Домашно № 1 по дисциплината "Дискретни структури" за специалност "Компютърни науки", I курс, II поток, зимен семестър на 2018/2019 уч. г. в СУ, ФМИ

Име: .... Факултетен № .... Група: .....

Задача	**1**	**2**	**3**	**4**	Общо
получени точки					
максимум точки	*25*	*35*	*30*	*10*	*100*

Забележка 1: *Всички отговори трябва да бъдат обосновани подробно.*

Забележка 2: *Не предавайте идентични решения дори когато работите заедно: идентичните решения ще бъдат анулирани!*

Задача 1. При хората има четири кръвни групи — A, B, AB и *0*. Кръв от човека X може да се прелее на човека Y само ако X е от нулевата група или Y е от групата AB, или X и Y са от една и съща група (включително ако са един и същи човек — т. нар. автотрансфузия).

В множеството на кръвните групи определяме бинарна релация ρ по следния начин:

$$x \, \rho \, y \iff \text{човек от групата } x \text{ може да дари кръв на човек от групата } y\,.$$

а) Докажете, че ρ е релация на наредба. **(6 точки)**

б) Представете ρ чрез таблица, множество, граф и диаграма на Хасе. **(8 точки)**

в) Линейна наредба ли е ρ? **(5 точки)**

г) В множеството на хората определяме бинарна релация R, както следва:

$$X \, R \, Y \iff \text{кръв от човека } X \text{ може да се прелее на човека } Y.$$

Релация на наредба ли е R ? **(6 точки)**

Задача 2.

а) Докажете, че множеството на всички алгоритми е изброимо. **(15 точки)**

б) Докажете, че множеството на всички алгоритмични задачи е неизброимо. **(15 точки)**

в) Какво следва от "а" и "б" ? **(5 точки)**

Задача 3. Нека $\mathbb{N}^+$ и $\mathbb{Q}^+$ са съответно множеството на целите положителни числа и множеството на положителните рационални числа.

Функцията $f : \mathbb{N}^+ \to \mathbb{Q}^+$ е дефинирана индуктивно:

$$f(1) = 1; \qquad f(2n) = f(n) + 1, \qquad f(2n+1) = \frac{1}{1 + f(n)} \text{ за всяко цяло } n \geq 1.$$

Докажете, че f е биекция.

Задача 4. Проверете еквивалентни ли са логическите изрази $(\neg q \vee p) \wedge (p \to q)$ и $p \leftrightarrow q$. Проверката да се извърши по два начина:

а) чрез табличния метод; **(5 точки)**

б) чрез еквивалентни преобразувания. **(5 точки)**

РЕШЕНИЯ

Задача 1.

а) Релацията ρ е релация на наредба, защото е рефлексивна, антисиметрична и транзитивна. Трите свойства се проверяват чрез разглеждане на всички възможности (пълно изчерпване).

б) Представяне на ρ чрез множество от наредени двойки:

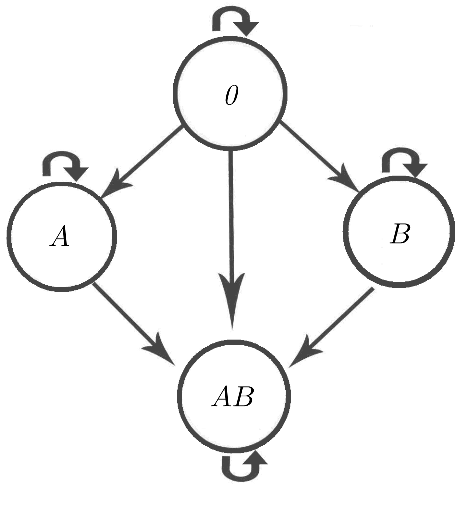
$$\Big\{ (0\,,\,0)\,,\ (0\,,\,A)\,,\ (0\,,\,B)\,,\ (0\,,\,AB)\,,\ (A\,,\,A)\,,\ (A\,,\,AB)\,,\ (B\,,\,B)\,,\ (B\,,\,AB)\,,\ (AB\,,\,AB) \Big\}.$$

Представяне на $x\,\rho\,y$ чрез таблица:

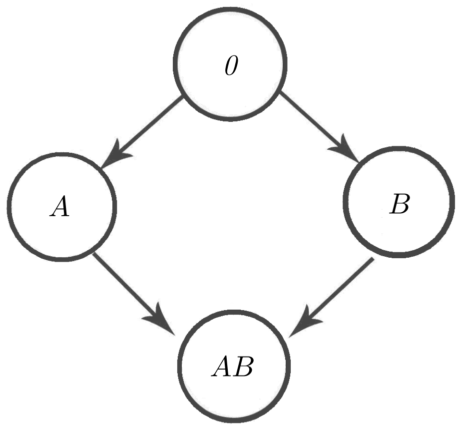
$x \backslash y$	0	A	B	AB
0	1	1	1	1
A	0	1	0	1
B	0	0	1	1
AB	0	0	0	1

x — кръводарител; y — получател.

