Изчислимост и сложност

Добре дошли! :)

Лекция №1

Примитивно рекурсивни и частично рекурсивни функции

1. Специфично за частичните функции

Ще разглеждаме функции в множеството на естествените числа

$$\mathbb{N} = \{0, 1, \dots\},$$

които са *частични*. Това означава, че в някои точки те могат да не са дефинирани, т.е. да нямат стойност. Такива ще са изчислимите функции, които основно ще изучаваме в този курс. Това ще са функциите, които се пресмятат – най-общо казано – с някаква програма. И тъй като програмите, както е известно, невинаги завършват, то значи и функциите, които те пресмятат, в общия случай трябва да са частични.

Ще пишем $f : \mathbb{N}^n \longrightarrow\!\!\!\!\!\!\circ\;\; \mathbb{N}$, за да означим, че f е частична функция на n аргумента в $\mathbb{N}$. Съвкупността от всички такива функции ще отбелязваме с $\mathcal{F}_n$, с други думи

$$\mathcal{F}_n = \{f \mid f : \mathbb{N}^n \longrightarrow\!\!\!\!\!\!\circ\;\; \mathbb{N}\}.$$

По-надолу ще предполагаме, че f е произволна n-местна частична функция. Ако тя е дефинирана в точката $(x_1, \dots, x_n)$, това ще отбелязваме така:

$$!f(x_1, \dots, x_n),$$

а ако не е дефинирана — ще пишем съответно $\neg!f(x_1, \dots, x_n)$.

Множеството от всички точки, в които f е дефинирана, ще наричаме *дефиниционно множество (домейн)* на f и ще означаваме с $Dom(f)$, или формално:

$$Dom(f) = \{(x_1, \dots, x_n) \mid !f(x_1, \dots, x_n)\}.$$

Ако $Dom(f) = \mathbb{N}^n$, ще казваме, че f е *тотална (навсякъде дефинирана)*. Разбира се, всяка тотална функция може да се разглежда и като частична, т.е. тя също принадлежи на $\mathcal{F}_n = \{f \mid f : \mathbb{N}^n \longrightarrow\!\!\!\!\circ\;\, \mathbb{N}\}$. Когато казваме ***функция***, в общия случай ще имаме предвид частична функция. Ако става въпрос за тотална функция, това ще бъде отбелязвано експлицитно, ако не се подразбира от контекста.

По-нататък n-торките $(x_1, \ldots, x_n)$ ще съкращаваме до $\bar{x}$, когато това не води до някаква неяснота.

1.1 Условно равенство

Когато пишем равенство между изрази, в които участват частични функции, е необходимо да уточним какво ще разбираме в случаите, когато някоя от двете страни (или и двете едновременно) не са дефинирани. За тази цел ще използваме нова релация, която ще наричаме *условно равенство* и ще означаваме с $\simeq$.

Определение 1.1. Нека $\alpha(\bar{x})$ и $\beta(\bar{x})$ са изрази, в които участват частични функции. Тогава

$$\begin{aligned} \alpha(\bar{x}) \simeq \beta(\bar{x}) \quad \stackrel{\text{деф}}{\iff} \quad & !\alpha(\bar{x}) \;\&\; !\beta(\bar{x}) \;\&\; \alpha(\bar{x}) = \beta(\bar{x}) \\ & \vee\; \neg!\alpha(\bar{x}) \;\&\; \neg!\beta(\bar{x}). \end{aligned}$$

С други думи, условното равенство има стойност *истина* или когато и двете му страни са дефинирани и имат една и съща стойност, или когато и двете му страни не са дефинирани. В останалите случаи то е *лъжа*. В частност, $f(\bar{x}) \simeq y$ ще е вярно точно когато f е дефинирана в $\bar{x}$ и нейната стойност е y.

Графиката G_f на частичната функция f въвеждаме по обичайния начин:

$$G_f = \{(x_1, \ldots, x_n, y) \mid f(x_1, \ldots, x_n) \simeq y\}.$$

Определение 1.2. За две n-местни частични функции f и g ще казваме, че са *равни* (и ще пишем $f = g$), ако $f(\bar{x}) \simeq g(\bar{x})$ за всяко $\bar{x} \in \mathbb{N}^n$.

Ясно е, че ако $f = g$, то $Dom(f) = Dom(g)$ и $f(\bar{x}) = g(\bar{x})$ за всяко $\bar{x} \in Dom(f)$. Равенството на две функции може да се разпише и по следния начин:

$$\begin{aligned} f = g &\iff \forall x_1 \ldots \forall x_n\; f(x_1, \ldots, x_n) \simeq g(x_1, \ldots, x_n) \\ &\iff \forall x_1 \ldots \forall x_n \forall y (f(x_1, \ldots, x_n) \simeq y \iff g(x_1, \ldots, x_n) \simeq y) \\ &\iff \forall x_1 \ldots \forall x_n \forall y ((x_1, \ldots, x_n, y) \in G_f \iff (x_1, \ldots, x_n, y) \in G_g) \\ &\iff G_f = G_g. \end{aligned}$$

Излезе (без да е изненадващо), че две частични функции са равни точно тогава, когато имат едни и същи графики.

1.2 Релацията включване

Сега ще въведем една релация между частични функции, която няма аналог при тоталните функции. Релацията е *включване* ($\subseteq$) и смисълът ѝ е, че ако $f \subseteq g$, то g "знае повече" от f, или g "носи повече информация" от f. Ето точното определение:

Определение 1.3. Нека $f, g \in \mathcal{F}_n$. Тогава

$$f \subseteq g \overset{\text{деф}}{\iff} \forall x_1 \dots \forall x_n \forall y\ (f(x_1, \dots, x_n) \simeq y \implies g(x_1, \dots, x_n) \simeq y).$$

Ако $f \subseteq g$, ще казваме още, че f е *подфункция* на g или обратно — че g е *продължение* на f. Преразказано, една функция се продължава от друга, ако там, където първата е дефинирана (и значи има някаква стойност), там и втората е дефинирана и има същата стойност.

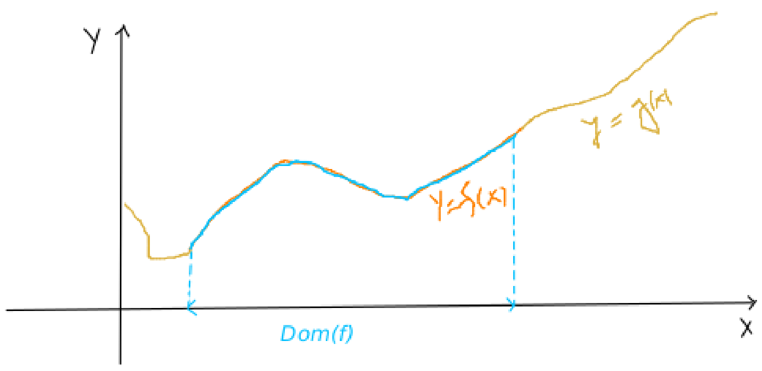
От определението се вижда, че

$$f \subseteq g \iff G_f \subseteq G_g,$$

което обяснява защо използваме теоретико-множествения символ $\subseteq$.
Да отбележим и още един очевиден факт, който ще използваме често:

$$f \subseteq g \implies Dom(f) \subseteq Dom(g).$$

Ето как изглеждат схематично графиките на f и g, такива че $f \subseteq g$:



Ако f е тотална и $f \subseteq g$, то очевидно $f = g$, т.е. върху тоталните функции релацията включване съвпада с релацията равенство.

Когато задаваме някаква функция f и искаме да кажем, че в т. $(x_1, \dots, x_n)$ тя няма стойност, това ще записваме и така: $f(x_1, \dots, x_n) \simeq \neg!$.

Ето един пример за две функции f и g, такива че f е подфункция на g:

Пример 1.1. Да дефинираме функциите f и g както следва:

$$f(x, y) \simeq \begin{cases} \lfloor \frac{x}{y} \rfloor, & \text{ако } y > 0 \\ \neg!, & \text{ако } y = 0, \end{cases}$$

$$g(x, y) = \begin{cases} \lfloor \frac{x}{y} \rfloor, & \text{ако } y > 0 \\ 0, & \text{ако } y = 0. \end{cases}$$

Ясно е, че в точките, в които е дефинирана, f има същата стойност като g, с други думи, $f \subseteq g$.

Релацията *строго включване* ($\subset$) се дефинира от $\subseteq$ по обичайния начин:

$$f \subset g \ \stackrel{\text{деф}}{\iff}\ f \subseteq g \ \& \ f \neq g.$$

За функциите f и g от *Пример* 1.1 от по-горе всъщност имаме $f \subset g$.

От наблюдението, че две функции са равни точно когато графиките им съвпадат, получаваме следната връзка между релациите $=$ и $\subseteq$:

$$\begin{aligned} f = g &\iff G_f = G_g \\ &\iff G_f \subseteq G_g \ \& \ G_g \subseteq G_f \\ &\iff f \subseteq g \ \& \ g \subseteq f. \end{aligned}$$

Излезе, че

$$f = g \iff f \subseteq g \ \& \ g \subseteq f.$$

От тази еквивалентност се вижда един начин да покажем, че две *частични* функции са равни — като проверим, че едната е подфункция на другата и обратно. Оказва се, че можем леко да отслабим това условие, като заменим включването $g \subseteq f$ с по-слабото $Dom(g) \subseteq Dom(f)$. Тази дребна наглед корекция в бъдеще ще ни спестява писане. Но да се убедим първо, че можем да направим това:

Задача 1.1. Нека f и g са n-местни функции. Докажете, че $f = g$ тогава и само тогава, когато са изпълнени условията:

1) $f \subseteq g$;

2) $Dom(g) \subseteq Dom(f)$.

Решение. Ако $f = g$, то $f \subseteq g$ и $g \subseteq f$ и от последното, в частност, следва и включването между домейните $Dom(g) \subseteq Dom(f)$.

Обратно, нека са верни 1) и 2). Трябва да покажем, че $f \subseteq g$ и $g \subseteq f$. Първото включване е точно условието 1). За да покажем, че и $g \subseteq f$, да приемем, че за произволни $\bar{x}, y$ $g(\bar{x}) \simeq y$. Тогава $\bar{x} \in Dom(g)$, а оттук съгласно 2) ще имаме и $\bar{x} \in Dom(f)$, т.е. $f(\bar{x}) \simeq z$ за някое z. Сега от условието 1) получаваме, че и $g(\bar{x}) \simeq z$, и значи $y = z$. И така, получихме, че за произволни $\bar{x}, y$:

$$g(\bar{x}) \simeq y \implies f(\bar{x}) \simeq y,$$

което по дефиницията на $\subseteq$ означава, че $g \subseteq f$. $\square$

От еквивалентността

$$f = g \iff G_f = G_g$$

се вижда, че релацията включване между функции се изразява чрез включване между *множества*, за което знаем, че е частична наредба. Следователно и релацията "подфункция" е*частична наредба*. Да обърнем внимание, че тя също е *частична*, т.е. не всеки две функции от $\mathcal{F}_n$ са свързани чрез нея. Такива са например константните функции f_0 и f_1, които за всяко $\bar{x} \in \mathbb{N}^n$ връщат 0 и 1, съответно. Не се заблуждавайте: вярно е, че $\forall \bar{x}\ f_0(\bar{x}) \leq f_1(\bar{x})$, обаче не е вярно, че $f_0 \subseteq f_1$. $\smile$

Интуитивно, $f \subseteq g$ означава, че f е "по-малко информативна" от g. Тогава "най-малко информативна" ще е функцията, която не е дефинирана в нито една точка.

Всъщност има безброй много такива функции, в зависимост от броя на аргументите им. За фиксирано $n \geq 1$ с $\emptyset^{(n)}$ ще означаваме n-местната функция, която не е дефинирана за нито една n-торка $\bar{x} \in \mathbb{N}^n$ и тази функция ще наричаме *никъде недефинираната* (или *празната*) функция на n аргумента. Да отбележим, че за всяка $f \in \mathcal{F}_n$ е в сила включването

$$\emptyset^{(n)} \subseteq f,$$

с други думи, никъде недефинираната функция $\emptyset^{(n)}$ е най-малкият (относно $\subseteq$) елемент на $\mathcal{F}_n$.

2. Примитивно рекурсивни функции

Дефиницията на примитивно рекурсивните функции идейно прилича на дефиницията на регулярните множества — тръгваме от някакви начални прости обекти и ги затваряме относно някакви прости операции. В нашия случай началните обекти се наричат *изходни (базисни) примитивно рекурсивни функции*, които въвеждаме по-долу заедно с операциите, относно които ще ги затворим. Така ще определим *примитивно рекурсивните* и *частично рекурсивните* функции, въведени от Ербран, Гьодел и Клини.

2.1 Изходни (базисни) примитивно рекурсивни функции

Идеята е това да са възможно най-простите функции над естествените числа, чиято "изчислимост" не оставя съмнение в никого. Разбира се, тези функции трябва да са и достатъчно изразителни, за да можем, тръгвайки от тях, да получим всички възможни изчислими функции.

Определение 1.4. *Изходните (базисните) примитивно рекурсивни функции* са следните:

1) функцията $\mathcal{S}$ (от successor), която дава наследника на всяко $x \in \mathbb{N}$:

$$\mathcal{S}(x) = x + 1;$$

2) *едноместната константна функция* $\mathcal{O}$, която за всяко $x \in \mathbb{N}$ връща 0:

$$\mathcal{O}(x) = 0;$$

3) *проектиращите функции* I^n_k, $n = 1, 2, \ldots$ и $1 \leq k \leq n$, дефинирани като:

$$I^n_k(x_1, \ldots, x_n) = x_k$$

за всяка n-торка $(x_1, \ldots, x_n)$ от $\mathbb{N}^n$.

В частност, при $k = n = 1$ получаваме $I^1_1(x) = x$ за всяко $x \in \mathbb{N}$, т.е. изходната функция I^1_1 е *идентитетът* в $\mathbb{N}$. Предназначението на проектиращите функции е по-скоро техническо.

2.2 Изходни (базисни) операции

Както при избора на изходните функции, и тук целта е тези начални операции да са възможно най-прости и едновременно с това — достатъчно мощни, за да може чрез тях да се зададат всички изчислими функции.

Изходните операции, които ще въведем, следвайки дефиницията на Ербран–Гьодел–Клини, са три — *суперпозиция, примитивна рекурсия и минимизация.* В този раздел ще определим първите две от тях, а в раздел 1.3 — третата.

Определение 1.5. Нека f е произволна n-местна функция, а $g_1, \ldots, g_n$ са n на брой функции, всички на k аргумента. *Суперпозицията* на тези функции е k-местната функция h, която се дефинира по следния начин:

$$h(x_1, \ldots, x_k) \simeq y \overset{\text{деф}}{\iff} \exists z_1 \ldots \exists z_n \; (g_1(x_1, \ldots, x_k) \simeq z_1 \; \& \; \ldots \; \& \\ g_n(x_1, \ldots, x_k) \simeq z_n \; \& \; f(z_1, \ldots, z_n) \simeq y) \qquad (1.1)$$

за всяко $(x_1, \ldots, x_k)$ от $\mathbb{N}^k$

Суперпозицията на f и $g_1, \ldots, g_n$ ще означаваме с

$$f(g_1, \ldots, g_n).$$

При $n = 1$ функцията $f(g)$ ще наричаме ***композиция*** на f и g и ще бележим с обичайното $f \circ g$.

От еквивалентността (1.1) следва в частност, че

$$!f(g_1, \ldots, g_n)(\bar{x}) \iff !g_1(\bar{x}) \; \& \; \ldots \& \; !g_n(\bar{x}) \; \& \; !f(g_1(\bar{x}), \ldots, g_n(\bar{x})).$$

Ако приемем, че така разбираме дефинираността на $f(g_1, \ldots, g_n)(\bar{x})$, определението за суперпозиция можем да запишем и по-кратко като:

$$f(g_1, \ldots, g_n)(\bar{x}) \overset{\text{деф}}{\simeq} f(g_1(\bar{x}), \ldots, g_n(\bar{x})).$$

Втората изходна операция е *примитивна рекурсия.* Най-общо, една функция f се задава с *рекурсия*, ако се определя "чрез себе си", което можем да си представим схематично така:

$$f(x) \;\simeq\; \ldots \; f, \; x \; \ldots .$$

Най-простата схема за дефиниция по рекурсия на *едноместна* функция f е следната: по дадени константа $c \in \mathbb{N}$ и $g \in \mathcal{F}_2$, функцията f определяме посредством равенствата:

$$\left| \begin{array}{l} f(0) = c \\ f(x+1) \simeq g(x, f(x)). \end{array} \right.$$

Горната рекурентна връзка обикновено се нарича *проста схема на примитивна рекурсия*. За f ще казваме, че е получена с *примитивна рекурсия* от c и g. Тук константата c задава *дъното* на рекурсията, а функцията g ни казва как да пресметнем $f(x+1)$, ако знаем $f(x)$.

Букварният пример за дефиниция чрез тази схема е рекурсивната дефиниция на функцията $f(x) = x!$. За нея имаме

$$\left| \begin{array}{l} f(0) = 1 \\ f(x+1) = (x+1).f(x). \end{array} \right.$$

Тук константата $c = 1$, а $g(x, y) = (x + 1)y$.

Да се опитаме да обобщим ситуацията за функция на повече променливи. Как би изглеждала възможно най-простата схема на рекурсия, ако f е на два аргумента примерно? Как да дефинираме рекурсивно $f(x, y)$ — с рекурсия по x, по y или и по двата аргумента?

Известно е, че рекурсия *и по двата аргумента* може да доведе до функции-чудовища. Пример за това е *функцията на Акерман*, която се задава със следната двойна рекурсия:

$$\left| \begin{array}{l} f(0, y) \simeq y + 1 \\ f(x + 1, 0) \simeq f(x, 1) \\ f(x + 1, y + 1) \simeq f(x, f(x + 1, y)). \end{array} \right.$$

Тази функция расте с шеметна скорост — например $f(4, 2)$ е число с 19 729 цифри в десетичния си запис!

Малко офтопик: ако си мислите, че горната рекурсивна схема *поначало* е сложна, защото рекурсията е двойна — ами не винаги е така. Да вземем ето тази рекурсивна схема, която се различава с всичко на всичко "една единица" от дефиницията на Акерман (в дъното на рекурсията):

$$\left| \begin{array}{l} g(0, y) \simeq y \\ g(x + 1, 0) \simeq g(x, 1) \\ g(x + 1, y + 1) \simeq g(x, g(x + 1, y)). \end{array} \right.$$

Иненадващо, тази рекурсивна схема определя функция, която е почти константа (вижте *Задача* 1.3 по-надолу).

Оказва се, че подходящото обобщение на простата схема на примитивна рекурсия е рекурсия само *по един* от аргументите на определяемата функция f. Ние ще изберем това да бъде последният аргумент.

Определение 1.6. Нека $g(x_1, \dots, x_n)$ и $h(x_1, \dots, x_n, y, z)$ са фиксирани частични функции, съответно на n и $n + 2$ аргумента. Казваме, че f се получава с *примитивна рекурсия* от g и h, ако за всички $\bar{x}, y$ са изпълнени равенствата:

$$\left| \begin{array}{l} f(x_1, \dots, x_n, 0) \simeq g(x_1, \dots, x_n) \\ f(x_1, \dots, x_n, y + 1) \simeq h(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y)). \end{array} \right. \tag{1.2}$$

За функцията f ще казваме още, че *е* примитивна рекурсия на g и h.

