

# DSL för linjär algebra

## Projektplan

Filip Nordmark, Sebastian Sjögren, Adam Eliasson,  
Linus Sundkvist, Daniel Nikolaev

28 februari 2022

# Innehåll

<b>1</b>	<b>Bakgrund</b>	<b>1</b>
<b>2</b>	<b>Syfte</b>	<b>2</b>
<b>3</b>	<b>Problem/Uppgift</b>	<b>2</b>
<b>4</b>	<b>Avgränsningar</b>	<b>3</b>
<b>5</b>	<b>Metod</b>	<b>3</b>
5.1	Genomförande . . . . .	3
5.2	Utvärdering . . . . .	5
<b>6</b>	<b>Etiska aspekter</b>	<b>5</b>
<b>7</b>	<b>Tidsplan</b>	<b>6</b>
	<b>Referenser</b>	<b>8</b>

# 1 Bakgrund

Ett grundläggande verktyg som datavetare bör kunna använda sig av är matematik. Det är klart att datavetenskap och matematik hör ihop då grunden för programmering har sina rötter inom diskret matematik. Trots detta finns det fortfarande de som förespråkar att matematik inte har någon betydelse i att kunna lära sig programera. Avsaknad av matematisk kunskap skulle dock begränsa nivån av analys som utvecklare kan utnyttja för att bevisa korrektheten av mjukvara. Därav är tesen för detta projekt att göra matematiken mer begriplig för datavetare och programmerare i allmänhet.

Trots att datavetenskap är en matematisk disciplin, främst inom diskret matematik, så saknar många datavetare kompetens inom områden som exempelvis linjär algebra eller analys av reella och komplexa tal. Tentamenstatistiken i kursen Linjär algebra [1] visar att det genomsnittliga antalet godkända studenter som går det datatekniska programmet på Chalmers tekniska högskola är 68.3%. I första anblicken ser detta inte ut som en dålig andel men i jämförelse med andra program på Chalmers, t.ex. industriell ekonomi där i genomsnitt 91.8% av studenterna klarar linjär algebra tentorna, ser man att det finns rum för förbättring.

Om man tolkar matematiken inom kontexten av *domänspecifika språk* [2], också förkortat *DSL*, kan man överbrygga klyftan mellan datavetenskap och andra matematiska discipliner. Det datavetenskapliga perspektivet ger en informativ synvinkel på matematik och i detta avseende kan utbildningen av matematik generellt ta nytta av tillvägagångssättet från datavetenskap [3]. Genom att implementera en mjukvarubaserad abstraktion av den matematiska syntaxen så kan man få en bättre insikt kring den tvetydiga semantiken inom den klassiska matematiken.

Inom utvecklingen av ett DSL, fokuseras det på att modellera en enda domän medan generella programmeringsspråk kan tillämpas i flera olika domäner. Fördelen med att specialisera inom en domän är att syntax och semantik kan göras mer lämpad för att lösa specifika problem. Generellt sätt är domänspecifika språk mindre komplexa och de kan variera i sin omfattning beroende på vilken domän de modellerar. Ett exempel på ett väldigt enkelt men kraftfullt domänspecifikt språk är reguljära uttryck som används för att parse och validera strängar.

Det finns många andra ämnen inom matematiken än diskret matematik som har användningsområden inom datavetenskap. Linjär algebra spelar bland annat en fundamental roll i datorgrafik, där matris-operationer används när man ska rendera grafiken i en 3D eller 2D miljö. Dessutom används också linjär algebra inom maskinlärning var man bland annat använder sig av linjär regression. På grund av bredden av användningsområdena i linjär algebra för datavetenskap, ska detta projekt fokusera på att kommunicera det matematiska språket kring linjär algebra i form av ett DSL för att demonstrera hur matematik från datavetenskapens perspektiv kan underlätta förståelsen av matematiska koncept för datavetare.

Detta projekt baseras framför allt på den valbara kursen *Domain Specific Languages of Mathematics (DSLofMath)* [3] som undervisas för datavetare och matematiker på Chalmers tekniska högskola och Göteborgs universitet. Kursens syfte är att presentera matematik utifrån ett datavetenskapligt perspektiv genom funktionell programmering. Mycket av metodiken som finns i den kursen kommer att inkorporeras i detta projekt. Detta innebär att ett DSL implementeras och utvecklas utifrån en matematisk domän baserat på ett funktionellt programmeringsspråk.

