CHALMERS | GÖTEBORGS UNIVERSITET

Hur vet vi att ett datorprogram gör vad det säger att det gör?

Formell verifiering av hypergeometriska rekursionsrelationer med polynomkoefficienter

How do we know that a computer program does what it says it does?

Formal verification of hypergeometric recurrence relations with polynomial coefficients

Examensarbete för kandidatexamen i matematik vid Göteborgs universitet
Kandidatarbete inom civilingenjörsutbildningen vid Chalmers

Louis Dao
Anna Davoodi
Erik Samuelsson

Institutionen för Matematiska vetenskaper
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2024

Hur vet vi att ett datorprogram gör vad det säger att det gör?

Formell verifiering av hypergeometriska rekursionsrelationer med polynomkoefficienter

Examensarbete för kandidatexamen i matematik inom Matematikprogrammet vid Göteborgs universitet
Erik Samuelsson

Examensarbete för kandidatexamen i matematik inom Matematikprogrammet, inriktning Tillämpad matematik, vid Göteborgs universitet
Louis Dao

Kandidatarbete i matematik inom civilingenjörsprogrammet Teknisk matematik vid Chalmers
Anna Davoodi

Handledare: Dennis Eriksson
Adam Keilthy

Institutionen för Matematiska vetenskaper
CHALMERS TEKNISKA HÖGSKOLA
GÖTEBORGS UNIVERSITET
Göteborg, Sverige 2024

Förord

Detta kandidatarbete är skrivet vid Instutitionen för Matematiska vetenskaper vid Chalmers Tekniska Högskola och Göteborgs Universitet. Vi vill rikta ett tack till våra handledare Adam Keilthy och Dennis Eriksson för den kontinuerliga handledningen samt värdefulla insikter.

Under arbetets gång har en logg i form av en dagbok samt en tidslogg förts där vi varje vecka dokumenterat händelseförloppet för den veckan, eventuella problem som uppstått samt vad varje person har bidragit med.

Nedan följer en uppdelning på författare till varje avsnitt. Vi vill poängtera att detta arbete har varit en gruppinsats där alla har varit lika delaktiga under hela processen. Observera att revideringen även har varit en gruppinsats, där personerna som inte är skrivna som författare har hjälpt till med revidering.

§	**Rubrik**	**Författare**
	Populärvetenskaplig presentation	Anna, Erik
	Sammandrag	Erik
	Abstract	Anna
1	Inledning	Anna, Louis
1.1	Problemformulering och syfte	Anna, Erik, Louis
1.2	Avgränsningar	Erik
1.3	Samhälleliga och etiska aspekter	Anna
2	Teori	Louis
2.1	Definitioner	Anna
2.2	Satser	Louis
3	Resultat och Diskussion	Anna
3.1	Fakultetfallet	Anna, Erik
3.1.1	Preliminära resultat	Anna, Louis
3.1.2	Lösning av Problem 2	*Se den detaljerade tabellen*
3.1.3	Lösning av Problem 3	Erik
3.2	Flervariabla fakultetfallet	Louis
3.3	Linjära fallet	Erik
3.4	Förslag till fortsatta riktningar	Erik, Louis
A	Appendix - teori	Louis
B	Appendix - källkod	Anna
C	Appendix - figurer	Erik

På grund av mängden material i Avsnitt 3.1.2 ger vi en detaljerad uppdelning av avsnittet i den följande tabellen.

Område i Avsnitt 3.1.2	**Författare**
Introducerande text	Erik
Formulering och bevis av Sats 3.3	Erik
Figurer	Anna
Formulering och bevis av Lemma 3.4	Anna
Formulering och bevis av Lemma 3.5	Louis
Formulering och bevis av Sats 3.6	Louis
Kod 1	Anna
Slutdiskussion	Anna, Louis

Populärvetenskaplig presentation

Ordet *algoritm* förekommer överallt idag. En algoritm kan i enkla termer beskrivas som utförandet av en uppgift givet viss information, likt ett recept för matlagning. Den givna informationen, *indatan*, kan betraktas som ingredienserna och receptbeskrivningen som stegen algoritmen tar. Resultatet av en algoritm kallas för *utdatan* och kan jämföras med den färdiga maträtten som fås efter att ha följt receptet.

Efter att fullföljt receptets instruktioner finns det en förväntan om att maträtten kommer smaka bra. Detsamma gäller för algoritmer. Det finns ett antagande att algoritmer ska bete sig på ett visst sätt. Resultatet ska ha en viss förutbestämd typ eller form. Fokuset för detta kandidatarbete är att avgöra om utdatan är ett heltal eller inte.

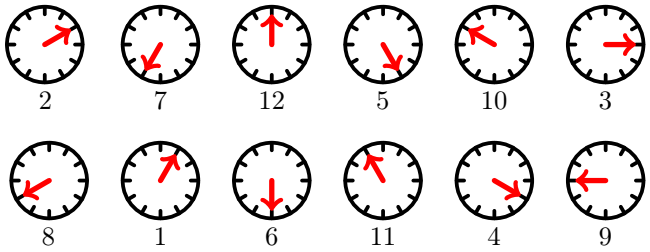
Ett exempel på en sådan algoritm kan vara en hiss som åker till olika våningsplan. Antag att hissen är på marknivå och att någon trycker på flera knappar. Om personen först trycker på knapp 4 och sedan på knapp 3 vet vi att hissen kommer att åka till dessa plan. Vi vet också att den här hissens algoritm tar 4 och 3 som indata samt ger ut 3 och 4 som utdata. Alltså kommer hissen åka till plan 3 först och sedan plan 4. I det här fallet gick allt bra, algoritmen gjorde inget oväntat. Skulle däremot hissen använda en algoritm som inte kontrollerats riskerar den att ge ut 3.5 som utdata. I så fall skulle hissen behöva åka till ”plan 3.5”, ett plan som inte existerar. För att garantera att hissens algoritm alltid ger ut heltal kan vi använda *formel verifiering*. Detta är en samlingsterm för olika matematiska verktyg som används för att kontrollera algoritmers beteende.

I vissa fall går det snabbt att verifiera att en algoritm gör som den ska, likt exemplet med hissen. Däremot finns det många algoritmer som är mer komplicerade att verifiera. Något som kan göra en algoritm svårare att verifiera är ifall den används flera gånger med olika indata. I så fall kan man inte direkt se om den kommer fungera som den ska. Arbetet kommer att fokusera på den här sortens algoritmer. Exemplet nedan beskriver hur en sådan algoritm kan användas.

Algoritmer används ofta i spelutveckling. För att få ett spel som känns intressant kanske man vill ha olika saker som är oförutsägbara. Det kanske rör sig om vilken riktning en fiende attackerar i, vilken färg det är på en fågel eller hur eld ska röra sig i vinden. I praktiken handlar detta om tillgången till slumptal. Alltså tal som inte tycks ha något mönster mellan varandra. Problemet blir att lyckas implementera detta på ett effektivt sätt.

För att lyckas få slumptal i spelet kan man använda flera olika metoder. En metod är att lagra en lång lista med slumptal på spelets minne. Problemen med denna metod är att listan kan ta slut och att den tar upp värdefullt minne. Vi vill hitta en metod som inte medför dessa problem. En bättre lösning är att använda en algoritm som använder sin utdata som indata! Fördelen med detta är att nästa utdata beror på den tidigare utdatan, alltså kan man starta algoritmen och så ger den ut massa tal. Detta betyder att man med en smart algoritm kan få ut mycket stor variation. En till fördel är att algoritmen bara kräver en bestämd mängd minne, oavsett hur många slumptal vi vill ha.

För att ge ett konkret exempel, säg att vi vill ha en klocka i ett TV-spel som styrs av en algoritm. Vi kommer ställa två krav på algoritmen. Dels att klockan visar en slumpmässig tid varje gång vi kollar på den. Samt att algoritmen ger oss alla tider någon gång. Här är det senare kravet viktigt för att vi inte ska få kanske bara två olika tider. Vi ska nu implementera detta i en algoritm och sedan verifiera att den gör vad vi förväntar oss. Hur kan vi implementera detta med en algoritm? Säg att klockans timvisare står på 2 första gången vi tittar på den. Algoritmen vi använder gör så att timvisaren flyttas 5 timmar framåt varje gång vi tittar på klockan. Alltså kommer klockan nästa gång att visa 7 för att sedan visa 12.



Figur 1: De olika klockorna som vår algoritm producerar. Notera att efter timvisaren stått på 12 så går den fem timmar fram och pekar då på siffran 5. Notera även att efter timvisaren pekat på 9 så repeterar vi samma klockor igen och igen.

I Figur 1 ser vi alla olika sätt klockan kommer se ut och vilken siffra timvisaren pekar på. I det här exemplet gick allt bra. Vår algoritm gör det vi förväntar oss. Den ger ut klockor i en ordning som ser lite oförutsägbar ut. Samt att den gav oss alla 12 möjliga klockor vilket vi ser i Figur 1 ovan. En sak som är värd att poängtera är valet av hopp. Tänk om vi tog 6 timmars hopp istället för 5. Då hade timvisaren pekat på 2, sedan på 8 och sedan tillbaks till 2. Alltså skulle vi inte få alla tider. Som tur är så lyckades vi välja hopp så att inga tider missades.

Detta exempel visar hur vi kan ställa krav på vår algoritm och sedan verifiera att den gör som den ska. Det finns många fler krav vi hade kunnat ställa och verifiera i exemplet ovan. Att timvisaren alltid kommer peka på ett heltal är självklart i exemplet ovan. Det vi kommer göra är att studera algoritmer som ger ut tal där nästa tal beror på det tidigare och avgöra om det blir heltal. Problemet är att ibland är det inte lika självklart som i exemplet ovan.

Sammandrag

Detta arbete använder formell verifiering för att undersöka loopar inom programmering. Looparna som behandlas ses ekvivalent som hypergeometriska rekursionsrelationer. Arbetet undersöker om lösningen u_n till dessa hypergeometriska rekursionsrelationer med polynomkoefficienter alltid är heltalsvärd. Detta genomförs under antagandet att startvärdet för rekursionsrelationen alltid är 1. För koefficienter som är vissa typer av linjära polynom ges villkor som avgör om u_n alltid är ett heltal. Även mer involverade typer av polynom undersöks, där en explicit algoritm ges som avgör om u_n alltid är ett heltal efter ändligt antal steg. Resultaten för det sistnämnda fallet generaliseras även till flera variabler.

Abstract

This thesis employs formal verification to examine loops in programming, treating them as equivalent to hypergeometric recurrence relations. It investigates whether the solution u_n to some hypergeometric recurrence relations with polynomial coefficients is integral, assuming the initial value of the recursion is always 1. Conditions which ensure that u_n is always integral for some linear polynomials are given. Additionally, more complex types of polynomials are explored, with an explicit algorithm provided to determine if u_n is always an integer after a finite number of steps. The results for the latter case are also generalized to multiple variables.

Innehåll

1 Inledning

Bedömningen av korrektheten hos ett datorprogram brukar genomföras på två olika sätt. En av dessa metoder är *validering*. Där evalueras ett programs korrekthet genom att utvärdera dess beteende, exempelvis genom simuleringar av programmet. Den andra metoden är *verifiering* där korrekthet avgörs med hjälp av rigorösa processer. I detta fall skapas en matematisk modell av programmet, sedan utvärderas modellen med hjälp av matematiska bevis som visar på det ursprungliga programmets funktion och korrekthet [7].

Formell verifiering (jrf. eng. "Formal verification") är en process där matematiska och logiska metoder används för att verifiera att programmen som konstrueras fungerar efter valda specifikationer. Trots att metoden är generellt tillämpbar inom en mängd olika områden har den historiskt varit mer relevant inom hårdvaruutveckling. Ett exempel på detta är att kontrollera elektriska nätverks funktioner med hjälp av matematiska formler [7]. Ett annat exempel är att komplettera under utvecklingen och implementationen av olika avancerade datorprocessorer. I sådana fall jämförs processorns beteende mot specifikationer givna i processorns instruktionsarkitektur. På senare tid tillämpas formell verifiering även på mjukvarusystem. Detta är en direkt effekt av att mjukvara blivit allt mer genomgripande inom såväl näringsliv som för privat bruk. Det är alltså av ytterst vikt att säkerställa korrektheten hos dessa mjukvaror för att kunna garantera deras tillförlitlighet [12].

Arbetets fokus är formell verifiering av enkla datorprogram, benämnda som loopar. Loopar är iterativa datorprogram med ändamålet att upprepa en uppsättning operationer ett specificerat antal gånger. Dessa exemplifieras med hjälp av följande kod:

```
u = 0, v = 2, w = 1
for i = 1,2,...,k:
    u = u + 1
    v = i*v+1
    w = 4*w
```

Matematiskt kan dessa loopar beskrivas som *rekursionsrelationer* vilket är ekvationer där ett värde i en sekvens beskrivs som ett uttryck av tidigare värden från samma sekvens. Kodexemplet ovan kan matematiskt beskrivas som följande rekursionsrelationer

$$\begin{cases} u_0 = 0, \ v_0 = 2, \ w_0 = 1 \\ u_{n+1} = u_n + 1 \\ v_{n+1} = nv_n + 1 \\ w_{n+1} = 4w_n \end{cases}, \quad n = 0, 1, \ldots, k-1.$$

Rekursionsrelationerna som kommer att undersökas är *hypergeometriska* och definieras av följande ekvation:

$$f(n)u_n = g(n)u_{n-1}, \ n \geq 1, \ \text{startvärde } u_0. \tag{1}$$

Koefficienterna f och g i hypergeometriska rekursionsrelationer är polynom. Fokuset för detta arbete kommer endast tillägnas polynom med positiva heltalskoefficienter där $u_0 = 1$. I kombination med att polynomen utvärderas i heltal medför detta att $u_n \in \mathbb{Q}_+$. Notera att med positiva heltal syftar vi till mängden $\mathbb{N} = \{1, 2, 3, \ldots\}$.

I detta arbete kommer vi undersöka i vilken mån som några olika hypergeometriska rekursionsrelationer på formen (1) kommer att enbart producera sekvenser av positiva heltal. Detta kallar vi för heltalsvärd. En mer precis definition ges i Definition 1 på sida 4.

De hypergeometriska rekursionsrelationerna som vi undersöker uppstår inte enbart vid studier av loopar utan dyker även upp inom flera olika matematiska områden. Ett sådant område är

primtalsräkning (jrf. eng. "prime-counting"). Om $\pi(x)$ är antalet primtal mindre än eller lika med x kommer

$$c_1 \frac{x}{\log x} \leq \pi(x) \leq c_2 \frac{x}{\log x}$$

för stora x och konstanter c_1, c_2. Ett sätt att visa att $c_1 \approx 0.92$ och $c_2 \approx 1.11$ fungerar är genom att visa att

$$U(n) = \frac{(30n)!n!}{(15n)!(10n)!(6n)!}$$

enbart antar heltal [2]. Dessa bråk av fakulteter går att betrakta som en rekursionsrelation med polynomkoefficienter, vilka kommer studeras mer generellt i arbetet.

