna, bo (indukcyjnie) $C_{m+1}^1(x) = C_m^1(x) + C_1^1(x)$;
6. C_m^n jest elementarna, bo $C_m^n(x_1, \ldots, x_n) = C_m^1(U_1^n(x_1, \ldots, x_n))$;
7. $s(x) = x + C_1^1(x)$;
8. Funkcja wykładnicza jest elementarna, bo $x^y = \prod_{z<y} U_1^2(x, z)$;
9. Silnia jest elementarna, bo $x! = \prod_{z<x} s(z)$;
10. Poprzednik jest elementarny, bo $p(x) = |x - C_1^1(x)| * sg(x)$.

Relacje elementarne

Definicja relacji elementarnej

Relację n-argumentową R nazywamy elementarną, jeśli jej funkcja charakterystyczna χ_R jest elementarna.

Taka definicja relacji elementarnej pociąga za sobą pewne pożyteczne własności relacji elementarnych.

Własności relacji elementarnych

Twierdzenie: Relacje elementarne mają następujące własności:

1. Jeżeli relacje n-argumentowe R i S są elementarne to relacje $R \cup S$, $R \cap S$, $R \setminus S$ także są elementarne.
2. Każda skończona relacja jest elementarna.
3. Binarne relacje $\leq, <, \geq, =, \neq$ są elementarne.

Definiowanie przez przypadki

Bardzo często przy definiowaniu funkcji posługujemy się definicjami wariantowymi (np. poprzednio podane sgn). Klasa funkcji elementarnych jest tak skonstruowana, że definiowanie przez przypadki (wariantowe) jest uprawnione. Mówi o tym następujące twierdzenie:

Twierdzenie

Niech $g_1, \ldots, g_m$ będą elementarnymi funkcjami o n argumentach, a $R_1, \ldots, R_m$ elementarnymi relacjami n wymiarowymi takimi, że $\bigcup_{t \leq m} R_t = \mathbb{N}^n$. Wtedy funkcja

$$f(x_1, \ldots, x_n) = \begin{cases} g_1(x_1, \ldots, x_n) & \text{gdy} \quad (x_1, \ldots, x_n) \in R_1 \\ g_2(x_1, \ldots, x_n) & \text{gdy} \quad (x_1, \ldots, x_n) \in R_2 \\ \vdots & \vdots \\ g_m(x_1, \ldots, x_n) & \text{gdy} \quad (x_1, \ldots, x_n) \in R_m \end{cases}$$

także jest elementarna.

Plan wykładu

Idea funkcji pierwotnie rekurencyjnych

Badania nad funkcjami, które mogą być realizowane algorytmicznie zostały podjęte niezależnie przez kilku badaczy już na przełomie lat dwudziestych i trzydziestych XX wieku. Powstało kilka konkurencyjnych podejść, w tym funkcje pierwotnie i częściowo rekurencyjne (Kleene).

Funkcje pierwotnie rekurencyjne

Kleene wyróżniał podstawowe funkcje obliczalne:

- Funkcja stała (równa 0) $\mathbf{0}(n) \equiv 0$;
- Następnik $s(n) = n + 1$;
- U_i^n – n-argumentowe funkcje rzutu na i-tą współrzędną dla wszystkich $n, i \in \mathbb{N}$ i $0 < i \leq n$.

Oraz wymagał by klasa tych funkcji była zamknięta ze względu na operacje **złożenia** i **rekursji pierwotnej**.

Rekursja pierwotna

Niech $g(\overrightarrow{x})$, $h(\overrightarrow{x}, m, y)$ dla $\overrightarrow{x} \in \mathbb{N}^{n-1}$ będą funkcjami odpowiednio $n-1$ i $n+1$ argumentowymi. Wtedy n-argumentową funkcję $f(\overrightarrow{x}, m)$ definiujemy przez pierwotną (prymitywną) rekursję jako:

$$f(\overrightarrow{x}, 0) = g(\overrightarrow{x})$$

$$f(\overrightarrow{x}, m+1) = h(\overrightarrow{x}, m, f(\overrightarrow{x}, m))$$

Tak zdefiniowaną operację rekursji można traktować jako przekształcenie:

$$Rek : \mathbb{N}^{\mathbb{N}^{n-1}} \times \mathbb{N}^{\mathbb{N}^{n+1}} \mapsto \mathbb{N}^{\mathbb{N}^n}$$

Mówimy, że funkcja $f(\overrightarrow{x}, m)$ jest zadana przez niesparametryzowaną rekursję pierwotną gdy g jest stała.

Klasa funkcji pierwotnie rekurencyjnych

Definicja

Klasą funkcji pierwotnie rekurencyjnych (**PREK**) nazwiemy najmniejszy zbiór funkcji naturalnych taki, że:

1. Funkcja zerowa, następnik i rzutowania należą do **PREK**;
2. **PREK** jest zamknięta na operacje złożenia, i rekursji pierwotnej (sparametryzowanej i niesparametryzowanej)

Powiemy, że relacje jest pierwotnie rekurencyjna, jeśli jej funkcja charakterystyczna należy do **PREK**.

Zauważmy, że wszystkie tak określone funkcje pierwotnie rekurencyjne są **totalne**.

Intuicyjnie możemy myśleć o klasie **PREK** jako o funkcjach które mogą być wyliczane przez programy wykorzystujące odwołania rekurencyjne, ale nie zawierające pętli `while` i instrukcji skoku `goto`. Dokładniej mówiąc, liczba wykonań każdej pętli musi być znana z góry.

Przykłady funkcji pierwotnie rekurencyjnych

Niemal wszystkie (intuicyjnie) podstawowe funkcje totalne są pierwotnie rekurencyjne. Na przykład:

- Suma dwóch liczb naturalnych jest w $\mathbf{PREK}$ gdyż

$$suma(x, 0) = U_1^1(x)$$

$$suma(x, y + 1) = s(U_3^3(x, y, suma(x, y)))$$

- Iloczyn dwóch liczb naturalnych jest w $\mathbf{PREK}$ gdyż

$$prod(x, 0) = \mathbf{0}(x)$$

$$prod(x, y) = x + prod(x, y - 1) =$$

$$= suma(U_1^3(x, y, prod(x, y)), U_3^3(x, y, prod(x, y - 1)))$$

- Funkcje takie jak: x^y, $x - y$, $sg(x)$, dzielenie całkowite, $\min$, $\max$, modulo, ...

Własności funkcji pierwotnie rekurencyjnych

Twierdzenie

Każda funkcja elementarna jest funkcją pierwotnie rekurencyjną. Instnieją nieelementarne funkcje pierwotnie rekurencyjne. Zatem

$$\mathbf{EL} \subsetneq \mathbf{PREK}$$

Twierdzenie – ograniczoność rekursji pierwotnej

Niech $g(\overrightarrow{x})$, $h(\overrightarrow{x}, m, y)$ będą elementarne dla $\overrightarrow{x} \in \mathbb{N}^{n-1}$, a funkcja f powstaje z g i h przez zastosowanie rekursji pierwotnej. Jeżeli istnieje n-argumentowa funkcja elementarna k taka, że dla każdego $(x_1, \ldots, x_n) \in \mathbb{N}^n$ zachodzi:

$$f(x_1, \ldots, x_n) \leq k(x_1, \ldots, x_n)$$

to f jest elementarna.

Twierdzenie (Robinson)

Każda funkcja z $f(x) \in \mathbf{PREK}^1$ (jednoargumentowa funkcja pierwotnie rekurencyjna) może być otrzymana z $s(x)$ i $q(x) =_{df} x - \lfloor\sqrt{x}\rfloor^2$ za pomocą skończonej liczby:

- Dodawań $(f+g)(x) = f(x) + g(x)$;
- Superpozycji $(f \circ g)(x) = f(g(x))$;
- Iteracji takich że: $(\mathcal{I}f)(0) = 0$, $(\mathcal{I}f)(n+1) = f((\mathcal{I}f)(n))$.

Uwaga: Tutaj $x - y = \begin{cases} x-y & \text{gdy} \quad x \geq y \\ 0 & \text{gdy} \quad x < y \end{cases}$.

Twierdzenie

Istnieje funkcja uniwersalna $F(n,x)$ dla $\mathbf{PREK}^1$ ale nie należy ona do $\mathbf{PREK}$. (Patrz następny slajd.)

Funkcja uniwersalna dla $\mathbf{PREK}^1$

Szkic dowodu twierdzenia z poprzedniego slajdu:
Z twierdzenia Robinsona wynika, że funkcji w $\mathbf{PREK}^1$ jest przeliczalnie wiele. Możemy je zatem ustawić w ciąg $(f_i)_{i\in\mathbb{N}}$.
Funkcją uniwersalna dla $\mathbf{PREK}^1$ będzie zdefiniowana jako $F(n,x) = f_n(x)$.
Pokażemy, że $F \notin \mathbf{PREK}$.

1. Przypuśćmy, że $F \in \mathbf{PREK}$, wtedy $g(x) = F(x,x) + 1 \in \mathbf{PREK}^1$.
2. Jeśli $g \in \mathbf{PREK}^1$ to istnieje takie n_0, że $g = f_{n_0}$.
3. Wtedy $g(n_0) = f_{n_0}(n_0)$, ale jednocześnie $g(n_0) = F(n_0,n_0) + 1 = f_{n_0}(n_0) + 1$ i **sprzeczność**.

Zatem $F \notin \mathbf{PREK}$.
UWAGI: Powyższy dowód jest prawdziwy pod warunkiem prawdziwości nieudowodnionego tw. Robinsona. Funkcja F nie jest pierwotnie rekurencyjna, ale intuicyjnie jest obliczalna, więc być może klasa $\mathbf{PREK}$ jest zbyt uboga.

Plan wykładu

Minimalizacja

Dla poradzenia sobie z problemami wynikającymi ze zbyt słabych własności klasy funkcji pierwotnie rekurencyjnych musimy wzbogacić nasz warsztat o operację minimum, która pozwala "domknąć" klasę funkcji obliczalnych.

Definicja

Niech $g : \mathbb{N}^{n+1} \mapsto \{0, 1\}$ Powiemy, że funkcja n-argumentowa f jest zdefiniowana przez **minimalizację** g, jeśli dla $\overrightarrow{x} \in \mathbb{N}^n$

$$f(\overrightarrow{x}) = \begin{cases} \min\{y : g(\overrightarrow{x}, y) = 0\} & \text{gdy znajdziemy takie } \overrightarrow{x}, y \\ \text{nieznana} & \text{w p.p.} \end{cases}$$

Zapisujemy to jako $f(\overrightarrow{x}) = \mu y g(\overrightarrow{x}, y)$.

Zauważmy, że tak zdefiniowana funkcja f może nie być określoną na całej dziedzinie (totalną) lecz funkcją częściową.
Jeżeli $f(\overrightarrow{x}) = \mu y g(\overrightarrow{x}, y)$ i z tego, że g jest określona dla wszystkich $\overrightarrow{x}$ i y wynika, że f jest określona dla wszystkich $\overrightarrow{x}$ to mówimy że f powstaje z g za pomocą **minimum efektywnego**.

Funkcje częściowo rekurencyjne

CZREK

Najmniejszą klasę funkcji zawierającą funkcje podstawowe (zero, następnik, rzutowania) i zamkniętą na złożenie, rekursję i minimum nazywamy klasą funkcji **częściowo rekurencyjnych** i oznaczamy $\mathbf{CZREK}$.

REK

Najmniejszą klasę funkcji totalnych zawierającą funkcje podstawowe (zero, następnik, rzutowania) i zamkniętą na złożenie, rekursję i minimum efektywne nazywamy klasą funkcji **rekurencyjnych** i oznaczamy $\mathbf{REK}$.

W oczywisty sposób:

$$\mathbf{EL} \subsetneq \mathbf{PREK} \subsetneq \mathbf{REK} \subsetneq \mathbf{CZREK}$$

UWAGA: Bardziej będzie nas interesować klasa $\mathbf{CZREK}$, gdyż to ona lepiej odpowiada intuicji klasy funkcji obliczalnych.