Представяне на ρ чрез граф:



Представяне на ρ чрез диаграма на Хасе:



в) Релацията ρ не е линейна наредба, защото има несравними елементи. Такива са елементите A и B: нито $A\rho B$, нито $B\rho A$.

г) R не е релация на наредба, защото не е антисиметрична: от $X R Y$ и $Y R X$ не следва $X = Y$. Действително, ако може да се прелее кръв от човека X на човека Y и от Y на X, то следва само това, че X и Y са от една и съща кръвна група, но може да са различни хора: $X \neq Y$.

Задача 2.

а) Нека $\mathcal{A}$ е множеството на алгоритмите. Всеки алгоритъм може да бъде описан на някакъв език за програмиране, например на C. Нека $\mathcal{P}$ е множеството на всички програми на езика C. Всяка програма реализира точно един алгоритъм, а всеки алгоритъм може да бъде реализиран чрез поне една програма (а най-често — чрез повече от една). Следователно $|\mathcal{A}| \leq |\mathcal{P}|$. Тогава, за да докажем, че множеството $\mathcal{A}$ е изброимо, е достатъчно да установим, че $\mathcal{P}$ е изброимо.

Всяка програма е текст, т.е. крайна редица от знаци от някаква азбука, например ASCII. Нека $\mathcal{T}$ е множеството от текстовете над избраната азбука. Тъй като всяка програма е текст, но не всеки текст е програма, то $|\mathcal{P}| \leq |\mathcal{T}|$. Тогава, за да докажем, че $\mathcal{P}$ е изброимо множество, е достатъчно да установим, че множеството $\mathcal{T}$ е изброимо.

Ще докажем, че множеството $\mathcal{T}$ е изброимо, като подредим всичките му елементи в редица. За целта нареждаме текстовете по дължина — първо по-късите. Текстовете с еднаква дължина нареждаме по азбучен ред (според кодовете на знаците). Така първият член на редицата е празният текст (с дължина 0). Следващите 256 члена са всички текстове с дължина 1, тоест знаците от кода ASCII — знак № 0, знак № 1, ... , знак № 255. Следващите 256^2 члена са всички текстове с дължина 2 и т.н. За всяко цяло число $n \geq 0$ има 256^n текста с дължина n, значи, краен брой. Следователно всеки текст е предхождан (в смисъла на въведената наредба) от краен брой текстове, поради което рано или късно ще бъде срещнат в редицата. Изводът е, че така построената редица съдържа всички текстове. Следователно множеството $\mathcal{T}$ е изброимо, което трябваше да се докаже.

б) Нека $\mathcal{Z}$ е множеството на всички алгоритмични задачи. Ще докажем, че $\mathcal{Z}$ е неизброимо, като посочим негово подмножество $\mathcal{S}$, чиято неизброимост се установява по-лесно.

Нека $\mathbb{N}$ е множеството на естествените числа. За всяко подмножество X на $\mathbb{N}$ разглеждаме задача $z(X)$: "Да се провери дали дадено естествено число n принадлежи на множеството X." Това е алгоритмична задача: решението ѝ, ако има такова, представлява алгоритъм от вида `bool f(unsigned int n)`. Тялото на функцията `f` е различно за различни задачи. Например, ако X е множеството на четните числа, то $z(X)$ е задачата за разпознаване дали n е четно; ако X е множеството на простите числа, то $z(X)$ е задачата за разпознаване дали n е просто.

Нека $\mathcal{S} = \left\{ z(X) \,\middle|\, X \in 2^{\mathbb{N}} \right\}$ е множеството на всички алгоритмични задачи от този вид. Съответствието $z : 2^{\mathbb{N}} \to \mathcal{S}$ е биекция, затова $\mathcal{S}$ е равномощно на $2^{\mathbb{N}}$. Понеже $2^{\mathbb{N}}$ е неизброимо, то и $\mathcal{S}$ е неизброимо. А тъй като $\mathcal{Z} \supseteq \mathcal{S}$, то множеството $\mathcal{Z}$ също е неизброимо.

в) Щом алгоритмичните задачи са неизброимо много, а алгоритмите са изброимо количество, то не за всяка задача има алгоритъм, тоест съществуват алгоритмично нерешими задачи. Това разсъждение е неконструктивно: показва, че съществуват нерешими задачи, но не ни помага да съчиним такава задача. Все пак в теорията на алгоритмите са намерени конкретни задачи, за които е доказано, че са алгоритмично нерешими. Такава е например задачата за спирането: даден алгоритъм при дадени входни данни ще спре ли работа, или ще се изпълнява безкрай?