Ett annat område där hypergeometriska rekursionsrelationer framträder inom matematiken är lösningar av hypergeometriska differentialekvationer som kan tillämpas i olika fysikaliska problem. Hypergeometriska differentialekvationer kan se ut på följande sätt

$$z(1-z)\frac{d^2w(z)}{dz^2} + [c - (a+b-1)z]\frac{dw(z)}{dz} - abw(z) = 0, \quad a, b, c \in \mathbb{C}.$$

Genom att ansätta $w = \sum_{n=0}^{\infty} u_n z^n$ kommer problemet att reduceras till den hypergeometriska rekursionsrelationen på formen

$$u_{n+1} = \frac{(n+a)(n+b)}{(n+1)(n+c)} u_n, \text{ se [14].}$$

Koefficienterna u_n i en potensserie är avgörande för att bestämma egenskaperna och beteendet hos serien. Genom att studera dessa koefficienter fås värdefulla insikter om lösningens egenskaper, som dess asymptotiska beteende eller hur den konvergerar. Här medför heltalskoefficienter att potensserien antingen är ett polynom eller har en konvergensradie mindre än ett [1].

I syfte att upprätthålla transparens vill författarna notera att en artikel som innehåller en komplett lösning av Problem 2 på sida 3 anträffades i ett tidigt skede av arbetet. Detta rapporterades till gruppens handledare, där ett gemensamt beslut togs att inte ta del av denna lösning i syfte att upprätthålla den akademiska integriteten. Istället utforskades och utvecklades egna bevis, och problemet löstes på egen hand. Efter att ha löst problemet jämfördes lösningen med artikelns lösning. Sammantaget är det samma villkor som ställts upp, men härledningarna och bevisföringen kring villkoren är annorlunda. Se Landaus "Sur les conditions de divisibilité d'un produit de factorielles par un autre" för hela artikeln med denna lösning [9].

1.1 Problemformulering

Linjära fallet

Vi kommer undersöka följande typ av rekursionsrelation

$$(n+c)u_n = (an+b)u_{n-1}, \;\; u_0 = 1 \tag{2}$$

för konstanter $a, b, c \in \mathbb{N}$. Specifikt kommer vi att undersöka vilka krav som dessa konstanter bör uppfylla för att rekursionen ska vara heltalsvärd, alltså att $u_n \in \mathbb{N}$ för alla $n \in \mathbb{N}$ (se Definition 1 på sida 4).

Problem 1. Ge en algoritm som avgör om rekursionsrelationen (2) är heltalsvärd i avseendet av Definition 1. Algoritmen måste avslutas efter ändligt många steg.

Fakultetfallet

Låt $\boldsymbol{a} \in \mathbb{N}^k$, $\boldsymbol{b} \in \mathbb{N}^\ell$ för $k, \ell \in \mathbb{N}$ vara givna och låt u_n vara lösningen till rekursionsrelationen definierad av

$$\left[\prod_{i=1}^{\ell}(b_i n)(b_i n - 1)\cdots(b_i n - b_i + 1)\right] u_n = \left[\prod_{i=1}^{k}(a_i n)(a_i n - 1)\cdots(a_i n - a_i + 1)\right] u_{n-1}, \tag{3}$$

för alla $n \in \mathbb{N}$, där vi antar begynnelsevillkoret $u_0 = 1$. Att detta kallas för fakultetfallet motiveras av att u_n kan skrivas som ett bråk av fakulteter, se ekvation (5) på sida 5. För att avgöra om denna rekursionsrelation inte är heltalsvärd så räcker det att hitta ett n sådant att $u_n \notin \mathbb{N}$. Däremot krävs det att varje $n \in \mathbb{N}$ ska kontrolleras för att visa att rekursionsrelationen är heltalsvärd. För att möjliggöra kontrollen kommer vi ta fram en algoritm som avgör om (3) är heltalsvärd och som avslutas efter ändligt många steg.

Problem 2. Ge en algoritm som avgör om rekursionsrelationen (3) är heltalsvärd i avseendet av Definition 1. Algoritmen måste avslutas efter ändligt många steg.

En ytterligare fråga är ifall det existerar en övre gräns för hur många n som måste kontrolleras för att kunna avgöra ifall (3) är heltalsvärd eller inte. Det är även viktigt att gränsen är effektivt beräkningsbar, alltså att den går att explicit ta fram för en given rekursion.

Problem 3. Låt u_n var definierad som i (3). Visa existensen av ett effektivt beräkningsbart $N \in \mathbb{N}$ sådant att $u_n \in \mathbb{N}$ för alla $n \in \mathbb{N}$ om $u_n \in \mathbb{N}$ för alla $n \leq N$.

Flervariabla fakultetfallet

Vi är intresserade av att utöka den slutna formen för fakultetfallet, se (5) på sida 5 till flera variabler. Detta görs så att u_n blir ett bråk av fakulteter i flera variabler, se (14) på sida 13. Från detta får vi följande rekursionsrelation

$$\left[\prod_{i=1}^{\ell}\prod_{r=0}^{b_{ij}-1}(b_{ij}m_j - r + \Lambda_{bj})\right]u_{\mathbf{m}} = \left[\prod_{i=1}^{k}\prod_{r=0}^{a_{ij}-1}(a_{ij}m_j - r + \Lambda_{aj})\right]u_{\mathbf{m}-\mathbf{e}_j} \tag{4}$$

med koefficienter $\boldsymbol{a}_j \in \mathbb{N}^k$, $\boldsymbol{b}_j \in \mathbb{N}^\ell$ för $k, \ell \in \mathbb{N}$ där

$$u_{\mathbf{m}} = u(m_1, m_2, \ldots, m_n),\ m_j \in \mathbb{N} \cup \{0\},$$
$$\Lambda_{aj} = \sum_{\substack{1 \leq J \leq n \\ J \neq j}} a_{iJ}m_J,\ \Lambda_{bj} = \sum_{\substack{1 \leq J \leq n \\ J \neq j}} b_{iJ}m_J,$$

och $\mathbf{e}_j$ är den j-te basvektorn i $\mathbb{R}^n, j = 1, 2, ..., n$. Notera att $u_{\mathbf{m}}$ är väldefinierat för $\mathbf{m} \neq \mathbf{0}$. Här antas begynnelsevillkoret $u_{\mathbf{0}} = 1$.

Problem 4. Givet koefficienter $\boldsymbol{a}_j \in \mathbb{N}^k, \boldsymbol{b}_j \in \mathbb{N}^\ell, j = 1, 2, ..., n$, ge en algoritm som avgör om den flervariabla rekursionrelationen (4) är heltalsvärd. Med heltalsvärd gäller den intuitiva utökningen utifrån Definition 1 det vill säga $u_{\mathbf{m}} \in \mathbb{N}\ \forall \mathbf{m} \neq \mathbf{0}$. Algoritmen måste avslutas efter ändligt många steg.

1.2 Avgränsningar

I det linjära fallet avgränsas arbetet till att undersöka rekursionsrelationen $(dn + c)u_n = (an + b)u_{n-1}$ med $d = 1$. Detta motiveras av komplexiteten som ett generellt d tillförde. På grund av tidsbrist detta fall därmed inte generaliserats till $d \neq 1$. Arbetet avgränsas även till att visa existensen av en algoritm som löser Problem 1, och inte ge en explicit algoritm. Däremot kan villkoren från Sats 3.13 på sida 18 implementeras i en algoritm.

I Problem 3 ges en explicit övre gräns för N, men en förbättring av denna gräns avstås från att undersökas. Vidare undersöks aldrig om N är av rätt storleksordning eller om det går att lyckas bättre asymptotiskt i avseende på koefficienterna.

En rekursionsrelation behöver ges tillsammans med begynnelsevillkor för att vara väldefinierad. Genomgående för hela arbetet kommer det antas att begynnelsevillkoren alltid är lika med 1. Om en rekursionsrelation är heltalsvärd då begynnelsevillkoret är 1 kommer den även vara heltalsvärd för alla begynnelsevillkor som är positiva heltal, men inte nödvändigtvis omvänt.

1.3 Samhälleliga och etiska aspekter

Algoritmer är en integrerad del av olika sammanhang och påverkar således en stor mängd människor. Många viktiga processer är beroende av algoritmer, speciellt att de fungerar enligt valda specifikationer. Några exempel på sådana processer är objektigenkänning på kamerorna hos självkörande bilar [10], optimering av poliskontroller [8] samt beräkningen av urvalet och ordningen av resultaten som visas vid en googling [3]. Objektigenkänning hos självkörande bilar påverkar trafiksäkerheten om medtrafikanter inte klassificeras korrekt. Att optimera poliskontroller till områden med mycket tidigare aktivitet riskerar att oskyldiga vistandes i dessa områden kan bli felaktigt kontrollerade. Sökresultatens ordning kan påverka tillgången till information och spridningen av olika perspektiv. Dessa exempel betonar behovet att diskutera samhälleliga och etiska aspekter vid någon form av arbete med eller studier av algoritmers beteenden.

En viktig aspekt av arbetet är kontrollen av korrektheten av resultaten från en algoritm. Givet exemplen som gavs i tidigare stycke finns det ofta en förväntan på resultatet från algoritmer ty deras samhälleliga betydelse. Att formellt verifiera resultat och därför garantera korrektheten av algoritmer är därför gynnsamt för samhället.

Ett problem med algoritmer är resurshantering. Som med många andra beräkningar kan även algoritmer behöva kräva betydande datorkraft för att köras [13]. Att effektivisera kontrollen om huruvida de rekursionsrelationer som undersöks är heltalsvärda eller ej medför därför en fördelaktig effekt för miljöpåverkan genom att minska mängden resurser som används.

Gällande arbetetsprocessen har detta utförts utan någon involvering av känslig data. Sådan data har varken använts, lagrats eller producerats. Processen har till stor del bestått av matematiska härledningar och bevisförande av rekursionsrelationer. Därför är samhälleliga och etiska aspekter inte relevanta att diskutera för själva arbetsprocessen.

2 Teori

I detta avsnitt är alla viktiga definitioner och satser som används inom arbetet samlade.

2.1 Definitioner

Definition 1. Den hypergeometriska rekursionsrelationen $f(n)u_n = g(n)u_{n-1}$, $n \in \mathbb{N}$ sägs vara *heltalsvärd* om det för alla $n \in \mathbb{N}$ gäller att $u_n \in \mathbb{N}$ för $u_0 = 1$.

Definition 2. Givet en hypergeometrisk rekursionsrelation u_n på formen från Definition 1, låt den *slutna formen*, $U(n)$ av u_n ges av

$$U(n) := \frac{g(n)g(n-1)\cdots g(1)}{f(n)f(n-1)\cdots f(1)}.$$

Definition 3. *Golvfunktionen* $\lfloor x \rfloor$ för ett givet $x \in \mathbb{R}$ ges av det största heltalet $m \leq x$.

Definition 4. Låt $n \in \mathbb{Z} \setminus \{0\}$ och låt p vara ett primtal. Den *p-adiska valueringen* av n, $\nu_p(n)$ är det talet sådant att $n = p^{\nu_p(n)}m$ för något $m \in \mathbb{N}$ där $p \nmid m$. Enligt konvention definieras $\nu_p(0) = \infty$. Denna definition kan utökas från $\mathbb{N}$ till $\mathbb{Q}$ via $\nu_p(\frac{a}{b}) := \nu_p(a) - \nu_p(b)$, för $a, b \in \mathbb{N}$.

2.2 Satser

Stirlings olikhet. *För stora $n \in \mathbb{N}$ ges Stirlings approximation av*

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n.$$

En utveckling av approximationen ger att för alla $n \geq 1$ så gäller

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\left(\frac{1}{12n+1}\right)} < n! < \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\left(\frac{1}{12n}\right)}.$$

Bevis. Se [6].

Dirichlets sats om aritmetiska talföljder. *För två relativt prima tal $a, b \in \mathbb{N}$ gäller det att sekvensen*

$$an + b,\ n \geq 0$$

kommer generera oändligt många primtal.

Bevis. Se [5].

Legendres formel. *Legendres formel säger att*

$$\nu_p(n!) = \sum_{i=1}^{\infty} \left\lfloor \frac{n}{p^i} \right\rfloor.$$

Bevis. Se [11].

Bertrands postulat. *För $n > 3$ så måste det existera minst ett primtal p mellan n och $2n - 2$. Detta är ekvivalent med att om $n > 1$ så måste det existera minst ett primtal q mellan n och $2n$.*

Bevis. Se [4].

3 Resultat och Diskussion

I detta avsnitt presenteras resultaten först för fakultetfallet, sedan för det flervariabla fakultetfallet och till sist för det linjära fallet. Ordningen av denna presentation motiveras av att resultaten för det sistnämnda fallet är mer komplexa och involverade än resultaten för de förstnämnda. Detta kan vid första anblick anses vara ointuitivt då formen på rekursionen från det linjära fallet nog är den som ser enklast ut.

3.1 Fakultetfallet

Vi undersöker huruvida det finns en algoritm som uppfyller kraven i Problem 2 och ger en övre gräns för hur stora n-värden som behöver kontrolleras, vilket löser Problem 3. Först formulerar vi den slutna formen för (3). Sedan konstaterar vi några preliminära resultat om nödvändiga villkor kring sambandet mellan koefficienterna i täljaren och nämnaren för rekursionsrelationens slutna form.

Utvecklar vi rekursionsrelationen kommer följande samband att erhållas:

$$u_n = \frac{\left[\prod_{i=1}^{k}(a_i n)(a_i n - 1)\cdots(a_i n - a_i + 1)\right]}{\left[\prod_{i=1}^{\ell}(b_i n)(b_i n - 1)\cdots(b_i n - b_i + 1)\right]} u_{n-1} = \cdots = \frac{(a_1 n)!\cdots(a_k n)!}{(b_1 n)!\cdots(b_\ell n)!} u_0.$$

Den slutna formen för (3) ges då av

$$\frac{(a_1 n)!\cdots(a_k n)!}{(b_1 n)!\cdots(b_\ell n)!} = U(n). \tag{5}$$

Hädanefter, när ett samband mellan täljar- och nämnarkoefficienterna nämns så refererar vi till koefficienterna $a_1, \ldots, a_k$ och $b_1, \ldots, b_\ell$ från (5).

Då vi antar begynnelsevillkoret $u_0 = 1$ gäller det att en rekursionsrelation är heltalsvärd om och endast om dess slutna form är heltalsvärd. Vi kommer därför vidare att fokusera på $U(n)$ när vi ska avgöra om (3) är heltalsvärd eller ej. Alltså om

$$U(n) \in \mathbb{N}, \forall n \in \mathbb{N}.$$

3.1.1 Preliminära resultat

Ett resultat som beskriver ett nödvändigt villkor för sambandet mellan koefficienterna för den slutna formen (5) ges i följande proposition. Denna proposition är användbar och används i bevisen av flera resultat längre ner.

