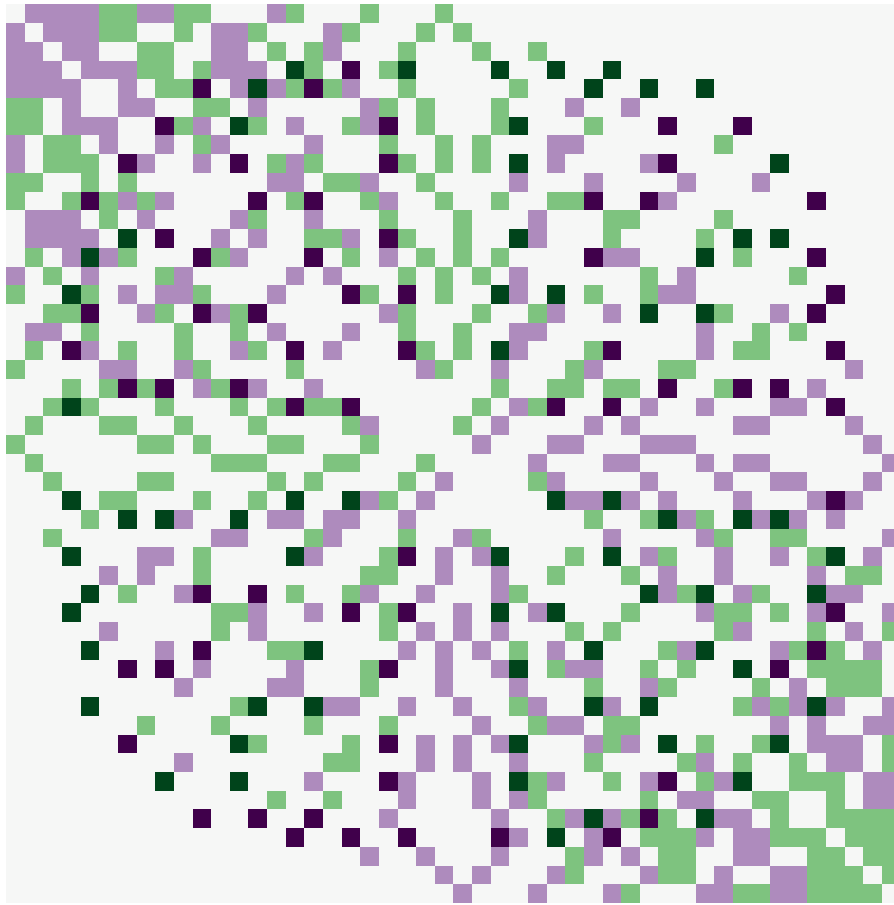




CHALMERS



Mjukvara för hantering av grassmanntal i Python och numeriskt bevis av CDSW-förmodan

Kandidatarbete inom teknisk fysik

Alexandru Golic
David Hambræus
Eli Ismailov

INSTITUTIONEN FÖR FYSIK
CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige 2021
www.chalmers.se

Mjukvara för hantering av grassmanntal i Python och numeriskt bevis av CDSW-förmodan
Alexandru Golic, David Hambræus, Eli Ismailov

© Alexandru Golic, David Hambræus, Eli Ismailov, 2021.

Handledare: Gabriele Ferretti
Examinator: Jan Swenson

Kandidatarbete 2021
TIFX04-21-83
Institutionen för fysik
Chalmers Tekniska Högskola
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Framsida: Visualisering av strukturkonstanterna för den exceptionella gruppen F_4 i Chevalleybasen.

Typsatt i L^AT_EX
Göteborg, Sverige 2021

Mjukvara för hantering av grassmanntal i Python och numeriskt bevis av CDSW-förmodan

Alexandru Golic
David Hambraeus
Eli Ismailov

Sammandrag

I detta arbete presenteras mjukvara utvecklad i Python som genererar strukturkonstanterna för en godtycklig enkel kompakt liealgebra i Chevalley-basen. Dessutom presenteras mjukvara för att programmatiskt hantera polynom av grassmanntal. Mjukvaran används sedan i syfte att numeriskt bevisa en förmodan formulerad av Freddy Cachazo, Michael R. Douglas, Nathan Seiberg och Edward Witten, förkortad CDSW-förmodan, i [1]. Två olika metoder testades för att försöka bevisa förmodan, en baserad på projektion av ett polynom av grassmanntal på ett antal nollpolynom och en baserad på att hitta en specifik uppsättning grassmanntal som löser förmodan. Den första metoden fungerade och förmodan kunde numeriskt bevisas för den exceptionella gruppen G_2 för första gången. Tidskomplexiteten för denna metod visades dock vara exponentiell och metoden kan ej användas för större grupper.

Software for Handling Polynomials of Grassmann Numbers in Python and a Numerical Proof of the CDSW Conjecture

Alexandru Golic
David Hambræus
Eli Ismailov

Abstract

This project presents software designed in Python that can generate the structure constants for any simple compact liealgebra in the Chevalley basis. Further, software that enables handling of Grassmann numbers programatically is presented. The software is then used to numerically prove a conjecture by Freddy Cachazo, Michael R. Douglas, Nathan Seiberg and Edward Witten, the CDSW conjecture, in [1]. Two methods for proving the conjecture were tested, one based on the projection of a polynomial of Grassmann numbers on a set of zero-polynomials, and one based on trying to find a specific set of Grassmann numbers satisfying the conjecture. The first method was successful and the conjecture was numerically proven for the exceptional group G_2 for the first time. However, the time complexity of this method was exponential and it could not be used for larger groups.

Innehåll

1	Introduktion	1
2	Bakgrund	3
2.1	Det kirala kondensatet $\langle \bar{q}q \rangle$	3
2.2	Ren $\mathcal{N} = 1$ supersymmetrisk Yang-Mills teori	4
2.3	Gluino- och glueballsperfältet	5
3	Teori	7
3.1	Några egenskaper hos grassmanntal	7
3.2	Liegrupper och liealgebror	8
3.2.1	Liegrupper och deras fysiska relevans	8
3.2.2	Från liegrupper till liealgebror	8
3.2.3	Strukturkonstanter	10
3.2.4	Killingformen	11
3.2.5	Rötter och några av deras egenskaper	11
3.2.6	Klassificering av liegrupper	12
3.2.7	Chevalley-basen	12
3.2.8	Dualt Coxeter-tal	13
4	CDSW-förmodan	15
4.1	Generell formulering av CDSW-förmodan	15
4.2	Specifik form av S och ekvivalensrelationerna	16

5	Generering av strukturkonstanter	17
5.1	Linjärkombinationer av enkla rötter som ger rötter	17
5.2	Generering av rötter	17
5.2.1	Finna enkelt kopplade rötter	18
5.2.2	Generera rötter via enkla rotkedjor	18
5.3	Kommutationer	19
6	Programdesign	21
6.1	En klass för polynom av grassmanntal	21
6.1.1	Termklassen	21
6.1.2	Polynomklassen	22
6.2	Projektion	23
6.2.1	Ett enkelt exempel	23
6.2.2	Generalisering	25
6.3	Att hitta en specifik uppsättning grassmanntal	26
6.3.1	Åter till det enkla exemplet	26
6.3.2	Generalisering	26
6.3.3	Gissa och testa	27
6.3.4	Semi-analytiskt tillvägagångssätt	27
7	Resultat och Diskussion	29
7.1	Resultat av projektionsmetoden	29
7.2	Resultat av att försöka hitta en specifik uppsättning	30
7.3	Framtida forskning	31

Kapitel 1

Introduktion

Den mest framgångsrika teorin vi har för att beskriva partikelfysik är standardmodellen. Denna utvecklades under andra halvan av 1900-talet och sedan dess har flera av dess förutsägelser verifierats experimentellt. Trots detta vet vi att modellen inte är fullständig, då det finns flera fenomen som den inte kan förklara som exempelvis mörk materia, mörk energi, varför higgsbosonen har den massan den har samt hur gravitationen verkar på partikelnivå.

På grund av detta har flera fysiker försökt ta fram andra modeller som bättre skulle kunna förklara dessa fenomen. En klass av sådana teorier är supersymmetriska teorier. Dessa postulerar bland annat att det till varje fermion finns en ”superpartner” som är en boson, och vice versa.

På grund av att supersymmetriska teorier, som namnet antyder, innehåller fler symmetrier än standardmodellen är vissa problem matematiskt enklare att lösa i en supersymmetrisk modell än i standardmodellen. Ett exempel på detta är det som kallas kiralt kondensat vilket bland annat ger upphov till vakuumfluktuationer, som i standardmodellen beskrivs av $\langle \bar{q}q \rangle \approx -(250 \text{ MeV})^3$, där q är Dirac-fältet för en upp-kvark eller en ner-kvark. Detta värde har man approximerat med vissa numeriska metoder men något exakt analytiskt värde har ej bestämts än. [2]

Inom supersymmetriska modeller har man dock kommit längre. År 2002 formulerade Cachazo, Douglas, Seiberg och Witten (CDSW) en förmodan om den supersymmetriska motsvarigheten till $\bar{q}q$ inom ren $\mathcal{N} = 1$ supersymmetrisk Yang-Mills teori som kallas S [1]. Inom en sådan teori har man möjligheten att välja vilka symmetrier som varje partikel uppvisar, och varje symmetri beskrivs av en liegrupp. Beroende på vilka symmetrier som gäller ändras formen av S .

Kortfattat påstår den del av förmodan vi är intresserade av att $S^h \sim 0$ för alla enkla liegrupper. S kommer alltid vara ett polynom av antikommutativa tal, så kallade grassmanntal, och det ska vara ekvivalent med 0 givet ett antal ekvivalensrelationer som även de är polynom av grassmanntal. Formen på S och ekvivalensrelationerna beror på den valda liegruppens struktur, de så kallade strukturkonstanterna. Denna förmodan har redan bevisats matematiskt för de fyra oändliga familjerna av enkla liegrupper,

$SU(N)$, $SO(2N)$, $SO(2N + 1)$ och $Sp(2N)$, och är även bevisad för den minsta av de exceptionella grupperna G_2 . Det kvarstår dock att bevisa den för de fyra återstående exceptionella grupperna, F_4 , E_6 , E_7 och E_8 .

Syftet med detta arbetet är att utveckla en numerisk metod för att bevisa CDSW-förmodan. För att göra det behövs två delar: För det första behövs en metod för att explicit beräkna de så kallade strukturkonstanterna, f_{ab}^c , för liegrupperna vi är intresserade av. Dessa avgör både formen på S och ekvivalensrelationerna. För det andra behövs ett sätt att hantera grassmantal programmatiskt. De är inte vanliga tal och ny mjukvara behöver därför utvecklas för att kunna undersöka deras egenskaper.

Kapitel 2

Bakgrund

I detta avsnitt beskrivs varför vi är intresserade av CDSW-förmodan; det sätter resten av arbetet i sitt sammanhang. För att fullt ut förstå delarna om supersymmetri krävs dock en förståelse av kvantfältteori. Den intresserade läsaren kan såklart läsa igenom dem ändå, men det inte är nödvändigt att förstå de delarna för att begripa resten av arbetet.

Först diskuteras vad som menas med ett kiralt kondensat och varför det är intressant. Därefter introduceras ren $\mathcal{N} = 1$ supersymmetrisk Yang-Mills teori. Det är en teori som ger gluonen (som är en boson) en fermion-partner som kallas gluino. I beskrivningen av gluino uppkommer de grassmannantal som S innehåller. Genomgående i kapitlet används Einsteins summationskonvention.

