Ръководството съдържа задачи върху основните раздели на курса *Семантика на езиците за програмиране*, който се чете на студентите от специалност Информатика на Факултета по математика и информатика при СУ „Св. Климент Охридски“ през последните няколко години. Всеки параграф започва с подробно изложение на теоретичния материал, необходим за решаване на задачите в него, което прави книгата независима от други учебни пособия. Голяма част от задачите имат пълни решения или подробни упътвания, а останалите — отговори.

Освен на студентите, за които е предназначено, ръководството може да бъде полезно и на по-широк кръг от читатели с интереси в областта на теоретичната информатика.

Александра Соскова   Стела Николова

СЕМАНТИКА НА ЕЗИЦИТЕ ЗА ПРОГРАМИРАНЕ
Ръководство

Редактор: Грозданка Благоева

Формат 60×84/16

ИК СОФТЕХ
тел. (02) 866 37 11

2008

ISBN 978–954–8495–41–7

СЪДЪРЖАНИЕ

ПРЕДГОВОР

В ръководството са разгледани широк кръг от задачи в областта на теоретичното програмиране, обособени в следните теми: *Методи за верификация на програми (итеративни и рекурсивни), Компактни оператори* и *Денотационна семантика на езиците за програмиране.*

В началото на всеки раздел са дадени основните понятия и теоретични резултати, необходими за решаването на предложените задачи. Читателят може да се запознае подробно с теорията от учебника „Теория на програмите“ на И. Сосков и А. Дичев. Някои от задачите тук са всъщност теореми от цитирания учебник. Те са придружени с доказателства в случаите, когато тези доказателства илюстрират полезни идеи. На част от задачите (сред тях задължително тези, които са характерни за даден клас от проблеми) са дадени подробни доказателства и коментари, друга част имат само упътвания и отговори, а някои са оставени за самостоятелна работа. Трудните задачи са отбелязани със звездичка.

Всички задачи, решения и упътвания, поместени тук, са подготвени със съвместните усилия и на двамата автори, но окончателният подбор е направен по следния начин:

— § 1.1, § 1.2, § 1.3, § 2.1 и трета глава — от С. Николова;

— § 1.4, § 2.2, § 2.3 и четвърта глава — от А. Соскова.

Бихме искали да изкажем своята благодарност на всички, които по някакъв начин допринесоха за появата на тази книга. Преди всичко сме признателни на нашите учители и колеги от Катедрата по математическа логика за колегиалното отношение и моралната подкрепа. Специално благодарим на проф. Иван Сосков, доц. Тинко Тинчев и доц. Ангел Дичев, които отзивчиво откликваха с компетентни мнения и приятелски съвети по всички въпроси, свързани с тематиката на книгата. Тук искаме да споменем и нашия колега Валентин Горанко, с когото започнахме упражненията по тази дисциплина. Благодарни сме и на нашите по-млади колеги Весела Балева и Владимир Сосков за уместните забележки по ръкописа на книгата. Благодарим и на г-жа Мария Велчева за техническата помощ, която ни оказа.

ОСНОВНИ ОЗНАЧЕНИЯ

Първа глава

ВЕРИФИКАЦИЯ НА ИТЕРАТИВНИ ПРОГРАМИ

§ 1.1. Принцип на структурната индукция в множества с фундирана наредба

Принцип на математическата индукция. *Нека P е свойство в множеството $\mathbb{N} = \{0, 1, \ldots\}$ на естествените числа, за което е изпълнено:*

(i) *0 притежава свойството P;*

(ii) *при всеки избор на $n \in \mathbb{N}$, ако n притежава свойството P, то и $n+1$ притежава свойството P.*

Тогава всяко естествено число n притежава свойството P.

В някои случаи е по-удобно да се използва следната модификация на този принцип, известна като

Принцип на възвратната (пълната) индукция. *Нека P е свойство на естествените числа, за което е изпълнено:*

(i′) $P(0)$;

(ii′) *за произволно $n \in \mathbb{N}$, ако всяко от числата $0, 1, \ldots, n$ има свойството P, то и $n+1$ има свойството P.*

Тогава за всяко $n \in \mathbb{N}$ е вярно $P(n)$.

Всъщност двата принципа са еквивалентни (в смисъла, уточнен в зад. 1). Първата формулировка на индуктивния принцип отразява начина, по който могат да се построят естествените числа: всяко естествено число е или 0, или се получава от друго чрез добавяне на единица. Втората формулировка е по-тясно свързана с наредбата $<$ („по-малко“) на естествените числа, както се вижда от следния еквивалентен запис на принципа на възвратната индукция:

Нека P е свойство в $\mathbb{N}$ и при всеки избор на $n \in \mathbb{N}$ е изпълнено условието

$$(\forall m_{m<n} P(m)) \Longrightarrow P(n).$$

Тогава $\forall n P(n)$.

Ще покажем, че индуктивен принцип от този тип е валиден и за множества от произволни обекти, в които е въведена подходяща бинарна релация $<$.

Нека X е множество, а $<$ е бинарна релация в X. Казваме, че $<$ е *(строга) частична наредба* на X, или $(X, <)$ е *(строго) частично наредено множество*, ако за всяко x, y, z от X са изпълнени условията:

1) $\neg(x < x)$ — антирефлексивност;

2) $x < y \ \& \ y < z \Longrightarrow x < z$ — транзитивност.

Нека $(X, <)$ е частично наредено множество и $Y \subseteq X$. Казваме, че y е *минимален елемент* на Y, ако $y \in Y$ и не съществува $z \in Y$, за което е изпълнено $z < y$. Частичната наредба $<$ наричаме *фундирана* (или множеството $(X, <)$ — *фундирано множество*), ако всяко непразно подмножество на X притежава поне един минимален елемент.

Принцип на структурната индукция. *Нека $(X, <)$ е фундирано множество, а P е свойство в X, за което при всеки избор на $x \in X$ е изпълнено условието:*

(*) *ако за всяко $y < x$ е вярно $P(y)$, то е вярно и $P(x)$.*

Тогава $P(x)$ е в сила за всяко $x \in X$.

Доказателство. Да допуснем, че за някое $x \in X$ не е вярно $P(x)$ и да образуваме множеството $Y = \{x \,|\, x \in X \ \& \ \neg P(x)\}$. Y не е празно и следователно притежава поне един минимален елемент x_0. Да отбележим, че по определение $x_0 \in Y$, т. е. вярно е $\neg P(x_0)$.

Нека $y < x_0$. Тогава $y \notin Y$ и следователно е в сила $P(y)$. Тъй като това е валидно за всяко $y < x_0$, то от (*) следва, че $P(x_0)$ е в сила — противоречие с избора на x_0.

Задача 1. Нека P е свойство в множеството на естествените числа.

а) Ако за P е вярно $P(0)$ и $\forall n(P(0) \ \& \ldots \& \ P(n) \Longrightarrow P(n+1))$, като се използва принципът на обичайната математическа индукция, да се докаже, че $\forall n P(n)$.

б) Ако за P е вярно $P(0)$ и $\forall n(P(n) \Longrightarrow P(n+1))$, като се използва принципът на възвратната индукция, да се докаже, че $\forall n P(n)$.

Доказателство. а) Нека за P е вярно

$$P(0) \quad \text{и} \quad \forall n(P(0) \ \& \ldots \& \ P(n) \Longrightarrow P(n+1)).$$

Да дефинираме свойство Q в $\mathbb{N}$ чрез еквивалентността:

$$Q(n) \iff P(0) \ \& \ldots \& \ P(n).$$

Ясно е, че Q е вярно за 0. Ще покажем, че $\forall n(Q(n) \Longrightarrow Q(n+1))$.

Нека за произволно n е вярно $Q(n)$, т.е. $P(0) \ \& \ldots \& \ P(n)$. Тогава поради избора на P ще е вярно и $P(n+1)$. Получихме, че

от $Q(n)$ следва $P(n+1)$. Но от $Q(n)$ следва и $P(0) \ \& \ldots \& \ P(n)$. Окончателно, от $Q(n)$ следва $P(0) \ \& \ldots \& \ P(n) \ \& \ P(n+1)$, или все едно, вярна е импликацията $Q(n) \Longrightarrow Q(n+1)$. Оттук по принципа на математическата индукция, приложен за Q, получаваме $\forall n Q(n)$ и следователно $\forall n P(n)$.

б) Нека за свойството P е изпълнено

$$P(0) \ \& \ \forall n(P(n) \Longrightarrow P(n+1)).$$

От последното условие очевидно следва

$$\forall n(P(0) \ \& \ldots \& \ P(n) \Longrightarrow P(n+1)),$$

откъдето с принципа на възвратната индукция отново получаваме $\forall n P(n)$.

Задача 2. Да се докаже с помощта на принципа на възвратната индукция, че всяко естествено число може да се представи като произведение от прости множители.

Задача 3. Нека $(X, <)$ е фундирано множество и Y е подмножество на X. Да се докаже, че

$$\forall x_{x \in X}(\{y \,|\, y \in X \ \& \ y < x\} \subseteq Y \Longrightarrow x \in Y) \Longrightarrow Y = X.$$

Задача 4. Нека $(X, <)$ е частично наредено множество. Да се докаже, че следните две условия са еквивалентни:

(1) всяко непразно подмножество на X притежава поне един минимален елемент;

(2) не съществува безкрайно намаляваща (относно $<$) редица от елементи на X.

Доказателство. Нека е изпълнено (1) и да допуснем, че в X има безкрайно намаляваща редица $x_0 > x_1 > \ldots$ Тогава множеството $Y = \{x_n \,|\, n = 0, 1, \ldots\}$ няма минимални елементи, защото ако y е минимален елемент на Y, то $y = x_n$ за някое n и следователно $x_{n+1} < y$.

Обратно, нека частичната наредба $<$ има свойството (2). Да допуснем, че съществува непразно множество $Y \subseteq X$, което няма минимални елементи. Ще построим безкрайно намаляваща редица $y_0 > y_1 > y_2 > \ldots$ от елементи на Y (а следователно и на X), което ще ни доведе до противоречие с условието (2).

Нека y_0 е произволен елемент на Y. Да приемем, че за някое $n \geqq 0$ сме построили редица $y_0, y_1, \ldots, y_n$ от елементи на Y, за която е изпълнено

$$y_0 > y_1 > \cdots > y_n.$$

Тъй като в Y няма минимални елементи, за $y_n \in Y$ ще съществува поне едно $z \in Y$, за което $z < y$. Полагаме $y_{n+1} = z$.

Забележка. Така описаната конструкция е всъщност индуктивна дефиниция на функция $h : \mathbb{N} \to Y$, където $h(n) = y_n$.

Задача 5. Нека $(X, <_1)$ и $(Y, <_2)$ са фундирани множества и $X \cap Y = \varnothing$. В множеството $X \cup Y$ въвеждаме бинарната релация $<$ по следния начин:

$$y < x \iff$$

$$(x, y \in X \text{ и } x <_1 y) \text{ или } (x, y \in Y \text{ и } x <_2 y) \text{ или } (x \in X \text{ и } y \in Y).$$

Да се докаже, че множеството $(X \cup Y, <)$ също е фундирано.

Доказателство. Непосредствено се проверява, че $<$ е частична наредба в $X \cup Y$. Да допуснем, че съществува безкрайно намаляваща редица

$$z_1 > z_2 > \ldots$$

от елементи на $X \cup Y$. Ако $z_n \in X$ за всяко $n \in \mathbb{N}$, то редицата $\{z_n\}_n$ ще е безкрайно намаляваща относно наредбата $<_1$ — противоречие с фундираността на $(X, <_1)$.

Следователно съществува n, така че $z_n \in Y$. Тогава съгласно определението на релацията $<$ получаваме, че $z_k \in Y$ за всяко $k \geqq n$. Но в такъв случай редицата

$$z_n >_2 z_{n+1} >_2 \ldots$$

е безкрайно намаляваща в Y, което противоречи на фундираността на $(Y, <_2)$.

Нека $(X, <)$ е частично наредено множество и $Y \subseteq X$. Казваме, че $y \in Y$ е *най-малък елемент* на Y ($y = \min(Y)$), ако за всяко $z \in Y$ е вярно, че $y = z$ или $y < z$.

Задача 6. Дайте пример за частично наредено множество $(X, <)$, за което е вярно, че:

а) X има поне един минимален елемент, но няма най-малък елемент;

б) X има точно един минимален елемент, но няма най-малък елемент;

в) X няма минимални елементи.

Частично нареденото множество $(X, <)$ наричаме *добре наредено*, ако всяко непразно подмножество на X има най-малък елемент.

Казваме, че частичната наредба $<$ е *линейна наредба* в X, ако при всеки избор на x, y от X е изпълнено условието

$$x < y \lor x = y \lor y < x.$$

Задача 7. Да се докаже, че частично нареденото множество $(X, <)$ е добре наредено точно тогава, когато $(X, <)$ е фундирано и $<$ е линейна наредба в X.

Задача 8. Кои от изброените множества са фундирани? Кои от тях са добре наредени?

а) $(\mathbb{N}, <)$;

б) $(\mathbb{Z}, <)$, където $\mathbb{Z}$ е множеството на целите числа, а $<$ е релацията „по-малко“ в $\mathbb{Z}$;

в) $(\mathcal{For}, <)$, където $\mathcal{For}$ е множеството на съждителните формули, а

$$\varphi < \psi \iff \varphi \text{ е подформула на } \psi \text{ и } \varphi \neq \psi;$$

г) $(2^{\mathbb{N}}, \subset)$, където $2^{\mathbb{N}}$ е съвкупността от всички подмножества на $\mathbb{N}$, а $\subset$ е теоретико-множествената релация „строго включване“;

д) $(\mathcal{Fin}(\mathbb{N}), \subset)$, където $\mathcal{Fin}(\mathbb{N})$ е съвкупността от всички крайни подмножества на $\mathbb{N}$;

е) $(\mathbb{N}^+, |)$, където $\mathbb{N}^+ = \{1, 2, \ldots\}$, а $m|n \iff m \neq n$ и m дели n.

Упътване. г) Нека $\mathbb{N}_n = \mathbb{N} \setminus \{0, \ldots, n\}$. Редицата $\{\mathbb{N}_n\}_n$ е безкрайно намаляваща и следователно множеството $(2^{\mathbb{N}}, \subset)$ не е фундирано.

д) Докажете, че всяко крайно подмножество на $\mathbb{N}$ има краен брой предшественици относно наредбата $\subset$. От това ще следва, че не съществува безкрайно намаляваща редица от крайни подмножества на $\mathbb{N}$ и следователно $(\mathcal{Fin}(\mathbb{N}), \subset)$ е фундирано множество.

Нека $\mathbb{N}\times\mathbb{N}$ е множеството от всички наредени двойки от естествени числа. *Лексикографска наредба* на $\mathbb{N}^2$ ще наричаме бинарната релация $\prec$ със следното определение:

$$(x,y)\prec(x',y') \iff x<x' \vee (x=x' \;\&\; y<y').$$

Задача 9. Да се докаже, че $(\mathbb{N}^2,\prec)$ е добре наредено множество.

Доказателство. Аксиомите за антирефлексивност и транзитивност на релацията $\prec$ следват лесно от факта, че релацията $<$ е наредба в $\mathbb{N}$. За да покажем, че $\prec$ е добра наредба, избираме произволно непразно множество $Y\subseteq\mathbb{N}^2$. Да положим

$$x_0=\min\{x\,|\,(x,y)\in Y \text{ за някое } y\in\mathbb{N}\};$$
$$y_0=\min\{y\,|\,(x_0,y)\in Y\}.$$

Да съобразим, че (x_0,y_0) е най-малкият елемент на Y. Наистина, нека (x,y) е произволна наредена двойка от Y. Тогава съгласно избора на x_0 ще имаме $x_0\leqq x$. Ако $x_0<x$, то $(x_0,y_0)\prec(x,y)$; а ако $x_0=x$, то $y_0\leqq y$ и следователно $(x_0,y_0)\prec(x,y)$ или $(x_0,y_0)=(x,y)$.

Задача 10. Нека $(X,<)$ е фундирано множество.

а) В декартовата степен X^n на X, $n\geqq 1$, въвеждаме бинарната релация $<_n$ с условието:

$$(x_1,\ldots,x_n)<_n(y_1,\ldots,y_n) \iff$$
$$\exists i_{1\leqq i\leqq n}(x_1=y_1 \;\&\;\ldots\;\&\; x_{i-1}=y_{i-1} \;\&\; x_i<y_i).$$

Да се докаже, че $<_n$ е фундирана наредба в X^n.

б) Да означим с X^* съвкупността от всички крайни редици от елементи на X. Нека $<_*$ е следната бинарна релация в X^*:

$$(x_1,\ldots,x_m)<_*(y_1,\ldots,y_n) \iff (m<n \;\&\; x_1=y_1 \;\&\;\ldots\;\&\; x_m=y_m)\vee$$
$$\exists i_{1\leqq i\leqq \min(m,n)}(x_1=y_1 \;\&\;\ldots\;\&\; x_{i-1}=y_{i-1} \;\&\; x_i<y_i).$$

Да се докаже, че $<_*$ е частична наредба в X^*, която (в общия случай) не е фундирана.

Упътване. б) Разгледайте нареденото множество $(X,<)$, където $X=\{x,y\}$ и $x<y$. Редицата $y>_* xy>_* xxy>_*\ldots$ е безкрайно намаляваща в X^* и следователно $<_*$ не е фундирана наредба.

Задача 11. Нека за функцията $f:\mathbb{N}^2\to\mathbb{N}$ е изпълнено:

$$f(x,y)=\begin{cases} y, & \text{ако } x=0,\\ f(x,y-x), & \text{ако } y\geqq x>0,\\ f(y,x), & \text{ако } y<x.\end{cases}$$

Да се докаже, че $f(x,y)=\text{НОД}(x,y)$ за всяко $(x,y)\in\mathbb{N}^2$. (Тук приемаме, че $\text{НОД}(0,0)=0$.)

Доказателство. Първи начин: със структурна индукция по лексикографската наредба. Да определим свойството P в множеството $\mathbb{N}^2$ с еквивалентността:

$$P(x,y) \iff f(x,y)=\text{НОД}(x,y).$$

Нека (x,y) е произволна наредена двойка от $\mathbb{N}^2$. Да приемем, че за всяка двойка (x',y'), за която $(x',y')\prec(x,y)$, е в сила $P(x',y')$. Ще покажем, че е в сила и $P(x,y)$.

Ще разгледаме последователно различните случаи от условието, което удовлетворява функцията f.

I *случай*: $x=0$. Имаме $f(0,y)=y$ и $\text{НОД}(0,y)=y$ (включително при $y=0$, съгласно уговорката $\text{НОД}(0,0)=0$).

II *случай*: $y\geqq x>0$. Тогава $f(x,y)=f(x,y-x)$. Тъй като $x>0$, то $y-x<y$ и следователно $(x,y-x)\prec(x,y)$. Тогава от индуктивното предположение за $(x,y-x)$ ще имаме $f(x,y-x)=\text{НОД}(x,y-x)$, а от свойствата на НОД — $\text{НОД}(x,y-x)=\text{НОД}(x,y)$. Така получаваме

$$f(x,y)=f(x,y-x)=\text{НОД}(x,y).$$

III *случай*: $y<x$. Тъй като $(y,x)\prec(x,y)$, от индуктивното предположение за (y,x) получаваме $f(y,x)=\text{НОД}(y,x)$. От друга страна, в този случай $f(x,y)=f(y,x)$ и следователно

$$f(x,y)=f(y,x)=\text{НОД}(x,y).$$

Тъй като разгледаните случаи изчерпват всички възможности за двойката (x,y), можем да твърдим, че е в сила импликацията

$$(\forall(x'y')_{\prec(x,y)}P(x',y'))\Longrightarrow P(x,y).$$

Сега от принципа на структурната индукция, приложен за фундираното множество $(\mathbb{N}^2, \prec)$, получаваме $\forall (x,y)_{\in \mathbb{N}^2} P(x,y)$, т. е. $f(x,y) = \text{НОД}\ (x,y)$ за всяко $(x,y) \in \mathbb{N}$.

Втори начин. Нека Q е следното свойство в $\mathbb{N}$:

$$Q(z) \iff \forall x_{\in\mathbb{N}} \forall y_{\in\mathbb{N}} (x+y=z \iff f(x,y) = \text{НОД}(x,y)).$$

С възвратна индукция по $z \in \mathbb{N}$ ще покажем, че $\forall z Q(z)$, откъдето очевидно ще следва, че $f(x,y) = \text{НОД}(x,y)$ за всяко $(x,y) \in \mathbb{N}^2$.

Ако $z = 0$, то $x = y = 0$ и $f(0,0) = \text{НОД}(0,0)$. Да изберем произволно $z > 0$ и да допуснем, че за всяко $z_0 < z$ е вярно $Q(z_0)$. Нека още x и y са такива, че $x + y = z$. Отново разглеждаме различните възможности за x и y.

I *случай*: $x = 0$. Тогава по определение $f(x,y) = y = \text{НОД}(x,y)$.

II *случай*: $y \geqq x > 0$. Тогава $x + (y-x) < x + y = z$. Като използвате индуктивното предположение за $z_0 = x + (y-x)$, проверете, че отново $f(x,y) = \text{НОД}(x,y)$.

III *случай*: $y < x$. От избора на функцията f имаме

$$f(x,y) = f(y,x). \tag{1}$$

Ако $y = 0$, то $f(y,x) = f(0,x) = x$ и $\text{НОД}(y,x) = \text{НОД}(0,x) = x$.

Нека $y > 0$. Имаме още $y < x$, и оттук, съгласно избора на f,

$$f(y,x) = f(y, x-y). \tag{2}$$

От друга страна, от индуктивното предположение за

$$z_0 = x + (y-x) < x + y$$

е вярно, че

$$f(y, x-y) = \text{НОД}(y, x-y) = \text{НОД}(x,y). \tag{3}$$

От равенствата (1), (2) и (3) следва $f(x,y) = \text{НОД}\ (x,y)$.

Получихме окончателно

$$(\forall z_{0<z} Q(z_0)) \Longrightarrow Q(z).$$

Тъй като z е произволно, от принципа на възвратната индукция ще следва, че $\forall z Q(z)$.

Задача 12 (функция 91 на Маккарти). Нека F е функция в множеството на целите числа, удовлетворяваща условието

$$F(x) = \begin{cases} x - 10, & \text{ако } x > 100, \\ F(F(x+11)), & \text{ако } x \leqq 100. \end{cases}$$

Да се докаже, че

$$F(x) = \begin{cases} x - 10, & \text{ако } x > 100, \\ 91 & \text{иначе.} \end{cases}$$

Доказателство. Нека $\mathbb{Z}$ е множеството на целите числа. Ще покажем, че за всяко $x \leqq 100$ е изпълнено $F(x) = 91$. За целта в $\mathbb{Z}$ въвеждаме следната наредба $\prec$:

$$x \prec y \iff y < x \ \&\ x \leqq 100.$$

(Тук $<$ е обичайната наредба в $\mathbb{Z}$).

Лесно се проверява, че $(\mathbb{Z}, \prec)$ е частично наредено множество. Ако допуснем, че съществуват цели числа $x_0, x_1, x_2, \ldots$, за които

$$x_0 \succ x_1 \succ x_2 \succ \cdots,$$

то $x_0 < x_1 < x_2 \cdots < 101$, което е невъзможно. Следователно множеството $(\mathbb{Z}, \prec)$ е фундирано. С индукция по наредбата $\prec$ ще покажем, че $F(x) = 91$ за всяко $x \leqq 100$.

1. Нека $x = 100$. В този случай $F(x) = F(F(111)) = F(101) = 91$.
2. Нека $x < 100$. Допускаме, че $\forall z \prec x\, F(z) = 91$.

а) $x + 11 > 100$, т.е. $89 < x < 100$. Тогава

$$F(x) = F(F(x+11)) = F(x+1).$$

Но $x + 1 \prec x$ и от индукционното предположение за $x + 1$ имаме, че $F(x+1) = 91$.

б) $x + 11 \leqq 100$, т.е. $x \leqq 89$. Тогава $F(x) = F(F(x+11))$. Но $x + 11 < 101$ и следователно $x + 11 \prec x$. Тогава от индукционното предположение за $x + 11$ имаме, че $F(x+11) = 91$. Така $F(x) = F(91)$. Но $91 \prec x$ и използвайки индукционното предположение за 91, получаваме, че $F(x) = F(91) = 91$.

Задача 13. Нека за функцията $F : \mathbb{N}^2 \to \mathbb{N}$ е изпълнено условието

$$F(x,y) = \begin{cases} 1, & \text{ако } x = y, \\ (y+2).(y+1).F(x,y+2), & \text{в противен случай.} \end{cases}$$

Да се докаже, че $\forall x \forall y (x \geqq y \ \& \ (x-y)$ е четно $\Longrightarrow F(x,y) = \frac{x!}{y!})$.

Задача 14 (функция на Акерман). Да се докаже, че съществува единствена функция F, която удовлетворява условията

$$(4) \quad \left| \begin{array}{l} F(0,y) = y+1 \\ F(x+1,0) = F(x,1) \\ F(x+1,y+1) = F(x, F(x+1,y)), \end{array} \right.$$

и тази функция е тотална.

Упътване. Съществуване. За да покажем, че такава функция съществува, с индукция по лексикографската наредба дефинираме следната редица $\{a_{x,y}\}_{x,y}$ от естествени числа:

$$a_{x,y} = \begin{cases} y+1, & \text{ако } x = 0, \\ a_{x-1,1}, & \text{ако } x > 0 \text{ и } y = 0, \\ a_{x-1,a_{x,y-1}}, & \text{ако } x > 0 \text{ и } y > 0. \end{cases}$$

Проверете, че функцията

$$F(x,y) = a_{x,y} \text{ за всяко } x,\ y \in \mathbb{N},$$

е тотална и удовлетворява равенствата (4).

Единственост. Да допуснем, че функциите F_1 и F_2 удовлетворяват тези равенства, и да положим

$$P(x,y) \Longleftrightarrow F_1(x,y) = F_2(x,y).$$

Като използвате индукция по лексикографската наредба, докажете, че $\forall (x,y) \in \mathbb{N}^2 \ P(x,y)$, с други думи $F_1 = F_2$.

§ 1.2. Верификация на блок-схеми с един цикъл

В този и следващия параграф ще разглеждаме програми, написани на езика на блок-схемите (с общоприетия синтаксис). Нека P е произволна програма с k входни променливи $x_1, \dots, x_k$ и l изходни променливи $y_1, \dots, y_l$.

Всяко *входно условие* (или *предусловие*) $A(x_1, \dots, x_k)$ за програмата P задава най-общо ограниченията, които трябва да удовлетворяват входните данни, за да е възможно изпълнението на P (например в условието A се описват типа на данните, над които работи програмата, някои основни връзки между тях и т.н.).

Всяко *изходно условие* (или *постусловие*) $C(x_1, \dots, x_k, y_1, \dots, y_l)$ за програмата P определя някакви основни зависимости между входните и изходните променливи на P, когато изпълнението ѝ завърши.

Нека A и C са съответно входно и изходно условие за програмата P. Казваме, че P е *частично коректна относно* A и C, ако при всеки вход $\overline{\mathrm{x}} = (\mathrm{x}_1, \dots, \mathrm{x}_k)$, за който е вярно $A(\overline{\mathrm{x}})$, е изпълнено:

ако P завършва при вход $\overline{\mathrm{x}}$ с резултат $\overline{\mathrm{y}} = (\mathrm{y}_1, \dots, \mathrm{y}_l)$, то е вярно $C(\overline{\mathrm{x}}, \overline{\mathrm{y}})$.

Казваме, че P *завършва при условие* A, ако за всеки вход $\overline{\mathrm{x}}$, който удовлетворява входното условие A, е вярно, че P завършва изпълнението си.

Програмата P е *тотално коректна относно* условията A и C, ако P е частично коректна относно A и C и завършва при условие A.

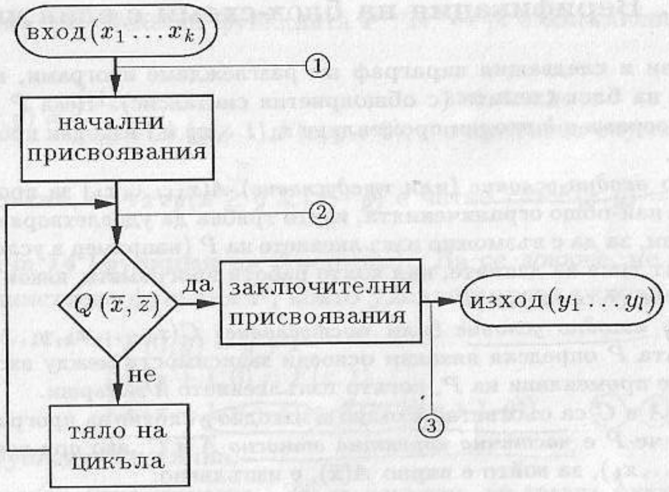
Нека записът $P(\overline{\mathrm{x}}) \to \overline{\mathrm{y}}$ означава, че програмата P завършва над вход $\overline{\mathrm{x}}$ с резултат $\overline{\mathrm{y}}$. Тогава условията за частична и тотална коректност могат да се запишат по-кратко по следния начин:

$$P \text{ е частично коректна относно } A \text{ и } C \Longleftrightarrow$$
$$\forall \overline{\mathrm{x}} \forall \overline{\mathrm{y}} ((A(\overline{\mathrm{x}}) \ \& \ P(\overline{\mathrm{x}}) \to \overline{\mathrm{y}}) \Longrightarrow C(\overline{\mathrm{x}}, \overline{\mathrm{y}}));$$
$$P \text{ е тотално коректна относно } A \text{ и } C \Longleftrightarrow$$
$$\forall \overline{\mathrm{x}} (A(\overline{\mathrm{x}}) \Longrightarrow \exists \overline{\mathrm{y}} (P(\overline{\mathrm{x}}) \to \overline{\mathrm{y}} \ \& \ C(\overline{\mathrm{x}}, \overline{\mathrm{y}}))).$$

Тук ще опишем един начин за доказване на частична коректност на блок-схеми с един цикъл, който се основава на понятието инвариант на цикъл. В следващия параграф ще разгледаме обобщение на този метод, известен като метод на индуктивните твърдения на Флойд.

На фиг. 0 е показана блок-схема с един цикъл P. С $(x_1, \dots, x_k)$ сме означили входните променливи на програмата P, с $(y_1, \dots, y_l)$ — изходните променливи на P, а списъкът $\overline{z} = (z_1, \dots, z_m)$ задава работните променливи на P. Ще предполагаме, че $\{y_1, \dots, y_l\} \subseteq \{z_1, \dots, z_m\}$ и че никоя входна променлива x_i не се среща в лявата част на оператор за присвояване (и следователно съдържанието ѝ не се изменя в процеса на изчисление).

Нека B е условие, свързващо текущите стойности $\overline{\mathrm{x}} = (\mathrm{x}_1, \dots, \mathrm{x}_k)$ и



Фиг. 0

$\overline{z} = (z_1, \dots, z_m)$ на променливите $(x_1, \dots, x_k)$ и $(z_1, \dots, z_m)$. (За да не претоварваме означенията, ще използваме едни и същи символи, но в различен шрифт, за имената на променливите и техните стойности в определен момент от работата на програмата). Да предположим, че се интересуваме от верността на $B(\overline{x}, \overline{z})$ непосредствено преди някое изпълнение на оператора за проверка. Образно бихме могли да си мислим, че сме фиксирали контролна точка от стрелката, водеща към този оператор (означена като точка 2 на фиг. 0), и разглеждаме условието B за текущите стойности на променливите в момента на преминаване през тази точка.

Да фиксираме и още две контролни точки от P: точка 1 — непосредствено след оператора за вход, и точка 3 — непосредствено преди оператора за изход.

По-нататък, когато казваме, че едно условие е изпълнено при някакво попадане в дадена точка, ще имаме предвид, че това условие е вярно за текущите стойности на променливите в момента на преминаване през тази точка.

Да фиксираме A и C — съответно входно и изходно условие за програмата P. Казваме, че условието $B(\overline{x}, \overline{z})$ е *инвариант* на (цикъла на) програмата P *относно* A и C, ако са изпълнени следните условия:

(i) ако в точка 1 е изпълнено входното условие A, то при първо попадане в точка 2 е изпълнено условието B;

(ii) ако при някое попадане в точка 2 е изпълнено условието B, и (преминавайки през тялото на цикъла) отново отидем в точка 2, то там отново е изпълнено B;

(iii) ако при някое попадане в точка 2 са изпълнени едновременно B и условието Q от оператора за проверка, то в точка 3 ще е изпълнено изходното условие C.

Забележка. Когато условията A и C се подразбират, ще казваме само, че B е инвариант на P.

При горните предположения е вярно следното

Твърдение. *Нека A и C са съответно входно и изходно условие за програмата P. Да предположим, че сме намерили инвариант на P относно A и C. Тогава програмата P е частично коректна относно условията A и C.*

!

Доказателство. Нека B е инвариант на P относно A и C, а входните данни $\overline{x}$ удовлетворяват входното условие A. Да предположим още, че P е завършила работата си при вход $\overline{x}$ с резултат $\overline{y}$. Тогава изпълнението на програмата е преминало краен брой пъти (например t) през точка 2.

С индукция по $n = 1, \dots, t$ ще покажем, че при n-тото по ред попадане в точка 2 е изпълнено условието B. Наистина при $n = 1$ това твърдение следва непосредствено от (i), а ако допуснем, че за някое $n < t$ то е в сила, верността му за $n + 1$ ще следва от (ii).

В частност, условието B ще е в сила при t-тото (последното) попадане в точка 2. Но тогава поради избора на t ще е вярно и условието Q от оператора за проверка. Оттук по (iii) получаваме, че в точка 3 ще е изпълнено изходното условие C. Тъй като текущите стойности на входните и изходните променливи в този момент са точно $\overline{x}$ и $\overline{y}$, то ще е вярно $C(\overline{x}, \overline{y})$.

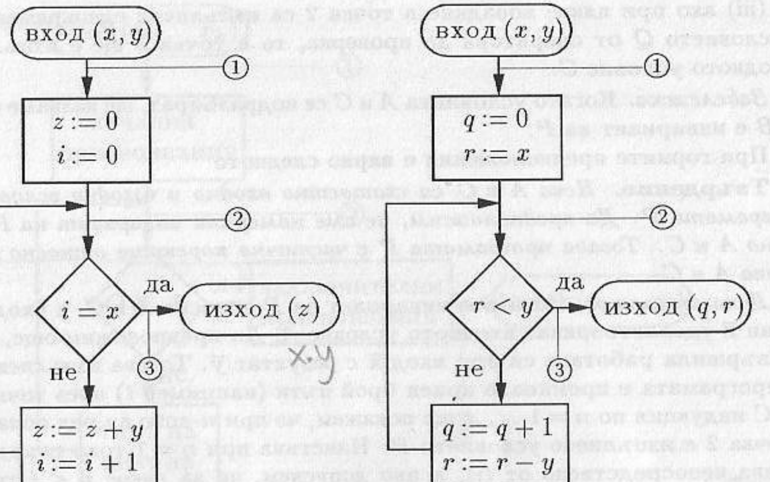
Нека P е програма с входни променливи $x_1, \dots, x_k$ и изходна променлива y. Казваме, че P пресмята функцията $f : X^k \to Y$, ако P е тотално коректна относно входно условие $A : \overline{x} \in X^k$ и изходно условие $C : y = f(\overline{x})$.

Задача 1. Да се докаже, че програмата P_1 (фиг. 1) е тотално коректна относно входно условие $A : x \in \mathbb{N}$, $y \in \mathbb{N}$ и изходно условие $C : z = x.y$.

Доказателство. Най-напред ще проверим, че програмата P_1 е частично коректна относно A и C, а след това ще покажем, че P_1 завършва всеки път, когато е в сила A.

Да фиксираме точките 1, 2 и 3, както е показано на фиг. 1, и да положим $B(x, y, z, i) : z = i.y$. Ще докажем, че B е инвариант на P_1. Наистина, нека за входа (x, y) е изпълнено условието A, т.е. x и y са естествени числа.

(i) За началните стойности z и i на работните променливи z и i имаме $\mathrm{z} = \mathrm{i} = 0$ и следователно $\mathrm{z} = \mathrm{i}.\mathrm{y}$, т.е. условието B е



Фиг. 1, 2

изпълнено при първо попадане в точка 2.

(ii) Нека при някое попадане в точка 2 с текущи стойности на променливите z и i — съответно z и i, е изпълнено условието B, т.е. $z = i.y$. Да предположим, че отново сме попаднали в точката 2 и z' и i' са новите стойности на променливите z и i. Тогава $z' = z + y$, $i' = i + 1$ и от предположението $z = i.y$ получаваме последователно

$$z' = z + y = i.y + y = (i + 1).y = i'.y.$$

Това означава, че B е в сила и за новите стойности на z и i.

(iii) Нека в точка 2 е в сила $B(x, y, z, i)$ и освен това $i = x$, т.е. изпълнено е условието от оператора за проверка. Тогава от $z = i.y$ и $i = x$ ще имаме $z = x.y$, и следователно в точка 3 е изпълнено изходното условие C.

За да покажем, че програмата P_1 завършва при (x, y), да означим с i_n, $n \geqq 1$, съдържанието на променливата i при n-тото попадане в точка 2 (ако такова има). С индукция по броя на преминаванията през точка 2 показваме, че за всяко $n \leqq x + 1$ стойността i_n е определена и $i_n = n - 1$. Тогава при $n = x + 1$ ще имаме $i_n = x$ и следователно програмата P_1 ще завърши.

Задача 2. Да се докаже, че програмата P_2 (фиг. 2) е:

а) тотално коректна относно входно условие $A : x \in \mathbb{N},\ y \in \mathbb{N}^+$ и изходно условие $C : x = q.y + r \ \&\ 0 \leqq r < y$ (т.е. P_2 намира частното и остатъка при делението на x с y, ако $x, y \in \mathbb{N}$ и $y \neq 0$);

б) частично коректна относно входно условие $A' : x \in \mathbb{N},\ y \in \mathbb{N}$ и изходно условие $C : x = q.y + r \ \&\ 0 \leqq r < y$, но не завършва при условие A'.

Доказателство. а) Да изберем $B(x, y, q, r) : x = q.y + r \ \&\ r \geqq 0$. Ще покажем, че това условие е инвариант на програмата P_2 относно A и C. Отново фиксираме точки 1, 2, и 3, както е означено на фиг. 2, и избираме (x, y), за които $A(x, y)$.

(i) Началните стойности на променливите q и r са съответно 0 и x. Тъй като $A(x, y)$, то $x \geqq 0$ и $x = 0.y + x$.

(ii) Нека при някакво попадане в точка 2 е изпълнено условието $B(x, y, q, r)$, където q и r са текущите стойности на променливите q и r. Да предположим, че (преминавайки през оператора за проверка), отново сме се върнали в точка 2. Тогава $\neg(r < y)$, т.е. $r \geqq y$. Имаме $q' = q + 1$ и $r' = r - y$ за q', r' — новите текущи стойности на q и r. От условието $B(x, y, q, r)$ получаваме

$$x = q.y + r = (q + 1).y + r - y = q'.y + r'.$$

Освен това, тъй като $r \geqq y$, ще е изпълнено $r' = r - y \geqq 0$. Получихме окончателно $B(x, y, q', r')$.

(iii) От $B(x, y, q, r)$ и $r < y$ очевидно следва $C(x, y, q, r)$.

Ще покажем, че програмата P_2 завършва при условие A. Да допуснем, че при някой вход (x, y), за който $x \in \mathbb{N}$, $y \in \mathbb{N}^+$, P_2 не завършва. Да означим с r_n текущата стойност на променливата r при n-тото попадане в точка 2. От допускането, че P_2 не завършва, получаваме че за всяко $n \geqq 1$ стойността r_n е определена и $\neg(r_n < y)$, т.е. $r_n \geqq y$. Така $r_1 = x \in \mathbb{N}$ и $r_{n+1} = r_n - y \in \mathbb{N}$ (индукция по n). От входното условие имаме $y > 0$ и следователно $\{r_n\}_n$ е строго намаляваща редица от естествени числа — противоречие. Следователно допускането, че P_2 не завършва при вход (x, y), е погрешно.

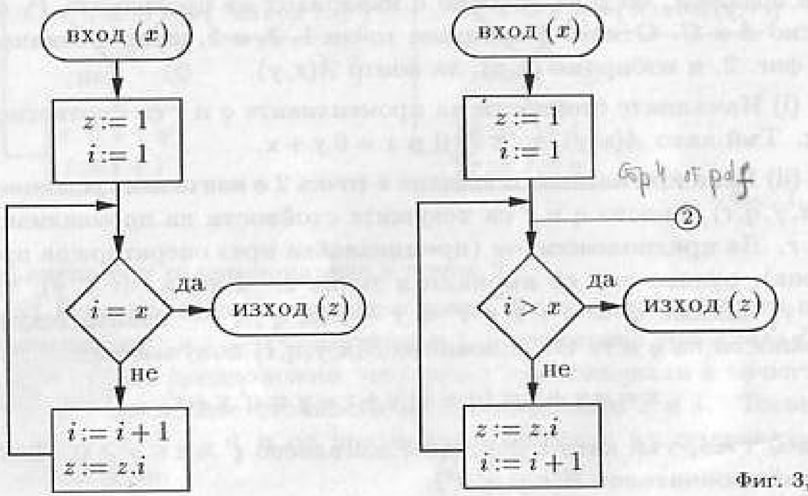
Ясно е, че ако $y = 0$, програмата P_2 зацикля, т.е. не завършва при условие A'.

Задача 3. Да положим $A\colon x \in \mathbb{N}$, $A'\colon x \in \mathbb{N}^+$, $C\colon z = x!$. Да се докаже, че:

а) програмата P_3 (фиг. 3) е частично коректна относно A и C, но не завършва при условие A;

б) програмата P_3 е тотално коректна относно A' и C;

в) програмата P_4 (фиг. 4) е тотално коректна относно A и C.



Фиг. 3, 4

Упътване. в) Нека $\mathrm{x} \in \mathbb{N}$. Не е трудно да се види, че текущите стойности z и i на променливите z и i при всяко преминаване през точка 2 са свързани с равенството $\mathrm{z} = (\mathrm{i}-1)!$. Съобразете още, че $\mathrm{i} \in \mathbb{N}$ и $\mathrm{i} \leqq \mathrm{x}+1$. Следователно вероятният инвариант на цикъла е условието

$$B : z = (i-1)! \ \&\ i \in \mathbb{N} \ \&\ i \leqq x+1.$$

Непосредствено се проверява, че това условие удовлетворява (i) и (ii). За да установим (iii), да забележим, че

$$B(x,z,i) \ \&\ i > x \Longrightarrow x < i \leqq x+1.$$

Освен това $i \in \mathbb{N}$ (от $B(x,z,i)$) и $x \in \mathbb{N}$ (от $A(x)$). Следователно $i = x+1$. Но при $i = x+1$ от равенството $z = (i-1)!$ получаваме $z = x!$, т.е. $C(x,z)$.

Задача 4. Да се докаже, че програмата P_5 (фиг. 5) пресмята функцията $f(n) = [\sqrt{n}]$, $n \in \mathbb{N}$.

Упътване. Трябва да покажем, че P_5 е тотално коректна относно входно условие $A : n \in \mathbb{N}$ и изходно условие $C : x = [\sqrt{n}]$. Лесно се съобразява, че при всяко преминаване през началото на оператора за проверка за текущите стойности x, y, и s на променливите x, y и s са в сила равенствата

$$\mathrm{y} = 2\mathrm{x}+1 \quad \text{и} \quad \mathrm{s} = 1+3+\cdots+(2\mathrm{x}+1) = (\mathrm{x}+1)^2.$$

Следователно инвариантът B на цикъла, който търсим, трябва да включва равенствата $y = 2x+1$ и $s = (x+1)^2$.

От третото условие от определението за инвариант имаме, че за B трябва да е изпълнено още

$$B(n,x,y,s) \ \&\ s > n \Longrightarrow C(n,x).$$

Като използваме определението за цяла част, изходното условие C преобразуваме по следния начин:

$$x = [\sqrt{n}] \iff x \leqq \sqrt{n} < x+1 \iff x^2 \leqq n < (x+1)^2.$$

Верността на второто неравенство $n < (x+1)^2$ от C следва от равенството $s = (x+1)^2$ (което причислихме към B) и условието за излизане от цикъла $s > n$. Другото неравенство $x^2 \leqq n$ от изходното условие C не можем да получим от $s > n$ и това, което до момента сме включили в B (равенствата $y = 2x+1$ и $s = (x+1)^2$). Затова добавяме $x^2 \leqq n$ към B. Така предполагаемият инвариант на цикъла става

$$B(n,x,y,s)\colon \ y = 2x+1 \ \&\ s = (x+1)^2 \ \&\ x^2 \leqq n.$$

Сега проверяваме дали това условие B е в сила при всяко преминаване през точка 2, т.е. дали са изпълнени изискванията (i) и (ii) от определението за инвариант.

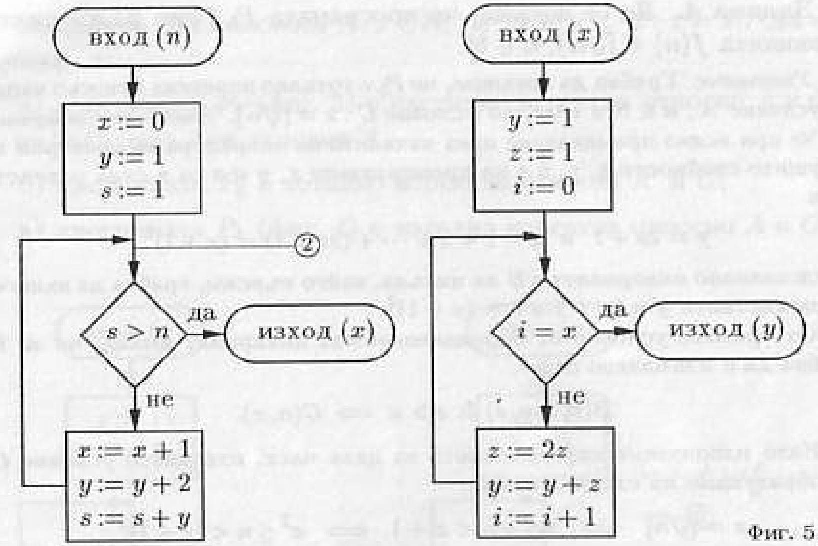
Условието (i) се свежда до проверката на $B(n,0,1,1)$, което е вярно.

За да установим (ii), трябва да проверим, че

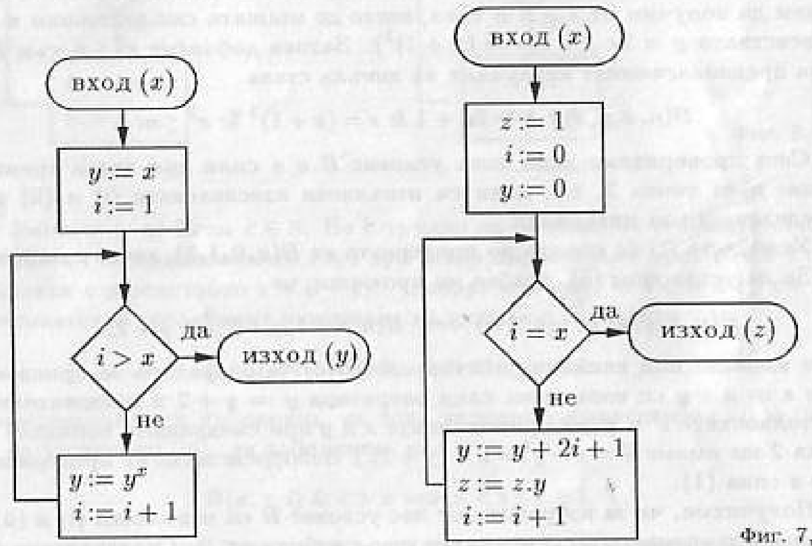
(1) $$B(n,x,y,s) \ \&\ \neg(s > n) \Longrightarrow B(n,x+1,y+2,s+y+2).$$

(Тук вземаме под внимание обстоятелството, че оператора за присвояване $s := s+y$ се изпълнява след оператора $y := y+2$ и следователно за стойностите s' и y' на променливите s и y при следващото попадане в точка 2 ще имаме $\mathrm{s}' = \mathrm{s}+\mathrm{y}' = \mathrm{s}+(\mathrm{y}+2)$.) Непосредствено се проверява, че е в сила (1).

Получихме, че за избраното от нас условие B са изпълнени (i) и (ii). Последното условие (iii) осигурихме още с избора на B. Следователно B е инвариант на програмата P_5.



Фиг. 5, 6



Фиг. 7, 8

Задача 5. Да се намери функцията $f_i : \mathbb{N} \to \mathbb{N}$, $i = 1, 2, 3$, която се пресмята съответно с всяка от програмите P_6 (фиг. 6), P_7 (фиг. 7), P_8 (фиг. 8).

Забележка. Тук имаме предвид да се намери функцията f и да се докаже, че съответната програма я пресмята.

Отговор. $f_1(x) = 2^{x+1} - 1$; $f_2(x) = x^{x^x}$, като $0^0 = 1$; $f_3(x) = (x!)^2$.

Задача 6. Да се докаже, че програмата P_9 (фиг. 9) е тотално коректна относно входно условие $A : n, k \in \mathbb{N} \,\&\, n \geq k$ и изходно условие $C : b = \binom{n}{k}$ (предполагаме, че $\binom{0}{0} = 1$).

Задача 7. Да се докаже, че:

а) програмата P_{10} (фиг. 10) е тотално коректна относно входно условие $A : n \in \mathbb{N} \,\&\, x \in \mathbb{R} \,\&\, x > 1$ и изходно условие $C : y = \dfrac{x^{n+1} - 1}{x - 1}$;

б) програмата P_{11} (фиг. 11) е тотално коректна относно входно условие $A : x \in \mathbb{N} \,\&\, y \in \mathbb{N}$ и изходно условие $C : z = \text{НОД}(x, y)$ $(\text{НОД}(0, 0) = 0)$;

в) програмата P_{12} (фиг. 12) е тотално коректна относно входно условие $A : x \in \mathbb{N}^+$ и изходно условие $C : y = (1 + \cdots + x)^2$.

Задача 8. Да се докаже, че всяка от следващите програми е тотално коректна относно съответните входни и изходни условия:

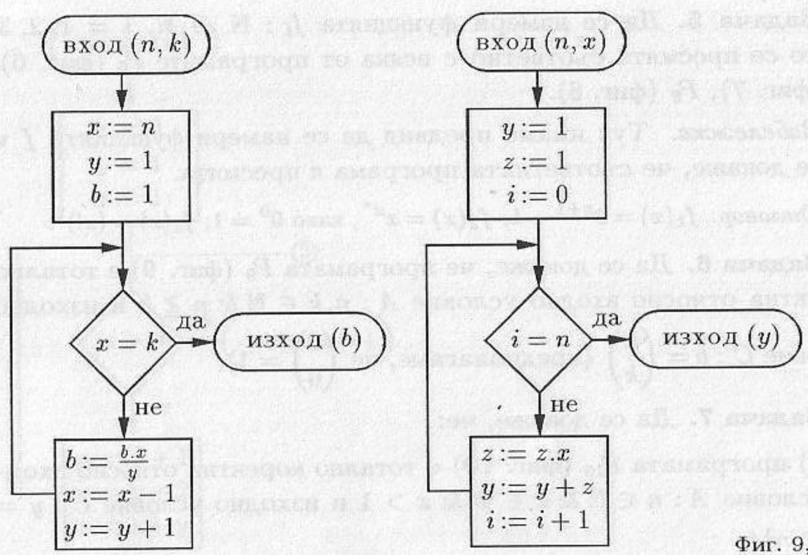
а) програмата P_{13} (фиг. 13) и условията $A : x \in \mathbb{N}^+$ и $C : y = x^2$;

б) програмата P_{14} (фиг. 14) и условията $A : x \in \mathbb{R} \,\&\, x \geqq 0$ и $C : z = [x/2]$;

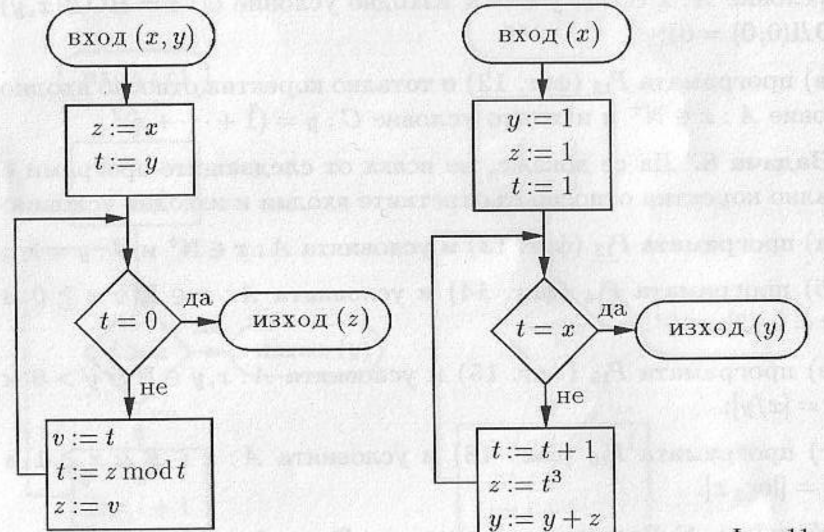
в) програмата P_{15} (фиг. 15) и условията $A : x, y \in \mathbb{N} \,\&\, y > 0$ и $C : t = [x/y]$;

г) програмата P_{16} (фиг. 16) и условията $A : x \in \mathbb{R} \,\&\, x \geqq 1$ и $C : z = [\log_2 x]$.

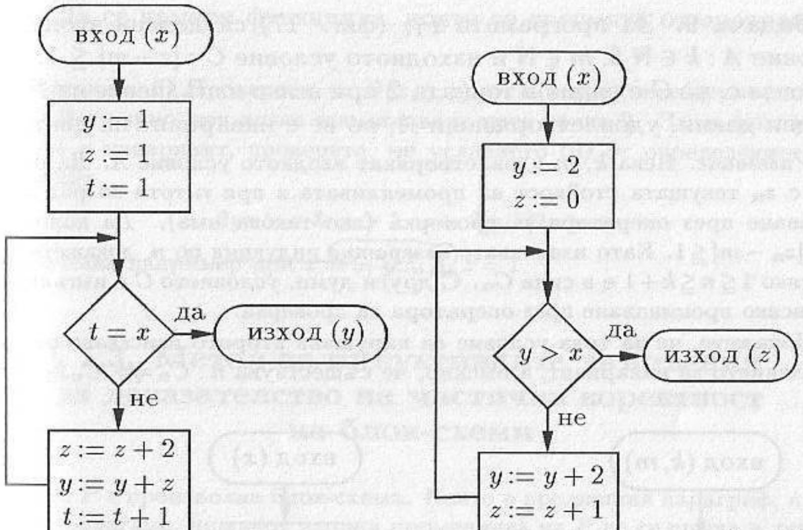
Упътване. б) Проверете, че условието $B : y = 2z + 2 \,\&\, 2z \leqq x$ е инвариант на P_{14}.



