Съдържание

Глава 1

Оператори

1.1 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. Като цяло, такива ще са функциите, които ще изучаваме в този курс. Това ще са функции, които се пресмятат – най-общо казано – с някаква програма. И тъй като програмите, както е известно, невинаги завършват, то и функциите, които те пресмятат, в общия случай трябва да са частични.

Ще пишем $f : \mathbb{N}^n \multimap \mathbb{N}$, 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

$$\mathcal{F}_n = \{f \mid f : \mathbb{N}^n \multimap \mathbb{N}\}.$$

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 е от множеството $\mathcal{F}_n = \{f \mid f : \mathbb{N}^n \multimap \mathbb{N}\}$. Когато

казваме *функция*, изобщо казано, ще имаме предвид частична функция. Ако е важно, че функцията е тотална, това ще бъде отбелязвано експлицитно, в случай че не се подразбира от контекста.

По-нататък n-торките $(x_1, \ldots, x_n)$ понякога ще съкращаваме до $\bar{x}$.

1.1.1 Условно равенство

Когато пишем знак за 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) не са дефинирани. За тази цел ще използваме нов символ $\simeq$, който ще наричаме *условно равенство*. Това равенство се дефинира по следния начин:

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 $\bar{x} \in Dom(f)$. Да разпишем по-подробно условието за равенство на две функции:

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

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

Нека $\alpha(x_1, \ldots, x_n)$ е израз, в който участват променливите $x_1, \ldots, x_n$, и нека сме избрали някаква част от тях — да кажем $x_1, \ldots, x_k$. Тогава с

$$\lambda x_1, \ldots, x_k \,.\, \alpha(x_1, \ldots, x_n)$$

ще означаваме функцията g на променливите $x_1, \ldots, x_k$,, която при фиксирани $x_{k+1}, \ldots, x_n$ се дефинира по следния начин:

$$g(x_1, \ldots, x_k) \overset{\text{деф}}{\simeq} \alpha(x_1, \ldots, x_n)$$

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

Примери: $\lambda x \,.\, x$ е функцията *идентитет*, $\lambda x \,.\, 0$ е едноместната константна функция, която има стойност 0, $\lambda x, y \,.\, x + y$ е функцията събиране, докато $\lambda x \,.\, x + y$ е линейната функция f, която при фиксирано y се дефинира като $f(x) = x + y$.

1.1.2 Релацията включване

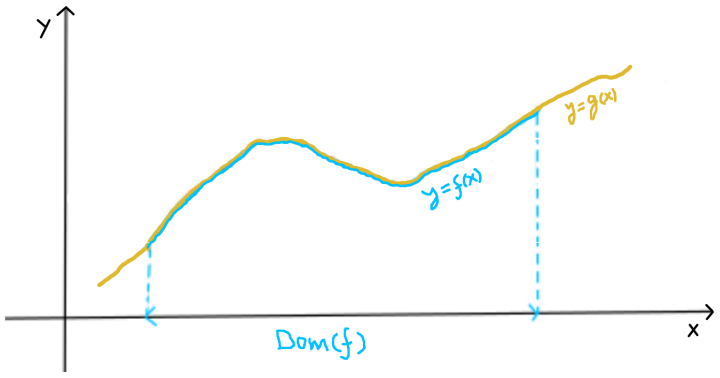
Тук ще въведем една релация между частични функции, която няма аналог при тоталните функции. Релацията е *включване* ($\subseteq$) и смисълът ѝ е, че ако $f \subseteq g$, то "g знае повече от f" или "g носи повече информация от f". Ето точното определение:

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

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

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 (т.е. има някаква стойност), там и втората е дефинирана и има същата стойност.

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



Разбира се, тъй като функциите f и g са дискретни, то графиките им би трябвало да са дискретни множества. Ние, обаче, за по-нагледно навсякъде в курса ще ги чертаем като непрекъснати.

От определението на релацията $\subseteq$ се вижда, че

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

което обяснява защо използваме същия символ $\subseteq$ както при включване между множества.

Да отбележим и още един очевиден факт, който ще използваме често:

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

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. $(x_1, \ldots, x_n)$ тя няма стойност, това ще записваме и така: $f(x_1, \ldots, x_n) \simeq \neg!$.

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 по следния начин:

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

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

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:

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

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 някои моменти ще ни спестява доста писане. Но първо да се убедим, че можем да направим това:

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

1) $f \subseteq g$;

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

Доказателство. 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 $\bar{x} \in Dom(f)$, и значи $f(\bar{x}) \simeq z$ за някое z. Сега от условието 1) получаваме, че и $g(\bar{x}) \simeq z$, откъдето $y = z$. И така, получихме, че за произволни $\bar{x}, y$:

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

което съгласно дефиницията на релацията $\subseteq$ означава, че $g \subseteq f$. □

Да напомним, че една бинарна релация е *частична наредба*, ако е рефлексивна, транзитивна и антисиметрична. Релацията включване, която току-що въведохме, е точно такава:

Твърдение 1.2. За всяко $n \geq 1$ релацията $\subseteq$ е частична наредба в $\mathcal{F}_n$.

Доказателство. Следва от еквивалентността

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

и от факта, че теоретико-множествената релация включване е частична наредба. □

Да отбележим, че горната релация наистина е *частична*, т.е. не всеки две функции от $\mathcal{F}_n$ са свързани чрез нея. Такива са например константните функции $f_0 = \lambda x\,.\,0$ и $f_1 = \lambda x\,.\,1$.

Не се заблуждавайте: вярно е, че $f_0(x) \leq f_1(x)$ за всяко x, обаче не е вярно, че $f_0 \subseteq f_1$. ☺

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 (празната) функция* на n аргумента. Ясно е, 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 $\mathcal{F}_n$.

1.1.3 Суперпозиция на функции

Ще завършим този раздел с дефиницията на операцията *суперпозиция* на частични функции.

Определение 1.4. 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:

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

Да обърнем внимание, че за да има стойност изразът $f(g_1(\bar{x}), \ldots, g_n(\bar{x}))$, трябва всеки от изразите $g_1(\bar{x}), \ldots, g_n(\bar{x})$ да има стойност, независимо от това, че някои от тези стойности могат да не се използват за крайния резултат $f(g_1(\bar{x}), \ldots, g_n(\bar{x}))$. Ето два примера:

Пример 1.2.

1) Нека $f = \lambda x, y \,.\, x$ (което означаваше: $f(x, y) = x$ за всяко x, y), нека още $g = \lambda x \,.\, x$, а $h = \emptyset^{(1)}$. Тогава

$$f(g, h)(x) \simeq f(g(x), h(x)) \simeq f(x, \neg!) \simeq \neg!,$$

въпреки че вторият аргумент на f е фиктивен и стойността на $f(x, y)$ не зависи от него.

2) Нека сега $f = \lambda x, y\ .\ x.y$, $g = \lambda x\,.\,0$, а $h = \emptyset^{(1)}$ отново. Тогава

$$f(g,h)(x) \simeq f(g(x),h(x)) \simeq 0.\neg! \simeq \neg!.$$

От друга страна, когато единият множител е 0, има някаква логика да искаме резултатът от умножението също да бъде 0, без въобще да се пресмята другият множител. Във функционалното програмиране подобно явление е известно като *отложено пресмятане (lazy evaluation)*, т.е. пресмятането на изразите се отлага дотогава, докогато е възможно, а на функционалните променливи се подават *имената* на изразите (а не техните *стойности*).

Ето един съвсем прост пример за рекурсивна програма, при която двата основни начина за подаване на параметрите — по стойност и по име — водят до различен резултат:

Пример 1.3. Нека R е следната програма:

$F(X,\ Y)\ =\ \texttt{if}\quad X\ =\ 0\quad \texttt{then}\ 0\quad \texttt{else}\quad F(X-1,\ F(X,\ Y)\ )$

При извикването по стойност $F(5,\ 10)$ ще имаме:

$\underline{F}(5,\ 10)\ \longrightarrow\ F(4,\ \underline{F}(5,\ 10)\ )\ \longrightarrow\ F(4,\ F(4,\ \underline{F}(5,\ 10)\ )\ )\ \longrightarrow\ \ldots$

което очевидно зацикля, докато при извикването по име ще стигнем до базовия случай $X = 0$ и съответно резултатът ще е 0:

$\underline{F}(5,\ 10)\ \longrightarrow\ \underline{F}(4,\ F(5,\ 10)\ )\ \longrightarrow\ \underline{F}(3,\ F(4,\ F(5,\ 10)\ )\ )\ \longrightarrow\ \ldots\ \longrightarrow$

$\underline{F}(0,\ F(1,\ldots\ F(4,\ F(5,\ 10)\ )\ \ldots\ )\ )\ \longrightarrow\ 0.$

Ясно е, че с *call by value* R ще пресметне функцията

$$f_{CV}(x,y) \simeq \begin{cases} 0, & \text{ако } x = 0 \\ \neg!, & \text{ако } x > 0, \end{cases}$$

докато с *call by name* — функцията

$$f_{CN}(x,y) = 0 \quad \text{за всяко } x, y \in \mathbb{N}.$$

Излезе, че $f_{CV} \subseteq f_{CN}$, или другояче казано — всичко, което се извежда от R по стойност, се извежда и по име. По-нататък ще видим, че това наблюдение е в сила за *всяка* рекурсивна програма R.

1.2 Компактни оператори

1.2.1 Оператори

Навсякъде в курса под *оператор* ще разбираме тотално изображение, което преработва функции във функции. Операторите ще означаваме с главни гръцки букви — $\Gamma, \Delta, ...$, евентуално с индекси. Първо ще разгледаме случая, когато тези оператори имат един аргумент. Тогава те ще са изображения от вида:

$$\Gamma\colon \mathcal{F}_k \longrightarrow \mathcal{F}_m$$

за някои естествени $k \geq 1$ и $m \geq 1$. Константите k и m определят *типа на оператора* Γ, който ще означаваме с $(k \ \rightarrow \ m)$.

Пример 1.4.1) Операторът *идентитет*: $\Gamma_{id}(f) = f$.

Този оператор е от тип $(k \ \rightarrow \ k)$.

2) *Константният оператор* $\Gamma_c(f) = f_0$, където f_0 е фиксирана функция. Този оператор може да е от произволен тип $(k \ \rightarrow \ m)$.

3) *Операторът за диагонализация* $\Gamma_d(f)(x) \simeq f(x,x)$,

който е от тип $(2 \ \rightarrow \ 1)$. (Ако се чудите откъде идва името на този оператор, представете си стойностите на f, разположени в безкрайна таблица, в която (i,j)-тият елемент е $f(i,j)$.)

4) *Операторът за сумиране* $\Gamma_{sum}(f)(x) \simeq \sum_{z=0}^{x} f(z)$ от тип $(1 \ \rightarrow \ 1)$.

5) $\Gamma(f)(x) \simeq$ `if` $x = 0$ `then` 1 `else` $x.f(x-1)$.

Този оператор е тясно свързан с рекурсивната дефиниция на функцията *факториел*. Очевидно е от тип $(1 \ \rightarrow \ 1)$.

1.2.2 Монотонни оператори

Монотонните изображения са типични за структури с частична наредба — това са тези изображения, които запазват тази наредба.

Определение 1.5. Нека $\Gamma : \mathcal{F}_k \longrightarrow \mathcal{F}_m$. Казваме, че Γ е *монотонен* оператор, ако за всяка двойка функции $f, g \in \mathcal{F}_k$ е изпълнено условието:

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

Монотонни са всички оператори, които дадохме като примери по-горе. Да проверим монотонността на един от тях:

Задача 1.1. Докажете, че е монотонен операторът за диагонализация

$$\Gamma(f)(x) \overset{\text{деф}}{\simeq} f(x,x).$$

Решение. Да вземем две функции f и g от $\mathcal{F}_2$, такива че $f \subseteq g$. За да видим, че и $\Gamma(f) \subseteq \Gamma(g)$, прилагаме определението на релацията $\subseteq$:

$$\Gamma(f) \subseteq \Gamma(g) \overset{\text{деф}}{\iff} \forall x \forall y\ (\Gamma(f)(x) \simeq y \implies \Gamma(g)(x) \simeq y).$$

Наистина, да вземем произволни естествени x и y и да приемем, че $\Gamma(f)(x) \simeq y$. Трябва да покажем, че и $\Gamma(g)(x) \simeq y$.

От $\Gamma(f)(x) \simeq y$ и определението на Γ получаваме, че $f(x,x) \simeq y$. Но $f \subseteq g$, следователно и $g(x,x) \simeq y$, т.е. $\Gamma(g)(x) \simeq y$. Понеже x и y бяха произволни, значи импликацията

$$\Gamma(f)(x) \simeq y \implies \Gamma(g)(x) \simeq y$$

ще е в сила за всички естествени x и y, с други думи, $\Gamma(f) \subseteq \Gamma(g)$. $\square$

Разбира се, не всички оператори са монотонни, обаче всички примери за немонотонни оператори са в някакъв смисъл неестествени. Сега ще дадем пример за един такъв оператор, а по-надолу ще обясним защо са неестествени операторите от този тип.

Пример 1.5. Да вземем отново константните функции $f_0 = \lambda x.0$ и $f_1 = \lambda x.1$ и да дефинираме оператор $\Gamma\colon \mathcal{F}_1 \longrightarrow \mathcal{F}_1$ по следния начин:

$$\Gamma(f) = \begin{cases} f_0, & \text{ако } f = \emptyset^{(1)} \\ f_1, & \text{иначе.} \end{cases}$$

Този оператор не е монотонен, защото ако вземем коя да е едноместна функция $f \neq \emptyset^{(1)}$, ще имаме $\emptyset^{(1)} \subseteq f$, докато

$$\Gamma(\emptyset^{(1)}) \overset{\text{деф}}{=} f_0 \not\subseteq f_1 \overset{\text{деф}}{=} \Gamma(f).$$

1.2.3 Компактни оператори

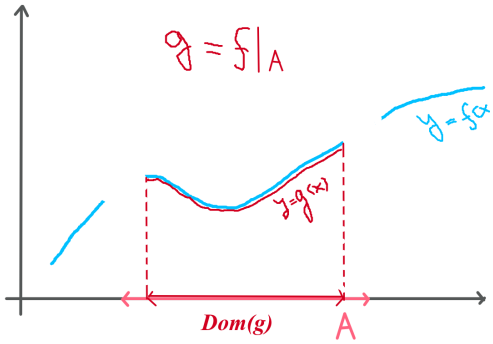
За дефиницията на компактност ще ни трябва понятието *крайна функция*. Една функция е *крайна*, ако е дефинирана в краен брой точки. Всяка крайна функция носи само *крайна информация* — информация за стойностите си в точките от дефиниционното си множество. За сравнение: една *тотална* едноместна функция f се характеризира с безкрайната редица от стойностите си $f(0), f(1), \ldots$.

Навсякъде в този курс с θ, евентуално с индекси, ще означаваме крайни функции. Да отбележим, че никъде недефинираната функция $\emptyset^{(1)}$ е крайна.

Нека f е произволна n-местна функция, а A е подмножество на $\mathbb{N}^n$. *Рестрикция на f до множеството A* ще наричаме функцията $g \in \mathcal{F}_n$, за която:

$$Dom(g) \;=\; Dom(f) \cap A \quad \& \quad g(\bar{x}) \simeq f(\bar{x}) \text{ за всяко } \bar{x} \in Dom(g).$$

Рестрикцията на f до множеството A ще означаваме с $f \restriction A$.



Определение 1.6. Операторът $\Gamma : \mathcal{F}_k \longrightarrow \mathcal{F}_m$ наричаме *компактен*, ако за всяка функция $f \in \mathcal{F}_k$, всяко $\bar{x} \in \mathbb{N}^m$ и всяко $y \in \mathbb{N}$ е в сила еквивалентността:

$$\Gamma(f)(\bar{x}) \simeq y \iff \exists \theta \; (\theta \subseteq f \;\&\; \theta \text{ е крайна } \;\&\; \Gamma(\theta)(\bar{x}) \simeq y). \tag{1.2}$$

Интуитивно, за един компактен оператор Γ е вярно, че *ако* $\Gamma(f)(\bar{x})$ има стойност, то тази стойност се получава като се използва само крайна информация от аргумента f — това е точно крайната функция θ от горното определение. Разбира се, точките, в които тази крайна θ е дефинирана, могат да зависят както от f, така и от $\bar{x}$.

Например, при оператора за диагонализация Γ_d имаме, че ако

$$\Gamma_d(f)(x) \overset{\text{деф}}{\simeq} f(x,x) \simeq y,$$

то резултатът y зависи от стойността на f само в една точка — точката (x,x). Следователно най-малката функция $\theta \subseteq f$, от която се определя резултатът $\Gamma_d(f)(x)$ е с дефиниционна област $\{(x,x)\}$, т.е.

$$\theta = f \restriction \{(x,x)\}.$$

За оператора

$$\Gamma_{sum}(f)(x) \simeq f(0) + \cdots + f(x)$$

имаме, че ако $\Gamma_{sum}(f)(x) \simeq y$, то y се определя от стойностите на f в точките $0, 1, \ldots, x$, и следователно $Dom(\theta)$ трябва *да включва* точките $0, 1, \ldots, x$, а най-малката θ с това свойство е тази, за която $Dom(\theta) = \{0, 1, \ldots, x\}$.

При оператора Γ, който се дефинира с условието $\Gamma(f)(x) \simeq f(f(x))$ е ясно, че ако $\Gamma(f)(x) \simeq y$, то $Dom(\theta)$ трябва да включва точките x и $f(x)$, като втората точка вече зависи и от f.

Следващото НДУ за компактност ще използваме както в теорията, така и в задачите.

Твърдение 1.3. Операторът $\Gamma : \mathcal{F}_k \longrightarrow \mathcal{F}_m$ е компактен тогава и само тогава, когато са изпълнени условията:

1) Γ е монотонен;

2) За всички $f \in \mathcal{F}_k$, $\bar{x} \in \mathbb{N}^m$ и $y \in \mathbb{N}$ е в сила импликацията:

$$\Gamma(f)(\bar{x}) \simeq y \implies \exists\theta(\theta \subseteq f \ \& \ \theta \text{ е крайна } \& \ \Gamma(\theta)(\bar{x}) \simeq y).$$

Забележка. Условието 2) всъщност е правата посока на условието за компактност (1.2). Операторите с това свойство понякога се наричат *крайни*.

Доказателство. Нека Γ е компактен. Поради горната забележка остава да видим само, че Γ е монотонен. За целта да вземем две функции f и g, такива че $f \subseteq g$, и да приемем, че за някои $\bar{x}, y$:

$$\Gamma(f)(\bar{x}) \simeq y.$$

Тогава от правата посока на (1.2) ще съществува крайна функция $\theta \subseteq f$, за която $\Gamma(\theta)(\bar{x}) \simeq y$. Имаме $\theta \subseteq f$ и $f \subseteq g$, и от транзитивността на релацията $\subseteq$ (*Твърдение* 1.2) получаваме $\theta \subseteq g$. Сега отново от условието за компактност на Γ, но прочетено наобратно, достигаме до $\Gamma(g)(\bar{x}) \simeq y$. Получихме общо, че

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

и понеже $\bar{x}$ и y бяха произволни, то наистина $\Gamma(f) \subseteq \Gamma(g)$.

Нека сега са в сила условията 1) и 2). Трябва да проверим само обратната посока на условието за компактност. Ако се вгледаме в него, виждаме, че то е някаква специална монотонност на Γ, отнасяща се само за случаите, когато функцията вляво на $\subseteq$ е крайна.
Наистина, нека дясната част на (1.2) е в сила, т.е. за някоя крайна $\theta \subseteq f$ е вярно, че

$$\Gamma(\theta)(\bar{x}) \simeq y.$$

Но операторът Γ е монотонен, и щом $\theta \subseteq f$, то и $\Gamma(\theta) \subseteq \Gamma(f)$. Оттук, имайки предвид, че $\Gamma(\theta)(\bar{x}) \simeq y$, веднага получаваме, че и $\Gamma(f)(\bar{x}) \simeq y$, което и трябваше да покажем. $\square$

Следствие 1.1. Всеки компактен оператор е монотонен.

Както не е трудно да се предположи, горното следствие не може да се обърне. Ето един контрапример:

Пример 1.6. Следният оператор Γ е монотонен, но не е компактен:

$$\Gamma(f) = \begin{cases} \emptyset^{(1)}, & \text{ако } f \text{ е крайна} \\ f, & \text{иначе.} \end{cases}$$

Доказателство. Монотонността на Γ се проверява непосредствено като се разгледат двете възможности за f:

1 сл. f е крайна. Тогава $\Gamma(f) \overset{\text{деф}}{=} \emptyset^{(1)} \subseteq \Gamma(g)$.

2 сл. f не е крайна. Понеже $f \subseteq g$, то тогава и g не е крайна и значи $\Gamma(f) \overset{\text{деф}}{=} f \subseteq g \overset{\text{деф}}{=} \Gamma(g)$.

За да се убедим, че Γ не е компактен, е достатъчно да вземем коя да е тотална функция f. За произволно естествено x имаме $\Gamma(f)(x) \overset{\text{деф}}{\simeq} f(x)$ и следователно $\Gamma(f)(x)$ има стойност. От друга страна, за всяка крайна θ, $\Gamma(\theta) \overset{\text{деф}}{=} \emptyset^{(1)}$ и следователно $\Gamma(\theta)(x)$ няма стойност, т.е. условието за компактност (1.2) няма как да е в сила. □

Да обърнем внимание, че горният оператор е доста неестествен от изчислителна гледна точка в следния смисъл. Да предположим, че разполагаме с програма за f. За да пресметнем $\Gamma(f)(x)$, трябва да проверим дали функцията f е крайна (т.е. дали е дефинирана в краен брой точки) — нещо, което интуитивно е ясно, че няма как да стане алгоритмично за краен брой стъпки. Съвсем друго е положението с оператора за диагонализация, например. За да пресметне $\Gamma_d(f)(x)$, на Γ_d му е нужна само една стойност на f — тази в точката (x, x).

Въобще, всички оператори, които дадохме като примери в началото на този раздел, са компактни. Да поверим компактността на някои от тях:

Задача 1.2. Докажете, че следващите оператори са компактни:

1) операторът за диагонализация $\Gamma_d(f)(x) \simeq f(x, x)$;

2) операторът композиция $\Gamma_{comp}(f) = f \circ f$;

3) операторът за сумиране $\Gamma_{sum}(f)(x) \simeq \sum_{z=0}^{x} f(z)$.

Решение. 1) Видяхме по-горе, че Γ_d е монотонен, и значи съгласно *Твърдение* 1.3 е достатъчно да покажем само правата посока на условието за компактност. Наистина, нека $\Gamma_d(f)(x) \simeq y$, т.е. $f(x, x) \simeq y$. Очевидно резултатът y зависи само от стойността на f в точката (x, x) и тогава е ясно коя ще е крайната подфункция θ на f, за която $\Gamma(\theta)(x) \simeq y$: просто полагаме θ да е рестрикцията на f до множеството $\{(x, x)\}$, т.е.

$\theta := f \restriction \{(x,x)\}$. Тогава от избора на θ ще имаме, че $\theta(x,x) \simeq f(x,x) \simeq y$ и следователно $\Gamma_d(\theta)(x) \overset{\text{деф}}{\simeq} \theta(x,x) \simeq y$.

Ще пропуснем проверката на монотонността на другите два оператора и ще се насочим директно към второто условие на *Твърдение* 1.3:

2) Нека $\Gamma_{comp}(f)(x) \simeq (f \circ f)(x) \simeq f(f(x)) \simeq y$. Очевидно резултатът y зависи от стойностите на f в двете точки x и $f(x)$ (обърнете внимание, че $f(x)$ трябва да има стойност, съгласно нашата дефиниция за суперпозиция). Ясно е, че можем да вземем крайната подфункция θ на f да бъде

$$\theta = f \restriction \{x,\ f(x)\}.$$

Тогава $\Gamma_{comp}(\theta)(x) \overset{\text{деф}}{\simeq} \theta(\theta(x)) \simeq \theta(f(x)) \simeq f(f(x)) \simeq y$.

3) Нека $\Gamma_{sum}(f)(x) \simeq \sum_{z=0}^{x} f(z) \simeq y$, т.е. $f(0)+\ldots+f(x) \simeq y$. В частност, $!f(0), \ldots, !f(x)$ и лесно се вижда, че ако положим

$$\theta = f \restriction \{0, \ldots, x\},$$

то $\Gamma_{sum}(\theta)(x) \simeq y$. □

Твърдението, с което ще завършим този раздел показва, че компактните оператори притежават свойство, аналогично на едно свойство на непрекъснатите функции в реалните числа: *ако две непрекъснати функции съвпадат върху всички рационални числа, то те съвпадат върху всички (реални) числа.* (Това ако приемем, че аналогът на рационално число е крайна функция, а на реално — произволна функция). Този факт идва да ни подскаже, че компактните оператори се държат в някакъв смисъл като непрекъснати. В следващия раздел ще видим, че това наистина е така — разбира се, след като дадем точната дефиниция за непрекъснатост на оператор.

Твърдение 1.4. Нека Γ_1 и Γ_2 са компактни оператори от един и същи тип $(k \to m)$, такива че за всяка крайна $\theta \in \mathcal{F}_k$ е изпълнено $\Gamma_1(\theta) = \Gamma_2(\theta)$. Тогава за всяка функция $f \in \mathcal{F}_k$ е вярно, че $\Gamma_1(f) = \Gamma_2(f)$.

Доказателство. За произволни $f \in \mathcal{F}_k$, $\bar{x} \in \mathbb{N}^m$ и $y \in \mathbb{N}$ имаме от определението за компактност:

$$\begin{aligned}
\Gamma_1(f)(\bar{x}) \simeq y &\iff \exists\theta(\theta \subseteq f \& \ \Gamma_1(\theta)(\bar{x}) \simeq y) \\
&\iff \exists\theta(\theta \subseteq f \ \& \ \Gamma_2(\theta)(\bar{x}) \simeq y) \\
&\iff \Gamma_2(f)(\bar{x}) \simeq y.
\end{aligned}$$

□

1.2.4 Оператори на много променливи

Теорията, която развихме до тук, се обобщава непосредствено и за оператори на много променливи. Но първо да ги дефинираме.

Оператор на произволен брой аргументи е изображение от вида

$$\Gamma\colon\ \mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k} \ \longrightarrow\ \mathcal{F}_n.$$

Типа на Γ ще означаваме с $(m_1, \ldots, m_k \ \to\ n)$.

Пример 1.7.

1) Операторът *композиция*: $\Gamma_{comp}(f,g)(\bar{x}) \simeq f(g(\bar{x}))$ от тип $(1, n \ \to\ n)$.

2) Операторът $\Gamma_{mult}(f,g)(\bar{x}) \simeq f(\bar{x}).g(\bar{x})$ от тип $(n, n \ \to\ n)$.

3) Операторът *суперпозиция* $\Gamma_{sup}(f, g_1, \ldots, g_n) = f(g_1, \ldots, g_n)$ от тип $(n, \underbrace{k, \ldots, k}_{n \text{ пъти}} \ \to\ k)$.

Определението за монотонност на оператор на много променливи е аналогично на определението за монотонност на оператор на една променлива:

Определение 1.7. Казваме, че $\Gamma\colon\ \mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k} \ \longrightarrow\ \mathcal{F}_n$ е *монотонен* оператор, ако за произволни $(f_1, ..., f_k)$ и $(g_1, ..., g_k)$ от $\mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k}$ е изпълнено условието:

$$f_1 \subseteq g_1 \ \&\ \ldots\ \&\ f_k \subseteq g_k \ \Longrightarrow\ \Gamma(f_1, \ldots, f_k) \subseteq \Gamma(g_1, \ldots, g_k).$$

Също директно се обобщава и определението за компактност:

Определение 1.8. Операторът $\Gamma\colon\ \mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k} \ \longrightarrow\ \mathcal{F}_n$ наричаме *компактен*, ако за произволни $(f_1, \ldots, f_k) \in \mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k}$, $\bar{x} \in \mathbb{N}^n$ и $y \in \mathbb{N}$ е в сила еквивалентността:

$$\begin{aligned}\Gamma(f_1, \ldots, f_k)(\bar{x}) \simeq y \ \Longleftrightarrow\ & \exists\theta_1 \ldots \exists\theta_k(\theta_1 \subseteq f_1\ \&\ \ldots\ \&\ \theta_k \subseteq f_k \quad \& \\ & \theta_1, \ldots, \theta_k \text{ са крайни } \&\ \Gamma(\theta_1, \ldots, \theta_k)(\bar{x}) \simeq y).\end{aligned}$$

Твърдение 1.3 продължава да е вярно и за компактни оператори, които имат повече от един аргумент. В новото доказателство няма нищо по-различно от старото, затова го пропускаме.

1.3 Непрекъснати оператори

В този раздел ще се запознаем с важното свойство *непрекъснатост* на оператор. За да го въведем, ще тръгнем от едно понятие, което ви е добре известно от лекциите по ДИС — понятието *точна горна граница* (*супремум*). Разликата е, че тук няма да разглеждаме супремум на множество от реални числа, а на множество от *функции*.

1.3.1 Точна горна граница

Определение 1.9. Нека $\mathcal{F} \subseteq \mathcal{F}_k$ е множество от k-местни функции. Казваме, че функцията g е *горна граница* (или *мажоранта*) на $\mathcal{F}$, ако за всяка функция $f \in \mathcal{F}$ е вярно, че

$$f \subseteq g.$$

Определение 1.10. Казваме, че g е *точна горна граница* на множеството $\mathcal{F}$, ако:

1) g е горна граница на $\mathcal{F}$;

2) за всяка горна граница h на $\mathcal{F}$ е вярно, че $g \subseteq h$.

По-нататък "точна горна граница" понякога ще съкращаваме до "т.г.г.". Лесно се съобразява, че ако съществува, точната горна граница е единствена: наистина, ако f и g са точни горни граници на множеството $\mathcal{F}$, то те са и негови горни граници, откъдето съгласно условие 2) ще имаме, че $f \subseteq g$ и $g \subseteq f$, което значи $f = g$. Точната горна граница на $\mathcal{F}$ ще означаваме с $\bigcup \mathcal{F}$.

Не всяко множество от функции притежава горна граница. Най-простият пример: да вземем две различни тотални функции f и g. Тогава множеството $\mathcal{F} = \{f, g\}$ няма горна граница: наистина, ако допуснем, че h е такава, то $f \subseteq h$ и $g \subseteq h$. Но f и g са тотални, следователно $f = h$ и $g = h$, т.е. $f = g$ — противоречие.

По-нататък ще се интересуваме основно от мажоранти на изброими редици от функции $f_0, f_1, \ldots$ Тези редици ще записваме по обичайния начин:

$$\{f_n\}_{n=0}^{\infty} \quad \text{или само} \quad \{f_n\}_n.$$

Ако g е (точна) горна граница на множеството $\mathcal{F} = \{f_0, f_1, \ldots\}$ от членовете на една такава редица, то g ще наричаме (точна) горна граница на *редицата* $\{f_n\}_n$.

Оказва се, че всяка *монотонно растяща* редица $f_0 \subseteq f_1 \ldots$ има точна горна граница, като при това тя се определя по един съвсем естествен начин. Да се убедим:

Твърдение 1.5. Всяка монотонно растяща редица $f_0 \subseteq f_1 \subseteq \ldots$ от функции в $\mathcal{F}_k$ притежава точна горна граница g, която се дефинира с еквивалентността

$$g(x_1, \ldots, x_k) \simeq y \iff \exists n \; f_n(x_1, \ldots, x_k) \simeq y$$

за всички естествени $x_1, \ldots, x_k, y$.

Забележка. От горната еквивалентност се вижда, че за графиката на g е изпълнено:

$$(\bar{x}, y) \in G_g \iff g(\bar{x}) \simeq y \overset{\text{деф}}{\iff} \exists n \; f_n(\bar{x}) \simeq y \iff \exists n \; (\bar{x}, y) \in G_{f_n}$$

за произволни $\bar{x}, y$. Следователно

$$G_g = \bigcup_{n=0}^{\infty} G_{f_n}.$$

Това, че графиката на g се явява обединение на графиките на функциите от редицата $\{f_n\}_n$, обяснява означението $\bigcup \mathcal{F}$ за точна горна граница на $\mathcal{F}$.

Доказателство. Най-напред да се убедим, че функцията g е еднозначна. Наистина, нека за някои $\bar{x}, y$ и z е изпълнено

$$g(\bar{x}) \simeq y \quad \text{и} \quad g(\bar{x}) \simeq z.$$

Тогава трябва да съществуват индекси l и m, за които

$$f_l(\bar{x}) \simeq y \quad \text{и} \quad f_m(\bar{x}) \simeq z.$$

Без ограничение на общността можем да считаме, че $l \leq m$. Тогава $f_l \subseteq f_m$ и щом $f_l(\bar{x}) \simeq y$, то и $f_m(\bar{x}) \simeq y$. Но ние имахме $f_m(\bar{x}) \simeq z$, следователно $y = z$.

Сега да видим защо g е точната горна граница на редицата $\{f_n\}_n$. Погоре забелязахме, че $G_g = \bigcup_n G_{f_n}$, откъдето за всяко n ще имаме $G_{f_n} \subseteq G_g$, или все едно, $f_n \subseteq g$, което означава, че g е горна граница на редицата $\{f_n\}_n$.

За да да убедим, че g е най-малката горна граница, да вземем друга горна граница на редицата — да кажем, h. Трябва да покажем, че $g \subseteq h$. За целта, за произволни $\bar{x}$ и y приемаме, че $g(\bar{x}) \simeq y$. От определението на g имаме, че тогава за някое n трябва да е изпълнено $f_n(\bar{x}) \simeq y$. Но h е горна граница на редицата $\{f_k\}_k$, а f_n е член на тази редица,

следователно $f_n \subseteq h$, откъдето в частност $h(\bar{x}) \simeq y$. Получихме, че за произволните $\bar{x}$ и y е в сила импликацията

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

което по дефиниция означава, че $g \subseteq h$. Следователно g е точната горна граница на редицата $\{f_n\}_n$. □

Ако съществува, точната горна граница на редицата $\{f_n\}_n$ ще означаваме с $\bigcup\limits_n f_n$ или само с $\bigcup f_n$.

В някои учебници, по аналогия с означението за *супремум (sup)* в анализа, точната горна граница на редицата $\{f_n\}_n$ се отбелязва с $\underset{n}{lub}\ f_n$ (от ***l**east **u**pper **b**ound*).

Да напишем още веднъж условието, задаващо точната горна граница $\bigcup f_n$ на редица $\{f_n\}_n$, което изведохме по-горе и което ще използваме многократно по-нататък:

$$(\bigcup f_n)(\bar{x}) \simeq y \iff \exists n\ f_n(\bar{x}) \simeq y. \tag{1.3}$$

1.3.2 Непрекъснати оператори

Сега ще въведем важното свойство *непрекъснатост* на оператор, което по-нататък в курса ще обобщим за произволни структури с частична наредба.

Определение 1.11. Операторът $\Gamma\colon \mathcal{F}_k \longrightarrow \mathcal{F}_m$ наричаме *непрекъснат*, ако за всяка монотонно растяща редица $f_0 \subseteq f_1 \subseteq \ldots$ от функции в $\mathcal{F}_k$ е изпълнено:

$$\Gamma(\bigcup_n f_n) \ = \ \bigcup_n \Gamma(f_n).$$

Забележка. В горното равенство имаме предвид, че точната горна граница $\bigcup\limits_n \Gamma(f_n)$ на редицата $\{\Gamma(f_n)\}_n$ съществува и е равна на $\Gamma(\bigcup\limits_n f_n)$. Малко по-надолу ще видим, че ако Γ е непрекъснат, то той е и монотонен, и следователно редицата $\{\Gamma(f_n)\}_n$ е монотонно растяща.

Твърдение 1.6. Всеки непрекъснат оператор е монотонен.

Доказателство. Нека $\Gamma\colon \mathcal{F}_k \longrightarrow \mathcal{F}_m$ е непрекъснат оператор и нека $f, g \in \mathcal{F}_k$ са такива, че $f \subseteq g$. Трябва да покажем, че и $\Gamma(f) \subseteq \Gamma(g)$. Да разгледаме монотонно растящата редица

$$f \subseteq g \subseteq g \subseteq \ldots.$$

Ясно е, че тази редица клони към g, т.е. нейната точна горна граница е g. Да приложим определението за непрекъснатост на Γ към тази редица. Ще получим, че $\Gamma(g)$ е точна горна граница на редицата

$$\Gamma(f), \Gamma(g), \Gamma(g), \ldots,$$

която има точно два различни елемента — $\Gamma(f)$ и $\Gamma(g)$. Но това означава, че $\Gamma(f) \subseteq \Gamma(g)$, което и трябваше да проверим. □

Всъщност от монотонността на един оператор следва едната половина от условието за непрекъснатост.

Твърдение 1.7. Нека $\Gamma\colon \mathcal{F}_k \longrightarrow \mathcal{F}_m$ е монотонен оператор, а редицата $f_0 \subseteq f_1 \subseteq \ldots$ е монотонно растяща в $\mathcal{F}_k$. Тогава

$$\bigcup_n \Gamma(f_n) \subseteq \Gamma(\bigcup_n f_n).$$

Доказателство. Като приложим монотонния оператор Γ към всяка от функциите от редицата $f_0 \subseteq f_1 \subseteq \ldots$, ще получим отново монотонно растяща редица

$$\Gamma(f_0) \subseteq \Gamma(f_1) \subseteq \ldots.$$

Тогава тази редица ще има точна горна граница $\bigcup \Gamma(f_n)$, съгласно *Твърдение* 1.5. От друга страна, тъй като $f_n \subseteq \bigcup f_n$, то отново от монотонността на Γ ще имаме

$$\Gamma(f_n) \subseteq \Gamma(\bigcup f_n).$$

Тъй като последното включване е в сила за всяко n, това означава, че функцията $\Gamma(\bigcup f_n)$ мажорира всеки член на редицата $\{\Gamma(f_n)\}_n$. Но в такъв случай $\Gamma(\bigcup f_n)$ ще мажорира и точната ѝ горна граница $\bigcup \Gamma(f_n)$, с други думи, ще имаме $\bigcup \Gamma(f_n) \subseteq \Gamma(\bigcup f_n)$. □

Дали има оператори Γ и монотонно растящи редици $\{f_n\}_n$, за които горното равенство е строго? Би трябвало, защото иначе ще излезе, че понятията монотонност и компактност съвпадат. Опитайте се сами да намерите примери:

Задача. Дайте пример за монотонен оператор $\Gamma\colon \mathcal{F}_1 \longrightarrow \mathcal{F}_1$ и редица от едноместни функции $f_0 \subseteq f_1 \subseteq \ldots$, такива че

$$\bigcup_n \Gamma(f_n) \subset \Gamma(\bigcup_n f_n).$$

Ето и още една лесна задача, която да се пробвате да решите самостоятелно:

Задача. Нека множеството $\mathcal{F} \subseteq \mathcal{F}_k$ има горна граница. Вярно ли е, че в такъв случай $\mathcal{F}$ има и точна горна граница? Обосновете се.

1.3.3 Еквивалентност между непрекъснатост и компактност

В оставащата част на този раздел целта ни ще бъде да покажем, че за операторите, които разглеждаме (преработващи аритметични функции), понятията непрекъснатост и компактност съвпадат. Едната половина на това твърдение можем да съобразим още сега.

Твърдение 1.8. Всеки непрекъснат оператор е компактен.

Доказателство. Най-напред да съобразим, че всяка функция $f \in \mathcal{F}_k$ можем да представим като точна горна граница на редица от *крайни* функции. Тези функции се явяват *крайни апроксимации* (приближения) на функцията f.
Да фиксираме $f \in \mathcal{F}_k$ и да означим с f_n рестрикцията на f до крайното множество $\{0, \ldots, n\}^k$, което означавахме така:

$$f_n = f \restriction \{0, \ldots, n\}^k.$$

Ясно е, че всяка f_n е крайна подфункция на f. Ще покажем, че редицата $f_0, f_1, \ldots$ е монотонно растяща и нейната точна горна граница е f.

Да фиксираме n и да се убедим, че $f_n \subseteq f_{n+1}$. Наистина, нека $f_n(x_1, \ldots, x_k) \simeq y$ за произволни $\bar{x}$ и y. Тогава и $f(x_1, \ldots, x_k) \simeq y$, и освен това $(x_1, \ldots, x_k) \in \{0, \ldots, n\}^k$. От последното ще имаме, че $(x_1, \ldots, x_k)$ ще е и в хиперкубчето $\{0, \ldots, n+1\}^k$, което заедно с $f(x_1, \ldots, x_k) \simeq y$ ни дава търсеното $f_{n+1}(x_1, \ldots, x_k) \simeq y$.
Интуитивно е съвсем ясно, че т.г.г. на редицата $f_0 \subseteq f_1 \subseteq \ldots$ трябва да е f, но да го докажем все пак. Наистина, по дефиниция $f_n \subseteq f$ за всяко n, което означава, че f е горна граница на редицата $\{f_n\}_n$.

Нека g е друга горна граница на тази редица. Трябва да покажем, че $f \subseteq g$. За целта да приемем, че за произволни $x_1, \ldots, x_k, y$ е изпълнено $f(x_1, \ldots, x_k) \simeq y$. Нека $n = max\{x_1, \ldots, x_k\}$. Тогава очевидно $(x_1, \ldots, x_k) \in \{0, \ldots, n\}^k$, откъдето $(x_1, \ldots, x_k) \in Dom(f_n)$ и значи $f_n(\bar{x}) \simeq y$. Но $f_n \subseteq g$ и следователно $g(\bar{x}) \simeq y$. Понеже $\bar{x}$ и y бяха произволни, това означава, че $f \subseteq g$. Така доказахме, че т.г.г. на редицата $f_0 \subseteq f_1 \subseteq \ldots$ от крайните приближения на f е самата f.

Сега се насочваме към доказателството на компактността на оператора $\Gamma \colon \mathcal{F}_k \longrightarrow \mathcal{F}_m$. За целта да вземем произволна функция $f \in \mathcal{F}_k$, както и произволни $\bar{x} \in \mathbb{N}^m$ и y. Понеже Γ е непрекъснат, за монотонно растящата редица $f_0 \subseteq f_1 \subseteq \ldots$ от крайните приближения на f ще имаме

$$\Gamma(\bigcup f_n) = \bigcup \Gamma(f_n).$$

Но $\bigcup f_n$ е точно f, както вече се убедихме. С други думи, имаме, че $\Gamma(f)$ е точната горна граница на редицата $\{\Gamma(f_n)\}_n$, която е монотонно растяща. Тогава за избраните по-горе $\bar{x}$ и y ще имаме, съгласно (1.3), че

$$\Gamma(f)(\bar{x}) \simeq y \iff \exists n\ \Gamma(f_n)(\bar{x}) \simeq y.$$

Тази еквивалентност можем да препишем още така:

$$\Gamma(f)(\bar{x}) \simeq y \iff \exists n\ (f_n \subseteq f\ \&\ f_n \text{ е крайна } \&\ \Gamma(f_n)(\bar{x}) \simeq y).$$

Оттук, в частност, получаваме импликацията

$$\Gamma(f)(\bar{x}) \simeq y \implies \exists \theta(\theta \subseteq f\ \&\ \theta \text{ е крайна } \&\ \Gamma(\theta)(\bar{x}) \simeq y).$$

И тъй като $f, \bar{x}$ и y бяха произволни, така всъщност показахме правата посока на условието за компактност (1.2). Това, заедно с монотонността на Γ и *Твърдение* 1.3 ни дават общо, че Γ е компактен. □

Да обърнем внимание, че горната апроксимация на f с *крайни* функции беше възможна само защото множеството $\mathbb{N}^k$ е изброимо. Функциите над реалните числа, например, не могат да се апроксимират с изброими редици от крайни функции.

За да докажем обратното на *Твърдение* 1.8 ще ни е нужна следната спомагателна лема.

Лема 1.1. Нека $f_0 \subseteq f_1 \subseteq \ldots$ е монотонно растяща редица от функции в $\mathcal{F}_k$ и нека за крайната функция θ е изпълнено

$$\theta \subseteq \bigcup_n f_n.$$

Тогава $\theta \subseteq f_n$ за някое n.

Доказателство. Нека $Dom(\theta) = \{\bar{x}^1, \ldots \bar{x}^l\}$. Можем да предполагаме, че $l \geq 1$, защото ако $l = 0$, т.е. $\theta = \emptyset^{(k)}$, то със сигурност $\theta \subseteq f_0$.

Да фиксираме $1 \leq i \leq l$ и нека

$$\theta(\bar{x}^i) \simeq y_i.$$

Понеже $\theta \subseteq \bigcup f_n$, значи и $(\bigcup f_n)(\bar{x}^i) \simeq y_i$. От последното, като използваме дефиницията за т.г.г. (1.3) получаваме, че съществува n_i, за което

$$f_{n_i}(\bar{x}^i) \simeq y_i.$$

Нека $n = max\{n_1, \ldots, n_l\}$. Тогава очевидно $n_i \leq n$ и следователно $f_{n_i} \subseteq f_n$. Сега от $f_{n_i}(\bar{x}^i) \simeq y_i$ ще имаме, че и

$$f_n(\bar{x}^i) \simeq y_i.$$

Финално, получихме, че за всяко $\bar{x}^i \in Dom(\theta)$ е изпълнено $\theta(\bar{x}^i) = f_n(\bar{x}^i)$, което означава, че $\theta \subseteq f_n$. □

Вече може да се заемем с обратната посока на *Твърдение* 1.8.

Твърдение 1.9. Всеки компактен оператор е непрекъснат.

Доказателство. Нека $\Gamma : \mathcal{F}_k \longrightarrow \mathcal{F}_m$ е компактен. Да фиксираме монотонно растяща редица $f_0 \subseteq f_1 \subseteq \ldots$ в $\mathcal{F}_k$. Трябва да покажем, че $\Gamma(\bigcup f_n) = \bigcup \Gamma(f_n)$. Но това е равенство между две функции и значи е достатъчно да покажем двете включвания

$$\Gamma(\bigcup f_n) \supseteq \bigcup \Gamma(f_n) \qquad \text{и} \qquad \Gamma(\bigcup f_n) \subseteq \bigcup \Gamma(f_n).$$

От *Следствие* 1.1 имаме, че всеки компактен оператор е монотонен. Тогава първото включване $\Gamma(\bigcup f_n) \supseteq \bigcup \Gamma(f_n)$ ще следва от *Твърдение* 1.7.

Да се насочим към проверката на второто включване $\Gamma(\bigcup f_n) \subseteq \bigcup \Gamma(f_n)$. За целта да вземем произволни $\bar{x} \in \mathbb{N}^m$ и y и да приемем, че за лявата функция $\Gamma(\bigcup f_n)$ е изпълнено

$$\Gamma(\bigcup f_n)(\bar{x}) \simeq y.$$

Трябва да видим, че и $(\bigcup \Gamma(f_n))(\bar{x}) \simeq y$. Наистина, от $\Gamma(\bigcup f_n)(\bar{x}) \simeq y$ и компактността на Γ следва, че съществува крайна функция θ, такава че

$$\theta \subseteq \bigcup f_n \quad \& \quad \Gamma(\theta)(\bar{x}) \simeq y.$$

Понеже $\theta \subseteq \bigcup f_n$, съгласно горната *Лема* 1.1 ще съществува n, за което $\theta \subseteq f_n$. Тогава $\Gamma(\theta) \subseteq \Gamma(f_n)$, понеже Γ е и монотонен. От дефиницията на точна горна граница имаме $\Gamma(f_n) \subseteq \bigcup \Gamma(f_n)$, или общо

$$\Gamma(\theta) \subseteq \Gamma(f_n) \subseteq \bigcup \Gamma(f_n).$$

Следователно $\Gamma(\theta) \subseteq \bigcup \Gamma(f_n)$, и понеже $\Gamma(\theta)(\bar{x}) \simeq y$, то значи и

$$(\bigcup \Gamma(f_n))(\bar{x}) \simeq y,$$

което и трябваше да покажем. □

От последните две твърдения получаваме общо, че

Следствие 1.2. Един оператор е компактен точно тогава, когато е непрекъснат.

1.4 Неподвижни точки на оператори

1.4.1 Неподвижни и най-малки неподвижни точки

Функцията f наричаме *неподвижна точка* на оператора Γ, ако

$$\Gamma(f) = f.$$

Ясно е, че за да говорим за неподвижни точки на Γ, трябва броят на аргументите на f и на $\Gamma(f)$ да е един и същ, т.е. трябва Γ да е оператор от специалния тип $(k \to k)$ за някое k.

Определение 1.12. Казваме, че f е *най-малка неподвижна точка (н.м.н.т.)* на оператора Γ, ако:

1) f е неподвижна точка на Γ;

2) за всяка неподвижна точка g на Γ е вярно, че $f \subseteq g$.

Ако съществува, най-малката неподвижна точка на Γ е единствена. Наистина, ако Γ има две най-малки неподвижни точки f и g, то от второто условие на дефиницията ще имаме, че $f \subseteq g$ и $g \subseteq f$, и следователно $f = g$. Тази единствена най-малка неподвижна точка на Γ ще означаваме с f_Γ. Друго често срещано означение е $lfp(\Gamma)$ (от ***l**east **f**ixed **p**oint*).

Основната мотивация за интереса ни към неподвижните точки на операторите са рекурсивните програми. Ще разгледаме няколко съвсем прости примера, илюстриращи връзките между рекурсивните програми, операторите и техните неподвижни точки. Да започнем с буквария пример за рекурсивна програма — тази, която пресмята функцията $\lambda x.x!$.

Пример 1.8. Нека R е следната рекурсивна програма:

$R:\quad F(X) = \texttt{if}\ \ X = 0\ \ \texttt{then}\ 1\ \ \texttt{else}\ \ X.F(X-1)$

На тялото на R можем да съпоставим следния оператор Γ $(= \Gamma_R)$:

$$\Gamma(f)(x) \simeq \begin{cases} 1, & \text{ако } x = 0 \\ x.f(x-1), & \text{иначе.} \end{cases}$$

Ясно е, че функцията f, която R пресмята, удовлетворява условието

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

или все едно

$$f(x) = \Gamma(f)(x).$$

Тъй като горното равенство е в сила за всяко $x \in \mathbb{N}$, то $f = \Gamma(f)$, т.е. f е неподвижна точка на оператора Γ.

Функцията *факториел* очевидно е неподвижна точка на Γ. Дали този оператор има и други неподвижни точки? Не. За да се убедим в това, вземаме произволна неподвижна точка f на Γ. Тогава за f е изпълнено:

$$f(x) \simeq \begin{cases} 1, & \text{ако } x = 0 \\ x.f(x-1), & \text{иначе.} \end{cases}$$

С индукция относно $x \in \mathbb{N}$ проверяваме, че $\forall x\ f(x) = x!$.

При $x = 0$ имаме $f(0) = 1 \stackrel{\text{деф}}{=} 0!$, а ако допуснем, че $f(x) = x!$ за някое $x \geq 0$, то за $x + 1$ получаваме последователно:

$$f(x+1) \simeq (x+1).f(x) = (x+1).x! = (x+1)!.$$

Да изследваме неподвижните точки на операторите, идващи от две пособени програми:

Пример 1.9. $R:\ F(X) = g(X)$, където g е фиксирана функция.

Очевидно програмата R (която всъщност не е рекурсивна, но няма пречка да се разглежда като такава), пресмята функцията g. Операторът, определен от нея, е константният оператор

$$\Gamma_c(f) \stackrel{\text{деф}}{=} g,$$

за всяка $f \in \mathcal{F}_1$. Ясно е, че и този оператор има една единствена н.т. и това е функцията g.

Пример 1.10. $R:\ F(X) = F(X)$

Тази програма пресмята никъде недефинираната функция $\emptyset^{(1)}$. Операторът, който тя определя, е операторът идентитет

$$\Gamma_{id}(f) \stackrel{\text{деф}}{=} f.$$

на който очевидно *всяка* функция е неподвижна точка, а най-малката неподвижна точка ще е точно $\emptyset^{(1)}$.

И един последен пример — за рекурсивна програма, на която съответства оператор с изброимо много неподвижни точки:

Пример 1.11. Нека R е програмата

$R: \quad F(\ X\ ) \quad = \quad if \quad X \ = \ 0 \quad then\ 0 \quad else\ F(\ X + 1\ )$

Да означим с Γ оператора, който се определя от R:

$$\Gamma(f)(x) \simeq \begin{cases} 0, & \text{ако } x \ = \ 0 \\ f(x+1), & \text{иначе.} \end{cases}$$

Ако f е неподвижна точка на Γ, то за нея е вярно, че

$$f(x) \simeq \begin{cases} 0, & \text{ако } x \ = \ 0 \\ f(x+1), & \text{иначе.} \end{cases}$$

Следователно $f(0) = 0$, а при всяко $x > 0$ би трябвало $f(x) \simeq f(x+1)$, което означава, че

$$f(1) \simeq f(2) \simeq f(3) \simeq \dots \ .$$

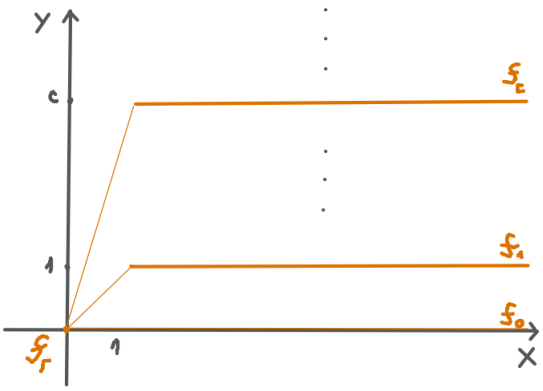
Следователно f трябва да има една и съща стойност при $x > 0$ или въобще да няма стойност. С други думи, f или е някоя от функциите f_c (за $c \in \mathbb{N}$), където f_c има вида

$$f_c(x) \simeq \begin{cases} 0, & \text{ако } x = 0 \\ c, & \text{ако } x > 0, \end{cases}$$

или f е $f_{\neg!}$, където

$$f_{\neg!}(x) \simeq \begin{cases} 0, & \text{ако } x = 0 \\ \neg!, & \text{ако } x > 0, \end{cases}$$

Ясно е, че най-малката неподвижна точка на Γ ще е функцията $f_{\neg!}$.



Видяхме, че разнообразието при неподвижните точки на операторите е голямо. Те могат да имат една, няколко или безброй много неподвижни точки. Обаче едно нещо се набиваше на очи — че всички те имат най-малка неподвижна точка.

Дали това винаги е така? Не. Ще завършим тази встъпителна част с още два примера — за оператор, който няма *най-малка* неподвижна точка (но има неподвижни точки) и за оператор, който въобще няма неподвижни точки. Особеното и при двата оператора е, че те, за разлика от вече разгледаните примери, "не идват" от рекурсивни програми.

Пример 1.12. Нека f_0 и f_1 са две различни тотални функции (бихме могли да си мислим отново за константните функции $\lambda x.0$ и $\lambda x.1$.) Да определим операторите Γ и Δ както следва:

$$\Gamma(f) = \begin{cases} f_0, & \text{ако } f = f_0 \\ f_1, & \text{ако } f \neq f_0, \end{cases}$$

$$\Delta(f) = \begin{cases} f_1, & \text{ако } f = f_0 \\ f_0, & \text{ако } f \neq f_0. \end{cases}$$

За да определим неподвижните точки на Γ, да приемем, че $\Gamma(f) = f$. Като разгледаме двете възможности за f — да е равна или да е различна от f_0, стигаме до извода, че $f = f_0$ или $f = f_1$. Следователно Γ има две неподвижни точки — f_0 и f_1, но няма най-малка неподвижна точка.

С подобни разсъждения се показва, че операторът Δ няма никакви неподвижни точки.

1.4.2 Теорема на Кнастер-Тарски

Примерите, които разгледахме дотук, показват, че на всяка рекурсивна програма R можем да съпоставим оператор Γ, като по смисъла на този оператор, функцията, която R пресмята, ще е една от неподвижните точки на Γ.

Да се опитаме да обобщим ситуацията: ако R е рекурсивна програма с n входни променливи, можем да си мислим, че тя изглежда най-общо така:

$R:\ F(\ X_1,\ \ldots,\ X_n\ )\ =\ \ldots\ F, X_1, \ldots, X_n\ \ldots$

Тогава операторът Γ от тип $(n\ \rightarrow\ n)$, който R определя, ще изглежда така:

$\Gamma(f)(x_1, \ldots, x_n) \simeq \ldots\ f, x_1, ..., x_n\ \ldots$

Ясно е, че ако R пресмята функцията f_R, то тази функция е неподвижна точка на Γ. Но Γ може да има и много други неподвижни точки, какъвто

беше операторът от *Пример* 1.11, да кажем. Възниква въпросът дали f_R има някакъв специален статут сред всички неподвижни точки на Γ?