2.1 Det kirala kondensatet $\langle \bar{q}q \rangle$

Kvantkromodynamiken (eller QCD) är teorin som ligger till grunden av vår förståelse av den starka kraften och är en av de teorier som utgör standardmodellen. QCD är dock en matematiskt komplicerad teori, vilket gör att många analytiska beräkningar om fenomen inom QCD görs med diverse approximationer. En av de vanligaste för detta ändamål är störningsteori, där man betraktar ett komplicerat kvantsystem som en liten störning av ett enklare system [3].

Dock förblir vissa problem som av olika anledningar inte kan lösas med störningsräkning olösta. Ett viktigt exempel på dessa är värdet av det så kallade kirala kondensatet $\langle \bar{q}q \rangle$. Detta kan liknas t.ex till elektronkondensat i supraleddare; vid låga temperaturer tenderar elektroner att "kondenseras" i par av spinn-upp och spinn-ner elektroner (s.k Cooper-par) vilket ger materialet mycket annorlunda elektriska egenskaper än vid högre temperaturer. Analogt tenderar lågenergikvarkar att para ihop sig i kvark-antikvark-par [4]. Det är förväntansvärdet på dessa lågenergetiska interaktioner som kallas det kirala kondensatet, och med numeriska metoder har man beräknat att dess värde är c.a $-(250\text{MeV})^3$. [5]

Att detta värde är nollskilt har fundamentala konsekvenser för hur universum fungerar. Det är detta fenomen som till exempel bidrar med majoriteten av nukleoners massa [4].

Trots att man numeriskt beräknat det kirala kondensatet är vägen till en analytisk lösning lång, och vi har fortfarande inte full förståelse för varför det sker.

Detta är den fysiska anledningen till varför CDSW-förmodan är intressant; det förmodan behandlar är en supersymmetrisk analog till det kirala kondensatet, som tack vare de extra symmetrierna till synes ger en mycket enklare analytisk lösning. Förhoppningen är att lösningen till det enklare problemet ger insikter om hur man skulle gå tillväga för att lösa problemet i QCD.

2.2 Ren $\mathcal{N} = 1$ supersymmetrisk Yang-Mills teori

År 1954 publicerade Cheng Ning Yang och Robert Mills en artikel [6] där de framförde en kvantiserad $SU(2)$ invariant gaugeteori för att beskriva en isospinnrelation mellan protoner och neutroner. Symmetrin de postulerade var fel men la grunden till ett fält av teorier kallade Yang-Mills teorier där partiklars egenskaper förklarades med icke-abelska liealgebror och en gaugeinvariant lagrangian.

Alla gaugeteorier som beskriver olika partiklars egenskaper med hjälp av specifika icke-abelska¹ liealgebror kallas nu för Yang-Mills teorier. Bortsett från teorier om gravitation är de flesta teorier i modern fysik kvantiserade Yang-Mills teorier. Några exempel är QCD, där symmetrigruppen är $SU(3)$, elektrosvaga växelverkan med gruppen $SU(2) \times U(1)$ eller standardmodellen med gruppen $SU(2) \times U(1) \times SU(3)$.

En gaugeinvariant lagrangian för en Yang-Mills teori har formen

$$\mathcal{L} = \bar{\Psi}_j (i\gamma^\mu (\delta_i^j \partial_\mu + igA_\mu^a (T_a)_i^j)) \Psi^i - \frac{1}{4} F^{a\mu\nu} F_{a\mu\nu} \quad (2.1)$$

där Ψ är fermioniska fältet, γ^μ är Dirac-matriserna. g är den starka kopplingskonstanten, A_μ^a är gaugefältet, T_a är generatorerna för algebran som gäller och $F^{a\mu\nu}$ är gaugefältets styrketensor. Detta uttryck är uppdelat i två termer, $\bar{\Psi}_j (i\gamma^\mu (\delta_i^j \partial_\mu + igA_\mu^a (T_a)_i^j)) \Psi^i$ och $-\frac{1}{4} F^{a\mu\nu} F_{a\mu\nu}$, där den andra termen kallas för den rena delen av lagrangianen. I en ren teori är det endast den rena delen som är relevant. I exempelvis QCD gäller första termen kvarkar medan andra termen gäller gluoner, i ren QCD hanteras alltså endast gluonen med lagrangianen

$$\mathcal{L}_{\text{ren}} = -\frac{1}{4} F^{a\mu\nu} F_{a\mu\nu}. \quad (2.2)$$

Att en teori är supersymmetrisk innebär att det finns en symmetri mellan bosoner och fermioner. För varje fermion finns en supersymmetrisk partner kallad superpartner som är en boson och vice versa för bosoner. Representationen för paret av partikel och superpartner kallas för en supermultiplett. Exempelvis gäller för gluonen, som är en boson, att det finns en supersymmetrisk partner kallad gluino som är en fermion.

När en ren Yang-Mills teori supersymmetriseras måste gluonens superpartner, gluino,

¹Att en grupp är icke-abelsk innebär att dess gruppemoment inte är kommutativa.

inkluderas i den rena lagrangianen. Resultatet blir då

$$\mathcal{L}_{\text{ren}} = \bar{\lambda}_c i \gamma^\mu (\delta_b^c \partial_\mu + i g A_\mu^a (T_a)_b^c) \lambda^b - \frac{1}{4} F^{a\mu\nu} F_{a\mu\nu} \quad (2.3)$$

där λ^b är superfältet för gluinon. Gluinon är sin egen antipartikel, precis som gluonen, och kan därmed representeras av Majorana-spinorer med fyra komponenter. Men då den är sin egen antipartikel är de första två komponenterna relaterade till de andra två genom laddningskonjugering, och det är därför endast två oberoende komponenter. Därmed kan vi istället beskriva gluinon med Weyl-spinorer, som bara har två komponenter, vilket reducerar komplexiteten.

Supersymmetri har många egenskaper gemensamt med standardmodellen som exempelvis vakuumförväntningsvärden, vilka är väldigt svåra att arbeta med inom vanlig QCD. Eftersom supersymmetrin ger ytterligare verktyg blir många av dessa gemensamma egenskaper lättare att hantera. Ett exempel på detta är det som kallas för kiralt symmetribrott, som i standardmodellen beskrivs av $\langle \bar{q}q \rangle \approx -(250 \text{ MeV})^3$, där q är Dirac-fältet för en upp-kvark eller en ner-kvark. Detta värde har man approximerat med vissa numeriska metoder men något exakt analytiskt värde har ej bestämts än. Under supersymmetri finns metoder att nyttja för att lösa dessa problem analytiskt där CDSW-förmodan är ett steg mot detta.

I supersymmetri finns alltså operatorer som transformerar mellan boson och fermion, Q och Q^\dagger ,

$$Q |\text{Boson}\rangle = |\text{Fermion}\rangle, \quad Q |\text{Fermion}\rangle = |\text{Boson}\rangle. \quad (2.4)$$

En egenskap för Q och Q^\dagger är att de kommuterar med fyrvektorn för rörelsemängd P^μ , rumtidsrotationer och translationer. Q och Q^\dagger kommuterar även med generatorer för gauge-transformationer och därmed kommer alla partiklar och dess superpartners tillhöra samma gaugegrupp och representation. Detta innebär att alla partiklar inom samma supermultiplett har samma egenvärde till operatorerna för dessa kommuterande operatorer och alltså kommer en partikel och dess superpartner ha samma massa och gaugekvanttal. Att det endast finns en supersymmetri, alltså ett unikt Q , är det som gör teorin till $\mathcal{N} = 1$. För en mer djupgående förklaring till supersymmetri och implikationer för $\mathcal{N} = 1$ refereras [7].

2.3 Gluino- och glueballsUPERFÄLTET

Ekvation (2.3) ger gluinosuperfältet $\lambda_\alpha^a(x, \theta, \bar{\theta})$, där $\alpha = 1, 2$ ger komponenter i spinorn. En informell definition av ett superrum, vilket alla superfält existerar inom, är ett rum som har de bekanta rumtidskoordinaterna x men även grassmannkoordinater $\theta, \bar{\theta}$. Förmodan kommer vara oberoende av var i superrummet man befinner sig och därmed kommer $\lambda_\alpha^a(x, \theta, \bar{\theta})$ alltid skrivas som λ_α^a . Produkten av gluinosuperfältet $\bar{\lambda}\lambda$ i sig är inte en gaugeinvariant kvantitet vilket är önskat för att möjligtvis kunna få något observerbart. Med syftet av att bilda en gaugeinvariant storhet från gluinosuperfältet kan man härleda den så kallade superfältstyrkan W_α . För att få λ_α^a och W_α på samma form kan man uttrycka den i en bas av generatorer för en algebra som $W_\alpha = w_\alpha^a T_a$. Precis som för λ_α^a ignoreras koordinaterna i superrummet då de är irrelevanta för arbetet. Relationen mellan gluinosuperfältet och superfältstyrkan är inte nödvändig för förmodan men för den intresserade

läsaren beskrivs den i [8].

W_α själv är inte gaugeinvariant men under godtycklig gaugetransformation med generatorer T_a transformerar fältet som följande:

$$W_\alpha \rightarrow e^{i\beta^a T_a} W_\alpha e^{-i\beta^a T_a}. \quad (2.5)$$

Betrakta nu gaugetransformationen av $\text{Tr}(W^\alpha W_\alpha)$.

$$\begin{aligned} \text{Tr}(W^\alpha W_\alpha) &\rightarrow \text{Tr}(e^{i\beta^a T_a} W^\alpha e^{-i\beta^a T_a} e^{i\beta^a T_a} W_\alpha e^{-i\beta^a T_a}) \\ &= \text{Tr}(e^{i\beta^a T_a} e^{-i\beta^a T_a} W^\alpha W_\alpha) = \text{Tr}(W^\alpha W_\alpha). \end{aligned} \quad (2.6)$$

Även om superfältstyrkan inte är invariant under gaugetransformationer ser man att $\text{Tr}(W^\alpha W_\alpha)$ är gaugeinvariant. Detta gaugeinvarianta superfält kallas för glueballsuperfältet². Glueballsuperfältet, S , definieras:

$$S \propto \text{Tr}(W^\alpha W_\alpha) = 2\text{Tr}(W_1 W_2). \quad (2.7)$$

Enligt CDSW-förmodan genererar polynom av S , S^k , för något k , den så kallade kirala ringen, se bilaga A för en avgränsad definition. Dessutom gäller $S^h = 0$ för den klassiska kirala ringen, notera att det är detta mjukvaran ska bevisa för en given grupp. När den klassiska ringen deformeras till den kvantmekaniska ringen ändras sambandet till

$$S^h = \Lambda^{3h}, \quad (2.8)$$

där Λ är en konstant med enhet massa, denna deformation är dock utanför ramen för detta arbete. Givet $\langle S^h \rangle = \langle S \rangle^h$ och $\langle \Lambda^h \rangle = \Lambda^h$ från [9] blir vakuumbäntningsvärdet $\langle S \rangle = \Lambda^3$ nollskilt. Vi påminner oss om vakuumbäntningsvärdet för $\bar{q}q$ som nämndes i avsnitt 2.1 och liknelsen mellan dessa, $\bar{q}q$ gäller för kvarkar medan S gäller för gluinon.

²För det faktiska glueballsuperfältet saknas en faktor, $-\frac{1}{32\pi^2}$. Denna är dock irrelevant för CDSW-förmodan då proportionalitet är tillräckligt.

