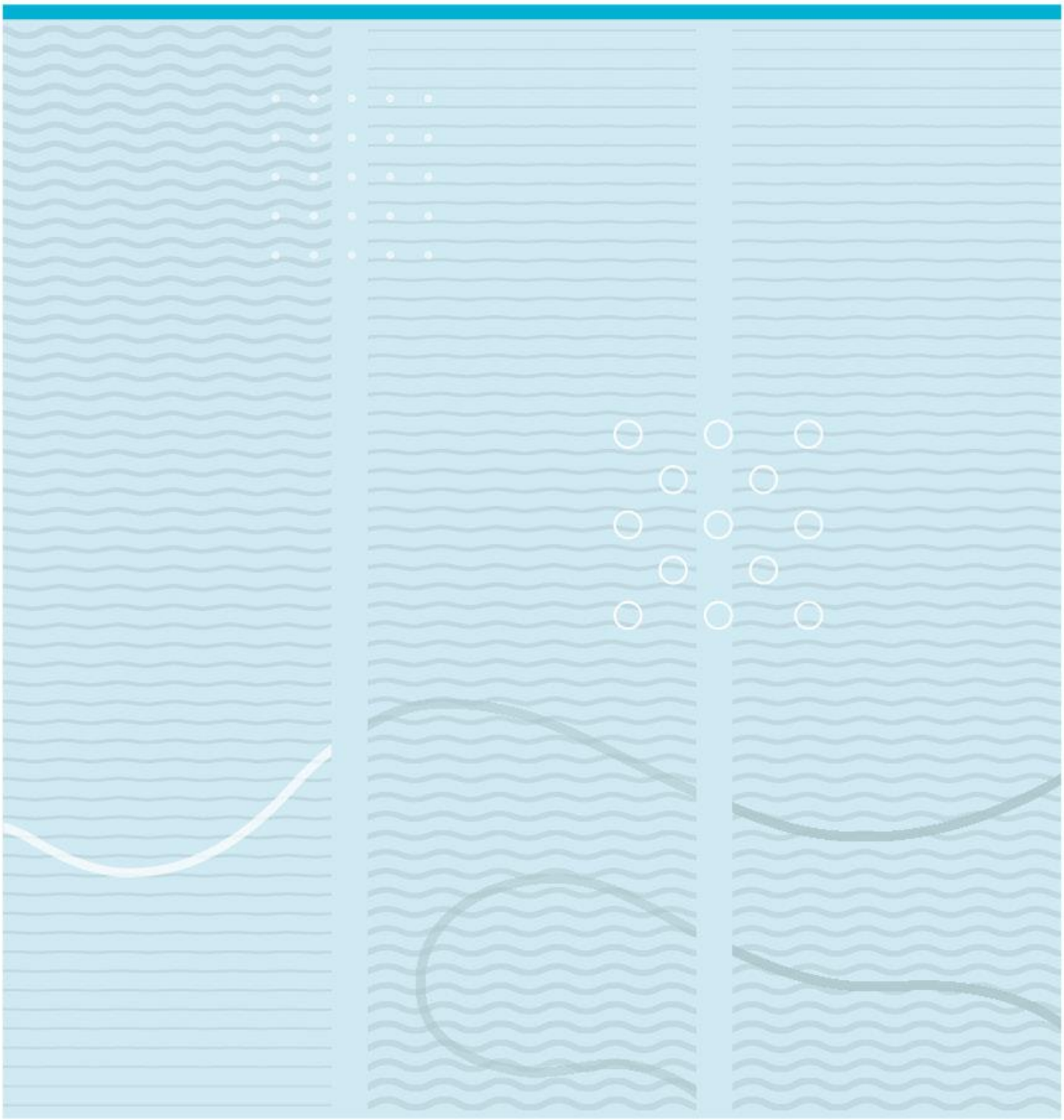
Universitetet i Sørøst-Norge
Fakultet for humaniora, idretts- og utdanningsvitenskap
–
Mastergradsavhandling
Grunnskolelærerutdanningen 5-10: Matematikk
Vår 2022

Anders Holm

Konkretiser det abstrakte med pedagogisk programmering

Et kvasieksperiment om programmering som ressurs i algebraopplæringen



Universitetet i Sørøst-Norge
Fakultet for humaniora, idretts- og utdanningsvitenskap
Institutt for matematikk og naturfag
Postboks 235
3603 Kongsberg

http://www.usn.no

Denne avhandlingen representerer 45 studiepoeng

Sammendrag

Dette er en kvasieksperimentell studie som undersøker hvordan pedagogisk programmering som et alternativ til vanlig algebraopplæring påvirker effekten på læring av algebra – nærmere bestemt utforming av algebraiske uttrykk. En hypotese om at pedagogisk programmering gir større effekt på læring av algebra kontra normal algebraopplæring skal bekreftes eller avkreftes. Dette ble gjort ved å måle algebraferdigheter i to tilnærmet like grupper både før og etter en intervensjon. Intervensjonen var at den ene gruppen fikk normal algebraundervisning, mens den andre gruppen fikk pedagogisk programmering som et alternativ til normal algebraundervisning. Etter intervensjonen fant jeg en svak kausalitet mellom programmering og algebraferdigheter. Ifølge en «independent t-test» som viste $p \approx 0.32$, kunne man ikke konkludere med at pedagogisk programmering fungerte bedre. Det viser derimot en stor mulighet for å drive med pedagogisk programmering på lik linje som vanlig algebraundervisning, og åpner opp for en mulighet til å dekke flere kompetansemål i matematikkfaget samtidig, samtidig som at man dekker flere kjerneelementer og elementer fra matematikkfagets fagrelevans og sentrale verdier.

Innholdsfortegnelse

1 Innledning 7

1.1 Bakgrunn for valg av studie 7

1.1.1 Algebrautfordringer hos norske ungdomsskoleelever 7

1.1.2 Programmering fremtrer i grunnopplæringen 8

1.1.3 Algebra + programmering = sant? 8

1.2 Forskningsspørsmål 9

1.3 Oppgavens struktur 10

2 Teoretisk rammeverk 11

2.1 Innledning til det teoretiske rammeverket 11

2.2 Læringsteorier 12

2.2.1 Relasjonell forståelse og nett av sammenhenger 12

2.2.2 Duvals semiotiske registre 13

2.2.3 Det sosiokulturelle læringssynet 14

2.3 Algebra 15

2.3.1 Introduksjon til algebra 15

2.3.2 Konseptualisering: Generalisering og studiet av strukturer 15

2.3.3 Konseptualisering: Studiet av funksjoner og algebra som modelleringsspråk 19

2.3.4 Algebraisk tenkning og kvasialgebra 24

2.4 Programmering og algoritmisk tenkning 28

2.4.1 Innledning til programmering i skolen 28

2.4.2 Konseptualisering av programmering i skolesammenheng 30

2.4.3 Konseptualisering av algoritmisk tenkning 34

2.5 Synergien mellom algebra og programmering i opplæringssammenheng 36

2.5.1 Kunnskapsstatus om sammenhengen 36

3 Metode 43

3.1 Introduksjon til metode 43

3.2 Epistemologisk standpunkt 44

3.3 Forskningsmetode 45

3.3.1 Kvantitativ metode 45

3.3.2 Kvasieksperimentell design 46

3.3.3 Hypotesetesting 48

3.4 Gjennomføring 51

3.4.1 Forskningsetikk 51
3.4.2 Operasjonalisering og indikatorer i pre og posttest 52
3.4.3 Kontroll- og eksperimentell gruppe 56
3.4.4 Undervisningsopplegg i kontroll- og eksperimentell gruppe 60
4 Analyse og resultater 64
4.1 Observasjoner 64
4.2 Resultater 65
5 Diskusjon 69
5.1.1 Refleksjon rundt masteravhandlingens tema og funn 69
5.1.2 Konklusjon 73
6 Referanser 74
7 Vedlegg 79
7.1 Vedlegg 1: Eksempeloppgave fra eksamen etter ny læreplan av utdanningsdirektoratet med hjelpemidler – oppgave 7 80
7.2 Vedlegg 2: Eksempeloppgave fra eksamen etter ny læreplan av utdanningsdirektoratet med hjelpemidler – oppgave 9 81
7.3 Vedlegg 3: Eksempeloppgave fra eksamen etter ny læreplan av utdanningsdirektoratet med hjelpemidler – oppgave 10 82
7.4 Vedlegg 4: Løsning av praktisk situasjon med likning med to ukjente. Dette er et løsningsforslag av oppgaven i vedlegg 1 83
7.5 Vedlegg 5: Matemagisk sitt forslag til terminprøve for 8.trinn etter den nye læreplanen – oppgave 10 84
7.6 Vedlegg 6: Enkel sannsynlighetsfordeling med kast av to terninger. Eksempel fra egen undervisning i matematikk på 9.trinn 85
7.7 Vedlegg 7: Hjørnespillet 86
7.8 Vedlegg 8: Samtykkeerklæring 87
7.9 Vedlegg 9: Pretest 89
7.10 Vedlegg 10: Posttest 93
7.11 Vedlegg 11: Pedagogisk plan kontrollgruppe 97
7.12 Vedlegg 12: Pedagogisk plan eksperimentell gruppe 98
7.13 Vedlegg 13: Oppgaveark kontrollgruppe 99
7.14 Vedlegg 14: Oppgaveark eksperimentell gruppe 108
7.15 Vedlegg 15: Økonomiprogram som viser hensiktsmessig bruk av for-løkke eller while-løkke 118

Forord

I en stadig mer digitalisert verden, og med et samfunn som innlemmer programmering i grunnopplæringen, vil det være naturlig å se effekten av dette utover selve programmeringen. Jeg håper flere vil se på denne oppgaven og bli motivert til å forsøke det samme, men gjerne i større omfang. Jeg har sett mange lærere de siste årene droppe å jobbe med programmering i klasserommet eller sett på det som en byrde, noe som må krysses på en liste. Hensikten med denne masteroppgaven er å vise hvordan det ikke er en byrde, men heller en fordel. Noe som kan motivere elevene og gi dem ny innsikt i matematiske fenomener.

Min hensikt med denne masteroppgaven er også at man kan lære noe av den, derav den relativt store litteraturgjennomgangen. Forhåpentligvis vil noen lese det og få ideer til egen undervisning. Jeg har også forsøkt å forklare og begrunne alle valg underveis, spesielt i metodedelen, slik at andre kan få inspirasjon til å gjennomføre lignende forskningsprosjekter hvis de leser denne avhandlingen.

Jeg vil takke min veileder Pål-Erik Eidsvig for gode innspill. Jeg vil også takke Mette Krogh for at hun har hatt troen på meg og dette masterprosjektet.

Horten, mai 2022
Anders Holm

1 Innledning

I denne mastergradsavhandlingen vil det bli foretatt en kvasieksperimentell studie. To elevgrupper med henholdsvis likt gjennomsnitt og standardavvik, basert på en pretest, vil ta del i to ulike undervisningsopplegg. Målet er at variasjonen vi eventuelt finner på en posttest, skal springe ut fra undervisningsoppleggenes forskjeller – en systematisk variasjon, og forhåpentligvis si noe om undervisningsoppleggenes effekt på læring av algebra hos ungdomsskoleelever. En hypotese om at pedagogisk programmering i algebraopplæringen fører til høyere læring av algebra kontra normal algebraundervisning, skal bekreftes eller avkreftes. Derfor vil den ene gruppen, kontrollgruppen, utsettes for normal algebraundervisning, og den andre gruppen, den eksperimentelle gruppen, utsettes for pedagogisk programmering, før vi deretter måler effekten i form av algebraferdigheter.

1.1 Bakgrunn for valg av studie

1.1.1 Algebrautfordringer hos norske ungdomsskoleelever

I Norge presterer vi svakere i algebra kontra andre matematiske temaer på ungdomsskolen om vi ser på TIMSS-undersøkelser fra 2019 og foregående år (TIMSS 2019: Kortrapport, 2020, s. 18). Ferdighetene i matematikk synker dessuten fra barneskolen til ungdomsskolen, noe som vesentlig skyldes at algebradomenet, som på alvor aktualiseres på ungdomsskolen, trekker ned gjennomsnittet (TIMSS 2019: Kortrapport, 2020, ss. 16–18). Pramesti & Retnawati beskriver overgangen fra barneskolen til ungdomsskolen som en overgangsfase fra kjent og konkret matematikk til ukjent og abstrakt matematikk (2019, s. 2). Denne overgangen virker til å være en utfordring i norsk skole. Kieran artikulerer at elever som presterer sterkt i den resultatorienterte aritmetikken på barneskolen, ikke nødvendigvis håndterer overgangen til algebra på ungdomsskolen (2004, s. 140). Algebra kan ses på som blant annet generalisering og studiet av strukturer (Kaput, 2008, s. 11). Det løftes derimot frem at elevers erfaringer med å lære aritmetikk sjeldent fostrer verdsettelsen av generalisering og strukturer (Arcavi et al., 2017, s. 58). Det blir derfor krevende når elevene skal bedrive meningsfull matematisk aktivitet, en aktivitet som beskrives som at elever klarer å se abstrakte ideer gjemt bak matematiske symboler, i motsetning til noe som skal kalkuleres; en strukturell forståelse kontra en operasjonell forståelse (Sfard & Linchevski, 1994, ss. 193-224). Derfor bør den resultatorienterte aritmetikken som fostrer operasjonell forståelse utfordres som en inngang til algebra, og alternative innganger som kan fostre strukturell forståelse bør diskuteres.

Naalsund forklarer, etter sin doktorgradsavhandling om algebravansker i Norge, at mange elever ser på algebra som «meningsløs manipulasjon av symboler» (sitert i Jakobsen, 2012). Dette er urovekkende, spesielt med tanke på at den første setningen i matematikkfagets sentrale verdier understreker at matematikk handler om å forstå mønstre og sammenhenger, og videre hvordan dette skal kommuniseres gjennom abstraksjon og generalisering – algebra (Utdanningsdirektoratet, 2020a). Hvis algebra reduseres til meningsløs manipulasjon av symboler, vil det være krevende for elever å håndtere kompetansemålene i matematikk som nå ser en dreining mot vektlegging av blant annet modellering og generalisering, samt tre kjerneelementer som eksplisitterer og impliserer bruken av algebra: modellering og anvendelser, resonnering og argumentasjon, abstraksjon og generalisering (Utdanningsdirektoratet, 2020b; Utdanningsdirektoratet, 2020c).

1.1.2 Programmering fremtrer i grunnopplæringen

OECD beskriver flere tiltak som bør igangsettes i utdanning rundt om i verden for å tilfredsstille fremtidens behov, og understreker hvordan grunnopplæringen bør sikre at elever utstyres med grunnleggende IKT-ferdigheter og problemløsningsferdigheter (OECD, 2016, s. 3). Den digitaliserende verden kan ikke stanses. I stortingsmelding 28, igangsetteren av fagfornyelsen, løftes det frem at flere land nå legger større vekt på at elevene skal mestre mer avanserte IKT-ferdigheter og problemløsning, og at elevene skal forstå og produsere IKT, fremfor at de skal være konsumenter av IKT ((2015–2016), s. 32). Videre nevnes det programmering som eget valgfag, at tilbudet om undervisning i programmering generelt skal bygges ut, og at IKT skal integreres i fagene (Meld. St. 28, (2015–2016), ss. 32-54). Det nevnes ikke noe spesifikt om programmering i matematikkfaget der, men i læreplanen ser vi at matematikkfaget har fått det største ansvaret (Utdanningsdirektoratet, 2020f). Dette er et globalt fenomen, og vi kan se at flere titalls land har kastet seg på bølgen med å innlemme programmering i grunnopplæringen (Sentance & Csizmadia, 2017, s. 470).

1.1.3 Algebra + programmering = sant?

Argumentet om sammenhengen mellom matematikk og programmering er én av grunnene til å inkorporere programmering i matematikkfaget (Bocconi et al., 2018, ss. 14–16). Kaufmann & Stenseth nevner hele fem studier som argumenterer for at programmering gir sterke matematiske ferdigheter hos elever (2021, s. 1030). Det som attpåtil er bemerkelsesverdig, er at flere forskere har sett på sammenhengen mellom programmering og algebraferdigheter spesifikt. Mason, en anerkjent algebradidaktiker, nevner slektskapet mellom algebra og programmering, og hvordan arbeid med programmeringsspråk påkaller algebraisk tenkning (2018, s. 335). Tidligere forskning fra 1989

viste at programmering i matematikkundervisningen ga økt forståelse for å se det strukturelle bak algebraiske symboler, og økt aksept for variabelen n – mengden med naturlige tall (Healy et al., 2002, s. 239. Nyere forskningsartikler beskriver derimot også en sammenheng mellom programmering og algebraisk tenkning, der det løftes frem at programmering kan støtte elevene i arbeid med variabler, matematisk struktur og mønstre, samt generalisering (Kilhamn & Bråting, 2019, s. 572). Det nevnes blant annet også at fokuset havner på prosessen fremfor resultatet, og at dette kan styrke elevene i å se det strukturelle kontra det operasjonelle, slik som etterspurt i underkapittel 1.1.1 (Bråting & Kilhamn, 2021, s. 181). Flere forskere har løftet opp sammenhengen; banebrytende forskning er derimot vanskelig å finne (Benton et al., 2017, s. 118). TIMSS-resultater fra England viser imidlertid er signifikant forbedring i algebraferdigheter hos sine elever i 2015 og 2019 (TIMS 2019: National report for England, 2020, s. 89), og det har blitt artikulert hvordan programmering har gjennomgått en renessanse i engelske skoler i denne perioden – innført som del av eget fag i 2014 (Benton et al, 2017, s. 116). Det kan derfor være nærliggende å tenke at programmering kan ha en positiv effekt på algebraferdigheter.

1.2 Forskningsspørsmål

Norske elevers utfordring med algebra, ideen om andre innganger til algebra, og en mulig redning i form av programmering, er grunnen til at jeg ønsker å svare på følgende forskningsspørsmål:

Vil pedagogisk programmering som en ressurs i algebraopplæringen, gi signifikant forbedring i algebraferdigheter – nærmere bestemt utforming av algebraiske uttrykk – hos norske ungdomsskoleelever i forhold til normal algebraopplæring?

Som man kan se, foreligger det grunnlag for å teste ut om det finnes en kausalitet mellom disse fenomenene. Altså om endring i variabelen programmering, om den er til stede eller ikke, fører til endring i variabelen algebraferdigheter (Creswell & Creswell, 2018, s. 92). Det vil legges stor vekt på utforming av algebraiske uttrykk da det er viktig å mestre dette for å arbeide med de tre kjerneelementene nevnt i underkapittel 1.1.1; kommer tilbake til dette i kapittel 2 i denne avhandlingen. Det kan være at vi allerede ser en forbedring i algebraferdigheter de neste årene ettersom at programmering nå har fått en større plass i norsk skole – en korrelasjon. Det forteller oss ingenting om kausaliteten derimot, og det ønsker jeg å svare på i denne masteravhandlingen. For det er nettopp bevisstheten rundt kausaliteten mellom programmering og algebraferdigheter som er viktig, for dette kan hjelpe lærere med å legge til rette for undervisningsopplegg der programmering og algebra kan eksistere i en synergi. Da kan man simultant dekke flere

kompetansemål og kjerneelementer, tid vil bespares, og sammen vil de gjøre resultatet enda sterkere enn hver for seg for elevene.

1.3 Oppgavens struktur

Denne oppgaven er strukturert på følgende måte:

1. **Innledning**: Bakgrunn for valg av studie og forskningsspørsmål som du nå har lest.
2. **Teoretisk rammeverk**: En tydelig gjennomgang av teori og konseptualisering av begreper som skal brukes videre i avhandlingen. Dette legger grunnlag for hvordan begreper og ferdigheter skal operasjonaliseres videre i metodedelen. De teoretiske antakelsene om fenomener vil baseres på forskningen som blir løftet frem her – det som skal testes ut og forskes på.
3. **Metodedel**: En grundig begrunnelse og oversikt for hvordan hypotesene skal testes ut. Her vil jeg løfte frem selve forskningsprosjektet som skal ta sted og dets styrker og svakheter.
4. **Analyse**: Her vil datainnsamlingen, det empiriske materialet, analyseres. Statistiske metoder vil bli tatt i bruk. Da får vi se om pedagogisk programmering ga økt effekt på læring av algebra hos en gruppe ungdomsskoleelever, og hvorvidt denne effekten er signifikant.
5. **Diskusjon**: Her vil resultatene av forskningsprosjektet diskuteres opp mot tidligere forskning, læringsteorier og masteravhandlingens hensikt og mål.

2 Teoretisk rammeverk

2.1 Innledning til det teoretiske rammeverket

Dette er den teoretiske delen av avhandlingen – litteraturgjennomgangen. Litteraturgjennomgangen er det arbeidet man utfører for å knytte forskningen til et større teoretisk og empirisk fagfelt, til annen forskning, teorier eller debatter, og som gir en bredere forståelse av forskningsspørsmålet (Fink, 2014; Hart, 1998, referert i Høgheim, 2022, s. 62). Innenfor kvantitativ forskning, som er metoden i denne avhandlingen, skal litteraturgjennomgangen være omfattende, nettopp fordi jeg skal foreta teoretiske antakelser om et fenomen på bakgrunn av teori (Creswell & Creswell, 2018, s. 68). Man skal bruke litteraturgjennomgangen til å forklare forventede sammenhenger mellom fenomener, samt beskrive og løfte frem hensiktsmessige teorier, for det er nettopp her hypotesene har sine røtter (Creswell & Creswell, 2018, s. 68). Begreper man bruker skal også tydelig konseptualiseres – begrepsdannelse, for dette er essensielt i forskning for å ha en gjennomgående konsensus med leseren og forskningsfellesskapet om hva som er betydningen av diverse begreper som forskes på og brukes (Creswell & Creswell, 2018, ss. 79–81).

For å sikre troverdigheten til kildene mine, har jeg for det meste tatt i bruk fagfellevurdert litteratur; hovedsakelig forskningsartikler. Når noe er fagfellevurdert betyr det at andre forskere har lest, vurdert og bedømt arbeidet som tilfredsstillende vitenskapelig kvalitet for utgivelse (Høgheim, 2020, s. 71). En del vitenskapelig bøker vil bli brukt, og disse går gjennom samme kvalitetssikring (Høgheim, 2020, s. 72). Noen lærerbøker vil bli brukt for å videre belyse temaer og læringsteorier, eller få innsikt i undervisningsopplegg og programmering. Slike bøker vil ikke nødvendigvis ha gjennomgått samme kvalitetssikring, og antakelser om fenomener eller begreper vil derfor ikke baseres på disse (Høgheim, 2020, s. 72). I selve litteratursøket er det tatt i bruk søkemotorer som er dedikerte til forskningslitteratur, henholdsvis ERIC, Google Scholar, Oria, JSTOR og andre (Høgheim, 2020, ss. 67–68). Mye forskning er også hentet fra Springer.

Jeg vil først nevne sentrale teorier og læringssyn som vil bli brukt gjennom avhandlingen. Deretter skal jeg konseptualisere algebrabegrepet i opplæringssammenheng med støtte i relevant kunnskapsstatus og læreplanverk. Dette vil videre brukes når algebrabegrepet skal operasjonaliseres i metodedelen. Videre vil jeg konseptualisere programmeringsbegrepet i opplæringssammenheng med støtte fra læreplanverk, policy-dokumenter og kunnskapsstatus. Deretter vil jeg gå inn på hvordan programmering kan støtte algebraopplæring.

2.2 Læringsteorier

Jeg vil først løfte frem læringssyn og teorier om læring som vil bli brukt gjennom avhandlingen. Dette er avgjørende for å kunne bedømme hva læring egentlig er innenfor matematikkfaget, og hva som kan kvalifiseres som matematisk kunnskap, for det er jo nettopp dette som skal vurderes og forskes på. Mange vil nok lene seg på Kilpatrick et al. sine fem tråer av matematisk kompetanse, som metaforisk sett tvinnet sammen, utgjør matematisk kompetanse (2001, ss. 115–118). Disse tråene vil henholdsvis være konseptuell forståelse, operasjonell forståelse, strategisk kompetanse, logisk resonnering og produktiv holdning (Kilpatrick et al., 2001, s. 115–118). I denne avhandlingen vil konseptuell forståelse vektlegges fordi dette tar for seg en høy orden av forståelse innenfor matematikkfaget der elevene har organisert kunnskapen sin i en sammenhengende helhet, som gjør dem i stand til å lære nye ideer ved å koble disse ideene til det de allerede vet, og de vet i hvilke kontekster ideene er brukbare og nyttige (Kilpatrick et al., 2001, s. 118). Elevene kan også i stor grad representere matematiske objekter på forskjellige måter, og forstår hvordan noen representasjoner er nyttigere enn andre, samt å veksle mellom disse (Kilpatrick et al., 2001, s. 119). Strategisk kompetanse vil også vektlegges i den forstand at elevene klarer å formulere en situasjon eller et problem i matematisk drakt, noe som er et kriterium i strategisk matematisk kompetanse (Kilpatrick et al., 2001, s. 124). Logisk resonnering vil også vektlegges, og det er her de to trådene nevnt ovenfor blir tvinnet sammen, der man navigerer gjennom de mange fakta, prosedyrer, konsepter og løsningsmetode og kan se hvordan alt passer sammen, at de skaper mening, og at vi kan formulere meningsfylte matematiske bevis (Kilpatrick et al., 2001, s. 129). For å gå dypere til verks i disse kunnskapsområdene, vil noen læringsteorier og et læringssyn nå bli introdusert.