## 2 Syfte

Projektets syfte är att skapa ett domänspecifikt språk för linjär algebra. Språket ska utvecklas med fokus på domänens lagar så att korrekthet i största mån kan garanteras. För att uppnå detta kommer projektet därför att specificera och bevisa egenskaper hos uttryck i språket. Dessa specifikationer och bevis kan ses som ett komplement till den huvudsakliga produkten: det domänspecifika språket. Förhoppningen är att språket ska kunna användas till att lösa konkreta problem, öka användarens förståelse av linjär algebra och som ett hjälpmedel för bevis av matematiska satser.

## 3 Problem/Uppgift

Uppgiften är att utveckla ett domänspecifikt språk som kan ge en pedagogisk insyn i den matematiska grenen linjär algebra från ett datavetenskapligt perspektiv. Utvecklingen ska ske i de funktionella programspråken Haskell [4] och Agda [5]. Uppbyggnad av det domänspecifika språket syftar till att öka förståelsen inom linjär algebra för projektmedlemmarna samt att ge framtida studenter, eller andra intresserade mer material att arbeta med. Det ska genomföras genom specifikation av introducerade begrepp, lagar och uppmärksammande av syntax och typer.

Processen att utveckla domänspecifika språk för ett ämne är en lång process bestående av flertal olika moment. Uppbyggnaden av projektet delas därav upp i följande olika delproblem/uppgifter. Genomförande av samtliga deluppgifter ska resultera i en slutprodukt vilket skall vara ett komplement till linjär algebra ur ett datavetenskapligt perspektiv.

### Delproblem/uppgifter

**Analys av linjära algebrans grund:** För en holistisk bild över vad som behöver inkluderas inom DSL implementeringen behövs en litteraturgenomgång av linjär algebra. Det inkluderar studerande av vektorer, matriser, linjära rum, linjära avbildningar och linjära ekvationssystem.

**Matematiken i funktionell programmering:** Med vad som studeras fram i linjära algebra analysen behöver matematiken kodas fram i funktionell programmering. Utveckling av en miljö för att möjliggöra exekvering av generella matematiska operationer samt specialiserade operationer för grenen linjär algebra. Det inkluderar framförallt uppbyggnad av nödvändiga matematiska typer, specifikationer och lagar.

Ytterligare, finns det väsentliga skillnader mellan matematik och dess tolkning i mjukvara. Datorer har en exekverings- och platskostnad som begränsar användning av vissa matematiska objekt. Till exempel kan inte reella tal implementeras exakt utan måste approximeras – vilket motsäger deras matematiska definition. Projektet kommer därför att behöva göra vissa begränsningar så att språket ska vara användbart. Dessa avvikelser behöver göras med avvägning att den: semantiskt sätt passar domänen; inte kompromissar korrekthet; och har en rimlig komplexitet.

**Testning och bevisande av korrekthet:** Testande och bevisande av den matematiska korrektheten i det utvecklade domänspecifika språket. Detta innebär att den utvecklade miljön behöver inkludera ett demo så att den linjära algebrans matematiska korrekthet kan visas.

**Skapandet av en interaktiv upplevelse:** Utveckla ett kompletterande material, där framtida studenter eller andra intresserade själva kan utföra tester och bevis med de framtagna modulerna för att fördjupa sin egen förståelse av linjär algebra.

## 4 Avgränsningar

Projektet ska vara avgränsat till grundläggande principer inom linjär algebra, vilket inkluderar: vektorer, matriser, egenvärden, linjära rum, linjära avbildningar och linjära ekvationssystem. Böckerna *Linear Algebra Done Right* av Sheldon Axler [6] samt *Linjär algebra* av Gunnar Sparr [7] kommer att följas. Därmed avgränsas projektet till matematiken som tas upp i dessa böcker. Det betyder inte att allt material i böckerna kommer att representeras, utan de ger riktlinjer för vad och i vilken ordning det domänspecifika språket ska utvecklas.

Specifikt kring matematikens implementation kommer vårt program att hantera godtyckligt stora matriser, men inte oändliga matriser och vektorer. Det här görs eftersom hantering av oändliga matriser och vektorer är för komplext och tidskrävande att implementera, speciellt kring operationer som till exempelvis matrismultiplikation. Linjär algebra är ett brett ämne där mycket kan implementeras och därför borde det undvikas att överkomplicera vissa delar av det för att möjliggöra en bredare implementation.

## 5 Metod

Projektet delas upp i två huvudmoment som kommer att beskrivas i denna sektion. Det kommer omfatta genomförandet av arbetet mot de uppsatta delmålen från sektion 3 samt hur utvärdering kommer göras av hur framgångsrikt projektet är.

### 5.1 Genomförande

Fokus för projektet är korrekthet och utifrån detta ska ett flertal läroböcker inom området linjär algebra analyseras. Detta är för att säkerställa att gruppen har tillräcklig kunskap gällande vilka typer och matematiska lagar som gäller för att kunna realisera matematiken ur ett datatekniskt perspektiv.

Därefter ska det utvecklas ett DSL för linjär algebra i Haskell utifrån litteraturstudien. Följandet av litteraturen säkerställer att matematiken byggs upp från dess grund och därav sker utvecklingen i en strukturerad ordning. Primärt ska utveckling ske i Haskell och domänspecifika språkets korrekthet testas med QuickCheck [8] i den mån det är möjligt. I komplement till testandet i Haskell ska en del matematiska lagar bevisas i Agda. Därefter ska det utvecklas en interaktiv miljö för att interagera med det domänspecifika språket.

Mer specifikt utifrån de tidigare etablerade deluppgifterna ska projektet utgå från punkterna nedan.

- **Analys av linjära algebras grund:**

I första hand ska Sheldon Axlers bok *Linear Algebra Done Right* [6] studeras, både innan och kontinuerligt under utvecklandet. Detta för att säkerställa en bra förståelse för matematiken när det implementeras som kod. Det här är ytterst viktigt för att spara tid långsiktigt genom att minimera grundläggande misstag kring matematiken.

I andra hand ska även boken *Linjär Algebra* av Gunnar Sparr [7] studeras. Det behövs ifall oklarheter skulle uppstå från läsandet av den förstnämnda boken och fler källor behövs.

- **Matematiken i funktionell programmering:**

Utöver inläsning av matematiken ska boken *Domain-Specific Languages of Mathematics* [9] från kursen *DSLofMath* [3] studeras. Detta för att få en konkret bild av hur matematik kan överföras till Haskell.

En viktig del av att designa ett DSL är att hitta bra datatypsrepresentationer för de grundtyper man är intresserad av. Inom linjär algebra motsvarar dessa vektorer och matriser, med avgränsningen i detta projekt på vektorer och matriser av givna dimensioner. Ett fokusområde är därför hur Haskells typ-system kan nyttjas så att vektorers dimensioner är väl integrerade i det domänspecifika språket. Exempelvis bör addition av vektorer med olika dimensioner inte kompilera och därav inte påverka språkets korrekthet genom exekveringsfel.

- **Testning och bevisande av korrekthet:**

Testningen av det domänspecifika språket ska ske med QuickCheck. För testning i QuickCheck behövs det sättas upp egenskaper och generatorer, där egenskaper är de lagar som ska testas och generatorer används för att testa slumpmässigt genererade argument. Testandet ska ske på de lagar inom linjär algebra som implementeras och med slumpmässig generering av matriser och vektorer.

Fortsättningsvis ska bevisande av lagar ske i Agda. Detta ska göras för ett utval av lagar som definierats i Haskell. Det här görs eftersom Agda använder sig av propositioner som typer vilket resulterar i att de bevis som ska testas konstrueras av språket under typ kontrollen. Det här ger ett starkare bevis än de generade funktions argument som produceras av quickcheck.

- **Skapandet av en interaktiv upplevelse:**

Det slutliga steget är att med det utvecklade domänspecifika språket möjliggöra interaktion för användare. Ett bibliotek av utvalda filer ska sammanställas och laddas upp som öppen källkod. Filerna kommer inkludera nödvändiga moduler, som kombinerat ger möjlighet för användaren att exekvera funktionalitet från matematiken beskrivet i det domänspecifika språket. Biblioteket ska vara lätt tillgänglig för alla användare att ladda ner till och köra på sin egen enhet.

För att göra processen än mer pedagogisk och materialet lättanvänt för samtliga användare ska filerna även inkludera handledning. Handledningen ska vara i form av kommentarer i källkoden som förklarar vad som kan utföras i respektive filer kombinerat med exempel för att förenkla processen för användaren.

## 5.2 Utvärdering

Under projektets gång ska en kontinuerlig dialog föras med handledare Patrik Jansson, examinator för kursen DAT326/ DIT982 *DSLofMath* på Chalmers och GU[3]. All källkod ska även vara tillgänglig för handledaren under arbetets gång att fritt analysera och kommentera. Genom denna kontinuerliga respons på arbetet kan under projektets gång flertal mindre utvärderingar genomföras av hur väl arbetet mot delmålen går. Vid projektets konklusion ska arbetet utvärderas genom följande process.

Inkluderar domänspecifika språket de matematiska modeller som avsetts vara inkluderade enligt litteraturstudien och avgränsningarna? Det vill säga identifieras de grundtyper och koncept nödvändiga för ett DSL inom linjär algebra enligt de uppsatta avgränsningarna. Utvärdering ska huvudsakligen ske genom två moment. Först och främst genom den kontinuerliga dialogen under projektets gång med handledaren där det kontrolleras att all avsedd matematik inkluderas. Moment två av utvärderingen ska inkludera testning och bevisande av matematikens korrekthet. Utförandet av det delmålet bygger på att den implementerade matematiken omfattar tillräcklig och korrekt funktionalitet. Därav kan det avgöras hur väl språket inkluderar nödvändig matematik från resultaten av testning och bevisande av dess korrekthet.

Denna utvärdering ska ske tillsammans med en granskning av vad de matematiska lagarna säger från litteratur analysen. Syftet med testningen och bevis av matematiska lagarna är att påvisa det domänspecifika språkets korrekthet. Desto fler egenskaper och satser inom linjär algebra som kan testas och bevisas vara korrekta inom det domänspecifika språket, desto bättre verifieras dess korrekthet.

Utvärdering av den interaktiva upplevelsen ska ske från respons av användare. Tidig tillgänglighet till materialet ska ges till en grupp av frivilliga testare. De kommer få använda materialet fritt och bygga en bild av hur användbart de anser det vara. Testarna ska förfrågas att besvara ett anonymt frågeformulär där svaren kan hjälpa ge en indikation över hur väl den interaktiva upplevelsen fungerar och ifall det är några förändringar som behöver göras.

Då utvärderingen av deluppgifterna är färdigställd, ska en generell analys av arbetet genomföras. Den slutliga produkten ska jämföras med syftet. Produkten och utvärderingen av arbetet ska sedan presenteras på projektets slutredovisning, se figur 1.

## 6 Etiska aspekter

Detta material är designat för de med viss erfarenhet inom programmering och linjär algebra. Därför kommer materialet att gynna vissa mer än andra, vilket skapar en ojämn möjlighet kring användbarheten.

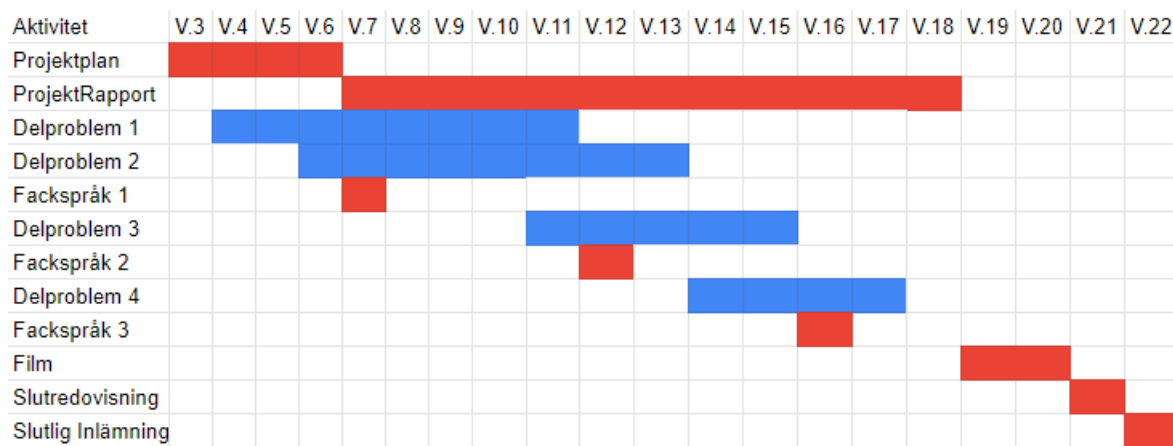
En positiv faktor är att projektets källkod ska vara öppet till alla. Det här är viktigt för att undvika återskapandet av program i framtiden vilket sparar resurser och arbetstid samt tillåter enkel vidare utveckling.

Användartester kommer att följa GDPR och anonymitet [10]. Detta är viktigt för att säkerställa

testarnas sekretess och anonymitet kring informationen de ger oss. Användare kan frivilligt välja att delta som testare.

## 7 Tidsplan

Tidsplanen beskrivs i form av en tabell 1 vilket visar inlämningar och viktiga datum samt ett Gantt schema 1 som visar vår tidsfördelning genom projektet.



Figur 1: Gantt schema



**Datum:   Aktivitet**

tis 25/1	Inlämning: Påbörjad projektdokumentation
fre 4/2	Inlämning: Rapportspråk
fre 11/2	Inlämning: Projektplan/Planeringsrapport
tis 8/3	Muntlig halvtidsredovisning
sön 13/3	Inlämning: Egen utvärdering 1
tor 28/4	Inlämning: Slutrapport, förhandsgranskning
tor 12/5	Inlämning: Sammanställd slutrapport
lör 14/5	Inlämning: Bidrags rapport
mån 16/5	Inlämning: Film samt 3 frågor om projekt
tor 19/5	Inlämning: Skriftlig individuell opposition
fre 20/5	Inlämning: Engelsk titel på rapport
tis 24/5	Närvaro slutredovisning
ons 25/5	- Avtal om e-publicering av kandidatarbete - Närvaro slutredovisning - Närvaro virtuell utställning
sön 29/5	Inlämning: Egen utvärdering 2
fre 3/6	Inlämning: Slutlig inlämning, Sammanställd slutrapport

Tabell 1: Planering för inlämningar i Kandidat DSLsofMath

## Referenser

- [1] Fysikteknologsektionen. (2022) Course statistics at chalmers. Hämtad: 2022/02/08. [Online]. Available: <https://stats.ftek.se/>
- [2] Wikipedia. (2022) Domain-specific language. Hämtad: 2022/02/09. [Online]. Available: [https://en.wikipedia.org/wiki/Domain-specific\\_language](https://en.wikipedia.org/wiki/Domain-specific_language)
- [3] P. Jansson, C. Ionescu, and J.-P. Bernardy. (2022) Domain-specific languages of mathematics. Hämtad: 2022/02/11. [Online]. Available: <https://github.com/DSLsofMath/DSLsofMath>
- [4] Haskell.org. (2022) *Haskell, An advanced, purely functional programming language*. Hämtad: 2022/02/09. [Online]. Available: <https://www.haskell.org/>
- [5] wiki portal Chalmers. (2021) *Agda*. Hämtad: 2022/02/09. [Online]. Available: <https://wiki.portal.chalmers.se/agda/pmwiki.php>
- [6] S. Axler, *Linjer Algebra Done Right*. Springer, 1995.
- [7] G.Sparr, *Linjär Algebra*. Studentlitteratur, 1997.
- [8] K. Claessen. (2020) *QuickCheck*. Hämtad: 2022/02/09. [Online]. Available: <https://hackage.haskell.org/package/QuickCheck>
- [9] P. Jansson, C. Ionescu, and J.-P. Bernardy, *Domain-Specific Languages of Mathematics*, ser. Texts in Computing. College Publications, Jan. 2022, vol. 24.
- [10] I. Consulting. (2016) General data protection regulation gdpr. [Online]. Available: <https://gdpr-info.eu/>