В примерите, които разгледахме по-горе, наблюдавахме, че f_R винаги е *най-малката неподвижна точка* на съответния оператор Γ, с други думи, $f_R = f_\Gamma$. Всъщност една от най-важните цели на този курс е да покажем, че това е така за *всяка* рекурсивна програма R. За тази цел ще ни трябва вариант на теоремата за неподвижната точка (*fixed point theorem*), формулирана за подходяща структура с частична наредба. Сега ще докажем тази теорема в един частен случай на такава структура — множеството $\mathcal{F}_n$ с частичната наредба $\subseteq$. В следващата глава ще обобщим тази теорема за произволни структури с подходяща наредба, т. нар. области на Скот.

Нека Γ е оператор от тип $(k \to k)$, а f е произволна k-местна функция. За всяко естествено число n, с $\Gamma^n(f)$ ще означаваме функцията, която се дефинира с индукция по n както следва:

$$\begin{aligned}\Gamma^0(f) &= f\\ \Gamma^{n+1}(f) &= \Gamma(\Gamma^n(f)).\end{aligned}$$

Можем да запишем, че $\Gamma^n(f) = \underbrace{\Gamma(\ldots\Gamma}_{n \text{ пъти}}(f)\ldots)$, с други думи, $\Gamma^n(f)$ е функцията, която се получава след n-кратно прилагане на оператора Γ към f.

Следващата теорема играе централна роля при дефиниране на формална семантика на рекурсивните програми.

Теорема 1.1. Кнастер-Тарски Нека $\Gamma\colon \mathcal{F}_k \longrightarrow \mathcal{F}_k$ е непрекъснат оператор. Тогава Γ притежава най-малка неподвижна точка f_Γ и за нея е изпълнено:

$$f_\Gamma \;=\; \bigcup_n \Gamma^n(\emptyset^{(k)}).$$

Забележка. Тази теорема е известна още като *Теорема на Кнастер-Тарски-Клини*, защото Клини посочва начина, по който се *конструира* най-малката неподвижна точка f_Γ — като точна горна граница на редицата от функции $\{\Gamma^n(\emptyset^{(k)})\}_n$.

Доказателство. Да означим с f_n функцията $\Gamma^n(\emptyset^{(k)})$. Тогава очевидно

$$f_{n+1} = \Gamma^{n+1}(\emptyset^{(k)}) = \Gamma(\Gamma^n(\emptyset^{(k)})) = \Gamma(f_n),$$

и следователно редицата $\{f_n\}_n$ удовлетворява рекурентната схема

$$\left|\begin{array}{l} f_0 = \emptyset^{(k)}\\ f_{n+1} = \Gamma(f_n).\end{array}\right.$$

Първо да се убедим, че тази редица е монотонно растяща. С индукция относно n ще покажем, че за всяко естествено n

$$f_n \subseteq f_{n+1}.$$

База $n = 0$: по определение $f_0 = \emptyset^{(k)}$ и тогава очевидно $f_0 \subseteq f_1$.
Сега да приемем, че за някое $n \in \mathbb{N}$

$$f_n \subseteq f_{n+1}.$$

Операторът Γ е непрекъснат, и в частност — монотонен, съгласно *Твърдение* 1.6. Тогава от горното включване ще имаме

$$\Gamma(f_n) \subseteq \Gamma(f_{n+1}),$$

или $f_{n+1} \subseteq f_{n+2}$, с което индукцията е приключена.

Щом редицата $f_0, f_1, \ldots$ е монотонно растяща, според *Твърдение* 1.5 тя ще има точна горна граница — да я означим с g:

$$g = \bigcup_n f_n.$$

Нашата цел е да покажем, че g е най-малката неподвижна точка на Γ, т.е. $g = f_\Gamma$.

Това, че g е неподвижна точка, се вижда от следната верига от равенства:

$$\Gamma(g) \overset{\text{деф}}{=} \Gamma(\bigcup_n f_n) = \bigcup_n \Gamma(f_n) = \bigcup_{n=0}^{\infty} f_{n+1} = \bigcup_{n=1}^{\infty} f_n = \bigcup_{n=0}^{\infty} f_n \overset{\text{деф}}{=} g.$$

Нека сега h е друга неподвижна точка на Γ. Тъй като g е точна горна граница на $\{f_n\}_n$, за да покажем, че $g \subseteq h$, е достатъчно да видим, че h е горна граница на тази редица, с други думи, че $f_n \subseteq h$ за всяко $n \geq 0$. Това ще проверим отново с индукция относно n. За $n = 0$ имаме по определение

$$f_0 \overset{\text{деф}}{=} \emptyset^{(k)} \subseteq h.$$

Да предположим, че за някое n

$$f_n \subseteq h.$$

Прилагаме Γ към двете страни на неравенството и получаваме

$$\Gamma(f_n) \subseteq \Gamma(h) = h, \tag{1.4}$$

т.е. $f_{n+1} \subseteq h$. Сега вече можем да твърдим, че $f_n \subseteq h$ за всяко $n \geq 0$, с други думи, че h е мажоранта на редицата $\{f_n\}_n$. Следователно h мажорира и точната ѝ горна граница g, т.е. $g \subseteq h$. □

Всъщност непрекъснатостта на Γ е само достатъчно условие за съществуването на f_Γ. Вярно е, че всеки *монотонен* оператор също притежава най-малка неподвижна точка.

Както отбелязахме, теоремата на Кнастер-Тарски не само твърди, че всеки непрекъснат оператор има най-малка неподвижна точка, но ни дава и *начин* за нейното конструиране. Нека я приложим към оператора от *Пример* 1.8.

Задача 1.3. Като използвате теоремата на Кнастер-Тарски, намерете най-малката неподвижна точка на следния оператор:

$$\Gamma(f)(x) \simeq \begin{cases} 1, & \text{ако } x = 0 \\ x.f(x-1), & \text{иначе.} \end{cases}$$

Решение. Нашата цел ще бъде да намерим *явния вид* на всяка функция от дефинираната по-горе редица $f_0, f_1, \ldots$, чиято граница се явява f_Γ. Функциите $f_0, f_1, \ldots$ имат смисъл на последователни *приближения* (*апроксимации*) на f_Γ.

Да напомним, че редицата $\{f_n\}_n$ удовлетворява условията:

$$\left| \begin{array}{l} f_0 = \emptyset^{(1)} \\ f_{n+1} = \Gamma(f_n). \end{array} \right. \tag{1.5}$$

Започваме с първата апроксимация f_1 на f_Γ:

$$f_1(x) \overset{(1.5)}{\simeq} \Gamma(\emptyset^{(1)})(x) \overset{\text{деф } \Gamma}{\simeq} \begin{cases} 1, & \text{ако } x = 0 \\ x.\emptyset^{(1)}(x-1), & \text{ако } x > 0 \end{cases} \simeq \begin{cases} 1, & \text{ако } x = 0 \\ \neg!, & \text{ако } x > 0. \end{cases}$$

За следващата апроксимация f_2 ще имаме:

$$f_2(x) \overset{(1.5)}{\simeq} \Gamma(f_1)(x) \simeq \begin{cases} 1, & \text{ако } x = 0 \\ x.f_1(x-1), & \text{иначе} \end{cases} \overset{\text{деф } \Gamma}{\simeq} \begin{cases} 1, & \text{ако } x = 0 \\ 1.1, & \text{ако } x = 1 \\ \neg!, & \text{ако } x > 1. \end{cases}$$

Функцията f_2 можем да препишем още по следния начин:

$$f_2(x) \simeq \begin{cases} x!, & \text{ако } x < 2 \\ \neg!, & \text{иначе,} \end{cases}$$

което ни дава идея какъв би могъл да е общият вид на f_n:

$$f_n(x) \simeq \begin{cases} x!, & \text{ако } x < n \\ \neg!, & \text{иначе.} \end{cases}$$

Ще използваме индукция относно $n \in \mathbb{N}$, за да се убедим, че това е така.

На практика вече проверихме случаите $n = 0, 1$ и 2. Да предположим сега, че f_n има горния вид. Тогава за f_{n+1} ще имаме последователно:

$$f_{n+1}(x) \overset{(1.5)}{\simeq} \Gamma(f_n)(x) \overset{\text{деф } \Gamma}{\simeq} \begin{cases} 1, & \text{ако } x = 0 \\ x.f_n(x-1), & \text{ако } x > 0 \end{cases}$$

$$\overset{\text{и.х.}}{\simeq} \begin{cases} 1, & \text{ако } x = 0 \\ x.(x-1)!, & \text{ако } x > 0 \ \&\ x-1 < n \\ \neg!, & \text{ако } x - 1 \geq n \end{cases} \simeq \begin{cases} x!, & \text{ако } x < n+1 \\ \neg!, & \text{иначе,} \end{cases}$$

и значи индукционната хипотеза се потвърждава и за $n + 1$. Остана да съобразим, че границата на редицата $\{f_n\}_n$ е функцията $x!$, но това следва директно от дефиницията за точна горна граница (1.3).

□

1.4.3 Прednеподвижни точки на оператори

За някои приложения на теоремата на Кнастер-Тарски се оказва удобно да разполагаме с едно нейно уточнение. То се отнася за понятието прednеподвижна (или още квазинеподвижна точка, prefixed point) на даден оператор. По определение, f е *прednеподвижна точка* на оператора Γ, ако $\Gamma(f) \subseteq f$ и съответно f е *най-малка прednеподвижна точка* на Γ, ако тя е най-малката функция със свойството

$$\Gamma(f) \subseteq f.$$

Отново е ясно, че ако съществува, най-малката прednеподвижна точка е единствена.

На прednеподвижните точки на оператора Γ можем да гледаме като на решения на *неравенството*

$$\Gamma(X) \subseteq X,$$

докато неподвижните му точки са решения на *уравнението*

$$\Gamma(X) = X.$$

Тъй като неравенството е нестрого, очевидно всяко решение на уравнението е решение и на неравенството, с други думи, всяка неподвижна точка на Γ е и нейна прednеподвижна точка. Дали вярно и обратното? Невинаги — вижте например *Задача* 1.4 по-долу. Оказва се, обаче, че за *най-малката* прednеподвижна точка това е така, по-точно, най-малката прednеподвижна точка на всеки *непрекъснат* оператор е точно f_Γ.

Твърдение 1.10. Нека Γ е непрекъснат оператор от тип $(k \to k)$. Тогава f_Γ се явява и най-малка преднеподвижна точка на Γ.

Доказателство. Това твърдение следва съвсем непосредствено от доказателството на теоремата на Кнастер-Тарски. Две неща трябва да съобразим:

1) f_Γ е преднеподвижна точка, което както вече отбелязахме, е така.

2) Да вземем друга преднеподвижна точка h, т.е. функция, за която $\Gamma(h) \subseteq h$, и да покажем, че $f_\Gamma \subseteq h$. За целта следваме съвсем пунктуално доказателството на *Теорема* 1.1, като единствената разлика е в условието (1.4), където вместо $\Gamma(h) = h$ трябва да напишем $\Gamma(h) \subseteq h$. □

Всъщност горният факт е в сила и за операторите, които са само *монотонни*, и е известен като Лема на Тарски. Доказателството ѝ е съвсем кратко, да се убедим:

Твърдение 1.11. (Лема на Тарски) Нека Γ е монотонен оператор от тип $(k \to k)$, а f е неговата най-малка преднеподвижна точка. Тогава f е и най-малката неподвижна точка на Γ.

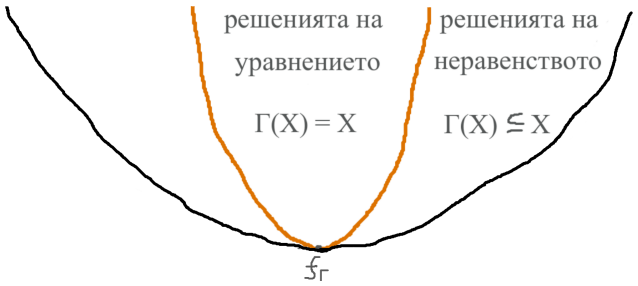
Доказателство. Имаме $\Gamma(f) \subseteq f$ и след почленно прилагане на Γ към двете страни на неравенството получаваме

$$\Gamma(\Gamma(f)) \subseteq \Gamma(f).$$

Излезе, че $\Gamma(f)$ също е преднеподвижна точка на Γ, и тъй като f е най-малката, то

$$f \subseteq \Gamma(f).$$

Но обратното включване $\Gamma(f) \subseteq f$ също е вярно, и значи общо $\Gamma(f) = f$, т.е. f е неподвижна точка на Γ. Дали е най-малката — да, защото ако g е друга н.т. на Γ, то тя ще е и негова преднеподвижна точка, което означава, че $f \subseteq g$. □



Задача 1.4. (Задача за ЕК) Опишете всички преднеподвижни точки на следния оператор Γ:

$$\Gamma(f)(x) \simeq \begin{cases} 0, & \text{ако } x = 0 \\ f(x+1), & \text{иначе.} \end{cases}$$

1.5 Индуктивен принцип на Скот

Ще го срещнете под най-различни имена:

Индуктивен принцип на Скот и Де Бакер

Правило/принцип за μ-индукция на Скот

Индукционно правило на Скот

На английски се среща още като

Scott's fixed-point induction principle

Индуктивният принцип на Скот е метод за доказване на свойства на най-малката неподвижна точка f_Γ на непрекъснат оператор Γ. Разбира се, някои свойства на f_Γ можем да доказваме с *обичайна индукция*. В *Пример* 1.8 така показахме, че единствената неподвижна точка на оператора

$$\Gamma(f)(x) \simeq \begin{cases} 1, & \text{ако } x \ = \ 0 \\ x.f(x-1), & \text{иначе.} \end{cases}$$

е $x!$, и значи f_Γ е $x!$.

В други случаи може да се наложи да използваме *пълна индукция*, както е във следващия пример.

Пример 1.13. Нека Γ е следният оператор:

$$\Gamma(f)(x) \simeq \begin{cases} 1, & \text{ако } x = 0 \\ (f(\frac{x}{2}))^2, & \text{ако } x > 0 \text{ е четно} \\ 2(f(\frac{x-1}{2}))^2, & \text{ако } x \text{ е нечетно.} \end{cases}$$

Докажете, че Γ има единствена неподвижна точка и това е функцията 2^x.

Решение. Ще покажем, че ако f е неподвижна точка на Γ, то $f(x) = 2^x$ за всяко x. Следователно 2^x ще е най-малката н.т. на Γ. Наистина $f = \Gamma(f)$ означава, че за всяко x:

$$f(x) \simeq \begin{cases} 1, & \text{ако } x = 0 \\ (f(\frac{x}{2}))^2, & \text{ако } x > 0 \text{ е четно} \\ 2(f(\frac{x-1}{2}))^2, & \text{ако } x \text{ е нечетно.} \end{cases}$$

Виждаме, че при $x > 0$ функцията f вика себе си в точки от вида $\frac{x}{2}$ или $\frac{x-1}{2}$, които са строго по-малки от x и значи трябва да разсъждаваме с *пълна индукция*, за да покажем, че

$$f(x) = 2^x \text{ за всяко } x \in \mathbb{N}.$$

При $x = 0$ имаме $f(0) \stackrel{\text{деф}}{=} 1 = 2^0$.

Сега да фиксираме някакво $x > 0$ и да предположим, че за всички $x' < x$ е вярно, че $f(x') = 2^{x'}$. Ако x е четно, за него ще имаме:

$$f(x) \simeq (f(\frac{x}{2}))^2 \stackrel{\text{и.х.}}{=} (2^{\frac{x}{2}})^2 = 2^x,$$

а ако x е нечетно, то отново от избора на f и индукционното предположение получаваме:

$$f(x) \simeq 2(f(\frac{x-1}{2}))^2 \stackrel{\text{и.х.}}{=} 2.(2^{\frac{x-1}{2}})^2 = 2^x.$$

□

Накрая, да разгледаме един оператор със следната дефиниция:

Пример 1.14.

$$\Gamma(f)(x) \simeq \begin{cases} \frac{x}{2}, & \text{ако } x \text{ е четно} \\ f(f(\frac{3x+1}{2})), & \text{ако } x \text{ е нечетно.} \end{cases}$$

Ясно е, че тук пълна индукция относно x няма да върви, най-малкото защото аргументът $\frac{3x+1}{2}$ на вътрешното f от израза $f(f(\frac{3x+1}{2}))$ е по-голям от x в $\Gamma(f)(x)$. В този раздел ще опишем метод за доказване на свойства на f_Γ, който не се базира на индукция относно x (както беше в горните примери), а е съвършено различен индуктивен принцип — т. нар. *принцип за μ-индукция на Скот.*

За да формулираме този принцип, най-напред ще дефинираме един специален вид свойства — *непрекъснатите свойства.*

1.5.1 Непрекъснати свойства

Нека P е свойство на функциите от $\mathcal{F}_k = \{f \mid f : \mathbb{N}^k \rightharpoonup \mathbb{N}\}$, или все едно, P е унарен предикат в $\mathcal{F}_k$.

Определение 1.13. Казваме, че свойството P е *непрекъснато*, ако за всяка монотонно растяща редица $f_0 \subseteq f_1 \subseteq \ldots$ в $\mathcal{F}_k$ е изпълнено условието:

$$\forall n P(f_n) \implies P(\bigcup_n f_n). \tag{1.6}$$

С други думи, P е непрекъснато, ако от това, че всеки член на растящата редица $\{f_n\}_n$ има свойството P следва, че и нейната точна горна граница $\bigcup f_n$ ще има свойството P.

Условието (1.6) се нарича още условие за *затвореност* на P. Ще си обясните защо, ако си представите P като множество — множеството M_P на тези функции, които имат свойството P.

Задача 1.5. Проверете дали е непрекъснато всяко от изброените по-долу свойства в $\mathcal{F}_1$:

1) $P_1(f) \iff \neg!f(1)$;

2) $P_2(f) \iff !f(0)$;

3) $P_3(f) \iff f$ е тотална;

4) $P_4(f) \iff f$ е крайна;

5) $P_5(f) \iff f$ е с безкраен домейн, но не е тотална.

Решение. Да си припомним дефиницията (1.3) за точна горна граница f на дадена монотонно растяща редица $f_0 \subseteq f_1 \subseteq \ldots$ за случая, когато тези функции са едноместни:

$$\forall x \forall y(\ f(x) \simeq y \iff \exists n\ f_n(x) \simeq y\ ). \tag{1.7}$$

1) Нека $f_0 \subseteq f_1 \subseteq \ldots$ е монотонно растяща редица и нека за всяко $n, P_1(f_n)$ е вярно, т.е. вярно е, че

$$\neg!f_0(1),\ \neg!f_1(1),\ \neg!f_2(1), \ldots\ .$$

Но тогава от дефиницията (1.7) на f е ясно, че $\neg!f(1)$, т.е. и $P_1(f)$ е в сила.

2) Нека сега всяка функция от редицата $\{f_n\}_n$ има свойството P_2, т.е.

$$\forall n\ !f_n(0).$$

Ако допуснем, че граничната функция $f = \bigcup f_n$ не е дефинирана за $x = 0$, то съгласно (1.7) би трябвало за никое n да не е дефинирано $f_n(0)$ — противоречие.

3) P_3 също е непрекъснато, защото ако редицата $f_0 \subseteq f_1 \subseteq \ldots$ се състои само от тотални функции, то тогава $f_0 = f_1 = \ldots$ и следователно $\bigcup f_n = f_0$ също ще е тотална.

4) Свойството P_4 вече не е непрекъснато: да вземем една едноместна функция f, която не е крайна. Нека f_n е рестрикцията на f до началния сегмент $\{0, \ldots, n\}$:

$$f_n = f \restriction \{0, \ldots, n\}.$$

Ясно е, че така получената редица $f_0 \subseteq f_1 \subseteq \ldots$ от крайни приближения на f е монотонно растяща и нейната т.г.г. е f. Очевидно за всяко n, $P_4(f_n)$ е в сила, докато за функцията f това не е така.

5) Адаптирайте доказателството от по-горе, като този път тръгнете от тотална функция f. $\square$

1.5.2 Индуктивен принцип на Скот

Вече сме готови да формулираме *индуктивния принцип на Скот* за доказване на свойства на най-малките неподвижни точки на непрекъснати оператори.

Твърдение 1.12.(Индуктивен принцип на Скот) Нека $\Gamma\colon \mathcal{F}_k \longrightarrow \mathcal{F}_k$ е непрекъснат оператор, а P е свойство в $\mathcal{F}_k$, за което са изпълнени условията:

1) $P(\emptyset^{(k)})$;

2) $P(f) \implies P(\Gamma(f))$ за всяка функция $f \in \mathcal{F}_k$;

3) свойството P е непрекъснато.

Тогава P е вярно за най-малката неподвижна точка f_Γ на оператора Γ.

Доказателство. От теоремата на Кнастер-Тарски знаем, че

$$f_\Gamma = \bigcup_n \Gamma^n(\emptyset^{(k)}).$$

Да положим отново $f_n = \Gamma^n(\emptyset^{(k)})$. В доказателството на теоремата видяхме, че редицата $\{f_n\}_n$ е монотонно растяща. С индукция по n ще покажем, че свойството P е вярно за всяка функция f_n от тази редица.

База $n = 0$: имаме $f_0 = \emptyset^{(k)}$ и $P(f_0)$ е точно условието 1).

Да приемем, че за някое n е вярно $P(f_n)$. Но тогава съгласно 2) ще е вярно и $P(\Gamma(f_n))$, т.е. ще е вярно $P(f_{n+1})$.

Получихме, че всеки член на монотонно растящата редица $f_0, f_1, \ldots$ има свойството P. Но P е непрекъснато, следователно то ще бъде в сила и за точната горна граница f_Γ на тази редица. □

Разбира се, далеч не всяко свойство на f_Γ може да се докаже с индуктивния принцип на Скот. Да вземем, да кажем, свойството

$$P(f) \iff \forall x\ f(x) = 2^x.$$

В *Пример* 1.13 видяхме, че то е изпълнено за н.м.н.т. f_Γ на съответния оператор Γ. Това свойство няма как да се докаже с принципа на Скот, защото още първото му условие $P(\emptyset^{(1)})$ пропада. В следващия раздел ще разгледаме няколко типа свойства, които принципно могат да се атакуват с този принцип (непрекъснати са и са верни за $\emptyset^{(1)}$).

1.5.3 Непрекъснатост на някои типове свойства

Типичното приложение на индуктивния принцип на Скот е за свойствата от тип частична коректност. Да си припомним определенията за коректност на програма.

Нека $\mathbf{P}$ е произволна програма с k входни променливи $x_1, \ldots, x_k$ и една изходна променлива y. Нека $I(x_1, \ldots, x_k)$ и $O(x_1, \ldots, x_k, y)$ са съответно някакви *входно* и *изходно условие* за програмата $\mathbf{P}$.

Свойство от тип частична коректност (относно даденото входно условие I и изходно условие O) ще наричаме условието $P_{p.c.}(f)$, което се дефинира по следния начин:

$$P_{p.c.}(f) \iff \forall \bar{x}\ (I(\bar{x})\ \&\ !f(\bar{x})) \implies O(\bar{x}, f(\bar{x}))).$$

Преразказано, свойството $P_{p.c.}(f)$ ни говори, че една програма е частично коректна, ако при всеки коректен вход $\bar{x}$, **ако** програмата завърши върху $\bar{x}$, то резултатът ще е коректен.

В частност, когато $I(\bar{x})$ е вярно за всяко $\bar{x} \in \mathbb{N}^k$ (т.е. нямаме изискване за входните данни), горното условие за частична коректност добива вида:

$$P_{p.c.}(f) \iff \forall \bar{x}\ (!f(\bar{x}) \implies O(\bar{x}, f(\bar{x}))).$$

При свойствата от тип *тотална коректност* искаме при всеки коректен вход програмата задължително да завършва и резултатът отново да е коректен, разбира се. Или формално:

$$P_{t.c.}(f) \iff \forall \bar{x}\ (I(\bar{x}) \implies !f(\bar{x})\ \&\ O(\bar{x}, f(\bar{x}))).$$

За съжаление, свойствата от тип тотална коректност не могат да се доказват с индуктивния принцип на Скот, защото очевидно пропада условието те да са в сила за никъде недефинираната функция $\emptyset^{(k)}$:

$$P_{t.c.}(\emptyset^{(k)}) \iff \underbrace{\forall \bar{x}\ (I(\bar{x}) \implies \underbrace{\underbrace{!(\emptyset^{(k)}(\bar{x})}_{false}\ \&\ O(\bar{x}, \emptyset^{(k)}(\bar{x}))}_{false})}_{false,\ \text{ако } I(\bar{x})=true \text{ за поне едно } \bar{x}}.$$

За сметка на това, обаче, всяко свойство от тип частична коректност е вярно за $\emptyset^{(k)}$ по тривиални причини ("от лъжата следва всичко"):

$$P_{p.c.}(\emptyset^{(k)}) \iff \underbrace{\forall \bar{x}\ (\underbrace{I(\bar{x})\ \&\ \underbrace{!\emptyset^{(k)}(\bar{x})}_{false}}_{false} \implies O(\bar{x}, \emptyset^{(k)}(\bar{x})))}_{true}$$

За щастие, всяко такова свойство се оказва и непрекъснато.

Твърдение 1.13. Всяко свойство от тип частична коректност е непрекъснато.

Доказателство. Да вземем произволно свойство $P_{p.c.}$ от този тип:

$$P_{p.c.}(f) \iff \forall \bar{x}\ (I(\bar{x})\ \&\ !f(\bar{x})) \implies O(\bar{x}, f(\bar{x}))).$$

Да вземем и монотонно растяща редица $f_0 \subseteq f_1 \subseteq \ldots$, такава че всеки неин член има свойството $P_{p.c.}$. Нека f е точната горна граница на тази редица. Да фиксираме произволно $\bar{x} \in \mathbb{N}^k$ и да предположим, че за него предпоставката в горната импликация е вярна, т.е. верни са $I(\bar{x})$ и $!f(\bar{x})$.

Щом $!f(\bar{x})$, съгласно дефиницията (1.3) за точна горна граница, ще съществува n, такова че $!f_n(\bar{x})$ и съответно $f_n(\bar{x}) = f(\bar{x})$. Но $P_{p.c.}(f_n)$ е в сила, вярна е и предпоставката $I(\bar{x})\ \&\ !f_n(\bar{x})$ на това свойство, откъдето следва, че е вярно и следствието $O(\bar{x}, f_n(\bar{x}))$, с други думи, вярно е $O(\bar{x}, f(\bar{x}))$. □

Свойствата от тип тотална коректност също са непрекъснати (убедете се сами!). За съжаление, те не могат да се третират с правилото на Скот, защото както отбелязахме по-горе, не са верни за никъде недефинираната функция $\emptyset^{(k)}$.

В задачите ще ни се налага да доказваме свойства, които са конюнкция на две или повече други, по-прости свойства. В тези случаи ще ни е от полза следващото твърдение:

Твърдение 1.14. Нека свойствата P_1 и P_2 са непрекъснати. Тогава $P_1\ \&\ P_2$ също е непрекъснато.

Доказателство. Фиксираме монотонно растяща редица $f_0 \subseteq f_1 \subseteq \ldots$ с точна горна граница f и приемаме, че

$$\forall n(P_1\ \&\ P_2)(f_n).$$

Тогава $\forall n P_1(f_n)$ и $\forall n P_2(f_n)$, откъдето (поради непрекъснатостта на P_1 и P_2) получаваме $P_1(f)$ и $P_2(f)$, а значи и $(P_1\ \&\ P_2)(f)$. □

Да се убедим, че дизюнкцията също запазва непрекъснатостта:

Твърдение 1.15. Ако P_1 и P_2 са непрекъснати, то и $P_1 \vee P_2$ е непрекъснато.

Доказателство. Тук вече не можем да твърдим, че

$$\forall n(P_1 \vee P_2)(f) \implies \forall n P_1(f)\ \vee\ \forall n P_2(f).$$

Затова ще разсъждаваме другояче.

Отново избираме растяща редица $f_0 \subseteq f_1 \subseteq \ldots$ с точна горна граница f и приемаме, че за всяко n е вярно $(P_1 \vee P_2)(f_n)$. Тогава за безброй много n ще е вярно $P_1(f_n)$ или $P_2(f_n)$ (или и двете). Да приемем, че се е случило първото.

Нека $\{f_{n_i}\}_i$ е подредицата на $\{f_n\}_n$, за която е вярно $\forall i P_1(f_{n_i})$. От непрекъснатостта на P_1 ще имаме, че то е в сила и за точната горна граница на редицата $\{f_{n_i}\}_i$. Да я означим с g:

$$g = \bigcup_i f_{n_i}.$$

Да се убедим, че g е точна горна граница и на цялата редица $\{f_n\}_n$, с други думи, $g = f$. Интуитивно е ясно, че g трябва да е подфункция на f, защото g е граница на подредица на редицата с граница f. Наистина, да приемем, че за произволни $\bar{x}, y$: $g(\bar{x}) \simeq y$. От определението за т.г.г. следва, че трябва да съществува i, за което $f_{n_i}(\bar{x}) \simeq y$. Но f е т.г.г. на цялата редица $\{f_n\}_n$ и значи и $f(\bar{x}) \simeq y$. Понеже $\bar{x}$ и y бяха произволни, то $g \subseteq f$.

Обратно, ако $f(\bar{x}) \simeq y$, то значи $f_n(\bar{x}) \simeq y$ за някое n. Нека i е такова, че $n_i \geq n$. Тогава $f_{n_i} \supseteq f_n$, което означава, че и $f_{n_i}(\bar{x}) \simeq y$, а оттук и $g(\bar{x}) \simeq y$. Така получихме, че от $f(\bar{x}) \simeq y$ следва, че и $g(\bar{x}) \simeq y$, и това е изпълнено за произволните $\bar{x}$ и y. Следователно $g \subseteq f$.

Сега от $P_1(g)$ и $g = f$ получаваме $P_1(f)$, откъдето и $(P_1 \vee P_2)(f)$. $\square$

Твърдение 1.16. Нека Γ и Δ са непрекъснати оператори от един и същи тип $(k \to m)$. Да дефинираме свойството P в $\mathcal{F}_k$ по следния начин:

$$P(f) \overset{\text{деф}}{\iff} \Gamma(f) \subseteq \Delta(f).$$

Тогава P е непрекъснато свойство.

Доказателство. Да вземем монотонно растяща редица $f_0 \subseteq f_1 \subseteq \ldots$ с точна горна граница f и да приемем, че за всяко n $P(f_n)$.

Непрекъснатостта на Γ и Δ означава, че

$$\Gamma(f) = \Gamma(\bigcup_n f_n) = \bigcup_n \Gamma(f_n) \qquad \text{и} \qquad \Delta(f) = \Delta(\bigcup_n f_n) = \bigcup_n \Delta(f_n).$$

Следователно нашето условие $\Gamma(f) \subseteq \Delta(f)$ е еквивалентно с

$$\bigcup_n \Gamma(f_n) \subseteq \bigcup_n \Delta(f_n), \tag{1.8}$$

верността на което се съобразява ето как:

За фиксирано n, понеже $P(f_n)$ е вярно, ще имаме, че $\Gamma(f_n) \subseteq \Delta(f_n)$. Ясно е, че $\Delta(f_n) \subseteq \bigcup_n \Delta(f_n)$, откъдето общо

$$\Gamma(f_n) \ \subseteq \ \bigcup_n \Delta(f_n).$$

Но n беше произволно, следователно функцията $\bigcup_n \Delta(f_n)$ мажорира *всеки* член на редицата $\{\Gamma(f_n)\}_n$, и значи тя мажорира и точната ѝ горна граница $\bigcup_n \Gamma(f_n)$, което е точно условието (1.8). □

Твърдение 1.17. Нека Γ и Δ са непрекъснати оператори от един и същи тип. Тогава е непрекъснато и свойството

$$P(f) \ \stackrel{\text{деф}}{\iff} \ \Gamma(f) = \Delta(f).$$

Доказателство. Директно от предишните две твърдения, тъй като

$$P(f) \quad \iff \quad \Gamma(f) \subseteq \Delta(f) \ \& \ \Delta(f) \subseteq \Gamma(f).$$

□

Сега да се върнем към оператора Γ от нашия мотивационен *Пример* 1.14 от началото на този раздел, за да видим как с индуктивния принцип на Скот ще можем да докажем свойство на f_Γ.

Задача 1.6. Нека Γ е дефиниран по следния начин:

$$\Gamma(f)(x) \simeq \begin{cases} \frac{x}{2}, & \text{ако } x \text{ е четно} \\ f(f(\frac{3x+1}{2})), & \text{ако } x \text{ е нечетно.} \end{cases}$$

Докажете, че за f_Γ е изпълнено:

$$\forall x(!f_\Gamma(x) \implies f_\Gamma(x) \leq \frac{x}{2}).$$

Решение. Ще приемем наготово, че Γ е непрекъснат и следователно можем да приложим индуктивния принцип на Скот.

Да означим с P свойството от условието на задачата:

$$P(f) \iff \forall x(!f(x) \implies f(x) \leq \frac{x}{2}).$$

За да покажем, че $P(f_\Gamma)$ е вярно, трябва да проверим трите изисквания от правилото на Скот:

1) $P(\emptyset^{(1)})$;

2) $P(f) \implies P(\Gamma(f))$ за всяка $f \in \mathcal{F}_1$;

3) свойството P е непрекъснато.

Но P е свойство от тип частична коректност и съгласно *Твърдение* 1.13, то е непрекъснато. За такива свойства забелязахме също, че първото изискване $P(\emptyset^{(1)})$ е винаги вярно.

Следователно остана да проверим само условието 2), което впрочем е единственото, в което участва Γ.

Наистина, да фиксираме произволна функция $f \in \mathcal{F}_1$ и да приемем, че за нея $P(f)$ е вярно. Това се явява *индуктивната хипотеза* при този тип индукция.

Да направим индуктивната стъпка означава да покажем, че е вярно и $P(\Gamma(f))$. Свойството $P(\Gamma(f))$ изглежда така:

$$P(\Gamma(f)) \iff \forall x(!\Gamma(f)(x) \implies \Gamma(f)(x) \leq \frac{x}{2}).$$

Избираме произволно x и приемаме, че за него $!\Gamma(f)(x)$. Трябва да покажем, че

$$\Gamma(f)(x) \leq \frac{x}{2}.$$

Но на колко е равно $\Gamma(f)(x)$? Поглеждаме към определението на Γ и виждаме, че това зависи от четността на x. Разглеждаме поотделно двете възможности за x.

1 сл. x е четно. Този случай е базисен за Γ, т.е. стойността на $\Gamma(f)(x)$ въобще не зависи от f. По-точно, имаме, че $\Gamma(f)(x) \overset{\text{деф}}{=} \frac{x}{2} \leq \frac{x}{2}$.

2 сл. x е нечетно. В този случай $\Gamma(f)(x) \overset{\text{деф}}{\simeq} f(f(\frac{3x+1}{2}))$.

Нашето предположение е, че $\Gamma(f)(x)$ има стойност, т.е. изразът $f(f(\frac{3x+1}{2}))$ има стойност. Това означава, съгласно определението за суперпозиция, че и изразът $f(\frac{3x+1}{2})$ трябва да има стойност. Сега прилагаме двукратно индуктивното предположение $P(f)$ и получаваме последователно:

$$f(f(\frac{3x+1}{2})) \overset{\text{и.х. } P(f)}{\leq} \frac{f(\frac{3x+1}{2})}{2} \overset{\text{и.х. } P(f)}{\leq} \frac{\frac{\frac{3x+1}{2}}{2}}{2} = \frac{3x+1}{8} \leq \frac{x}{2}.$$

С това приключи индуктивната стъпка, т.е. преходът $P(f) \implies P(\Gamma(f))$, и понеже f беше произволна, можем да твърдим, че условието 2) е в сила за всяка функция $f \in \mathcal{F}_1$.

Да обобщим: показахме, че и трите изисквания от индуктивния принцип на Скот са изпълнени, и значи можем да твърдим, че свойството P е в сила и за f_Γ, с други думи, вярно е, че

$$\forall x(!f_\Gamma(x) \implies f_\Gamma(x) \leq \frac{x}{2}).$$

□

Глава 2

Области на Скот

Областите на Скот (ОС) са абстрактната математическа среда, в която се развива т. нар. Теория на неподвижните точки — математически подход, чрез който се дефинира *денотационна семантика* или *семантика с неподвижни точки* (*fixpoint semantics*) на някои видове програми — рекурсивни, логически и пр. Всичко, което правихме дотук в глава Оператори, беше всъщност работа в една конкретна област на Скот (*Твърдение* 2.1).

2.1 Определение и примери

2.1.1 Пълни наредби

Да напомним, че бинарната релация $\leqslant$ в множеството A е *частична наредба* на A, ако тя е рефлексивна, транзитивна и антисиметрична.

Ще казваме, че частичната наредба $\leqslant$ е *пълна* (*complete*), ако всяка монотонно растяща редица

$$a_0 \leqslant a_1 \leqslant a_2 \ \ldots$$

от елементи на A (или всяка *верига в* A) има точна горна граница $\underset{n}{lub}\, a_n$, която принадлежи на A.

В конкретните ОС, които по-нататък ще ни интересуват, ще използваме специфични означения за точната горна граница — например $\bigcup_n f_n$, както беше в предишната глава, или $\bigsqcup_n f_n$, както ще е в раздел 3.5, където ще разглеждаме една друга важна ОС. За общата теория на областите на Скот предпочитаме, обаче, по-неутралното *lub* (от **l**east **u**pper **b**ound).

Определение 2.1. *Област на Скот (Scott domain)* наричаме наредена тройка $\boldsymbol{A} = (A, \leqslant, a_0)$, за която са изпълнени условията:

1) A е непразно множество;

2) $\leqslant$ е пълна частична наредба на A;

3) $a_0 \in A$ е най-малкият елемент на A (относно наредбата $\leqslant$), с други думи, $a_0 \leqslant a$ за всяко $a \in A$.

Множеството A ще наричаме *носител* или *домейн* на структурата $\boldsymbol{A}$. В англоезичната литература областите на Скот се наричат още *cpo* — от ***c**omplete **p**artial **o**rder*.

Горните аксиоми на ОС са минималните изисквания, които са необходими, за да може в структурата $(A, \leqslant, a_0)$ да се докаже теорема на Кнастер-Тарски.

2.1.2 Областта на Скот $(\mathcal{F}_n,\ \subseteq,\ \emptyset^{(n)})$ и други примери

Ето няколко примера за структури, които са (или не са) области на Скот.

Примери:

1) Нека M е произволно множество, а $\mathscr{P}(M) = \{A \mid A \subseteq M\}$. Тогава наредената тройка $(\mathscr{P}(M),\ \subseteq,\ \emptyset)$ е област на Скот.
Доказателство. Добре известно е, че релацията $\subseteq$ е частична наредба в $\mathscr{P}(M)$. Това, че тя е пълна, следва от факта, че за всяка фамилия $\{A_i |\ i \in I\}$ от подмножества на M, обединението

$$\bigcup_{i \in I} A_i$$

се явява точна горна граница на тази фамилия. В частност, това ще е вярно и за всяка *редица* $A_0, A_1, \ldots$ в $\mathscr{P}(M)$. Да отбележим, че тук не е необходимо редицата да е монотонно растяща, за да притежава точна горна граница. □

2) Структурата $(\mathbb{N},\ \leq,\ 0)$ *не* е област на Скот (тук $\leq$ е обичайното неравенство в $\mathbb{N}$).
Доказателство. Релацията $\leq$ е наредба на $\mathbb{N}$ (дори е тотална наредба), но очевидно не всяка монотонно растяща редица има граница. Пример за такава редица е, да кажем, редицата $0,\ 1,\ 2, \ldots$. □

3) Област на Скот е разширената структура $(\mathbb{N} \cup \{\infty\},\ \leq,\ 0)$, където $0 \leq \infty,\ 1 \leq \infty, \ldots$, т.е. ∞ е най-големият елемент на $\mathbb{N} \cup \{\infty\}$.
Доказателство. Множеството $\mathbb{N} \cup \{\infty\}$ изглежда така: $0 \leq 1 \leq \cdots \leq \infty$. Сега ако растящата редица $a_0 \leq a_1 \leq \ \ldots$ е ограничена, то тя очевидно

има вида $a_0 \leq a_1 \leq \ldots a_n = a_{n+1} = \ldots$ и значи нейната граница е a_n. Ако тази редица е неограничена, то нейната граница ще е ∞. □

4) Структурата $(\{a,b\}^\star, \leqslant, \varepsilon)$, където $\leqslant$ е релацията "префикс", а ε е празният низ, *не* е област на Скот.

Доказателство. Редицата $a \sqsubseteq aa \sqsubseteq aaa\ldots$ очевидно няма точна горна граница. □

За една друга структура — $(\mathcal{F}_n, \subseteq, \emptyset^{(n)})$ — на практика вече знаем, че е област на Скот. И тъй като този факт ще е от особена важност, ще го формулираме като отделно твърдение, което ще цитираме многократно по-нататък.

Твърдение 2.1. За всяко $n \geq 1$ структурата $\boldsymbol{\mathcal{F}_n} = (\mathcal{F}_n, \subseteq, \emptyset^{(n)})$ е област на Скот.

Доказателство. Имаме, че за всяка $f \in \mathcal{F}_n$

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

т.е. празната функция $\emptyset^{(n)}$ е най-малкият елемент на $\mathcal{F}_n$. Освен това вече видяхме, че релацията $\subseteq$ е частична наредба (*Твърдение* 1.2), която при това е пълна (*Твърдение* 1.5).

Следователно $\boldsymbol{\mathcal{F}_n} = (\mathcal{F}_n, \subseteq, \emptyset^{(n)})$ е област на Скот. □

Да отбележим, че ако се ограничим до множеството на всички *крайни* n-местни функции $\mathcal{F}_n^{fin} \subseteq \mathcal{F}_n$, структурата с носител $\mathcal{F}_n^{fin}$ вече не е област на Скот.

Задача 2.1. Докажете, че за всяко $n \geq 1$ структурата $\boldsymbol{\mathcal{F}_n^{fin}} = (\mathcal{F}_n^{fin}, \subseteq, \emptyset^{(n)})$ *не* е област на Скот.

Решение. Проблемът е в това, че границата на монотонно растяща редица от крайни функции може да е безкрайна, или другояче казано, свойството

$$P(f) \iff f \text{ е крайна}$$

не е непрекъснато в $\mathcal{F}_n^{fin}$ — нещо, което вече установихме в *Задача* 1.5. Това означава, че частичната наредба $\subseteq$ в множеството на *крайните* n-местни функции не е пълна. □

Накрая да отбележим и очевидния факт, че ако разглеждаме частични функции в *произволно множество* D, наредени с релацията $\subseteq$, отново имаме област на Скот.

Задача 2.2. Да фиксираме $n \geq 1$. Нека D е произволно множество, а $F_n = \{f \mid f : D^n \rightharpoonup D\}$. За $f, g \in F_n$ да положим $f \subseteq g \iff G_f \subseteq G_g$. Нека още $\emptyset^{(n)}$ е никъде недефинираната функция в D. Докажете, че наредената тройка $(F_n, \subseteq, \emptyset^{(n)})$ е област на Скот.

Решение. Повтаря доказателството на *Твърдение* 2.1. □

2.1.3 Плоската наредба и плоската ОС $(D_\bot, \sqsubseteq, \bot)$

Сега ще въведем една съвсем проста бинарна релация в произволно множество D, която ще се окаже частична наредба — тъй нареченатa *плоска наредба (flat order)*. Макар и да е много проста, тази наредба стои в основата на едната от двете най-важни за теоретичната информатика области на Скот — областта, която е модел за *call-by-name*.

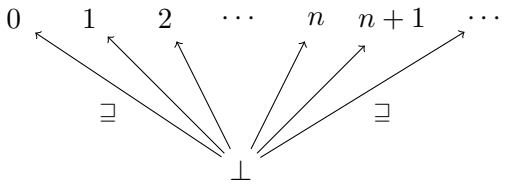
Да фиксираме произволно непразно множество D и да изберем един обект $\bot$, такъв че $\bot \notin D$. Предназначението на $\bot$ е да бъде нещо като *име на недефинираността*, т.е. грубо казано, в този модел вместо $\neg!f(5)$ ще пишем $f(5) = \bot$. Ако ви се струва, че това е едно и също, изчакайте до следващата глава $\ddot\smile$.

Нека $D_\bot = D \cup \{\bot\}$. В $D_\bot$ дефинираме следната бинарна релация:

$$a \sqsubseteq b \overset{\text{деф}}{\iff} a = \bot \ \vee \ a = b. \tag{2.1}$$

Ясно е, че $\bot \sqsubseteq a$ за всяко $a \in D_\bot$, т.е. $\bot$ е най-малкият елемент на $D_\bot$. Образно казано, той е на дъното на $D_\bot$, затова понякога се нарича *bottom* елемент. Ясно е още, че $a \sqsubseteq a$ за всяко $a \in D_\bot$, и това всъщност са всички връзки между елементите на $D_\bot$.

Ето как изглежда графично релацията $\sqsubseteq$ в множеството $\mathbb{N}_\bot$ (без примките $n \sqsubseteq n$):



Твърдение 2.2. За всяко множество D структурата $\boldsymbol{D_\bot} = (D_\bot, \sqsubseteq, \bot)$ е област на Скот.

Доказателство. Вече забелязахме, че $\bot$ е най-малкият елемент на $D_\bot$. Да проверим условията от дефиницията за частична наредба:

- *рефлексивност*: При $a = b$ дясната част на (2.1) е винаги вярна и следователно $a \sqsubseteq a$ за всяко $a \in D_\bot$ (което вече отбелязахме по-горе).
- *транзитивност*: За произволни $a, b, c \in D_\bot$ нека $a \sqsubseteq b \ \& \ b \sqsubseteq c$. Трябва да видим, че $a \sqsubseteq c$. Ако $a = \bot$, то очевидно $a \sqsubseteq c$. Ако $a \neq \bot$, то съгласно (2.1) трябва $a = b$, и понеже $b \sqsubseteq c$, значи отново $a \sqsubseteq c$.

- *антисиметричност*: За произволни a и b от $D_\perp$ нека $a \sqsubseteq b$ & $b \sqsubseteq a$. Трябва да покажем, че $a = b$. От $a \sqsubseteq b$ имаме според (2.1) два случая: $a = b$ или $a = \perp$. Ако е налице първият случай — чудесно; ако пък $a = \perp$, то от $b \sqsubseteq a$ ще имаме, че и $b = \perp$, и значи отново $a = b$.

Остана да видим, че $\sqsubseteq$ е пълна. Да вземем една монотонно растяща редица в $D_\perp$:

$$a_0 \sqsubseteq a_1 \sqsubseteq \dots$$

Как изглежда тя? Ако първият ѝ член a_0 не е $\perp$, то тогава

$$a_0 = a_1 = a_2 = \dots$$

и значи границата на тази редица е a_0.

Ако $a_0 = \perp$, отново имаме два случая: първият е всички елементи на редицата да са $\perp$, който е ясен, а вторият — да съществува $n : a_n \neq \perp$. Ако n е първото естествено число, за което това се случва, то редицата ще изглежда така:

$$\underbrace{\perp, \dots \perp}_{n \text{ пъти}}, a_n, a_n, \dots$$

и очевидно нейната граница е a_n. □

Определение 2.2. Наредбата $\sqsubseteq$ на $D_\perp$ ще наричаме *плоска наредба*, а наредената тройка $(D_\perp,\ \sqsubseteq,\ \perp)$ — *плоска област на Скот*.

Да отбележим отново, че всяка монотонно растяща редица (верига) $a_0 \sqsubseteq a_1 \sqsubseteq \dots$ в областта $(D_\perp,\ \sqsubseteq,\ \perp)$ изглежда по един от следните три начина:

- $\perp,\ \perp,\ \perp, \dots$
- $\underbrace{\perp, \dots \perp}_{n \geq 1}, a,\ a, \dots$
- $a,\ a,\ a, \dots$

Точната горна граница на редицата $a_0 \sqsubseteq a_1 \sqsubseteq \dots$ ще означаваме с

$$\bigsqcup_n a_n$$

или само с $\bigsqcup a_n$. За нас най-голям интерес ще представлява плоската област с носител $\mathbb{N}_\perp$.

2.2 Конструкции на области на Скот

По-горе отбелязахме, че плоската ОС $(\mathbb{N}_\perp, \sqsubseteq, \perp)$ ще ни трябва за моделиране на семантиката с отложени пресмятания на рекурсивните програми в $\mathbb{N}$. Всъщност истината е, че за да моделираме тези пресмятания, ще ни е нужна една по-друга област — тази на *тоталните функции* в $\mathbb{N}_\perp$, и по-общо — на тоталните функции в $\mathbb{N}_\perp^n$. Целта на този раздел е да покажем, че тези функции също образуват област на Скот. Този факт ще получим като частен случай на две много общи конструкции, чрез които по дадени ОС ще получаваме нова ОС. Първата конструкция е декартово произведение.

2.2.1 Декартово произведение на ОС

Нека са дадени k на брой области на Скот

$$\boldsymbol{A_1} = (A_1, \leqslant_1, \perp_1), \ \ldots\ , \boldsymbol{A_k} = (A_k, \leqslant_k, \perp_k).$$

В декартовото произведение $A = A_1 \times \cdots \times A_k$ определяме нова релация $\leqslant$, породена от локалните наредби във всяка от дадените области. Поконкретно, за произволни $(a_1, \ldots, a_k) \in A$ и $(b_1, \ldots, b_k) \in A$ полагаме:

$$(a_1, \ldots, a_k) \leqslant (b_1, \ldots, b_k) \overset{\text{деф}}{\iff} a_1 \leqslant_1 b_1 \ \&\ \ldots\ \&\ a_k \leqslant_k b_k. \tag{2.2}$$

Твърдение 2.3. Така въведената релация $\leqslant$ е частична наредба в $A = A_1 \times \cdots \times A_k$ с най-малък елемент $\perp = (\perp_1, \ldots, \perp_k)$.

Доказателство. Очевидно от дефинициите $\ddot\smile$. □

Тази наредба ще наричаме *покомпонентна наредба*. Да се убедим, че тя е пълна:

Твърдение 2.4. Нека $\boldsymbol{A_1} = (A_1, \leqslant_1, \perp_1), \ \ldots\ , \boldsymbol{A_k} = (A_k, \leqslant_k, \perp_k)$ са области на Скот. Тогава е вярно, че:

1) Редицата $\{(a_1^n, \ldots, a_k^n)\}_n$ е монотонно растяща в $A_1 \times \cdots \times A_k$ тогава и само тогава, когато всяка от редиците $\{a_i^n\}_n$ е монотонно растяща в A_i, за $i = 1, \ldots, k$.

2) Ако редицата $\{(a_1^n, \ldots, a_k^n)\}_n$ е монотонно растяща в $A_1 \times \cdots \times A_k$, то тя има точна горна граница $(b_1, \ldots, b_k)$, където $b_i, 1 \leq i \leq k$ е точната горна граница на редицата $\{a_i^n\}_n$ в A_i.

Доказателство. 1) За произволно n имаме по определение:

$$(a_1^n, \ldots, a_k^n) \leqslant (a_1^{n+1}, \ldots, a_k^{n+1}) \iff a_1^n \leqslant_1 a_1^{n+1} \ \& \ \ldots \ \& \ a_k^n \leqslant_k a_k^{n+1}.$$

Следователно редицата от k-орките е растяща в $A_1 \times \cdots \times A_k$ точно когато локално по всяка компонента са растящи редиците $\{a_i^n\}_n$.

2) Нека $\{(a_1^n, \ldots, a_k^n)\}_n$ е монотонно растяща в $A_1 \times \cdots \times A_k$. Току-що видяхме, че тогава и всяка от редиците $\{a_i^n\}_n$ ще е монотонно растяща в A_i. Но $(A_i, \leqslant_i, \bot_i)$ е област на Скот. Следователно там $\{a_i^n\}_n$ има точна горна граница, да я означим с b_i:

$$b_i = \underset{n}{lub}\ a_i^n.$$

Изглежда логично k-орката $(b_1, \ldots, b_k)$ да е точната горна граница на редицата $\{(a_1^n, \ldots, a_k^n)\}_n$. Това наистина е така и в доказателството няма нищо неочаквано, но да го проведем все пак.

Да фиксираме някакво n. Имаме, че $a_i^n \leqslant_i b_i$ за всяко $i = 1, \ldots\ k$, което означава, че

$$(a_1^n, \ldots, a_k^n) \ \leqslant \ (b_1, \ldots, b_k).$$

Понеже това е за всяко n, значи $(b_1, \ldots, b_k)$ мажорира всеки член на редицата $\{(a_1^n, \ldots, a_k^n)\}_n$, т.е. $(b_1, \ldots, b_k)$ е горна граница на тази редица. Сега ако $(c_1, \ldots, c_k)$ е друга нейна горна граница, то за произволно n ще е изпълнено:

$$(a_1^n, \ldots, a_k^n) \ \leqslant \ (c_1, \ldots, c_k).$$

В частност, при фиксирано i ще имаме $a_i^n \leqslant_i c_i$, и това е за всяко n, т.е. c_i е горна граница за редицата $\{a_i^n\}_n$. Но b_i е нейната точна горна граница и значи

$$b_i \leqslant_i c_i.$$

И тъй като това е вярно за всяко $i = 1, \ldots, k$, то $(b_1, \ldots, b_k) \leqslant (c_1, \ldots, c_k)$. □

Точната горна граница на редицата $\{(a_1^n, \ldots, a_k^n)\}_n$ ще означаваме с $\underset{n}{lub}\ (a_1^n, \ldots, a_k^n)$. От доказаното по-горе можем да запишем:

$$\underset{n}{lub}\ (a_1^n, \ldots, a_k^n) \ = \ (\underset{n}{lub}\ a_1^n, \ldots, \underset{n}{lub}\ a_k^n) \tag{2.3}$$

Твърдения 2.3 и 2.4 ни дават общо, че декартово произведение на ОС е ОС:

Твърдение 2.5. Нека $\boldsymbol{A_1} = (A_1, \leqslant_1, \bot_1), \ldots, \boldsymbol{A_k} = (A_k, \leqslant_k, \bot_k)$ са ОС. Тогава декартовото им произведение $\boldsymbol{A} = (A_1 \times \cdots \times A_k, \leqslant, (\bot_1, \ldots, \bot_k))$ също е област на Скот.

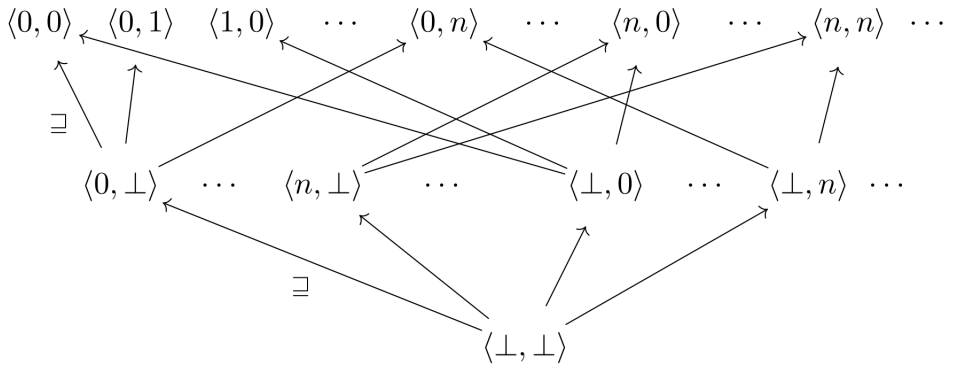
От това твърдение получаваме, че две важни за нашите разглеждания структури са области на Скот.

Следствие 2.1. Структурата $\mathbb{N}_\perp^{\boldsymbol{n}} = (\underbrace{\mathbb{N}_\perp \times \cdots \times \mathbb{N}_\perp}_{n \text{ пъти}},\ \sqsubseteq,\ \underbrace{(\perp, \ldots, \perp)}_{n \text{ пъти}})$ е област на Скот.

По-нататък декартовото произведение $\underbrace{\mathbb{N}_\perp \times \cdots \times \mathbb{N}_\perp}_{n \text{ пъти}}$ ще съкращаваме до $\mathbb{N}_\perp^n$. Да обърнем внимание, че това е означение за $(\mathbb{N}_\perp)^n$, а *не* за $(\mathbb{N}^n)_\perp \stackrel{\text{деф}}{=} \mathbb{N}^n \cup \{\perp\}$ — това множество няма да представлява никакъв интерес за нас. Това, което ще ни е нужно, е елементът $\perp$ да присъства във всяка компонента на декартовото произведение $\mathbb{N}_\perp^n$.

Разбира се, наредбата $\sqsubseteq$ в новата ОС $\mathbb{N}_\perp^{\boldsymbol{n}} = (\mathbb{N}_\perp^n,\ \sqsubseteq,\ (\perp, \ldots, \perp))$ е релация между n-торки в $\mathbb{N}_\perp^n$, макар че ние ще я означаваме със същия символ, с който бележехме плоската наредба в първоначалната ОС $(\mathbb{N}_\perp,\ \sqsubseteq,\ \perp)$. Областта на Скот $\mathbb{N}_\perp^{\boldsymbol{n}}$ ще изучим подробно, когато стигнем до денотационната семантика с предаване на параметрите по име в раздел 3.5.

Ето как изглежда тази наредба при $n = 2$, т.е. за областта на Скот $\mathbb{N}_\perp^{\boldsymbol{2}} = (\mathbb{N}_\perp \times \mathbb{N}_\perp,\ \sqsubseteq,\ (\perp, \perp))$.



Вече видяхме (*Твърдение* 2.1), че за произволно n структурата $(\mathcal{F}_n,\ \subseteq,\ \emptyset^{(n)})$ е ОС. Сега отново прилагаме доказаното по-горе *Твърдение* 2.5 и получаваме още една важна за нас ОС:

Следствие 2.2. За произволни положителни $n_1, \ldots, n_k$, структурата $\boldsymbol{\mathcal{F}} = (\mathcal{F}_{n_1} \times \cdots \times \mathcal{F}_{n_k},\ \subseteq,\ (\emptyset^{(n_1)}, \ldots, \emptyset^{(n_k)})\ )$ е област на Скот.

Точно в тази ОС в следващата глава ще дефинираме денотационната семантика с предаване на параметрите *по стойност*.

2.2.2 Функционални пространства над ОС

Нека M е произволно множество, а $\boldsymbol{A} = (A, \leqslant, a_0)$ е някаква ОС. Ще ни интересуват *тоталните* изображения от M към A. Нека $\mathcal{F}$ е съвкупността от всички такива изображения:

$$\mathcal{F} = \{f \mid f \colon M \longrightarrow A\}.$$

В $\mathcal{F}$ въвеждаме бинарна релация $\leqslant\!\!\!\!\leqslant$, породена от наредбата в A, по следния естествен начин: за произволни $f, g \in \mathcal{F}$ полагаме

$$f \leqslant\!\!\!\!\leqslant g \overset{\text{деф}}{\iff} \forall x \in M \; f(x) \leqslant g(x). \tag{2.4}$$

Както ще се убедим след малко, тази релация също е частична наредба. Ще я наричаме *поточкова наредба*. Нещо повече, тази наредба ще се окаже пълна. Ясно кой ще е най-малкият елемент на $\mathcal{F}$ — това ще е функцията, която винаги връща най-малкия елемент на A. Тази функция ще означаваме с Ω:

$$\Omega(x) \overset{\text{деф}}{=} a_0$$

за всяко $x \in M$.
Наистина, да вземем произволна $f \in \mathcal{F}$. Имаме, че за всяко $x \in M$:

$$\Omega(x) = a_0 \leqslant f(x),$$

и следователно $\Omega \leqslant\!\!\!\!\leqslant f$, т.е. действително Ω е най-малката функция в $\mathcal{F}$. Вече сме готови да покажем, че:

Твърдение 2.6. При означенията от по-горе, структурата $\boldsymbol{\mathcal{F}} = (\mathcal{F}, \leqslant\!\!\!\!\leqslant, \Omega)$ е област на Скот.

Доказателство. Аксиомите за частична наредба се проверяват непосредствено. Да видим, например, защо $\leqslant\!\!\!\!\leqslant$ е транзитивна:
Наистина, нека за $f, g, h \in \mathcal{F}$ е вярно, че

$$f \leqslant\!\!\!\!\leqslant g \text{ и } g \leqslant\!\!\!\!\leqslant h.$$

За да видим, че и $f \leqslant\!\!\!\!\leqslant h$, да вземем произволно $x \in M$. От определение (2.4) имаме, че

$$f(x) \leqslant g(x) \quad \text{и} \quad g(x) \leqslant h(x).$$

Но $f(x), g(x)$ и $h(x)$ са елементи на носителя A на ОС $\boldsymbol{A} = (A, \leqslant, a_0)$. Тогава от транзитивността на $\leqslant$ ще имаме $f(x) \leqslant h(x)$. Но $x \in M$ беше произволно и значи

$$\forall x \in M \; f(x) \leqslant h(x),$$

което съгласно (2.4) означава точно $f \leqslant\!\!\!\!\leqslant h$.

За да се убедим, че новата наредба $\leqslant\!\!\!\cdot$ е пълна, да вземем една монотонно растяща редица в $\mathcal{F}$:

$$f_0 \leqslant\!\!\!\cdot\ f_1 \leqslant\!\!\!\cdot\ \ldots$$

Тогава от определението на релацията $\leqslant\!\!\!\cdot$ ще имаме, че за кое да е $x \in M$:

$$f_0(x) \leqslant f_1(x) \leqslant \ldots,$$

с други думи, тази редица е монотонно растяща в A. Но там релацията $\leqslant$ е пълна, следователно тя има т. г. граница

$$\underset{n}{lub}\ f_n(x).$$

Звучи логично точната горна границата на редицата $\{f_n\}_n$ да е функцията f, която се дефинира с равенството

$$f(x) \overset{\text{деф}}{=} \underset{n}{lub}\ f_n(x) \tag{2.5}$$

за всяко $x \in M$.

За да се убедим, че това е така, да проверим най-напред, че

$$\forall n\ f_n \leqslant\!\!\!\cdot\ f,$$

т.е. че f е горна граница на редицата $\{f_n\}_n$. За целта да фиксираме произволно $n \in \mathbb{N}$. Неравенството $f_n \leqslant\!\!\!\cdot\ f$ е еквивалентно на

$$\forall x \in M\ f_n(x) \leqslant f(x),$$

което е вярно, защото по дефиниция $f(x)$ мажорира $f_n(x)$ за всяко $x \in M$.

Нека сега g е друга горна граница на редицата $\{f_n\}_n$, т.е.

$$\forall n\ f_n \leqslant\!\!\!\cdot\ g.$$

Тогава съгласно определение (2.4):

$$\forall n\ \forall x \in M\ f_n(x) \leqslant g(x).$$

Последното е еквивалентно на

$$\forall x \in M\ \forall n\ f_n(x) \leqslant g(x).$$

Сега да фиксираме $x \in M$. Имаме $\forall n\ f_n(x) \leqslant g(x)$, което означава, че $g(x)$ е горна граница на редицата $\{f_n(x)\}_n$. Но $f(x)$ е точната горна граница на тази редица и значи $f(x) \leqslant g(x)$. Това неравенство е изпълнено за всяко $x \in M$, което ни дава финално $f \leqslant\!\!\!\cdot\ g$, т.е. f е най-малката сред горните граници на $\{f_n\}_n$. $\square$

Да запишем още веднъж как се дефинира точната горна граница $\underset{n}{lub}\ f_n$ на една монотонно растяща редица $f_0 \leqslant f_1 \leqslant \ldots$:

$$(\underset{n}{lub}\ f_n)(x) \ = \ \underset{n}{lub}\ f_n(x) \tag{2.6}$$

Тук първата точна горна граница е в областта на Скот $\boldsymbol{\mathcal{F}}$, а втората — в областта $\boldsymbol{A}$.

Да приложим току-що доказаното за случая, когато множеството M е $\mathbb{N}_\perp^n$, а областта на Скот $\boldsymbol{A}$ е $(\mathbb{N}_\perp,\ \sqsubseteq,\ \perp)$. Новополучената функционална ОС ще е с домейн — множеството от всички *тотални* n-местни функции в $\mathbb{N}_\perp$. Това множество ще означаваме с $\mathcal{F}_n^\perp$, по подобие на множество на n-местните *частични* функции в $\mathbb{N}$, което означавахме с $\mathcal{F}_n$. С други думи,

$$\mathcal{F}_n^\perp = \{f \mid f\colon\ \mathbb{N}_\perp^n \longrightarrow \mathbb{N}_\perp\}.$$

Нека още

$$\Omega^{(n)}(x_1, \ldots, x_n) \ \stackrel{\text{деф}}{=} \ \perp$$

за всяко $(x_1, \ldots, x_n) \in \mathbb{N}_\perp^n$.

Сега от *Твърдение* 2.6 получаваме важното

Следствие 2.3. Структурата $\boldsymbol{\mathcal{F}_n^\perp} = (\mathcal{F}_n^\perp,\ \sqsubseteq,\ \Omega^{(n)})$ е област на Скот.

Комбинираме този факт с доказаното по-горе за декартови произведения на ОС (*Твърдение* 2.5) и получаваме по-общо, че

Следствие 2.4. За произволни положителни $n_1, \ldots, n_k$, структурата

$$\boldsymbol{\mathcal{F}^\perp} = (\mathcal{F}_{n_1}^\perp \times \cdots \times \mathcal{F}_{n_k}^\perp,\ \sqsubseteq,\ (\Omega^{(n_1)}, \ldots, \Omega^{(n_k)}))$$

е област на Скот.

Точно в тази ОС ще дефинираме денотационната семантика с предаване на параметрите *по име*.

2.3 Теорията на неподвижните точки за произволна ОС

Тук по същество ще повторим дефинициите и твърденията от последните два раздела 1.4 и 1.5 на предишната глава, като този път ще работим не в познатата ни ОС $(\mathcal{F}_n,\ \subseteq,\ \emptyset^{(n)})$, а в произволна област на Скот. Доказателствата на твърденията в този раздел също ще бъдат много подобни на съответните им, които доказахме дотук.

2.3.1 Непрекъснати изображения в ОС

Ще предполагаме, че са фиксирани две области на Скот

$$\boldsymbol{\mathcal{F}} = (\mathcal{F},\ \leqslant,\ \Omega) \quad \text{и} \quad \boldsymbol{\mathcal{F}'} = (\mathcal{F}',\ \leqslant',\ \Omega').$$

Съвсем умишлено ги означаваме с $\boldsymbol{\mathcal{F}}$ и $\boldsymbol{\mathcal{F}'}$ (а не с $\boldsymbol{A}, \boldsymbol{A_1}, \ldots$, както беше в предишния раздел), защото резултатите, които сега ще получим, ще прилагаме за области на Скот, които са с носители — множества от функции. Ще ни интересуват основно множествата $\mathcal{F}_n$, $\mathcal{F}_n^{\perp}$ и техните декартови произведения.

Когато областта на Скот е произволна (в смисъл, неуточнена), точните горни граници в нея ще означаваме с lub. Когато сме в ОС $(\mathcal{F}_n,\ \subseteq,\ \emptyset^{(n)})$, ще използваме добре познатото ни означение $\bigcup$ за точна горна граница. Накрая, ако областта на Скот е $(\mathcal{F}_n^{\perp},\ \sqsubseteq,\ \Omega^{(n)})$, точната горна граница ще отбелязваме със знака $\bigsqcup$. Същите означения ще използваме и при декартови произведения на ОС от първия и втория тип.

Изображенията, които ще разглеждаме, ще действат от $\mathcal{F}$ към $\mathcal{F}'$. Вече отбелязахме, че тези ОС в нашите приложения ще са свързани с функции, затова изображенията в тях ще наричаме отново *оператори*.

Определение 2.3. Ще казваме, че операторът $\Gamma\colon \mathcal{F} \longrightarrow \mathcal{F}'$ е *монотонен*, ако за всяка двойка $f, g \in \mathcal{F}$ е изпълнено:

$$f \leqslant g \implies \Gamma(f) \leqslant' \Gamma(g).$$

Определение 2.4. Операторът $\Gamma\colon \mathcal{F} \longrightarrow \mathcal{F}'$ ще наричаме *непрекъснат*, ако за всяка монотонно растяща редица $f_0 \leqslant f_1 \leqslant \ldots$ в $\mathcal{F}$ е изпълнено:

$$\Gamma(\underbrace{\underset{n}{lub}\ f_n}_{\text{т.г.гр. е в } \mathcal{F}}) = \underbrace{\underset{n}{lub}\ \Gamma(f_n)}_{\text{т.г.гр. е в } \mathcal{F}'}. \tag{2.7}$$

Забележка. В горното равенство имаме предвид, че точната горна граница $lub\ \Gamma(f_n)$ на редицата в дясно $\{\Gamma(f_n)\}_n$ съществува и съответно е равна на $\Gamma(\underset{n}{lub}\ f_n)$.

Твърдение 2.7. Всеки непрекъснат оператор $\Gamma\colon \mathcal{F} \longrightarrow \mathcal{F}'$ е монотонен.

Доказателство. Да фиксираме $f, g \in \mathcal{F}$, такива че $f \leqslant g$. Трябва да покажем, че $\Gamma(f) \leqslant' \Gamma(g)$. Да разгледаме монотонно растящата редица

$$f \leqslant g \leqslant g \leqslant \ldots,$$

чиято т.г. граница очевидно е g. Прилагаме определението за непрекъснатост на Γ към тази редица. Така получаваме, че $\Gamma(g)$ ще е точна горна граница на редицата

$$\Gamma(f), \Gamma(g), \Gamma(g) \ldots$$

Но тази редица има точно два различни члена — $\Gamma(f)$ и $\Gamma(g)$. Това означава, че $\Gamma(f) \leqslant' \Gamma(g)$ и значи Γ наистина е монотонен. □

Да отбележим, че ако приложим *монотонния* оператор $\Gamma\colon \mathcal{F} \longrightarrow \mathcal{F}'$ към елементите на монотонно растящата редица

$$f_0 \leqslant f_1 \leqslant \ldots$$

с граница f, то той ще запази наредбата на нейните елементи, т.е. ще имаме

$$\Gamma(f_0) \leqslant' \Gamma(f_1) \leqslant' \ldots\ .$$

Но $\boldsymbol{\mathcal{F}'} = (\mathcal{F}',\ \leqslant',\ \Omega')$ е област на Скот и там наредбата $\leqslant'$ е пълна. Следователно редицата $\{\Gamma(f_n)\}_n$ ще има точна горна граница — да я означим с g.

Ние имаме, че за всяко n $f_n \leqslant f$ и следователно отново за всяко n $\Gamma(f_n) \leqslant' \Gamma(f)$. Последното означава, че $\Gamma(f)$ е горна граница за редицата $\{\Gamma(f_n)\}_n$, но g беше точната ѝ горна граница, и значи $\Gamma(f) \leqslant' g$.

Получихме, че и при произволна област на Скот за всеки монотонен оператор Γ е изпълнено

$$\underset{n}{lub}\ \Gamma(f_n)\ \leqslant'\ \Gamma(\underset{n}{lub}\ f_n).$$

Това, което в добавка ни дава *непрекъснатостта* на Γ е, че горното неравенство се превръща в равенство.

В бъдеще ще ни интересуват ОС, които са декартово произведение на други ОС. За изображенията в тях ще ни е необходима операцията *декартово произведение на оператори*. За целта да фиксираме $k+1$ области на Скот

$$\boldsymbol{\mathcal{F}} = (\mathcal{F},\ \leqslant,\ \Omega),\ \boldsymbol{\mathcal{F}_1} = (\mathcal{F}_1,\ \leqslant_1,\ \Omega_1),\ \ldots\ ,\ \boldsymbol{\mathcal{F}_k} = (\mathcal{F}_k,\ \leqslant_k,\ \Omega_k).$$

Да отбележим специално, че множествата $\mathcal{F}_1, \ldots \mathcal{F}_k$ са съвършено *произволни* и нямат връзка с означението $\mathcal{F}_i$ за множеството от всички частични функции на i аргумента.

Нека са ни дадени и k на брой оператора, действащи от $\mathcal{F}$ към всяко от множествата $\mathcal{F}_i$:

$$\Gamma_i\colon\ \mathcal{F} \longrightarrow\ \mathcal{F}_i,\ i = 1, \ldots, k.$$

Да дефинираме оператор $\Gamma\colon\ \mathcal{F}\ \longrightarrow\ \mathcal{F}_1 \times \cdots \times \mathcal{F}_k$ по следния начин:

$$\Gamma(f)\ \stackrel{\text{деф}}{=}\ (\Gamma_1(f), \ldots, \Gamma_k(f)).$$

Този оператор ще означаваме така:

$$\Gamma\ =\ \Gamma_1 \times \cdots \times \Gamma_k$$

и ще наричаме *декартово произведение* на операторите $\Gamma_1, \ldots, \Gamma_k$. Да отбележим, че съгласно *Твърдение* 2.5, декартовото произведение

$$\boldsymbol{\mathcal{F}_1} \times \cdots \times \boldsymbol{\mathcal{F}_k}$$

на областите на Скот $\boldsymbol{\mathcal{F}_1}, \ldots, \boldsymbol{\mathcal{F}_k}$ също е област на Скот и следователно можем да говорим за непрекъснатост на горния оператор Γ. Наредбата в домейна $\mathcal{F}_1 \times \cdots \times \mathcal{F}_k$ на тази ОС е покомпонентната наредба, така както я дефинирахме в раздел 2.2.1.

При горните означения:

Твърдение 2.8. Ако операторите $\Gamma_1, \ldots, \Gamma_k$ са непрекъснати, то тяхното декартово произведение $\Gamma = \Gamma_1 \times \cdots \times \Gamma_k$ също е непрекъснат оператор.

Доказателство. Да вземем произволна монотонно растяща редица в $\mathcal{F}$

$$f_0 \leqslant f_1 \leqslant \ldots.$$

Трябва да покажем, че $\Gamma(\underset{n}{lub}\ f_n) = \underset{n}{lub}\ \Gamma(f_n)$. Понеже всеки от операторите Γ_i е непрекъснат в ОС $\boldsymbol{\mathcal{F}_i}$, то за всяко $i = 1, \ldots, k$ ще е изпълнено

$$\Gamma_i(\underset{n}{lub}\ f_n) = \underset{n}{lub}\ \Gamma_i(f_n).$$

Тогава ще имаме последователно

$$\Gamma(\underset{n}{lub}\ f_n)\ \stackrel{\text{деф}\ \Gamma}{=}\ (\Gamma_1(\underset{n}{lub}\ f_n), \ldots, \Gamma_k(\underset{n}{lub}\ f_n)\ )\ =\ (\underset{n}{lub}\,\Gamma_1(f_n), \ldots, \underset{n}{lub}\,\Gamma_k(f_n))$$

$$\stackrel{\text{деф}\ lub}{=}\ \underset{n}{lub}\ \underbrace{(\Gamma_1(f_n), \ldots, \Gamma_k(f_n))}_{(\Gamma_1 \times \cdots \times \Gamma_k)(f_n)}\ =\ \underset{n}{lub}\ \underbrace{(\Gamma_1 \times \cdots \times \Gamma_k)}_{\Gamma}(f_n)\ = \underset{n}{lub}\ \Gamma(f_n).$$

Под "деф lub" по-горе имаме предвид условието (2.3), с което, както се убедихме в раздел 2.2.1, се дефинира точната горна граница в декартово произведение на ОС. □

2.3.2 Теорема на Кнастер–Тарски за произволна ОС

Да фиксираме една област на Скот $\boldsymbol{\mathcal{F}} = (\mathcal{F}, \leqslant, \Omega)$ и нека $\Gamma : \mathcal{F} \longrightarrow \mathcal{F}$ е изображение в носителя на тази ОС. Определенията за неподвижна и преднеподвижна точка, както и за най-малка (пред)неподвижна точка на Γ, които знаем от раздел 1.4, се пренасят директно.

Казваме, че $f \in \mathcal{F}$ е *неподвижна точка* на Γ, ако

$$\Gamma(f) = f.$$

Казваме, че $f \in \mathcal{F}$ е *преднеподвижна точка* на Γ, ако

$$\Gamma(f) \leqslant f.$$

Казваме, че f е *най-малка неподвижна точка* на оператора Γ, ако:

1) f е неподвижна точка на Γ;

2) за всяка неподвижна точка g на Γ е вярно, че $f \leqslant g$.

Казваме, че f е *най-малка преднеподвижна точка* на оператора Γ, ако:

1) f е преднеподвижна точка на Γ;

2) за всяка преднеподвижна точка g на Γ е вярно, че $f \leqslant g$.

Лесно се вижда, че ако съществува, най-малката неподвижна точка на Γ е единствена. Ще я означаваме отново с f_Γ. Нека имаме предвид, обаче, че когато множеството $\mathcal{F}$, в което действа Γ, е някакво декартово произведение $\mathcal{F}_1 \times \cdots \times \mathcal{F}_k$, то f_Γ всъщност ще е *вектор* $(f_1, \ldots, f_k)$.

Да формулираме и централния резултат за този раздел — обобщената Теорема на Кнастер-Тарски (или Кнастер-Тарски-Клини), чрез която ще дефинираме денотационна семантика.

Теорема 2.1. (Теорема на Кнастер-Тарски за произволна ОС) Нека $\boldsymbol{\mathcal{F}} = (\mathcal{F}, \leqslant, \Omega)$ е област на Скот, а $\Gamma : \mathcal{F} \longrightarrow \mathcal{F}$ е непрекъснат оператор. Тогава Γ притежава най-малка неподвижна точка f_Γ, за която е изпълнено:

$$f_\Gamma \;=\; \underset{n}{lub}\, \Gamma^n(\Omega).$$

В добавка, f_Γ е и най-малката преднеподвижна точка на оператора.

Доказателство. Разсъжденията следват тези от доказателството на *Теорема* 1.1 от раздел 1.4. Отново за по-кратко означаваме $\Gamma^n(\Omega)$ с f_n. Тогава от равенствата

$$f_{n+1} \stackrel{\text{деф}}{=} \Gamma^{n+1}(\Omega) = \Gamma(\Gamma^n(\Omega)) = \Gamma(f_n)$$

получаваме, че редицата $\{f_n\}_n$ удовлетворява рекурентната връзка

$$\begin{cases} f_0 = \Omega \\ f_{n+1} = \Gamma(f_n). \end{cases}$$

Тази редица е монотонно растяща, т.е. за всяко n имаме $f_n \leqslant f_{n+1}$ — да го проверим с бърза индукция по n. Наистина, при $n = 0$ това е така, защото $f_0 = \Omega$, и понеже Ω е най-малкият елемент на $\mathcal{F}$, то $\Omega \leqslant f_1$.

Сега ако приемем, че за някое $n \in \mathbb{N}$

$$f_n \leqslant f_{n+1},$$

то от това, че Γ е непрекъснат, ще имаме съгласно *Твърдение* 2.7, че той е и монотонен, и като го приложим към двете страни на горното неравенство, ще получим

$$\underbrace{\Gamma(f_n)}_{f_{n+1}} \leqslant \underbrace{\Gamma(f_{n+1})}_{f_{n+2}},$$

или все едно, $f_{n+1} \leqslant f_{n+2}$. Вече можем да твърдим, че редицата

$$f_0 \ \leqslant \ f_1 \leqslant \ \ldots$$

е монотонно растяща. Но понеже $(\mathcal{F}, \leqslant, \Omega)$ е област на Скот, там наредбата $\leqslant$ е пълна, и следователно тази редица има точна горна граница

$$g \ = \ \underset{n}{lub}\ f_n.$$

Ще покажем, че g е най-малката неподвижна точка на Γ.

Това, че g е неподвижна точка, се вижда от следната верига равенства:

$$\begin{aligned} \Gamma(g) \stackrel{\text{деф}}{=} \Gamma(\underset{n}{lub}\ f_n) \ &= \ \underset{n}{lub}\ \Gamma(f_n) \ = \ \overset{\infty}{\underset{n=0}{lub}}\ f_{n+1} \\ = \ \overset{\infty}{\underset{n=1}{lub}}\ f_n \ &= \ \overset{\infty}{\underset{n=0}{lub}}\ f_n \stackrel{\text{деф}}{=} g. \end{aligned}$$

За предпоследното равенство използвахме очевидния факт

$$lub\ \{f_1, f_2, \ldots\} \ = \ lub\ \{\Omega, f_1, f_2, \ldots\}.$$

Нека h е друга неподвижна точка на Γ, т.е. нека $\Gamma(h) = h$. Защо $g \leqslant h$? Достатъчно е да видим, че h е само *горна граница* на тази редица, с други думи, да видим, че

$$f_n \leqslant h \text{ за всяко } n \geq 0.$$

Това ще проверим отново с индукция относно n. За $n = 0$ очевидно

$$f_0 \stackrel{\text{деф}}{=} \Omega \leqslant h.$$

Да предположим, че за някое n, $f_n \leqslant h$. Но Γ е непрекъснат, в частност е монотонен, и оттук

$$\underbrace{\Gamma(f_n)}_{f_{n+1}} \leqslant \Gamma(h) = h, \tag{2.8}$$

или все едно, $f_{n+1} \leqslant h$.

Получихме, че за всяко n $f_n \leqslant h$, с други думи, h е мажоранта на редицата $\{f_n\}_n$. Следователно h мажорира и точната ѝ горна граница g, т.е. $g \leqslant h$.

Така получихме, че g е най-малката неподвижна точка на Γ, с други думи

$$f_\Gamma = g \stackrel{\text{деф}}{=} \operatorname*{lub}_n \Gamma^n(\Omega).$$

Да се убедим, че g е и най-малката *преднеподвижна* точка на оператора. Наистина, понеже $\Gamma(g) = g$, то в частност $\Gamma(g) \leqslant g$, и следователно g е и преднеподвижна точка на Γ. Сега да вземем друга преднеподвижна точка h на Γ. За нея по определение е изпълнено

$$\Gamma(h) \leqslant h.$$

Със същата рутинна индукция по n показваме, че за всяко n $f_n \leqslant h$, като за прехода от n към $n+1$ в (2.8) вместо $\Gamma(h) = h$ пишем $\Gamma(h) \leqslant h$. □

Да приложим току-що доказаната теорема към два различни оператора, свързани с една и съща рекурсивна програма, която разгледахме в началото на курса. Става въпрос за следната програма R:

R: $F(X, Y) =$ `if X = 0 then 0 else F(X − 1, F(X, Y))`

Като разсъждавахме операционно, тогава лесно съобразихме, че с *call by value* R ще пресметне функцията

$$f_{CV}(x, y) \simeq \begin{cases} 0, & \text{ако } x = 0 \\ \neg!, & \text{ако } x > 0, \end{cases}$$

докато с *call by name* — функцията

$$f_{CN} = \lambda x, y.0.$$

Сега ще се опитаме да погледнем на R денотационно. Все още не знаем какво формално означава това; в случая идеята ни е по-скоро да видим кои ще са операторите, определени от тялото на R във всяка от двете области на Скот $\boldsymbol{\mathcal{F}_2}$ и $\boldsymbol{\mathcal{F}_2^{\perp}}$, и съответно — кои ще са най-малките неподвижни точки на тези оператори

Първата област на Скот $\boldsymbol{\mathcal{F}_2} = (\mathcal{F}_2,\ \subseteq,\ \emptyset^{(2)})$ познаваме много добре. В нея R определя оператора

$$\Gamma\colon\ \mathcal{F}_2 \longrightarrow \mathcal{F}_2,$$

който се дефинира по следния начин:

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

Този оператор очевидно е компактен, и следователно — непрекъснат, така че към него можем да приложим теоремата на Кнастер-Тарски, според която за f_Γ е в сила представянето:

$$f_\Gamma \ = \bigcup_n \Gamma^n(\emptyset^{(2)}).$$

Да означим с f_n функцията $\Gamma^n(\emptyset^{(2)})$. Да напомним, че функциите от монотонно растящата редица

$$f_0 \subseteq f_1 \subseteq f_2 \ldots$$

имат смисъл на последователните приближения на f_Γ. Искаме да намерим *явния вид* на всяка f_n, а оттам — и на самата f_Γ.

По определение редицата $\{f_n\}_n$ удовлетворява рекурентната връзка

$$\left| \begin{array}{l} f_0 = \emptyset^{(2)} \\ f_{n+1} = \Gamma(f_n). \end{array} \right. \tag{2.9}$$

Така за първата апроксимация f_1 на f_Γ ще имаме:

$$f_1(x,y) \overset{(2.9)}{\simeq} \Gamma(\emptyset^{(2)})(x,y) \overset{\text{деф } \Gamma}{\simeq} \begin{cases} 0, & \text{ако } x = 0 \\ \emptyset^{(2)}(x-1, \emptyset^{(2)}(x,y)), & \text{ако } x > 0 \end{cases} \simeq \begin{cases} 0, & \text{ако } x = 0 \\ \neg!, & \text{ако } x > 0. \end{cases}$$

За следващата апроксимация f_2 получаваме:

$$f_2(x,y) \overset{(2.9)}{\simeq} \Gamma(f_1)(x,y) \overset{\text{деф } \Gamma}{\simeq} \begin{cases} 0, & \text{ако } x = 0 \\ f_1(x-1, \underbrace{f_1(x,y)}_{\neg!}), & \text{ако } x > 0 \end{cases} \overset{\text{деф } f_1}{\simeq} \begin{cases} 0, & \text{ако } x = 0 \\ \neg!, & \text{ако } x > 0. \end{cases}$$

Оказа се, че двете апроксимации f_1 и f_2 съвпаднаха. Това най-вероятно означава, че и всички следващи апроксимации ще са като f_1, защото от

$$f_1 \;=\; f_2,$$

прилагайки почленно Γ, ще получим

$$\Gamma(f_1) \;= \Gamma(f_2), \text{ т.е. } f_2 \;=\; f_3 \text{ и т.н.}$$

Формалното доказателство е с индукция: с индукция относно n ще покажем, че

$$f_n = f_1 \text{ за всяко } n \geq 1.$$

Случаят $n = 1$ е очевиден, а приемайки, че

$$f_n \;=\; f_1,$$

след почленно прилагане на Γ ще имаме

$$\Gamma(f_n) \;=\; \Gamma(f_1),$$

или все едно, $f_{n+1} \;=\; f_2$. Но вече знаем, че $f_2 = f_1$, откъдето веднага и $f_{n+1} = f_1$, с което индуктивната стъпка е проведена. Следователно редицата от последователните приближения на f_Γ изглежда така:

$$f_0 \stackrel{\text{деф}}{=} \emptyset^{(2)} \subset \; f_1 \;=\; f_2 \;= f_3 \ldots.$$

Ясно е, че границата на тази редица е f_1, и значи

$$f_\Gamma \;=\; f_1.$$

Оказа се, че всъщност $f_\Gamma \;=\; f_{CV}$, което, както вече имахме повод да коментираме, не е случайно.

Сега да видим как ще изглежда операторът, определен от програмата R, в другата област на Скот

$$\boldsymbol{\mathcal{F}_2^\perp} = (\mathcal{F}_2^\perp, \sqsubseteq, \Omega^{(2)}).$$

Да напомним, че с $\mathcal{F}_2^\perp$ отбелязвахме множеството от всички *тотални* функции на два аргумента в $\mathbb{N}_\perp \stackrel{\text{деф}}{=} \mathbb{N} \cup \{\perp\}$, т.е.

$$\mathcal{F}_2^\perp = \{f |\; f\colon\; \mathbb{N}_\perp^2 \longrightarrow \mathbb{N}_\perp\}.$$

Да означим с Δ оператора, който съответства на R в ОС $\boldsymbol{\mathcal{F}_2^\perp}$. Да дефинираме $\Delta\colon\; \mathcal{F}_2^\perp \longrightarrow \; \mathcal{F}_2^\perp$ по следния начин:

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

Каква е логиката да искаме $\Delta(g)(\bot, y) = \bot$? Да погледнем още веднъж към програмата R:

$R \quad F(\,X,\,Y) \;=\; \texttt{if}\;\; \texttt{X} \;=\; \texttt{0}\;\; \texttt{then}\;\; \texttt{0}\;\; \texttt{else}\;\; \texttt{F(}\,\texttt{X} - \texttt{1},\, \texttt{F(}\,\texttt{X},\,\texttt{Y)}\,\texttt{)}$

При извикването $F(\bot, y)$ влизаме в проверката $\bot = 0$, за която е логично да приемем, че има стойност $\bot$ (а не *false*). Това е т. нар. *естествено продължение* на базисните функции и предикати, за което ще говорим подробно по-нататък в курса.

Засега ще приемем наготово, че горният оператор Δ е непрекъснат в ОС $(\mathcal{F}_2^{\bot}, \sqsubseteq, \Omega^{(2)})$. Тогава теоремата на Кнастер-Тарски, приложена към Δ, ще ни даде, че

$$g_\Delta \;=\; \bigsqcup_n \Delta^n(\Omega^{(2)}).$$

За да намерим g_Δ, ще действаме както при другата ОС — ще намерим явния вид на апроксимациите $\Delta^n(\Omega^{(2)})$ на g_Δ. Нека за по-кратко ги означим е с g_n, т.е.

$$g_n \overset{\text{деф}}{=} \Delta^n(\Omega^{(2)}).$$

Редицата $\{g_n\}_n$ удовлетворява условията

$$\left|\begin{array}{l} g_0 = \Omega^{(2)} \\ g_{n+1} = \Delta(g_n). \end{array}\right. \tag{2.10}$$

Отново се стремим да намерим *явния вид* на всяка от функциите g_n.

За първата функция g_1 ще имаме:

$$g_1(x,y) \overset{(2.10)}{=} \Delta(\Omega^{(2)})(x,y) \overset{\text{деф } \Delta}{=} \begin{cases} 0, & \text{ако } x = 0 \\ \Omega^{(2)}(x-1, \Omega^{(2)}(x,y)), & \text{ако } x > 0 \\ \bot, & \text{ако } x = \bot \end{cases}$$

$$= \quad \begin{cases} 0, & \text{ако } x = 0 \\ \bot, & \text{ако } x > 0 \\ \bot, & \text{ако } x = \bot \end{cases} \quad = \quad \begin{cases} 0, & \text{ако } x = 0 \\ \bot, & \text{ако } x > 0 \;\vee\; x = \bot. \end{cases}$$

За следващата апроксимация g_2 получаваме последователно

$$g_2(x,y) \overset{(2.10)}{=} \Delta(g_1)(x,y) \overset{\text{деф } \Delta}{=} \begin{cases} 0, & \text{ако } x = 0 \text{ (за всяко } y, \text{ вкл. за } y = \bot) \\ g_1(x-1, g_1(x,y)), & \text{ако } x > 0 \\ \bot, & \text{ако } x = \bot \end{cases}$$

$$= \quad \begin{cases} 0, & \text{ако } x = 0 \\ g_1(0, \underbrace{g_1(1,y)}_{\bot}), & \text{ако } x = 1 \\ \bot, & \text{ако } x > 1 \\ \bot, & \text{ако } x = \bot \end{cases} \quad = \quad \begin{cases} 0, & \text{ако } x = 0 \\ 0, & \text{ако } x = 1 \\ \bot, & \text{ако } x > 1 \;\vee\; x = \bot. \end{cases}$$

Наблюдавайки какво се случи при $n = 2$, можем да направим предположение, че g_n ще изглежда по подобен начин:

$$g_n(x,y) = \begin{cases} 0, & \text{ако } x < n \text{ (\textbf{за всяко} } y \in \mathbb{N}_\bot) \\ \bot, & \text{ако } x \geq n \ \vee \ x = \bot, \end{cases}$$

Да го покажем с индукция относно n. Вече видяхме, че за $n = 0, 1, 2$ g_n наистина има този вид. Сега ако допуснем, че това е така и за произволна функция g_n, то за следващата g_{n+1} ще имаме:

$$g_{n+1}(x,y) \overset{(2.10)}{=} \Delta(g_n)(x,y) \overset{\text{деф } \Delta}{=} \begin{cases} 0, & \text{ако } x = 0 \\ g_n(x-1, g_n(x,y)), & \text{ако } x > 0 \\ \bot, & \text{ако } x = \bot \end{cases}$$

$$\overset{\text{и.х.}}{=} \begin{cases} 0, & \text{ако } x = 0 \\ 0, & \text{ако } x > 0 \ \& \ x - 1 < n \\ \bot, & \text{ако } x - 1 \geq n \\ \bot, & \text{ако } x = \bot \end{cases} = \begin{cases} 0, & \text{ако } x < n{+}1 \\ \bot, & \text{ако } x \geq n{+}1 \ \vee \ x = \bot. \end{cases}$$

с което индуктивната ни хипотеза се потвърди и за $n + 1$.

Сега като знаем как изглежда всяка от функциите g_n, не е трудно да съобразим, че $g_\Delta \overset{\text{деф}}{=} \sqcup g_n$ ще има вида

$$g_\Delta(x,y) \ = \ \begin{cases} 0, & \text{ако } x \in \mathbb{N} \\ \bot, & \text{ако } x = \bot. \end{cases}$$

Наистина, ако $x = \bot$, а $y \in \mathbb{N}_\bot$ — произволно, то за всяко n ще имаме, че $g_n(x,y) = \bot$, откъдето съгласно (2.6) и $g_\Delta(x,y) = \bot$.

Ако x е естествено число, тогава редицата от стойностите на $g_n(x,y)$ ще изглежда по този начин:

$$\underbrace{g_0(x,y)}_{\bot}, \ \ldots \ , \underbrace{g_x(x,y)}_{\bot}, \ \underbrace{g_{x+1}(x,y)}_{0}, \ \underbrace{g_{x+2}(x,y)}_{0}, \ \ldots$$

С други думи, редицата от стойностите е

$$\underbrace{\bot, \ \ldots \ , \bot}_{x+1 \text{ пъти}}, \ 0, \ 0, \ \ldots$$

и нейната граница очевидно е 0.

Ясно е, че функцията g_Δ няма как да е равна на f_{CN}, защото g_Δ е функция в $\mathbb{N}_\bot$, докато f_{CN} е функция в $\mathbb{N}$. Ако, обаче, "върнем" g_Δ в света на естествените числа, двете функции вече ще съвпадат. Как точно става това "ограничаване" ще разберем отново в раздел 3.5.

2.3.3 Теорема на Кнастер-Тарски за системи от уравнения

Теоремата на Кнастер-Тарски, която доказахме в предишния раздел, беше формулирана за оператори в *произволна* област на Скот $\boldsymbol{\mathcal{F}} = (\mathcal{F}, \leqslant, \Omega)$. Когато тази ОС е декартово произведение на други области (какъвто ще бъде най-типичният случай в бъдеще), тази теорема можем да преформулираме по един по-естествен начин.

За целта да фиксираме k отново на брой произволни ОС

$$\boldsymbol{\mathcal{F}_1} = (\mathcal{F}_1, \leqslant_1, \Omega_1), \ \ldots \ , \boldsymbol{\mathcal{F}_k} = (\mathcal{F}_k, \leqslant_k, \Omega_k).$$

Да означим с $\mathcal{F} = \mathcal{F}_1 \times \ldots \times \mathcal{F}_k$ декартово произведение на *носителите* на тези ОС, а с

$$\boldsymbol{\mathcal{F}} = (\mathcal{F}, \leqslant, \Omega)$$

— декартовото произведение на самите ОС. Тук релацията $\leqslant$ е стандартната покомпонентна наредба, индуцирана от наредбите $\leqslant_1, \ldots, \leqslant_k$, а $\Omega = (\Omega_1, \ldots, \Omega_k)$.

Съгласно *Твърдение* 2.5, $\boldsymbol{\mathcal{F}}$ също е ОС. Сега да вземем произволен оператор Γ в тази ОС:

$$\Gamma\colon \ \mathcal{F}_1 \times \ \ldots \ \times \mathcal{F}_k \ \longrightarrow \ \mathcal{F}_1 \times \ \ldots \ \times \mathcal{F}_k.$$

Тогава за всяка к-орка $(f_1, \ldots, f_k) \in \mathcal{F}$ съществува $(g_1, \ldots, g_k) \in \mathcal{F}$:

$$\Gamma(f_1, \ldots, f_k) \ = \ (g_1, \ldots, g_k).$$

За всяко $i = 1, \ldots, k$ да означим с

$$\Gamma_i : \ \mathcal{F}_1 \times \ \cdots \times \ \mathcal{F}_k \ \longrightarrow \ \mathcal{F}_i$$

оператора, който връща i-тата компонента на $\Gamma(f_1, \ldots, f_k)$, т.е.

$$\Gamma_i(f_1, \ldots, f_k) \ = \ g_i.$$

Тогава за всяка $(f_1, \ldots, f_k) \in \mathcal{F}$ ще имаме

$$\Gamma(f_1, \ldots, f_k) \ = \ (\Gamma_1(f_1, \ldots, f_k), \ \ldots \ , \Gamma_k(f_1, \ldots, f_k)),$$

което в нашите означения за декартово произведение на оператор може да се запише като

$$\Gamma \ = \ \Gamma_1 \times \ \ldots \ \times \Gamma_k.$$

Нека $\bar{f} = (f_1, \ldots, f_k)$ е неподвижна точка на Γ. Тогава означава, че $\Gamma(\bar{f}) = \bar{f}$, или все едно

$$(\Gamma_1(\bar{f}), \ \ldots \ , \Gamma_k(\bar{f})) = (f_1, \ldots, f_k).$$

Разписваме покомпонентно горното равенство и стигаме до k на брой равенства

$$\left|\begin{array}{rcl} \Gamma_1(f_1,\ldots,f_k) &=& f_1 \\ \vdots & & \\ \Gamma_k(f_1,\ldots,f_k) &=& f_k. \end{array}\right.$$

Можем да кажем, че $(f_1,\ldots,f_k)$ всъщност е *решение на системата от уравнения*

$$\left|\begin{array}{rcl} \Gamma_1(\mathbb{X}_1,\ldots,\mathbb{X}_k) &=& \mathbb{X}_1 \\ \vdots & & \\ \Gamma_k(\mathbb{X}_1,\ldots,\mathbb{X}_k) &=& \mathbb{X}_k. \end{array}\right. \tag{2.11}$$

Вярно е и обратното: ако $(f_1,\ldots,f_k)$ е решение на горната система, то $(f_1,\ldots,f_k)$ е неподвижна точка на Γ.
Да резюмираме:

$(f_1,\ldots,f_k)$ е неподвижна точка на $\Gamma = \Gamma_1 \times \cdots \times \Gamma_k$

точно когато $(f_1,\ldots,f_k)$ е решение на системата (2.11).

Аналогично се вижда, че

$(f_1,\ldots,f_k)$ е най-малка неподвижна точка на Γ

точно когато $(f_1,\ldots,f_k)$ е най-малко решение на системата (2.11).

Ако $\bar{f} = (f_1,\ldots,f_k)$ е преднеподвижна точка на Γ, то по определение $\Gamma(\bar{f}) \leqslant \bar{f}$, което ще рече

$$(\Gamma_1(\bar{f}),\ \ldots\ ,\Gamma_k(\bar{f})) \leqslant (f_1,\ \ldots\ f_k).$$

Горното неравенство е по отношение на покомпонентната наредба, което означава, че всъщност са в сила следните k на брой неравенства:

$$\left|\begin{array}{rcl} \Gamma_1(f_1,\ldots,f_k) &\leqslant_1& f_1 \\ \vdots & & \\ \Gamma_k(f_1,\ldots,f_k) &\leqslant_k& f_k. \end{array}\right.$$

или все едно — $(f_1,\ldots,f_k)$ е решение *на системата от неравенства*

$$\left|\begin{array}{rcl} \Gamma_1(\mathbb{X}_1,\ldots,\mathbb{X}_k) &\leqslant_1& \mathbb{X}_1 \\ \vdots & & \\ \Gamma_k(\mathbb{X}_1,\ldots,\mathbb{X}_k) &\leqslant_k& \mathbb{X}_k. \end{array}\right. \tag{2.12}$$

Отново можем да заключим, че $(f_1, \ldots, f_k)$ е преднеподвижна точка на $\Gamma = \Gamma_1 \times \cdots \times \Gamma_k$ точно когато $(f_1, \ldots, f_k)$ е решение на *системата от неравенства* (2.12), и съответно $(f_1, \ldots, f_k)$ е най-малката преднеподвижна точка на Γ точно когато $(f_1, \ldots, f_k)$ е най-малкото решение на системата (2.12).

При дефиниране на денотационна семантика на рекурсивна програма R ние ще се движим по-скоро по обратния път: на всяка рекурсивна дефиниция на R ще съпоставяме по един оператор Γ_i, а след това от тези оператори $\Gamma_1, \ldots, \Gamma_k$ ще образуваме тяхното декартово произведение Γ. При това ще ни е нужно да знаем как се изразява f_Γ чрез отделните оператори $\Gamma_1, \ldots, \Gamma_k$.

За да разберем как става това, да фиксираме произволни области на Скот

$$\boldsymbol{\mathcal{F}_1} = (\mathcal{F}_1,\ \leqslant_1,\ \Omega_1),\ \ldots\ ,\ \boldsymbol{\mathcal{F}_k} = (\mathcal{F}_k,\ \leqslant_k,\ \Omega_k).$$

Да положим $\mathcal{F} = \mathcal{F}_1 \times\ \ldots\ \times \mathcal{F}_k$, $\Omega = (\Omega_1, \ldots, \Omega_k)$ и да означим с $\boldsymbol{\mathcal{F}}$ тяхното декартово произведение:

$$\boldsymbol{\mathcal{F}} = (\mathcal{F},\ \leqslant, \Omega).$$

Нека са дадени непрекъснатите оператори

$$\Gamma_i :\ \mathcal{F}_1 \times\ \cdots \times\ \mathcal{F}_k\ \longrightarrow\ \mathcal{F}_i$$

за $i = 1, \ldots k$. Съгласно *Твърдение* 2.8, непрекъснат оператор ще е и тяхното декартово произведение $\Gamma = \Gamma_1 \times \cdots \times \Gamma_k$, който дефинираме по този начин:

$$\Gamma(\bar{f}) \stackrel{\text{деф}}{=} (\Gamma_1(\bar{f}), \ldots, \Gamma_k(\bar{f})).$$

Ясно е, че Γ е изображение от вида

$$\mathcal{F}_1 \times\ \cdots \times\ \mathcal{F}_k\ \longrightarrow\ \mathcal{F}_1 \times\ \cdots \times\ \mathcal{F}_k,$$

т.е. типът на входа и на изхода на Γ е един и същ, следователно можем да говорим за неподвижни точки на Γ. Както отбелязахме по-горе, този оператор е непрекъснат в ОС $\boldsymbol{\mathcal{F}}$. Следователно можем да приложим теоремата на Кнастер-Тарски за произволни ОС и да получим, че Γ има н.м.н.т. f_Γ, за която е изпълнено:

$$f_\Gamma\ =\ \underset{n}{lub}\ \Gamma^n(\Omega).$$

Да видим как се разписва това условие чрез операторите $\Gamma_1, \ldots, \Gamma_k$, с помощта на които дефинирахме Γ. (Ако си представяте f_Γ просто като $(f_{\Gamma_1}, \ldots, f_{\Gamma_k})$ — ами не е чак толкова просто ☺).

Удобно е отново да въведем означение за n-тата апроксимация $\Gamma^n(\Omega)$ на оператора Γ. Имаме, че $\Gamma^n(\Omega) \in \mathcal{F}_1 \times \ \cdots \times \mathcal{F}_k$, т.е тази апроксимация всъщност е вектор. Нека го означим с $(f_1^n, \ldots, f_k^n)$, т.е. имаме

$$\Gamma^n(\Omega) \ = \ (f_1^n, \ldots, f_k^n).$$

Редицата от последователните апроксимации $\{(f_1^n, \ldots, f_k^n)\}_n$ удовлетворява следната рекурентна схема:

$$(f_1^0, \ldots, f_k^0) \stackrel{\text{деф}}{=} \Gamma^0(\Omega) \ = \ \Omega \ = \ (\Omega_1, \ldots, \Omega_k)$$

$$(f_1^{n+1}, \ldots, f_k^{n+1}) \ = \ \Gamma(f_1^n, \ldots, f_k^n) \stackrel{\text{деф } \Gamma}{=} \ (\Gamma_1(f_1^n, \ldots, f_k^n), \ldots, \Gamma_k(f_1^n, \ldots, f_k^n)).$$

Тогава по всяка от компонентите $i = 1, \ldots, k$ ще имаме, че

$$\begin{vmatrix} f_i^0 \ = \ \Omega_i \\ f_i^{n+1} \ = \ \Gamma_i(f_1^n, \ldots, f_k^n). \end{vmatrix} \tag{2.13}$$

По-нататък, от доказателството на теоремата на Кнастер-Тарски за ОС знаем още, че последователните приближения $\Gamma^n(\Omega)$ на f_Γ образуват монотонно растяща редица. С други думи, имаме, че

$$(f_1^0, \ldots, f_k^0) \ \leqslant \ (f_1^1, \ldots, f_k^1) \ \leqslant \ldots,$$

откъдето по *Твърдение* 2.4 получаваме, че ще е монотонно растяща и редицата по всяка от компонентите. За i-тата компонента това означава, че редицата

$$f_i^0 \ \leqslant_i \ f_i^1 \ \leqslant_i \ f_i^2 \ \leqslant_i \ \ldots$$

е монотонно растяща в ОС $\boldsymbol{\mathcal{F}_i} = (\mathcal{F}_i, \ \leqslant_i \ \Omega_i)$, и следователно там тя ще има т.г. граница $\underset{n}{lub}\ f_i^n$, $1 \leq i \leq k$.

От *Твърдение* 2.4 знаем, че точната горна граница в ОС, която е декартово произведение, се дефинира по следния начин:

$$\underset{n}{lub}\ (f_1^n, \ldots, f_k^n) \ = \ (\underset{n}{lub}\ f_1^n, \ \ldots, \ \underset{n}{lub}\ f_k^n).$$

Най-малката неподвижна точка f_Γ също е вектор; да го означим така:

$$f_\Gamma \ = \ (f_\Gamma^1, \ldots, f_\Gamma^k).$$

Тогава ще имаме последователно

$$(f_\Gamma^1, \ldots, f_\Gamma^k) \ = \ \underset{n}{lub}\ \Gamma^n(\Omega) \ = \ \underset{n}{lub}\ (f_1^n, \ldots, f_k^n) \ = \ (\underset{n}{lub}\ f_1^n, \ldots, \underset{n}{lub}\ f_k^n).$$

Следователно f_Γ^i е точна горна граница на редицата, определена с рекурентната връзка (2.13), за всяко $i = 1, \ldots, k$.

Да резюмираме: когато Γ е декартово произведение от вида $\Gamma_1 \times \cdots \times \Gamma_k$, неговата най-малка неподвижна точка е векторът $f_\Gamma = (f_\Gamma^1, \ldots, f_\Gamma^k)$, в който i-тата функция f_Γ^i, $i = 1, \ldots, k$, е точна горна граница на редицата, определена с рекурентната схема (2.13).

2.3.4 Индуктивен принцип на Скот за произволна ОС

Ще завършим тази глава с обобщение на индуктивния принцип на Скот за случая на произволна ОС $\boldsymbol{\mathcal{F}} = (\mathcal{F}, \leqslant, \Omega)$.

Първо обобщаваме дефиницията за непрекъснатост. Свойството P в множеството $\mathcal{F}$ наричаме *непрекъснато*, ако за всяка монотонно растяща редица $f_0 \leqslant f_1 \leqslant \ldots$ в $\mathcal{F}$ е изпълнено условието:

$$\forall n\ P(f_n) \implies P(\underset{n}{lub}\ f_n).$$

Лесно се съобразява, че двете основни *Твърдение* 1.14 и *Твърдение* 1.17 за непрекъснати свойства, които доказахме за ОС $\boldsymbol{\mathcal{F}} = (\mathcal{F}_n, \subseteq, \emptyset^{(n)})$, остават в сила и за произволни ОС.

Твърдение 2.9. (Индуктивен принцип на Скот за произволна ОС) Нека $\boldsymbol{\mathcal{F}} = (\mathcal{F}, \leqslant, \Omega)$ е ОС, а $\Gamma\colon \mathcal{F} \longrightarrow \mathcal{F}$ е непрекъснат оператор. Нека още P е свойство в $\mathcal{F}$, за което са изпълнени условията:

1) $P(\Omega)$;

2) за всяка $f \in \mathcal{F}$ е вярно, че $P(f) \implies P(\Gamma(f))$;

3) свойството P е непрекъснато.

Тогава P е вярно за най-малката неподвижна точка f_Γ на оператора Γ.

Доказателство. По същество повтаря доказателството на *Твърдение* 1.12, но да го проведем все пак. Отново използваме представянето

$$f_\Gamma = \underset{n}{lub}\ \Gamma^n(\Omega).$$

от теоремата на Кнастер-Тарски за ОС. Нека

$$f_n = \Gamma^n(\Omega).$$

От доказателството на горната теорема знаем още, че редицата $f_0, f_1, \ldots$ е монотонно растяща. С индукция по n показваме, че $P(f_n)$ е вярно за всяко f_n.

При $n = 0$ по определение $f_0 = \Omega$ и верността на $P(f_0)$ следва от условието 1).

Да приемем, че за някое n е вярно $P(f_n)$. Но тогава съгласно 2) ще е вярно и $P(\Gamma(f_n))$, или все едно, ще е вярно $P(f_{n+1})$.

Така получихме, че за всяко n, $P(f_n)$ е в сила, откъдето поради непрекъснатостта на P ще имаме, че $P(\underset{n}{lub}\ f_n)$ също ще е в сила. Но $\underset{n}{lub}\ f_n = f_\Gamma$ и следователно $P(f_\Gamma)$ е налице. □

Глава 3

Семантики на рекурсивните програми

В тази глава ще въведем един много прост учебен език за функционално програмиране — езикът REC. За този език ще дефинираме операционна и денотационна семантика в двете версии — с предаване на параметрите по стойност и по име. Тук ще докажем и един от централните резултати в курса, а именно, че операционният и денотационният подход към рекурсивните програми водят до един и същ резултат.

3.1 Езикът REC

3.1.1 Синтаксис

Най-напред да разгледаме един пример за програма на този език:

$$\begin{array}{lcl} F(X,1) & & \texttt{where} \\ F(X,Y) & = & \texttt{if } X == 0 \ \texttt{ then } \ Y \ \texttt{ else } \ F(X-1, G(X,Y)) \\ G(X,Y) & = & \texttt{if } X == 0 \ \texttt{ then } \ 0 \ \texttt{ else } \ G(X-1,Y) + Y \end{array}$$

Елементите на езика REC включват:

- по една *константа* n за всяко $n \in \mathbb{N}$
- изброимо много *обектови* променливи $X_1, X_2, X_3, \ldots$
- изброимо много *функционални* променливи $F_1, F_2, F_3, \ldots$
- *базисни операции* op от множеството $\{+, *, -, <, ==, ||, \&\&, \ldots\}$. Ще предполагаме, че всички базисни операции са на два аргумента и това е единствено с цел да си спестим писането на индекси в доказателствата по-нататък.

Всяка функционална променлива F_i притежава своя *местност* (или *арност*) — естествено число, определящо броя на аргументите ѝ. Когато формулираме дефиниции, твърдения и пр., обикновено експлицитно ще казваме какъв е този брой, ако контекстът не го подсказва. Има и друг начин — да приемем, че за всяко i имаме цяла редица от променливи $F_i^1, F_i^2, \ldots, F_i^m, \ldots$, като всяка от променливите F_i^m е на m аргумента. Това, разбира се, ще е за сметка на претоварения запис, който ние се стремим да избягваме.

Определение 3.1. *Програмен терм* (или само *терм*) τ в езика REC дефинираме по следния начин:

$$\tau ::= n \mid X_i \mid (\tau\ op\ \tau) \mid \texttt{if}\ \tau\ \texttt{then}\ \tau\ \texttt{else}\ \tau \mid F_i(\tau, \ldots, \tau)$$

Да отбележим че зад горната Бекус-Наурова форма всъщност се крие следното индуктивно определение, което ще използваме в следващите дефиниции и доказателства:

1) За всяко $n \in \mathbb{N}$, константата n е терм.

2) За всяко $i = 1, 2, \ldots$, обектовата променлива X_i е терм.

3) Ако τ_1 и τ_2 са термове, а op е базисна операция, то $(\tau_1\ op\ \tau_2)$ е терм.

4) Ако τ_1, τ_2 и τ_3 са термове, то $\texttt{if}\ \tau_1\ \texttt{then}\ \tau_2\ \texttt{else}\ \tau_3$ е терм.

5) Ако F_i е m-местна функционална променлива, а $\tau_1, \ldots, \tau_m$ са термове, то $F_i(\tau_1, \ldots, \tau_m)$ е терм.

Примери: $5, \quad X_1, \quad 5X_1, \quad F_1(X_1, 5), \quad \texttt{if}\ X_1 > 5\ \texttt{then}\ X_2\ \texttt{else}\ F_1(X_2)$

По-нататък ще пишем $\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$, за да означим, че променливите на терма τ са измежду $X_1, \ldots, X_n, F_1, \ldots, F_k$.

3.1.2 Програми на езика REC

Определение 3.2. *Програма на езика REC* (или *рекурсивна програма*) е синтактичен обект от следния вид:

$$\begin{aligned}
&\tau_0(X_1, \ldots, X_n, F_1, \ldots, F_k) \qquad \texttt{where} \\
&F_1(X_1, \ldots, X_{m_1}) \;=\; \tau_1(X_1, \ldots, X_{m_1}, F_1, \ldots, F_k) \\
&\vdots \\
&F_k(X_1, \ldots, X_{m_k}) \;=\; \tau_k(X_1, \ldots, X_{m_k}, F_1, \ldots, F_k)
\end{aligned}$$

където $\tau_0, \ldots, \tau_k$ са програмни термове.

Терма $\tau_0(X_1, \ldots, X_n, F_1, \ldots, F_k)$ ще наричаме *глава* на програмата, а следващите k на брой равенства определят *тялото* на програмата.

Ще казваме, че термът τ_i задава *дефиницията* (или декларацията) на i-тата функционална променлива F_i. Всяка функционална променлива участва със своята дефиниция в програмата. На нея можем да гледаме като на система от k уравнения с k неизвестни — функционалните променливи на програмата.

Една програма можем да си представяме и в следния по-симетричен вид:

$$\begin{aligned} F_0(X_1, \ldots, X_n) &= \tau_0(X_1, \ldots, X_n, F_1, \ldots, F_k) \\ F_1(X_1, \ldots, X_{m_1}) &= \tau_1(X_1, \ldots, X_{m_1}, F_1, \ldots, F_k) \\ &\vdots \\ F_k(X_1, \ldots, X_{m_k}) &= \tau_k(X_1, \ldots, X_{m_k}, F_1, \ldots, F_k) \end{aligned}$$

като приемаме, че F_0 е *главната функция* (main function). Забележете, че симетрията не е пълна — в τ_0 не участва функционалната променлива F_0.

Идеята е, грубо казано, че тялото на програмата

$$\begin{aligned} F_1(X_1, \ldots, X_{m_1}) &= \tau_1(X_1, \ldots, X_{m_1}, F_1, \ldots, F_k) \\ &\vdots \\ F_k(X_1, \ldots, X_{m_k}) &= \tau_k(X_1, \ldots, X_{m_k}, F_1, \ldots, F_k) \end{aligned}$$

определя k на брой функции $f_1, \ldots, f_k$ (в смисъл, който ще уточняваме по-нататък), и тези функции, заместени в τ_0, задават n-местната функция, която тази програма пресмята.

Да определим термовете τ_i в примерната програма, с която започнахме:

$$\overbrace{F_1(X_1, 1)}^{\tau_0(X_1, F_1)} \quad \texttt{where}$$

$$F_1(X_1, X_2) = \underbrace{\texttt{if } X_1 == 0 \texttt{ then } X_2 \texttt{ else } F_1(X_1 - 1, F_2(X_1, X_2))}_{\tau_1(X_1, X_2, F_1, F_2)}$$

$$F_2(X_1, X_2) = \underbrace{\texttt{if } X_1 == 0 \texttt{ then } 0 \texttt{ else } F_2(X_1 - 1, X_2) + X_2}_{\tau_2(X_1, X_2, F_1, F_2)}$$

Да отбележим, че някои от термовете могат да имат фиктивни променливи, каквато е например F_1 в терма τ_2. За нас ще е важно да си представяме τ_1 и τ_2 като термове *и на двете* променливи F_1 и F_2, за да можем да съставим подходяща *система*, която съответства на рекурсивната програма. За да определим тази система, първо ще трябва да

определим понятието оператор, зададен чрез терм, или *термален оператор.*

3.2 Денотационна семантика с предаване на параметрите по стойност

3.2.1 Термални оператори

Нека $\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$ е произволен терм, в който всяка от функционалните променливи F_i е на m_i аргумента, $1 \leq i \leq k$. За да зададем *стойност* на τ, фиксираме n на брой естествени числа $x_1, \ldots, x_n$, с които да се заместят обектовите променливи $X_1, \ldots, X_n$ и k на брой функции $f_1 \in \mathcal{F}_{m_1}, \ldots, f_k \in \mathcal{F}_{m_k}$ — за функционалните променливи $F_1, \ldots, F_k$.

Стойността на терма τ в точката $(x_1, \ldots, x_n, f_1, \ldots, f_k)$ ще означаваме така:

$$\tau(x_1, \ldots, x_n, f_1, \ldots, f_k), \text{ или за по-кратко } - \tau(\bar{x}, \bar{f}).$$

Дефиницията на $\tau(\bar{x}, \bar{f})$ е с индукция по построението на терма τ:

Определение 3.3. 1) Ако τ е константата n, то $\tau(\bar{x}, \bar{f}) = n$.

2) Ако τ е обектовата променлива X_i, то $\tau(\bar{x}, \bar{f}) = x_i$.

3) Ако τ е от вида $(\tau_1 \; op \; \tau_2)$, то $\tau(\bar{x}, \bar{f}) \simeq \tau_1(\bar{x}, \bar{f}) \; op \; \tau_2(\bar{x}, \bar{f})$.

4) Ако τ е от вида `if` τ_1 `then` τ_2 `else` τ_3, то

$$\tau(\bar{x}, \bar{f}) \simeq \begin{cases} \tau_2(\bar{x}, \bar{f}), & \text{ако } \tau_1(\bar{x}, \bar{f}) > 0 \\ \tau_3(\bar{x}, \bar{f}), & \text{ако } \tau_1(\bar{x}, \bar{f}) \simeq 0 \\ \neg!, & \text{ако } \neg!\tau_1(\bar{x}, \bar{f}). \end{cases}$$

5) Ако τ е от вида $F_i(\tau_1, \ldots, \tau_{m_i})$, то $\tau(\bar{x}, \bar{f}) \simeq f_i(\tau_1(\bar{x}, \bar{f}), \ldots, \tau_{m_i}(\bar{x}, \bar{f}))$.

От дефиницията на стойност на израз от вида `if` τ_1 `then` τ_2 `else` τ_3 се вижда, че числовата константа 0 интерпретираме като булевата константа $false$, а всички положителни числа отъждествяваме с $true$. Това правим с цел да си спестим въвеждането на допълнителен булев тип. Разбира се, в задачите условията ще бъдат предикати ($=$, $\leq$, $\geq$, $\ldots$), които връщат булеви стойности.

Да вземем отново произволен терм $\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$, в който всяка от променливите F_i е на m_i аргумента. По един естествен начин този

терм определя оператор Γ_τ, който ще наричаме *термален оператор*. Този оператор ще е на k на брой аргумента $f_1, \ldots, f_k$ (които са с местност $m_1, \ldots, m_k$), а резултатът $\Gamma_\tau(f_1, \ldots, f_k)$ ще е n-местна функция. Ето точното определение:

Определение 3.4. Нека $\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$ е терм, в който всяка функционална променлива F_i е на m_i аргумента, $1 \leq i \leq k$. *Термалният оператор* $\Gamma_\tau : \mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k} \longrightarrow \mathcal{F}_n$ дефинираме както следва:

$$\Gamma_\tau(f_1, \ldots, f_k)(x_1, \ldots, x_n) \simeq \tau(x_1, \ldots, x_n, f_1, \ldots, f_k) \tag{3.1}$$

за всички $f_1 \in \mathcal{F}_{m_1}, \ldots f_k \in \mathcal{F}_{m_k}$ и $(x_1, \ldots, x_n) \in \mathbb{N}^n$.

Нашата близка цел бъде да докажем, че всеки термален оператор е компактен. Да напомним дефинициите за монотонност и компактност на оператор на много променливи от раздел 1.2.4.

Операторът $\Gamma \colon \mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k} \longrightarrow \mathcal{F}_n$ е *монотонен*, ако е вярно, че

$$f_1 \subseteq g_1 \ \& \ \ldots \ \& \ f_k \subseteq g_k \quad \Longrightarrow \quad \Gamma(f_1, \ldots, f_k) \subseteq \Gamma(g_1, \ldots, g_k)$$

за всички $(f_1, \ldots, f_k) \in \mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k}$ и $(g_1, \ldots, g_k) \in \mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k}$.

Операторът $\Gamma \colon \mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k} \longrightarrow \mathcal{F}_n$ наричаме *компактен*, ако за всяка $f_1 \in \mathcal{F}_{m_1}, \ldots, f_k \in \mathcal{F}_{m_k}$, за всяко $\bar{x} \in \mathbb{N}^n$ и за всяко $y \in \mathbb{N}$ е в сила еквивалентността:

$$\begin{aligned}\Gamma(f_1, \ldots, f_k)(\bar{x}) \simeq y \iff \exists\theta_1 \ldots \exists\theta_k(&\theta_1 \subseteq f_1 \ \& \ \ldots \ \& \ \theta_k \subseteq f_k \ \& \\ &\theta_1, \ldots, \theta_k \text{ са крайни } \& \ \Gamma(\theta_1, \ldots, \theta_k)(\bar{x}) \simeq y).\end{aligned}$$

От *Твърдение* 1.3 знаем, че един оператор е компактен точно когато е монотонен и краен. По определение операторът Γ е *краен*, ако е в сила правата посока на условието за компактност:

$$\begin{aligned}\Gamma(f_1, \ldots, f_k)(\bar{x}) \simeq y \implies \exists\theta_1 \ldots \exists\theta_k(&\theta_1 \subseteq f_1 \ \& \ \ldots \ \& \ \theta_k \subseteq f_k \ \& \\ &\theta_1, \ldots, \theta_k \text{ са крайни } \& \ \Gamma(\theta_1, \ldots, \theta_k)(\bar{x}) \simeq y).\end{aligned}$$

По-надолу ще пишем $\bar{f} \subseteq \bar{g}$ като съкращение за $f_1 \subseteq g_1 \ \& \ \ldots \ \& \ f_k \subseteq g_k$.

Твърдение 3.1. За всеки терм τ операторът Γ_τ е монотонен.

Доказателство. Отново ще предполагаме, че всяка от функционалните променливи F_i на терма $\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$ е на m_i аргумента, $1 \leq i \leq k$.

Избираме вектори от функции $\bar{f}$ и $\bar{g}$ от $\mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k}$, такива че $\bar{f} \subseteq \bar{g}$. Трябва да покажем, че $\Gamma_\tau(\bar{f}) \subseteq \Gamma_\tau(\bar{g})$. Последното означава да покажем, че за всяко $\bar{x} \in \mathbb{N}^n$ и y е изпълнена импликацията:

$$\Gamma_\tau(\bar{f})(\bar{x}) \simeq y \implies \Gamma_\tau(\bar{g})(\bar{x}) \simeq y.$$

Наистина, да фиксираме $\bar{x}$ и y и да предположим, че $\Gamma_\tau(\bar{f})(\bar{x}) \simeq y$. За да покажем, че и $\Gamma_\tau(\bar{g})(\bar{x}) \simeq y$, ще използваме индукция по построението на терма τ.

1) Ако τ е константата n, то $\Gamma_\tau(\bar{f})(\bar{x}) \overset{\text{деф } \Gamma_\tau}{\simeq} \tau(\bar{x}, \bar{f}) \overset{\text{деф}}{=} n = \Gamma_\tau(\bar{g})(\bar{x})$.

2) Ако τ е обектовата променлива X_i за някое $1 \leq i \leq n$, то

$$\Gamma_\tau(\bar{f})(\bar{x}) \overset{\text{деф } \Gamma_\tau}{\simeq} \tau(\bar{x}, \bar{f}) \overset{\text{деф}}{=} x_i = \Gamma_\tau(\bar{g})(\bar{x}).$$

3) Нека τ е от вида $(\tau_1 \; op \; \tau_2)$. Тогава

$$\Gamma_\tau(\bar{f})(\bar{x}) \overset{\text{деф } \Gamma_\tau}{\simeq} \tau(\bar{x}, \bar{f}) \overset{\text{деф}}{\simeq} \tau_1(\bar{x}, \bar{f}) \; op \; \tau_2(\bar{x}, \bar{f}) \simeq \Gamma_{\tau_1}(\bar{f})(\bar{x}) \; op \; \Gamma_{\tau_2}(\bar{f})(\bar{x}).$$

Ние имаме $\Gamma_\tau(\bar{f})(\bar{x}) \simeq y$, т.е.

$$\Gamma_{\tau_1}(\bar{f})(\bar{x}) \; op \; \Gamma_{\tau_2}(\bar{f})(\bar{x}) \simeq y.$$

Оттук, съгласно дефиницията на суперпозиция, *трябва* да съществуват естествени числа z_1 и z_2, такива че

$$\Gamma_{\tau_1}(\bar{f})(\bar{x}) \simeq z_1, \; \Gamma_{\tau_2}(\bar{f})(\bar{x}) \simeq z_2 \quad \text{и} \quad z_1 \; op \; z_2 \simeq y.$$

Но термовете τ_1 и τ_1 са построени *преди* τ и за тях индукционната хипотеза е в сила. Следователно $\Gamma_{\tau_1}(\bar{g})(\bar{x}) \simeq z_1$ и $\Gamma_{\tau_2}(\bar{g})(\bar{x}) \simeq z_2$, откъдето

$$\Gamma_\tau(\bar{g})(\bar{x}) \simeq \Gamma_{\tau_1}(\bar{g})(\bar{x}) \; op \; \Gamma_{\tau_1}(\bar{g})(\bar{x}) \overset{\text{и.х.}}{\simeq} z_1 \; op \; z_2 \simeq y.$$

4) Нека τ е от вида `if` τ_1 `then` τ_2 `else` τ_3.

Имаме по допускане, че $\Gamma_\tau(\bar{f})(\bar{x}) \simeq y$. Съгласно дефиницията на стойност на такъв терм, са възможни два случая:

1 сл. $\Gamma_{\tau_1}(\bar{f})(\bar{x}) > 0$ и $\Gamma_{\tau_2}(\bar{f})(\bar{x}) \simeq y$ или

2 сл. $\Gamma_{\tau_1}(\bar{f})(\bar{x}) \simeq 0$ и $\Gamma_{\tau_3}(\bar{f})(\bar{x}) \simeq y$.

Ще се спрем само на първия случай, тъй като другият е аналогичен на него. Понеже τ_1 и τ_2 са построени *преди* текущия терм τ, то по индуктивната хипотеза ще имаме, че $\Gamma_{\tau_1}(\bar{g})(\bar{x}) > 0$ и $\Gamma_{\tau_2}(\bar{g})(\bar{x}) \simeq y$, или все едно — $\tau_1(\bar{x}, \bar{g}) > 0$ и $\tau_2(\bar{x}, \bar{g}) \simeq y$. Но тогава съгласно дефиницията на стойност на условен терм ще е вярно, че $\tau(\bar{x}, \bar{g}) \simeq y$, или преписано чрез термалния оператор — $\Gamma_\tau(\bar{g})(\bar{x}) \simeq y$.

5) Последната възможност за τ е да е от вида $F_i(\tau_1, \ldots, \tau_{m_i})$. Тук

$$\Gamma_\tau(\bar{f})(\bar{x}) \overset{\text{деф } \Gamma_\tau}{\simeq} \tau(\bar{x}, \bar{f}) \overset{\text{деф}}{\simeq} f_i(\tau_1(\bar{x}, \bar{f}), \ldots, \tau_{m_i}(\bar{x}, \bar{f}))$$

$$\simeq\ f_i(\Gamma_{\tau_1}(\bar{f})(\bar{x}),\ldots,\Gamma_{\tau_{m_i}}(\bar{f})(\bar{x})).$$

По условие $\Gamma_\tau(\bar{f})(\bar{x}) \simeq y$, и следователно

$$f_i(\Gamma_{\tau_1}(\bar{f})(\bar{x}),\ldots,\Gamma_{\tau_{m_i}}(\bar{f})(\bar{x})) \simeq y.$$

От дефиницията за суперпозиция имаме, че *трябва* да съществуват естествени числа $z_1,\ldots,z_{m_i}$, такива че

$$\Gamma_{\tau_1}(\bar{f})(\bar{x}) \simeq z_1,\ldots,\ \Gamma_{\tau_{m_i}}(\bar{f})(\bar{x}) \simeq z_{m_i} \text{ и } f_i(z_1,\ldots,z_{m_i}) \simeq y.$$

Индукционната хипотеза за термовете $\tau_1,\ldots,\tau_{m_i}$ ни дава

$$\Gamma_{\tau_1}(\bar{g})(\bar{x}) \simeq z_1,\ldots,\ \Gamma_{\tau_{m_i}}(\bar{g})(\bar{x}) \simeq z_{m_i}.$$

Тук вземаме предвид, че по условие $f_i \subseteq g_i$ (впрочем, това е единственото място, където използваме, че $\bar{f} \subseteq \bar{g}$). Тогава от $f_i(z_1,...,z_{m_i}) \simeq y$ ще получим, че и $g_i(z_1,\ldots,z_{m_i}) \simeq y$. Следователно общо ще имаме:

$$\Gamma_{\tau_1}(\bar{g})(\bar{x}) \simeq z_1,\ldots,\ \Gamma_{\tau_{m_i}}(\bar{g})(\bar{x}) \simeq z_{m_i} \text{ и } g_i(z_1,\ldots,z_{m_i}) \simeq y.$$

Оттук

$$\Gamma_\tau(\bar{g})(\bar{x}) \simeq g_i(\Gamma_{\tau_1}(\bar{g})(\bar{x}),\ldots,\Gamma_{\tau_{m_i}}(\bar{g})(\bar{x})) \simeq y.$$

□

Твърдение 3.2. За всеки терм τ операторът Γ_τ е краен.

Доказателство. За да си спестим писането на двойни индекси нека приемем, че τ има само две функционални променливи F_1 и F_2, всяка от които е на аргументи m_1 и m_2, т.е. τ е от вида $\tau(X_1,\ldots,X_n,F_1,F_2)$.

Да вземем произволни $f \in \mathcal{F}_{m_1}, g \in \mathcal{F}_{m_2}, \bar{x} \in \mathbb{N}^n$ и $y \in \mathbb{N}$ и да приемем, че

$$\Gamma_\tau(f,g)(\bar{x}) \simeq y.$$

Трябва да покажем, че съществуват крайни функции $\theta \subseteq f$ и $\delta \subseteq g$, такива че

$$\Gamma_\tau(\theta,\delta)(\bar{x}) \simeq y.$$

Отново ще разсъждаваме с индукция, която следва построението на τ.

1) Ако τ е константата n, то

$$\Gamma_\tau(f,g)(\bar{x}) \overset{\text{деф } \Gamma_\tau}{\simeq} \tau(\bar{x},f,g)\ =\ n$$

и стойността n очевидно не зависи от f и g. Тук бихме могли да изберем $\theta = \emptyset^{(m_1)}$ и $\delta = \emptyset^{(m_2)}$. И двете функции са крайни, подфункции са на f и g и за тях също е изпълнено

$$\Gamma_\tau(\theta,\delta)(\bar{x})\ =\ n.$$

2) Случаят $\tau = X_i$ е аналогичен на горния, защото $\Gamma_\tau(f,g)(\bar{x}) = x_i$ и тази стойност отново не зависи от f и g.

3) Нека τ е от вида $(\tau_1\ op\ \tau_2)$. Тогава

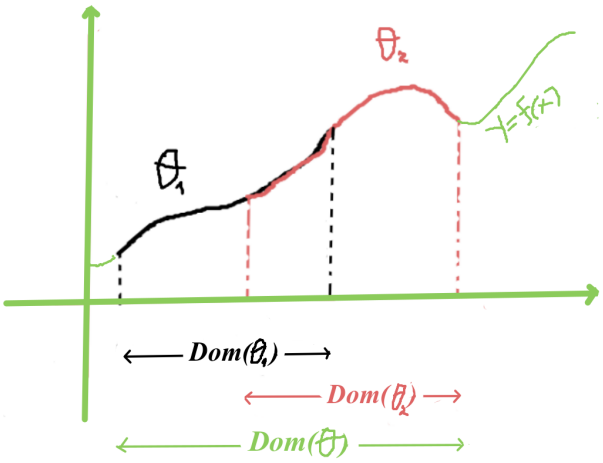
$$\Gamma_\tau(f,g)(\bar{x}) \ \simeq\ \Gamma_{\tau_1}(f,g)(\bar{x})\ op\ \Gamma_{\tau_2}(f,g)(\bar{x}) \simeq y.$$

Нека z_1 и z_2 са такива, че

$$\Gamma_{\tau_1}(f,g)(\bar{x}) \simeq z_1,\ \Gamma_{\tau_2}(f,g)(\bar{x}) \simeq z_2 \quad \text{и} \quad z_1\ op\ z_2\ \simeq y.$$

От индукционната хипотеза за τ_1 и τ_2 следва, че съществуват крайни функции $\theta_1 \subseteq f,\ \delta_1 \subseteq g,\ \theta_2 \subseteq f$ и $\delta_2 \subseteq g$, такива че

$$\Gamma_{\tau_1}(\theta_1,\delta_1)(\bar{x}) \simeq z_1 \quad \text{и} \quad \Gamma_{\tau_2}(\theta_2,\delta_2)(\bar{x}) \simeq z_2.$$



Да положим

$$\theta := \theta_1 \cup \theta_2 \quad \text{и} \quad \delta := \delta_1 \cup \delta_2.$$

Тук имаме предвид, че θ и δ са функциите, чиито графики са *обединение* на графиките на θ_1, θ_2 и δ_1, δ_2, съответно.

Тъй като $G_{\theta_1} \subseteq G_f$ и $G_{\theta_2} \subseteq G_f$, то $G_{\theta_1} \cup G_{\theta_2} \subseteq G_f$, т.е. $G_\theta \subseteq G_f$. Последното включване означава, че θ е подфункция на f, като разбира се, θ отново е крайна функция. По същия начин се вижда, че и δ е крайна подфункция на g.

Сега разсъждението продължава така: от индукционната хипотеза за τ_1 имаме, че $\Gamma_{\tau_1}(\theta_1,\delta_1)(\bar{x}) \simeq z_1$. Освен това очевидно $\theta_1 \subseteq \theta$ и $\delta_1 \subseteq \delta$. Но от предишното твърдение имаме, че Γ_{τ_1} е монотонен, и значи и $\Gamma_{\tau_1}(\theta,\delta)(\bar{x}) \simeq z_1$. Аналогично показваме, че $\Gamma_{\tau_2}(\theta,\delta)(\bar{x}) \simeq z_2$. Като отчетем и факта, че $z_1\ op\ z_2\ \simeq y$, получаваме общо

$$\Gamma_\tau(\theta,\delta)(\bar{x}) \ \simeq\ \Gamma_{\tau_1}(\theta,\delta)(\bar{x})\ op\ \Gamma_{\tau_2}(\theta,\delta)(\bar{x}) \ \simeq\ z_1\ op\ z_2\ \simeq\ y.$$

4) Когато τ е от вида **if** τ_1 **then** τ_2 **else** τ_3, разсъжденията са много подобни на горните.

5) Остана случаят, при който τ започва с функционална променлива. Нека за определеност това е F_1, т.е. τ е $F_1(\tau_1, \ldots, \tau_{m_1})$. В този случай

$$\Gamma_\tau(f,g)(\bar{x}) \simeq f(\Gamma_{\tau_1}(f,g)(\bar{x}), \ldots, \Gamma_{\tau_{m_1}}(f,g)(\bar{x})).$$

Ние приехме, че

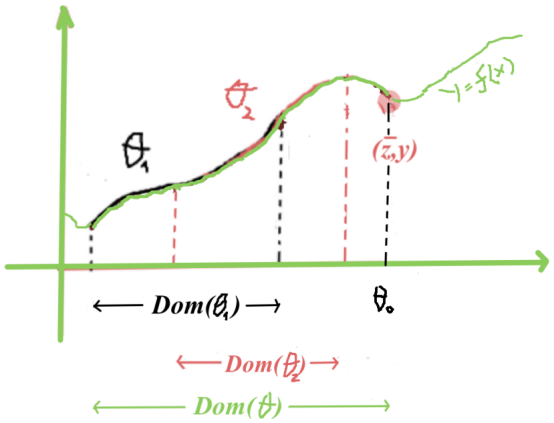
$$f(\Gamma_{\tau_1}(f,g)(\bar{x}), \ldots, \Gamma_{\tau_{m_i}}(f,g)(\bar{x})) \simeq y,$$

откъдето следва, че съществуват $z_1, \ldots, z_{m_1}$, такива че

$$\Gamma_{\tau_1}(f,g)(\bar{x}) \simeq z_1, \ \ldots \ , \Gamma_{\tau_{m_1}}(f,g)(\bar{x}) \simeq z_{m_1} \quad \text{и} \quad f(z_1, \ldots, z_{m_1}) \simeq y.$$

Прилагайки индукционното предположение за всеки от операторите $\Gamma_{\tau_1}, \ldots, \Gamma_{\tau_{m_1}}$, получаваме, че съществуват крайни функции
$\theta_1 \subseteq f, \ \delta_1 \subseteq g, \ \ldots \ , \theta_{m_1} \subseteq f, \ \delta_{m_1} \subseteq g$, такива че

$$\Gamma_{\tau_1}(\theta_1, \delta_1)(\bar{x}) \simeq z_1, \ \ldots \ , \Gamma_{\tau_{m_1}}(\theta_{m_1}, \delta_{m_1})(\bar{x}) \simeq z_{m_1}. \tag{3.2}$$



Крайните функции θ и δ, които искаме да конструираме, избираме почти както при случай 3) по-горе:

$$\theta := \underline{\theta_0} \cup \theta_1 \cup \cdots \cup \theta_{m_1} \quad \text{и} \quad \delta := \delta_1 \cup \cdots \cup \delta_{m_1}.$$

Разликата е в това, че към θ сме добавили още една крайна функция θ_0. Това е точно функцията с графика $\{(z_1, \ldots, z_{m_1}, y)\}$. Условието $f(z_1, \ldots, z_{m_1}) \simeq y$ означава точно, че $\theta_0 \subseteq f$, което заедно с

$$\theta_1 \subseteq f, \ \ldots \ , \theta_{m_1} \subseteq f$$

ни дава общо $\theta \subseteq f$. Разбира се, имаме и включването $\delta \subseteq g$.

От избора на θ и δ се вижда още, че

$$\theta_1 \subseteq \theta, \ldots, \theta_{m_1} \subseteq \theta \quad \text{и} \quad \delta_1 \subseteq \delta, \ldots, \delta_{m_1} \subseteq \delta.$$

Оттук, като вземем предвид монотонността на операторите $\Gamma_{\tau_1}, \ldots, \Gamma_{\tau_{m_1}}$ и равенствата (3.2), получаваме

$$\Gamma_{\tau_1}(\theta, \delta)(\bar{x}) \simeq z_1, \ \ldots \ , \Gamma_{\tau_{m_1}}(\theta, \delta)(\bar{x}) \simeq z_{m_1}.$$

Освен това се погрижихме да осигурим $\theta(z_1, \ldots, z_{m_1}) \simeq y$. Сега вече

$$\Gamma_\tau(\theta, \delta)(\bar{x}) \stackrel{\text{деф}}{\simeq} \theta(\Gamma_{\tau_1}(\theta, \delta)(\bar{x}), \ldots, \Gamma_{\tau_{m_1}}(\theta, \delta)(\bar{x})) \simeq \theta(z_1, \ldots, z_{m_1}) \simeq y,$$

което и трябваше да покажем.

□

Сега обединяваме *Твърдение* 3.1 и *Твърдение* 3.2 и получаваме

Твърдение 3.3. За всеки терм τ операторът Γ_τ е компактен.

3.2.2 Как дефинираме $D_V(R)$?

Настъпи дългоочакваният момент да дефинираме формална семантика на програмите от нашия език REC посредством най-малки неподвижни точки (*fixpoint semantics*). Тази семантика обикновено се нарича *денотационна семантика*. Ние ще ѝ викаме денотационна семантика *по стойност*, защото както ще видим в следващия раздел, тя съответства на операционната семантика с предаване на параметрите *по стойност*.

Да фиксираме произволна програма R от езика REC:

$$\begin{aligned} &\tau_0(X_1, \ldots, X_n, F_1, \ldots, F_k) \qquad \texttt{where} \\ &F_1(X_1, \ldots, X_{m_1}) = \tau_1(X_1, \ldots, X_{m_1}, F_1, \ldots, F_k) \\ &\vdots \\ &F_k(X_1, \ldots, X_{m_k}) = \tau_k(X_1, \ldots, X_{m_k}, F_1, \ldots, F_k) \end{aligned}$$

Всеки от термовете $\tau_i(X_1, \ldots, X_{m_i}, F_1, \ldots, F_k)$ определя оператор Γ_{τ_i} от тип $(m_1, \ldots, m_k \ \to \ m_i)$, т.е.

$$\Gamma_{\tau_i} : \mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k} \ \longrightarrow \ \mathcal{F}_{m_i}.$$

Да означим с $\Gamma = \Gamma_{\tau_1} \times \cdots \times \Gamma_{\tau_k}$ декартовото произведение на термалните оператори $\Gamma_{\tau_1}, \ldots, \Gamma_{\tau_k}$. Току-що се убедихме (*Твърдение* 3.3), че тези оператори са компактни, което съгласно *Твърдение* 1.9 означава, че те са и непрекъснати. Но тогава според *Твърдение* 2.8 и Γ ще е непрекъснат като декартово произведение на непрекъснати оператори. По определение

$$\Gamma(f_1, \ldots, f_k) \stackrel{\text{деф}}{=} (\Gamma_1(f_1, \ldots, f_k), \ldots, \Gamma_k(f_1, \ldots, f_k)).$$

Следователно Γ е изображение от следния вид:

$$\Gamma \colon \mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k} \longrightarrow \mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k}.$$

Можем да говорим *за неподвижни точки* на Γ, тъй като входът и изходът на Γ са от един и същи тип. Знаем, че Γ е непрекъснат в областта на Скот

$$\boldsymbol{\mathcal{F}} = (\mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k}, \subseteq, (\emptyset^{(m_1)}, \ldots, \emptyset^{(m_k)})).$$

Да приложим към оператора Γ теоремата на Кнастер-Тарски за тази област на Скот. Така ще получим, че Γ има най-малка неподвижна точка

$$f_\Gamma = (f_\Gamma^1, \ldots, f_\Gamma^k).$$

Да напомним, че съгласно това, което изведохме в раздел 2.3.3, векторът $(f_\Gamma^1, \ldots, f_\Gamma^k)$ се явява най-малко решение на системата

$$\left|\begin{array}{l} \mathbb{X}_1 = \Gamma_{\tau_1}(\mathbb{X}_1, \ldots, \mathbb{X}_k) \\ \vdots \\ \mathbb{X}_k = \Gamma_{\tau_k}(\mathbb{X}_1, \ldots, \mathbb{X}_k). \end{array}\right.$$

Чрез това най-малко решение $(f_\Gamma^1, \ldots, f_\Gamma^k)$ определяме денотационната семантика по стойност на рекурсивната програма R.

Определение 3.5. *Денотационна семантика с предаване на параметрите по стойност* на програмата R е функцията

$$D_V(R) : \mathbb{N}^n \multimap \mathbb{N},$$

която се определя посредством равенството:

$$D_V(R)(x_1, \ldots, x_n) \simeq \tau_0(x_1, \ldots, x_n, f_\Gamma^1, \ldots, f_\Gamma^k)$$

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

3.2.3 Един пример

Да се върнем към примерната програма R, с която започнахме тази глава, и да пресметнем $D_V(R)$ по начина, който описахме току-що.

$$\begin{array}{lcl} F(X,1) & & \texttt{where} \\ F(X,Y) & = & \texttt{if } X=0 \texttt{ then } Y \texttt{ else } F(X-1,G(X,Y)) \\ G(X,Y) & = & \texttt{if } X=0 \texttt{ then } 0 \texttt{ else } G(X-1,Y)+Y \end{array}$$

Да означим с $\Gamma : \mathcal{F}_2 \times \mathcal{F}_2 \longrightarrow \mathcal{F}_2$ и $\Delta : \mathcal{F}_2 \times \mathcal{F}_2 \longrightarrow \mathcal{F}_2$ операторите, определени от дефинициите на F и G:

$$\Gamma(f,g)(x,y) \simeq \begin{cases} y, & \text{ако } x = 0 \\ f(x-1, g(x,y)), & \text{ако } x > 0 \end{cases}$$

$$\Delta(f,g)(x,y) \simeq \begin{cases} 0, & \text{ако } x = 0 \\ g(x-1,y)+y, & \text{ако } x > 0. \end{cases}$$

Нека $\mathbf{\Gamma} = \Gamma \times \Delta$ е декартовото произведение на Γ и Δ. Имаме, че

$$\mathbf{\Gamma} : \ \mathcal{F}_2 \times \mathcal{F}_2 \longrightarrow \mathcal{F}_2 \times \mathcal{F}_2$$

е непрекъснат и по теоремата на Кнастер-Тарски той има най-малка неподвижна точка (f^*, g^*). Да си спомним, че всяка от компонентите f^* и g^* на $f_{\mathbf{\Gamma}}$ се получава като граница на рекурентна редица, определена от равенствата (2.13). В случая конкретно ще имаме, че

$$f^* \ = \ \bigcup_n f_n, \qquad g^* \ = \ \bigcup_n g_n,$$

където функциите от редиците $f_0, f_1, \ldots$ и $g_0, g_1, \ldots$ се дефинират със следната взаимна рекурсия:

$$\left| \begin{array}{lcl} f_0 & = & \emptyset^{(2)} \\ f_{n+1} & = & \Gamma(f_n, g_n) \quad \text{и} \end{array} \right. \tag{3.3}$$

$$\left| \begin{array}{lcl} g_0 & = & \emptyset^{(2)} \\ g_{n+1} & = & \Delta(f_n, g_n). \end{array} \right. \tag{3.4}$$

Понеже операторът Δ не зависи от първия си аргумент f, е удобно първо да пресметнем функциите от редицата $\{g_n\}_n$:

$$g_1(x,y) \overset{(3.4)}{\simeq} \Delta(\emptyset^{(2)}, \emptyset^{(2)})(x,y) \overset{\text{деф } \Delta}{\simeq} \begin{cases} 0, & \text{ако } x = 0 \\ \emptyset^{(2)}(x-1,y)+y, & \text{ако } x > 0 \end{cases} \simeq \begin{cases} 0, & \text{ако } x = 0 \\ \neg !, & \text{ако } x > 0. \end{cases}$$

За следващата апроксимация g_2 ще имаме:

$$g_2(x,y) \overset{(3.4)}{\simeq} \Delta(f_1,g_1)(x,y) \overset{\text{деф } \Delta}{\simeq} \begin{cases} 1, & \text{ако } x=0 \\ g_1(x-1,y)+y, & \text{ако } x>0 \end{cases} \simeq \begin{cases} 0, & \text{ако } x=0 \\ 0+y, & \text{ако } x=1 \\ \neg!, & \text{ако } x>1, \end{cases}$$

което можем да препишем като:

$$g_2(x,y) \simeq \begin{cases} x.y, & \text{ако } x<2 \\ \neg!, & \text{иначе.} \end{cases}$$

Това ни подсказва, че g_n може би ще изглежда по този начин:

$$g_n(x,y) \simeq \begin{cases} x.y, & \text{ако } x<n \\ \neg!, & \text{иначе.} \end{cases} \tag{3.5}$$

Наистина, по-горе видяхме, че за началните стойности на n това е така. Сега ако допуснем, че за произволно n горното представяне (3.5) е в сила, то за $n+1$ ще имаме:

$$g_{n+1}(x,y) \overset{(3.4)}{\simeq} \Delta(f_n,g_n)(x,y) \overset{\text{деф } \Delta}{\simeq} \begin{cases} 1, & \text{ако } x=0 \\ g_n(x-1,y)+y, & \text{ако } x>0 \end{cases} \overset{(3.5)}{\simeq}$$

$$\begin{cases} 0, & \text{ако } x=0 \\ (x-1).y+y, & \text{ако } x>0 \ \&\ x-1<n \\ \neg!, & \text{ако } x-1 \geq n \end{cases} \simeq \begin{cases} 0, & \text{ако } x=0 \\ xy, & \text{ако } 0<x<n+1 \\ \neg!, & \text{ако } x \geq n+1 \end{cases} \simeq \begin{cases} xy, & \text{ако } x<n+1 \\ \neg!, & \text{ако } x \geq n+1. \end{cases}$$

Границата g^* на редицата $\{g_n\}_n$ е функцията $x.y$ (макар че формално g^* няма да ни трябва при определянето на $D_V(R)$).

Като знаем общия вид на g_n, можем да пристъпим към пресмятането на функциите от редицата $\{f_n\}_n$:

$$f_1(x,y) \overset{(3.3)}{\simeq} \Gamma(\emptyset^{(2)},\emptyset^{(2)})(x,y) \overset{\text{деф } \Gamma}{\simeq} \begin{cases} y, & \text{ако } x=0 \\ \emptyset^{(2)}(x-1,\emptyset^{(2)}(x,y)), & \text{ако } x>0 \end{cases} \simeq \begin{cases} y, & \text{ако } x=0 \\ \neg!, & \text{ако } x>0. \end{cases}$$

За следващата функция f_2 ще имаме:

$$f_2(x,y) \overset{(3.3)}{\simeq} \Gamma(f_1,g_1)(x,y) \overset{\text{деф } \Gamma}{\simeq} \begin{cases} y, & \text{ако } x=0 \\ f_1(x-1,\underbrace{g_1(x,y)}_{\neg!}), & \text{ако } x>0 \end{cases} \simeq \begin{cases} y, & \text{ако } x=0 \\ \neg!, & \text{ако } x>0. \end{cases}$$

Получихме, че $f_1 = f_2$, което обаче не означава, че непременно ще имаме $f_1 = f_2 = f_3 \ldots$, защото дефиницията на следващата апроксимация f_3 зависи и от g_2, а тя е различна от g_1. Да се убедим:

$$f_3(x,y) \overset{(3.3)}{\simeq} \Gamma(f_2,g_2)(x,y) \overset{\text{деф } \Gamma}{\simeq} \begin{cases} y, & \text{ако } x = 0 \\ f_2(x-1, g_2(x,y)), & \text{ако } x > 0 \end{cases}$$

$$\simeq \begin{cases} y, & \text{ако } x = 0 \\ \underbrace{g_2(1,y)}_{y}, & \text{ако } x = 1 \\ \neg!, & \text{ако } x > 1 \end{cases} \simeq \begin{cases} x!.y, & \text{ако } x < 2 \\ \neg!, & \text{ако } x \geq 2. \end{cases}$$

Очертава се хипотезата, че при $n \geq 2$ функцията f_n би трябвало да изглежда така:

$$f_n(x,y) \simeq \begin{cases} x!.y, & \text{ако } x < n-1 \\ \neg!, & \text{иначе.} \end{cases} \tag{3.6}$$

Наистина, експериментите ни по-горе потвърдиха, че f_2 и f_3 имат горния вид. Да приемем, че и за произволно $n \geq 2$ това е така. Тогава за $n+1$ ще имаме:

$$f_{n+1}(x,y) \overset{(3.3)}{\simeq} \Gamma(f_n,g_n)(x,y) \overset{\text{деф } \Gamma}{\simeq} \begin{cases} y, & \text{ако } x = 0 \\ f_n(x-1, g_n(x,y)), & \text{ако } x > 0 \end{cases}$$

$$\overset{(3.5)}{\simeq} \begin{cases} y, & \text{ако } x = 0 \\ f_n(x-1, x.y), & \text{ако } x > 0 \ \&\ \underbrace{x < n}_{x-1<n-1} \\ \neg!, & \text{в останалите случаи} \end{cases} \overset{(3.6)}{\simeq} \begin{cases} y, & \text{ако } x = 0 \\ (x-1)!(x.y), & \text{ако } x > 0 \ \&\ x < n \\ \neg!, & \text{в останалите случаи} \end{cases}$$

$$\simeq \begin{cases} x!.y, & \text{ако } x < n \\ \neg!, & \text{иначе.} \end{cases}$$

Сега като имаме общия вид (3.6) на всяка от функциите f_n, не е трудно да съобразим, че тяхната граница $f^* = \bigcup_n f_n$ ще е функцията $x!.y$. Тогава по дефиниция

$$D_V(R)(x) \simeq \tau_0(x,y,f^*,g^*) \simeq f^*(x,1) = x!.$$

3.3 Операционна семантика на програмите от езика REC

3.3.1 Правила за извод на опростявания

Да напомним, че записът

$$\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$$

означаваше, че τ е терм с обектови променливи, които са сред $X_1, \ldots, X_n$, и функционални променливи измежду $F_1, \ldots, F_k$. Тук ще ни интересуват един специален вид термове, в които не участват обектови променливи. Такива термове ще наричаме *функционални термове* и ще отбелязваме с $\mu, \mu_1, \mu_2, \ldots$.

Примери: $5, \quad F_1(5), \quad 5 \; op \; F_1(5),$
$\texttt{if } 5 \; op \; 10 \; \texttt{then } F_1(5) \; \texttt{else } F_1(F_2(10))$

Определение 3.6. *Опростяване* ще наричаме синтактичен израз от вида

$$\mu \; \rightarrow \; c,$$

където μ е *функционален терм,* а c е *константа.*

Мъглявата (засега) идея, която стои зад израза $\mu \; \rightarrow \; c$ е, че термът μ *се опростява (или се редуцира)* до константата c. Разбира се, това опростяване е спрямо фиксирана рекурсивна програма. Всъщност рекурсивните дефиниции от тялото на програмата ще определят правилата за опростяване.

Преди да дадем строгите дефиниции, да обясним с пример защо се интересуваме от функционалните термове и опростяванията. Да разгледаме рекурсивната програма за функцията $x!$, написана на езика REC:

$R:$ $\quad F(X) \quad \texttt{where}$
$\quad\quad F(X) \; = \; \texttt{if } X == 0 \; \texttt{ then } \; 1 \; \texttt{ else } \; X.F(X-1)$

Когато искаме да видим какво пресмята тази програма да речем при $X = 3$, тръгваме от *функционалния терм* $F(3)$ и се опитваме чрез поредица от преобразования да достигнем до резултат, който е число. Преобразованията, които правим, изглеждат така:

$$F(3) \; \rightarrow \; 3.F(2) \; \rightarrow \; 3.2.F(1) \; \rightarrow \; 3.2.1.F(0) \; \rightarrow \; 6.$$

Можем да си представяме, че сме *опростили* функционалния терм $F(3)$ до константата 6 на базата на дефиницията на функционалната променлива F на програмата R. В такъв случай ще казваме, че от R *сме извели* опростяването $F(3) \; \rightarrow \; 6$.

Ако τ е терм, чийто обектови променливи са сред $X_1, \ldots, X_n$, а $\rho_1, \ldots, \rho_n$ са произволни термове, ще пишем

$$\tau[X_1/\rho_1,\ \ldots\ , X_n/\rho_n],$$

за да означим терма, който се получава, когато в τ *едновременно* се заместят променливите $X_1, \ldots, X_n$ с термовете $\rho_1, \ldots, \rho_n$ съответно. Изразът $\tau[X_1/\rho_1,\ \ldots\ , X_n/\rho_n]$ понякога ще съкращаваме до $\tau[\bar{X}/\bar{\rho}]$.

Формалната дефиниция е с индукция по построението на терма τ:

Определение 3.7. Нека τ е терм с обектови променливи измежду $X_1, \ldots, X_n$, а $\rho_1, \ldots, \rho_n$ са произволни термове. Тогава

1) ако $\tau = c$, то $\tau[X_1/\rho_1,\ \ldots\ , X_n/\rho_n] = c$;

2) ако $\tau = X_i$, то $\tau[X_1/\rho_1,\ \ldots\ , X_n/\rho_n] = \rho_i$;

3) ако $\tau = (\tau_1\ op\ \tau_2)$, то

$$\tau[\bar{X}/\bar{\rho}]\ =\ (\tau_1[\bar{X}/\bar{\rho}]\ op\ \tau_2[\bar{X}/\bar{\rho}]);$$

4) ако $\tau = \texttt{if}\ \tau_1\ \texttt{then}\ \tau_2\ \texttt{else}\ \tau_3$, то

$$\tau[\bar{X}/\bar{\rho}]\ =\ \texttt{if}\ \tau_1[\bar{\texttt{X}}/\bar{\rho}]\ \texttt{then}\ \tau_2[\bar{\texttt{X}}/\bar{\rho}]\ \texttt{else}\ \tau_3[\bar{\texttt{X}}/\bar{\rho}];$$

5) ако $\tau = F_i(\tau_1, \ldots, \tau_m)$, то

$$\tau[\bar{X}/\bar{\rho}]\ =\ F_i(\tau_1[\bar{X}/\bar{\rho}], \ldots, \tau_m[\bar{X}/\bar{\rho}]).$$

Да разгледаме няколко примера.

Примери. 1) Нека $\tau = F_1(X_1,\ F_1(X_1, X_2))$, $\rho_1 = F_2(X_2)$ и $\rho_2 = 5$. Тогава

$$\tau[X_1/F_2(X_2),\ X_2/5)]\ =\ F_1(F_2(X_2),\ F_1(F_2(X_2),\ 5)).$$

2) Нека $\tau(X_1, X_2)\ =\ X_1\ +\ X_2$. Да заместим синтактично променливите X_1 и X_2 с *константите* 5 и 10, съответно:

$$\tau[X_1/5,\ X_2/10]\ =\ 5 + 10.$$

Нека обърнем внимание, че резултатът от горното заместване е *синтактичният израз* $5 + 10$. Не го бъркайте със *семантичното понятие* стойност на терм. В случая за стойността на терма τ при $X_1 = 5$ и $X_2 = 10$ ще бъде

$$\tau(5, 10)\ =\ 5 + 10 = 15.$$

Ясно е, че ако в $\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$ заместим променливите $X_1, \ldots, X_n$ с функционалните термове $\mu_1, \ldots, \mu_n$, резултатът от заместването

$$\tau[X_1/\mu_1, \ \ldots \ , X_n/\mu_n]$$

отново ще е функционален терм. Доказателство на това наблюдение е със съвсем рутинна индукция по построението на терма τ, затова ще го пропуснем.

Сега да фиксираме произволна програма R от нашия език REC:

$$\begin{aligned}
&\tau_0(X_1, \ldots, X_n, F_1, \ldots, F_k) \qquad \texttt{where} \\
&F_1(X_1, \ldots, X_{m_1}) = \tau_1(X_1, \ldots, X_{m_1}, F_1, \ldots, F_k) \\
&\vdots \\
&F_i(X_1, \ldots, X_{m_i}) = \tau_i(X_1, \ldots, X_{m_i}, F_1, \ldots, F_k) \\
&\vdots \\
&F_k(X_1, \ldots, X_{m_k}) = \tau_k(X_1, \ldots, X_{m_k}, F_1, \ldots, F_k)
\end{aligned}$$

Определение 3.8. *Правила за синтактичен извод по стойност* от програмата R:

(0) $\quad c \ \to \ c \quad$ за всяка константа $c \in \mathbb{N}$.

(Четем: "c се опростява до c".)

(1) $\quad$ Ако $\quad \mu_1 \ \to \ c_1, \ \mu_2 \ \to \ c_2 \quad$ и $\quad c_1 \ op \ c_2 = c,$
$\quad$ то $\quad \mu_1 \ op \ \mu_2 \ \to \ c.$

(Ако μ_1 се опростява до c_1, μ_2 се опростява до c_2 и освен това $c_1 \ op \ c_2 = c$, то $\mu_1 \ op \ \mu_2$ се опростява до c.)

($2_{\mathbf{t}}$) $\quad$ Ако $\quad \mu_1 \ \to \ c_1, \ c_1 > 0 \quad$ и $\quad \mu_2 \ \to \ c,$
$\quad$ то $\quad \texttt{if } \mu_1 \texttt{ then } \mu_2 \texttt{ else } \mu_3 \ \to \ \texttt{c}.$

($2_{\mathbf{f}}$) $\quad$ Ако $\quad \mu_1 \ \to \ 0 \quad$ и $\quad \mu_3 \ \to \ c,$
$\quad$ то $\quad \texttt{if } \mu_1 \texttt{ then } \mu_2 \texttt{ else } \mu_3 \ \to \ \texttt{c}.$

(3_V) $\quad$ За всяко $1 \leq i \leq k$:
$\quad$ ако $\quad \mu_1 \to c_1, \ \ldots \ , \mu_{m_i} \to c_{m_i} \quad$ и $\quad \tau_i[X_1/c_1, \ldots, X_{m_i}/c_{m_i}] \to \ c,$
$\quad$ то $\quad F_i(\mu_1, ..., \mu_{m_i}) \ \to \ c.$

Да отбележим, че клаузата (0) всъщност е *аксиома*, а клаузите (1), ($2_{\mathbf{t}}$), ($2_{\mathbf{f}}$) и (3_V) са *правила*. Можем да считаме, че аксиомите също са правила, само че с 0 на брой предпоставки.

Определение 3.9. *Правила за синтактичен извод по име* от програмата R:

(0) $c \ \to \ c$ за всяка константа $c \in \mathbb{N}$.

(1) Ако $\mu_1 \ \to \ c_1,\ \mu_2 \ \to \ c_2$ и $c_1 \ op \ c_2 = c$,
то $\mu_1 \ op \ \mu_2 \ \to \ c$.

($2_\mathbf{t}$) Ако $\mu_1 \ \to \ c_1,\ c_1 > 0$ и $\mu_2 \ \to \ c$,
то $\texttt{if}\ \mu_1\ \texttt{then}\ \mu_2\ \texttt{else}\ \mu_3 \ \to \ \texttt{c}$.

($2_\mathbf{f}$) Ако $\mu_1 \ \to \ 0$ и $\mu_3 \ \to \ c$,
то $\texttt{if}\ \mu_1\ \texttt{then}\ \mu_2\ \texttt{else}\ \mu_3 \ \to \ \texttt{c}$.

(3_N) За всяко $1 \leq i \leq k$: ако $\tau_i[X_1/\mu_1, \ldots, X_{m_i}/\mu_{m_i}] \to c$,
то $F_i(\mu_1, \ldots, \mu_{m_i}) \ \to \ c$.

Правилата $(0), (1), (2_\mathbf{t})$ и $(2_\mathbf{f})$ са общи за двете системи за извод. Освен това те са едни и същи за всяка програма. Последните правила (3_V) и (3_N) вече зависят от конкретната програма R. При тях е и разликата между двете системи за извод. Правило (3_V) изисква параметрите да имат *стойности*, преди да бъдат подадени на съответната подпрограма, докато при правило (3_N) подаваме директно техните *имена* — термовете $\mu_1, \ldots, \mu_{m_i}$.

Горните правила могат да се запишат и по следния по-прегледен начин:

$$\frac{}{c \ \to \ c} \quad (0)$$

$$\frac{\mu_1 \ \to \ c_1, \quad \mu_2 \ \to \ c_2, \quad c_1 \ op \ c_2 = c}{\mu_1 \ op \ \mu_2 \ \to \ c} \quad (1)$$

$$\frac{\mu_1 \ \to \ c_1, \quad c_1 > 0, \quad \mu_2 \ \to \ c}{\texttt{if}\ \mu_1\ \texttt{then}\ \mu_2\ \texttt{else}\ \mu_3 \ \to \ \texttt{c}} \quad (2_\mathbf{t})$$

$$\frac{\mu_1 \ \to \ 0, \quad \mu_3 \ \to \ c}{\texttt{if}\ \mu_1\ \texttt{then}\ \mu_2\ \texttt{else}\ \mu_3 \ \to \ \texttt{c}} \quad (2_\mathbf{f})$$

$$\frac{\mu_1 \to c_1, \quad \ldots \quad , \ \mu_{m_i} \to c_{m_i}, \quad \tau_i[X_1/c_1, \ldots, X_{m_i}/c_{m_i}] \to c}{F_i(\mu_1, \ldots, \mu_{m_i}) \ \to \ c} \quad (3_V)$$

$$\frac{\tau_i[X_1/\mu_1, \ldots, X_{m_i}/\mu_{m_i}] \to c}{F_i(\mu_1, \ldots, \mu_{m_i}) \ \to \ c} \quad (3_N)$$

Пример. Ако $\mu_1 \ \to \ 5$ и $\mu_2 \ \to \ 10$, то понеже $5 + 10 = 15$, можем да твърдим на базата на правило (1), че $\mu_1 + \mu_2 \ \to \ 15$. Схематично:

$$\frac{\mu_1 \to 5, \quad \mu_2 \to 10, \quad 5 + 10 = 15}{\mu_1 + \mu_2 \ \to \ 15} \quad (1)$$

3.3.2 Как дефинираме $O_V(R)$ и $O_N(R)$?

Вече сме в състояние да дефинираме функциите $O_V(R)$ и $O_N(R)$ — *операционната семантика по стойност и по име* на програмата R. За целта въвеждаме няколко последни синтактични понятия.

Отново си мислим фиксирана произволна програма R от езика REC:

$$\begin{aligned}
&\tau_0(X_1,\ldots,X_n,F_1,\ldots,F_k) \qquad \texttt{where}\\
&F_1(X_1,\ldots,X_{m_1}) = \tau_1(X_1,\ldots,X_{m_1},F_1,\ldots,F_k)\\
&\vdots\\
&F_i(X_1,\ldots,X_{m_i}) = \tau_i(X_1,\ldots,X_{m_i},F_1,\ldots,F_k)\\
&\vdots\\
&F_k(X_1,\ldots,X_{m_k}) = \tau_k(X_1,\ldots,X_{m_k},F_1,\ldots,F_k)
\end{aligned}$$

Извод по стойност от програмата R ще наричаме редицата от опростявания

$$\mu_1 \;\to\; c_1, \quad \mu_2 \;\to\; c_2, \;\ldots\;, \mu_l \;\to\; c_l,$$

където всяко опростяване $\mu_i \to c_i$ е или аксиома, или се получава от *предишни* опростявания $\mu_j \to c_j$, $j < i$, на базата на някое от правилата $(1), (2_{\mathbf{t}}), (2_{\mathbf{f}})$ и (3_V).

Дължината l на тази редица ще наричаме *дължина на извода*. Ясно е, че ако дължината на извода е 1, то $\mu_1 \;\to\; c_1$ е аксиома и следователно $\mu_1 = c_1$.

Определение 3.10. Ще казваме, че от *от R се извежда по стойност* опростяването $\mu \;\to\; c$ и ще пишем

$$R \;\vdash_V\; \mu \;\to\; c,$$

ако съществува извод по стойност $\mu_1 \to c_1, \;\ldots\;, \mu_l \to c_l$, чийто последен член $\mu_l \to c_l$ е точно опростяването $\mu \to c$.

Аналогично въвеждаме и *извод по име* от R: това отново е редица от опростявания

$$\mu_1 \;\to\; c_1, \quad \mu_2 \;\to\; c_2, \;\ldots\;, \mu_l \;\to\; c_l,$$

където всяко $\mu_i \to c_i$ е или аксиома, или се получава от *предишни* опростявания $\mu_j \to c_j$, $j < i$, на базата на някое от правилата $(1), (2_{\mathbf{t}}), (2_{\mathbf{f}})$ и (3_N).

Определение 3.11. Ще казваме, че от *от R се извежда по име* опростяването $\mu \ \to \ c$ и ще пишем

$$R \ \vdash_N \ \mu \ \to \ c,$$

ако съществува извод по име $\mu_1 \ \to \ c_1, \ \ldots \ , \mu_l \ \to \ c_l$, чийто последен член е опростяването $\mu \ \to \ c$.

Определение 3.12. *Операционна семантика по стойност* на програмата R е функцията $O_V(R)\colon \ \mathbb{N}^n \multimap \mathbb{N}$, която се дефинира с еквивалентността:

$$O_V(R)(c_1, \ldots, c_n) \simeq d \iff R \ \vdash_V \ \tau_0[X_1/c_1, \ \ldots \ , X_n/c_n] \ \to \ d \quad (3.7)$$

за всички $c_1, \ldots, c_n, d \in \mathbb{N}$.

Аналогично определяме и другата операционна семантика на R:

Определение 3.13. *Операционна семантика по име* на програмата R е функцията $O_N(R)\colon \ \mathbb{N}^n \multimap \mathbb{N}$, която се дефинира с еквивалентността:

$$O_N(R)(c_1, \ldots, c_n) \simeq d \iff R \ \vdash_N \ \tau_0[X_1/c_1, \ \ldots \ , X_n/c_n] \ \to \ d$$

за всички $c_1, \ldots, c_n, d \in \mathbb{N}$.

Да обърнем специално внимание, че в горните определения $c_1, \ldots, c_n$ и d имат различен смисъл. Вляво те означават *числа*, а вдясно — *константи*, които участват в синтактично заместване. Разбира се, от контекста е съвсем ясно в какъв смисъл се употребяват те.

Операционните семантики, които въведохме, са от тип *"big step semantics"*. Характерното при тях е, че детайлите по извода не са уточнени и системата за извод е недетерминирана. Следователно че теоретично е възможно да бъдат изведени опростяванията

$$R \vdash_V \ \tau_0[X_1/c_1, \ \ldots \ , X_n/c_n] \ \to \ d \quad \text{и} \quad R \vdash_V \ \tau_0[X_1/c_1, \ \ldots \ , X_n/c_n] \ \to \ e,$$

като $d \neq e$, което означава, че $O_V(R)$ няма да е еднозначна функция. В следващия раздел ще покажем, че за всяка програма R

$$O_V(R) \subseteq D_V(R) \quad \text{и} \quad O_N(R) \subseteq D_N(R),$$

откъдето, в частност, ще следва, че $O_V(R)$ и $O_N(R)$ *са* еднозначни функции, тъй като $D_V(R)$ и $D_N(R)$ очевидно са такива.

3.3.3 Доказателство на включването $O_V(R) \subseteq O_N(R)$

Тук ще покажем включването $O_V(R) \subseteq O_N(R)$ или преразказано: *всичко, което се извежда по стойност, се извежда и по име.* Но първо да отбележим един очевиден факт, който формулираме като лема, на която ще се позоваваме многократно. Лемата ще изкажем за изводимостта *по име.*

Лема 3.1. Нека $R \vdash_N \mu \ \to \ c$ с дължина на извода l. Тогава:

0) Ако μ е константа, то $\mu = c$.

1) Ако μ е от вида $\mu_1 \ op \ \mu_2$, то съществуват константи c_1 и c_2, такива че *преди това* от R са били изведени по име опростяванията $\mu_1 \ \to \ c_1$ и $\mu_2 \ \to \ c_2$, и освен това $c_1 \ op \ c_2 = c$.
Тук под "*преди това*" имаме предвид, че *всяко* от тези две опростявания е имало дължина на извода, по-малка от l. Същата уговорка ще важи и за следващите подточки.

2) Ако μ е от вида $\texttt{if} \ \mu_1 \ \texttt{then} \ \mu_2 \ \texttt{else} \ \mu_3$, то е вярно едно от двете:
— преди това от R са били изведени по име опростяванията $\mu_1 \ \to \ c_1$ за някое $c_1 > 0$ и $\mu_2 \ \to \ c$ или
— преди това от R са били изведени по име $\mu_1 \ \to \ 0$ и $\mu_3 \ \to \ c$.

3) Ако μ е от вида $F_i(\mu_1, ..., \mu_{m_i})$, то преди това от R е било изведено по име опростяването $\tau_i[X_1/\mu_1, \ldots, X_{m_i}/\mu_{m_i}] \to c$.

Разбира се, такова твърдение е в сила и за изводимостта по стойност. Единствената разлика е в последния пункт 3), който изглежда така:
3) Ако μ е от вида $F_i(\mu_1, ..., \mu_{m_i})$, то за някои константи $c_1, \ldots, c_{m_i}$ преди това от R са били изведени по стойност опростяванията $\mu_1 \ \to \ c_1, \ \ldots \ , \mu_{m_i} \ \to \ c_{m_i}$ и $\tau_i[X_1/\mu_1, \ldots, X_{m_i}/\mu_{m_i}] \to c$.

Доказателство. Доказателството е съвсем очевидно от вида на правилата, но за да стане съвсем ясно, ще разгледаме първите два случая.
1) Ако μ е константа, то *единственият* начин да имаме $R \vdash_N \mu \ \to \ c$ е като сме използвали правилото (0) от *Дефиниция* 3.9, защото опростяванията $\mu \ \to \ c$, които стоят "под чертата" на останалите правила $(1), (2_{\mathbf{t}}), (2_{\mathbf{f}})$ и (3_N) се отнасят за термове μ, които очевидно не са константи. Но правилото (0) казва, че са изводими само опростявания от вида $c \ \to \ c$, и следователно $\mu = c$.
2) Ако μ е от вида $\mu_1 \ op \ \mu_2$, отново *единственият* начин от R да бъде изведено по име опростяването $\mu_1 \ op \ \mu_2 \ \to \ c$ е като се приложи правилото (1) от *Дефиниция* 3.9, защото само при него под чертата стои опростяване от този вид. Но правило (1) казва, че опростяването $\mu_1 \ op \ \mu_2 \ \to \ c$ може да бъде изведено само ако преди това са били изведени $\mu_1 \ \to \ c_1$ и $\mu_2 \ \to \ c_2$ за някои c_1 и c_2, такива че $c_1 \ op \ c_2 = c$. □

Сега се насочваме към доказателството на едно спомагателно твърдение, известно като *Лема за симулацията*. Чрез тази лема в края на този раздел лесно ще изведем важното включване $O_V(R) \subseteq O_N(R)$.

Лема 3.2. (Лема за симулацията) Нека $\rho(Y_1, \ldots, Y_n, F_1, \ldots, F_k)$ е произволен терм. Нека още $\mu_1, ..., \mu_n$ са функционални термове, $c_1, \ldots, c_n$ са константи и за тях е дадено, че

$$R \vdash_N \mu_1 \to c_1, \ \ldots \ , R \vdash_N \mu_n \to c_n.$$

Тогава

$$R \vdash_N \rho[Y_1/c_1, \ldots, Y_n/c_n] \to d \implies R \vdash_N \rho[Y_1/\mu_1, \ldots, Y_n/\mu_n] \to d. \quad (3.8)$$

Доказателство. Нека $R \vdash_N \rho[Y_1/c_1, \ldots, Y_n/c_n] \to d$. Ще разсъждаваме с пълна индукция по дължината l на извода на това опростяване, за да докажем, че $R \vdash_N \rho[Y_1/\mu_1, \ldots, Y_n/\mu_n] \to d$.
Фиксираме произволно $l \geq 1$ и приемаме, че за всички изводи по име на опростявания от вида $\rho_0[Y_1/c_1, \ldots, Y_n/c_n] \to d$ с дължина, по-малка от l, твърдението е вярно. Ще го докажем и за l. Разглеждаме различните възможности за терма ρ:

1) ρ е константата c. Тогава

   $$\rho[Y_1/c_1, \ldots, Y_n/c_n] = c \quad \text{и} \quad \rho[Y_1/\mu_1, \ldots, Y_n/\mu_n] = c$$

   и тогава импликацията (3.8) е очевидна.

2) ρ е обектовата променлива Y_i. Тогава $\rho[Y_1/c_1, \ldots, Y_n/c_n] \stackrel{\text{деф}}{=} c_i$.
   Ние имаме, че

   $$R \vdash_N \underbrace{\rho[Y_1/c_1, \ldots, Y_n/c_n]}_{c_i} \to d,$$

   или все едно, $R \vdash_N c_i \to d$, и значи $c_i = d$. Но по условие $R \vdash_N \mu_i \to c_i$, откъдето веднага получаваме $R \vdash_N \mu_i \to d$. Но това е точно условието $R \vdash_N \underbrace{\rho[Y_1/\mu_1, \ldots, Y_n/\mu_n]}_{\mu_i} \to d$, което искаме да покажем в този случай, защото $\rho[Y_1/\mu_1, \ldots, Y_n/\mu_n] = \mu_i$.

3) ρ е от вида $\rho_1 \ op \ \rho_2$. Тогава

   $$\rho[Y_1/c_1, \ldots, Y_n/c_n] \stackrel{\text{деф}}{=} \rho_1[Y_1/c_1, \ldots, Y_n/c_n] \ op \ \rho_2[Y_1/c_1, \ldots, Y_n/c_n].$$

   В този случай условието $R \vdash_N \rho[Y_1/c_1, \ldots, Y_n/c_n] \to d$ се преписва така:

   $$R \vdash_N \underbrace{\rho_1[Y_1/c_1, \ldots, Y_n/c_n]}_{\nu_1} \ op \ \underbrace{\rho_2[Y_1/c_1, \ldots, Y_n/c_n]}_{\nu_2} \ \to \ d.$$

Съгласно *Лема* 3.1, ще съществуват константи d_1 и d_2, такива че $d_1\ op\ d_2 = d$ и

$$R \vdash_N\ \rho_1[Y_1/c_1, \ldots, Y_n/c_n]\ \to\ d_1 \quad \text{и} \quad R \vdash_N\ \rho_2[Y_1/c_1, \ldots, Y_n/c_n]\ \to\ d_2,$$

като тези изводи са с дължина, по-малка от l. Тогава съгласно индуктивната хипотеза:

$$R \vdash_N\ \rho_1[Y_1/\mu_1, \ldots, Y_n/\mu_n]\ \to\ d_1 \quad \text{и} \quad R \vdash_N\ \rho_2[Y_1/\mu_1, \ldots, Y_n/\mu_n]\ \to\ d_2.$$

Оттук, като вземем предвид и това, че $d_1\ op\ d_2 = d$, прилагайки правилото (1) от от *Дефиниция* 3.9, достигаме до

$$R \vdash_N\ \rho_1[Y_1/\mu_1, \ldots, Y_n/\mu_n]\ op\ \rho_2[Y_1/\mu_1, \ldots, Y_n/\mu_n]\ \to\ d,$$

което в случая е точно $R \vdash_N\ \rho[Y_1/\mu_1, \ldots, Y_n/\mu_n]\ \to\ d$.

4) ρ е от вида $\texttt{if}\ \rho_1\ \texttt{then}\ \rho_2\ \texttt{else}\ \rho_3$. Този случай се разглежда аналогично на предишния.

5) ρ е от вида $F_i(\rho_1, \ldots, \rho_{m_i})$. В този случай от дефиницията за синтактично заместване имаме:

$$F_i(\rho_1, \ldots, \rho_{m_i})[Y_1/c_1, \ldots, Y_n/c_n]\ =\ F_i(\rho_1[Y_1/c_1, \ldots, Y_n/c_n], \ldots, \rho_{m_i}[Y_1/c_1, \ldots, Y_n/c_n]).$$

Нека за по-кратко терма вдясно съкратим до $F_i(\rho_1[\bar{Y}/\bar{c}], \ldots, \rho_{m_i}[\bar{Y}/\bar{c}])$. Тук имаме дадено, че

$$R \vdash_N\ F_i(\underbrace{\rho_1[\bar{Y}/\bar{c}]}_{\nu_1}, \ldots, \underbrace{\rho_{m_i}[\bar{Y}/\bar{c}]}_{\nu_{m_i}})\ \to\ d.$$

Ако си представим за момент аргументите на F_i като някакви термове $\nu_1, \ldots, \nu_{m_i}$, то ще имаме

$$R \vdash_N\ F_i(\nu_1, \ldots, \nu_{m_i})\ \to\ d.$$

Прилагайки отново *Лема* 3.1, можем да твърдим, че това е станало, защото *преди това* (т.е. за *по-малко* от l стъпки) е било изведено опростяването

$$R \vdash_N\ \tau_i[X_1/\nu_1, \ldots, X_{m_i}/\nu_{m_i}] \to d.$$

Но

$$\begin{aligned}\tau_i[X_1/\nu_1, \ldots, X_{m_i}/\nu_{m_i}]\ &=\ \tau_i[X_1/\rho_1[\bar{Y}/\bar{c}], \ldots, X_{m_i}/\rho_{m_i}[\bar{Y}/\bar{c}]] \\ &=\ \tau_i[X_1/\rho_1, \ldots, X_{m_i}/\rho_{m_i}][\bar{Y}/\bar{c}].\end{aligned}$$

Значи всъщност имаме, че

$$R \vdash_N \ \tau_i[X_1/\rho_1, \ldots, X_{m_i}/\rho_{m_i}][\bar{Y}/\bar{c}] \ \to \ d$$

и този извод е направен за по-малко от l стъпки. Тогава от индуктивната хипотеза получаваме

$$R \vdash_N \ \tau_i[X_1/\rho_1, \ldots, X_{m_i}/\rho_{m_i}][\bar{Y}/\bar{\mu}] \ \to \ d,$$

или все едно,

$$R \vdash_N \ \tau_i[X_1/\underbrace{\rho_1[\bar{Y}/\bar{\mu}]}_{\mu_1^*}, \ldots, X_{m_i}/\underbrace{\rho_{m_i}[\bar{Y}/\bar{\mu}]}_{\mu_{m_i}^*}] \ \to \ d.$$

Щом имаме горната изводимост, значи можем да приложим правилото (3_N) от от *Дефиниция* 3.9. Така получаваме

$$R \vdash_N \ F_i(\underbrace{\rho_1[\bar{Y}/\bar{\mu}]}_{\mu_1^*}, \ldots, \underbrace{\rho_{m_i}[\bar{Y}/\bar{\mu}]}_{\mu_{m_i}^*}) \ \to \ d,$$

или все едно,

$$R \vdash_N \ F_i(\rho_1, \ldots, \rho_{m_i})[\bar{Y}/\bar{\mu}] \ \to \ d.$$

Но $F_i(\rho_1, \ldots, \rho_{m_i})$ беше нашето ρ, следователно

$$R \vdash_N \ \rho[\bar{Y}/\bar{\mu}] \ \to \ d.$$

□

Вече сме в състояние да докажем обещаното по-горе важно твърдение:

Теорема 3.1. За всяка рекурсивна програма R

$$O_V(R) \subseteq O_N(R).$$

Доказателство. Вземаме произволна програма R. Трябва да покажем, че за произволни $c_1, \ldots, c_n, c$ е вярна импликацията:

$$R \vdash_V \ \tau_0[X_1/c_1, \ \ldots \ , X_n/c_n] \ \to \ c \quad \Longrightarrow \quad R \vdash_N \ \tau_0[X_1/c_1, \ \ldots \ , X_n/c_n] \ \to \ c.$$

Ясно е, че ще ни се наложи да покажем по-общата импликация

$$R \vdash_V \ \mu \ \to \ c \quad \Longrightarrow \quad R \vdash_N \ \mu \ \to \ c \tag{3.9}$$

за всеки функционален терм μ.

Отново разсъждаваме с пълна индукция по дължината l на извода по стойност на опростяването $\mu \ \to \ c$.

Приемаме, че за всички изводи с дължина, по-малка от l, е в сила условието (3.9) (и това е вярно за произволни опростявания $\mu \ \to \ c$).

Нека сега $R \vdash_V \ \mu \ \to \ c$ с дължина на извода l. Разглеждаме различните възможности за функционалния терм μ.

1) μ е константата d. Тогава $R \vdash_V \ \mu \ \to \ c$ ще означава $R \vdash_V \ d \ \to \ c$ и следователно $d = c$. Тогава очевидно $R \vdash_N \ d \ \to \ c$.

2) μ е обектова променлива. Този случай е невъзможен, защото μ е *функционален* терм.

3) μ е от вида $\mu_1 \ op \ \mu_2$. Тогава даденото ни $R \vdash_V \ \mu \ \to \ c$ означава

$$R \vdash_V \ \mu_1 \ op \ \mu_2 \ \to \ c.$$

Оттук по *Лема* 3.1 получаваме, че с дължина по-малка от l са били изведени

$$R \vdash_V \ \mu_1 \ \to \ c_1 \quad \text{и} \quad R \vdash_V \ \mu_2 \ \to \ c_2$$

за някои c_1 и c_2, такива че $c_1 \ op \ c_2 \ = c$. Сега по индукционната хипотеза ще имаме

$$R \vdash_N \ \mu_1 \ \to \ c_1 \quad \text{и} \quad R \vdash_N \ \mu_2 \ \to \ c_2,$$

което заедно с факта, че $c_1 \ op \ c_2 \ = c$ по правилото за извод (1) ни дава $R \vdash_N \ \mu \ \to \ c$.

4) μ е от вида $\texttt{if} \ \mu_1 \ \texttt{then} \ \mu_2 \ \texttt{else} \ \mu_3$. Разсъждаваме начин, много подобен на горния.

5) μ е от вида $F_i(\mu_1, \ldots, \mu_{m_i})$. Това е случаят, в който ще се възползваме от лемата за симулацията. Имаме, че

$$R \vdash_V \ F_i(\mu_1, \ldots, \mu_{m_i}) \ \to \ c$$

за l стъпки. Това означава, че преди това за по-малко от l стъпки трябва да сме извели опростяванията от условията над чертата на правило (3_V)

$$R \vdash_V \mu_1 \to c_1, \ \ldots \ , R \vdash_V \mu_{m_i} \to c_{m_i} \quad \text{и} \quad R \vdash_V \tau_i[X_1/c_1, \ldots, X_{m_i}/c_{m_i}] \to c.$$

Прилагаме индуктивната хипотеза и получаваме

$$R \vdash_N \mu_1 \to c_1, \ \ldots \ , R \vdash_N \mu_{m_i} \to c_{m_i} \quad \text{и} \quad R \vdash_N \tau_i[X_1/c_1, \ldots, X_{m_i}/c_{m_i}] \to c.$$

Сега вече можем да приложим Лемата за симулацията. Така ще получим

$$R \vdash_N \tau_i[X_1/\mu_1, \ldots, X_{m_i}/\mu_{m_i}] \to c.$$

Прилагайки правилото за извод (3_N), получаваме окончателно

$$R \vdash_N F_i(\mu_1, \ldots, \mu_{m_i}) \to c, \quad \text{т.е. } R \vdash_N \ \mu \ \to \ c.$$

□

3.4 Еквивалентност между денотационната и операционната семантика по стойност

Предстои ни да докажем една от най-важните теореми в нашия курс, която се изказва съвсем кратко: за всяка рекурсивна програма R:

$$O_V(R) \;=\; D_V(R).$$

Този резултат хвърля мост между двата съвършено различни подхода към определянето на формална семантика на рекурсивните програми, които разгледахме дотук. При денотационния поход функцията, която се определя от програмата R, в нашите означения — $D_V(R)$, се дефинира посредством най-малката неподвижна точка на подходящ непрекъснат оператор в подходяща област на Скот, *Определение* 3.5 (или еквивалентно, посредством най-малкото решение на системата от функционални уравнения 2.11, която този оператор задава).

От друга страна, функцията $O_V(R)$, която се пресмята операционно от програмата R, въведохме по един по-естествен начин — чрез правилата за синтактичен извод 3.9. Фактът, че тези два типа семантики съвпадат, е сам по себе си интересен и нетривиален. Но той също така е от изключителна важност за верификацията на рекурсивните програми, защото дава възможност да се използват техники от типа на μ-индуктивния принцип на Скот, които вървят за най-малки неподвижни точки. Този феномен — че функцията, която се пресмята от една рекурсивна програма може да се опише чрез подходяща най-малка неподвижна точка — позволява тези техники да се използват и за доказване на коректност на рекурсивни програми.

В следващия раздел ще докажем, че подобна характеризация е в сила и при предаване на параметрите *по име*.

Да покажем равенството на двете частични функции $D_V(R)$ и $O_V(R)$ означава да покажем двете включвания

$$O_V(R) \subseteq D_V(R) \quad \text{и} \quad D_V(R) \subseteq O_V(R).$$

Това ще направим в следващите два подраздела.

3.4.1 Доказателство на включването $O_V(R) \subseteq D_V(R)$

Ще започнем с една спомагателна лема, която, ще използваме многократно по-надолу.

Лема 3.3. Нека $\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$ е терм, в който всяка от функционалните променливи F_i е на m_i аргумента, $1 \leq i \leq k$. Нека още $c_1, \ldots, c_n$ са естествени числа, а $f_1, \ldots, f_k$ са частични функции, съответно на $m_1, \ldots, m_k$ аргумента. Тогава

$$\tau[X_1/c_1, \ \ldots \ , X_n/c_n](f_1, \ldots, f_k) \ \simeq \ \tau(c_1, \ldots, c_n, f_1, \ldots, f_k).$$

Доказателството на тази лема е със съвсем рутинна индукция по построението на терма τ, затова ще го пропуснем.

Да обърнем отново внимание, че в лявата страна на горното равенство $c_1, \ldots, c_n$ се употребяват като *константи*, докато в дясната страна — като *числа*. Термът $\tau[X_1/c_1, \ \ldots \ , X_n/c_n]$ вляво е *функционален*, т.е. терм без обектови променливи, и затова пресмятаме неговата стойност в точка от вида $(f_1, \ldots, f_k)$.

Пример. Нека $\tau(X, F)$ е $F(X) + X$. Да означим с S функцията "прибавяне на 1", т.е. $S(x) = x + 1$ за всяко x. Тогава

$$\tau(5, S) = S(5) + 5 = 11.$$

От друга страна, за стойността на функционалния терм $\tau[X/5]$в т. S ще имаме:

$$\tau[X/5](S) = (F(5) + 5)(S) = S(5) + 5 = 11.$$

Оттук до края на този раздел ще считаме, че е фиксирана произволна програма R от нашия език REC:

$$\begin{aligned}
&\tau_0(X_1, \ldots, X_n, F_1, \ldots, F_k) \qquad \texttt{where} \\
&F_1(X_1, \ldots, X_{m_1}) = \tau_1(X_1, \ldots, X_{m_1}, F_1, \ldots, F_k) \\
&\vdots \\
&F_i(X_1, \ldots, X_{m_i}) = \tau_i(X_1, \ldots, X_{m_i}, F_1, \ldots, F_k) \\
&\vdots \\
&F_k(X_1, \ldots, X_{m_k}) = \tau_k(X_1, \ldots, X_{m_k}, F_1, \ldots, F_k)
\end{aligned}$$

В раздел 3.2.2 показахме, че операторът $\Gamma \ = \ \Gamma_{\tau_1} \times \cdots \times \Gamma_{\tau_k}$ е непрекъснат в областта на Скот

$$\boldsymbol{\mathcal{F}} = (\mathcal{F}_{m_1} \times \cdots \times \mathcal{F}_{m_k}, \ \subseteq, \ (\emptyset^{(m_1)}, \ldots, \emptyset^{(m_k)}) \ )$$

и следователно той има най-малка неподвижна точка $\bar{f}_\Gamma = (f_\Gamma^1, \ldots, f_\Gamma^k)$.

Чрез $\bar{f}_\Gamma$ определихме функцията $D_V(R)$ — *денотационната семантика по стойност* на програмата R посредством равенството:

$$D_V(R)(c_1, \ldots, c_n) \simeq \tau_0(c_1, \ldots, c_n, f_\Gamma^1, \ldots, f_\Gamma^k)$$

за всички естествени $c_1, \ldots, c_n$.
По-надолу това равенство ще съкращаваме до

$$D_V(R)(\bar{c}) \simeq \tau_0(\bar{c}, \bar{f}_\Gamma).$$

В раздел 2.3.3 забелязахме, че вместо да говорим за неподвижни точки на оператора Γ, по-интуитивно е да си представяме решенията на следната система, определена от R:

$$\left|\begin{array}{l} \mathbb{X}_1 = \Gamma_{\tau_1}(\mathbb{X}_1, \ldots, \mathbb{X}_k) \\ \vdots \\ \mathbb{X}_k = \Gamma_{\tau_k}(\mathbb{X}_1, \ldots, \mathbb{X}_k). \end{array}\right. \tag{3.10}$$

Видяхме, че всяка неподвижна точка на Γ е решение на системата (3.10) и обратно, всяко решение на (3.10) е н.т. на оператора Γ. В частност, най-малката неподвижна точка $(f_\Gamma^1, \ldots, f_\Gamma^k)$ на Γ ще бъде най-малко решение на тази система.

Близката ни цел е да покажем включването $O_V(R) \subseteq D_V(R)$, което може да се изкаже неформално така: всяка стойност, която се извежда от R с правилата за извод, се получава на определен етап от итерацията на оператора $\Gamma = \Gamma_{\tau_1} \times \cdots \times \Gamma_{\tau_k}$.

Съгласно определението за подфункция, $O_V(R) \subseteq D_V(R)$, ако за всички естествени $c_1, \ldots, c_n$ и d е в сила импликацията

$$O_V(R)(\bar{c}) \simeq d \implies D_V(R)(\bar{c}) \simeq d,$$

или преписана чрез съответните дефиниции на $O_V(R)$ и $D_V(R)$:

$$R \vdash_V \tau_0[\bar{X}/\bar{c}] \to d \quad \Longrightarrow \quad \tau_0(\bar{c}, \bar{f}_\Gamma) \simeq d.$$

Съгласно *Лема* 3.3, стойността на израза вдясно можем да запишем и така:

$$\tau_0(\bar{c}, \bar{f}_\Gamma) \simeq \underbrace{\tau_0[\bar{X}/\bar{c}]}_{\mu}(\bar{f}_\Gamma).$$

Да означим за момент с μ функционалния терм $\tau_0[\bar{X}/\bar{c}]$. Тогава импликацията по-горе добива вида:

$$R \vdash_V \mu \to d \quad \Longrightarrow \quad \mu(\bar{f}_\Gamma) \simeq d. \tag{3.11}$$

Ясно е, че ще се наложи да докажем (3.11) не само за $\mu = \tau_0[\bar{X}/\bar{c}]$, а за *всеки* функционален терм $\mu(F_1, \ldots, F_k)$, чиито функционални променливи $F_1, \ldots, F_k$ са като тези на програмата R, т.е. имат местност $m_1, \ldots, m_k$, съответно.

В доказателството на импликацията (3.11), което ще проведем по-долу, никъде няма да използваме, че $\bar{f}_\Gamma$ е *най-малката* сред неподвижните точки на оператора Γ. Затова ще формулираме твърдението за *произволна* неподвижна точка $\bar{f}$ на Γ.

Твърдение 3.4. Нека $\mu(F_1, \ldots, F_k)$ е функционален терм, а $\bar{f} = (f_1, \ldots, f_k)$ е неподвижна точка на оператора $\Gamma = \Gamma_{\tau_1} \times \cdots \times \Gamma_{\tau_k}$. Тогава е в сила импликацията:

$$R \vdash_V \mu \ \to \ d \quad \Longrightarrow \quad \mu(\bar{f}) \simeq d. \tag{3.12}$$

Преди да сме пристъпили към доказателството на твърдението, да изкажем следствието, което получаваме от него при $\mu = \tau_0[\bar{X}/\bar{c}]$:

Следствие 3.1. За всяка рекурсивна програма R

$$O_V(R) \subseteq D_V(R).$$

Доказателство. Нека $R \vdash_V \mu \to d$ с дължина на извода l. Ще покажем, че $\mu(\bar{f}) \simeq d$, като разсъждаваме с пълна индукция по l.

Наистина, да приемем, че за всяко опростяване $\mu' \ \to \ d'$:

$$\text{ако } R \vdash_V \mu' \ \to \ d' \text{ с дължина на извода } l' < l, \quad \text{то } \mu'(\bar{f}) \simeq d'$$

(*индуктивната хипотеза*).

Разглеждаме различните възможности за функционалния терм μ.

1) μ е константа. Тогава $R \vdash_V \mu \ \to \ d$ може да е изпълнено само ако $\mu = d$ и в такъв случай $\mu(\bar{f}) \stackrel{\text{деф}}{=} d$.

2) μ е обектова променлива. Този случай е невъзможен, защото μ е функционален терм.

3) μ е от вида $\mu_1 \ op \ \mu_2$. Имаме по условие, че

   $$R \vdash_V \ \mu_1 \ op \ \mu_2 \ \to \ d,$$

   откъдето по *Лема* 3.1 получаваме, че с дължина на извода, по-малка от l, са били изведени условията над чертата на правило (1):

   $$R \vdash_V \ \mu_1 \ \to \ c_1 \quad \text{и} \quad R \vdash_V \ \mu_2 \ \to \ c_2$$

   за някои c_1 и c_2, такива че $c_1 \ op \ c_2 \ = d$. Прилагаме индукционната хипотеза към тези опростявания и получаваме, че

   $$\mu_1(\bar{f}) \simeq c_1 \quad \text{и} \quad \mu_2(\bar{f}) \simeq c_2.$$

Тогава от дефиницията за *стойност на терм* и факта, че $c_1\ op\ c_2\ = d$ ще имаме:

$$\mu(\bar{f})\ \overset{\text{деф}}{\simeq}\ \mu_1(\bar{f})\ op\ \mu_2(\bar{f})\ \simeq\ c_1\ op\ c_2 = d.$$

4) μ е от вида $\texttt{if}\ \mu_1\ \texttt{then}\ \mu_2\ \texttt{else}\ \mu_3$, разсъждаваме по много подобен начин, като разглеждаме двата случая, отговарящи на правилата $(2_{\mathbf{t}})$ и $(2_{\mathbf{f}})$.

5) Последната възможност за μ е да е от вида $F_i(\mu_1, \ldots, \mu_{m_i})$. Тук имаме

$$R \vdash_V\ F_i(\mu_1, \ldots, \mu_{m_i})\ \to\ d$$

с дължина на извода l. Прилагаме отново *Лема* 3.1 и получаваме, че преди това (т.е. с дължина на извода, по-малка от l) *трябва* да са били изведени условията в предпоставката на правило (3_V), т.е. за някои константи $c_1, \ldots, c_{m_i}$ са били изведени

$$R \vdash_V \mu_1 \to c_1,\ \ldots\ , R \vdash_V \mu_{m_i} \to c_{m_i} \quad \text{и} \quad R \vdash_V \tau_i[X_1/c_1, \ldots, X_{m_i}/c_{m_i}] \to d$$

с дължина на извода, по-малка от l. Прилагаме индуктивната хипотеза към тези опростявания и получаваме, че

$$\mu_1(\bar{f}) \simeq c_1,\ \ldots\ , \mu_{m_i}(\bar{f}) \simeq c_{m_i} \quad \text{и} \quad \tau_i[\bar{X}/\bar{c}](\bar{f}) \simeq d.$$

Следователно

$$\mu(\bar{f}) \overset{\text{деф}}{\simeq} f_i(\mu_1(\bar{f}), \ldots, \mu_{m_i}(\bar{f}))\ \simeq\ f_i(c_1, \ldots, c_{m_i}). \tag{3.13}$$

Този е моментът, в който ще използваме, че векторът $\bar{f} = (f_1, \ldots, f_k)$ е неподвижна точка на оператора $\Gamma = \Gamma_{\tau_1} \times \cdots \times \Gamma_{\tau_k}$. По дефиниция това означава, че $\Gamma(\bar{f}) = \bar{f}$, или разписано:

$$\Gamma(\bar{f})\ \overset{\text{деф}}{=}\ (\Gamma_{\tau_1}(\bar{f}), \ldots, \Gamma_{\tau_k}(\bar{f}))\ = (f_1, \ldots, f_k).$$

В частност, за нашето i ще имаме

$$\Gamma_{\tau_i}(\bar{f}) = f_i.$$

Оттук при $\bar{c} = (c_1, \ldots, c_{m_i})$ ще е изпълнено

$$\Gamma_{\tau_i}(\bar{f})(\bar{c}) \simeq f_i(\bar{c}). \tag{3.14}$$

Но колко е $\Gamma_{\tau_i}(\bar{f})(\bar{c})$? От определението на термален оператор (3.1) имаме

$$\Gamma_{\tau_i}(\bar{f})(\bar{c})\ \overset{\text{деф}}{\simeq}\ \tau_i(\bar{c}, \bar{f})\ \overset{\text{Лема 3.3}}{\simeq}\ \tau_i[\bar{X}/\bar{c}](\bar{f})\ \overset{\text{и.х.}}{\simeq}\ d.$$

Сега от равенствата (3.13) и (3.14) получаваме финално

$$\mu(\bar{f})\ \simeq\ f_i(\bar{c})\ \simeq\ \Gamma_{\tau_i}(\bar{f})(\bar{c})\ \simeq\ d,$$

което и трябваше да покажем.

□

3.4.2 Доказателство на включването $D_V(R) \subseteq O_V(R)$

За да покажем това включване, се оказва полезно да въведем k на брой спомагателни функции, които ще означим с $g_1, \ldots, g_k$. Техният интуитивен смисъл е следният: $g_i : \mathbb{N}^{m_i} \rightharpoonup \mathbb{N}$ е функцията, която се определя операционно по стойност от F_i. Формалното определение на g_i е следното:

$$g_i(c_1, \ldots, c_{m_i}) \simeq d \quad \overset{\text{деф}}{\iff} \quad R \vdash_V F_i(c_1, \ldots, c_{m_i}) \ \to \ d \tag{3.15}$$

за всички естествени $c_1, \ldots, c_{m_i}$ и d.

Разбира се, от това определение никак не следва, че $g_1, \ldots, g_k$ наистина са *еднозначни* функции. Това ще се изясни по-надолу. Нещо повече, ще се окаже, че векторът $(g_1, \ldots, g_k)$ всъщност е най-малката неподвижна точка на оператора Γ, т.е. ще бъде вярно, че

$$\bar{g} = \bar{f}_\Gamma.$$

Когато докажем това, ще сме на една крачка от включването $D_V(R) \subseteq O_V(R)$, което е нашата цел в този раздел.

Най-напред да забележим, че когато викаме F_i с аргументи, които са *константи* (а не произволни *термове*), е в сила еквивалентността:

Лема 3.4. За произволни константи $c_1, \ldots, c_{m_i}$ и d:

$$R \vdash_V F_i(c_1, \ldots, c_{m_i}) \ \to \ d \quad \iff \quad R \vdash_V \tau_i[\bar{X}/\bar{c}] \ \to \ d$$

за всяко $1 \leq i \leq k$.

Доказателство. Ако $R \vdash_V F_i(c_1, \ldots, c_{m_i}) \ \to \ d$, то съгласно *Лема* 3.1 преди това трябва да е било изведено опростяването

$$\tau_i[X_1/c_1, \ldots, X_{m_i}/c_{m_i}] \to d.$$

Обратно, нека $R \vdash_V \tau_i[\bar{X}/\bar{c}] \ \to \ d$. Трябва да покажем, че са изпълнени предпоставките на правило (3_V), което е единственото правило, от което можем да изведем $R \vdash_V F_i(c_1, \ldots, c_{m_i}) \ \to \ d$:

$$\frac{R \vdash_V c_1 \to c_1 \quad \cdots \quad R \vdash_V c_{m_i} \to c_{m_i} \quad R \vdash_V \tau_i[X_1/c_1, \ldots, X_{m_i}/c_{m_i}] \to d}{R \vdash_V F_i(c_1, \ldots, c_{m_i}) \ \to \ d} \quad (3_V)$$

Но това наистина е така, защото първите опростявания над чертата са аксиоми, а това, че $R \vdash_V \tau_i[X_1/c_1, \ldots, X_{m_i}/c_{m_i}] \to d$ ни е дадено по условие. $\square$

Сега можем да пристъпим към доказателството на едната половина от равенството $\bar{g} = \bar{f_\Gamma}$, а именно:

Твърдение 3.5.

$$\bar{g} \subseteq \bar{f_\Gamma}.$$

Доказателство. Да фиксираме $1 \leq i \leq k$, както и произволни константи $c_1, \ldots, c_{m_i}, d$ и да приложим *Твърдение* 3.4 към функционалния терм $\mu = F_i(c_1, \ldots, c_{m_i})$ и н.м.н.т. $\bar{f_\Gamma} = (f_\Gamma^1, \ldots, f_\Gamma^k)$ на Γ. Ще получим

$$R \vdash_V \underbrace{F_i(c_1, \ldots, c_{m_i})}_{\mu} \to d \quad \Longrightarrow \quad \underbrace{F_i(c_1, \ldots, c_{m_i})}_{\mu}(\bar{f_\Gamma}) \simeq d,$$

или все едно,

$$\underbrace{R \vdash_V F_i(c_1, \ldots, c_{m_i}) \to d}_{g_i(c_1, \ldots, c_{m_i}) \simeq d} \quad \Longrightarrow \quad f_\Gamma^i(c_1, \ldots, c_{m_i}) \simeq d.$$

Но предпоставката на тази импликация е точно дефиницията (3.15) на g_i. Излезе, че за произволни $c_1, \ldots, c_{m_i}$ и d

$$g_i(c_1, \ldots, c_{m_i}) \simeq d \quad \Longrightarrow \quad f_\Gamma^i(c_1, \ldots, c_{m_i}) \simeq d,$$

което означава, че $g_i \subseteq f_\Gamma^i$. Понеже i беше произволно, имаме всъщност $\bar{g} \subseteq \bar{f_\Gamma}$. $\square$

Да помислим как да покажем обратното включване $f_\Gamma^i \subseteq g_i$. Разсъждаваме така: щом g_i са функциите, които се пресмятат операционно по стойност от F_i, то звучи логично операционната семантика по стойност на R — функцията $O_V(R)$ — да може да се дефинира и така:

$$O_V(R)(\bar{c}) \ \simeq \ \tau_0(\bar{c}, \bar{g}).$$

Оказва се, че това наистина е възможно. Всъщност търсеното от нас включване $\bar{f_\Gamma} \subseteq \bar{g}$ ще следва от едната половина на горното равенство $O_V(R) = \lambda \bar{c}.\tau_0(\bar{c}, \bar{g})$, затова ще докажем само тази половина. (Другата половина също е вярна, но няма да ни трябва, затова няма да я разглеждаме.)

Става въпрос за включването $\lambda \bar{c}.\tau_0(\bar{c}, \bar{g}) \subseteq O_V(R)$, което разписано поточково изглежда така: за произволни $c_1, \ldots, c_{m_i}$ и d:

$$\tau_0(\bar{c}, \bar{g}) \simeq d \quad \Longrightarrow \quad O_V(R)(\bar{c}) \simeq d,$$

или преписано чрез дефиницията на O_V:

$$\tau_0(\bar{c}, \bar{g}) \simeq d \quad \Longrightarrow \quad R \vdash_V \tau_0[\bar{X}/\bar{c}] \ \to \ d.$$

От *Лема* 3.3 имаме, че $\tau_0(\bar{c}, \bar{g})$ можем да си представяме отново като $\tau_0[\bar{X}/\bar{c}](\bar{g})$. Следователно горната импликация можем да запишем по следния начин:

$$\underbrace{\tau_0[\bar{X}/\bar{c}]}_{\mu}(\bar{g}) \simeq d \quad \Longrightarrow \quad R \vdash_V \underbrace{\tau_0[\bar{X}/\bar{c}]}_{\mu} \ \to \ d.$$

Ясно е, че ще ни се наложи да доказваме тази импликация за *произволен* функционален терм μ.

Твърдение 3.6. За всеки функционален терм $\mu(F_1, \ldots, F_k)$ и всяко естествено d е в сила:

$$\mu(\bar{g}) \simeq d \quad \Longrightarrow \quad R \vdash_V \mu \ \to \ d.$$

Доказателство. Доказателството отново ще е с индукция, но този път по построението на функционалния терм μ. Фиксираме произволен функционален терм $\mu(F_1, \ldots, F_k)$ и приемаме, че за всички функционални термове, построени *преди* него твърдението е вярно (*индуктивна хипотеза*). Разглеждаме различните възможности за μ.

1) Нека μ е константа. Тогава равенството $\mu(\bar{g}) \simeq d$ може да се случи само при $\mu = d$, откъдето очевидно $R \vdash_V d \ \to \ d$.

2) Случаят $\mu = X_i$ е невъзможен, защото μ е функционален терм.

3) μ е от вида $\mu_1 \ op \ \mu_2$. Имаме по условие, че

$$\mu(\bar{g}) \overset{\text{деф}}{\simeq} \mu_1(\bar{g}) \ op \ \mu_2(\bar{g}) \ \simeq \ d.$$

Тогава трябва да съществуват числа d_1 и d_2, такива че

$$\mu_1(\bar{g}) \simeq d_1, \quad \mu_2(\bar{g}) \simeq d_2 \quad \text{и} \quad d_1 \ \mathbf{op} \ d_2 \simeq d.$$

Като приложим индукционната хипотеза към μ_1 и μ_2, от горните равенства ще имаме

$$R \vdash_V \mu_1 \ \to \ d_1, \quad R \vdash_V \mu_2 \ \to \ d_2 \quad \text{и} \quad d_1 \ \mathbf{op} \ d_2 \ = \ d.$$

Но това са точно условията над чертата на правилото (1). Прилагаме го и получаваме, че

$$R \vdash_V \mu_1 \ op \ \mu_2 \ \to \ d.$$

4) Нека μ е от вида $\texttt{if}\ \mu_1\ \texttt{then}\ \mu_2\ \texttt{else}\ \mu_3$. По дефиниция имаме:

$$\mu(\bar{g}) \simeq \begin{cases} \mu_2(\bar{g}), & \text{ако } \mu_1(\bar{g}) > 0 \\ \mu_3(\bar{g}), & \text{ако } \mu_1(\bar{g}) = 0 \\ \neg!, & \text{ако } \neg!\mu_1(\bar{g}). \end{cases}$$

Понеже по условие $\mu(\bar{g}) \simeq d$, не е възможно $\neg!\mu_1(\bar{g})$, и значи остават случаите $\mu_1(\bar{g}) > 0$ и $\mu_1(\bar{g}) = 0$. Ще разгледаме само първия; при втория се разсъждава аналогично.

Нека $\mu_1(\bar{g}) > 0$. Тогава $\mu_1(\bar{g}) \simeq d_1$ за някое $d_1 > 0$. От дефиницията на стойност на условен терм се вижда, че в този случай $\mu(\bar{g}) \overset{\text{деф}}{\simeq} \mu_2(\bar{g}) \simeq d$. Да ги запишем общо:

$$\mu_1(\bar{g}) \simeq d_1 > 0 \quad \text{и} \quad \mu_2(\bar{g}) \simeq d.$$

Понеже термовете μ_1 и μ_2 са построени преди μ, то за тях индуктивната хипотеза е в сила. Така от горните две равенства получаваме

$$R \vdash_V \mu_1 \ \to\ d_1 \quad \text{за } d_1 > 0 \quad \text{и} \quad R \vdash_V \mu_2 \ \to\ d,$$

които са точно предпоставките на правилото $(2_{\mathbf{t}})$. Прилагаме го и получаваме търсеното

$$R \vdash_V \mu \ \to\ d.$$

5) Последният случай $\mu = F_i(\mu_1, \ldots, \mu_{m_i})$ отново е най-интересен. Точно тук ще използваме специалния избор на функциите $\bar{g}$. Имаме по условие, че $\mu(\bar{g}) \simeq d$, което по дефиницията за стойност на терм означава, че

$$g_i(\mu_1(\bar{g}), \ldots, \mu_{m_i}(\bar{g})) \simeq d.$$

Тогава съществуват числа $d_1, \ldots, d_{m_i}$, такива че

$$\mu_1(\bar{g}) \simeq d_1, \ldots, \mu_{m_i}(\bar{g}) \simeq d_{m_i} \quad \text{и} \quad g_i(d_1, \ldots, d_{m_i}) \simeq d.$$

Понеже термовете $\mu_1, \ldots, \mu_{m_i}$ са построени на по-ранен етап от μ, то за тях индуктивната хипотеза е в сила, т.е. ще имаме изводимостите

$$R \vdash_V \ \mu_1 \ \to\ d_1, \ \ldots\ , R \vdash_V \ \mu_{m_i} \ \to\ d_{m_i}. \tag{3.16}$$

По-горе получихме $g_i(d_1, \ldots, d_{m_i}) \simeq d$, което по дефиницията (3.15) на g_i означава, че

$$R \vdash_V \ F_i(d_1, \ldots, d_{m_i}) \ \to\ d.$$

Оттук по *Лема* 3.4 ще имаме

$$R \vdash_V \tau_i[\bar{X}/\bar{d}] \ \to\ d.$$

Тази изводимост, заедно с другите от (3.16), всъщност са точно предпоставките на правило (3_V). Прилагаме го и получаваме търсеното

$$R \vdash_V \underbrace{F_i(\mu_1, \ldots, \mu_{m_i})}_{\mu} \rightarrow d.$$

□

Сега да видим как от твърдението, което току-що доказахме, ще следва търсеното от нас включване $\bar{f}_\Gamma \subseteq \bar{g}$:

Твърдение 3.7. $\bar{f}_\Gamma \subseteq \bar{g}$.

Доказателство. По определение $\bar{f}_\Gamma$ е най-малката неподвижна точка на оператора Γ. От теоремата на Кнастер-Тарски за произволни ОС знаем, че $\bar{f}_\Gamma$ е и най-малката *преднеподвижна точка* на Γ, т.е. тя е най-малкото решение и на *неравенството*

$$\Gamma(\bar{f}) \subseteq \bar{f}.$$

Ние ще покажем, че векторът $\bar{g} = (g_1, \ldots, g_k)$ също е решение на това неравенство, т.е. $\Gamma(\bar{g}) \subseteq \bar{g}$. Оттук, понеже $\bar{f}_\Gamma$ е най-малкото решение, ще получим желаното включване $\bar{f}_\Gamma \subseteq \bar{g}$.

Да покажем $\Gamma(\bar{g}) \subseteq \bar{g}$ означава да покажем, че за всяко $i = 1, \ldots k$

$$\Gamma_{\tau_i}(\bar{g}) \subseteq g_i.$$

Да фиксираме произволно $1 \leq i \leq k$, както и произволни естествени числа $c_1, \ldots, c_{m_i}$ и d. Ще покажем, че за тях е в сила импликацията:

$$\Gamma_{\tau_i}(\bar{g})(\bar{c}) \simeq d \implies g_i(\bar{c}) \simeq d. \tag{3.17}$$

Наистина, от определението за термален предикат имаме:

$$\Gamma_{\tau_i}(\bar{g})(\bar{c}) \overset{\text{деф}}{\simeq} \tau_i(\bar{c}, \bar{g}) \overset{\text{Лема 3.3}}{\simeq} \tau_i[\bar{X}/\bar{c}](\bar{g}).$$

Сега нека приемем, че предпоставката $\Gamma_{\tau_i}(\bar{g})(\bar{c}) \simeq d$ на импликацията (3.17) е в сила. Оттук, като използваме горните равенства, получаваме

$$\underbrace{\tau_i[\bar{X}/\bar{c}]}_{\mu}(\bar{g}) \simeq d.$$

Тук е моментът да приложим последното *Твърдение* 3.6 за $\mu = \tau_i[\bar{X}/\bar{c}]$. Ще получим, че

$$R \vdash_V \tau_i[\bar{X}/\bar{c}] \to d.$$

Съгласно *Лема* 3.4 това условие е еквивалентно на

$$R \vdash_V F_i(\bar{c}) \to d,$$

откъдето по дефиницията (3.15) на g_i, получаваме търсеното $g_i(\bar{c}) \simeq d$. С това проверката на (3.17) е завършена и следователно $\Gamma_{\tau_i}(\bar{g}) \subseteq g_i$. Тъй като i беше произволно, това ни дава общо, че $\Gamma(\bar{g}) \subseteq \bar{g}$, т.е. $\bar{g}$ е преднеподвижна точка на оператора Γ и следователно $\bar{f}_\Gamma \subseteq \bar{g}$. □

Като следствие от доказаните до тук *Твърдение* 3.5 и *Твърдение* 3.7 получаваме това, което беше нашата цел:

Следствие 3.2. $\bar{f_\Gamma} \;=\; \bar{g}.$

Вече подготвихме всичко за финалната

Теорема 3.2. За всяка рекурсивна програма R

$$O_V(R) \;=\; D_V(R).$$

Доказателство. Първото включване $O_V(R) \subseteq D_V(R)$ е точно *Следствие* 3.1, което доказахме в предишния раздел.

За да покажем и обратното включване $D_V(R) \subseteq O_V(R)$, да приемем, че за произволни $c_1, \ldots, c_{m_i}$ и d

$$D_V(R)(\bar{c}) \simeq d.$$

Това означава, съгласно дефиницията на $D_V(R)$, че

$$\tau_0(\bar{c}, \bar{f_\Gamma}) \simeq d.$$

Но $\bar{f_\Gamma} = \bar{g}$ и значи

$$\tau_0(\bar{c}, \bar{g}) \simeq d,$$

което, използвайки, *Лема* 3.3 можем да препишем като

$$\tau_0[\bar{X}/\bar{c}](\bar{g}) \simeq d.$$

Но тогава от *Твърдение* 3.6 ще имаме, че

$$R \vdash_V \;\; \tau_0[\bar{X}/\bar{c}] \;\rightarrow\; d,$$

което по дефиницията (3.7) на операционна семантика означава точно

$$O_V(R)(\bar{c}) \simeq d.$$

Това завършва доказателството на включването $D_V(R) \subseteq O_V(R)$ и на теоремата. □

Накрая да направим важното уточнение, че никъде в доказателството на тази теорема не използвахме, че програмата R е над естествените числа. Това означава, че този резултат остава в сила и за рекурсивни програми над *произволен* тип данни.

3.5 Денотационна семантика с предаване на параметрите по име

3.5.1 Функционалната област на Скот $\boldsymbol{\mathcal{F}_k^\perp} = (\mathcal{F}_k^\perp,\ \sqsubseteq,\ \Omega^{(k)})$

Тази ОС въведохме, когато разглеждахме общата теория на областите на Скот. Сега ще припомним основните понятия и факти от раздели 2.1.3 и 2.2.2, които ще използваме тук.

Фиксираме някакъв елемент $\perp \notin \mathbb{N}$ и полагаме

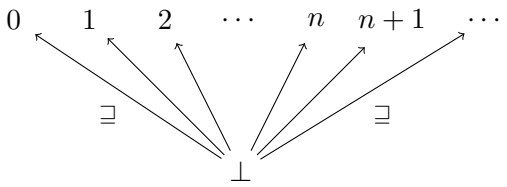
$$\mathbb{N}_\perp = \mathbb{N} \cup \{\perp\}.$$

Плоската наредба на $\mathbb{N}_\perp$ се дефинира посредством еквивалентността:

$$x \sqsubseteq y \overset{\text{деф}}{\iff} x = \perp \ \vee \ x = y. \tag{3.18}$$

Ясно е, че $\perp \sqsubseteq x$ за всяко $x \in N_\perp$, т.е. $\perp$ е най-малкият елемент на $\mathbb{N}_\perp$. Образно казано, той е на дъното на $\mathbb{N}_\perp$, затова се нарича *bottom* елемент. Това ще е елементът, с който ще обозначаваме, че една функция няма стойност в дадена точка; например $f(5) = \perp$ ще означава, че f няма стойност в 5.

Ето как изглеждаше графиката на релацията $\sqsubseteq$ върху множеството $\mathbb{N}_\perp$ (без примките $n \sqsubseteq n$):



Да наблегнем отново на това, че релацията $\sqsubseteq$ няма нищо общо с числовото $\leq$. Две *числа* x и y са свързани с тази релация, т.е. $x \sqsubseteq y$, само когато $x = y$.

От *Твърдение* 2.2 знаем, че структурата $\boldsymbol{N_\perp} = (N_\perp, \sqsubseteq, \perp)$ е ОС; ще я наричаме *плоска област на Скот*. Точната горна граница на монотонно растящата редица $x_0 \sqsubseteq x_1 \sqsubseteq \ldots$ в $N_\perp$ ще означаваме с

$$\bigsqcup_n x_n.$$

В декартовото произведение $\mathbb{N}_\bot^k = \underbrace{\mathbb{N}_\bot \times \cdots \times \mathbb{N}_\bot}_{k}$ дефинираме *покомпонентната наредба*, индуцирана от плоската наредба в $\mathbb{N}_\bot$, по следния начин:

$$(x_1, \ldots, x_k) \sqsubseteq (y_1, \ldots, y_k) \overset{\text{деф}}{\iff} x_1 \sqsubseteq y_1 \ \& \ \ldots \ \& \ x_k \sqsubseteq y_k.$$

Забележка. Тази наредба в $\mathbb{N}_\bot^k$ ще означаваме със същия символ $\sqsubseteq$ и също ще наричаме плоска наредба.

Ще пишем $\bar{x} \sqsubset \bar{y}$, за да означим, че $\bar{x} \sqsubseteq \bar{y}$ и $\bar{x} \neq \bar{y}$.

Съгласно *Твърдение* 2.5, структурата

$$\mathbb{N}_\bot^{\boldsymbol{k}} = (\underbrace{\mathbb{N}_\bot \times \cdots \times \mathbb{N}_\bot}_{k}, \ \sqsubseteq, \ \underbrace{(\bot, \ldots, \bot)}_{k})$$

също е област на Скот. И нея ще наричаме *плоска* ОС.

Да напомним и дефиницията (2.3) на *функционалната* област на Скот, породена от плоската ОС $\mathbb{N}_\bot^{\boldsymbol{k}}$:

$$\boldsymbol{\mathcal{F}_k^\bot} = (\mathcal{F}_k^\bot, \ \sqsubseteq, \ \Omega^{(k)}).$$

Тази област е с домейн множеството $\mathcal{F}_k^\bot$ от всички *тотални* k-местни функции в $\mathbb{N}_\bot$:

$$\mathcal{F}_k^\bot = \{f |\ f\colon \ \mathbb{N}_\bot^k \ \to \mathbb{N}_\bot\}.$$

Приехме да я означаваме така по аналогия с ОС на *частичните* функции $\boldsymbol{\mathcal{F}_k} = (\mathcal{F}_k, \ \subseteq, \ \emptyset^{(k)})$, в която работехме дотук.

Наредбата в $\mathcal{F}_k^\bot$ е *поточковата наредба*, индуцирана от плоската наредба в $\mathbb{N}_\bot^k$, по-точно:

$$f \sqsubseteq g \overset{\text{деф}}{\iff} \forall x_1 \in \mathbb{N}_\bot \ldots \forall x_k \in \mathbb{N}_\bot \ f(x_1, \ldots, x_k) \sqsubseteq g(x_1, \ldots, x_k). \quad (3.19)$$

Наредбата $\sqsubseteq$ е пълна, т.е. всяка монотонно растяща редица $f_0 \sqsubseteq f_1 \sqsubseteq \ldots$ в $\mathcal{F}_k^\bot$ има *точна горна граница*. Тази граница ще означаваме с $\bigsqcup\limits_n f_n$. Очаквано, и тя се дефинира поточково:

$$\underbrace{(\bigsqcup_n f_n)(\bar{x})}_{lub \text{ в } \mathcal{F}_k^\bot} \overset{\text{деф}}{=} \underbrace{\bigsqcup_n f_n(\bar{x})}_{lub \text{ в } \mathbb{N}_\bot}. \quad (3.20)$$

Най-малкият елемент на $\mathcal{F}_k^\bot$ — функцията $\Omega^{(k)}$ — дефинираме така: за всяка $(x_1, \ldots, x_k) \in \mathbb{N}_\bot^k$

$$\Omega^{(k)}(x_1, \ldots, x_k) \overset{\text{деф}}{=} \bot$$

При доказателствата в този раздел се оказва удобен следният еквивалентен запис на релацията $\sqsubseteq$:

Твърдение 3.8. За произволни функции $f, g \in \mathcal{F}_k^\perp$:

$$f \sqsubseteq g \iff \forall \bar{x} \in \mathbb{N}_\perp^k \ (f(\bar{x}) \neq \perp \implies f(\bar{x}) = g(\bar{x})).$$

Доказателство. Директно от дефиницията (3.18) на плоска наредба и това, че дизюнкцията $p \vee q$ е еквивалентна на $\neg p \implies q$:

$$f \sqsubseteq g \overset{(3.19)}{\iff} \forall \bar{x} \in \mathbb{N}_\perp^k \ (f(\bar{x}) \sqsubseteq g(\bar{x})) \overset{(3.18)}{\iff} \forall \bar{x} \in \mathbb{N}_\perp^k \ (f(\bar{x}) = \perp \vee f(\bar{x}) = g(\bar{x}))$$
$$\iff \forall \bar{x} \in \mathbb{N}_\perp^k \ (f(\bar{x}) \neq \perp \implies f(\bar{x}) = g(\bar{x})).$$

□

Обърнете внимание колко си приличат горната характеризация на $\sqsubseteq$ и дефиницията на релацията $\subseteq$ между частични функции:

$$f \subseteq g \iff \forall \bar{x} \in \mathbb{N}^k \ (!f(\bar{x}) \implies f(\bar{x}) \simeq g(\bar{x})).$$

Подобна аналогия забелязваме и между дефиницията на точна горна граница $g = \bigcup_n f_n$ на монотонно растяща редица $f_0 \subseteq f_1 \subseteq \ldots$ в $\mathcal{F}_k$:

$$g(\bar{x}) \simeq y \iff \exists n \ f_n(\bar{x}) \simeq y$$

и следващото свойство на точната горна граница в $\mathcal{F}_k^\perp$:

Твърдение 3.9. Нека $f_0 \sqsubseteq f_1 \sqsubseteq \ldots$ е монотонно растяща редица в $\mathcal{F}_k^\perp$ и $g = \bigsqcup_n f_n$ е нейната точна горна граница. Тогава за всяка k-орка $\bar{x} \in \mathbb{N}_\perp^k$ и $y \in \mathbb{N}$ е вярно, че:

а) $g(\bar{x}) = \perp \iff \forall n \ f_n(\bar{x}) = \perp$;

б) $g(\bar{x}) = y \iff \exists n \ f_n(\bar{x}) = y$.

Доказателство. Нека $f_0 \sqsubseteq f_1 \sqsubseteq \ldots$ е монотонно растяща. За произволно $\bar{x} \in \mathbb{N}_\perp^k$ и да означим $y_n \overset{\text{деф}}{=} f_n(\bar{x})$. От това, че редицата от функции е монотонно растяща в $\mathcal{F}_k^\perp$ следва, че и редицата от стойностите им ще е монотонно растяща в $\mathbb{N}_\perp$, т.е. ще имаме

$$y_0 \sqsubseteq y_1 \sqsubseteq y_2 \sqsubseteq \ldots.$$

От основните свойства на плоската наредба от раздел 2.1.3 знаем, че всяка монотонно растяща редица в $\mathbb{N}_\perp$ изглежда по един от следните два начина:

– $\perp, \perp, \perp, \ldots$ с граница $\perp$;

– $\underbrace{\perp, \ldots \perp}_{n \geq 0}, y, y, \ldots$ с граница $y \in \mathbb{N}$.

Сега вече условията от твърдението изглеждат съвсем очевидни. □

3.5.2 Точни функции

Ще въведем едно подмножество на функциите от $\mathcal{F}_k^{\perp}$, което в някакъв смисъл е аналог на частичните функции $\mathcal{F}_k$.

Определение 3.14. Казваме, че функцията $f\colon \mathbb{N}_{\perp}^k \longrightarrow \mathbb{N}_{\perp}$ е *точна* (или *стриктна, strict*), ако за всички $(x_1, \dots\ x_k) \in \mathbb{N}_{\perp}^k$ е изпълнено:

$$(\exists i\colon x_i = \perp) \implies f(x_1, \dots\ x_k) = \perp.$$

С други думи, една функция е точна, ако всеки път, когато някой от аргументите ѝ е $\perp$, стойността ѝ също е $\perp$.

На пръв поглед между една частична функция $f\colon \mathbb{N}^k \rightharpoonup \mathbb{N}$ и една точна функция $f\colon \mathbb{N}_{\perp}^k \longrightarrow \mathbb{N}_{\perp}$ няма разлика, ако приемем, че случаят $\neg!f(\bar{x})$ отговаря на $f(\bar{x}) = \perp$. Да не забравяме, обаче, че сред аргументите на $f\colon \mathbb{N}_{\perp}^k \longrightarrow \mathbb{N}_{\perp}$ може да има $\perp$, за разлика от аргументите на частичната функция $f\colon \mathbb{N}^k \rightharpoonup \mathbb{N}$, които са само числа.

Множеството на всички k-местни точни функции ще означаваме с $\mathcal{S}_k$, с други думи

$$\mathcal{S}_k \ = \ \{f \mid f\colon \mathbb{N}_{\perp}^k \longrightarrow \mathbb{N}_{\perp} \ \&\ f \text{ е точна}\}.$$

Частичните функции от $\mathcal{F}_k$ "потапяме" в множеството $\mathcal{S}_k \subseteq \mathcal{F}_k^{\perp}$ по следния начин: на всяка функция $f \in \mathcal{F}_k$ съпоставяме следната точна функция:

$$f^*(\bar{x}) \ = \ \begin{cases} f(\bar{x}), & \text{ако } \bar{x} \in \mathbb{N}^k \ \&\ !f(\bar{x}) \\ \perp, & \text{иначе} \end{cases}$$

за всяко $\bar{x} \in \mathbb{N}_{\perp}^k$. Случаят "иначе" ще рече, че или сред елементите на k-орката $\bar{x}$ има $\perp$, или $\bar{x} \in \mathbb{N}^k$, но $\neg!f(\bar{x})$.

Функцията f^* ще наричаме *естествено продължение* на f. Тя очевидно е точна функция.

Примери. Ето няколко примера за естествени продължения на функции и предикати:
а) Нека $f(x, y) = x + y$. Нейното естествено продължение f^* е функцията

$$f^*(x, y) = \begin{cases} x + y, & \text{ако } x \in \mathbb{N} \ \&\ y \in \mathbb{N} \\ \perp, & \text{ако } x = \perp \ \vee \ y = \perp. \end{cases}$$

б) Нека x `div` y е функцията целочислено деление

$$x \texttt{ div } y \simeq \begin{cases} \lfloor \frac{x}{y} \rfloor, & \text{ако } y > 0 \\ \neg!, & \text{ако } y = 0. \end{cases}$$

Нейното естествено продължение $x\ \mathtt{div}^* y$ вече е тоталната функция

$$x\ \mathtt{div}^* y = \begin{cases} \lfloor \frac{x}{y} \rfloor, & \text{ако } x \in \mathbb{N}\ \&\ y \in \mathbb{N}^+ \\ \bot, & \text{ако } x = \bot\ \vee\ y = \bot\ \vee\ y = 0. \end{cases}$$

в) Да означим с E предиката "равенство". Неговото естествено продължение E^* има следната дефиниция:

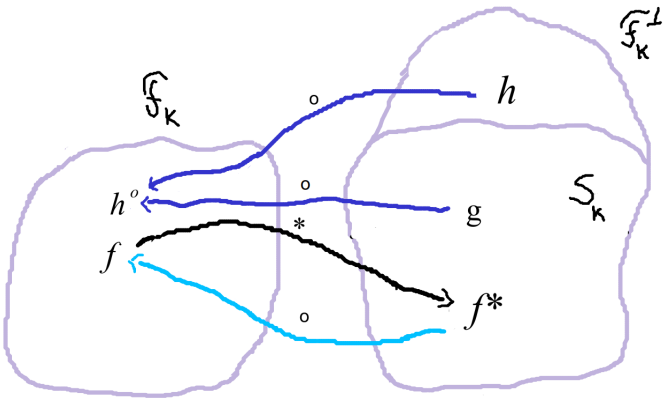
$$E^*(x,y) = \begin{cases} 1, & \text{ако } x \in \mathbb{N}\ \&\ y \in \mathbb{N}\ \&\ x = y \\ 0, & \text{ако } x \in \mathbb{N}\ \&\ y \in \mathbb{N}\ \&\ x \neq y \\ \bot, & x = \bot\ \vee\ y = \bot. \end{cases}$$

Лесно се съобразява, че изображението * е биекция между $\mathcal{F}_k$ и $\mathcal{S}_k$. Следователно една функция от $\mathcal{F}_k^\bot$ е точна тогава и само тогава, когато се явява естествено продължение на някоя частична функция от $\mathcal{F}_k$.

Да дефинираме и обратното изображение $^\circ$, което се прилага не само върху точните, а върху всички функции от $\mathcal{F}_k^\bot$. За всяка $f \in \mathcal{F}_k^\bot$ полагаме

$$f^\circ(\bar{x}) \simeq \begin{cases} f(\bar{x}), & \text{ако } \ f(\bar{x}) \neq \bot \\ \neg!, & \text{иначе.} \end{cases} \tag{3.21}$$

за всяко $\bar{x} \in \mathbb{N}^k$.



Като лесно упражнение съобразете, че:

Задача 3.1. а) За всяка функция $f \in \mathcal{F}_k$ е вярно, че $(f^*)^\circ = f$.
б) За всяка *точна* функция от $\mathcal{F}_k^\bot$ е вярно, че $(f^\circ)^* = f$.
в) Дайте пример за функция $f \in \mathcal{F}_k^\bot$, за която това равенство вече не е в сила.

Видяхме, че точните функции са някакъв аналог на частичните функции, затова не е изненадващо, че те също образуват област на Скот.

Задача 3.2. Докажете, че наредената тройка $\boldsymbol{\mathcal{S}_k} = (\mathcal{S}_k, \sqsubseteq, \Omega^{(k)})$ е област на Скот.

Решение. Релацията $\sqsubseteq$, която е частична наредба на $\mathcal{F}_k^{\perp}$, разбира се, ще е такава и върху множеството $\mathcal{S}_k$. Ясно е също, че функцията $\Omega^{(k)}$ е точна (и съответно се явява най-малък елемент на $\mathcal{S}_k$). Остана да видим, че границата на монотонно растяща редица от точни функции също е точна.

Наистина, нека

$$f_0 \sqsubseteq f_1 \sqsubseteq \dots$$

е редица от точни функции и нека

$$g = \bigsqcup_n f_n$$

е нейната граница в $\mathcal{F}_k^{\perp}$. Да вземем една k-орка $\bar{x} = (x_1, \dots x_k)$, в която участва $\perp$. Тогава за всяко n ще имаме $f_n(\bar{x}) = \perp$, откъдето по *Твърдение* 3.9 а) получаваме, че и $g(\bar{x}) = \perp$. □

Областта на Скот $\boldsymbol{\mathcal{S}_k} = (\mathcal{S}_k, \sqsubseteq \Omega^{(k)})$ на практика е идентична с областта $\boldsymbol{\mathcal{F}_k} = (\mathcal{F}_k, \subseteq, \emptyset^{(k)})$, поради което няма да представлява особен интерес за нас. Ако търсим модел, в който да дефинираме денотационна семантика за *call by name*, е ясно, че ще трябва да отидем отвъд множеството на точните функции.

Ако се питате защо този модел да не е *цялата* ОС $\boldsymbol{\mathcal{F}_k^{\perp}} = (\mathcal{F}_k^{\perp}, \sqsubseteq, \Omega^{(k)})$ — погледнете примера по-долу $\ddot{\smile}$. Проблемът е, че има термове τ, за които съответният им термален оператор Γ_τ не е непрекъснат и значи идеята за семантика с най-малка неподвижна точка не може да се осъществи.

Ето един прост пример за такъв терм τ:

Пример 3.1. Да разгледаме терма $\tau(X, F, G) = F(G(X))$. Да се убедим, че операторът Γ_τ, който той определя, не е монотонен, и следователно не е непрекъснат.

Доказателство. Термалният оператор $\Gamma_\tau \colon \mathcal{F}_1^{\perp} \times \mathcal{F}_1^{\perp} \longrightarrow \mathcal{F}_1^{\perp}$, който съответства на терма $\tau = F(G(X))$, е следният:

$$\Gamma_\tau(f, g)(x) \stackrel{\text{деф}}{=} f(g(x))$$

за всяко $x \in \mathbb{N}_\perp$. Да покажем, че Γ_τ не е монотонен означава да посочим две двойки функции (f_1, g_1) и (f_2, g_2), такива че $(f_1, g_1) \sqsubseteq (f_2, g_2)$, но $\Gamma_\tau(f_1, g_1) \not\sqsubseteq \Gamma_\tau(f_2, g_2)$. Последното, съгласно определение (3.19) означава, че за поне едно $x \in \mathbb{N}_\perp$:

$$\Gamma_\tau(f_1, g_1)(x) \not\sqsubseteq \Gamma_\tau(f_2, g_2)(x).$$

Да вземем например g_1 и g_2 да са следните функции:

$$g_1(x) = \begin{cases} \bot, & \text{ако } x = 0 \\ x, & \text{иначе} \end{cases} \qquad \text{и} \qquad g_2(x) = \begin{cases} 0, & \text{ако } x = 0 \\ x, & \text{иначе.} \end{cases}$$

Ясно е, че $g_1 \sqsubseteq g_2$. Нека f е функцията, която се дефинира като:

$$f(x) = \begin{cases} 5, & \text{ако } x = \bot \\ 10, & \text{иначе.} \end{cases}$$

Очевидно $(f, g_1) \sqsubseteq (f, g_2)$. Да видим какво се случва с $\Gamma_\tau(f, g_1)$ и $\Gamma_\tau(f, g_2)$. Да пресметнем стойността им за $x = 0$:

$$\Gamma_\tau(f, g_1)(0) \overset{\text{деф}}{=} f(g_1(0)) = f(\bot) = 5, \text{ а}$$

$$\Gamma_\tau(f, g_2)(0) \overset{\text{деф}}{=} f(g_2(0)) = f(0) = 10.$$

Следователно $\Gamma_\tau(f, g_1)(0) \not\sqsubseteq \Gamma_\tau(f, g_2)(0)$, откъдето и

$$\Gamma_\tau(f_1, g_1) \not\sqsubseteq \Gamma_\tau(f_2, g_2),$$

и значи Γ_τ наистина е много "дефектен" — той дори не е монотонен. $\square$

Ако разгледаме по-внимателно горния контрапример, виждаме, че функцията f също е дефектна в известен смисъл. Ако гледаме на нея като на функция, която се пресмята от някаква програма, тогава фактът, че $f(\bot) = 5$ може да се тълкува така: резултатът 5 е получен от аргумент $\bot$, който на практика не съществува. Следователно той е получен без да се използва този аргумент (или все едно — аргументът x на $f(x)$ е фиктивен аргумент). Но това означава, че би трябвало $f(x) = 5$ за *всяко* $x \in \mathbb{N}$, а това не е така. В този смисъл f не е "хубава" функция.

3.5.3 Монотонни и непрекъснати функции в ОС $\mathcal{F}_k^\bot$

Дали не е възможно негативният резултат от примера по-горе да се дължи на факта, че приложихме Γ_τ върху "неподходящи" функции? Точно това е причината! Оказва се, че ако ограничим термалните оператори до "подходящи" функции, тогава те ще бъдат не само монотонни, но и непрекъснати. Въпросните подходящи функции са всъщност *монотонните* функции.

Определение 3.15. Казваме, че функцията $f : \mathbb{N}_\bot^k \longrightarrow \mathbb{N}_\bot$ е *монотонна*, ако е изпълнено условието: за всяко $\bar{x} \in \mathbb{N}_\bot^k$ и $\bar{y} \in \mathbb{N}_\bot^k$:

$$\bar{x} \sqsubseteq \bar{y} \implies f(\bar{x}) \sqsubseteq f(\bar{y}).$$

Един първи пример за монотонни функции са точните функции, които въведохме по-горе:

Твърдение 3.10. Всяка точна функция е монотонна.

Доказателство. Нека $f : \mathbb{N}_\bot^k \longrightarrow \mathbb{N}_\bot$ е точна функция. Да вземем произволни $\bar{x} = (x_1, \ldots, x_k)$ и $\bar{y} = (y_1, \ldots, y_k)$, такива че $\bar{x} \sqsubseteq \bar{y}$. Интересен е случаят, когато

$$(x_1, \ldots, x_k) \sqsubset (y_1, \ldots, y_k).$$

Тогава за поне едно $i \in \{1, \ldots, k\}$ ще е изпълнено $x_i \sqsubset y_i$, което съгласно дефиницията на плоска наредба (3.18) означава, че $x_i = \bot$. Но f е точна и значи $f(\bar{x}) = \bot$, откъдето $f(\bar{x}) \sqsubseteq f(\bar{y})$. Следователно f е монотонна. □

При фиксирано $k \geq 1$, множеството на всички k-местни монотонни функции ще означаваме с $\mathcal{M}_k$:

$$\mathcal{M}_k \;=\; \{f \mid f\colon\ \mathbb{N}_\bot^k \longrightarrow \mathbb{N}_\bot \ \&\ f \text{ е монотонна}\}.$$

От горното твърдение имаме, че $\mathcal{S}_k \subseteq \mathcal{M}_k$. Дали е строго включването? Абсолютно. Най-простият пример е може би следващият:

Пример 3.2. Нека $f : \mathbb{N}_\bot \longrightarrow \mathbb{N}_\bot$ е константната функция $\lambda x.c$, където $c \in \mathbb{N}$. Имаме $f(\bot) = c$ и следователно f не е точна. Тя, обаче, е монотонна, защото при $x \sqsubseteq y$ ще имаме:

$$f(x) \;= c \;=\; f(y), \quad \text{следователно} \quad f(x) \sqsubseteq f(y).$$

Ето как изглеждат класовете от функции в $N_\bot$, които въведохме дотук:

Следващата задача обобщава резултата от горния пример и описва всички монотонни функции на един аргумент.

Задача 3.3. Докажете, че функцията $f : \mathbb{N}_\perp \longrightarrow \mathbb{N}_\perp$ е монотонна тогава и само тогава, когато е точна или е константна.

Доказателство. $\Longleftarrow$ Следва от *Твърдение* 3.10 и *Пример* 3.2.

$\Longrightarrow$ Нека f е едноместна монотонна функция. Имаме две възможности:

1 сл. $f(\perp) = \perp$ и в този случай f е точна.

2 сл. $f(\perp) = y \in \mathbb{N}$. Понеже за всяко $x \in \mathbb{N}_\perp$: $\perp \sqsubseteq x$, то от монотонността на f ще имаме

$$f(\perp) \sqsubseteq f(x), \quad \text{т.е.} \quad y \sqsubseteq f(x).$$

Но $y \in \mathbb{N}$ и тогава $y \sqsubseteq f(x)$ означава $y = f(x)$, като това е за всяко $x \in \mathbb{N}_\perp$. Следователно функцията f е константна. $\square$

В горната задача беше важно, че f е едноместна функция. Да вземем, обаче, ето тази двуместна функция $f : \mathbb{N}_\perp^2 \longrightarrow \mathbb{N}_\perp$:

$$f(x,y) \;=\; \begin{cases} \perp, & \text{ако } x = \perp \\ 5, & \text{ако } x \neq \perp. \end{cases}$$

Тогава f не е точна, защото например $f(0, \perp) = 5$. Очевидно тя не е и константна. Обаче f е монотонна: наистина, нека

$$(x,y) \;\sqsubset\; (x',y').$$

Строгото включване по-горе означава, че *задължително* някоя от компонентите на (x,y) (може и двете) да е $\perp$. Да разгледаме поотделно двете възможности за x:

1 сл. $x = \perp$. Тук

$$f(\perp, y) \overset{\text{деф}}{=} \perp \;\sqsubseteq\; f(x',y').$$

2 сл. $x \neq \perp$. Тогава непременно $y = \perp$. От $(x, \perp) \sqsubseteq (x', y')$ и факта, че $x \in N$ следва, че $x = x'$. Следователно

$$f(x,\perp) \overset{\text{деф}}{=} 5 \overset{\text{деф}}{=} f(x',y'), \quad \text{и значи} \quad f(x,y) \;\sqsubseteq\; f(x',y').$$

Така получихме, че f е монотонна.

За една функция $f : \mathbb{N}_\perp^k \longrightarrow \mathbb{N}_\perp$ ще казваме, че е *монотонна по i-тия си аргумент*, ако за всички $x_1, \ldots, x_i, \ldots, x_k, x_i' \in \mathbb{N}_\perp$ е изпълнено условието:

$$x_i \sqsubseteq x_i' \quad \Longrightarrow \quad f(x_1, \ldots, x_i, \ldots, x_k) \sqsubseteq f(x_1, \ldots, x_i', \ldots, x_k).$$

Задача 3.4. Функцията $f : \mathbb{N}_\perp^k \longrightarrow \mathbb{N}_\perp$ е монотонна тогава и само тогава, когато е монотонна по всеки от аргументите си.

Решение. Правата посока е ясна. Да приемем сега, че f е монотонна по всеки от аргументите си и да вземем произволни $\bar{x} = (x_1, \ldots, x_k)$ и $\bar{y} = (y_1, \ldots, y_k)$, такива че $(x_1, \ldots, x_k) \sqsubseteq (y_1, \ldots, y_k)$. Тогава и $(x_1, \ldots, x_k) \sqsubseteq (y_1, x_2, \ldots, x_k)$, откъдето поради монотонността на f по първия ѝ аргумент ще имаме, че

$$f(x_1, \ldots, x_k) \sqsubseteq f(y_1, \ldots, y_k).$$

Аналогично, от това че $(y_1, x_2, \ldots, x_k) \sqsubseteq (y_1, y_2, x_3, \ldots, x_k)$ и монотонността на f по втория аргумент ще имаме, че

$$f(y_1, x_2, \ldots, x_k) \sqsubseteq f(y_1, y_2, x_3, \ldots, x_k).$$

Повтаряме неколкократно това разсъждение, докато стигнем до последното включване $f(y_1, \ldots, y_{k-1}, x_k) \sqsubseteq f(y_1, \ldots, y_k)$. Така получаваме

$$f(x_1, \ldots, x_k) \sqsubseteq f(y_1, x_2, x_3, \ldots, x_k) \sqsubseteq \ \ldots \ \sqsubseteq f(y_1, \ldots, y_k)$$

и значи общо $f(\bar{x}) \sqsubseteq f(\bar{y})$. □

Оказва се, че монотонните функции от $\mathcal{F}_k$ съвпадат с *непрекъснатите функции*. Този факт изглежда странен, защото знаем, че за изображенията (или операторите), действащи върху частични функции, свойството непрекъснатост е по-силно от свойството монотонност (пример за оператор, който е монотонен, но не е непрекъснат в ОС $(\mathcal{F}_n, \subseteq, \emptyset^{(n)})$ е да кажем ето този).

За изображенията f, действащи в плоската ОС $(\mathbb{N}_\perp^k, \sqsubseteq, \underbrace{(\perp, \ldots, \perp)}_{k})$, обаче, това не е така, и причината е в следващото наблюдение.

Задача 3.5. Всяка монотонно растяща редица в областта на Скот $(\mathbb{N}_\perp^k, \sqsubseteq, \underbrace{(\perp, \ldots, \perp)}_{k})$ има краен брой различни елементи.

Забележка. Това твърдение е обобщение на предишно наблюдение, че всяка монотонно растяща редица в $\mathbb{N}_\perp$ изглежда по един от двата начина:

$$\perp,\ \perp,\ \perp, \ldots \qquad \text{или} \qquad \underbrace{\perp, \ldots \perp}_{n \geq 0}, y,\ y, \ldots \text{ за някое } y \in \mathbb{N}.$$

Решение. Наистина, ако $(x_1, \ldots, x_k) \sqsubset (y_1, \ldots, y_k)$, то за поне едно $i \in \{1, \ldots, k\}$ ще е вярно, че $x_i \sqsubset y_i$. Това според дефиницията на плоска наредба (3.18) означава, че $x_i = \perp$, а $y_i \in \mathbb{N}$. Когато имаме монотонно растяща редица от k-орки, е ясно, че това може да се случи най-много k пъти и следователно различните елементи в редицата са най-много $k+1$.

Ето един пример за $k = 3$, който казва всичко:

$$(\bot, \bot, \bot) \sqsubset (\bot,\ 5, \bot) \sqsubset (\bot, 5, 3) \sqsubset (0, 5, 3).$$

□

Да препишем определението 2.7 за непрекъснатост на изображение, което имаме от общата теория, за случая на изображение f, действащо от плоската ОС $(\mathbb{N}_\bot^k,\ \sqsubseteq,\ \underbrace{(\bot, \ldots, \bot)}_{k})$ към плоската ОС $(\mathbb{N}_\bot,\ \sqsubseteq,\ \bot)$. То ще изглежда така:

Функцията $f\colon\ \mathbb{N}_\bot^k \longrightarrow \mathbb{N}_\bot$ наричаме *непрекъсната*, ако за всяка монотонно растяща редица $\bar{x}^0 \sqsubseteq \bar{x}^1 \sqsubseteq \ \ldots$ в $\mathbb{N}_\bot^k$ е в сила равенството:

$$f\ (\ \underbrace{\bigsqcup_n \bar{x}^n}_{\text{т.г.гр. е в } \mathbb{N}_\bot^k}\ ) \ =\ \underbrace{\bigsqcup_n f(\bar{x}^n)}_{\text{т.г.гр. е в } \mathbb{N}_\bot}. \tag{3.22}$$

От общата теория (*Твърдение* 2.7) имаме, че всяко непрекъснато изображение е монотонно. Да видим, че за функциите от $\mathcal{F}_k^\bot$ е вярно и обратното.

Задача 3.6. Нека $k \geq 1$. Функцията $f\colon\ \mathbb{N}_\bot^k \longrightarrow \mathbb{N}_\bot$ е непрекъсната в ОС $(\mathbb{N}_\bot^k,\ \sqsubseteq,\ \underbrace{(\bot, \ldots, \bot)}_{k})$ точно тогава, когато е монотонна.

Решение. Необходимо е да покажем само, че от монотонност следва непрекъснатост.

За целта да вземем една монотонна функция $f : \mathbb{N}_\bot^k \longrightarrow \mathbb{N}_\bot$. Нека $\bar{x}^0 \sqsubseteq \ \bar{x}^1 \sqsubseteq \ \ldots$ е растяща редица в $\mathbb{N}_\bot^k$. Съгласно *Твърдение* 3.5, ще съществува индекс m, от който нататък редицата се стабилизира. Ясно е, че нейната граница ще е $\bar{x}^m$, т.е. редицата изглежда така:

$$\bar{x}^0 \sqsubseteq \bar{x}^1 \sqsubseteq \ldots \sqsubseteq \bar{x}^m \ = \bar{x}^m = \ \ldots \qquad \text{с граница } \bar{x}^m.$$

Понеже f е монотонна, оттук веднага получаваме, че редицата

$$f(\bar{x}^0) \sqsubseteq f(\bar{x}^1) \sqsubseteq \ldots \sqsubseteq f(\bar{x}^m) = f(\bar{x}^m)\ \ \ldots \text{ е с граница } f(\bar{x}^m). \tag{3.23}$$

Но тогава

$$f(\underbrace{\bigsqcup_n \bar{x}^n}_{\bar{x}^m}) \ =\ f(\bar{x}^m) \overset{(3.23)}{=} \bigsqcup_n f(\bar{x}^n)$$

и следователно f е непрекъсната. □

Ще завършим този раздел с важното наблюдение, че монотонните функции също образуват област на Скот.

Твърдение 3.11. За всяко $k \geq 1$ структурата $\boldsymbol{\mathcal{M}_k} = (\mathcal{M}_k,\ \sqsubseteq,\ \Omega^{(k)})$ е област на Скот.

Доказателство. Функцията $\Omega^{(k)}$ очевидно е монотонна. Релацията $\sqsubseteq$, която е частична наредба на $\mathcal{F}_k$, ще е частична наредба и на $\mathcal{M}_k$. Единственото, което трябва да проверим, е че тази наредба е пълна и в $\mathcal{M}_k$.

Наистина, нека $f_0 \sqsubseteq f_1 \sqsubseteq \ldots$ е растяща редица от монотонни функции и нека $g = \bigsqcup_n f_n$ е нейната граница. Трябва да видим, че g също е монотонна. За целта вземаме произволни $\bar{x}$ и $\bar{y}$ от $\mathbb{N}_\perp^k$, такива че $\bar{x} \sqsubseteq \bar{y}$. Защо $g(\bar{x}) \sqsubseteq g(\bar{y})$?

Да изберем произволно $n \in \mathbb{N}$. От монотонността на f_n и от това, че $f_n \sqsubseteq g$ ще имаме

$$f_n(\bar{x}) \ \sqsubseteq \ f_n(\bar{y}) \ \sqsubseteq \ g(\bar{y}).$$

Тъй като n беше произволно, това означава, че $g(\bar{y})$ е горна граница на редицата $\{f_n(\bar{x})\}_n$. Но $g(\bar{x})$ е точната ѝ горна граница (спомнете си за поточковата дефиниция (3.20) на т.г.гр.), и значи $g(\bar{x}) \sqsubseteq g(\bar{y})$, което и искахме да покажем.

Ако това доказателство ви се вижда твърде отвлечено, ето ви едно "по-земно" $\ddot{\smile}$. То използва *Твърдение* 3.9, което може би интуитивно е по-ясно, защото има аналог и при частичните функции, с които вече сме стари познайници.

Отново тръгваме от произволни $\bar{x}$ и $\bar{y}$ от $\mathbb{N}_\perp^k$, за които $\bar{x} \sqsubseteq \bar{y}$. За да покажем, че $g(\bar{x}) \sqsubseteq g(\bar{y})$, разглеждаме двата случая за стойността $g(\bar{x})$:

1 сл. $g(\bar{x}) = \perp$.
Тук няма какво да се доказва, защото $\perp \sqsubseteq g(\bar{y})$, каквото и да е $g(\bar{y})$.

2 сл. $g(\bar{x}) = z \in \mathbb{N}$.
Тогава съгласно *Твърдение* 3.9 б) съществува $n \in \mathbb{N}$, такова че $f_n(\bar{x}) = z$. Но $\bar{x} \sqsubseteq \bar{y}$, а f_n е монотонна. Така $f_n(\bar{x}) \sqsubseteq f_n(\bar{y})$, и понеже $f_n(\bar{x}) = z \in \mathbb{N}$, то и $f_n(\bar{y}) = z \in \mathbb{N}$. Сега от обратната посока на същото *Твърдение* 3.9 б) получаваме, че и $g(\bar{y}) = z$ и значи $g(\bar{x}) = g(\bar{y})$, откъдето, разбира се, и $g(\bar{x}) \sqsubseteq g(\bar{y})$. □

Като комбинираме резултатът, който току-що получихме, с доказаното в предишната глава *Твърдение* 2.5 за декартови произведения на ОС, можем да твърдим, че:

Следствие 3.3. За произволни положителни $m_1, \ldots, m_k$, структурата

$$\boldsymbol{\mathcal{M}} \ = \ (\mathcal{M}_{m_1}^{\perp} \times \cdots \times \mathcal{M}_{m_k}^{\perp},\ \sqsubseteq,\ (\Omega^{(m_1)}, \ldots, \Omega^{(m_k)}))$$

е област на Скот.

В тази област на Скот по-надолу ще дефинираме *денотационната семантика по име* на произволна рекурсивна програма от нашия език *REC*.

3.5.4 Непрекъснатост на термалните оператори в ОС $\mathcal{M}$

Областта на Скот, в която ще дефинираме денотационната семантика по име $D_N(R)$ на рекурсивна програма R, вече се изясни. От опита си с денотационната семантика *по стойност* $D_V(R)$ знаем, че тя се въвежда посредством най-малка неподвижна точка на подходящ *непрекъснат* оператор, определен от дефинициите на R. Задачата ни в този раздел е да видим, че въпросният операторът, който вече ще действа в новата ОС

$$\boldsymbol{\mathcal{M}} \;=\; (\mathcal{M}_{m_1}^{\perp} \times \cdots \times \mathcal{M}_{m_k}^{\perp},\ \sqsubseteq,\ (\Omega^{(m_1)}, \ldots, \Omega^{(m_k)}))$$

също е непрекъснат.

За целта ще повторим пътя, изминат при дефинирането на $D_V(R)$: ще дадем дефиниция за *стойност на терм* $\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$, когато X_i се заместват с елементи на $\mathbb{N}_\perp$, а F_j — с тотални функции от $\mathcal{F}_{m_j}^{\perp}$, и ще покажем, че е непрекъснат всеки термален оператор, разглеждан като изображение над *монотонните функции*.

Да напомним индуктивната дефиницията на *терм* в езика REC:

1) Всяка константа n е терм, $n \in \mathbb{N}$.

2) За всяко $i = 1, 2, \ldots$, обектовата променлива X_i е терм.

3) Ако τ_1 и τ_2 са термове, а *op* е базисна операция, то $(\tau_1\ op\ \tau_2)$ е терм.

4) Ако τ_1, τ_2 и τ_3 са термове, то `if` τ_1 `then` τ_2 `else` τ_3 е терм.

5) Ако F_i е m-местна функционална променлива, а $\tau_1, \ldots, \tau_m$ са термове, то $F_i(\tau_1, \ldots, \tau_m)$ е терм.

Нека $\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$ е терм, в който всяка от функционалните променливи F_i е на m_i аргумента, $1 \leq i \leq k$. Ще зададем *стойност* на τ, когато неговите обектови променливи X_i заместваме с елементи на $\mathbb{N}_\perp$, а функционалните променливи F_j — с функции от $\mathcal{F}_{m_j}^{\perp}$.

За целта избираме произволни $x_1 \in \mathbb{N}_\perp, \ldots, x_n \in \mathbb{N}_\perp$ и k на брой функции $f_1 \in \mathcal{F}_{m_1}^{\perp}, \ldots, f_k \in \mathcal{F}_{m_k}^{\perp}$. Тези функции са *произволни*; в този момент не е необходимо да са монотонни.

Определение 3.16. *Стойността* на $\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$ в точката $(x_1, \ldots, x_n, f_1, \ldots, f_k)$:

$$\tau(x_1, \ldots, x_n, f_1, \ldots, f_k) \qquad (\text{или } \ \tau(\bar{x}, \bar{f})),$$

дефинираме с индукция по построението на терма τ както следва:

1) Ако τ е константата n, то $\tau(\bar{x}, \bar{f}) = n$.

2) Ако τ е обектовата променлива X_i, то $\tau(\bar{x}, \bar{f}) = x_i$.

3) Ако τ е от вида τ_1 *op* τ_2, то

$$\tau(\bar{x}, \bar{f}) \;=\; \tau_1(\bar{x}, \bar{f})\ op^*\ \tau_2(\bar{x}, \bar{f}).$$

Забележете, че всяка базисна функция *op* (която е функция над $\mathbb{N}$) разширяваме до функция над $\mathbb{N}_\perp$ като вземаме нейното *естествено продължение* op^*.

4) Ако τ е от вида `if` τ_1 `then` τ_2 `else` τ_3, то

$$\tau(\bar{x}, \bar{f}) \;=\; \begin{cases} \tau_2(\bar{x}, \bar{f}), & \text{ако } \tau_1(\bar{x}, \bar{f}) > 0 \\ \tau_3(\bar{x}, \bar{f}), & \text{ако } \tau_1(\bar{x}, \bar{f}) = 0 \\ \perp, & \text{ако } \tau_1(\bar{x}, \bar{f}) = \perp. \end{cases}$$

5) Ако τ е от вида $F_i(\tau_1, \ldots, \tau_{m_i})$, то

$$\tau(\bar{x}, \bar{f}) \;=\; f_i(\tau_1(\bar{x}, \bar{f}), \ldots, \tau_{m_i}(\bar{x}, \bar{f})).$$

Да отбележим, че тук вече всеки всеки терм τ има стойност, която е елемент на $\mathbb{N}_\perp$.

Определение 3.17. Нека $\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$ е терм, в който всяка от променливите F_i е на m_i аргумента, $1 \leq i \leq k$ Термът τ определя *термалния оператор*

$$\Gamma_\tau \colon\ \mathcal{F}^\perp_{m_1} \times \cdots \times \mathcal{F}^\perp_{m_k} \;\longrightarrow\; \mathcal{F}^\perp_n,$$

дефиниран като:

$$\Gamma_\tau(f_1, \ldots, f_k)(x_1, \ldots, x_n) \;=\; \tau(x_1, \ldots, x_n, f_1, \ldots, f_k)$$

за всички $\bar{x} \in \mathbb{N}^n_\perp$ и $f_i \in \mathcal{F}^\perp_{m_i}, 1 \leq i \leq k$.

Вече видяхме (*Пример* 3.1), че за някои термове τ операторът Γ_τ може дори да не е монотонен. Оказва се, че когато го ограничим до изображение върху *монотонните функции*, той вече е не само монотонен, но и непрекъснат.

Първо да съобразим, че приложен върху монотонни функции, всеки термален оператор отново връща монотонна функция.

Твърдение 3.12. Нека $\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$ е терм, в който всяка функционална променлива F_i е на m_i аргумента, $1 \leq i \leq k$. Нека още $f_1 \in \mathcal{M}^\perp_{m_1}, \ldots, f_k \in \mathcal{M}^\perp_{m_k}$ са монотонни функции. Тогава и функцията $\Gamma_\tau(f_1, \ldots, f_k)$ е монотонна.

Доказателство. Избираме произволни $\bar{x}$ и $\bar{y}$ от $\mathbb{N}_\perp^n$, за които $\bar{x} \sqsubseteq \bar{y}$. Трябва да покажем, че

$$\Gamma_\tau(\bar{f})(\bar{x}) \sqsubseteq \Gamma_\tau(\bar{f})(\bar{y}).$$

Ще разсъждаваме с индукция по построението на терма τ.

1) Ако τ е константата n, то $\Gamma_\tau(\bar{f})$ е константната функция $\lambda\bar{x}.n$, която, разбира се, е монотонна.

2) Ако τ е обектовата променлива X_i, то

$$\Gamma_\tau(\bar{f})(\bar{x}) = x_i, \quad \text{а} \quad \Gamma_\tau(\bar{f})(\bar{y}) = y_i.$$

Но $\bar{x} \sqsubseteq \bar{y}$, откъдето и $x_i \sqsubseteq y_i$, и следователно $\Gamma_\tau(\bar{f})(\bar{x}) \sqsubseteq \Gamma_\tau(\bar{f})(\bar{y})$.

3) Нека $\tau = \tau_1 \; op \; \tau_2$. Тогава по определение

$$\Gamma_\tau(\bar{f})(\bar{x}) = \Gamma_{\tau_1}(\bar{f})(\bar{x}) \; op^* \; \Gamma_{\tau_2}(\bar{f})(\bar{x}).$$

Избираме отново $\bar{x} \sqsubseteq \bar{y}$. За да покажем, че $\Gamma_\tau(\bar{f})(\bar{x}) \sqsubseteq \Gamma_\tau(\bar{f})(\bar{y})$, ще се възползваме от *Твърдение* 3.8.

Нека $\Gamma_\tau(\bar{f})(\bar{x}) = z \in \mathbb{N}$. Понеже op^* е точна, непременно ще съществуват *числа* z_1 и z_2, такива че

$$\Gamma_{\tau_1}(\bar{f})(\bar{x}) = z_1, \quad \Gamma_{\tau_2}(\bar{f})(\bar{x}) = z_2 \quad \text{и} \quad z_1 \; op^* z_2 = z.$$

По индукционното предположение

$$\Gamma_{\tau_i}(\bar{f})(\bar{x}) \sqsubseteq \Gamma_{\tau_i}(\bar{f})(\bar{y}), \quad \text{за} \quad i = 1, 2,$$

и следователно $\Gamma_{\tau_i}(\bar{f})(\bar{y}) = z_i, \; i = 1, 2$. Тогава веднага

$$\Gamma_\tau(\bar{f})(\bar{y}) \stackrel{\text{деф}}{=} \Gamma_{\tau_1}(\bar{f})(\bar{y}) \; op^* \; \Gamma_{\tau_2}(\bar{f})(\bar{y}) = z_1 \; op^* z_2 = z.$$

4) Случаят $\tau = \texttt{if } \tau_1 \texttt{ then } \tau_2 \texttt{ else } \tau_3$ е съвсем аналогичен на горния.

5) Нека накрая τ е от вида $F_i(\tau_1, \ldots, \tau_{m_i})$. По определение

$$\Gamma_\tau(\bar{f})(\bar{x}) = f_i(\Gamma_{\tau_1}(\bar{f})(\bar{x}), \ldots, \Gamma_{\tau_{m_i}}(\bar{f})(\bar{x})).$$

Да означим

$$\Gamma_{\tau_1}(\bar{f})(\bar{x}) = z_1, \; \ldots \; , \Gamma_{\tau_{m_i}}(\bar{f})(\bar{x}) = z_{m_i},$$

и съответно

$$\Gamma_{\tau_1}(\bar{f})(\bar{y}) = t_1, \; \ldots \; , \Gamma_{\tau_{m_i}}(\bar{f})(\bar{y}) = t_{m_i}.$$

От индуктивната хипотеза за всеки от термовете $\tau_1, \; \ldots \; , \tau_{m_i}$ ще имаме

$$z_1 \sqsubseteq t_1, \; \ldots \; , z_{m_i} \sqsubseteq t_{m_i}.$$

Тук е моментът да използваме, че функциите $f_1, \; \ldots \; , f_k$ са монотонни, в случая — че f_i е монотонна. Така от горните неравенства ще получим

$$\Gamma_\tau(\bar{f})(\bar{x}) \stackrel{\text{деф}}{=} f_i(z_1, \; \ldots \; , z_{m_i}) \sqsubseteq f_i(t_1, \; \ldots \; , t_{m_i}) \stackrel{\text{деф}}{=} \Gamma_\tau(\bar{f})(\bar{y}).$$

□

Сега се насочваме към твърдението, че всеки термален оператор, ограничен до монотонните функции, е непрекъснат. Първо да проверим, че всеки такъв оператор е монотонен.

Твърдение 3.13. За всеки терм $\tau(X_1, \ldots, X_l, F_1, \ldots, F_k)$ операторът

$$\Gamma_\tau : \mathcal{M}_{m_1} \times \cdots \times \mathcal{M}_{m_k} \longrightarrow \mathcal{M}_l$$

е монотонен.

Да отбележим, че съгласно предишното твърдение, стойностите на Γ_τ наистина попадат в $\mathcal{M}_l$.

Доказателство. Да вземем функции

$$\bar{f} \in \mathcal{M}_{m_1} \times \cdots \times \mathcal{M}_{m_k} \quad \text{и} \quad \bar{g} \in \mathcal{M}_{m_1} \times \cdots \times \mathcal{M}_{m_k},$$

такива, че $\bar{f} \sqsubseteq \bar{g}$. Трябва да покажем, че $\Gamma_\tau(\bar{f}) \sqsubseteq \Gamma_\tau(\bar{g})$.
Отново разсъждаваме с индукция, която следва построението на τ.

1) и 2) Когато τ е константа или променлива, $\Gamma_\tau(\bar{f})$ не зависи от $\bar{f}$ и твърдението е очевидно.

3) Нека τ е от вида $\tau_1 \; op \; \tau_2$. От индуктивното предположение имаме

$$\Gamma_{\tau_1}(\bar{f}) \sqsubseteq \Gamma_{\tau_1}(\bar{g}) \quad \text{и} \quad \Gamma_{\tau_2}(\bar{f}) \sqsubseteq \Gamma_{\tau_2}(\bar{g}).$$

Сега да вземем произволно $\bar{x} \in \mathbb{N}_\perp^l$. От горните неравенства, следвайки дефиницията на поточковата наредба $\sqsubseteq$, получаваме

$$\Gamma_{\tau_1}(\bar{f})(\bar{x}) \sqsubseteq \Gamma_{\tau_1}(\bar{g})(\bar{x}) \quad \text{и} \quad \Gamma_{\tau_2}(\bar{f})(\bar{x}) \sqsubseteq \Gamma_{\tau_2}(\bar{g})(\bar{x}).$$

Отчитайки, че op^* е точна, и следователно — монотонна, ще имаме:

$$\Gamma_\tau(\bar{f})(\bar{x}) \stackrel{\text{деф}}{=} \Gamma_{\tau_1}(\bar{f})(\bar{x}) \; op^* \; \Gamma_{\tau_2}(\bar{f})(\bar{x}) \sqsubseteq$$

$$\Gamma_{\tau_1}(\bar{g})(\bar{x}) \; op^* \; \Gamma_{\tau_2}(\bar{g})(\bar{x}) \stackrel{\text{деф}}{=} \Gamma_\tau(\bar{g})(\bar{x}).$$

Сега от $\Gamma_\tau(\bar{f})(\bar{x}) \sqsubseteq \Gamma_\tau(\bar{g})(\bar{x})$ и това, че $\bar{x}$ беше произволно, достигаме финално до $\Gamma_\tau(\bar{f}) \sqsubseteq \Gamma_\tau(\bar{g})$.

4) Нека τ е от вида `if` τ_1 `then` τ_2 `else` τ_3. Да вземем отново произволна $\bar{x} \in \mathbb{N}_\perp^l$. Имаме 3 възможности:

1 сл. $\Gamma_{\tau_1}(\bar{f})(\bar{x}) = 0$. Тогава

$$\Gamma_\tau(\bar{f})(\bar{x}) \stackrel{\text{деф}}{=} \Gamma_{\tau_3}(\bar{f})(\bar{x}).$$

От и.х. имаме, че $\Gamma_{\tau_1}(\bar{f})(\bar{x}) \sqsubseteq \Gamma_{\tau_1}(\bar{g})(\bar{x})$ и следователно $\Gamma_{\tau_1}(\bar{g})(\bar{x}) = 0$ също. Но в такъв случай

$$\Gamma_\tau(\bar{g})(\bar{x}) \overset{\text{деф}}{=} \Gamma_{\tau_3}(\bar{g})(\bar{x}) \overset{\text{и.х.}}{\sqsupseteq} \Gamma_{\tau_3}(\bar{f})(\bar{x}) \overset{\text{деф}}{=} \Gamma_\tau(\bar{f})(\bar{x}).$$

2 сл. $\Gamma_{\tau_1}(\bar{f})(\bar{x}) > 0$ — аналогичен на горния.

3 сл.$\Gamma_{\tau_1}(\bar{f})(\bar{x}) = \bot$. Този случай е най-лесен: тук по определение $\Gamma_\tau(\bar{f})(\bar{x}) = \bot$ и следователно $\Gamma_\tau(\bar{f})(\bar{x}) \sqsubseteq \Gamma_\tau(\bar{g})(\bar{x})$.

5) Когато τ е от вида $F_i(\tau_1, \ldots, \tau_{m_i})$, разсъждаваме както в случай 3), като използваме монотонността на f_i.

□

Вече сме готови да покажем, че всеки термален оператор е непрекъснат.

Твърдение 3.14. За всеки терм $\tau(X_1, \ldots, X_l, F_1, \ldots, F_k)$ операторът

$$\Gamma_\tau : \mathcal{M}_{m_1} \times \cdots \times \mathcal{M}_{m_k} \longrightarrow \mathcal{M}_l$$

е непрекъснат.

Доказателство. Да вземем произволна монотонно растяща редица

$$\bar{f}^0 \sqsubseteq \bar{f}^1 \sqsubseteq \ldots \quad \text{в декартовото произведение } \mathcal{M}_{m_1} \times \cdots \times \mathcal{M}_{m_k}$$

и да означим с $\bar{g} = \bigsqcup_n \bar{f}^n$ нейната точна горна граница. Съгласно *Твърдение* 3.13, което току-що доказахме, Γ_τ е монотонен, и значи ще е монотонно растяща и редицата

$$\Gamma_\tau(\bar{f}^0) \sqsubseteq \Gamma_\tau(\bar{f}^1) \sqsubseteq \ldots \tag{3.24}$$

Тогава тази редица ще има точна горна граница $h = \bigsqcup_n \Gamma_\tau(\bar{f}^n)$. Трябва да покажем, че

$$\Gamma_\tau(\bigsqcup_n \bar{f}^n) = \bigsqcup_n \Gamma_\tau(\bar{f}^n), \quad \text{т.е.} \quad \Gamma_\tau(\bar{g}) = h. \tag{3.25}$$

Едната половина на това равенство — включването $\bigsqcup_n \Gamma_\tau(\bar{f}^n) \sqsubseteq \Gamma_\tau(\bigsqcup_n \bar{f}^n)$ получаваме директно от монотонността на Γ_τ. Наистина, за произволен индекс n имаме, че

$$\bar{f}^n \sqsubseteq \bigsqcup_n \bar{f}^n,$$

откъдето по монотонността на Γ_τ:

$$\Gamma_\tau(\bar{f}^n) \sqsubseteq \Gamma_\tau(\bigsqcup_n \bar{f}^n),$$

т.е. $\Gamma_\tau(\bigsqcup_n \bar{f}^n)$ е *горна граница* за редицата $\{\Gamma_\tau(\bar{f}^n)\}_n$. Следователно $\Gamma_\tau(\bigsqcup_n \bar{f}^n)$ мажорира и точната ѝ горна граница, с други думи, вярно е, че

$$\bigsqcup_n \Gamma_\tau(\bar{f}^n) \ \sqsubseteq \ \Gamma_\tau(\bigsqcup_n \bar{f}^n). \tag{3.26}$$

Сега се насочваме към обратното включване

$$\Gamma_\tau(\bigsqcup_n \bar{f}^n) \ \sqsubseteq \ \bigsqcup_n \Gamma_\tau(\bar{f}^n) \quad \text{или все едно,} \quad \Gamma_\tau(\bar{g}) \sqsubseteq h. \tag{3.27}$$

Да видим, че за целта ще е достатъчно да покажем импликацията:

$$\Gamma_\tau(\bar{g})(\bar{x}) = y \in \mathbb{N} \implies \exists n \ \Gamma_\tau(\bar{f}^n)(\bar{x}) = y. \tag{3.28}$$

Наистина, да приемем за момент, че горното условие е в сила за всяко $\bar{x} \in \mathbb{N}_\perp^l$. За да покажем включването (3.27), вземаме произволно $\bar{x} \in \mathbb{N}_\perp^l$ и приемаме, че $\Gamma_\tau(\bar{g})(\bar{x}) = y \in \mathbb{N}$. Трябва да покажем, че и $\bigsqcup_n \Gamma_\tau(\bar{f}^n)(\bar{x}) = y$.

Наистина, прилагайки (3.28) към $\Gamma_\tau(\bar{g})(\bar{x}) = y \in \mathbb{N}$, получаваме, че съществува индекс n, такъв че $\Gamma_\tau(\bar{f}^n)(\bar{x}) = y$. По-горе видяхме ((3.24)), че редицата $\{\Gamma_\tau(\bar{f}^n)\}_n$ е монотонно растяща в $\mathcal{F}_l^\perp$:

$$\Gamma_\tau(\bar{f}^0) \ \sqsubseteq \Gamma_\tau(\bar{f}^1) \ \sqsubseteq \ldots,$$

откъдето по определението за поточковата наредба $\sqsubseteq$ ще имаме, че за избраното от нас $\bar{x} \in \mathbb{N}_\perp^l$ ще е монотонно растяща в $\mathbb{N}_\perp$ редицата от стойностите

$$\Gamma_\tau(\bar{f}^0)(\bar{x}) \ \sqsubseteq \ \Gamma_\tau(\bar{f}^1)(\bar{x}) \ \sqsubseteq \ \ldots.$$

Понеже n-тият член $\Gamma_\tau(\bar{f}^n)(\bar{x})$ на тази редица има стойност *естественото число* y, то и точната горна граница на редицата $\{\Gamma_\tau(\bar{f}^n)(\bar{x})\}_n$ ще има същата стойност y. Така показахме, че

$$\Gamma_\tau(\bar{g})(\bar{x}) \ = \ y \ = \ (\bigsqcup_n \Gamma_\tau(\bar{f}^n))(\bar{x}).$$

и понеже $\bar{x} \in \mathbb{N}_\perp^l$ беше произволно, то значи $\Gamma_\tau(\bar{g}) \sqsubseteq h$, и следователно (3.27) е в сила. Това включване, заедно с обратното (3.26) ни дават общо верността на (3.25), т.е. на условието за непрекъснатост на Γ_τ.

Сега вече имаме мотивация да покажем, че за всяко $\bar{x} \in \mathbb{N}_\perp^l$ е вярно (3.28):

$$\Gamma_\tau(\bar{g})(\bar{x}) = y \in \mathbb{N} \implies \exists n \ \Gamma_\tau(\bar{f}^n)(\bar{x}) = y.$$

Отново ще разсъждаваме с индукция по строежа на τ.

1) и 2) Когато τ е константа или обектова променлива, $\Gamma_\tau(\bar{f})$ е константен оператор и твърдението е очевидно.

3) Нека τ е от вида $\tau_1\ op\ \tau_2$. Избираме $\bar{x} \in \mathbb{N}_\perp^l$ и приемаме, че $\Gamma_\tau(\bar{g})(\bar{x}) = y \in \mathbb{N}$. Тогава

$$\Gamma_\tau(\bar{g})(\bar{x}) \stackrel{\text{деф}}{=} \Gamma_{\tau_1}(\bar{g})(\bar{x})\ op^*\ \Gamma_{\tau_2}(\bar{g})(\bar{x}) \ = \ y.$$

Понеже op^* е точна функция, а $y \in \mathbb{N}$, то трябва да съществуват $z_1 \in \mathbb{N}$ и $z_2 \in \mathbb{N}$, такива че:

$$\Gamma_{\tau_1}(\bar{g})(\bar{x}) \ = \ z_1, \quad \Gamma_{\tau_2}(\bar{g})(\bar{x}) \ = \ z_2 \quad \text{и} \quad z_1\ op^*\ z_2 \ = \ y.$$

Тогава от индукционното предположение ще имаме, че съществуват индекси n_1 и n_2:

$$\Gamma_{\tau_1}(\bar{f}^{n_1})(\bar{x}) \ = \ z_1 \quad \text{и} \quad \Gamma_{\tau_2}(\bar{f}^{n_2})(\bar{x}) \ = \ z_2.$$

Нека $n = max\{n_1, n_2\}$. Тогава със сигурност $\bar{f}^{n_1} \sqsubseteq \bar{f}^n$ и $\bar{f}^{n_2} \sqsubseteq \bar{f}^n$, откъдето поради монотонността на операторите Γ_{τ_i} ще имаме, че

$$\Gamma_{\tau_1}(\bar{f}^{n_1}) \ \sqsubseteq \ \Gamma_{\tau_1}(\bar{f}^n) \quad \text{и} \quad \Gamma_{\tau_2}(\bar{f}^{n_2}) \ \sqsubseteq \ \Gamma_{\tau_2}(\bar{f}^n).$$

Оттук, в частност, за $\bar{x} \in \mathbb{N}_\perp^l$ ще имаме (вече в $\mathbb{N}_\perp$):

$$\Gamma_{\tau_1}(\bar{f}^{n_1})(\bar{x}) \ \sqsubseteq \ \Gamma_{\tau_1}(\bar{f}^n)(\bar{x}) \quad \text{и} \quad \Gamma_{\tau_2}(\bar{f}^{n_2})(\bar{x}) \ \sqsubseteq \ \Gamma_{\tau_2}(\bar{f}^n)(\bar{x}).$$

Сега като вземем предвид, че $\Gamma_{\tau_i}(\bar{f}^{n_1})(\bar{x}) = z_i \ \in \mathbb{N}$, $i = 1, 2$, можем да твърдим, че

$$\Gamma_{\tau_1}(\bar{f}^n)(\bar{x}) \ = \ z_1 \quad \text{и} \quad \Gamma_{\tau_2}(\bar{f}^n)(\bar{x}) \ = \ z_2.$$

Така получаваме, че за това n условието (3.28) ще е вярно, защото:

$$\Gamma_\tau(\bar{f}^n)(\bar{x}) \stackrel{\text{деф}}{=} \Gamma_{\tau_1}(\bar{f}^n)(\bar{x})\ op^*\ \Gamma_{\tau_2}(\bar{f}^n)(\bar{x}) \ = \ z_1\ op^*\ z_2 \ = \ y.$$

4) Случаят $\tau \ = \ \texttt{if}\ \tau_1\ \texttt{then}\ \tau_2\ \texttt{else}\ \tau_3$ се разглежда по много подобен начин.

5) Нека $\tau \ = \ F_i(\tau_1, \ldots, \tau_{m_i})$. Да вземем произволно $\bar{x} \in \mathbb{N}_\perp^l$ и да приемем, че $\Gamma_\tau(\bar{g})(\bar{x}) = y \in \mathbb{N}$. По определение

$$\Gamma_\tau(\bar{g})(\bar{x}) \stackrel{\text{деф}}{=} g_i(\Gamma_{\tau_1}(\bar{g})(\bar{x}), \ldots, \Gamma_{\tau_{m_i}}(\bar{g})(\bar{x})) \ = \ y.$$

Тогава съществуват $z_1, \ldots, z_{m_i}$ от $\mathbb{N}_\perp$ (за които вече не можем да твърдим, че са естествени числа!), такива че

$$\Gamma_{\tau_1}(\bar{g})(\bar{x}) = z_1, \ldots, \Gamma_{\tau_{m_i}}(\bar{g})(\bar{x}) = z_{m_i} \quad \text{и} \quad g_i(z_1, \ldots, z_{m_i}) = y.$$

Да напомним, че растящата редица от k-орки

$$\bar{f}^0 \ \sqsubseteq \bar{f}^1 \ \sqsubseteq \ \ldots \qquad \text{има граница } \bar{g} = (g_1, \ldots, g_k).$$

Тогава i-тата компонента на тази редица

$$f_i^0 \sqsubseteq f_i^1 \sqsubseteq \ldots \quad \text{ще има граница } g_i.$$

Понеже $g_i(z_1, \ldots, z_{m_i}) = y \in \mathbb{N}$, по по *Твърдение* 3.9 б) ще съществува индекс n, за който

$$f_i^n(z_1, \ \ldots \ , z_{m_i}) = y.$$

Сега да вземем произволно $j \in \{1, \ \ldots \ m_i\}$. За $\Gamma_{\tau_j}(\bar{g})(\bar{x})$ имаме две възможности:
— $\Gamma_{\tau_j}(\bar{g})(\bar{x}) \stackrel{\text{деф}}{=} z_j \in \mathbb{N}$. Тогава по и.х. за терма τ_j ще съществува индекс n_j, за който $\Gamma_{\tau_j}(\bar{f}^{n_j})(\bar{x}) = z_j$.

— $\Gamma_{\tau_j}(\bar{g})(\bar{x}) \stackrel{\text{деф}}{=} z_j = \bot$. В този случай полагаме $n_j \stackrel{\text{деф}}{=} 0$.

Нека $N = max\{n_1, \ldots, n_{m_i}, n\}$. Тогава за всяко $j = 1, \ \ldots \ , m_j$, $\bar{f}^{n_j} \sqsubseteq \bar{f}^N$. Оттук поради монотонността на всеки от операторите Γ_{τ_j} ще имаме, че

$$\Gamma_{\tau_j}(\bar{f}^{n_j}) \sqsubseteq \Gamma_{\tau_j}(\bar{f}^N),$$

откъдето за фиксираното $\bar{x} \in \mathbb{N}_\bot^l$ ще имаме в $\mathbb{N}_\bot$, че:

$$\Gamma_{\tau_j}(\bar{f}^{n_j})(\bar{x}) \sqsubseteq \Gamma_{\tau_j}(\bar{f}^N)(\bar{x}).$$

Оттук, като разгледате двете възможности за z_j — да е число или да е $\bot$, получаваме

$$z_j \stackrel{\text{деф}}{=} \Gamma_{\tau_j}(\bar{g})(\bar{x}) \sqsubseteq \Gamma_{\tau_j}(\bar{f}^{n_j})(\bar{x}) \sqsubseteq \Gamma_{\tau_j}(\bar{f}^N)(\bar{x}). \tag{3.29}$$

Настъпи моментът да използваме, че нашите оператори работят върху монотонни функции. В случая ще се възползваме от монотонността f_i^N, за да покажем, че $\Gamma_\tau(\bar{f}^N)(\bar{x}) = y$. Да видим как става това:

$$\begin{aligned}
\Gamma_\tau(\bar{f}^N)(\bar{x}) &\stackrel{\text{деф}}{=} f_i^N(\Gamma_{\tau_1}(\bar{f}^N)(\bar{x}), \ldots, \Gamma_{\tau_{m_i}}(\bar{f}^N)(\bar{x})) \\
&\stackrel{(3.29)}{\sqsupseteq} f_i^N(\underbrace{\Gamma_{\tau_1}(\bar{g})(\bar{x})}_{z_1}, \ldots, \underbrace{\Gamma_{\tau_{m_i}}(\bar{g})(\bar{x})}_{z_{m_i}}) \qquad //f_i^N \text{ е монотонна} \\
&= f_i^N(z_1, \ \ldots \ , z_{m_i}) \stackrel{f_i^N \sqsupseteq f_i^n}{\sqsupseteq} f_i^n(z_1, \ \ldots \ , z_{m_i}) = y \stackrel{\text{деф}}{=} \Gamma_\tau(\bar{g})(\bar{x}),
\end{aligned}$$

Получихме, че $\Gamma_\tau(\bar{f}^N)(\bar{x}) \sqsupseteq y$, като y е число (не е $\bot$). Следователно $\Gamma_\tau(\bar{f}^N)(\bar{x}) = y$, с което приключва проверката на (3.28), а с това и доказателството на твърдението.

□

3.5.5 Как дефинираме $D_N(R)$?

Вече имаме опит в прилагането на *денотационния подход* към програмите от функционалния език REC и това, което ще направим тук, за да дефинираме *денотационна семантика по име*, ще ни изглежда доста познато.

Да вземем произволна програма R от езика REC:

$$R \begin{cases} \tau_0(X_1, \ldots, X_n, F_1, \ldots, F_k) \qquad \texttt{where} \\ F_1(X_1, \ldots, X_{m_1}) = \tau_1(X_1, \ldots, X_{m_1}, F_1, \ldots, F_k) \\ \vdots \\ F_i(X_1, \ldots, X_{m_i}) = \tau_i(X_1, \ldots, X_{m_i}, F_1, \ldots, F_k) \\ \vdots \\ F_k(X_1, \ldots, X_{m_k}) = \tau_k(X_1, \ldots, X_{m_k}, F_1, \ldots, F_k) \end{cases}$$

В този и другия раздел (с които всъщност прилючва главата за семантиките) ще считаме, че R е фиксирана.

Всеки от термовете $\tau_i(X_1, \ldots, X_{m_i}, F_1, \ldots, F_k)$ определя термален оператор Γ_{τ_i}, който разглеждаме като оператор над *монотонните функции* в $\mathbb{N}_\perp$:

$$\Gamma_{\tau_i} : \ \mathcal{M}_{m_1} \times \cdots \times \mathcal{M}_{m_k} \ \longrightarrow \ \mathcal{M}_{m_i}.$$

Твърдение 3.14 ни гарантира, че термалните оператори, ограничени до *монотонните функции*, вече са непрекъснати. Да означим с

$$\Gamma \ = \ \Gamma_{\tau_1} \times \cdots \times \Gamma_{\tau_k}$$

декартовото произведение на операторите $\Gamma_{\tau_1}, \ldots, \Gamma_{\tau_k}$. Тогава и Γ ще е непрекъснат, съгласно *Твърдение* 2.8.

Операторът Γ е от следния вид:

$$\Gamma : \mathcal{M}_{m_1} \times \cdots \times \mathcal{M}_{m_k} \ \longrightarrow \ \mathcal{M}_{m_1} \times \cdots \times \mathcal{M}_{m_k}.$$

Той е непрекъснат в областта на Скот

$$\boldsymbol{\mathcal{M}} \ = \ (\mathcal{M}_{m_1} \times \cdots \times \mathcal{M}_{m_k}, \ \sqsubseteq, \ (\Omega^{(m_1)}, \ldots, \Omega^{(m_k)})).$$

Тогава към Γ можем да приложим теоремата на Кнастер-Тарски и да получим, че той има най-малка неподвижна точка

$$f_\Gamma \ = \ (f_\Gamma^1, \ldots, f_\Gamma^k).$$

Да означим с $h : \mathbb{N}_\perp^n \longrightarrow \mathbb{N}_\perp$ функцията, определена от главата $\tau_0(X_1, \ldots, X_n, F_1, \ldots, F_k)$ на програмата R по следния начин:

$$h(x_1, \ldots, x_n) \stackrel{\text{деф}}{=} \tau_0(x_1, \ldots, x_n, f_\Gamma^1, \ldots, f_\Gamma^k).$$

Ако разсъждаваме по аналогия с денотационната семантика *по стойност*, би трябвало да положим $D_N(R) \stackrel{\text{деф}}{=} h$. Проблемът е, че h е функция в $\mathbb{N}_\perp$, докато нашата програма работи само над естествени числа. Затова трябва да "върнем" h в света на естествените числа, което ставаше чрез изображението $^\circ$, дефинирано с равенството (3.21). Сега вече можем да положим $D_N(R) = h^\circ$.

Определение 3.18. *Денотационна семантика с предаване на параметрите по име* на програмата R е функцията

$$D_N(R) : \mathbb{N}^n \multimap \mathbb{N},$$

която се определя с условното равенство:

$$D_N(R)(x_1, \ldots, x_n) \simeq \begin{cases} \tau_0(x_1, \ldots, x_n, f_\Gamma^1, \ldots, f_\Gamma^k)), & \text{ако } \tau_0(\bar{x}, f_\Gamma^1, \ldots, f_\Gamma^k)) \neq \perp \\ \neg!, & \text{иначе.} \end{cases}$$

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

3.6 Еквивалентност между денотационната и операционната семантика по име

Тук ще докажем, че денотационният и операционният подход към дефиниране на семантика *по име* са еквивалентни, т.е. определят една и съща функция. Ще се придържаме към схемата от раздел 3.3, в който доказахме същото, но за семантиките *по стойност*. Стъпките, които ще следваме, са следните:

1) Ще докажем твърдение от тип "теорема за коректност", което в случая гласи: за всяка рекурсивна програма R

$$O_N(R) \subseteq D_N(R).$$

Доказателството ще бъде подобно на доказателството на аналогичното *Твърдение* 3.4 за $O_V(R) \subseteq D_V(R)$, и дори малко по-просто от него, тъй като правилото за извод (3_N) не е толкова "капризно", колкото съответстващото му правило (3_V).

2) Отново ще дефинираме спомагателни функции $g_1, \ldots, g_k$, които се определят операционно, но този път *по име*, от декларациите на $F_1, \ldots, F_k$. Нашата цел ще бъде да покажем, че векторът $(g_1, \ldots, g_k)$ съвпада с най-малката неподвижна точка на оператора $\Gamma - f_\Gamma = (f_\Gamma^1, \ldots, f_\Gamma^k)$.

Тук възниква малък проблем, сързан с това, че от една страна, функциите g_i се определят чрез синтактична изводимост и следователно са функции в $\mathbb{N}$ (при това *частични*), докато функциите f_Γ^i са тотални функции в $\mathbb{N}_\perp$. Ще го разрешим, като въведем константата $\perp$ към базисните *термове* и съответно разширим дефиницията на "изводимост по име".

3) Като използваме тези нови функции, ще докажем твърдение от тип "теорема за пълнота", от която ще следва и $D_N(R) \subseteq O_N(R)$.

3.6.1 Доказателство на включването $O_N(R) \subseteq D_N(R)$

Ще продължим да спазваме уговорката функционалните термове да означаваме с $\mu, \mu_1, \mu_2 \ldots$. До края на лекцията под"стойност на терм" ще имаме предвид стойността му в света с $\perp$-ите, т.е. обектовите променливи ще се заместват с елементи на $\mathbb{N}_\perp$, а функционалните променливи — с тотални функции в $\mathbb{N}_\perp$.

По-надолу ще ни е нужна една спомагателна лема, която е обобщение на *Лема* 3.3.

Лема 3.5. (Лема за оценките) Нека $\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$, $\mu_1(F_1, \ldots, F_k), \ldots, \mu_n(F_1, \ldots, F_k)$ са термове, в които всяка от функционалните променливи F_i е на m_i аргумента, $1 \leq i \leq k$. Нека още $f_1 \in \mathcal{F}_{m_1}^\perp, \ldots, f_k \in \mathcal{F}_{m_k}^\perp$ са дадени тотални функции. Тогава

$$\tau[X_1/\mu_1, \ \ldots \ , X_n/\mu_n](f_1, \ldots, f_k) \ = \ \tau(\mu_1(\bar{f}), \ldots, \mu_n(\bar{f}), f_1, \ldots, f_k).$$

Доказателството е с рутинна индукция по построението на терма τ.

В частния случай, когато термовете $\mu_1, \ldots, \mu_n$ са константите $c_1, \ldots, c_n$, ще имаме, разбира се, че $\mu_i(\bar{f}) = c_i(\bar{f}) = c_i$, откъдето

$$\tau[X_1/c_1, \ \ldots \ , X_n/c_n](f_1, \ldots, f_k) \ = \ \tau(c_1, \ldots, c_n, f_1, \ldots, f_k). \qquad (3.30)$$

Сега се насочваме към доказателството на едно по-общо твърдение, от което като частен случай ще получим и включването $O_N(R) \subseteq D_N(R)$.Да напомним, че векторът $\bar{f} = (f_1, \ldots, f_k)$ е неподвижна точка на оператора

$\Gamma = \Gamma_{\tau_1} \times \cdots \times \Gamma_{\tau_k}$ точно когато той удовлетворява системата

$$(*) \begin{cases} f_1 = \Gamma_{\tau_1}(f_1, \ldots, f_k) \\ \vdots \\ f_i = \Gamma_{\tau_i}(f_1, \ldots, f_k) \\ \vdots \\ f_k = \Gamma_{\tau_k}(f_1, \ldots, f_k). \end{cases}$$

Твърдение 3.15. Нека $\mu(F_1, \ldots, F_k)$ е функционален терм, а $\bar{f} = (f_1, \ldots, f_k)$ е неподвижна точка на оператора $\Gamma = \Gamma_{\tau_1} \times \cdots \times \Gamma_{\tau_k}$. Тогава

$$R \vdash_N \mu \ \to \ d \quad \Longrightarrow \quad \mu(\bar{f}) = d. \tag{3.31}$$

Да се убедим най-напред, че от това твърдение наистина ще следва включването, което ни интересува:

Следствие 3.4. За всяка рекурсивна програма R

$$O_N(R) \subseteq D_N(R).$$

Доказателство. Да фиксираме произволна n-торка $\bar{c} \in \mathbb{N}^n$. Прилагаме *Твърдение* 3.15 за $\mu = \tau_0[\bar{X}/\bar{c}]$ и $\bar{f} = \bar{f}_\Gamma$. Ще имаме

$$\begin{aligned} O_N(R)(\bar{c}) \simeq d \quad &\overset{\text{деф}}{\Longleftrightarrow} \quad R \vdash_N \underbrace{\tau_0[\bar{X}/\bar{c}]}_{\mu} \ \to \ d \quad \overset{(3.31)}{\Longrightarrow} \quad \underbrace{\tau_0[\bar{X}/\bar{c}]}_{\mu}(\bar{f}_\Gamma) = d \\ &\overset{(3.30)}{\Longleftrightarrow} \quad \tau_0(\bar{c}, \bar{f}_\Gamma) = d \quad \overset{\text{деф}}{\Longleftrightarrow} \quad D_N(R)(\bar{c}) \simeq d. \end{aligned}$$

□

Сега се насочваме към доказателството на самото *Твърдение* 3.15.

Доказателство. Ще следваме схемата на доказателство на аналогичното *Твърдение* 3.4 при семантиките по стойност.

Нека $R \vdash_N \mu \ \to \ d$ с дължина на извода l. Отново разсъждаваме с пълна индукция по l, за да покажем, че $\mu(\bar{f}) = d$.

Тук би трябвало да разгледаме различните случаи от индуктивната дефиниция на функционалния терм μ. Но тъй като при всички тях, с изключение на последния, разсъжденията са съвсем същите, като при доказателството на *Твърдение* 3.4, ще ги пропуснем и ще се насочим директно към случая $\mu = F_i(\mu_1, \ldots, \mu_{m_i})$. (Да си спомним, че всички правила за извод, с изключение на последното правило (3_N), бяха едни и същи, тъй че горното наблюдение не е изненадващо.)

И тъй, нека с дължина на извода l

$$R \vdash_N \ F_i(\mu_1, \ldots, \mu_{m_i}) \ \to \ d$$

Ясно е, че това опростяване е изведено след прилагане на правилото (3_N) (както твърди и *Лема* 3.1). Следователно с дължина на извода, *по-малка* от l, е било изведено

$$R \vdash_N \tau_i[X_1/\mu_1, \ldots, X_{m_i}/\mu_{m_i}] \to d.$$

Тогава по индуктивната хипотеза

$$\tau_i[X_1/\mu_1, \ \ldots \ , X_{m_i}/\mu_{m_i}](\bar{f}) = d.$$

От лемата за оценките (*Лема* 3.5) имаме, че

$$\tau_i[X_1/\mu_1, \ \ldots \ , X_{m_i}/\mu_{m_i}](f_1, \ldots, f_k) \ = \ \tau_i(\mu_1(\bar{f}), \ldots, \mu_{m_i}(\bar{f}), f_1, \ldots, f_k),$$

откъдето

$$\tau_i(\underbrace{\mu_1(\bar{f})}_{x_1 \in \mathbb{N}_\perp}, \ldots, \underbrace{\mu_{m_i}(\bar{f})}_{x_{m_i} \in \mathbb{N}_\perp}, f_1, \ldots, f_k) = d.$$

Оттук

$$\Gamma_{\tau_i}(\bar{f})(x_1, \ldots, x_{m_i}) \stackrel{\text{деф}}{=} \tau_i(x_1, \ldots, x_n, f_1, \ldots, f_k) = d.$$

Но $\bar{f}$ е неподвижна точка на Γ, т.е. функциите $\bar{f}$ удовлетворяват системата $(*)$, откъдето в частност $f_i = \Gamma_{\tau_i}(\bar{f})$. Сега вече можем да намерим стойността на нашия терм μ в т. $\bar{f}$:

$$\begin{aligned} \mu(\bar{f}) \ &= \ F_i(\mu_1, \ldots, \mu_{m_i})(\bar{f}) \stackrel{\text{деф}}{=} f_i(\mu_1(\bar{f}), \ \ldots \ , \mu_{m_i}(\bar{f})) \\ &= \ f_i(x_1, \ldots, x_{m_i}) \ = \ \Gamma_{\tau_i}(\bar{f})(x_1, \ldots, x_{m_i}) = d. \end{aligned}$$

□

3.6.2 Доказателство на включването $D_N(R) \subseteq O_N(R)$

Искаме да въведем спомагателни функции $g_1, \ldots, g_k$, които са такива, че $g_i : \mathbb{N}^{m_i} \multimap \mathbb{N}$ е функцията, която се определя операционно по име от F_i. По-конкретно, ако започнем изчислението по име от началния терм $F_i(c_1, \ldots, c_{m_i})$, резултатът, който ще получим (ако изчислението въобще завърши), ще е стойността на g_i в т. $(c_1, \ldots, c_{m_i})$. Идеята ни, по аналогия със случая на семантиките по стойност, е да покажем, че функциите $g_1, \ldots, g_k$ съвпадат с $(f^1_\Gamma, \ldots, f^k_\Gamma)$. Но функциите f^i_Γ са изображения в $\mathbb{N}_\perp$, докато функциите g_i очевидно са дефинирани само върху естествените числа.

Това несъответствие ще разрешим така: ще разширим понятието *терм*, като добавим клауза, че $\perp$ също е терм. Новото понятие ще наричаме *обобщен терм*. Неговата дефиниция повтаря дефиницията 3.1 на терм, като разликата е само и единствено в първия пункт от дефиницията, където искаме и $\perp$ да е терм.

1) Всяко $c \in \mathbb{N}_\perp$ е обобщен терм.

2) Всяка обектовата променлива X_i е обобщен терм за $i = 1, 2, \ldots$.

3) Ако τ_1 и τ_2 са обобщени термове, а op е базисна операция, то $(\tau_1\ op\ \tau_2)$ е обобщен терм.

4) Ако τ_1, τ_2 и τ_3 са обобщени термове, то $\texttt{if}\ \tau_1\ \texttt{then}\ \tau_2\ \texttt{else}\ \tau_3$ е обобщен терм.

5) Ако F_i е m-местна функционална променлива, а $\tau_1, \ldots, \tau_m$ са обобщени термове, то $F_i(\tau_1, \ldots, \tau_m)$ е обобщен терм.

Примери. $\perp, \quad \perp + X_1, \quad F_1(X_1, \perp), \quad \texttt{if}\ X_1 == \perp\ \texttt{then}\ 5\ \texttt{else}\ \perp$

Под *опростяване* $\mu\ \to\ c$ ще разбираме израз, в който μ е функционален обобщен терм. Константата c също може да е както от $\mathbb{N}$, така и $\perp$, но както ще съобразим след малко, от R няма да можем да извеждаме опростявания от вида $\mu\ \to\ \perp$. Това ще е така, защото системата за *синтактичен извод по име* на опростяване видоизменяме по следния начин:

$$\text{за всяко } c \in \mathbb{N}: \qquad \frac{}{c\ \to\ c} \quad (0)$$

$$\frac{\mu_1\ \to\ c_1 \quad \mu_2\ \to\ c_2 \quad c_1\ op^*\ c_2 = c \in \mathbb{N}}{\mu_1\ op\ \mu_2\ \to\ c} \quad (1)$$

$$\frac{\mu_1\ \to\ c_1 \quad \mu_2\ \to\ c \quad c_1 > 0}{\texttt{if}\ \mu_1\ \texttt{then}\ \mu_2\ \texttt{else}\ \mu_3\ \to\ \texttt{c}} \quad (2_\mathbf{t})$$

$$\frac{\mu_1\ \to\ 0 \quad \mu_3\ \to\ c}{\texttt{if}\ \mu_1\ \texttt{then}\ \mu_2\ \texttt{else}\ \mu_3\ \to\ \texttt{c}} \quad (2_\mathbf{f})$$

$$\frac{\tau_i[X_1/\mu_1, \ldots, X_{m_i}/\mu_{m_i}] \to c}{F_i(\mu_1, \ldots, \mu_{m_i})\ \to\ c} \quad (3_N)$$

Лема 3.6. Ако $R \vdash_N \mu\ \to\ c$, то $c \in \mathbb{N}$.

Доказателство. Банална индукция по дължината на извода $R \vdash_N \mu\ \to\ c$. □

Да отбележим също, че не е възможно $R \vdash_N \perp\ \to\ c$ за никоя константа c.

Неформално можем да кажем, че ако $R \vdash_N \mu\ \to\ c$ и в μ има $\perp$, то той на практика не е участвал в извода на опростяването и с какъвто и функционален терм да го заменим, ако означим новия терм с μ^*, то ще продължава да е вярно, че $R \vdash_N \mu^*\ \to\ c$ (като изводът ще е абсолютно същият).

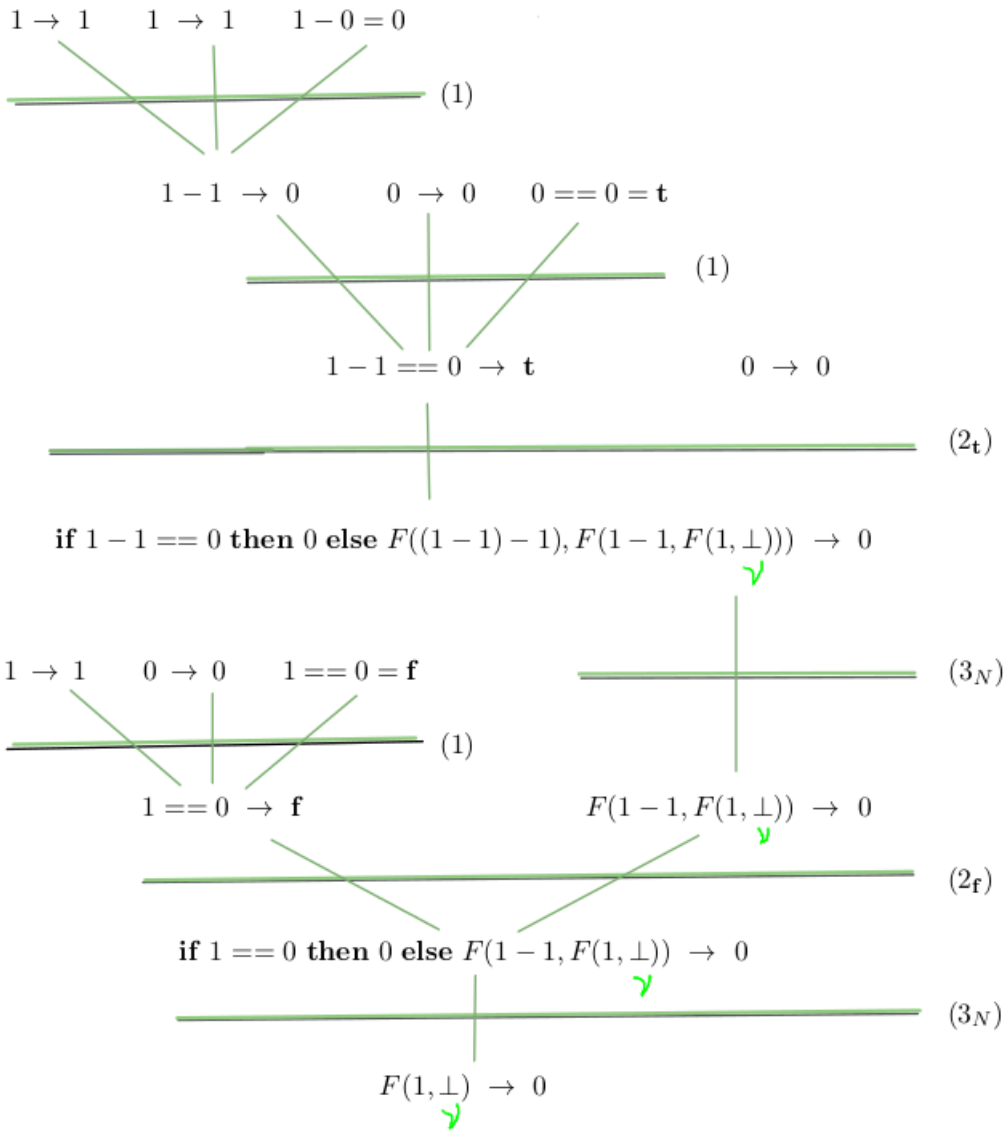
По-надолу ще формализираме и докажем строго това наблюдение, но ето един пример, който илюстрира явлението.

Пример. Дадена е програмата

R : `F( X, Y)  where`

`F( X, Y) = if X = 0 then 0 else F( X − 1, F( X, Y) )`

Това, което следва, е дървото на извода на опростяването $F(1, \perp) \to 0$ от R. Ясно е, че в него можем да заместваме $\perp$ с какъвто си искаме затворен терм ν — същият този извод ще е извод и на $F(1, \nu) \to 0$.



За да формализираме горното наблюдение, въвеждаме следното спомагателно понятие: при фиксиран функционален терм ν полагаме

$S(\mu) = \{\mu^* | \mu^*$ се получава от μ след заместване на *някои* участия на $\bot$ с $\nu\}$.

В горното определение под *някои* участия имаме пред вид евентуално и 0 на брой участия. Ето и формалното определение на множеството $S(\mu)$, което е с индукция по построението на μ:

0) Ако $\mu = c \in \mathbb{N}$, то $S(\mu) = \{c\}$; ако $\mu = \bot$, то $S(\mu) = \{\bot, \nu\}$.

1) Ако $\mu = \mu_1 \; op \; \mu_2$, то

$$S(\mu) = \{\mu_1^* \; op \; \mu_2^* \mid \mu_1^* \in S(\mu_1) \;\&\; \mu_2^* \in S(\mu_2)\}.$$

2) Ако $\mu = \texttt{if}\ \mu_1\ \texttt{then}\ \mu_2\ \texttt{else}\ \mu_3$ то

$$S(\mu) = \{\texttt{if}\ \mu_1^*\ \texttt{then}\ \mu_2^*\ \texttt{else}\ \mu_3^* \mid \mu_1^* \in \texttt{S}(\mu_1) \;\&\; \mu_2^* \in \texttt{S}(\mu_2) \;\&\; \mu_3^* \in \texttt{S}(\mu_3)\}.$$

3) Ако $\mu = F_i(\mu_1, ..., \mu_{m_i})$, то

$$S(\mu) = \{F_i(\mu_1^*, ..., \mu_{m_i}^*) \mid \mu_1^* \in S(\mu_1) \;\&\; \ldots \;\&\; \mu_{m_i}^* \in S(\mu_{m_i})\}.$$

Определението на множеството $S(\mu)$ е за дадено фиксирано ν, така че по-коректно би било да пишем $S_\nu(\mu)$. За да не претоварваме записа, по-надолу ще пишем и само $S(\mu)$, ако това не води до неясноти.

Пример. Ако $\mu = F(\bot, \bot)$, то $S(\mu) = \{F(\bot, \bot), F(\nu, \bot), F(\bot, \nu), F(\nu, \nu)\}$.

Ето и формално доказателство на това, което наблюдавахме по-горе, при това, в по-обща постановка. По-точно, ще докажем, че ако имаме извод на опростяване $\mu \to c$ и в μ участват $\bot$-и, то произволен брой участия на $\bot$ в μ могат да бъдат заместени с друг функционален терм ν, като същият извод ще извежда и новото опростяване.

Твърдение 3.16. Нека ν е обобщен функционален терм и $R \vdash_N \mu \to c$. Тогава $R \vdash_N \mu^* \to c$ за всеки терм $\mu^* \in S_\nu(\mu)$.

Доказателство. Индукция по дефиницията на терма μ.

1) Ако $\mu = c \in \mathbb{N}$, то $S(\mu) = \{c\}$ и твърдението е очевидно. Случаят $\mu = \bot$ е невъзможен, както вече отбелязахме.

2) $\mu = X_i$ — също невъзможен.

3) Ако $\mu = \mu_1\ op\ \mu_2$, то за някои константи c_1 и c_2 *преди това* от R са били изведени по име опростяванията $\mu_1 \to c_1$ и $\mu_2 \to c_2$, и освен това $c_1\ op\ c_2 = c$.

   Да вземем произволен терм $\mu^* \in S(\mu)$. От определението на $S(\mu)$ се вижда, че в такъв случай $\mu^* = \mu_1^*\ op\ \mu_2^*$, за някои $\mu_1^* \in S(\mu_1)$ и $\mu_2^* \in S(\mu_2)$. По индуктивната хипотеза $R \vdash_N \mu_1^* \to c_1$ и $R \vdash_N \mu_2^* \to c_2$, откъдето по правилото (1) веднага получаваме $R \vdash_N \underbrace{\mu_1^*\ op\ \mu_2^*}_{\mu^*} \to c$.

4) Случаят $\mu = \texttt{if}\ \mu_1\ \texttt{then}\ \mu_2\ \texttt{else}\ \mu_3$ се разглежда аналогично на горния.

5) Нека накрая $\mu = F_i(\mu_1, ..., \mu_{m_i})$. Ясно е, че преди това е било изведено

$$R \vdash_N \tau_i[X_1/\mu_1, \ldots, X_{m_i}/\mu_{m_i}] \to c.$$

   Нека $\mu^* \in S(\mu)$. Тогава $\mu^* = F_i(\mu_1^*, ..., \mu_{m_i}^*)$ за някои $\mu_1^* \in S(\mu_1), \ldots$ $\mu_{m_i}^* \in S(\mu_{m_i})$. По индуктивната хипотеза

$$R \vdash_N \tau_i[X_1/\mu_1^*, \ldots, X_{m_i}/\mu_{m_i}^*] \to c,$$

   откъдето по правилото (3_N): $R \vdash_N \underbrace{F_i(\mu_1^*, ..., \mu_{m_i}^*)}_{\mu^*} \to c$.

□

Сега вече полагаме

$$g_i(c_1, \ldots, c_{m_i}) \stackrel{\text{деф}}{=} \begin{cases} d, & \text{ако } R \vdash F_i(c_1, \ldots, c_{m_i}) \to d \\ \bot, & \text{иначе.} \end{cases} \tag{3.32}$$

за всички $c_1 \in \mathbb{N}_\bot, \ldots, c_{m_i} \in \mathbb{N}_\bot$ и $d \in \mathbb{N}$.

Дали g_i са еднозначни функции? Да, с рутинна индукция по дължината на извода се вижда, че

$$R \vdash_N \mu \to c \quad \text{и} \quad R \vdash_N \mu \to d \quad \Longrightarrow \quad c = d.$$

Да проверим също, че всяка от функциите g_i е монотонна. Така ще имаме база да сравняваме вектора $(g_1, \ldots, g_k)$ с най-малката неподвижна точка $f_\Gamma = (f_\Gamma^1, \ldots, f_\Gamma^k)$, която беше елемент на $\mathcal{M}_{m_1} \times \cdots \times \mathcal{M}_{m_k}$.

Твърдение 3.17. За всяко $i = 1, \ldots, k$, функцията g_i е монотонна.

Доказателство. Съгласно *Задача* 3.4 е достатъчно да видим, че g_i е монотонна по всеки от аргументите си. Наистина, да фиксираме $1 \leq j \leq m_i$ и нека $c_1, \ldots, c_{j-1}, c_{j+1}, \ldots, c_{m_i}$ са произволни елементи на $\mathbb{N}_\bot$, а $c \in \mathbb{N}$. Имаме

$$(c_1, \ldots, c_{j-1}, \bot, c_{j+1}, \ldots, c_{m_i}) \sqsubset (c_1, \ldots, c_{j-1}, c, c_{j+1}, \ldots, c_{m_i}).$$

Трябва да проверим, че

$$g_i(c_1,\ldots,c_{j-1},\bot,c_{j+1},\ldots,c_{m_i}) \sqsubseteq g_i(c_1,\ldots,c_{j-1},c,c_{j+1},\ldots,c_{m_i}).$$

Наистина, нека $g_i(c_1,\ldots,c_{j-1},\bot,c_{j+1},\ldots,c_{m_i}) = d \in \mathbb{N}$. Тогава съгласно дефиницията на g_i ще имаме

$$R \vdash_N \underbrace{F_i(c_1,\ldots,c_{j-1},\bot,c_{j+1},\ldots,c_{m_i})}_{\mu} \;\to\; d.$$

Прилагаме *Твърдение* 3.16 към терма $\mu^* = F_i(c_1,\ldots,c_{j-1},\mathbf{c},c_{j+1},\ldots,c_{m_i})$. Така получаваме

$$R \vdash_N \underbrace{F_i(c_1,\ldots,c_{j-1},c,c_{j+1},\ldots,c_{m_i})}_{\mu^*} \;\to\; d,$$

което означава, че и $g_i(c_1,\ldots,c_{j-1},c,c_{j+1},\ldots,c_{m_i}) = d$. □

Сега вече сме сигурни, че $(g_1,\ldots,g_k) \in \mathcal{M}_{m_1} \times \cdots \times \mathcal{M}_{m_k}$, точно както и $(f^1_\Gamma,\ldots,f^k_\Gamma)$. При семантиките *по стойност* видяхме, че тези два вектора съвпадат. И тук ще е така, но на нас ще ни е достатъчно да покажем само, че $(f^1_\Gamma,\ldots,f^k_\Gamma) \sqsubseteq (g_1,\ldots,g_k)$. За целта е достатъчно да видим, че $\bar{g} = (g_1,\ldots,g_k)$ е преднеподвижна точка на оператора Γ, с други думи, е решение на неравенството $\Gamma(\bar{f}) \sqsubseteq \bar{f}$. Този факт ще получим като директно следствие от следния по-общ резултат:

Твърдение 3.18. Нека $\bar{g} = (g_1,\ldots,g_k)$ са функциите, определени с равенството (3.32). Тогава за всеки обобщен функционален терм $\mu(F_1,\ldots,F_k)$ и всяко $d \in \mathbb{N}$ е в сила:

$$\mu(\bar{g}) = d \quad\Longrightarrow\quad R \vdash_N \mu \;\to\; d.$$

Доказателство. Доказателството ще е с индукция по построението на обобщения функционален терм μ.

Да фиксираме μ и да предположим, че за всички обобщени функционални термове, построени *преди* него, твърдението е вярно. Разглеждаме различните възможности за μ.

1) Ако μ е константа, то $\mu(\bar{g}) \stackrel{\text{деф}}{=} \mu$. Ние имаме, че $\mu(\bar{g}) = d$, следователно $\mu = d$, откъдето очевидно $R \vdash_N d \;\to\; d$. Да отбележим, че случаят $\mu = \bot$ е невъзможен, защото $d \in \mathbb{N}$ по условие.

2) Случаят $\mu = X_i$ също е невъзможен, защото μ е функционален терм.

3) Нека $\mu = \mu_1\ op\ \mu_2$. Тогава

$$\mu(\bar{g}) \;\stackrel{\text{деф}}{=}\; \mu_1(\bar{g})\ op^*\ \mu_2(\bar{g}) \;=\; d.$$

Нека $\mu_1(\bar{g}) = c_1$ и $\mu_2(\bar{g}) = c_2$. Тогава $c_1\ op^*\ c_2 = d$. Понеже $d \in \mathbb{N}$, а op^* е точна функция, то непременно c_1 и c_2 са числа (не могат да са $\bot$!). Като приложим индукционната хипотеза към μ_1 и μ_2, от горните равенства получаваме

$$R \vdash_N \ \mu_1 \ \to \ c_1, \quad R \vdash_N \ \mu_2 \ \to \ c_2 \quad \text{и} \quad c_1\ op^*\ c_2 \ = \ d.$$

Но това са точно условията над чертата на правилото (1). Прилагаме го и получаваме, че

$$R \vdash_N \underbrace{\mu_1\ op\ \mu_2}_{\mu} \ \to \ d.$$

4) Ако $\mu = \texttt{if}\ \mu_1\ \texttt{then}\ \mu_2\ \texttt{else}\ \mu_3$, разсъжденията са много подобни на горните.

5) Остана случаят $\mu = F_i(\mu_1, \ldots, \mu_{m_i})$. Тук вече ще използваме, че функциите $\bar{g}$ са избрани специално. Имаме по условие, че $\mu(\bar{g}) = d$ и следователно

$$g_i(\mu_1(\bar{g}), \ldots, \mu_{m_i}(\bar{g})) = d.$$

Нека $\mu_1(\bar{g}) = c_1, \ldots \mu_{m_i}(\bar{g}) = c_{m_i}$, където $c_1, \ldots, c_{m_i}$ са елементи на $\mathbb{N}_\bot$.

От това, че $g_i(c_1, \ldots, c_{m_i}) = d \in \mathbb{N}$, съгласно избора на g_i можем да твърдим, че

$$R \vdash_N \ F_i(c_1, \ldots, c_{m_i}) \ \to \ d.$$

Нашата цел е да получим, че $R \vdash_N F_i(\mu_1, \ldots, \mu_{m_i}) \ \to \ d$. Ясно е, че ще трябва да използваме някак индуктивната хипотеза за μ_j.

Наистина, ако $c_j \in \mathbb{N}$, то от $\mu_j(\bar{g}) = c_j$ и по индуктивната хипотеза за μ_j ще имаме $R \vdash_N \ \mu_j \ \to \ c_j$. Разбира се, не може да се каже същото за случаите, когато $c_j = \bot$. Но в тези случаи знаем, че $\bot$ може да се замести с всякакъв обобщен функционален терм ν, както вече съобразихме в *Твърдение* 3.16. Тръгвайки от $R \vdash_N \ F_i(c_1, \ldots, c_{m_i}) \ \to d$, прилагаме това твърдение последователно толкова пъти, колкото са на брой елементите $\bot$ сред аргументите на F_i. При всяко от тези прилагания, ако заместваме елемент $\bot$, който е на j-та позиция т.е. $c_j = \bot$, то този $\bot$ заместваме с терма μ_j. Вече можем да твърдим, че

$$R \vdash_N \ F_i(\nu_1, \ldots, \nu_{m_i}) \ \to \ d,$$

където

$$\nu_j \overset{\text{деф}}{=} \begin{cases} \mu_j, & \text{ако } c_j = \bot \\ c_j, & \text{ако } c_j \in \mathbb{N}. \end{cases}$$

Сега трябва да елиминираме и оставащите константи c_j в терма $F_i(\nu_1, \ldots, \nu_{m_i})$. Ясно е, че c_j участва в този терм само ако е от $\mathbb{N}$.

Но както видяхме по-горе, в този случай е приложима индуктивната хипотеза и съответно $R \vdash_N \ \mu_j \ \to \ c_j$. В този момент се сещаме за Лемата за симулацията, според която от извода на опростяването $R \vdash_N \ F_i(\nu_1, \ldots, \nu_{m_i}) \ \to \ d$ можем да конструираме нов извод, при който всяка от константите $c_j \in \mathbb{N}$ е заменена с μ_j. Но така получаваме точно

$$R \vdash_N \ F_i(\mu_1, \ldots, \mu_{m_i}) \ \to \ d,$$

което и трябваше да докажем.

□

Като непосредствено следствие от току-що доказаното твърдение ще получим, че $\bar{g}$ е решение на неравенството $\Gamma(\bar{f}) \sqsubseteq \bar{f}$, откъдето веднага ще следва и търсеното от нас включване

Твърдение 3.19. $\bar{f}_\Gamma \sqsubseteq \bar{g}$.

Доказателство. Да покажем, че $\Gamma(\bar{g}) \sqsubseteq \bar{g}$ означава да покажем, че за всяко $i = 1, \ldots k$

$$\Gamma_{\tau_i}(\bar{g}) \sqsubseteq g_i,$$

или все едно, $\Gamma_{\tau_i}(\bar{g})(\bar{c}) \sqsubseteq g_i(\bar{c})$ за всяка $\bar{c} \in \mathbb{N}^{m_i}$.

Наистина, да фиксираме $1 \leq i \leq k$. Съгласно *Твърдение* 3.8 е достатъчно да покажем, че за произволни $c_1, \ldots, c_{m_i}$ от $\mathbb{N}_\perp$ и $d \in \mathbb{N}$ е вярно, че

$$\Gamma_{\tau_i}(\bar{g})(\bar{c}) = d \implies g_i(\bar{c}) = d. \tag{3.33}$$

От определението за термален предикат имаме:

$$\Gamma_{\tau_i}(\bar{g})(\bar{c}) \overset{\text{деф}}{=} \tau_i(\bar{c}, \bar{g}) \overset{\text{Лема 3.30}}{=} \tau_i[\bar{X}/\bar{c}](\bar{g}).$$

Наистина, да приемем, че $\Gamma_{\tau_i}(\bar{g})(\bar{c}) = d$. Тогава ще имаме

$$\underbrace{\tau_i[\bar{X}/\bar{c}]}_{\mu}(\bar{g}) \ = \ d.$$

Сега прилагаме горното *Твърдение* 3.18 за $\mu = \tau_i[\bar{X}/\bar{c}]$ и получаваме

$$R \vdash_N \tau_i[\bar{X}/\bar{c}] \ \to \ d.$$

Оттук по правилото за извод (3_N) ще следва, че

$$R \vdash_V F_i(\bar{c}) \ \to \ d,$$

което според дефиницията (3.32) на g_i означава, че $g_i(\bar{c}) = d$. С това проверката на (3.33) е завършена и следователно $\Gamma_{\tau_i}(\bar{g}) \sqsubseteq g_i$. Тъй като i беше произволно, това ни дава общо, че $\Gamma(\bar{g}) \sqsubseteq \bar{g}$, т.е. $\bar{g}$ е преднеподвижна точка на оператора Γ. Но теоремата на Кнастер-Тарски за произволни ОС ни казва че $\bar{f}_\Gamma$ е най-малкото решение на това неравенство, откъдето добиваме желаното $\bar{f}_\Gamma \sqsubseteq \bar{g}$. □

Сега вече имаме всичко, за да докажем важната

Теорема 3.3. За всяка рекурсивна програма R

$$O_N(R) = D_N(R).$$

Доказателство. Посоката $O_N \subseteq D_N(R)$ е точно *Следствие* 3.4. Обратно, нека за някои естествени $\bar{c}$ и d е изпълнено $D_N(R)(\bar{c}) \simeq d$. От дефиницията на $D_N(R)$ имаме, че това се случва, ако $\tau_0(\bar{c}, \bar{f_\Gamma}) = d$, или все едно — $\Gamma_{\tau_0}(\bar{f_\Gamma}, \bar{c}) = d$. Но $\bar{f_\Gamma} \sqsubseteq \bar{g}$, съгласно доказаното току-що *Твърдение* 3.19. Обаче всеки термален оператор е монотонен (*Твърдение* 3.13). Следователно и $\Gamma_{\tau_0}(\bar{g}, \bar{c}) = d$, или все едно, $\tau_0(\bar{c}, \bar{g}) = d$. Последното можем да препишем още като

$$\underbrace{\tau_0[\bar{X}/\bar{c}]}_{\mu}(\bar{g}) = d.$$

Сега вече от *Твърдение* 3.18 ще имаме, че $R \vdash_N \underbrace{\tau_0[\bar{X}/\bar{c}]}_{\mu} \rightarrow d$. Но последното означава точно, че $O_N(R)(\bar{c}) \simeq d$. □

Забележка. Да обърнем внимание, че в доказателството и на тази теорема никъде не използвахме, че типът данни, над които работят програмите от езика REC, са естествените числа. Резултатът за съвпадение на двете семантики е в сила за всеки друг тип данни.

Като непосредствено следствие от горната теорема и *Теорема* 3.1 получаваме, че:

Теорема 3.4. За всяка рекурсивна програма R

$$D_V(R) \subseteq D_N(R).$$

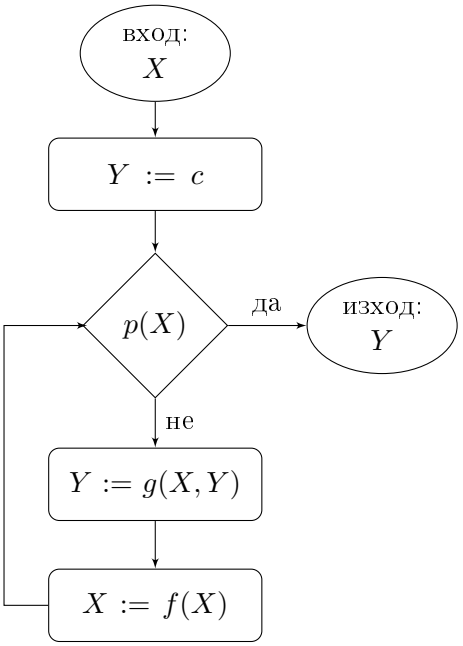
Глава 4

Сравнителна схематология

4.1 Схеми на програми

Преди да дадем точните дефиниции, да си изясним разликата между *схема на програма* и програма.

Да разгледаме блок-схемата S от по-долу. Да се запитаме какво прави тази блок-схема; коя функция пресмята тя?



Пример 4.1.

Да, въпросът какво пресмята S е безсмислен, защото тази картинка все още не е програма; тя е само *схема на програма*. Не знаем върху какви обекти работи, не знаем и какъв е смисълът на символите c, f, g и p, участващи в нея.

Една подходяща аналогия в случая е връзката между *формула* от първи ред и нейната *вярност*, за която можем да говорим след като сме придали смисъл на нелогическите символи, участващи във формулата. Например, за формулата φ

$$\forall X \exists Y p(X, Y)$$

е безсмислено да се питаме дали е вярна. За да е смислен въпросът, трябва да знаем къде варират променливите X и Y и какъв е смисълът на буквата p. Това става чрез фиксирането на *структура* в сигнатурата на φ, която в случая се състои само от двуместния предикатен символ p. Ако вземем структурата $\mathcal{N} = (\mathbb{N},\ >)$, в която p се интерпретира като предиката $>$, то очевидно φ не е вярна в $\mathcal{N}$. Ако сменим универсума с множеството $\mathbb{Z}$ на целите числа и разгледаме структурата $\mathcal{Z} = (\mathbb{Z},\ >)$, в нея вече φ ще е вярна.

Да се върнем отново на схемата S от *Пример* 4.1. *Сигнатурата* на схемата е

$$\Sigma \ = \ (c;\ f,\ g;\ p),$$

където c е константен символ, f и g са функционални символи (на 1 и на 2 аргумента, съответно), а p е едноместен предикатен символ. *Структура* $\mathcal{A}$ в тази сигнатура е редица от вида

$$\mathcal{A} \ = \ (D;\ c^{\mathcal{A}};\ f^{\mathcal{A}}, g^{\mathcal{A}};\ p^{\mathcal{A}}),$$

където D е непразно множество — *универсум* (или *носител*) на $\mathcal{A}$, а останалите елементи задават интерпретацията на символите от Σ: $c^{\mathcal{A}} \in D$, $f^{\mathcal{A}}$ и $g^{\mathcal{A}}$ са функции в D (съответно на 1 и 2 аргумента), а $p^{\mathcal{A}}$ е едноместен предикат в D.

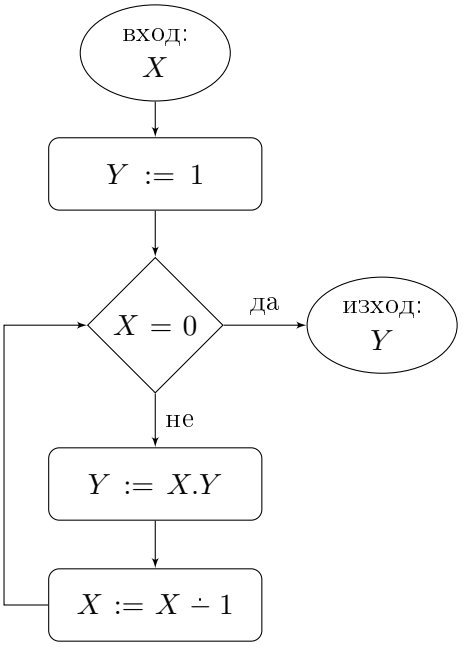
Ето два примера за структури $\mathcal{A}_1$ и $\mathcal{A}_2$ в горната сигнатура $\Sigma \ = \ (c;\ f,\ g;\ p)$. Да дефинираме $\mathcal{A}_1$ по следния начин:

$$\mathcal{A}_1 \ = \ (\mathbb{N};\ 1;\ x \dot{-} 1,\ x.y;\ x = 0?),$$

където функцията $\dot{-}$ (*отсечена разлика*) се определя с равенството:

$$x \dot{-} y = \begin{cases} x - y, & \text{ако } \ x \geq y \\ 0, & \text{иначе.} \end{cases}$$

В тази структура от схемата S получаваме следната *програма* $(S, \mathcal{A}_1)$:



Пример 4.2.

Лесно се вижда, че програмата $(S, \mathcal{A}_1)$ пресмята функцията $x!$. В този случай ще казваме, че $Sem(S, \mathcal{A}_1)$ е функцията $x!$.

Сега да разгледаме структурата $\mathcal{A}_2$ с носител множеството A^* на всички думи над дадена крайна азбука A:

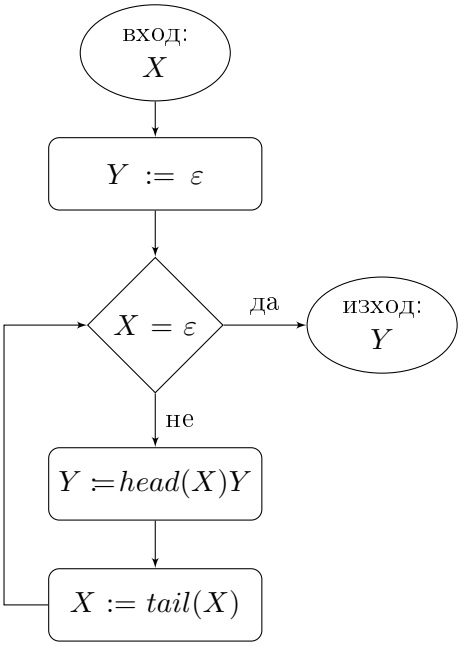
$$\mathcal{A}_2 \ = \ (A^*;\ \varepsilon;\ tail(x),\ head(x)y;\ x = \varepsilon?).$$

Тук ε е празният низ, функциите *head* и *tail* се дефинират по обичайния начин:

$$head(x) \ = \ \begin{cases} a_1, & \text{ако } \ x = a_1 \dots a_n \text{ и } n > 0 \\ \varepsilon, & \text{ако } \ x = \varepsilon \end{cases} \quad \text{и}$$

$$tail(x) \ = \ \begin{cases} a_2 \dots a_n & \text{ако } \ x = a_1 \dots a_n \text{ и } n > 0 \\ \varepsilon, & \text{ако } \ x = \varepsilon, \end{cases}$$

а $head(x)y$ е *конкатенацията* на $head(x)$ и y. В структурата $\mathcal{A}_2$ от схемата S получаваме следната програма $(S, \mathcal{A}_2)$, която вече работи над низове:



Пример 4.3.

Убедете се, че програмата $(S, \mathcal{A}_2)$ пресмята функцията $reverse$, или формално

$$Sem(S, \mathcal{A}_2) = reverse.$$

Една схема може да прави различни неща, в зависимост от това в каква структура я интерпретираме. Схемата S, която разгледахме, е пример за *стандартна схема*, записана на блок-схемен език. Този тип схеми ще дефинираме формално в следващия раздел.

Под *рекурсивни схеми* ще разбираме синтактични обекти, подобни на рекурсивните програми от езика REC, които разглеждахме досега, като разбира се, те вече няма да са (само) над естествените числа. Точното определение също ще дадем по-нататък; сега ще се ограничим само един пример. Рекурсивната схема R, която следва, е в същата сигнатура $\Sigma = (c;\ f,\ g;\ p)$ като по-горе:

$F(X, c)$ `where`
$F(X, Y) =$ `if` $p(X)$ `then` Y `else` $F(f(X), g(X, Y))$

Рекурсивната програма, която схемата R определя в структурата $\mathcal{A}$, ще отбелязваме с $(R, \mathcal{A})$, а $Sem(R, \mathcal{A})$ ще е нейната семантика, т.е. функцията, която програмата $(R, \mathcal{A})$ пресмята. В тази глава под "семантика" ще разбираме денотационната семантика по стойност на рекурсивната програма $(R, \mathcal{A})$. Защо избираме тази семантика, а не денотационната семантика по име на $(R, \mathcal{A})$ ще стане ясно по-нататък.

4.2 Транслируемост

Определение 4.1. Нека S_1 и S_2 са схеми в една и съща сигнатура Σ. Ще казваме, че те са *еквивалентни* (и ще пишем $S_1 \equiv S_2$), ако за всяка структура $\mathcal{A}$ в сигнатурата Σ е вярно, че

$$Sem(S_1, \mathcal{A}) \;=\; Sem(S_2, \mathcal{A}).$$

С други думи, две схеми са еквивалентни, ако във всяка структура пресмятат една и съща функция.

Да си мислим сега, че имаме два класа от схеми — например стандартните и рекурсивните схеми, които споменахме по-горе. Отново ще предполагаме, че те са в една и съща сигнатура Σ. Можем да ги сравняваме по отношение на тяхната изразителна сила, т.е. по отношение на нещата, които могат да се програмират с техните изразни средства във всяка конкретна структура. Ето по-точните определения:

Определение 4.2. Нека $\mathcal{K}_1$ и $\mathcal{K}_2$ са два класа от схеми в една и съща сигнатура Σ. Ще казваме, че класът $\mathcal{K}_1$ *се транслира* в класа $\mathcal{K}_2$ (и ще пишем $\mathcal{K}_1 \leqslant_t \mathcal{K}_2$), ако

$$\forall S_1 \in \mathcal{K}_1 \; \exists S_2 \in \mathcal{K}_2 \; \mathcal{S}_1 \equiv S_2.$$

Забележка. Това е т.нар. *силна* (*синтактична* или още *равномерна*) транслируемост, при която на схемно ниво по дадената схема S_1 се конструира еквивалентната на нея схема S_2. Има и по-слабо понятие за неравномерна или *семантична* транслируемост, което се изказва така: $\mathcal{K}_1$ е *слабо транслируем* в $\mathcal{K}_2$, ако

$$\forall S_1 \in \mathcal{K}_1 \; \underline{\forall \mathcal{A} \; \exists S_2 \in \mathcal{K}_2} \; Sem(S_1, \mathcal{A}) = Sem(S_2, \mathcal{A}).$$

За да видим по-добре разликата между двете понятия, да разпишем условието определението за транслируемост 4.1:

$$\forall S_1 \in \mathcal{K}_1 \; \underline{\exists S_2 \in \mathcal{K}_2 \; \forall \mathcal{A}} \; Sem(S_1, \mathcal{A}) = Sem(S_2, \mathcal{A}).$$

Виждаме, че вътрешните два квантора са разменени, което прави второто понятие за транслируемост по-слабо.

Определение 4.3. Ще казваме, че класовете $\mathcal{K}_1$ и $\mathcal{K}_2$ са *еквивалентни* по отношение на транслируемост (и ще пишем $\mathcal{K}_1 \equiv_t \mathcal{K}_2$), ако

$$\mathcal{K}_1 \leqslant_t \mathcal{K}_2 \quad \text{и} \quad \mathcal{K}_2 \leqslant_t \mathcal{K}_1.$$

Записът $\mathcal{K}_1 \not\leqslant_t \mathcal{K}_2$ ще означава, че $\mathcal{K}_1$ не се транслира в $\mathcal{K}_2$.

И едно последно понятие за *строга транслируемост* $<_t$, което се получава от нестрогата релация $\leqslant_t$ по обичайния начин:

$$\mathcal{K}_1 <_t \mathcal{K}_2 \quad \overset{\text{деф}}{\iff} \quad \mathcal{K}_1 \leqslant_t \mathcal{K}_2 \text{ и } \mathcal{K}_2 \not\leqslant_t \mathcal{K}_1.$$

На картинката по-долу сме използвали следните означения за класове от схеми:

$\mathcal{S}$ — класа на всички стандартни схеми

$\mathcal{R}$ — класа на всички рекурсивни схеми

$\mathcal{S}+nc$ — класа на всички стандартни схеми с n брояча

$\mathcal{S}+ks$ — класа на всички стандартни схеми с k стека

$\mathcal{S}+nc+ks$ — класа на всички стандартни схеми с n брояча и k стека

$\mathcal{LS}$ — класа на всички логически схеми

Ето каква е връзката между тези класове:

4.3 Стандартни схеми

4.3.1 Синтаксис

Нашата цел до края на курса ще бъде да докажем неравенството $\mathcal{S} <_t \mathcal{R}$, което може да се изкаже накратко така: *рекурсията е по-мощна от итерацията.* Това включва две неща:

1) $\mathcal{S} \leqslant_t \mathcal{R}$ — *теорема на Маккарти*, която казва, че всяка стандартна схема се транслира в рекурсивна.

2) $\mathcal{R} \not\leqslant_t \mathcal{S}$ — *пример на Патерсън и Хюит*, който показва, че има рекурсивни схеми, които не могат да се преведат в стандартни.

За доказателствата ще ни трябва строга дефиниция на класа на стандартните схеми. Тези схеми могат да се въведат по най-различни (еквивалентни) начини. Тук ще разгледаме една стандартна дефиниция, която е подходяща за нашите цели.

Да фиксираме произволна сигнатура

$$\Sigma \ = \ (\mathbf{c}_1, \ldots, \mathbf{c}_p;\ \mathbf{f}_1, \ldots, \mathbf{f}_s;\ \mathbf{p}_1, \ldots,\ \mathbf{p}_t),$$

в която $\mathbf{c}_1, \ldots, \mathbf{c}_p$ са константни символи, $\mathbf{f}_1, \ldots, \mathbf{f}_s$ — функционални символи, а $\mathbf{p}_1, \ldots,\ \mathbf{p}_t$ — предикатни символи. Няма да разглеждаме функция за арност (местност), а ще считаме, че функционалните и предикатните символи вървят със своята местност.

Да фиксираме и една редица от *променливи* (или *регистри*) $X_1, X_2, \ldots$.

Следва определението за *оператор* в сигнатурата Σ:

`оператор за присвояване` $::=\ X_i := X_j \ \mid\ X_i := \mathbf{c}_j \ \mid\ X_i := \mathbf{f}_j(X_{i_1}, \ldots, X_{i_k})$

`оператор за преход` $::=$ `goto` $l \ \mid$ `if` $\mathbf{p}_j(X_{i_1}, \ldots, X_{i_n})$ `then goto` l `else goto` l'

`оператор` $::=$ `оператор за присвояване` $\mid$ `оператор за преход` $\mid$ `stop`

Определение 4.4. *Стандартна схема* в сигнатурата Σ е синтактичен обект от следния вид:

$S:$ `input`$(X_1, \ldots, X_n)$; `output`(X_k)
$1:\ O_1$
$\vdots$
$l:\ O_l$
$\vdots$
$q:\ O_q$

където $O_l, 1 \leq l \leq q$, е оператор в сигнатурата Σ.

Числото l ще наричаме *адрес* на оператора O_l. Ще искаме всички адреси, към които сочат операторите за преход, да са в интервала $[1, q]$, а последният оператор S_q да е `stop`.

Блок-схемата S от началото на тази глава (*Пример* 4.1), записана в този синтаксис, ще изглежда така:

$S:$ `input`(X); `output`(Y)
$1:\ Y := \mathbf{c}$
$2:$ `if` $\mathbf{p}(X)$ `then goto` 6 `else goto` 3
$3:\ Y := \mathbf{g}(X, Y)$
$4:\ X := \mathbf{f}(X)$
$5:$ `goto` 2
$6:$ `stop`

Ще казваме, че променливата X_i *участва* в схемата S, ако тя се среща в някой оператор на S. Нека

$$m \ = \ max\{i \mid X_i \text{ участва в } S\}.$$

$(X_1, \ldots, X_m)$ ще наричаме *памет* на S. Ясно е, че броят n на входните променливи е по-малък или равен на m. Разбира се, не е задължително всички X_i, $i \leq m$, да се срещат в схемата. Например в нея могат да участват само X_1, X_3 и X_5 (и тогава m ще е 5).