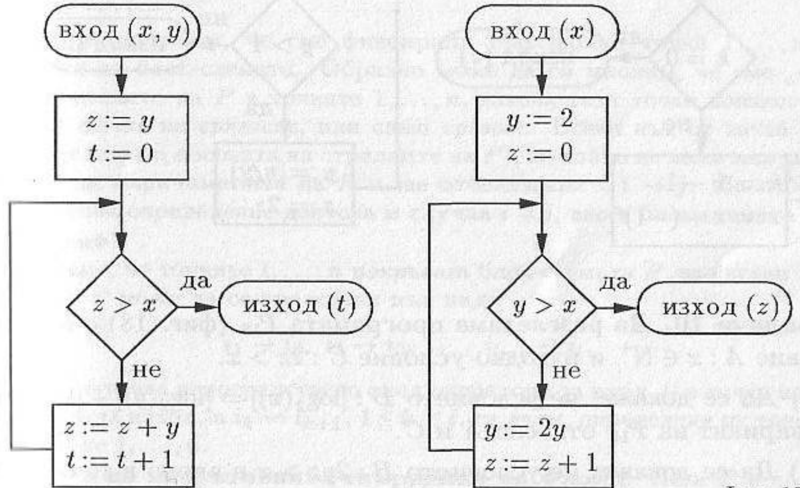
Фиг. 9, 10



Фиг. 11, 12



Фиг. 13, 14

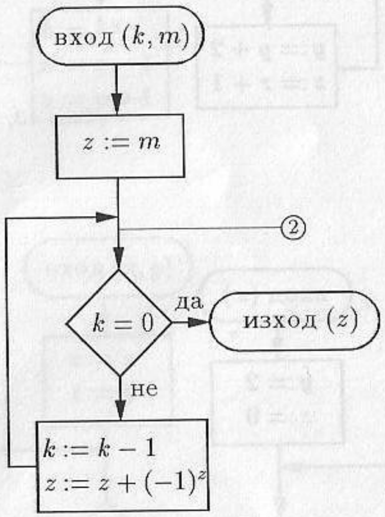


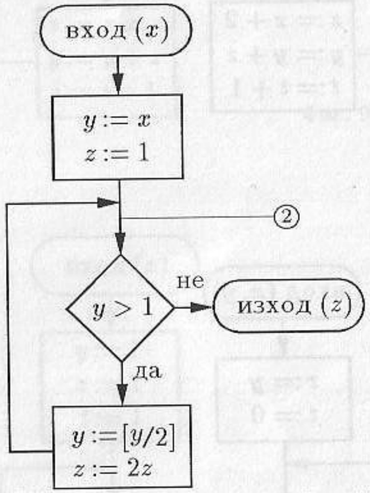
Фиг. 15, 16

Задача 9. За програмата P_{17} (фиг. 17) са дадени входното условие $A: k \in \mathbb{N} \ \& \ m \in \mathbb{N}$ и изходното условие $C: |z - m| \leqq 1$. Да се докаже, че C е вярно в точката 2 при всяко изпълнение на P_{17} с входни данни, удовлетворяващи A, но не е инвариант на цикъла.

Упътване. Нека k, m удовлетворяват входното условие A. Да означим с z_n текущата стойност на променливата z при n-тото по ред преминаване през оператора за проверка (ако такова има). Да положим $C_n: |z_n - m| \leqq 1$. Като използвате *възвратна* индукция по n, докажете, че за всяко $1 \leqq n \leqq k+1$ е в сила C_n. С други думи, условието C е изпълнено при всяко преминаване през оператора за проверка.

Покажете, че за това условие се нарушава второто изискване от определението за инвариант, а именно, че съществува $n: C_n \not\Rightarrow C_{n+1}$





Фиг. 17, 18

Задача 10. Да разгледаме програмата P_{18} (фиг. 18) с входно условие $A: x \in \mathbb{N}^+$ и изходно условие $C: 2z > x$.

а) Да се докаже, че условието $D: [\log_2(x)] = [\log_2(y)] + [\log_2(z)]$ е инвариант на P_{18} относно A и C.

б) Да се докаже, че условието $B: 2yz > x$ е вярно в точката 2 при всяко изпълнение на програмата с входни данни, удовлетворяващи A, но не е инвариант на P_{18} относно A и C.

в) Да се намери функцията, която се пресмята от програмата P_{18}.

Упътване. б) Проверете, че $D(x, y, z) \Rightarrow B(x, y, z)$, откъдето ще следва, че B е вярно при всяко преминаване през точка 2. За да покажете, че B не е инвариант, проверете, че условието (ii) от определението за инвариант

$$2yz > x \ \& \ y > 1 \implies 2[y/2]2z > x$$

се нарушава например при $x = 5$, $y = 3$, $z = 1$.

§ 1.3. Метод на индуктивните твърдения за доказателство на частична коректност на блок-схеми

Нека P е произволна блок-схема. Както в предишния параграф, и тук ще предполагаме, че никоя входна променлива на P не се среща в лявата част на оператор за присвояване.

На блок-схемата P можем да гледаме като на ориентиран граф с върхове — ромбовете, правоъгълниците и овалите на схемата и ребра — стрелките на схемата. Всеки краен път в този граф, започващ с оператора за вход и завършващ с оператора за изход, ще наричаме *траектория* в P.

Да предположим, че сме фиксирали произволни точки $1, \ldots, n$ от стрелките на блок-схемата. Образно може да си мислим, че сме „срязали" стрелките на P в точките $1, \ldots, n$, затова тези точки понякога се наричат *точки на срязване*, или само *срезове*. Всеки път от точка i до точка j (следващ посоката на стрелките на P), на който не лежи междинна точка l, ще наричаме *дъга* на P и ще отбелязваме с $i \to j$. Да отбележим, че това определение допуска и случая $i = j$, ако в блок-схемата има цикли.

Казваме, че точките $1, \ldots, n$ *покриват* блок-схемата P, ако всяка траектория в P може да се представи във вида

$$i_1 \to i_2, \ i_2 \to i_3, \ \ldots, \ i_{l-1} \to i_l,$$

където i_1 е точка непосредствено след оператора за вход, i_l е точка преди оператора за изход, а $i_k \to i_{k+1}$, $1 \leqq k < l$, са дъги, определени от точките на срязване $1, \ldots, n$.

Метод на индуктивните твърдения на Флойд. Нека P е произволна блок-схема, а A и C са съответно входно и изходно условие за P. Да предположим, че сме направили следното:

I. Избрали сме точки на срязване $1, \dots, n$ от стрелките на блок-схемата, които покриват P.

II. С всяка точка i, $1 \leqq i \leqq n$, сме „свързали“ условие A_i, определящо някаква зависимост между текущите стойности на променливите при преминаване през тази точка. При това, с началната точка сме свързали входното условие A, а с точките преди операторите за изход — изходното условие C. Условията $A_1, \dots, A_n$ сме избрали така, че за всяка дъга $i \to j$ да е изпълнено:

(*) ако в точката i е в сила A_i и (преминавайки по дъгата $i \to j$), попаднем в точката j, то там е в сила A_j.

Твърдим, че в такъв случай P е частично коректна относно A и C.

Доказателство. Да предположим, че при входни данни $\overline{x}$, за които $A(\overline{x})$, програмата P е завършила с резултат $\overline{y}$. Това изпълнение е определило някаква траектория, която според избора на точките на срязване може да се представи като

$$i_1 \to i_2, \dots, i_{l-1} \to i_l,$$

където $i_1 = 1$ е началната точка, а i_l е точка преди оператора за изход.

С индукция по $k = 1, \dots, l$ ще покажем, че при всяко преминаване през точка i_k е в сила условието A_{i_k}, свързано с нея. Наистина, за началната точка i_1, с която е свързано входното условие A, това твърдение е вярно по предположение, а преходът от k към $k+1$ се основава на факта, че за дъгата $i_k \to i_{k+1}$ е изпълнено условието (*).

Тогава при $k = l$ ще имаме, че в точката i_l (т.е. при завършване на изпълнението на програмата) ще е в сила условието $A_{i_l} = C$. Но в точката i_l стойностите на входните и изходните променливи на P са съответно $\overline{x}$ и $\overline{y}$ (съобразете защо); следователно $C(\overline{x}, \overline{y})$.

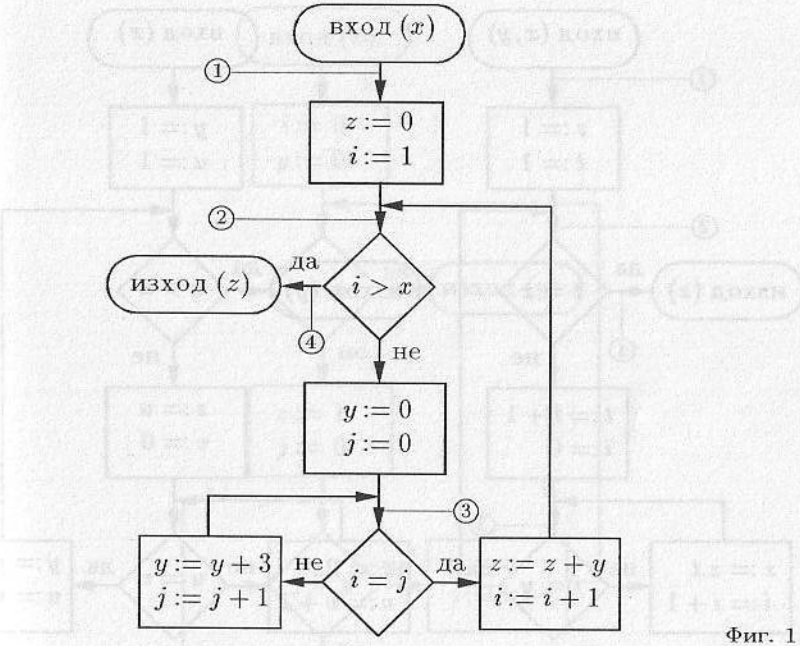
Забележка. Точките на срязване $1, \dots, n$, покриващи P, можем да избираме по различни начини. Препоръчително е тези точки да са помалко, за да са по-малко и проверките на верифициращите условия (*).

Условията $A_1, \dots, A_n$ обикновено се наричат *индуктивни твърдения*, откъдето идва и името на метода.

Да *верифицираме* дъгата $i \to j$ означава да проверим, че за нея е изпълнено условието (*).

Задача 1. Да се докаже, че програмата P_1 (фиг. 1) е тотално коректна относно входно условие $A : x \in \mathbb{N}$ и изходно условие $C : z = \dfrac{3x(x+1)}{2}$.

Упътване. Лесно се съобразява, че точките на срязване 1, 2, 3, 4, означени на фиг. 1, са достатъчни, за да покрият P_1. Нека с т. 2 свържем условието B, а с т. 3 — условието B', където



Фиг. 1

$$B(x, z, i) : z = \frac{3i(i-1)}{2} \;\&\; i \leqq x + 1; \quad B'(x, z, i, y, j) : z = \frac{3i(i-1)}{2} \;\&\; i \leqq x \;\&\; y = 3j.$$

Сега верифицирането на дъгите, получени при срязването, което избрахме, се свежда до проверката на следните импликации:

за дъгата $1 \to 2$: $A(x, y) \Longrightarrow B(x, 0, 1)$;

за дъгата $2 \to 3$: $B(x, z, i) \;\&\; \neg(i > x) \Longrightarrow B'(x, z, i, 0, 0)$;

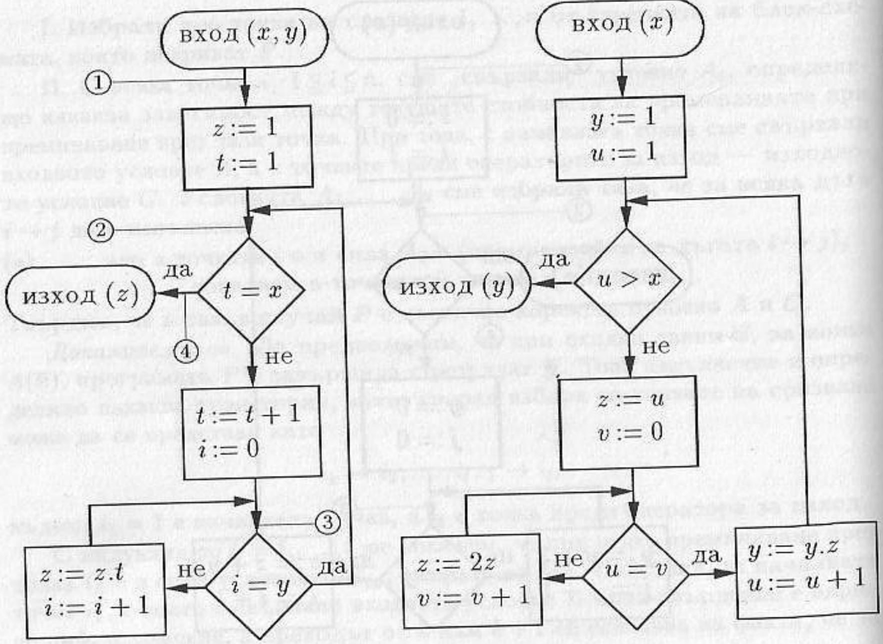
за дъгата $3 \to 3$: $B'(x, z, i, y, j) \;\&\; i = j \Longrightarrow B'(x, z, i, y + 3, j + 1)$;

за дъгата $3 \to 2$: $B'(x, z, i, y, j) \;\&\; i = j \Longrightarrow B(x, z + y, i + 1)$;

за дъгата $2 \to 4$: $B(x, z, i) \;\&\; i > x \Longrightarrow C(x, z)$.

Непосредствено се проверява, че всяка от тези импликации е вярна. Следователно P_1 е частично коректна относно A и C.

Да допуснем, че за някое $x \in \mathbb{N}$ програмата P_1 не завършва. Съобразете, че това не се дължи на зацикляне във вътрешния цикъл. Оттук ще следва, че за всяко $n = 1, 2, \dots$ преминаваме през точката 2 от външния цикъл. Така получаваме редицата $i_1 < i_2 < \dots$ от текущите стойности на променливата i при n-тото попадане в точката 2, като при това $i_n \in \mathbb{N}$ и $i_n \leqq x$ — противоречие.



Фиг. 2, 3

Задача 2. Да се докаже, че програмата P_2 (фиг. 2) е:

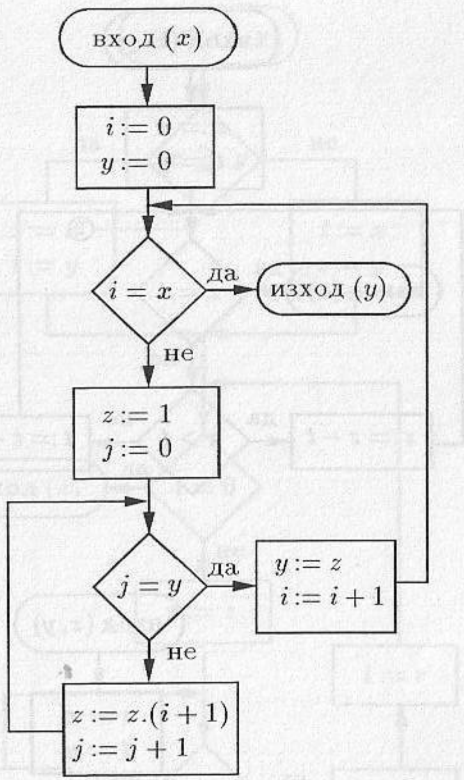
а) частично коректна относно входно условие $A : x, y \in \mathbb{N}$ и изходно условие $C : z = (x!)^y$, но не завършва при условие A;

б) тотално коректна относно входно условие $A' : x \in \mathbb{N}^+ \,\&\, , y \in \mathbb{N}$ и изходно условие $C : z = (x!)^y$.

Упътване. За да се досетим как да изберем условието B в точка 2, използваме, че за него трябва да е вярно верифициращото условие за дъгата $2 \to 4$: $B(x, y, z, t) \,\&\, x = t \Longrightarrow z = (x!)^y$.

Следователно B трябва да е (или да включва) равенството $z = (t!)^y$. Тогава при движение по дъгата $2 \to 3$ в точката 3 ще е изпълнено $z = ((t-1)!)^y$. Оттук лесно се съобразява, че условието B', което трябва да свържем с точката 3, е $B'(x, y, z, t, i) : z = ((t-1)!)^y.t^i$.

Задача 3. Да се докаже, че програмата P_3 (фиг. 3) е тотално коректна относно входно условие $A : x \in \mathbb{N}$ и изходно условие $C : y = x!2^{\frac{x(x+1)}{2}}$.



Фиг. 4

Задача 4. Да се намери функцията $f : \mathbb{N} \to \mathbb{N}$, която се пресмята от програмата P_4 (фиг. 4).

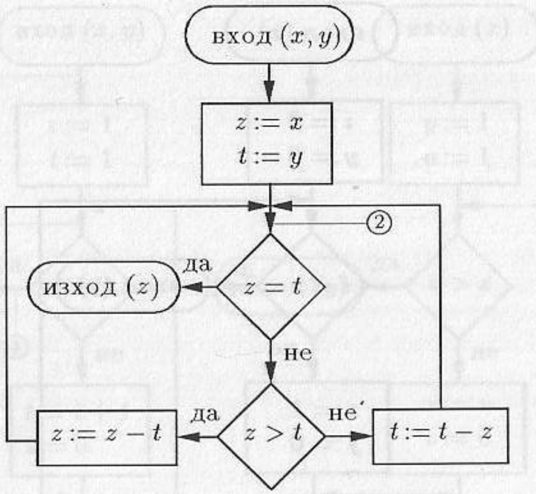
Задача 5. Нека $A : x, y \in \mathbb{N}$, $A' : x, y \in \mathbb{N}^+$, $C : z = \text{НОД}(x, y)$. Да се докаже, че:

а) програмата P_5 (фиг. 5) е частично коректна относно A и C, но не завършва при условие A; 2013/23 →

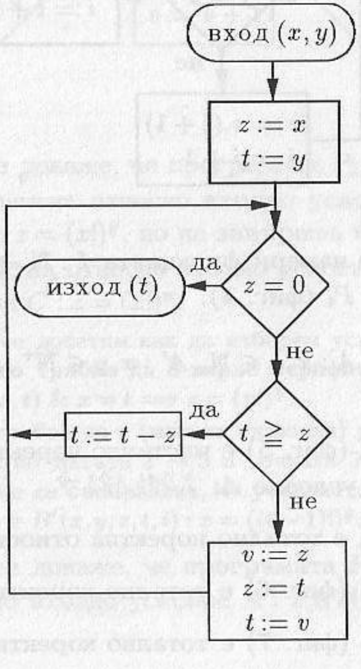
б) програмата P_5 е тотално коректна относно A' и C;

в) програмата P_6 (фиг. 6) е тотално коректна относно A и C;

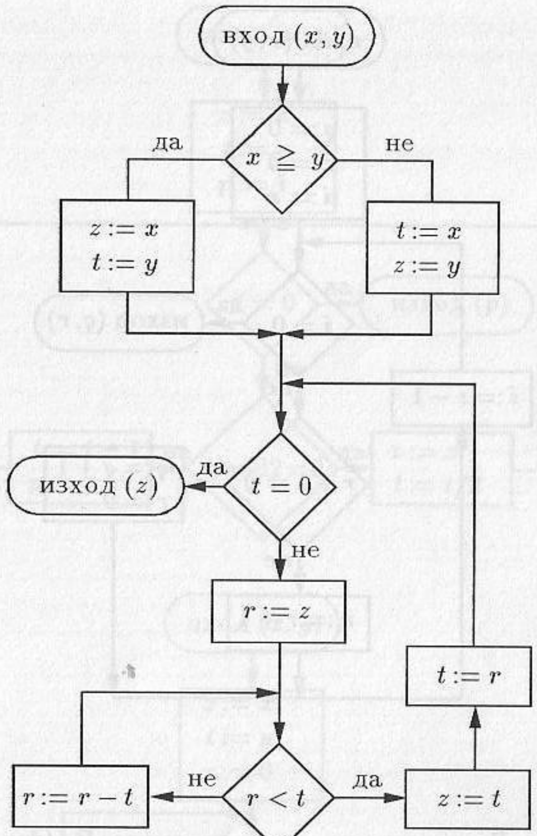
г) програмата P_7 (фиг. 7) е тотално коректна относно A и C.



Фиг. 5



Фиг. 6



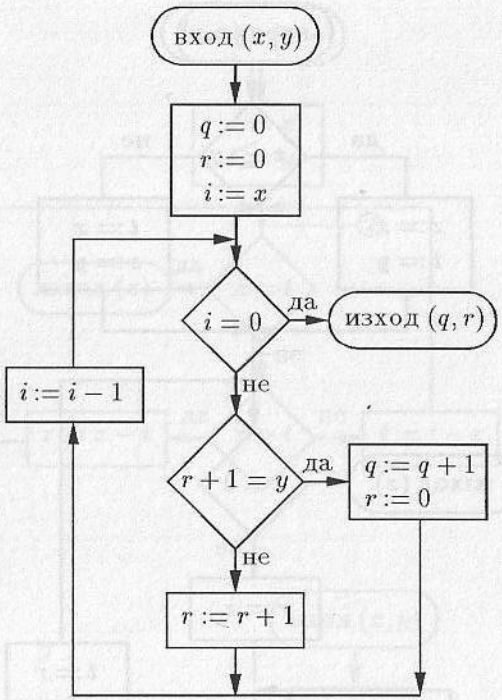
Фиг. 7

Упътване. Да допуснем, че при някой вход (x, y), удовлетворяващ A, програмата P_5 не завършва. Да означим със z_n и t_n текущите стойности на z и t при n-тото преминаване през точка 2. Покажете, че за всяко $n = 1, 2 \ldots$

1) $z_n, t_n \in \mathbb{N}$;

2) $(z_n, t_n) \succ (z_{n+1}, t_{n+1})$.

Така ще получите, че редицата $\{(z_n, t_n)\}_n$ е безкрайно намаляваща относно лексикографската наредба в $\mathbb{N}^2$, което е противоречие с факта, че тя е фундирана (зад. 9, § 1.1).



Фиг. 8

Задача 6. Да се докаже, че програмата P_8 (фиг. 8) намира частното и остатъка от деление на x с y, ако $x \in \mathbb{N}$ и $y \in \mathbb{N}^+$.

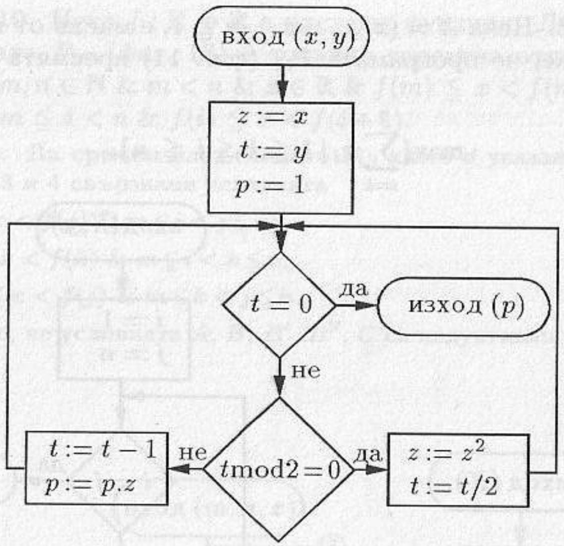
Задача 7. Да се докаже тотална коректност на:

а) бързия алгоритъм за степенуване (фиг. 9);

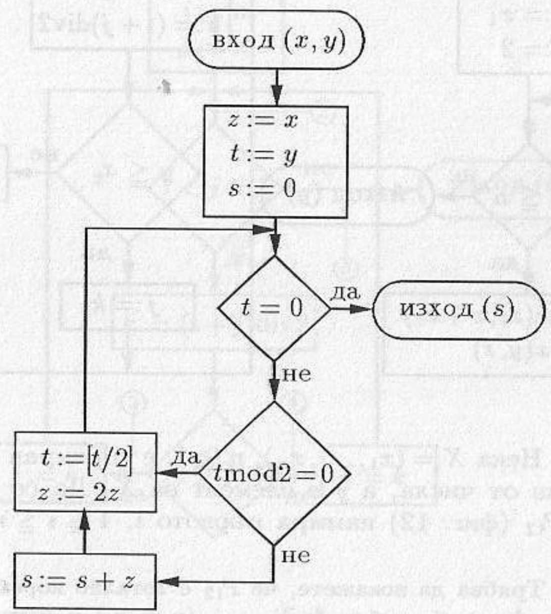
б) бързия алгоритъм за умножение (фиг. 10).

Упътване. а) Използвайте следното свойство на операцията степенуване:

$$a^b = \begin{cases} (a^2)^{b/2}, & \text{ако } b \text{ е четно,} \\ a.a^{b-1}, & \text{ако } b \text{ е нечетно.} \end{cases}$$



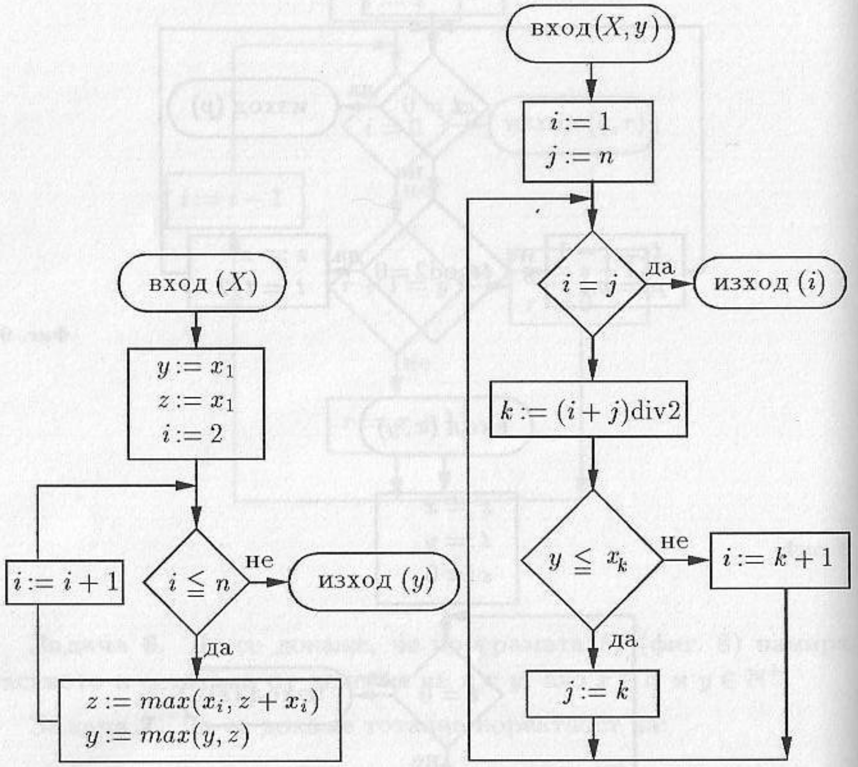
Фиг. 9



Фиг. 10

Задача 8. Нека $X = (x_1, \dots, x_n)$, $n \geqq 1$, е масив от цели числа. Да се докаже, че програмата P_{11} (фиг. 11) пресмята стойността на израза

$$\max\{\sum_{i=k}^{l} x_i \,|\, 1 \leqq k \leqq l \leqq n\}.$$



Фиг. 11, 12

Задача 9. Нека $X = (x_1, \dots, x_n)$, $n \geqq 1$, е сортиран във възходящ ред масив от числа, а y е елемент на X. Да се докаже, че програмата P_{12} (фиг. 12) намира първото i, $1 \leqq i \leqq n$, за което $y = x_i$.

Упътване. Трябва да покажете, че P_{12} е тотално коректна относно входно условие $A : x_1 \leqq \dots \leqq x_n \;\&\; \exists i_{1 \leqq i \leqq n}(y = x_i)$ и изходно условие $C : y = x_i \;\&\; (i > 1 \Longrightarrow x_{i-1} < x_i)$.

Задача 10. Нека $f : \mathbb{R} \to \mathbb{R}$ е растяща функция. Да се докаже, че програмата P_{13} (фиг. 13) е тотално коректна относно входно условие $A : m, n \in \mathbb{N} \;\&\; m < n \;\&\; x \in \mathbb{R} \;\&\; f(m) \leqq x < f(n)$ и изходно условие $C : m \leqq i < n \;\&\; f(i) \leqq x < f(i+1)$.

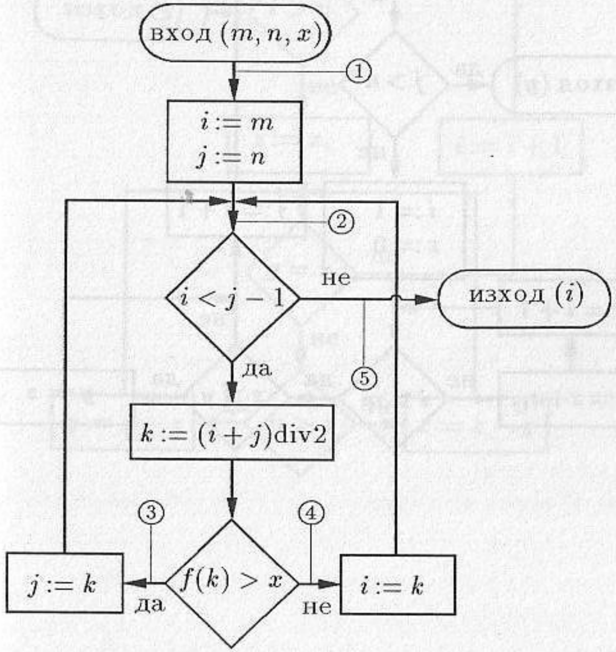
Упътване. Да срежем блок-схемата P_{13} както е указано на фиг. 13. С точките 2, 3 и 4 свързваме условията

B: $f(i) \leqq x < f(j) \;\&\; m \leqq i < j \leqq n$;

B': $f(i) \leqq x < f(k) \;\&\; m \leqq i < k \leqq n$;

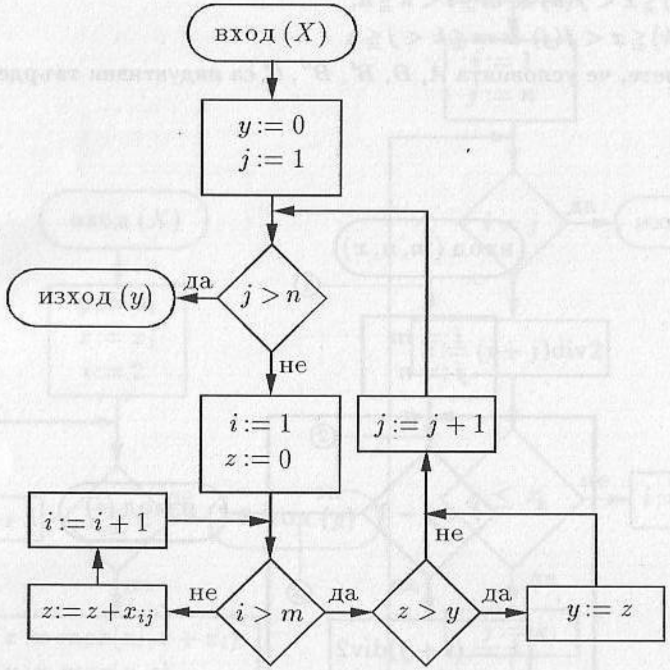
B'': $f(k) \leqq x < f(j) \;\&\; m \leqq k < j \leqq n$.

Проверете, че условията A, B, B', B'', C са индуктивни твърдения за P_{13}.



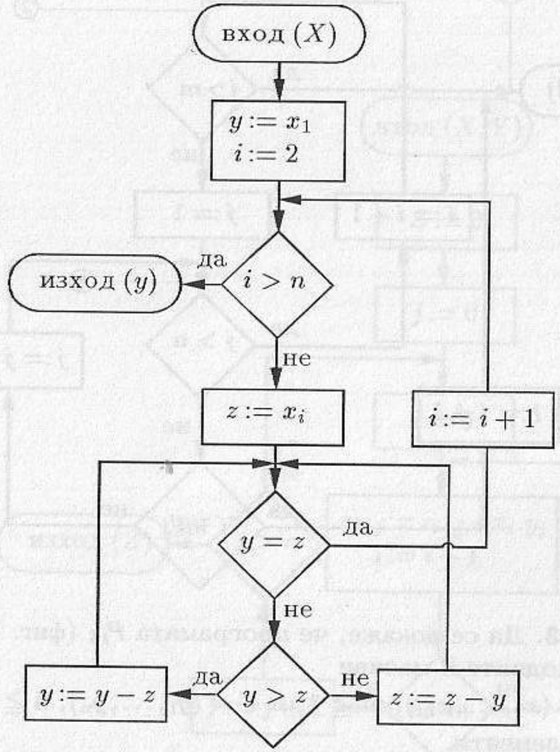
Фиг. 13

Задача 11. Нека $X = (x_{i,j})$, $1 \leqq i \leqq m$, $1 \leqq j \leqq n$, е масив от неотрицателни числа. Да се докаже, че програмата P_{14} (фиг. 14) намира стойността на израза $\max\{\sum_{l=1}^{m} x_{l,k} \mid 1 \leqq k \leqq n\}$.

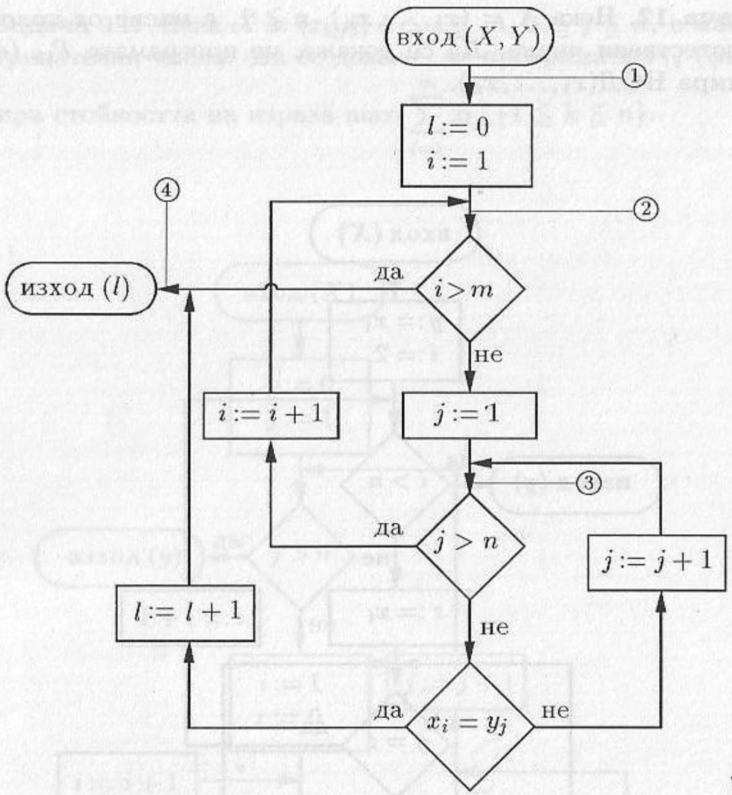


Фиг. 14

Задача 12. Нека $X = (x_1, \ldots, x_n)$, $n \geqq 1$, е масив от положителни естествени числа. Да се докаже, че програмата P_{15} (фиг. 15) намира НОД$(x_1, \ldots, x_n)$.



Фиг. 15



Фиг. 16

Задача 13. Да се докаже, че програмата P_{16} (фиг. 16) разпознава дали входните ѝ масиви

$$X = (x_1, \ldots, x_m),\ m \geqq 1, \text{ и } Y = (y_1, \ldots, y_n),\ n \geqq 1,$$

имат общи елементи.

Упътване. С всяка от точките $i = 1, 2, 3, 4$ свързваме условие A_i, където

A_1: $x_1 = x_1$;

A_2: $\{x_1, \ldots, x_{i-1}\} \bigcap \{y_1, \ldots, y_n\} = \varnothing \ \&\ i \leqq m+1 \ \&\ i \in \mathbb{N}$;

A_3: $\{x_1, \ldots, x_{i-1}\} \bigcap \{y_1, \ldots, y_n\} = \varnothing \ \&$

$\{x_i\} \bigcap \{y_1, \ldots, y_{j-1}\} = \varnothing \ \&\ i \leqq m,\ j \leqq n+1 \ \&\ i, j \in \mathbb{N}$;

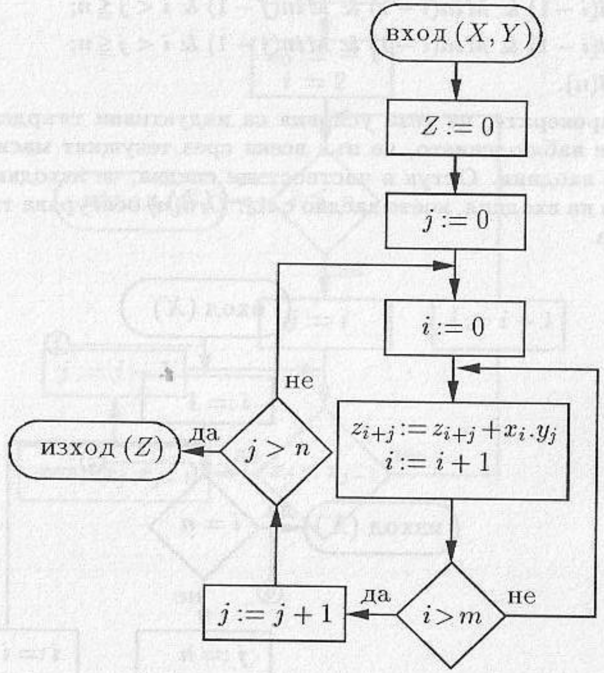
A_4: $(l = 0 \ \&\ \{x_1, \ldots, x_m\} \bigcap \{y_1, \ldots, y_n\} = \varnothing) \vee$

$(l = 1 \ \&\ \{x_1, \ldots, x_m\} \bigcap \{y_1, \ldots, y_n\} \neq \varnothing)$.

Проверете, че A_1, A_2, A_3, A_4 са индуктивни твърдения за P_{16}. Съобразете още, че тя завършва при всеки вход.

Задача 14. Нека $X = (x_0, \ldots, x_m)$, $m \geqq 0$, и $Y = (y_0, \ldots, y_n)$, $n \geqq 0$, са масиви от числа. Да се докаже, че i-тият елемент на изходния масив $Z = (z_0, \ldots, z_{m+n})$ на програмата P_{17} (фиг. 17) е коефициентът пред t^i в нормалния вид на полинома

$$P(t) = (x_0 + x_1 t + \cdots + x_m t^m)(y_0 + y_1 t + \cdots + y_n t^n).$$



Фиг. 17

Задача 15. Да се докаже, че програмата P_{18} (фиг. 18) сортира във възходящ ред входния масив $X = (x_1, \ldots, x_n)$, $n \geqq 1$.

Упътване. Нека $Ord(i)$ е условието $x_1 \leqq \cdots \leqq x_i$, а $Min(i)$ — условието $x_i = \min\{x_i, \ldots, x_n\}$ (приемаме за удобство, че $Ord(0)$ и $Min(0)$ са в сила). С всеки срез i, $i = 1, \ldots, 6$, да свържем следните условия:

A_1: $Ord(0)$;

A_2: $Ord(i-1)$ & $Min(i-1)$ & $i \leqq n$;

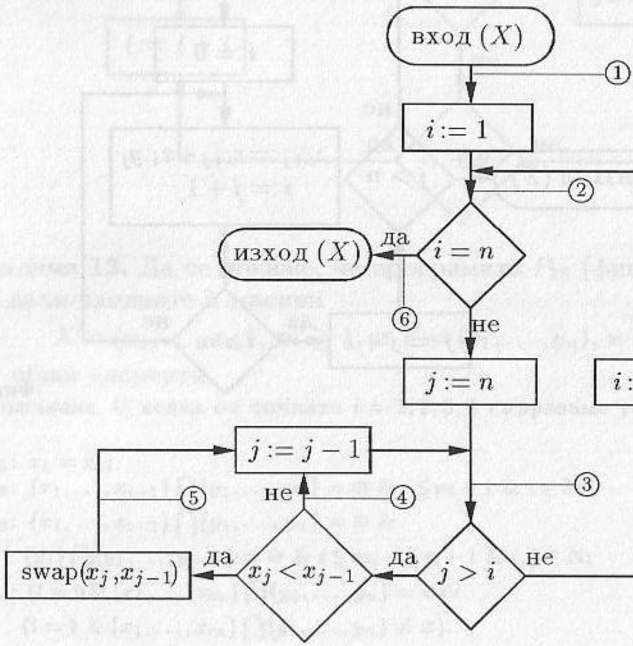
A_3: $Ord(i-1)$ & $Min(i-1)$ & $Min(j)$ & $i \leqq j \leqq n$;

A_4: $Ord(i-1)$ & $Min(i-1)$ & $Min(j-1)$ & $i < j \leqq n$;

A_5: $Ord(i-1)$ & $Min(i-1)$ & $Min(j-1)$ & $i < j \leqq n$;

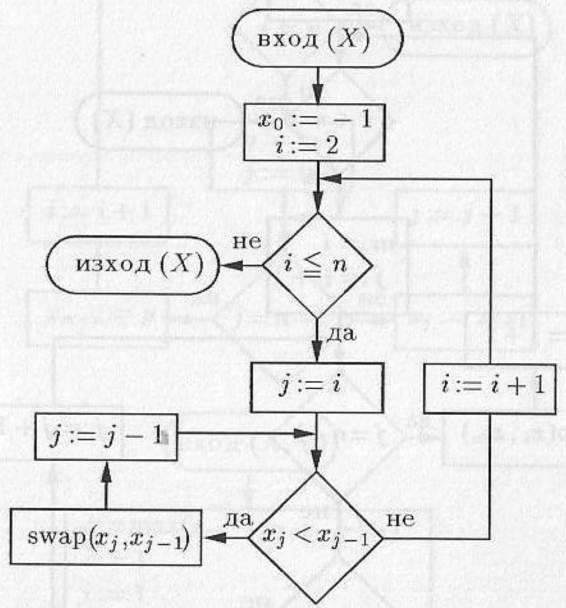
A_6: $Ord(n)$.

За да проверите, че тези условия са индуктивни твърдения за P_{18}, използвайте наблюдението, че във всеки срез текущият масив X е пермутация на входния. Оттук в частност ще следва, че изходният масив е пермутация на входния, което заедно с A_6: $Ord(n)$ осигурява твърдението от задачата.



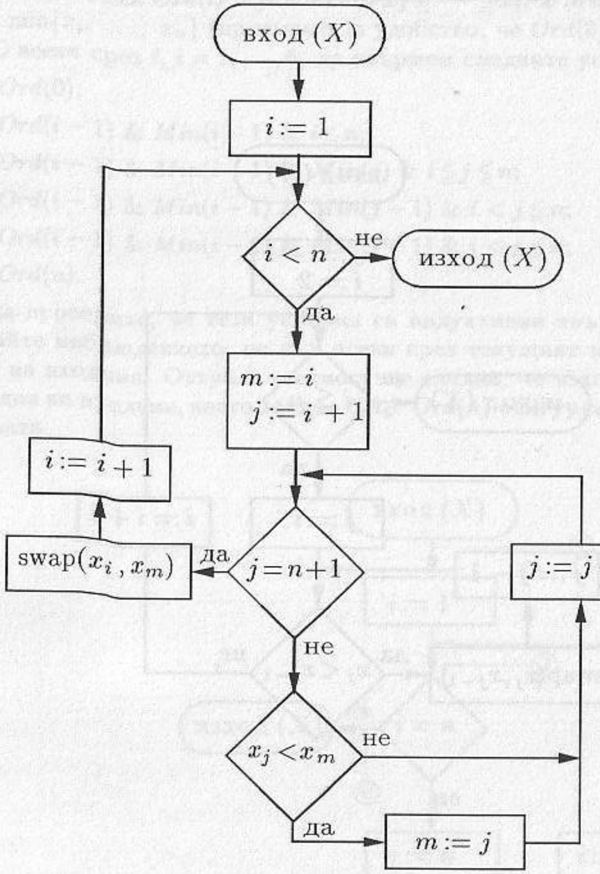
Фиг. 18

Задача 16. Нека $X = (x_1, \ldots, x_n)$, $n \geqq 1$, е масив от неотрицателни числа. Да се докаже, че програмата P_{19} (фиг. 19) сортира елементите на X във възходящ ред.



Фиг. 19

Задача 17. Да се докаже, че програмата P_{20} (фиг. 20) сортира във възходящ ред входния масив $X = (x_1, \dots, x_n)$, $n \geqq 1$.

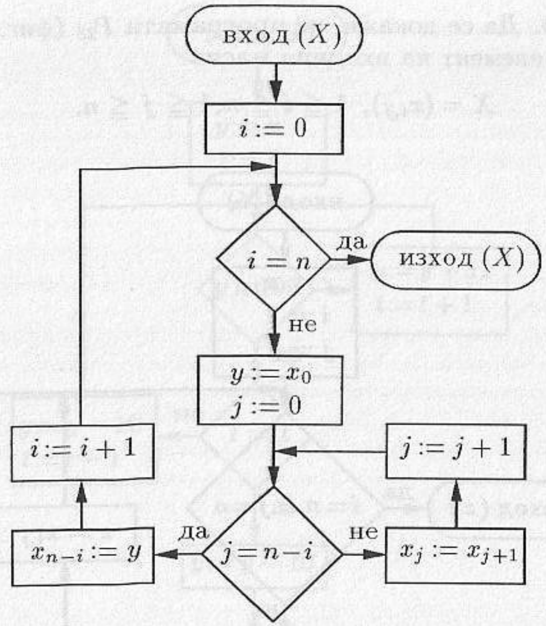


Фиг. 20

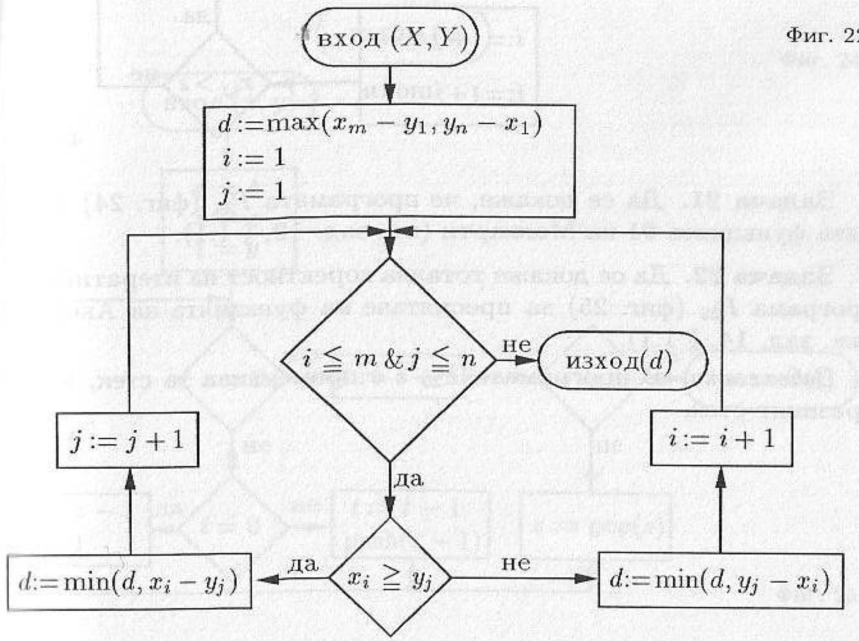
Задача 18. Да се докаже, че програмата P_{21} (фиг. 21) обръща входния масив $X = (x_0, \dots, x_n)$, $n \geqq 0$.

Задача 19. Нека $X = (x_1, \dots, x_m)$, $m \geqq 1$, и $Y = (y_1, \dots, y_n)$, $n \geqq 1$, са сортирани във възходящ ред масиви от числа. Да се докаже, че програмата P_{22} (фиг. 22) намира разстоянието между X и Y, т.е. пресмята стойността на израза

$$\min\{|x_i - y_j| : 1 \leqq i \leqq m \,\&\, 1 \leqq j \leqq n\}.$$



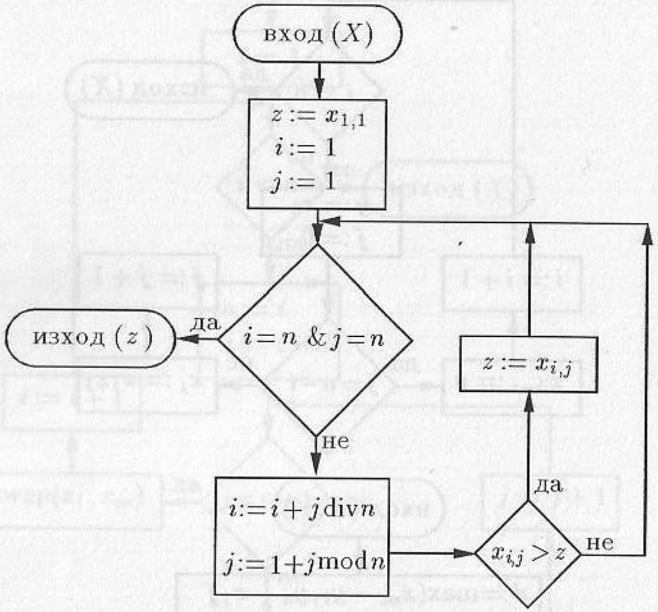
Фиг. 21



Фиг. 22

Задача 20. Да се докаже, че програмата P_{23} (фиг. 23) намира максималния елемент на входния масив

$$X = (x_{i,j}),\ 1 \leqq i \leqq n, 1 \leqq j \leqq n.$$

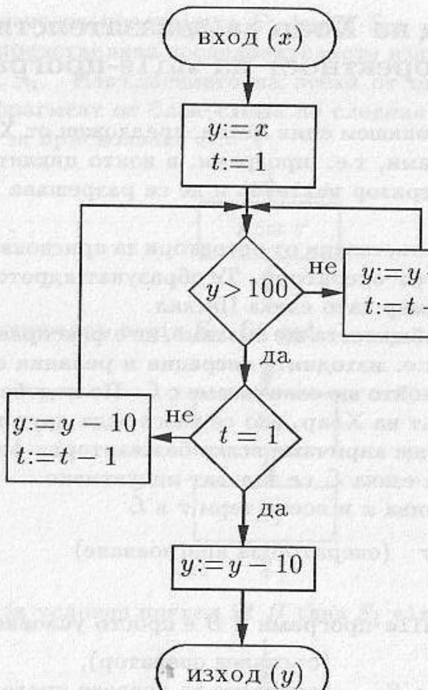


Фиг. 23

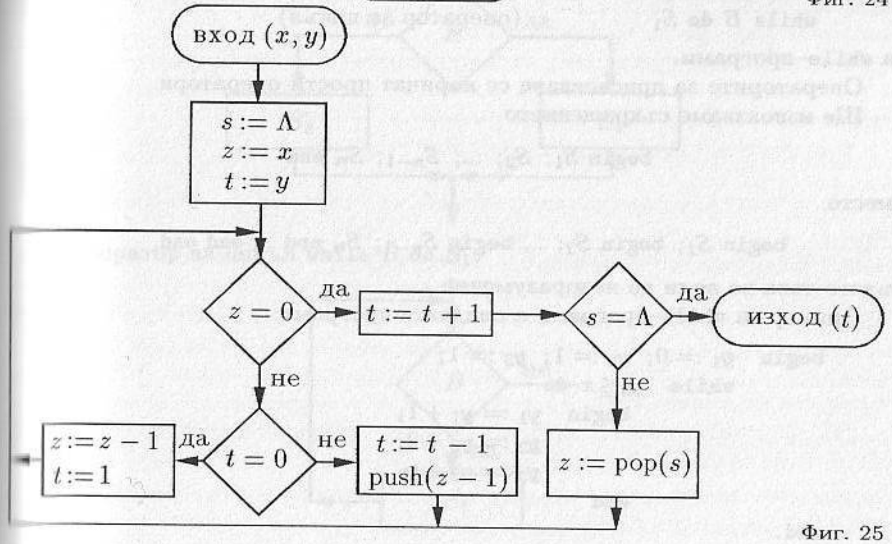
Задача 21. Да се докаже, че програмата P_{24} (фиг. 24) пресмята функцията 91 на Маккарти (вж. зад. 12, § 1.1).

Задача 22. Да се докаже тотална коректност на итеративната програма P_{25} (фиг. 25) за пресмятане на функцията на Акерман (вж. зад. 14, § 1.1).

Забележка. В програмата P_{25} s е променлива за стек, а Λ е празният стек.



Фиг. 24



Фиг. 25

§ 1.4. Метод на Хоар за доказателство на частична коректност на while-програми

В този параграф ще опишем един метод, предложен от Хоар, за верификация на while-програми, т.е. програми, в които циклите се осъществяват с помощта на оператор while-do и не се разрешава използването на оператор go to.

While-програмите са съставени от оператори за присвояване, if-then-else–оператори и while-do–оператори. Те образуват ядрото на всеки алголоподобен език, например като езика Паскал.

Без ограничение на общността ще смятаме, че е фиксиран типът данни на естествените числа, т.е. изходните операции и релации са зададени в езика на аритметиката, който ще означаваме с $\mathcal{L}$. По подобен начин може да се илюстрира методът на Хоар, ако се разглежда друг тип данни.

Просто условие в $\mathcal{L}$ ще наричаме всяка безкванторна формула в $\mathcal{L}$.

While–програмите за езика $\mathcal{L}$ се задават индуктивно:

(1) за всяка променлива x и всеки терм τ в $\mathcal{L}$

$x := \tau$ (оператор за присвояване)

е while–програма;

(2) ако S_1 и S_2 са while–програми и B е просто условие в $\mathcal{L}$, то

begin S_1; S_2 end (съставен оператор),
if B then S_1 else S_2 (оператор за условен преход),
while B do S_1 (оператор за цикъл)

са while–програми.