Равенствата 1.6 определят *общата схема* на примитивната рекурсия. Ясно е, че при $n = 0$ от нея получаваме простата схема, която въведохме по-горе.

Да се убедим най-напред, че *Определение* 1.6 наистина определя точно една функция.

Твърдение 1.1. Нека $g \in \mathcal{F}_n$, а $h \in \mathcal{F}_{n+2}$. Съществува единствена функция f, която е примитивна рекурсия на g и h.

Доказателство. Ще покажем, че за всяко $\bar{x} \in \mathbb{N}^n$ и всяко $y \in \mathbb{N}$, $f(\bar{x}, y)$ е *еднозначно определена*. Последното означава, че или $f(\bar{x}, y)$ няма стойност, или има стойност и тази стойност е единствена.

За целта да фиксираме $\bar{x} \in \mathbb{N}^n$. С индукция по y ще покажем, че $\forall y P(y)$, където P е следното свойство:

$$P(y) \quad \stackrel{\text{деф}}{\Longleftrightarrow} \quad f(\bar{x}, y) \ \text{ е еднозначно определена.}$$

Базовият случай $P(0)$ е ясен, защото тогава $f(\bar{x}, 0) \simeq g(\bar{x})$.
Сега да допуснем, че за някое y е в сила $P(y)$, т.е. $f(\bar{x}, y)$ е еднозначно определена. Но тогава същото ще е вярно и за $f(\bar{x}, y+1)$, тъй като в този случай

$$f(\bar{x}, y+1) \stackrel{(1.2)}{\simeq} h(\bar{x}, y, f(\bar{x}, y)).$$

Така показахме и $P(y+1)$, с което приключва проверката на $\forall y P(y)$, а оттук и доказателството на твърдението. □

С разсъждение, подобно на горното, лесно се убеждаваме, че операцията примитивна рекурсия запазва тоталността, т.е. приложена върху тотални функции, тя връща отново тотална функция.

Твърдение 1.2. Нека $g \in \mathcal{F}_n$ и $h \in \mathcal{F}_{n+2}$ са тотални функции. Тогава и тяхната примитивна рекурсия е тотална функция.

Доказателство. Да означим с f примитивната рекурсия на g и h. Сега трябва да видим, че за всяко $\bar{x} \in \mathbb{N}^n$ и всяко $y \in \mathbb{N}$, $!f(\bar{x}, y)$. Както в предишното доказателство, фиксираме $\bar{x} \in \mathbb{N}^n$ и с рутинна индукция по y показваме, че $\forall y Q(y)$, където Q се определя така:

$$Q(y) \quad \stackrel{\text{деф}}{\Longleftrightarrow} \quad !f(\bar{x}, y).$$

□

2.3 Примитивно рекурсивни функции

Определение 1.7. Казваме, че една функция е *примитивно рекурсивна (пр. р.)*, ако тя може да се получи от изходните примитивно рекурсивни функции чрез краен брой прилагания на операциите суперпозиция и примитивна рекурсия.

Да отбележим, че горната дефиниция всъщност е *индуктивна*. Тя може да бъде изказана и по следния начин:

1) Всяка от изходните функции $\mathcal{S}, \mathcal{O}$ и I^n_k е примитивно рекурсивна.

2) Ако f и $g_1, \ldots g_n$ са примитивно рекурсивни, то и тяхната суперпозиция $f(g_1, \ldots, g_n)$ е примитивно рекурсивна.

3) Ако f и g са примитивно рекурсивни, а h е получена с примитивна рекурсия от тях, то и h е примитивно рекурсивна.

Индуктивният характер на тази дефиниция означава, че всяко свойство, отнасящо се до примитивно рекурсивните функции, ще трябва да се доказва с индукция, следваща пунктовете на дефиницията. Такъв тип индукция се нарича *структурна индукция*. За илюстрация да докажем следващото твърдение.

Твърдение 1.3. Всяка примитивно рекурсивна функция е тотална.

Доказателство. Нека h е примитивно рекурсивна. Ако тя е получена по т. 1) от горната дефиниция, то h е някоя от изходните функции $\mathcal{S}, \mathcal{O}$ и I^n_k, които са тотални. Нека сега h е получена по т. 2) от дефиницията. Това означава, че $h = f(g_1, \ldots, g_n)$, като за f и $g_1, \ldots, g_n$ индуктивната хипотеза е вярна, т.е. те са тотални функции. Но тогава и h ще е такава, защото суперпозицията очевидно запазва тоталността.

По подобен начин разсъждаваме ако h е получена по последния пункт 3) от дефиницията, като се възползваме от току-що доказаното *Твърдение* 1.2. □

По-нататък ще се убедим, че макар и примитивна по име, с такава рекурсия могат да се дефинират супербързорастящи функции — примерно експоненти (двойни, тройни и всякакви). Всъщност всяка тотална изчислима функция в естествените числа, за която се досетите, със сигурност ще е примитивно рекурсивна. Както ще видим по-нататък в курса, ще се изискват доста знания, за да конструираме тотална изчислима функция, която да не е примитивно рекурсивна.