Kapitel 3

Teori

I detta avsnitt går vi igenom grundläggande teori. Först beskrivs några egenskaper och räkneregler hos grassmanntal. Efter detta hanteras det tyngre ämnet liegrupper och liealgebror. Under detta beskrivs strukturkonstanter, vilket är det som bestämmer våra ekvivalensrelationer. För att förstå hur vi explicit beräknat strukturkonstanterna behövs även teori om baser och rötter. Avslutningsvis beskriver vi vad det duala Coxeter-talet h i CDSW-förmodan är.

3.1 Några egenskaper hos grassmanntal

Låt $\{\theta_i\}$ vara en mängd grassmanntal, $i = 1, \dots, n$ där n är godtycklig. Grassmanntal är speciella i det att de är antikommutativa. Detta innebär att de uppfyller följande relation

$$\theta_i \theta_j = -\theta_j \theta_i \quad \forall i, j. \quad (3.1)$$

Antikommutativiteten innebär även att $\theta_i^2 = 0$. Dock kommuterar de med vanliga komplexa tal.

$$\theta_i z = z \theta_i, \quad z \in \mathbb{C}. \quad (3.2)$$

Grassmanntal används ofta inom partikelfysik, detta är på grund av antisymmetrin för fermioner från Paulis uteslutningsprincip. Specifikt för supersymmetri framkommer de ofta upp på grund av Grassmann-riktningarna som finns i ett superrum som nämndes i avsnitt 2.2. Dessa riktningar är de fermiska dimensionerna och relaterar även till Pauli-principen. Denna ytliga förklaring är mer än tillräcklig för följande arbete, läsaren med vidare intresse hänvisas till [10].

3.2 Liegrupper och liealgebror

Ett matematiskt fält som är centralt för att förstå CDSW-förmodan (samt vår numeriska lösningsmetod) är läran om liegrupper och liealgebror. Då det är ett stort område kan vi omöjligen infatta allting i detta avsnitt, utan tar här endast med de resultat från lieteori som krävs för att förstå arbetet. Den lieteori som presenteras nedan utgår från böckerna [11], [12] och [13], och dessa behandlar ämnet mer genomgående.

3.2.1 Liegrupper och deras fysiska relevans

En liegrupp är väsentligen en grupp vars gruppverkan motsvarar en kontinuerlig transformation. Ett klassiskt exempel på en sådan grupp är rotationsgruppen i två dimensioner (kallad $SO(2)$). En godtycklig rotation med en vinkel θ i två dimensioner av en vektor \mathbf{v} kan beskrivas genom ekvationen

$$\mathbf{v}' = R(\theta)\mathbf{v}, \quad (3.3)$$

där

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \quad (3.4)$$

I detta exempel utgör alltså mängden $\{R(\theta)\}$ elementen i liegruppen, som i detta fall har gruppkompositionen vanlig matrismultiplikation. Andra exempel på liegrupper är rotationer i högre dimensioner samt translationer i godtyckliga dimensioner.

Läran om liegrupper har stor relevans för fysiken eftersom det låter oss matematiskt beskriva de symmetrier som vårt universum följer. Detta har konsekvenser både för hur vi strukturerar våra fysiska teorier och för hur vi kan gå tillväga för att lösa fysiska problem. Till exempel vet vi att om vi flyttar hela universum 5 meter åt något håll eller roterar det med någon vinkel så kommer ingenting att ändras; universum är alltså både rotations- och translationsinvariant, vilket sätter krav på de fysiska teorier som vi använder oss av. Detta kan vi beskriva matematiskt genom liegrupper.

3.2.2 Från liegrupper till liealgebror

Även om liegrupper är viktiga för att beskriva symmetrier är de ofta matematiskt svåra att använda direkt. Om vi tar liegruppen som motsvarar rotationer i 3 dimensioner (kallad $SO(3)$) är gruppelmenten inte kommutativa, och matematiskt är det svårt att parametrisera dem som i 2 dimensioner.

Istället för att betrakta alla element i en liegrupp är det oftast lättare att betrakta infinitesimala element, eller "derivatorna" av gruppen. Om vi igen betraktar rotationsgruppen i två dimensioner, vet vi att en godtycklig rotation med vinkeln θ kan beskrivas enligt (3.4). Detta är dock ekvivalent med att rotera två gånger med vinkeln $\theta/2$, d.v.s $R(\theta) = R(\theta/2)R(\theta/2) = R(\theta/2)^2$. Mer generellt måste det vara, på grund av egenskaperna av rotationer, ekvivalent att rotera n gånger med vinkeln θ/n , d.v.s $R(\theta) = R(\theta/n)^n$. Om vi nu låter n gå mot oändligheten och använder första gradens taylorapproximation av \cos

och sin får vi

$$R(\theta) = (R(\theta/n))^n = \begin{bmatrix} \cos(\theta/n) & -\sin(\theta/n) \\ \sin(\theta/n) & \cos(\theta/n) \end{bmatrix}^n \approx \begin{bmatrix} 1 & -\theta/n \\ \theta/n & 1 \end{bmatrix}^n. \quad (3.5)$$

Låt nu

$$\mathcal{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \frac{\partial}{\partial \theta} R(\theta)|_{\theta=0}. \quad (3.6)$$

Denna matris kallas *generatorn* för $SO(2)$ och är derivatan av $SO(2)$ vid rotationsvinkeln $\theta = 0$. Om I är identitetsmatrisen får vi slutligen

$$R(\theta) = \lim_{n \rightarrow \infty} \begin{bmatrix} 1 & -\theta/n \\ \theta/n & 1 \end{bmatrix}^n = \lim_{n \rightarrow \infty} (I + \theta \mathcal{J}/n)^n = e^{\theta \mathcal{J}}, \quad (3.7)$$

där $e^A := \lim_{n \rightarrow \infty} (I + A/n)^n = 1 + A + A^2/2! + A^3/3! \dots$ är den så kallade *matrisexponentialen* för en matris A . Vi ser alltså att vi kan reducera komplexiteten av $SO(2)$ ner till en enda matris, generatorn, som vi sedan matrisexponentierar för att få ut alla element av $SO(2)$. Man kan därför i många fall undersöka gruppens egenskaper genom att endast studera generatorn. Notera dock att även matriser på formen $A = \lambda \mathcal{J}$ genererar elementen i $SO(2)$, bara med en extra faktor λ för vinkeln, ty de ger $e^{\theta A} = e^{\theta \lambda \mathcal{J}} = R(\lambda \theta)$. Generatorerna till $SO(2)$ ges således av alla reella multipler av matrisen \mathcal{J} .

Kraften av detta är inte helt uppenbar när man betraktar en så enkel grupp som $SO(2)$ men redan för $SO(3)$ minskar denna metod komplexiteten märkbart. I två dimensioner kan man väsentligen bara rotera kring en axel, vilket ger en enda generator \mathcal{J} . I tre dimensioner kan man dock rotera kring flera axlar, vilket ger många fler möjliga derivator. Dock går även detta att förenkla. Eftersom alla möjliga rotationsaxlar (och deras reella multipler) i 3 dimensioner ges av vektorrummet \mathbb{R}^3 kommer alla generatorer till $SO(3)$ (som är derivatorna kring alla rotationsaxlar) att vara isomorft¹ med \mathbb{R}^3 , och kommer alltså bilda ett vektorrum med vanlig matrisaddition, -subtraktion och multiplikation med skalärer. Speciellt innebär detta att vi kan hitta en bas för generatorerna till $SO(3)$, vilket t.ex kan tas som rotationerna kring koordinataxlarna. Basgeneratorn för en axel blir då en 2-dimensionell rotation i planet som spänns upp av resterande två axlar och får formen

$$L_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, L_y = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, L_z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (3.8)$$

Vi kan alltså i fallet för $SO(3)$ betrakta dess generatorer istället för att betrakta rotationerna i sig, som är svåra att parametrисera och icke-kommutativa. Då generatorerna utgör ett vektorrum, är de mycket enklare objekt att arbeta med. En sista bit av struktur som krävs utöver detta vektorrum för att kunna beskriva liegruppen fullständigt är hur gene-

¹För de som inte läst gruppteori innebär isomorfi att det finns en 1 till 1 korrespondens mellan två grupper, och att de väsentligen har samma struktur (grupperna är i någon mån "samma"). Exempelvis finns en isomorfi mellan \mathbb{R}^2 och \mathbb{C} , genom ekvivalensen $(x, y) \sim x + iy$, eftersom båda kommer bete sig likadant under komplex/vektor- addition och subtraktion. Här är alltså x, y reella tal och (x, y) betecknar en vektor.

ratorerna kommuterar med varandra. Även om generatorerna utgör ett vektorrum under matrisaddition och -subtraktion, får vi ut deras gruppverkan genom matris*multiplikation*. Under matrismultiplikation är inte generatorerna heller kommutativa, men generatorerna är *slutna* under kommutationsoperatoren $[\ , \]$, det vill säga om A, B är generatorer är även $C = [A, B] = AB - BA$ en generator. Vektorrummet som motsvarar generatorerna tillsammans med kommutationsrelationerna givet av $[\ , \]$ (som i lieteori kallas en *lieparantes*) kallas för en *liealgebra*, och är unik för varje liegrupp.

Även om elementen i en liealgebra i regel inte kommuterar innebär det inte att det inte finns vissa element som kommuterar med varandra. I själva verket kommer det för varje bas till en liealgebra alltid finnas en delmängd bestående av basgeneratorer som alla kommuterar med varandra. Antalet basgeneratorer r i den största sådana delmängd kallas liealgebras rank och betecknas $\text{Rank}(\mathfrak{g})$.

Eftersom varje liegrupp har en unik associerad liealgebra som i regel har en mycket enklare struktur än själva gruppen studeras ofta liealgebror för att bevisa satsen om liegrupper. Notationsmässigt brukar man beteckna liegrupper med stora bokstäver, och deras associerade liealgebror med små bokstäver i typsnittet fraktur. Detta gör också att terminologin för de två (matematiskt väldigt olika objekten) ofta förväxlas; exempelvis när man talar om dimensionen av en liegrupp (som enligt vanlig grupp teori bör vara oändlig) så menar man egentligen dimensionen av dess liealgebra.

3.2.3 Strukturkonstanter

Man kan vidare förenkla beskrivningen av en liealgebra och reducera strukturen till endast en uppsättning tal. Låt G vara en liegrupp och \mathfrak{g} dess liealgebra, och introducera en bas $T_a, a = 1, 2, \dots, \dim(G)$ till \mathfrak{g} . Varje element $X \in \mathfrak{g}$ (d.v.s varje generator) kan då skrivas som en linjärkombination

$$X = \sum_{a=1}^{\dim(G)} X^a T_a, \quad (3.9)$$

där X^a är konstanter.

Vidare vet vi att elementen i liealgebran relaterar till varandra med lieparantesen $[\ , \]$. Om vi explicit uttrycker kommutationen mellan två generatorer X, Y och använder oss av basen T_a ser vi den kan uttryckas i termer av basvektorerna:

$$[X, Y] = XY - YX = \sum_{a,b=1}^{\dim(G)} (X^a T_a)(Y^b T_b) - (Y^b T_b)(X^a T_a) = \quad (3.10)$$

$$= \sum_{a,b=1}^{\dim(G)} (X^a Y^b)(T_a T_b - T_b T_a) = \sum_{a,b=1}^{\dim(G)} X^a Y^b [T_a, T_b]. \quad (3.11)$$

Vidare, eftersom generatorerna är slutna under kommutationsoperatoren måste $[T_a, T_b] = Z$, där $Z \in \mathfrak{g}$ för alla basgeneratorer T_a, T_b . Men Z kan i sin tur uttryckas som en linjärkombination av basvektorer med koefficienter Z^c . För givna a, b kallar vi dessa Z^c för *strukturkonstanterna* f_{ab}^c som för den givna basen uppfyller

$$[T_a, T_b] = \sum_{c=1}^{\dim(G)} f_{ab}^c T_c. \quad (3.12)$$

Hela verkan av lieparantesen $[,]$ för en given liealgebra går alltså att reducera till uppsättningen strukturkonstanter f_{ab}^c . Notera dock att dessa fortfarande är basberoende, något som är väsentligt att ta hänsyn till om man ska utföra numeriska beräkningar.