4.3.2 Семантика

Нека S е произволна стандартна схема в сигнатурата

$$\Sigma \ = \ (\mathbf{c}_1, \ldots, \mathbf{c}_p;\ \mathbf{f}_1, \ldots, \mathbf{f}_s;\ \mathbf{p}_1, \ldots,\ \mathbf{p}_t).$$

За да въведем *семантиката* на S, трябва да фиксираме структура в тази сигнатура. Вече разгледахме няколко примера за структури; сега ще дадем официалната дефиниция.

Определение 4.5. *Структура* в сигнатурата Σ е обект от вида

$$\mathcal{A} \;=\; (D;\; c_1, \ldots, c_p;\; f_1, \ldots, f_s;\; p_1, \ldots, p_t),$$

където:

- D е непразно множество — *универсум/носител* на $\mathcal{A}$;
- $c_1 \in D, \ldots, c_p \in D$ са интерпретациите на константните символи $\mathbf{c}_1, \ldots, \mathbf{c}_p$;
- за всяко $1 \leq i \leq s$, f_i е частична функция в D, с която се интерпретира функционалният символ $\mathbf{f_i}$ (като местността на f_i се определя от местността на $\mathbf{f_i}$);
- за всяко $1 \leq i \leq t$, p_i е предикат в D, с който се интерпретира предикатният символ $\mathbf{p_i}$ (като отново местността на p_i се определя от местността на $\mathbf{p_i}$).

Да вземем произволна стандартна схема S:

$S:$ $\texttt{input}(X_1, \ldots, X_n);$ $\texttt{output}(X_k)$
$1:$ O_1
$\vdots$
$q:$ O_q

Схемата S в структурата $\mathcal{A}$ се превръща в *стандартна програма* $(S, \mathcal{A})$, пресмятаща частичната функция

$$Sem(S, \mathcal{A}):\; D^n \;\multimap\; D.$$

Да фиксираме и произволна структура $\mathcal{A}$ от сигнатурата на S. Ще дефинираме функцията $Sem(S, \mathcal{A})$ — *семантиката* на S в $\mathcal{A}$ в типичен операционен стил — чрез едностъпково преобразование, което ще итерираме дотогава, докато стигнем до оператора `stop` (точно както при машините на Тюринг се итерира δ-функцията на преходите, докато се достигне финално състояние).

За целта дефинираме *конфигурация* (или моментна снимка), която е наредена двойка от вида

$$(l,\; (x_1, \ldots, x_m)),$$

където l е адресът на текущия оператор, който се изпълнява в дадения момент, а $(x_1, \ldots, x_m) \in D^m$ е *текущото състояние на паметта* в този момент (като x_i е съдържанието на i-тия регистър X_i).

Начална конфигурация за входа $(x_1, \ldots, x_n)$ е конфигурацията

$$(1,\; (x_1, \ldots, x_n, \underbrace{x_n, \ldots, x_n}_{m-n \text{ пъти}})).$$

Да отбележим, че регистрите, които не са входни, считаме, че имат стойност x_n при стартиране на програмата. Със същия успех бихме могли да приемем, че тази стойност е x_1 (обаче няма как да искаме да е 0, защото тя трябва да бъде елемент на D).

Заключителна/финална конфигурация е конфигурация от вида

$$(q,\ (y_1, \ldots, y_m)).$$

По-нататък ще пишем $(l, x_1, \ldots, x_m)$ вместо $(l,\ (x_1, \ldots, x_m))$.

Да означим с $Q = \{1, \ldots, q\}$ съвкупността от всички адреси на оператори на S. Ще дефинираме едностъпково преобразование $Step$ (или *функция-стъпка*), което ще е изображение от вида:

$$Step : Q \times D^m \ \multimap \ Q \times D^m$$

Стойността на $Step$ върху фиксирана конфигурация $(l, x_1, \ldots, x_m)$ дефинираме с разглеждане на различните случаи за вида на оператора O_l както следва:

$$Step(l, x_1, \ldots, x_m) \simeq \begin{cases} (l{+}1, x_1, ...x_{i-1}, x_j, x_{i+1}..., x_m), & \text{ако } O_l \text{ е } X_i := X_j \\ (l{+}1, x_1, ...x_{i-1}, c_j, x_{i+1}..., x_m), & \text{ако } O_l \text{ е } X_i := \mathbf{c}_j \\ (l{+}1, x_1, ... \underbrace{f_j(x_{i_1}, .., x_{i_k})}_{(i)} ..., x_m), & \text{ако } O_l \text{ е } X_i := \mathbf{f}_j(X_{i_1}, \ldots, X_{i_k}) \\ (l', x_1, \ldots, x_m), & \text{ако } O_l \text{ е } \texttt{goto } l' \\ (l', x_1, \ldots, x_m), & \text{ако } O_l \text{ е } \texttt{if } \mathbf{p}_j(X_{i_1}, ..., X_{i_n}) \texttt{ then goto } l' \\ & \texttt{else goto } l'' \quad \& \quad p_j(x_{i_1}, ..., x_{i_n}) = \mathbf{t} \\ (l'', x_1, \ldots, x_m), & \text{ако } O_l \text{ е } \texttt{if } \mathbf{p}_j(X_{i_1}, ..., X_{i_n}) \texttt{ then goto } l' \\ & \texttt{else goto } l'' \quad \& \quad p_j(x_{i_1}, ..., x_{i_n}) = \mathbf{f} \\ (l, x_1, \ldots, x_m), & \text{ако } O_l \text{ е } \texttt{stop}. \end{cases}$$

Да напомним, че ако $f : M \ \multimap \ M$ е частична функция в M, то

$$f^n \stackrel{\text{деф}}{=} \underbrace{f \circ \cdots \circ f}_{n \text{ пъти}}.$$

Определение 4.6. *Семантиката* на програмата $(S, \mathcal{A})$ е функцията

$$Sem(S, \mathcal{A}) : D^n \ \multimap \ D,$$

която се дефинира с еквивалентността

$$Sem(S, \mathcal{A})(x_1, \ldots, x_n) \simeq y \stackrel{\text{деф}}{\iff} \exists t \exists y_1 \ldots \exists y_m \ Step^t(1, x_1, ..., x_n, \underbrace{x_n, ..., x_n}_{m-n \text{ пъти}})$$
$$\simeq (q, y_1, \ldots, y_m) \ \&\ y = y_k. \qquad (4.1)$$

Понеже функцията $Sem(S, \mathcal{A})$ е дефинирана с условие, включващо квантори за съществуване, трябва да проверим, че тя е *еднозначна*.

Твърдение 4.1. (Коректност на дефиницията.)

$$Step^t(l, \bar{x}) \simeq (q, \bar{y}) \;\&\; Step^{t'}(l, \bar{x}) \simeq (q, \bar{y}') \quad \Longrightarrow \quad \bar{y} = \bar{y}'.$$

Доказателство. Ще използваме, че

$$f^{n+k}(x) \overset{\text{деф}}{\simeq} \underbrace{f \circ \cdots \circ f}_{n+k \text{ пъти}}(x) \;\simeq\; \underbrace{f \circ \cdots \circ f}_{n \text{ пъти}} \circ \underbrace{f \circ \cdots \circ f}_{k \text{ пъти}}(x) \;\simeq\; f^n(f^k(x)).$$

Без ограничение на общността можем да предполагаме, че $t' \geq t$. Тогава

$$Step^{t'}(l, \bar{x}) \;\simeq\; Step^{t'-t}(Step^t(l, \bar{x})) \;\simeq\; Step^{t'-t}(q, \bar{y}) \overset{\text{деф}}{\simeq} (q, \bar{y}),$$

и следователно $(q, \bar{y}) = (q', \bar{y}')$. □

4.4 Рекурсивни схеми

Синтаксисът на рекурсивните схеми идейно следва дефиницията на рекурсивните програми от езика REC. Разликата е в това, че тези програми бяха дефинирани в естествените числа $\mathbb{N}$ с базисни операции, които означавахме най-общо с op, докато схемите се дефинират за произволна сигнатура Σ. Да я фиксираме отново:

$$\Sigma \;=\; (\mathbf{c}_1, \ldots, \mathbf{c}_p;\; \mathbf{f}_1, \ldots, \mathbf{f}_s;\; \mathbf{p}_1, \ldots,\; \mathbf{p}_t).$$

Терм τ в сигнатурата Σ дефинираме по подобие на старата дефиниция за терм

$$\tau ::= \mathbf{c}_i \;\mid\; X_i \;\mid\; \mathbf{f}_i(\tau, \ldots, \tau) \;\mid\; \texttt{if } \mathbf{p}_i(\tau, \ldots, \tau) \texttt{ then } \tau \texttt{ else } \tau \;\mid\; F_i(\tau, \ldots, \tau)$$

Да отбележим, че в условията $\mathbf{p}_i(\tau, \ldots, \tau)$ на термове `if then else` стоят изрази, които (вече) имат булеви стойности.

И тук ще пишем $\tau(X_1, \ldots, X_n, F_1, \ldots, F_k)$, за да означим, че променливите на τ са измежду $X_1, \ldots, X_n, F_1, \ldots, F_k$.

Определение 4.7. *Рекурсивна схема* R в сигнатурата Σ ще наричаме синтактичен обект от вида:

$$R \begin{cases} \tau_0(X_1, \ldots, X_n, F_1, \ldots, F_k) \qquad \texttt{where} \\ F_1(X_1, \ldots, X_{m_1}) = \tau_1(X_1, \ldots, X_{m_1}, F_1, \ldots, F_k) \\ \vdots \\ F_k(X_1, \ldots, X_{m_k}) = \tau_k(X_1, \ldots, X_{m_k}, F_1, \ldots, F_k) \end{cases}$$

в който термовете $\tau_0, \ldots, \tau_k$ са термове в сигнатурата Σ.

Сега да вземем произволна структура в тази сигнатура:

$$\mathcal{A} \ = \ (D;\ a_1, \ldots, a_p;\ f_1, \ldots, f_s;\ p_1, \ldots, p_t).$$

В структурата $\mathcal{A}$ рекурсивната схема R се превръща в *рекурсивна програма* $(R, \mathcal{A})$. Приемаме, че семантиката на тази програма е денотационната (или еквивалентно — операционната) семантика *по стойност* на програмата $(R, \mathcal{A})$. За целите, които сме си поставили тук — да покажем, че всяка стандартна схема се транслира в рекурсивна, ще се окаже, че не е от значение дали вземаме семантиката по стойност или по име. Това ще е така, защото всяка стандартна схема ще се превежда в рекурсивна схема от много прост вид — в която няма вложени участия на функционални променливи.

4.5 Транслируемост на стандартните схеми в рекурсивни схеми

4.5.1 Опашкови функции

Да вземем отново произволна стандартна схема S в сигнатурата $\Sigma \ = \ (\mathbf{c}_1, \ldots, \mathbf{c}_p;\ \mathbf{f}_1, \ldots, \mathbf{f}_s;\ \mathbf{p}_1, \ldots,\ \mathbf{p}_t)$.

$S:$ $\mathtt{input}(X_1, \ldots, X_n);\ \mathtt{output}(X_k)$
$1:\ O_1$
$\vdots$
$l:\ O_l$
$\vdots$
$q:\ O_q,$

Отново ще предполагаме, че паметта на S е $(X_1, \ldots, X_m)$. Да фиксираме и произволна структура $\mathcal{A} \ = \ (D;\ a_1, \ldots, a_p;\ f_1, \ldots, f_s;\ p_1, \ldots, p_t)$ в сигнатурата Σ.

За всяко $l = 1, \ldots, q$ ще въведем *опашкови функции* (*tail functions*) g_l, които са изображения от вида:

$$g_l\colon\ D^m \ \multimap \ D.$$

Дефиницията на опашковите функции е в духа на *Определение* (4.1) на семантиката $Sem(S, \mathcal{A})$ — посредством функцията $Step$:

Определение 4.8. За всяко $l = 1, \ldots, q$ полагаме

$$g_l(x_1, ..., x_m) \simeq y \ \overset{\text{деф}}{\Longleftrightarrow} \ \exists t \exists y_1 ... \exists y_m \ Step^t(l, x_1, ..., x_m) \simeq (q, y_1, ..., y_m) \ \& \ y = y_k. \tag{4.2}$$

Смисълът на тези функции, в общи линии, се вижда от определението им — те са това, което ще пресметне програмата, когато я стартираме не от първия, а от l-тия оператор. Какво е тяхното предназначение ще стане ясно по-нататък.

Да обърнем внимание, че опашковите функции действат върху цялата памет $(x_1, \ldots, x_m)$, т.е. са функции на m аргумента, за разлика от $Sem(S, \mathcal{A})$, която е функция само на входните променливи $(x_1, \ldots, x_n) \in D^n$.

Непосредствено от определението получаваме и следната връзка между първата опашкова функция g_1 и $Sem(S, \mathcal{A})$:

$$Sem(S, \mathcal{A})(x_1, \ldots, x_n) \simeq g_1(x_1, \ldots, x_n, \underbrace{x_n, ..., x_n}_{m-n \text{ пъти}}). \tag{4.3}$$

Понеже и в дефиницията (4.2) участват квантори за съществуване, трябва отново да докажем коректност на на тази дефиниция, но тъй като тя се показва точно както при доказателството на *Твърдение* 4.1, ще я пропуснем.

За упражнение да намерим опашковите функции на стандартната програма $(S, \mathcal{A}_1)$, определена с блок-схемата 4.2, записана в езика на стандартните схеми.

Задача 4.1. Намерете явния вид на всяка от опашковите функции $g_1, \ldots, g_6$ на следната стандартна програма S:

`input`(X); `output`(Y)
1 : $Y := \mathbf{1}$
2 : `if` $X = 0$ `then goto` 6 `else goto` 3
3 : $Y := X.Y$
4 : $X := X \dot{-} 1$
5 : `goto` 2
6 : `stop`

Решение. Лесно се съобразява, че първите две опашкови функции са

$$g_1(x, y) = x! \quad \text{и} \quad g_2(x, y) = x!.y.$$

За g_3 се вижда, че връща 0 при $x = 0$, а при $x > 0$ е същата като $g_2(x, y)$, или:

$$g_3(x, y) = \begin{cases} 0, & \text{ако } x = 0 \\ x!.y, & \text{ако } x > 0. \end{cases}$$

Тъй като четвъртият оператор е присвояването $X := X \dot{-} 1$, то би трябвало

$$g_4(x, y) \simeq g_5(x \dot{-} 1, y).$$

Но коя е функцията g_5? Ами тя е същата като g_2, защото операторът, от който се стартира пресмятането на g_5, е `goto` 2. Сега вече можем да намерим и явния вид на g_4:

$$g_4(x,y) \simeq g_5(x \dot{-} 1, y) \simeq g_2(x \dot{-} 1, y) = (x \dot{-} 1)!.y.$$

Последната функция g_6 очевидно винаги връща y, т.е. $g_6(x,y) = y$. □

Всъщност с разсъжденията по-горе не само намерихме опашковите функции $g_1, \ldots, g_6$, но и забелязахме нещо по-общо, което ще е ключово за теоремата която предстои да формулираме. Това е наблюдението, че в зависимост от вида на оператора O_l, опашковата функция g_l удовлетворява определено равенство. Тези равенства, събрани заедно, ни дават следната *система*, която се удовлетворява от функциите $g_1, \ldots, g_6$:

$$\left|\begin{array}{lcl} g_1(x,y) & = & g_2(x,1) \\ g_2(x,y) & = & \texttt{if } x == 0 \texttt{ then } g_6(x,y) \texttt{ else } g_3(x,y) \\ g_3(x,y) & = & g_4(x, x.y) \\ g_4(x,y) & = & g_5(x \dot{-} 1, y) \\ g_5(x,y) & = & g_2(x,y) \\ g_6(x,y) & = & y \end{array}\right.$$

Ако искаме да сме една крачка по-близо до рекурсивната програма, еквивалентна на стандартната програма, от която тръгнахме, можем да кажем, че векторът от опашковите функции $(g_1, \ldots, g_6)$ е решение на системата от уравнения, съответстваща на ето тези дефиниции на функционалните променливи $F_1, \ldots, F_6$:

$$\begin{array}{lcl} F_1(X,Y) & = & F_2(X,1) \\ F_2(X,Y) & = & \texttt{if } X == 0 \texttt{ then } F_6(X,Y) \texttt{ else } F_3(X,Y) \\ F_3(X,Y) & = & F_4(X, X.Y) \\ F_4(X,Y) & = & F_5(X \dot{-} 1, Y) \\ F_5(X,Y) & = & F_2(X,Y) \\ F_6(X,Y) & = & Y \end{array}$$

Тези декларации формират тялото на рекурсивната програма, която искаме да е еквивалентна на стандартната програма от *Задача* 4.1. А коя е главата ѝ? Спомняйки си за равенството (4.3), което дава връзката между първата опашкова функция g_1 и семантиката на една стандартна програма, стигаме до извода, че главата на тази програма трябва да е $F_1(X,X)$. Така получаваме следната рекурсивна програма R:

$$
\begin{array}{lll}
R: & F_1(X,X) & \texttt{where} \\
& F_1(X,Y) & = F_2(X,1) \\
& F_2(X,Y) & = \texttt{if } X == 0 \texttt{ then } F_6(X,Y) \texttt{ else } F_3(X,Y) \\
& F_3(X,Y) & = F_4(X,X.Y) \\
& F_4(X,Y) & = F_5(X \dot{-} 1,Y) \\
& F_5(X,Y) & = F_2(X,Y) \\
& F_6(X,Y) & = Y
\end{array}
$$

Разбира се, тази програма може да се оптимизира, като много от функционалните променливи отпаднат. Всъщност остава само променливата F_2 и новата програма, която получаваме, е следната:

$$
\begin{array}{lll}
R^*: & F_2(X,1) & \texttt{where} \\
& F_2(X,Y) & = \texttt{if } X == 0 \texttt{ then } Y \texttt{ else } F_2(X-1,X.Y)
\end{array}
$$

Да отбележим, че следвайки горните разсъждения, можем да отидем и още по-далече, като "повдигнем" цялата конструкция на синтактично ниво. Алгоритъмът, който преобразува всяка стандартна *схема* в еквивалентна на нея рекурсивна *схема*, ще опишем подробно при доказателството на теоремата на Маккарти. Завършвайки, тук само ще отбележим, че стандартната схема от *Пример* 4.1, с която започнахме

S : $\texttt{input}(X)$; $\texttt{output}(Y)$

1 : $Y := \mathbf{c}$

2 : $\texttt{if } \mathbf{p}(X) \texttt{ then goto } 6 \texttt{ else goto } 3$

3 : $Y := \mathbf{g}(X,Y)$

4 : $X := \mathbf{f}(X)$

5 : $\texttt{goto } 2$

6 : $\texttt{stop}$

този алгоритъм ще преработи в следната рекурсивна схема R:

$$
\begin{array}{lll}
R: & F_1(X,X) & \texttt{where} \\
& F_1(X,Y) & = F_2(X,c) \\
& F_2(X,Y) & = \texttt{if } \mathbf{p}(X) \texttt{ then } F_6(X,Y) \texttt{ else } F_3(X,Y) \\
& F_3(X,Y) & = F_4(X,\mathbf{g}(X,Y)) \\
& F_4(X,Y) & = F_5(\mathbf{f}(X),Y) \\
& F_5(X,Y) & = F_2(X,Y) \\
& F_6(X,Y) & = Y
\end{array}
$$

4.5.2 Теорема на Маккарти

Теорема 4.1. (Теорема на Маккарти.) Нека Σ е произволна сигнатура. За всяка стандартна схема S в сигнатурата Σ съществува еквивалентна на нея рекурсивна схема R.

Доказателство. Ще опишем алгоритъма, който на произволна стандартна схема съпоставя еквивалентна на нея рекурсивна схема. По същество ще следваме стъпките при преобразуването на стандартната схема от примера в предишния раздел, като този път приложим и доказателства към разсъжденията си.

Стандартната схема S, от която тръгваме, има следния общ вид:

$S:$ $\texttt{input}(X_1, \ldots, X_n)$; $\texttt{output}(X_k)$
$1:$ O_1
$\vdots$
$l:$ O_l
$\vdots$
$q:$ O_q

Нека паметта на S е $(X_1, \ldots, X_m)$. Рекурсивната схема R, която ще конструираме, ще има q на брой функционални променливи $F_1, \ldots, F_q$, като всички са на един и същ брой аргументи — точно m. Дефинициите на $F_1, \ldots, F_q$ ще отразяват връзките между опашковите функции, които наблюдавахме, когато решавахме *Задача* 4.1. Идеята е във всяка структура $\mathcal{A}$ функцията, която се пресмята от F_l да е точно l-тата опашкова функция.

Да си спомним за връзката (4.3) между $Sem(S, \mathcal{A})$ и първата опашкова функция:

$$Sem(S, \mathcal{A})(x_1, \ldots, x_n) \simeq g_1(x_1, \ldots, x_n, \underbrace{x_n, ..., x_n}_{m-n \text{ пъти}}).$$

Като имаме предвид това и факта, че първата опашкова функция съответства на F_1, полагаме главата на бъдещата схема R да е следният терм τ_0:

$$\tau_0(X_1, \ldots, X_n, F_1) = F_1(X_1, \ldots, X_n, \underbrace{X_n, ..., X_n}_{m-n \text{ пъти}}). \tag{4.4}$$

Тялото на рекурсивната схема R е декларация от вида:

$$\begin{aligned} &F_1(X_1, \ldots, X_m) = \tau_1(X_1, \ldots, X_m, F_1, \ldots, F_q) \\ &\vdots \\ &F_q(X_1, \ldots, X_m) = \tau_q(X_1, \ldots, X_m, F_1, \ldots, F_q) \end{aligned} \tag{4.5}$$

Всеки от горните термове τ_l определяме в зависимост от вида на l-тия оператор на схемата S. Разглеждаме поотделно случаите за O_l.

– ако O_l е $X_i := X_j$, то

$$\tau_l(X_1, \ldots, X_m, F_1, \ldots F_q) \stackrel{\text{деф}}{=} F_{l+1}(X_1, \ldots, X_{i-1}, X_j, X_{i+1}, \ldots, X_m);$$

– ако O_l е $X_i := \mathbf{c}_j$, то

$$\tau_l(X_1, \ldots, X_m, F_1, \ldots F_q) \stackrel{\text{деф}}{=} F_{l+1}(X_1, \ldots, X_{i-1}, \mathbf{c}_j, X_{i+1}, \ldots, X_m);$$

– ако O_l е $X_i := \mathbf{f}_j(X_{i_1}, \ldots, X_{i_k})$, то

$$\tau_l(X_1, \ldots, X_m, F_1, \ldots F_q) \stackrel{\text{деф}}{=} F_{l+1}(X_1, \ldots, X_{i-1}, \mathbf{f}_j(X_{i_1}, \ldots, X_{i_k}), X_{i+1}, \ldots, X_m);$$

– ако O_l е `goto` l', то

$$\tau_l(X_1, \ldots, X_m, F_1, \ldots F_q) \stackrel{\text{деф}}{=} F_{l'}(X_1, \ldots, X_m);$$

– ако O_l е `if` $\mathbf{p}_j(X_{i_1}, \ldots, X_{i_n})$ `then goto` l' `else goto` l'', то

$$\tau_l(X_1, \ldots, X_m, F_1, \ldots F_q) \stackrel{\text{деф}}{=} \texttt{if}\ \mathbf{p}_j(X_{i_1}, ..., X_{i_n})\ \texttt{then}\ F_{l'}(X_1, ..., X_m)\ \ \texttt{else}\ F_{l''}(X_1, ..., X_m);$$

– ако O_l е `stop`, то

$$\tau_l(X_1, \ldots, X_m, F_1, \ldots F_q) \stackrel{\text{деф}}{=} X_k.$$

Идеята зад всяко от тези полагания е съвсем прозрачна. Например ако O_l е оператор за присвояване, отиваме на *следващия* оператор, при това с памет, изменена според това, което "казва" O_l. Ако O_l е оператор за преход или `stop`, смисълът на термовете τ_l е още по-ясен. Ясно е също, че новата схема R ще бъде в същата сигнатура като тази на S.

За да покажем, че получената рекурсивна схема R е еквивалентна на S, фиксираме произволна структура в сигнатурата Σ:

$$\mathcal{A} \ = \ (D;\ a_1, \ldots, a_p;\ f_1, \ldots, f_s;\ p_1, \ldots, p_t).$$

Трябва да покажем, че

$$Sem(S, \mathcal{A}) \ = \ Sem(R, \mathcal{A}).$$

Да означим с $(R, \mathcal{A})$ рекурсивната *програма*, която се получава от схемата R в структурата $\mathcal{A}$. Нека още $g_1, \ldots, g_q$ са опашковите функции на

стандартната програма $(S, \mathcal{A})$. Ще покажем, че те са най-малко решение на системата, определена от тялото (4.5) на програмата $(R, \mathcal{A})$.

Но преди това да се убедим, че от този факт ще следва теоремата. По определение

$$Sem(R, \mathcal{A}) \;=\; D_V(R, \mathcal{A})$$

Като вземем предвид главата (4.4) на схемата R, за денотационната семантика $D_V(R, \mathcal{A})$ ще имаме:

$$D_V(R,\mathcal{A})(x_1,\ldots,x_n) \overset{\text{деф}}{\simeq} \tau_0(x_1,\ldots,x_n,g_1) \overset{(4.4)}{=} g_1(x_1,\ldots,x_n,\underbrace{x_n,\ldots,x_n}_{m-n \text{ пъти}}).$$

От друга страна, вече отбелязахме, че $Sem(S, \mathcal{A})$ е свързана с първата опашкова функция g_1 по следния начин:

$$Sem(S,\mathcal{A})(x_1,\ldots,x_n) \simeq g_1(x_1,\ldots,x_n,\underbrace{x_n,\ldots,x_n}_{m-n \text{ пъти}}).$$

От горните две равенства получаваме, че за всички $(x_1, \ldots, x_n) \in D^n$:

$$Sem(S,\mathcal{A})(x_1,\ldots,x_n) \simeq D_V(R,\mathcal{A})(x_1,\ldots,x_n),$$

и следователно $Sem(S, \mathcal{A}) = D_V(R, \mathcal{A}) \overset{\text{деф}}{=} Sem(R, \mathcal{A})$.

Сега се насочваме към доказателството на факта, че опашковите функции $g_1, \ldots, g_q$ са най-малко решение на системата, съответна на тялото на програмата $(R, \mathcal{A})$. Да означим с $f_1, \ldots, f_q$ неизвестните функции в тази система. Тогава тя ще изглежда така:

$$\left|\begin{array}{l} f_1 = \Gamma_{\tau_1}(f_1,\ldots,f_q) \\ \vdots \\ f_q = \Gamma_{\tau_q}(f_1,\ldots,f_q), \end{array}\right. \tag{4.6}$$

където всеки от термалните оператори $\Gamma_{\tau_1}, \ldots, \Gamma_{\tau_q}$ се определя с равенството

$$\Gamma_{\tau_i}(\bar{f})(\bar{x}) \overset{\text{деф}}{\simeq} \tau_i(\bar{x}, \bar{f}).$$

Да проверим, че опашковите функции $g_1, \ldots, g_q$ са решение на горната система означава да проверим, че за всяко $l \in \{1, \ldots, q\}$ и всяко $(x_1, \ldots, x_m) \in D^m$ е изпълнено:

$$g_l(x_1,\ldots,x_m) \simeq \tau_l(x_1,\ldots,x_m,g_1,\ldots,g_q).$$

Всички те са очевидни от съответните дефиниции на опашкова функция и на терма τ_l.

По-интересно е да видим защо опашковите функции $g_1, \ldots, g_q$ са *най-малкото решение* на системата (4.6). За целта нека $h_1, \ldots, h_q$ са друго решение на (4.6). Това означава, че за всяко $(x_1, \ldots, x_m) \in D^m$ са за тях е изпълнено:

$$\left|\begin{array}{l} h_1(x_1, \ldots, x_m) \simeq \tau_1(x_1, \ldots, x_m, h_1, \ldots, h_q) \\ \vdots \\ h_q(x_1, \ldots, x_m) \simeq \tau_q(x_1, \ldots, x_m, h_1, \ldots, h_q) \end{array}\right. \tag{4.7}$$

Трябва да покажем, че

$$\text{за всяко } l \in \{1, \ldots, q\}: \quad g_l \subseteq h_l,$$

което по дефиниция означава

$$\forall l \in \{1, \ldots, q\}\ \forall x_1 \ldots \forall x_m \forall y\ (g_l(x_1, \ldots, x_m) \simeq y \implies h_l(x_1, \ldots, x_m) \simeq y).$$

Да препишем още веднъж това условие, като минем през дефиницията (4.2) на опашкова функция:

$$\forall l \forall \bar{x} \forall y\ (\exists t \exists \bar{y}\ Step^t(l, \bar{x}) \simeq (q, \bar{y})\ \&\ y = y_k \implies h_l(x_1, \ldots, x_m) \simeq y). \tag{4.8}$$

Тук си спомняме за един логически закон, който ви е известен от ЛП ☺:

$$\forall \bar{x}(\exists \bar{y} \varphi(\bar{x}, \bar{y}) \implies \psi(\bar{x}, \bar{y})) \iff \forall \bar{x} \forall \bar{y}(\varphi(\bar{x}, \bar{y}) \implies \psi(\bar{x}, \bar{y})).$$

Прилагаме го към формулата (4.8) и получаваме

$$\forall l \forall \bar{x} \forall y \forall t \forall \bar{y}\ (Step^t(l, \bar{x}) \simeq (q, \bar{y})\ \&\ y = y_k \implies h_l(\bar{x}) \simeq y).$$

Разбира се, оттук можем да изключим променливата y_k. Ще получим

$$\forall l \forall \bar{x} \forall t \forall \bar{y}\ (Step^t(l, \bar{x}) \simeq (q, \bar{y}) \implies h_l(\bar{x}) \simeq y_k). \tag{4.9}$$

Понеже всички квантори в (4.9) са еднотипни, можем да ги разместваме на воля. Да ги препишем така:

$$\forall t \underbrace{\forall l \forall \bar{x} \forall \bar{y}\ (Step^t(l, \bar{x}) \simeq (q, \bar{y}) \implies h_l(x_1, \ldots, x_m) \simeq y_k)}_{P(t)} \tag{4.10}$$

Да означим с P следното свойство над $\mathbb{N}$:

$$P(t) \overset{\text{деф}}{\iff} \forall l \forall \bar{x} \forall \bar{y}\ (Step^t(l, \bar{x}) \simeq (q, \bar{y}) \implies h_l(\bar{x}) \simeq y_k).$$

Ще докажем, че $\forall t P(t)$ като използваме обикновена индукция относно $t \in \mathbb{N}$, или другояче казано — индукция по броя стъпки, за които спира програмата $Sem(S, \mathcal{A})$, извикана от l-тия си оператор O_l върху вход $(x_1, \ldots, x_m)$.

База $t = 0$. Нека за произволни $l \in \{1, \ldots, q\},\ \bar{x} \in D^m$ и $\bar{y} \in D^m$

$$Step^0(l, \bar{x}) \simeq (q, \bar{y}).$$

Но $Step^0(l, \bar{x}) \stackrel{\text{деф}}{=} (l, \bar{x})$, следователно $(l, \bar{x}) = (q, \bar{y})$, което означава, че $l = q$ и $\bar{x} = \bar{y}$. Значи O_l е операторът `stop`. Но тогава по дефиниция $\tau_q = X_k$, и q-тото уравнение на системата (4.6) ще е

$$F_q(X_1, \ldots, X_m) \ = \ X_k.$$

Това означава, че за h_q ще е изпълнено:

$$h_q(x_1, \ldots, x_m) = x_k.$$

Но по-горе видяхме, че всъщност $\bar{x} = \bar{y}$, в частност, $x_k = y_k$, откъдето получаваме търсеното $h_l(\bar{x}) = y_k$.

Нека сега за произволно t е в сила $P(t)$, т.е. вярно е, че

$$\forall l \forall \bar{x} \forall \bar{y}\ (Step^t(l, \bar{x}) \simeq (q, \bar{y}) \implies h_l(\bar{x}) \simeq y_k).$$

Искаме да покажем и $P(t{+}1)$, което разписано означава, че за фиксирани $l \in \{1, \ldots, q\},\ \bar{x} \in D^m$ и $\bar{y} \in D^m$ трябва да покажем импликацията

$$Step^{t+1}(l, \bar{x}) \simeq (q, \bar{y}) \implies h_l(\bar{x}) \simeq y_k.$$

Наистина, да приемем, че предпоставката $Step^{t+1}(l, \bar{x}) \simeq (q, \bar{y})$ е вярна. Разглеждаме различните възможности за оператора O_l.

Ако той е оператор за присвояване, ще се спрем само на един от трите случая, защото всички те се разглеждат аналогично. Нека например O_l е $X_i := X_j$. Тогава от дефиницията на функцията $Step$ ще имаме, че

$$Step(l, \bar{x}) = (l + 1, x_1, ..., x_{i-1}, x_j, x_{i+1}, ..., x_m),$$

откъдето

$$Step^{t+1}(l, \bar{x}) \stackrel{\text{деф}}{\simeq} Step^t(Step(l, \bar{x})) \simeq Step^t(l{+}1, x_1, ..., x_{i-1}, x_j, x_{i+1}, ..., x_m) \simeq (q, \bar{y}).$$

Получихме, че

$$Step^t(l + 1, x_1, ..., x_{i-1}, x_j, x_{i+1}, ..., x_m) \simeq (q, \bar{y}),$$

което съгласно индукционното предположение $P(t)$, означава, че

$$h_{l+1}(x_1, ..., x_{i-1}, x_j, x_{i+1}, ..., x_m) \simeq y_k. \tag{4.11}$$

Да си спомним, че функциите $h_1, \ldots, h_q$ удовлетворяват равенствата (4.7). В случая ни трябва l-тото от тях:

$$h_l(x_1, \ldots, x_m) \simeq \tau_l(x_1, \ldots, x_m, h_1, \ldots, h_q). \tag{4.12}$$

Но колко е τ_l , когато операторът O_l е $X_i := X_j$? Поглеждаме към съответните дефиниции по-горе и виждаме, че в този случай то е

$$\tau_l(X_1, \ldots, X_m, F_1, \ldots F_q) \stackrel{\text{деф}}{=} F_{l+1}(X_1, \ldots, X_{i-1}, X_j, X_{i+1}, \ldots, X_m).$$

Следователно равенството (4.12) можем да препишем така:

$$h_l(x_1, \ldots, x_m) \simeq h_{l+1}(x_1, ..., x_{i-1}, x_j, x_{i+1}, ..., x_m).$$

Но от (4.11) имаме, че $h_{l+1}(x_1, ..., x_{i-1}, x_j, x_{i+1}, ..., x_m) \simeq y_k$, откъдето получаваме търсеното

$$h_l(x_1, \ldots, x_m) \simeq y_k.$$

Да разгледаме и случая, в който O_l е оператор за преход. Нека той е `goto` l'. Тогава по дефиниция

$$Step(l, \bar{x}) = (l', \bar{x}),$$

откъдето

$$Step^{t+1}(l, \bar{x}) \stackrel{\text{деф}}{\simeq} Step^t(Step(l, \bar{x})) \simeq Step^t(l', \bar{x}) \simeq (q, \bar{y}).$$

Получихме, че

$$Step^t(l', \bar{x}) \simeq (q, \bar{y}),$$

което съгласно индукционното предположение $P(t)$, означава, че

$$h_{l'}(\bar{x}) \simeq y_k. \tag{4.13}$$

Отново от това, че $h_1, \ldots, h_q$ удовлетворяват равенствата (4.7) ще имаме

$$h_l(x_1, \ldots, x_m) \simeq \tau_l(x_1, \ldots, x_m, h_1, \ldots, h_q). \tag{4.14}$$

Остана да видим как сме дефинирали τ_l в този случай:

$$\tau_l(X_1, \ldots, X_m, F_1, \ldots F_q) \stackrel{\text{деф}}{=} F_{l'}(X_1, \ldots, X_m).$$

В такъв случай

$$\tau_l(x_1, \ldots, x_m, h_1, \ldots, h_q) \simeq h_{l'}(x_1, \ldots, x_m).$$

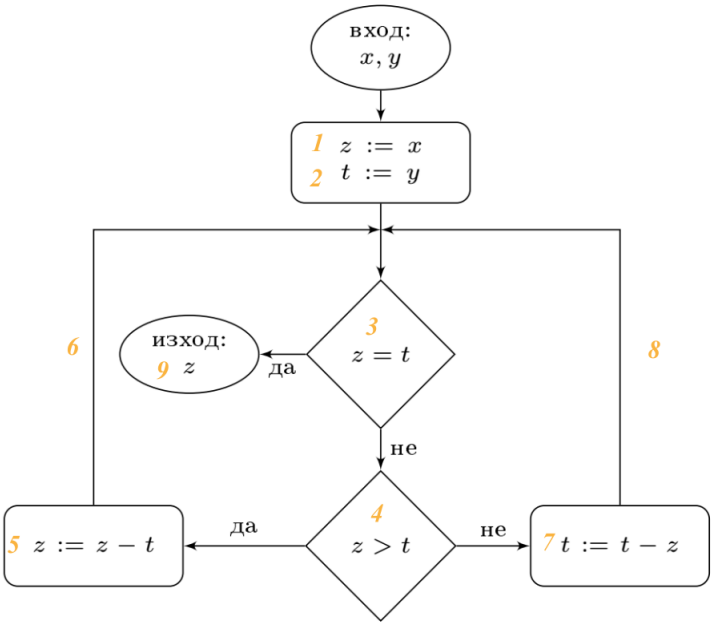
Сега отново комбинираме това равенство с горните (4.13) и (4.14), за да получим, че

$$h_l(x_1, \ldots, x_m) \simeq y_k.$$

□

От доказателството, което проведохме, се вижда, че преводът на всяка стандартна схема в еквивалентната на нея рекурсивна е алгоритмичен. С други думи, не просто е вярно, че за всяка стандартна схема S съществува рекурсивна схема R, еквивалентна на нея, но имаме *общ метод* по схемата S *да конструираме* резултантната схема R. Този метод се нарича *метод на опашковите функции* и се използва за намиране на ефективни рекурсивни програми, тъй като рекурсията в R е изцяло *линейна*.

Задача 4.1. Като приложите метода на опашковите функции, намерете рекурсивната схема, еквивалентна на следната блок-схема:



Решение. Първо да препишем блок-схемата като стандартна програма:

$\texttt{input}(X, Y);\ \texttt{output}(Z)$
1 : $Z := X$
2 : $T := Y$
3 : $\texttt{if } Z = T \texttt{ then goto } 9 \texttt{ else goto } 4$
4 : $\texttt{if } Z > T \texttt{ then goto } 5 \texttt{ else goto } 7$
5 : $Z := Z - T$
6 : $\texttt{goto } 3$
7 : $T := T - Z$
8 : $\texttt{goto } 3$
9 : $\texttt{stop}$

Следвайки описания по-горе алгоритъм, от S получаваме следната рекурсивна програма R:

$$
\begin{array}{lll}
R\colon & F_1(X,Y,Y,Y) & \texttt{where} \\
& F_1(X,Y,Z,T) & = F_2(X,Y,X,T) \\
& F_2(X,Y,Z,T) & = F_3(X,Y,Z,Y) \\
& F_3(X,Y,Z,T) & = \texttt{if } Z=T \texttt{ then } F_9(X,Y,Z,T) \texttt{ else } F_4(X,Y,Z,T) \\
& F_4(X,Y,Z,T) & = \texttt{if } Z>T \texttt{ then } F_5(X,Y,Z,T) \texttt{ else } F_7(X,Y,Z,T) \\
& F_5(X,Y,Z,T) & = F_6(X,Y,Z-T,T) \\
& F_6(X,Y,Z,T) & = F_3(X,Y,Z,T) \\
& F_7(X,Y,Z,T) & = F_8(X,Y,Z,T-Z) \\
& F_8(X,Y,Z,T) & = F_3(X,Y,Z,T) \\
& F_9(X,Y,Z,T) & = Z
\end{array}
$$

След няколко тривиални опростявания достигаме до

$$
\begin{array}{lll}
R\colon & F_1(X,Y,Y,Y) & \texttt{where} \\
& F_1(X,Y,Z,T) & = F_3(X,Y,X,Y) \\
& F_3(X,Y,Z,T) & = \texttt{if } Z=T \texttt{ then } Z \\
& & \qquad \texttt{else if } Z>T \texttt{ then } F_3(X,Y,Z-T,T) \\
& & \qquad\qquad \texttt{else } F_3(X,Y,Z,T-Z)
\end{array}
$$

Всъщност F_3 не зависи от X и Y, затова я преписваме като

$$
\begin{array}{lll}
& F_3(Z,T) & = \texttt{if } Z=T \texttt{ then } Z \\
& & \qquad \texttt{else if } Z>T \texttt{ then } F_3(Z-T,T) \texttt{ else } F_3(Z,T-Z)
\end{array}
$$

или все едно

$$
\begin{array}{lll}
& F_3(X,Y) & = \texttt{if } X=Y \texttt{ then } X \\
& & \qquad \texttt{else if } X>Y \texttt{ then } F_3(X-Y,Y) \texttt{ else } F_3(X,Y-X)
\end{array}
$$

Тогава първото уравнение става

$$F_1(X,Y,Z,T) = F_3(X,Y)$$

Сега изключваме Z и T и от F_1, елиминираме F_3, защото става излишна, и получаваме добре познатата ни рекурсивна програма

$$
\begin{array}{lll}
R\colon & F(X,Y) & \texttt{where} \\
& F(X,Y) & = \texttt{if } X=Y \texttt{ then } X \\
& & \qquad \texttt{else if } X>Y \texttt{ then } F(X-Y,Y) \texttt{ else } F(X,Y-X)
\end{array}
$$

□

4.6 Пример на Патерсън и Хюит

Тук ще посочим пример на рекурсивна схема, която не е еквивалентна на никоя стандартна схема, което означава, че класът на рекурсивните схеми не е транслируем в класа на стандартните схеми ($\mathcal{R} \not\leqslant_t \mathcal{S}$).

Теорема 4.2. (Пример на Патерсън и Хюит.) Да означим с R следната рекурсивна схема:

$F(X)$ `where`
$F(X) \;=\; \texttt{if}\;\; \mathbf{p}(X)\;\; \texttt{then}\;\; X\;\; \texttt{else}\;\; \mathbf{f}(F(\mathbf{g}(X)),\; F(\mathbf{h}(X)))$

За тази схема R *не* съществува стандартна схема S, такава че $R \equiv S$.

Доказателство. Схемата R е в сигнатурата $(\mathbf{f}, \mathbf{g}, \mathbf{h};\; \mathbf{p})$, където $\mathbf{f}, \mathbf{g}$ и $\mathbf{h}$ са функционални символи с местност $2, 1$ и 1, а $\mathbf{p}$ е едноместен предикатен символ. Тъй като за доказателството ще са ни нужни *ербранови структури*, към тази сигнатура добавяме и един константен символ $\mathbf{c}$. Новата сигнатура ще означаваме със Σ:

$$\Sigma \;=\; (\mathbf{c};\; \mathbf{f}, \mathbf{g}, \mathbf{h};\; \mathbf{p}).$$

Всяка ербранова структура в тази сигнатура е с носител множеството H от всички затворени термове в Σ. И тук, както при семантиките, затворените термове ще означаваме с $\mu, \mu_1, \mu_2, \ldots$ Разбира се, сега те ще са в сигнатурата Σ, т.е. ще са обекти от този вид: $\mathbf{c},\; \mathbf{g}(\mathbf{c}),\; \mathbf{h}(\mathbf{c}),\; \mathbf{g}(\mathbf{h}(\mathbf{c})),\; \mathbf{f}(\mathbf{c}, \mathbf{c}),$ $\mathbf{f}(\mathbf{g}(\mathbf{c}), \mathbf{c}),\; \ldots$

За всеки затворен терм μ въвеждаме *норма* на μ (означение: $|\mu|$) — естествено число, равно на броя на функционалните символи $\mathbf{g}$ и $\mathbf{h}$, участващи в μ. Например

$$|\mathbf{c}| \;=\; |\mathbf{f}(\mathbf{c}, \mathbf{c})| \;=\; 0, \quad |\mathbf{g}(\mathbf{c})| \;=\; |\mathbf{h}(\mathbf{c})| \;=\; |\mathbf{f}(\mathbf{g}(\mathbf{c}), \mathbf{c})| \;=\; 1 \;\text{ и пр.}$$

За всяко $n \geq 0$ с p_n ще означаваме едноместния предикат в H, който се дефинира както следва:

$$p_n(\mu) = \begin{cases} \mathbf{t}, & \text{ако } |\mu| = n \\ \mathbf{f}, & \text{иначе.} \end{cases}$$

Да означим с $\mathcal{A}_n$ следната *ербранова* структура:

$$\mathcal{A}_n \;=\; (H;\; c^{\mathcal{A}_n};\; f^{\mathcal{A}_n},\; g^{\mathcal{A}_n},\; h^{\mathcal{A}_n};\; p_n).$$

По дефиниция това означава, че:

— носителят на структурата е множеството $H \stackrel{\text{деф}}{=} \{\mu \mid \mu \text{ е затворен терм в } \Sigma\}$;

— всеки константен и функционален символ се интерпретира "със себе си", което ще рече:

$$c^{\mathcal{A}_n} = \underbrace{\mathbf{c}}_{\in H},\ \ f^{\mathcal{A}_n}(\mu_1,\mu_2) = \underbrace{\mathbf{f}(\mu_1,\mu_2)}_{\in H},\ \ g^{\mathcal{A}_n}(\mu) = \underbrace{\mathbf{g}(\mu)}_{\in H} \text{ и } h^{\mathcal{A}_n}(\mu) = \underbrace{\mathbf{h}(\mu)}_{\in H}.$$

Схемата R от условието на теоремата в структурата $\mathcal{A}_n$ определя следната рекурсивна *програма* $(R, \mathcal{A}_n)$:

$$\begin{array}{ll} F(X) & \texttt{where} \\ F(X) & = \texttt{if}\ \ |X| = n\ \ \texttt{then}\ \ X\ \ \texttt{else}\ \ \mathbf{f}(F(\mathbf{g}(x)),\ F(\mathbf{h}(X))) \end{array} \quad (4.15)$$

Нека

$$\varphi_n \stackrel{\text{деф}}{=} Sem(R, \mathcal{A}_n),$$

т.е. $\varphi_n : H \ \multimap\ H$ е функцията, която се пресмята от горната рекурсивна програма $(R, \mathcal{A}_n)$.

Да видим как ще изглежда φ_n за $n = 0, 1$ и 2. Ключовото наблюдение е, че при всяко рекурсивно извикване $F(\mu)$ се обръща към $F(\mathbf{g}(\mu))$ и $F(\mathbf{h}(\mu))$, и значи нормата на аргумента μ нараства с 1. Оттук следва, че $\varphi_0(\mu)$ ще е дефинирана само за термове μ, които са с норма 0:

$$\varphi_0(\mu) \simeq \begin{cases} \mu, & \text{ако } |\mu| = 0 \\ \neg!, & \text{иначе.} \end{cases}$$

При $n = 1$ ще имаме:

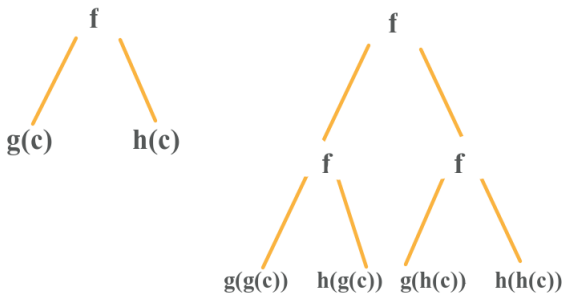
$$\varphi_1(\mu) \simeq \begin{cases} \mu, & \text{ако } |\mu| = 1 \\ f(g(\mu), h(\mu)), & \text{ако } |\mu| = 0 \\ \neg!, & \text{иначе.} \end{cases}$$

В частност, ще имаме, че $\varphi_0(\mathbf{c}) = \mathbf{c}$ и $\varphi_1(\mathbf{c}) = \mathbf{f}(\mathbf{g}(\mathbf{c}), \mathbf{h}(\mathbf{c}))$.

При $n = 2$ ще пресметнем $\varphi_2(\mu)$ само за $\mu = \mathbf{c}$:

$$\begin{array}{l} \varphi_2(\mathbf{c}) \stackrel{|\mathbf{c}|\neq 2}{=} \mathbf{f}(\varphi_2(\mathbf{g}(\mathbf{c})),\ \varphi_2(\mathbf{h}(\mathbf{c}))) \stackrel{|\mathbf{g}(\mathbf{c})|=|\mathbf{h}(\mathbf{c})|\neq 2}{=} \\ \quad \mathbf{f}(\ \mathbf{f}(\varphi_2(\mathbf{g}(\mathbf{g}(\mathbf{c})))),\ \varphi_2(\mathbf{h}(\mathbf{g}(\mathbf{c})))),\ \mathbf{f}(\ \varphi_2(\mathbf{g}(\mathbf{h}(\mathbf{c}))),\ \varphi_2(\mathbf{h}(\mathbf{h}(\mathbf{c}))))) = \\ \quad \mathbf{f}(\ \mathbf{f}(\mathbf{g}(\mathbf{g}(\mathbf{c})),\ \mathbf{h}(\mathbf{g}(\mathbf{c}))),\ \mathbf{f}(\ \mathbf{g}(\mathbf{h}(\mathbf{c})),\ \mathbf{h}(\mathbf{h}(\mathbf{c})))). \end{array}$$

Термовете $\varphi_1(\mathbf{c})$ и $\varphi_2(\mathbf{c})$ можем да си представим графично така:



Нашата близка цел ще бъде да покажем, че за всяко n, $\varphi_n(\mathbf{c})$ е пълното двоично дърво с дълбочина n. За това се оказва удобно да дефинираме с рекурсия по n редицата от термове $\{\tau_n\}_n$, всеки от които е на една променлива X. Дефиницията е ето тази:

$$\begin{aligned} \tau_0(X) &= X \\ \tau_{n+1}(X) &= f(\tau_n[X/\mathbf{g}(X)],\ \tau_n[X/\mathbf{h}(X)]). \end{aligned} \tag{4.16}$$

Тук ще ни е нужен един вариант на Лемата за оценките, който в случая се изказва така: ако $\tau(X)$ и $\rho(X)$ са термове на една свободна променлива X, а μ е затворен терм, то

$$\tau[X/\rho](\mu) \ = \ \tau(\rho(\mu)). \tag{4.17}$$

Разбира се, в това равенство имаме предвид стойност на терм в *ербранова* структура, каквато е $\mathcal{A}_n$. Оттук получаваме следното изразяване на $\tau_{n+1}(\mu)$ чрез $\tau_n(\mathbf{g}(\mu))$ и $\tau_n(\mathbf{h}(\mu))$:

$$\tau_{n+1}(\mu) \overset{\text{деф}}{=} \mathbf{f}(\tau_n[X/\mathbf{g}(X)](\mu),\ \tau_n[X/\mathbf{h}(X)](\mu)) \overset{(4.17)}{=} \mathbf{f}(\tau_n(\mathbf{g}(\mu)),\ \tau_n(\mathbf{h}(\mu))).$$

Чрез термовете τ_n лесно можем да опишем явния вид на всяка функция φ_n — функцията, която се пресмята от програмата (4.15). Твърдим, че за всяко n:

$$\varphi_n(\mu) \simeq \begin{cases} \tau_{n-k}(\mu), & \text{ако } |\mu| = k \leq n \\ \neg!, & \text{ако } |\mu| > n \end{cases} \tag{4.18}$$

Ако μ е терм, за който $|\mu| > n$, то

$$\varphi_n(\mu) \overset{(4.15)}{\simeq} \mathbf{f}(\varphi_n(\underbrace{\mathbf{g}(\mu)}_{|\mathbf{g}(\mu)|>n}),\ \varphi_n(\underbrace{\mathbf{h}(\mu)}_{|\mathbf{h}(\mu)|>n})) \ \simeq \ \ldots$$

и очевидно програмата $Sem(R, \mathcal{A}_n)$ забива, което значи, че $\neg!\varphi_n(\mu)$.

Нека сега $|\mu| = k \leq n$. С индукция по $l = n - k$ ще покажем, че

$$\varphi_n(\mu) \ = \ \tau_{n-k}(\mu).$$

Ако $l = 0$, то $n = k$, т.е. $|\mu| = n$ и тогава

$$\varphi_n(\mu) \overset{(4.15)}{=} \mu \overset{(4.16)}{=} \tau_{n-n}(\mu).$$

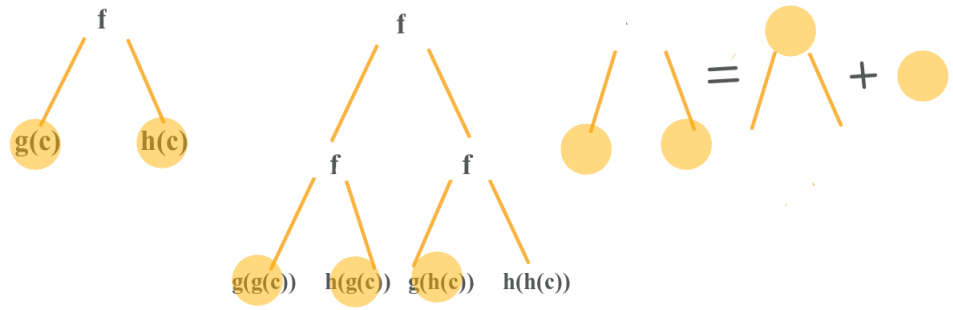
Да допуснем, че за l твърдението е вярно и да го проверим за $l+1$. Нека $|\mu| = k$, като сега $n - k = l + 1$. Ясно е, че в такъв случай $|\mathbf{g}(\mu)| = |\mathbf{h}(\mu)| = k + 1$, и понеже $n - (k + 1) = l$, то за термовете $\mathbf{g}(\mu)$ и $\mathbf{h}(\mu)$ индуктивната хипотеза ще е в сила. Тогава за $\varphi_n(\mu)$ ще имаме:

$$\varphi_n(\mu) \overset{(4.15)}{=} \mathbf{f}(\, \varphi_n(\mathbf{g}(\mu)),\ \varphi_n(\mathbf{h}(\mu))\,) \overset{\text{и.х.}}{=} \mathbf{f}(\, \tau_{n-(k+1)}(\mathbf{g}(\mu)),\ \tau_{n-(k+1)}(\mathbf{h}(\mu))\,)$$

$$= \mathbf{f}(\tau_{(n-k)-1}(\mathbf{g}(\mu)),\ \tau_{(n-k)-1}(\mathbf{h}(\mu))\,) \overset{(4.17)}{=} \tau_{n-k}(\mu).$$

Така показахме, че φ_n наистина има общия вид (4.18). В частност, понеже $|\mathbf{c}| = 0$, $\varphi_n(\mathbf{c}) = \tau_n(\mathbf{c})$.

Да се убедим (идейно), че за пресмятането на пълното двоично дърво $\tau_n(c)$ с дълбочина n със стандартна програма в ербранова структура в сигнатурата $\Sigma = (\mathbf{c};\ \mathbf{f}, \mathbf{g}, \mathbf{h};\ \mathbf{p})$ ще са нужни поне $n + 1$ регистъра.



Сега да допуснем, че съществува стандартна схема S, която е еквивалентна на схемата R от условието на теоремата. Според *Определение* 4.1, за всяка структура $\mathcal{A}$ в сигнатурата Σ ще е в сила равенството

$$Sem(R, \mathcal{A}) = Sem(S, \mathcal{A}).$$

Нека схемата S има памет $X_1, \ldots, X_m$, с други думи, в S се срещат най-много m различни променливи. За да получим противоречие с допуснатото, да се спрем на ербрановата структура $\mathcal{A}_m$. Съгласно горното равенство, би трябвало $Sem(R, \mathcal{A}_m) = Sem(S, \mathcal{A}_m)$, и значи

$$Sem(S, \mathcal{A}_m) = \varphi_m.$$

Тогава $Sem(S, \mathcal{A}_m)(\mu) \simeq \varphi_m(\mu)$ за всеки затворен терм μ. В частност, при $\mu = \mathbf{c}$ ще имаме, че $Sem(S, \mathcal{A}_m)(\mathbf{c}) = \tau_m(\mathbf{c})$, т.е. стандартната програма $(S, \mathcal{A}_m)$ при вход $\mathbf{c}$ връща пълното двоично дърво с дълбочина m. Но това противоречи на нашето наблюдавахме по-горе, че за пресмятането на $\tau_m(\mathbf{c})$ със стандартна програма са нужни $m + 1$ регистъра, а S има най-много m. □

Азбучен указател