Операторите за присвояване се наричат прости оператори.

Ще използваме съкращението

begin S_1; S_2; ...; S_{n-1}; S_n end

вместо

begin S_1; begin S_2; ... begin S_{n-1}; S_n end ... end end,

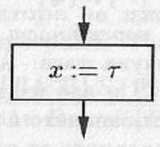
където това не води до недоразумение.

Пример за while–програма е следната програма:

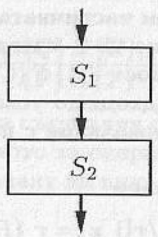
```
begin  y1 := 0;  y2 := 1;  y3 := 1;
       while  y3 ≦ x  do
              begin  y1 := y1 + 1;
                     y2 := y2 + 2;
                     y3 := y3 + y2
              end
end.
```


Изпълнението на програмата S: begin S_1; ... ; S_n end над дадени входни данни представлява последователното изпълнение на while–програмите $S_1, \ldots, S_n$. Изпълнението на всеки от операторите може да се илюстрира с фрагмент от блок–схема по следния начин.

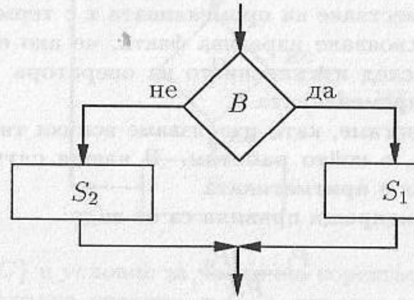
Оператор за присвояване $x := \tau$:



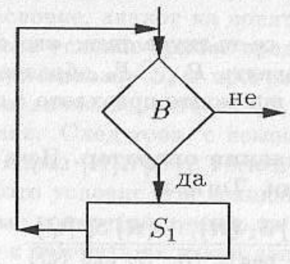
Съставен оператор begin S_1; S_2 end:



Оператор за условен преход if B then S_1 else S_2:



Оператор за цикъл while B do S_1:



Както се вижда, всяка while–програма може да се представи с блок-схема от специален вид.

Ако P и Q са условия в езика $\mathcal{L}$, а S е while–програма в $\mathcal{L}$, то израз от вида
$$\{P\}\,S\,\{Q\}$$
се нарича *условие за частична коректност*.

Изразът $\{P\}\,S\,\{Q\}$ се тълкува така: ако условието P е в сила за входните данни на програмата S и ако изпълнението на S завърши над тези входни данни, то след изпълнението е в сила условието Q. С други думи, $\{P\}\,S\,\{Q\}$ изразява условието за частична коректност на S при входно условие P и изходно условие Q.

Така, ако искаме да покажем частичната коректност на дадена while–програма S при входно условие A и изходно условие C, разглеждаме условието за частична коректност $\{A\}\,S\,\{C\}$. С всяка подпрограма S_i на програмата S свързваме подходящо условие за частична коректност $\{P_i\}\,S_i\,\{Q_i\}$, чиято валидност показваме с помощта на следните верифициращи правила.

(I) Правило за присвояване.
$$\{P[x/\tau]\}\ x := \tau\ \{P\},$$
където P е условие, x е променлива, τ е терм в $\mathcal{L}$, а $P[x/\tau]$ е условие, получено от P чрез заместване на променливата x с терма τ.

Правилото за присвояване изразява факта, че ако едно свойство е вярно за терма τ, то след изпълнението на оператора $x := \tau$ същото свойство е вярно и за променливата x.

Това правило прилагаме, като използваме всички твърдения, които са верни в типа данни, в който работим. В нашия случай използваме всички верни твърдения в аритметиката.

Следващите верифициращи правила са от вида:
$$\frac{P_1 \ldots, P_n}{P},$$
където $P_1 \ldots P_n$ и P са условия за частична коректност или твърдения в езика $\mathcal{L}$.

Правило от такъв тип се тълкува така: ако са в сила $P_1, \ldots$ и P_n, то е изпълнено и P. Условията $P_1, \ldots P_n$ образуват предпоставката на правилото, т.е. условията, при които правилото е приложимо, а P е заключението на правилото.

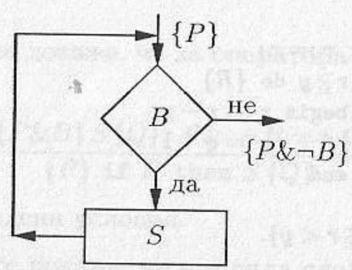
(II) Правило за съставния оператор. Нека P, Q и R са условия, а S_1 и S_2 са while–програми. Тогава
$$\frac{\{P\}\,S_1\,\{R\},\quad \{R\}\,S_2\,\{Q\}}{\{P\}\ \mathtt{begin}\ S_1;\ S_2\ \mathtt{end}\ \{Q\}}.$$

(III) Правило за условния преход. Нека P и Q са условия, B е просто условие, а S_1 и S_2 са while–програми. Тогава
$$\frac{\{P\,\&\,B\}\,S_1\,\{Q\},\ \{P\,\&\,\neg B\}\ S_2\,\{Q\}}{\{P\}\ \mathtt{if}\ B\ \mathtt{then}\ S_1\ \mathtt{else}\ S_2\{Q\}}.$$

(IV) Правило за оператора за цикъл. Нека P е условие, B е просто условие, а S е while–програма. Тогава
$$\frac{\{P\,\&\,B\}\,S\,\{P\}}{\{P\}\ \mathtt{while}\ B\ \mathtt{do}\ S\,\{P\,\&\,\neg B\}}.$$

(V) Правило за следствието. Нека P, Q, R и T са условия, а S е while–програма. Тогава
$$\frac{P \Longrightarrow Q,\ \{Q\}\,S\,\{R\},\ R \Longrightarrow T}{\{P\}\,S\,\{T\}}.$$

Смисълът на правилата за съставния оператор, за условния преход и за следствието е ясен. Правилото за оператора за цикъл изразява свойството, че условието P е инвариант на цикъла в точката преди тестването на условието B, т.е.



Нека $\{A\}\,S\,\{C\}$ е условие за частична коректност, където S е while–програма, A — входно условие, а C — изходно условие. За да приложим тези правила към $\{A\}\,S\,\{C\}$, най-напред с всеки оператор за цикъл свързваме подходящо условие, аналог на понятието инвариант на цикъл (§ 1.2). Естествено за това условие трябва предварително да се досетим. За всеки оператор за присвояване S_i в програмата S показваме валидността на условия за частична коректност $\{P_i\}S_i\{Q_i\}$, като използваме правилото за присвояване. След това, с помощта на останалите правила, извеждаме желания израз $\{A\}S\{C\}$. Тогава програмата S е частично коректна относно входното условие A и изходното условие C.

Преди да разгледаме конкретен пример за прилагане на този метод, да забележим първо, че с помощта на някои логически правила можем да изведем допълнителни верифициращи правила, които ще ни помогнат да съкратим разсъжденията.

(I′) $$\frac{P \Longrightarrow Q[x/\tau]}{\{P\}\, x := \tau\, \{Q\}},$$

където P и Q са условия, x е променлива, а τ е терм.

Доказателство. Нека $P \Longrightarrow Q[x/\tau]$. От правилото за заместване имаме $\{Q[x/\tau]\}\, x := \tau\, \{Q\}$. Освен това $Q \Longrightarrow Q$. Следователно $\{P\}\, x := \tau\, \{Q\}$ (по правилото за следствието).

(II′) Нека P_0, P_1, ... P_n са условия, а S_1, ..., S_n са while-програми. Тогава

$$\frac{\{P_0\}S_1\{P_1\},\ \{P_1\}S_2\{P_2\}, \ldots \{P_{n-i}\}S_n\{P_n\}}{\{P_0\}\ \texttt{begin}\ S_1;\ S_2;\ \ldots;S_n\ \texttt{end}\ \{P_n\}}.$$

Ясно е, че като приложим няколко пъти правилото за съставния оператор, се получава (II′).

Пример. Да разгледаме програмата за намиране на частно и остатък при целочислено делене:

```
(1) {x ≧ 0 & y > 0}
      begin  q := 0;  r := x;
          while r ≧ y do {R}
              begin r := r − y;
                    q := q + 1
              end
      end
      {x = q.y + r & 0 ≦ r < y}.
```

Трябва да докажем, че условието за частична коректност (1) е изпълнено, т.е. че горната програма е частично коректна относно входно условие $x \geqq 0 \& y > 0$ и изходно условие $x = q.y + r \& 0 \leqq r < y$. Основната задача е да намерим условие R, което е инвариант на цикъла, точно както постъпваме в зад. 2, § 1.2. За целта тръгваме отзад напред, т.е. от изходното условие: имаме $x = q.y + r \& 0 \leqq r < y$. Забелязваме, че при излизане от while–цикъла имаме $r < y$. Затова полагаме $R : x = qy + r \& 0 \leqq r$. Достатъчно е да покажем, че

(2) $\{x \geqq 0 \& y > 0\}$ begin $q := 0$; $r := x$ end $\{R\}$,

(3) $\{R \& r \geqq y\}$ begin $r := r - y$; $q := q + 1$ end $\{R\}$.

Тогава по правило (IV) ще имаме

(4) $\{R\}$ while $r \geqq y$ do begin $r := r - y$; $q := q + 1$ end $\{x = q, y + r \& 0 \leqq r < y\}$.

Комбинирайки (2) и (4) по правилото за съставния оператор (II), имаме, че (1) е в сила, т.е. програмата е частично коректна относно посочените условия.

Проверка на (3):

(5) $(x = q.y + r \& r \geqq 0 \& r \geqq y) \Longrightarrow (x = (q+1).y + (r-y) \& r - y \geqq 0)$ (твърдение, вярно в аритметиката),

(6) $\{x = (q+1).y + (r-y) \& r - y \geqq 0\}\ r := r - y\ \{x = (1+q).y + r \& r \geqq 0\}$ (по правилото за присвояване),

(7) $\{x = (1+q).y + r \& r \geqq 0\}\ q := q + 1\ \{x = q.y + r \& r \geqq 0\}$ (по правилото за присвояване).

От правилата (I′) и (II′) имаме, че (3) е в сила.

Проверка на (2):

(8) $x \geqq 0 \& y > 0 \Longrightarrow x = 0.y + x \& x \geqq 0$,

(9) $\{x = 0.y + x \& x \geqq 0\}\ q := 0\ \{x = q.y + x \& x \geqq 0\}$ (по правилото за присвояване),

(10) $\{x = q.y + x \& x \geqq 0\}\ r := x\ \{x = q.y + r \& r \geqq 0\}$ (по правилото за присвояване).

От правилата (I′), (II′) и от (8), (9) и (10) получаваме, че (2) е в сила.

Задача 1. Да се докаже, че за оператора if B then S е в сила следното правило:

$$\frac{\{P \& B\}\, S\, \{Q\},\ P \& \neg B \Longrightarrow Q}{\{P\}\ \texttt{if}\ B\ \texttt{then}\ S\, \{Q\}},$$

където P и Q са дадени условия.

Задача 2. Да се покаже, че е в сила следното правило:

$$\frac{P \Longrightarrow R,\ \{R \& B\}\, S\{R\},\ R \& \neg B \Longrightarrow Q}{\{P\}\ \texttt{while}\ B\ \texttt{do}\ S\, \{Q\}},$$

където P, R и Q са условия.

Задача 3. Да се докаже, че за оператора repeat S until B е в сила правилото

$$\frac{\{P\}\, S\{Q\},\ Q \& \neg B \Longrightarrow P}{\{P\}\ \texttt{repeat}\ S\ \texttt{until}\ B\, \{Q \& B\}}.$$

Упътване. Използвайте, че repeat S until B = begin S; while $\neg B$ do S end.

Задача 4. Да се докаже, че

$$\{a \geqq 0 \,\&\, b \geqq 0\}$$
if $a > b$ then $a := a - b$
else $b := b - a$
$$\{a \geqq 0 \,\&\, b \geqq 0\}.$$

Задача 5. Нека $fib(n)$ е n-тото число на Фибоначи, т.е.

$$\left| \begin{array}{l} fib(0) = 0 \\ fib(1) = 1 \\ fib(n+2) = fib(n) + fib(n+1). \end{array} \right.$$

Да се докаже, че

```
{n ≧ 0}
begin a := 1; b := 0; i := 1;
    while i ≦ n   do {R}
        begin a := a + b; b := a − b;
              i := i + 1
        end
end
{b = fib(n)}.
```

Доказателство. Нека

$$R : a = fib(i) \,\&\, b = fib(i-1) \,\&\, i \leqq n+1.$$

Имаме:

(1) $n \geqq 0 \Longrightarrow n+1 \geqq 1$;

(2) $\{1 = fib(1)\}$ $a := 1$ $\{ a = fib(1)\}$ (по правило (I));

(3) $\{0 = fib(0)\}$ $b := 0$ $\{ b = fib(0)\}$ (по правило (I));

(4) $\{a = fib(1) \,\&\, b = fib(0) \,\&\, n+1 \geqq 1\}$ $i := 1$
$\{a = fib(i) \,\&\, b = fib(i-1) \,\&\, n+1 \geqq i\}$ (по правило (I));

Така по (II′) получаваме

(5) $\{n \geqq 0\}$ begin $a := 1$; $b := 0$; $i = 1$ end $\{R\}$.

Остава да покажем

(6) $\{R \,\&\, i \leqq n\}$ begin $a := a + b$; $b := a - b$; $i := i + 1$ end $\{R\}$.

Тогава по правило (IV) и (V) ще следва че програмата е частично коректна относно входното условие $n \geqq 0$ и изходното $b = fib(n)$, тъй като $i > n \,\&\, i \leqq n+1 \Longrightarrow i = n+1$.

За доказателството на (6) проверяваме последователно:

(7) $a = fib(i) \,\&\, b = fib(i-1) \Longrightarrow a + b = fib(i+1)$;

(8) $\{a + b = fib(i+1)\}$ $a := a + b$ $\{a = fib(i+1)\}$;

(9) $fib(i+1) = fib(i) + fib(i-1) \Longrightarrow fib(i) = fib(i+1) - fib(i-1)$;

(10) $a = fib(i+1) \,\&\, b = fib(i-1) \,\&\, fib(i) = fib(i+1) - fib(i-1) \Longrightarrow a = fib(i+1) \,\&\, fib(i) = a - b$;

(11) $\{a = fib(i+1) \,\&\, a - b = fib(i)\}$ $b := a - b$ $\{a = fib(i+1) \,\&\, b = fib(i)\}$;

(12) $\{a = fib(i+1) \,\&\, b = fib((i+1)-1) \,\&\, i \leqq n\}$ $i := i + 1$ $\{R\}$.

Задача 6. Да се покаже, че

```
{x ≧ 0}
    begin y1 := 0; y2 := 1; y3 := 1;
        while y3 ≦ x do {R}
            begin y1 := y1 + 1;
                  y2 := y2 + 2;
                  y3 = y3 + y2
            end
    end
{y1 = [√x]},
```

където $R : y_1^2 \leqq x \,\&\, y_3 = (y_1+1)^2 \,\&\, y_2 = 2y_1 + 1$.

Упътване. Използвайте, че $y_1 = [\sqrt{x}] \iff y_1^2 \leqq x < (y_1+1)^2$.

Задача 7. Да се покаже, че

```
{x > 0 & y > 0}
begin r := x; q := 0; w := y;
    {∃n(w = 2^n.y) & x = q.w + r & 0 ≦ r}
    while w ≦ r do w := 2.w
    {∃n(w = 2^n.y) & x = q.w + r & 0 ≦ r < w}
    while w ≠ y do
        begin q := 2.q; w := [w/2];
            if w ≦ r then
                begin r := r − w; q := q + 1
                end
        end
end
{x = q.y + r & 0 ≦ r < y}.
```

Задача 8. Да се провери, че

```
{a > 0 & b > 0}
begin x := a; y := b;
      {x > 0 & y > 0 & НОД(x, y) = НОД(a, b)}
      repeat
            begin while x > y do x := x − y;
                  while y > x do y := y − x
            end
      until x = y
end
{x = y = НОД(a, b)},
```

където с НОД(a, b) е означен най-големият общ делител на a и b.

Задача 9. Да се докаже, че

```
{a ≧ 0, b ≧ 0}
begin x := a; y := b; z := 0;
      while x ≠ 0 do
            begin if odd(x) then z := z + y;
                  y := 2y;
                  x := [x/2]
            end
end
{z = a.b},
```

където $odd(x)$ е предикатът „x е нечетно“.

Упътване. Покажете, че инвариант на цикъла е $ab = x.y + z$.

Задача 10. Да се покаже, че:

```
{a > 0 & b > 0}
begin x := a; y := b; u := b, v := 0;
      while x ≠ y do
            begin while x > y do
                        begin x := x − y;
                              v := v + u
                        end
                  while y > x do
                        begin y := y − x;
                              u := u + v
                        end
            end
      u := u + v;
end
{x = НОД(a, b) & u = НОК(a, b)}.
```


Упътване. Проверете, че следните условия са инвариантни:

$$\text{НОД}(x, y) = \text{НОД}(a, b), \quad u.x + v.y = a.b.$$

Задача 11. Нека $a_1 \leqq \ldots \leqq a_n$, $n \geqq 1$, е масив от числа, който е сортиран по големина, и x е елемент в масива. Да се докаже, че за следната програма е изпълнено:

```
{n ≧ 1 & a_1 ≦ ... ≦ a_n & ∃i(1 ≦ i ≦ n & x = a_i)}
begin i := 1; j := n;
      while i ≠ j do
            begin k := [(i + j)/2];
                  if x ≦ a_k then j := k
                             else i := k + 1
            end
end
{1 ≦ i ≦ n & x = a_i}.
```

Упътване. Докажете, че инвариантно условие на цикъла е

$$R : A \,\&\, 1 \leqq i \leqq j \leqq n \quad \& \quad a_i \leqq x \leqq a_j,$$

където A е входното условие. Използвайте и, че $i < j \Longrightarrow i \leqq [(i+j)/2] < j$.

Задача 12. Нека $a_0, \ldots a_n$, $n \geqq 0$, е масив от числа. Да се докаже, че

```
{n > 0}
begin r := a([n/2]); i := 0; j := n;
      while i ≦ j do
            begin while a_i < r do i := i + 1;
                  while r < a_j do j := j − 1;
                  if i ≦ j then begin swap (a_i, a_j);
                                      i := i + 1; j := j − 1
                                end
            end
end
{j < i & ∀k∀l(0 ≦ k < i & j < l ≦ n ⟹ a_k ≦ a_l)}.
```

Тук операторът swap (a_i, a_j) разменя местата на a_i и a_j.

Съобразете още, че елементите на получения масив са пермутация на елементите на входния.

Упътване. Програмата пренарежда масива $a_0, \ldots, a_n$, като го разделя на две части така, че всеки елемент на първата част $a_0, \ldots, a_{i-1}$ е по-малък или равен на всеки елемент от втората част $a_{j+1}, \ldots, a_n$, като $0 \leqq j < i \leqq n$.

Въвеждаме означенията:

$a[0..i) \leqq a(j..n] \iff \forall k \forall l (0 \leqq k < i \,\&\, j < l \leqq n \Longrightarrow a_k \leqq a_l)$,

$a[0..i) \leqq r \iff \forall k (0 \leqq k < i \Longrightarrow a_k \leqq r)$,

$r \leqq a(j..n] \iff \forall l (j < l \leqq n \Longrightarrow r \leqq a_l)$.

Покажете, че

```
{n > 0}
 begin r := a[(n/2)]; i := 0; j := n;
   {0≦i & a[0..i) ≦ r & j≦n & r ≦ a(j..n]}
   while i≦j do
         begin while a_i < r do i := i+1;
   {0≦i & a[0..i) ≦ r & j≦n & r≦a(j..n] & r≦a_i}
               while r < a_j do j := j−1;
   {0≦i & a[0..i) ≦ r & j≦n & r≦a(j..n] & a_j≦r≦a_i}
               if i≦j then begin swap (a_i, a_j);
                                 i := i+1;
                                 j := j−1
                           end
   {0≦i & a[0..i) ≦ r & j≦n & r≦a(j..n]}
         end
 end
{j < i & a[0..i) ≦ a(j..n]}.
```

Задача 13. Нека $a[1..n] = (a_1, \dots, a_n)$ и $b[1..m] = (b_1, \dots, b_m)$ са сортирани в нарастващ ред масиви от положителни числа. Приемаме, че 0 е знак за край на масива. Нека $c[1..\max(n,m)]$ е масив с дължина, равна на дължината на по-дългия от двата масива $a[1..n]$ и $b[1..m]$, като c в началото е запълнен с 0. Да се докаже, че следващата програма намира общите елементи на $a[1..n]$ и $b[1..m]$ и ги записва в масива c в нарастващ ред.

```
begin i := 1; j := 1; k := 1;
  while (a_i ≠ 0 & b_j ≠ 0) do
    begin if a_i = b_j then
            begin c_k := a_i;
                  k := k+1; i := i+1;
                  j := j+1
            end
          else if a_i < b_j
                  then i := i+1
                  else j := j+1
    end
end.
```


С други думи, докажете, че програмата е частично коректна относно входното условие

$$0 < a_1 < \ldots < a_n \,\&\, 0 < b_1 < \ldots < b_m \,\&\, c[1..\max(m,n)] = 0$$

и изходното условие

$$c = a \cap b \,\&\, c \text{ е сортиран},$$

като $c = a \cap b$ означава, че c се състои от общите елементи на a и b.

Упътване. Докажете, че условието

$c[1..k-1] = a[1..i-1] \cap b[1..j-1] \,\&\, b[1..j-1] \cap a[i..n] = \varnothing \,\&\, a[1..i-1] \cap b[j..m] = \varnothing$

е инвариант на цикъла.

Задача 14. Нека x и y са естествени числа, записани в десетична бройна система, а a и b са масиви, съдържащи съответно цифрите на числата x и y, т.е. $x = \sum_{i=0}^{n-1} a_i.10^i$, $y = \sum_{i=0}^{m-1} b_i.10^i$. В резултат на изпълнението на дадената програма за събиране на числата x и y се получава масив s, съдържащ цифрите на числото $x+y$. Да се докаже, че

$$\{x = \sum_{i=0}^{n-1} a_i.10^i \,\&\, y = \sum_{i=0}^{m-1} b_i.10^i\}$$

```
begin i := 0; l := 0;
  while i ≠ max(n, m) do
    begin z := l + a_i + b_i;
          if z ≧ 10 then begin s_i := z − 10;
                               l := 1
                         end;
                    else begin s_i := z;
                               l := 0
                         end;
          i := i+1
    end;
  s_i := l
end
```

$\{x+y = \sum_{i=o}^{\max(n,\,m)} s_i.10^i\}$.

Забележка. Предполагаме, че ако $n > m$, то

$$b_m = b_{m+1} = \dots b_{n-1} = 0.$$

Задача 15. Нека $a_0 \leqq \dots \leqq a_{n-1}$ и $b_0 \leqq \dots \leqq b_{m-1}$ са сортирани масиви от числа, съответно с $n \geqq 1$ и $m \geqq 1$ елемента и

$$\mathrm{mindist}(a,\, b) = \min_{0 \leqq i < n, 0 \leqq j < m} |a_i - b_j|.$$

Да се покаже, че

$\{a_0 \leqq \dots \leqq a_{n-1} \,\&\, b_0 \leqq \dots \leqq b_{m-1}\}$
begin $d := \max(a_{n-1} - b_0,\, b_{m-1} - a_0)$;
  $i := 0;\ j := 0;$
  while $i \neq n \,\&\, j \neq m$ do
    if $a_i \geqq b_j$ then begin $d := \min(d,\, a_i - b_j)$;
              $j := j + 1$
            end
        else begin $d := \min(d,\, b_j - a_i)$;
              $i := i + 1$
            end
end
$\{d = \mathrm{mindist}(a,\, b)\}$.

Задача 16. Дадена е редицата от числа $a_0, \dots a_{n-1}$. Монотонна подредица на дадената редица наричаме такава подредица

$$a_i,\, a_{i+1}, \dots a_{i+k}, \quad 0 \leqq i \leqq i + k < n,$$

за която $a_i \leqq a_{i+1} \leqq \dots \leqq a_{i+k}$ или $a_i \geqq a_{i+1} \geqq \dots \geqq a_{i+k}$.

Да се покаже, че

$\{a_0 \dots a_{n-1}$ е редица от числа $\& \, n > 0\}$
begin $i := 1;\ q := 1;\ v := 1;\ w := 1;$
  while $i \neq n$ do
  begin if $a_i > a_{i-1}$ then begin $w := 1;\ v := v + 1$ end;
    else if $a_i = a_{i-1}$ then begin $w := w + 1$;
              $v := v + 1$
            end;
        end;
      else begin $w := w + 1$;
              $v := 1$
            end;
    $q := \max(q,\, \max(v,\, w));\ i := i + 1$
  end
end
$\{q$ е броят на елементите на най-дългата монотонна подредица на $a_0, \dots a_{n-1}\}$.

Задача 17. Нека $x_0, \dots, x_{n-1}$ е пермутация без повторение на числата $0,\, 1, \dots, n-1$. Да се докаже, че следващата програма за всяко x_i намира броя на числата, по-малки от x_i, които са преди x_i в тази пермутация.

$\{x_0, \dots, x_{n-1}$ е пермутация без повторение на числата $0, 1, \dots, n-1\}$
begin $k := 0$;
  while $k < n$ do $v_k := x_k$;
  $i := n$;
  while $i \neq 0$ do
    begin $j := 0;\ i := i - 1$;
      while $j \neq i$ do
        begin if $v_i < v_j$ then $v_j := v_j - 1$;
          $j := j + 1$
        end
    end
end
$\{\forall k (0 \leqq k < n)(v_k =$ броя на тези $l : (0 \leqq l < k \,\&\, x_l < x_k)\}$.

Упътване: Да означим с $y(k)$ броя на тези l, за които

$$0 \leqq l < k \,\&\, x_l < x_k.$$

Разгледайте условието:

$v_0, \dots, v_{i-1}$ е пермутация на числата от 0 до $i - 1 \,\&$
$\forall k (0 \leqq k < i)$(броят на тези $l(0 \leqq l < k \,\&\, v_l < v_k)$ е $y(k)) \,\&$
$\forall k (i \leqq k < n)\ (v_k = y(k))$.

В т о р а   г л а в а

КОМПАКТНИ ОПЕРАТОРИ

§ 2.1. Частични функции и операции с тях

Нека X и Y са множества, а f е функция, преобразуваща елементи на X в елементи на Y, като за някои $x \in X$ стойността на f в точката x може да не е определена. В такъв случай казваме, че f е *частична* функция от X към Y и записваме

$$f : X \twoheadrightarrow Y.$$

Записът $!f(x)$ ще означава, че f е определена в точката x, а $\neg!f(x)$ — че f не е определена в x (или няма смисъл $f(x)$).

Дефиниционна област на функцията f наричаме множеството

$$\mathrm{dom}(f) = \{x \mid x \in X \,\&\, !f(x)\}$$

(от англ. domain — област). Ако $\mathrm{dom}(f) = X$, казваме, че f е *навсякъде определена (тотална)*. В този случай използваме обичайния запис $f : X \to Y$.

Област от стойностите на f е множеството

$$\mathrm{range}(f) = \{y \mid \exists x(x \in \mathrm{dom}(f) \,\&\, f(x) = y)\}$$

(от англ. range — област, обхват).

Функцията f наричаме *сюрективна* (или изображение *върху* Y), ако

$$\mathrm{range}(f) = Y.$$

Казваме, че частичната функция $f : X \twoheadrightarrow Y$ е *инективна* (или *обратима*), ако за всяко $x,\ y \in \mathrm{dom}(f)$ е изпълнено условието

$$x \neq y \Longrightarrow f(x) \neq f(y).$$

Частичната функция $f : X \twoheadrightarrow Y$ наричаме *биективна*, ако тя е едновременно инективна и сюрективна. Ако f е тотална, казваме, че f е *биекция* между X и Y.

Когато работим с изрази, които не винаги имат стойност, се налага да додефинираме релацията $=$ за случаите, когато някой от аргументите ѝ не е определен. Разширената релация ще наричаме *условно равенство* и ще отбелязваме с $\simeq$. Тя има стойност „истина“ когато или и двата ѝ аргумента са дефинирани и имат една и съща стойност, или и двата ѝ аргумента са недефинирани.

Например условното равенство $\dfrac{x^2}{x} \simeq \dfrac{x^3}{x^2}$ е вярно при всяка стойност на x, включително и при $x = 0$, тъй като тогава и двете му страни са неопределени. За разлика от него, условното равенство $\dfrac{x^2}{x} \simeq x$ не е вярно при $x = 0$, защото в този случай лявата му страна е неопределена, а дясната има стойност (равна на 0).

Графика на функцията $f : X \twoheadrightarrow Y$ наричаме множеството

$$G_f = \{(x, y) \mid x \in X \,\&\, f(x) \simeq y\}.$$

Рестрикция на $f : X \twoheadrightarrow Y$ върху множеството $X_0 \subseteq X$ наричаме функцията

$$g : X_0 \twoheadrightarrow Y,$$

за която е изпълнено

$$g(x) \simeq f(x)$$

при всеки избор на $x \in X_0$.

Рестрикцията на f върху X_0 означаваме с $f|_{X_0}$.

Нека f и g са частични функции от X към Y. Казваме, че f и g *съвпадат* ($f = g$), ако $f(x) \simeq g(x)$ за всяко $x \in X$. С други думи, f съвпада с g, ако за всяко $x \in X$ е изпълнено точно едно от следните две условия:

(1) $!f(x)$, $!g(x)$ и $f(x) = g(x)$;

(2) $\neg!f(x)$ и $\neg!g(x)$.

Частичните функции можем да сравняваме по отношение на тяхната определеност. Казваме, че f е *подфункция* на g (или симетрично, g е продължение на f), ако за всяко $x \in X$ и всяко $y \in Y$ е изпълнено условието

$$f(x) \simeq y \Longrightarrow g(x) \simeq y.$$

В такъв случай пишем $f \subseteq g$.

Задача 1. Нека f и g са произволни частични функции. Да се докаже, че следните условия са еквивалентни:

(1) $f \subseteq g$;

(2) $G_f \subseteq G_g$;

(3) $\mathrm{dom}(f) \subseteq \mathrm{dom}(g)$ и $\forall x(x \in \mathrm{dom}(f) \Longrightarrow f(x) = g(x))$.

Задача 2. Нека f и g са частични функции. Да се докаже, че изброените условия са еквивалентни:

(1) $f = g$;

(2) $G_f = G_g$;

(3) $\mathrm{dom}(f) = \mathrm{dom}(g)$ и $\forall x (x \in \mathrm{dom}(f) \Longrightarrow f(x) = g(x))$;

(4) $f \subseteq g \,\&\, g \subseteq f$.

Задача 3. Нека $f : X \twoheadrightarrow Y$ и $g : X \twoheadrightarrow Y$. Да се докаже, че $f = g$ тогава и само тогава, когато при всеки избор на $x \in X$ и $y \in Y$ са изпълнени условията:

(1) $f(x) \simeq y \Longrightarrow g(x) \simeq y$;

(2) $\neg ! f(x) \Longrightarrow \neg ! g(x)$.

Доказателство. Условията (1) и (2) следват непосредствено от определението за съвпадане на f и g.

Обратно, да предположим, че (1) и (2) са в сила, и да изберем произволно $x \in X$. Трябва да покажем, че $f(x) \simeq g(x)$. Наистина, ако $x \in \mathrm{dom}(f)$, то от (1) следва, че $f(x) = y = g(x)$, т.е. $f(x) \simeq g(x)$. Ако $x \notin \mathrm{dom}(f)$, от второто условие получаваме, че $x \notin \mathrm{dom}(g)$ и следователно отново $f(x) \simeq g(x)$.

Задача 4. Нека $f : X \twoheadrightarrow Y$, $X_0 \subseteq X$ и g е рестрикцията на f върху X_0. Да се докаже, че g е подфункция на f. При какво условие g съвпада с f?

Задача 5. Да се докаже, че релацията $\subseteq$ е частична наредба в съвкупността от всички частични функции от X към Y.

Оттук нататък ще разглеждаме частични функции в множеството $\mathbb{N}$ на естествените числа. Ако това не води до някаква неяснота, наредената n-орка $(x_1, \ldots, x_n) \in \mathbb{N}^n$ ще означаваме съкратено с $\overline{x}$. Записът $f(\overline{x})$ ще употребяваме в два случая: за да означим стойността на f в точката $\overline{x}$, или (по-рядко) — за да отбележим, че f е функция на аргументи, означени с $\overline{x}$.

Нека α е израз, в който се срещат променливите $x_1, \ldots, x_n$. Да предположим още, че част от тези променливи — например $x_{k+1}, \ldots, x_n$, $k \leqq n$, са фиксирани. Тогава записът

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

ще е означение за функцията $f : \mathbb{N}^k \twoheadrightarrow \mathbb{N}$, където

$$f(x_1, \ldots, x_k) \simeq \alpha(x_1, \ldots, x_n)$$

при всеки избор на $\overline{x} \in \mathbb{N}^k$.

Нека $f_0 : \mathbb{N}^n \twoheadrightarrow \mathbb{N}$, $f_1 : \mathbb{N}^k \twoheadrightarrow \mathbb{N}, \ldots, f_n : \mathbb{N}^k \twoheadrightarrow \mathbb{N}$. Функцията

$$f_0(f_1, \ldots, f_n) : \mathbb{N}^k \twoheadrightarrow \mathbb{N},$$

дефинирана посредством еквивалентността

$$f_0(f_1, \ldots, f_n)(\overline{x}) \simeq y \Longleftrightarrow$$

$$\exists z_1 \ldots \exists z_n (f_1(\overline{x}) \simeq z_1 \,\&\, \ldots \,\&\, f_n(\overline{x}) \simeq z_n \,\&\, f_0(z_1, \ldots, z_n) \simeq y)$$

наричаме *суперпозиция* на $f_0, f_1, \ldots, f_n$.

При $n = 1$ функцията $f_0(f_1)$ се нарича *композиция* на f_0 и f_1 и се отбелязва с $f_0 \circ f_1$.

Ясно е, че $f_0(f_1, \ldots, f_n)$ е определена в $\overline{x}$ тогава и само тогава, когато $!f_1(\overline{x}), \ldots, !f_n(\overline{x})$ и $!f_0(f_1(\overline{x}), \ldots, f_n(\overline{x}))$. В такъв случай горното определение бихме могли да запишем по обичайния за суперпозиция на тотални функции начин

$$f_0(f_1, \ldots, f_n)(\overline{x}) \simeq f_0(f_1(\overline{x}), \ldots, f_n(\overline{x})),$$

но като се уговорим да подразбираме, че един такъв израз е дефиниран само когато са дефинирани всички подизрази, участващи в него.

От определението на суперпозиция се вижда, че ако всяка от функциите $f_0, f_1, \ldots, f_n$ е тотална, то и $f_0(f_1, \ldots, f_n)$ също е тотална функция.

§ 2.2. Компактни оператори

Нека за всяко $n \geqq 1$ с $\mathcal{F}_n$ означим съвкупността от всички частични функции

$$f : \mathbb{N}^n \twoheadrightarrow \mathbb{N}.$$

Да припомним, че ако f и g са елементи на $\mathcal{F}_n$, то функцията g е *продължение* на функцията f ($f \subseteq g$), ако

$$f(x_1, \ldots, x_n) \simeq y \Longrightarrow g(x_1, \ldots, x_n) \simeq y$$

при всеки избор на естествените числа $x_1, \ldots, x_n$ и y.

Всяко тотално изображение $\Gamma : \mathcal{F}_n \to \mathcal{F}_r$ наричаме *оператор* от тип (n, r).

Нека Γ е оператор от тип (n, r). Казваме, че Γ е *монотонен*, ако при всеки избор на f и g от $\mathcal{F}_n$ е в сила

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

Операторът Γ е *компактен*, ако за всеки елемент f на $\mathcal{F}_n$ е изпълнено

$$\Gamma(f)(x_1, \ldots, x_r) \simeq y \iff \exists \theta \subseteq f (\theta \text{ е крайна } \& \ \Gamma(\theta)(x_1, \ldots, x_r) \simeq y)$$

за всяко $x_1, \ldots, x_r$ и y.

Твърдение 1. *Всеки компактен оператор е и монотонен.*

Отношението $\subseteq$ задава частична наредба в множеството $\mathcal{F}_n$. Тъй като никъде недефинираната функция $\emptyset^{(n)}$ се продължава от всяка частична функция, тя е най-малкият елемент на $\mathcal{F}_n$.

Нека $X \subseteq \mathcal{F}_n$. Казваме, че g е *горна граница* на X, ако

$$\forall f (f \in X \Longrightarrow f \subseteq g).$$

Функцията g е *точна горна граница* на X, ако g е горна граница на X и $g \subseteq h$ за всяка горна граница h на X. Точната горна граница на X ще означаваме с $\bigcup X$.

Всяка монотонно растяща редица $f_1 \subseteq f_2 \subseteq \dots \subseteq f_k \subseteq \dots$ от елементи от $\mathcal{F}_n$ има точна горна граница $\bigcup_k f_k$ или само $\bigcup f_k$ (зад. 9). Тя се определя с еквивалентността

$$(\bigcup_k f_k)(\overline{x}) \simeq y \iff \exists k (f_k(\overline{x}) \simeq y).$$

Операторът Γ се нарича (*изброимо*) *непрекъснат*, ако при всеки избор на монотонно растяща редица $f_0 \subseteq \dots \subseteq f_k \subseteq \dots$ от елементи на $\mathcal{F}_n$ е изпълнено

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

Тук имаме предвид, че точната горна граница на множеството $\{\Gamma(f_k) \mid k \in \mathbb{N}\}$ съществува и е равна на $\Gamma(\bigcup f_k)$.

Твърдение 2. *Компактните оператори съвпадат с непрекъснатите.*

Нека Γ е оператор от тип (n, n). Казваме, че функцията f е *неподвижна точка* на Γ, ако $\Gamma(f) = f$. Функцията f е *най-малка неподвижна точка* на Γ, ако $\Gamma(f) = f$ и всеки път, когато $g \in \mathcal{F}_n$ и $\Gamma(g) = g$, имаме $f \subseteq g$.

Ясно е, че ако въобще съществува, най-малката неподвижна точка на Γ е единствена. Ще я означаваме с f_Γ.

Теорема на Кнастер — Тарски. *Всеки непрекъснат оператор от тип (n, n) притежава най-малка неподвижна точка.*

Задача 1. Да се докаже, че всеки от изброените оператори е монотонен:

а) $\Gamma(f) = f$, т.е. Γ е операторът идентитет;

б) $\Gamma(f) = f_0$, където f_0 е фиксиран елемент на $\mathcal{F}_n$ (константен оператор);

в) $\Gamma(f)(x) \simeq 2^{f(x)}$;

г) $\Gamma(f)(x) \simeq f(f(x))$;

д) $\Gamma(f)(x) \simeq \sum_{y<x} f(y)$;

е) $\Gamma(f)(x) \simeq f(x, x)$;

ж) $\Gamma(f)(\overline{x}, y) \simeq \mu z_{<y}[f(\overline{x}, z) \simeq 0]$;

з) $\Gamma(f)(\overline{x}) \simeq \mu z[f(\overline{x}, z) \simeq 0]$;

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

к) $\Gamma(f)(x) \simeq f^{\mathrm{L}(x)}(\mathrm{R}(x))$.

Упътване. и) Нека $f \subseteq g$. Да предположим, че $\Gamma(f)(x, y) \simeq z$. Ще покажем, че $\Gamma(g)(x, y) \simeq z$.

Ако $x = 0$, то от определението на Γ следва, че $\Gamma(g)(x, y) = 0 = z$.

Ако $x \neq 0$, то $z = f(x-1, f(x, y))$. Следователно $!f(x, y)$ и тъй като $f \subseteq g$, то $f(x, y) = g(x, y)$. Нека $u = f(x, y)$. Имаме още $!f(x-1, u)$ и следователно $f(x-1, u) = g(x-1, u) = z$. Така $\Gamma(g)(x, y) = g(x-1, g(x, y)) = z$. Показахме, че $\Gamma(f) \subseteq \Gamma(g)$.

Задача 2. Да се докаже, че операторът Γ, определен с равенството

$$\Gamma(f)(x) \simeq \begin{cases} f(x), & \text{ако dom}(f) \text{ е крайна}, \\ \neg!, & \text{ако dom}(f) \text{ е безкрайна}, \end{cases}$$

не е монотонен.

Упътване. Разгледайте една крайна функция θ с непразна дефиниционна област и функция f с безкрайна дефиниционна област, която я продължава, т.е. $\theta \subseteq f$. Тогава $\Gamma(\theta) \not\subseteq \Gamma(f)$.

Задача 3. Да се докаже, че ако Γ е компактен оператор, то Γ е монотонен.

Доказателство. Нека Γ е компактен и $f \subseteq g$. Ако $\Gamma(f)(\overline{x}) \simeq y$, то за някоя крайна функция $\theta \subseteq f$ е изпълнено $\Gamma(\theta)(\overline{x}) \simeq y$. Но $\theta \subseteq f \subseteq g$ и тъй като Γ е компактен, следва, че $\Gamma(g)(\overline{x}) \simeq y$.

Задача 4. Нека Γ_1 и Γ_2 са компактни оператори от тип (n, r), за които $\Gamma_1(\theta) = \Gamma_2(\theta)$ за всяка крайна функция θ. Да се докаже, че $\Gamma_1 = \Gamma_2$.

Доказателство.

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

Следователно $\Gamma_1(f) = \Gamma_2(f)$ за всяка частична функция $f \in \mathcal{F}_n$.

Задача 5. Да се докаже, че операторът Γ е компактен тогава и само тогава, когато едновременно са изпълнени условията:

(1) Γ е монотонен;

(2) ако $\Gamma(f)(\overline{x}) \simeq y$, то $\exists\theta(\theta$ е крайна & $\theta \subseteq f$ & $\Gamma(\theta)(\overline{x}) \simeq y)$ за произволни $\overline{x}, y \in \mathbb{N}$.

Доказателство. Ако Γ е компактен оператор, то от зад. 3 знаем, че (1) е налице.

Нека Γ удовлетворява (1) и (2). Да предположим, че за някоя крайна функция $\theta \subseteq f$ е изпълнено $\Gamma(\theta)(\overline{x}) \simeq y$. Тъй като Γ е монотонен, то $\Gamma(\theta) \subseteq \Gamma(f)$. Следователно $\Gamma(f)(\overline{x}) \simeq y$.

Да напомним уговорката, че навсякъде, където пишем $f(\overline{x}) > 0$, имаме предвид $!f(\overline{x})$ и $f(\overline{x}) > 0$.

Задача 6. Да се докаже, че операторите от зад. 1 са компактни.

Доказателство. Тъй като тези оператори са монотонни, съгласно зад. 5 е достатъчно да проверим, че те удовлетворяват условие (2) (от същата задача).

Да разгледаме оператора $\Gamma(f)(\overline{x}) \simeq \mu z[f(\overline{x}, z) \simeq 0]$ от зад. 1 з).

Да предположим, че за някои $\overline{x}, y$ имаме $\Gamma(f)(\overline{x}) \simeq y$. Тогава $f(\overline{x}, 0) > 0$, $f(\overline{x}, 1) > 0$, $\ldots$, $f(\overline{x}, y-1) > 0$ и $f(\overline{x}, y) = 0$. Нека θ е крайна подфункция на f с дефиниционна област $\{(\overline{x}, 0), (\overline{x}, 1), \ldots, (\overline{x}, y)\}$. Ясно е, че $\Gamma(\theta)(\overline{x}) \simeq y$.

Да разгледаме оператора от зад. 1 и):

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

Нека $\Gamma(f)(x,y) \simeq z$. Ако $x = 0$, разгледайте $\theta = \varnothing^{(2)}$. Ако $x \neq 0$, то $\Gamma(f)(x,y) \simeq f(x-1, f(x,y))$ и следователно $!f(x,y)$ и $!f(x-1, f(x,y))$. Да положим $u = f(x,y)$ и нека θ е крайната функция с дефиниционна област $\{(x,y), (x-1,u)\}$, за която

$$\theta(x,y) = f(x,y), \ \theta(x-1,u) = f(x-1,u).$$

Тогава от определението на θ получаваме $\theta \subseteq f$. В този случай имаме $\Gamma(\theta)(x,y) \simeq \theta(x-1, \theta(x,y))$. Следователно $\Gamma(\theta)(x,y) \simeq z$.

Задача 7. За всеки от следните оператори да се провери дали е монотонен и дали е компактен:

а) $\Gamma(f)(x) \simeq \begin{cases} 0, & \text{ако } \neg!f(x), \\ \neg!, & \text{ако } !f(x); \end{cases}$

б) $\Gamma(f)(x) \simeq \begin{cases} f(x), & \text{ако } \mathrm{dom}(f) \text{ е безкрайно множество}, \\ \neg!, & \text{ако } \mathrm{dom}(f) \text{ е крайно множество}; \end{cases}$

в) $\Gamma(f)(x) \simeq \begin{cases} 0, & \text{ако } \forall y(f(x,y) > 0), \\ 1, & \text{ако } \forall y(!f(x,y)) \ \& \ \exists y(f(x,y) = 0), \\ \neg!, & \text{в противен случай}; \end{cases}$

г) $\Gamma(f)(x) \simeq \begin{cases} 0, & \text{ако } !f(x), \\ \neg!, & \text{в противен случай.} \end{cases}$

Упътване. а) Γ не е монотонен.
Операторите от примерите б) и в) са монотонни, но не са компактни.

Задача 8. Да се докаже, че съществуват подмножества на $\mathcal{F}_n$, които не притежават горни граници.

Упътване. Разгледайте множество, съдържащо две различни тотални функции от $\mathcal{F}_n$.

Задача 9. Нека $f_0 \subseteq f_1 \subseteq \ldots \subseteq f_k \subseteq \ldots$ е монотонно растяща редица от елементи на $\mathcal{F}_n$ и $G = \{(\overline{x}, y) \mid \exists k(f_k(\overline{x}) \simeq y)\}$. Да се докаже, че G е графика на функцията, която е точна горна граница на редицата $\{f_k\}_{k \in \mathbb{N}}$.

Задача 10. Нека X е множество от n-местни частични функции, което притежава горна граница. Да се докаже, че X има точна горна граница.

Задача 11. Да се докаже, че ако Γ и Δ са компактни оператори съответно от тип (n,k) и (k,m), то и операторът $\Delta \circ \Gamma = \lambda f.\, \Delta(\Gamma(f))$ е компактен.

Доказателство. Покажете, че $\Delta \circ \Gamma$ е монотонен.

Нека $\Delta(\Gamma(f))(\overline{x}) \simeq y$. Ще покажем, че съществува крайна подфункция θ^* на f, за която $\Delta(\Gamma(\theta^*))(\overline{x}) \simeq y$.

Тъй като Δ е компактен, то за някоя крайна подфункция θ на $\Gamma(f)$ е изпълнено $\Delta(\theta)(\overline{x}) \simeq y$.

Ако $\theta = \varnothing^{(k)}$, то нека $\theta^* = \varnothing^{(n)} \subseteq f$. Имаме $\varnothing^{(k)} \subseteq \Gamma(\varnothing^{(n)})$. Тогава $\Delta(\varnothing^{(k)}) \subseteq \Delta(\Gamma(\varnothing^{(n)}))$. Следователно $\Delta(\Gamma(\varnothing^{(n)}))(\overline{x}) \simeq y$.

Нека $\mathrm{dom}(\theta) = \{\overline{z}_1, \ldots, \overline{z}_l\}$ и $\theta(\overline{z}_i) = u_i$. От $\theta \subseteq \Gamma(f)$ следва, че $\Gamma(f)(\overline{z}_i) \simeq u_i$. Използваме, че Γ е компактен. За всяко i, $i = 1, \ldots, l$, съществува крайна функция $\theta_i \subseteq f$, за която $\Gamma(\theta_i)(\overline{z}_i) \simeq u_i$. От зад. 10 следва, че функциите $\theta_1, \dots, \theta_l$ имат точна горна граница $\theta^* \subseteq f$. Ясно е, че θ^* е крайна. Освен това $\theta \subseteq \Gamma(\theta^*)$. От монотонността на Δ получаваме $\Delta(\theta) \subseteq \Delta(\Gamma(\theta^*))$. Следователно $\Delta(\Gamma(\theta^*))(\overline{x}) \simeq y$.

Задача 12. Нека $f_0 \subseteq f_1 \subseteq \dots \subseteq f_k \subseteq \dots$ е монотонно растяща редица от елементи на $\mathcal{F}_n$ и θ е крайна подфункция на $\bigcup f_k$. Да се докаже, че съществува m, за което $\theta \subseteq f_m$.

Задача 13. Нека Δ е монотонен оператор върху множеството на крайните функции, т.е. $\theta_1 \subseteq \theta_2 \Longrightarrow \Delta(\theta_1) \subseteq \Delta(\theta_2)$ за произволни крайни функции θ_1 и θ_2. Да се покаже, че съществува единствен компактен оператор Γ, такъв че $\Gamma(\theta) = \Delta(\theta)$ за всяка крайна функция θ.

Упътване. Разгледайте оператора

$$\Gamma(f)(\overline{x}) \simeq y \iff \exists\theta(\theta \subseteq f \ \&\ \Delta(\theta)(\overline{x}) \simeq y).$$

Задача 14. Да се докаже, че всеки компактен оператор е непрекъснат.

Упътване. Нека Γ е компактен оператор от тип (n,r) и $f_0 \subseteq f_1 \subseteq \dots \subseteq f_k \subseteq \dots$ е монотонно растяща редица от елементи на $\mathcal{F}_n$.

Тъй като Γ е монотонен, $\Gamma(f_k) \subseteq \Gamma(\bigcup f_k)$ за всяко k, т.е. $\Gamma(\bigcup f_k)$ е горна граница на редицата $\{\Gamma(f_k)\}_{k\in\mathbb{N}}$, откъдето $\bigcup \Gamma(f_k) \subseteq \Gamma(\bigcup f_k)$.

За обратната посока на последното неравенство използвайте, че Γ е компактен и зад. 12.

Задача 15. Да се докаже, че всеки непрекъснат оператор е компактен.

Упътване. Нека Γ е непрекъснат оператор от тип (n,r) и $f \subseteq g$, f, $g \in \mathcal{F}_n$. За да покажете, че Γ е монотонен, разгледайте редицата

$$f \subseteq g \subseteq \dots \subseteq g \subseteq \dots ,$$

чиято точна горна граница е g. Използвайте непрекъснатостта на Γ, за да покажете, че $\Gamma(f) \subseteq \Gamma(g)$.

Нека $f \in \mathcal{F}_n$. Според зад. 5, достатъчно е да покажете, че условие (2) е налице. За целта разгледайте монотонно растящата редица

$$\theta_0 \subseteq \theta_1 \subseteq \dots \subseteq \theta_k \subseteq \dots,$$

където θ_k е рестрикцията на f върху множеството $\{0, \cdots, k\}$. Съобразете, че f е точна горна граница на тази редица, откъдето ще следва, че $\Gamma(f) = \Gamma(\bigcup \theta_k) = \bigcup \Gamma(\theta_k)$.

Сега ако $\Gamma(f)(\overline{x}) \simeq y$, то $(\bigcup \Gamma(\theta_k))(\overline{x}) \simeq y$ и следователно $\Gamma(\theta_k)(\overline{x}) \simeq y$ за някое k.

Задача 16. Да се намерят неподвижните точки на изброените оператори.

а) $\Gamma(f) = f$ от тип (n,n);

б) $\Gamma(f)(\overline{x}) \simeq g(\overline{x})$, където g е фиксирана функция;

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

г) $\Gamma(f)(x) \simeq \begin{cases} 1, & \text{ако } x = 0, \\ 2.f(x-1), & \text{ако } x > 0; \end{cases}$

д) $\Gamma(f)(x) \simeq \begin{cases} 0, & \text{ако } x = 0, \\ 2x-1+f(x-1), & \text{ако } x > 0; \end{cases}$

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

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

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

Упътване. а) Всяка функция $f \in \mathcal{F}_n$ е неподвижна точка на Γ;

б) Γ има единствена неподвижна точка — функцията g;

в) Покажете, че единствената неподвижна точка на Γ е $\lambda x.x!$;

г) $f_\Gamma = \lambda x.2^x$ и тя е единствена;

д) Използвайте тъждеството

$$1 + 3 + \dots + 2x - 1 = x^2.$$

Функцията $f = \lambda x.x^2$ е неподвижна точка на Γ, тъй като за f е в сила

$$x^2 \simeq \begin{cases} 0, & \text{ако } x = 0, \\ 2x-1+(x-1)^2, & \text{ако } x > 0. \end{cases}$$

След това с индукция по x докажете, че ако

$$g(x) \simeq \begin{cases} 0, & \text{ако } x = 0, \\ 2x-1+g(x-1), & \text{ако } x > 0, \end{cases}$$

то $g(x) = f(x)$ за всяко x.

е) $f_\Gamma = \lambda x.2^x$. Използвайте тъждеството

$$2^x = \begin{cases} 1, & \text{ако } x = 0,\\ \left(2^{x/2}\right)^2, & \text{ако } x \neq 0 \text{ и } x \text{ е четно},\\ 2.\left(2^{(x-1)/2}\right)^2, & \text{ако } x \text{ е нечетно},\end{cases}$$

на което се основава двоичният алгоритъм за пресмятане на 2^x.

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

Забележете, че ако f е неподвижна точка на Γ, то

$$f(x) \simeq f(x+1) \text{ за } x > 0.$$

Съобразете, че всяка неподвижна точка f на Γ, която е различна от f_Γ, има вида:

$$f(x) = \begin{cases} 0, & \text{ако } x = 0,\\ a, & \text{ако } x > 0,\end{cases}$$

за някое фиксирано a.

з) Ако f е неподвижна точка на Γ, то

$$x \geqq y \Longrightarrow f(x,y) \simeq f(x,y+1) \simeq \cdots \simeq f(x,x) \simeq 0 \text{ и}$$

$$x < y \Longrightarrow f(x,y) \simeq f(x,y+1) \simeq \cdots .$$

Да положим

$$f(x,y) \simeq \begin{cases} 0, & \text{ако } x \geqq y,\\ \neg!, & \text{ако } x < y.\end{cases}$$

Ще покажем, че $\Gamma(f) = f$. Разглеждаме следните три случая.

Ако $x = y$, то $\Gamma(f)(x,y) = 0 = f(x,y)$.