3. Частично рекурсивни функции

Сега се насочваме към последната изходна операция — *минимизация*, която участва в дефиницията на частично рекурсивните функции.

Определение 1.8. Нека $f : \mathbb{N}^{n+1} \multimap\to \mathbb{N}$ е произволна частична функция. Казваме, че n-местната функция g се получава с *минимизация* (или с *μ-операция*) от f, и пишем

$$g(x_1, \dots, x_n) \simeq \mu y[f(x_1, \dots, x_n, y) \simeq 0],$$

ако за g е изпълнено:

$$g(x_1, \dots, x_n) \simeq y \iff f(x_1, \dots, x_n, y) \simeq 0 \ \& \ \forall z_{z<y} \ f(x_1, \dots, x_n, z) > 0$$

за всички естествени $x_1, \dots, x_n$ и y.

Да обърнем внимание на следната особеност в горната дефиниция: ако $g(\bar{x}) \simeq y$, то y е не просто първото естествено число, за което $f(\bar{x}, y) \simeq 0$; за него трябва да е вярно още, че $f(\bar{x}, z) > 0$ за всяко $z < y$. С други думи, ако $g(\bar{x}) \simeq y$, то за всички $z < y$, $f(\bar{x}, z)$ *има стойност* и тя е различна от 0.

Сигурно се питате защо да не вземем минимизация, която просто връща първото естествено y, такова че $f(\bar{x}, y) \simeq 0$, иначе казано, защо да не вземем следната μ-операция:

$$\mu y[f(\bar{x}, y) \simeq 0] \overset{\text{деф}}{\simeq} min\{y \mid f(\bar{x}, y) \simeq 0\}.$$

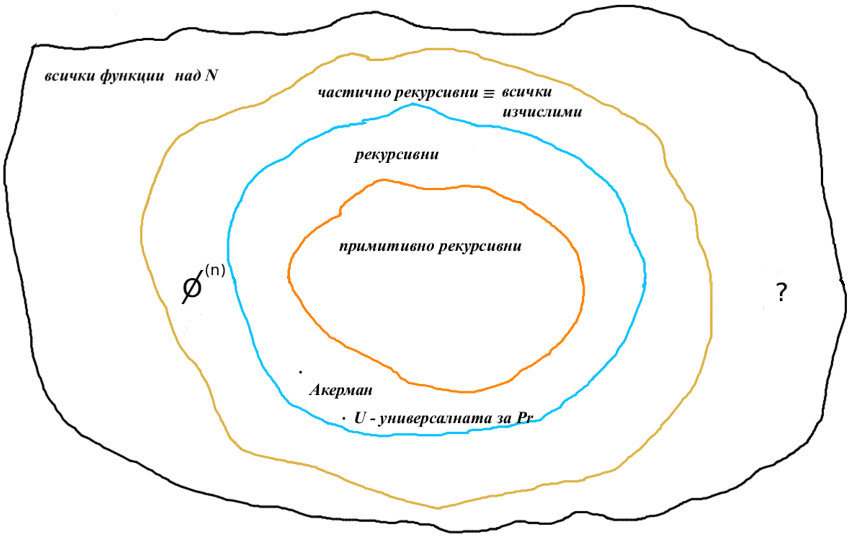
По-нататък в курса ще покажем, че при такава минимизация ще съществуват функции, които са изчислими (с някаква програма), докато тяхната минимизация вече не е изчислима с никаква програма. Това означава, че тази операция извежда извън класа на изчислимите функции, а това е последното нещо, което бихме искали да имаме за една базисна операция.

Определение 1.9. Казваме, че една функция е *частично рекурсивна (ч. р.)*, ако тя може да се получи от изходните примитивно рекурсивни функции чрез краен брой прилагания на операциите суперпозиция, примитивна рекурсия и минимизация.

Определение 1.10. Казваме, че една функция е *рекурсивна*, ако тя е частично рекурсивна и тотална.

Ясно е, че всички примитивно рекурсивни функции са и рекурсивни, защото те са:
– частично рекурсивни (в частност);
– тотални, съгласно *Твърдение* 1.3.

Ето как изглеждат на картинка класовете от функции, които въведохме, като в нея под *изчислима* функция засега ще разбираме функция, изчислима с *някаква* програма. В следващата глава ще дадем строга дефиниция на това понятие и ще докажем, че изчислимите функции съвпадат с частично рекурсивните.



В тази диаграма всички включвания са строги. Примери за функции, които са рекурсивни, но не са примитивно рекурсивни са, да кажем, функцията на Акерман, както и универсалната функция за всички примитивно рекурсивни функции (нали не очаквате в този момент да формулираме и докажем такова твърдение? $\ddot\smile$). Съвсем лесно за доказване е, обаче, че частично рекурсивните функции се включват строго в рекурсивните, просто защото те могат да са частични. Тривиални примери са празната функция $\emptyset^{(n)}$ или да кажем функцията f от *Пример* 1.1 (ще го докажем следващия час).