Този факт има практически последствия: не е възможно да се разбере с пълна сигурност дали работеща компютърна програма се е зациклила. Други важни задачи също са нерешими, например дали дадена програма ще влезе някога в дадено разклонение.

Споменатите задачи и други като тях са нерешими в общия случай. Отделни частни случаи може да имат решения.

Задача 3. Щом функцията f е дефинирана индуктивно, то и доказателството трябва да бъде индуктивно. Но индукция, имаща вида "от n към $2n$ и $2n+1$", не е особено удобна за работа. Затова ще използваме твърдение, което е равносилно на принципа на математическата индукция: "Всяко непразно множество от естествени числа има най-малък елемент."

Че стойностите на f са положителни рационални числа, е очевидно: тъй като започваме пресмятането на $f(n)$ от някакво цяло положително число n и използваме само операциите прибавяне на единица и взимане на реципрочна стойност, то както междинните стойности, така и крайният резултат ще бъдат все положителни рационални числа.

Щом $f(n) > 0$, то $f(2n) > 1$ и $f(2n+1) < 1$ за всяко цяло $n \geq 1$. Тоест стойността на f е по-голяма от 1 при четен аргумент и по-малка от 1 при нечетен (с изключение на $f(1) = 1$).

Да допуснем, че f не е инекция. Следователно съществуват различни числа k_1 и k_2 от $\mathbb{N}^+$, за които $f\left(k_1\right) = f\left(k_2\right)$. Нека k_1 е най-малкото такова число (тук използваме горния принцип). Общата стойност y на лявата и дясната страна е положително число. Възможни са три случая.

Първи случай: $y = 1$. Тогава $k_1 = 1$ и $k_2 = 1$, което противоречи на $k_1 \neq k_2$.

Втори случай: $y > 1$. Следователно k_1 и k_2 са четни положителни числа. С други думи, $k_1 = 2n_1$ и $k_2 = 2n_2$ за някакви цели положителни n_1 и n_2. Заместваме:

$$f\left(k_1\right) = f\left(k_2\right) \iff f\left(2n_1\right) = f\left(2n_2\right) \iff f\left(n_1\right) + 1 = f\left(n_2\right) + 1 \iff f\left(n_1\right) = f\left(n_2\right).$$

Понеже $n_1 = \dfrac{k_1}{2} \neq \dfrac{k_2}{2} = n_2$, то $n_1 \neq n_2$, тоест $\left(n_1, n_2\right)$ е нова двойка аргументи, която нарушава изискването от определението за инекция. От $n_1 = \dfrac{k_1}{2}$ и $k_1 > 0$ следва, че $n_1 < k_1$, а това противоречи на минималността на k_1.

Трети случай: $y < 1$. Тогава k_1 и k_2 са нечетни числа, по-големи от единица. Ето защо $k_1 = 2n_1 + 1$ и $k_2 = 2n_2 + 1$ за някакви цели положителни n_1 и n_2. Заместваме:

$$f\left(k_1\right) = f\left(k_2\right) \iff f\left(2n_1 + 1\right) = f\left(2n_2 + 1\right) \iff \frac{1}{1 + f\left(n_1\right)} = \frac{1}{1 + f\left(n_2\right)} \iff$$

$$\iff 1 + f\left(n_1\right) = 1 + f\left(n_2\right) \iff f\left(n_1\right) = f\left(n_2\right).$$

Понеже $n_1 = \dfrac{k_1 - 1}{2} \neq \dfrac{k_2 - 1}{2} = n_2$, то $n_1 \neq n_2$, тоест $\left(n_1, n_2\right)$ е нова двойка аргументи, нарушаваща изискването от определението за инекция. От $n_1 = \dfrac{k_1 - 1}{2}$ и $k_1 > 0$ следва, че $n_1 < k_1$, а това противоречи на минималността на k_1.

Щом във всички случаи стигаме до противоречие, то направеното допускане не е вярно. Следователно f е инекция.

Ще докажем, че f е сюрекция, като проверим, че всяко положително рационално число може да се получи от базата $f(1) = 1$ с помощта на краен брой операции прибавяне на единица и взимане на реципрочна стойност.

За целите числа това е очевидно: достатъчна е едната операция — прибавяне на единица.

За дробните числа има два случая — да са по-малки или по-големи от единица.

По-малките от единица се получават от по-големите чрез взимане на реципрочна стойност.

По-големите от единица дробни числа са сбор от цяла и дробна част. За цялата част знаем как се получава: тя е сбор от единици. Дробната част е по-малка от 1, тоест тя се свежда до предишния случай.

Двата случая се свеждат един до друг. Това може да изглежда като кръгово разсъждение, но в действителност е косвена рекурсия. Тя не е бездънна: при отделянето на цялата част числителят на дробта намалява, а при взимане на реципрочна стойност досегашният знаменател става числител и на свой ред намалява при следващото отделяне на цяла част. Не съществува безкрайна намаляваща редица от цели положителни числа, тоест все някога процесът ще спре: дробната част ще изчезне и ще остане цяло число, а този случай вече е решен.

Пример: Търсим цяло положително n, за което $f(n) = \dfrac{215}{62}\cdot$ Преработваме дробта:

$$\frac{215}{62} = 3+\frac{29}{62} = 3+\cfrac{1}{\cfrac{62}{29}} = 3+\cfrac{1}{2+\cfrac{4}{29}} = 3+\cfrac{1}{2+\cfrac{1}{\cfrac{29}{4}}} = 3+\cfrac{1}{2+\cfrac{1}{7+\cfrac{1}{4}}}\cdot$$

Полученият запис се нарича верижна дроб и има редица интересни свойства. Тук се налага да го преработим още малко — заместваме участващите цели числа със сборове от единици:

$$\frac{215}{62} = 1+1+1+\cfrac{1}{1+1+\cfrac{1}{1+1+1+1+1+1+1+\cfrac{1}{1+1+1+1}}}\cdot$$

Заместваме последната единица с $f(1)$:

$$\frac{215}{62} = 1+1+1+\cfrac{1}{1+1+\cfrac{1}{1+1+1+1+1+1+1+\cfrac{1}{1+1+1+f(1)}}}\cdot$$

Опростяваме получения израз отдолу нагоре с помощта на рекурентните формули за f:

$$\frac{215}{62} = 1+1+1+\cfrac{1}{1+1+\cfrac{1}{1+1+1+1+1+1+1+\cfrac{1}{1+1+f(2)}}};$$

$$\frac{215}{62} = 1+1+1+\cfrac{1}{1+1+\cfrac{1}{1+1+1+1+1+1+1+\cfrac{1}{1+f(4)}}};$$

$$\frac{215}{62} = 1+1+1+\cfrac{1}{1+1+\cfrac{1}{1+1+1+1+1+1+1+f(9)}};$$

$$\frac{215}{62} = 1+1+1+\cfrac{1}{1+1+\cfrac{1}{1+1+1+1+1+1+f(18)}};$$

$$\frac{215}{62} = 1+1+1+\cfrac{1}{1+1+\cfrac{1}{1+1+1+1+1+f(36)}};$$

$$\frac{215}{62} = 1+1+1+\cfrac{1}{1+1+\cfrac{1}{1+1+1+1+f(72)}}\,;$$

$$\frac{215}{62} = 1+1+1+\cfrac{1}{1+1+\cfrac{1}{1+1+1+f(144)}}\,;$$

$$\frac{215}{62} = 1+1+1+\cfrac{1}{1+1+\cfrac{1}{1+1+f(288)}}\,;$$

$$\frac{215}{62} = 1+1+1+\cfrac{1}{1+1+\cfrac{1}{1+f(576)}}\,;$$

$$\frac{215}{62} = 1+1+1+\frac{1}{1+1+f(1153)}\,;$$

$$\frac{215}{62} = 1+1+1+\frac{1}{1+f(2306)}\,;$$

$$\frac{215}{62} = 1+1+1+f(4613)\,;$$

$$\frac{215}{62} = 1+1+f(9226)\,;$$

$$\frac{215}{62} = 1+f(18452)\,;$$

$$\frac{215}{62} = f(36904)\,.$$

И така, решението на уравнението $f(n) = \dfrac{215}{62}$ е числото $n = 36904$.

Аналогично, за всяко рационално $q > 0$ съществува цяло $n > 0$, което удовлетворява уравнението $f(n) = q$. Затова функцията f е сюрекция.

Щом f е инекция и сюрекция, то тя е биекция.

Задача 4. Логическите изрази $(\neg q \vee p) \wedge (p \to q)$ и $p \leftrightarrow q$ са еквивалентни.

а) Проверка по табличния метод:

p	q	$\neg q$	$\neg q \vee p$	$p \to q$	$(\neg q \vee p) \wedge (p \to q)$	$p \leftrightarrow q$
0	0	1	1	1	1	1
0	1	0	0	1	0	0
1	0	1	1	0	0	0
1	1	0	1	1	1	1

Еквивалентността на двата логически израза следва от еднаквостта на последните два стълба.

б) Чрез еквивалентни преобразувания: Изразяваме операцията еквивалентност като конюнкция на две импликации, после изразяваме едната импликация чрез дизюнкция и отрицание:

$p \leftrightarrow q \equiv$

$(q \to p) \wedge (p \to q) \equiv$

$(\neg q \vee p) \wedge (p \to q)$.