3.2.4 Killingformen

I vissa sammanhang är det användbart att introducera en skalärprodukt på en liealgebra \mathfrak{g} till en liegrupp G , kallad *killingformen* eller en *kvadratisk form* på liegruppen/liealgebran. Denna bestämmer formen på S . Killingformen betecknas Tr och definieras på två basvektorer som

$$\text{Tr}(T_a, T_b) = \sum_{i,j=1}^{\dim(G)} f_{aj}^i f_{bi}^j, \quad (3.13)$$

och för två godtyckliga generatorer X, Y som

$$\text{Tr}(X, Y) = \sum_{a,b=1}^{\dim(G)} X^a Y^b \text{Tr}(T_a, T_b) = \sum_{a,b,i,j=1}^{\dim(G)} X^a Y^b f_{aj}^i f_{bi}^j. \quad (3.14)$$

Notera att även om man skriver ut killingformen i termer av strukturkonstanter (som är basberoende) är killingformen oberoende av bas.

3.2.5 Rötter och några av deras egenskaper

Låt G vara en liegrupp med liealgebra \mathfrak{g} , med dimension n och rank r . Till varje liegrupp kommer det finnas en uppsättning av $n - r$ stycken r -dimensionella vektorer kallade liegruppens *rötter*. Mängden av alla rötter till en liegrupp betecknas Φ och kallas för (liegruppens) *rotsystem*. Även dessa är unika till varje liegrupp.

Dessa rötter uppfyller flera egenskaper men de viktigaste för detta arbete är följande:

Enkla, positiva och negativa rötter

Givet ett rotsystem Φ , ifall vektorn α är en rot (d.v.s $\alpha \in \Phi$) är även vektorn $-\alpha$ en rot.

Vidare kan man till varje rotsystem Φ hitta en uppsättning av r stycken s.k *enkla rötter*

$\gamma_{(k)} \in \Phi, k = 1, 2, \dots, r$, som för varje godtycklig rot $\alpha \in \Phi$ uppfyller

$$\alpha = \sum_{k=1}^r \lambda_{\alpha}^k \gamma_{(k)}, \quad (3.15)$$

där λ_{α}^k är heltalskoefficienter som antingen alla är icke negativa, eller alla icke positiva. En mängd enkla rötter betecknas Δ . Givet en sådan uppsättning Δ kallas rötter som endast har icke negativa λ_{α}^k för *positiva rötter*, medan de med endast icke positiva λ_{α}^k kallas *negativa rötter*.

Rotkedjor

En annan användbar egenskap som rötterna i ett rotsystem Φ besitter är följande: Låt $\alpha, \beta \in \Phi$ vara rötter i samma rotsystem Φ . Om nu även $\beta + N\alpha, N = 1, 2, 3, \dots$ är en rot kommer $\beta + n\alpha$ också vara en rot, för alla heltal $0 \leq n \leq N$. Motsvarande gäller ifall $\beta - N\alpha$ är en rot.

Denna egenskap motiverar följande definition:

Definition 1. Låt $\alpha, \beta \in \Phi$ vara rötter i samma rotsystem Φ . Då definieras α -kedjan genom β som mängden $\{\beta - q\alpha, \beta - (q-1)\alpha, \dots, \beta, \dots, \beta + p\alpha\}$, där p, q är de största heltalen sådana att $\beta - q\alpha, \beta + p\alpha \in \Phi$. Vidare kallar vi en α -kedja för enkel ifall α är en enkel rot.

Vi kallar även två rötter α, β *kopplade med en δ -kedja* ifall α är del av δ -kedjan genom β (eller ekvivalent β är del av δ -kedjan genom α) och *enkelt kopplade* ifall de är kopplade med en enkel kedja.

3.2.6 Klassificering av liegrupper

Genom att använda sig av egenskaper av rotsystemen kan man visa att alla liegrupper kan brytas ner till olika kombinationer av s.k *enkla liegrupper*. De enkla liegrupperna består av 4 oändliga familjer, $SU(N), SO(2N), SO(2N+1), Sp(2N)$, som utgör de *klassiska* grupperna samt 5 *exceptionella* grupper, G_2, F_4, E_6, E_7, E_8 .

Eftersom alla liegrupper utgörs av kombinationer av dessa enkla liegrupper går det att bevisa flera universella egenskaper för liegrupper genom att endast bevisa dem för dessa.

3.2.7 Chevalley-basen

En bas som är speciellt användbar vid numeriska beräkningar av strukturkonstanter är Chevalley-basen. För varje liegrupp G av dimension n och rank r , med rotsystem Φ och enkla rötter Δ , delas generatorerna i Chevalley-basen upp i två olika sorter: r -stycken

ömsesidigt kommuterande generatorer, betecknade h , och $n - r$ stycken generatorer som inte är ömsesidigt kommuterande, betecknade e .

Varje generator i Chevalley-basen är associerad med en rot. För varje rot $\alpha \in \Phi$ finns en associerad icke-kommuterande generator e_α . Dock har endast enkla rötter associerade kommuterande generatorer, som på motsvarande sätt betecknas h_γ för varje enkel rot $\gamma \in \Delta$.

Generatorerna till Chevalley-basen uppfyller följande kommutationsregler:

$$[h_{\gamma(i)}, h_{\gamma(j)}] = 0 \quad (3.16)$$

$$[h_\gamma, e_\alpha] = 2 \frac{(\gamma, \alpha)}{(\alpha, \alpha)} e_\alpha \quad (3.17)$$

$$[e_\alpha, e_\beta] = c_{\alpha, \beta} e_{\alpha + \beta}, \text{ om } \alpha + \beta \text{ är en rot och } \alpha \neq -\beta \quad (3.18)$$

$$[e_\alpha, e_{-\alpha}] = \lambda_\alpha^j \frac{(\gamma(j), \gamma(j))}{(\alpha, \alpha)} h_{\gamma(j)}, \text{ där } \gamma(j) \text{ är de enkla rötterna.} \quad (3.19)$$

där (α, β) betecknar skalärprodukten av vektorerna α och β , och $c_{\alpha, \beta} = \pm(k + 1)$, där k är det största heltalet för vilket $\beta - k\alpha$ är en rot. Notera att Chevalley-basen inte är entydigt bestämd, då tecknen i (3.18) kan väljas på flera sätt.

Den största fördelen med Chevalley-basen är att all information om strukturkonstanterna kan fås från liegruppens rötter. Detta låter en undvika att utföra explicita beräkningar av matriskommutationer och att dela upp resultatet av sådana kommutationer i termer av den bas man valt. Vidare är alla strukturkonstanter i Chevalley-basen heltal vilka kan sparas numeriskt exakt.

3.2.8 Dualt Coxeter-tal

I förmodan används en konstant för en given liealgebra kallad duala Coxeter-talet. För vårt arbete är det inte viktigt var denna konstant kommer ifrån, men här beskrivs kortfattat hur den beräknas för det intresserade läsaren.

En given representation av en liealgebra kan specificeras med r antal heltal, där r är ranken av gruppen. Dessa tal kallas för representationens Dynkin-märke, exempelvis har den adjungerade representationen för $SU(3)$, som har rank 2, Dynkin-märket $\Lambda = (1 \ 1)$. Med Dynkin-märket kan man framställa en representations Dynkin-index l ,

$$l = (\Lambda, \Lambda + 2\delta) \quad (3.20)$$

där Λ är Dynkin-märket, och δ är en r -dimensionell vektor av ettor $\delta = (1 \ 1 \dots 1)$. $(,)$ är en inre produkt med en Cartan-matris. Dessa produkter är beräknade i tabeller i [14] där även Dynkin-märke och index förklaras mer djupgående. I den adjungerade representationen ger Dynkin-indexet det duala Coxeter-talet h med följande relation:

$$h = \frac{1}{2}l_{adj} = \frac{1}{2}(\Lambda, \Lambda + 2\delta) \quad (3.21)$$

Tabell 3.1 ger de relevanta konstanterna för de grupper som är intressanta.

Tabell 3.1: Tabellen visar, för de enkla kompakta liegrupperna G , dimension d , ranken r och dualt Coxeter-tal h .

G	$SU(N)$	$SO(2N)$	$SO(2N+1)$	$Sp(2N)$	G_2	F_4	E_6	E_7	E_8
d	$N^2 - 1$	$2N^2 - N$	$2N^2 + N$	$2N^2 + N$	14	52	78	133	248
r	$N - 1$	N	N	N	2	4	6	7	8
h	N	$2N - 2$	$2N - 1$	$N + 1$	4	9	12	18	30

Kapitel 4

CDSW-förmodan

Förmodan formulerades först i [1] och består egentligen av två delar. Den första delen säger att S genererar den kirala ringen¹. Detta är dock inget som kan bevisas numeriskt och har redan bevisats analytiskt för både de klassiska och exceptionella grupperna. Den andra delen, vilken är den vi försöker bevisa, presenteras utförligt nedan.

4.1 Generell formulering av CDSW-förmodan

Objektet av intresse för förmodan är S som vi definierar som $S = \text{Tr}(W_1 W_2)$, där Tr är en invariant kvadratisk form på liealgebran som betraktas. W_1 och W_2 är de så kallade superfältstyrkorna för gluinon, och rent konkret så är de matriser där varje matriselement är en summa av grassmanntal. Hur W_1 och W_2 ser ut beror på vilken liealgebra vi betraktar, men S kommer vara ett polynom av grassmanntal med två grassmanntal per term.

Vidare tar vi ekvivalensrelationerna

$$\{W_\alpha, W_\beta\} \sim 0, \quad \alpha, \beta = 1, 2. \quad (4.1)$$

där $\{W_\alpha, W_\beta\}$ är den s.k. *antikommutatorn* av W_α, W_β som ges av $\{W_\alpha, W_\beta\} = W_\alpha W_\beta + W_\beta W_\alpha$. Dessa kommer i grund och botten från egenskaper för *den kirala ringen* men vi tar de helt enkelt som postulat. Mer konkret ger dessa ekvivalensrelationer ett antal polynom av grassmanntal som är ekvivalenta med 0.

CDSW-förmodan säger att $S^{h-1} \approx 0$ och $S^h \sim 0$ under ekvivalensrelationerna, där h är det duala Coxeter-talet av liegruppen. Med andra ord påstår CDSW-förmodan att polynomet S^{h-1} inte kan skrivas som en summa av polynom som är ekvivalenta med 0 medan S^h kan det.

¹Den intresserade läsaren kan läsa mer om den kirala ringen i appendix A.

4.2 Specifik form av S och ekvivalensrelationerna

Vi nämnde i föregående avsnitt att W_1 och W_2 , och därmed S och ekvivalensrelationerna, beror på vilken liealgebra som betraktas. I detta avsnitt härleds mer explicita uttryck för dem så att de kan beräknas numeriskt.