Разбира се, и най-външното включване е строго, което се вижда най-лесно с мощностни съображения. За целта първо съобразяваме, че всички частично рекурсивни функции са изброимо много, а после използваме, че множеството от всички функции над $\mathbb{N}$ е с мощността на континуума (както е добре известно).

Ето и едно директно доказателство на този факт, което се базира на *диагоналния метод на Кантор.*

Задача 1.2. Докажете, че съществуват функции, които не са частично рекурсивни (и следователно не могат да се пресметнат с никаква програма).

Решение. Ще конструираме едноместна функция, която не е частично рекурсивна.

Както вече отбелязахме, всички частично рекурсивни функции са изброимо много, а оттук и *едноместните* ч.р.ф. също ще са изброимо много.

Да ги подредим в редичка:

$$f_0, f_1, \ \dots \ f_n, \ \dots$$

Ще конструираме функция $d(x)$ — *диагонална функция*, такава че

$$d \neq f_n$$

за всяко n, и следователно d не може да е частично рекурсивна.

Условието $d \neq f_n$ означава, че $d(x) \not\simeq f_n(x)$ за поне едно x. Ние ще осигурим това различие за $x = n$, т.е. функцията d ще се различава от f_n в точката n.

За целта да вземем например

$$d(x) \simeq \begin{cases} \neg !, & \text{ако } !f_x(x) \\ 0, & \text{ако } \neg !f_x(x). \end{cases}$$

Да допуснем, че d е частично рекурсивна. Тогава $d = f_n$ за някое n и значи

$$d(x) \simeq f_n(x)$$

за всяко x. Но при $x = n$ имаме проблем, защото от една страна, би трябвало

$$d(n) \simeq f_n(n),$$

а от друга — функцията d избрахме точно с цел това да не се случва. □

Задача 1.3. Нека за g е изпълнено:

$$\left| \begin{array}{l} g(0, y) \simeq y \\ g(x+1, 0) \simeq g(x, 1) \\ g(x+1, y+1) \simeq g(x, g(x+1, y)). \end{array} \right. \tag{1.3}$$

Докажете, че g има следния явен вид:

$$g(x, y) = \begin{cases} y, & \text{ако } x = 0 \\ 1, & \text{иначе.} \end{cases}$$

Решение. Ясно е, че $g(0, y) = y$. Трябва да покажем, че

$$\forall x_{x \geq 1} \underbrace{\forall y \ g(x, y) = 1}_{P(x)}.$$

С индукция по $x \geq 1$ показваме, че $\forall x P(x)$, където

$$P(x) \iff \forall y \ g(x, y) = 1.$$

База $x = 1$, т.е. доказваме, че

$$\forall y \ \underbrace{g(1, y) = 1}_{Q(y)}.$$

Сега нека с Q означим свойството $Q(y) \overset{\text{деф}}{\iff} g(1, y) = 1$. Ще докажем, че $\forall y \ Q(y)$, като този път използваме индукция относно y. При $y = 0$ от (1.3) получаваме:

$$g(1, 0) \ = \ g(0, 1) \ = \ 1.$$

Да допуснем, че за някое y е вярно $Q(y)$. Тогава за $Q(y + 1)$ ще имаме, съгласно (1.3):

$$g(1, y + 1) \ = \ g(0, g(1, y)) \ \overset{\text{и.х. } Q(y)}{=} \ g(0, 1) \ = \ 1.$$

С това приключва проверката на $\forall y \ Q(y)$, или все едно — на твърдението $P(1)$. Сега да допуснем, че за някое $x \geq 1$ е изпълнено $P(x)$. Трябва да покажем, че и $P(x + 1)$ е вярно, т.е.

$$\forall y \ \underbrace{g(x + 1, y) = 1}_{R(y)}.$$

Трябва да покажем, че $\forall y \ R(y)$, където $R(y) \overset{\text{деф}}{\iff} g(x + 1, y) = 1$. Действаме отново с индукция относно y.

При $y = 0$ ще имаме

$$g(x + 1, 0) \ = \ g(x, 1) \ \overset{\text{и.х. } P(x)}{=} \ 1.$$

Сега да приемем, че $R(y)$ е вярно за някое y. Тогава за $y + 1$ ще имаме, съгласно (1.3):

$$g(x + 1, y + 1) \ = \ g(x, g(x + 1, y)) \ \overset{\text{и.х. } R(y)}{=} \ g(x, 1) \ \overset{\text{и.х. } P(x)}{=} \ 1.$$

□

4. Примитивна рекурсивност на някои функции

В този раздел ще докажем примитивната рекурсивност на една дълга редица от функции в естествените числа. Фактът, че те са примитивно рекурсивни, ще е важен за това, което следва.