Proposition 3.1. För a_i och b_i som i Problem 2 gäller

$$\sum_{i=1}^{k} a_i < \sum_{i=1}^{\ell} b_i \Longrightarrow \exists n \in \mathbb{N} : U(n) \notin \mathbb{N}.$$

Bevis. Vi antar att $\sum_{i=1}^{k} a_i < \sum_{i=1}^{\ell} b_i$ och visa att det existerar ett n sådant att $U(n)$ inte är heltalsvärd. Vi tillämpar Stirlings olikhet på täljaren i (5) och får

$$\prod_{i=1}^{k} \sqrt{2\pi a_i n}\left(\frac{a_i n}{e}\right)^{a_i n} e^{\left(\frac{1}{12a_i n+1}\right)} < (a_1 n)! \cdots (a_k n)! < \prod_{i=1}^{k} \sqrt{2\pi a_i n}\left(\frac{a_i n}{e}\right)^{a_i n} e^{\left(\frac{1}{12a_i n}\right)}.$$

En liknande tillämpning utförs på nämnaren och vi får följande olikhet

$$U(n) < \frac{\prod_{i=1}^{k} \sqrt{2\pi a_i n}\left(\frac{a_i n}{e}\right)^{a_i n} e^{\left(\frac{1}{12a_i n}\right)}}{\prod_{i=1}^{\ell} \sqrt{2\pi b_i n}\left(\frac{b_i n}{e}\right)^{b_i n} e^{\left(\frac{1}{12b_i n+1}\right)}}.$$

Vi bryter ut alla n^n faktorer då dessa växer snabbast för stora n vilket resulterar i

$$\frac{\prod_{i=1}^{k} \sqrt{2\pi a_i n}\left(\frac{a_i n}{e}\right)^{a_i n} e^{\left(\frac{1}{12a_i n}\right)}}{\prod_{i=1}^{\ell} \sqrt{2\pi b_i n}\left(\frac{b_i n}{e}\right)^{b_i n} e^{\left(\frac{1}{12b_i n+1}\right)}} = n^{n\left(\sum_{i=1}^{k} a_i - \sum_{i=1}^{\ell} b_i\right)} \frac{\prod_{i=1}^{k} \sqrt{2\pi a_i n}\left(\frac{a_i}{e}\right)^{a_i n} e^{\left(\frac{1}{12a_i n}\right)}}{\prod_{i=1}^{\ell} \sqrt{2\pi b_i n}\left(\frac{b_i}{e}\right)^{b_i n} e^{\left(\frac{1}{12b_i n+1}\right)}}.$$

Vi får slutligen termen n^{-Cn} där $C := \sum_{i=1}^{\ell} b_i - \sum_{i=1}^{k} a_i > 0$. Detta medför att högerledet går mot 0 för stora n men $U(n) \neq 0$, så vid $n >> 1$ får vi att $U(n) \notin \mathbb{N}$ för något n.

■

En annan nödvändig egenskap för sambandet mellan täljar- och nämnarkoefficienterna ges av följande proposition.

Proposition 3.2. För a_i och b_i som i Problem 2 gäller

$$U(n) \in \mathbb{N} \ \forall n \in \mathbb{N} \Longrightarrow \exists a_i \geq \max\{b_1, \ldots, b_\ell\}.$$

Bevis. Vi bevisar detta med hjälp av ett kontrapositivt argument, det vill säga

$$\nexists a_i \geq \max\{b_1, \ldots, b_\ell\} \Longrightarrow U(n) \notin \mathbb{N} \text{ för något } n \in \mathbb{N}.$$

Notera att

$$\nexists a_i \geq \max\{b_1, \ldots, b_\ell\} \Longleftrightarrow \max\{a_1, \ldots, a_k\} < \max\{b_1, \ldots, b_\ell\}.$$

Låt $M_a := \max\{a_1, \ldots, a_k\}$ och $M_b := \max\{b_1, \ldots, b_\ell\}$. För $n \in \mathbb{N}$ gäller att $M_a n < M_b n$. Vi är intresserade av om det kommer gå att hitta ett primtal p där $M_a n < p \leq M_b n$. Skulle ett sådant primtal existera kommer inte faktorerna i täljaren ta ut faktorerna i nämnaren varpå $U(n) \notin \mathbb{N}$.

Vi kommer nu visa att

$$\exists n : M_b n - 1 = p.$$

Vi har

$$M_b n - 1 = M_b n - 1 + M_b - M_b = M_b(n-1) + (M_b - 1).$$

Notera att $M_b,\ M_b - 1 > 0$ samt att $\mathrm{SGD}(M_b, M_b - 1) = 1$. Från Dirichlets sats om aritmetiska talföljder följer alltså direkt att det finns oändligt många primtal p som kan produceras av sekvensen $M_b(n-1) + (M_b - 1) = M_b n - 1$. Observera att

$$M_b n - 1 = p > M_a n \Longrightarrow (M_b - M_a)n > 1 \Longrightarrow n > 1 \text{ ty } (M_b - M_a) \geq 1.$$

Detta krav på n är dock inte inskränkande ty det fortfarande kan produceras oändligt många primtal från sekvensen $M_b n - 1$. ■

Sammantaget har vi från dessa två propositioner en möjlighet att avstå från att undersöka (5) för många olika kombinationer av koefficienter som inte uppfyller de angivna kraven från propositionerna. För att visa detta ges följande exempel. Den slutna formen

$$\frac{(3n)!(6n)!}{(5n)!(5n)!}$$

kommer enligt Proposition 3.1 inte vara heltalsvärd då nämnarens koefficientsumma är strikt större än täljarens. Samtidigt skulle

$$\frac{(3n)!(5n)!}{(n)!(6n)!}$$

inte vara heltalsvärd enligt Proposition 3.2 då $5 < 6$. Däremot kan dessa propositioner inte direkt hantera fallet

$$\frac{(3n)!(32n)!}{(8n)!(25n)!} \qquad (6) \qquad \text{eller} \qquad \frac{(14n)!(6n)!}{(11n)!(7n)!}. \qquad (7)$$

I nuläget har vi inga krav som kan avgöra om de slutna formerna i exemplen ovan är heltalsvärda eller ej. Om vi med hjälp av dator kontrollerar rekursionsrelationerna från dessa exempel upp till $n = 20$ verkar båda vara heltalsvärda. Detta är emellertid inte en precis eller hållbar kontroll. Krav som kan hantera dessa koefficientkombinationer kommer istället ges i nästa avsnitt. Vi återkommer till dessa exempel efter att ha presenterat kraven.

3.1.2 Lösning av problem 2

För att lösa Problem 2 måste vi kunna avgöra om rekursionsrelationen med den slutna formen given i (5) är heltalsvärd. För närvarande är den enda metoden för att avgöra detta att undersöka bråkets värde för varje $n \in \mathbb{N}$. För att kunna få en bättre förståelse vill vi omformulera problemet med hjälp av p-adisk valuering. Det finns två viktiga egenskaper med $\nu_p(n)$ som vi kommer använda. Den första är att $\nu_p(nm) = \nu_p(n) + \nu_p(m)$ för något $n, m \in \mathbb{N}$, som direkt följer från definitionen. Den andra egenskapen är att $\frac{n}{m} \in \mathbb{N}$ om och endast om $\nu_p(n) \geq \nu_p(m)$ för alla primtal p. En intuitiv förståelse för den senare egenskapen kan uppnås genom att tänka på $\nu_p(n)$ som antalet gånger primtalet p förekommer i n. Alltså är $\frac{n}{m}$ ett heltal om och endast om primtalet p förekommer fler gånger i täljaren än i nämnaren. Tillämpar vi nu ν_p på (5) får vi att $U(n) \in \mathbb{N}$ för alla $n \in \mathbb{N}$ om och endast om

$$\nu_p((a_1 n)! \cdots (a_k n)!) \geq \nu_p((b_1 n)! \cdots (b_\ell n)!), \forall n \in \mathbb{N}, \forall p \text{ primtal}.$$

Eftersom $\nu_p(nm) = \nu_p(n) + \nu_p(m)$ är detta i sin tur ekvivalent med att

$$\sum_{i=1}^{k} \nu_p((a_i n)!) \geq \sum_{i=1}^{\ell} \nu_p((b_i n)!), \forall n \in \mathbb{N}, \forall p \text{ primtal}.$$

Använder vi Legendres formel får vi att detta är ekvivalent med

$$\sum_{i=1}^{k} \sum_{j \geq 1} \left\lfloor \frac{a_i n}{p^j} \right\rfloor \geq \sum_{i=1}^{\ell} \sum_{j \geq 1} \left\lfloor \frac{b_i n}{p^j} \right\rfloor, \forall n \in \mathbb{N}, \forall p \text{ primtal}.$$

Vi kan alltså sammanfatta denna omskrivningen som att

$$U(n) \in \mathbb{N}, \forall n \in \mathbb{N} \iff \sum_{i=1}^{k} \sum_{j \geq 1} \left\lfloor \frac{a_i n}{p^j} \right\rfloor \geq \sum_{i=1}^{\ell} \sum_{j \geq 1} \left\lfloor \frac{b_i n}{p^j} \right\rfloor, \forall n \in \mathbb{N}, \forall p \text{ primtal}. \qquad (8)$$

Denna ekvivalenta formulering av problemet kommer vi använda i beviset av nästa sats där vi visar att $U(n) \in \mathbb{N}$ för alla $n \in \mathbb{N}$ om och endast om en viss reellvärd funktion är positiv. Vi kommer också vilja definiera $S_j(x, p)$ som

$$S_j(x, p) := \sum_{i=1}^{k} \left\lfloor \frac{a_i x}{p^j} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_i x}{p^j} \right\rfloor, \ j \in \mathbb{N},$$

för primtal p och reella tal x. Denna funktion kommer vara viktig i beviset av den kommande satsen och i lösningen av Problem 2.

Sats 3.3. *Det gäller att*

$$U(n) \in \mathbb{N}\,\forall n \in \mathbb{N} \iff \sum_{i=1}^{k} \left\lfloor \frac{a_i x}{2} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_i x}{2} \right\rfloor \geq 0\,\forall x \in [0,2].$$

Bevis. Låt oss börja med "$\Longleftarrow$". Vi kommer nu visa att högerledet i (8) gäller. För att kunna göra detta måste vi skriva om det på följande ekvivalenta form

$$\sum_{j\geq 1}\left(\sum_{i=1}^{k} \left\lfloor \frac{a_i n}{p^j} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_i n}{p^j} \right\rfloor\right) \geq 0\,\forall n \in \mathbb{N}, \forall p \text{ primtal}. \tag{9}$$

Att uttrycken är ekvivalenta följer av att summan över $j \geq 1$ är ändlig för varje givet n. Undersökningen av differensen utvidgas till att även inkludera reella x. Härnäst härleds tre egenskaper för $S_j(x,p)$. Det gäller att

$$S_j\left(x+p^j,p\right) = \sum_{i=1}^{k}\left(\left\lfloor \frac{a_i x}{p^j} \right\rfloor + a_i\right) - \sum_{i=1}^{\ell}\left(\left\lfloor \frac{b_i x}{p^j} \right\rfloor + b_i\right) = S_j(x,p) + C, \tag{10}$$

där vi använde oss av att $\lfloor x+a \rfloor = \lfloor x \rfloor + a$ då $a \in \mathbb{Z}$. Alltså har vi nu visat att $S_j(x,p)$ är kvasiperiodisk (jfr. eng. "quasiperiodic") i x med period p^j i den mån att det tillkommer en konstant förskjutning $C := \sum a_i - \sum b_i$ vid varje period. Observera att $C \geq 0$ ty $C = S_1(2,2) \geq 0$ enligt antagandet. Resterande två egenskaper beskriver beteendet hos $S_j(x,p)$ för olika värden på j och p. För samma p men två olika j_1, j_2 får vi att

$$S_{j_2}(x,p) = \sum_{i=1}^{k} \left\lfloor \frac{a_i x}{p^{j_2}} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_i x}{p^{j_2}} \right\rfloor = \sum_{i=1}^{k} \left\lfloor \frac{a_i\left(\frac{p^{j_1}}{p^{j_2}}x\right)}{p^{j_1}} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_i\left(\frac{p^{j_1}}{p^{j_2}}x\right)}{p^{j_1}} \right\rfloor = S_{j_1}\left(\frac{p^{j_1}}{p^{j_2}}x, p\right), \tag{11}$$

som resulterar i en skalning i x-led. För samma j men två olika primtal p, q gäller det att

$$S_j(x,q) = \sum_{i=1}^{k} \left\lfloor \frac{a_i x}{q^j} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_i x}{q^j} \right\rfloor = \sum_{i=1}^{k} \left\lfloor \frac{a_i\left(\frac{p^j}{q^j}x\right)}{p^j} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_i\left(\frac{p^j}{q^j}x\right)}{p^j} \right\rfloor = S_j\left(\frac{p^j}{q^j}x, p\right), \tag{12}$$

som också resulterar i en skalning i x-led. Detta innebär att $S_j(x,p)$ endast är en skalad variant av $S_1(x,2)$. Enligt antagande är $S_1(x,2) \geq 0$ för alla $x \in [0,2]$. Nu ger (10) samt $C \geq 0$ att $S_1(x,2) \geq 0$ för alla $x \in \mathbb{R}_+$. Samtidigt gäller det från (11) och (12) att $S_j(x,p) \geq 0$ för alla $x \in \mathbb{R}_+$, $j \in \mathbb{N}$ och primtal p. Alltså gäller

$$\sum_{j\geq 1} S_j(x,p) \geq 0 \implies (9),$$

vänsterledet ovan gäller för alla $x \in \mathbb{R}_+$ och därmed även för alla $n \in \mathbb{N}$ så vi är klara med den första implikationen.

Låt oss nu visa "$\Longrightarrow$" med hjälp av ett kontrapositivt argument. Vi vill visa att

$$\exists x \in [0,2] : \sum_{i=1}^{k} \left\lfloor \frac{a_i x}{2} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_i x}{2} \right\rfloor < 0 \implies \exists n \in \mathbb{N} : U(n) \notin \mathbb{N}.$$

Detta är enligt (8) ekvivalent med att

$$\exists x \in [0,2] : S_1(x,2) < 0 \implies \exists n \in \mathbb{N}, \exists p \text{ primtal} : \sum_{j\geq 1}\left(\sum_{i=1}^{k} \left\lfloor \frac{a_i n}{p^j} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_i n}{p^j} \right\rfloor\right) < 0. \tag{13}$$

Notera att om $S_1(2,2) < 0$ så kommer Proposition 3.1 ge att $U(n)$ inte är heltalsvärd. Fortsättningsvis antas därför att $S_1(2,2) \geq 0$. Vi kommer visa (13) genom att först visa att det existerar ett halvöppet intervall I där $S_1(x,2) < 0$. Vi vet att det existerar ett $x_0 \in [0,2)$ sådant att $S_1(x_0,2) < 0$. Funktionen $S_1(x,2)$ är en ändlig summa av skalade golvfunktioner, alltså kommer $S_1(x,2)$ vara diskontinuerlig i ändligt många punkter i $[0,2)$. Det måste därför existera ett $\varepsilon > 0$ sådant att $S_1(x,2)$ är kontinuerlig i $[x_0, x_0+\varepsilon)$ förutom möjligvis i punkten x_0. Vidare kommer $S_1(x,2)$ att vara konstant i detta intervall eftersom $S_1(x,2)$ är högerkontinuerlig. Alltså uppfyller intervallet $I = [x_0, x_0+\varepsilon)$ de önskade kraven. Från första delen av beviset så har vi sett att olika p-värden skalar om funktionen $S_1(x,p)$. Enligt (12) kommer därför $I \cdot \frac{p}{2}$ att vara ett intervall sådant att $S_1(x,p) < 0$ för alla $x \in I \cdot \frac{p}{2}$.