Låt G vara en enkel kompakt liegrupp med tillhörande liealgebra \mathfrak{g} och en bas T_a , $a = 1, 2, \dots, \dim(G)$ för \mathfrak{g} . Då ges superfältstyrkorna av

$$W_1 = w_1^a T_a, \quad W_2 = w_2^a T_a. \quad (4.2)$$

där w_1^a och w_2^a är två uppsättningar med grassmanntal. Med detta uttryck för W_1 och W_2 kan S och ekvivalensrelationerna beräknas.

Ekvivalensrelationerna blir

$$\begin{aligned} 0 \sim \{W_\alpha, W_\beta\} &= W_\alpha W_\beta + W_\beta W_\alpha = w_\alpha^a w_\beta^b T_a T_b + w_\beta^b w_\alpha^a T_b T_a = \\ &= w_\alpha^a w_\beta^b T_a T_b - w_\alpha^a w_\beta^b T_b T_a = w_\alpha^a w_\beta^b [T_a, T_b], \end{aligned} \quad (4.3)$$

där vi använt oss av antikommutativiteten av grassmanntal. Eftersom alla T_a utgör en bas för \mathfrak{g} vet vi även att kommutatorn enligt ekvation (3.12) är lika med $f_{ab}^c T_c$, där f_{ab}^c är strukturkonstanterna av \mathfrak{g} i basen T_a . Ekvivalensrelationerna blir alltså

$$0 \sim \{W_\alpha, W_\beta\} = w_\alpha^a w_\beta^b f_{ab}^c T_c. \quad (4.4)$$

Vi noterar nu att eftersom T_c utgör en bas för \mathfrak{g} kan ekvation (4.4) endast stämma om koefficienterna för varje T_c är noll. Detta låter oss slutligen skriva ekvivalensrelationerna som

$$w_\alpha^a w_\beta^b f_{ab}^c \sim 0, \quad c = 1, 2, \dots, \dim(G), \quad \alpha, \beta = 1, 2. \quad (4.5)$$

För att beräkna S använder vi oss av att killingformen av två basgeneratorer är $\text{Tr}(T_a T_b) = f_{aj}^i f_{bi}^j$. Detta ger att

$$S = \text{Tr}(W_1 W_2) = w_1^a w_2^b \text{Tr}(T_a T_b) = w_1^a w_2^b f_{aj}^i f_{bi}^j. \quad (4.6)$$

Med något sätt att beräkna strukturkonstanterna f_{ab}^c och en mjukvara som kan hantera polynom av grassmanntal är det möjligt att programmatiskt hantera objekten i förmodan. I kapitel 5 presenteras vår metod för strukturkonstantsgenerering, och i kapitel 6 presenteras vår mjukvara för hantering av grassmannpolynom. Allt som återstår är sedan att bevisa förmodan.

Kapitel 5

Generering av strukturkonstanter

Första delen av vårt program genererar strukturkonstanterna för en godtycklig liegrupp. Vi valde att generera dessa i Chevalley-basen, eftersom dessa går att generera med en universell metod som inte kräver att man utför några explicita kommutationer. Istället kan vi härleda fram dessa genom att använda oss av liegruppens rötter.

5.1 Linjärkombinationer av enkla rötter som ger rötter

De enkla rötterna går att finna i tabell 5.1, och från dessa kan man generera resten av rötterna. Alla rötter i ett rotsystem kan enligt ekvation (3.15) fås genom linjärkombinationer av de enkla rötterna, där koefficienterna λ_α^k är heltal. Dock ger inte alla kombinationer av heltals- λ^k rötter, utan endast en del av dem.

För alla klassiska grupper utom $Sp(2N)$ och samtliga exceptionella grupper ges rötterna av de vektorer som är (heltals-)linjärkombinationer av de enkla rötterna, och som har samma längd som någon av de enkla rötterna.

Rötterna som tillhör $Sp(2N)$ har dock ett striktare krav på sig. Dess rötter kommer vara vektorer i ett r -dimensionellt vektorrum. Som man kan se i tabell 5.1 har $Sp(2N)$ två olika typer av enkla rötter: $r - 1$ -stycken på formen $\mathbf{e}_i - \mathbf{e}_{i+1}$ av längd $\sqrt{2}$, och en på formen $2\mathbf{e}_r$ av längd 2. Resten av rötterna till $Sp(2N)$ kan fås av vektorer som är (heltals-)linjärkombinationer av de enkla rötterna, och som antingen har längd $\sqrt{2}$ eller är på formen $2\mathbf{e}_k$.

5.2 Generering av rötter

Ett sätt man skulle kunna generera rötter på är genom att testa alla möjliga kombinationer av heltalskoefficienter för att se vilka som motsvarar rötter. Detta sättet är dock väldigt ineffektivt.

Givet en liealgebra \mathfrak{g} med rotsystem Φ och enkla rötter Δ , kan alla rötter enligt avsnitt 3.2.5 skrivas som $\sum_{k=1}^{\text{Rank}(\mathfrak{g})} \lambda^k \gamma_{(k)}$, där $\gamma_{(k)} \in \Delta$. Låt λ_{\max} beteckna det största värdet bland koefficienter λ^k som resulterar i rötter. För att hitta alla positiva rötter genom att testa alla möjliga kombinationer, måste man alltså variera alla λ^k mellan värdena $0, \dots, \lambda_{\max}$, d.v.s över $\lambda_{\max} + 1$ s.t värden. Då det finns $r = \text{Rank}(\mathfrak{g})$ stycken koefficienter ökar alltså antalet beräkningar exponentiellt med stigande rank.

Ett effektivare sätt att generera rötterna är via rotkedjor. I avsnitt 3.2.5 noterades det att alla rötter är kopplade till varandra via enkla rotkedjor. En metod som bygger på detta presenteras nedan:

5.2.1 Finna enkelt kopplade rötter

Givet en rot α tillhörandes ett rotsystem Φ med enkla rötter Δ kan man finna alla rötter som är enkelt kopplade till α genom att för varje enkel rot $\gamma \in \Delta$ utföra följande algoritm:

Steg 1 Låt $n = 1$.

Steg 2 Beräkna vektorn $\beta = \alpha + n\gamma$.

Steg 3 Ifall β är en rot enligt avsnitt 5.1, spara den. Sätt då $n = n + 1$ och repetera från **Steg 2**.

Steg 4 Låt $m = 1$.

Steg 5 Beräkna vektorn $\beta = \alpha - m\gamma$.

Steg 6 Ifall β är en rot enligt avsnitt 5.1, spara den. Sätt då $m = m + 1$ och repetera från **Steg 5**.

5.2.2 Generera rötter via enkla rotkedjor

Givet en uppsättning enkla rötter Δ (som för de enkla liegrupperna finns tabulerade bland annat i [12]) kan man generera alla positiva rötter med följande algoritm:

Steg 1 Låt I vara en ordnad lista som till en början endast innehåller de enkla rötterna, och låt $I(k)$ beteckna det k :te elementet i I . Sätt även $n = 1$

Steg 2 Låt $\alpha_n = I(n)$. Finn alla rötter β som är enkelt kopplade till α_n enligt avsnitt 5.2.1.

Steg 3 För varje funnen rot β , ifall β inte redan är ett element i I , lägg till β som sista elementet i I .

Steg 4 Ifall α_n inte är det sista elementet i I , öka värdet av n med 1 och repetera från **Steg 2**.

När man väl genererat alla positiva rötter kan man trivialt få fram de negativa genom att helt enkelt multiplicera varje positiv rot med faktorn (-1) .

5.3 Kommutationer

Givet alla enkla och positiva rötter kan strukturkonstanterna i Chevalley-basen konstrueras genom kommutationsreglerna i ekvationerna 3.16-3.19. Dock behöver man ta hänsyn till att välja tecknen i ekvation 3.18 på ett konsekvent sätt. Den algoritm som använts för att bestämma dessa tecken beskrivs i [15].

Tabell 5.1: Tabellen visar de enkla liegruppernas enkla rötter, samt deras rank. \mathbf{e}_i betecknar den i :te enhetsvektorn i en ON-bas. Tabellen är tagen från [12].

Grupp	Rank	Enkla rötter
$SU(N)$	$r = N - 1$	$\mathbf{e}_i - \mathbf{e}_{i+1}, i = 1, 2, \dots, r$
$SO(2N)$	$r = N$	$\mathbf{e}_i - \mathbf{e}_{i+1}, i = 1, 2, \dots, r - 1$ \mathbf{e}_r
$SO(2N + 1)$	$r = N$	$\mathbf{e}_i - \mathbf{e}_{i+1}, i = 1, 2, \dots, r - 1$ $\mathbf{e}_{r-1} + \mathbf{e}_r$
$Sp(2N)$	$r = N$	$\mathbf{e}_i - \mathbf{e}_{i+1}, i = 1, 2, \dots, r - 1$ $2\mathbf{e}_r$
G_2	$r = 2$	$\mathbf{e}_1 + \mathbf{e}_2$ $-2\mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3$
F_4	$r = 4$	$\mathbf{e}_1 - \mathbf{e}_2$ $\mathbf{e}_2 - \mathbf{e}_3$ \mathbf{e}_3 $\frac{1}{2}(\mathbf{e}_4 - \mathbf{e}_1 - \mathbf{e}_2 - \mathbf{e}_3)$
E_6	$r = 6$	$\mathbf{e}_i - \mathbf{e}_{i+1}, i = 3, 4, 5, 6$ $\mathbf{e}_6 + \mathbf{e}_7$ $\frac{1}{2}(\mathbf{e}_8 + \mathbf{e}_7 - \sum_{i=1}^6 \mathbf{e}_i)$
E_7	$r = 7$	$\mathbf{e}_i - \mathbf{e}_{i+1}, i = 2, 3, 4, 5, 6$ $\mathbf{e}_6 + \mathbf{e}_7$ $\frac{1}{2}(\mathbf{e}_8 + \mathbf{e}_7 - \sum_{i=1}^6 \mathbf{e}_i)$
E_8	$r = 8$	$\mathbf{e}_i - \mathbf{e}_{i+1}, i = 1, 2, 3, 4, 5, 6$ $\mathbf{e}_6 + \mathbf{e}_7$ $\frac{1}{2}(\mathbf{e}_8 + \mathbf{e}_7 - \sum_{i=1}^6 \mathbf{e}_i)$

Kapitel 6

Programdesign

I detta avsnitt beskrivs designen för programmet och de olika metoderna som vi testat för att försöka bevisa förmodan. Avsnitt 6.1 behandlar delen av programmet som möjliggör hantering av grassmanntal programmatiskt. Efter detta beskriver vi de två bevismetoder vi testat: projektionsmetoden i avsnitt 6.2 och hitta-en-specifik-uppsättning-metoden i avsnitt 6.3. Båda avsnitten inleds med ett enklare exempel och metoderna generaliseras sedan. Genomgående i kapitlet används Einsteins summationskonvention för att göra ekvationerna tydligare.

6.1 En klass för polynom av grassmanntal

Grassmanntal är som bekant inte vanliga reella eller komplexa tal; ingen mängd vanliga tal förutom 0 har egenskapen att $ab = -ba$. Således behövs något sätt att programmatiskt hantera dessa om man vill utforska egenskaper hos dem.

Vi är specifikt intresserade av egenskaper hos polynom av dessa grassmanntal, och har därför implementerat en klass för hantering av dessa polynom. Den består i huvudsak av två delar: `Term` och `Polynomial`. För att ge en enklare notation skriver vi i detta och nästkommande avsnitt U^a istället för w_1^a och V^a istället för w_2^a .