Ще започнем с едно спомагателно твърдение, което нататък ще използваме систематично. Ще го формулираме за трите типа функции, които въведохме, макар че основно ще го използваме за примитивно рекурсивните.

Твърдение 1.4. Нека $f(x_1, \dots, x_k)$ е произволна примитивно рекурсивна/частично рекурсивна/рекурсивна функция. Нека още $i_1, \dots, i_k$ са някакви числа между 1 и n (допускаме и повтарящи се). Да дефинираме n-местната функция g по следния начин:

$$g(x_1, \dots, x_n) \simeq f(x_{i_1}, \dots, x_{i_k})$$

за всяко $x_1, \dots, x_n \in \mathbb{N}^n$. Тогава g също е примитивно рекурсивна/частично рекурсивна/рекурсивна.

Доказателство. Функцията g можем да препишем и така:

$$g(x_1, \dots, x_n) \stackrel{\text{деф}}{\simeq} f(x_{i_1}, \dots, x_{i_k}) \simeq f(I^n_{i_1}(x_1, \dots, x_n), \dots, I^n_{i_k}(x_1, \dots, x_n)),$$

откъдето се вижда, че $g = f(I^n_{i_1}, \dots, I^n_{i_k})$. Тъй като $I^n_{i_1}, \dots, I^n_{i_k}$ са изходни пр.р. функции, то ясно е, че ако f е примитивно (частично) рекурсивна, то и g ще е такава, а ако f е рекурсивна, то и g ще е рекурсивна, защото изходните функции са тотални. $\square$

Това твърдение ще използваме в ситуации като следните:

- $g(x_1, x_2) = f(x_1)$ — въвеждане на фиктивна променлива (тук $i_1 = 1$);
- $g(x_1, x_2) = f(x_2, x_1)$ — разместване на променливи (тук $i_1 = 2, i_2 = 1$);
- $g(x_1) = f(x_1, x_1)$ — удвояване на променлива (тук $i_1 = i_2 = 1$).

С C^n_a ще означаваме n-местната константна функция, която връща винаги a:

$$C^n_a(x_1, \dots, x_n) \stackrel{\text{деф}}{=} a$$

за всяка $(x_1, \dots, x_n) \in \mathbb{N}^n$. Всички константни функции са примитивно рекурсивни, както се вижда от задачата по-долу.

Задача 1.4. Докажете, че за всяко $a \in \mathbb{N}$ и $n \in \mathbb{N}^+$ константната функция C^n_a е примитивно рекурсивна.

Решение. За всяко $\bar{x} \in \mathbb{N}^n$ имаме $C^n_a(\bar{x}) = \underbrace{\mathcal{S}(\dots\mathcal{S}}_{a \text{ пъти}}(\mathcal{O}(I^n_1(\bar{x})))\dots)$ и значи C^n_a е следната композиция на изходни функции:

$$C^n_a = \underbrace{\mathcal{S} \circ \dots \circ \mathcal{S}}_{a \text{ пъти}} \circ \mathcal{O} \circ I^n_1.$$

$\square$

Предстои ни да видим как, тръгвайки от съвсем скромната функция "прибавяне на единица", можем да получим всички аритметични действия — събиране, изваждане, умножение, деление and much more $\ddot\smile$.

Твърдение 1.5. Следните функции са примитивно рекурсивни:

а) $f(x, y) = x + y$ (събиране).

Доказателство. Ще конструираме примитивно рекурсивна схема за f. Избираме си рекурсията да е по втория аргумент на f (тя е комутативна, тъй че няма значение кой от двата ще изберем). Базисният случай е ясен:

$$f(x, 0) = x + 0 = x.$$

Сега трябва да намерим връзка между $f(x, y)$ и $f(x, y + 1)$. В случая тя се вижда веднага:

$$f(x, y + 1) = x + (y + 1) = (x + y) + 1 = f(x, y) + 1 = \mathcal{S}(f(x, y)).$$

Получаваме общо

$$\left| \begin{array}{l} f(x, 0) = I^1_1(x) \\ f(x, y + 1) = h(x, y, f(x, y)) \end{array} \right.$$

за $h(x, y, z) = \mathcal{S}(z)$. Тази функция е примитивно рекурсивна, защото се получава от изходната $\mathcal{S}$ с добавяне на две фиктивни променливи (тук използваме доказаното по-горе *Твърдение* 1.4). Финално, f се получава с примитивна рекурсия от пр.р. функции I^1_1 и h, и следователно f е примитивно рекурсивна. $\square$

Въпрос: Как мислите, защо не използвахме по-краткото разсъждение, че събирането може да се представи като композиция на изходните функции $\mathcal{S}$ и I^2_1, което се вижда от следните равенства:

$$x + y = x + \underbrace{1 + \dots + 1}_{y \text{ пъти}} = \underbrace{\mathcal{S}(\dots \mathcal{S}}_{y \text{ пъти}}(x) \dots) = \underbrace{\mathcal{S}(\dots \mathcal{S}}_{y \text{ пъти}}(I^2_1(x, y)) \dots)?$$