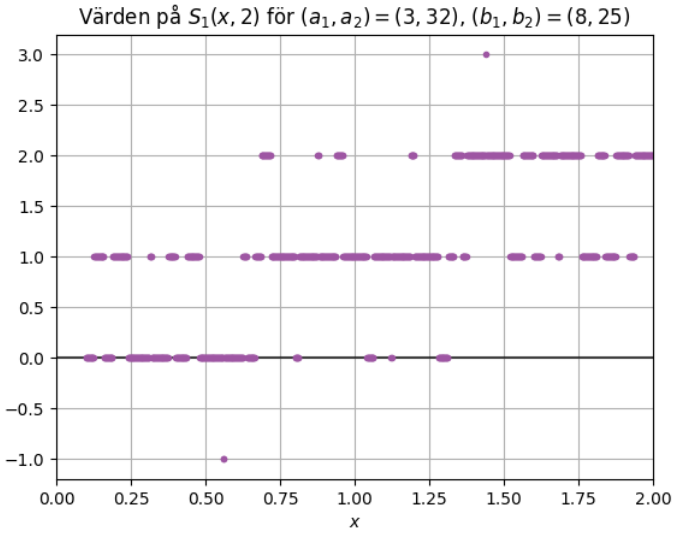
Härnäst vill vi visa högerledet i (13), alltså att summan av $S_j(n,p)$ över $j \geq 1$ är strikt negativ för något primtal p och heltal n. Detta verkställs genom att visa att ett primtal q kan väljas sådant att $S_j(x,q) = 0$ för alla $j \geq 2$ och alla $x \in I \cdot \frac{q}{2}$. Vi visar nu existensen av ett sådant primtal. Låt $M := \max\{a_1, \ldots, a_k, b_1, \ldots, b_\ell\}$. Vi börjar med att visa att $S_j(x,p) = 0$ för alla $x < \frac{p^j}{M}$. Detta följer av att $x < \frac{p^j}{M} \iff x = \frac{p^j}{M+t}$ för något $t \in \mathbb{R} > 0$, så att

$$S_j(x,p) = \sum_{i=1}^{k} \left\lfloor \frac{a_i x}{p^j} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_i x}{p^j} \right\rfloor = \sum_{i=1}^{k} \left\lfloor \frac{a_i}{M+t} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_i}{M+t} \right\rfloor = 0.$$

Nu visar vi att primtalet q uppfyller dessa krav om $q \geq M$. För ett sådant q gäller det att givet ett $x \in I \cdot \frac{q}{2}$ kommer $x < q$ ty intervallet $I \subseteq [0,2)$. Enligt antagandet är $q \geq M$ varför $x < \frac{q^j}{M}$ för alla $j \geq 2$. Alltså kommer $S_j(x,q) = 0$ för alla $j \geq 2$ och alla $x \in I \cdot \frac{q}{2}$. Detta betyder att

$$\sum_{j \geq 1} S_j(x,q) < 0, \forall x \in I \cdot \frac{q}{2},$$

vilket är väldigt nära högerledet i (13). Det som återstår är att visa att det existerar något heltal n i intervallet $I \cdot \frac{q}{2}$. Eftersom I är halvöppet kommer q kunna väljas sådant att det finns ett heltal $n \in I \cdot \frac{q}{2}$. ■



Figur 2: Värden på $S_1(x,2)$, $x \in [0,2)$ från Sats 3.3 för exempel (6).

Från Sats 3.3 kan vi nu avgöra om (6) är heltalsvärd. Figur 2 illustrerar $S_1(x,2)$ för exemplet. Notera att vid $x = 0.56$ kommer $S_1(x,2) < 0$. Sats 3.3 medför direkt att existensen av ett sådant x innebär att (6) inte är heltalsvärd.

Det ursprungliga problemet kring vilka koefficienter som innebär att $U(n)$ är heltalsvärd har nu omformulerats till en kontroll av en reellvärd golvfunktion. Det som återstår för att lösa Problem 2 är att visa att denna reellvärda funktion endast behöver kontrolleras på en mängd med ändlig kardinalitet, något som skulle innebära att vi kan formulera en algoritm som avslutas efter ändligt många steg. För att formulera och bevisa en sådan sats kommer vi först att presentera två lemman.

Lemma 3.4. Låt

$$f(x) := \lfloor kx \rfloor \,,\ k \in \mathbb{N}$$

vara definierad på intervallet $[0, 1]$. I denna definitionsmängd kommer vi endast att ha en diskontinuerlig punkt vid $x = \frac{m}{k}$ för $m \in \mathbb{N}$ där $0 \leq m \leq k$.

Se Bevis av Lemma 3.4 i Appendix A.

Notera att Lemma 3.4 används både i bevisen för Lemma 3.5 och för Sats 3.7.

Lemma 3.5. Låt

$$f_n(x) := \sum_{i=1}^{n} T_i \lfloor k_i x \rfloor \geq 0,\ k_i \in \mathbb{N},\ T_i \in \{-1, 1\}$$

vara en summa av linjära golvfunktioner definierad på intervallet $[0, 1]$. Då gäller att

$$f_n(x) \geq 0,\ \forall x \in [0, 1] \iff f_n(x) \geq 0\, \forall x \in \Omega \subset [0, 1]$$

där $\Omega = \{x = \frac{m}{k_i} \in \mathbb{Q}, 0 \leq m \leq k_i, m \in \mathbb{N}, i = 1, \ldots, n\}$ samt att $|\Omega| \leq \sum_{i=1}^{n} k_i$.

Se Bevis av Lemma 3.5 i Appendix A.

Sats 3.6. *Det gäller att*

$$U(n) \in \mathbb{N}\, \forall n \in \mathbb{N} \iff \sum_{i=1}^{k} \lfloor a_i y \rfloor - \sum_{i=1}^{\ell} \lfloor b_i y \rfloor \geq 0\, \forall y \in \Omega.$$

där $\Omega = \{y = \frac{m}{c_i}, 0 \leq m \leq c_i, m \in \mathbb{N}, i = 1, \ldots, k + \ell\} \subset [0, 1]$ *och* $\boldsymbol{c}$ *är vektorn med koefficienter* a_i, b_i *som element. Det gäller att* $|\Omega| \leq \sum_{i=1}^{k} a_i + \sum_{i=1}^{\ell} b_i$.

Bevis. Beviset inleds genom att visa "$\Longleftarrow$". Från Lemma 3.5 får vi att

$$\sum_{i=1}^{k} \lfloor a_i y \rfloor - \sum_{i=1}^{\ell} \lfloor b_i y \rfloor \geq 0\, \forall y \in \Omega \implies \sum_{i=1}^{k} \lfloor a_i y \rfloor - \sum_{i=1}^{\ell} \lfloor b_i y \rfloor \geq 0\, \forall y \in [0, 1].$$

Genom variabelbytet $y = x/2$ och med hjälp av Sats 3.3 får vi att

$$\sum_{i=1}^{k} \left\lfloor \frac{a_i x}{2} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_i x}{2} \right\rfloor \geq 0\, \forall x \in [0, 2] \implies U(n) \in \mathbb{N}\, \forall n \in \mathbb{N}.$$

Vi visar "$\Longrightarrow$" genom att göra det kontrapositiva argumentet,

$$\exists y \in \Omega : \sum_{i=1}^{k} \lfloor a_i y \rfloor - \sum_{i=1}^{\ell} \lfloor b_i y \rfloor < 0 \implies \sum_{i=1}^{k} \left\lfloor \frac{a_i x}{2} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_i x}{2} \right\rfloor < 0, x = y/2$$

där $x \in [0, 2]$ och igen tillämpa Sats 3.3

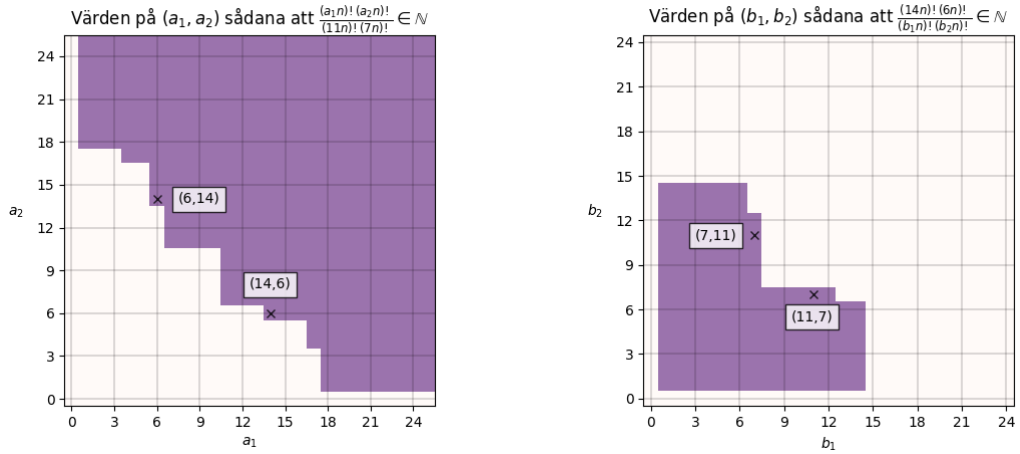
$$\exists x \in [0, 2] : \sum_{i=1}^{k} \left\lfloor \frac{a_i x}{2} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_i x}{2} \right\rfloor < 0 \implies \exists n \in \mathbb{N} : U(n) \notin \mathbb{N}.$$

varpå vi är klara. ■

```
def coeff_sums_check(num_coeffs,den_coeffs):
    '''
    Kontroll av om en rekursionsrelation på formen U(n) är heltalsvärd eller ej.
    Input:
        - lista med täljarkoefficienter num_coeffs
        - lista med nämnarekoefficienter den_coeffs
    Output:
        - check: boolean Sant/Falskt om r.r är heltalsvärd eller ej
        - n: -1 om r.r heltalsvärd, annars det värde på n där r.r
            först upptäckts inte vara heltalsvärd
    '''

    #beräkna lcm och listan med x-värden som bör kontrolleras
    lcm, x = x_generator(num_coeffs,den_coeffs)
    check = True
    n = -1
    for i,el in enumerate(x):

        #beräkna täljarsumman
        num_sum = 0
        for i in range(1,len(num_coeffs)+1):
            num_sum += floor(num_coeffs[i-1]*el/lcm)

        #beräkna nämnarsumman
        den_sum = 0
        for i in range(1,len(den_coeffs)+1):
            den_sum += floor(den_coeffs[i-1]*el/lcm)

        if den_sum > num_sum:
            check = False
            n = el
            return check, n

    return check, n
```

Kod 1: Algoritm som undersöker om en rekursionsrelation på formen (3) för givna koefficienter $\mathbf{a} = (a_1, \ldots, a_k)$, $\mathbf{b} = (b_1, \ldots, b_\ell)$ är heltalsvärd eller ej. Kontrollen genomförs på en summa av golvfunktioner från Sats 3.3 och på ändligt många steg enligt Sats 3.6. Funktionen `x_generator` ges i Kod 4 i Appendix B.

Sammantaget har vi bevisat att det räcker med att kontrollera tecknet hos en viss reellvärd funktion på en mängd med ändligt antal punkter för att avgöra huruvida en rekursionsrelation på formen (3) är heltalsvärd eller ej. Villkoren från Sats 3.3 och Sats 3.6 används för att formulera en algoritm som efter ändligt antal steg avgör om en rekursionsrelation på formen (3) är heltalsvärd eller ej. Ett exempel på en sådan algoritm ges i Kod 1. För att undvika avrundningsfel så multipliceras alla punkter med minsta gemensamma multipeln, detta resulterar i att alla punkter som undersöks är heltal. Väl i beräkningen av värdena hos den reellvärda funktionen bör vi alltså dela värdet på punkterna som undersöks med denna multipel för att erhålla korrekt resultat.



(a) Värden på (a_1, a_2) där rekursionsrelationen med fix $(b_1, b_2) = (11, 7)$ är heltalsvärd.

(b) Värden på (b_1, b_2) där rekursionsrelationen med fix $(a_1, a_2) = (14, 6)$ är heltalsvärd.

Figur 3: Värden på (a_1, a_2) eller (b_1, b_2) som medför en heltalsvärd rekursionsrelation, dessa värden är markerade med en lila kvadrat. När ena koefficientparet varieras hålls det andra fix. Figurerna genererades med hjälp av algoritmen från Kod 1. De svarta kryssen representerar rekursionsrelationen med den slutna formen från exempel (7). I båda figurer syns det att rekursionsrelationen är heltalsvärd. Observera att inbördes ordning för respektive koefficientpar inte spelar roll på grund av kommutativitet.

Med hjälp av algoritmen kan vi generera figurer som visar vilka kombinationer av täljar- respektive nämnarkoefficienter som ger en heltalsvärd rekursionsrelation när vi håller den andra delen (nämnar- respektive täljarkoefficienterna) konstant. Figur 3 illustrerar just detta. I Figur 3a låter vi nämnarkoefficienterna vara fix med $b_1 = 11$ och $b_2 = 7$ medan täljaren får variera. På analogt sätt så kommer täljarkoefficienterna i Figur 3b vara konstanta med $a_1 = 14$ och $a_2 = 6$. Exempel (7) har markerats ut med svarta kryss i båda delfigurer. Här observeras att den slutna formen av rekursionsrelationen från exempel (7) är heltalsvärd. I Figur 3a ser vi att för tillräckligt stora a_1 och a_2 kommer rekursionensrelationerna alltid vara heltalsvärda. Analogt kan vi se i Figur 3b att för tillräckligt stora b_1 och b_2 kommer rekursionsrelationerna aldrig vara heltalsvärda. Notera att dessa exempel endast har två koefficienter i täljare respektive nämnare i visualiseringssyfte, själva algoritmen kan avgöra om en rekursionsrelation är heltalsvärd för obegränsat antal koefficienter.

Slutsatsen från detta avsnitt är att vi funnit en algoritm som kan avgöra om rekursionsrelationen (3) är heltalsvärd eller ej efter ändligt antal steg, varpå Problem 2 är löst.

3.1.3 Lösning av problem 3

I beviset av Sats 3.3 såg vi att om rekursionsrelationen (3) inte är heltalsvärd för alla n kommer det existera något halvöppet intervall I och primtal q som skalar intervallet. Skalningen gav oss ett motsvarande intervall $I \cdot \frac{q}{2}$ som vi kunde visa måste innehålla ett heltal för något q. Undersöker vi detta argumentet noggrannare får vi följande sats som löser Problem 3.

Sats 3.7. *Låt* $\sum_{i=1}^{k} a_i \geq \sum_{i=1}^{\ell} b_i$ *och definiera* $L := \mathrm{MGM}(a_1, \ldots, a_k, b_1, \ldots, b_\ell)$. *Då gäller*

$$U(n) \in \mathbb{N}\, \forall n \in \mathbb{N} \iff U(n) \in \mathbb{N}\, \forall n \leq 2L.$$

Bevis. Vi börjar med att notera att "$\Longrightarrow$" följer direkt och visar därför "$\Longleftarrow$". Detta bevis utförs med hjälp av ett kontrapositivt argument. Vi gör detta genom att visa att om $U(n)$ inte är heltalsvärd för alla $n \in \mathbb{N}$, kommer det att existera ett $n_0 \leq 2L$ sådant att $U(n_0) \notin \mathbb{N}$. Från Sats 3.3 vet vi att $\exists n \in \mathbb{N} : U(n) \notin \mathbb{N} \implies \exists x \in [0, 2) : S_1(x, 2) < 0$. Notera att det x-värde som gör $S_1(x, 2)$ negativt inte kan vara $x = 2$ då vi antar att $\sum a_i \geq \sum b_i$. Enligt beviset av Sats 3.3 medför detta

att det existerar ett halvöppet intervall $I \subseteq [0,2)$ sådant att

$$\sum_{j\geq 1}\left(\sum_{i=1}^{k}\left\lfloor\frac{a_i x}{p^j}\right\rfloor-\sum_{i=1}^{\ell}\left\lfloor\frac{b_i x}{p^j}\right\rfloor\right)<0, \forall x\in I\cdot\frac{p}{2},\ p \text{ primtal } \geq M$$

där $M := \max\{a_1,\ldots,a_k,b_1,\ldots,b_\ell\}$. Vi kan använda Lemma 3.4 på termerna i $S_1(x,2)$. Detta ger att $S_1(x,2)$ endast kan vara diskontinuerlig i punkterna $x=\frac{2m}{c}$ för något $m\in\mathbb{N}$ och ett $c\in\{a_1,\ldots,a_k,b_1,\ldots,b_\ell\}$. Detta medför att $S_1\left(\frac{2x}{L},2\right)$ endast är diskontinuerlig i punkterna $x=\frac{L}{c}m$ vilka måste vara heltal. Eftersom $S_1(x,2)$ inte är diskontinuerlig i I kan vi välja ett halvöppet intervall $I\subseteq J\subseteq[0,2)$ sådant att $S_1(x,2)$ är kontinuerlig i J och $|J|=\frac{2}{L}$. Detta betyder att ett primtal $p\geq L$ gör att intervallet $J\cdot\frac{p}{2}$ måste innehålla ett heltal n eftersom $\left|J\cdot\frac{p}{2}\right|\geq\frac{2}{L}\cdot\frac{L}{2}=1$. Eftersom $J\subseteq[0,2)$ så måste $n\leq p$. Alltså måste det enligt Bertrands postulat gälla att

$$\exists n\leq 2L : U(n)\notin\mathbb{N}.$$

■

I formuleringen av Problem 3 står det att vi söker existensen av ett ”effektivt beräkningsbart $N\in\mathbb{N}$”. Vi noterar här att det N vi har tagit fram enkelt kan beräknas. Notera även att antagandet att $\sum a_i\geq\sum b_i$ i satsen ovan är rimligt ty annars ger Proposition 3.1 direkt att den motsvarande rekursionen inte är heltalsvärd.

3.2 Flervariabla fakultetfallet

I detta avsnitt undersöker vi Problem 4. Vi börjar med att utveckla (4) och får för varje $j=1,\ldots,n$ att

$$u_{\mathbf{m}}=\frac{\prod_{i=1}^{k}\prod_{r=0}^{a_{ij}-1}(a_{ij}m_j-r+\Lambda_{aj})}{\prod_{i=1}^{\ell}\prod_{r=0}^{b_{ij}-1}(b_{ij}m_j-r+\Lambda_{bj})}u_{\mathbf{m}-\mathbf{e}_j}=\cdots=\frac{\prod_{i=1}^{k}(a_{i1}m_1+\cdots+a_{in}m_n)!}{\prod_{i=1}^{\ell}(b_{i1}m_1+\cdots+b_{in}m_n)!}u_{\mathbf{0}}.$$

Den motsvarande slutna formen för det flervariabla rekursionsrelation (4) ges av

$$U(\mathbf{m})=\frac{\prod_{i=1}^{k}(a_{i1}m_1+\cdots+a_{in}m_n)!}{\prod_{i=1}^{\ell}(b_{i1}m_1+\cdots+b_{in}m_n)!}. \tag{14}$$

För att (4) ska vara vara heltalsvärd bör den slutna formen vara heltalsvärd, vilket skrivs

$$U(\mathbf{m})\in\mathbb{N}\,\forall m_j\in\mathbb{N}\cup\{0\},\mathbf{m}\neq\mathbf{0},j=1,\ldots,n. \tag{15}$$

Vi formulerar först om problemet med hjälp av p-adisk valuering och Legendres formel likt fakultetfallet. Detta medför att (15) är ekvivalent med att

$$\sum_{j\geq 1}\sum_{i=1}^{k}\left\lfloor\frac{(a_{i1}m_1+\cdots+a_{in}m_n)}{p^j}\right\rfloor-\sum_{j\geq 1}\sum_{i=1}^{\ell}\left\lfloor\frac{b_{i1}m_1+\cdots+b_{in}m_n}{p^j}\right\rfloor\geq 0, \tag{16}$$
$$\forall m_j\in\mathbb{N}\cup\{0\},\mathbf{m}\neq\mathbf{0},j=1,\ldots,n,\forall p \text{ primtal}.$$

Med hjälp av denna ekvivalenta formulering undersöker vi Problem 4 för två variabler, det vill säga $U(\mathbf{m})$ för $\mathbf{m}\in\mathbb{N}\cup\{0\}\times\mathbb{N}\cup\{0\},\mathbf{m}\neq(0,0)$.

Sats 3.8. *Låt $\mathbf{a},\mathbf{c}\in\mathbb{N}^k$ och $\mathbf{b},\mathbf{d}\in\mathbb{N}^\ell$. Då kommer*

$$\frac{\prod_{i=1}^{k}(a_in+c_im)!}{\prod_{i=1}^{\ell}(b_in+d_im)!}\in\mathbb{N}\,\forall n,m\in\mathbb{N}\cup\{0\},(n,m)\neq(0,0) \tag{17}$$

vara ekvivalent med att

$$\sum_{i=1}^{k}\left\lfloor\frac{a_ix+c_iy}{2}\right\rfloor-\sum_{i=1}^{\ell}\left\lfloor\frac{b_ix+d_iy}{2}\right\rfloor\geq 0\,\forall x,y\in[0,2].$$

Bevis. Låt oss börja med att visa "$\Longleftarrow$". Notera att vi använder den ekvivalenta omformuleringen av (17), det vill säga (16) för två variabler som blir

$$\sum_{j\geq 1}\left(\sum_{i=1}^{k}\left\lfloor\frac{a_in+c_im}{p^j}\right\rfloor-\sum_{i=1}^{\ell}\left\lfloor\frac{b_in+d_im}{p^j}\right\rfloor\right)\geq 0\,\forall n,m\in\mathbb{N}\cup\{0\},(n,m)\neq(0,0),\forall p \text{ primtal.} \quad (18)$$

Vi undersöker (18) genom att först utvidga till de reella talen x, y, så definiera

$$S_j(x,y,p):=\sum_{i=1}^{k}\left\lfloor\frac{a_ix+c_iy}{p^j}\right\rfloor-\sum_{i=1}^{\ell}\left\lfloor\frac{b_ix+d_iy}{p^j}\right\rfloor,\ j\in\mathbb{N}.$$

Vi fixerar y-ledet och på likande sätt som i beviset av Sats 3.3 visar vi att $S_j(x,y,p)$ är kvasiperiodisk i x-led med period p^j samt konstant förskjutning $C_1:=\sum_{i=1}^{k}a_i-\sum_{i=1}^{\ell}b_i$, det vill säga $S_j\left(x+p^j,y,p\right)=S_j(x,y,p)+C_1$. På samma sätt får vi likande resultat för y-ledet där $C_2:=\sum_{i=1}^{k}c_i-\sum_{i=1}^{\ell}d_i$, vilket blir att $S_j\left(x,y+p^j,p\right)=S_j(x,y,p)+C_2$. Analogt erhålls skalningsegenskapen för $S_j(x,y,p)$, det vill säga för olika värden på j och p är det enda som sker är att funktionen skalas om i både x- och y-led.

Vi drar slutsatsen att all nödvändig information finns i $S_1(x,y,2)$. Från antagandet är $S_1(x,y,2)\geq 0\ \forall x,y\in[0,2]$. Enligt kvasiperiodiciteten hos $S_j(x,y,p)$ i både x- och y-led samt att $C_1=S_1(2,0,2), C_2=S_1(0,2,2)\geq 0$ innebär detta att $S_1(x,y,2)\geq 0\ \forall x,y\in\mathbb{R}_+$. Vidare medför skalning att $S_j(x,y,p)\geq 0\ \forall x,y\in\mathbb{R}_+,\forall j\in\mathbb{N},\forall p$ primtal. Vi har därmed att $\sum_{j\geq 1}S_j(x,y,p)\geq 0\ \forall x,y\in\mathbb{R}_+$ som innehåller (18) och vi är klara med första implikationen.

Nu visar vi "$\Longrightarrow$" genom att göra det kontrapositiva argumentet. Vi gör detta genom att anta existensen av $x_0,y_0\in[0,2]$ sådana att

$$\sum_{i=1}^{k}\left\lfloor\frac{a_ix_0+c_iy_0}{2}\right\rfloor-\sum_{i=1}^{\ell}\left\lfloor\frac{b_ix_0+d_iy_0}{2}\right\rfloor<0, \quad (19)$$

och visar att detta medför att

$$\exists n,m\in\mathbb{N}\cup\{0\},(n,m)\neq(0,0):\frac{\prod_{i=1}^{k}(a_in+c_im)!}{\prod_{i=1}^{\ell}(b_in+d_im)!}\notin\mathbb{N}. \quad (20)$$

Med hjälp av p-adisk valuering är (20) ekvivalent med att

$$\begin{aligned}&\exists n,m\in\mathbb{N}\cup\{0\},(n,m)\neq(0,0),\exists p \text{ primtal}:\\&\sum_{j\geq 1}\left(\sum_{i=1}^{k}\left\lfloor\frac{a_in+c_im}{p^j}\right\rfloor-\sum_{i=1}^{\ell}\left\lfloor\frac{b_in+d_im}{p^j}\right\rfloor\right)<0.\end{aligned} \quad (21)$$

För randen $(x_0,y_0)=(2,y_0)$ där $y_0\in[0,2)$ noterar vi att

$$S_1(2,y_0,2)=\sum_{i=1}^{k}a_i-\sum_{i=1}^{\ell}b_i+\sum_{i=1}^{k}\left\lfloor\frac{c_iy_0}{2}\right\rfloor-\sum_{i=1}^{\ell}\left\lfloor\frac{d_iy_0}{2}\right\rfloor=S_1(2,0,2)+S_1(0,y_0,2)<0.$$

Vi får att antingen kommer $S_1(2,0,2)<0$ eller $S_1(0,y_0,2)<0$ som kan reduceras till det endimensionella fallet genom att låta $m=0$ respektive $n=0$. Randen $(x_0,y_0)=(x_0,2)$ där $x_0\in[0,2)$ behandlas analogt. Det som kvarstår att undersöka är randpunkten $(x_0,y_0)=(2,2)$ som täcks av Lemma A.1.

Nästa steg som vi tar är att utöka egenskapen i (19) till ett område $D\subseteq[0,2)\times[0,2)$ där $S_1(x,y,2)<0\ \forall(x,y)\in D$. Vi gör samma argument som i beviset av Sats 3.3 på funktionen

$S_1(x, y_0, 2)$ där y-ledet är fixerat på y_0. Vi får att $\exists \varepsilon_1 > 0$ sådant att $S_1(x, y_0, 2)$ är högerkontinuerlig för alla x i intervallet $I = [x_0, x_0 + \varepsilon_1)$. Detta medför att $S_1(x, y_0, 2) < 0$ för alla x i intervallet I. Ett analogt resonemang med x_0 fixerat ger intervallet $I_2 = [y_0, y_0 + \varepsilon_2)$ och vidare bildar $D = [x_0, x_0 + \varepsilon) \times [y_0, y_0 + \varepsilon),\ \varepsilon = \min\{\varepsilon_1, \varepsilon_2\}$.

Vi vet att olika p-värden endast skalar om funktionen $S_1(x, y, p)$ så $D \cdot \frac{p}{2}$ kommer att vara ett område där $S_1(x, y, p) < 0\ \forall (x, y) \in D \cdot \frac{p}{2}$.

Kvar återstår det att visa att summan av $S_j(n, m, p)$ över $j \geq 1$ är negativ för något primtal p och heltal n, m. Vi visar detta genom att välja ett primtal q sådant att $S_j(x, y, q) = 0\ \forall (x, y) \in D \cdot \frac{q}{2}, \forall j \geq 2$. Låt M vara definierat som det största elementet i $\mathbf{a}, \mathbf{b}, \mathbf{c}$ eller $\mathbf{d}$. Vi visar nu att primtalet q uppfyller dessa krav så fort $q \geq M$. För ett sådant q gäller det att $x, y < q\ \forall (x, y) \in D \cdot \frac{q}{2}$ ty $D \subseteq [0, 2) \times [0, 2)$. Då $q \geq M$ enligt antagandet får vi att $x, y < \frac{q^j}{M}\ \forall (x, y) \in D \cdot \frac{q}{2}, \forall j \geq 2$, något vi känner igen från Sats 3.3. Alltså kommer $S_j(x, y, q) = 0\ \forall (x, y) \in D \cdot \frac{q}{2}, \forall j \geq 2$. Detta medför att

$$\sum_{j \geq 1} S_j(x, y, q) < 0\ \forall (x, y) \in D \cdot \frac{q}{2},$$

där D är en produkt av halvöppna intervall och q kan väljas på ett sådant sätt att $\exists n, m \in D \cdot \frac{q}{2} : n, m \in \mathbb{N}$, vilket precis är (21). ■

Övergången från analysen av två variabler till flera variabler är minimal i skillnader av de underliggande logiska argumenten och bevismetoder.

Korollarium 3.9. Det gäller att den slutna formen $U(\mathbf{m})$ är heltalsvärd om och endast om

$$f(\mathbf{x}) := \sum_{i=1}^{k} \left\lfloor \frac{a_{i1}x_1 + \cdots + a_{in}x_n}{2} \right\rfloor - \sum_{i=1}^{\ell} \left\lfloor \frac{b_{i1}x_1 + \cdots + b_{in}x_n}{2} \right\rfloor \geq 0$$
$$\forall x_j \in [0, 2], j = 1, \ldots, n.$$

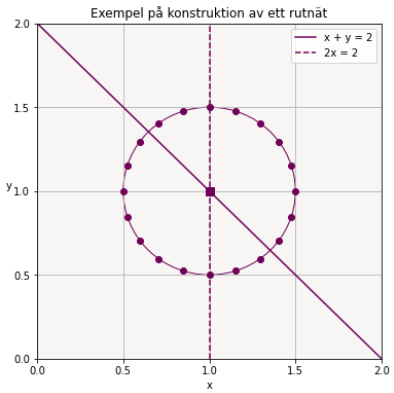
Bevis. Beviset följer analogt från Sats 3.8. ■

Sammanfattningsvis har vi fastställt att kontrollera om en viss funktion med reella värden är positiv är allt som behövs för att avgöra om en flervariabel rekursionsrelation är heltalsvärd eller inte. Det kvarstår att formulera en algoritm som efter ändligt antal steg avgör om en flervariabel rekursionsrelation är heltalsvärd eller ej för givna koefficienter. Detta visade sig vara svårt så istället illustreras hur en sådan algoritm kan konstrueras.

Vi vet att golvfunktionen i det endimensionella fakultetfallet är konstant och ändrar värde endast vid de diskontinuerliga punkterna. Ett alternativt perspektiv är att de diskontinuerliga punkterna delar definitionsmängden till olika intervall där funktionsvärdena är olika. För det flervariabla fallet utökas definitionsmängden till ett område och golvfunktionen ändras när följande termer är heltal, det vill säga när

$$b_{i1}x_1 + b_{i2}x_2 + \cdots + b_{in}x_n = 2m,$$
$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = 2m, m \in \mathbb{N}.$$

Vi har ett begränsat antal av dessa termer ty $x_j \leq 2, j = 1, \ldots, n$ och vi får begränsningen $m \leq nM$ där M är det största elementet i vektorerna $\mathbf{a}_j$ eller $\mathbf{b}_j\ \forall j$. Varje sådan term bildar ett slags plan i n dimensioner av diskontinuerliga punkter.
Definitionsmängden är begränsad ty $x_j \in [0, 2]\ \forall j$, så dessa ändliga plan delar definitionsmängden i ändliga områden där funktionsvärdena kan vara olika. Så om vi kan på något vis



Figur 4: Linjerna delar definitionsmängden för golvfunktioner i det två dimensionella fallet i fyra olika områden där funktionsvärdena kan vara olika. Från skärningspunkterna bildas ett rutnät med minst en punkt inom varje område.

hitta en punkt inom varje enskilt område är vi klara.

Vi ger ett förslag på en möjlig idé för en sådan algoritm i det två dimensionella fallet. Först hittas alla skärningspunkter och sedan bildas ett rutnät kring alla dessa punkter för att få minst en punkt från de olika områdena. Vi kommer alltid kunna hitta ett rutnät som garanterar detta ty dessa områden inte består av isolerade punkter, något vi får från beviset av Sats 3.8. Figur 4 illustrerar en konstruktion av en sådan algoritm. Detta kan möjligtvis utökas till det flervariabla fallet men vi väljer att inte undersöka detta vidare.

3.3 Linjära fallet

Innan vi löser Problem 1 så undersöker vi vad som händer om en av konstanterna a, b eller c är 0. Först begränsar vi oss till fallet då $a = 0$. Då är $U(n) = \frac{b^n}{(n+c)\cdots(1+c)}$, alltså kommer $\lim_{n\to\infty} U(n) = 0$, men $U(n) \neq 0$ så $U(n) \notin \mathbb{N}$ för något $n \in \mathbb{N}$. Om $b = 0$ får vi att

$$U(n) = \frac{a^n n!}{(n+c)\cdots(1+c)} = \frac{a^n n! c!}{(n+c)!}.$$

Eftersom $c \geq 1$ måste det existera ett primtal $p = n + c > \max\{n, c\}$, där p inte delar a. Alltså kan inte u_n vara heltalsvärd. Fallet då $c = 0$ är svårare så vi börjar med fallet då $c = 0$ och $a = 1$. Då är

$$U(n) = \frac{(n+b)\cdots(1+b)}{n\cdots 1} = \frac{(n+b)!}{n!b!} = \binom{n+b}{n}. \tag{22}$$

Vi noterar att yttersta högerledet i (22) är en binomialkoefficient varpå det direkt följer att rekursionen är heltalsvärd oavsett värde på b.

Vidare utvidgar vi begränsningen $a = 1$ och undersöker problemet för ett generellt $a \in \mathbb{N}$. I detta fall går det inte alltid att skriva om den slutna formen som en binomialkoefficient. Ett sådant exempel är när $nu_n = (2n+1)u_{n-1}$. Då kommer $U(n) = \frac{(2n+1)(2n-1)\cdots 5\cdot 3}{n!}$ där vi ser att täljaren enbart består av udda tal, alltså hade $(2n+1)!$ inkluderat för många faktorer. Om vi försöker skriva täljaren som en fakultet och delar med alla jämna tal som inte ska vara med får vi att

$$U(n) = \frac{(2n+1)!}{(2n)(2n-2)\cdots 4\cdot 2\cdot n!} = \frac{(2n+1)!}{2^n(n!)^2} = \frac{2n+1}{2^n}\binom{2n}{n}.$$

Här ser vi att 2^n dyker upp och därför kan vi inte uttrycka den slutna formen som en binomialkoefficient. Innan vi kan ge ett villkor som avgör om u_n är heltalsvärd eller ej behöver vi ha följande lemma.

Lemma 3.10. Låt p vara ett givet primtal, om p inte delar a så kommer $\nu_p((an+b)\cdots(a+b)) \geq \nu_p(n!)$ för alla $n \in \mathbb{N}$.

Bevis. Enligt Legendres formel är

$$\nu_p(n!) = \sum_{j\geq 1} \left\lfloor \frac{n}{p^j} \right\rfloor,$$

vilket är en ändlig summa för varje n. Intuitionen bakom varje term i summan är att den anger hur många p^j som är mindre än eller lika med n för varje givet j. Vi ska nu visa olikheten genom att visa att $(an+b)\cdots(a+b)$ innehåller fler termer som p^j delar än vad $n!$ gör. För ett givet p^j och n kommer det finnas heltal $\alpha, \beta \geq 0$ sådana att $n = p^j\alpha + \beta$ med $\beta < p^j$. Detta betyder att vi kan dela upp $[1, n]$ i α stycken intervall av storlek p^j, och ett intervall av storleken β. Vi visar nu att det existerar ett $n^* \in (kp^j, (k+1)p^j]$ sådant att $p^j | (an^* + b)$ för alla $k \in \{0, \ldots, \alpha - 1\}$. Vi bevisar detta genom att visa att $a(kp^j + m) + b$ tillhör varje kongruensklass modulo p^j för något $1 \leq m \leq p^j$. Vi vet att

$$a(kp^j + m_1) + b \equiv a(kp^j + m_2) + b \mod p^j \iff a(m_1 - m_2) \equiv 0 \mod p^j$$

vilket är ekvivalent med att $m_1 = m_2$ eftersom $(a, p^j) = 1$. Alltså kommer $a(kp^j + m) + b$ att passera p^j olika kongruensklasser, vilket betyder att $a(kp^j + m) + b$ passerar alla. Det existerar då ett $n^* \in (kp^j, (k+1)p^j]$ sådant att $an^* + b \equiv 0 \mod p^j$. Detta betyder att $(an + b) \cdots (a + b)$ delas av p^j minst α antal gånger. Frågan är nu vad som händer i det sista intervallet, alltså för $m \in (p^j\alpha, p^j\alpha + \beta]$. Det visar sig att detta intervall inte är viktigt eftersom $\beta < p^j$ och därför kan det inte finnas något $m \in (p^j\alpha, p^j\alpha + \beta]$ sådant att $p^j | m$. Detta avslutar beviset. ■

Ett villkor för att avgöra om fallet då $c = 0$ är heltalsvärd eller inte ges nu av följande proposition.

Proposition 3.11. Rekursionsrelationen $nu_n = (an + b)u_{n-1}$ är heltalsvärd om och endast om det för alla primtal p gäller att $p|a \implies p|b$.

Bevis. Att $nu_n = (an + b)u_{n-1}$ är heltalsvärd för ett givet $n \in \mathbb{N}$ är ekvivalent med att

$$\frac{(an + b) \cdots (a + b)}{n!} \in \mathbb{N} \iff \nu_p\left((an + b) \cdots (a + b)\right) \geq \nu_p(n!)\ \forall p, \text{ primtal.} \tag{23}$$

Antag att p är ett primtal sådant att $p|a$. Då gäller för ett givet n att

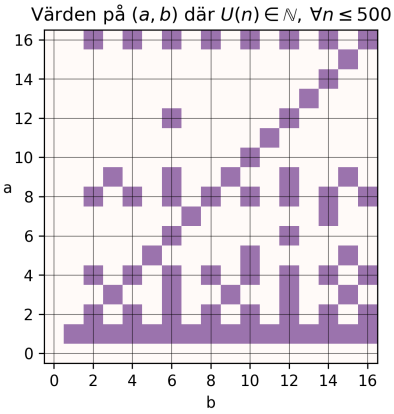
$$\nu_p(n!) = \sum_{j \geq 1} \left\lfloor \frac{n}{p^j} \right\rfloor \leq n \sum_{j \geq 1} \frac{1}{p^j} \leq n.$$

Om $b \equiv 0 \mod p$, alltså att $b = p \cdot r$ för något heltal r, så kommer

$$\nu_p\left((an + b) \cdots (a + b)\right) = \sum_{m=1}^{n} \nu_p(am + pr) = \sum_{m=1}^{n} \left(\nu_p(p) + \nu_p\left(\frac{a}{p}m + r\right)\right) \geq n.$$

Om vi dock har att $b \not\equiv 0 \mod p$ så kommer $\nu_p(am + b) = 0$ för alla $m \in \mathbb{N}$ ty om $p|(am + b)$ så måste $p|b$. Alltså kommer (23) inte gälla för $n = p$ och därför inte för alla $n \in \mathbb{N}$. Antag nu att p är ett primtal sådant att $p \nmid a$. Lemma 3.10 ger nu resultatet direkt. ■

För att bättre förstå propositionen kan vi visualisera kravet som den ställer på a och b. För att se vilka (a, b) i $\mathbb{N}^2$ som medför att $nu_n = (an + b)u_{n-1}$ är heltalsvärd för alla $n \leq 500$ går det att utföra beräkningen med dator. Vi får då Figur 5. Vi ser att precis de punkterna som markeras är de propositionen säger bör markeras. Vi kan nu använda propositionen för att avgöra att exemplet med $nu_n = (2n + 1)u_{n-1}$ från ovan inte kan vara heltalsvärt. På samma sätt ser vi att exempelvis $a = 5^3 \cdot 7, b = 2 \cdot 3^2 \cdot 5^2 \cdot 7^{10}$ skulle ge en heltalsvärd rekursion, trots att $a \nmid b$ och $b \nmid a$.



Figur 5: En lila kvadrat markerar punkten (a, b) i $\mathbb{N}^2$ om och endast om $nu_n = (an + b)u_{n-1}$ är heltalsvärd för alla $n \leq 500$. Notera att för $a = 2, 4, 8, 16$ markeras alla b där $2|b$.

Villkoret på a och b som ges i Proposition 3.11 kan användas för att skapa en algoritm som avgör om en rekursionsrelation på formen från Problem 1 med $c = 0$ är heltalsvärd eller ej för givna koefficienter a och b. En sådan algoritm kan formuleras för att avklara detta på ett ändligt antal steg ty den inte behöver kontrollera primtal $p > a$ då inga sådana primtal kan dela a. Ett exempel på en sådan algoritm visas i Kod 5 i Appendix B. Notera också att det finns icke-triviala rekursioner som är heltalsvärda. Se Figur 6 i Appendix C för en skalad motsvarighet till Figur 5. Se även Figur 7-9 i Appendix C för motsvarande bilder i fallet då $c \neq 0$.

Innan vi kan lösa Problem 1 måste vi formulera ett lemma som används i beviset av Sats 3.13 nedan. Beviset är placerat i appendix då det inte tillför mer än själva formuleringen.

Lemma 3.12. Låt c vara skriven i bas p enligt $c = c_0 + c_1 p + \ldots + c_r p^r$, $0 \leq c_i < p$ för $i = 0, \ldots, r$. Då kommer

$$\nu_p(c!) = \sum_{j=1}^{r} c_j \sum_{i=0}^{j-1} p^i.$$

Se Bevis av Lemma 3.12 i Appendix A.

Det kommer vara bekvämt att införa notationen $\delta_p(j) := \sum_{i=0}^{j-1} p^i$ inför beviset av den kommande satsen. Lemmat ovan säger då att $\nu_p(c!) = \sum_{j=0}^{r} c_j \delta_p(j)$. Notera att $\delta_p(0) = 0$ så vi kan ändra startindex för den yttre summan i lemmat till 0.

För rekursionsrelationen i Problem 1 krävs två villkor för att avgöra om den är heltalsvärd eller ej. Dessa villkor ges av följande sats. Efter beviset av satsen kommer vi visa att Problem 1 går att lösa då satsens villkor går att verifiera efter ändligt många steg för varje given rekursion på formen (2).

Sats 3.13. *Rekursionsrelationen $(n + c)u_n = (an + b)u_{n-1}$ är heltalsvärd om och endast om det för varje primtal p gäller att något av följande villkor gäller:*

I. *Om $p|a$ medför att $p|b$ och att $u_n \in \mathbb{N}$ för alla $0 < n < 4 + c$.*

II. *Skriv c i bas p som $c = c_0 + c_1 p + \ldots + c_r p^r$ för $0 \leq c_i < p$, $i = 0, \ldots, r$ och låt $c_i = 0$ för alla $i > r$ samt definiera*

$$n_m := p^m - \sum_{i=0}^{m-1} c_i p^i.$$

Om $\nu_p(a) = 0$ medför att det för alla $m \in \mathbb{N}$ existerar $n_m^* \leq n_m$ sådant att $(an_m^* + b) \equiv 0 \mod p^m$.

Bevis. Att $(n + c)u_n = (an + b)u_{n-1}$ är heltalsvärd för alla $n \leq m \in \mathbb{N}$ är ekvivalent med att $U(n) \in \mathbb{N}$, $\forall n = 1, \ldots, m$ vilket i sin tur är ekvivalent med att

$$\nu_p\left((an + b) \cdots (a + b)\right) \geq \nu_p((n + c)!) - \nu_p(c!), \tag{24}$$

gäller för alla $n = 1, \ldots, m$ och alla primtal p. Vi delar upp beviset i olika fall.

I. Antag att $p|a$ och notera att $\nu_p((n + c)!) - \nu_p(c!) \geq \nu_p(n!)$ vilket följer av att $n!|\frac{(n+c)!}{c!}$. Detta betyder att $\nu_p((p + c)!) - \nu_p(c!) \geq \nu_p(p!) = 1$. I kontrast, om $p \nmid b$ så kommer $\nu_p(an + b) = 0$ för alla $n \in \mathbb{N}$ ty om $p|(am + b)$ så måste $p|b$, en motsägelse. Därmed måste $\nu_p(b) \geq 1$ om (24) ska gälla. Så antag att $\varrho := \min\{\nu_p(a), \nu_p(b)\}$ är större än eller lika med 1, då kommer (24) för givet p och $n \in \mathbb{N}$ att vara ekvivalent med

$$\nu_p\left(\left(Ap^{\nu_p(a)-\varrho}n + Bp^{\nu_p(b)-\varrho}\right) \cdots \left(Ap^{\nu_p(a)-\varrho} + Bp^{\nu_p(b)-\varrho}\right)\right) + n\varrho \geq \nu_p((n + c)!) - \nu_p(c!).$$

Notera att om $\varrho = 1$ kommer vänsterledet ovan att vara lika med $\nu_p((An + B) \cdots (A + B))$ för $p \nmid A, B$. Alltså kommer Lemma 3.10 ge att $\nu_p((An + B) \cdots (A + B)) \geq \nu_p(n!)$. Legendres formel ger nu att

$$\nu_p(n!) = \sum_{j \geq 1} \left\lfloor \frac{n}{p^j} \right\rfloor \geq \left\lfloor \frac{n}{p} \right\rfloor \geq \frac{n}{p} - 1$$

samt att

$$\nu_p(n!) = \sum_{j \geq 1} \left\lfloor \frac{n}{p^j} \right\rfloor \leq \sum_{j \geq 1} \frac{n}{p^j} = \frac{n}{p - 1}.$$

Detta i kombination med att $\nu_p \geq 0$ ger att $n\varrho \geq \frac{n+c}{p-1} - \left(\frac{c}{p} - 1\right) \implies$ (24) om $\varrho > 1$ och att $\frac{n}{p} - 1 + n \geq \frac{n+c}{p-1} - \left(\frac{c}{p} - 1\right) \implies$ (24) om $\varrho = 1$. Dock är $\frac{n}{p} - 1 + n \leq n\rho$ för alla n om $\varrho > 1$ så vi kan samla båda fallen till att $\frac{n}{p} - 1 + n \geq \frac{n+c}{p-1} - \left(\frac{c}{p} - 1\right) \implies$ (24) för alla $\varrho \geq 1$. Detta ger nu

att $n \geq \frac{2\left(p^2-p\right)+c}{p^2-p-1} \implies (24)$. Vi observerar att $p^2-p-1 \neq 0$ för alla primtal p. Alltså håller (24) för alla $n \in \mathbb{N}$ i detta fallet om och endast om (24) gäller för alla $n < \frac{2\left(p^2-p\right)+c}{p^2-p-1}$. Vi ser nu att

$$\frac{2\left(p^2-p\right)+c}{p^2-p-1} = \frac{2\left(p^2-p-1\right)+2+c}{p^2-p-1} \leq 2+2+c = 4+c,$$

alltså kommer (24) gälla för alla $n \in \mathbb{N}$ om och endast om (24) gäller för alla $n < 4+c$.

II. Antag att $p \nmid a$ och skriv c som i satsformuleringen. Denna form på c kommer bli användbar då den unikt beskriver var c ligger mellan potenser av p. Vi kommer nu visa att

$$\forall m \in \mathbb{N}, \exists n_m^* \leq n_m : (an_m^* + b) \equiv 0 \mod p^m \iff (24) \text{ gäller } \forall n \in \mathbb{N}. \tag{25}$$

Vi visar först att n_m är det minsta n-värdet där $\nu_p((n+c)!) - \nu_p(c!)$ har en ökning i värde större än eller lika med m. Notera att $p^m | (n_m + c)$ vilket vi ser från definitionen av n_m. Notera också att om det skulle existera ett mindre n-värde, n_m^- där en ökning större än eller lika med m sker så skulle p^m vara en delare till $n_m^- + c$. Därför måste $n_m^- \leq n_m - p^m$. Dock är

$$n_m - p^m = -\sum_{i=0}^{m-1} c_i p^i \leq 0.$$

Därför är n_m det minsta sådana n-värdet ty $\{n_m : m \in \mathbb{N}\} \subseteq \mathbb{N}$.

Låt oss nu börja med "$\implies$" i (25). Låt $s \in \mathbb{N}$ vara givet. Vi kommer nu visa att $\nu_p((an+b)\cdots(a+b))$ gör en ökning större än eller lika med s för ett $n = n_s^* \leq n_s$. Vänsterledet i (25) säger att $(an_s + b), \ldots, (a+b)$ innehåller en faktor $(an_s^* + b)$ som har p^s som delare och därför gör $\nu_p((an+b)\cdots(a+b))$ en ökning av minst storlek s vid $n = n_s^* \leq n_s$. Eftersom n_s är det minsta n-värdet där $\nu_p((n+c)!) - \nu_p(c!)$ gör en ökning större än eller lika med s måste något annat sådant n-värde vara på formen $n_s + rp^s$ för något $r \in \mathbb{N}$. Så vid $n = n_s + rp^s$ kommer $\nu_p((n+c)!) - \nu_p(c!)$ att ha gjort r stycken ökningar av storlek större än eller lika med s. Eftersom $p^s | (an_s^* + b)$ så kommer vi ha att $p^s \mid (a\left(n_s^* + rp^s\right) + b)$ och vi får att $n_s^* + rp^s \leq n_s + rp^s$ varpå vi är klara. Se Figur 10 i Appendix C för exempel.

Låt oss nu göra "$\impliedby$" i (25). Låt n_s^* vara det minsta n-värdet där $\nu_p((an+b)\cdots(a+b))$ har en ökning av storlek större än eller lika med s. Vi ska nu visa att $n_s^* \leq n_s$ för varje $s \in \mathbb{N}$ då detta är den precisa innebörden av vänsterledet i (25). Antag för att få en motsägelse att $E \in \mathbb{N}$ är det minsta index sådant att $n_E^* > n_E$. Vi har då att att $n_m^* \leq n_m$ för alla $m < E$. Lemma 3.12 ger nu att

$$\nu_p((n_E+c)!) - \nu_p(c!) = \nu_p\Big(\Big(p^E - \sum_{i=0}^{E-1} c_i p^i + \sum_{i=0}^{r} c_i p^i\Big)!\Big) - \sum_{i=0}^{r} c_i \delta_p(i) =$$
$$= \nu_p\Big(\Big(p^E + \sum_{i=E}^{r} c_i p^i\Big)!\Big) - \sum_{i=0}^{r} c_i \delta_p(i).$$

För att utvärdera uttrycket ovan ser vi att om $c_E = p-1$ kan vi inte direkt tillämpa Lemma 3.12. Vi får därför att

$$\nu_p\Big(\Big(p^E + \sum_{i=E}^{r} c_i p^i\Big)!\Big) = \nu_p\Big(\Big((c_E+1)p^E + \sum_{i=E+1}^{r} c_i p^i\Big)!\Big) \geq \delta_p(E) + \sum_{i=E}^{r} c_i \delta_p(i),$$

där olikheten kommer ifrån antagandet att $c_E < p-1$. Sammantaget får vi att

$$\nu_p((n_E+c)!) - \nu_p(c!) \geq \delta_p(E) + \sum_{i=E}^{r} c_i \delta_p(i) - \sum_{i=0}^{r} c_i \delta_p(i) = \delta_p(E) + \sum_{i=0}^{E-1} c_i \delta_p(i).$$

Vi ska nu jämföra detta med värdet av $\nu_p((an_E+b)\cdots(a+b))$. För ett givet naturligt tal $s < E$ vet vi att både $\nu_p((an_s+b)\cdots(a+b))$ och $\nu_p((n_s+c))! - \nu_p(c!)$ tar ett steg av minst storlek s

i intervallet $(0, n_s]$ och vi visade ovan att de kommer fortsätta att ta steg av minst storlek s i varje intervall $(n_s + kp^s, n_s + (k+1)p^s]$ för alla $k \geq 0$. Alltså kommer $\nu_p((an_E + b)\cdots(a+b))$ att ta k stycken steg av minst storlek s om och endast om k är det största heltalet sådant att $n_s + (k-1)p^s \leq n_E$. Detta ger att

$$k \leq \frac{n_E - n_s}{p^s} + 1 = \frac{1}{p^s}\left[p^E - \sum_{i=0}^{E-1} c_i p^i - p^s + \sum_{i=0}^{s-1} c_i p^i\right] + 1 = p^{E-s} - \sum_{i=s}^{E-1} c_i p^{i-s}.$$

Vi kan nu summera hur många steg som tas av varje storlek s och observerar att det inte tas något steg av storlek E. Detta ger att

$$\nu_p((an_E + b)\cdots(a+b)) \leq \sum_{s=1}^{E-1}\left[p^{E-s} - \sum_{i=s}^{E-1} c_i p^{i-s}\right] =$$

$$= \delta_p(E) - 1 - \sum_{s=1}^{E-1}\sum_{i=s}^{E-1} c_i p^{i-s} = \delta_p(E) - 1 - \sum_{i=0}^{E-1} c_i \delta_p(i).$$

Detta är ett mindre än vårt värde för $\nu_p((n_E + c)!) - \nu_p(c!)$ vilket är en motsägelse. Alltså kan inget sådant E existera och vi får att vänsterledet i (25) gäller för alla $m \in \mathbb{N}$. ■

Det är i nuläget inte tydligt att Sats 3.13 kan användas för att ge en algoritm som löser Problem 1. Härnäst visar vi att för givna $a, b, c \in \mathbb{N}$ ger Sats 3.13 att vi kan avgöra huruvida rekursionsrelationen är heltalsvärd efter ändligt många steg. Villkoret $I.$ behöver bara kontrolleras för primtal $p|a$. Det kvarstår att visa att villkoret $II.$ kan avgöras på ändligt många steg. Villkoret behöver verifieras för alla $m \in \mathbb{N}$ och alla primtal p där $p \nmid a$. Notera att för $p^m > c$ kommer $n_m = p^m - c$. Att det existerar ett $n_m^* \leq n_m$ sådant att $an_m^* + b \equiv 0 \mod p^m$ är ekvivalent med att $an + b \not\equiv 0 \mod p^m$ för alla $n \in (n_m, p^m] = (p^m - c, p^m]$. Detta ger att $a(p^m - n) \not\equiv -b \mod p^m$ för alla $n = 0, \ldots, c-1$. Förenkling ger $an \not\equiv b \mod p^m$ för alla $n = 0, \ldots, c-1$. För $p^m > \max\{a(c-1), b\}$ kommer detta vara ekvivalent med att $an \neq b$ för alla $n = 0, \ldots, c-1$. Detta villkor går att avgöra på ändligt antal steg. Vi har därmed skissat en lösning till Problem 1 eftersom en algoritm som löser problemet kan konstrueras med hjälp av villkoren.

3.4 Förslag till fortsatta riktningar

I det här avsnittet identifieras exempel på problem och områden som kan vara av intresse för framtida arbeten inom ämnet. En naturlig fortsättning är att vidare undersöka och minska den övre gränsen given i Sats 3.7. För stora koefficientkombinationer saknar gränsen relevans eftersom många iterationer ändå behöver genomföras för ett stort antal steg.

Andra förslag på fortsättningar berör Problem 1 och demonstreras av följande. Här går det att undersöka när rekursionsrelationen $f(n)u_n = g(n)u_{n-1}$ är heltalsvärd för polynom f, g av grad högre än 1. Studier av problemet kan vidare generaliseras till att inkludera mer generella startvillkor än $u_0 = 1$. Till exempel genom att se vilka villkor som uppstår om u_0 tillåts vara ett positivt rationellt tal. Ytterligare arbete krävs även för att presentera en explicit algoritm som löser Problem 1, som visats kommer existera.

Vidare är det även av intresse att ge en explicit algoritm som efter ändligt antal steg kan avgöra huruvida ett flervariabelt fakultetproblem är heltalsvärd eller inte, likt algoritmen som givits i resultatdelen för fakultetfallet.

Referenser

[1] Bell JP, Bruin N och Coons M. *Transcendence of generating functions whose coefficients are multiplicative*. I: *Trans. Am. Math. Soc.* 2012; 364(2); s. 933–59. DOI: 10.48550/arXiv.1003.2221.

[2] Bober JW. *Factorial ratios, hypergeometric series, and a family of step functions*. I: *J. London Math. Soc.* 2009; 79(2); s. 422–44. DOI: 10.1112/jlms/jdn078.

[3] Brin S och Page L. *The anatomy of a large-scale hypertextual Web search engine*. I: *Computer Networks and ISDN Systems*. 1998; 30(1); s. 107–17. DOI: 10.1016/S0169-7552(98)00110-X.

[4] Chebyshev PL. *Mémoire sur les nombres premiers*. I: *J. Math. Pures Appl.* 1852; 17; s. 366–90.

[5] Dirichlet PGL. *There are infinitely many prime numbers in all arithmetic progressions with first term and difference coprime*. Stephan R, översättare. 2014. DOI: 10.48550/arXiv.0808.1408.

[6] Feller W. *An Introduction to Probability Theory and Its Applications*. 3 uppl. Hoboken, USA: Wiley, 1968. Kap. 2.9, Elements of Combinatorial Analysis, Stirling's Formula; s. 52–4.

[7] Grout I. *Digital Systems Design with FPGAs and CPLDs*. Burlington, USA: Newnes, 2008. Kap. 2, Electronic Systems Design; s. 43–121. DOI: 10.1016/B978-0-7506-8397-5.00002-7.

[8] Kim D, Kan Y, Aum Y, Lee W och Yi G. *Hotspots-based patrol route optimization algorithm for smart policing*. I: *Heliyon*. 2023; 9(10). DOI: 10.1016/j.heliyon.2023.e20931.

[9] Landau E. *Sur les conditions de divisibilité d'un produit de factorielles par un autre*. I: *Nouv. Ann. Math.* 1900; 19; s. 344–62.

[10] Ljungbergh W, Johnander J, Petersson C och Felsberg M. *Raw or cooked? Object detection on RAW images*. I: Gade R, Felsberg M och Kämäräinen J, redaktörer. Image Analysis. Scandinavian Conference on Image Analysis. 18–21 april 2023; Sirkka, Finland. Cham, Schweiz; Springer; 2023; s. 374–85. DOI: 10.1007/978-3-031-31435-3_25.

[11] Moll VH. *Numbers and Functions: From a Classical Experimental Mathematicians Point of View*. Providence, USA: American Mathematical Society, 2012. Kap. 2.6, Factorials and binomial coefficients: The prime factorization of $n!$; s. 76–7.

[12] Nguyen MD, Wedler M, Stoffel D och Kunz W. *Formal hardware/software co-verification by interval property checking with abstraction*. I: 2011 48th ACM/EDAC/IEEE Design Automation Conference. 5–9 juni 2011; San Diego, USA. IEEE; 2011; s. 510–5. DOI: 10.1145/2024724.2024843.

[13] Rashid M, Ardito L och Torchiano M. *Energy Consumption Analysis of Algorithms Implementations*. I: 2015 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. 22–23 okt. 2015; Beijing, Kina. IEEE; 2015; s. 1–4. DOI: 10.1109/ESEM.2015.7321198.

[14] Weisstein EW. *Hypergeometric Function*. Från MathWorld–A Wolfram Web Resource. [Internet. Hämtad 28 april, 2024]. URL: https://mathworld.wolfram.com/HypergeometricFunction.html.

A Appendix – teori

Bevis av Lemma 3.4. Golvfunktionen är tydligt diskontinuerlig vid varje heltal eftersom högerrespektive vänstergränsvärdena inte konvergerar mot samma värde. ■

Bevis av Lemma 3.5. Likt lemmat ovan så följer det från definitionen av golvfunktionen att funktionen $f_n(x)$ endast kommer att ändra värde vid de diskontinuerliga punkterna. Är funktionsvärdet positivt (eller lika med 0) i dessa punkter är lemmat bevisat. Lemma 3.4 medför att varje enskild term har begränsat antal diskontinuerliga punkter. Då summan av linjära funktioner är linjär kommer antalet diskontinuiteter som mest vara summan av antalet diskontinuiteter för varje funktion. ■

Lemma A.1. För att

$$U(\mathbf{m}) \in \mathbb{N}\, \forall m_j \in \mathbb{N} \cup \{0\}, \mathbf{m} \neq \mathbf{0}, j = 1, \ldots, n$$

måste

$$\sum_{j=1}^{n}\sum_{i=1}^{k} a_{ij} \geq \sum_{j=1}^{n}\sum_{i=1}^{\ell} b_{ij.}$$

Bevis. Vi bevisar detta endast för $n = 2$ då alla andra fall följer analogt. För $n = 2$ har vi precis (17)

$$\frac{(a_1 n + c_1 m)!(a_2 n + c_2 m)! \cdots (a_k n + c_k m)!}{(b_1 n + d_1 m)!(b_2 n + d_2 m)! \cdots (b_\ell n + d_\ell m)!}.$$

Låt $m = n$ i ovan, detta ger oss

$$\frac{((a_1 + c_1)n)!((a_2 + c_2)n)! \cdots ((a_k + c_k)n)!}{((b_1 + d_1)n)!((b_2 + d_2)n)! \cdots ((b_\ell + d_\ell)n)!}.$$

Något som vi känner igen från fakultet fallet och med hjälp av Proposition 3.1 få

$$\sum_{i=1}^{k}(a_i + c_i) \geq \sum_{i}^{\ell}(b_i + d_i).$$

■

Bevis av Lemma 3.12. Legendres formel ger att

$$\nu_p(c!) = \sum_{j \geq 1} \left\lfloor \frac{c_0 + c_1 p + \ldots + c_r p^r}{p^j} \right\rfloor.$$

Skriver vi ut termerna för $j = 1, \ldots, r$ får vi

$$\begin{aligned} j = 1: &\quad c_1 + \ldots + c_r p^{r-1} \\ j = 2: &\quad c_2 + \ldots + c_r p^{r-2} \\ &\vdots \\ j = r-1: &\quad c_{r-1} + c_r p \\ j = r: &\quad c_r. \end{aligned}$$

Notera att för $j > r$ blir termerna noll. Detta ger att hela summan blir

$$\sum_{j=1}^{r} \left\lfloor \frac{c_0 + c_1 p + \ldots + c_r p^r}{p^j} \right\rfloor = c_1 + c_2(1 + p) + \ldots + c_r \sum_{i=0}^{r-1} p^i.$$

Vilket är precis vad vi ville visa. ■

B Appendix – källkod

```
def divides(num,den):
'''
Returnerar boolean Sant/Falskt beroende på om num/den är ett heltal.
'''
return num%den == 0
```

Kod 2: Hjälpfunktion som avgör om en given täljare delat med en given nämnare är ett heltal eller ej.

```
def is_prime(n):
'''
Avgör om ett givet tal är ett primtal.

Input:
    - positivt heltal n

Output:
    - boolean Sant/Falskt om n är ett primtal eller ej
'''

if n < 2: # 1 ej primtal
     return False
if n % 2 == 0:
     return n == 2  # Falskt om n =/= 2
p = 3
while p*p <= n:
     if n % p == 0:
         return False
     p += 2
return True
```

Kod 3: Algoritm som avgör om ett givet positivt heltal n är ett primtal eller ej.

```
def x_generator(num_coeffs, den_coeffs):
'''
Beräknar och returnerar en ändlig lista med x-värden som behöver undersökas
för att avgöra given r.r (fakultet) är heltalsvärd eller ej.

Input:
    - lista med täljarkoefficienter num_coeffs
    - lista med nämnarkoefficienter den_coeffs

Output:
    - lista i stigande ordning med unika punkter som ska kontrolleras
    - MGM för problemet (dvs MGM av täljar- och nämnarkoeff)

'''

#beräkna mgm
mgm = 1
for i in set(num_coeffs + den_coeffs):
    mgm = mgm*i//gcd(mgm, i)

# beräkna x där golvfunktionen kommer hoppa (diskontinuiteter)
x_list = []
for i,el in enumerate(num_coeffs):
    for n in range(1,el + 1):
        x_list.append(mgm*n/el)


for i,el in enumerate(den_coeffs):
    for n in range(1,el + 1):
        x_list.append(mgm*n/el)

x = sorted(set(x_list)) #ta bort dubletter och sortera i stigande ordning

return mgm, x
```

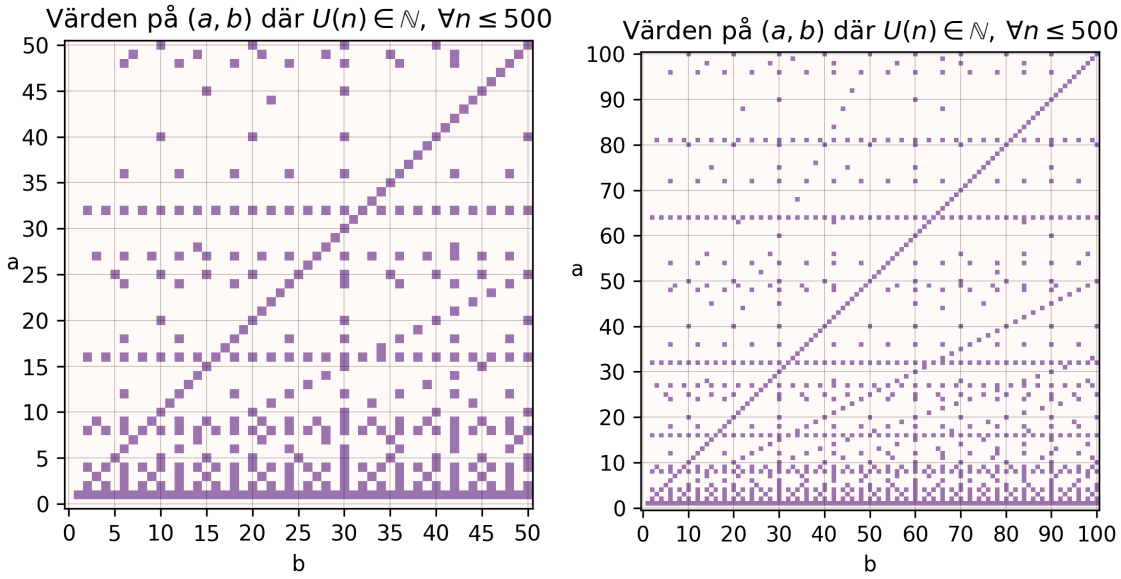
Kod 4: Algoritm som beräknar en lista med x-värden enligt Sats 3.6 som används i algoritmen från kod 1. Observera att varje koordinat multipliceras med minsta gemensamma multipeln (MGM) för att kringgå avrundningsfel.

```python
def linear_case_c0(a,b):
'''
Undersöker om rekursionen
    n*u_n = (a*n+b)u_{n-1}
är heltalsvärd för givet a,b. Oberoende av n.
Input:
    - a, b
Output:
    - boolean Sant eller Falskt beroende på om rekursionen är heltalsvärd.
'''
integrality = True
for p in range(2,a+1):
    if divides(a,p) and is_prime(p):
        if not divides(b,p):
            integrality = False
            return integrality
return integrality
```

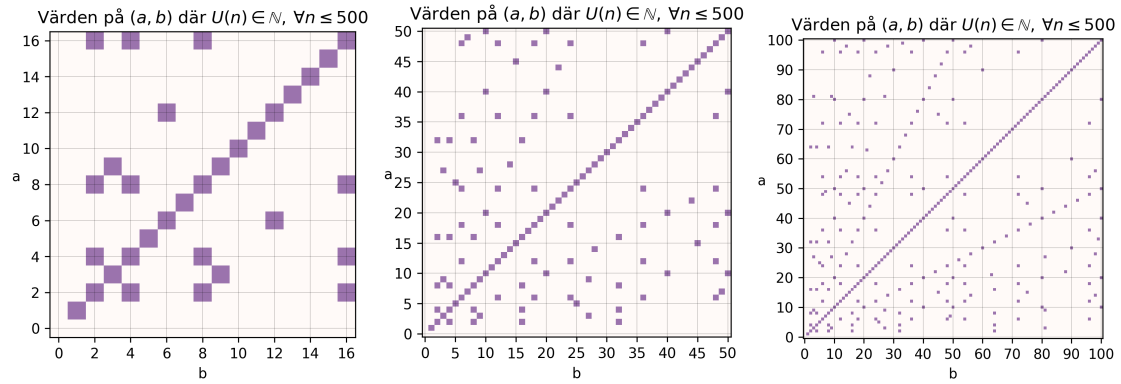
Kod 5: Algoritm som undersöker om rekursionsrelationen från Problem 1 är heltalsvärd eller ej för givet a, b och $c = 0$. Algoritmen avslutas efter ändligt antal steg då vi endast undersöker $p \in [2, a]$.

C Appendix – figurer

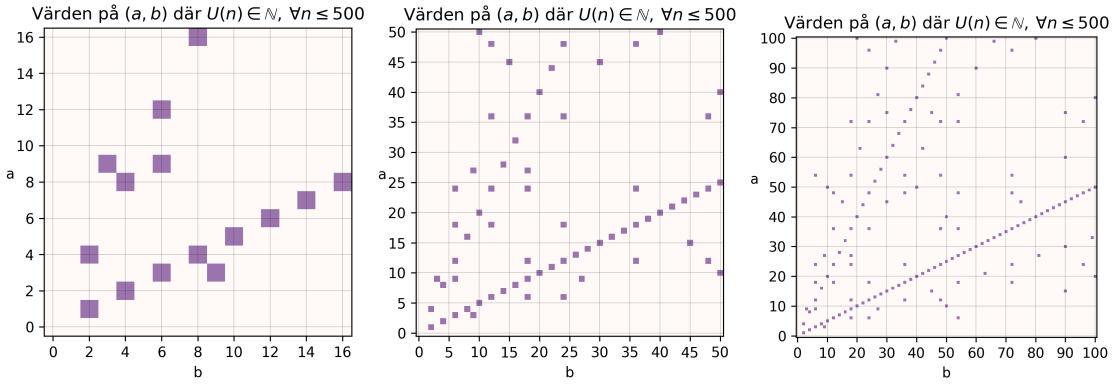
Samling med figurer som stödjer rapporten.



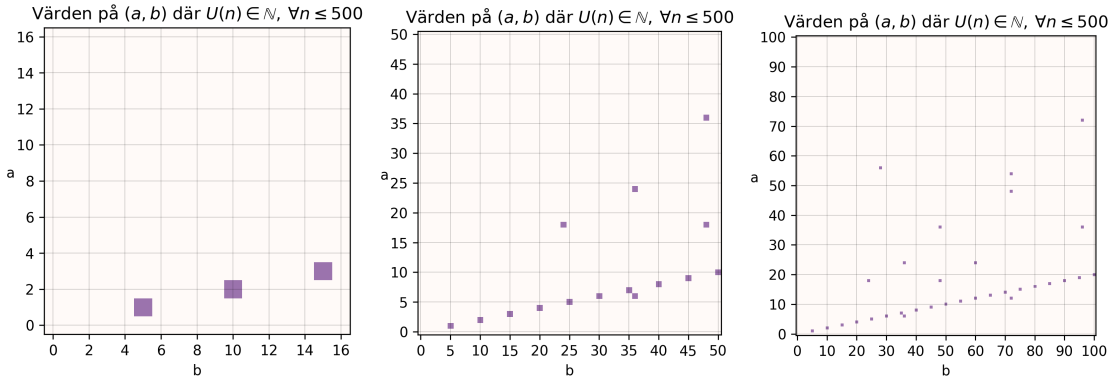
Figur 6: Två olika skalade bilder där en lila kvadrat markerar punkten (a, b) i $\mathbb{N}^2$ om och endast om $nu_n = (an + b)u_{n-1}$ är heltalsvärd för alla $n \leq 500$. Notera att villkoret i Proposition 3.11 ger exakt samma bild.



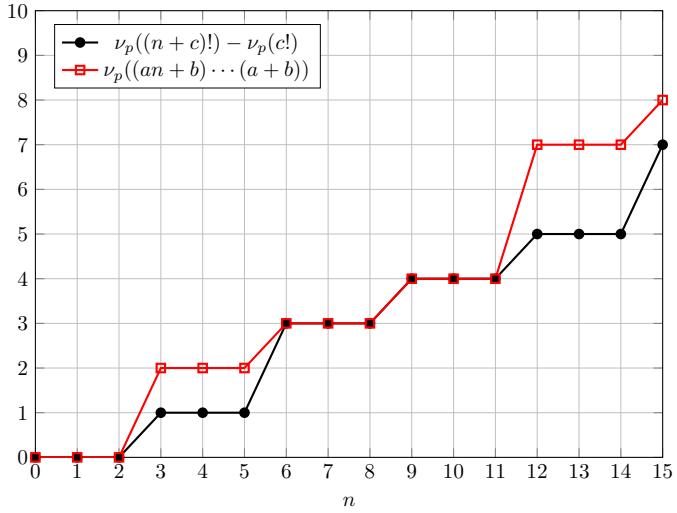
Figur 7: Tre olika skalade bilder där en lila kvadrat markerar punkten (a, b) i $\mathbb{N}^2$ om och endast om $(n + 1)u_n = (an + b)u_{n-1}$ är heltalsvärd för alla $n \leq 500$. Notera att bilderna ser ut att vara symmetriska kring $a = b$ linjen.



Figur 8: Tre olika skalade bilder där en lila kvadrat markerar punkten (a, b) i $\mathbb{N}^2$ om och endast om $(n+2)u_n = (an+b)u_{n-1}$ är heltalsvärd för alla $n \leq 500$. Notera att bilderna inte ser ut att vara symmetriska kring $a = b$ linjen.



Figur 9: Tre olika skalade bilder där en lila kvadrat markerar punkten (a, b) i $\mathbb{N}^2$ om och endast om $(n+5)u_n = (an+b)u_{n-1}$ är heltalsvärd för alla $n \leq 500$. Noter att antalet icke-triviala lösningar (där $b \neq c \cdot a$) ser ut att minska för stora c.



Figur 10: Graf av ett exempel där $\nu_p((an+b)\cdots(a+b)) \geq \nu_p((n+c)!) - \nu_p(c!)$ för alla $n \leq 10$. Notera att i detta fallet är $n_1^* = 3 \leq 3 = n_1$ och $n_2^* = 3 \leq 6 = n_2$. Sedan ser vi att $n_3^* = 12$ vilket är mindre än n_3 då vi inte ser var n_3 sker.