Funkcje istotnie częściowo rekurencyjne

Dobrym przykładem funkcji istotnie częściowo rekurencyjnej jest funkcja odwrotna do funkcji różnowartościowej. Jeśli $f : \mathbb{N} \mapsto \mathbb{N}$, taka że $f(x) = f(y) \Rightarrow x = y$ to (częściową) funkcję odwrotną

$$f^{-1}(x) = \begin{cases} y & \text{gdy} \quad \exists_y f(y) = x \\ \text{nieznana} & \text{w p.p.} \end{cases}$$

możemy wyznaczyć jako $\mu y(f(y) = x)$.

Wcześniej widzieliśmy już przykład funkcji uniwersalnej dla $\mathbf{PREK}^1$, która nie była pierwotnie rekurencyjna. Można wszakże pokazać, że funkcja uniwersalna dla $\mathbf{PREK}^1$ jest w $\mathbf{REK}$.

Niestety wspomniana funkcja uniwersalna nie jest zdefiniowana konstruktywnie i jakkolwiek stanowi przykład funkcji pochodzącej z $\mathbf{CZREK} \setminus \mathbf{PREK}$, to nie jest zbyt przydatna.

Funkcja Ackermana

Przykład (bardzo istotny) funkcji, która jest rekurencyjna, ale nie pierwotnie rekurencyjna stanowi **funkcja Ackermana**.

Funkcja Ackermana

Dla $m, n \in \mathbb{N}$:

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

Funkcja Ackermana rośnie szybciej niż każda funkcja w $\mathbf{PREK}^1$ tzn.

$$\forall_{f \in \mathbf{PREK}^1} \exists_m \forall_{n>m} Ack(n, n) > f(n)$$

Nota bene: Funkcja Ackermana jest wcześniejsza niż teoria funkcji obliczalnych. Także zakres przydatności funkcji Ackermana wykracza poza teorię obliczeń.

Ograniczoność minimum efektywnego

Nieco światła na wzajemne relacje między klasą funkcji pierwotnie rekurencyjnych, a funkcjami rekurencyjnymi rzuca twierdzenie o **minimum ograniczonym**.

Twierdzenie o minimum ograniczonym

Niech:

1. $g(\overrightarrow{x}, y) \in \mathbf{PREK}$ dla $\overrightarrow{x} \in \mathbb{N}^n, y \in \mathbb{N}$;
2. $f(\overrightarrow{x}) = \mu y(g(\overrightarrow{x}, y) = 0)$ i to minimum jest efektywne (f jest totalna);
3. $k(\overrightarrow{x}) \in \mathbf{PREK}$ dla $\overrightarrow{x} \in \mathbb{N}^n$;
4. $\forall_{\overrightarrow{x} \in \mathbb{N}^n} f(\overrightarrow{x}) \leq k(\overrightarrow{x})$.

Wtedy f jest pierwotnie rekurencyjna.

Plan wykładu

Problemy obliczalne jako zbiory

Zagadnienie obliczalności (opisywalności) różnych zagadnień można przeformułować jako zagadnienie opisywania zbioru obiektów spełniających określone warunki.
Na przykład:

- W zbiorze wszystkich grafów nieskierowanych wyznaczyć podzbiór grafów, które mają klikę o zadanej mocy $k \in \mathbb{N}$;
- W zbiorze liczb naturalnych wyznaczyć zbiór wszystkich liczb pierwszych.

Charakteryzacji zbiorów tego typu dokonujemy przez określenie ich funkcji charakterystycznej. W ten sposób możemy wprowadzić pojęcie zbioru pierwotnie rekurencyjnego i rekurencyjnego. Problem może się pojawić przy funkcjach częściowo rekurencyjnych.

Zbiory rekurencyjne

Zbiory pierwotnie rekurencyjne

Powiemy, że zbiór $A \subset \mathbb{N}$ jest pierwotnie rekurencyjny jeśli jego funkcja charakterystyczna χ_A jest w $\mathbf{PREK}$.