6.1.1 Termklassen

För att representera enskilda termer, det vill säga en produkt av ett antal grassmanntal, har vi implementerat en klass `Term`. Den består av dels en numerisk faktor och dels en tupel av heltal som representerar grassmanntalen.

Varje heltal representerar ett grassmanntal, och i och med att de är element i en tupel har de en bestämd ordning. För vårt specifika problem har vi valt att låta U^a representeras av $-a$ och V^a av a , men programmet antar ingenting sådant, hur man väljer att indexera sina grassmanntal lämnas helt upp till användaren.

Eftersom grassmanntalen indexerats med heltal har de en naturlig ordning, nämligen stigande index. Alla implementerade operationer på `Term` lämnar heltalstupeln sorterad, vilket gör att alla termer är på en standardform och kan därmed enkelt jämföras.

En `Term` kan skapas med en konstruktor med två argument: en tupel med index för grassmanntalen, och en numerisk faktor. Om tupeln inte redan är sorterad sorteras den och den numeriska faktorn byter tecken om sorteringen resulterade i en udda permutation av originallistan.

Genom implementation av Pythons så kallade *special methods* kan `Term`-objekt adderas, subtraheras, negeras och multipliceras med de vanliga operatorerna `+`, `-` och `*`. Notera att om termerna inte innehåller samma faktorer kommer addition och subtraktion resultera i ett objekt av typ `Polynomial`. Notera även att multiplikation använder en modifierad version av *merge*-algoritmen i [16] som även håller koll på pariteten för att sortera indexen.

Utöver dessa funktioner har vi även implementerat `__repr__()` som gör att `print(term)` fungerar som förväntat. Här tar vi vår notation för grassmanntal och representerar -1 som `U1` och 1 som `V1`. Nedan visas ett exempel på hur `Term` kan användas:

```
>>> t = Term(gms=(-3, -1, 1), factor=7)
>>> print(t)
+7 U3 U1 V1
>>> s = Term(gms=(2,))
>>> print(s)
+1 V2
>>> print(t*s)
+7 U3 U1 V1 V2
>>> print(s*t)
-7 U3 U1 V1 V2
```

6.1.2 Polynomklassen

För att representera polynom av grassmanntal har vi även implementerat en klass kallad `Polynomial`. Ett `Polynomial` kan konstrueras från en lista med termer, och kan som väntat adderas, subtraheras, negeras och multipliceras med operatorerna `+`, `-` och `*`. Ett `Polynomial` kan adderas med, och subtraheras från, både ett annat `Polynomial` och en `Term`. Det kan också multipliceras med ett annat `Polynomial`, en `Term` eller ett vanligt tal.

Man kan tänka sig att `Polynomial` borde implementeras som en lista med termer, men för att operationerna i föregående stycke ska vara snabba även för mycket stora polynom krävs en annan metod. Anledningen är att för att till exempel addera en `Term` till ett `Polynomial` måste det avgöras om polynomet redan har en term med de faktorerna. Om `Polynomial` implementeras som en lista av grassmanntal har detta tidskomplexitet $O(n)$, där n är antalet termer i polynomet.

Vi har valt att istället implementera `Polynomial` som en avbildningstabell med klassen

`dict` i Python. För varje term i polynomet innehåller avbildningstabellen en avbildning (kallas ofta på engelska *key-value pair*) med tupeln med index som söknyckel och faktorn för termen som värde. Det ger en tidskomplexitet $O(1)$ för att kontrollera om termen redan finns i polynomet.

Fastän `Polynomial` internt implementeras som en avbildningstabell kan konstruktorn ta både en vanlig lista med `Term`-variabler och en avbildningstabell enligt ovan. Det första är lättare för användaren, medan det andra går något snabbare.

Även för `Polynomial` är `__repr__()` implementerad så att `print(p)` fungerar. Nedan visas ett exempel på hur klassen kan användas:

```
>>> t = Term(gms=(-3, -1, 1), factor=7)
>>> s = Term(gms=(2,))
>>> u = Term(gms=(3,5), factor=3)
>>> p = Polynomial([s, t, u])
>>> print(p)
P( +1 V2 +7 U3 U1 V1 +3 V3 V5 )
>>> q = u + s
>>> print(q)
P( +3 V3 V5 +1 V2 )
>>> print(p*q)
P( +21 U3 U1 V1 V3 V5 +7 U3 U1 V1 V2 +6 V2 V3 V5 )
>>> print(p-q)
P( +7 U3 U1 V1 )
```

6.2 Projektion

Den första metoden som testades byggde på projektion. I stora drag går den ut på att översätta både S och polynomen som är ekvivalenta med 0, nollpolynomen, till vektorer och använda väl beprövade metoder från linjär algebra för att lösa matrisekvationer. I detta avsnitt börjar vi med ett enkelt exempel på hur det kan fungera och går sedan vidare och generaliserar metoden.

6.2.1 Ett enkelt exempel

För att få en känsla för hur projektionsmetoden fungerar tar vi ett enkelt exempel. S är ett polynom av grassmanntal, och i avsnitt 4.2 visades att varje term i S innehåller ett w_1^a och ett w_2^b , eller ett U och ett V . Detta innebär att termerna i någon potens av S , S^k kommer innehålla k stycken U :n och k stycken V :n. För exemplet begränsar vi oss till 6 stycken grassmanntal U^1, U^2, U^3, V^1, V^2 och V^3 , och tar ett P på en form som S^2 skulle kunna ha i en hypotetisk algebra:

$$P = U^1U^2V^1V^2 - U^1U^3V^1V^3. \quad (6.1)$$

Låt våra nollpolynom i denna hypotetiska situation vara

$$\begin{aligned} Z_1 &= U^1U^2 + U^1U^3 \\ Z_2 &= V^1V^2 + V^1V^3. \end{aligned} \tag{6.2}$$

Dessa är våra ekvivalensrelationer som motsvaras av $f_{ab}^c U^a U^b$, $f_{ab}^c U^a V^b$ och $f_{ab}^c V^a V^b$.

Att fråga om polynomet P är nollskilt givet ekvivalensrelationerna $Z_1 \sim 0$ och $Z_2 \sim 0$ är detsamma som att fråga om P ej kan skrivas som en linjärkombination av Z_1 och Z_2 med faktorer som är grassmanntal. I vårt enkla exempel kan vi genom att lägga pannan i djupa veck se att

$$P = V^1V^2Z_1 - U^1U^3Z_2, \tag{6.3}$$

och därmed att P måste vara ekvivalent med 0.

Frågan om huruvida ett objekt är en linjärkombination av en mängd andra objekt är ett väldigt vanligt problem, och det lättaste sättet att lösa det på är med linjär algebra. Skrivet i termer av vektorer och matriser, den linjära algebrans språk, blir problemet

$$\begin{bmatrix} | & | \\ Z_1 & Z_2 \\ | & | \end{bmatrix} \mathbf{x} = \begin{bmatrix} | \\ P \\ | \end{bmatrix} \text{ eller } Z\mathbf{x} = \mathbf{P}. \tag{6.4}$$

Denna kan sedan lösas med minstakvadratmetoden. Om den har en exakt lösning blir $\mathbf{P} - Z\mathbf{x} = 0$, annars blir den nollskild.

Det enda problemet som kvarstår är hur polynomen ska översättas till vektorer. För att använda vanliga implementationer av matrisekvationslösare måste våra vektorer och matriser bestå av reella tal, inte grassmanntal. Ett sätt att göra detta på är att låta vårt vektorrum vara alla möjliga polynom med 4 grassmanntal i varje term. Vi kan dessutom begränsa oss till endast polynom med lika många U och V eftersom vi vet att P är på den formen. Ett rimligt val av basvektorer är då alla entermopolynom, ty från de kan alla andra polynom konstrueras som linjärkombinationer av dessa och därmed vet vi att de spänner upp rummet. Vektorn $\mathbf{e}_1 = [1, 0, 0, \dots]^\top$ motsvarar alltså polynomet $U^1U^2V^1V^2$, $\mathbf{e}_2 = [0, 1, 0, \dots]^\top$ motsvarar $U^1U^2V^1V^3$ och så vidare.

Två saker är viktigt att notera med dessa basvektorer. För det första inkluderas ej några termer som är trivialt noll, så som $U^1U^1V^1V^1$. För det andra inkluderas ej heller termer som är permutationer av redan inkluderade termer: $U^1U^2V^2V^1$ är inte en distinkt basvektor från \mathbf{e}_1 utan skrivs som $[-1, 0, 0, \dots]^\top$.

Detta tillvägagångssätt gör dock att våra nollpolynom, Z_1 och Z_2 inte ligger i vårt vektorrum, ty de har endast 2 faktorer per term. Vi måste därmed använda nya nollpolynom som fås genom att multiplicera Z_1 och Z_2 med grassmanntal. De nya nollpolynomen blir

$$\begin{aligned} &V^1V^2Z_1, \quad V^1V^3Z_1, \quad V^2V^3Z_1, \\ &U^1U^2Z_2, \quad U^1U^3Z_2, \quad U^2U^3Z_2 \end{aligned} \tag{6.5}$$

Översätter vi dessa till vektorer som vi tar som kolumnerna i en matris, som i ekva-

tion (6.4), får vi

$$\begin{array}{l}
 U^1U^2V^1V^2 \\
 U^1U^2V^1V^3 \\
 U^1U^2V^2V^3 \\
 U^1U^3V^1V^2 \\
 U^1U^3V^1V^3 \\
 U^1U^3V^2V^3 \\
 U^2U^3V^1V^2 \\
 U^2U^3V^1V^3 \\
 U^2U^3V^2V^3
 \end{array}
 \begin{bmatrix}
 1 & 0 & 0 & 1 & 0 & 0 \\
 0 & 1 & 0 & 1 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 1 & 0 \\
 0 & 1 & 0 & 0 & 1 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0
 \end{bmatrix}
 \mathbf{x} =
 \begin{bmatrix}
 1 \\
 0 \\
 0 \\
 0 \\
 -1 \\
 0 \\
 0 \\
 0 \\
 0
 \end{bmatrix}
 \quad (6.6)$$

vilken har lösningen $\mathbf{x} = [1, 0, 0, 0, -1, 0]^\top$, samma lösning som i ekvation (6.3).

6.2.2 Generalisering

Här generaliseras problemet i föregående text till större polynom. Ett generellt polynom med lika många U och V och $2m$ grassmanntal per term kan skrivas

$$P = k_{a_1 a_2 \dots a_n b_1 b_2 \dots b_m} U^{a_1} \dots U^{a_m} V^{b_1} \dots V^{b_m}. \quad (6.7)$$

Våra nollpolynom ges av strukturkonstanterna för liealgebran genom

$$f_{bc}^a U^b U^c, \quad f_{bc}^a U^b V^c, \quad f_{bc}^a V^b V^c. \quad (6.8)$$

Precis som i föregående stycke multiplicerar vi dessa med grassmanntal så att de blir polynom med $2m$ faktorer per term och har lika många U och V . Då får vi de nya nollpolynomen

$$\begin{aligned}
 & (f_{bc}^a U^b U^c) U^{d_1} \dots U^{d_{m-2}} V^{e_1} \dots V^{e_m}, \\
 & (f_{bc}^a U^b V^c) U^{d_1} \dots U^{d_{m-1}} V^{e_1} \dots V^{e_{m-1}}, \\
 & (f_{bc}^a V^b V^c) U^{d_1} \dots U^{d_m} V^{e_1} \dots V^{e_{m-2}}.
 \end{aligned}
 \quad (6.9)$$

Som innan låter vi sedan alla polynom med en term, $2m$ faktorer per term och lika många U som V vara en bas och skriver om problemet som en matrisekvation.