2.2.1 Relasjonell forståelse og nett av sammenhenger

Relasjonell forståelse kan beskrives som en forståelse der det ikke holder å bare vite hva man skal gjøre i en situasjon, men også hvorfor man gjør som man gjør (Skemp, 1976, s. 21). I matematisk forstand vil det si å løse et problem med forståelse, en forståelse som lar oss løse nye og ukjente problemer (Skemp, 1976, s. 29). Man kan trekke koblinger mellom forskjellige matematiske konsepter og temaer, der disse bygger på hverandre som meningsfylte verktøy (Skemp, 1976, s. 29). Motpolen til dette er når elever lærer regler og rutiner som kan brukes for å løse et matematisk problem – instrumentell forståelse – «rules without reason» (Skemp, 1976, s. 28). Ulempen med dette er at elevene ikke forstår hvorfor de gjør noe, for de kjenner kun igjen en situasjon og følger en innøvd rutine, og kan derfor ikke løse nye og ukjente problemer, nettopp fordi de ikke kjenner til disse situasjonene – de ser det ikke i relasjon til noe annet. Denne undervisningsmåten er enklere og mer behagelig for eleven og læreren, og som lærer er man egentlig bare djevelens advokat (Skemp,

1976, s. 28). Dette vil likne mye på det nye begrepet dybdelæring som beskrives som at man gradvis utvikler kunnskap eller metoder man kan anvende i ukjente situasjoner – videre i livet (Meld. St. 28, (2015–2016), s. 14). Utdanningsdirektoratet forklarer at det skal legges til rette for dybdelæring for at barn og unge skal utvikle kompetanse de trenger i en framtid som endrer seg raskt (2019a). Derfor kreves det at all matematikk skal være meningsfylt, og rutiner og regler vil ikke bli betraktet som matematisk kunnskap i denne avhandlingen. Det kan sammenliknes med å se på et matematisk uttrykk på en strukturell måte versus operasjonell måte, slik som nevnt innledningsvis.

Noss & Hoyles nevner læringsteorien kalt «nettet av sammenhenger», og hevder i likhet med meg selv, at læring handler om å skape sammenhenger mellom det eleven kan fra før av, og det nye som skal læres (1996, s. 105). Elevene trenger noe å bygge videre på, slik som i skjemateorien til Piagét (Hinna et al., 2011, s. 890), og jobben til læreren er å finne dette, og koble sammen elevens nett av sammenhenger – lærerens jobb er å være en sammenkobler (Noss & Hoyles, 1996, s. 106). Det å finne gode analogier og metaforer ved å bruke elevenes forkunnskaper er hjertet i god undervisning (Noss & Hoyles, 1996, s. 105). Dette er nødvendig for å få til den lovfestede rettigheten om tilpasset opplæring, altså at læringen skal tilpasses elevenes nivå, og derfor må man ta utgangspunkt i det elevene kan fra før av (Opplæringsloven, 2008). Slik kan vi som lærere koble sammen elevenes erfaringer, gjerne virkelige situasjoner, med matematiske temaer. Videre kan også de matematiske temaene kobles sammen. Dette er en løsning på hvordan å skape relasjonell forståelse.

2.2.2 Duvals semiotiske registre

Matematiske objekter kan ikke direkte observeres eller måles med noe instrument, og tilgangen til objektene er bundet til bruk av et system av representasjoner, kulturelt skapte representasjoner (Duval, 2006, ss. 106–107). Det er her det kognitive møter det sosiokulturelle, nettopp fordi forståelse for det matematiske objektet er tvunget til bruken av kulturelle artefakter. Ingen matematisk aktivitet kan gjøres uten å ta i bruk semiotiske representasjonssystemer – meningsskapende registre (Duval, 2006, s. 107). Dette kan være alt fra det visuelle registre, eksempelvis geometri, eller verbalspråk for å skape kontekst, eller registeret bestående av matematiske symboler (Duval, 2006, s. 110). Matematisk forståelse handler om å kunne navigere mellom ulike representasjoner, gjerne i ulikt register, og det kan da oppnå en synergi mellom representasjonene som da øker forståelsen for det matematiske objektet (Duval, 2006, s. 126). For eksempel kan den symbolske representasjonen til rektangeltallene, $R_n = n(n+1)$, få god støtte av den visuelle representasjonen som viser figurtallene, og sammen kan disse gi styrket forståelse for

det matematiske objektet. Vi må også ta med i betraktning hvilket register som brukes, da noen kan være mer krevende å tolke – slik som for eksempel i det kontekstuelle i forhold til det symbolske (Duval, 2006, s. 109). Det kan være vanskeligere å se det matematiske objektet i det kontekstuelle registeret kontra det symbolske, siden det registeret kan brukes til så mye mer enn matematikk. Vi må også se på overgangen mellom registrene, da noen er ikke reversible, som vil si at det er et større kognitivt gap den ene veien, kontra den andre veien (Duval, 2006, ss. 123-127). Eksempelvis funksjonsuttrykk til graf og omvendt, eller formel til tallfølge og omvendt. At eleven kan beskrive det matematiske objektet på flere måter er derfor viktig. Dette er også nyttig, fordi vi kan utvide horisonten av representasjonssystemer, samt at man må ha i bakhodet begrensingene og reglene i de ulike registrene.

2.2.3 Det sosiokulturelle læringssynet

Vygotsky forklarte at læring skjer først sosialt, før det skjer individuelt; interpsykologisk før intrapsykologisk (Hinna et al, 2011, s. 899). Derfor er det sosiokulturelle læringssynet et fint kompliment til de kognitive teoriene nevnt ovenfor, nettopp fordi vi må også ta i betraktning faktorer rundt eleven. Når vi bedriver matematisk aktivitet, så bruker vi kulturelle artefakter, og hovedsakelig språk. Derfor må det kulturelle aspektet betraktes, fordi dette er kulturelle konstruksjoner og tankemåter som skal internaliseres, og dette skjer gjennom deltakelse i fellesskap (Hinna et al, 2011, s. 900). Vi må også se på konteksten eller situasjonen det læres i, for idealet er at konteksten skal være autentisk – realistisk (Hinna et al, 2011, s. 901). Det betyr at sammenhengen eller situasjonen som læring foregår i, har betydning for hva som læres, og hvordan det læres (Hinna et al, 2011, s. 901). Dialog blir også sett på som sentralt, og teorien om sonen der det er visse ting elevene kan gjøre med hjelp fra andre før de er i stand til å gjøre alene kaller vi for den proksimale utviklingssonen, og denne «noen andre» kalles en medierende hjelper – ofte læreren (Hinna et al, 2011, s. 908). Elevene bør derfor ikke etterlates helt for seg selv og sine kognitive egenskaper, men de bør støttes, og dette skjer vanligvis i en dialog. Mediatorene som kan være medelever eller læreren, skal tenkes å bygge stillas for eleven, og det er et viktig kriterium at mediatoren er mer kompetent enn eleven (Hinna et al, 2011, s. 909). Det siste poenget vi i det sosiokulturelle læringssynet vektlegger at læring er mediert, som vil si at kunnskap og tankemåter blir formidlet gjennom bruken av kulturelle artefakter, som for eksempel linjaler eller datamaskiner, og semiotiske redskaper, som for eksempel matematiske symboler eller språk (Hinna et al, 2011, s. 904–905). Derfor er det viktig å betrakte hvordan disse kan påvirke læring og forståelse.

2.3 Algebra

2.3.1 Introduksjon til algebra

Algebra er ikke et enkelt begrep å definere, slik som Balacheff understreker i sitt postskript i en bok om nettopp algebradidaktikk: Hvis denne boken har gitt meg noe, så er det en styrket usikkerhet på hva algebra egentlig er, og derfor løfter jeg frem det didaktiske dilemmaet som innebærer å lære symbolsk aritmetikk versus å lære algebra (2002, s. 259). Kieran nevner også dette dilemmaet, for algebra er jo så mye mer enn kun symbolsk aritmetikk, og for elever som kun jobber i det hun kaller for transformasjonsdomenet av algebra, vil algebra miste mye av sin egentlige funksjon (Kieran, 2007, ss. 747-748). Derfor vil forståelsen for algebra vektlegges i denne oppgaven, og ikke selve aritmetikken. For det ønskes nettopp at elevene skal ha en relasjonell forståelse for algebra, der de vet hvorfor de gjør som de gjør og hva algebra kan brukes til – algebraens funksjon. Hvis man ikke vet hva de algebraiske uttrykkene betyr, blir det kun meningsløs manipulasjon av symboler – som nevnt innledningsvis. Denne avhandlingen vil derfor ta for seg selve utformingen av algebraiske uttrykk, for det blir også essensielt å mestre dette for å håndtere det mangfoldet av læringsmål og kjerneelementer som er avhengig av nettopp det, som nevnt innledningsvis, der man kan virkelig kan dra nytte av algebra – for eksempel abstraksjon eller modellering. Jeg skal derfor videre belyse og avgrense hvordan algebra kan være mye mer enn kun symbolsk aritmetikk, og hvordan vi kan definere algebra i norsk skolesammenheng, og hvilke innganger vi har til dette i undervisningen. Som nevnt innledningsvis, er dette det matematiske temaet som vi sliter mest med her til lands, for ser vi på TIMSS-resultatene, så er det matematiske temaet der vi scorer vesentlig lavest (TIMSS 2019. Kortrapport, 2020, s. 18). Vi ligger også langt etter flere land internasjonalt i matematikk generelt på ungdomsskolen, eksempelvis Japan, Russland, Irland, Israel, USA, England, og Singapore som presterer hele 22,5% bedre enn oss i Norge (TIMSS 2019. Kortrapport, 2020, s. 15). Vi ligger også under alle de andre nordiske landene i domenet algebra spesifikt (TIMSS 2019. Kortrapport, 2020, s. 18). Det er synd da det epistemologiske grunnlaget for algebra ofte blir omtalt som en inngangsport til høyere matematikk (Kaput et al., 2008, s. 23).

2.3.2 Konseptualisering: Generalisering og studiet av strukturer

For å konseptualisere algebrabegrepet, vil jeg hovedsakelig lene meg på definisjonen til algebradidaktikeren James Kaput. Han er av den oppfatning at generalisering og symbolisering er kjernen av algebraisk aktivitet, nettopp fordi den eneste måten man kan komme med et enkelt utsagn som kan gjelde flere tilfeller, generalisering, er å referere til flere forekomster gjennom et slags samlende uttrykk som refererer til dem alle – via variabler (Kaput et al., 2008, s. 20). Det vil

si at vi jobber abstrakt, nettopp fordi de objektene vi betegner ikke eksisterer, de er ubestemte og samlende (Arcavi etal., 2017). Den symbolske algebraen, et kulturelt artefakt slik som all annet matematikk, blir derfor hovedredskapet for å uttrykke denne generaliseringen (Kaput, 2008, s. 8). Det at disse uttrykksmåtene har blitt formet gjennom århundrer med kognitiv aktivitet, gjør at det er langt ifra trivielt for elevene å forstå disse (Radford, 2010, s. 4). Hvordan den symbolske algebraen kan internaliseres, blir derfor et stort spørsmål. Den symbolske algebraen er et semiotisk redskap som medierer den algebraiske tenkemåten, men det kan være hensiktsmessig å ta i bruk andre medierende artefakter som er mer nærliggende eleven.

Kaput løfter også frem hvordan det snevre synet på algebra som har dominert skolealgebra i årevis, et syn som primært verdsetter syntaktisk, veiledet, symbolsk manipulasjon, undervurderer alle de sidene av algebra som gjør det til det redskapet som det er (2008, s. 8). Videre løfter han frem hva han mener disse sidene her: Studiet av strukturer og relasjoner, studiet av funksjoner, og algebra som et modelleringsspråk både inne og utenfor matematikk (2008, s. 11).

Den første delen vil inneholde et krysningspunkt mellom det å uttrykke matematiske strukturer, en form for generalisering, og matematiske resonnement. Et eksempel på matematisk struktur vil kunne være å gjenkjenne den samme strukturen i flere former; det handler om å se på et aritmetisk uttrykk på en ny måte, og innebærer å se på formen i motsetning til den kalkulerte verdien (Kaput, s. 12). Ellemor-Collins & Wright hevder at det å se det strukturelle i tall betyr å organisere tall mer formelt, etablere regulariteter i tall, relatere tall til andre tall, og konstruere symmetrier og mønstre i tall (2009, s. 53). Eksempelvis kan vi tenke oss at elevene studerer partall, henholdsvis 8 eller 10, og oppdager sammenhenger ved at $8 = 4 + 4 = 2 * 4$ og $10 = 5 + 5 = 2 * 5$. De vil da komme nærmere oppfattelsen om at alle partall har tallet to som faktor, og kan argumentere for at partall er to multiplisert med ett eller annet heltall. Det handler om å se gjennom det matematiske objekt, og klare å dekomponere og rekomponere dem på forskjellige strukturelle måter (Kieran, 2018, s. 80). Dette krever en løsrivelse fra det operasjonelle, altså noe som skal kalkuleres, og over til det strukturelle. Det krever også god kontroll på å behandle det matematiske objektet innenfor det symbolske registeret, så noe forståelse for aritmetikk er nødvendig. Mange elever sliter med dette da de mangler noe som beskrives best på engelsk som «acceptance of lack of closure»: Nemlig at elever trenger å akseptere at et matematisk uttrykk som for eksempel $10 + 1$ eller $5 + 5 + 1$ kan legitimeres på lik linje som deres kalkulerte verdi på 11 (Collis, 1975). Elever som blir for opptatt av rutiner, eller har en instrumentell forståelse, kan fort gå glipp av matematiske strukturer, og

heller falle i en kalkuleringsrutine der man primært forholder seg til regler (Arcavi et al., 2017, s. 57; Skemp, 1976, s. 21).

For å uttrykke denne strukturen, blir det nødvendig å generalisere (Kaput, 2008, ss. 12–13). Denne «ett eller annet heltall» må defineres på en eller annen måte. Kaput forklarer som nevnt innledningsvis, at dette er hjertet i algebra, der vi bruker variabler til å forene mangfoldet og ubestemtheten, der man skaper det symbolske objektet (Kaput et al., 2008, s. 20). Elever som utvikler evne til å resonnere om generelle sammenhenger mellom størrelse, vil ha det konseptuelle grunnlaget for å lære algebra (Smith & Thompson, 2008, ss. 111–113). For å uttrykke disse sammenhengene, blir det derfor sentralt å bruke bokstaver som står for mengder (Kaput, 2008, s. 13). Det betyr at du kan regne med disse ubestemte mengdene som om du kjente dem, som om de var spesifikke tall (Radford, 2010, s. 2). Dette gjør oss i stand til å sammenligne generalisering, undersøke deres form og belyse nye sider av matematiske mønstre eller strukturer (Kaput et al., 2008, s. 23). En rekke studier viser dessverre at elevene har utfordringer med å forstå seg på variabler, og å bruke variabler som plassholdere, en ukjent, et generalisert tall eller en mengde (Jupri & Drijvers, 2016, s. 2482).

Radford løfter frem hvordan man kan bruke figurtall for å øve på generalisering i forhold til mengder: det vil si å først se på forholdet eller sammenhengen mellom hver figur, for så å gå forbi alle figurene og tenke på en generell figur som uttrykker mønsteret eller sammenhengen (Radford, 2018, s. 9; Radford, 2010, s. 7). Denne generelle figuren skal deretter beskrives algebraisk ved å ta i bruk symbolspråk. Man kan blant annet tenke seg at hver figur representerer hvert partall i tallfølgen med partall, der vi refererer til hver figur med å bruke variabelen n. Videre kan elevene lage det algebraiske uttrykket, eller formel om du vil: $2n$ eller $P_n = 2n$. Radford nevner derimot at å beskrive det algebraisk på den måten, er langt mer krevende enn å se selve strukturen: De fleste elevene klarer å se strukturen mellom figurene, men kun få av dem klarer å uttrykke det algebraisk (2010, s. 8). Et eksempel på en slik oppgave ser vi på Matemagisk, et læreverk etter fagfornyelsen, sin terminprøve i matematikk – se vedlegg 5.

Det er kjente utfordringer med å forstå bokstaver som variabler i opplæringssammenheng, der variabelen n er en av de vanskeligste å forstå se på (Rystedt et al., 2016, s. 5). Utfordringen for elever er at algebra ofte handler om å lære et kommunikasjonsspråk istedenfor å uttrykke forhold og strukturer, og at nettopp dette blir det store hinderet (Rystedt et al, 2016, s. 5). Elevene håndterte å se strukturer i figurtall som nevnt ovenfor, men da de skulle ta i bruk det semiotiske redskapet

variabler for å beskrive strukturen, ble det problematisk. Det er en utfordring at kommunikasjonsspråket som egentlig skal mediere algebraisk tenkning, i realiteten hindrer det. Forskning viser også at det er mer krevende å uttrykke det generelle med variabler, men at å forstå seg på likninger med ukjente variabler eller konstanter var lettere å begripe, nettopp fordi deres aritmetiske bakgrunn kunne håndtere det (Rystedt et al., 2016, s. 10). Det krever ikke det samme nivået av abstrakt tenkning og generalisering. Likninger med ukjente vil derfor ikke bli særlig vektlagt videre i denne oppgaven.

Denne delen av algebra, ifølge Kaput, handler også om å resonnere om matematiske forhold i generelle former, som for eksempel å bevise hvorfor summen at to oddetall blir et partall (2008, ss. 12–13). Dreyfus et al. som observerte syvendeklassinger som i samarbeid konstruerte et algebraisk bevis, rapporterte at å konstruere bevis innebar først å generalisere prosessen, og deretter rettferdiggjøre den ved hjelp av en algebraisk representasjon (Kieran, 2007, s. 726). For hvis elevene skal bevise noe, så krever det generalisering, og Balacheff beskriver dette som en stor utfordring når det gjelder matematiske bevis, der generalisering er nødvendig for å få til dekontekstualisering, der man forlater spesifikke objekter og entrer domenet klasser av objekter – det generelle – og tar i bruk variabler (Balacheff, 1988). For man ønsker jo nettopp at sammenhengen man har funnet skal gjelde i alle mulige tilfeller, hvis ikke blir det bare det som Balacheff benevner som naiv empirisme (1988). Gyldigheten av beviset eller resonnementet kommer an på det etablerte fellesskapets normer for godkjennelse (Balacheff, 1988). Hele poenget med bevis eller algebra som sådan, er å oppfatte et eller annet mønster eller regelmessighet, og deretter prøve å uttrykke det kortfattet slik at du kan kommunisere din oppfatning til noen andre (Radford, 2018, s. 5).

For å bevise og kommunisere hvorfor summen av to oddetall blir et partall, må elevene først tenke generelt om strukturen til oddetall. Som nevnt ovenfor, oppdager de at partall kan skrives på formen $2n$, og videre at oddetall kan skrives på formen $2n-1$. Deretter blir det: $2n-1+2n-1=4n-2=2(2n-1)$. Som er det samme som $2*(mengden\ av\ naturlige\ tall)$. Den matematiske strukturen blir derfor generalisert til å gjelde i alle tilfeller med naturlige tall. De kan også bruke variabler til å uttrykke generelle tall i divisjon med to brøker for å bevise hvorfor man kan snu brøken og multiplisere, og videre hvordan dette da kan gjelde for alle mulige tall: $\frac{\frac{a}{b}}{\frac{c}{d}}=\frac{\frac{a}{b}*b*d}{\frac{c}{d}*b*d}=\frac{a*d}{c*b}$.
Slike eksempler vil være sentralt å jobbe med videre. Vi ser dette virkelig aktualisert i Utdanningsdirektoratets utkast til kommende matteeksamen etter nye læreplan, våren 2022 – en

eksamen som for øvrig har blitt avlyst (se vedlegg 2). Der blir elevene blant annet spurt om å bevise hvorfor summen av fem påfølgende heltall er delelig med fem. De blir bedt om å vise sin kompetanse i abstraksjon og generalisering for å få til dette; dette krever at elevene også har god forståelse for disse begrepene. Dette kan de få til ved å tenke abstrakt: $n + (n + 1) + (n + 2) + (n + 3) + (n + 4) = 5n + 10 = 5(n + 5)$. For å få til dette, kreves det god kompetanse i elementene løftet frem i dette underkapittelet.

Vi finner stor støtte fra dette i læreplanen. Kjerneelementet abstraksjon og generalisering nevner at generalisering i matematikk handler om at elevene oppdager sammenhenger og strukturer og ikke blir presentert for en ferdig løsning, og hvordan dette skal formaliserer ved å bruke algebra (Utdanningsdirektoratet, 2020c). Kjerneelementet resonnering og argumentasjon støtter også oppunder dette, der det blant annet nevnes at elevene skal begrunne matematiske regler, slik som eksempelet ovenfor med divisjon av brøk (Utdanningsdirektoratet, 2020c). Vi ser også at et kompetansemål etter 8.-klasse heter å beskrive og generalisere mønstre med egne ord og algebraisk (Utdanningsdirektoratet, 2020b). Derfor vil generalisering og studiet av strukturer bli vektlagt videre i oppgaven, og som del av operasjonaliseringen. Dette er viktige elementer for å få til utforming av algebraiske uttrykk, samt den relasjonelle forståelsen for algebraiske uttrykk.

2.3.3 Konseptualisering: Studiet av funksjoner og algebra som modelleringsspråk

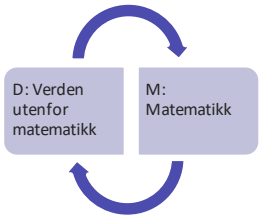
På grunn av sin abstrakte karakter, er ikke algebra et enkelt emne å undervise i, spesielt når algebra presenteres uten noen kontekst, og som nakne likninger og formler (Arcavi, Drijvers, & Stacey, 2017, s. 81). Gledeligvis er algebra så mye mer enn dette, for den andre og tredje delen Kaput nevner i sin beskrivelse av algebra, er nemlig studiet av funksjoner og algebra som et modelleringsspråk både i og utenfor matematikken (2008, s. 11). Førstnevnte, altså funksjoner, beskriver han som en form for generalisering der man beskriver systematisk variasjon av flere tilfeller på tvers av et domene – variabelen x (2008, s. 13). Vi ønsker altså å beskrive variasjonen for ulike verdier av variabelen x, ved å tildele hver verdi av x til en verdi av y. For eksempen på formen $y = ax + b$. Da vil endring i den ene mengden, x, føre til en bestemt endring i den andre mengden, y. Man ønsker å uttrykke ulike former for endring, eksempelvis lineær endring eller eksponentiell vekst (Kaput, 2008, s. 14). Denne grenen inneholder en rekke variabler og symboler, så vel som tabeller, grafer og pedagogiske systemer som funksjonsmaskiner som behandler input og output (Kaput, 2008, s. 14). Funksjonsmaskiner kjenner vi igjen fra norsk skolesammenheng, og er en god metode for elever å bli kjent med variabler, der man setter inn en verdi for x, og maskinen foretar symbolsk manipulasjon og spytter ut en y-verdi (Kongsnes & Wallace, 2020a, s. 208).

Kieran beskriver hvordan det er i denne delen av algebra at vi kan bedrive virkelig meningsfull matematisk aktivitet, for her vil det introduseres kontekst og formål med den algebraiske aktiviteten (2007, s. 714). Det er her vi kan ta i bruk generaliseringen og studiet av strukturer til å bruke algebraen som et verktøy i aktiviteter utenfor selve algebraen (Kieran, 2007, s. 714). Dette vil virkelig åpne opp for mer relasjonell forståelse da elevene kan få meningsfylte kontekster å relatere det symbolske til, og det åpner derfor opp for flere representasjoner av de matematiske objektene. Det er artikulert en reformistisk måte å lære algebra på som vektlegger funksjoner, ulike måter å representere funksjonelle situasjoner på, og til løsning av virkelig problemer med andre metoder enn manuell symbolsk manipulasjon, slik som i for eksempel teknologistøttede læringsmiljøer som geogebra (Kieran, 2007, s. 709). Disse undervisningsmetodene reflekterer holdningen formulert av Fey & Good om at å øve på manipulerende ferdigheter bør erstattes med studiet av familier med elementære funksjoner (1985, referert i Kieran, 2007, s. 709). Dette støtter også oppunder poenget nevnt i forrige underkapittel om at symbolsk manipulasjon ikke bør vektlegges i så stor grad i algebraopplæringen. For går vi tilbake til Duval sin beskrivelse av matematisk forståelse, så er det som nevnt å navigere mellom ulike representasjoner av matematiske objekter, og ved å jobbe med funksjoner, kan man navigere mellom symbolsk og grafisk representasjon for å øke forståelsen for blant annet algebraiske eller matematiske strukturer. Det introduseres et nytt semiotisk register – det visuelle (Duval, 2006, s. 108). Dette beskrives som teknologiens største innvirkning på algebraopplæringen ved at elever øker forståelse ved hjelp av grafiske representasjoner og hvordan disse representasjonene lar et betydelig antall elever visualisere den symbolske formen til en funksjon – eksempelvis geogebra (Kieran, 2007, s. 727). Studier viser dessuten også at når elever som presterte svakt i symbolsk manipulasjon fikk mulighet til å løse virkelige problemsituasjoner med teknologiske redskaper som støtter grafisk representasjoner av funksjoner, utkonkurrerte disse elevene de elevene som presterte høyt i symbolsk manipulasjon (Kieran, 2007, s. 728). Vi skal videre også se hvordan andre digitale læringsmiljøer kan støtte algebraopplæringen på dypere kognitive plan.

Arbeid med funksjoner kan derfor betraktes som algebra. Vi kan også lage funksjoner for å beskrive endring i figurtall, areal av figurer og andre matematiske temaer som også kan beskrives algebraisk. Funksjoner kan derimot også brukes til å beskrive mye utenfor matematikken, til blant annet realistiske situasjoner slik som ytret ovenfor. Kaput har som nevnt satt studiet av funksjoner og algebra som et modelleringsspråk i hver sin gren av algebradefinisjonen. Jeg velger derimot i likhet med Kieran, å sette disse sammen, nettopp på grunn av slektskapet mellom disse to grenene (2007,

s. 714). Matematisk modellering er en algebraisk aktivitet der vi gjerne beskriver en situasjon i algebraisk bekledning ved å ta i bruk variabler, hvor variabelen enten kan representere en klasse av situasjoner eller en ukjent, hvor vi deretter gjennomfører symbolsk aritmetikk for å for eksempel løse et problem eller belyse nye sider av situasjonen (Kaput, 2008, s. 14). Det kan være at vi ønsker å beskrive et mønster i situasjoner eller fenomener både i og utenfor matematikken, slik som for eksempel å beskrive oddetall på generalisert form eller hvordan bankkontoen vil se ut om 10 år inkludert renter. Mønstrene vi beskriver fra virkelige situasjoner tar ofte form som en funksjon, og det er nettopp her disse grenene forenes, slik som nevnt ovenfor om hvordan realistiske problemer ofte assosieres med funksjoner. For den funksjonelle tilnærmingen til påkalling av algebraisk tenkning, legger fokuset på arbeid med funksjoner gjennom møter med virkelig situasjoner der kvantitative forhold kan beskrives av disse matematiske modellene (Kieran, 2007, s. 709). Så vi ser hele tiden en glidning over til matematisk modellering. Derfor tenker jeg som flere, å bruke matematisk modellering som et begrep som er bredere, for den vil nettopp omfavne både funksjoner, forming av algebraiske uttrykk, generalisering, abstraksjon, resonnement og mer. Dette gir oss mulighet til å bruke kontekster eller situasjoner som elever kjenner til fra før av, slik at vi kan bruke elevenes nett av sammenhenger til å skape relasjonell forståelse. Da kan vi bygge på elevenes tidligere erfaringer, og elevene vil se den symbolske algebraen i relasjon til noe annet. På den måten kan vi få til tilpasset opplæring, fordi vi kan møte elevene på deres bakgrunnskunnskap, og vi kan inkludere alle ved å bygge på situasjoner som alle kjenner til. Matematikken vil også være mer meningsfylt og relevant – mer hensikt som etterlyst i starten av dette underkapittelet.

En matematisk modell i bredere forstand består av den virkelige verden, D, og det matematiske domenet M, og arbeid frem og tilbake mellom disse – slik som i figur 2.1 (Niss et al., 2007, s. 4). Det kan være deskriptive matematiske modeller som beskriver en del av verden, eller normative matematiske modeller som forsøker å forutsi eller skape deler av verden (Blum, 2012, s. 77).



***Figur 2.1**: Matematisk modellering (Niss et al., 2007, s. 4)*

Denne brede definisjonen beskriver at matematisk modellering handler om å jobbe matematisk med situasjoner utenfor selve matematikken. Slik vil all algebraopplæring vi bedriver i klasserommet hele tiden være meningsfylt, samlende og ha en kontekst, slik som etterlyst i begynnelsen av dette underkapittelet; det kan bygges en bro mellom klasserommet og verden utenfor. Blum beskriver dette som en kognitiv krevende aktivitet for elevene, for modellering er krevende; det krever både abstraksjon og selvstendighet (2012, s. 78).

For å få til matematisk modellering, er vi avhengig av å matematisere. Matematisering, samme med abstraksjon, innebærer å lage matematikk i symboler – nærmere bestemt i algebraisk bekledning (Jupri & Drijvers, 2016, s. 2483). Matematisering refererer til aktiviteten med å organisere og studere enhver form for virkelighet med matematiske midler, som vil si å oversette et realistisk problem til den symbolske matematiske verden, og vice versa, samt å reorganisere og rekonstruere innenfor matematikken verden for å få ny innsikt – en viktig del av matematisk modellering (Jupri & Drijvers, 2016, s. 2483). Den første delen, det å oversette realistiske problemer eller situasjoner til den symbolske matematiske verden, kalles horisontal matematisering, og det ansees også som den mest kognitivt krevende delen av blant annet modelleringsprosessen (Jupri & Drijvers, 2016, ss. 2483-2499–2450); dette står i likhet med at å beskrive en struktur algebraisk også var det mest krevende i generaliseringsporsessen slik som Radford nevnte i forrige underkapittel. Det å løse tekstoppgaver i algebra innebærer horisontal matematisering, og det nevnes at det å løse tekstoppgaver i algebra er for mange elever over hele verden en av hovedutfordringene i matematikk (Jupri & Drijvers, 2016, s. 2481). Mange studier nevner elevenes utfordringer med å lære algebra i prosessen med matematisering av hverdagssituasjoner til matematiske former (Jupri & Drijvers, 2016, s. 2482). Dette krever en overgang fra det verbale registeret, kontekst, til det symbolske registeret. Denne overgangen ansees som svært kognitivt krevende, men kan på en annen side skape en synergi og en større forståelse for det matematiske objektet (Duval, 2006, s. 124). Denne overgangen blir sentral videre i avhandlingen, for det er nettopp denne man må få til,

som nevnt innledningsvis, for å håndtere kjerneelementet modellering og anvendelser, samt flere andre kompetansemål. Det krever også relasjonell forståelse, nettopp fordi man må se det symbolske i relasjon til det kontekstuelle.

Den andre delen kalles vertikal matematisering og refererer til aktiviteten med å reorganisere og rekonstruere innenfor symbolverdenen som inkluderer løsning av problemer, generalisering og formalisering – altså en kombinasjon av generalisering, strukturer og aritmetikk (Jupri & Drijvers, 2016, s. 2483). Her vil man navigere frem og tilbake mellom ulike representasjoner av det matematiske objektet innenfor samme semiotiske register – det symbolske. Dette krever god kompetanse i strukturell forståelse av matematiske uttrykk. Det er viktig å ha et reflektert forhold til disse to delene av matematisering, slik at man kan analysere og kartlegge hvor i modelleringsprosessen elevene vil ha størst utfordring og eventuelt hvordan man kan håndtere dette som lærer.

Utdanningsdirektoratet har som nevnt laget et utkast til kommende eksamen i matematikk, våren 2022, etter den nye læreplanen. Her ser vi eksempler på hvordan elevene skal matematisere og lage modeller i oppgave 10 (se vedlegg 3). Vi ser også hvordan elevene må matematisere og lage en matematisk modell for situasjonen i oppgave 7 (se vedlegg 1), henholdsvis ved å lage to likninger for så å eventuelt løse disse grafisk slik som på vedlegg 4. Da er det selve den horisontale matematiseringen som blir utfordringen for elevene som nevnt, da selve aritmetikken kan overlates til dataprogrammet Geogebra. Det må også nevnes at matematisk modellering ikke nødvendigvis behøver abstraksjon; altså å bruke variabler, selv om definisjonen til Kaput krever dette. Det kreves likevel en form for abstraksjon når man skal presentere modellen, samt når man skal bedrive såkalt vertikal matematisering der man bruker modellen til å belyse nye sider eller hvis man tar i bruk matematiske funksjoner. Det å jobbe abstrakt er noe vi kommer tilbake til når begrepet algoritmisk tenkning blir tatt opp videre i avhandlingen. Poenget her er hvordan elementene nevnt i dette underkapittelet blir aktualisert på en kommende og standardisert eksamen på ungdomstrinnet i norske skole.

Den brede beskrivelsen av matematisk modellering, som også legger mer fokus på den virkelige verden, er den definisjonen jeg velger å lene meg på i denne avhandlingen, og dette er en definisjon som så vel kan inkludere funksjoner i sitt repertoar. Burkhardt beskrev hvordan fremtiden av algebraundervisning bør inneholde modellering som en bro til algebra (Kieran, 2007, s. 726). En måte å sørge for at elevene oppfatter hensikten med algebra på er å få dem til å jobbe med oppgaver

som har et meningsfullt resultat for elevene selv, og kan gi dem følelse av myndiggjøring som gjør dem i stand til å forstå situasjoner de ellers vil forbli delvis forstått uten (Arcavi et al., 2017, s. 81). Autentiske kontekster kan invitere til algebraisk aktivitet, og kunstige situasjoner bør dessuten unngås (Arcavi et al., 2017, s. 85). Da vil elevene ha virkelig situasjoner som kan øke forståelse for variabler og abstraksjon, samt som en motivator. Dette ser vi støtte fra helt tilbake i opplæringslovens formålsparagraf: Opplæringen skal utvikle kunnskap for at elevene skal kunne mestre livene sine og delta i samfunnet (Opplæringsloven, 2008, § 1–1). Det betyr at arbeid i klasserommet skal knyttes opp mot virkeligheten. Det understrekes også i matematikkfagets relevans og sentrale verdier at matematikkfaget skal forberede elevene på samfunnet (Utdanningsdirektoratet, 2020a). Derfor blir det å jobbe med matematikk opp mot virkeligheten, så vel som algebra, en selvfølgelighet. Det står også i opplæringslovens at ett av formålene med utdanning er at elevene skal lære å tenke kritisk (Opplæringsloven, 2008, § 1-1). Det tverrfaglige temaet demokrati og medborgerskapet artikulerer hvordan matematikkfagets ansvar er at faget skal gjøre elevene bevisste på forutsetninger og premisser for matematiske modeller som ligger til grunn for beslutninger i deres eget liv og i samfunnet (Utdanningsdirektoratet, 2020d). Arbeid med matematisk modellering blir derfor en del av skolens bredere samfunnsmandat om å utvikle dannede mennesker til et demokrati (Utdanningsdirektoratet, 2020e). Kjerneelementet modellering og anvendelser beskriver at en matematisk modell er en beskrivelse av virkeligheten i matematisk språk, og at å forstå seg på matematisk modellering for elevene er viktig for å leve i et samfunn som styres av slike modeller, og for å kunne få bedre innsikt i virkelige situasjoner (Utdanningsdirektoratet, 2020c). Vi ser derfor hvordan kunnskapsstatusen går hånd i hånd med læreplanen og til og med lovverk. Dessuten ytrer kompetansemålene etter 10.klasse at man skal kunne lage likningssett av praktiske situasjoner, bruke funksjoner i modellering, samt å modellere situasjoner knyttet til reelle datasett (Utdanningsdirektoratet, 2020b). Utfordringer i horisontal matematisering i algebra vil derfor ha betydelige innvirkninger på elevene utover algebraen i seg selv. Derfor vil dette kapittelet om konseptualiseringen vektlegge modellering, funksjoner, og det å matematisere virkelige situasjoner – abstraksjon. For å få til dette må elevene også ha god kompetanse i elementene nevnt i underkapittel 2.3.2. Alt dette knyttes også sammen med utforming av algebraiske uttrykk, som er der vekten i denne masteravhandlingen vil ligge.

2.3.4 Algebraisk tenkning og kvasialgebra

Jeg har kommet frem til en aktuell definisjon av algebra i dagens opplæring som vektlegger studiet av strukturer, generalisering og matematisk modellering. For å få til dette bruker vi symboler i form

av variabler – som nevnt. Kaput nevner som nevnt symbolisering er den del av kjernen i algebraisk aktivitet (Kaput et al, 2008, s. 20). Tegn er en meningsbærende enhet i kommunikasjon mellom mennesker, og læren om tegn, hva de uttrykker, og hvordan de tolkes, kalles semiotikk (Hinna et al., 2011, ss. 1059–1061). Noen tegn er vanskeligere å tolke enn andre, slik som symboler, nettopp fordi symboler ikke likner på det som tegnet assosieres med eller står for (Hinna et al., 2011, ss. 1062–1066). Vi sier at tegn, eller symboler om du vil, står for noe annet, og dette «noe annet» er en referent (Kaput et al., 2008, s. 27). Det er ofte et stort kognitivt gap mellom symbol og referent (Kaput er al., 2008, s. 27). Undervisning generelt handler egentlig om å skape referenter for elever ut ifra deres bakgrunnskunnskap eller erfaring (Kaput et al., 2008, ss. 27–28). Sånn at vi kan bruke og utvikle elevenes nett av sammenhenger. Algebraiske symboler har derimot ingen konkrete referenter; de er enda mer abstrakte. I en dyp og langsiktig forstand tillater symboliseringsprosessen i algebra et slags løft fra grensene for konkret, referert tenkning, den lar oss overvinne minnet og prosesseringsgrensene til rå menneskelig erkjennelse, og introduserer nye muligheter for abstraksjon bort fra erfaringens spesifikasjoner (Kaput et al., 2008, s. 23). Radford beskriver hva som skjedde med elevene da de fikk til generalisering og abstraksjon ved hjelp av symboler: Ved å gjøre dette, gikk elevene inn i en ny form for algebraisk forståelse og beveget seg inn i et dypt område av sonen for fremvekst av algebraisk tenkning, der de beveget seg fra en referanseforståelse av tegn, til en morfologisk forståelse – og det er her den virkelig strukturelle forståelse oppstår (2008, s. 24). Det kan derfor bli krevende å tenke algebraisk uten å ha noe som helst å referere til. Men hele poenget er at algebra er noe å tenke med, ikke på (Kaput et al., 2008, s. 25). Og det blir derfor vanskelig når det man er så avhengig av å bruke algebraisk symbolspråk for å tenke algebraisk. Elevene trenger å bli kjent med dette semiotiske redskapet for å kunne tenke som mennesker gjør i matematikk. Algebraisk symbolspråk medierer denne tenkemåten, og muliggjør den i stor grad for oss mennesker, men den er dessverre ikke så enkelt internalisert som beskrevet ovenfor. Dessverre forbinder mange matematikk og algebra med alt annet enn tenkning, men som et kommunikasjonsspråk som nevnt (Hinna et al., 2011, s. 174). Derfor kan det være at vi kan åpne opp for andre innganger til algebraisk tenkning, innganger som har overføringsverdi til algebraisk symbolspråk, slik at internaliseringen av dette ikke blir så krevende.

Tidlig algebra-bevegelsen er i gang verden over, og er en bevegelse som forsøker å avgjøre om unge elever virkelig kan begynne å lære algebra og om tidlig eksponering for elementære algebraiske konsepter kan lindre velkjente vansker som ungdom møter i algebraopplæringen (Radford, 2018, ss. 2–3). Slike ideer strider mot den tradisjonelle læreplanoppfatningen om at algebra bare kan læres etter at elevene har tilstrekkelig kunnskap om aritmetikk, som inntil nylig

ekskluderte algebra fra barneskolen i mange læreplaner rundt om i verden (Radford, 2018, ss. 2–3). Mer vekt blir i denne bevegelsen lagt på mønstre, systemer og variable størrelser, og ikke uttrykksformen (Hinna et al., 2011, s. 175). For hele poenget er jo at elevene skal tenke algebraisk, og fra tidlig alder, og ved å jobbe med slike aktiviteter, kan de dette uten å være avhengig av avansert aritmetikk. Forskernes erfaring er at barneskoleelever, som går denne veien som vektlegger algebraisk tenkning først, kan lære å bruke bokstaver til å uttrykke en variabel størrelse relativt tidlig – før ungdomsskolen (Hinna et al., 2011, s. 175). Altså å lage algebraiske uttrykk.

Gå vi derimot bort fra tanken om at algebraisk tenkning må inneholde symboler som det eneste semiotiske systemet for å få til dette på, kan det være andre semiotiske systemer vi kan bruke for å uttrykke algebra eller tenke algebraisk (Radford, 2010, ss. 1–2). Radford foreslår at kjernen i algebraisk tenkning er å behandle ubestemthet på analytiske måter, og han hevder at det bør være flere semiotiske systemer som er rustet til å uttrykke dette (2010, s. 3). Ontologisk sett er det rom for en stor konseptuell sone hvor elevene kan begynne å tenke algebraisk, selv om de ennå ikke tyr til algebraiske symboler, og denne sonen kan kalles for fremvekst av algebraisk tenkning (Radford, 2010, s. 3). Vi kan derfor tenke oss at elever kan begynne med dette allerede på barneskolen slik som i algebra tidlig-bevegelsen. Elevene kan for eksempel argumentere for sammenhengen mellom konkrete figurtall ved å bruke rytmekoordinerte gester eller språk – finne en struktur eller et mønster (Radford, 2010, s. 7). De kan deretter videre få beskjed om å håndtere ubestemtheten ved å forsøke å beskrive en generell figur ved å bruke lingvistikk, som for eksempel ved å formidle en fortelling eller annet (Radford, 2010, s. 7). Her øver elevene på å tenke generelt før de har lært grunnleggende aritmetikk og kommunikasjonsspråk. De kan bruke ord som blant annet figurnummer eller generell figur (Radford, 2010, s. 8). Elevene er allerede her i gang med å uttrykke ubestemthet ved å bruke et annet semiotisk register – verbalspråk eller kontekst. Det er etter dette at det blir krevende, for det er her elevene må gå over til å uttrykke det med symbolspråk (Radford, 2010, s. 8). Det er derimot en god øvelse, og følger man denne framgangen vil elevene bli bedre rustet til å håndtere generalisering. Det siste nivået av abstraksjon understreker også det aller mest krevende, og det er når elevene har funnet en algebraisk formel som tilfredsstiller figurtallene, men hvor formelen følger figurenes narrativ (Radford, 2010, s. 11). Det kan være at den ikoniske formelen, som likner på figurene, blir beskrevet slik: $(n+1)+(n+2)$. Da refererer formelen, eller det algebraiske uttrykket, til noe delvis konkret. Det er fortsatt ikke helt algebraisk og abstrakt. Hvis de derimot får til å kollapse narrative, kan de gjøre en transformasjon: $(n+1)+(n+2)=2n+3$, og det er da vi virkelig går inn i algebras abstrakte karakter (Radford, 2010, s. 11). Vi ser også hvordan behandling innenfor det samme semiotiske register muliggjør høyere abstraksjon og

generalisering. Poenget er at det er mange nivåer elevene må gjennom, for noe som kan virke ganske trivielt for oss som matematikklærere. Det er også mange måter å få til algebraisk tenkning på – ubestemthet. Vi kan derfor arbeide med de tidligere nivåene allerede på barneskolen, slik at elevene øve på ubestemthet – algebraisk tenkning.

Dette åpner opp for nye kommunikasjonsformer for algebra. Det inviterer også til nye måter å lære algebra på, nye måter å skape relasjonell forståelse på, der elevene forstår hvorfor de kan gjøre som de gjør. Kvasivariabler definerer vi som kvasialgebraisk hvis den tilfredsstiller samme betingelser som definisjonen på algebra bortsett fra at den kan bruke alle symboler, inkludert tradisjonelle aritmetiske, eller mer uformelle som for eksempel verbalspråk (Kaput et al., 2008, s. 49). Hvis man bruker dette for å uttrykke generalisering, kvalifiserer det som kvasialgebra (Kaput et al., 2008, s. 49–50). Dette vil derimot begrense elevenes mulighet til å bedrive symbolsk manipulasjon eller vertikal matematisering, men det er ikke her problemet ligger slik som vi så i underkapittel 2.3.2 og 2.3.3. Uansett så åpner dette opp for at vi kan bedrive kvasialgebraisk aktivitet i klasserommet som en inngang til å lære algebraisk tenkning. Det åpner også opp for andre semiotiske registre, slik som for eksempel programmeringsspråk som jeg kommer tilbake til. Ved å tenke på algebraisk tenkning generelt som arbeid med ubestemthet, vil man overkomme problemet nevnt tidligere om at algebra blir et kommunikasjonsspråk. Derfor oppsummeres dette underkapittelet ved at man kan tenke algebraisk uten å nødvendigvis ta i bruk konvensjonell algebraisk symbolisering, og at algebraisk tenkning kan betraktes som behandling av ubestemthet på en analytisk måte. Masteravhandlingens tittel har også sitt utspring her, nettopp fordi algebraiske uttrykk er abstrakte, og de mangler konkrete referenter. Hvordan vi kan gjøre det abstrakte mer konkret, er noe som skal løftes frem i underkapittel 2.5.1.

2.4 Programmering og algoritmisk tenkning

2.4.1 Innledning til programmering i skolen

Som nevnt innledningsvis, er programmering noe som har blitt innlemmet i grunnopplæringen på global basis (Sentance & Csizmadia, 2017, s. 470). Det er en del av det globale tiltaket om å styrke elevers digitale kompetanse og problemløsningskompetanse (OECD, 2016, s. 3). Med tanke på det økende digitaliserende samfunnet, har programmering og algoritmisk tenkning dukket opp som nødvendige ferdigheter ikke bare for informatikere og ingeniører, men også for alle innbyggere (Bråting & Kilhamn, 2021, s. 170). Algoritmisk tenkning kan kort fortalt beskrives som en problemløsningsmetode der vi tenker som en informatiker – og derfor knyttes problemløsning og programmering sammen (Utdanningsdirektoratet, 2019b). Det innebærer den mentale aktiviteten ved å formulere problemet til å kunne løses av en datamaskin, og får å få til dette er man avhengig av abstraksjon (Wing, 2017, s. 8).

Problemløsning ansees som en sentral kompetanse i samfunn og arbeidsliv, og matematikkfaget har her fått et stort ansvar (Utdanningsdirektoratet, 2020a). Problemløsningsoppgaver er oppgaver der man ikke på forhånd har en ferdig oppskrift for å løse oppgaven, oppgaver med høye kognitive krav, der elever utfordres til å bruke det de kan i nye sammenhenger, samt å utvikle elevenes forståelse for matematiske konsepter gjennom arbeidet med å løse oppgavene (Karlsen, 2014, ss. 33–35). Det åpner opp for mer relasjonell forståelse, og matematikkfaget blir mer enn bare ferdighetstrening (Karlsen, 2014, s. 34). Vi ser derfor at problemløsning er bra for matematikkferdigheter så vel som videre deltakelse i samfunnet, og at algoritmisk tenkning blir en viktig problemløsningsmetode. Jo Boaler er en aktiv forkjemper for dette, og forklarer at matematikkundervisning bør være bestående av åpne og utforskende oppgaver, gjerne problemløsningsoppgaver: «ask the problem before showing the method» (2016, ss. 76–81). Ved å jobbe med programmering kan det bygges bro mellom teori og praksis der elevene jobber med og skaper teknologi; og slik vil det de lærer og jobber med ha overføringsverdi ut av klasserommet (Sevik, 2016, s. 18). Det er nettopp det som er målet med relasjonell forståelse og dybdelæring: at kunnskapen skal ha overføringsverdi til nye situasjoner. Vi ser at programmering, algoritmisk tenkning og problemløsning har fått større plass med fagfornyelsen i 2020, og hvordan disse tre elementene komplementerer hverandre (Utdanningsdirektoratet, 2020f).

Vi kan dele argumentene for å innlemme programmering i grunnopplæringen i to deler – jeg har valgt å dele det inn i mikro- og makronivå. I den første delen ser vi på samfunnet på mikronivå, for

Marc Durando, leder for European Schoolnet, nevner at det ble observert i England at elever ikke hadde gode nok digitale ferdigheter til å håndtere karrierene sine, hvor da programmering og algoritmisk tenkning ble sett på som sentralt manglende ferdigheter (Bocconi et al., 2018). Han nevner også at det er nødvendig å forstå seg på hvordan programvarekode og algoritmer blir satt i arbeid i ulike sosiale, politiske, kulturelle og økonomiske kontekster (Bocconi et al., 2018). Dette argumentet, som går på mikronivå, dreier seg om elevenes muligheter. Deres mulighet til å delta i samfunnet på en god måte. Som nevnt innledningsvis, ble det etterlyst av igangsetteren til fagfornyelse at grunnopplæringen skal legge vekt på at elevene skal forstå og produsere IKT, fremfor å være konsumenter av det, samt at programmeringstilbudet skulle bygges ut (Meld. St. 28, (2015–2016), ss. 32-54). Forståelsen for IKT er viktig, for de tverrfaglige temaene folkehelse og livsmestring og demokrati og medborgerskap aktualiserer dette fra et mikroperspektiv i læreplanen: For å kunne delta i samfunnsdebatten kreves det forståelse for statistiske og teknologiske modeller (Utdanningsdirektoratet, 2020d). En undersøkelse fra 2013 viser at nærmere en fjerdedel av norske elever på 9. trinn har så svake digitale ferdigheter at de vil ha problemer med å kunne delta fullt ut i skole-, yrkes-, og samfunnsliv (Meld. St. 28, (2015–2016), s. 32).

Et annet argument på mikronivå peker på at elevene skal tilegne seg ny kompetanse gjennom blant annet programmering, kompetanse de vanligvis ikke kunne lært – slik som nevnt innledningsvis (Wing, 2017, s. 1). Utdanningsdirektoratet nevner blant annet at elevene skal bruke programmering til å utforske og løse problemer, og hvordan det kan være en god metode for å utvikle matematisk forståelse (Utdanningsdirektoratet, 2020f). Det er også argumentert om matematikk og programmerings delte gener, samt forskning som nevner denne likheten med blant annet slagordet: «coding to learn» (Popat & Starkey, 2019, s. 373; Misfeldt & Ejsing-Duun, 2016, s. 2524). Dette er ikke så mye nevnt i dagens policy-dokumenter, men vi ser hvordan matematikkfaget har fått hovedansvaret med programmering og algoritmisk tenkning (Utdanningsdirektoratet, 2020f). Dette kommer jeg tilbake til i kapittel 2.5 – kjernen i avhandlingen.

Det andre delen av argumentasjonen omhandler samfunnet på makronivå, og handler om økonomisk vekst og innovasjon på lands- og globalbasis (OECD, 2016, ss. 1–2). Dette handler om hva samfunnet trenger i sine aktører, for utdanning skal sørge for økonomisk vekst innad landene (Biesta, 2018, s. 248). Det kan trekkes så langt som Williamson påpeker: Denne utformingen av befolkningens digitale ferdigheter forbereder dem som idealdeltakere mot den digitale styringen og propaganda fra den motvillige stat (2016, s. 54).

Jeg skal derfor videre konseptualisere hva programmering i skolen vil innebære og hvordan det blir aktualisert i dagens skole – vi har jo nå sett hvorfor det skal bli inkorporert. Deretter vil jeg konseptualisere det sentrale begrepet algoritmisk tenkning som har dukket opp i flere policy-dokumenter, læreplanverk og litteratur. Avslutningsvis vil jeg gå inn på den mulige kausaliteten mellom programmering i skolen og algebraferdigheter, som jo vil være grunnlaget for det som skal testes ut senere i avhandlingen – kjernen i denne avhandlingen.

2.4.2 Konseptualisering av programmering i skolesammenheng

2.4.2.1 Hva er programmering?

Et dataprogram er en samling detaljerte instruksjoner som beskriver hvordan et problem kan løses, og i det vi skriver programmet, programmerer vi (Bueie, 2019, s. 22). Et annet begrep kalt «koding» brukes også i flere sammenhenger, et begrep som gir konnotasjoner til noe mer lekende og ufarlig (Sevik, 2016, s. 9). Jeg vil forholde meg til begrepet programmering da dette også omfavner prosessen med å komme frem til programkoden samt identifisering av problemer og mulige løsninger (Sevik, 2016, s. 9). Prosessoren i en datamaskin håndterer hele tiden et hav av binærkode – sekvenser med 0 og 1 (Haverbeke, 2009, ss. 11–12). Dette er datamaskinens språk, språket som blir overført gjennom ledninger via for eksempel ulike spenningsnivåer. Forskjellige kombinasjoner av 0 og 1 vil være forskjellige handlinger eller kommandoer for datamaskinen. En byte, bestående av 8 binære tall, utgjør totalt $2^8 = 256$ mulige forskjellige kommandoer. Når en prosessor behandler flere millioner bytes i sekundet, forteller det noe om hva en datamaskin kan gjennomføre i forhold til oss mennesker. Vi sier ofte at å kode er det nærmeste vi kommer superkrefter, for kan du kode, kan du utrette ting som før var utenkelig (Lær Kidsa Koding, uå). Dette språket, også kjent som maskinkode, er altfor krevende for oss å skrive – spesielt i opplæringssammenheng. Derfor finnes det andre programmeringsspråk kalt høynivåspråk som vi kan skrive med som blir kompilert, via en kompilator eller en tolk (interpreter), til binærkode som datamaskinen forstår.

Programmeringsspråket som gjerne blir brukt i norsk skolesammenheng heter Python (Lær Kidsa Koding, uå). Det beskrives som et programmeringsspråk med lav inngangsterskel og stor takhøyde (Bueie, 2019, s. 30). Språket er svært allsidig og blir på grunn av sin lettleste og veldefinerte skrivemåte, syntaks, et naturlig valg i opplæringsøyemed (Bueie, 2019, s. 31). Det at programmeringsspråk og programmering som sådan defineres som aktivitet med lav inngangsterskel og stor takhøyde, gjør at elevenes lovfestede rettighet om tilpasset opplæring skinner gjennom disse aktivitetene (Sevik, 2016, s. 17; Opplæringsloven, 2008). Flere nye pedagogiske innganger til programmering slik som programmeringsspråket Scratch, et blokkbasert

dra-og-slipp-språk, gjør også at flere har mulighet til å begynne å programmere (Sevik, 2016, s. 17). Dette er et pedagogisk virkemiddel for barn og ungdom, og vi kaller disse brukergrensesnittene for blokkbasert brukergrensesnitt, men dette er ikke noe jeg vil bruke videre i denne avhandlingen (Bueie, 2019, s. 26).

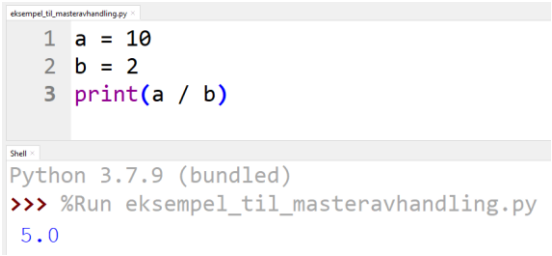
Mange elever ser på Python som noe matnyttig og motiverende da dette er et verktøy som de kan få direkte bruk for senere i livet, i tillegg til at grunnstrukturene i Python er lett overførbare til de fleste andre populære programmeringsspråk (Haraldsrud et al., 2020, s. 14). Haraldsrud et al. argumenterer for bruken av Python i skolesammenheng ved 1) det er ett av de enkleste og mest lettleste tekstbaserte språkene 2) den rene syntaksen gjør det enklere å oversette matematikk til Python-kode 3) det brukes over hele verden, både i akademia og næringslivet 4) det har veldig gode biblioteker med ferdigskrevet kode som kan brukes til praktiske formål, spesielt realfaglige sammenhenger (2020, s. 14). Et eksempel på bruk av biblioteker kan man se i vedlegg 6 og 15, eksempler fra min egen undervisning der man henter metoder fra et bibliotek for å simulere et tilfeldig tall til en sannsynlighetsfordeling, og ett der vi bruker et bibliotek for å få ut en metode for grafisk representasjon av veksten på en sparekonto. Et populært bibliotek kalt Pygame kan også brukes for viderekommende elever som ønsker å lage spill. Da kan elevene se hvordan dataskjermen bruker koordinater slik som et koordinatsystem i matematikken. Dette kan man se i vedlegg 7, men er et eksempel for den mer viderekommende programmereren.

I klasseromsforstand kan vi derfor si at programmering innebærer å utforske og løse matematiske problemer, samt å gjøre dette gjennom programmering og planlegging av programkode; overførbart. Vi ser også matematiske temaer som er direkte overførbare til programmering, noe som gjør at man kan slå to om ikke tre fluer i én smekk: Elevene for jobbet med programmering, de får jobbet med matematikk, og de blir forberedt på det digitale samfunnet.

2.4.2.2 Programmeringsspråkets syntaks og kommandoer som skal brukes videre

Det ville tatt lang tid å løfte frem alle elementene som vil bli brukt i utdanningssammenheng med programmering, men jeg skal nevne de som er relevant for avhandlinger her – de som skal benyttes i eksperimentet. Det første vil være bruk av variabler, for ethvert program er nødt til å oppbevare og holde på data – informasjon (Haraldsrud et al., 2020, s. 21). Variablene blir lagret midlertidig i datamaskinens arbeidsminne – RAM. Dette kan betegnes som datamaskinens indre lager, og her finner datamaskinen en passende lagerplass for verdien til variabelen – og variabelen refererer eller peker til den lagerplassen (Haraldsrud et al., 2020, s. 24). En variabel i matematikk har mye til

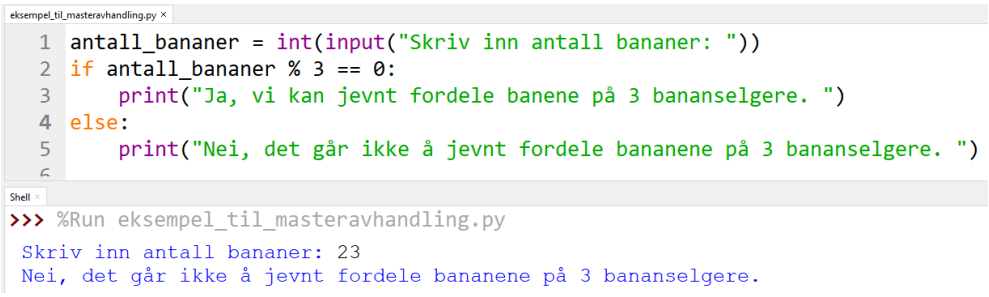
felles med en variabel i programmering (Bueie, 2019, s. 33). Disse variablene kan blant annet holde på tall – heltall eller desimaltall. På heltall så vel som variabler, kan vi gjennomføre regneoperasjoner eller aritmetikk, for vi kan gjøre det meste som vi kan i matematikk så vel som heltalldivisjon (//) og modulo (%) (Haraldsrud et al., 2020, s. 27). Vi kan også gjøre flere operasjoner som for eksempel kvadratrot ved å importere biblioteker med flere lignende funksjoner. Du ser i figur 2.2 hvordan vi kan tilordne tallverdier til variabler, og hvordan vi kan gjennomføre regneoperasjoner på disse som om de var verdier vi kjente til i det vi skriver det ut til konsollen; akkurat som med algebra – slik som Radford forklarte i underkapittel 2.3.



***Figur 2.2**: Variabler i programmering*

Regneoperasjoner i Python følger vanlig regnerekkefølge som vi har i matematikkfaget (Haraldsrud et al, 2020, s. 28). Dette er én av sammenhengene mellom programmering og algebra, og lar elevene enkelt overføre matematiske modeller og utregninger til Python, hvor dataprogrammet tar for seg kalkuleringen og elevene kan realisere sine egne matematiske modeller (Haraldsrud et al., 2020, s. 162).

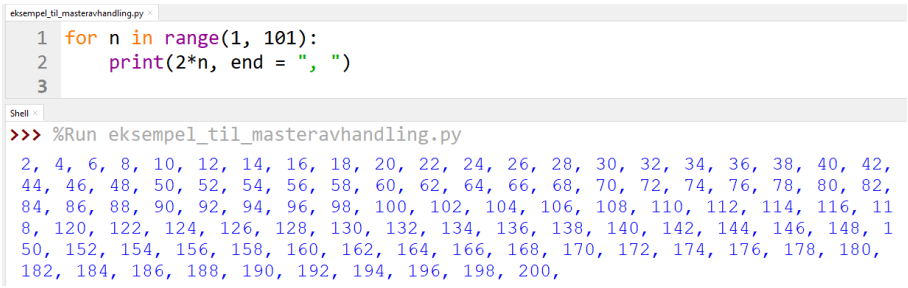
Videre har vi vilkår, noe som betegnes som at noe skjer basert på betingelser – logiske tester som er sanne eller usanne (Haraldsrud et al, 2020, s. 42). Dette kan gjøre dataprogrammene mer dynamiske og i stand til å ta egne valg. Vi kan da programmere mer avanserte modeller. Et eksempel kan være hvis man skal sjekke om et vilkårlig tall er delelig med tre. Tenk at vi har en mengde bananer og lurer på om vi kan fordele disse jevnt på tre bananselgere. Ikke veldig avansert, men et enkelt eksempel.



***Figur 2.3**: Vilkår*

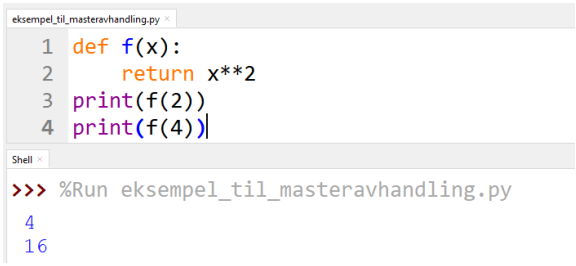
Vi ser i figur 3.3 hvordan vi bruker innebygde funksjoner slik som input() for å ta imot en inntastet heltallsverdi til variabelen som deretter blir gjort om til heltall med funksjonen int(). Deretter blir det foretatt vilkår ved å bruke modulo-operatoren, samt sammenlikning med (==). Her ser vi et skille mellom programmering og matematikk der likhetstegnet brukes forskjellig, for i dette tilfellet betyr det likhet i motsetning til da vi tilordnet verdi til variabler i figur 2.3 ved å bruke ett likhetstegn (Haraldsrud et al., 2020, s. 22). Et tydelig eksempel på hvordan symboler kan oppføre seg forskjellig i ulike semiotiske registre. Det er derfor ikke direkte overførbart mellom de ulike registrene, noe som kan være en utfordring.

Det neste er løkker. Det er her mange setter seg fast med programmeringen, for her er ikke kodeflyten lenger er lineær (Haraldsrud er al., 2020, s. 47). Det er her vi kan gjenta kodelinjer flere ganger, hvor hver repetisjon kalles en iterasjon. Det er her vi virkelig kan ta i bruk maskinkraften og sparer mye skriving. Det kan man se aktualisert i vedlegg 15, der kan se to forskjellige tilfeller der bruken av den ene typen løkke er hensiktsmessig i forhold til den andre: i det første programmet ser man for-løkke med et bestemt antall iterasjoner og i det andre en while-løkke som stopper når et spesifikt kronebeløp er nådd. Vi forholder oss henholdsvis til for-løkker hvis vi har et bestemt antall iterasjoner, eller while-løkker hvis vi ikke vet hvor mange iterasjoner som skal foretas (Haraldsrud, Sveinsson, & Løvold, 2020, ss. 47–49). Det kan være at vi ønsker å iterere gjennom et antall partall fra 2 til og med 200, og dette ser man aktualisert i figur 2.4. Det kan også gjøres med en tredje parameter i range()-funksjonen, men for å øke elevers forståelse for matematiske strukturer, er ikke det å foretrekke. Vi ser også at for-løkken har med seg en variabel som øker med én i verdi for hver iterasjon. Denne opplistingen har mye til felles med det vi kaller tallfølger i matematikk, og muliggjør i stor grad arbeid med dette (Bueie, 2019, s. 88).



***Figur 2.4**: For-løkke*

Man kan dessuten også lage egne funksjoner i Python så vel som de fleste andre programmeringsspråk. Det kan være at man vil finne en bestemt ut-verdi for noe som skal skrives inn. Dette kan likne mye på funksjoner i matematikk (Haraldsrud et al., 2020, s. 52). Vi lager funksjoner når vi ønsker å gjenta en kodesekvens, for da kan vi bare kalle på funksjonen når den skal brukes – slik sparer vi kode (Haraldsrud et al., 2020, s. 52). Her er et eksempel, i figur 3.5, på hvordan å lage en funksjon som tilsvarer $f(x) = x^2 \Rightarrow f(2) = 4 \Rightarrow f(4) = 16$.



Figur 2.5: Funksjon

Vi lager funksjonen ved å skrive «def» før funksjonsnavnet, der vi tar med oss en variabel som innverdi inni parentesene – dette heter også en parameter. Det som funksjonen skal spytte ut, må skrives etter «return». Elever kan for eksempel jobbe med å lage valutakalkulatorer som kan gjentas flere ganger i koden. Mulighetene for hva de kan lage er uendelige. Dette lar elever lage algoritmer og matematiske modeller som kan realiseres av et dataprogram. Vi ser derfor hvordan programmering kan brukes i enkel forstand i opplæringssammenheng, og hvordan det samsvarer med matematikkfaget.

2.4.3 Konseptualisering av algoritmisk tenkning

Jeg skal nå definere hvordan vi tenker og jobber når vi lager dataprogrammer, for dette er en etterlyst og nødvendig ferdighet hos deltakerne i samfunnet som nevnt i underkapittel 2.4.1. Begrepet algoritmisk tenkning nevnes eksplisitt i matematikkfaget innenfor kjerneelementet utforsking og problemløsing, der det beskrives som en viktig prosess med å utvikle strategier og

fremgangsmåter for å løse problemer, og at det innebærer å bryte ned et problem i delproblemer som kal løses systematisk, og hvordan delproblemene kan løses med digitale hjelpemidler (Utdanningsdirektoratet, 2020c). Dette er ikke langt ifra definisjonen på programmering – mye av det samme – men tar for seg mer av aktiviteten utenfor datamaskinen. Som nevnt innledningsvis, er dette en problemløsningsmetode der vi tenker som en informatiker (Wing, 2017, s. 8). Slik kan vi kombinere intelligensen til mennesker og intelligensen til maskiner (Wing, 2017, s. 8). Algoritmisk tenkning handler ikke bare om problemløsning, men like mye om formulering av en algoritme som kan løse problemer ved hjelp av en datamaskin (Wing, 2017, s. 8). Begrepet algoritme betyr en presis beskrivelse av en serie operasjoner som skal utføres for å oppnå et visst resultat (Haraldsrud et al., 2020, s. 183). Det kan være noe så enkelt som en middagsoppskrift, derav kallenavnet oppskrift, eller den velkjente divisjonsalgoritmer som vi bruker i opplæringen, eller mer avanserte matematiske modeller som for eksempel styrer en selvkjørende bil. Wing beskriver abstraksjon som nøkkelen i algoritmisk tenkning, for hvis vi får til det, kan vi få ett objekt til å stå for mange, og vi kan skalere og lage store automatiserte systemer (Wing, 2017, s. 8). Algoritmer vi lager kan også være overførbare til andre lignende problemer, og det kreves også abstraksjon hvis vi skal presentere algoritmen, og for å få til det må vi generalisere (Wing, 2017, ss. 8–9). Vi ser derfor allerede her en stor likhet mellom algoritmisk tenkning og algebra, der abstraksjon og generalisering brukes i begge.

Den algoritmiske tenkeren må være systematisk og analytisk i arbeidet, bryte ned problemer i delproblemer, prøve og feile (utvikling av kognitiv kondis), samt å generalisere i den forstand at metodene kan brukes til å løse lignende problemer eller mer komplekse problemer (Utdanningsdirektoratet, 2019b). Da kommer bruken av variabler, vilkår og løkker virkelig til nytte. Etter at elevene har delt opp et problem i løsbare deler, kan de skissere programmet før de lager dem, og da kan de gjerne skrive pseudokode (Haraldsrud et al., 2020, s. 189). Det vil si at vi skriver en uformell beskrivelse av dataprogrammet for å få oversikt over hva programmet bør inneholder før vi programmerer det – alternativt et flytdiagram. Viktige deler i prosessen er analyse, algoritmer, dekomposisjon, mønstre, abstraksjon og evaluering (Utdanningsdirektoratet, 2019b).

2.5 Synergien mellom algebra og programmering i opplæringssammenheng

2.5.1 Kunnskapsstatus om sammenhengen

Når elevene bruker programmering til å utforske og løse problemer, kan det være et godt verktøy for å utvikle matematisk forståelse (Utdanningsdirektoratet, 2020f). Dette er et viktig argument fra læreplanverket om grunnen til at matematikkfaget har fått hovedansvaret for programmeringsopplæringen etter fagfornyelsen. Dette beskriver Bueie som naturlig ettersom programmering henger svært tett sammen med matematikkfaget og på mange måter har sitt utspring fra matematikkfaget (2019, s. 23). Papert, grunnleggeren av det pedagogiske LOGO-programmeringsspråket som ble brukt mye i opplæringssammenheng på 70- og 80-tallet, skrev at å jobbe med logo, ga elever mer forståelse for spesifikke matematiske temaer som vinkler, grader og variabler, så vel som problemløsning, dekomposisjon og abstraksjon (Papert, 1980, referert i Benton et al, 2017, s. 117). Vi ser mye forskning om dette fra denne perioden, altså programmering for å lære matematikk i klasserommet, før det senere dabbet av. Benton et al. påpeker kjente utfordringer ved programmering i opplæringssammenheng der den avanserte syntaksen og håndtering av feilmeldinger og lite tilgang på digitalt utstyr har vært kjente utfordringer i foregående år, men at dette nå ikke er reelle problemer lenger med nyere programmeringsspråk og mer digitalt utstyr i klasserommene (2017, s. 116). Vi skal dermed se at nyere forskning nå har begynt å se på sammenhengen; mye mulig grunnet mer digitalisering og bedre egnede programmeringsspråk.

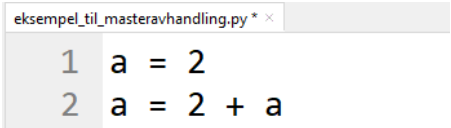
Som nevnt innledningsvis i avhandlingen, er det flere som løfter frem slektskapet mellom disse. Misfeldt & Duun nevner programmering og matematikkens delte gener, og løfter frem en mulig synergi mellom læring av matematikk og programmering – det vil si de støtter hverandre og sammen gjør hverandre sterkere (2016, s. 2525). Kaufmann & Stenseth nevner hele fem studier som argumenterer for at programmering gir sterke matematiske ferdigheter hos elever (2021, s. 1030). Jeg vil ta i bruk synergi-begrepet til Misfeldt & Duun og bruke dette i denne avhandlingen, men jeg vil videreføre dette begrepet til læring av algebra og programmering spesifikt. Dette er for øvrig et begrep som ble introdusert til utdanningssammenheng av Duval, som nevnt i underkapittel 2.2.2, der tanker var at flere representasjoner av et matematisk objekt sammen kan skape større forståelse for det matematiske objektet i forhold til representasjonene hver for seg. Her tenkes det derfor at arbeid med programmering som er et eget avgrenset tema sammen algebraisk symbolspråk som et eget avgrenset tema, vil kunne gi større forståelse for de algebraiske matematiske objektene.

Misfeldt & Duun løfter frem en modell om læringspotensialet gjennom programmering, en modell som forklarer at elevene gjennom programmering vil 1) være produsenter av egen kunnskap 2) lære abstraksjon 3) lære å tenke i algoritmer (2016, s. 2524). Det at de er produsenter av egen kunnskap støtter godt oppunder aktivitetsteori og skjemateorien til Piaget, men i masteravhandlingens tilfelle, relasjonell forståelse og nett av sammenhenger (Hinna et al., 2011, ss. 888–891). Eleven konstruerer kunnskapen selv, basert på det de kan fra før, og ser kunnskapen i relasjon eller sammenheng til andre elementer, og de danner sammenhenger mellom matematikk og programmering, og dette bygges gjennom å lage dataprogrammer. Det at abstraksjon løftes opp som en sammenheng virker naturlig ved bruk av variabler og den algoritmiske tenkemåten – et viktig element i algebraisk tenkning og algoritmisk tenkning som nevnt. Én av ideene deres om abstraksjon stammer fra radikal konstruktivisme der man hevder at all abstrakt tenkning har et konkret utgangspunkt, for hvis datamaskinen kan gjøre numeriske beregninger på inkrementerende variabler i løkker, kan dette være et konkret utgangspunkt, et skjema, som kan videreutvikles til abstrakt algebra, slik som etterlyst i underkapittel 2.3.4 (Misfeldt & Ejsing-Duun, 2016, ss. 2525–2526). Tenker vi i baner av relasjonell forståelse og nett av sammenhenger vil det utvide elevenes nett av sammenhenger og de kan se abstraksjon i relasjon til noe nytt. Måten datamaskinen arbeider med variabler på, kan gi en ny forståelse, en slags referent, til algebraisk tenkning. Slik kan også datamaskinen fungere som et medierende artefakt, og kanskje dette kulturelle artefaktet kan støtte elevene med å innordne seg algebraiske tenkemåter (Rabardel & Bourmaud, 2003, referert i Misfeldt & Ejsing-Duun, 2016, s. 2526).

Vi ser derfor at det sosiokulturelle aspektet skinner gjennom her, og at utfordringer med å tenke abstrakt grunnet mangel på referenter som nevnt i underkapittel 2.3.4 med tanke på semiotikken, kan håndteres ved hjelp av hvordan datamaskinen medierer slike referenter. Noss & Hoyles er også inne på dette: Ved å tilby elevene en datamaskin der de kan uttrykke sine ideer, kan datamaskinen bidra til å gjøre eksplisitt det som er implisitt, både for elev og lærer – konkretisere det abstrakte (5) (1996, s. 5). Noss & Hoyles ser blant annet også på abstraksjon som sentralt i matematikken, og løfter frem tanken om Vygotskys bruk av medierende artefakter, og hvordan datamaskinen er et slikt et for å skape mening for elever i det abstrakte (1996, s. 105). Datamaskinen er derfor et kulturelt artefakt som kan mediere tankemåter som igjen kan videreføres til semiotiske redskaper slik som algebraisk symbolspråk. Det nevnes også hvordan datamaskinen, programmering spesifikt, kan skape nye sammenhenger for elever med eksperimenter og konstruksjon – med tanke på elevenes nett av sammenhenger; litt som at man jobber på labben i naturfag (Noss & Hoyles, 1996,

s. 106). De ser blant annet også hvordan elever som kan skrive dataprogrammer kan skissere halvforståtte matematiske ideer og få ut et semikonkret bilde av dette på skjermen for å få økt strukturell forståelse for matematikk og algebra (1996, s. 55). Dette samsvarer med kunnskapsstatusen i kapittel 2.3.2 om at elever trenger strukturell forståelse for matematiske objekter.

Bråting & Kilhamn beskriver sammenhengen mellom algebra og programmering, og introduserer ideen om programmering som et eget semiotisk register for å representere matematiske objekter og strukturer (2021, s. 171). Dette er interessant, og ifølge Duval kan dette åpne opp for en større forståelse av matematiske objekter, og kanskje dette kan falle innunder begrepet kvasialgebra som nevnt i underkapittel 2.3.4. Forfatterne sammenlikner blant annet algebraisk tenkning og algoritmisk tenkning der fellesnevneren er at begge krever et skifte, et skifte bort fra resultat og over til prosess og struktur – slik som etterspurt innledningsvis for å håndtere algebra (Bråting & Kilhamn, 2021, s. 181). Grunnen til dette er at når man programmerer er det datamaskinen som har ansvar for beregninger, mens vi som programmerere har ansvar for å strukturere matematiske uttrykk og algoritmer som dataen skal prosessere og beregne. I følge Duval fungerer tegn forskjellige i ulike registre, noe som kan være en utfordring for elever, spesielt med tanke på overføring av det matematiske objektet mellom ulike registre (2006, s. 110). Bråting & Kilhamn løfter frem en slik utfordring med likhetstegnet i programmeringssammenheng kontra i det symbolske matematiske registeret: I programmeringssammenheng betyr likhetstegnet å tilordne verdi til en variabel, mens i matematikken betyr det likhet (2021, s. 181). Dette kan være en stor utfordring siden forståelse for hvordan likhetstegnet signaliserer likhet på begge sider er så viktig for å forstå seg på matematiske strukturer som nevnt i underkapittel 2.3.2. Likhetstegnet fungerer ulikt i de forskjellige registrene, noe som kan skape misoppfatninger. På linje 2 i figur 2.6 ser man hvordan dette kan ta for seg, da dette ville vært katastrofalt i matematisk symbolspråk. En måte å håndtere det på kunne vært å bruke kommandoen «+=» på linje 2. Men bevisstheten for læreren er her viktig om de potensielle misoppfatningene.



Figur 2.6: Tilordning av variabler

De er også bekymret for at algebra vil bli for diffust, og at vi mister fokus for hva som egentlig er algebra, siden mye kan kalles algebra om vi bruker definisjonen til Radford (2021, s. 182). Læreren

må derfor være opplyst og konsekvent når programmering og algebra skal støtte oppunder hverandre slik at misoppfatninger ikke oppstår. Programmeringsspråket er ikke direkte overførbart til symbolsk algebra, men hvis man tar forhåndsregler, er tankemåten overførbar. De konkluderer også med en konsensus med den anerkjente matematikkdidaktikeren Mason om at programmering fostrer algebraisk tenkning (2021, s. 181). De oppsummerer også en mulig sammenheng mellom algebra og programmering slik som arbeid med likhet, arbeid med variabler, strukturer og mønster, samt generalisering og symbolisering (2019, s. 572).

Egara & Nzeadibe løfter frem hvordan utfordringer med læring av algebra kan begrunnes med hvordan det blir presentert i klasserommet, dets dårlige fundament, den abstrakte natur av matematiske konsepter, og lite hensiktsmessige læringsmetoder, og hvordan alle disse faktorene påvirker dårlige matematisk forståelse hos elever over hele verden (2018, ss. 2–3). Hvordan det blir presentert og læringsmetodene blir viktig. Dette tar inn elementer fra hvordan læring er situert om vi ser fra det sosiokulturelle perspektivet, som nevnt i underkapittel 2.2.3. For det er jo nettopp flere faktorer som påvirker elevenes læring enn selve elevens kognitive ferdigheter, og vi må derfor se på hvordan de algebraiske konseptene blir presentert. Dette ser vi i direkte sammenlikning med at algebraundervisning bør omhandle mer matematisk modellering, slik at kontekst og mening introduseres i algebraopplæringen, som nevnt i kapittel 2.3.3. Dette kan påvirke elevenes interesse og motivasjon, samt forståelse. Likevel ser vi som nevnt på starten av dette avsnittet, at algebraen presenteres på måter som ikke vekker elevenes engasjement eller interesse. Dette kan derimot overkommes ved å lage dataprogrammer som simulerer abstrakte modeller, som tallfølger eller sannsynlighetsfordelinger (Egara & Nzeadibe, 2018, s. 3). Disse programmene tillater elever til å direkte manipulere modellene for å se konsekvensen av dette, uten å være avhengig av abstrakt tenkning (Egara & Nzeadibe, 2018, s. 3). Elever ser arbeid med simulering som mer interessant og motiverende, samt mer knyttet opp mot virkeligheten (Alessi & Trollip, 2001, referert i Egara & Nzeadibe, 2018, s. 5). Simuleringer viser seg også å gjøre overføring av læring til den virkelige verden enklere, samt at elever for øvd på modellerings og problemløsingsferdigheter (Egara & Nzeadibe, 2018, s. 5). For det er ofte sånn at mange har angst og bekymringer rundt arbeid med matematikk, men vi ser hvordan arbeid med dataprogrammering gir økt motivasjon og mindre angst for å lære matematiske temaer, noe som igjen kan gi effekt på læringen (Egara et al, 2022). Studiet til Egara & Nzeadibe konstaterte dessuten at arbeid med simuleringer i matematikkfaget ga økt algebraferdigheter – et lignende studie som i denne masteravhandlingen (2018, ss. 11–12).

Healy et al. påpekte hvordan fremgangen av programmeringsspråk og andre nyere teknologier slik som regneark skapte nye ruter til læring av algebra, noe som utfordret validiteten til tidligere læreplaner, og de løfter frem læring og undervisning av algebra mediert eller formidlet av digitale omgivelser (2002, s. 231). Forskning fra 1981 viste at elever hadde betydelige problemer med å akseptere og bruke ideen om en bokstav som representerer en variabel i algebra og med å forstå det systematiske forholdet mellom variabler i algebraiske uttrykket – slik som også nevnt i kunnskapsstatusen om algebra i nåtiden (Healy et al., 2002, ss. 233–234). Dette begrunnes med det underliggende faktum at algebra er mer abstrakt, og at bakgrunnen for dette ligger i at det er et stort kognitivt gap mellom den operasjonelle aritmetikken – som man ofte lærer først – og algebra (Healy et al, 2002, s. 234). De sikter også til forskning om hvordan elever har utfordringer med å beskrive matematiske mønstre verbalt, og spesielt algebraisk, og at generaliseringen sjeldent er berettiget i annet en spesifikke instanser (2011, s. 235). Dette er naturlig ettersom det foretrekkes at man har konkrete referenter, og vi ser igjen at elevene sliter med å dekontekstualisere slik som nevnt i underkapittel 2.3.2. Healy et al. gjennomgikk en treårig lang studie på 80-tallet der de brukte det pedagogiske programmeringsspråket LOGO for å se om elever kunne lære matematikk gjennom aktiv konstruksjon av egen kunnskap, og hvordan dette kan tilrettelegges i datamiljø gjennom en itererende prosess med redigeringer og tilbakemeldinger fra datamaskinen (2002, s. 237). Hele poenget var om noen av utfordringene nevnt ovenfor kunne unngås, og det er interessant hvordan dette går i hånd i hånd med utfordringene beskrevet i denne avhandlingen – 2.1. Resultatene fra forskningen ga bevis for å støtte en økende overbevisning om at noen av vanskene som elever har kan overvinnes gjennom interaksjon med datamiljøer – for eksempel programmering (Healy et al., 2002, s. 237). Studiene fant ut at elever kunne akseptere at en bokstav kan stå for et generelt tall, og at de kunne akseptere og arbeide med åpne algebraiske uttrykk og kunne uttrykke generelle matematiske sammenhenger uttrykt med programmeringsspråk (Healy et al., 2002, s. 237). Dette løser de utfordringene som nevnt ovenfor og innledningsvis, spesielt med at elever må se på matematiske uttrykk som noe åpent, noe med struktur, og ikke noe som skal kalkuleres. Elever var også kapable til å bruke de algebralignende notasjonene i programmeringsspråket LOGO når de skulle skrive på papir og løse algebraoppgaver – det hadde overføringsverdi (Healy et al., 2002, s. 238). Dette er banebrytende forskning, forskning som kanskje har ligget i skyggen av at verden ikke var klar for digitalisering enda på den tiden.

De forklarer som nevnt hvordan arbeid med programmering eller regneark illustrerer hvordan slike aktiviteter har et skifte fra fokus på kalkulering og aritmetikk og over til det strukturelle. Begrunnelse for dette virker til å ligge i det at datamaskinen får ansvaret for kalkulasjonene, og

elevene blir mer i stand til og får mer tid og plass til å reflektere over datamaskinens tilbakemelding på skjermen som for eksempel kan være en lang rekke med tall i en tallfølge (Healy et al., 2002, s. 239). Eleven jobber mer med det strukturelle, og hva som kan endres der og effekten av dette. Det forklares at tilbakemeldingene som programmering eller arbeid i slike miljøer gir, tilbyr en ny ressurs for studenter til å forstå matematiske abstrakte ideer. Hele poenget ligger i at man kan redigere formelen til en tallfølge i for eksempel en løkke, og øyeblikkelig se effekten av dette som en tilbakemelding på skjermen, og man er ikke avhengig av å mentalt se for seg denne matematiske og abstrakte strukturen. Dermed vil datamaskinen konkretisere det abstrakte.

Healy et al. nevner også ideen om stillasbygging som vi kjenner fra det sosiokulturelle læringssynet om barnets proksimale utviklingssone hvor datamaskinen kan fungere som en mediator, at datamaskinens egenskaper kan gjøre mye av jobben som læreren gjør, at datamaskinen kan ha en stillasbyggende funksjon for eleven (2002, s. 240). De kan bygge forståelse som de ikke hadde klart på egenhånd, uten verktøyet. I forskningen deres observerte de dessuten hvordan elever kunne svitsje mellom det spesifikke og generelle hele tiden når de pratet med hverandre i det de lagde dataprogrammer (2011, s. 245). Dette er et essensielt poeng der det sosiokulturelle med dets aspekter slik som språk, fellesskap og samarbeid lar elevene bygge forståelse, og arbeid med dataprogrammering muliggjør denne algebraiske tenkemåten som elevene brukte da de pratet med hverandre. Man kan derfor tenke seg mulighetene som realistiske prosjekter kan by på, der for eksempel elevene samarbeider om å lage dataprogram, eller matematiske modeller om du vil. Lærerens jobb blir da å være en veileder, og elevene og datamaskinen kan støtte hverandre. Elevene kan for eksempel samarbeide med medelever i arbeid med dataprogrammer. De konkluderte med at datamiljøer lar elevene få et avbrekk fra aritmetikkfokusert algebraundervisning, noe som vi har sett kan gi mer forståelse (2011, s. 245).

Avslutningsvis i dette underkapittelet kan vi derfor med stor sikkerhet påstå at man kan lære algebra gjennom arbeid med programmering. Vi kan til og med si at det muligens kan gi økt innsikt i algebraiske fenomener i forhold til det normal undervisning gir. Det er her forskningsspørsmålet om at pedagogisk programmering kan gi økt kompetanse i utforming av algebraiske uttrykk kontra normal undervisning har sine røtter. Misfeldt & Duun nevnte hvordan det støtter oppunder elevaktivitet i den forstand at man produserer egen kunnskap gjennom å lage dataprogrammer. De løfter frem hvordan denne aktiviteten fostrer abstraksjon, noe som er et av grunnelementene i algebra. Det gir også kompetanse i algoritmisk tenkning som er en essensiell problemløsningsmetode som nevnt i underkapittel 2.4.3. Det gir også elevene mer konkrete

referenter med tanke på variabler, slik som etterspurt i underkapittel 2.3.4. Vi kan derfor utvide elevenes nett av sammenhenger til å skape en ny relasjonell forståelse. Bråting & Kilhamn nevner hvordan det legges fokus på det strukturelle kontra det operasjonelle som etterspurt innledningsvis i denne avhandlingen. De nevner også direkte sammenhenger mellom programmering og algebra slik som variabler, generalisering og symbolisering. Egara et al. nevnte hvordan dataprogrammering gir kontekst og mening i undervisningen, noe som kan styrke elevene interesse og motivasjon, og hvordan de så at arbeid med dataprogrammer i algebraundervisningen ga økte algebraferdigheter. Forskningen til Healy et al. viste at utfordringer som elever har med algebra kan overvinnes ved arbeid med dataprogrammering, og at arbeid med dataprogrammer fostret den matematiske samtalen mellom elever. Derfor oppsummeres dette underkapittelet med tanken om at datamaskinen kan skape mer hensikt i algebraundervisningen og bygge en bro mellom det spesifikke og generelle, og hvordan man kan konkretisere det abstrakte. Tanken er derfor at elever kan lage dataprogram i undervisningen for å bli bedre på utforming av algebraiske uttrykk.

3 Metode

Forskningsspørsmålet*: Vil pedagogisk programmering som en ressurs i algebraopplæringen, gi signifikant forbedring i algebraferdigheter – nærmere bestemt utforming av algebraiske uttrykk – hos norske ungdomsskoleelever i forhold til normal algebraopplæring?*

3.1 Introduksjon til metode

Hansen & Simonsen påpeker at vitenskap handler om å skaffe til veie kunnskap om verden (Kvarv, 2021, s. 14). Metode innenfor vitenskapelig forskning handler om hvordan man går frem for å skaffe til veie denne kunnskapen (Lund & Haugen, referert i Høgheim, 2020, s. 27). For det er nettopp det jeg ønsker å få til med denne avhandlingen, skaffe til veie kunnskap om verden på en eller annen måte; i mitt tilfelle ulike undervisningsoppleggs effekt på læring av algebra. Et sentralt mål med all forskning er validitet, og validitet kan defineres som grad av sannhet i en slutning (Høgheim, 2020, s. 80). Vitenskapelig forskning skal være kritisk, som betyr at man bør kjenne til mulige styrker og svakheter ved det arbeidet man gjør, og det er her vurderingen av validitet kommer inn (Høgheim, 2020, s. 25). Derfor kommer jeg til å løfte frem styrker og svakheter ved arbeidet jeg gjør – metoden spesielt. Forskning skal skille seg fra hverdagserfaringer fordi de ideelt sett bygger på systematiske iakttakelser, logiske slutninger og eksplisitter forutsetninger (Kvarv, 2021, s. 35). Derfor vil fremgangsmåten også tydelig begrunnes og løftes frem. Intersubjektiv åpenhet, som vil si offentlig åpenhet i den forstand at utenforstående skal ha mulighet til å kikke forskeren, meg, i kortene og bli presentert for de verdimessige, teoretiske og metodiske forutsetningene som ligger til grunn for en undersøkelser, er et av de viktigste kravene som stilles til forskning (Kvarv, 2021, s. 38). På den måten vil forskningen være etterprøvbar, og det kan være lettere for fellesskapet å ha en konsensus rundt validiteten i slutningene som jeg kommer frem til (Kvarv, 2021, s. 35). Derfor vil mine antakelser og valg begrunnes og løftes frem.

Jeg har med meg en forutforståelse om hvorvidt programmering fungerer bedre enn normal algebraopplæring når det gjelder effekten på læring av algebra, og den tilsier at jeg er partisk. Jeg er altså ikke helt objektiv når slutningen skal tas. Min egenerfaring gjør at jeg lener meg mot å tro at programmering fungerer bedre, ettersom at det hjalp meg til å forstå algebra i min egen utdanning. Jeg er også overbevisst om at det er en legitim metode å drive algebraopplæring på gjennom å ha lest forskningen som ble presentert i underkapittel 2.5.1. Innenfor naturvitenskapelig forskning ønsker man gjerne å redusere til et minimum den påvirkningen forskerens verdier og forutforståelse har på slutningene som en tar (Bryman, 2012, s. 177). Derfor har jeg valgt å gå veien om

naturvitenskapelig forskning, der man ofte snakker om hypotetisk-deduktiv metode, der hypoteser basert på forskningsspørsmålet skal testes ut (Kvarv, 2021, s. 27). Dette faller naturlig sammen med det at jeg har et forklarende forskningsspørsmål, som har som formål å skape kunnskap om antatte forhold mellom fenomener og deres egenskaper, nettopp siden det er et utgangspunkt for å anta at noe skal være på en gitt måte, og målet med forskningen er derfor å bekrefte eller avkrefte antakelsen (Høgheim, 2020, s. 42). Stiller man forklarende forskningsspørsmål, vil man antakelig ha best utbytte av å velge kvantitativ metode, der vi gjerne undersøker antatte sammenhenger, påvirkninger eller effekter, og måler dette med tall (Høgheim, 2020, s. 59). Min forutforståelse i kombinasjon med det forklarende forskningsspørsmålet setter en del føringer for hvordan resten av forskningsprosjektet utformer seg, og dette skal jeg videre forklare nå. For målet er jo å være objektiv i slutningene som tas.

3.2 Epistemologisk standpunkt

Kardinalpunktet og det som ideelt sett er hovedintensjonen i all forskning, er søking etter sannhet (Kvarv, 2021, s. 13). Ikke alt kan kvalifiseres som sannhet derimot, og det er tvetydig enighet rundt hva som er veien til sannhet, og en viktig del av vitenskapelig virksomhet handler derfor om å skille mellom hva vi kan vite noe om, og hva som er spekulativt begrunnet (Kvarv, 2021, s. 18). Erkjennelsesteori, epistemologi, handler om den tenkningen og sansingen som gir oss kunnskap om verden, for dette forteller oss noe om mulighetene og grensene for den menneskelige erkjennelse, hva vi kan vite noe om, som igjen forteller oss om forskningens validitet (Kvarv, 2021, s. 16). Derfor ønsker jeg å ha et reflektert forhold til erkjennelseskildene som jeg tar i bruk, for det er disse jeg bruker for å vurdere «sannheten» i hypotesene.

Empirisme er det standpunktet i kunnskapsteorien der kunnskap baseres på erfaring – sansefornemmelser (Kvarv, 2021, s. 18). Der hevder man at det som observeres kan få status som kunnskap, i motsetning til hos rasjonalistene der man tenker at logisk tenkning strekker seg over sanseinntrykkene (Høgheim, 2020, s. 21). Ser man veldig svart-hvitt på dette, vil de si at empirisme lener seg mot objektiv kunnskap, mens rasjonalisme lener seg mer mot subjektiv tolkning som åpner opp for menneskets fortolkning av det som sanses (Høgheim, 2020, s. 21). Det sammenfaller godt med mitt ønske om å være objektiv, nettopp fordi jeg skal observere og tallfeste noe, og mine egne tolkninger påvirker derfor slutningene i mindre grad. Innenfor empirisme oppnås objektivitet gjennom å gi et så korrekt bilde av naturen og dens lover som mulig – gjenspeile virkeligheten på en så klar, dekkende og presis måte som mulig – korrespondansteorien (Kvarv, 2021, s. 21). Slutningsprinsippet som dette fører med deg kalles induksjon, som vil si at vi trekker generelle

konklusjoner på grunnlag av enkeltstående observasjoner, og det skal jeg gjøre i denne avhandlingen (Kvarv, 2021, s. 25). Det er nettopp slik man må gjøre det fordi man ikke har tilgang til å observere alle mulige tilfeller av et fenomen. Problemet med dette er at vi aldri kan være sikker på kunnskapen vi finner, noe man må ha et reflektert forhold til. Dette er en stor svakhet med slikt arbeid. Det er også ikke enkelt å gi et korrekt bilde av virkeligheten i et klasserom, og det er ikke nødvendigvis slik at det man observerer at fungerte i ett klasserom vil fungere i et annet klasserom; for læring skjer ikke i et vakuum (Hinna et al., 2011, ss. 898–900). Det kan også være krevende – på grunnlag av sosiale fakta eller ting – å etablere lover av samme type som vi finner i naturen, siden mennesker må bygge på søken etter mening og forståelse av subjekt-subjekt-relasjon (Kvarv, 2021, s. 74). Derfor må man være kritisk og reflektert til observasjonene, samtidig som sosiale faktorer og annet må kartlegges. Vi kan altså ikke bare gå inn i et klasserom, se at noe funker, og konkludere med det.

Dette synet på kunnskapsanskaffelse faller innenfor positivisme, som var en frigjøringsanskuelse på 1800-tallet (Kvarv, 2021, s. 66). Tanken var at det skulle frigjøre mennesket fra den åndelige dominans som kunne tilskrives metafysisk spekulasjon ved å basere seg på verdinøytrale observasjoner (Kvarv, 2021, s. 66). Ulempen med dette er at den nøytrale vitenskap som ikke lar seg påvirke av interesser i samfunnet er en illusjon, og derfor ble kritisk teori, og senere postpositivisme, løftet frem som hadde intensjon om å belyse interessene hos det erkjennende subjekt (Kvarv, 2021, s. 77). Derfor må man løfte frem og være reflektert rundt valgene man tar. Det er viktig å være reflektert rundt at det man observerer ikke nødvendigvis er objektivt tilegnet kunnskap, nettopp fordi det sies at alt arbeidet man gjør før selve slutningen, er kvalitativt (Høgheim, 2020, s. 97). Planleggingen, valg av indikatorer for å fange begreper, konseptualisering av begreper og alt dette er arbeid som ikke nødvendigvis ses på som objektivt eller kvantitativt arbeid, er med på å påvirke slutningene. Derfor håper jeg at andre ser at jeg spiller med åpne kort i alle valgene jeg tar, og at andre kan etterprøve forskningen min.

3.3 Forskningsmetode

3.3.1 Kvantitativ metode

Valg og bruk av kvantitativ metode assosieres gjerne med positivisme (Kvarv, 2021, s. 131). Som nevnt så sammenfaller dette også godt med det forklarende forskningsspørsmålet og målet om objektivitet. Kvantitativ metode blir definert som å forklare et fenomen gjennom å samle inn numeriske data og analysere disse ved bruk av matematisk metode (Aliaga & Gundersen, 2002,

referert i Høgheim, 2020, s. 97). Man kvantifiserer eller tallfester det man forsker på, og analyser tallene; i mitt tilfelle algebraferdigheter (Høgheim, 2020, s. 29). Man jobber bekreftende, og forskningsspørsmålet driver forskningen, og ut ifra forskningsspørsmålet lages det hypoteser, testbare påstander som skal bekreftes eller avkreftes basert på tallmaterialet (Høgheim, 2020, s. 99).

Måten man arbeider på er slik som denne masteravhandlingen er delt opp. Først arbeider man i det konseptuelle domenet, der man går igjennom teori, konseptualiserer begreper, genererer hypoteser og identifiserer variabler (Field, 2018, s. 38). Variablene er det som skal måles og undersøkes. Dette arbeidet er deduktivt da vi beveger oss fra det generelle til det spesifikke (Bryman, 2012, s. 160). For å teste ut hypotesene, må vi bevege oss over til det observerbare domenet, og det er der vi er nå (Field, 2018, s. 40). Hypotesen, så vel som variabler, må gjøres målbart, som vil si at vi operasjonaliserer (Field, 2018, s. 40). Dette er en del av det kvalitative arbeidet som nevnt i kapittel 3.2, der vi finner ut hvilke indikatorer vi ønsker å bruke for å fange begreper og fenomener, for det er egentlig bare selve analysen av tallene som ansees som objektiv. Nesten hele forskningsprosjektets validitet baserer seg på at jeg faktisk måler det jeg er ute etter (Field, 2018, s. 53). Derfor må det tenkes nøye gjennom hvilke indikatorer jeg skal bruke for å måle blant annet algebraferdigheter. Innenfor kvantitativ forskning er vi som regel ute etter å trekke kausale slutninger, som vil si årsak-effekt-forhold mellom variabler (Høgheim, 2020, s. 111). Dette er en deterministisk filosofi, der noe bestemmer utkommet – ofte assosiert med postpositivisme (Creswell & Creswell, 2018, s. 44). Variabelen som vi tror er en årsak er kjent som en uavhengig variabel, og variabelen som vi tror er en effekt er kjent som avhengig variabel fordi verdien av denne avhenger av årsaken (Field, 2018, s. 47). I dette tilfellet vil den dikotome, eller binære, variabelen programmering være den uavhengige variabelen, og vi ønsker å se om den påvirker endring i den avhengige variabelen algebraferdigheter, som er en diskrete intervallvariabel.

3.3.2 Kvasieksperimentell design

Man kan finne et kausalt forhold mellom variabler på to måter: 1) å observere naturlige korrelasjoner som forekommer i verden 2) intervenere med virkeligheten ved å manipulere den uavhengige variabelen og se effekten av dette (Field, 2018, s. 55). Ulempen ved den første er at korrelasjoner man observerer ikke forteller oss noe om kausalt forhold mellom variabler, nettopp fordi man ikke kan vite med sikkerhet hva som er den uavhengige variabelen, for det kan eksistere forvirrende variabler, som vil si andre faktorer enn den uavhengige variabelen som påvirker den avhengige variabelen (Field, 2018, ss. 55-56). Vi kan aldri være sikre på hva som er årsaken til effekten i den avhengige variabalen. Man kunne for eksempel undersøkt matematikkarakteren til

elever som har programmering som valgfag, kontra elever som ikke har det; men man har ingen oversikt over forvirrende variabler som for eksempel matematikkferdigheter i utgangspunktet, interesser eller tid brukt på matematikk. Derfor bruker man gjerne den andre metoden, som handler om at hvis årsaken er til stede, bør effekten være til stedet, og vice versa (Field, 2018, s. 56). Denne måten å utlede kausalitet på er å sammenligne to kontrollerte situasjoner der årsaken er til stede i den ene, og der årsaken er fraværende i den andre (Field, 2018, s. 57). Finnes det en forskjell i effekten i de to, kan vi konkludere med at det er på grunn av den uavhengige variabelen – programmering i mitt tilfelle. Dette kalles for eksperimentelt design, og er godt egnet for å blant annet undersøke hvordan ulike undervisningsopplegg påvirker læring (Høgheim, 2020, s. 112). I mitt tilfelle vil forskerens, min, intervensjon med virkeligheten være to ulike undervisningsopplegg. Variasjonen man finner kalles for systematisk variasjon, og det er den vi ønsker å oppdage (Field, 2018, s. 59). Usystematisk variasjon er endring i den avhengige variabelen grunnet ukjente faktorer, og det ønsker vi å redusere til et minimum (Field, 2018, s. 58). Dermed kan én gruppe drive med normal algebraundervisning, kontrollgruppen, og den andre gruppen drive med pedagogisk programmering, eksperimentell gruppe. Finner vi en forskjell i gruppene etter eksperimentet, kan man være rimelig sikker på at undervisningsoppleggenes forskjeller påvirker algebraferdighetene.

Jeg har valgt å ta i bruk det som kalles for «independent design» som vil si at jeg vil sammenlikne to ulike elevgrupper (Field, 2018, s. 60). Ulempen med dette er at systematisk variasjon kan oppstå på grunn av forskjeller i gruppene; for eksempel kan den ene gruppen i utgangspunktet ha høyere algebraferdigheter, og da vil måling av dette være meningsløst. Måten man vanligvis håndterer dette på er ved å lage to tilfeldig tildelte grupper, slik at jeg som forsker ikke har noe påvirkning på gruppeinndelingen, som er et viktig element i hvorvidt vi finner forskjeller i gruppene (Creswell & Creswell, 2018, s. 224). Dette kalles for et ekte eksperiment, og er på mange måter gullstandarden for kausale undersøkelser, for da blir gruppene like basert på sannsynlighet, og vi kan bruke sannsynlighetsregning for å konkludere signifikans i funnene våre (Høgheim, 2020, s. 113). Sannsynlighetsregning brukes for å overveie sannsynligheten for ulike grupper. Ved å gjøre dette sikrer man indre validitet, som vil si at man utelukker alternative forklaringer, og øker sikkerheten i de kausale slutningene (Engel & Schutt, 2017, s. 336). Det sikrer derimot ikke ytre validitet, generaliserbarhet, altså at det man finner kan gjelde i flere tilfeller, nettopp fordi det kan være komplett tilfeldig at vi finner forskjellens som vi ønsket å finne (Engel & Schutt, 2017, s. 336). Derfor har jeg valgt kvasieksperimentell design, som vil si at gruppeinndelingen ikke er 100% tilfeldig inndelt, og jeg som forsker har en delvis kontroll over gruppeinndelingen (Creswell & Creswell, 2018, s. 225). Gruppene vil bli delt basert på en pretest der algebraferdigheter kartlegges

eller måles, ofte kalt «matching», og på den måten kan uønsket systematisk variasjon holdes til et minimum, og ytre validitet styrkes (Engel & Schutt, 2017, s. 337). Elevene vil bli delt inn i par basert på algebraferdighetene på pretesten, og ved bruk av en RNG (random number generator), delt inn i hver sin gruppe. Ulempen med dette, og grunnen til at dette ikke kalles et ekte eksperiment, er at vi simpeltent ikke kan kartlegge alt ved deltakerne, slik som motivasjon, hvor fort man lærer, dagsform osv., og gruppene vil derfor aldri bli helt like – som man delvis påstår (Engel & Schutt, 2017, s. 337). De vil derimot bli tilnærmet like. Derfor må funnene betraktes med varsomhet. Etter gruppeinndelingen vil gruppene utsettes for to ulike undervisningsopplegg, der tilfeldig bakgrunnsstøy vil bli kartlagt, slik at potensielle medierende faktorer ikke påvirker resultatet – usystematisk variasjon (Field, 2018, ss. 59–60). Avslutningsvis gjennomføres det en posttest for å måle effekten i gruppene, og vi kan konkludere med at variasjonen man finner på posttesten skyldes undervisningsoppleggenes forskjeller – programmering eller ikke.

3.3.3 Hypotesetesting

For å vite om differansen i algebraferdigheter fra posttesten mellom gruppene er signifikant nok til å konkludere med noe, entrer vi domenet av slutningsstatistikk, der vi bruker matematiske metoder for å si om funnene våre er verdt å tolke (Høgheim, 2020, s. 178). Det er som nevnt i kapittel 3.2 ikke mulig å observere alle tilfeller i en populasjon, og vi må derfor ta en stikkprøve av populasjonen, i mitt tilfelle 28 ungdomsskoleelever, og bruke denne dataen til å si noe om hele populasjonen – induksjon. Vi kan derfor ikke si noe med sikkerhet, slik som nevnt om positivismen, men vi kan estimere en sannsynlighet for sannheten i hypotesen vår (Field, 2018, s. 126). Å undersøke holdbarheten, sannhetsinnholdet, i hypoteser, kalles for hypotesetesting (Lysø, 2015, s. 464). Den grunnleggende metoden er at man vanligvis har to konkurrerende hypoteser, en som sier at effekten eksisterer, alternativ hypotese, og den andre som sier at effekten ikke eksisterer, nullhypotesen (Field, 2018, s. 145). Det er den alternative hypotesen man ønsker å teste ut, og se om kan erstatte nullhypotesen. I mitt tilfelle vil påstanden om at programmering gir økt effekt i utforming av algebraiske uttrykk være den alternative hypotesen.

I en ideell verden vil data være distribuert symmetrisk rundt gjennomsnittet, og mange naturlige forekomster har ofte denne fordelingen, slik som for eksempel matematikkferdigheter (Field, 2018, s. 62). Denne fordelingen kalles for en normalfordeling og karakteriseres av to størrelser, den ene er μ, gjennomsnittet i populasjonen, og den andre er σ, standardavviket i populasjonen, eller variansen, σ^2, om du vil (Lysø, 2015, s. 415). Arealet under kurven forteller noe om

sannsynligheten for å få en viss verdi hvis vi tar en stikkprøve fra populasjonen (Lysø, 2015, s. 415). Standardavviket er nøkkelen her, for den beskriver variasjonen i fordeling, altså hvor langt vekk fra gjennomsnittet man kan forvente å trekke en tilfeldig stikkprøve – sannsynligheten for dette. Hvis stikkprøven er signifikant avvikende fra det som forventes å finne, μ, vil man med sannsynlighet kunne si at nullhypotesen ikke lenger er representativ. Man ønsker gjerne at det skal være p-verdi, altså sannsynlighet, på $p \leq 0.05$ for å finne det man finner for å konkludere med at funnene våre er signifikant differensiert fra populasjonen (Field, 2018, ss. 126–127). Det vil si at det er mindre enn fem prosent sannsynlighet for at vi feilaktig forkaster nullhypotesen, noe som vil si at vi bekrefter den nye alternative hypotesen som representativ for populasjonen. Dette forteller oss for ofte vi er forberedt på å ta feil, og i dette tilfellet, 1 av 25 ganger (Field, 2018, s. 126). Dette betyr at man aldri kan være helt sikker på det man finner, som jo igjen er en stor svakhet ved disse metodene (Field, 2018, s. 131).

I mitt tilfelle har vi ikke tilgang på standardavviket til populasjonen, som jo egentlig alt baserer seg på, og derfor bruker vi ikke normalfordeling, men heller t-fordelingen, en sannsynlighetsfordeling som aksepterer mer variasjon i standardavviket ved lavere antall observasjoner, nettopp fordi vi må estimere standardavviket (Lysø, 2015, s. 446). Det betyr at vi aksepterer mer variasjon i funnene våre, noe som er naturlig siden at et lavt antall utfall kan variere mye; med tanke på store talls lov. Dette gjør vi ved å regne ut antall frihetsgrader, ofte betegnet som df (Field, 2018, ss. 106–107). I mitt tilfelle skal vi ikke sammenlikne noe opp mot en populasjon for å se om det «nye» vi finner vil forkaste en nullhypotese, men vi vil sammenlikne to stikkprøvers gjennomsnitt. Gjennomsnittet av posttesten til den eksperimentelle gruppen opp mot kontrollgruppen, og vi ønsker å se om differansen er signifikant til å konkludere med noe (Field, 2018, s. 590). Metoden vi da bruker kalles for «independent t-test», og vi ser om differansen mellom gjennomsnittene i gruppene er avvikende det vi forventer å finne (Field, 2018, ss. 589–590). Siden alt baserer seg på standardavvik, burde man i teorien ha identisk varians i begge gruppene, altså standardavviket kvadrert, slik at vi har en konsensus på standardavviket i populasjonen (Field, 2018, s. 358). I programmet som jeg skal bruke for å gjøre utregningene, SPSS, gjennomføres det «Levene's test», som sjekker om variansene er omtrent like, og hvis de ikke er signifikant avvikende med en p på 0.05 eller under, så kan vi tolke dataene på vanlig måte (Field, 2018, s. 358).

Jeg skal gjennomføre en ensidig t-test, nettopp fordi jeg kun er interessert i den positive differansen ved kontrollgruppens gjennomsnitt subtrahert fra den eksperimentelle gruppens gjennomsnitt, ikke ulikhet – begge sider. Og hvis differansen er langt nok over det vi forventer å finne, så kan vi

forkaste nullhypotesen, og konkludere med at den alternative hypotesen kan representere populasjonen. Det betyr at de skal være mindre enn eller 5% sannsynlig at vi finner denne differansen: $p \leq 0.05$. Nå skal det nevnes at valg av signifikansnivå egentlig skal være et meningsfylt og gjennomtenkt valgt, men i teorien er det ikke det, fordi nesten alle bruker 0.05 uten å tenke noe mer over det (Field, 2018, s. 127). Dette holder sjansen for type 1-feil lav, som vil si at vi feilaktig konkluderer med at det er et årsak-effekt-forhold mellom variablene (Field, 2018, s. 136). Det motsatte av dette er type 2-feil, som vil si at vi feilaktig beholder nullhypotesen, altså at vi ikke tror det er noe årsak-effekt-forhold her (Field, 2018, s. 136). Siden hypotesetesting egentlig baserer seg på tilfeldig grupper eller stikkprøver, har jeg valgt å øke signifikansnivået til $p = 0.2$ – 20%. Dette er på grunnlag av at gruppene er tilnærmet like basert på algebraferdigheter i utgangspunktet, og at det derfor ikke behøves så stor margin for å overveie potensielle forskjeller med sannsynlighetsregning – derfor lavere sjanse for type 2-feil. Cohen argumenterer for at 20% er det maksimalt aksepterte signifikansnivået (1992, referert i, Field, 2018, s. 136). Det påpekes at valg av signifikansnivå bør være et «educated guess» basert på helheten av prosjektet (Field, 2018, s. 136). Derfor er signifikansnivået satt til 0,2 i denne masteravhandlingen.

Hypotesene som skal testes ut i denne masteravhandlingen blir som i tabell 3.1. Nullhypotesen i mitt eksempel er av et noe alternativt preg, for den er på en måte i enighet med kunnskapsstatusen, men kanskje ikke helt standardisert. Ikke alle vil kunne være enige i at man kan drive med programmering på lik linje som normal algebraundervisning og få samme effekt på læring. Jeg er derimot på jakt etter å finne en løsning på problemet som norsk skole står overfor, nemlig at vi sliter med algebra, som nevnt innledningsvis. Derfor må det overbevisende argumenter til for å kunne si at programmering skal innlemmes som standard i algebraopplæringen, noe som det nå ikke er. Det er egentlig mest opp til lærernes tolkning av læreplan om hvorvidt programmering kan benyttes i algebraopplæringen. Derfor velger jeg den radikale varianten, med å undersøke om det faktisk gir økt effekt på læring kontra vanlig undervisning, da dette vil være mer overskriftsskapende. Dette vil gi mer overbevisende argumenter for å innlemme programmering som standard i algebraopplæringen. Gjennomsnittene i testene er kontinuerlige variabler, noe som muliggjør bruken av t-test, da dette er et krav for å drive med normalfordeling.

Hypoteser:	H_0: Programmering som et alternativ i algebraopplæringen vil gi lik kompetanse i utforming av algebraiske uttrykk hos ungdomsskoleelever som ved normal algebraopplæring.

	H_a: Programmering som et alternativ i algebraopplæringen vil gi økt kompetanse i utforming av algebraiske uttrykk hos ungdomsskoleelever i forhold til normal algebraopplæring.
Skal testes:	H_0: $\mu_0 = \mu_1$ H_a: $\mu_0 < \mu_1$ Signifikansnivå: 0,2

Tabell 3.1: Hypoteser som skal testes ut

3.4 Gjennomføring

3.4.1 Forskningsetikk

Det er viktig å ha et bevisst forhold til forskningsetiske retningslinjer, for forskningsetikkloven sier at all forskning skal skje i samsvar med etiske prinsipper (Forskningsetikkloven, 2017, § 1). Forskningsetikk retter blant annet fokus mot dem som man forsker på, mot deltakerne i ulike undersøkelser (Høgheim, 2020, s. 86). Friheten, interessene og integriteten til de som deltar i forskningen skal ivaretas (Høgheim, 2020, s. 88). Det skal være frivillig å delta på forskningen, og de som deltar skal gi sitt samtykke til å delta (NESH, 2021). Deltakerne skal også gis tilstrekkelig informasjon om forskninger de inviteres til å delta i, slik at de kan ta en overveid avgjørelse om å delta (NESH, 2021). Kravet om personvern er viktig, og skal man behandle personopplysninger digitalt, må man sende meldeskjema til NSD, men tar man forhåndsregler trenger man ikke dette (NSD, uå.). Nå skal jeg ikke jeg behandle personopplysninger på noen måte, så personopplysningsloven gjelder ikke, men jeg valgte likevel basert på NESH sine retningslinjer å innhente samtykke fra alle deltakerne (NESH, 2021). Siden deltakerne er barn, må jeg innhente samtykke fra foresatte også (NESH, 2021). I vedlegg 8 kan du se samtykkeerklæringen som ble delt ut til elevene som skulle delta på forskningsprosjektet. Det var viktig for meg å ikke informere for mye om forskningsprosjektet, siden dette kan påvirke forskningsfunnene (Høgheim, 2020, s. 91). Intervensjonen ville da muligens få en forsterket effekt da det kunne ført til en wow-effekt for den eksperimentelle gruppen, likeledes ville kontrollgruppen derfor blitt sett på som mindre spennende og derfor påvirket resultatene (Creswell & Creswell, 2018, s. 243). Deltakerne i kontrollgruppen ville antagelig også opplevd det som kjedelig eller urettferdig, og igjen ikke prestert like bra (Creswell & Creswell, 2018, s. 243). Derfor ble de kun informert om at det var to ulike undervisningsopplegg, og elevene visste ikke hvilke som var kjernen i eksperimentet, eller om det i det hele tatt var to grupper som skulle sammenliknes.

Alle de 28 ungdomsskoleelevene ordnet samtykke. Nå sies det også at man ikke skal motivere deltakere med belønninger, for det kan være etisk uforsvarlig (Høgheim, 2020, s. 88). Det var derimot en bekymring for meg å motivere elevene til å være med på dette prosjektet, og de fikk derfor en kinotur med hele klassen som en belønning. Dette fikk elevene til å ta det seriøst, og de ønsket å gi av seg selv, og bidra til forskningen. Det kan være krevende for elever å ta undervisningen seriøst, når de må bruke ekstra mye tid på matematikk, et fag som i utgangspunktet kan ansees som utfordrende. Dette kunne påvirket resultatet, og lignet lite på virkeligheten hvis dette var noe de gjennomførte med lite motivasjon, og ikke tok seriøst, eller ble sett på som et tillegg til vanlig undervisning. De ble i tillegg informert om at dette var relevant for dem met tanke på neste terminprøve, og derfor en del av deres utdanningsløp. Dette var også med på å skape en mer realistisk læringssituasjon.

3.4.2 Operasjonalisering og indikatorer i pre og posttest

Jeg skal nå forklare hvordan jeg skal måle algebraferdighetene. Dette er nødvendig slik at jeg viser at min forutforståelse ikke påvirker det kvalitative arbeidet med å skape indikatorer, slik som løftet frem i kapittel 3.2. Derfor blir forskningsfellesskapet tatt med på begrunnelsen og operasjonalisering, og det er mulig å se forskeren, meg, i kortene. Jeg forklarte i underkapittel 2.3 hva som ansees som algebraferdigheter, og hvordan jeg konseptualiserer begrepet. Dette skal nå gjøres målbart, og heldigvis er ikke dette en altfor krevende jobb. For her kan det benyttes direkte indikatorer, da dette er noe som lar seg enkelt måle, i motsetning til begreper som for eksempel motivasjon, der man muligens må ta i bruk indirekte indikatorer (Bryman, 2012, s. 164). Da slipper man å gå veien om variabler som fungerer som indirekte indikatorer, og man kan konsentrere seg om variabelen algebraferdigheter. Det betyr at algebraferdigheter kan direkte måles, og hva dette innebærer fant vi ut gjennom konseptualiseringen. Når det er snakk om måleinstrument og indikatorer, kan man ikke unngå begrepet reliabilitet. Reliabilitet refererer til påliteligheten av et mål på et konsept; altså nøyaktigheten til et måleinstrument (Bryman, 2012, s. 169). Altså om måleinstrumentene vil være like nøyaktige hos alle deltakerne. Dette er ikke noe som vil være så relevant i denne avhandlingen da dette er noe som vektlegges mer i spørreundersøkelser og andre typer eksperimenter som bruker indirekte indikatorer. Likevel ble reliabilitet i form av stabilitet i måleinstrumentene vektlagt når det gjaldt valg av forskningsdesign, for hvis jeg heller hadde gått for et såkalt «repeated measure design» der man heller måler den samme gruppen deltakere både før og etter manipulasjonen av den uavhengige variabelen med den samme testen før og etter, noe som ville holdt usystematisk variasjon til et minimum, ville modningsprosessen til elevene i stor grad

påvirket resultatet i testen etter eksperimentet (Bryman, 2012, ss. 169–170; Field, 2018, s. 59). Dette ville muligens ført til lav reliabilitet, og man kunne ikke konkludert med at forbedring på ettertesten skyldes manipulasjonen av den uavhengige variabelen. Dette er også en av hovedgrunnene til at jeg har valgt et «independent design», og forskjellige tester før og etter eksperimentet.

Siden hele planen er å lage to like grupper basert på pretesten, slik at en potensiell forskjell i gruppene etter intervensjonen vil basere seg på intervensjonen, altså kun manipulasjonen, må disse gruppene ha like algebraferdigheter i utgangspunktet. Derfor skal algebraferdigheter måles både før og etter eksperimentet. Det er ytterst viktig for validiteten til forskningsprosjektet at selve måleinstrumentene, testene, måler det jeg er ute etter, altså algebraferdigheter (Bryman, 2012, s. 170). Dette vil avgjøre verdien til variabelen algebraferdigheter hos de ulike deltakerne, som brukes både til å måle algebraferdigheter i pre- og posttest. Dette er som nevnt i kapittel 3.3.1 en diskrete intervallvariabel. Dette er fordi testene, som jo er måleinstrumentene, vil ha en poengskala fra 0 til 18. Det er min erfaring, og en standard for retting av mange matematikkprøver på ungdomsskoler generelt, at man vurderer oppgaver som feil, delvis riktig, eller riktig. Derfor vil oppgavene vurderes som 0 poeng for feil, 1 poeng for delvis riktig og 2 poeng for helt riktig.

I vedlegg 9 og 10 ser man pre- og post-testen, og det kan være hensiktsmessig å ha disse oppe når jeg nå går gjennom begrunnelsen for oppgavene. Det er her algebraferdigheter med stor vekt på utforming av algebraiske uttrykk skal måles. På både pre- og posttesten var oppgave 1 en oppgave som skiller seg fra de andre. Der skulle deltakerne bruke språket, noe som viser god forståelse for hva de driver med om vi ser på det sosiokulturelle som nevnt i kapittel 2.2.3 – når de kan sette ord på noe, så forstår de det (Hinna et al., 2011, ss. 899–909). Det vil også vise god forståelse for hvorfor man arbeider med noe, relasjonell forståelse. De ser det de driver med i relasjon til noe annet, og de ser elementene i algebra i og matematikk i relasjon til hverandre. Dette er et interessant spørsmål, for vi kan da etter eksperimentet se om programmering ga dem større forståelse for bruken av algebra, kontra den vanlige undervisningen – læring med hensikt. Da får vi også sett om deltakerne nevner noe av det som er nevnt i kapittel 2.3.2 om generalisering, og om de nevner horisontal matematisering som nevnt i kapittel 2.3.3. For det er jo nettopp meningsløst å drive med algebra hvis man ikke vet om disse fenomenene, ikke begrepene vel å merke, men det som ligger bak. Uten dette blir det bare meningsløs manipulasjon av symboler, som nevnt i innledningen. Som Arcavi et al. nevner i underkapittel 2.3.2, så er en måte å sørge for at elevene oppfatter hensikten med algebra på å få dem til å jobbe med oppgaver som har meningsfullt resultat for elevene selv, og

derfor kan vi se om for eksempel programmering kan styrke elevenes oppfatning av hensikt. En deltaker som skriver at det lar oss tenke generelt om matematiske strukturer, arbeide med ukjente tall som om vi kjente dem, bruke matematikk til å beskrive og løse praktiske situasjoner, vil nok få to poeng. Hvis en deltaker kun skriver at det lar oss tenke generelt, vil de få ett poeng. Her er det noe rom for tolkning. Derfor lot jeg dessuten en medstudent rette posttestene, slik at jeg ikke ville være partisk i tolkningen av oppgavebesvarelse, og dessuten visste ikke medstudenten hvilke posttester som tilhørte hvilke grupper. Dette er et viktig grep for å sikre validitet og generaliserbarhet.

Oppgave 2 har tre deler. Denne oppgaven er viktig da den får frem horisontal matematisering, slik som nevnt i underkapittel 2.3.3, om at elevene håndterer overgangen fra det kontekstuelle registeret, og over til det symbolske. Her må elevene bruke algebraen som et verktøy i aktiviteter utenfor selve matematikken, som nevnt i kapittel 2.3.3. Det er også den mest kognitivt krevende delen i modelleringsprosessen. Skal elevene jobbe med modellering, som er nevnt flere steder i læreplanen, må de øve på dette. Dette handler i stor grad om utforming av algebraiske uttrykk. Det er klart at man kunne gjort mye mer i disse oppgavene slik som utforming av funksjonsuttrykk og sammenlikning av disse, men dette er 8-klasseelever som ikke har gjennomgått mye av dette enda. Derfor er tanken at de skal gjøre en situasjon abstrakt, samt at denne aktiviteten kan gi dem ny innsikt i situasjonen. Oppgave 2a på begge testene handler om å identifisere antall t-skjorter eller bananer som noe som varierer, en variabel. 2 poeng på denne vil tilsvare $150t$ eller $5b$. Bruker de en annen bokstav eller andre småfeil, vil det tilsvare ett poeng. På oppgave 2b kreves det enda mer tolkning av situasjonen, det kontekstuelle, og det introduseres også et konstantledd, samt at de må klare å bruke uttrykket – vertikal matematisering. Oppgave 2c på førstesten krever at elevene tolker uttrykket, setter ord på det, og bruker det. De kan for eksempel sette opp en likning, noe de har jobbet litt med: $0 = 50t - 1500$. På posttesten må de tolke situasjonen selv, og de må også ha med et negativt konstantledd. Dette krever god kompetanse i horisontal matematisering. Det kan være at arbeid med å lage dataprogram som vi skal se på i underkapittel 3.4.3 gir dem bedre kompetanse i å tolke hvilke variabler, konstanter og hvilke regneoperasjoner man må bruke i uttrykket for å få ut ønsket resultat. Kompetanse i dette viser høy grad av relasjonell forståelse, nettopp fordi deltakerne ser relasjonen mellom praktiske situasjoner og den symbolske matematikken. Det kreves også at de klarer å tenke i bane av matematiske strukturer, kontra det operasjonelle. De må også generalisere.

Oppgave 3 på både pre- og posttesten er ganske rett frem, og er oppgaver vi typisk kan se på terminprøver og i lærebøker – som nevnt i underkapittel 2.3.2. Deltakerne må tolke figurtall og

finne et mønster, som Radford forklarte som en god aktivitet til generalisering, som nevnt i kapittel 2.3.2. Figurtallene på posttesten er noe vanskeligere å tolke, og begrunnelsen for dette er at deltakerne jobbet en del med figurtall i begge gruppene i løpet av eksperimentet, og bør derfor forsøke seg på en som kan være mer krevende. Dette handler også om testenes reliabilitet, slik at den første testen ikke påvirker den andre, og stabilitet i måleinstrumentene blir opprettholdt. Her handler det også om at de må tenke generelt og benytte bokstaven n for å gjøre det abstrakt. Utforming av algebraiske uttrykk er igjen nøkkelen. Forskning viser at det er utrolig krevende å bruke denne variabelen til å uttrykke mønster og generalitet som nevnt i kapittel 2.3.2. Denne oppgaven kan også være fin for de som ikke er så gode på å tolke tekstoppgaver. Da fungerer ikke leseferdigheter som en flaskehals for matematikkferdighetene. Denne oppgaven lar også elevene få mulighet til å se mønsteret uten å generalisere, slik at de som ikke får til generalisering kan uttrykke at de ser mønsteret på en annen måte, og man kan konstatere at det er selve den algebraiske representasjonen som er utfordrende. Oppgaven krever høy evne til å tenke abstrakt, og en løsrivelse fra konkrete representasjoner som nevnt i underkapittel 2.3.2 av Ballacheff. Vi får her se om arbeid med løkker gir større forståelse for denne formen for matematisering slik som artikulert i underkapittel 2.5.1.

Oppgave 4 går også på matematiske strukturer og generalisering, samt matematiske mønster hvis man vil tenke den veien. Oppgaven er inspirert av Lisbet Karlsen (Karlsen, 2014, s. 71). Her må elevene se strukturen i de symbolske uttrykkene, og forsøke å få det til å samsvare med de visuelle representasjonene. Det handler om å se mønster, struktur og tenke generelt om dette. Det krever ikke så høyt nivå av abstraksjon som i oppgaven under, og elevene kan i bunn og grunn prøve og feile. Tanken er her at elevene beveger seg motsatt vei enn i oppgave 3, og vi sier at denne overgangen er mindre kognitivt krevende enn fra figurtall til formel, slik som forklart i underkapittel 2.3.2. Derfor er disse figurtallene mer krevende enn i oppgave 3. I begge oppgavene, a og b, er det to riktige svar de kan krysse av for. Grunnen til dette er at de ikke skal låse seg til et narrativ, de skal altså klare å kollapse sitt eget narrativ, som Radford nevnte i kapittel 2.3.2, for å oppnå det høyeste nivået av generalisering. Nå skal det sies at det er mulig å tolke begge formlene ved å se på figuren, men hvis eleven finner ett av uttrykkene, så kan de enkelt se ved litt symbolsk manipulasjon hva som også er det andre. Det viser høy grad av generalisering, og elevene navigerer mellom ulike registre, og viser derfor stor matematisk forståelse, samt kompetanse i vertikal matematisering. Her får elevene 1 poeng for riktig, og -1 poeng for feil. Dette er for å hindre elevene i å tippe tilfeldig, eller eventuelt krysse av på alle. Dette gir dem mulighet til å få 2 poeng på hver. Løsning av denne oppgaven viser stor relasjonell forståelse for de algebraiske uttrykkene.

Oppgave 5 er en viktig og relevant oppgave i begge testene, og det er her det introduseres generalisering og matematiske strukturer til å bedrive matematiske bevis. Læreplanen støtter godt oppunder dette, som nevnt i underkapittel 2.3.2, og vi ser et lignende eksempel i utdanningsdirektoratets utkast til matematikkeksamen. Her kreves det høy nivå av relasjonell forståelse, der elevene må bruke flere elementer samtidig som de jobber abstrakt. Her kreves det at elevene klarer å tenke på et uttrykks struktur, at de klarer å på det som noe som ikke nødvendigvis skal utregnes, men som et generelt uttrykk. Deltakerne er avhengig av å lage algebraiske uttrykk for å i det hele tatt få til oppgaven. På posttesten ville det vært ønskelig om deltakerne trakk ut to som faktor slik: $2n-1+2n-1=4n-2=2(2n-1)$. Siden du får to som faktor så er det et partall, men dette er nok litt for avansert for deltakerne. Derfor ble det hintet til at de kunne se om uttrykket kunne deles på to, som er en noe enklere variant. Dette er på grunnlag av at deltakerne ikke har jobbet så mye med å lage parentesuttrykk. Også det faktumet at den vertikale matematiseringen ikke er det viktigste her, men heller at de får til å lage et algebraisk uttrykk og til en viss grad mestrer å bruke det. Noen av deltakerne kan derimot få det til. Men dette er krevende, som nevnt, og krever et høyt nivå av generalisering, og derfor er hint lagt til. Tanken er som nevnt i underkapittel 2.5.1 at arbeid med løkker konkretiserer for elevene den abstrakte tenkninger som kreves her. Derfor kan man se om ulike måter å arbeide på gir ulik effekt på denne oppgaven.

3.4.3 Kontroll- og eksperimentell gruppe

Gruppeinndelingen er et viktig element for validiteten til prosjektet da dette kan være en av de største faktorene som påvirker målingene i posttesten. Som nevnt er det viktig for et ekte eksperiment at gruppene er 100% tilfeldig, som vil si at jeg som forsker ikke har noe påvirkning på inndelingen. Jeg har valgt kvasieksperimentelt design, noe som betyr at gruppene ikke er 100% tilfeldig, men kun delvis. Deltakerne ble delt i to grupper basert på resultatene på pretesten. Kontrollgruppen får tildelt normal undervisning, og den eksperimentelle gruppen får tildelt eksperimentell undervisning – nærmere bestemt læring av algebra gjennom programmering. Hva som er normal undervisning, eller normal algebraopplæring, kan være krevende å definere og påstå, og er et viktig element i validiteten i denne masteravhandlingen. Hva normal undervisning vil tilsvare, vil videre bli forklart og begrunnet i neste underkapittel.

Resultatene på pretesten ble seende slik ut som i tabell 3.2. Gruppene var bestående av 28 ungdomsskoleelever, nærmere bestemt 8-klassinger. Man kunne gått ut ifra karakterene som elevene hadde i matematikk fra før av, basert på terminprøve på høsten, men dette kunne by på

utfordringer. For terminprøven ble gjennomført en god tid før selv eksperimentet som ble gjennomført i april, og kunne derfor ikke gi tydelige indikatorer på hva de har lært i mellomtiden. Nå er jeg kun interessert i algebrakompetanse også, og derfor kan for eksempel karakteren 4 ikke være en reel indikator da elevene kunne ha prestert svakt i algebra, og sterkt i et annet tema. Det kunne også vært mulig å ha delt elevene etter interesser eller motivasjon, men det hadde blitt for mye arbeid. Derfor er disse egenskapene opp til tilfeldighetene, siden en RNG ble brukt for å dele opp like par basert på pretesten. Det er enkelt å forholde seg til algebraferdigheter, for det er jo nettopp det som skal måles i etterkant også, og for å styrke ytre validitet og generaliserbarhet, valgte jeg det. I tabell 3.2 ser du variabelen algebraferdigheter på alle deltakerne, og hver deltaker identifisert med et deltakernummer – dette er resultatene på pretestene. I tabell 3.3 ser man parene, som igjen ble noe justert for å få likt gjennomsnitt i begge gruppene. Gjennomsnittene ble for øvrig identisk i begge gruppene, men standardavviket divergerte noe, men er innenfor rimelighetens rammer.

Deltakernummer	Algebraferdigheter
1	2
2	9
3	11
4	0
5	5
6	5
7	8
8	13
9	5
10	0
11	1
12	6
13	9
14	15
15	4
16	3
17	1
18	15
19	4
20	12
21	5
22	6
23	17
24	8
25	12
26	2

27	4
28	6

Tabell 3.2: Resultater på pretest

Eksperimentell Gruppe		Kontrollgruppe	
AF	DN	DN	AF
2	**1**	**26**	2
0	**10**	**4**	0
6	**22**	**11**	1
5	**5**	**15**	4
5	**21**	**6**	5
6	**12**	**9**	5
6	**28**	**27**	4
4	**19**	**16**	3
9	**13**	**24**	8
8	**7**	**2**	9
12	**20**	**3**	11
15	**14**	**25**	12
1	**17**	**23**	17
15	**18**	**8**	13
μ	6,71428571		6,71428571
σ	4,6974391		5,06007862

Tabell 3.3: Eksperimentell- og kontroll-gruppe

I tabell 3.3 vil venstresiden tilsvare den eksperimentelle gruppen, der variabelen DN tilsvarer deltakernummer slik som beskrevet i tabell 3.2. Variabelen AF vil tilsvare algebraferdighet slik som mål av pretesten og som beskrevet i tabell 3.2. Høyresiden tilsvarer kontrollgruppen. De deltakernumrene som står inntil hverandre på samme linje på midten, er parene, som det kan tenkes at blir splittet på midten i hver sin gruppe for å holde på likheten i gruppene. Gjennomsnittet og standardavviket står nederst.

Nå som begge gruppene er inndelt, må man se på andre faktorer som kan påvirke forskjeller i gruppene. Det er mange trusler til det vi kaller for indre validitet som kan være med på å påvirke forskerens mulighet til å konkludere at det er den manipulerte variabelen som påvirker resultatet (Creswell & Creswell, 2018, s. 242). Derfor må disse andre faktorene kartlegges, slik at verdiene til andre variabler er så identiske som mulig i begge gruppene, slik som tid på dagen eller vær, eller annen tilfeldig bakgrunnsstøy (Field, 2018, ss. 69–70). Et annet viktig element er drop-outs, for hvis noen deltakere ikke møter opp eller trekker seg, vil dette resultere i at resultatet kan bli påvirket

(Creswell & Creswell, 2018, s. 243). For hvis en deltaker mangler i den ene gruppen, grunnet for eksempel sykdom, må deltakeren i det paret som er i den andre gruppen også «fjernes». Hvis ikke ville ikke gruppene vært like lenger. Dette kan bli problematisk da parene ble noe justert for å få like grupper. Det er også et velkjent fenomen at deltakerne kan snakke med hverandre om de ulike behandlingene som de utsettes for, og dette kan påvirke resultatene (Creswell & Creswell, 2018, s. 243). Deltakerne ble derfor oppfordret til å ikke snakke med hverandre om eksperimentet i løpet av dagene som eksperimentet foregikk. Det ble laget en tabell for å utelukke forvirrende variabler, eller medierende faktorer, som kan påvirke resultatet; se tabell 3.4.

Variabel	Håndtering
Tid og tid på dagen	Motivasjonen kan variere i løpet av døgnet. Derfor ble undervisningsopplegget i begge gruppene gjennomført på samme tid. Begge gruppene jobbet i 2.5 time. Gruppe 1 mandag 08:30-10:00 og 10:15-11:15. Gruppe 2 tirsdag 08:30-10:00 og 10:15-11:15.
Vær	Det kan være mer motiverende å jobbe med matematikk når det regner ute, kontra på en solskinnsdag der man heller ønsker å være ute og for eksempel spille fotball. Derfor ble eksperimentet plassert på to dager med likt vær, basert på prognosene til yr.no.
Lærer	Forskning viser at læreren er den største faktoren på elevenes læring enn noen andre variabler (Boaler, 2016, s. 57). Derfor måtte læreren være den samme i begge gruppene, slik at ulik lærer ikke påvirker resultatene. Jeg underviste derfor begge gruppene. Derfor er jeg en stor faktor her, og det ble viktig at jeg underviste med lik innlevelse og på samme måte i begge gruppene. Bortsett fra den manipulerte variabelen.
Motivasjon	Elevene fikk ikke vite hvilken gruppe som var den eksperimentelle. De fikk vite at vi skulle se på to ulike undervisningsopplegg. Det ble også sørget for at den eksperimentelle gruppen gjennomførte etter kontrollgruppen. Dette var for at den andre gruppen ikke så på det som en nedtur at de ikke jobbet med datamaskinen, hvis de på et eller annet vis hadde fått det med seg. Begge gruppene ble oppfordret til å ta dette seriøst, slik at de kunne dra på kino sammen i ettertid.

Tabell 3.4: Medierende faktorer

3.4.4 Undervisningsopplegg i kontroll- og eksperimentell gruppe

Dette er på mange måter det viktigste med forskningsprosjektet. For det er her selve forskerens intervensjon kommer inn. Det er her det skal endres på den uavhengige variabelen (programmering), slik at vi muligens ser endring i den avhengige variabelen (algebraferdigheter), og kan trekke en kausalitet mellom variablene. Jeg ønsker som nevnt, å se om programmering kan være et alternativ til vanlig undervisning, ikke et kompliment, derfor var det viktig som nevnt i tabell 3.4 at begge undervisningsoppleggene tok like lang tid. Det er også elementært at kontrollgruppens undervisningsopplegg følger normal standard, og at dette vektlegges på lik linje som undervisningsopplegget i den eksperimentelle gruppen. Derfor er det viktig at kontrollgruppens opplegg samsvarer med kunnskapsstatusen nevnt i denne avhandlingen, og at den eksperimentelle gruppen følger det samme, men med programmering i tillegg. Noe av det viktigste her er også å ikke skape en spennende wow-effekt med den eksperimentelle gruppen (Creswell & Creswell, 2018, s. 243); derfor skal begge gruppene jobbe på en moderne og prosjektbasert realistisk måte; de må være likestilte, og kontrollgruppens undervisningsopplegg må samsvare med måten det undervises på til vanlig i norsk ungdomsskole. Modellering vil være nøkkelen i kombinasjon med algebra slik som nevnt i underkapittel 2.3.3. Begge undervisningsoppleggene skal ha aktive elever som konstruere sin egen kunnskap. De skal konstruere sin egen kunnskap ved å bygge på det de allerede kan fra før av og oppdage nye ting i relasjon til noe annet, og jeg som lærer er der kun som en veileder, en som sammenkobler elevenes nett av sammenhenger – det vil passe bra med læringsteoriene nevnt i underkapittel 2.2. Det vil derimot være korte introduksjoner til hvert tema før deltakerne setter i gang selvstendig arbeid. Det sosiokulturelle læringssynet påvirker mye av undervisningsoppleggene i det forstand at elevene er aktive, de samarbeider og prater med hverandre og samarbeider i par, og at jeg som lærer fungerer som en mediator. Vi får også se effekten av datamaskinens stillasbyggende funksjon i den eksperimentelle gruppen på læringen. Elevene jobber i læringspar i begge gruppene, og konsentrerer seg om heftene som du ser i vedlegg 13 og 14 – ha disse oppe når jeg går gjennom oppgavene. Jeg gjennomførte en pilotrunde som foreslått av Creswell & Creswell for å justere det eksperimentelle gruppens undervisningsopplegg i tilfelle komplikasjoner eller misoppfatninger kunne oppstått – på en annen klasse så klart (2018, s. 228). Elevene i den eksperimentelle gruppen fikk også nødvendig utstyr slik som datamus, slik at lite tilgang på digitalt utstyr ikke fungerte som en mulig flaskehals. Nødvendige dataprogrammer slik som Thonny var installert på forhånd.

I vedlegg 11 og 12 kan man se den pedagogiske planen, og selve gjennomføringen. Begge gruppene hadde en 15 minutters oppfriskning i starten. I kontrollgruppen hadde vi en klasseromssamtale om algebra, og elevene hentet frem det de husket. Vi laget noen eksempler på algebraiske uttrykk og pratet om hvorfor vi drev med algebra. Dette anses som en fin oppstartaktivitet der elevene eller deltakerne henter frem kunnskapen sin (Karlsen, 2014, s. 65). Det gir blant annet også læreren god oversikt over hva elevene kan. I den eksperimentelle gruppen ble dette gjort på en annen måte. Der ble heller variabler og print()-funksjonen introdusert på en liknende måte, og elevene kunne se hvordan man på datamaskinen kunne lage algebraiske uttrykk. Dette baserer seg på det som ble løftet frem i underkapittel 2.5.1 om at dataprogrammering ga større forståelse for bruken av variabler, og at ved å la datamaskinen gjør utregningene, kunne man få en større forståelse for det strukturelle, og ikke vektlegge det operasjonelle. På den måten kan man gjøre to ting samtidig, friske opp algebra samtidig som man lærer programmering. Vi utvider og tar i bruk elevens nett av sammenhenger for å bygge relasjonell forståelse. Dette er noe av grunnlaget i det Radford løfter frem i underkapittel 2.3.4 om kvasialgebraiske aktiviteter, og som Bråting & Kilhamn nevnte i underkapittel 2.5.1. Aktiviteten som etterfulgte denne gjennomgangen i kontrollgruppen er en moderne undervisningsmetode som kalles for «the odd one out», og den muliggjør refleksjon og forståelse, og stimulerer til samtale i gruppene. De jobber på det strukturelle planen, og må bruke språket til å begrunne og reflektere. Et noe lignende ett ble gjennomført i den eksperimentelle gruppen der de skulle tolke algebraiske uttrykk på datamaskinen. De kunne også teste ut dette selv, for da får man aktualisert uttrykkene, og det blir mer konkret for elevene som nevnt i underkapittel 2.5.1. Fellesnevneren i begge disse oppleggene er å se det strukturelle i matematiske eller algebraiske uttrykk, og det samsvarer godt med kunnskapsstatusen i 2.3.2. Den eneste forskjellen er at programmering er til stede i den ene gruppen, og ikke til stede i den andre.

Etter dette begynte virkelig det viktige arbeidet, og vi entrer domenet av horisontal matematisering. Det er her de skal beskrive realistiske situasjoner i matematikkens bekledning, og muligens få nyere innsikt i et situasjonene, slik som beskrevet i underkapittel 2.3.3. Dette måtte man naturligvis gjøre i begge gruppene for å opprettholde en likhet i undervisningsoppleggene. Oppgave 2c i kontrollgruppen bærer i tillegg preg av problemløsning, der de må være gode i vertikal matematisering. Det samme med oppgave 3c. Uansett så er det viktigste her er at elevene får til å identifisere hvilke variabler og tall man skal bruke; for det er jo nettopp det utforming av algebraiske uttrykk handler om. I oppgave 2 i den eksperimentelle gruppen skal deltakerne lage dataprogrammer, og det hele dreier seg om de algebraiske uttrykkene som de bruker inne i print()-funksjonen. Her skinner programmeringen virkelig, da elevene kan lage små dataprogrammer ved

hjelp av input()-funksjonen, og får derfor en ny konkret referent. Det ser vi aktualisert i hele oppgave 2. Tanken er at dette blir mer motiverende og konkret for elevene som nevnt i underkapittel 2.5.1. Det er mulig at dataprogrammene virker mer hensiktsfulle og funksjonelle for elevene, som nevnt i underkapittel 2.5.1. Vi kan allerede her si at deltakerne i den eksperimentelle gruppen lager algoritmer som brukes av en datamaskin. I den enkleste forstand vel å merke.

I kontrollgruppens oppgave 4 jobbet de med en bevisoppgave. Her pratet de sammen i par om partall på generell form og jobbet på en vanlig måte slik som man ser i utkast til matematikkeksamen i vedlegg 2. De jobbet deretter videre med figurtall. Dette ansees som svært vanlig, som nevnt i underkapittel 2.3.2, og vi ser også hvordan man jobber på lignende måte med figurtall i Matemagisk 8, en lærebok etter fagfornyelsen som brukes mye i Norge (Kongsnes & Wallace, 2020a, ss. 116–128). Jeg har tidligere nevnt hvorfor vi arbeider med figurtall og matematiske bevis. I den eksperimentelle gruppen forgår det litt annerledes. I stedet for å prate sammen om partall på generell form, skulle de utforske og eksperimentere dette i oppgave 3a, b og c. Det blir som nevnt i underkapittel 2.5.1 som om at vi er på naturfagslaben – bare at vi jobber med matematikk. Dette gir forhåpentligvis en ny forståelse for abstrakt tenkning. Det som gjør dette unikt er at det fungerer som en synergi, slik som nevnt av Misfeldt og Duun i underkapittel 2.5.1, og her utforsker de matematiske sammenhenger samtidig som de jobber med og lærer programmering, og muligens vil den kombinerte effekten gi enda høyere effekt på læring. Slik kan vi virkelig slå flere fluer i en smekk. De jobbet deretter videre med de samme figurtallene som i kontrollgruppen, men med den fordelen av at de kunne forsøke å skrive det algebraiske uttrykket i print()-funksjonen i en for-løkke for å teste ut. Tanken er ikke her at de skal prøve og feile, men at det kan fungere som en støtte for å justere det algebraiske uttrykket og umiddelbart se effekten av dette, uten å måtte være avhengig av den kognitivt krevende abstrakte tenkningen, som nevnt i underkapittel 2.5.1. Videre i utdanningsløpet til deltakerne kunne de gjort en lignende aktivitet når de skulle lære seg funksjoner i matematikkfaget, der de lager funksjonsmaskiner som dataprogrammer, og gir disse til hverandre og kan gjette funksjonsuttrykket til hverandre etter et antall forsøk.

Den siste viktige delen av undervisningsoppleggene er hvordan jeg som lærer har planlagt å gi rettferdig eller lik behandling til begge gruppene. Merk at det ikke er forventet å «skape» realistiske klasseromssituasjoner. Jeg er innforstått med at disse kan bli noe kunstige, men hvis man kartlegger så mye som mulig, lar undervisningene fungere på normal måte, kan vi få et så valid resultat som mulig, og har derfor mulighet til å generalisere resultatene. En viktig faktor er at jeg kjenner

deltakerne, og vet derfor hvem jeg skal gi ekstra støtte til. Det vil derfor også virke naturlig og mer realistisk siden jeg er noen som elevene kjenner, og derfor vil ikke en ukjent lærer påvirke resultatene negativt. Derfor vil det man observerer, resultatene, kunne være mer normale i den forstand at det gir et korrekt bilde av hvordan det fungerer i skolen. Dette var heller ikke ekstra undervisning for elevene da eksperimentet var i skoletiden, noe som igjen førte til at elevene tok dette seriøst. Jeg måtte hele tiden gjøre det samme i begge gruppene. Veilede på lik linje, med lik innlevelse, og hele tiden bruke elevenes nett av sammenhenger.

4 Analyse og resultater

4.1 Observasjoner

Gjennomføringen i begge gruppene gikk bra, og alle deltakerne deltok i begge gruppene under eksperimentet og på posttesten. Dette bidro til at likheten basert på pretesten ble opprettholdt. Alle deltakerne syntes det var spennende å få være med på et forskningsprosjekt, så bekymringen løftet frem om deltakernes mulige mangel på motivasjon i underkapittel 3.4.1 var heldigvis bare en bekymring. Deltakerne tok arbeidet seriøst, og klasseromssituasjonen fikk en naturlig flyt, slik som i klasserommet til vanlig. Det virket lite kunstig, og etter min erfaring som lærer, var det som en hvilken som helst time. De umiddelbare observasjonene var at deltakerne i den eksperimentelle gruppen ble mer motivert, og ville til og med utvide funksjonaliteten i dataprogrammene som de lagde. En deltaker utvidet blant annet valutaprogrammet til å regne med flere valutaer, slik som yen, pund og til og med kryptovaluta. Det var fint å se at deltakerne hygget seg da de samtidig drev med så læringsfylte aktiviteter som i tillegg, ifølge kunnskapsstatusen i kapittel 2.5.1, fostrer algebraisk tenkning. Synergien mellom programmering og algebra kommer da virkelig frem. Deltakere som til vanlig sliter med algebra, så ikke på algebraen som noen utfordringer, og spurte heller om hjelp når det var noe programmeringsteknisk de ikke håndterte. De spurte sjeldent om hjelp til algebraen, fordi det ble naturlig for dem hva som skulle stå inne i print()-funksjonene – altså de algebraiske uttrykkene. Dette kan være fordi, som nevnt, at det ble mer konkret for deltakerne, og variablene fikk en kontekst. I kontrollgruppen ble det gitt mest veiledning til utforming av de algebraiske uttrykkene, og det matematiske beviset. Det virket som at deltakerne i den eksperimentelle gruppen forsto dette med generelle tall etter at de arbeidet med dette i løkker, for de trengte lite veiledning til de oppgavene. To av deltakerne nevnte blant annet at de endelig fikk en forståelse for dette med variabelen n etter at de jobbet med løkker for å skrive ut tallfølger. Jeg kan derfor ikke understreke hvor lærerikt og interessant det ville vært for elevene å drive med dette over en lengre periode, slik at de kan jobbe på store og meningsfylte prosjekter innenfor programmering, og der de virkelig ville beveget seg inn i den algoritmiske tankegangen. Dette er bare en liten del av toppen av isfjellet. Det virket som abstraksjon krevde mindre kognitiv aktivitet for dem i den eksperimentelle gruppen da denne ble en naturlig del av programmeringen. Deltakerne i kontrollgruppen arbeidet også veldig godt, og flere fortalte i ettertid at de fikk bedre kontroll på algebra, og kunne se nytten av det, slik som nevnt i underkapittel 2.3.2. Dette er bra, og et viktig kriterium for å opprettholde likhet mellom gruppene. Kontrollgruppen jobbet mer konkret rettet mot posttesten, noe som ga meg bekymring for om dette kunne ha vært et feilsteg, og igjen ville gi dem en fordel på posttesten. Det

var derimot bra at begge undervisningsoppleggene ble opplevd som lærerike, og på lik linje med hverandre. For derfor blir også resultatene mer reelle.

4.2 Resultater

Resultatene på de rettete testene ser man imidlertid i tabell 4.1. Der ser man den inkrementerende variabelen DN (deltakernummer), som kan ansees som en primærnøkkel, sammen med den dikotome elle binære variabelen PR (programmering), som enten har verdi 0 for ikke til stede, eller 1 for til stede. Dette er altså om intervensjonen er til stede eller ikke, selve manipulasjonen av den uavhengige variabelen. Kontrollgruppen har derfor 0 i verdi her, og den eksperimentelle har 1. Variablene «Pretest» og «Posttest» som er resultatene fra måleinstrumentene, testene, kommer etter førstnevnte, og viser som nevnt en diskrete intervallvariabel. Dette er selve deltakernes verdier i algebraferdigheter. Det er derimot satt opp annerledes for å gi en oversikt over pre og post-testene. Her får man også en oversikt over forbedring eller forverring hos de ulike deltakerne. Størst differanse mellom prøvene var hos deltaker 5 i den eksperimentelle gruppen som fikk en forbedring på hele 7 poeng, noe som kan tilsi at deltageren virkelig knakk en kode etter å ha jobbet med programmering. Denne deltakeren fikk 5 poeng på pretesten, men hele 12 poeng på posttesten. Nå er det ikke differansen mellom pretesten og posttesten vi skal se på i denne avhandlingen, men heller differansen mellom posttestene – som nevnt i kapittel 3. Det er likevel fint å se slike forbedringer.

DN	PR	Pretest	Posttest
2	0	9	15
3	0	11	11
4	0	0	0
6	0	5	5
8	0	13	16
9	0	5	5
11	0	1	3
15	0	4	4
16	0	3	6
23	0	17	17
24	0	8	9
25	0	12	12
26	0	2	1
27	0	4	5
1	1	2	5
5	1	5	12
7	1	8	6
10	1	0	2

12	1	6	6
13	1	9	14
14	1	15	16
17	1	1	3
18	1	15	16
19	1	4	5
20	1	12	12
21	1	5	8
22	1	6	7
28	1	6	10

Tabell 4.1: Resultater på pre- og post-test

PR	σ (Std)	μ (Gj.snitt)
0	5,5771315	7,78571429
1	4,6480529	8,71428571

Tabell 4.2: Gjennomsnitt av algebraferdigheter i begge gruppene

I tabell 4.2 ser man de oppsummerende resultatene, målingene gjort på posttesten, og vi kan se på gjennomsnittet at den eksperimentelle gruppen presterte bedre enn kontrollgruppen. Der kan man se kontrollgruppen med verdi 0 i PR (programmering) og den eksperimentelle gruppen med verdi 1 i PR. Etterfulgt av dette ser vi standardavvik og gjennomsnitt. Vi ser at det er et årsak-effekt-forhold mellom programmering og algebraferdigheter, siden vi fikk forskjellige verdier i de kontinuerlige variablene gjennomsnitt. Det er nå man kan bruke slutningsstatistikk for å sammenlikne de to kontinuerlige variablene: μ_0 og μ_1. Uansett så er dette banebrytende funn, for det viser bedre resultat i den eksperimentelle gruppen, og det viser at man kan drive med pedagogisk programmering i algebraopplæringen på lik linje som vanlig algebraopplæring, og at det i tillegg kan gi bedre effekt på algebraferdigheter – synergien kommer frem. Som nevnt innledningsvis, kan dette derfor være et svar på utfordringene vi har i norsk skole innenfor algebra, samtidig som vi inkluderer de nye elementene av læreplanen. Dette forteller derimot ikke noe om den langsiktige påvirkning av pedagogisk programmering. Det kan tenkes at over en lengre periode, kan man også se om den påståtte motivasjonen som pedagogisk programmering gir, slik som nevnt i underkapittel 2.5.1 av Egara, og egne observasjoner, på læring av algebra over en lengre periode. Denne avhandlingen hadde derimot ikke tid eller ressurser nok til å ta for seg en slik studie.

Det store spørsmålet er om differansen mellom resultatene på posttesten er signifikant nok til å kunne påstå at dette ikke skyldes tilfeldigheter, og at om derfor kan generalisere funnene til å gjelde for hele populasjonen av ungdomsskoleelever i Norge. Jeg ønsker altså å se om i tillegg til at det

kan være på lik linje med algebraopplæring, at man også kan si at det er bedre, slik at flere vil inkorporere dette i sin undervisning. Siden jeg valgte den mer radikale, eller overskriftsskapende, tilnærmingen, valgte jeg å se om dette til og med kunne være bedre enn vanlig undervisning. Dette var også mye på grunn av kunnskapsstatusen i underkapittel 2.5.1 som påsto at det kunne gi elevene nye referenter eller utvide sitt nett av sammenhenger. Hypotesetesten, som ble testet med en «independent t-test» ser du i tabell 4.3 og 4.4. Fordelen med SPSS er at den også sjekker om variansen var signifikant avvikende eller ikke, ved å bruke «Levene's test», noe som den ikke var. Variansene var avvikende med $p = 0.387$, som du ser i tabell 4.3. Den var altså ikke signifikant avvikende med $p < 0.05$. Derfor kan man konkludere med at variansene er like nok til å gjennomføre en vanlig t-test. Jeg ønsket å se en signifikans på $p \leq 0{,}2$ på t-testen, noe som tilsvarer 20% eller mindre signifikansnivå, i den ensidige t-testen. Den viste derimot signifikans på $p \approx 0{,}32$ som man kan se i tabell 4.4. Det betyr at det er 32% sannsynlig å feilaktig forkaste nullhypotesen. Dette gir altfor stor sjanse for type 1 feil. Det betyr at jeg vil ta feil 1 av 3 ganger hvis jeg konkluderer med den alternative hypotesen. Differansen er derfor ikke stor nok til å konkludere. Jeg kan derfor ikke komme med påstanden om at pedagogisk programmering sannsynligvis fungerer bedre enn vanlig algebraopplæring når det gjelder effekt på læring av algebra – nærmere bestemt utforming av algebraiske uttrykk. Men det at gjennomsnittet på posttesten i den eksperimentelle gruppen i det hele tatt var større, er banebrytende. Hele poenget med signifikanstesting som nevnt i underkapittel 3.3.3, er å ta hensyn til tilfeldighetene i gruppene. Siden jeg har redusert disse tilfeldighetene til et minimum, vil jeg kunne påstå at denne avhandlingen har funnet det den ønsket. Nemlig at det er mulig å drive med pedagogisk programmering som et alternativ til vanlig undervisning, og at det i tillegg muligens kan gi bedre effekt på læring av algebra. Det kan tenkes at det derfor med flere deltakere, og gjerne over lenger tid, vil kunne være mulig å finne en differanse med lavere signifikansnivå – til og med ned på 5%-nivået. Ikke bare kan det fungere på lik linje, men man kan også med veldig lav sikkerhet, påstå at det kan være bedre. Ser vi derfor tilbake på nullhypotesen som forteller at det fungerer på lik linje, kan vi derfor ikke påstå at denne skal forkastes og erstattes med den alternative hypotesen som artikulerer at pedagogisk programmering fungerer enda bedre enn tradisjonell undervisning. Siden nullhypotesen også er av et noe alternativt preg, og vi fant en differanse mellom de to, vil jeg påstå at man med stor sikkerhet kan si at man kan drive med pedagogisk programmering som et alternativ til vanlig undervisning. Jeg kan ikke konkludere mer at det fungerer bedre.

Levene's Test for Equality of Variances	
F	Sig.
0,773	0,387

Tabell 4.3

t-test for Equality of Means							
t	df	Significance		Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
		One-Sided p	Two-Sided p			Lower	Upper
-0,479	26	**0,318**	0,636	-0,929	1,940	-4,917	3,060

Tabell 4.4

5 Diskusjon

5.1.1 Refleksjon rundt masteravhandlingens tema og funn

I introduksjonen til denne masteravhandlingen ble det artikulert en bekymring rundt norske ungdomsskoleelevers prestasjoner i algebra basert på TIMSS-undersøkelser. Dette er som nevnt urovekkende, spesielt med tanke på det mangfoldet av kompetansemål og kjerneelementer i matematikkfaget som nå mer eller mindre er avhengig av algebra – nærmere bestemt utforming av algebraiske uttrykk og horisontal matematisering (Utdanningsdirektoratet, 2020b). Jeg kan trekke frem tre kjerneelementer bare for å nevne noe: modellering og anvendelser, resonnering og argumentasjon, abstraksjon og generalisering (Utdanningsdirektoratet, 2020c). Skal man ha en relasjonell forståelse for algebra, slik som denne masteravhandlingen etterlyste, så trenger man å forstå hva algebraiske uttrykk betyr – alle disse elementene fra læreplanen så vel som utforming av algebraiske uttrykk krever dette. Fagfornyelsen legger som nevnt implisitt mer vekt på algebra. Utfordringene som vi i norsk skole har på dette feltet, vil derfor kunne føre til at fagfornyelsen setter en enorm demper på matematikkferdighetene. For som Kaput et al. nevnte i underkapittel 2.3.1, så blir det epistemologiske grunnlaget for algebra ofte omtalt som en inngangsport til høyere matematikk. Skal vi derfor drive med meningsfull matematikk som løftet frem i 2.3, en matematikk som er avhengig av å bruke algebra, en matematikk som fagfornyelsen inviterer til, så må vi endre på noe i norsk skole.

Grunnen til at vi har utfordringer på dette temaet, kan som nevnt ligge i at vi legger for stor vekt på den operasjonelle aritmetikken på barneskolen som nevnt innledningsvis av Kieran. Denne inngangen til algebra er som nevnt i underkapittel 2.3.2 lite hensiktsmessig. Dette korrelerer godt med TIMSS-undersøkelsene, for vi kunne se at på barneskolen presterte vi godt over norm, før vi presterte langt under våre kollegaer i Norden på ungdomsskolen – når algebra blir introdusert. Det ble forklart gjennom kunnskapsstatusen hvordan definisjonen til Kaput om at algebra handler om generalisering, studiet av strukturer og modellering, er en gjeldende og relevant definisjon. At elever ser på et matematisk uttrykk som noe som skal kalkuleres eller beregnes, kan være et stort hinder for disse kunnskapsområdene. Elevene trengte en løsrivelse fra det operasjonelle, og over til det strukturelle. Det ble løftet frem hvordan man er avhengig av et høyt nivå av abstrakt tenkning i underkapittel 2.3.4, og at denne kognitivt krevende aktiviteten blir utfordrende uten konkrete referenter. På barneskolen jobber man mer konkret. Disse utfordringene er sentrale å håndtere for å kunne få til den horisontale matematiseringen og utformingen av algebraiske uttrykk.

Programmeringens fremtreden i matematikkfaget ble sett på som en mulig velsignelse i forkledning. På bakgrunn av TIMSS-undersøkelser i England og en kunnskapsstatus som løfter frem sammenhengen mellom programmering og algebraferdigheter, kan man se at det finnes en korrelasjon mellom disse fenomenene. Om det er en kausalitet derimot, er det som denne masteravhandlingen forsøker å svare på. For det er nettopp sånn at det kan finnes en synergi mellom disse, slik som nevnt i underkapittel 2.5.1. Det betyr at kombinert vil disse to fenomenene gi større effekt på læring av algebra, enn hvis man arbeider med dem hver for seg. Det er også enormt tidsbesparende, fordi da kan man krysse av flere kompetansemål samtidig. Programmering som kompetansemål, algebra som kompetansemål, et mangfold av kjerneelementer, en god del av matematikkfagets fagrelevans og sentrale verdier, og problemløsningsmetoden algoritmisk tenkning. Kombinasjonen av dette vil muligens gi enda større effekt på læring av algebra, samtidig som at det er mer motiverende og hensiktsfult for elevene som nevnt av Egara et al. Jobber man med flere temaer samtidig, vil det som nevnt også styrke relasjonell forståelse og dybdelæring. Dette kan ikke beskrives som noe annet enn det engelske begrepet en «grand slam home run».

Radford forklarte videre hvordan man kunne ta i bruk flere metoder for å bedrive algebraisk aktivitet, ikke nødvendigvis algebraisk symbolspråk. Målet er at elevene skal forstå, og hvis algebra blir redusert til et kommunikasjonsspråk, der språket ikke medierer algebraisk tenkning, men heller fungerer som en flaskehals for algebraisk tenkning, vil ikke den relasjonell forståelse oppnås. Tanken var at vi kunne bruke elevenes nett av sammenhenger, utvide til flere semiotiske registre, og ta i bruk programmering for å styrke den relasjonelle forståelsen for algebra. Datamaskinen kunne fungere som en mediator, samtidig som at programmeringsspråket kunne mediere algebraisk tenkning. Sammen kunne de ulike representasjonen gi ny forståelse for algebra. Derfor ville det være naturlig å undersøke dette på en mer systematisk måte gjennom et forskningseksperiment.

Bråting og Kilhamn fortalte at programmering gikk et steg mot det strukturelle, kontra det operasjonelle. Forskningen til Healy et al viste for eksempel at elever aksepterte at bokstaver kunne stå for generelle tall, akseptere og arbeide med åpne algebraiske uttrykk, etter arbeid med programmering. Misfeldt og Duun nevnte hvordan man kan få et mer konkret utgangspunkt til abstrakt tenkning, altså at datamaskinen gir en slags ny referent. Jeg ville gjøre som Healy et al. nemlig å undersøke om man kan lære matematiske temaer, algebra spesifikt, gjennom aktiv konstruksjon av egen kunnskap i et datamiljø gjennom itererende prosesser med redigeringer og tilbakemeldinger uten å være avhengig av abstrakt tenkning. Masteravhandlingen viste i samsvar med Healy et al. at datamaskinen fikk en stillasbyggende funksjon samtidig som det også medierte

nye tankemåter, som resulterte i relasjonell forståelse av algebra. Vi så også, i samsvar med dem, at arbeid på datamaskinen fostret matematiske samtaler da dette ble noe mer håndfast og konkret for elevene å arbeide med og prate om. Tanken min er derfor at hvis elever begynner tidlig med programmering på barneskolen, litt som i algebra tidlig-bevegelsen nevnt i underkapittel 2.3.2, så kan overgangen til algebra gå mer naturlig kontra fra den tradisjonelle aritmetikkens om elever jobber med på barneskolen. Utforming av algebraiske uttrykk blir mer naturlig for elevene, de får en relasjonell forståelse for uttrykkene, noe som er et godt utgangspunkt for videre vertikal matematisering.

Det virker som dette var en god metode for å bruke elevenes nett av sammenhenger. For ved å gi dem en ny referent, dataprogram, klarte de fort å knytte en bro mellom det og symbolsk algebra. De klarte å se det i relasjon til hverandre, og derfor få forståelse. Dette lar dem også se dette i relasjon til andre matematiske temaer igjen, som jo er ønskelig som nevnt i kapittel 2. Det virker også som at dette nye registeret som vi kan kalle det, ga dem en synergi, slik som nevnt av Duval, en synergi senere nevnt av Misfeldt og Duun, som styrket forståelsen for algebra – på grunnlag av resultatene. Med meget lav sikkerhet derimot. Dette er gode funn. Det kan derimot kreves mer forskning, slik som nevnt av Braating og Kilhamn om sammenhengen mellom de ulike registrene. Utfordringer mellom dem og i overgangene mellom. Det så derimot ut som at dette var håndterbart i dette forskningsprosjektet.

Det sosiokulturelle læringssynet skinte virkelig gjennom her. Arbeid med dataprogram fostret samarbeid og elevene pratet sammen om mengder og generalisering, slik som nevnt av Healyt et al. Det fostret matematisk samtale, uten at elevene kanskje helt tenkte over at de pratet matematikk. Deltakerne samarbeidet med hverandre, og var ikke avhengig av å individuelt gjøre utregninger. Det var mer relevant hvordan uttrykk skulle struktureres, og det pratet om det strukturelle aspektet, slik som etterlyst av flere forfattere; blant annet Sfard. Derfor kan vi også si at det virket som at datamaskinen og programmeringsspråk fostret algebraisk tenkning og muliggjorde utforming av algebraiske uttrykk. Dette bygger seg på teorien om læring som mediert, som vil si at kulturelt skapte artefakter gir en måte å tenke på. Datamaskinen ga elevene en måte å tenke algebraisk på. Man er derfor ikke kun avhengig av algebraisk symbolspråk – slik som Radford nevnte.

Kanskje det aller mest banebrytende her er hvor lite jeg som lærer trengte å bidra med i timene. Det er jo ønskelig at elevene skal være aktive, og at læreren skal bedrive minst tavleundervisning (Hinna et al., 2011, ss. 910–914). Slagordet «leaerning by doing» er ytterst relevant. For å få til

dette må vi snakke om den proksimale utviklingssonen. Og for å komme seg gjennom disse sonene må man ha støtte fra en veileder, mediator, som gjerne er en lærer. Dette kan også være en medelev – derfor er samarbeid bra. Likevel var det en teori om datamaskinens stillasbyggende funksjon som nevnt av Healy et al. Det så ut som denne var ytterst reell. Både med tanke på at resultatene var bedre i den eksperimentelle gruppen, men også med observasjonen om at de sjeldent spurte om hjelp med det algebraiske. Datamaskinen støttet dem i å finne uttrykkene, nettopp fordi det ble naturlig for elevene.

Som vi kan se i resultatene, ser vi en kausalitet mellom den uavhengige og avhengige variabelen. Kausaliteten er riktignok ikke signifikant nok til å kunne konkludere med å si at programmering fungerer bedre enn normal undervisning når det gjelder læring av algebra. Hele poenget med hypotesetesting er at man finner en sannsynlighet for å utelukke randomisert variasjon i stikkprøvene, noe som man egentlig også utelukker ved «matching pairs» basert på en pretest. Dette gjør at man i teorien kan tåle et mye høyere signifikansnivå. Jeg har likevel valgt å ta en mer nøktern tilnærming til resultatene i denne oppgaven. Derfor kan man ikke si at resultatet generaliserbar til hele populasjonen. Hele poenget med studiet var jo å se om man kan lære algebra gjennom aktiviteter med programmering, og at med tanke på at vi i tillegg ser en økning i algebrakompetanse i forhold til vanlig undervisning, er banebrytende. Selv om synergien ikke viste seg å være så sterk i dette forskningsprosjektet som forventet. Jeg valgte å ta det noe radikale valget med å se om det var en forbedring hos i den eksperimentelle gruppen, nettopp fordi vi trenger nyere måter å lære algebra på da det ser ut til å være utfordringer med algebra her til lands.

Dette er fremtiden av matematikkundervisning, for det er ingenting som tilsier at teknologien kommer til å stoppe opp. Vi som lærere er avhengige av å henge med på skiftet. På denne måten får vi også aktive elever som skaper ting i et datamiljø. Vi unngår passiv skjermbruk der elevene får digitale lærebøker eller videoer som de ser på. Vi kan virkelig benytte datamaskinens kraft. Jeg håper dette prosjektet viser hvordan vi kan lære matematikk samtidig som at elevene skaper noe med programmering. Den store demperen til dette prosjektet er masteravhandlingens lille tilgang på ressurser og tid. Dette bør testes ut i større omfang og over lenger tid. Det kan ikke understrekes nok hvor lærerikt det kan være for elevene hvis de arbeider med større og mer meningsfylte dataprogrammer. Vi har kun med denne masteravhandlingen tatt på toppen av isfjellet for hva algoritmisk tenkning kan by på.

5.1.2 Konklusjon

For å avslutte denne masteravhandlingen konkluderer jeg med at vi kan drive med pedagogisk programmering i algebraopplæringen på lik linje som normal algebraopplæring. Dette er på bakgrunn av at det ble testet ut om pedagogisk programmering kan gi bedre effekt på læring av algebra enn normal undervisning, og at dette resulterte i en svak positiv effekt på læring. Den var ikke sterk nok til å konkludere med noe, men den viser uansett en kausalitet. Siden vi med lav sikkerhet kan si at det fungerer bedre, ikke høy nok sikkerhet til å generalisere dette, kan man med stor sikkerhet påstå at de fungerer på lik linje. Hvis vi derfor ser tilbake på forskningsspørsmålet, som jo er det som har drevet denne forskningen, kan jeg påstå at vi har delvis fått svart på forskningsspørsmålet. Pedagogisk programmering som en ressurs i algebraopplæringen vil gi delvis, men ikke signifikant, forbedring i algebraferdigheter hos norske ungdomsskoleelever i forhold til normal algebraopplæring. Dette er bra, for det betyr at man som lærer kan krysse av flere kompetansemål samtidig. Det vil være interessant å se hvordan effekten vil være på læring av algebra så vel som andre matematiske temaer over en lengre periode. Kanskje ved større forskningsprosjekter, over lengre tid, eller med elever som har høy kompetanse i programmering fra før av, kan man foreta studier som kan være med på å generalisere disse funnene. Uansett så avslutter jeg med å si at man med å lage dataprogram i matematikkundervisningen kan konkretisere det abstrakte.

6 Referanser

Arcavi, A., Drijvers, P., & Stacey, K. (2017). *The Learning and Teaching of Algebra.* New York: Routledge.

Balacheff, N. (1988). Aspects of proof in pupils' practice of school mathematics. I D. Pimm, *Mathematics, teachers and children* (ss. 216–235). Londres: Hodder & Stoughton.

Balacheff, N. (2002). Symbolic Arithmetic vs Algebra: The Core of a Didactical Dilemma. I R. Sutherland, T. Rojano, A. Bell, & R. Lins, *Perspectives on School Algebra* (ss. 249–260). New York: Kluwer Academic Publishers.

Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. *Digit Exp Math Educ 3*, 115–138.

Biesta, G. (2018). Interrupting the politics of learning. I K. Illeris, *Contemporary theories of learning: learning theorists ... in their own words* (ss. 243–257). