VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
TIKIMYBIŲ TEORIJOS IR SKAIČIŲ TEORIJOS KATEDRA

Linas Klimavičius

Algebrinių skaičių multiplikatyvusis priklausomumas

Multiplicative dependence of algebraic numbers

Bakalauro baigiamasis darbas

Leidžiu ginti .....

Darbo vadovas doc. dr. Paulius Drungilas

Vilnius 2017

Turinys

Įvadas

Šiame darbe nagrinėjama algebrinių skaičių multiplikatyvumo problema. Algebrinis skaičius α vadinams priklausomu virš aibės $M \subset \mathbb{Q}$, jeigu skaičiai $\alpha + x_i, x_i \in M$ yra multiplikatyviai priklausomi, t. y. egzistuoja tokie skaičiai $x_1, x_2, \ldots, x_m$, $y_1, \ldots, y_n \in M$, kad

$$(\alpha + x_1)(\alpha + x_2)\ldots(\alpha + x_m) = (\alpha + y_1)(\alpha + y_2)\ldots(\alpha + y_n). \qquad (1)$$

Jeigu algebrinis skaičius α priklausomas virš aibės M, jam galima apibrėžti ilgį $l(\alpha, M)$ kaip mažiausią įmanomą (1) tenkinantį skaičių $m + n$. Šiame darbe nagrinėjamos mažų laipsnių paslinktųjų algebrinių skaičių multiplikatyvumo savybės.

Paslinktųjų sveikųjų ir racionaliųjų skaičių multiplikativusis priklausomumas yra paprasta ir išspresta problema. Jos paprastumas kyla iš to, kad racionaliuosius skaičius galima sandauga išsireikšti virš pirminių bazės ir yra žinoma, jog pirminių skaičių yra santykinai nedaug tarp sveikųjų skaičių, asimptotiškai jų yra $\approx N/log(N)$. Iš to seka, kad visiems $r \in \mathbb{Q}$ kažkokiam N skaičiai $r + 1, r + 2, \ldots, r + N$ turės mažiau nei N skirtingų pirminių daliklių ir bus multiplikatyviai priklausomi.

Šis argumentas neveikia aukštesnio nei 1 laipsnio algebriniams skaičiams. Nors skaičių kūno $\mathbb{Q}(\alpha)$ sveikųjų skaičių žiedo idealams galioja vienareikšmiško išskaidymo teorema, tačiau bendru atveju nėra pakankamų rezultatų apie pirminių idealų tankį šiuose žieduose.

Cassels [2] parodė, kad visiems algebrinimas iracionaliesiems skaičiams α pakankamai dideliems N bent pusė skaičių $\alpha + 1, \alpha + 2, \ldots, \alpha + N$ bus multiplikatyviai nepriklausomi. Šis teiginys rodo, kad daug skaičių nepriklausomų, tačiau nepasako, kada tarp jų bus priklausomų. Iki šiol nėra žinoma, ar egzistuoja toks α, kuriam visi skaičiai $\alpha + n, n \in \mathbb{Z}$ yra multiplikatyviai nepriklausomi.

Algebrinio skaičiaus multiplikatyvusis priklausomumas yra svarbus tiriant Ler-

cho dzeta funkcijos nulių pasiskirstymą ir universalumo savybę (žr. [4, 5]).

Drungilas ir Dubickas [3] įrodė, kad bet kuris kvadratinis algebrinis skaičius α yra miltiplikatyviai priklausomas virš $\mathbb{Z}_t = \{x \in \mathbb{Z} \mid x \geqslant t\}$ su bet kuriuo $t \in \mathbb{R}$ ir teisinga nelygybė $l(\alpha, \mathbb{Z}_t) \leqslant 8$. Be to, jie įrodė, kad kiekvienas kubinis algebrinis skaičius α yra multiplikatyviai priklausomas virš $\mathbb{Q}$ ir teisinga nelygybė $l(\alpha, \mathbb{Q}) <= 8$.

Šiame darbe įrodyta tokia teorema:

1 teorema. *Tarkime, kad a, c ir m – tokie sveikieji skaičiai, kad $c \neq 0$ ir*

$$|2am - 3m^2 - 1| > |c|(|a| + |c| + 1).$$

Tada daugianaris $x^3 + ax^2 + (2am - 3m^2 - 1)x + c$ yra neredukuojamas, o jo šaknis α yra multiplikatyviai priklausoma virš $\mathbb{Z}$ ir teisinga nelygybė $l(\alpha, \mathbb{Z}) \leqslant 4$.

Be to, įrodyta, kad iš algebrinio skaičiaus multiplikatyviojo priklausomumo išplaukia jo minimaliojo polinomo reikšmių aibės multiplikatyvusis priklausomumas.

1 skyrius

Sąsajos su kitomis problemomis

1.1 Prouhet–Tarry–Escott problema

Tegu $\mathbf{x} = (x_1, x_2 \ldots, x_d) \in \mathbb{Z}^d$. Pažymėkime $\sigma_1(\mathbf{x}), \sigma_2(\mathbf{x}), \ldots, \sigma_d(\mathbf{x})$ elementariąsias simetrines funkcijas: $\sigma_1(\mathbf{x}) = x_1 + x_2 + \cdots + x_d$, $\sigma_2(\mathbf{x}) = x_1x_2 + x_2x_3 + \cdots + x_{d-1}x_d, \ldots, \sigma_d(\mathbf{x}) = x_1x_2 \ldots x_d$. Suformuluosiu ir įrodysiu tokią teoremą:

2 teorema. *Tegu α – algebrinis skaičius, kuris nėra sveikasis algebrinis skaičius ir kurio minimalusis daugianaris yra $P(x) = a_1x^{d-1} + a_2x^{d-2} + \cdots + a_d$. Jei skaičiui α teisinga* (1) *lygybė, tai $l(\alpha, \mathbb{Z}) \geqslant 2d$, o lygybė galioja tada ir tik tada, kai egzistuoja tokie du skirtingi vektoriai $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^d$, kad $(\sigma_1(\boldsymbol{x}) - \sigma_1(\boldsymbol{y}) : \sigma_2(\boldsymbol{x}) - \sigma_2(\boldsymbol{y}) : \cdots : \sigma_n(\boldsymbol{x}) - \sigma_n(\boldsymbol{y})) = (a_1 : a_2 : \cdots : a_d)$*

Įrodymas. Iš pradžių įrodysiu, kad galiojant šios teoremos sąlygoms (1) lygybėje $m = n$. Tarkime priešingai, $m \neq n$. Neprarasdami bendrumo galime tarti, kad $m > n$, tai reikštų, kad α yra daugianario $Q(t) = (t + x_1)(t + x_2) \ldots (t + x_m) - (t + y_1)(t + y_2) \ldots (t + y_n)$ šaknis. Iš to seka, kad $Q(t)$ dalinasi iš $P(t)$, t.y. $\exists R(t) \in \mathbb{Z}[t]$, toks, kad $Q(t) = R(t) \cdot P(t)$. Sulyginę laipsnius prie vyriausiųjų koeficientų gauname, kad 1 dalinasi iš a, tačiau tai reikštų, kad α algebrinis sveikasis, prieštara. Taigi $m = n$. Tada daugianario $Q(t)$ narys t^n išsiprastina. Taigi $d - 1 = deg(P(t)) \leqslant deg(Q(t)) \leqslant n - 1$. Tada $l(\alpha, \mathbb{Z}) = 2n \geqslant 2d$. Jeigu galioja lygybė, tai $Q(\alpha) = (\alpha + x_1) \cdot (\alpha + x_2) \cdot \cdots \cdot (\alpha + x_d) - (\alpha + y_1) \cdot (\alpha + y_2) \cdot \cdots \cdot (\alpha + y_d) = (\sigma_1(x) - \sigma_1(y))\alpha^{d-1} + (\sigma_2(x) - \sigma_2(y))\alpha^{d-2} + \cdots + (\sigma_d(x) - \sigma_d(y)) = 0$. $Q(t)$ laipsnis sutampa su minimalaus α daugianario laipsniu, tai $Q(t) = C \cdot P(t)$ kažkokiai konstantai C. Taigi šių daugianarių koeficientai proporcingi, t.y., $(\sigma_1(\mathbf{x}) - \sigma_1(\mathbf{y}) : \sigma_2(\mathbf{x}) - \sigma_2(\mathbf{y}) :$

$\cdots : \sigma_n(\mathbf{x}) - \sigma_n(\mathbf{y})) = (a_1 : a_2 : \cdots : a_d)$, ką ir reikėjo įrodyti.