Då m och liealgebrans dimension n ökar ökar dimensionen på vektorerna mycket snabbt. Det finns $\binom{n}{m}^2$ möjliga entermspolynom med m stycken U och m stycken V . Antalet nollpolynom blir därmed

$$N_Z = 2n \cdot \binom{n}{m-2} \cdot \binom{n}{m} + n \cdot \binom{n}{m-1}^2. \quad (6.10)$$

6.3 Att hitta en specifik uppsättning grassmanntal

En annan möjlighet som kan användas för att visa ena halvan av satsen, att $S^{h-1} \neq 0$, är att hitta en specifik uppsättning grassmanntal som uppfyller $S^{h-1} \neq 0$, och ekvivalensrelationerna. För att illustrera idén ges först ett exempel med vanliga reella tal.

För att visa att funktionen $f(x, y) = xy$ inte är identiskt lika med 0 givet kravet att $g(x, y) = x + y = 0$ räcker det med *ett* motexempel, en uppsättning x, y som löser $g(x, y) = 0$ och $f(x, y) \neq 0$. I detta fall skulle det kunna vara $x = 1, y = -1$. På samma sätt kan vi visa att $S^{h-1} \neq 0$ genom att hitta en specifik uppsättning grassmanntal som löser våra ekvivalensrelationer och ger $S^{h-1} \neq 0$.

6.3.1 Åter till det enkla exemplet

För att få en känsla för principen bakom hitta-specifik-uppsättning-idén återgår vi till det enklare exemplet i avsnitt 6.2.1, men denna gång med ett $+$ istället för $-$ i P :

$$P = U^1 U^2 V^1 V^2 + U^1 U^3 V^1 V^3. \quad (6.11)$$

Denna går inte att skriva som en linjärkombination av Z_1 och Z_2 . Ett sätt att visa detta på är att hitta en specifik uppsättning grassmanntal U^i och V^i som både löser $Z_1 = 0$ och $Z_2 = 0$ och gör att P är nollskild.

Men grassmanntal är ju som bekant inte vanliga reella tal, så hur ser denna uppsättning ut? I detta fallet skulle vi kunna ta fyra grassmanntal, kalla dem Ψ^1, Ψ^2, Ψ^3 och Ψ^4 . Vi kan sedan låta $U^1 = \Psi^1, U^2 = \Psi^2, U^3 = -\Psi^2$ och $V^1 = \Psi^3, V^2 = \Psi^4, V^3 = -\Psi^4$. Sätter vi in detta i Z_1 och Z_2 ser vi att de blir 0, och sätter vi in det i P får vi $2\Psi^1\Psi^2\Psi^3\Psi^4$ vilket är nollskilt.

6.3.2 Generalisering

Polynomet S^k har $2k$ faktorer i varje term. För att någon term ska vara nollskild måste vår specifika uppsättning innehålla minst $2k$ distinkta grassmanntal. Om den innehöll färre skulle varje term innehålla minst 2 av något grassmanntal, och på grund av att de är antikommutativa är ett grassmanntal multiplicerat med sig själv 0. Vi tar därmed $2k$ grassmanntal, kalla de $\Psi^1 \dots \Psi^{2k}$. Vidare kan varje U och V skrivas som godtyckliga linjärkombinationer av dessa $\Psi^1 \dots \Psi^{2k}$.

$$U^a = c_i^a \Psi^i, \quad V^a = d_i^a \Psi^i \quad (6.12)$$

och vårt mål är alltså att hitta något c och d som löser ekvivalensrelationerna och som gör att S^k är nollskild.

6.3.3 Gissa och testa

Det är omöjligt att testa alla möjliga c och d — det finns oändligt många av dem — så vi har testat en liten delmängd av dem. Kom ihåg att allt som behövs är ett exempel, så förhoppningen är att det finns ett sådant i den delmängd som testas.

Delmängden vi testat är den då varje U^a är lika med något Ψ^i . Ingen summa av flera Ψ^i och ingen numerisk faktor framför men flera olika U^a kan vara samma Ψ^i . Detta är en relativt liten delmängd, men det är den största som vi kan fullständigt testa.

6.3.4 Semi-analytiskt tillvägagångssätt

Ett bättre tillvägagångssätt vore att analytiskt bestämma möjligheterna för c och d som löser ekvivalensrelationerna. I detta avsnitt börjar vi härleda några egenskaper för dem. Eftersom formen på ekvivalensrelationerna är desamma för U och V kommer vi fortsättningsvis endast skriva om U , men samma resonemang kan repeteras för V . Ekvivalensrelationerna som innehåller U och V blandat återstår sedan att ta hänsyn till.

Tyvärr har vi inte kunnat utforska denna metod helt och hållet på grund av tidsbrist. Vi tror dock att den är lovande och att framtida forskning kan bygga vidare på det vi presenterar i detta avsnitt.

Grassmantalen U^a kan ses som en n -dimensionell vektor av grassmantal. Ekvationen $U^a = c_i^a \Psi^i$, eller $U = c\Psi$ på matrisform, ses då som en matrisekvation och c som en $n \times k$ -matris. Ekvivalensrelationerna blir även de matrisekvationer då vi låter f_{ab}^c representera en $n \times n$ -matris för något fixt c . De blir $U^\top [f_{ab}^c] U \sim 0$. För att underlätta notationen skriver vi matrisen f_{ab}^1 som A , f_{ab}^2 som B och så vidare. Sätter vi in $U = c\Psi$ i den första ekvivalensrelationen får vi $\Psi^\top c^\top A c \Psi \sim 0$. Sammantaget vill vi alltså finna en $n \times k$ -matris c med ett så stort k som möjligt som uppfyller

$$\begin{aligned}\Psi^\top c^\top A c \Psi &= 0 \\ \Psi^\top c^\top B c \Psi &= 0 \\ \Psi^\top c^\top C c \Psi &= 0 \\ &\vdots\end{aligned}\tag{6.13}$$

Betrakta först den första ekvivalensrelationen: $\Psi^\top c^\top A c \Psi \sim 0$, hur ser det mest generella c som löser denna ut? Antaget att grassmantalen i Ψ är oberoende så gäller att för att relationen ska vara uppfylld måste $c^\top A c$ vara symmetrisk. Men transponerar vi matrisprodukten får vi $(c^\top A c)^\top = c^\top A^\top c = -c^\top A c$ eftersom A är antisymmetrisk. De enda matris som är både symmetrisk och antisymmetrisk är nollmatrisen, och således är ekvivalensrelationen endast uppfylld om $c^\top A c = 0_{k \times k}$. Från detta följer att om c är en lösning kan en ny lösning erhållas genom multiplikation av godtycklig $k \times k$ -matris från höger. Detta kommer användas senare.

Eftersom A är reell och antisymmetrisk kan den *skew-diagonaliseras*, det vill säga skrivas

på formen

$$A = Z_A T_A Z_A^\top, \quad T_A = \begin{bmatrix} \lambda_1 \varepsilon & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 \varepsilon & \cdots & 0 & 0 & \cdots & 0 \\ & & \ddots & & & & \\ 0 & 0 & \cdots & \lambda_k \varepsilon & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \varepsilon = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (6.14)$$

där Z_A är en reell ortogonal matris och λ_i är positiva reella tal. Vi kan absorbera Z_A i c och fokusera på problemet $\tilde{c}^\top T_A \tilde{c} = 0$ med $\tilde{c} = Z_A^\top c$. Nu använder vi vår frihet att multiplicera c med godtycklig $k \times k$ -matris från höger för att låta \tilde{c}^\top vara på reducerad trappstegsform och dess rader vara normerade.

Det mest generella \tilde{c} på den formen som löser $\tilde{c}^\top T_A \tilde{c}$ är

$$\begin{bmatrix} \cos \theta_1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \sin \theta_1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cos \theta_2 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \sin \theta_2 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ & & \ddots & & & & & \\ 0 & 0 & \cdots & \cos \theta_k & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & \sin \theta_k & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ & & & & & & \ddots & \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (6.15)$$

vilket är relaterat till c genom $c = Z_A \tilde{c}$. Problemet är alltså hur vi ska välja dessa θ_i så att även de andra ekvivalensrelationerna löses.

Kapitel 7

Resultat och Diskussion

Det viktigaste resultatet av projektet var det numeriska beviset av CDSW-förmodans första del genom projektionsmetoden i avsnitt 6.2 för G_2 . Programmet gör detta på cirka 250 sekunder, vilket vittnar om att det är på gränsen till vad som är möjligt med detta tillvägagångssätt. Vi har även försökt hitta en specifik uppsättning grassmanntal, se avsnitt 6.3, för de större grupperna och har verifierat att S upphöjd med ranken är nollskilt men har inte kunnat säga något om högre potenser.

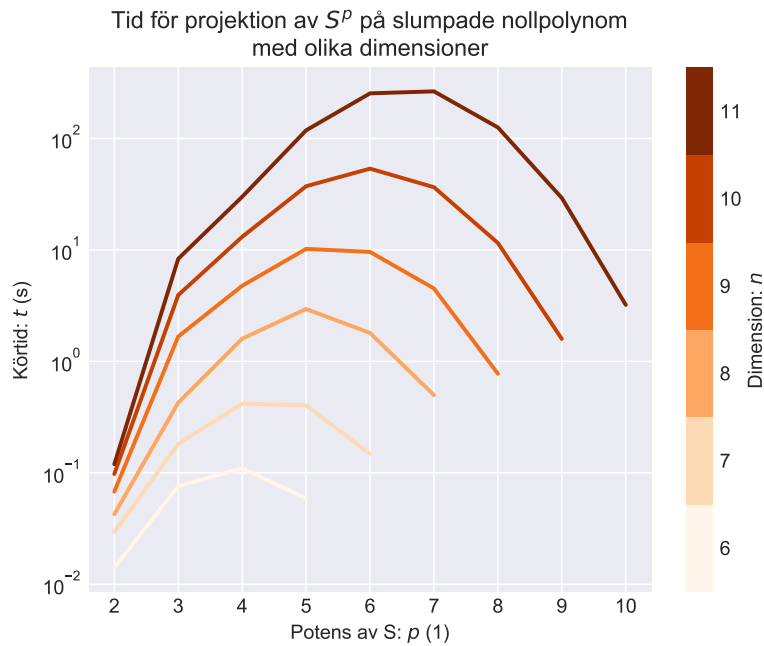
Den färdiga koden för generering av strukturkonstanter, hantering av polynom av grassmanntal och bevis av CDSW-förmodan finns på Github: github.com/hamdav/kandidatarbete-kod.

7.1 Resultat av projektionsmetoden

För gruppen G_2 kunde programmet säga att skillnaden mellan S^3 och projektionen av S^3 på rummet som spänns upp av nollpolynomen var nollskild; den innehöll 3177 termer med faktorer större än 0,1 varav den största faktorn var 7,54. Skillnaden mellan S^4 och projektionen av S^4 hade däremot inga termer med faktorer större än 10^{-6} , men flera termer med nollskilda faktorer i storleksordningar 10^{-7} och lägre. Att faktorerna ej blir exakt noll kommer av att detta är en numerisk lösningsmetod och därmed uppkommer flyttalsfel i beräkningarna.

Samma program kan inte användas för att bevisa satsen för större grupper än G_2 . Detta för att körtiden ökar mycket snabbt med dimensionen på algebran. För att kvantisera hur snabbt den ökar testas projektion på slumpmässigt genererade ekvivalensrelationer med olika dimensioner. Resultatet av detta finns i figur 7.1.

Från figuren ser vi att med varje steg dimensionen ökar höjs tidskurvan ungefär ett konstant steg på den logaritmiska skalan: körtiden ökar alltså exponentiellt när dimensionen ökas. Naivt räknat verkar körtiden öka med en faktor 100 då dimensionen ökar med 3. Den näst minsta exceptionella gruppen är F_4 och har en algebra med dimension 52, vilket med ovanstående räkning ger en körtid på 10^{27} sekunder, eller 10^{19} år.



Figur 7.1: Körzeit för projektion av S^p på slumpmässigt genererade nollpolynom i en hypotetisk algebra med dimension n . Dimensionen varierar från 6 till och med 11 och för varje dimension testas samtliga möjliga p .

Rutinerna för matrisekvationslösning kommer från `scipy.sparse`-modulen och är redan väl optimerade. Även på en kraftigare dator kommer den exponentiella tillväxten snabbt att överta de vinningar man kan göra på större beräkningskraft med en snabbare dator. Vi anser därmed att F_4 ej kan lösas inom en rimlig tidsgräns med denna metod, även om programmet teoretiskt sett skulle klara det tillslut.

7.2 Resultat av att försöka hitta en specifik uppsättning

Resultatet av att testa alla uppsättningar i avsnitt 6.3.3 var att sätta alla $U^a = 0$ förutom de som svarar mot mängden ömsesidigt kommuterande generatorer. Denna lösning var redan innan känd och bevisad av Witten[9], men att programmet verifierar detta visar att det fungerar som det ska.

Alla möjligheter av specifika uppsättningar på formen i avsnitt 6.3.3 testades för algebrorna G_2 och F_4 med programmet men ingen lösning hittades som gav nollskilt S^{h-1} . Detta innebär att denna enklare form var *för* enkel, och vi kan dra slutsatsen att en mer komplex form krävs.

7.3 Framtida forskning

Projektionsmetoden, trots att det är den som gett bäst resultat av de testade, visar mycket lite utrymme för förbättring. Dess exponentiella tidskomplexitet gör att optimering av algoritmen inte kommer vara tillräckligt för den näst minsta exceptionella gruppen F_4 . Detta i kombination med att metoden använder Pythonbibliotek som redan är mycket optimerade gör att vi inte tror vidare forskning om denna metod kommer ge bättre resultat.

Den metod som uppvisar störst potential är att hitta en specifik uppsättning grassmanntal som uppfyller förmodan. Speciellt det semi-analytiska tillvägagångssätt som presenterades i avsnitt 6.3.4 tror vi är lovande. Problemet med detta sätt är att man måste hitta vilka θ_i som ger lösningar till alla ekvivalensrelationer, vilket är svårt. Men om man lyckas skapa ett bra sätt att hitta dessa θ_i kan sedan alla möjligheter testas och det kan numeriskt avgöras vilka som ger ett nollskilt S^{h-1} .

Litteratur

- [1] Freddy Cachazo m. fl. "Chiral rings and anomalies in supersymmetric gauge theory". I: *JHEP* 12 (2002), s. 071. DOI: [10.1088/1126-6708/2002/12/071](https://doi.org/10.1088/1126-6708/2002/12/071). arXiv: [hep-th/0211170](https://arxiv.org/abs/hep-th/0211170).
- [2] M.A. Shifman, A.I. Vainshtein och V.I. Zakharov. "QCD and resonance physics. theoretical foundations". I: *Nuclear Physics B* 147.5 (1979), s. 385–447. ISSN: 0550-3213. DOI: [https://doi.org/10.1016/0550-3213\(79\)90022-1](https://doi.org/10.1016/0550-3213(79)90022-1). URL: <https://www.sciencedirect.com/science/article/pii/0550321379900221>.
- [3] Pietro Colangelo och Alexander Khodjamirian. "QCD sum rules, a modern perspective". I: *At The Frontier of Particle Physics* (2001-04), s. 1495–1576. DOI: [10.1142/9789812810458_0033](https://doi.org/10.1142/9789812810458_0033). URL: http://dx.doi.org/10.1142/9789812810458_0033.
- [4] Michael E. Peskin och Daniel V. Schroeder. *An Introduction to quantum field theory*. Reading, USA: Addison-Wesley, 1995. ISBN: 978-0-201-50397-5.
- [5] B. L. Ioffe. "Condensates in quantum chromodynamics". I: *Physics of Atomic Nuclei* 66.1 (2003-01), s. 30–43. ISSN: 1562-692X. DOI: [10.1134/1.1540654](https://doi.org/10.1134/1.1540654). URL: <http://dx.doi.org/10.1134/1.1540654>.
- [6] C. N. Yang och R. L. Mills. "Conservation of Isotopic Spin and Isotopic Gauge Invariance". I: *Phys. Rev.* 96 (1 1954-10), s. 191–195. DOI: [10.1103/PhysRev.96.191](https://doi.org/10.1103/PhysRev.96.191). URL: <https://link.aps.org/doi/10.1103/PhysRev.96.191>.
- [7] Stephen P. Martin. "A supersymmetry primer". I: *Advanced Series on Directions in High Energy Physics* (1998-07), s. 1–98. ISSN: 1793-1339. DOI: [10.1142/9789812839657_0001](https://doi.org/10.1142/9789812839657_0001). URL: http://dx.doi.org/10.1142/9789812839657_0001.
- [8] Riccardo Argurio, Gabriele Ferretti och Rainer Heise. "An introduction to supersymmetric gauge theories and matrix models". I: *International Journal of Modern Physics A* 19.13 (2004-05), s. 2015–2078. ISSN: 1793-656X. DOI: [10.1142/S0217751x04018038](https://doi.org/10.1142/S0217751x04018038). URL: <http://dx.doi.org/10.1142/S0217751x04018038>.
- [9] E. Witten. "Chiral Ring of Sp(N) and SO(N) Supersymmetric Gauge Theory in Four Dimensions". I: *Chinese Annals of Mathematics* 24.04 (2003-10), s. 403–414. ISSN: 0252-9599. DOI: [10.1142/S0252959903000402](https://doi.org/10.1142/S0252959903000402). URL: <http://dx.doi.org/10.1142/S0252959903000402>.
- [10] Alice Rogers. *Supermanifolds*. World Scientific, 2007. DOI: [10.1142/1878](https://doi.org/10.1142/1878). eprint: <https://www.worldscientific.com/doi/pdf/10.1142/1878>. URL: <https://www.worldscientific.com/doi/abs/10.1142/1878>.
- [11] Howard Georgi. *Lie algebras in particle physics: from isospin to unified theories*. CRC Press, 2018.

- [12] Anthony Zee. *Group theory in a nutshell for physicists*. Vol. 17. Princeton University Press, 2016.
- [13] Roger W. Carter. *Simple Groups of Lie Type*. John Wiley & Sons Ltd, 1972.
- [14] Robert Feger, Thomas W. Kephart och Robert J. Saskowski. "LieART 2.0 – A Mathematica application for Lie Algebras and Representation Theory". I: *Computer Physics Communications* 257 (2020-12), s. 107490. ISSN: 0010-4655. DOI: [10.1016/j.cpc.2020.107490](https://doi.org/10.1016/j.cpc.2020.107490). URL: <http://dx.doi.org/10.1016/j.cpc.2020.107490>.
- [15] R.B Howlett, L.J Rylands och D.E Taylor. "Matrix Generators for Exceptional Groups of Lie Type". I: *Journal of Symbolic Computation* 31.4 (2001), s. 429–445. ISSN: 0747-7171. DOI: <https://doi.org/10.1006/jSCO.2000.0431>. URL: <https://www.sciencedirect.com/science/article/pii/S0747717100904310>.
- [16] Robert Sedgewick. *Algorithms (4th Edition)*. Addison-Wesley Professional, 2011-03. ISBN: 032157351X. URL: <https://www.xarg.org/ref/a/032157351X/>.

Bilaga A

Kiral Ring

Den fulla definition inom kvantfältteori för en kiral ring är överdriven för följande arbete och därför kommer en begränsad definition av en kiral ring ges vilket är tillräckligt för att formulera förmodan. En ring kan ses som en grupp med en extra operand, multiplikation. Ett exempel av en ring följer den formella definition. Begreppet är viktigt för förmodan eftersom den är formulerad med polynom av grassmanntal, specifikt en ring av gaugeinvarianta polynom av grassmanntal.

Definition 2. Om R är en **ring** har den två operander, addition, $+$, och multiplikation, \cdot , för alla $a, b, c \in R$ gäller:

- Under addition.
 - $a \in R \implies -a \in R$, det existerar en invers.
 - $a + 0 = a$, det existerar ett identitetslement.
 - $a + (b + c) = (a + b) + c$, associativitet.
 - $a, b \in R \implies a + b \in R$, sluten.
 - $a + b = b + a$, addition är kommutativ.
- Under multiplikation.
 - $a \cdot (b \cdot c) = (a \cdot b) \cdot c$, associativitet.
 - $a, b \in R \implies a \cdot b \in R$, sluten.
- $a \cdot (b + c) = a \cdot b + a \cdot c$, $(b + c) \cdot a = b \cdot a + c \cdot a$, multiplikation distribuerar över addition.

Mängden av alla polynom med komplexa koefficienter har just en ringstruktur, för att ge läsaren mer intuition redogörs detta explicit. Betrakta mängden av alla polynom med komplexa koefficienter $\mathbb{C}[x]$. För alla $p \in \mathbb{C}[x]$ existerar givetvis en additiv invers, $-p \in \mathbb{C}[x]$ och identitetslement är det triviella polynomet 0. Polynom är associativa i addition och multiplikation, $p, q, h \in \mathbb{C}[x]$, $p + (q + h) = (p + q) + h$ och $p \cdot (q \cdot h) = (p \cdot q) \cdot h$. Alltså skapar komplexvärda polynom en ring.

Definition 3. En **delring** S till en ring R är en delmängd av R och en ring med avseende på samma operader, \cdot och $+$.

Definition 4. I kallas för ett **ideal** till ringen R om:

- I är en delring till R .
- För alla $i \in I$ och $r \in R$ gäller $r \cdot i \in I$ och $i \cdot r \in I$

Sats 1. Givet en ring, R och ett ideal, I är R/I en ny ring, kallad kvotringen, där givet $a, b \in R$ gäller $a \sim b$ om och endast om det existerar ett $i \in I$ så att $a + i = b$, där $+$ är addition i ringen.

Givet en liealgebra \mathfrak{g} av dimension G , låt R vara den ring som genereras av polynom i grassmantalen w_α^a . Idealet I till R genereras av de så kallade ekvivalensrelationerna:

$$I = \{\alpha, \beta, c \mid \alpha, \beta = 1, 2 \text{ och } c = 1, \dots, G \text{ där } w_\alpha^a w_\beta^b f_{ab}^c \in I\}, \quad (\text{A.1})$$

där f_{ab}^c är strukturkonstanterna för \mathfrak{g} . På grund av ett ideals egenskaper ger detta ekvivalensrelationerna $w_\alpha^a w_\beta^b f_{ab}^c \sim 0$. Den begränsade definitionen av en kirala ring som är lämplig för CDSW-förmodan kan nu ges. Den kirala ringen för en given liealgebra \mathfrak{g} är ringen $(R/I)^{\text{in}}$ där $(\)^{\text{in}}$ är den gaugeinvarianta delringen av R/I .

INSTITUTIONEN FÖR FYSIK
CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige 2021
www.chalmers.se



CHALMERS