Ако $x > y$, то $\Gamma(f)(x,y) \simeq f(x,y+1)$, но $x \geqq y+1$ и имаме $f(x,y+1) = 0 = f(x,y)$.

Ако $x < y$, то $\Gamma(f)(x,y) \simeq f(x,y+1)$, и тъй като $x < y+1$, то $f(x,y)$ и $f(x,y+1)$ са недефинирани.

Нека g е произволна неподвижна точка на Γ. Ще покажем, че $f \subseteq g$.

Ако $f(x,y) \simeq z$, то от избора на f следва, че $x \geqq y$ и $z = 0$. В този случай $g(x,y) \simeq \cdots \simeq g(x,x) \simeq 0 = z$. Следователно $f \subseteq g$.

Проверете, че всички неподвижни точки на Γ са функции от вида

$$g(x,y) \simeq \begin{cases} 0, & \text{ако } x \geqq y,\\ h(x), & \text{ако } x < y,\end{cases}$$

за някоя фиксирана функция h.

Задача 17. Да се докаже, че единствената неподвижна точка на оператора

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

е функцията

$$h(x,y) \simeq \begin{cases} x - y, & \text{ако } x \geqq y,\\ \neg!, & \text{ако } x < y.\end{cases}$$

Упътване. Проверете, че h е неподвижна точка. Нека $\Gamma(g) = g$. Ще покажем, че $h = g$. С индукция по u покажете, че свойството $P(u)$, дефинирано с

$$P(u) \iff \forall x \forall y (x \geqq y \,\&\, u = x - y \Longrightarrow g(x,y) \simeq x - y),$$

е вярно за всяко естествено число u.

Нека $x < y$. Допускаме, че $!g(x,y)$ и $g(x,y) = z$. Тогава от дефиницията на Γ следва, че за всяко m $!g(x,y+m)$ и $g(x,y) = g(x,y+m)+m$. Следователно $z = g(x,y) = g(x,y+z+1)+z+1$, което е невъзможно.

Задача 18. Нека Γ е монотонен оператор от тип (n,n) и f е най-малкото решение на неравенството $\Gamma(\mathbb{X}) \subseteq \mathbb{X}$, т. е. $\Gamma(f) \subseteq f$ и $\forall g(\Gamma(g) \subseteq g \Longrightarrow f \subseteq g)$. Да се докаже, че f е най-малката неподвижна точка на Γ.

Задача 19. Нека Γ е компактен оператор от тип (n,n). Да се докаже, че Γ има най-малка неподвижна точка, без да се използва теоремата на Кнастер — Тарски. Като следствие да се изведе теоремата на Кнастер — Тарски.

Доказателство. Разглеждаме редицата

$$f_0 = \emptyset^{(n)}, f_1 = \Gamma(f_0), \ldots, f_{k+1} = \Gamma(f_k), \ldots$$

От монотонността на Γ следва, че редицата $\{f_k\}$ е монотонно растяща. Действително $f_0 = \emptyset^{(n)} \subseteq f_1$ и

$$f_k \subseteq f_{k+1} \Longrightarrow \Gamma(f_k) \subseteq \Gamma(f_{k+1}) \Longrightarrow f_{k+1} \subseteq f_{k+2}.$$

Полагаме $f = \bigcup f_k$. Ще покажем, че f е най-малката неподвижна точка на Γ.

Имаме

$$\begin{aligned} f(\overline{x}) \simeq y &\Longrightarrow \exists k (f_k(\overline{x}) \simeq y) \text{ (по определението на } f)\\ &\Longrightarrow \exists k (k > 0 \,\&\, \Gamma(f_{k-1})(\overline{x}) \simeq y) \text{ (по определението на } \{f_k\})\\ &\Longrightarrow \Gamma(f)(\overline{x}) \simeq y \text{ (от монотонността на } \Gamma).\end{aligned}$$

Обратно

$$
\begin{aligned}
\Gamma(f)(\overline{x}) \simeq y &\Longrightarrow \exists\theta \subseteq f(\theta \text{ е крайна} \,\&\, \Gamma(\theta)(\overline{x}) \simeq y) \text{ (}\Gamma\text{ е компактен)}\\
&\Longrightarrow \exists\theta\exists k(\theta \subseteq f_k \,\&\, \theta \text{ е крайна } \&\, \Gamma(\theta)(\overline{x}) \simeq y) \text{ (от зад. 12)}\\
&\Longrightarrow \exists k(f_{k+1}(\overline{x}) \simeq y) \text{ (}\Gamma\text{ е монотонен)} \Longrightarrow f(\overline{x}) \simeq y.
\end{aligned}
$$

Следователно $\Gamma(f)=f$. За да покажем, че f е най-малка неподвижна точка, да предположим, че $\Gamma(g)=g$. Ясно е, че $f_0=\varnothing^{(n)} \subseteq g$. Ако $f_k \subseteq g$, то от монотонността на Γ следва, че $f_{k+1} = \Gamma(f_k) \subseteq \Gamma(g) = g$. Така $f_k \subseteq g$ за всяко k. Получаваме, че g е горна граница за редицата $\{f_k\}$. Но f е точна горна граница за $\{f_k\}$ и следователно $f \subseteq g$.

Тъй като всеки непрекъснат оператор е компактен (зад. 15), то теоремата на Кнастер — Тарски следва непосредствено от показаното по-горе.

Забележка. Доказателството на теоремата на Кнастер — Тарски дава един общ метод за построяване на най-малките неподвижни точки на компактните (непрекъснатите) оператори.

Нека Γ е компактен оператор от тип (n,n). За да построим най-малката неподвижна точка на Γ, първо намираме явния вид на всяка от функциите

$$f_0 = \varnothing^{(n)}, \quad f_1 = \Gamma(f_0), \ldots, \quad f_{k+1} = \Gamma(f_k), \ldots$$

След това използваме, че най-малката неподвижна точка на Γ е $\bigcup f_k$. Този метод ще наричаме за краткост метод на Кнастер — Тарски.

Задача 20. Да се докаже, че най-малките неподвижни точки на операторите от зад. 16 съществуват. С метода на Кнастер — Тарски да се намери явният им вид.

Упътване. За да покажете, че всеки от операторите има най-малка неподвижна точка, е достатъчно да проверите, че той е компактен или непрекъснат.

а) Функцията $\varnothing^{(n)}$ е най-малката неподвижна точка на Γ.

д) Проверяваме, че Γ е компактен. Следователно най-малката неподвижна точка f_Γ на Γ съществува и се получава като точна горна граница на редицата

$$f_0 = \varnothing^{(1)}, \ f_1 = \Gamma(f_0), \ldots, \ f_{k+1} = \Gamma(f_k), \ldots$$

С индукция по k ще покажем, че

$$(*) \qquad f_k(x) \simeq \begin{cases} x^2, & \text{ако } x < k,\\ \neg!, & \text{ако } x \geqq k.\end{cases}$$

Наистина, за $k=0$ условното равенство $(*)$ е налице. Нека $(*)$ е вярно за k. Тогава

$$f_{k+1}(x) \simeq \begin{cases} 0, & \text{ако } x = 0,\\ 2x - 1 + f_k(x-1), & \text{ако } x > 0,\end{cases}$$

$$f_{k+1}(x) \simeq \begin{cases} 0, & \text{ако } x = 0,\\ 2x - 1 + (x-1)^2, & \text{ако } 0 \leqq x - 1 < k,\\ \neg!, & \text{ако } x - 1 \geqq k,\end{cases}$$

$$f_{k+1}(x) \simeq \begin{cases} x^2, & \text{ако } x < k+1,\\ \neg!, & \text{ако } x \geqq k+1.\end{cases}$$

Оттук, използвайки, че $(*)$ е вярно за всяко k и

$$(\bigcup f_k)(x) \simeq y \iff \exists k(f_k(x) \simeq y),$$

получаваме, че $f_\Gamma(x) = x^2$ за всяко x.

ж) Нека $f_0 = \varnothing^{(1)}$ и $f_{k+1} = \Gamma(f_k)$ за всяко k. Тогава при $k > 0$

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

Следователно

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

з) С индукция по k се показва, че за $k > 0$

$$f_k(x,y) \simeq \begin{cases} 0, & \text{ако } x = y \vee \cdots \vee x = y + k - 1,\\ \neg!, & \text{в останалите случаи.}\end{cases}$$

Тогава

$$f_\Gamma(x,y) \simeq \begin{cases} 0, & \text{ако } x \geqq y,\\ \neg!, & \text{ако } x < y.\end{cases}$$

Нека $f : \mathbb{N}^{n+1} \twoheadrightarrow \mathbb{N}$, $n \geqq 1$. Казваме, че g се получава с операцията *минимизация* (или с *μ-операция*) от f (и записваме $g(\overline{x}) \simeq \mu y[f(\overline{x}, y) \simeq 0]$), ако за g е изпълнено условието:

$$g(\overline{x}) \simeq y \Longleftrightarrow f(\overline{x}, y) \simeq 0 \,\&\, \forall z(z < y \Longrightarrow !f(\overline{x}, z) \,\&\, f(\overline{x}, z) > 0).$$

От определението на функцията g се вижда, че ако $!g(\overline{x})$, то $g(\overline{x})$ е най-малкото y, за което $f(\overline{x}, y) \simeq 0$, но при условие, че $f(\overline{x}, z)$ е дефинирана за всяко $z < y$.

Задача 21. Нека h е частична функция на $n+1$ аргумента. Да се докаже, че най-малката неподвижна точка на оператора:

$$\text{а) } \Gamma(f)(\overline{x}, y) \simeq \begin{cases} y, & \text{ако } h(\overline{x}, y) \simeq 0, \\ f(\overline{x}, y+1), & \text{ако } h(\overline{x}, y) > 0, \\ \neg!, & \text{ако } \neg!h(\overline{x}, y) \end{cases}$$

е функцията $\lambda\overline{x}, y.\mu z_{\geqq y}[h(\overline{x}, z) \simeq 0]$;

$$\text{б) } \Delta(f)(\overline{x}, y) \simeq \begin{cases} 0, & \text{ако } h(\overline{x}, y) \simeq 0, \\ f(\overline{x}, y+1)+1, & \text{ако } h(\overline{x}, y) > 0, \\ \neg!, & \text{ако } \neg!h(\overline{x}; y) \end{cases}$$

е функцията $\lambda\overline{x}, y.\mu z[h(\overline{x}, z+y) \simeq 0]$.

Да се изрази операцията минимизация с помощта на най-малките неподвижни точки на операторите Γ и Δ.

Упътване. а) I *начин.* Докажете, че при $k > 0$

$$f_k(\overline{x}, y) \simeq \begin{cases} y, & \text{ако } h(\overline{x}, y) \simeq 0, \\ y+1, & \text{ако } h(\overline{x}, y) > 0 \,\&\, h(\overline{x}, y+1) \simeq 0, \\ \dots, & \\ y+k-1, & \text{ако } h(\overline{x}, y) > 0 \,\&\, \cdots \,\&\, h(\overline{x}, y+k-2) > 0 \,\&\, \\ & \quad h(\overline{x}, y+k-1) \simeq 0, \\ \neg!, & \text{в останалите случаи,} \end{cases}$$

или

$$f_k(\overline{x}, y) \simeq \begin{cases} \mu z_{\geqq y}[z < y+k \,\&\, h(\overline{x}, z) \simeq 0], & \text{ако } \exists z(y \leqq z < y+k \,\&\, h(\overline{x}, z) \simeq 0 \\ & \quad \&\, \forall u(y \leqq u < z \Longrightarrow h(\overline{x}, u) > 0)) \\ \neg!, & \text{в останалите случаи.} \end{cases}$$

II *начин.* Проверете, че $f(\overline{x}, y) \simeq \mu z_{\geqq y}[h(\overline{x}, z) \simeq 0]$ е неподвижна точка на Γ като за целта разгледате случаите:

1. $\neg!h(\overline{x}, y)$;
2. $h(\overline{x}, y) \simeq 0$;
3. $h(\overline{x}, y) > 0$. Тук съобразете, че
$$f(\overline{x}, y) \simeq \mu z_{\geqq y+1}[h(\overline{x}, z) \simeq 0] \simeq f(\overline{x}, y+1).$$

За да докажете, че f е най-малката неподвижна точка, може да разсъждавате така. Нека $\Gamma(g) = g$. Да забележим, че ако $f(\overline{x}, y) \simeq z$, то от дефиницията на f следва, че $z \geqq y$. Покажете, че наредбата

$$(y, z) \prec (y', z') \iff 0 \leqq (z-y) < (z'-y')$$

е фундирана. Със структурна индукция по тази наредба докажете, че

$$f(\overline{x}, y) \simeq z \Longrightarrow g(\overline{x}, y) \simeq z.$$

Нека $f(\overline{x}, y) \simeq z$.

Ако $z - y = 0$, то $z = y$ и $h(\overline{x}, y) \simeq 0$. Тогава $g(\overline{x}, y) = y = z$.

Нека $z - y > 0$ и за всяко $(y', z') \prec (y, z)$ е вярно предположението $f(\overline{x}, y') \simeq z' \Longrightarrow g(\overline{x}, y') \simeq z'$. Имаме $z > y$. Следователно $h(\overline{x}, y) > 0$ и $f(\overline{x}, y) = f(\overline{x}, y+1) = z$. Тъй като $0 \leqq z - (y+1) < z - y$, то по индукционното предположение $f(\overline{x}, y+1) \simeq g(\overline{x}, y+1) \simeq \Gamma(g)(\overline{x}, y) \simeq g(\overline{x}, y) = z$. Така $f \subseteq g$.

Забележете, че $f(\overline{x}, 0) \simeq \mu z[h(\overline{x}, z) \simeq 0]$.

Задача 22. Да се докаже, че операторът

$$\Gamma(f)(x, y) \simeq \begin{cases} y, & \text{ако } f(x, y) \simeq 0, \\ f(x, y+1), & \text{ако } f(x, y) > 0, \\ \neg! & \text{ако } \neg!f(x, y) \end{cases}$$

има безброй много неподвижни точки.

Задача 23. Даден е операторът

$$\Gamma(f)(x, y) \simeq \begin{cases} x+y, & \text{ако } x = 0 \text{ или } y = 0, \\ f(x-1, f(x, y-1)), & \text{в противен случай.} \end{cases}$$

Да се провери дали функциите:

$$\text{а) } g(x, y) \simeq \begin{cases} x+y, & \text{ако } x = 0 \text{ или } y = 0, \\ \neg!, & \text{в противен случай;} \end{cases}$$

$$\text{б) } h(x, y) \simeq \begin{cases} x+y, & \text{ако } x = 0 \text{ или } y = 0, \\ 1, & \text{в противен случай;} \end{cases}$$

са неподвижни точки на оператора Γ.

Да се докаже, че Γ има най-малка неподвижна точка и да се намери явният ѝ вид.

Упътване. Функцията g не е неподвижна точка на Γ.

Задача 24. Да се докаже, че операторите

$$\Gamma(f)(x) \simeq \begin{cases} 0, & \text{ако } \exists n(x = 2^n), \\ f(x+1)+1, & \text{в противен случай,} \end{cases}$$

$$\Delta(f)(x, y) \simeq \begin{cases} y, & \text{ако } x = 0, \\ f(x, y-x), & \text{ако } 0 < x \leqq y, \\ f(y, x), & \text{ако } y < x \end{cases}$$

имат най-малки неподвижни точки и да се намери явният им вид.

Задача 25. Дадени са операторите:

а) $\Gamma(f)(x,y) \simeq \begin{cases} 0, & \text{ако } x = 0\ , \\ f(x-1, f(x,y)), & \text{в противен случай;} \end{cases}$

б) $\Gamma(f)(x,y) \simeq \begin{cases} y, & \text{ако } f(x,y) \simeq 0, \\ f(x, y+1), & \text{ако } f(x,y) > 0, \\ \neg! & \text{ако } \neg! f(x,y). \end{cases}$

Да се докаже, че дадените оператори имат повече от една неподвижна точка.

Да се провери, че те имат най-малки неподвижни точки и да се намери явният им вид.

Упътване. а) Нека $f_0 = \varnothing^{(2)}$ и $f_{k+1} = \Gamma(f_k)$ за всяко k. Тогава за $k > 0$ имаме

$$f_k(x,y) \simeq \begin{cases} 0, & \text{ако } x = 0, \\ \neg!, & \text{в противен случай.} \end{cases}$$

Задача 26. Да се докаже, че операторът

$$\Gamma(f)(x,y) \simeq \begin{cases} 0, & \text{ако } x < y, \\ f(x-y, y) + 1, & \text{в противен случай} \end{cases}$$

има най-малка неподвижна точка и да се намери явният ѝ вид. Има ли Γ други неподвижни точки?

Задача 27. Да се даде пример за оператор, който

а) няма неподвижни точки;

б) има неподвижни точки, но няма най-малка неподвижна точка;

в) има неподвижни точки, но няма неподвижни точки, които са тотални функции;

г) има неподвижна точка, която е тотална функция, но най-малката му неподвижна точка не е тотална;

д) има безброй много неподвижни точки;

е) има точно три неподвижни точки.

Задача 28. Нека g и h са фиксирани несравними едноместни функции, т.е. $g \not\subseteq h \ \& \ h \not\subseteq g$, които са различни от $\varnothing^{(1)}$. Да се докаже, че операторът

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

няма най-малка неподвижна точка.

Нека $n_i \geq 1$, $i = 1, \ldots, r$ са фиксирани естествени числа. Всяко изображение $\Gamma : \mathcal{F}_{n_1} \times \ldots \times \mathcal{F}_{n_r} \to \mathcal{F}_n$, $n \geq 1$, се нарича оператор от тип $(n_1, \ldots, n_r \to n)$.

Нека Γ е оператор от тип $(n_1, \ldots, n_r \to n)$. Казваме, че Γ е *монотонен*, ако при всеки избор на $f_1, g_1 \in \mathcal{F}_{n_1}, \ldots, f_r, g_r \in \mathcal{F}_{n_r}$

$$f_1 \subseteq g_1 \ \& \ \ldots \ \& \ f_r \subseteq g_r \Longrightarrow \Gamma(f_1, \ldots, f_r) \subseteq \Gamma(g_1, \ldots, g_r).$$

Операторът Γ е *компактен*, ако при всеки избор на $f_1 \in \mathcal{F}_{n_1}, \ldots, f_r \in \mathcal{F}_{n_r}$, $\overline{x} \in \mathbb{N}^n$ и $y \in \mathbb{N}$ е в сила еквивалентността

$\Gamma(f_1, \ldots, f_r)(\overline{x}) \simeq y \iff$

$$\exists \theta_1 \ldots \exists \theta_r [\theta_i \text{ е крайна подфункция на } f_i \text{ и } \Gamma(\theta_1, \ldots, \theta_r)(\overline{x}) \simeq y].$$

Нека $\Gamma_1, \ldots, \Gamma_r$ са оператори, които са от тип $(n_1, \ldots, n_r \to n_1), \ldots, (n_1, \ldots, n_r \to n_r)$ съответно. *Решение* на системата

(3) $\qquad \Gamma_i(\mathbb{X}_1, \ldots, \mathbb{X}_r) = \mathbb{X}_i, \ i = 1, \ldots, r,$

наричаме такива функции $f_1 \in \mathcal{F}_{n_1}, \ldots, f_r \in \mathcal{F}_{n_r}$, за които

$$\left| \begin{array}{l} \Gamma_1(f_1, \ldots, f_r) = f_1, \\ \ldots \\ \Gamma_r(f_1, \ldots, f_r) = f_r. \end{array} \right.$$

Най-малко решение на (3) наричаме такова решение $f_1, \ldots, f_r$ на (3), за което е изпълнено: ако $g_1, \ldots, g_r$ е решение на (3), то $f_1 \subseteq g_1, \ldots, f_r \subseteq g_r$.

Задача 29. Да се докаже, че всеки многоместен компактен оператор е монотонен.

Задача 30. Да се покаже, че всяко компактно изображение $\Gamma : P(\mathbb{N}) \to P(\mathbb{N})$ е монотонно и изброимо непрекъснато, т.е. удовлетворява равенството $\Gamma(\bigcup A_k) = \bigcup \Gamma(A_k)$ при всеки избор на монотонно растяща редица $A_0 \subseteq A_1 \subseteq \ldots \subseteq A_k \subseteq \ldots$ от подмножества на $\mathbb{N}$.

Задача 31. Нека $\Gamma_1, \ldots, \Gamma_r$ са компактни оператори съответно от тип $(n_1, \ldots, n_r \to n_1), \ldots, (n_1, \ldots, n_r \to n_r)$. Да се докаже, че съществува най-малко решение $f_1, \ldots, f_r$ на системата

$$\Gamma_i(\mathbb{X}_1, \ldots, \mathbb{X}_r) = \mathbb{X}_i, \ i = 1, \ldots, r.$$

Задача 32. Да се докаже, че следните многоместни оператори са компактни:

а) $\Gamma(f, g)(x) \simeq f(g(x))$ — композиция на функции;

б) $\Gamma(g, f_1, \ldots, f_k)(\overline{x}) \simeq g(f_1(\overline{x}), \ldots, f_k(\overline{x}))$ — суперпозиция на функции;

в) $\Gamma(f, g)(\overline{x}) \simeq$ if $f(\overline{x}) \simeq 0$ then $g(\overline{x})$;

г) $\Gamma(f, g, h)(\overline{x}) \simeq$ if $f(\overline{x}) \simeq 0$ then $g(\overline{x})$ else $h(\overline{x})$;

д) $\Gamma(f, g)(x) \simeq [f, g](x)$ — итерация, където

$$[f,g](x) \simeq y \iff \exists x_0 \exists x_1 \cdots \exists x_n (x_0 = x \;\&\; x_n = y \;\&\; g(y) \simeq 0 \;\& \\ \forall i_{<n}(x_{i+1} \simeq f(x_i) \;\&\; g(x_i) > 0)).$$

Задача 33. Нека f и g са едноместни частични функции. Да се докаже, че най-малкото решение на уравнението

$$\chi(x) \simeq \texttt{ if } g(x) \simeq 0 \texttt{ then } x \texttt{ else } \chi(f(x))$$

е функцията $\chi = [f, g]$.

Упътване. Докажете, че $[f, g]$ е най-малката неподвижна точка на оператора

$$\Gamma(\chi)(x) \simeq \texttt{ if } g(x) \simeq 0 \texttt{ then } x \texttt{ else } \chi(f(x)).$$

Задача 34. Да се докаже, че следните системи имат най-малко решение:

а) $\left| \begin{array}{l} f_1(x) \simeq \texttt{ if } x = 0 \texttt{ then } 1 \texttt{ else } f_1(x-1) + f_2(x-1) \\ f_2(x) \simeq \texttt{ if } x = 0 \texttt{ then } 1 \texttt{ else } f_2(x+1); \end{array} \right.$

б) $\left| \begin{array}{l} f_1(x) \simeq \texttt{ if } x = 1 \texttt{ then } 0 \texttt{ else } f_1(f_2(x)) \\ f_2(x) \simeq \texttt{ if } x = 1 \texttt{ then } 1 \\ \qquad\qquad \texttt{else if } x \text{ е четно } \texttt{then } x/2 \\ \qquad\qquad\qquad\qquad \texttt{else } f_2(f_2((3x+1)/2)). \end{array} \right.$

§ 2.3. Индукционно правило на Скот

Ще разгледаме едно правило за доказване на свойства на най-малките неподвижни точки на непрекъснатите (компактните) оператори, известно като индукционно правило на Скот. Забележителното в него е, че то не изисква намирането на явния вид на най-малката неподвижна точка на оператора, което в някои случаи е доста трудно.

Нека Γ е компактен оператор от тип (n, n). С f_Γ ще означаваме най-малката неподвижна точка на Γ. От теоремата на Кнастер — Тарски знаем, че $f_\Gamma = \bigcup f_k$, където $f_0 = \varnothing^{(n)}$ и $f_{k+1} = \Gamma(f_k)$.

Нека P е някакво свойство на n-местните частични функции, което желаем да докажем за f_Γ. Да предположим, че сме доказали $P(\varnothing^{(n)})$ и $\forall f \in \mathcal{F}_n(P(f) \Longrightarrow P(\Gamma(f)))$. Тогава свойството P е изпълнено за всяка от функциите f_k. За да бъде вярно и за най-малката неподвижна точка на Γ, достатъчно е P да бъде изброимо непрекъснато.

Казваме, че свойството P в $\mathcal{F}_n$ е *(изброимо) непрекъснато*, ако при всеки избор на монотонно растяща редица $f_0 \subseteq f_1 \subseteq \ldots \subseteq f_k \subseteq \ldots$ от елементи на $\mathcal{F}_n$, такава че $P(f_k)$ за всяко k, е в сила $P(\bigcup f_k)$.

Индукционно правило на Скот. *Нека Γ е непрекъснат (компактен) оператор от тип (n, n) и P е непрекъснато свойство в $\mathcal{F}_n$. Нека предположим, че $P(\varnothing)$ и $\forall f(P(f) \Longrightarrow P(\Gamma(f)))$. Тогава P е в сила за най-малката неподвижна точка на Γ.*

Най-напред ще разгледаме някои достатъчни условия за непрекъснатост на дадено свойство.

Всяко свойство P, което е от вида

$$P(f) \iff \forall \overline{x}(!f(\overline{x}) \;\&\; I(\overline{x}) \Longrightarrow O(\overline{x}, f(\overline{x}))),$$

за някои предикати $\lambda\overline{x}.I(\overline{x})$ и $\lambda\overline{x}, y.O(\overline{x}, y)$ в множеството на естествените числа, наричаме *условие за частична коректност*.

Нека f е частична функция, изчислима с дадена програма. Свойството $P(f)$ означава, че всеки път, когато входните данни удовлетворяват предиката I и програмата завършва, резултатът удовлетворява предиката O. Сравнете това понятие с въведеното в § 1.2 понятие за частична коректност на програми.

Условие за тотална коректност наричаме всяко свойство от вида

$$P(f) \iff \forall \overline{x}(I(\overline{x}) \Longrightarrow !f(\overline{x}) \;\&\; O(\overline{x}, f(\overline{x}))).$$

Така, ако f е изчислима с дадена програма, свойството $P(f)$ изразява факта, че при всеки вход, удовлетворяващ предиката I, тази програма за f завършва с изход, който удовлетворява предиката O.

Задача 1. Нека P е условие за частична коректност. Да се провери, че свойството P е непрекъснато.

Задача 2. Да се докаже, че всяко условие за тотална коректност е непрекъснато. Обяснете защо индукционното правило на Скот е неприложимо при доказване на условия за тотална коректност.

Доказателство. Нека

$$P(f) \iff \forall \overline{x}(I(\overline{x}) \Longrightarrow !f(\overline{x}) \;\&\; O(\overline{x}, f(\overline{x}))).$$

Разглеждаме $f_0 \subseteq f_1 \subseteq \dots \subseteq f_k \subseteq \dots$ — монотонно растяща редица от n-местни частични функции, и полагаме $f = \bigcup f_k$. Да предположим, че за всяко k е в сила $P(f_k)$. Нека $I(\overline{x})$ е вярно. Тъй като е изпълнено $P(f_k)$, то $!f_k(\overline{x})$ и $O(\overline{x}, f_k(\overline{x}))$ за $k \in \mathbb{N}$. Но $f(\overline{x}) \simeq y \iff \exists k(f_k(\overline{x}) \simeq y)$. Ясно е тогава, че $!f(\overline{x})$ и $O(\overline{x}, f(\overline{x}))$.

Индукционното правило на Скот не е приложимо към условие за тотална коректност, тъй като такова условие не е вярно за никъде недефинираната функция (стига I да не е тоталната лъжа).

Задача 3. Да се докаже, че свойството, определено с еквивалентността

а) $P(f) \iff \forall \overline{x}(\neg !f(\overline{x}))$

е непрекъснато;

б) $Q(f) \iff \exists \overline{x}(\neg !f(\overline{x}))$

не е непрекъснато.

Упътване. б) Разгледайте редицата $f_0, f_1, \dots, f_k, \dots$, дефинирана с

$$f_k(x) \simeq \begin{cases} x, & \text{ако } x < k, \\ \neg !, & \text{ако } x \geqq k. \end{cases}$$

Тогава $(\bigcup f_k)(x) = x$ за всяко x.

Ясно е, че $\forall k(Q(f_k))$, но $\neg Q(\bigcup f_k)$.

Задача 4. Нека P_1 и P_2 са непрекъснати свойства. Да се докаже, че свойството, определено чрез еквивалентността

а) $P(f) \iff P_1(f) \;\&\; P_2(f)$,

също е непрекъснато;

б) $Q(f) \iff \neg P_1(f)$,

не винаги е непрекъснато.

Непрекъснато ли е свойството R, където

$$R(f) \iff P_1(f) \;\vee\; P_2(f)?$$

Упътване. б) Разгледайте например свойството

$$P_1(f) \iff \forall \overline{x}(!f(\overline{x})).$$

То е непрекъснато, но $\neg P_1$ е точно свойството от зад. 3 б).

Задача 5. Да се покаже, че ако Γ_1 и Γ_2 са непрекъснати оператори, то свойствата

а) $P(f) \iff \Gamma_1(f) \subseteq \Gamma_2(f)$;

б) $Q(f) \iff \Gamma_1(f) = \Gamma_2(f)$

са непрекъснати.

Упътване. а) Ако $f_0 \subseteq f_1 \subseteq \dots \subseteq f_k \subseteq \dots$ и $\Gamma_1(f_k) \subseteq \Gamma_2(f_k)$ за всяко k, то

$$\Gamma_1(\bigcup f_k) = \bigcup \Gamma_1(f_k) \subseteq \bigcup \Gamma_2(f_k) = \Gamma_2(\bigcup f_k).$$

б) Директно следва от а) и зад. 4 а).

Задача 6. Нека Γ е оператор, определен с равенството

$$\Gamma(f)(x,y) \simeq \texttt{if } x < y \texttt{ then } x \texttt{ else } f(x-y,y).$$

Да се докаже, че $\forall x \forall y(!f_\Gamma(x,y) \Longrightarrow f_\Gamma(x,y) < y)$.

Доказателство. Разглеждаме условието за частична коректност $P(f)$, определено чрез еквивалентността

$$P(f) \iff \forall x \forall y(!f(x,y) \Longrightarrow f(x,y) < y).$$

От зад. 1 знаем, че P е непрекъснато.

Ясно е, че $P(\varnothing^{(2)})$, тъй като $\neg !\varnothing^{(2)}(x,y)$ за всяко x, y.

Да предположим, че $P(f)$ е вярно за произволна $f \in \mathcal{F}_2$, и да разгледаме $\Gamma(f)$. Ще покажем, че $P(\Gamma(f))$ също е налице.

Нека $!\Gamma(f)(x,y)$. Искаме да проверим, че $\Gamma(f)(x,y) < y$. Имаме два случая:

1. $x < y$. Тогава $\Gamma(f)(x,y) \simeq x$. Следователно $\Gamma(f)(x,y) < y$.
2. $x \geqq y$. Имаме $\Gamma(f)(x,y) \simeq f(x-y,y)$. Следователно $!f(x-y,y)$, и от $P(f)$ получаваме $f(x-y,y) < y$. Така $\Gamma(f)(x,y) < y$.

Оттук съгласно правилото на Скот имаме $P(f_\Gamma)$.

Задача 7. Даден е операторът

$$\Gamma(f)(x) \simeq \texttt{if } x = 0 \texttt{ then } 1 \texttt{ else } f(x-1).x.$$

Да се докаже, че:

а) $\forall x \in \mathrm{dom}(f_\Gamma)(f_\Gamma(x) \simeq x!)$;

б) $\forall x(f_\Gamma(x) = x!)$.

Упътване. а) Разгледайте условието за частична коректност $P(f)$, определено чрез еквивалентността

$$P(f) \iff \forall x(!f(x) \Longrightarrow f(x) \simeq x!).$$

б) От а) имаме $f_\Gamma \subseteq \lambda x.x!$. Следователно трябва да докажем, че f_Γ е тотална. Както знаем от зад. 2, този проблем не може да се реши с помощта на правилото на Скот. Затова покажете, че $\forall x(!f_\Gamma(x))$ с обикновена индукция по x.

Задача 8. Операторът Γ е определен с равенството

$$\Gamma(f)(x) \simeq \texttt{if } x > 100 \texttt{ then } x - 10 \texttt{ else } f(f(x+11)).$$

Да се докаже, че

а) $\forall x(!f_\Gamma(x) \ \&\ x \leqq 101 \Longrightarrow f_\Gamma(x) \simeq 91)$.

б) f_Γ е навсякъде дефинирана.

Забележка. Сравнете със зад. 12, § 1.1.

Задача 9. Даден е операторът

$$\Gamma(f)(x,y) \simeq \texttt{if } x = 0 \texttt{ then } y \texttt{ else } f(x+1,y) \dot{-} 1.$$

Да се докаже, че $\forall x \forall y(f_\Gamma(x,y) \dot{-} 1 \simeq f_\Gamma(x, y \dot{-} 1))$.

Доказателство. Разглеждаме свойството

$$P(f) \iff \forall x \forall y(f(x,y) \dot{-} 1 \simeq f(x, y \dot{-} 1)).$$

Нека $\Gamma_1(f) = \lambda x,y.f(x,y) \dot{-} 1$ и $\Gamma_2(f) = \lambda x,y.f(x, y \dot{-} 1)$. Операторите Γ_1 и Γ_2 са компактни и следователно непрекъснати. Да забележим, че $P(f) \iff \Gamma_1(f) = \Gamma_2(f)$. Тогава от зад. 5 б) имаме, че P също е непрекъснато.

Свойството P е вярно за $\varnothing^{(2)}$.

Да предположим, че $P(f)$ е изпълнено за произволна частична функция $f \in \mathcal{F}_2$. Ще покажем, че е вярно $P(\Gamma(f))$. Имаме

$$\begin{aligned}\Gamma(f)(x,y) \dot{-} 1 &\simeq [\texttt{if } x = 0 \texttt{ then } y \texttt{ else } f(x+1,y) \dot{-} 1] \dot{-} 1\\ &\simeq \texttt{if } x = 0 \texttt{ then } y \dot{-} 1 \texttt{ else } (f(x+1,y) \dot{-} 1) \dot{-} 1.\end{aligned}$$

Съгласно индукционното предположение

$$f(x+1,y) \dot{-} 1 \simeq f(x+1, y \dot{-} 1).$$

Следователно

$$\begin{aligned}\Gamma(f)(x,y) \dot{-} 1 &\simeq \texttt{if } x = 0 \texttt{ then } y \dot{-} 1 \texttt{ else } f(x+1, y \dot{-} 1) \dot{-} 1\\ &\simeq \Gamma(f)(x, y \dot{-} 1),\end{aligned}$$

откъдето $P(f_\Gamma)$ е в сила по индукционното правило на Скот.

Задача 10. Даден е операторът

$$\Gamma(f)(x,y) \simeq \texttt{if } x \equiv 1 \ (\mathrm{mod}\ 2) \texttt{ then } y+1 \texttt{ else } f(x/2,y)+1.$$

Да се докаже, че $\forall x \forall y(f_\Gamma(x,y) + 1 \simeq f_\Gamma(x, y+1))$.

Задача 11. Да разгледаме оператора

$$\Gamma(f)(x,y) \simeq \texttt{ if } p(x) \texttt{ then } y \texttt{ else } h(f(k(x),y)),$$

където p е тотален едноместен предикат, а h и k са тотални едноместни функции.

Да се докаже, че $\forall x \forall y(h(f_\Gamma(x,y)) \simeq f_\Gamma(x, h(y)))$.

Задача 12. Нека p е тотален едноместен предикат, h е навсякъде определена едноместна функция, а Γ е операторът, зададен с равенството

$$\Gamma(f)(x) \simeq \texttt{ if } p(x) \texttt{ then } x \texttt{ else } f(f(h(x))).$$

Да се докаже, че $\forall x(f_\Gamma(x) \simeq f_\Gamma(f_\Gamma(x)))$.

Упътване. Покажете, че свойството

$$P(f) \iff \forall x(!f(x) \Longrightarrow p(f(x)))$$

е непрекъснато и приложете правилото на Скот. Използвайте, че f_Γ е неподвижна точка на оператора Γ.

Задача 13. Нека Γ е оператор, определен с равенството

$$\Gamma(f)(x) \simeq \texttt{if } x \equiv 0 \ (\mathrm{mod}\ 2) \texttt{ then } x/2 \texttt{ else } f(f((3x+1)/2)).$$

Да се докаже, че $\forall x(!f_\Gamma(x) \Longrightarrow f_\Gamma(x) \leqq x/2)$.

Тотална ли е f_Γ?

Доказателство. Да разгледаме условието за частична коректност

$$P(f) \iff \forall x(!f(x) \Longrightarrow f(x) \leqq x/2).$$

То е в сила за $\varnothing^{(1)}$. Да предположим, че $P(f)$ е вярно за произволна $f \in \mathcal{F}_1$.

Нека $!\Gamma(f)(x)$. Ще покажем, че $\Gamma(f)(x) \leqq x/2$.

Ако x е четно, то $\Gamma(f)(x) = x/2$.

Ако x е нечетно, то $\Gamma(f)(x) = f(f((3x+1)/2))$. В такъв случай $!f((3x+1)/2)$ и $!f(f((3x+1)/2))$. Да положим $u = f((3x+1)/2)$. Тъй като $P(f)$ е в сила и $!f(u)$, то $f(u) \leqq u/2$. Прилагаме индукционното предположение още веднъж: от $P(f)$ и $!f((3x+1)/2)$ следва, че $f((3x+1)/2) \leqq (3x+1)/4$. Така

$$\Gamma(f)(x) = f(f((3x+1)/2)) \leqq f((3x+1)/2)/2 \leqq (3x+1)/8 \leqq x/2,$$

тъй като x е нечетно.

Функцията f_Γ не е тотална, например $\neg!f_\Gamma(1)$.

Задача 14. Даден е операторът

$$\Gamma(f)(x,y) \simeq \texttt{if } x = y \texttt{ then } 0 \texttt{ else } f(x, y+1) + 1.$$

Да се докаже, че

а) $\forall x \forall y(!f_\Gamma(x,y) \Longrightarrow f_\Gamma(x,y) \simeq x - y)$;

б) $\forall x \forall y(x < y \Longrightarrow \neg!f_\Gamma(x,y))$.

Упътване. б) Разгледайте условието P, дефинирано с

$$P(f) \iff \forall x \forall y(x < y \Longrightarrow \neg!f(x,y)).$$

Покажете, че P е непрекъснато, и приложете правилото на Скот.

Задача 15. Нека

$$\Gamma(f)(x) \simeq \texttt{if } x \text{ е точен квадрат } \texttt{then } \sqrt{x} \texttt{ else } f(f(x+1)).$$

Да се докаже, че $\forall x_{>1}(!f_\Gamma(x) \Longrightarrow f_\Gamma(x) < x)$.

Доказателство. Да разгледаме свойството

$$P(f) \iff \forall x(!f(x) \mathbin{\&} x \leqq 1 \Longrightarrow f(x) = x) \mathbin{\&}$$
$$\forall x(!f(x) \mathbin{\&} x > 1 \Longrightarrow f(x) < x).$$

Свойството P е конюнкция на две условия за частична коректност и по зад. 1 и зад. 4 а) следва, че е непрекъснато.

Ясно е, че $P(\varnothing^{(1)})$. Да предположим, че $P(f)$, за някоя $f \in \mathcal{F}_1$.

Нека $!\Gamma(f)(x)$. Ще покажем, че ако $x \leqq 1$, то $\Gamma(f)(x) = x$, и ако $x > 1$, то $\Gamma(f)(x) < x$.

1. Ако $x \leqq 1$, то x е точен квадрат и от дефиницията на Γ имаме $\Gamma(f)(x) = x$.

2. Нека $x > 1$. Тогава имаме два случая:

а) Ако x е точен квадрат, то $\Gamma(f)(x) = \sqrt{x} < x$, тъй като $x > 1$.

б) Ако x не е точен квадрат, то $\Gamma(f)(x) = f(f(x+1))$ и следователно $!f(x+1)$ и $!f(f(x+1))$. Имаме две възможности:

б1) $f(x+1) \leqq 1$. Сега от $P(f)$ и $!f(f(x+1))$ следва, че

$$f(f(x+1)) = f(x+1) \leqq 1.$$

Тогава $\Gamma(f)(x) \leqq 1 < x$, тъй като сме предположили, че $x > 1$.

б2) $f(x+1) > 1$. От $P(f)$ и $!f(f(x+1))$ следва, че

$$f(f(x+1)) < f(x+1).$$

От $P(f)$ и $!f(x+1)$ имаме $f(x+1) < x+1$, тъй като $x+1 > 1$. Следователно

$$\Gamma(f)(x) = f(f(x+1)) < f(x+1) < x+1.$$

Откъдето и в двата случая (а) и (б) получаваме $\Gamma(f)(x) < x$.

Задача 16. Да се докаже, че всеки от изброените оператори е рекурсивен и да се покаже, че посоченото свойство е в сила за най-малката му неподвижна точка f_Γ.

а) $\Gamma(f)(x) \simeq \texttt{if } x > 202 \texttt{ then } x - 3 \texttt{ else } f(f(x+4))$;

$$\forall x(!f_\Gamma(x) \Longrightarrow f_\Gamma(x) \geqq 200).$$

б) $\Gamma(f)(x,y) \simeq \texttt{if } x = 0 \texttt{ then } y \texttt{ else } f(x-1, y) + 1$;

$$\forall x \forall y(!f_\Gamma(x,y) \Longrightarrow f_\Gamma(x,y) \leqq x + y).$$

в) $\Gamma(f)(x) \simeq$

$$\texttt{if } x \leqq 1 \texttt{ then } 1 \texttt{ else if } x \equiv 0 \pmod 2 \texttt{ then } x/2 \texttt{ else } f(f((3x+1)/2));$$

$$\forall x_{>1}(!f_\Gamma(x) \Longrightarrow f_\Gamma(x) \leqq x/2).$$

г) $\Gamma(f)(x) \simeq \texttt{if } x \equiv 0 \pmod 3 \texttt{ then } x/3 \texttt{ else } f(f(x+1))$;

$$\forall x_{>0}(!f_\Gamma(x) \Longrightarrow f_\Gamma(x) < x).$$

Задача 17. а) Даден е операторът

$$\Gamma(f)(x) \simeq \texttt{if } x \equiv 0 \pmod 3 \texttt{ then } x/3 \texttt{ else } f(f(3x-1)).$$

Да се докаже, че $\forall x(!f_\Gamma(x) \Longrightarrow f_\Gamma(x) \leqq x/3)$.

б) Нека k е естествено число и $k > 1$. За оператора

$$\Gamma(f)(x) \simeq \texttt{if } x \equiv 0 \pmod k \texttt{ then } x/k \texttt{ else } f(f(kx-1)).$$

да се докаже, че $\forall x(!f_\Gamma(x) \Longrightarrow f_\Gamma(x) \leqq x/k)$.

Задача 18. За оператора

$$\Gamma(f)(x) \simeq \texttt{if } x \equiv 0 \pmod 3 \texttt{ then } x/3 \texttt{ else } f(f(2x+1))$$

да се докаже, че $\forall x(!f_\Gamma(x) \Longrightarrow f_\Gamma(x) \leqq x/3)$.

Тотална ли е функцията f_Γ?

Задача 19. Даден е операторът

а) $\Gamma(f)(x) \simeq$

$$\begin{aligned}&\texttt{if } x \equiv 0 \pmod 2 \texttt{ then } x/2\\ &\qquad \texttt{else if } x = 1 \texttt{ then } 1\\ &\qquad\qquad \texttt{else } f(f(x+1)) + f(f(x-1)).\end{aligned}$$

Да се докаже, че $\forall x(!f_\Gamma(x) \Longrightarrow 2f_\Gamma(x) \geqq x)$.

б) $\Gamma(f)(x) \simeq$

$$\begin{aligned}&\texttt{if } x \equiv 0 \pmod 3 \texttt{ then } x/3\\ &\qquad \texttt{else if } x \equiv 1 \pmod 3 \texttt{ then } f(3f(x-1)+1)\\ &\qquad\qquad \texttt{else } f(3f(x+1) \dot{-} 1).\end{aligned}$$

Да се докаже, че $\forall x(!f_\Gamma(x) \Longrightarrow f_\Gamma(x) \leqq x/3)$.

Задача 20. За дадените оператори покажете посочените свойства:

а) $\Gamma(f)(x,y) \simeq$

$$\begin{aligned}&\texttt{if } x = 0 \texttt{ then } y\\ &\qquad \texttt{else if } y = 0 \texttt{ then } f(x-1,1) + \mathrm{sg}(x-1)\\ &\qquad\qquad \texttt{else } f(x-1, f(x-1,y-1)) + 1.\end{aligned}$$

$$\forall x \forall y(!f_\Gamma(x,y) \Longrightarrow f_\Gamma(x,y) \geqq \max(x,y)).$$

б) $\Gamma(f)(x,y) \simeq$

$$\begin{aligned}&\texttt{if } y = 0 \texttt{ then } 0\\ &\qquad \texttt{else if } x = 0 \texttt{ then } f(1,y-1)\\ &\qquad\qquad \texttt{else } f(f(x-1,y-1),y-1) + 1.\end{aligned}$$

$$\forall x \forall y(!f_\Gamma(x,y) \Longrightarrow f_\Gamma(x,y) \geqq \min(x,y)).$$

Докажете, че и в двата случая функцията f_Γ е навсякъде определена.

Задача 21. Нека

$$\begin{aligned}\Gamma(f)(x) \simeq\ &\texttt{if } x \leqq 1 \texttt{ then } 1\\ &\qquad \texttt{else if } x \equiv 0 \pmod 2 \texttt{ then } x/2\\ &\qquad\qquad \texttt{else } f(f(3[x/2]+2)).\end{aligned}$$

Да се докаже, че $\forall x_{>1}(!f_\Gamma(x) \Longrightarrow f_\Gamma(x) \leqq x/2)$.

Задача 22. За дадените оператори покажете посочените свойства:

а) $\Gamma(f)(x) \simeq \texttt{if } \exists n(x = 2^n) \texttt{ then } \log_2(x) \texttt{ else } f(f(x+2))$.

$$\forall x_{>2}(!f_\Gamma(x) \Longrightarrow f_\Gamma(x) < x-1).$$

б) $\Gamma(f)(x) \simeq \texttt{if } x = 0 \texttt{ then } 0$

$$\begin{aligned}&\qquad \texttt{else if } \exists n(x = 2^n) \texttt{ then } \log_2(x)\\ &\qquad\qquad \texttt{else } f(f(x+1)).\end{aligned}$$

$$\forall x_{>1}(!f_\Gamma(x) \Longrightarrow f_\Gamma(x) < x).$$

Задача 23*. Нека означим с q примитивно рекурсивната функция $\lambda x.\ ((x+1) - 2^{[\log_2(x+1)]})$, където $[\log_2(x+1)]$ е цялата част на $\log_2(x+1)$. Нека φ е двуместна примитивно рекурсивна функция, дефинирана посредством разглеждане на случаи:

$$\varphi(x,y) = \begin{cases} x+1, & \text{ако } y = q(x),\\ 0 & \text{в противен случай.}\end{cases}$$

Да се докаже, че за най-малката неподвижна точка на оператора

$$\Gamma(f)(x,y) \simeq \texttt{if } \varphi(x,y) = 0 \texttt{ then } f(x, f(y,0)) \texttt{ else } \varphi(x,y).$$

е в сила свойството $\forall x(!f_\Gamma(x,0) \Longrightarrow f_\Gamma(x,0) \simeq x+1)$.

Да се докаже, че $\lambda x.f_\Gamma(x,0)$ е навсякъде определена.

Задача 24. Нека

а) $\Gamma(f)(x) \simeq \texttt{if } x \equiv 0 \pmod 2 \texttt{ then } x/2 \texttt{ else } f(f(3x+1))$.

Да се докаже, че f_Γ е дефинирана за всяко естествено число.

б)** *Открит проблем*

$$\begin{aligned}\Gamma(f)(x) \simeq\ &\texttt{if } x = 1 \texttt{ then } 0\\ &\qquad \texttt{else if } x \equiv 0 \pmod 2 \texttt{ then } f(x/2)\\ &\qquad\qquad \texttt{else } f(3x+1).\end{aligned}$$

Дефинирана ли е f_Γ за всяко положително естествено число?

Упътване. а) Използвайте представянето на естествените числа в двоична бройна система.

В следващите два примера ще илюстрираме метода на структурната индукция за доказване на свойства на най-малките неподвижни точки на компактни оператори. Тук индукцията е по структурата на данните. В § 1.1 вече разгледахме някои илюстрации на този метод.

Задача 25. Дадени са операторите

$$\Gamma_1(f)(x) \simeq \texttt{if } x=0 \texttt{ then } 1 \texttt{ else } x.f(x-1),$$

$$\Gamma_2(f)(x,y) \simeq \texttt{if } x=y \texttt{ then } 1 \texttt{ else } f(x,y+1).(y+1).$$

Функциите $\lambda x.f_{\Gamma_1}(x)$ и $\lambda x.f_{\Gamma_2}(x,0)$ пресмятат $x!$ по два различни начина. Да се докаже, че $\forall x(f_{\Gamma_2}(x,0) \simeq f_{\Gamma_1}(x))$.

Доказателство. Да разгледаме свойството p в естествените числа

$$p(x) \iff \forall y(f_{\Gamma_2}(x+y,y).f_{\Gamma_1}(y) \simeq f_{\Gamma_1}(x+y)).$$

С индукция по x ще покажем, че $p(x)$ е вярно за всяко x.

Ако $x=0$, то $p(0)$ е $\forall y(f_{\Gamma_2}(y,y).f_{\Gamma_1}(y) \simeq f_{\Gamma_1}(y))$. Ясно е, че $p(0)$ е вярно.

Нека $x>0$ и е вярно $p(x-1)$. Да разгледаме произволно $y \in \mathbb{N}$. Имаме

$f_{\Gamma_2}(x+y,y).f_{\Gamma_1}(y) \simeq$

$\simeq f_{\Gamma_2}(x+y,y+1).(y+1).f_{\Gamma_1}(y)$ (по дефиницията на f_{Γ_2} и $x+y>y$)

$\simeq f_{\Gamma_2}(x+y,y+1).f_{\Gamma_1}(y+1)$ (по дефиницията на f_{Γ_1} и $y+1>0$)

$\simeq f_{\Gamma_2}((x-1)+(y+1),y+1).f_{\Gamma_1}(y+1)$ ($x>0$)

$\simeq f_{\Gamma_1}((x-1)+(y+1))$ (по индукционното предположение за $x-1$)

$\simeq f_{\Gamma_1}(x+y)$.

Задача 26. Нека Γ_1 е определен като в зад. 25 и

$$\Gamma_2(f)(x,y) \simeq \texttt{if } x=0 \texttt{ then } y \texttt{ else } f(x-1,x.y).$$

Да се докаже, че $\forall x(f_{\Gamma_1}(x) \simeq f_{\Gamma_2}(x,1))$.

Упътване. С индукция по x покажете, че свойството p в естествените числа

$$p(x) \iff \forall y(f_{\Gamma_1}(x).y \simeq f_{\Gamma_2}(x,y))$$

е вярно за всяко x.

Трета глава

ОБЛАСТИ НА СКОТ

§ 3.1. Определение и примери за области на Скот

Нека A е произволно множество. Бинарната релация $\leqq$ в A наричаме (*частична*) *наредба* в A, ако тази релация е рефлексивна, антисиметрична и транзитивна.

Както е обичайно, с $\geqq$ ще отбелязваме обратната на релацията $\leqq$, т.е. ще считаме, че $a \geqq b$, ако $b \leqq a$. Ще пишем $a<b$, за да означим, че $a \leqq b$ и $a \neq b$.

Нека B е подмножество на A. Казваме, че $a \in A$ е *горна граница* (или *мажоранта*) на B, ако за всяко $b \in B$ е вярно, че $b \leqq a$. Казваме, че a е *точна горна граница* на множеството B, ако са изпълнени условията:

(1) a е горна граница на B;

(2) ако c е горна граница на B, то $a \leqq c$.

С други думи, точната горна граница на множеството B е най-малката сред горните му граници.

Нека b_1 и b_2 са точни горни граници на B. От условието (2) имаме $b_1 \leqq b_2$ и $b_2 \leqq b_1$, откъдето $b_1=b_2$. Следователно (ако въобще съществува) точната горна граница на B е единствена. Ще я означаваме с $\bigcup B$.

Нека $\{a_k\}$ е редица от елементи на A. Точната горна граница на редицата $\{a_k\}$ (или все едно, на множеството $\{a_k \mid k \in \mathbb{N}\}$) ще отбелязваме с $\bigcup\limits_k a_k$ (или само $\bigcup a_k$, ако няма опасност от някаква неяснота).

Казваме, че редицата $\{a_k\}$ е *монотонно растяща*, ако

$$a_k \leqq a_{k+1} \text{ за всяко } k=0,1,\ldots$$

Наредената тройка $(A,\leqq,\perp)$ наричаме *област на Скот*, ако са изпълнени следните условия:

(1) релацията $\leqq$ е частична наредба в множеството A;

(2) $\perp \in A$ е най-малкият елемент на A (т.е. $\perp \leqq a$ за всяко $a \in A$);

(3) всяка монотонно растяща редица $\{a_k\}$ от елементи на A притежава точна горна граница.

Забележка. Частична наредба, удовлетворяваща условието (3), се нарича още *пълна* наредба.

Нека D е произволно множество, а $\perp$ е обект, който не е от D. Да положим

$$D_\perp = D \cup \{\perp\}.$$

В множеството $D_\perp$ въвеждаме бинарната релация $\sqsubseteq$ по следния начин:

$$a \sqsubseteq b \iff a = \perp \vee a = b.$$

Задача 1. Да се докаже, че

а) релацията $\sqsubseteq$ е частична наредба в $D_\perp$;

б) наредената тройка $(D_\perp, \sqsubseteq, \perp)$ е област на Скот.

Забележка. Наредбата $\sqsubseteq$ се нарича *плоска наредба*, а структурата $(D_\perp, \sqsubseteq, \perp)$ — *плоска област на Скот*. Когато по-нататък говорим за частична наредба в множества от вида $D_\perp$, ще подразбираме плоската наредба в $D_\perp$.

Упътване. б) Нека $\{a_k\}$ е произволна монотонно растяща редица от елементи на $D_\perp$. За тази редица са възможни следните два случая:

I *случай.* $a_k = \perp$ за всяко $k = 0, 1, \ldots$ Тогава $\bigcup_k a_k = \perp$.