Zbiory rekurencyjne

Powiemy, że zbiór $A \subset \mathbb{N}$ jest pierwotnie rekurencyjny jeśli jego funkcja charakterystyczna χ_A jest rekurencyjna. Zbiory rekurencyjne mają ścisły związek z obliczalnością i rozstrzygalnością.

Rodzina zbiorów pierwotnie rekurencyjnych (rekurencyjnych) jest zamknięta ze względu na $\cap, \cup, \setminus$.

- $\emptyset, \mathbb{N}$ są pierwotnie rekurencyjne z funkcjami charakterystycznymi równymi (odpowiednio) stale 0 i 1.
- Dowolny skończony podzbiór $A = \{n_1, \ldots, n_k\}, n_i, k \in \mathbb{N}$ jest pierwotnie rekurencyjny z funkcją charakterystyczną $\chi_A(x) = sg(|(x - n_1) \cdot \ldots \cdot (x - n_k)|)$.

Zbiory rekurencyjnie przeliczalne

Definicja

Zbiór $A \subset \mathbb{N}$ jest **rekurencyjnie przeliczalny** (**re** – *recursively enumerable*), jeśli istnieje funkcja pierwotnie rekurencyjna $F(x, y)$ taka, że:

$$x \in A \Leftrightarrow \exists_{y \in \mathbb{N}}(F(x, y) = 0).$$

Zbiory rekurencyjnie przeliczalne są istotne, gdyż za ich pomocą można charakteryzować zbiory funkcji (częściowo, pierwotnie) rekurencyjnych i pokazywać istotne związki między nimi. Omówimy kilka takich zależności.

Twierdzenie o przeciwobrazie

Niech $f \in \mathbf{PREK}$ oraz $f(x) \geq x$ dla $x \in \mathbb{N}$. Wtedy przeciwobraz $R_f = \{y : \exists_{x \in \mathbb{N}} f(x) = y\}$ jest zbiorem pierwotnie rekurencyjnym.

Dowód: Ponieważ $f(y) \geq y$, wystarczy rozpatrywać

$$\chi_{R_f}(y) = \prod_{x=0}^{y} sg(|f(x) - y|).$$

Własności zbiorów rekurencyjnie przeliczalnych

Twierdzenie

Niech $\emptyset \neq A \subset \mathbb{N}$. A jest $\mathbf{re}$ wtedy i tylko wtedy, gdy istnieje $f \in \mathbf{PREK}^1$ taka, że $R_f = A$.

Twierdzenie (Post)

Jeżeli $A \subset \mathbb{N}$, A jest $\mathbf{re}$ i $\mathbb{N} \setminus A$ jest $\mathbf{re}$ to A jest zbiorem rekurencyjnym.

Zdefiniujemy pojęcie **wykresu** funkcji $f : D_f \mapsto \mathbb{N}$ dla $D_f \subset \mathbb{N}^n$ przez:

$$GF(f) = \{(\overrightarrow{x}, y) \in \mathbb{N}^{n+1} : \overrightarrow{x} \in D_f \wedge f(\overrightarrow{x}) = y\}.$$

Twierdzenie o wykresie

Funkcja $f : \mathbb{N}^n \mapsto \mathbb{N}$ jest częściowo rekurencyjna wtedy i tylko wtedy, gdy $GF(f)$ jest $\mathbf{re}$.

Pożytki ze zbiorów **re**

Dzięki zbiorom rekurencyjnie przeliczalnym można pokazać kilka istotnych faktów dotyczących interesującej nas klasy funkcji częściowo rekurencyjnych.
Jeżeli $f \in \mathbf{CZREK}^1$ i $f \subset g$, to g nazwiemy **rozszerzeniem** f. Jeżeli g jest określona dla wszystkich argumentów to nazywamy ją **uzupełnieniem** f.

Twierdzenie

Istnieje funkcja $f \in \mathbf{CZREK}^1$ taka, że $R_f = \{0, 1\}$ i nie istnieje rekurencyjne uzupełnienie dla f.

Wniosek pośredni

Istnieje zbiór rekurencyjnie przeliczalny, który nie jest rekurencyjny.