Prouhet–Tarry–Escott problema klausia, ar duotiems $n, k \in \mathbb{N}$, $n \geqslant k$ egzistuoja du skirtingi sveikųjų skaičių rinkiniai $\mathbf{a} = (a_1, a_2, \ldots, a_n), \mathbf{b} = (b_1, b_2, \ldots, b_n)$ tokie, kad

$$\sum_{j=1}^{n} a_j^i = \sum_{j=1}^{n} b_j^i,$$

$j = 1, 2, \ldots, k$ Pasinaudojant Vieto teorema, nesunku įsitikinti, kad, jeigu $n = k$, tai rinkiniai sutampa, todėl $n > k$. Sudėtingiausias klausimas lieka, kuriems n šis teiginys galioja kai $k = n - 1$. Atsakymas į šį klausimą yra teigiamas, kai $3 \leqslant n \leqslant 10$ ir kai $n = 12$, tačiau nėra žinomas jokiam kitam n. Kadangi elementariosios simetrinės fukcijos vienareikšmiškai nusakomos simetriniais Niutono daugianariais, $(\sigma_j(\mathbf{a}) = \sigma_j(\mathbf{b})$, $j = 1, 2, \ldots, n - 1$ Taigi taip suformuluotas klausimas yra ekvivalentus teiginiui

$$(\sigma_1(\mathbf{a}) - \sigma_1(\mathbf{b})) : (\sigma_2(\mathbf{a}) - \sigma_2(\mathbf{b})) : \cdots : (\sigma_n(\mathbf{a}) - \sigma_n(\mathbf{b})) = (0 : 0 : \cdots : 1).$$

1.2 Sąsajos su normos multiplikatyvumu virš racionaliųjų skaičių

Nagrinėkime skaičių kūną $\mathbb{Q}(\alpha)$. Tegul $\beta \in \mathbb{Q}$, o $Nm(\beta)$ - skaičiaus β norma plėtinyje $\mathbb{Q} \subset \mathbb{Q}(\alpha)$ (žr. [6]). Žinoma, kad norma $Nm : \mathbb{Q}(\alpha) \to \mathbb{Q}$ yra multiplikatyvi, t. y.,

$$Nm(\beta\gamma) = Nm(\beta)Nm(\gamma), \quad \beta, \gamma \in \mathbb{Q}(\alpha).$$

Tegu $P \in \mathbb{Q}[x]$ - algebrinio skaičiaus α minimalusis daugianaris, kurio koeficientas prie vyriausiojo nario lygus 1. Tegul $\alpha = \alpha_1, \alpha_2, \ldots, \alpha_n$ – visi α skirtingi algebriniai jungtiniai skaičiai. Tada $P(x) = (x - \alpha_1)(x - \alpha_2)\ldots(x - \alpha_d)$. Bet kuriam racionaliajam skaičiui r teisinga lygybė

$$\begin{aligned} Nm(\alpha + r) &= (\alpha_1 + r)(\alpha_2 + r)\ldots(\alpha_d + r) = \\ &= (-1)^d(-r - \alpha_1)(-r - \alpha_2)\ldots(-r - \alpha_d) = (-1)^d P(-r). \end{aligned}$$

Taigi suskaičiavę abiejų (1) lygybės pusių normą, gauname

$$(-1)^{md} P(-x_1)P(-x_2)\ldots P(-x_m) = (-1)^{nd} P(-y_1)P(-y_2)\ldots P(-y_n).$$

Pakėlę kvadratu, gauname

$$P(-x_1)^2 P(-x_2)^2 \ldots P(-x_m)^2 = P(-y_1)^2 P(-y_2)^2 \ldots P(-y_n)^2.$$

Vadinasi, jei algebrinis skaičius α yra multiplikatyviai priklausomas virš M, tai racionaliųjų skaičių aibė $\{P(-m) \mid m \in M\}$ yra multiplikatyviai priklausoma.

2 skyrius

Antro laipsnio algebriniai skaičiai

2.1 Kvadratinių agebrinių skaičių priklausomumas bendru atveju

Realiajam skaičiui t pažymėkime $\mathbb{Z}_t$ sveikuosius skaičius, didesnius už arba lygius t. Drungilas ir Dubickas [3] įrodė, kad visiems antrojo laipsni algebriniams skaičiams α egzistuoja toks pirminis p, kad bet kokiam realiajam t egzistuja be galo daug lygties

$$p(\alpha + x_1)(\alpha + x_2) = (\alpha + y_1)(\alpha + y_2)$$

sprendinių sveikaisiais skaičiais $x_1, x_2, y_1, y_2 > t$. Iš to seka, kad paėmę du tokius sprendinius (x_1, x_2, y_1, y_2) ir (y_3, y_4, x_3, x_4) ir sudauginę atitinkamas multiplikatyvaus priklausomumo lygybes, gauname

$$(\alpha + x_1)(\alpha + x_2)(\alpha + x_3)(\alpha + x_4) = (\alpha + y_1)(\alpha + y_2)(\alpha + y_3)(\alpha + y_4).$$

Vadinasi, visi antrojo laipsnio algebriniai skaičiai α yra multiplikatyviai priklausomi virš $\mathbb{Z}_t$. Be to,

$$l(\alpha, \mathbb{Z}_t) \leqslant 8. \tag{2.1}$$

Straipsnyje taip pat iškeliama hipotezė, kad ši nelygybė yra griežta, tai yra, egzistuoja toks antro laipsnio algebrinis skaičius α ir realusis t, kad $l(\alpha, \mathbb{Z}_t) = 8$ ir kaip galimą pavyzdį pateikia daugianarį $2\alpha^2 + 3$ ir teigia, kad atlikus skaičiavimus, programa rado, kad jokiems $x_1, x_2, x_3, y_1, y_2, y_3 \in \{1, 2, \ldots, 1000\}$ negalioja lygybė

$$(\alpha + x_1) \cdot (\alpha + x_2) \cdot (\alpha + x_3) = (\alpha + y_1) \cdot (\alpha + y_2) \cdot (\alpha + y_3).$$

Iš tiesų, šis teiginis nėra tesingas. Pavyzdžiui, tegul

$$\mathbf{x} := (x_1, x_2, x_3) = (3, 4, 46) \quad \text{ir} \quad \mathbf{y} := (y_1, y_2, y_3) = (2, 11, 24).$$

Tada

$$\begin{aligned}
&\sigma_1(\mathbf{x}) = 53,\\
&\sigma_2(\mathbf{x}) = 3 \cdot 4 + 4 \cdot 46 + 46 \cdot 3 = 334,\\
&\sigma_3(\mathbf{x}) = 3 \cdot 4 \cdot 46 = 552,\\
&\sigma_1(\mathbf{y}) = 37,\\
&\sigma_2(\mathbf{y}) = 2 \cdot 11 + 11 \cdot 24 + 24 \cdot 2 = 334,\\
&\sigma_3(\mathbf{y}) = 2 \cdot 11 \cdot 24 = 528.
\end{aligned}$$

$$\begin{aligned}
&(\alpha + x_1) \cdot (\alpha + x_2) \cdot (\alpha + x_3) - (\alpha + y_1) \cdot (\alpha + y_2) \cdot (\alpha + y_3) =\\
&= (\sigma_1(x) - \sigma_1(y))\alpha^2 + (\sigma_2(x) - \sigma_2(y))\alpha + (\sigma_3(x) - \sigma_3(y)) =\\
&= (53 - 37)\alpha^2 + (334 - 334)\alpha + (552 - 528) = 16\alpha^2 + 24 = 8(2\alpha^2 + 3) = 0.
\end{aligned}$$

Taigi, $l(\alpha, \mathbb{Z}_1) = 6$. Kita vertus, kaip bus matyti iš pateikiamų skaičiavimų, egzistuoja tokie daugianariai su mažais koeficientais, kad jei α yra tokio daugianario šaknis ir jai teisinga (1) lygybė su $n = m = 3$, tai $\max(x_1, x_2, x_3, y_1, y_2, y_3) \geqslant 200$.

2.2 Skaičiavimai, padedantys ieškoti algebrinių skaičių ilgius

Sukūrėme *python* programą (kodas pateikiamas šio darbo priede), skirtą kvadratinių algebrinių skaičių, kurių minimalusis polinomas turi pavidalą $ax^2 + c$, ir kuriems teisinga lygybė $l(\alpha, \mathbb{Z}_0) = 8$, paieškai. Tuo tikslu nagrinėjami trejetai $\mathbf{x} = (x_1, x_2, x_3)$ ir $\mathbf{y} = (y_1, y_2, y_3)$, tenkinantys sąlygą

$$(\sigma_1(\mathbf{x}) - \sigma_1(\mathbf{y})) : (\sigma_2(\mathbf{x}) - \sigma_2(\mathbf{y})) : (\sigma_3(\mathbf{x}) - \sigma_3(\mathbf{y})) = (a : 0 : c).$$

Atliekant skaičiavimus laikyta, kad $1 \leqslant x_1, x_2, x_3, y_1, y_2, y_3 \leqslant n$, kur n – programoje kontroliuojamas parametras. Paprasčiausias algoritmas perrinkti visus trejetus yra labai neefektyvus, nes jam įgyvendinti prireiktų $O(n^6)$ operacijų, todėl jį reikia patobulinti. Tam praverčia sąlyga $\sigma_2(\mathbf{x}) = \sigma_2(\mathbf{y})$. Ją galima perrašyti taip:

$$(y_1 + y_2) \cdot (y_1 + y_3) = x_1x_2 + x_2x_3 + x_3x_1 + y_1^2.$$

Tada užtenka perrinkti tik visus skaičius x_1, x_2, x_3, y_1 ir, nagrinėjant skaičiaus

$$x_1x_2 + x_2x_3 + x_3x_1 + y_1^2$$

daliklius, rasti įmanomas $(\sigma_1(\mathbf{x}) - \sigma_1(\mathbf{y})) : (\sigma_2(\mathbf{x}) - \sigma_2(\mathbf{y}))$ reikšmes.

Atlikti skaičiavimai parodė, kad jei α yra algebrinis skaičius, kurio minimalusis polinomas yra $4x^2 + 1$ arba $10x^2 + 1$, ir jam teisinga (1) lygybė su $m = n = 3$, tai bent vienas iš skaičių $x_1, x_2, x_3, y_1, y_2, y_3$ yra didesnis už 200. Taigi tikėtina, kad šiems algebriniams skaičiams (2.1) nelygybė yra griežta.

3 skyrius

Aukštesnio laipsnio algebriniai skaičiai

3.1 Teorema apie trečiojo laipsnio sveikųjų algebrinių skaičių ilgį

Straipsnyje [3] taip pat nagrinėjamas aukštesnio laipsnio algebrinių skaičius multiplikatyvusis priklausomumas. Įrodyta, kad kiekvienas kubinis algebrinis skaičius α yra multiplikatyviai priklausomas virš $\mathbb{Q}$ ir teisinga nelygybė $l(\alpha, \mathbb{Q}) <= 8$. Šiame darbe pavyko įrodyti, kad tam tikri kubiniai sveikieji algebriniai skaičiai yra multiplikatyviai priklausomi virš $\mathbb{Z}$.

3 teorema. *Tarkime, kad a, c ir m – tokie sveikieji skaičiai, kad $c \neq 0$ ir*

$$|2am - 3m^2 - 1| > |c|(|a| + |c| + 1).$$

Tada daugianaris $x^3 + ax^2 + (2am - 3m^2 - 1)x + c$ yra neredukuojamas, o jo šaknis α yra multiplikatyviai priklausoma virš $\mathbb{Z}$ ir teisinga nelygybė $l(\alpha, \mathbb{Z}) \leqslant 4$.

Įrodymas. Nesunku įsitikinti, kad teisinga tokia lygybė:

$$\alpha^3 + a\alpha^2 + (2am - 3m^2 - 1)\alpha + c =$$
$$= (\alpha + m)^3 + (a - 3m)(\alpha + m)^2 - (\alpha + m) + c - am^2 + 2m^3 + m.$$

Pažymėję $r = c - am^2 + 2m^3$ ir pasinaudoję tuom, kad šis reiškinys lygus 0, gauname, kad

$$(\alpha + m)^2(\alpha + a - 2m) = (\alpha + m) - r = \alpha + (m - r).$$

Vadinasi, $l(\alpha, \mathbb{Z}) \leqslant 4$. Lieka įrodyti, kad daugianaris

$$P = x^3 + ax^2 + (2am - 3m^2 - 1)x + c$$

yra neredukuojamas virš sveikųjų skaičių. Iš tiesų, jeigu jis būtų redukuojamas, tai, kadangi jo laipsnis 3, bent viena jo komponenčių būtų tiesinė ir daugianaris turėtų sveikąją šaknį, kurią pažymėkime x_0. Tai reikštų, kad $x_0|c$ ir $|x_0| \leqslant |c|$. Tada

$$\begin{aligned} |2am - 3m^2 - 1| = |-x_0^2 - ax_0 - c/x_0| \leqslant \\ \leqslant |x_0|^2 + |ax_0| + |c/x_0| \leqslant \\ \leqslant |c|^2 + |ac| + |c| = |c|(|a| + |c| + 1), \end{aligned}$$

o tai prieštarauja teoremos prielaidai. □

4 išvada. *Bet kuriems sveikiesiems skaičiams a ir $c \neq 0$ egzistuoja be galo daug tokių sveikųjų skaičių b, kad daugianaris $x^3 + ax^2 + bx + c$ yra neredukuojamas, o jo šaknis α yra multiplikatyviai priklausoma virš $\mathbb{Z}$ ir teisinga nelygybė $l(\alpha, \mathbb{Z}) \leqslant 4$.*

Multiplicative dependence of algebraic numbers

Summary

A length of an algebraic number α over a set $M \subset \mathbb{Q}$ is defined as the minimum number $m + n$ such that there exists two distinct collections $x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_n \in M$ satisfying

$$(\alpha + x_1) \cdot (\alpha + x_2) \ldots (\alpha + x_m) = (\alpha + y_1) \cdot (\alpha + y_2) \ldots (\alpha + y_n),$$

provided that such collections exist. The analysis of certain second and third degree algebraic numbers is provided as well as calculations that suggest bound $l(\alpha, \mathbb{Z}_0) \leqslant 8$ for quadratic algebraic numbers is sharp.

Literatūra

[1] P. Borwein, C. Ingalls *The Prouhet–Tarry-Escott problem revisited*, Enseign. Math. 40 (2) (1994), 177-184.

[2] J.W.S. Cassels, *Footnote to a note of Davenport and Heilbronn,* J. London Math. Soc. 36 (1961), 177–184.

[3] P. Drungilas, A. Dubickas, *Multiplicative dependence of shifted algebraic numbers,* Col. Math. 96 (1) (2003), 75-81.

[4] A. Laurinčikas, *Limit theorems for the Riemann zeta–function,* Kluwer, Dordrecht, 1996.

[5] A. Laurinčikas, K. Matsumoto *The joint universality and the functional independence of Lerch zeta–functions,* 157 (2000), 221-227.

[6] M. R. Murty, J. Esmonde, *Problems in Algebraic Number Theory*, 2nd ed., Springer-Verlag New York, 2005.

Priedai

python 3 programa antro laipsnio algebrinių skaičių, kuriems

$$l(\alpha, \{1, 2, \ldots, n\}) = 8.$$

Programa išspausdina trupmenas $\frac{a}{b}$, kurioms galioja, kad daugianario $ax^2 + c$ šaknis α tenkina minėtą sąlygą.

```
from fractions import Fraction as Fr


def get_primes(n):
    numbers = set(range(n, 1, -1))
    primes = []
    while numbers:
        p = numbers.pop()
        primes.append(p)
        numbers.difference_update(set(range(p*2, n+1, p)))
    return primes


def factorize(n, primes):
    factors = []
    for p in primes:
        if p*p > n:
                break
        i = 0
        while n % p == 0:
```

```
            n //= p
            i += 1
        if i > 0:
            factors.append((p, i))
    if n > 1:
        factors.append((n, 1))
    return factors


def sigma2(x):
    return x[0]*x[1] + x[1]*x[2] + x[2]*x[0]


def rel_fraction(x, y):
    den = x[0]*x[1]*x[2]-y[0]*y[1]*y[2]
    if den != 0:
        return Fr(sum(x) - sum(y), den)
    else:
        return None


def triplet_generator(n):
    for x in range(1, n):
        for y in range(1, n):
            for z in range(y, n):
                yield (x, y, z)


def get_simple_fractions():
    res = set()
    for i in range(-10, 11):
        for j in range(-10, 11):
            if j != 0:
                res.add(Fr(i, j))
    return res
```

```
def get_divisors_by_num(n):
    res = {}
    primes = get_primes(n)
    for i in range(1, n+1):
        res[i] = factorize(i, primes)
    return res


def main():
    n = 200
    print('Generated fractions:')
    divisors_by_num = get_divisors_by_num(4*n*n + 3)
    all_fractions = set()
    for x in triplet_generator(n):
        for y in range(1, int(round((sigma2(x)/3.0)**.5))+2):
            n2 = sigma2(x) + y**2
            divisors = map(lambda r: r[0], divisors_by_num[n2])
            divisors = filter(lambda r: r > y and r*r <= n2, divisors)
# y = min(x_1, x_2, x_3, y_1, y_2, y_3)
            for d in divisors:
                y2 = d - y
                y3 = n2//d-y
                rel_f = rel_fraction(x, (y, y2, y3))
                if rel_f and rel_f not in all_fractions:
                    all_fractions.add(rel_f)
                    print('%s %s, %s' % (rel_f, x, (y,y2,y3)))
    print('not generated fractions:\n')
    simple_fractions = get_simple_fractions()  # paprastomis trupmenomis l
    for fr in simple_fractions:
        if fr not in all_fractions:
            print('%s' % fr)

if __name__ == '__main__':
    main()
```