II *случай.* Съществува $k \in \mathbb{N}$, за което $a_k \neq \perp$. Нека k_0 е най-малкото число с това свойство. Тогава от $a_{k_0} \sqsubseteq a_{k_0+1} \sqsubseteq \ldots$ и определението на плоска наредба ще имаме $a_{k_0} = a_{k_0+1} = \ldots$ Следователно редицата $\{a_k\}$ има вида

$$\underbrace{\perp, \ldots, \perp}_{k_0-1 \text{ пъти}}, a_{k_0}, a_{k_0}, \ldots$$

В този случай $\bigcup_k a_k = a_{k_0}$.

Нека X и Y са произволни множества. Да означим с $\mathcal{F}$ съвкупността от всички частични функции от X към Y. Естествена частична наредба в класа $\mathcal{F}$ е релацията включване $(\subseteq)$, където по определение

$$f \subseteq g \iff \forall x_{x \in X} \forall y_{y \in Y} (f(x) \simeq y \Longrightarrow g(x) \simeq y).$$

В бъдеще, когато разглеждаме някаква съвкупност от частични функции, ще имаме предвид, че частичната наредба в тази съвкупност е релацията включване.

Задача 2. Нека $\mathcal{F} = \{f \mid f : X \dashrightarrow Y\}$, а $\varnothing \in \mathcal{F}$ е функцията, която за всяко $x \in X$ е неопределена. Да се докаже, че $(\mathcal{F}, \subseteq, \varnothing)$ е област на Скот.

Упътване. Нека $\{f_k\}$ е монотонно растяща редица в $\mathcal{F}$. Ще покажем, че функцията $g : X \dashrightarrow Y$, която се определя с еквивалентността

$$g(x) \simeq y \iff \exists k (f_k(x) \simeq y),$$

е точната горна граница на редицата $\{f_k\}$.

Преди всичко да съобразим, че g е коректно дефинирана. Наистина, ако допуснем, че съществуват x, $y_1 \neq y_2$ и $k_1 < k_2$, за които $f_{k_1}(x) = y_1$ и $f_{k_2}(x) = y_2$, то от условието $f_{k_1} \subseteq f_{k_2}$ ще имаме $f_{k_1}(x) = y_1 = f_{k_2}(x)$, т.е. $y_1 = y_2$ — противоречие с допуснатото.

От определението на g се вижда, че $g \supseteq f_k$ за всяко $k = 0, 1, \ldots$ и следователно g е горна граница на редицата $\{f_k\}$. Нека h е друга горна граница на тази редица. Трябва да покажем, че $g \subseteq h$, т.е. $g(x) \simeq y \Longrightarrow h(x) \simeq y$ при всеки избор на x, y. Да предположим, че $g(x) \simeq y$. Тогава съществува $k : f_k(x) \simeq y$. Но $f_k \subseteq h$ и следователно $h(x) \simeq y$.

Задача 3. Да се докаже, че наредената тройка $(\mathcal{C}_n, \subseteq, \varnothing^{(1)})$ не е област на Скот.

Доказателство. Ще покажем, че релацията $\subseteq$, ограничена до $\mathcal{C}_1$, не е пълна, т.е. съществува монотонно растяща редица от изчислими функции, чиято точна горна граница не е изчислима функция.

Нека f е произволна неизчислима функция, а $f_k = f|_{\{0,1,\ldots,k\}}$. Ясно е, че $\{f_k\}$ е монотонно растяща и $\bigcup f_k = f$. Освен това всяка функция f_k е крайна, и следователно изчислима, докато точната ѝ горна граница f не е изчислима.

Очевидно конструкция, подобна на горната, е приложима и за произволен клас $\mathcal{C}_n$, $n > 1$.

Задача 4. Нека X е произволно множество, а $\subseteq$ е теоретико-множествената релацията включване. Да се докаже, че наредената тройка $(2^X, \subseteq, \varnothing)$ е област на Скот.

Нека X е произволно множество, а $(A, \leqq, \perp)$ е област на Скот. Да означим с $\mathcal{G}$ съвкупността от всички тотални функции от X към A. Наредбата $\leqq$ в A определя следната бинарна релация в $\mathcal{G}$, която ще означаваме със същия символ $\leqq$:

$$g \leqq h \iff \forall x_{x \in X} (g(x) \leqq h(x)).$$

Да положим $\Omega(x) = \perp$ за всяко $x \in X$.

Задача 5. При горните предположения да се докаже, че наредената тройка $(\mathcal{G}, \leqq, \Omega)$ е област на Скот.

Упътване. Нека $\{g_k\}$ е монотонно растяща редица от елементи на $\mathcal{G}$. Тогава за всяко фиксирано $x \in X$ $\{g_k(x)\}$ е монотонно растяща редица в областта на Скот A и следователно притежава точна горна граница $\bigcup g_k(x)$.

Нека $h : X \to A$ е функцията, определена с равенството

$$h(x) = \bigcup_k g_k(x).$$

При фиксирано $k \in \mathbb{N}$ имаме $g_k(x) \leqq \bigcup g_k(x) = h(x)$ за всяко x от X, откъдето следва, че $g_k \leqq h$. Понеже последното включване е в сила за произволно k, то h е горна граница на редицата $\{g_k\}$.

Нека h' е друга горна граница на тази редица, т.е. $h' \geqq g_k$ за всяко $k \in \mathbb{N}$. Нека x е произволен елемент на X. Тогава $h'(x) \geqq g_k(x)$ за всяко k и следователно $h'(x)$ е горна граница за редицата $\{g_k(x)\}_k$. Но $h(x)$ е точната ѝ горна граница, откъдето получаваме $h(x) \leqq h'(x)$. Това включване е валидно за произволно x от X, което означава, че $h \leqq h'$.

Да отбележим, че когато говорим за наредба в дадена съвкупност от тотални функции $g : X \to A$, ще имаме предвид горната *поточкова* наредба, определена от частичната наредба $\leqq$ в A. Така например ако $\mathcal{G} = \{g \mid g : \mathbb{N}_\perp \to \mathbb{N}_\perp\}$, то подразбиращата се частична наредба в $\mathbb{N}_\perp$ е плоската наредба $\sqsubseteq$, и оттук частичната наредба в $\mathcal{G}$ е релацията $\sqsubseteq$, определена с условието:

$$g \sqsubseteq h \iff \forall x_{x \in \mathbb{N}_\perp}(g(x) \sqsubseteq h(x)).$$

Нека $(A_1, \leqq_1, \perp_1), \dots, (A_n, \leqq_n, \perp_n)$ са области на Скот. *Декартово произведение* на тези области наричаме наредената тройка $(A, \leqq, \perp)$, където $A = A_1 \times \dots \times A_n$, $\perp = (\perp_1, \dots, \perp_n)$, а бинарната релация $\leqq$ в $A_1 \times \dots \times A_n$ се определя с еквивалентността

$$(a_1, \dots, a_n) \leqq (b_1, \dots, b_n) \iff a_1 \leqq_1 b_1 \ \& \ \dots \ \& \ a_n \leqq_n b_n.$$

От определението на релацията $\leqq$ непосредствено следва, че тя е частична наредба. Ще я наричаме *покомпонентна наредба* в $A_1 \times \dots \times A_n$.

Задача 6. Нека $(A_1, \leqq_1, \perp_1), \dots, (A_n, \leqq_n, \perp_n)$ са области на Скот. Нека още за всяко $i = 1, \dots, n$ са дадени редици $\{a_k^i\}_k$ от елементи на A_i.

а) Да се докаже, че всяка от редиците $\{a_k^i\}_k$ е монотонно растяща в A_i, $1 \leqq i \leqq n$, точно тогава, когато редицата $\{(a_k^1, \dots, a_k^n)\}_k$ е монотонно растяща в $A_1 \times \dots \times A_n$ (относно покомпонентната наредба).

б) Нека всяка от редиците $\{a_k^i\}_k$ е монотонно растяща в A_i. Да се докаже, че е изпълнено условието

$$\Big(\bigcup_k a_k^1, \dots, \bigcup_k a_k^n\Big) = \bigcup_k (a_k^1, \dots, a_k^n).$$

Упътване. б) Да означим с b^i точната горна граница на $\{a_k^i\}_k$ в A_i. Имаме $a_k^i \leqq_i b^i$ за всяко $1 \leqq i \leqq n$, откъдето

$$(a_k^1, \dots, a_k^n) \leqq (b^1, \dots, b^n).$$

Следователно $(b^1, \dots, b^n)$ е горна граница на редицата $\{(a_k^1, \dots, a_k^n)\}_k$. Съобразете, че за всяка друга горна граница $(c^1, \dots, c^n)$ е вярно, че $(b^1, \dots, b^n) \leqq (c^1, \dots, c^n)$.

Задача 7. Да се докаже, че декартово произведение на области на Скот е област на Скот.

Упътване. Нека $(A, \leqq, \perp)$ е декартово произведение на областите на Скот $(A_i, \leqq_i, \perp_i)$, $1 \leqq i \leqq n$. Нека още $\{(a_k^1, \dots, a_k^n)\}_k$ е монотонно растяща редица в $A_1 \times \dots \times A_n$. По зад. 6 а) всяка от редиците $\{a_k^i\}_k$ е монотонно растяща в A_i, откъдето по зад. 6 б) точната горна граница на редицата $\{(a_k^1, \dots, a_k^n)\}_k$ съществува (и е равна на $(\bigcup_k a_k^1, \dots, \bigcup_k a_k^n)$).

Задача 8. Нека $\mathcal{F} = \{f \mid f : X \rightharpoonup Y\}$. Да се докаже, че едно множество $\mathcal{F}_0 \subseteq \mathcal{F}$ притежава точна горна граница тогава и само тогава, когато $\mathcal{F}_0$ има мажоранта.

Упътване. Нека за функцията $g : X \rightharpoonup Y$ е вярно, че $f \subseteq g$ за всяка $f \in \mathcal{F}_0$. Покажете, че следната дефиниция на функция от $\mathcal{F}$ е коректна:

$$h(x) \simeq y \iff \exists f_{\in \mathcal{F}_0}(f(x) \simeq y).$$

Докажете, че h е точната горна граница на $\mathcal{F}_0$.

Задача 9. Нека $\mathcal{G} = \{g \mid g : X \rightharpoonup \mathrm{D}_\perp\}$. Да се докаже, че множеството $\mathcal{G}_0 \subseteq \mathcal{G}$ притежава точна горна граница тогава и само тогава, когато $\mathcal{G}_0$ притежава поне една горна граница.

Задача 10. Да се докаже, че всяка от изброените редици е монотонно растяща в съответната област на Скот и да се намери точната ѝ горна граница.

а) $f_k(x) \simeq \begin{cases} x, & \text{ако } x \leqq k, \\ \neg!, & \text{в противен случай} \end{cases}$

в областта $(\mathcal{F}_1 = \{f \mid f : \mathbb{N} \rightharpoonup \mathbb{N}\}, \subseteq, \emptyset)$;

б) $g_k(x) = \begin{cases} x, & \text{ако } 0 \leqq x \leqq k, \\ \perp, & \text{ако } x = \perp \vee x > k, \end{cases}$

в областта $(\mathcal{G}_1 = \{g \mid g : \mathbb{N}_\perp \to \mathbb{N}_\perp\}, \sqsubseteq, \lambda x.\perp)$;

в) $h_k(x) = \begin{cases} tt, & \text{ако } 0 \leqq x \leqq k \;\&\; x \text{ е четно}, \\ ff, & \text{ако } x = \perp \vee (1 \leqq x \leqq k \text{ и } x \text{ е нечетно}), \\ \perp & \text{ако } x > k \end{cases}$

в областта $(\mathcal{G} = \{h \mid h : \mathbb{N}_\perp \to \mathrm{B}_\perp\}, \sqsubseteq, \lambda x.\perp)$, където B е множеството от булевите константи tt (истина) и ff (лъжа);

г) $\Gamma_k(f) = f|_{\{0,1,\dots,k\}}$

в областта $(\mathcal{H} = \{\Gamma \mid \Gamma : \mathcal{F}_1 \to \mathcal{F}_1\}, \subseteq, \lambda f.\emptyset^{(1)})$, където

$$\mathcal{F}_1 = \{f \mid f : \mathbb{N} \rightharpoonup \mathbb{N}\}.$$

Упътване. г) Тъй като $\mathcal{H}$ е съвкупност от тотални изображения, наредбата по подразбиране в $\mathcal{H}$ е поточковата наредба, определена от наредбата $\subseteq$ в множеството $\mathcal{F}_1 = \{f \mid f : \mathbb{N} \rightharpoonup \mathbb{N}\}$. С други думи,

$$\Gamma_1 \subseteq \Gamma_2 \iff \forall f_{f \in \mathcal{F}_1}(\Gamma_1(f) \subseteq \Gamma_2(f)).$$

Нека f е произволна функция от $\mathcal{F}_1$, а $k \in \mathbb{N}$. От определението за рестрикция на функция имаме $f|_{\{0,1,\dots,k\}} \subseteq f|_{\{0,1,\dots,k+1\}}$, т.е. $\Gamma_k(f) \subseteq \Gamma_{k+1}(f)$. От това неравенство (тъй като f е произволна) получаваме, че $\Gamma_k \subseteq \Gamma_{k+1}$. Следователно редицата $\{\Gamma_k\}_k$ е монотонно растяща. Съобразете, че операторът Γ_{id}, който се определя с равенството $\Gamma_{\mathrm{id}}(f) = f$ за всяка $f \in \mathcal{F}_1$, е точна горна граница на $\{\Gamma_k\}_k$.

Задача 11. Кои от изброените множества притежават точна горна граница (в съответните области на Скот)?

а) $\{\theta \mid \theta \subseteq f_0 \;\&\; \theta \text{ е крайна}\}$

в областта $(\mathcal{F} = \{f \mid f : X \rightharpoonup Y\}, \subseteq, \emptyset)$, където f_0 — фиксирана функция от $\mathcal{F}$;

б) $\{\theta \mid \theta \text{ е крайна}\}$

в областта $(\mathcal{F} = \{f \mid f : X \rightharpoonup Y\}, \subseteq, \emptyset)$;

в) $\{g \mid g \sqsubseteq g_0\}$

в областта $(\mathcal{G} = \{g \mid g : X \to \mathrm{D}_\perp\}, \sqsubseteq, \lambda x.\perp)$, където g_0 е фиксирана функция от $\mathcal{G}$;

г) $\{\lambda x.d \mid d \in \mathrm{D}_\perp\}$

в областта $(\mathcal{G} = \{g \mid g : X \to \mathrm{D}_\perp\}, \sqsubseteq, \lambda x.\perp)$;

д) $\{f_n \mid n \in \mathbb{N}\}$, където $f_n(x) \simeq y \iff x \leqq n \;\&\; y = x^2$,

в областта $(\mathcal{F}_1 = \{f \mid f : \mathbb{N} \rightharpoonup \mathbb{N}\}, \subseteq, \emptyset)$;

е) $\{f_n \mid n \in \mathbb{N}\}$, където $f_n(x) \simeq y \iff x \leqq n \;\&\; y = x^n$,

в областта $(\mathcal{F}_1 = \{f \mid f : \mathbb{N} \rightharpoonup \mathbb{N}\}, \subseteq, \emptyset)$.

Упътване. а) Използвайте зад. 8.

в) Използвайте зад. 9.

д) Съобразете, че $\{f_n\}$ е монотонно растяща редица.

Нека $(A, \leq)$ е частично наредено множество. Казваме, че множеството $C \subseteq A$ е *верига*, ако за всяко $a,\ b \in C$ е вярно, че $a \leq b$ или $b \leq a$. С други думи, C е верига, ако частичната наредба $\leq$, ограничена до C, е линейна наредба.

Казваме, че множеството $S \subseteq A$ е *насочено*, ако всеки два елемента на S имат горна граница в S, т.е. изпълнено е условието

$$a,\ b \in S \Longrightarrow \exists c_{c \in S}(a \leq c \;\&\; b \leq c).$$

Задача 12. Нека $(A, \leqq)$ е частично наредено множество. Да се докаже, че:

а) за всяка монотонно растяща редица $\{a_k\}$ в A множеството $\{a_k \mid k \in \mathbb{N}\}$ е верига;

б) всяка верига е насочено множество.

Да се даде пример за частично наредено множество, в което понятията монотонно растяща редица, верига и насочено множество не съвпадат.

Забележка. Поради а) монотонно растящите редици се наричат още ω-вериги.

Упътване. Покажете, че в частично нареденото множество

$$(\mathcal{F} = \{f \mid f : X \rightharpoonup Y\}, \subseteq)$$

множеството $\mathrm{Fin}(f) = \{\theta \mid \theta \text{ е крайна подфункция на } f\}$ е насочено, но не е верига.

Задача 13. Нека $(A, \leqq, \perp)$ е област на Скот. Да положим $C(A) = \{C \mid C \subseteq A \text{ е верига}\}$. Да се докаже, че наредената тройка $(C(A), \subseteq, \emptyset)$ също е област на Скот.

Задача 14*. Нека $(A, \leqq, \perp)$ е област на Скот и A е изброимо множество. Да се докаже, че всяко насочено множество в A има точна горна граница.

Задача 15. Нека S е насочено множество в частично нареденото множество $(\mathcal{F} = \{f \mid f : X \to\!\!\!\circ\; Y\}, \subseteq)$. Да се докаже, че точната горната граница $\bigcup S$ на S съществува и се определя с равенството

$$(\bigcup S)(x) \simeq y \iff \exists f_{\in S}(f(x) \simeq y).$$

Упътване. Разсъждавайте както при доказателството на зад. 2.

Нека $(\mathrm{D}_\perp, \sqsubseteq, \perp)$ е произволна плоска област на Скот. Съгласно зад. 7 декартовото произведение $\mathrm{D}_\perp^n = \mathrm{D}_\perp \times \ldots \times \mathrm{D}_\perp$, наредено с покомпонентната наредба, определена от $\sqsubseteq$, и най-малък елемент $(\underbrace{\perp, \ldots, \perp}_{n \text{ пъти}})$, също е област на Скот. За да не претоварваме записа, по-нататък ще използваме един и същ символ $\sqsubseteq$ за плоската наредба както в различни плоски области на Скот, така и в техни декартови произведения.

Задача 16. Да се докаже, че в областта на Скот

$$(\mathrm{D}_\perp^n, \sqsubseteq, (\perp, \ldots, \perp)):$$

а) всяка монотонно растяща редица $\{\overline{a}_k\}$ има най-много $n+1$ различни членове;

б) всяка верига е крайна и има най-много $n+1$ елемента;

в) всяко насочено множество е крайно и има най-много 2^n елемента.

Задача 17. Нека S е произволно насочено множество в съвкупността $\mathcal{G} = \{g \mid g : X \to \mathrm{D}_\perp\}$ с частична наредба, определена с условието $g \sqsubseteq h \iff \forall x_{x \in X}(f(x) \sqsubseteq g(x))$. Да се докаже, че:

а) за всяко $x \in X$ множеството $\{g(x) \mid g \in S\}$ е $\{\perp\}$ или $\{\perp, d\}$ за някое $d \in \mathrm{D}$ и следователно има точна горна граница в $\mathrm{D}_\perp$;

б) множеството S притежава точна горна граница в $\mathcal{G}$, която се дефинира с равенството

$$(\bigcup S)(x) = \bigcup \{g(x) \mid g \in S\}.$$

Упътване. Разсъждавайте както при доказателството на зад. 5.

Нека D и D' са произволни множества, а $\perp \notin \mathrm{D} \cup \mathrm{D}'$. Допълнителния елемент $\perp$ от множествата $\mathrm{D}_\perp = \mathrm{D} \cup \{\perp\}$ и $\mathrm{D}'_\perp = \mathrm{D}' \cup \{\perp\}$ ще тълкуваме най-общо като някаква неопределеност (поради което ще го отбелязваме с една и съща буква за различните области).

Ще казваме, че функцията $g : \mathrm{D}_\perp^n \to \mathrm{D}'_\perp$ е *точна*, ако $g(a_1, \ldots, a_n) = \perp$ всеки път, когато $\perp$ е сред компонентите на $(a_1, \ldots, a_n)$.

Задача 18. Нека $\mathcal{G}_s = \{g \mid g : \mathrm{D}_\perp^n \to \mathrm{D}'_\perp \;\&\; g \text{ е точна}\}$, $\Omega(\overline{a}) = \perp$ за всяко $\overline{a} \in \mathrm{D}_\perp^n$, а $\sqsubseteq$ е релация в $\mathcal{G}_s$, определена с условието

$$f \sqsubseteq g \iff \forall \overline{a}_{\overline{a} \in \mathrm{D}_\perp^n}(f(\overline{a}) \sqsubseteq g(\overline{a})).$$

Да се докаже, че наредената тройка $(\mathcal{G}_s, \sqsubseteq, \Omega)$ е област на Скот.

Доказателство. Според зад. 5 релацията $\sqsubseteq$ е частична наредба в множеството $\mathcal{G} = \{g \mid g : \mathrm{D}_\perp^n \to \mathrm{D}'_\perp\}$, а функцията Ω е най-малкият му елемент. Оттук следва, че $\sqsubseteq$ е частична наредба и в $\mathcal{G}_s$, а функцията Ω (която очевидно е точна) е най-малкият елемент на $\mathcal{G}_s$.

Нека $\{g_k\}$ е монотонно растяща редица от точни функции. Отново по зад. 5 функцията $h \in \mathcal{G}$, която се определя с равенството

$$h(\overline{a}) = \bigcup_k g_k(\overline{a}),$$

е точна горна граница на редицата $\{g_k\}_k$ (разглеждана като редица в $\mathcal{G}$). Ще покажем, че h е точна функция.

Наистина, нека $(a_1, \ldots, a_n) \in \mathrm{D}_\perp^n$ и $a_i = \perp$ за някое $i \in \{1, \ldots, n\}$. Тъй като за всяко $k \in \mathbb{N}$ функцията g_k е точна, то $g_k(a_1, \ldots, a_n) = \perp$. Следователно

$$h(a_1, \ldots, a_n) = \bigcup_k g_k(a_1, \ldots, a_n) = \perp.$$

На всяка частична функция $f : \mathrm{D}^n \to\!\!\!\circ\; \mathrm{D}'$ съпоставяме тотална функция $f^* : \mathrm{D}_\perp^n \to \mathrm{D}'_\perp$ със следното определение:

$$f^*(\overline{a}) = \begin{cases} f(\overline{a}), & \text{ако } \overline{a} \in \mathrm{D}^n \,\&\, !f(\overline{a}), \\ \perp, & \text{в противен случай.} \end{cases}$$

Функцията f^* ще наричаме *точно* (или *естествено*) *продължение* на f.

Задача 19. Да се докаже, че функцията $g : \mathrm{D}_\perp^n \to \mathrm{D}'_\perp$ е точна тогава и само тогава, когато g е точно продължение на някоя частична функция $f : \mathrm{D}^n \to\!\!\!\circ\; \mathrm{D}'$.

Задача 20. Нека $f\colon \mathrm{D}^n \twoheadrightarrow \mathrm{D}'$ и $g\colon \mathrm{D}^n \twoheadrightarrow \mathrm{D}'$. Да се докаже, че

$$f \subseteq g \iff f^* \sqsubseteq g^*.$$

Функцията $g : \mathrm{D}^n_\perp \to \mathrm{D}'_\perp$ се нарича *монотонна*, ако за всяко $\overline{a},\ \overline{b} \in \mathrm{D}^n_\perp$ е изпълнено

$$\overline{a} \sqsubseteq \overline{b} \Longrightarrow g(\overline{a}) \sqsubseteq g(\overline{b}).$$

Задача 21. Да се докаже, че

а) всяка константна функция $g : \mathrm{D}^n_\perp \to \mathrm{D}'_\perp$ е монотонна;

б) всяка точна функция $g : \mathrm{D}^n_\perp \to \mathrm{D}'_\perp$ е монотонна.

Упътване. б) Покажете, че ако $\overline{a} \sqsubseteq \overline{b}$ и $\overline{a} \neq \overline{b}$, то $\perp$ е компонента на $\overline{a}$. Оттук ще следва, че $g(\overline{a}) = \perp \sqsubseteq g(\overline{b})$, каквато и да е стойността $g(\overline{b})$.

В следващата задача $\mathbb{Z}$ и $\mathbb{R}$ са означения съответно за множеството на целите и на реалните числа, $\mathrm{B} = \{tt, ff\}$, а D е произволно множество.

Задача 22. Проверете кои от изброените функции са монотонни и/или точни:

а) $g_1 : \mathbb{N}_\perp \to \mathbb{N}_\perp$, където $g_1(a) = \begin{cases} 0, & \text{ако } a = \perp, \\ 1, & \text{ако } a \neq \perp; \end{cases}$

б) $g_2 : \mathbb{Z}^2_\perp \to \mathbb{Z}_\perp$, където функцията g_2 удовлетворява равенствата:

$$g_2(a,b) = a.b, \text{ ако } a,\ b \in \mathbb{Z}$$
$$g_2(a,\perp) = g_2(\perp,a) = \perp, \text{ ако } a \in \mathbb{Z} \text{ и } a \neq 0,$$
$$g_2(0,\perp) = g_2(\perp,0) = 0,$$
$$g_2(\perp,\perp) = \perp;$$

в) $g_3 : \mathbb{R}^2_\perp \to \mathbb{R}_\perp$,

където $g_3(a,b) = \begin{cases} a \setminus b, & \text{ако } a \in \mathbb{R}\ \&\ b \in \mathbb{R} \setminus \{0\}, \\ \perp, & \text{в останалите случаи;} \end{cases}$

г) $g_4 : \mathbb{R}^2_\perp \to \mathbb{R}_\perp$,

където $g_4(a,b) = \begin{cases} a \setminus b, & \text{ако } a \in \mathbb{R}\ \&\ b \in \mathbb{R} \setminus \{0\}, \\ 0, & \text{ако } a \in \mathbb{R}\ \&\ b = \perp, \\ \perp, & \text{в останалите случаи;} \end{cases}$

д) $g_5 : \mathrm{D}_\perp \to \mathrm{B}_\perp$, където $g_5(a) = \begin{cases} tt, & \text{ако } a \in \mathrm{D}, \\ ff, & \text{ако } a = \perp; \end{cases}$

е) $g_6 : \mathrm{D}^2_\perp \to \mathrm{B}_\perp$, където $g_6(a,b) = \begin{cases} tt, & \text{ако } a = b, \\ ff, & \text{ако } a \neq b; \end{cases}$

ж) $g_7 : \mathrm{D}^2_\perp \to \mathrm{B}_\perp$, където $g_7(a,b) = \begin{cases} tt, & \text{ако } a = b, \\ ff, & \text{ако } a,\ b \in \mathrm{D}\ \&\ a \neq b, \\ \perp, & \text{в останалите случаи;} \end{cases}$

з) $g_8 : \mathrm{D}^2_\perp \to \mathrm{B}_\perp$, където $g_8(a,b) = \begin{cases} tt, & \text{ако } a,\ b \in \mathrm{D}\ \&\ a = b, \\ ff, & \text{ако } a,\ b \in \mathrm{D}\ \&\ a \neq b, \\ \perp, & \text{ако } a = \perp \vee b = \perp; \end{cases}$

и) $g_9 : \mathrm{B}_\perp \times \mathrm{D}^2_\perp \to \mathrm{D}_\perp$, където $g_9(a,b,c) = \begin{cases} b, & \text{ако } a = tt, \\ c, & \text{ако } a = ff, \\ \perp, & \text{ако } a = \perp. \end{cases}$

Упътване. б) Функцията g_2 не е точна (защото $g_2(0,\perp) = 0 \neq \perp$), но е монотонна.

в) Функцията g_3 е точна, и съгласно зад. 21 б) — монотонна.

д) Функцията g_5 не е монотонна, защото $g_5(\perp) = ff \not\sqsubseteq tt = g_5(a)$ при $a \in \mathrm{D}$.

е) Функцията g_6 (на която можем да гледаме като на условно равенство в D, ако елемента $\perp$ тълкуваме като неопределеност) не е монотонна, а оттук не е и точна.

з) Функцията g_8 е точно продължение на предиката „равенство“ в D, и следователно е точна и монотонна.

Задача 23. Нека $g : \mathrm{D}^n_\perp \to \mathrm{D}'_\perp$ е монотонна. Да се докаже, че g е константна функция или е вярно, че $g(\perp,\dots,\perp) = \perp$.

Задача 24. а) Да се докаже, че едноместната функция $g : \mathrm{D}_\perp \to \mathrm{D}'_\perp$ е монотонна точно тогава, когато е константна или е точна.

б) Да се даде пример за двуместна функция $g : \mathrm{D}^2_\perp \to \mathrm{D}'_\perp$, която е монотонна, но не е нито точна, нито константна.

Доказателство. а) Нека $g : \mathrm{D}_\perp \to \mathrm{D}'_\perp$ е монотонна. Ако $g(\perp) = \perp$, то g е точна. В противен случай $g(\perp) = d \in \mathrm{D}'$ и от монотонността на g ще следва, че за всяко $a \in \mathrm{D}$ е изпълнено $g(a) = d$. Следователно g е константна.

Обратната посока на а) следва от зад. 21.

б) Нека c е фиксиран елемент от D. Да положим

$$g(a,b) = \begin{cases} \perp, & \text{ако } a = \perp, \\ c, & \text{ако } a \neq \perp. \end{cases}$$

Очевидно функцията g не е константна. Тя не е и точна, защото $g(a,\perp) = c \neq \perp$ при $a \neq \perp$.

От друга страна, g е монотонна. Наистина, да вземем произволни $(a,b) \sqsubseteq (a',b')$ от $D^2_\perp$. Ако $a=\perp$, то $g(a,b) = \perp \sqsubseteq g(a',b')$. Ако $a \neq \perp$, от $(a,b) \sqsubseteq (a',b')$ ще имаме $a = a'$ и следователно $a' \neq \perp$. Оттук $g(a,b) = c = g(a',b')$.

Казваме, че функцията $g : D^n_\perp \to D'_\perp$ е *точна по i-тия си аргумент*, ако $g(a_1,\dots,a_n) = \perp$ всеки път, когато $a_i = \perp$.

Функцията $g : D^n_\perp \to D'_\perp$ е *монотонна по i-тия си аргумент*, ако при всеки избор на $a_1,\dots,a_i,\dots,a_n$, a'_i от $D_\perp$ е изпълнено условието

$$a_i \sqsubseteq a'_i \Longrightarrow g(a_1,\dots,a_i,\dots,a_n) \sqsubseteq g(a_1,\dots,a'_i,\dots,a_n).$$

Задача 25. Да се докаже, че $g : D^n_\perp \to D'_\perp$ е точна тогава и само тогава, когато g е точна по всеки от аргументите си.

Задача 26. Да се докаже, че функцията $g : D^n_\perp \to D'_\perp$ е монотонна тогава и само тогава, когато g е монотонна по всеки от аргументите си.

Задача 27. Нека $g : D^n_\perp \to D'_\perp$ и $1 \leqq i \leqq n$. Да се докаже, че ако функцията g е точна по a_i или g не зависи от a_i (т.е. a_i е фиктивен аргумент за g), то g е монотонна по a_i.

Упътване. Разсъждавайте както при доказателството на зад. 21.

Задача 28. За кои от изброените в зад. 22 функции, които не са монотонни или не са точни, може да се твърди, че са монотонни или точни по някой свой аргумент?

Упътване. Функцията g_4 не е монотонна (защото например $g_4(10,\perp) = 0 \not\sqsubseteq g_4(10,10) = 1$), но е монотонна по първия си аргумент.

Функцията g_9 е продължение на функцията $f : B \times D^2 \to D$, където

$$f(a,b,c) = \mathbf{if}\ a = tt\ \mathbf{then}\ b\ \mathbf{else}\ c.$$

Функцията g е точна по първия си аргумент ($g_9(\perp,b,c) = \perp$), и следователно е монотонна по него. Покажете, че g_9 не е точна, но е монотонна по втория и третия си аргумент.

Задача 29. Да се докаже, че съществува функция g, която е монотонна по някой свой аргумент a_i, но не е точна по a_i и този аргумент не е фиктивен за g.

Упътване. Да отбележим, че според тази задача твърдението, обратно на това от зад. 27 не е вярно. Ясно е от зад. 25, че функцията

g трябва да има поне два аргумента. Разгледайте например функцията $g : \mathbb{N}^2_\perp \to \mathbb{N}_\perp$, определена по следния начин:

$$g(a,b) = \begin{cases} a, & \text{ако } b = 0,\\ b, & \text{ако } b \neq 0 \text{ (т.е. } b = \perp \text{ или } b > 0). \end{cases}$$

Проверете, че g е монотонна по a, но не е точна по a и (очевидно) зависи от a.

Задача 30. Да се докаже, че съществува функция $g : D^n_\perp \to D_\perp$, която е монотонна по всеки от аргументите си, и едновременно с това съществува аргумент, който не е фиктивен за g и по който g не е точна.

Нека $(A, \leqq)$ е частично наредено множество и $B \subseteq A$. Ще казваме, че $a \in A$ е *точна долна граница* на B ($a = \bigcap B$), ако са изпълнени условията:

(1) $a \leqq b$ за всяко $b \in B$ (a е долна граница на B);

(2) ако c е долна граница за B, то $c \leqq a$ (a е най-голямата сред долните граници на B).

Задача 31. Да се докаже, че всяко непразно подмножество на $\mathcal{F} = \{f \mid f : X \mathrel{\circ\!\!\!\longrightarrow} Y\}$ притежава точна долна граница.

Упътване. Нека $\varnothing \neq \mathcal{F}_0 \subseteq \mathcal{F}$. Проверете, че функцията g, която се дефинира с условието

$$g(x) \simeq y \iff \forall f_{f \in \mathcal{F}_0} (f(x) \simeq y),$$

е точна долна граница на $\mathcal{F}_0$.

Задача 32. Да се докаже, че всяко непразно подмножество на $\mathcal{G} = \{g \mid g : X \to D_\perp\}$ притежава точна долна граница.

Упътване. Най-напред съобразете, че всяко непразно подмножество $B \subseteq D_\perp$ има точна долна граница $\bigcap B$.

Нека сега $\varnothing \neq \mathcal{G}_0 \subseteq \mathcal{G}$. Тогава за всяко $x \in X$ точната долна граница $\bigcap \{g(x) \mid g \in \mathcal{G}_0\}$ е определена. Да положим

$$h(x) = \bigcap \{g(x) \mid g \in \mathcal{G}_0\}.$$

Проверете, че $h = \bigcap \mathcal{G}_0$.

Задача 33. Нека $\mathcal{F} = \{f \mid f : X \mathrel{\circ\!\!\!\longrightarrow} Y\}$, а релацията $\subseteq'$ е обратната на релацията $\subseteq$. Да се докаже, че $\subseteq'$ е пълна наредба в $\mathcal{F}$, но по отношение на тази релация множеството $\mathcal{F}$ няма най-малък елемент.

Упътване. Използвайте зад. 31.

Задача 34. Нека $\mathcal{G} = \{g \mid g : X \to D_\perp\}$, а релацията $\sqsubseteq'$ е обратната на поточковата наредба $\sqsubseteq$ в $\mathcal{G}$, т.е.

$$g \sqsubseteq' h \iff \forall x_{x \in D_\perp}(h(x) \sqsubseteq g(x)).$$

Да се докаже, че тази релация е пълна наредба в $\mathcal{G}$, но по отношение на $\sqsubseteq'$ множеството $\mathcal{G}$ няма най-малък елемент.

Упътване. Използвайте зад. 32.

Задача 35. Нека X е произволно множество, а релацията $\subseteq'$ е обратната на релацията $\subseteq$. Да се докаже, че $(2^X, \subseteq', X)$ е област на Скот.

§ 3.2. Непрекъснати изображения в области на Скот

Нека $(A_1, \leqq_1, \perp_1)$ и $(A_2, \leqq_2, \perp_2)$ са области на Скот. Казваме, че изображението $f : A_1 \to A_2$ е *монотонно*, ако при всеки избор на a и b от A_1 е изпълнено условието

$$a \leqq_1 b \Longrightarrow f(a) \leqq_2 f(b).$$

Изображението $f : A_1 \to A_2$ наричаме *изброимо непрекъснато* (или само *непрекъснато*), ако за всяка монотонно растяща редица $\{a_k\}$ от елементи на A_1 е в сила равенството

$$f(\bigcup_k a_k) = \bigcup_k f(a_k).$$

Забележка. Това равенство се разбира в следния смисъл: точната горна граница $\bigcup_k f(a_k)$ на редицата $\{f(a_k)\}$ съществува и е равна на $f(\bigcup_k a_k)$.

В следващите няколко задачи ще предполагаме, че $(A_1, \leqq_1, \perp_1)$ и $(A_2, \leqq_2, \perp_2)$ са произволни области на Скот.

Задача 1. Да се докаже, че всяко непрекъснато изображение $f : A_1 \to A_2$ е монотонно.

Доказателство. Нека $a,\ b \in A_1$ и $a \leqq_1 b$. Да образуваме редицата $a, b, b, \ldots$ Тя е монотонно растяща и нейната точна горна

граница е b. От условието за непрекъснатост на f, приложено към тази редица, ще имаме

$$f(b) = \bigcup\{f(a), f(b), f(b), \ldots\},$$

откъдето получаваме, че $f(a) \leqq_2 f(b)$.

Задача 2. а) Нека $f : A_1 \to A_2$ е монотонно изображение, а $\{a_k\}$ е монотонно растяща редица от елементи на A_1. Да се докаже, че редицата $\{f(a_k)\}$ е монотонно растяща и за точната ѝ горна граница $\bigcup_k f(a_k)$ е изпълнено неравенството

$$\bigcup_k f(a_k) \leqq_2 f(\bigcup_k a_k)$$

б) Да се даде пример за монотонно изображение f и за монотонно растяща редица $\{a_k\}$, за които $\bigcup_k f(a_k) <_2 f(\bigcup_k a_k)$ (откъдето ще следва, че не всяко монотонно изображение е непрекъснато).

Доказателство. а) За произволно k имаме по условие $a_k \leqq_1 a_{k+1}$ и от монотонността на f получаваме $f(a_k) \leqq_2 f(a_{k+1})$. Следователно $\{f(a_k)\}$ е монотонно растяща редица в областта на Скот $(A_2, \leqq_2, \perp_2)$, откъдето по определение следва, че тази редица притежава точна горна граница.

Тъй като $a_k \leqq \bigcup_k a_k$, от монотонността на f ще имаме

$$f(a_k) \leqq_2 f(\bigcup_k a_k).$$

Понеже k е произволно, от това неравенство следва, че $f(\bigcup_k a_k)$ е горна граница на редицата $\{f(a_k)\}$. Следователно тя мажорира точната горна граница на тази редица, с други думи,

$$\bigcup_k f(a_k) \leqq_2 f(\bigcup_k a_k).$$

б) Нека $A_1 = A_2 = \{h \mid h : \mathbb{N} \rightharpoonup \mathbb{N}\}$, като множеството A_1 е наредено с обичайната наредба $\subseteq$. Да разгледаме изображението $\Gamma : A_1 \to A_1$ със следното определение:

$$\Gamma(h) = \begin{cases} O, & \text{ако } h \text{ е тотална,} \\ \varnothing, & \text{в противен случай.} \end{cases}$$

Монотонността на Γ се проверява непосредствено. Нека h е произволна тотална функция в $\mathbb{N}$, а h_k е рестрикцията на h върху крайното множество $\{0,\ldots,k\}$. Ясно е, че $\bigcup_k h_k = h$. Тъй като всяка от функциите h_k е крайна, то $\Gamma(h_k) = \varnothing$ и следователно

$$\bigcup_k \Gamma(h_k) = \varnothing \subset \Gamma(\bigcup_k h_k) = \Gamma(h) = O.$$

Нека $\mathcal{G}$ е някаква съвкупност от тотални функции от A_1 към A_2. Да напомним, че наредбата по подразбиране $\leqq$ в $\mathcal{G}$ е поточковата наредба, определена от частичната наредба $\leqq_2$ в A_2. Последното означава, че за всяка функция $f, g \in \mathcal{G}$ е изпълнено

$$f \leqq g \iff \forall a_{a \in A_1}(f(a) \leqq_2 g(a)).$$

Задача 3. Нека

$$\mathcal{G}_{\text{mon}} = \{g \,|\, g : A_1 \to A_2 \;\&\; g \text{ е монотонна функция}\},$$

а $\Omega = \lambda x.\ \perp_2$. Да се докаже, че наредената тройка $(\mathcal{G}_{\text{mon}}, \leqq, \Omega)$ е област на Скот.

Доказателство. Съгласно зад. 5, § 3.1, релацията $\leqq$ удовлетворява аксиомите за частична наредба. Освен това монотонната функция $\Omega = \lambda a.\ \perp_2$ е най малкият елемент на $\mathcal{G} = \{g \,|\, g : A_1 \to A_2\}$, а оттук и на $\mathcal{G}_{\text{mon}}$.

Нека $\{g_k\}$ е монотонно растяща редица от монотонни функции. Отново според зад. 5, § 3.1, тази редица (разглеждана като редица в $\mathcal{G}$) притежава точна горна граница $h \in \mathcal{G}$, която се определя с равенството $h(a) = \bigcup g_k(a)$.

Ще покажем, че функцията h също е монотонна. Наистина, нека a, $b \in A$ и $a \leqq_1 b$. Тъй като всяка от функциите g_k е монотонна, то $g_k(a) \leqq_2 g_k(b)$. Но $g_k(b) \leqq_2 h(b)$, съгласно определението на h. Следователно $g_k(a) \leqq_2 h(b)$ за всяко $k \in \mathbb{N}$, т.е. $h(b)$ е горна граница за редицата $\{g_k(a)\}_k$. Тогава $h(b)$ мажорира точната ѝ горна граница $\bigcup g_k(a)$, която по определение е равна на $h(a)$. Получихме, че $a \leqq_1 b \implies h(a) \leqq_2 h(b)$, което означава, че h е монотонна функция.

Задача 4. Нека

$$\mathcal{G}_c = \{g \,|\, g : A_1 \to A_2 \,\&\, g \text{ е непрекъсната функция}\}.$$

Да се докаже, че наредената тройка $(\mathcal{G}_c, \leqq, \Omega)$ е област на Скот.

Упътване. Нека $\{g_k\}$ е монотонно растяща редица в $\mathcal{G}_c$. Тъй като всяка от функциите g_k е непрекъсната, тя е и монотонна и по предишната задача точната ѝ горна граница $h = \bigcup g_k$ съществува и е монотонна функция. Ще покажем, че h е и непрекъсната функция.

Да изберем $\{a_n\}_n$ — произволна монотонно растяща редица от елементи на A_1 и нека $a = \bigcup_n a_n$. Тъй като h е монотонна, от зад. 2 а) ще имаме $\bigcup_n h(a_n) \leqq_2 h(a)$, т.е. $h(a)$ е горна граница на редицата $\{h(a_n)\}_n$.

Нека b е друга горна граница на тази редица. Ще покажем, че $h(a) \leqq_2 b$. Да фиксираме произволно $k \in \mathbb{N}$. Тъй като h мажорира g_k, то

$$g_k(a_n) \leqq_2 h(a_n) \leqq_2 b$$

при всеки избор на n от $\mathbb{N}$. Тогава

$$\bigcup_n g_k(a_n) \leqq_2 b.$$

Но функцията g_k е непрекъсната и следователно

$$\bigcup_n g_k(a_n) = g_k(\bigcup_n a_n) = g_k(a).$$

Получихме, че за всяко $k \in \mathbb{N}$ е в сила неравенството $g_k(a) \leqq_2 b$, откъдето $\bigcup_k g_k(a) \leqq_2 b$, т.е. $h(a) \leqq_2 b$.

Задача 5. Нека $(A_1, \leqq_1, \perp_1)$ е област на Скот, за която е известно, че всяка монотонно растяща редица от елементи на A_1 има само краен брой различни членове. Да се докаже, че ако изображението $f : A_1 \to A_2$ е монотонно, то f е непрекъснато.

Упътване. Нека $\{a_k\}$ е монотонно растяща редица в A_1. По условие съществува число k_0, такова че $a_k = a_{k_0}$ за всяко $k \geqq k_0$. Тогава $\bigcup a_k = a_{k_0}$. Съобразете, че $f(a_{k_0})$ е точната горна граница на редицата $\{f(a_k)\}$.

Задача 6. Нека D и D′ са произволни множества, а f е монотонно изображение от $\mathrm{D}_\perp^n$ в $\mathrm{D}'_\perp$. Да се докаже, че f е непрекъснато изображение.

Доказателство. Твърдението следва непосредствено от предишната задача, като се вземе предвид фактът, че всяка монотонно растяща редица в $\mathrm{D}_\perp^n$ има краен брой различни членове (зад. 16, § 3.1).

Задача 7. Нека $(A_1, \leqq_1, \perp_1)$ и $(A_2, \leqq_2, \perp_2)$ са области на Скот, а $f : A_1 \to A_2$ е биективно изображение, удовлетворяващо условието

$$\forall a_{a \in A_1} \forall b_{b \in A_1} (a \leqq_1 b \iff f(a) \leqq_2 f(b)).$$

Да се докаже, че изображението f е непрекъснато и $f(\perp_1) = \perp_2$.

В следващите няколко задачи ще предполагаме, че $(A, \leqq, \perp)$, $(A_1, \leqq_1, \perp_1), \ldots, (A_n, \leqq_n, \perp_n)$, са произволни области на Скот, а наредбата в декартовото произведение $A_1 \times \cdots \times A_n$ е покомпонентната наредба, определена от частичните наредби $\leqq_i$, $1 \leqq i \leqq n$.

Задача 8. Нека $f_1 : A \to A_1, \ldots, f_n : A \to A_n$ са непрекъснати изображения в съответните области. Да положим

$$f(a) = (f_1(a), \ldots, f_n(a)) \text{ за всяко } a \in A.$$

Да се докаже, че изображението $f : A \to A_1 \times \cdots \times A_n$ също е непрекъснато.

Доказателство. Нека $\{a_k\}$ е монотонно растяща редица в A. От непрекъснатостта на f_i, $1 \leqq i \leqq n$, имаме

$$f(\bigcup_k a_k) = (f_1(\bigcup_k a_k), \ldots, f_n(\bigcup_k a_k)) = (\bigcup_k f_1(a_k), \ldots, \bigcup_k f_n(a_k)).$$

Тъй като всяка от редиците $\{f_i(a_k)\}_k$, $1 \leqq i \leqq n$, е монотонно растяща, от зад. 6 б), § 3.1, получаваме

$$(\bigcup_k f_1(a_k), \ldots, \bigcup_k f_n(a_k)) = \bigcup_k (f_1(a_k), \ldots, f_n(a_k)),$$

откъдето окончателно

$$f(\bigcup_k a_k) = \bigcup_k (f_1(a_k), \ldots, f_n(a_k)) = \bigcup_k f(a_k).$$

Задача 9. Да се докаже, че изображението $f : A_1 \times \cdots \times A_n \to A$ е непрекъснато тогава и само тогава, когато за всяка монотонно растяща редица $\{a_k^i\}_k$ в A_i, $1 \leqq i \leqq n$, е изпълнено условието

$$f(\bigcup_k a_k^1, \ldots, \bigcup_k a_k^n) = \bigcup_k f(a_k^1, \ldots, a_k^n).$$

Упътване. Използвайте зад. 6, § 3.1.

Казваме, че изображението $f : A_1 \times \cdots \times A_n \to A$ е *непрекъснато по i-тия си аргумент*, ако за всяка монотонно растяща редица $\{a_k^i\}_k$ от елементи на A_i и за всяко $a^j \in A_j$, $1 \leqq j \leqq n \,\&\, j \neq i$, е в сила равенството

$$f(\bigcup_k (a^1, \ldots, a_k^i, \ldots, a^n)) = f(a^1, \ldots, \bigcup_k a_k^i, \ldots, a^n).$$

Задача 10. Нека $(A, \leqq, \perp)$, $(A_1, \leqq_1, \perp_1), \ldots, (A_n, \leqq_n, \perp_n)$ са области на Скот. Да се докаже, че изображението

$$f : A_1 \times \cdots \times A_n \to A$$

е непрекъснато точно тогава, когато е непрекъснато по всеки от аргументите си.

Задача 11. Нека $(A, \leqq, \perp)$, $(A_i, \leqq_i, \perp_i)$, $0 \leqq i \leqq n$, са области на Скот и изображенията

$$f_1 : A_0 \to A_1, \ldots, \; f_n : A_0 \to A_n, \; g : A_1 \times \cdots \times A_n \to A$$

са непрекъснати в съответните области. Да се докаже, че тяхната суперпозиция $h = g(f_1, \ldots, f_n) : A_0 \to A$ също е непрекъснато изображение.

Доказателство. Нека $\{a_k\}$ е монотонно растяща редица в A_0. Тогава от непрекъснатостта на $f_1, \ldots, f_n$ ще имаме

$$h(\bigcup a_k) = g(f_1(\bigcup a_k), \ldots, f_n(\bigcup a_k)) = g(\bigcup f_1(a_k), \ldots, \bigcup f_n(a_k)).$$

Тъй като всяка от редиците $\{f_i(a_k)\}$ е монотонно растяща, от зад. 9 получаваме

$$g(\bigcup f_1(a_k), \ldots, \bigcup f_n(a_k)) = \bigcup g(f_1(a_k), \ldots, f_n(a_k)),$$

откъдето окончателно

$$h(\bigcup a_k) = \bigcup g(f_1(a_k), \ldots, f_n(a_k)) = \bigcup h(a_k).$$

Оттук нататък ще разглеждаме само изображения, които преработват (частични) функции. Спазвайки терминологията от предишните глави, тези изображения ще наричаме оператори.

Нека D е произволно множество. В следващите задачи с $\mathcal{F}_n$ ще отбелязваме съвкупността от всички частични n-местни функции в D, (частично) наредена с релацията $\subseteq$.

Ако Γ е изображение от $\mathcal{F}_{n_1} \times \cdots \times \mathcal{F}_{n_r}$ в $\mathcal{F}_n$, ще казваме че Γ е *оператор от тип* $(n_1, \ldots, n_r \to n)$.

Задача 12. Нека Γ е оператор от тип $(n_1, \ldots, n_r \to n)$. Да се докаже, че ако операторът Γ е:

а) константен, т.е. $\Gamma(f_1, \ldots, f_r) = g_0$, където $g_0 \in \mathcal{F}_n$ е фиксирана;

б) проектиращ, т.е. $\Gamma(f_1, \ldots, f_r) = f_i$, където $i \in \{1, \ldots, r\}$,

то Γ е непрекъснат.

Упътване. Съгласно зад. 10 е достатъчно да покажете, че Γ е непрекъснат по всеки от аргументите си.

Задача 13. Нека $\Gamma_1, \ldots, \Gamma_m$ са непрекъснати оператори от тип $(n_1, \ldots, n_r \to n)$. Да се докаже, че следните оператори също са непрекъснати:

а) $\Gamma(f_1, \ldots, f_r) = g_0(\Gamma_1(f_1, \ldots, f_r), \ldots, \Gamma_m(f_1, \ldots, f_r))$
от тип $(n_1, \ldots, n_r \to n)$, където функцията $g_0 \in \mathcal{F}_m$ е фиксирана;

б) $\Gamma'(f_1, \ldots, f_r) = f_i(\Gamma_1(f_1, \ldots, f_r), \ldots, \Gamma_m(f_1, \ldots, f_r))$
от тип $(n_1, \ldots, n_r \to n)$, където $1 \leqq i \leqq r$.

Доказателство. а) Достатъчно е да покажем, че Γ е непрекъснат по всеки от аргументите си. За тази цел можем да се ограничим със случая $r = 1$. Да предположим още (без да ограничаваме общността), че $m = 2$. Непосредствено се проверява, че операторът

$$\Gamma(f) = g_0(\Gamma_1(f), \Gamma_2(f))$$

е монотонен. Нека $\{f_k\}$ е монотонно растяща редица от аргументи на Γ. Тъй като Γ е монотонен, според зад. 2 а) редицата $\{\Gamma(f_k)\}$ е монотонно растяща и

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

Ще покажем, че е в сила и обратното включване.

Нека $f = \bigcup f_k$ и $\Gamma(f)(\overline{x}) \simeq y$, т.е. $g_0(\Gamma_1(f)(\overline{x}), \Gamma_2(f)(\overline{x})) \simeq y$. От непрекъснатостта на Γ_1 и Γ_2 получаваме

$$\Gamma(f)(\overline{x}) \simeq g_0(\bigcup \Gamma_1(f_k)(\overline{x}), \bigcup \Gamma_2(f_k)(\overline{x})) \simeq y.$$

Следователно съществуват z_1 и z_2, за които

$$\bigcup \Gamma_1(f_k)(\overline{x}) \simeq z_1 \ \& \ \bigcup \Gamma_2(f_k)(\overline{x}) \simeq z_2 \ \& \ g_0(z_1, z_2) \simeq y.$$

Нека k_1 и k_2 са такива, че

$$\Gamma_1(f_{k_1})(\overline{x}) \simeq z_1 \text{ и } \Gamma_2(f_{k_2})(\overline{x}) \simeq z_2.$$

Тогава за $k = \max(k_1, k_2)$ ще имаме

$$g_0(\Gamma_1(f_k)(\overline{x}), \Gamma_2(f)(\overline{x}_k)) \simeq y \simeq \Gamma(f_k)(\overline{x}),$$

откъдето $\bigcup \Gamma(f_k)(\overline{x}) \simeq y$.

За б) разсъжденията са аналогични.

Задача 14. Нека Γ_1 и Γ_2 са непрекъснати оператори от тип $(n_1, \ldots, n_r \to n)$, а P е n-местен предикат в D^n. Да се докаже, че операторът

$$\Gamma(f_1, \ldots, f_r)(\overline{x}) \simeq \begin{cases} \Gamma_1(f_1, \ldots, f_r)(\overline{x}), & \text{ако } P(\overline{x}), \\ \Gamma_2(f_1, \ldots, f_r)(\overline{x}), & \text{ако } \neg P(\overline{x}) \end{cases}$$

също е непрекъснат.

Доказателство. Както в предишната задача, без ограничение на общността можем да предполагаме, че $r = 1$. Нека $\{f_k\}$ е монотонно растяща редица от аргументи на Γ, а $\overline{x} \in \mathrm{D}^n$. Ако $P(\overline{x})$, то (поради непрекъснатостта на Γ_1)

$$\Gamma(\bigcup f_k)(\overline{x}) \simeq \Gamma_1(\bigcup f_k)(\overline{x}) \simeq (\bigcup \Gamma(f_k))(\overline{x}).$$

Ако $\neg P(\overline{x})$, разсъждавайки както по-горе, отново получаваме

$$\Gamma(\bigcup f_k)(\overline{x}) \simeq (\bigcup \Gamma(f_k))(\overline{x}).$$

Следователно горното равенство е в сила за всяко $\overline{x} \in \mathrm{D}^n$, което означава, че операторът Γ е непрекъснат.

Задача 15. Като използвате предишните две задачи, покажете, че всеки от изброените оператори е непрекъснат.

а) $\Gamma_1(f) = f \circ f$;

б) $\Gamma_2(f)(x, y) \simeq f(g_0(x), f(x, g_1(y)))$, където g_0, g_1 са фиксирани функции;

в) $\Gamma_3(f)(x) \simeq \begin{cases} 1, & \text{ако } x \leqq 1, \\ f(x-1) + f(x-2), & \text{ако } x > 1, \end{cases}$

където $\Gamma_3 : \mathcal{F} = \{f \mid f : \mathbb{N} \multimap \mathbb{N}\} \to \mathcal{F}$;

г) $\Gamma_4(f)(x) \simeq \begin{cases} x, & \text{ако } \mathrm{lh}(x) \leqq 1, \\ \mathrm{append}(f(\mathrm{tail}(x)), \langle \mathrm{head}(x) \rangle), & \text{ако } \mathrm{lh}(x) > 1, \end{cases}$

където $\Gamma_4 : \mathcal{F} = \{f \mid f : \Sigma^* \rightarrow\!\!\!\!\rightarrow \Sigma^*\} \to \mathcal{F}$, а Σ^* е съвкупността от всички крайни списъци над азбуката Σ.

Упътване. а) За оператора Γ_1 можем да запишем $\Gamma_1(f) = f(\Gamma(f))$, където $\Gamma(f) = f$ е непрекъснат съгласно зад. 12 б). Оттук по зад. 13 б) и Γ_1 ще е непрекъснат.

В следващите няколко задачи ще разглеждаме оператори, преработващи монотонни функции $g : \mathrm{D}_\perp^n \to \mathrm{D}_\perp$. Да напомним, че $g : \mathrm{D}_\perp^n \to \mathrm{D}_\perp$ е монотонна, ако за всяко $\overline{x}$, $\overline{y} \in \mathrm{D}_\perp^n$ е изпълнено условието

$$\overline{x} \sqsubseteq \overline{y} \Longrightarrow g(\overline{x}) \sqsubseteq g(\overline{y}).$$

Да положим $\mathcal{G}_n = \{g \mid g : \mathrm{D}_\perp^n \to D_\perp \,\&\, g \text{ е монотонна}\}$. Съгласно зад. 3 $(\mathcal{G}_n, \sqsubseteq, \Omega)$ е област на Скот.

Задача 16. За кой от изброените оператори може да се твърди, че е непрекъснат или монотонен в областта на Скот $(\mathcal{G}_1, \sqsubseteq, \Omega)$:

а) $\Delta_1(g) = g_0$, където $g_0 : \mathrm{D}_\perp \to \mathrm{D}_\perp$ е фиксирана монотонна функция;

б) $\Delta_2(g) = g \circ g_0$, където $g_0 : \mathrm{D}_\perp \to \mathrm{D}_\perp$ е фиксирана монотонна функция;

в) $\Delta_3(g)(x) = \begin{cases} \perp, & \text{ако } x = \perp, \\ g(x), & \text{ако } x \neq \perp, \end{cases}$

г) $\Delta_4(g) = \begin{cases} g_1, & \text{ако } g = \Omega, \\ g_2, & \text{ако } g \neq \Omega, \end{cases}$

където g_1, $g_2 \in \mathcal{G}_1$ са фиксирани;

д) $\Delta_5(g) = \begin{cases} \Omega, & \text{ако } \perp \in \mathrm{range}(g), \\ g_0, & \text{в противен случай,} \end{cases}$

където $g_0 \in \mathcal{G}_1$ е фиксирана?

Упътване. д) Операторът Δ_5 е монотонен, но не винаги е непрекъснат.

Както по-горе, изображението $\Delta : \mathcal{G}_{n_1} \times \cdots \times \mathcal{G}_{n_r} \to \mathcal{G}_n$ ще наричаме оператор от тип $(n_1, \ldots, n_r \to n)$.

Задача 17. Нека $\Delta : \mathcal{G}_{n_1} \times \cdots \times \mathcal{G}_{n_r} \to \mathcal{G}_n$. Да се докаже, че ако:

а) Δ е константен, т.е. $\Delta(g_1, \ldots, g_r) = h_0$, където $h_0 \in \mathcal{G}_n$ е фиксирана;

б) Δ е проектиращ, т.е. $\Delta(g_1, \ldots, g_r) = g_i$, където $i \in \{1, \ldots, r\}$, то Δ е непрекъснат.

Задача 18. Нека $\Delta_1, \ldots, \Delta_m$ са непрекъснати оператори от тип $(n_1, \ldots, n_r \to n)$, а $h_0 : \mathrm{D}_\perp^m \to \mathrm{D}_\perp$ е фиксирана монотонна функция. Да положим

$$\Delta(g_1, \ldots, g_r) = h_0(\Delta_1(g_1, \ldots, g_r), \ldots, \Delta_m(g_1, \ldots, g_r)).$$

Да се докаже, че

а) при всеки избор на $\overline{g} \in \mathcal{G}_{n_1} \times \cdots \times \mathcal{G}_{n_r}$ функцията $\Delta(\overline{g})$ е монотонна;

б) операторът $\Delta : \mathcal{G}_{n_1} \times \cdots \times \mathcal{G}_{n_r} \to \mathcal{G}_n$ е непрекъснат.

Упътване. а) Нека $\overline{g} \in \mathcal{G}_{n_1} \times \cdots \times \mathcal{G}_{n_r}$. Всяка от функциите $\Delta_1(\overline{g}), \ldots, \Delta_m(\overline{g})$ е монотонна, следователно при всеки избор на $\overline{x}$, $\overline{y} \in \mathrm{D}_\perp^n$, за които $\overline{x} \sqsubseteq \overline{y}$, ще е изпълнено неравенството

$$(\Delta_1(\overline{g})(\overline{x}), \ldots, \Delta_m(\overline{g})(\overline{x})) \sqsubseteq (\Delta_1(\overline{g})(\overline{y}), \ldots, \Delta_m(\overline{g})(\overline{y})).$$

Оттук по монотонността на h_0 получаваме

$$h_0(\Delta_1(\overline{g})(\overline{x}), \ldots, \Delta_m(\overline{g})(\overline{x})) \sqsubseteq h_0(\Delta_1(\overline{g})(\overline{y}), \ldots, \Delta_m(\overline{g})(\overline{y})),$$

с други думи, $\Delta(\overline{g})(\overline{x}) \sqsubseteq \Delta(\overline{g})(\overline{y})$.

б) Съгласно току-що доказаното, операторът Δ преработва монотонни функции в монотонни функции. Тъй като е достатъчно да покажем, че операторът Δ е непрекъснат по всеки от аргументите си, можем да си мислим, че Δ има само един аргумент. Нека $\{g_k\}_k$ е монотонно растяща редица от аргументи на Δ. По зад. 3 точната горна граница $\bigcup g_k$ на тази редица също е монотонна функция и тя се определя с равенството

$$(\bigcup g_k)(\overline{x}) = \bigcup g_k(\overline{x}),$$

като точната горна граница в десния израз е в областта на Скот $(\mathrm{D}_\perp, \sqsubseteq, \perp)$. Като използвате наблюдението, че всяка монотонно растяща редица в $\mathrm{D}_\perp$ има само краен брой различни членове (зад. 16, § 3.1), съобразете, че за всяко $\overline{x} \in \mathrm{D}_\perp^n$ съществува k_0, за което

$$(\bigcup g_k)(\overline{x}) = g_{k_0}(\overline{x}).$$

Оттук нататък проверката за непрекъснатост на оператора Δ по същество повтаря разсъжденията от зад. 13 а).

Задача 19. Нека операторите $\Delta_1, \ldots, \Delta_m$ от тип $(n_1, \ldots, n_r \to n)$ са непрекъснати и $i \in \{1, \ldots, r\}$. Да положим

$$\Delta(g_1, \ldots, g_r) = g_i(\Delta_1(g_1, \ldots, g_r), \ldots, \Delta_m(g_1, \ldots, g_r)).$$

Да се докаже, че:

а) при всеки избор на $\overline{g} \in \mathcal{G}_{n_1} \times \cdots \times \mathcal{G}_{n_r}$ функцията $\Delta(\overline{g})$ е монотонна;

б) операторът $\Delta : \mathcal{G}_{n_1} \times \cdots \times \mathcal{G}_{n_r} \to \mathcal{G}_n$ е непрекъснат.

Упътване. Разсъждавайте както в доказателството на предишната задача.

Задача 20. Нека Δ_1 и Δ_2 са непрекъснати оператори от тип $(n_1, \ldots, n_r \to n)$, а $P : \mathrm{D}_\perp^n \to \{tt, ff\}_\perp$ е монотонно изображение. Да се докаже, че операторът

$$\Delta(g_1, \ldots, g_r)(\overline{x}) = \begin{cases} \Delta_1(g_1, \ldots, g_r)(\overline{x}), & \text{ако } P(\overline{x}) = tt, \\ \Delta_2(g_1, \ldots, g_r)(\overline{x}), & \text{ако } P(\overline{x}) = ff, \\ \perp, & \text{ако } P(\overline{x}) = \perp \end{cases}$$

също е непрекъснат.

Упътване. Повторете разсъжденията от доказателството на зад. 14.

Задача 21. Да се докаже, че всеки от изброените оператори е непрекъснат:

а) $\Delta_1(g) = g \circ g$, където $\Delta_1 : \mathcal{G}_1 = \{g \,|\, g : \mathrm{D}_\perp \to \mathrm{D}_\perp\} \to \mathcal{G}_1$;

б) $\Delta_2(g)(x) = \begin{cases} 1, & \text{ако } x = 0, \\ \perp, & \text{ако } x = \perp, \\ g(x-1).x, & \text{ако } x > 0, \end{cases}$

където $\Delta_2 : \mathcal{G}_1 = \{g \,|\, g : \mathbb{N}_\perp \to \mathbb{N}_\perp\} \to \mathcal{G}_1$;

в) $\Delta_3(g)(x,y) = \begin{cases} 0, & \text{ако } x = 0 \vee \ (x = \perp \ \& \, y = 0), \\ \perp, & \text{ако } x = \perp \ \& \, y > 0, \\ g(x-1, y) + x, & \text{ако } x > 0, \end{cases}$

където $\Delta_3 : \mathcal{G}_2 = \{g \,|\, g : \mathbb{Z}_\perp^2 \to \mathbb{Z}_\perp\} \to \mathcal{G}_2$.

Упътване. Разсъждавайте по аналогия с доказателството на зад. 15.

Нека X, X', Y и Y' са произволни множества. Да положим

$$\mathcal{F} = \{f \,|\, f : X \to\!\!\!\!\circ\; Y\} \text{ и } \mathcal{F}' = \{f \,|\, f : X' \to\!\!\!\!\circ\; Y'\}.$$

Казваме, че $\Gamma : \mathcal{F} \to \mathcal{F}'$ е *компактен*, ако за всяка функция $f \in \mathcal{F}$ е изпълнено условието

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

при всеки избор на $x \in X'$, $y \in Y'$.

В това определение имаме предвид, че функцията $\theta : X \to\!\!\!\!\circ\; Y$ е крайна, ако $\mathrm{dom}(\theta)$ е крайно множество. По-нататък с θ ще означаваме само крайни функции.

Задача 22. Да се докаже, че всеки компактен оператор $\Gamma : \mathcal{F} \to \mathcal{F}'$ е монотонен.

Упътване. Разсъждавайте както при доказателството на твърдението, че всеки компактен оператор, преработващ аритметични функции, е монотонен (зад. 3, § 2.2).

Задача 23. Нека $S \subseteq \mathcal{F}$ е насочено множество (т.е. за всяка функция f_1, $f_2 \in \mathcal{F}$ съществува функция $g \in S$, за която $f_1 \subseteq g$ и $f_2 \subseteq g$), а $\Gamma : \mathcal{F} \to \mathcal{F}'$ е монотонен оператор. Да се докаже, че

а) множеството $\Gamma(S) = \{\Gamma(f) \,|\, f \in S\} \subseteq \mathcal{F}'$ е насочено;

б) $\bigcup \Gamma(S) \subseteq \Gamma(\bigcup S)$.

Забележка. Да напомним, че съгласно зад. 15, § 3.1, точната горна граница $\bigcup S$ на всяко насочено множество S съществува и се определя с равенството

$$(\bigcup S)(x) \simeq y \iff \exists f_{\in S}(f(x) \simeq y).$$

Упътване. б) От монотонността на Γ имаме, че за всяка функция $f \in S$ е в сила включването $\Gamma(f) \subseteq \Gamma(\bigcup S)$. Следователно $\Gamma(\bigcup S)$ е горна граница на $\Gamma(S) = \{\Gamma(f) \,|\, f \in S\}$. Тъй като $\Gamma(S)$ е насочено множество, точната му горна граница $\bigcup \Gamma(S)$ съществува и $\bigcup \Gamma(S) \subseteq \Gamma(\bigcup S)$.

Задача 24. Нека $S \subseteq \mathcal{F}$ е насочено множество, а θ е крайна функция. Да се докаже, че е в сила условието

$$\theta \subseteq \bigcup S \Longrightarrow \exists f_{\in S}(\theta \subseteq f).$$

Упътване. Разсъждавайте както в зад. 12, § 2.2.

Задача 25. Нека $\Gamma : \mathcal{F} \to \mathcal{F}'$ е компактен оператор, а $S \subseteq \mathcal{F}$ е насочено множество. Да се докаже, че:

а) $\bigcup \Gamma(S) = \Gamma(\bigcup S)$;

б) всеки компактен оператор е непрекъснат (като изображение от областите на Скот $(\mathcal{F}, \subseteq, \varnothing)$ в $(\mathcal{F}', \subseteq, \varnothing)$).

Доказателство. а) Според зад. 22 операторът Γ е монотонен и оттук по зад. 23 б)

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

Ще покажем, че е в сила и обратното включване $\Gamma(\bigcup S) \subseteq \bigcup\Gamma(S)$.

Наистина, нека $\Gamma(\bigcup S)(x) \simeq y$. От компактността на Γ ще съществува крайна функция $\theta \subseteq f$, за която $\Gamma(\theta)(x) \simeq y$. От предишната задача следва, че θ се включва в f за някоя $f \in S$ и следователно $\Gamma(f)(x) \simeq y$. Оттук по определението на $\bigcup\Gamma(S)$ получаваме, че

$$(\bigcup\Gamma(S))(x) \simeq y.$$

б) Тъй като елементите на всяка монотонно растяща редица $\{f_k\}$ образуват насочено множество (зад. 12, § 3.1), от току-що доказаното следва в частност, че Γ е непрекъснат оператор.

Задача 26. Нека $\Gamma : \mathcal{F} = \{f \mid f : X \twoheadrightarrow Y\} \to \mathcal{F}'$ и множеството X е изброимо. Да се докаже, че

$$\Gamma \text{ е компактен} \iff \Gamma \text{ е непрекъснат}.$$

Доказателство. Правата посока на твърдението следва от предишната задача.

Обратно, нека Γ е непрекъснат и $f : X \twoheadrightarrow Y$. Нека $x_0, x_1, \ldots$ е произволно изброяване на елементите на множеството X. Да положим

$$f_k = f|_{\{x_0,\ldots,x_k\}}.$$

Ясно е, че всяка от функциите f_k е крайна и $f = \bigcup_k f_k$. От непрекъснатостта на Γ ще имаме $\Gamma(f) = \Gamma(\bigcup_k f_k) = \bigcup_k \Gamma(f_k)$. Тогава

$$\Gamma(f)(x) \simeq y \iff (\bigcup_k \Gamma(f_k))(x) \simeq y \iff \exists k(\Gamma(f_k)(x) \simeq y)).$$

Задача 27. Да се докаже, че следните условия са еквивалентни:

(1) $\Gamma : \mathcal{F} \to \mathcal{F}'$ е компактен оператор;

(2) за всяко насочено множество $S \subseteq \mathcal{F}$ е в сила $\bigcup\Gamma(S) = \Gamma(\bigcup S)$.

Упътване. Посоката (1) $\Rightarrow$ (2) следва от зад. 25.

Обратно, нека е изпълнено условието (2). За произволна функция $f \in \mathcal{F}$ да положим

$$S = \{f|_Z \mid Z \subseteq X \text{ е крайно}\}.$$

Покажете, че S е насочено множество и $\bigcup S = f$. Тогава

$$\Gamma(f) = \Gamma(\bigcup S) = \bigcup\Gamma(S) = \bigcup\{\Gamma(g) \mid g \in S\}.$$

Оттук ще имаме

$$\Gamma(f)(x) \simeq y \iff \exists g_{\in S}(\Gamma(g)(x) \simeq y) \iff \exists\theta_{\subseteq f}(\theta \text{ е крайна } \& \, \Gamma(\theta)(x) \simeq y)$$

и следователно Γ е компактен оператор.

Задача 28. Да се даде пример за оператор

$$\Gamma : \mathcal{F} = \{f \mid f : X \twoheadrightarrow Y\} \to \mathcal{F},$$

който е непрекъснат, но не е компактен.

Упътване. Забележете, че съгласно зад. 26 множеството X не трябва да е изброимо.

Нека X и Y са произволни множества, а $\theta : X \twoheadrightarrow Y$ е крайна функция. *Отворена околност* на θ ще наричаме множеството $\mathcal{N}_\theta = \{f \mid f \supseteq \theta\}$. Казваме, че множеството $\mathcal{O} \subseteq \mathcal{F} = \{f \mid f : X \twoheadrightarrow Y\}$ е *отворено*, ако е изпълнено условието

$$f \in \mathcal{O} \Longrightarrow \exists\theta_{\subseteq f}(\theta \text{ е крайна } \& \, \mathcal{N}_\theta \subseteq \mathcal{O}).$$

Задача 29. Да се докаже, че:

(1) $\varnothing$ и $\mathcal{F}$ са отворени множества;

(2) ако $\mathcal{O}_1$ и $\mathcal{O}_2$ са отворени, то и $\mathcal{O}_1 \cap \mathcal{O}_2$ е отворено;

(3) ако $\mathcal{F}^*$ е съвкупност от отворени множества, то и множеството $\bigcup\mathcal{F}^* = \{f \mid \exists\mathcal{O}(\mathcal{O} \in \mathcal{F}^* \,\&\, f \in \mathcal{O})\}$ също е отворено.

С други думи, фамилията $\mathcal{T} = \{\mathcal{O} \mid \mathcal{O} \subseteq \mathcal{F} \text{ е отворено}\}$ задава топология в $\mathcal{F}$.

Упътване. (2) Нека $f \in \mathcal{O}_1 \cap \mathcal{O}_2$. Тогава съществуват крайни функции $\theta_1 \subseteq f$ и $\theta_2 \subseteq f$, за които $\mathcal{N}_{\theta_1} \subseteq \mathcal{O}_1$ и $\mathcal{N}_{\theta_2} \subseteq \mathcal{O}_2$. Проверете, че $\mathcal{N}_{\theta_1 \cup \theta_2} \in \mathcal{O}_1 \cap \mathcal{O}_2$.

Нека Γ е изображение от $\mathcal{F} = \{f \mid f : X \twoheadrightarrow Y\}$ към $\mathcal{F}' = \{f \mid f : X' \twoheadrightarrow Y'\}$. Казваме, че Γ е *топологично непрекъснат* оператор, ако за всяко отворено множество $\mathcal{O}' \subseteq \mathcal{F}'$ е вярно, че множеството $\Gamma^{-1}(\mathcal{O}') = \{f \mid \Gamma(f) \in \mathcal{O}'\}$ е отворено.

Задача 30. Да се докаже, че операторът $\Gamma : \mathcal{F} \to \mathcal{F}'$ е компактен точно тогава, когато е топологично непрекъснат.

§ 3.3. Теорема за най-малката неподвижна точка

Нека $(A, \leqq, \perp)$ е област на Скот и $f : A \to A$. Казваме, че a е *неподвижна точка* на f, ако $f(a) = a$. Елементът a на A наричаме *най-малка неподвижна точка* на f, ако:

(1) $f(a) = a$ (a е неподвижна точка на f);

(2) $f(b) = b \Longrightarrow a \leqq b$ (a е най-малката сред неподвижните точки на f).

От условието (2) получаваме, че ако съществува, най-малката неподвижна точка на f е единствена.

Ще казваме, че $a \in A$ е *псевдонеподвижна точка* на f, ако $f(a) \leqq a$.

Задача 1. Да се даде пример за област на Скот $(A, \leqq, \perp)$ и изображение $f : A \to A$, за което е вярно, че:

а) f има неподвижна точка;

б) f няма неподвижни точки;

в) f има неподвижни точки, но няма най-малка неподвижна точка;

г) f има псевдонеподвижни точки, но няма неподвижна точка;

д) f няма псевдонеподвижни точки.

Упътване. а) Нека $f(a) = a$ за всяко $a \in A$. Тогава всеки елемент на A е неподвижна точка на f.

г) Разгледайте плоската област на Скот $(\mathrm{D}_\perp, \sqsubseteq, \perp)$ и изображението

$$f(a) = \begin{cases} \perp, & \text{ако } a \neq \perp, \\ a_0, & \text{ако } a = \perp, \end{cases}$$

където a_0 е фиксиран елемент от D.

д) Нека $(\mathrm{D}_\perp, \sqsubseteq, \perp)$ е плоска област на Скот и множеството D има поне два елемента. Нека $f : \mathrm{D}_\perp \to \mathrm{D}_\perp$ е изображение, удовлетворяващо условията

$$f(\perp) \in \mathrm{D}; \quad f(a) \notin \{\perp, a\} \text{ за всяко } a \in \mathrm{D}.$$

Проверете, че f няма псевдонеподвижни точки.

Задача 2 (Тарски). Нека f е монотонно изображение в областта на Скот $(A, \leqq, \perp)$ и $a \in A$ е най-малката псевдонеподвижна точка на f. Да се докаже, че a е най-малката неподвижна точка на f.

Доказателство. Имаме $f(a) \leqq a$ и от монотонността на f получаваме $f(f(a)) \leqq f(a)$, т.е. $f(a)$ е псевдонеподвижна точка на f.

Оттук $a \leqq f(a)$, тъй като a е най-малката сред псевдонеподвижните точки на f. От двете неравенства $a \leqq f(a)$ и $f(a) \leqq a$ получаваме $f(a) = a$, т.е. a е неподвижна точка на f.

Нека $f(b) = b$. Тогава в частност $f(b) \leqq b$ и следователно $a \leqq b$. Получихме, че всяка неподвижна точка b на f мажорира a. Следователно a е най-малката неподвижна точка на f.

Задача 3 (теорема на Кнастер — Тарски за съществуване на най-малка неподвижна точка). Нека $(A, \leqq, \perp)$ е област на Скот и $f : A \to A$ е непрекъснато изображение. Тогава f притежава най-малка неподвижна точка, която е точна горна граница на редицата $\{f^k(\perp)\}$.

Доказателство. Да съобразим първо, че редицата $\{f^k(\perp)\}$ е монотонно растяща. Наистина

$$f^0(\perp) = \perp \leqq f^1(\perp),$$

и ако допуснем, че за някое $k \in \mathbb{N}$ $f^k(\perp) \leqq f^{k+1}(\perp)$, то от монотонността на f ще следва, че

$$f(f^k(\perp)) \leqq f(f^{k+1}(\perp)), \text{ т.е. } f^{k+1}(\perp) \leqq f^{k+2}(\perp).$$

Тъй като наредбата $\leqq$ е пълна, редицата $\{f^k(\perp)\}$ притежава точна горна граница a. Ще покажем, че a е най-малката неподвижна точка на f.

(1) a е неподвижна точка на f, защото (от непрекъснатостта на f)

$$f(a) = f\Big(\bigcup_k f^k(\perp)\Big) = \bigcup_{k \geqq 0} f(f^k(\perp)) = \bigcup_{k \geqq 1} f^k(\perp) = \bigcup_{k \geqq 0} f^k(\perp) = a.$$

(2) Нека $f(b) = b$. Тогава $f^0(\perp) = \perp \leqq b$ и ако допуснем, че за някое k $f^k(\perp) \leqq b$, то от монотонността на f ще имаме, че

$$f^{k+1}(\perp) = f(f^k(\perp)) \leqq f(b) = b.$$

Следователно b мажорира всеки член на редицата $\{f^k(\perp)\}_k$, откъдето получаваме

$$\bigcup_k f^k(\perp) \leqq b,$$

т.е. $a \leqq b$.

Задача 4 (индукционно правило на Скот). Нека $f : A \to A$ е непрекъснато изображение в областта на Скот $(A, \leqq, \perp)$, а P е свойство в множеството A, за което са изпълнени следните три условия:

(1) $P(\perp)$;

(2) $P(a) \Longrightarrow P(f(a))$ за всяко $a \in A$;

(3) P е *непрекъснато*, т.е. за всяка монотонно растяща редица $\{a_k\}$ в A е вярно, че $(\forall k P(a_k)) \Longrightarrow P(\bigcup a_k)$.

Тогава P е в сила за най-малката неподвижна точка $f_{\min}$ на изображението f.

Доказателство. От теоремата на Кнастер — Тарски имаме

$$f_{\min} = \bigcup_k f^k(\perp).$$

Тъй като редицата $\{f^k(\perp)\}$ е монотонно растяща, а P е непрекъснато свойство, за да покажем $P(f_{\min})$ е достатъчно да проверим, че $P(f^k(\perp))$ за всяко k.

Наистина за $k = 0$ условието $P(f^0(\perp)) = P(\perp)$ е в сила съгласно (1), а ако допуснем, че $P(f^k(\perp))$, то от (2) веднага получаваме $P(f(f^k(\perp)))$, т.е. $P(f^{k+1}(\perp))$.

Задача 5. Нека $f : D_\perp \to D_\perp$ е монотонно изображение в плоската област на Скот $(D_\perp, \sqsubseteq, \perp)$. Да се докаже, че $f(\perp)$ е най-малката неподвижна точка на f.

Упътване. Според зад. 6, § 3.2, изображението f е непрекъснато, откъдето по теоремата на Кнастер — Тарски получаваме, че най-малката неподвижна точка $f_{\min}$ на f е определена и

$$f_{\min} = \bigcup f^k(\perp).$$

Съобразете, че $f^1(\perp) = f^2(\perp) = \cdots$, откъдето ще получите

$$\bigcup f^k(\perp) = f(\perp), \text{ т.е. } f_{\min} = f(\perp).$$

Задача 6. Нека $\mathcal{G} = \{g \mid g : \mathbb{Z}^2_\perp \to \mathbb{Z}_\perp\}$. Да разгледаме оператора $\Delta : \mathcal{G} \to \mathcal{G}$, определен с равенството

$$\Delta(g)(x,y) = \begin{cases} y+1, & \text{ако } x = y \in \mathbb{Z}, \\ g(x, g(x-1, y+1)), & \text{ако } x,\ y \in \mathbb{Z}\ \&\ x \neq y, \\ \perp, & \text{ако } x = \perp \vee y = \perp. \end{cases}$$

Проверете, че следните функции са неподвижни точки на оператора Δ:

$$g_1(x,y) = \begin{cases} x+1, & \text{ако } x,\ y \in \mathbb{Z}, \\ \perp, & \text{в противен случай,} \end{cases}$$

$$g_2(x,y) = \begin{cases} x+1, & \text{ако } x \geqq y, \\ y-1, & \text{ако } x < y, \\ \perp, & \text{ако } x = \perp \vee y = \perp, \end{cases}$$

$$g_3(x,y) = \begin{cases} x+1, & \text{ако } x \geqq y\ \&\ (x-y) \text{ е четно,} \\ \perp, & \text{в останалите случаи.} \end{cases}$$

Задача 7. Като използвате конструкцията от теоремата на Кнастер — Тарски, намерете най-малките неподвижни точки на следните оператори:

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

където $\Gamma : \mathcal{F} = \{f \mid f : \mathbb{N}^2 \rightharpoonup \mathbb{N}\} \to \mathcal{F}$;

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

където $\Delta : \mathcal{G} = \{g \mid g : \mathbb{N}^2_\perp \to \mathbb{N}_\perp\} \to \mathcal{G}$.

Забележка. В определението на оператора Δ имаме предвид, че функциите $\lambda x.x-1$ и $\lambda x,y.xy$ (като функции в $\mathbb{N}_\perp$) са точни продължения на съответните аритметични функции, т.е. $\perp -1 = \perp$ и $x.y = \perp$, ако $x = \perp$ или $y = \perp$.

Доказателство. а) За да приложим теоремата на Кнастер — Тарски, най-напред трябва да установим, че Γ е непрекъснат оператор. За тази цел можем да процедираме по два начина:

— да проверим непосредствено, че Γ е непрекъснат оператор (например както в доказателството на зад. 15, § 3.2) или

— да покажем, че операторът Γ е компактен, и да използваме факта, че всеки компактен оператор е непрекъснат (зад. 25 б), § 3.2).

Ние ще предпочетем втория начин, тъй като условието за компактност се проверява по-кратко.

Наистина, нека $f \in \mathcal{F}$ е произволна аритметична функция, а $(x, y) \in \mathbb{N}^2$.

Ако $x = 0$, то $\Gamma(f)(x, y)$ не зависи от f и следователно за всяка крайна подфункция θ на f ще имаме

$$\Gamma(\theta)(x, y) = \Gamma(f)(x, y) = 1.$$

Нека $x > 0$ и $\Gamma(f)(x, y) \simeq z$. Резултатът z зависи от стойността на функцията f в точките (x, y) и $(x-1, y')$, където $y' = f(x, y)$. Тъй като тези точки принадлежат на $\mathrm{dom}(f)$, за крайната функция

$$\theta = f|_{\{(x,y),(x-1,y')\}}$$

ще е изпълнено $\Gamma(\theta)(x, y) \simeq z$.

От теоремата на Кнастер — Тарски ще имаме, че най-малката неподвижна точка f_Γ на оператора Γ е точна горна граница на редицата $\{f_k\}$, където $f_k = \Gamma^k(\varnothing)$.

Да видим как изглежда всяка функция f_k от тази редица. По определение

$$f_0 = \varnothing \quad \text{и} \quad f_{k+1} = \Gamma^{k+1}(\varnothing) = \Gamma(\Gamma^k(\varnothing)) = \Gamma(f_k).$$

За $k = 1, 2$ ще имаме последователно

$$f_1(x, y) \simeq \Gamma(\varnothing)(x, y) \simeq \begin{cases} 1, & \text{ако } x = 0, \\ \varnothing(x-1, \varnothing(x, y)).x, & \text{ако } x > 0 \end{cases}$$

$$\simeq \begin{cases} 1, & \text{ако } x = 0, \\ \neg!, & \text{ако } x > 0, \end{cases}$$

$$f_2(x, y) \simeq \Gamma(f_1)(x, y) \simeq \begin{cases} 1, & \text{ако } x = 0, \\ f_1(x-1, f_1(x, y)).x, & \text{ако } x > 0 \end{cases}$$

$$\simeq \begin{cases} 1, & \text{ако } x = 0, \\ \neg!, & \text{ако } x > 0 \end{cases} \simeq f_1(x, y).$$

Ако допуснем, че $f_k = f_1$ за някое $k > 1$, то за f_{k+1} ще имаме $f_{k+1} = \Gamma(f_k) = \Gamma(f_1) = f_1$.

Следователно $\bigcup\limits_k f_k = f_1$, т.е. най-малката неподвижна точка на оператора Γ е

$$f_\Gamma(x, y) \simeq \begin{cases} 1, & \text{ако } x = 0, \\ \neg!, & \text{ако } x > 0. \end{cases}$$

б) Най-напред трябва да проверим, че операторът Δ е непрекъснат в областта на Скот ($\mathcal{G} = \{g \,|\, g : \mathbb{N}_\perp^2 \to \mathbb{N}_\perp\}, \sqsubseteq, \Omega$) (забележете, че в тази област не можем да говорим за компактност на Δ).

Нека $\Delta_1(g)(x, y) = g(x \dot- 1, g(x, y))$. Оператора Δ_1 можем да представим по следния начин: $\Delta_1(g) = g(\Delta_2(g), \Delta_3(g))$, където операторите $\Delta_2(g) = \lambda x.x \dot- 1$ и $\Delta_3(g) = g$ са непрекъснати, съгласно зад. 17, § 3.2. Тогава по зад. 19, § 3.2, и операторът Δ_2 е непрекъснат.

Нека $\Delta'(g) = g_0(\Delta_1(g), \Delta_4(g))$, където $g_0 = \lambda x, y.xy$, а $\Delta_4(g) = \boldsymbol{I}_1^2$. Съгласно зад. 18, § 3.2, операторът Δ' е непрекъснат. За да довършите проверката на непрекъснатостта на Δ, вземете под внимание този факт и зад. 20, § 3.2.

Сега от теоремата на Кнастер — Тарски получаваме, че най-малката неподвижна точка g_Δ на оператора Δ е точна горна граница на редицата $\{\Delta^k(\Omega)\}$ в областта на Скот

$$(\mathcal{G} = \{g \,|\, g : \mathbb{N}_\perp^2 \to \mathbb{N}_\perp\}, \sqsubseteq, \Omega).$$

За да намерим g_Δ, пресмятаме последователно всяка функция $g_k = \Delta^k(\Omega)$.

Имаме $g_0 = \Omega$, а от определението на Δ получаваме

$$g_1(x, y) = \Delta(\Omega)(x, y) = \begin{cases} 1, & \text{ако } x = 0, \\ \Omega(x-1, \Omega(x, y)).x, & \text{ако } x > 0, \\ \perp, & \text{ако } x = \perp \end{cases}$$

$$= \begin{cases} 1, & \text{ако } x = 0, \\ \perp, & \text{ако } x = \perp \vee x > 0, \end{cases}$$

$$g_2(x, y) = \Delta(g_1)(x, y) = \begin{cases} 1, & \text{ако } x = 0, \\ g_1(x-1, g_1(x, y)).x, & \text{ако } x > 0, \\ \perp, & \text{ако } x = \perp. \end{cases}$$

Да отбележим, че $g_1(0, z) = 1$ за всяко $z \in \mathbb{N}_\perp$, включително при $z = \perp$, откъдето

$$g_2(1, y) = g_1(0, g_1(1, y)).1 = g_1(0, \perp) = 1.$$

(Забележете разликата с функцията f_2 от предишния пример, където получихме

$$f_2(1, y) \simeq f_1(0, f_1(1, y)) \simeq \neg!,$$

тъй като стойността на израза $f_1(1,y)$ не е определена.)

За функцията g_2 получаваме окончателно

$$g_2(x,y) = \Delta(g_1)(x,y) = \begin{cases}1, & \text{ако } x=0,\\ 1, & \text{ако } x=1,\\ \perp, & \text{ако } x=\perp \vee x>1\end{cases}$$

$$= \begin{cases} x!, & \text{ако } 0 \leqq x < 2,\\ \perp, & \text{иначе.}\end{cases}$$

Да допуснем, че за някое $k \geqq 2$ е изпълнено

$$g_k(x,y) = \begin{cases} x!, & \text{ако } 0\leqq x<k,\\ \perp, & \text{иначе.}\end{cases}$$

Тогава

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

$$=\begin{cases}1, & \text{ако } x=0,\\ (x-1)!.x, & \text{ако } x>0 \ \&\ 0\leqq x-1<k,\\ \perp, & \text{в останалите случаи}\end{cases}$$

$$=\begin{cases}x!, & \text{ако } 0\leqq x<k+1,\\ \perp, & \text{в останалите случаи.}\end{cases}$$

Така получихме, че за всяко k

$$g_k(x,y) = \begin{cases} x!, & \text{ако } 0\leqq x<k,\\ \perp, & \text{иначе.}\end{cases}$$

Тогава за произволно $x\in\mathbb{N}$ ще имаме

$$g_\Delta(x,y) = (\bigcup_k g_k)(x,y) = \bigcup_k (g_k(x,y)) = x!,$$

а за $x=\perp$ получаваме

$$g_\Delta(\perp,y) = (\bigcup_k g_k)(\perp,y) = \bigcup_k (g_k(\perp,y)) = \bigcup\{\perp,\perp,\dots\} = \perp\ .$$

Следователно

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

Задача 8. Да се докаже, че най-малката неподвижна точка на оператора $\Gamma : \mathcal{F} = \{f \mid f: \mathbb{Z} \rightharpoonup \mathbb{Z}\} \to \mathcal{F}$:

$$\Gamma(f)(x) \simeq \begin{cases} x+1, & \text{ако } x<0,\\ f(f(x-2)), & \text{ако } x\geqq 0\end{cases}$$

е функцията $f_0(x) \simeq \begin{cases} x+1, & \text{ако } x<0,\\ 0, & \text{ако } x\geqq 0.\end{cases}$

Задача 9. Нека A е произволно множество, а $R\subseteq \mathbb{N}\times A$. Да се докаже, че съществува най-малка (относно частичната наредба $\subseteq$) функция $f:\mathbb{N}\times A \rightharpoonup \mathbb{N}\times A$, удовлетворяваща условията

$$(x,a)\in R \Longrightarrow f(x,a)=x; \quad (x,a)\notin R \Longrightarrow f(x,a) = f(x+1,a).$$

Да се намери явният вид на f.

Упътване. Покажете, че изображението

$$\Gamma(g)(x,a) \simeq \begin{cases} x, & \text{ако } (x,a)\in R,\\ g(x+1,a), & \text{в противен случай}\end{cases}$$

е непрекъснато в областта на Скот $(\{f\mid f:\mathbb{N}\times A \rightharpoonup \mathbb{N}\times A\}, \subseteq, \varnothing)$. Тогава съществуването на функцията f ще се гарантира от теоремата на Кнастер — Тарски.

Задача 10. Нека $(A_1, \leqq_1, \perp_1), \dots, (A_n, \leqq_n, \perp_n)$ са области на Скот, а f_i, $1\leqq i\leqq n$, са непрекъснати изображения от $A = A_1\times\dots\times A_n$ в A_i.

Да определим $f: A\to A$ с полагането:

$$f(\overline{a}) = (f_1(\overline{a}),\dots,f_n(\overline{a})).$$

Да се докаже, че най-малката неподвижна точка $f_{\min}$ на изображението f съществува и се определя с равенството

$$f_{\min} = (\bigcup_k a_k^1, \dots, \bigcup_k a_k^n),$$

където $a_0^i = \perp_i$ и $a_{k+1}^i = f_i(a_k^1,\dots,a_k^n)$ за $i=1,\dots,n$.

Упътване. Съгласно зад. 8, § 3.2, f е непрекъснато изображение. Приложете конструкцията от теоремата на Кнастер — Тарски към това изображение.

Задача 11. Да определим операторите

$$\Delta_1 : \mathcal{G} = \{g\mid g:\mathbb{N}_\perp\to\mathbb{N}_\perp\}\times\mathcal{G}\to\mathcal{G}, \quad \Delta_2:\mathcal{G}\times\mathcal{G}\to\mathcal{G}$$

по следния начин:

$$\Delta_1(g,h)(x) = \begin{cases} 1, & \text{ако } x = 0, \\ g(x-1) + h(x-1), & \text{ако } x > 0, \\ \perp, & \text{ако } x = \perp; \end{cases}$$

$$\Delta_2(g,h)(x) = \begin{cases} 0, & \text{ако } x = 0, \\ h(x+1), & \text{ако } x > 0, \\ \perp, & \text{ако } x = \perp. \end{cases}$$

Да се намерят всички неподвижни точки на оператора

$$\Delta : \mathcal{G} \times \mathcal{G} \to \mathcal{G} \times \mathcal{G},$$

който се определя с равенството

$$\Delta(g,h) = (\Delta_1(g,h), \Delta_2(g,h)).$$

Упътване. Нека (g,h) е неподвижна точка на Δ. Тогава

$$\Delta_1(g,h) = g \text{ и } \Delta_2(g,h) = h.$$

Покажете, че в такъв случай е изпълнено едно от следните две условия:

(1) $$g(x) = \begin{cases} 1, & \text{ако } x \leq 1 \\ \perp, & \text{иначе} \end{cases} \quad \text{и } h(x) = \begin{cases} 0, & \text{ако } x = 0, \\ \perp, & \text{иначе;} \end{cases}$$

съществува $a \in \mathbb{N}$, за което

(2) $$g(x) = \begin{cases} 1, & \text{ако } x \leq 1 \\ a.(x-1)+1, & \text{ако } x > 1 \\ \perp, & \text{ако } x = \perp \end{cases} \quad \text{и } h(x) = \begin{cases} 0, & \text{ако } x = 0, \\ a, & \text{ако } x > 0, \\ \perp, & \text{ако } x = \perp. \end{cases}$$

Задача 12. Нека $\mathcal{F} = \{f \mid f : \mathbb{N} \times \mathbb{N} \rightharpoonup \mathbb{Z}\}$. Да определим операторите $\Gamma_1 : \mathcal{F} \times \mathcal{F} \to \mathcal{F}$ и $\Gamma_2 : \mathcal{F} \times \mathcal{F} \to \mathcal{F}$, както следва:

$$\Gamma_1(f,g)(x,y) \simeq \begin{cases} 1, & \text{ако } x = y, \\ f(x, y-x) - g(x, y-x), & \text{ако } x < y, \\ g(y,x), & \text{ако } x > y, \end{cases}$$

$$\Gamma_2(f,g)(x,y) \simeq \begin{cases} 0, & \text{ако } x = y, \\ g(x, y-x), & \text{ако } x < y, \\ f(y,x), & \text{ако } x > y. \end{cases}$$

Да се докаже, че съществуват единствени функции f^* и g^*, за които

$$f^* = \Gamma_1(f^*, g^*), \; g^* = \Gamma_2(f^*, g^*),$$

и тези функции удовлетворяват условието:

$$f^*(x,y).x + g^*(x,y).y = \text{НОД}(x,y).$$

Наредената двойка $(G, \to_G)$, където G е множество, а $\to_G$ е бинарна релация в него, ще наричаме *граф*. Елементите на G са върховете на графа, а наредените двойки от релацията $\to_G$ са негови ребра. *Транзитивна обвивка* на $(G, \to_G)$ наричаме графа $(G, \Longrightarrow_G)$, където релацията $\Longrightarrow_G$ се дефинира с условието:

$$x \Longrightarrow_G y \iff$$
$$\exists n_{n>0} \exists z_0 \ldots \exists z_n (z_0 = x \;\&\; z_n = y \;\&\; z_0 \to_G z_1 \;\&\; \ldots \;\&\; z_{n-1} \to_G z_n).$$

Задача 13. Нека $(G, \to_G)$ е произволен граф. Да определим изображенията $\Gamma_i : G \times G \to G \times G$, $i = 1, 2, 3$, както следва:

$$(x,y) \in \Gamma_1(R) \iff x \to_G y \;\vee\; \exists z (x \to_G z \;\&\; (z,y) \in R);$$
$$(x,y) \in \Gamma_2(R) \iff x \to_G y \;\vee\; \exists z ((x,z) \in R \;\&\; z \to_G y);$$
$$(x,y) \in \Gamma_3(R) \iff x \to_G y \vee \exists z (x \to_G z \;\&\; z \to_G y) \vee$$
$$\exists z \exists t (x \to_G z \;\&\; (z,t) \in R \;\&\; t \to_G y).$$

Да се докаже, че изображенията Γ_1, Γ_2 и Γ_3 са монотонни в областта на Скот $(G \times G, \subseteq, \varnothing)$, където $\subseteq$ е покомпонентното включване. Проверете, че транзитивната обвивка $\Longrightarrow_G$ е най-малката неподвижна точка и на трите изображения.

Задача 14. Нека $(A, \leqq, \perp)$ е област на Скот, а $f : A^2 \to A$ е непрекъснато изображение. За всяко $a \in A$ да означим с $g(a)$ най-малката неподвижна точка на изображението $\lambda b.f(a,b)$. Да се докаже, че изображението $g : A \to A$ също е непрекъснато.

Упътване. С индукция по k дефинираме редица $\{g_k\}$, както следва:

$$g_0 = \lambda a.\perp; \quad g_{k+1} = \lambda a.f(a, g_k(a)).$$

Покажете, че всяко от изображенията g_k е непрекъснато и редицата $\{g_k\}$ е монотонно растяща (относно поточковата наредба, определена от $\leqq$). От зад. 4, § 3.2, ще получите, че точната горна граница $\bigcup g_k$ на тази редица също е непрекъснато изображение.

Съобразете, че $g = \bigcup g_k$, откъдето ще следва, че g е непрекъснато изображение.

Казваме, че частично нареденото множество $(A, \leqq)$ е *пълна решетка*, ако всяко непразно множество $B \subseteq A$ притежава точна горна и точна долна граница.

Задача 15 (теорема на Тарски). Нека $(A, \leqq)$ е пълна решетка, а $f : A \to A$ е монотонно изображение. Да се докаже, че най-малката неподвижна точка $f_{\min}$ и най-голямата неподвижна точка $f_{\max}$ на изображението f са определени и удовлетворяват равенствата

$$f_{\min} = \bigcap\{a \mid f(a) \leqq a\} \quad \text{и} \quad f_{\max} = \bigcup\{a \mid a \leqq f(a)\}.$$

Упътване. Нека $\top = \bigcup A$. Тъй като $\top$ е най-големият елемент на A, то $\top \in \{a \mid f(a) \leqq a\}$. Тогава множеството $\{a \mid f(a) \leqq a\} \neq \varnothing$ и следователно точната му долна граница

$$b = \bigcap\{a \mid f(a) \leqq a\}$$

е определена. Съобразете, че $f(b) \leqq b$. От това ще следва, че b е най-малката псевдонеподвижна точка на f, откъдето от зад. 2 ще получите, че b е най-малката неподвижна точка на f.

Нека $(A, \leqq, \perp)$ е област на Скот. Съгласно зад. 4, § 3.2, наредената тройка

$$(\mathcal{G}_c = \{g \mid g : A \to A \ \&\ g \text{ е непрекъсната}\}, \leqq, \Omega = \lambda x.\, \perp),$$

където

$$g \leqq h \iff \forall x_{x \in A}(g(x) \leqq h(x)),$$

също е област на Скот.

Задача 16. При горните означения да определим $\Gamma : \mathcal{G}_c \to A$ посредством равенството:

$$\Gamma(g) = \text{най-малката неподвижна точка на } g.$$

Да се докаже, че Γ е непрекъснато изображение от областта на Скот $(\mathcal{G}_c, \leqq, \Omega)$ към $(A, \leqq, \perp)$.

Упътване. Най-напред съобразете, че изображението

$$\Gamma(g) = \bigcup g^k(\perp)$$

е монотонно.

Да изберем произволна монотонно растяща редица $\{g_k\}$ от $\mathcal{G}_c$. Тъй като изображението Γ е монотонно, ще е изпълнено неравенството

$$\bigcup \Gamma(g_k) \leqq \Gamma(\bigcup g_k).$$

За да покажете обратната посока на това неравенство, с индукция по k съобразете, че

$$(\bigcup_k g_k)^n(\perp) = \bigcup_k ((g_k)^n(\perp)).$$

Тогава

$$\Gamma(\bigcup_k g_k) = \bigcup_n (\bigcup_k g_k)^n(\perp) = \bigcup_n \bigcup_k (g_k)^n(\perp) = \bigcup_k \bigcup_n (g_k)^n(\perp) = \bigcup_k \Gamma(g_k).$$

Задача 17. Нека f и g са непрекъснати изображения в областта на Скот $(A, \leqq, \perp)$. Нека b е най-малката неподвижна точка на $f \circ g = \lambda a. f(g(a))$ и c е най-малката неподвижна точка на $g \circ f$. Да се докаже, че $f(c)$ и $g(b)$ са най-малките неподвижни точки съответно на $f \circ g$ и $g \circ f$.

Задача 18. Нека f е монотонно изображение в областта на Скот $(A, \leqq, \perp)$. Да се докаже, че f притежава най-малка неподвижна точка.

Упътване. С трансфинитна индукция по ξ дефинирайте следната монотонно растяща редица $\{f^\xi\}_\xi$:

$$f^0 = \perp; \quad f^\xi = f(\bigcup_{\eta < \xi} f^\eta) \text{ при } \xi > 0.$$

Покажете, че съществува (най-малък) ординал ζ, за който $f^\zeta = f^{\zeta+1}$. Тогава

$$f^{\zeta+1} = f(\bigcup_{\eta < \zeta+1} f^\eta) = f(\bigcup_{\eta \leq \zeta} f^\eta) = f(f^\zeta),$$

и следователно f^ζ е неподвижна точка на f.

Нека b е друга неподвижна точка на f. С трансфинитна индукция по ξ лесно се съобразява, че $f^\xi \leqq b$ при всеки избор на ξ. Оттук, в частност, $f^\zeta \leqq b$, т.е. f^ζ е най-малката неподвижна точка на f.

Задача 19. Да се даде пример за изображение $f : A \to A$, което не е монотонно, но има най-малка неподвижна точка.

Упътване. Нека $(\mathrm{D}_\perp, \sqsubseteq, \perp)$ е плоска област на Скот и D има поне два елемента b и c. Да положим

$$f(a) = \begin{cases} b, & \text{ако } a = \perp, \\ c, & \text{ако } a \neq \perp. \end{cases}$$

Изображението f не е монотонно, защото $\perp \sqsubseteq c$, но $f(\perp) = b \not\sqsubseteq c = f(c)$. Съобразете, че c е единствената, а следователно и най-малката, неподвижна точка на f.

Четвърта глава

РЕКУРСИВНИ ПРОГРАМИ

§ 4.1. Денотационна семантика на рекурсивните програми с предаване на параметрите по стойност

Нека D е произволно непразно множество от обекти и в него са зададени някакви операции и релации. Множеството D, заедно с дадените операции и релации, образуват *основния* тип данни, който ще означаваме с Data. Всяка n-местна операция в D ще считаме, че е от тип $D^n \to D$. Предполагаме също, че имаме още един *булев* тип данни Bool, който се състои от множеството $B = \{tt, ff\}$ и операциите $\&, \vee, \neg$. Така релациите в основния тип данни ще разглеждаме като операции от тип $D^k \to B$, а булевите операции $\&, \vee$ от тип $B^2 \to B$ и $\neg$ от тип $B \to B$. За всички операции в двата типа данни ще използваме термина *основни операции*.

Примери:

а) $\mathrm{Nat} = \langle \mathbb{N};\ +,\ \dot{-}, *;\ \leqq, <, = \rangle$ е типът на естествените числа, в който операциите са $+, \dot{-}, *$, а релациите $\leqq, <, =$.

б) $\mathrm{Int} = \langle \mathbb{Z};\ +, -, *;\ \leqq, <, = \rangle$ е типът на целите числа;

в) $\mathrm{List} = \langle L^*; \Lambda,\ \mathrm{head},\ \mathrm{tail},\ \mathrm{cons}; \prec, = \rangle$ е типът на думите в азбуката L, където L е дадено множество от символи, L^* е множеството от думите в L, Λ е празната дума, а операциите са:

$\mathrm{head}(\Lambda) = \Lambda$ и ако a е непразна дума, то $\mathrm{head}(a)$ е първият символ на a;

$\mathrm{tail}(\Lambda) = \Lambda$ и ако a е непразна дума, то $\mathrm{tail}(a)$ е опашката на a, т.е. дума, получена от a без първия символ;

$\mathrm{cons}(\sigma, a)$ е думата с първи символ σ и опашка a. Тук $\sigma \in L$, $a \in L^*$;

$a \prec b$, ако a е поддума на b и $a \neq b$.

Синтаксиса на рекурсивните програми ще опишем на функционален език, който съдържа следните елементи.

Символи:

а) константи $a, b, c, \ldots$ за означаване на елементите на D и B;

б) функционални символи $f_1, f_2, \ldots,\ \pi, p_1, p_2, \ldots$, за основните операции;

в) обектови променливи $X_1, X_2, \ldots, Y_1, Y_2, \ldots$ от тип D;

г) функционални променливи F_k^n, $n = 1, 2 \ldots$; $k = 0, 1, \ldots$, където всяка променлива с горен индекс n е от тип $D^n \to D$.

Когато е ясно от контекста, ще изпускаме горните индекси на функционалните променливи.

Термове от тип D:

(1) Всяка константа от тип D;

(2) Всяка обектова променлива;

(3) Ако f е основна операция от тип $D^n \to D$, а $\tau_1, \ldots, \tau_n$ са термове от тип D, то $f(\tau_1, \ldots, \tau_n)$ е терм от тип D;

(4) Ако F е функционална променлива от тип $D^n \to D$, а $\tau_1, \ldots, \tau_n$ са термове от тип D, то $F(\tau_1, \ldots, \tau_n)$ е терм от тип D.

Термове от тип B:

(1) Булевите константи tt и ff;

(2) Ако p е основна операция от тип $D^n \to B$, а $\tau_1, \ldots, \tau_n$ са термове от тип D, то $p(\tau_1, \ldots, \tau_n)$ е терм от тип B;

(3) Ако π е основна операция от тип $B^n \to B$, а $p_1, \ldots, p_n$ са термове от тип B, то $\pi(p_1, \ldots, p_n)$ е терм от тип B.

Условни термове:

(1) Ако p е терм от тип B, а τ_1 и τ_2 са термове от тип D, то

$$\texttt{if } p \texttt{ then } \tau_1 \texttt{ else } \tau_2$$

е условен терм;

(2) Ако p е терм от тип B, а τ_1 и τ_2 са условни термове или термове от тип D, то

$$\texttt{if } p \texttt{ then } \tau_1 \texttt{ else } \tau_2$$

е условен терм.

Рекурсивна програма в типа Data се нарича синтактичен обект R от следния вид:

$$\tau_0(X_1, \ldots, X_n, F_1^{m_1}, \ldots, F_k^{m_k}), \quad \texttt{where}$$
$$F_1^{m_1}(X_1, \ldots, X_{m_1}) = \tau_1(X_1, \ldots, X_{m_1}, F_1^{m_1}, \ldots, F_k^{m_k})$$
$$\ldots\ldots\ldots$$
$$F_k^{m_k}(X_1, \ldots, X_{m_k}) = \tau_k(X_1, \ldots, X_{m_k}, F_1^{m_1}, \ldots, F_k^{m_k}),$$

като всеки от термовете $\tau_1, \ldots, \tau_k$ е от тип D или е условен терм, а τ_0 е от тип D.

Примери:

Програма в типа Nat:

$F(X, 0)$, where
$F(X, Y) = \texttt{if } X = Y \texttt{ then } 1 \texttt{ else } F(X, Y+1) * (Y+1).$

Програма в типа List:

$F(X, Y)$, where
$F(X, Y) = \texttt{if } X = \Lambda \texttt{ then } Y \texttt{ else } \mathrm{cons}(\mathrm{head}(X), F(\mathrm{tail}(X), Y)).$

Ще опишем денотационната семантика на рекурсивните програми с предаване на параметрите по стойност.

Нека е даден терм $\tau(X_1, \dots, X_n, F_1^{m}, \dots, F_k^{m_k})$. Нека $a_1, \dots, a_n \in \mathrm{D}$ и $\varphi_1, \dots, \varphi_k$ са частични функции съответно на $m_1, \dots, m_k$ аргумента в D.

Стойността $\tau(a_1, \dots, a_n, \varphi_1, \dots, \varphi_k)$, съкратено $\tau(\overline{a}, \overline{\varphi})$, на терма τ се дефинира индуктивно:

а) ако τ е константата c, то $\tau(\overline{a}, \overline{\varphi}) \simeq c$;

б) ако $\tau = X_i$, то $\tau(\overline{a}, \overline{\varphi}) \simeq a_i$;

в) ако $\tau = f(\tau_1, \dots, \tau_l)$, където f е l-местна основна операция, то
$$\tau(\overline{a}, \overline{\varphi}) \simeq f(\tau_1(\overline{a}, \overline{\varphi}), \dots, \tau_l(\overline{a}, \overline{\varphi}));$$

г) ако $\tau = \texttt{if } p \texttt{ then } \tau_1 \texttt{ else } \tau_2$, то
$$\tau(\overline{a}, \overline{\varphi}) \simeq \begin{cases} \tau_1(\overline{a}, \overline{\varphi}), & \text{ако } p(\overline{a}, \overline{\varphi}) \simeq tt, \\ \tau_2(\overline{a}, \overline{\varphi}), & \text{ако } p(\overline{a}, \overline{\varphi}) \simeq ff, \\ \neg!, & \text{ако } \neg! p(\overline{a}, \overline{\varphi}); \end{cases}$$

д) ако $\tau = F_i^{m_i}(\tau_1, \dots, \tau_{m_i})$, то
$$\tau(\overline{a}, \overline{\varphi}) \simeq \varphi_i(\tau_1(\overline{a}, \overline{\varphi}), \dots, \tau_{m_i}(\overline{a}, \overline{\varphi})).$$

Нека $\mathcal{F}_n$ е съвкупността от всички частични функции на n аргумента в D. Съгласно зад. 2, § 3.1, наредената тройка $(\mathcal{F}_n, \subseteq, \varnothing^{(n)})$, където $\subseteq$ е релацията включване, а $\varnothing^{(n)}$ е никъде недефинираната функция, е област на Скот.

Нека $\tau(X_1, \dots, X_n, F_1^{m_1}, \dots, F_k^{m_k})$ е терм от тип D или условен терм. Да означим с $\mathcal{F} = \mathcal{F}_{m_1} \times \dots \times \mathcal{F}_{m_k}$ декартовото произведение на множествата $\mathcal{F}_{m_1}, \dots, \mathcal{F}_{m_k}$. От зад. 7, § 3.1, знаем, че множеството $\mathcal{F}$, наредено с покомпонентната наредба включване и най-малък елемент $\varnothing$, е област на Скот. С терма τ свързваме изображението Γ_τ на $\mathcal{F}$ в $\mathcal{F}_n$, дефинирано с условното равенство
$$\Gamma_\tau(\varphi_1, \dots, \varphi_k)(a_1, \dots, a_n) \simeq \tau(a_1, \dots, a_n, \varphi_1, \dots, \varphi_k)$$
за всеки избор на $a_1, \dots, a_n \in \mathrm{D}$ и $\varphi_1, \dots, \varphi_k$ — частични функции съответно на $m_1, \dots, m_k$ аргумента в D.

Задача 1. Нека $\tau(X_1, \dots, X_n, F_1^{m_1}, \dots, F_k^{m_k})$ е терм от тип D или условен терм. Да се докаже, че изображението $\Gamma_\tau : \mathcal{F} \to \mathcal{F}_n$ е непрекъснато.

Упътване. Приложете зад. 13 и зад. 14, § 3.2, или покажете, че $\Gamma\tau$ е компактен оператор, и използвайте факта, че всеки компактен оператор е непрекъснат (зад. 25, § 3.2).

Нека R е програмата в типа Data
$\tau_0(X_1, \dots, X_n, F_1^{m_1}, \dots, F_k^{m_k})$, where
$F_1^{m_1}(X_1, \dots, X_{m_1}) = \tau_1(X_1, \dots, X_{m_1}, F_1^{m_1}, \dots, F_k^{m_k})$
....
$F_k^{m_k}(X_1, \dots, X_{m_k}) = \tau_k(X_1, \dots, X_{m_k}, F_1^{m_1}, \dots, F_k^{m_k}).$

От предишната задача следва, че операторите $\Gamma_{\tau_1}, \dots, \Gamma_{\tau_k}$ са непрекъснати изображения на $\mathcal{F}$ съответно в $\mathcal{F}_{m_1}, \dots, \mathcal{F}_{m_k}$. Определяме изображението $\Gamma^* : \mathcal{F} \to \mathcal{F}$ посредством равенството $\Gamma^*(\overline{f}) = (\Gamma_{\tau_1}(\overline{f}), \dots, \Gamma_{\tau_k}(\overline{f}))$. От зад. 8, § 3.2, знаем, че изображението Γ^* също е непрекъснато. И от теоремата на Кнастер — Тарски (зад. 3, § 3.3) получаваме, че Γ^* притежава най-малка неподвижна точка $(\varphi_1, \dots, \varphi_k)$. С други думи, $(\varphi_1, \dots, \varphi_k)$ е най-малкото решение на системата
$$\Gamma_{\tau_i}(\mathbb{X}_1, \dots, \mathbb{X}_k) = \mathbb{X}_i, \quad i = 1, \dots, k.$$

Дефиниция. *Денотационна семантика* на програмата R с *предаване на параметрите по стойност* се нарича частичната функция $D_V(R)$ на n аргумента в D, определена чрез равенството
$$D_V(R)(a_1, \dots, a_n) \simeq \tau_0(a_1, \dots, a_n, \varphi_1, \dots, \varphi_k), \quad a_1, \dots, a_n \in \mathrm{D}.$$

Ще направим уговорката, че всяка рекурсивна програма, която разглеждаме, е в тип данни, съдържащ като основни всички операции, участващи в нейната дефиниция.

В следващите пет задачи рекурсивните програми са зададени в типа Nat.

Задача 2. Дадена е рекурсивната програма R:
$F(X, Y)$, where
$F(X, Y) = \texttt{if } X < Y \texttt{ then } 0 \texttt{ else } F(X - Y, Y) + 1.$

Да се докаже, че
$$D_V(R)(a, b) \simeq \begin{cases} [a/b], & \text{ако } b \neq 0, \\ \neg!, & \text{ако } b = 0, \end{cases}$$
за всяко $a, b \in \mathbb{N}$.

Упътване. Разгледайте оператора

$$\Gamma(f)(a,b) \simeq \texttt{if } a < b \texttt{ then } 0 \texttt{ else } f(a-b,b)+1.$$

Покажете, че функцията

$$f(a,b) \simeq \begin{cases} [a/b], & \text{ако } b \neq 0, \\ \neg!, & \text{ако } b = 0 \end{cases}$$

е най-малката неподвижна точка на Γ.

Задача 3. За рекурсивната програма R:

а) $F(X,Y)$, where
$F(X,Y) = \texttt{if } X < Y \texttt{ then } X \texttt{ else } F(X-Y,Y)$

да се докаже, че

$$D_V(R)(a,b) \simeq \begin{cases} \mathrm{rem}(b,a), & \text{ако } b \neq 0, \\ \neg!, & \text{ако } b = 0. \end{cases}$$

Тук $\mathrm{rem}(b,a)$ е остатъкът от делението на a с b.

б) $F(X,Y)$, where
$F(X,Y) = \texttt{if } X = Y \texttt{ then } 0 \texttt{ else } F(X,Y+1)+1$

да се докаже, че

$$D_V(R)(a,b) \simeq \begin{cases} a-b, & \text{ако } a \geq b, \\ \neg!, & \text{ако } a < b. \end{cases}$$

в) $F(X,Y)$, where
$F(X,Y) = \texttt{if } X = Y \texttt{ then } 1 \texttt{ else } F(X,Y+1) * (Y+1)$

да се докаже, че

$$D_V(R)(a,b) \simeq \begin{cases} a!/b!, & \text{ако } a \geqq b, \\ \neg!, & \text{ако } a < b. \end{cases}$$

г) $F(X,Y)$, where
$F(X,Y) = \texttt{if } X = Y \texttt{ then } X \texttt{ else if } X > Y \texttt{ then } F(X-Y,Y)$
$\texttt{else } F(X,Y-X)$

да се докаже, че

$$D_V(R)(a,b) \simeq \begin{cases} \text{НОД}(a,b), & \text{ако } (a=0\,\&\,b=0) \vee (a \neq 0\,\&\,b \neq 0), \\ \neg!, & \text{в останалите случаи.} \end{cases}$$

Задача 4. Нека R е следната рекурсивна програма:

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

Да се намери функцията $D_V(R)$.

Упътване. Разглеждаме оператора
$\Gamma(f)(a,b) \simeq \texttt{if } a = 0 \texttt{ then } 0 \texttt{ else } f(a-1, f(a,b))$.
С метода на Кнастер — Тарски намираме $f_0 = \varnothing^{(2)}$ и $f_{k+1} = \Gamma(f_k)$. За всяко $k > 0$ имаме

$$f_k(a,b) \simeq \begin{cases} 0, & \text{ако } a = 0, \\ \neg!, & \text{в противен случай.} \end{cases}$$

Знаем, че $f_\Gamma = \bigcup f_k$. Следователно $D_V(R) = f_k$ за $k > 0$.

Задача 5. Нека R е следната рекурсивна програма:

$F_1(X)$, where
$F_1(X) = F_2(F_1(X))$
$F_2(X) = 0$.

Да се намери функцията $D_V(R)$.

Отговор. $D_V(R) = \varnothing^{(1)}$.

Задача 6. Нека R е програмата

$F_1(X)$, where
$F_1(X) = \texttt{if } X = 0 \texttt{ then } 1 \texttt{ else } F_1(X-1) + F_2(X-1)$.
$F_2(X) = \texttt{if } X = 0 \texttt{ then } 0 \texttt{ else } F_2(X+1)$.

Да се намери функцията $D_V(R)$.

Упътване. Разгледайте системата

$$\left| \begin{array}{l} \Gamma_1(f^1,f^2)(a) \simeq \texttt{if } a = 0 \texttt{ then } 1 \texttt{ else } f^1(a-1) + f^2(a-1). \\ \Gamma_2(f^1,f^2)(a) \simeq \texttt{if } a = 0 \texttt{ then } 0 \texttt{ else } f^2(a+1). \end{array} \right.$$

Нека (φ_1, φ_2) е най-малкото решение на системата

$$\Gamma_i(\mathbb{X}_1, \mathbb{X}_2) = \mathbb{X}_i, \quad i = 1,2.$$

Ясно е, че $\varphi_2(0) = 0$ и $\neg!\varphi_2(a)$ за всяко $a > 0$. Ако положим $f_0^i = \varnothing^{(1)}$ и $f_{k+1}^i = \Gamma_i(f_k^1, f_k^2)$, $i = 1,2$, то покажете, че за всяко $k > 0$

$$f_k^1(a) \simeq \begin{cases} 1, & \text{ако } a \leq 1 \\ \neg!, & \text{в противен случай.} \end{cases}$$

Задача 7. Нека R е програмата в типа Int

$F(X,Y)$, where

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

Да се провери коя от следните функции в целите числа задава денотационната семантика на програмата R с предаване на параметрите по стойност:

а) $f_1(a,b) = \texttt{if } a = b \texttt{ then } b+1 \texttt{ else } a+1$;

б) $f_2(a,b) = \texttt{if } a \geqq b \texttt{ then } a+1 \texttt{ else } b-1$;

в) $f_3(a,b) \simeq \begin{cases} a+1, & \text{ако } (a \geqq b) \,\&\, ((a-b) \text{ е четно}), \\ \neg!, & \text{в противен случай.} \end{cases}$

В следващите задачи ще разгледаме някои приложения на метода на структурната индукция за доказване на свойства на рекурсивните програми.

Задача 8. Нека R е следната рекурсивна програма в типа List:

$F(X,\Lambda)$, where

$F(X,Y) = \texttt{if } X = \Lambda \texttt{ then } Y \texttt{ else } F(\mathrm{tail}(X), \mathrm{cons}(\mathrm{head}(X), Y))$.

Да означим с rev функцията $\lambda x.D_V(R)(x)$. Да се докаже, че

а) функцията rev е навсякъде определена в L^*;

б) $\forall a[\mathrm{rev}(\mathrm{rev}(a)) = a]$.

Доказателство. а) Да забележим първо, че множеството $\langle L^*, \prec\rangle$ е фундирано.

Нека f_Γ е най-малката неподвижна точка на оператора Γ, съответен на R, т.е.

$$\Gamma(f)(a,b) \simeq \texttt{if } a = \Lambda \texttt{ then } b \texttt{ else } f(\mathrm{tail}(a), \mathrm{cons}(\mathrm{head}(a), b)).$$

Със структурна индукция по a ще проверим, че $\forall b(!f_\Gamma(a,b))$.

За $a = \Lambda$ имаме $f_\Gamma(a,b) = b$ и следователно $!f_\Gamma(a,b)$.

Да допуснем, че $a \neq \Lambda$ и за всяко $c \prec a$ е изпълнено, че $\forall b(!f_\Gamma(c,b))$. Нека $b \in L^*$. Тогава по дефиницията на Γ имаме

$$f_\Gamma(a,b) \simeq f_\Gamma(\mathrm{tail}(a), \mathrm{cons}(\mathrm{head}(a), b)).$$

Тъй като $\mathrm{tail}(a) \prec a$, по индукционното предположение

$$!f_\Gamma(\mathrm{tail}(a), \mathrm{cons}(\mathrm{head}(a), b)).$$

Следователно $!f_\Gamma(a,b)$. Оттук $!f_\Gamma(a,\Lambda)$ за всяко $a \in L^*$ и в частност функцията rev е навсякъде определена.

б) Пак със структурна индукция по a ще покажем по-общото твърдение q, дефинирано с еквивалентността

$$q(a) \iff \forall b[\mathrm{rev}(f_\Gamma(a,b)) = f_\Gamma(b,a)].$$

За $a = \Lambda$ и произволно $b \in L^*$ имаме $\mathrm{rev}(f_\Gamma(\Lambda,b)) = \mathrm{rev}(b) = f_\Gamma(b,\Lambda)$ по дефиницията на rev.

Нека $a \neq \Lambda$ и за всяко $c \prec a$ е вярно $q(c)$. Тогава

$\mathrm{rev}(f_\Gamma(a,b))$

$= \mathrm{rev}(f_\Gamma(\mathrm{tail}(a), \mathrm{cons}(\mathrm{head}(a), b)))$ (по дефиницията на Γ и $a \neq \Lambda$)

$= f_\Gamma(\mathrm{cons}(\mathrm{head}(a), b), \mathrm{tail}(a))$ (по инд. предположение, $\mathrm{tail}(a) \prec a$)

$= f_\Gamma(b,a)$ (по деф. на Γ, $\mathrm{cons}(\mathrm{head}(a), b) \neq \Lambda$ и $\mathrm{cons}(\mathrm{head}(a), \mathrm{tail}(a)) = a$).

Следователно $\mathrm{rev}(f_\Gamma(a,b)) = f_\Gamma(b,a)$ за всяко $a, b \in L^*$. Тогава за $b = \Lambda$ получаваме $\mathrm{rev}(f_\Gamma(a,\Lambda)) = f_\Gamma(\Lambda,a) = a$ или $\mathrm{rev}(\mathrm{rev}(a)) = a$.

Задача 9. Нека R е следната рекурсивна програма в типа List:

$F(X,Y)$, where

$F(X,Y) = \texttt{if } X = \Lambda \texttt{ then } Y \texttt{ else } \mathrm{cons}(\mathrm{head}(X), F(\mathrm{tail}(X), Y))$.

Означаваме с conc функцията $\lambda x.D_V(R)(x)$. Да се докаже, че:

а) функцията conc е навсякъде определена в L^*;

б) за всеки две думи a и b думата $\mathrm{conc}(a,b)$ се състои от символите на a, последвани от символите на b.

Задача 10. Нека rev и conc са функциите, определени в зад. 8 и зад. 9. Да се докаже, че всеки път, когато a и b са символи от L и $c \in L^*$, е изпълнено:

а) $\mathrm{rev}(\mathrm{conc}(c,a)) = \mathrm{cons}(a, \mathrm{rev}(c))$;

б) $\mathrm{rev}(\mathrm{cons}(a,c)) = \mathrm{conc}(\mathrm{rev}(c), a)$;

в) $\mathrm{rev}(\mathrm{cons}(a, \mathrm{conc}(c,b))) = \mathrm{cons}(b, \mathrm{conc}(\mathrm{rev}(c), a))$.

Задача 11. Нека R е следната рекурсивна програма в типа List:

$F(X)$, where

$F(X) = \texttt{if } X = \Lambda \texttt{ then } \Lambda \texttt{ else if } \mathrm{tail}(X) = \Lambda \texttt{ then } X$

$\texttt{else } \mathrm{cons}(\mathrm{head}(F(\mathrm{tail}(X))), F(\mathrm{cons}(\mathrm{head}(X), F(\mathrm{tail}(F(\mathrm{tail}(X)))))))$.

Да се докаже, че $D_V(R) = \mathrm{rev}$.

Упътване. Използвайте, че за всяка дума c от L^* е изпълнено: или $c = \Lambda$, или $c \in L$, т.е. $c \neq \Lambda \,\&\, \mathrm{tail}(c) = \Lambda$, или $c = \mathrm{cons}(a, \mathrm{conc}(l, b))$ за някои $a, b \in L$ и $l \in L^*$. Използвайте още свойствата от зад. 10. Разгледайте строгата наредба $a \prec_1 b \iff |a| < |b|$, където $|a|$ е броят на символите на думата a.

Задача 12. Нека R_1 и R_2 са следните рекурсивни програми в типа Nat:

R_1: $F_1(X)$, where
$F_1(X) = \mathtt{if}\ X = 0\ \mathtt{then}\ 1\ \mathtt{else}\ F_1(X-1) * X.$

R_2: $F_2(X, 0)$, where
$F_2(X,Y) = \mathtt{if}\ X = Y\ \mathtt{then}\ 1\ \mathtt{else}\ F_2(X, Y+1) * (Y+1).$

Да се докаже, че $D_V(R_1)(a) \simeq D_V(R_2)(a)$ за всяко естествено число a.

Упътване. Използвайте зад. 25, § 2.3.

Задача 13. Нека R е следната рекурсивна програма в типа Nat:

$$F(X, 0, 1, 0), \quad \mathtt{where}$$
$$F(X, Y_1, Y_2, Y_3) = \mathtt{if}\ Y_2 + Y_3 > X\ \mathtt{then}\ Y_1\ \mathtt{else}\ F(X, Y_1+1, Y_2+2, Y_2+Y_3).$$

Да се докаже, че:

а) $D_V(R)$ е навсякъде определена функция в $\mathbb{N}$;

б) $\forall a (D_V(R)(a) = [\sqrt{a}])$.

Упътване. Разгледайте следната частична наредба в $\mathbb{N}^4$:

$$(a, b_1, b_2, b_3) \prec (a', b_1', b_2', b_3') \iff (a + 1 - b_2 - b_3) \leqq 0 < (a' + 1 - b_2' - b_3')$$
$$\text{или} \quad 0 < (a + 1 - b_2 - b_3) < (a' + 1 - b_2' - b_3').$$

Докажете, че $\langle \mathbb{N}^4, \prec \rangle$ е фундирано множество.

Ако f_Γ е най-малката неподвижна точка на оператора, съответен на програмата R, със структурна индукция по горната наредба покажете, че е в сила следното свойство:

$$\forall a \forall b_1 \forall b_2 \forall b_3 \Big[b_1^2 \leqq a \,\&\, b_2 = 2b_1 + 1 \,\&\, b_3 = b_1^2 \Longrightarrow$$
$$!f_\Gamma(a, b_1, b_2, b_3) \,\&\, (f_\Gamma(a, b_1, b_2, b_3)^2 \leqq a < (f_\Gamma(a, b_1, b_2, b_3) + 1)^2) \Big].$$

Задача 14. Нека R е програмата в типа Nat:

$$F(X, Y), \quad \mathtt{where}$$
$$F(X,Y) = \mathtt{if}\ X = Y\ \mathtt{then}\ 1\ \mathtt{else}\ (Y+1) * (Y+2) * F(X, Y+2).$$

Да се докаже, че ако $a \geqq b$ и $a - b$ е четно, то $D_V(a, b) = a!/b!$.

Задача 15. Нека R е програмата в типа Data:

$$F(\overline{X}), \quad \mathtt{where}$$
$$F(\overline{X}) = \mathtt{if}\ p(\overline{X})\ \mathtt{then}\ g(\overline{X})\ \mathtt{else}\ h(\overline{X}, F(u_1(\overline{X}), \ldots, u_n(\overline{X}))),$$

където $\overline{X}$ е $X_1, \ldots, X_n$, p е тотална операция от тип $\mathrm{D}^n \to \mathrm{B}$, а g и u_i са тотални операции от тип $\mathrm{D}^n \to \mathrm{D}$ и h е тотална операция от тип $\mathrm{D}^{n+1} \to \mathrm{D}$.

Нека q е свойство за елементите на D^n и

$$W = \{d \in \mathrm{D}^n \mid q(d) = tt\} \neq \varnothing.$$

Да предположим, че съществува частична наредба $\prec$ в W, за която:

(1) множеството $\langle W, \prec \rangle$ е фундирано;

(2) ако $d \in W$ е минимален елемент в W, то $p(d) = tt$;

(3) за всяко $d \in W$, ако $p(d) = ff$, то $(u_1(d), \ldots, u_n(d)) \in W$ и $(u_1(d), \ldots, u_n(d)) \prec d$.

Да се докаже, че D_V е дефинирана за всяко $d \in \mathrm{D}^n$, за което $q(d) = tt$.

Задача 16. Да се докаже, че всяка частично рекурсивна функция може да се пресметне с помощта на рекурсивна програма с предаване на параметрите по стойност в следните типове данни:

а) $\langle \mathbb{N};\ \lambda x.x+1; = \rangle$;

б) $\langle \mathbb{N};\ \lambda x.x+1, \lambda x.x \dot{-} 1; x = 0? \rangle$.

Упътване. Използвайте индукция по построението на частично рекурсивните функции. Намерете рекурсивни програми R_0, R_1, R_k^n в посочения тип данни, за които $D_V(R_0) = \boldsymbol{O}$, $D_V(R_1) = \boldsymbol{S}$ и $D_V(R_k^n) = \boldsymbol{I}_k^n$. Покажете още, че операциите суперпозиция, примитивна рекурсия и минимизация запазват свойството, т.е. ако за $g, f_1, \ldots, f_k$ има рекурсивни програми $R, R_1, \ldots, R_k$, за които $D_V(R) = g$, $D_V(R_i) = f_i$, то и за функцията $g(f_1, \ldots, f_k)$ има рекурсивна програма R^*, която я пресмята с предаване на параметрите по стойност и т.н.

Задача 17*. Да се докаже, че в типа данни $\langle \mathbb{N}; \lambda x.x \dot{-} 1; x = 0? \rangle$ не всяка частично рекурсивна функция може да се пресметне с помощта на рекурсивна програма с предаване на параметрите по стойност.

Упътване. Покажете, че за функцията $\lambda x.x+1$ не съществува такава рекурсивна програма.

Задача 18. Да се докаже, че ако типът данни Data е в естествените числа и има изчислими основни операции, то за всяка рекурсивна програма R в типа данни Data функцията $D_V(R)$ е изчислима.

§ 4.2. Денотационна семантика на рекурсивните програми с предаване на параметрите по име

Нека $(\mathrm{D}_\perp, \sqsubseteq, \perp)$ е плоската област на Скот, базирана на множеството D, т.е. $\mathrm{D}_\perp = \mathrm{D} \cup \{\perp\}$, релацията „$\sqsubseteq$“ е плоската наредба, определена като $x \sqsubseteq y \iff x = \perp \vee x = y$. Ще използваме буквите x, y, евентуално с индекси, за да означаваме елементи на $\mathrm{D}_\perp$. С $a, b, \ldots$, както в предишния параграф, ще означаваме елементи на D.

Нека $\mathrm{D}^n_\perp = \mathrm{D}_\perp \times \dots \times \mathrm{D}_\perp$ и $\mathrm{B}^n_\perp = \mathrm{B}_\perp \times \dots \times \mathrm{B}_\perp$, $n \geq 1$. Съгласно зад. 7, § 3.1, наредените тройки $(\mathrm{D}^n_\perp, \sqsubseteq, \perp^n)$ и $(\mathrm{B}^n_\perp, \sqsubseteq, \perp^n)$, където $\sqsubseteq$ е покомпонентната плоска наредба, са области на Скот.

Да означим с $\mathcal{F}^\perp_n$ съвкупността от непрекъснатите изображения на $\mathrm{D}^n_\perp$ в $\mathrm{D}_\perp$, с $\mathcal{P}^\perp_n$ — съвкупността от непрекъснатите изображения на $\mathrm{D}^n_\perp$ в $\mathrm{B}_\perp$, а с $\mathcal{B}^\perp_n$ — съвкупността от непрекъснатите изображения на $\mathrm{B}^n_\perp$ в $\mathrm{B}_\perp$. От зад. 4, § 3.2, знаем, че $(\mathcal{F}^\perp_n, \sqsubseteq, \Omega)$, $(\mathcal{P}^\perp_n, \sqsubseteq, \Omega)$ и $(\mathcal{B}^\perp_n, \sqsubseteq, \Omega)$ са области на Скот, където ако φ и ψ са елементи на $\mathcal{F}^\perp_n(\mathcal{P}^\perp_n, \mathcal{B}^\perp_n)$, то

$$\varphi \sqsubseteq \psi \iff \forall \overline{x}(\varphi(\overline{x}) \sqsubseteq \psi(\overline{x}))$$

и $\Omega(\overline{x}) = \perp$ за всяко $\overline{x} \in \mathrm{D}^n_\perp(\mathrm{B}^n_\perp)$.

Тъй като областта $(\mathrm{D}^n_\perp, \sqsubseteq, \perp^n)$ не съдържа безкрайни монотонно растящи редици, то от зад. 5, § 3.2, имаме, че $\mathcal{F}^\perp_n$ съвпада с множеството на монотонните изображения на $\mathrm{D}^n_\perp$ в $\mathrm{D}_\perp$. Аналогично за $\mathcal{P}^\perp_n$ и $\mathcal{B}^\perp_n$.

Ще припомним още две понятия от § 3.1.

Нека $\varphi : \mathrm{D}^n_\perp \to \mathrm{D}_\perp$ ($\varphi : \mathrm{D}^n_\perp \to \mathrm{B}_\perp$, $\varphi : B^n_\perp \to \mathrm{B}_\perp$). Казваме, че функцията φ е *точна*, ако $\varphi(x_1, \dots, x_n) = \perp$ всеки път, когато $\perp \in \{x_1, \dots, x_n\}$.

Всяка точна функция е монотонна, а следователно и непрекъсната.

Нека $\varphi : \mathrm{D}^n \rightharpoonup \mathrm{D}$ ($\varphi : \mathrm{D}^n \rightharpoonup \mathrm{B}$, $\varphi : \mathrm{B}^n \rightharpoonup \mathrm{B}$). *Точно продължение* на φ ще наричаме функцията $\varphi^* : \mathrm{D}^n_\perp \to \mathrm{D}_\perp$ ($\varphi^* : \mathrm{D}^n_\perp \to \mathrm{B}_\perp$, $\varphi^* : B^n_\perp \to \mathrm{B}_\perp$), за която

$$\varphi^*(x_1,\dots,x_n) = \begin{cases} b, & \text{ако } \perp \notin \{x_1,\dots,x_n\} \mathbin{\&} \varphi(x_1,\dots,x_n) \simeq b,\\ \perp, & \text{в останалите случаи.}\end{cases}$$

Всяко точно продължение е точна и следователно непрекъсната функция.

Ще използваме понятието точно продължение на функция, за да представим основните операции на езика на рекурсивните програми. Всяка основна операция от тип $\mathrm{D}^n \to \mathrm{D}$, $\mathrm{D}^n \to \mathrm{B}$, $\mathrm{B}^n \to \mathrm{B}$ се представя с нейното точно продължение като елемент съответно на $\mathcal{F}^\perp_n$, $\mathcal{P}^\perp_n$, $\mathcal{B}^\perp_n$.

Нека $\tau(X_1,\dots,X_n,F_1^{m_1},\dots,F_k^{m_k})$ е терм, $x_1,\dots,x_n$ са елементи на $\mathrm{D}_\perp$, а $\varphi^1,\dots,\varphi^k$ принадлежат съответно на $\mathcal{F}^\perp_{m_1},\dots,\mathcal{F}^\perp_{m_k}$.

Стойността $\tau(x_1,\dots,x_n,\varphi^1,\dots,\varphi^k)$ на терма τ се определя със следната индуктивна дефиниция:

а) ако τ е константата c, то $\tau(\overline{x},\overline{\varphi}) = c$;

б) ако $\tau = X_i$, то $\tau(\overline{x},\overline{\varphi}) = x_i$;

в) ако $\tau = f(\tau_1,\dots,\tau_l)$ за някоя l-местна основна операция f, то

$$\tau(\overline{x},\overline{\varphi}) = f^*(\tau_1(\overline{x},\overline{\varphi}),\dots,\tau_l(\overline{x},\overline{\varphi})),$$

където f^* е точното продължение на f;

г) ако $\tau =$ if p then τ_1 else τ_2, то

$$\tau(\overline{x},\overline{\varphi}) = \begin{cases} \tau_1(\overline{x},\overline{\varphi}), & \text{ако } p(\overline{x},\overline{\varphi}) = tt,\\ \tau_2(\overline{x},\overline{\varphi}), & \text{ако } p(\overline{x},\overline{\varphi}) = ff,\\ \perp, & \text{ако } p(\overline{x},\overline{\varphi}) = \perp;\end{cases}$$

д) ако $\tau = F_i^{m_i}(\tau_1,\dots,\tau_{m_i})$, то

$$\tau(\overline{x},\overline{\varphi}) = \varphi^i(\tau_1(\overline{x},\overline{\varphi}),\dots,\tau_{m_i}(\overline{x},\overline{\varphi})).$$

За разлика от дефиницията за стойност на терм от предишния параграф, тук всеки терм има стойност, принадлежаща на $\mathrm{D}_\perp$ или на $\mathrm{B}_\perp$ в зависимост от типа на терма.

Нека $\tau(X_1,\dots,X_n,F_1^{m_1},\dots,F_k^{m_k})$ е терм. Да означим с $\mathcal{F}^\perp$ декартовото произведение $\mathcal{F}^\perp_{m_1}\times\ldots\times\mathcal{F}^\perp_{m_k}$. Ще използваме $\overline{\varphi}$, $\overline{\psi}$ за означаване на елементи на $\mathcal{F}^\perp$.

С терма τ свързваме изображението Γ_τ на $\mathcal{F}^\perp$ в множеството на тоталните функции на n аргумента в $\mathrm{D}_\perp$, дефинирано чрез

(1) $$\Gamma_\tau(\varphi^1,\dots,\varphi^k)(x_1,\dots,x_n) = \tau(x_1,\dots,x_n,\varphi^1,\dots,\varphi^k).$$

В задачи 2–5 ще покажем, че Γ_τ е непрекъснато изображение на $\mathcal{F}^\perp$ в $\mathcal{F}^\perp_n$.

Вече сме готови да определим денотационната семантика на рекурсивните програми с предаване на параметрите по име.

Нека R е рекурсивна програма в типа Data от вида

$\tau_0(X_1,\ldots,X_n,F_1^{m_1},\ldots,F_k^{m_k})$, where

$F_i^{m_i}(X_1,\ldots,X_{m_i}) = \tau_i(X_1,\ldots,X_{m_i},F_1^{m_1},\ldots,F_k^{m_k}), \quad i=1,\ldots,k.$

Определяме изображението $\Gamma^* : \mathcal{F}^\perp \to \mathcal{F}^\perp$ с равенството

$$\Gamma^*(\overline{\psi}) = (\Gamma_{\tau_1}(\overline{\psi}),\ldots,\Gamma_{\tau_k}(\overline{\psi})).$$

Тъй като $\Gamma_{\tau_1},\ldots,\Gamma_{\tau_k}$ са непрекъснати, то и Γ^* е непрекъснато изображение (зад. 8, § 3.2). По теоремата на Кнастер — Тарски изображението Γ^* притежава най-малка неподвижна точка $(\varphi^1,\ldots,\varphi^k)$. С други думи, $(\varphi^1,\ldots,\varphi^k)$ е най-малкото решение на системата

$$\Gamma_{\tau_i}(\mathbb{X}_1,\ldots,\mathbb{X}_k) = \mathbb{X}_i, \quad i=1,\ldots,k.$$

Нека

$$\varphi = \lambda x_1,\ldots,x_n.\tau_0(x_1,\ldots,x_n,\varphi^1,\ldots,\varphi^k).$$

С помощта на φ дефинираме n-местната частична функция $D_N(R)$ в множеството D като

$$D_N(R)(a_1,\ldots,a_n) \simeq \begin{cases} \varphi(a_1,\ldots,a_n), & \text{ако } \varphi(a_1,\ldots,a_n) \neq \perp, \\ \neg!, & \text{ако } \varphi(a_1,\ldots,a_n) = \perp. \end{cases}$$

Дефиниция. Функцията $D_N(R)$ се нарича *денотационна семантика на програмата R с предаване на параметрите по име.*

Ще казваме, че функциите $\varphi^1,\ldots,\varphi^k$ задават денотационната семантика на функционалните променливи $F_1^{m_1},\ldots,F_k^{m_k}$ на R с предаване на параметрите по име.

Теорема. *За всяка рекурсивна програма R е в сила $D_V(R) \subseteq D_N(R)$.*

Задача 1. Нека $\varphi_0 \sqsubseteq \varphi_1 \sqsubseteq \ldots \sqsubseteq \varphi_r \sqsubseteq \ldots$ е монотонно растяща редица от елементи на $\mathcal{F}_n^\perp$ и $\varphi = \bigcup \varphi_r$ е точната ѝ горна граница. Тогава при всеки избор на $\overline{x} \in \mathrm{D}_\perp^n$ са в сила:

а) $\varphi(\overline{x}) = \perp \iff \forall r(\varphi_r(\overline{x}) = \perp)$;

б) ако $b \neq \perp$, то $\varphi(\overline{x}) = b \iff \exists r(\varphi_r(\overline{x}) = b)$.

Упътване. Разгледайте поотделно случаите

$$\forall r(\varphi_r(\overline{x}) = \perp) \text{ и } \exists r(\varphi_r(\overline{x}) \neq \perp).$$

Нека $\tau(X_1,\ldots,X_n,F_1^{m_1},\ldots,F_k^{m_k})$ е терм и Γ_τ е изображението, свързано с него чрез равенството (1).

Задача 2. Да се докаже, че всеки път, когато $\overline{\varphi} \in \mathcal{F}^\perp$, функцията $\lambda x_1,\ldots,x_n.\Gamma_\tau(\overline{\varphi})$ е монотонна.

Забележка. Като следствие се получава $\Gamma_\tau(\overline{\varphi}) \in \mathcal{F}_n^\perp$, т.е. $\Gamma_\tau : \mathcal{F}^\perp \to \mathcal{F}_n^\perp$.

Задача 3. Да се докаже, че $\Gamma_\tau : \mathcal{F}^\perp \to \mathcal{F}_n^\perp$ е монотонно изображение.

Упътване. Използвайте индукция по построението на τ.

Задача 4. Нека $\overline{\varphi}_0 \sqsubseteq \overline{\varphi}_1 \sqsubseteq \ldots \sqsubseteq \overline{\varphi}_r \sqsubseteq \ldots$ е монотонно растяща редица от елементи на $\mathcal{F}^\perp$, $\overline{x} \in \mathrm{D}_\perp^n$, $a \in \mathrm{D}_\perp$ $(a \in \mathrm{B}_\perp)$ и $a \neq \perp$. Нека $\overline{\varphi} = \bigcup \overline{\varphi}_r$. Да се докаже, че

$$\Gamma_\tau(\overline{\varphi})(\overline{x}) = a \iff \exists r(\Gamma_\tau(\overline{\varphi}_r)(\overline{x}) = a).$$

Упътване. Използвайте индукция по построението на τ, като имате предвид, че основните операции и функциите $\overline{\varphi}$, $\overline{\varphi}_r$, $r = 0,1,\ldots$ са монотонни. Използвайте още твърденията от зад. 1 и зад. 3.

Задача 5. Да се докаже, че Γ_τ е непрекъснато изображение на $\mathcal{F}^\perp$ в $\mathcal{F}_n^\perp$.

Упътване. Използвайте предишната задача.

Задача 6. Нека $\varphi_0 \sqsubseteq \varphi_1 \sqsubseteq \cdots \sqsubseteq \varphi_r \sqsubseteq \cdots$ е монотонно растяща редица от елементи на $\mathcal{F}_n^\perp$. Да означим с φ точната горна граница на редицата $\{\varphi_r\}$. Да се докаже, че φ е точна тогава и само тогава, когато всяка от функциите φ_r, $r = 0,1,\ldots$, е точна.

Упътване. Използвайте зад. 1.

Дефинираме частична функция $o : \mathrm{D}_\perp \to\!\!\!\circ\; \mathrm{D}$ с условното равенство:

$$o(x) \simeq \begin{cases} x, & \text{ако } x \neq \perp, \\ \neg!, & \text{ако } x = \perp. \end{cases}$$

За всяко $x \in \mathrm{D}_\perp$ с x^0 ще означаваме $o(x)$.

Ако $\varphi \in \mathcal{F}_n^\perp$, то с φ° ще означаваме n-местната частична функция в D, дефинирана посредством

$$\varphi^\circ(\overline{x}) \simeq (\varphi(\overline{x}))^\circ.$$

Задача 7. Нека $\tau(X_1,\ldots,X_n,F_1^{m_1},\ldots,F_k^{m_k})$ е терм и $\varphi_1,\ldots,\varphi_k$ са функции съответно от $\mathcal{F}_{m_1}^\perp,\ldots,\mathcal{F}_{m_k}^\perp$. Нека

$$\chi = \lambda x_1,\ldots,x_n.\tau_N(x_1,\ldots,x_n,\varphi_1,\ldots,\varphi_k),$$

$$\psi = \lambda a_1,\ldots,a_n.\tau_V(a_1,\ldots,a_n,\varphi_1^\circ,\ldots,\varphi_k^\circ).$$

Да се докаже, че $\psi \subseteq \chi^\circ$. Да се даде пример за терм τ и функции $\varphi_1, \ldots, \varphi_k$, за които $\psi \neq \chi^\circ$.

Забележка. Тук стойността на τ в дефиницията на χ е съгласно определението от този параграф. Затова е означена с τ_N. В дефиницията на ψ стойността на τ е съгласно определението от § 4.1. Означена е с τ_V.

Упътване. Използвайте индукция по построението на терма τ.

Например, нека $\tau = F_i^{m_i}(\tau^1, \ldots, \tau^{m_i})$ и за $\tau^1, \ldots, \tau^{m_i}$ е вярно, че

$$\lambda\overline{a}.\tau_V^j(\overline{a}, \varphi_1^\circ, \ldots, \varphi_k^\circ) \subseteq (\lambda\overline{a}.\tau_N^j(\overline{a}, \overline{\varphi}))^\circ,$$

за $j = 1, \ldots, m_i$. Тогава

$$\chi(\overline{x}) = \varphi_i(\tau_N^1(\overline{x}, \overline{\varphi}), \ldots, \tau_N^{m_i}(\overline{x}, \overline{\varphi})),$$

$$\psi(\overline{a}) \simeq \varphi_i^\circ(\tau_V^1(\overline{a}, (\overline{\varphi})^\circ), \ldots, \tau_V^{m_i}(\overline{a}, (\overline{\varphi})^\circ)).$$

Нека $\psi(\overline{a}) \simeq b$. Ще покажем, че $\chi^\circ(\overline{a}) \simeq b$. Имаме

$$\varphi_i^\circ(\tau_V^1(\overline{a}, (\overline{\varphi})^\circ), \ldots, \tau_V^{m_i}(\overline{a}, (\overline{\varphi})^\circ)) \simeq b.$$

Следователно съществуват $b_1, \ldots, b_{m_i} \in \mathrm{D}$, за които $\tau_V^j(\overline{a}, (\overline{\varphi})^\circ) = b_j$ за $1 \leqq j \leqq m_i$, и $\varphi_i^\circ(b_1, \ldots, b_{m_i}) = b$. Съгласно индукционното предположение $(\tau_N^j(\overline{a}, \overline{\varphi}))^\circ = b_j$. Тъй като $b_j \neq \perp, b \neq \perp$, то $\tau_N^j(\overline{a}, \overline{\varphi}) = b_j$ и $\varphi_i(b_1, \ldots, b_{m_i}) = b$. Оттук $\chi(\overline{a}) = b$ и следователно $\chi^\circ(\overline{a}) \simeq b$.

Задача 8. При предположенията от предишната задача да се покаже, че ако $\varphi_1, \ldots, \varphi_k$ са точни функции, то $\chi^\circ = \psi$.

Упътване. От зад. 7 следва, че достатъчно е да се покаже, че $\chi^\circ \subseteq \psi$. Използвайте индукция по построението на терма τ.

Например, нека $\tau = F_i^{m_i}(\tau^1, \ldots, \tau^{m_i})$ и за $\tau^1, \ldots, \tau^{m_i}$ е вярно, че

$$(\lambda\overline{a}.\tau_N^j(\overline{a}, \overline{\varphi}))^\circ \subseteq \lambda\overline{a}.\tau_V^j(\overline{a}, \varphi_1^\circ, \ldots, \varphi_k^\circ),$$

за $j = 1, \ldots, m_i$.

Нека $\chi^\circ(\overline{a}) \simeq b$. С други думи, $\varphi_i(\tau_N^1(\overline{a}, \overline{\varphi}), \ldots, \tau_N^{m_i}(\overline{a}, \overline{\varphi})) = b$. Нека $\tau_N^j(\overline{a}, \overline{\varphi}) = b_j$ за $1 \leqq j \leqq m_i$. Имаме $\varphi_i(b_1, \ldots, b_{m_i}) = b$ и тъй като φ_i е точна, $b_1, \ldots, b_{m_i} \in \mathrm{D}$. Ясно е, че $(\tau_N^j(\overline{a}, \overline{\varphi}))^\circ = b_j$. Съгласно индукционното предположение $\tau_V^j(\overline{a}, (\overline{\varphi})^\circ) = b_j$, $j = 1, \ldots, m_i$. Тогава $\varphi_i^\circ(\tau_V^1(\overline{a}, (\overline{\varphi})^\circ), \ldots, \tau_V^{m_i}(\overline{a}, (\overline{\varphi})^\circ)) \simeq b$, откъдето $\psi(\overline{a}) \simeq b$.

Задача 9. Да разгледаме рекурсивната програма R:

$\sigma(X_1, \ldots, X_n, F_1^{m_1}, \ldots, F_k^{m_k})$, where
$F_i^{m_i}(X_1, \ldots, X_{m_i}) = \tau_i(X_1, \ldots, X_{m_i}, F_1^{m_1}, \ldots, F_k^{m_k}), i = 1, \ldots, k.$

Да се докаже, че ако функциите $\varphi_1, \ldots, \varphi_k$, които задават семантиката на функционалните променливи на R с предаване на параметрите по име, са точни, то $D_V(R) = D_N(R)$.

Упътване. По определение

$$D_N(R) = \lambda a_1, \ldots, a_n.(\sigma_N(a_1, \ldots, a_n, \varphi_1, \ldots, \varphi_k))^\circ,$$

а от зад. 8 знаем, че

$$D_N(R) = \lambda a_1, \ldots, a_n.\sigma_V(a_1, \ldots, a_n, \varphi_1^\circ, \ldots, \varphi_k^\circ).$$

Следователно проблемът се свежда до доказване на равенството

$$D_V(R) = \lambda a_1, \ldots, a_n.\sigma_V(a_1, \ldots, a_n, \varphi_1^\circ, \ldots, \varphi_k^\circ).$$

За да не претрупваме означенията, да предположим, че $k = 1$, т.е. R е програма от вида

$\sigma(X_1, \ldots, X_n, F)$, where
$F(X_1, \ldots, X_m) = \tau(X_1, \ldots, X_m, F)$.

Нека φ е функцията, която задава семантиката на функционалната променлива на R с предаване на параметрите по име. Да предположим, че φ е точна. Ще докажем, че $D_V(R) = \lambda\overline{a}.\sigma_V(\overline{a}, \varphi^\circ)$.

Нека $\psi_0 = \varnothing$, $\psi_{i+1} = \lambda\overline{a}.\tau_V(\overline{a}, \psi_i)$ и $\psi = \bigcup \psi_i$. Тъй като по дефиниция $D_V(R) = \lambda\overline{a}.\sigma_V(\overline{a}, \psi)$, достатъчно е да покажем, че $\varphi^\circ = \psi$.

Използваме, че $\varphi = \bigcup \chi_i$, където $\chi_0 = \Omega$, $\chi_{i+1} = \lambda\overline{a}.\tau_N(\overline{a}, \chi_i)$.

Да забележим, че съгласно зад. 6 щом φ е точна, то за всяко i функцията χ_i също е точна.

Като имате предвид този факт, с индукция по i покажете, че $\chi_i^\circ = \psi_i$. Тогава $\varphi^\circ = \psi$. Наистина, ако $\varphi^\circ(\overline{a}) \simeq b$, то $\varphi(\overline{a}) = b$. Оттук съгласно зад. 1 съществува i, за което $\chi_i(\overline{a}) = b$, а следователно $\psi_i(\overline{a}) \simeq b$. Така $\psi(\overline{a}) \simeq b$. Аналогично се вижда, че ако $\psi(\overline{a}) \simeq b$, то $\varphi^\circ(\overline{a}) \simeq b$.

Задача 10. Нека R е рекурсивна програма и $(\varphi_1, \ldots, \varphi_k) \in \mathcal{F}^\perp$, $(\psi_1, \ldots, \psi_k) \in \mathcal{F}$ задават семантиката на функционалните променливи на R с предаване на параметрите съответно по име и по стойност. Да се докаже, че $\psi_i \sqsubseteq \varphi_i^\circ$ за $i = 1, \ldots, k$.

Задача 11. За дадената рекурсивна програма R в Nat да се провери, че $D_V(R) = D_N(R)$.

$F_1(X)$, where
$F_1(X) =$ if $X = 0$ then 1 else $F_1(X-1) + F_2(X-1)$
$F_2(X) =$ if $X = 0$ then 0 else $F_2(X+1)$.

Доказателство. От зад. 6, § 4.1, знаем, че

$$D_V(R)(a) \simeq \begin{cases} 1, & \text{ако } a \leqq 1, \\ \neg!, & \text{ако } a > 1. \end{cases}$$

За да намерим $D_N(R)$, разглеждаме системата от оператори

$$\left|\begin{array}{l}\Delta_1(f,g)(x) \simeq \texttt{if } x =^* 0 \texttt{ then } 1 \texttt{ else } f(x -^* 1) +^* g(x -^* 1),\\ \Delta_2(f,g)(x) \simeq \texttt{if } x =^* 0 \texttt{ then } 0 \texttt{ else } g(x +^* 1).\end{array}\right.$$

Нека $\Omega = \lambda x.\bot$. Разглеждаме редицата $\{(f_k, g_k)\}$, дефинирана с $f_0 = \Omega$, $g_0 = \Omega$, $f_{k+1} = \Delta_1(f_k, g_k)$, $g_{k+1} = \Delta_2(f_k, g_k)$. Лесно се вижда, че за $k > 0$ е в сила равенството

$$g_k(x) = \begin{cases} 0, & \text{ако } x = 0,\\ \bot, & \text{в противен случай.}\end{cases}$$

Проверяваме, че

$$f_1(x) = \begin{cases} 1, & \text{ако } x = 0,\\ \bot, & \text{за } x > 0 \text{ или } x = \bot;\end{cases}$$

$$f_2(x) = \begin{cases} \bot, & \text{ако } x = \bot,\\ 1, & \text{ако } x = 0,\\ f_1(x -^* 1) +^* g_1(x -^* 1), & \text{ако } x > 0\end{cases} = \begin{cases} \bot, & \text{ако } x = \bot,\\ 1, & \text{ако } x = 0,\\ 1, & \text{ако } x = 1,\\ \bot, & \text{за } x > 1,\end{cases}$$

тъй като $+^*$ е точна. Аналогично се съобразява, че за $k > 1$

$$f_k(x) = \begin{cases} 1, & \text{ако } x \leqq 1,\\ \bot, & \text{за } x > 1 \text{ или } x = \bot.\end{cases}$$

Откъдето $D_N(R)(a) \simeq D_V(R)(a)$ за всяко $a \in \mathbb{N}$.

Задача 12. Нека R е унарна рекурсивна програма от вида

$F_1(X)$, where
$F_1(X) = \tau_1(X, \boldsymbol{F})$
....
$F_k(X) = \tau_k(X, \boldsymbol{F})$.

Да предположим, че никоя от функциите $\varphi_1, \dots, \varphi_k$, които задават семантиката на функционалните променливи на R с предаване на параметрите по име, не е константна. Да се докаже, че $D_N(R) = D_V(R)$.

Упътване. Съобразете, че всяка φ_i е точна и приложете зад. 9.

Рекурсивните програми от следващите задачи са в типа Nat.

Задача 13. Нека R е следната рекурсивна програма:

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

Да се определи функцията $D_N(R)$ и да се покаже, че е изпълнено $D_V(R) \neq D_N(R)$.

Отговор. $D_V(R)(a,b) \simeq \begin{cases} 0, & \text{ако } a = 0\\ \neg!, & \text{ако } a > 0,\end{cases}$ $\quad D_N(R)(a,b) = 0.$

Задача 14. Нека R е следната рекурсивна програма:

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

Да се покаже, че

$$D_V(R)(a,b) \simeq \begin{cases} 0, & \text{ако } a = 0,\\ \neg!, & \text{в противен случай,}\end{cases}$$

$D_N(R)(a,b) = a * (a+1)/2.$

Доказателство. Нека $\Gamma : \mathcal{F}_2 \to \mathcal{F}_2$ е дефиниран с

$$\Gamma(f)(a,b) \simeq \texttt{if } a = 0 \texttt{ then } 0 \texttt{ else } f(a-1, f(a,b)) + a.$$

Знаем, че $D_V(R) = f_\Gamma$.

Нека $f_k = \Gamma^k(\varnothing^{(2)})$. Тогава за всяко $k > 0$ имаме

$$f_k(a,b) \simeq \begin{cases} 0, & \text{ако } a = 0,\\ \neg!, & \text{ако } a > 0.\end{cases}$$

Следователно $D_V(R) = f_k$ за $k > 0$.

За да получим $D_N(R)$, вземаме изображението $\Delta : \mathcal{F}_2^\bot \to \mathcal{F}_2^\bot$, където

$$\Delta(g)(x,y) = \texttt{if } x =^* 0 \texttt{ then } 0 \texttt{ else } g(x -^* 1, g(x,y)) +^* x.$$

Нека $\Omega = \lambda x,y.\bot$. Разглеждаме редицата $g_0 = \Omega$, $g_{k+1} = \Delta(g_k)$. Нека $x, y \in \mathbb{N}_\bot$. Имаме

$$g_1(x,y) = \begin{cases} \bot, & \text{ако } x = \bot,\\ 0, & \text{ако } x = 0,\\ \bot, & \text{за } x > 0;\end{cases} \qquad g_2(x,y) = \begin{cases} \bot, & \text{ако } x = \bot,\\ 0, & \text{ако } x = 0,\\ 1, & \text{ако } x = 1,\\ \bot, & \text{за } x > 1;\end{cases}$$

$$g_k(x,y) = \begin{cases} \perp, & \text{ако } x = \perp,\\ 0, & \text{ако } x = 0,\\ 1, & \text{ако } x = 1,\\ \dots & \\ 1+2+\dots+k-1, & \text{ако } x = k-1,\\ \perp, & \text{за } x \geqq k \end{cases}$$

$$= \begin{cases} \perp, & \text{ако } x = \perp,\\ 1+2+\dots+x, & \text{ако } x < k,\\ \perp, & \text{за } x \geqq k. \end{cases}$$

Оттук получаваме

$$D_N(R)(a,b) = a*(a+1)/2.$$

Задача 15. За всяка от изброените рекурсивни програми R_i да се покаже, че $D_V(R_i) \neq D_N(R_i)$.

R_1: $F_1(X)$, where
$F_1(X) = F_2(F_1(X))$
$F_2(X) = 0$.

R_2: $F(X,Y)$, where
$F(X,Y) = \texttt{if } X = 0 \texttt{ then } 0$
$\texttt{else if } X \equiv 1 \pmod 2 \texttt{ then } F(X-1, F(X+1,Y))$
$\texttt{else } F(X,Y)$.

R_3: $F(X,1)$, where
$F(X,Y) = \texttt{if } X = 0 \texttt{ then } 1$
$\texttt{else } X * F(X-1, G(X-1,Y))$
$G(X,Y) = \texttt{if } X = 0 \texttt{ then } 1 \texttt{ else } G(X,Y)$.

R_4: $F(X,Y)$, where
$F(X,Y) = \texttt{if } X = 0 \texttt{ then } 0 \texttt{ else } F(X-1, G(Y))$
$G(Y) \quad = \texttt{if } Y = 0 \texttt{ then } 0 \texttt{ else } G(Y+1)$.

R_5: $F(X,X)$, where
$F(X,Y) = \texttt{if } X \equiv 0 \pmod 3 \texttt{ then } X/3$
$\texttt{else } F(X+1, F(2X+2,Y))$.

Упътване.

$$D_V(R_2)(a,b) \simeq \begin{cases} 0, & \text{ако } a = 0,\\ \neg!, & \text{ако } a > 0. \end{cases}$$

$$D_N(R_2)(a,b) \simeq \begin{cases} 0, & \text{ако } a = 0 \vee a = 1,\\ \neg!, & \text{ако } a > 1. \end{cases}$$

$$D_V(R_3)(a) \simeq \begin{cases} 1, & \text{ако } a \leq 1,\\ \neg!, & \text{ако } a > 1. \end{cases} \qquad D_N(R_3)(a) = a!.$$

Дефинираме $x \dot{-} y$ (*отсечена разлика*) по следния начин:

$$x \dot{-} y = \begin{cases} x - y, & \text{ако } x \geqq y,\\ 0, & \text{ако } x < y. \end{cases}$$

Задача 16. Да се докаже, че $D_V(R) \neq D_N(R)$, където R е програмата:

а) $F(X,Y)$, where
$F(X,Y) = \texttt{if } X = 0 \texttt{ then } 1$
$\texttt{else } X.F(X \dot{-} Y, F(X,Y))$.

б) $F(X,Y)$, where
$F(X,Y) = \texttt{if } X \leqq 10 \texttt{ then } 10$
$\texttt{else } F(X \dot{-} Y, F(X,Y) + 10)$.

Задача 17. Като се използва теоремата на Кнастер — Тарски, да се пресметнат $D_V(R)$ и $D_N(R)$, където R е програмата:

а) $F(X,Y)$, where
$F(X,Y) = \texttt{if } X \equiv 0 \pmod 2 \texttt{ then } X/2$
$\texttt{else } F(X.Y, F(X, 2Y))$.

б) $F(X,Y)$, where
$F(X,Y) = \texttt{if } X \equiv 0 \pmod 2 \texttt{ then } X/2$
$\texttt{else } F(X+Y, F(X,Y))$.

в)* $F(X,Y)$, where
$F(X,Y) = \texttt{if } X \equiv 0 \pmod 2 \texttt{ then } X/2$
$\texttt{else } F(X+Y, F(X, Y+1))$.

Задача 18. Като се използва теоремата на Кнастер — Тарски, да се пресметнат $D_V(R)$ и $D_N(R)$ за програмата R:

$F(X,X)$, where
$F(X,Y) = \texttt{if } X = 0 \texttt{ then } 1 \texttt{ else } F(X-1, G(2X-1)).X$
$G(X) = \texttt{if } X \equiv 0 \pmod 2 \texttt{ then } X/2 \texttt{ else } G(X)$.

Задача 19. Нека R_1 и R_2 са следните програми:

R_1: $F_1(X)$, where
$F_1(X) =$ if $X=0$ then 1 else $2*F_1(X-1)$.

R_2: $F_2(X)$, where
$F_2(X) =$ if $X=0$ then 1
else if $X \equiv 1 \pmod 2$ then $2*(F_2((X-1)/2))^2$
else $(F_2(X/2))^2$.

Да се покаже, че $D_N(R_1) = D_N(R_2)$.

Задача 20. За рекурсивните програми R_1 и R_2

R_1: $F(X,Y)$, where
$F(X,Y) =$ if $X \equiv 0 \pmod 2$ then $X/2$
else $F(X+Y, F(X,Y+1))$,

R_2: $F(X,Y)$, where
$F(X,Y) =$ if $X=0$ then Y else $X*F(X-1, G(X-1,Y))$,
$G(X,Y) =$ if $X=0$ then 1 else $G(X,Y)$,

да се намерят $D_V(R_i)$ и $D_N(R_i)$, $i=1,2$.

Задача 21*. Нека R е следната рекурсивна програма в типа Int:

$F(X,Y)$, where
$F(X,Y) =$ if $X \leqq a$ then b else $F((X-Y, F(X,Y)+c)$,

където a, b и c са фиксирани цели числа. Да се намерят $D_V(R)$ и $D_N(R)$.

Задача 22. Да се покаже, че всяка частично рекурсивна функция може да се пресметне с помощта на рекурсивна програма с предаване на параметрите по име в типа данни

$$\langle \mathbb{N};\ \lambda a.a+1, \lambda a.a \dot{-} 1;\ a=0? \rangle.$$

Забележка. Сравнете със зад. 16, § 4.1.

Задача 23. Да се покаже, че ако типът данни Data е в естествените числа и има изчислими основни операции, то за всяка рекурсивна програма R в Data функцията $D_N(R)$ е изчислима.

§ 4.3. Правило на Скот за доказване на свойства на рекурсивни програми

Нека $A_1, \ldots, A_k$ са области на Скот, $A = A_1 \times \ldots \times A_k$ е декартовото произведение на $A_1, \ldots, A_k$. Да означим с $\perp$ най-малкия елемент на A.

Нека P е свойство в A. Свойството P наричаме *непрекъснато*, ако при всеки избор на монотонно растяща редица $\overline{a}_0, \overline{a}_1, \ldots, \overline{a}_r, \ldots$ от елементи на A е в сила импликацията

$$\forall r (P(\overline{a}_r)) \Longrightarrow P(\bigcup \overline{a}_r).$$

Нека $f_i : A \to A_i$, $i = 1, \ldots, k$, са непрекъснати изображения и $\overline{c} = (c_1, \ldots, c_k)$ е най-малкото решение на системата

$$f_i(\mathbb{X}_1, \ldots, \mathbb{X}_k) = \mathbb{X}_i, \quad i = 1, \ldots, k.$$

Правило на Скот. *Нека P е непрекъснато свойство, за което е вярно $P(\perp)$, и при всеки избор на $\overline{a} \in A$ е в сила импликацията*

$$P(\overline{a}) \Longrightarrow P(f_1(\overline{a}), \ldots, f_k(\overline{a})).$$

Тогава е вярно $P(\overline{c})$.

Твърдението е непосредствено следствие от зад. 4, § 3.3.

Да разгледаме рекурсивната програма R:

$\tau_0(X_1, \ldots, X_n, F_1^{m_1}, \ldots, F_k^{m_k})$, where
$F_i^{m_i}(X_1, \ldots, X_{m_i}) = \tau_i(X_1, \ldots, X_{m_i}, F_1^{m_1}, \ldots, F_k^{m_k})$, $i = 1, \ldots, k$.

В зависимост от начина на предаване на параметрите имаме две семантики $D_V(R)$ и $D_N(R)$ на R. Ако разглеждаме свойства на $D_V(R)$, прилагаме правилото на Скот за $(\mathcal{F}, \subseteq, \varnothing)$, $\mathcal{F} = \mathcal{F}_{m_1} \times \ldots \times \mathcal{F}_{m_k}$, докато за доказателства на свойства на $D_N(R)$ работим в областта на Скот $(\mathcal{F}^{\perp}, \sqsubseteq, \Omega)$, $\mathcal{F}^{\perp} = \mathcal{F}^{\perp}_{m_1} \times \ldots \times \mathcal{F}^{\perp}_{m_k}$.

Задача 1. Да се докаже, че конюнкция на непрекъснати свойства е непрекъснато свойство.

Задача 2. Нека Γ_1 и Γ_2 са непрекъснати изображения над $\mathcal{F}$ (или $\mathcal{F}^{\perp}$). Да се докаже, че всяко от следните свойства е непрекъснато:

$$P(\overline{\varphi}) \iff \Gamma_1(\overline{\varphi}) \subseteq \Gamma_2(\overline{\varphi});$$

$$Q(\overline{\varphi}) \iff \Gamma_1(\overline{\varphi}) = \Gamma_2(\overline{\varphi}).$$

Ще покажем първо как правилото на Скот се прилага за доказване на свойства на $D_V(R)$.

Нека P е условие за частична коректност в $\mathcal{F}_n$ от вида

$$P(f) \iff \forall\overline{a}(!f(\overline{a}) \,\&\, I(\overline{a}) \Longrightarrow O(\overline{a}, f(\overline{a}))).$$

Ако искаме да докажем, че P е в сила за $D_V(R)$, разглеждаме свойството $\overline{P}$ в $\mathcal{F}$, дефинирано с еквивалентността

$$\overline{P}(\overline{\psi}) \iff \forall\overline{a}(!\tau_0(\overline{a},\overline{\psi}) \,\&\, I(\overline{a}) \Longrightarrow O(\overline{a}, \tau_0(\overline{a},\overline{\psi}))).$$

Задача 3. Да се докаже, че свойството $\overline{P}$ е непрекъснато.

Упътване. Използвайте, че операторът

$$\Gamma_{\tau_0}(\overline{\psi})(\overline{a}) \simeq \tau_0(\overline{a},\overline{\psi})$$

е непрекъснат, и следователно е компактен и монотонен (зад. 1, § 4.1).

Нека $\overline{\varphi} = (\varphi_1,\dots,\varphi_k)$ е най-малкото решение на системата, съответна на R в $\mathcal{F}$. От дефиницията на семантиката $D_V(R)$ следва, че $P(D_V(R)) \iff \overline{P}(\overline{\varphi})$. Както ще видим в следващите задачи, понякога се налага да усилим $\overline{P}$, като добавяме допълнителни условия, описващи свойствата на функциите $\varphi_1,\dots,\varphi_k$.

Задача 4. Нека R е следната рекурсивна програма в типа Nat:

$$\begin{aligned}
&F_1(X), \quad \texttt{where}\\
&F_1(X) = \texttt{if } X = 0 \texttt{ then } 1 \texttt{ else } \alpha(X) * (F_1(F_2(X)))^2\\
&F_2(X) = \texttt{if } X \leqq 1 \texttt{ then } 0 \texttt{ else } F_2(X-2) + 1,
\end{aligned}$$

където $\alpha(a) = \texttt{if } a \equiv 0 \pmod 2 \texttt{ then } 1 \texttt{ else } 2$.

Да се докаже, че

(1) $$\forall a(!D_V(R)(a) \Longrightarrow D_V(R)(a) \simeq 2^a).$$

Доказателство. Разглеждаме свойствата

$$P_1(\psi_1,\psi_2) \iff \forall a(!\psi_1(a) \Longrightarrow \psi_1(a) \simeq 2^a),$$

$$P_2(\psi_1,\psi_2) \iff \forall a(!\psi_2(a) \Longrightarrow \psi_2(a) \simeq [a/2]).$$

Всяко от тези свойства е условие от тип частична коректност и съгласно зад. 3 е непрекъснато. От зад. 1 следва, че тяхната конюнкция P е непрекъснато свойство. Ще приложим правилото на Скот за P.

Имаме $P(\varnothing,\varnothing)$. Да предположим $P(\psi_1,\psi_2)$. Нека

$$\rho_1(a) \simeq \texttt{if } a = 0 \texttt{ then } 1 \texttt{ else } \alpha(a) * (\psi_1(\psi_2(a)))^2,$$

$$\rho_2(a) \simeq \texttt{if } a \leqq 1 \texttt{ then } 0 \texttt{ else } \psi_2(a-2) + 1.$$

Използвайки индукционното предположение, ще проверим валидността на $P_1(\rho_1,\rho_2)$ и $P_2(\rho_1,\rho_2)$.

За да проверим $P_1(\rho_1,\rho_2)$, да предположим, че $!\rho_1(a)$.

Ако $a = 0$, то $\rho_1(a) = 1 = 2^0$.

Нека $a > 0$. Тогава $\rho_1(a) = \alpha(a) * (\psi_1(\psi_2(a)))^2$. Следователно $!\psi_2(a)$ и от индукционното предположение следва, че $\psi_2(a) = [a/2]$. Така $!\psi_1([a/2])$ и от $P_1(\psi_1,\psi_2)$ имаме, че $\rho_1(a) = \alpha(a) * (2^{[a/2]})^2 = 2^a$.

Остава да проверим, че $P_2(\rho_1,\rho_2)$ е изпълнено. Нека $!\rho_2(a)$.

Ако $a \leqq 1$, то $\rho_2(a) = 0 = [a/2]$.

Нека $a > 1$. Тогава $\rho_2(a) = \psi_2(a-2)+1$. Следователно $!\psi_2(a-2)$ и по индукционното предположение $\psi_2(a-2) = [(a-2)/2]$. Така $\rho_2(a) = [(a-2)/2] + 1 = [a/2]$.

Оттук, съгласно правилото на Скот, заключаваме, че P е в сила за най-малкото решение (φ_1,φ_2) на системата, съответна на R в $\mathcal{F}_1 \times \mathcal{F}_1$, откъдето следва (1).

Задача 5. Нека R е следната рекурсивна програма в типа Nat:

$$\begin{aligned}
&F_1(X), \quad \texttt{where}\\
&F_1(X) = \texttt{if } X = 0 \texttt{ then } 1 \texttt{ else } F_1(X-1) + F_2(X-1)\\
&F_2(X) = \texttt{if } X = 0 \texttt{ then } 0 \texttt{ else } F_2(X+1).
\end{aligned}$$

Да се докаже, че

$$\forall a(!D_V(R)(a) \Longrightarrow D_V(R)(a) \simeq 1).$$

Задача 6. Да се докаже, че за всяка от следващите две програми R в типа данни Nat е вярно условието за частична коректност

$$\forall a \forall b(!D_V(R)(a,b) \Longrightarrow D_V(R)(a,b) = \text{НОД}(a,b)).$$

Следва ли оттук, че двете програми пресмятат по стойност една и съща функция?

a) $$\begin{aligned}
&F(X,Y), \quad \texttt{where}\\
&F(X,Y) = \texttt{if } X = Y \texttt{ then } X\\
&\qquad\qquad \texttt{else if } X > Y \texttt{ then } F(G(X,Y),Y)\\
&\qquad\qquad\qquad \texttt{else } F(X,G(Y,X))\\
&G(X,Y) = \texttt{if } X = Y \texttt{ then } 0 \texttt{ else } G(X,Y+1)+1;
\end{aligned}$$

б) $F(X,Y)$, where
$F(X,Y) = \text{if } Y=0 \text{ then } X$
$\quad \text{else if } X>Y \text{ then } F(Y,G(X,Y))$
$\quad \text{else } F(X,G(Y,X))$
$G(X,Y) = \text{if } X<Y \text{ then } X \text{ else } G(X-Y,Y)$.

Задача 7. Нека R е следната програма в типа данни Int:

$F(X,X)$, where
$F(X,Y) = \text{if } X=Y \text{ then } G(X,Y)$
$\quad \text{else } F(X+1,Y)-1$
$G(X,Y) = \text{if } X=0 \text{ then } Y$
$\quad \text{else } G(X-1,Y)+1$

Да се докаже, че за всяко цяло число a: ·

$$!D_V(R)(a) \Longrightarrow D_V(R)(a) = 2a.$$

Задача 8. Да се докаже, че следващата програма пресмята функцията на Фибоначи:

$F(N,1,1)$, where
$F(N,X,Y) = \text{if } N=0 \text{ then } X$
$\quad \text{else } F(N-1,Y,X+Y)$.

Забележка. Това е т. нар. *акумулаторен алгоритъм* за функцията на Фибоначи.

Задача 9. Нека R е следната програма в типа данни Nat:

$F(X,X,0)$, where
$F(X,Y,S) = \text{if } Y=0 \text{ then } S$
$\quad \text{else } F(X,Y-1,G(X,S))$
$G(X,Y) = \text{if } X=0 \text{ then } Y$
$\quad \text{else } G(X-1,Y+1)$.

Да се докаже, че

$$\forall a(!D_V(R)(a) \Longrightarrow D_V(R)(a) = a^2).$$

Задача 10. Нека R е следната програма в типа данни Nat:

$F(X,X,1)$, where
$F(X,Y,P) = \text{if } Y=0 \text{ then } P$
$\quad \text{else } F(X,Y-1,G(X,P,0))$
$G(X,Y,S) = \text{if } X=0 \text{ then } S$
$\quad \text{else } G(X-1,Y,S+Y)$.

Да се докаже, че

$$\forall a(!D_V(R)(a) \Longrightarrow D_V(R)(a) = a^2).$$

Задача 11. Нека R е следната програма в типа данни Nat:

$F(X,Y,1)$, where
$F(X,Y,Z) = \text{if } X=1 \text{ then } Z$
$\quad \text{else } F(X-1,Y,Z.G(X,Y))$
$G(X,Y) = \text{if } Y=0 \text{ then } 1$
$\quad \text{else } X.G(X,Y-1)$.

Да се докаже, че

$$\forall a \forall b(!D_V(R)(a,b) \Longrightarrow D_V(R)(a,b) = (a!)^b).$$

Задача 12. Нека R е следната програма в типа данни Int:

$F(X,Y).X + G(X,Y).Y$, where
$F(X,Y) = \text{if } X=Y \text{ then } 1$
$\quad \text{else if } X>Y \text{ then } G(Y,X)$
$\quad \text{else } F(X,Y-X)-G(X,Y-X)$
$G(X,Y) = \text{if } X=Y \text{ then } 0$
$\quad \text{else if } X>Y \text{ then } F(Y,X)$
$\quad \text{else } G(X,Y-X)$.

Да се докаже, че за всеки две *естествени* числа a и b:

$$!D_V(R)(a,b) \Longrightarrow D_V(R)(a,b) = \text{НОД}(a,b).$$

Задача 13. Нека R е следната програма в типа данни Nat:

$G(X,Y)$, where
$F(X) = \text{if } X \leqq 1 \text{ then } X \text{ else } F(X-2)$
$G(X,Y) = \text{if } X=0 \text{ then } Y \text{ else } G(X-1,F(Y))$.

Кои от изброените условия:

а) $\forall x \forall y(!D_V(R)(x,y) \Longrightarrow D_V(R)(x,y) \leqq 1)$;

б) $\forall x \forall y(x>0 \Longrightarrow \neg !D_V(R)(x,y))$;

в) $\forall x \forall y(\neg !D_V(R)(x,y) \Longrightarrow x>0)$;

г) $\forall x \forall y(!D_V(R)(x,y) \Longrightarrow D_V(R)(x,y) \equiv y \pmod 2)$,

са верни за $D_V(R)$?

Задача 14. Да се намери функцията (в естествените числа), която се пресмята по стойност от следващата програма:

$$
\begin{aligned}
&F(X,2), \quad \texttt{where}\\
&F(X,Y) = \texttt{if } X \leqq Y \texttt{ then } X\\
&\qquad\qquad\qquad\qquad \texttt{else } G(Y, F(X, Y+1))\\
&G(X,Y) = \texttt{if } Y = 0 \texttt{ then } 1\\
&\qquad\qquad\qquad\qquad \texttt{else } X.G(X, Y-1).
\end{aligned}
$$

За да докажем едно непрекъснато свойство, задаващо връзката между семантиките $D_V(R_1)$ и $D_V(R_2)$ на две рекурсивни програми R_1 и R_2, първо преименуваме функционалните променливи на R_2 така, че те да не съвпадат с тези на R_1. След това прилагаме правилото на Скот към системата, съответстваща на рекурсивната програма, получена от обединяване на R_1 и преименуваната R_2. Съобразете, че най-малкото решение на тази нова система ще се състои от най-малкото решение на системата, съответна за R_1, и най-малкото решение на системата, съответна за R_2.

Задача 15. Нека R_1 и R_2 са следните рекурсивни програми в типа Nat:

$$
\begin{aligned}
&R_1 : F_1(X), \quad \texttt{where}\\
&\qquad F_1(X) = \texttt{if } X = 0 \texttt{ then } 1 \texttt{ else } F_1(X-1) * X\\
&R_2 : F_2(X,1), \quad \texttt{where}\\
&\qquad F_2(X,Y) = \texttt{if } X = 0 \texttt{ then } Y \texttt{ else } F_2(X-1, X*Y).
\end{aligned}
$$

Да се докаже, че

$$\forall a(D_V(R_1)(a) \simeq D_V(R_2)(a)).$$

Упътване. Разгледайте свойството P в $\mathcal{F}_1 \times \mathcal{F}_2$

$$P(\psi_1,\psi_2) \iff \forall a \forall b(\psi_1(a) * b \simeq \psi_2(a,b)).$$

Покажете, че P е непрекъснато свойство, като използвате зад. 2, и приложете правилото на Скот към системата

$$
\left|
\begin{aligned}
&\Gamma_1(\psi_1,\psi_2)(a) \simeq \texttt{if } a = 0 \texttt{ then } 1 \texttt{ else } \psi_1(a-1) * a\\
&\Gamma_2(\psi_1,\psi_2)(a,b) \simeq \texttt{if } a = 0 \texttt{ then } b \texttt{ else } \psi_2(a-1, a*b).
\end{aligned}
\right.
$$

Задача 16. Нека R_1 и R_2 са следните рекурсивни програми в типа Nat:

$$
\begin{aligned}
&R_1 : F_1(X,0,X), \quad \texttt{where}\\
&\qquad F_1(X,Y,Z) = \texttt{if } X = 0 \texttt{ then } Y \texttt{ else } F_1(X-1, Y+Z, Z)\\
&R_2 : F_2(X,0), \quad \texttt{where}\\
&\qquad F_2(X,Y) = \texttt{if } X = 0 \texttt{ then } Y \texttt{ else } F_2(X-1, 2*X+Y-1).
\end{aligned}
$$

Да се докаже, че

$$\forall a(D_V(R_1)(a) \simeq D_V(R_2)(a)).$$

Упътване. Разгледайте свойството P в $\mathcal{F}_3 \times \mathcal{F}_2$

$$P(\psi_1,\psi_2) \iff \forall a \forall b(a \geq b \Longrightarrow \psi_1(b, a*(a-b), a) \simeq \psi_2(b, a^2 - b^2)).$$

Покажете, че P е непрекъснато и приложете правилото на Скот.

Задача 17. Нека p е едноместна тотална релация в D, а k и h са едноместни тотални функции в D. Нека R_1 и R_2 са следните рекурсивни програми в типа Data:

$$
\begin{aligned}
&R_1 : F_1(X,Y), \quad \texttt{where}\\
&\qquad F_1(X,Y) = \texttt{if } p(X) \texttt{ then } Y \texttt{ else } h(F_1(k(X),Y))\\
&R_2 : F_2(X,Y), \quad \texttt{where}\\
&\qquad F_2(X,Y) = \texttt{if } p(X) \texttt{ then } Y \texttt{ else } F_2(k(X), h(Y)).
\end{aligned}
$$

Да се докаже, че

$$\forall a \forall b(D_V(R_1)(a,b) \simeq D_V(R_2)(a,b)). \tag{2}$$

Забележка. Сравнете със зад. 11, § 3.3.

Доказателство. Разглеждаме свойствата P_1 и P_2 в $\mathcal{F}_2 \times \mathcal{F}_2$:

$$P_1(\psi_1,\psi_2) \iff \forall a \forall b(\psi_1(a,b) \simeq \psi_2(a,b)).$$

$$P_2(\psi_1,\psi_2) \iff \forall a \forall b[\psi_2(a,h(b)) \simeq h(\psi_2(a,b))].$$

От зад. 1 и зад. 2 следва, че тяхната конюнкция P е непрекъснато свойство. Прилагаме правилото на Скот.

Имаме $P(\varnothing,\varnothing)$. Да предположим, че $P(\psi_1,\psi_2)$. Нека

$$\rho_1(a,b) \simeq \texttt{if } p(a) \texttt{ then } b \texttt{ else } h(\psi_1(k(a),b))$$

$$\rho_2(a,b) \simeq \texttt{if } p(a) \texttt{ then } b \texttt{ else } \psi_2(k(a),h(b)).$$

Ще покажем, че $P(\rho_1,\rho_2)$. Да проверим първо $P_1(\rho_1,\rho_2)$.

Ако $p(a)$, то ясно е, че $\rho_1(a,b) = b = \rho_2(a,b)$.

Нека $\neg p(a)$. Тогава

$$\rho_1(a,b) \simeq h(\psi_1(k(a),b)) \simeq h(\psi_2(k(a),b)) \text{ (от } P_1(\psi_1,\psi_2))$$
$$\simeq \psi_2(k(a),h(b)) \text{ (от } P_2(\psi_1,\psi_2)) \simeq \rho_2(a,b).$$

Сега да проверим $P_2(\rho_1,\rho_2)$.
Ако $p(a)$, то $\rho_2(a,h(b)) = h(b) = h(\rho_2(a,b))$.
Ако $\neg p(a)$, тогава

$$\rho_2(a,h(b)) \simeq \psi_2(k(a),h(h(b))) \simeq h(\psi_2(k(a),h(b))) \text{ (от } P_2(\psi_1,\psi_2))$$
$$\simeq h(\rho_2(a,b)).$$

От правилото на Скот следва, че (2) е вярно.

Задача 18. Нека p и q са едноместни, навсякъде определени релации в D, а k, g и h са едноместни тотални функции в D. Нека R_1 и R_2 са следните рекурсивни програми в типа Data:

$R_1 : F_1(X)$, where
 $F_1(X) = \texttt{if } p(X) \texttt{ then } F_1(F_2(h(X))) \texttt{ else } F_2(g(X))$
 $F_2(X) = \texttt{if } q(X) \texttt{ then } k(F_2(F_1(X))) \texttt{ else } k(h(X))$

$R_2 : F_3(X)$, where
 $F_3(X) = \texttt{if } p(X) \texttt{ then } F_3(k(F_4(h(X)))) \texttt{ else } k(F_4(g(X)))$
 $F_4(X) = \texttt{if } q(X) \texttt{ then } k(F_4(F_3(X))) \texttt{ else } h(X)$.

Да се докаже, че

$$\forall a(D_V(R_1)(a) \simeq D_V(R_2)(a)).$$

Упътване. Разгледайте свойството P

$$P(\psi_1,\psi_2,\psi_3,\psi_4) \iff \forall a(\psi_1(a) \simeq \psi_3(a) \,\&\, \psi_2(a) \simeq k(\psi_4(a))).$$

Задача 19. Нека R е следната рекурсивна програма в типа List:

$R : F(X,Y)$, where
 $F(X,Y) = \texttt{if } X = \Lambda \texttt{ then } Y \texttt{ else } \mathrm{cons}(\mathrm{head}(X), F(\mathrm{tail}(X),Y))$.

Означаваме с $\mathrm{conc}(a,b) = D_V(R)(a,b)$. Да се докаже, че операцията конкатенация е асоциативна, т.е.

$$\forall a \forall b \forall c[\mathrm{conc}(\mathrm{conc}(a,b),c) = \mathrm{conc}(a,\mathrm{conc}(b,c))].$$

Забележка. Съгласно зад. 9, § 4.1 функцията $\lambda a,b.D_V(R)(a,b)$ дефинира операцията конкатенация на две думи.

Доказателство. Разгледайте следното свойство P:

$$P(\psi) \iff \forall a \forall b \forall c[\mathrm{conc}(\psi(a,b),c) \simeq \psi(a,\mathrm{conc}(b,c))].$$

Покажете, че P е непрекъснато свойство. Прилагаме правилото на Скот.

Имаме $P(\varnothing)$. Да предположим, че $P(\psi)$. Нека

$$\rho(a,b) \simeq \texttt{if } a = \Lambda \texttt{ then } b \texttt{ else } \mathrm{cons}(\mathrm{head}(a), \psi(\mathrm{tail}(a),b)).$$

Ще покажем, че $P(\rho)$.

$\mathrm{conc}(\rho(a,b),c)$

$\simeq \mathrm{conc}([\texttt{ if } a = \Lambda \texttt{ then } b \texttt{ else } \mathrm{cons}(\mathrm{head}(a),\psi(\mathrm{tail}(a),b))],c)$

$\simeq \texttt{ if } a = \Lambda \texttt{ then } \mathrm{conc}(b,c) \texttt{ else } \mathrm{conc}(\mathrm{cons}(\mathrm{head}(a),\psi(\mathrm{tail}(a),b)),c)$

$\simeq \texttt{ if } a = \Lambda \texttt{ then } \mathrm{conc}(b,c) \texttt{ else } \mathrm{cons}(\mathrm{head}(a),\mathrm{conc}(\psi(\mathrm{tail}(a),b),c))$ (от деф. на conc)

$\simeq \texttt{ if } a = \Lambda \texttt{ then } \mathrm{conc}(b,c) \texttt{ else } \mathrm{cons}(\mathrm{head}(a),\psi(\mathrm{tail}(a),\mathrm{conc}(b,c)))$ (от $P(\psi)$)

$\simeq \rho(a,\mathrm{conc}(b,c))$ (от деф. на ρ).

Задача 20. Нека R_1 и R_2 са следните рекурсивни програми в типа List:

$R_1 : F(X,\Lambda)$, where
 $F(X,Y) = \texttt{if } X = \Lambda \texttt{ then } Y \texttt{ else } F(\mathrm{tail}(X), \mathrm{cons}(\mathrm{head}(X),Y))$.
$R_2 : F_1(X)$, where
 $F_1(X) \quad = \texttt{if } X = \Lambda \texttt{ then } X \texttt{ else } F_2(\mathrm{head}(X), F_1(\mathrm{tail}(X)))$.
 $F_2(X,Y) = \texttt{if } Y = \Lambda \texttt{ then } X \texttt{ else } \mathrm{cons}(\mathrm{head}(Y), F_2(X,\mathrm{tail}(Y)))$.

Да се докаже, че

$$\forall a(D_V(R_1)(a) \simeq D_V(R_2)(a)).$$

Упътване. Използвайте зад. 8 и зад. 9, § 4.1, или предишната задача.

Забележка. И двете програми от зад. 20 пресмятат функцията $\mathrm{rev}(a)$, която за всяка дума a връща дума със същите символи, но в обратен ред. Първата е известна като акумулаторна програма и има сложност $O(n)$, докато втората има сложност $O(n^2)$.

В следващите няколко примера ще покажем как правилото на Скот се прилага за доказване на свойства на $D_N(R)$.

Да разгледаме рекурсивната програма R:

$\tau_0(X_1, \ldots, X_n, F_1^{m_1}, \ldots, F_k^{m_k})$, where
$F_i^{m_i}(X_1, \ldots, X_{m_i}) = \tau_i(X_1, \ldots, X_{m_i}, F_1^{m_1}, \ldots, F_k^{m_k})$, $i = 1, \ldots, k$.

Да предположим, че искаме да покажем за $D_N(R)$ условие за частична коректност P от вида

$$P(f) \iff \forall \overline{a}(!f(\overline{a}) \,\&\, I(\overline{a}) \Longrightarrow O(\overline{a}, f(\overline{a}))).$$

Най-напред трябва да представим P като непрекъснато условие в областта на Скот $(\mathcal{F}^{\perp}, \sqsubseteq, \Omega)$, $\mathcal{F}^{\perp} = \mathcal{F}^{\perp}_{m_1} \times \ldots \times \mathcal{F}^{\perp}_{m_k}$. Нека условието $Q : \mathcal{F}^{\perp} \to \mathrm{B}$ е дефинирано с еквивалентността

$$Q(\overline{\psi}) \iff \forall \overline{a}_{\overline{a} \in \mathrm{D}^n}(\tau_0(\overline{a}, \overline{\psi}) \neq \perp \& I(\overline{a}) \Longrightarrow O(\overline{a}, \tau_0(\overline{a}, \overline{\psi}))).$$

Задача 21. Да се докаже, че условието Q е непрекъснато.

Упътване. Използвайте, че

$$\Gamma_{\tau_0}(\overline{\psi})(\overline{x}) = \tau_0(\overline{x}, \overline{\psi})$$

е непрекъснато изображение от $\mathcal{F}^{\perp}$ в $\mathcal{F}^{\perp}_n$ (зад. 5, § 4.2).

От дефиницията на $D_N(R)$ следва, че P е в сила за $D_N(R)$ точно тогава, когато Q е в сила за най-малкото решение $\overline{\varphi}$ на системата, съответна на R в $\mathcal{F}^{\perp}$.

За доказателството на $Q(\overline{\varphi})$ използваме правилото на Скот, приложено за системата, съответна на R в областта на Скот $(\mathcal{F}^{\perp}, \sqsubseteq, \Omega)$. Тук също се налага да усилим Q, за да покажем допълнителни свойства на $\overline{\varphi}$. Свойството Q ще наричаме съответно на условието P за програмата R.

Да отбележим, че тъй като $D_V(R) \subseteq D_N(R)$ за всяка рекурсивна програма R, то ако едно условие за частична коректност P е вярно за $D_N(R)$, то P е непременно вярно за $D_V(R)$.

Задача 22. Нека R е следната рекурсивна програма в Nat:

```
F(X),  where
F(X) = if X ≦ 1 then 1 else if X ≡ 0 (mod 2)
                            then X/2
                            else F(F(3 * [X/2] + 2)).
```

Да се докаже, че

(3) $$\forall a(!D_N(R)(a) \,\&\, a > 1 \Longrightarrow D_N(R)(a) \leqq a/2).$$

Доказателство. Разглеждаме следните условия в $\mathcal{F}^{\perp}_1$:

$Q_1(\psi) \iff \forall a_{a \in \mathbb{N}}(\psi(a) \neq \perp \& a \leqq 1 \Longrightarrow \psi(a) = 1)$,

$Q_2(\psi) \iff \forall a_{a \in \mathbb{N}}(\psi(a) \neq \perp \& a > 1 \Longrightarrow \psi(a) \leqq a/2)$,

$Q_3(\psi) \iff \psi(\perp) = \perp$.

Свойствата Q_1 и Q_2 са съответни на условия за частична коректност и от зад. 21 следва, че са непрекъснати. Използвайки зад. 1, § 4.2, съобразяваме, че свойството Q_3 също е непрекъснато. Нека $Q = Q_1 \& Q_2 \& Q_3$. Свойството Q също е непрекъснато съгласно зад. 1 от този параграф.

Имаме $Q(\Omega)$.

Да предположим, че е вярно $Q(\psi)$ за $\psi \in \mathcal{F}^{\perp}_1$. Нека

$$\leqq^*, +^*, \equiv^*, *^*, [\,]^*, /^*$$

са точните продължения на $\leqq, +, \equiv, *, [\,], /$ в $\mathcal{F}^{\perp}_2$.

Разглеждаме изображението ξ, дефинирано с

```
ξ(x) = if x ≦* 1 then 1 else if x ≡* 0 (mod 2)
                             then x/*2
                             else ψ(ψ(3 ** [x/*2]* +* 2)).
```

Ще покажем, че $Q(\xi)$.

Условието $Q_1(\xi)$ следва от дефиницията на ξ.

За да проверим, че $Q_2(\xi)$ е изпълнено, да предположим, че $a \in \mathbb{N}$, $a > 1$ и $\xi(a) \neq \perp$.

Ако a е четно, то $\xi(a) = a/2$.

Нека $a = 2k + 1$. Тогава $\xi(a) = \psi(\psi(3[a/2] + 2)) = \psi(\psi(3k + 2))$. От $Q_3(\psi)$ имаме $\psi(\perp) = \perp$. Тъй като $\xi(a) \neq \perp$, то $\psi(3k + 2) \neq \perp$. Имаме два случая:

1. $\psi(3k + 2) \leqq 1$. Тогава от $Q_1(\psi)$ следва, че $\xi(a) = 1 \leqq a/2$, защото $a > 1$ и a е нечетно.

2. $\psi(3k + 2) > 1$. Тогава по $Q_2(\psi)$ имаме, че

$$\psi(\psi(3k + 2)) \leqq (\psi(3k + 2))/2.$$

Тъй като $3k + 2 > 1$, то като приложим още веднъж $Q_2(\psi)$, получаваме $\psi(3k + 2) \leqq (3k + 2)/2$. Така

$$\psi(\psi(3k + 2)) \leqq (\psi(3k + 2))/2 \leqq (3k + 2)/4 \leqq a/2.$$

Условието $Q_3(\xi)$ следва от дефиницията на стойност на терм.

По правилото на Скот условието Q е вярно за най-малкото решение на системата, съответна на R в областта $\mathcal{F}_1^{\perp}$, откъдето следва (3).

Задача 23. Нека да разгледаме рекурсивната програма R в Nat:

$F_1(X)$, where
$F_1(X) = \texttt{if } X > 10 \texttt{ then } X - 10 \texttt{ else } F_1(F_2(X+13))$
$F_2(X) = \texttt{if } X > 10 \texttt{ then } X - 10 \texttt{ else } F_2(X+3)$.

Да се докаже, че

(4) $\quad \forall a(!D_N(R)(a) \,\&\, a \leqq 10 \Longrightarrow D_N(R)(a) \simeq \mathrm{rem}(3, a+1)+1)$.

Доказателство. Разглеждаме следните условия в $\mathcal{F}_1^{\perp} \times \mathcal{F}_1^{\perp}$:

$Q_1(\psi_1, \psi_2) \iff \forall a_{a\in\mathbb{N}}(\psi_1(a) \neq \perp \& \, a \leqq 10 \Longrightarrow \psi_1(a) = \mathrm{rem}(3, a+1)+1)$,

$Q_2(\psi_1, \psi_2) \iff \forall a_{a\in\mathbb{N}}(\psi_1(a) \neq \perp \& \, a > 10 \Longrightarrow \psi_1(a) = a - 10)$,

$Q_3(\psi_1, \psi_2) \iff \forall a_{a\in\mathbb{N}}(\psi_2(a) \neq \perp \& \, a > 10 \Longrightarrow \psi_2(a) = a - 10)$,

$Q_4(\psi_1, \psi_2) \iff \psi_1(\perp) = \perp$.

Непрекъснатостта на Q_1, Q_2 и Q_3 следва от зад. 21, а тази на Q_4 — от зад. 1, § 4.2. Да означим с Q конюнкцията на Q_1, Q_2, Q_3 и Q_4.

Ясно е, че $Q(\Omega, \Omega)$.

Да предположим, че ψ_1, ψ_2 са елементи на $\mathcal{F}_1^{\perp}$, за които е в сила $Q(\psi_1, \psi_2)$. Нека $>^*, -^*, +^*$ са точните продължения на $>, -, +$ в $\mathcal{F}_2^{\perp}$. Да разгледаме изображенията

$$\xi_1(x) = \texttt{if } x >^* 10 \texttt{ then } x -^* 10 \texttt{ else } \psi_1(\psi_2(x +^* 13)),$$

$$\xi_2(x) = \texttt{if } x >^* 10 \texttt{ then } x -^* 10 \texttt{ else } \psi_2(x +^* 3).$$

Ще проверим $Q(\xi_1, \xi_2)$.

За проверката на $Q_1(\xi_1, \xi_2)$ да предположим, че $a \in \mathbb{N}$, $a \leqq 10$ и $\xi_1(a) \neq \perp$. Тогава $\xi_1(a) = \psi_1(\psi_2(a+13))$. От $Q_4(\psi_1, \psi_2)$ следва, че $\psi_2(a+13) \neq \perp$. Тъй като $a + 13 > 10$, то от $Q_3(\psi_1, \psi_2)$ получаваме $\psi_2(a+13) = a + 13 - 10 = a + 3$. Имаме два случая:

Ако $a + 3 \leqq 10$, то от $Q_1(\psi_1, \psi_2)$ следва, че

$$\psi_1(a+3) = \mathrm{rem}(3, a+4) + 1 = \mathrm{rem}(3, a+1) + 1.$$

Ако $a + 3 > 10$, то от $Q_2(\psi_1, \psi_2)$ следва, че

$$\psi_1(a+3) = a + 3 - 10 = (a+1) - 9 + 1 = \mathrm{rem}(3, a+1) + 1,$$

тъй като $9 \leqq a + 1 \leqq 11$.

Другите условия следват директно от дефиницията на ξ_1 и ξ_2.

По правилото на Скот свойството Q е вярно за най-малкото решение на системата, съответна на R в областта $\mathcal{F}_1^{\perp} \times \mathcal{F}_1^{\perp}$, откъдето следва и (4).

Задача 24. Нека R е рекурсивната програма в типа Int:

$F_1(X)$, where
$F_1(X) = \texttt{if } X > 100 \texttt{ then } X - 10 \texttt{ else } F_1(F_2(X+11))$
$F_2(X) = \texttt{if } X > 100 \texttt{ then } X - 10 \texttt{ else } F_2(F_1(X+11))$.

Да се докаже, че

$$\forall a(!D_N(R)(a) \,\&\, a \leqq 101 \Longrightarrow D_N(R)(a) \simeq 91).$$

В следващите задачи рекурсивните програми са зададени в типа Nat.

Задача 25. Дадена е рекурсивната програма R:

$R : F_1(X)$, where
$F_1(X) = F_2(F_1(X))$
$F_2(X) = 0$.

Да се докаже, че

$$\forall a(!D_V(R)(a) \Longrightarrow D_V(R)(a) > 0).$$

Проверете, че това свойство не е в сила за $D_N(R)$.

Задача 26. Нека R е рекурсивната програма:

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

Да се докаже, че

$$\forall a \forall b(!D_N(R)(a, b) \,\&\, a > 0 \Longrightarrow D_N(R)(a, b) \simeq 1).$$

Упътване. Разгледайте следните условия в $\mathcal{F}_2^{\perp}$:

$Q_1(\psi) \iff \forall a_{a\in\mathbb{N}} \forall b_{b\in\mathbb{N}}(\psi(a, b) \neq \perp \& \, a > 0 \Longrightarrow \psi(a, b) = 1)$,

$Q_2(\psi) \iff \forall b_{b\in\mathbb{N}}(\psi(0,b) \neq \bot \Longrightarrow \psi(0,b) = b)$,

$Q_3(\psi) \iff \forall a_{a\in\mathbb{N}}(\psi(a,\bot) = \bot)$.

Задача 27. Нека R е рекурсивната програма:

$F(X,Y)$, `where`
$F(X,Y) =$ `if` $X * Y = 0$ `then` $X + Y + 1$ `else` $F(F(X-1,1), Y-1)$.

Да се докаже, че

$$\forall a \forall b(!D_N(R)(a,b) \Longrightarrow D_N(R)(a,b) \simeq a+b+1).$$

Задача 28. Нека R е следната програма:

$F(X,Y)$, `where`
$F(X,Y) =$ `if` $X \leqq 10$ `then` 10
`else` $F(X \dot{-} Y, F(X,Y)+10)$.

Да се докаже, че

$$\forall a \forall b(!D_N(R)(a,b) \Longrightarrow D_N(R)(a,b) = 10).$$

Задача 29. Нека R е следната програма:

$F(X,X)$, `where`
$F(X,Y) =$ `if` $X \leqq 1$ `then` 1
`else if` $X \equiv 0 \pmod 2$ `then` $F(X/2, F(X,Y)) + 1$
`else` $F(X-1, F(X,Y))$.

а) Да се намери $D_V(R)$.

б) Да се докаже, че

$$\forall a(!D_N(R)(a) \Longrightarrow D_N(R)(a) = lh(a)),$$

където $lh(a)$ е дължината на a в двоичен запис.

в) Да се докаже, че $D_N(R)(a)$ е дефинирана за всяко естествено число a.

Задача 30. Нека R е следната програма:

$F(X,Y)$, `where`
$F(X,Y) =$ `if` $X \equiv 0 \pmod 2$ `then` $X/2$
`else` $[F(X+Y, F(X,Y))/2]$.

Да се докаже, че

$$\forall a \forall b(!D_N(R)(a,b) \Longrightarrow D_N(R)(a,b) \leqq (a+b)/2).$$

Задача 31. Нека R е рекурсивната програма:

$F(X,Y,0)$, `where`
$F(X,Y,Z) =$ `if` $X = 0$ `then` Z
`else if` $X \equiv 0 \pmod 2$ `then` $F(X/2, 2*Y, Z)$
`else` $F(X-1, Y, Z+Y)$.

Да се докаже, че

$$\forall a \forall b(!D_N(R)(a,b) \Longrightarrow D_N(R)(a,b) \simeq a * b).$$

Задача 32. Нека R_1 и R_2 са следните рекурсивни програми:

$R_1 : F_1(X)$, `where`
$F_1(X) =$ `if` $X \leqq 1$ `then` 1 `else` $F_1(X-1) + F_1(X-2)$

$R_2 : F_2(X)$, `where`
$F_2(X) =$ `if` $X \leqq 1$ `then` 1 `else` $2 * F_2(X-2)$.

Да се докаже, че

$$\forall a \forall b(!D_N(R_1)(a) \,\&\, !D_N(R_2)(b) \,\&\, a \geqq b \Longrightarrow D_N(R_1)(a) \geqq D_N(R_2)(b)).$$

Задача 33. Нека R е следната рекурсивна програма:

$F(X,Y)$, `where`
$F(X,Y) =$ `if` $X = 0$ `then` 0 `else` $F(X \dot{-} Y, F(X,Y)) + X$

Да се докаже, че

$$\forall a \forall b(!D_N(R)(a,b) \Longrightarrow D_N(R)(a,b) \geqq a).$$

Задача 34. Нека R е следната рекурсивна програма:

$F(X,Y)$, `where`
$F(X,Y) =$ `if` $X = Y$ `then` 1 `else` $(Y+2) * (Y+1) * F(X, Y+2)$

Да се докаже, че

$$\forall a \forall b(a < b \Longrightarrow \neg !D_N(R)(a,b)).$$

Упътване. Проверете, че условието

$$P(\psi) \iff \forall a_{a\in\mathbb{N}} \forall b_{b\in\mathbb{N}}(a < b \Longrightarrow \psi(a,b) = \bot)$$

е непрекъснато.

Задача 35. Нека R е рекурсивната програма:

$F(X,Y)$, `where`
$F(X,Y) =$ `if` $X \equiv 0 \pmod 2$ `then` $X/2$
`else` $F((X+1)/2 + Y, F(X-1, Y+1))$

Да се докаже, че

$$\forall a \forall b (!D_N(R)(a,b) \Longrightarrow D_N(R)(a,b) \leqq a+b).$$

Упътване. Разгледайте условията

$Q_1(\psi) \iff \forall a_{a\in\mathrm{N}} \forall b_{b\in\mathrm{N}_\perp} (\psi(a,b) \neq \perp \,\&\, a$ е четно $\Longrightarrow \psi(a,b) = a/2)$

$Q_2(\psi) \iff \forall a_{a\in\mathrm{N}} \forall b_{b\in\mathrm{N}} (\psi(a,b) \neq \perp \,\&\, a$ е нечетно $\Longrightarrow \psi(a,b) \leqq a+b)$

$Q_3(\psi) \iff \forall a_{a\in\mathrm{N}} (a$ е нечетно $\Longrightarrow \psi(a,\perp) = \perp)$

$Q_4(\psi) \iff \forall b_{b\in\mathrm{N}_\perp} (\psi(\perp,b) = \perp)$

Задача 36. а) За рекурсивната програма R_1

$F(X,\Lambda)$, `where`
$F(X,Y) =$ `if` $X=\Lambda$ `then` Y `else` $F(\mathrm{tail}(X), \mathrm{cons}(\mathrm{head}(X),Y))$

в типа List да се докаже, че $D_N(R_1) = D_V(R_2) = \mathrm{rev}$.

б) За рекурсивната програма R_2

$F(X,Y)$, `where`
$F(X,Y) =$ `if` $X=\Lambda$ `then` Y `else` $\mathrm{cons}(\mathrm{head}(X), F(\mathrm{tail}(X),Y))$

в типа List да се докаже, че $D_N(R_2) = D_V(R_2) = \mathrm{conc}$.

в) Нека Γ е изображение на $\mathcal{F}_2$ в $\mathcal{F}_2$, определено с равенството

$\Gamma(\psi)(a,b) \simeq$ `if` $a=\Lambda$ `then` $\mathrm{rev}(b)$
`else` $\mathrm{conc}(\psi(\mathrm{tail}(a),b), \mathrm{head}(a))$.

Да се докаже, че ако $\varphi \in \mathcal{F}_2$ е неподвижна точка на Γ, то

$$\forall a \forall b (!\varphi(a,b)).$$

г) Да се докаже, че функциите

$$\lambda x,y.\mathrm{rev}(\mathrm{conc}(x,y)) \text{ и } \lambda x,y.\mathrm{conc}(\mathrm{rev}(y),\mathrm{rev}(x))$$

от $\mathcal{F}_2$ са неподвижни точки на Γ.

д) Да се докаже, че

$$\forall a \forall b (\mathrm{rev}(\mathrm{conc}(a,b)) = \mathrm{conc}(\mathrm{rev}(b),\mathrm{rev}(a))).$$

Упътване. в) Използвайте индукция по структурата на данните.

г) За да покажете, че функцията $\lambda x,y.\mathrm{rev}(\mathrm{conc}(x,y))$ е неподвижна точка на Γ, използвайте свойството от зад. 10 б), § 4.1.

$$\mathrm{rev}(\mathrm{cons}(a,c)) = \mathrm{conc}(\mathrm{rev}(c),a),$$

за всяко $a \in L$, $c \in L^*$.

За функцията $\lambda x,y.\mathrm{conc}(\mathrm{rev}(y),\mathrm{rev}(x))$ използвайте зад. 19, т.е. че операцията конкатенация е асоциативна.

д) Съобразете, че Γ има единствена неподвижна точка в $\mathcal{F}_2$.

В следващите две задачи ще третираме думите в L^* като линейни списъци (крайни редици) от елементи на L. С $|a|$ ще означаваме дължината на списъка a, т.е. броя на елементите от L, участващи в a. Освен това разполагаме с две операции:

$\mathrm{left}(a)$ — лявата половина на списъка a, т.е. списък от първите $[|a|/2]$ елемента на a,

$\mathrm{right}(a)$ — дясната половина на списъка a, т.е. списък от останалите елементи на a, като $\mathrm{conc}(\mathrm{left}(a),\mathrm{right}(a)) = a$.

Предполагаме още, че в типа List е зададена наредба $\leqq$ между символите в L. С $s(a)$ ще означаваме факта, че списъкът a е сортиран в нарастващ ред. С $a \sim b$ означаваме, че елементите на a са точно елементите на b, наредени евентуално в друг ред.

Задача 37. Нека R_1 е рекурсивната програма в типа List:

$F(X,Y)$, `where`
$F(X,Y) =$ `if` $X=\Lambda$ `then` Y
`else if` $Y=\Lambda$
`then` X
`else if` $\mathrm{head}(X) \leqq \mathrm{head}(Y)$
`then` $\mathrm{cons}(\mathrm{head}(X), F(\mathrm{tail}(X),Y))$
`else` $\mathrm{cons}(\mathrm{head}(Y), F(X,\mathrm{tail}(Y)))$

Да се докаже, че програмата R_1 по два сортирани списъка a и b връща сортиран списък с елементите на a и b. Или, ако положим $\mathrm{merge} = \lambda a,b.D_V(R_1)(a,b)$, то

$$\forall a \forall b (s(a) \,\&\, s(b) \,\&\, \mathrm{merge}(a,b) \simeq c \Longrightarrow c \sim \mathrm{conc}(a,b) \,\&\, s(c))$$

и функцията merge е тотална.

Упътване. Разгледайте условията за частична коректност:

$$P_1(\psi) \iff \forall a \forall b (s(a) \,\&\, s(b) \,\&\, !\psi(a,b) \Longrightarrow s(\psi(a,b))),$$

$$P_2(\psi) \iff \forall a \forall b (!\psi(a,b) \Longrightarrow \psi(a,b) \sim \mathrm{conc}(a,b))$$

и приложете правилото на Скот.

За тоталността на функцията merge използвайте индукция по структурата на данните.

Задача 38. Нека R_2 е рекурсивната програма в типа List:

$$\begin{array}{l}
G(X), \quad \texttt{where} \\
F(X,Y) = \texttt{if } X = \Lambda \texttt{ then } Y \\
\qquad\qquad\qquad \texttt{else if } Y = \Lambda \\
\qquad\qquad\qquad\qquad \texttt{then } X \\
\qquad\qquad\qquad\qquad \texttt{else if } \mathrm{head}(X) \leqq \mathrm{head}(Y) \\
\qquad\qquad\qquad\qquad\qquad \texttt{then } \mathrm{cons}(\mathrm{head}(X), F(\mathrm{tail}(X), Y)) \\
\qquad\qquad\qquad\qquad\qquad \texttt{else } \mathrm{cons}(\mathrm{head}(Y), F(X, \mathrm{tail}(Y))) \\
G(X) = \texttt{if } |X| = 0 \vee |X| = 1 \texttt{ then } X \\
\qquad\qquad\qquad\qquad \texttt{else } F(G(\mathrm{left}(X)), G(\mathrm{right}(X))).
\end{array}$$

Да се докаже, че за всеки списък a програмата R_2 връща сортиран списъка a. Или, ако означим $\mathrm{sort}, = \lambda a.D_V(R_2)(a)$, функцията sort е навсякъде определена и

$$\forall a(\mathrm{sort}(a) \simeq b \Longrightarrow b \sim a \,\&\, s(b)).$$

Упътване. Разгледайте условието за частична коректност

$$Q(\varphi) \iff \forall a(!\varphi(a) \Longrightarrow s(\varphi(a)) \,\&\, a \sim \varphi(a))$$

и приложете правилото на Скот, като използвате предишната задача.

ЛИТЕРАТУРА

1. Е р ш о в, Ю. Л. Теория нумерации. Наука, М., 1977.
2. М а л ь ц е в, А. Алгоритмы и рекурсивные функции. Наука, М., 1986.
3. Л а в р о в, И., Л. М а к с и м о в а. Задачи по теории множеств, математической логике и теории алгоритмов. Наука, М., 1984.
4. С к о р д е в, Д. Комбинаторные пространства и рекурсивность в них. БАН, С., 1980.
5. С о с к о в, И., А. Д и ч е в. Теория на програмите. Ун. изд. „Св. Кл. Охридски", С., 1996.
6. A l a g i c, S., M. A r b i b. The Design of Well-structured and Correct Programs. Springer-Verlag, N. Y. Inc, 1978.
7. A n d e r s o n, R. Proving Programs Correct. John Wiley & Sons Ltd, N. Y., 1979.
8. C u t l a n d, N. Computability. An Introduction to Recursive Function Theory. Cambridge University Press, Cambridge, 1980.
9. D i j k s t r a, E., W. F e i j e n. A Method of Programming. Addison-Wesley, 1988.
10. L o e c k x, J., K. S i e b e r. The Fondations of Program Verification. John Wiley & Sons Ltd and B. G. Teubner, Stuttgart, 1984.
11. M a n n a, Z. Mathematical Theory of Computation. McGraw-Hill, N. Y., 1974. (Български превод: М а н н а, З. Математическа теория на информатиката. Наука и изкуство, С., 1983.)
12. M e n d e l s o n, E. Introduction to Mathematical Logic. D. Van Nostrand Company Inc, N. Y., 1976.
13. P h i l i p s, I. Recursion Theory. In: Handbook of Logic in Computer Science (S. Abramsky, D. Gabbay, T. Maibaum eds.), vol. 1. Clarendon Press, Oxford, 1992.
14. R o g e r s, H. Theory of Recursive Functions and Effective Computability. McGraw-Hill, N. Y., 1967.
15. S h o e n f i e l d, J. R. Mathematical Logic. Addison-Wesley, 1967.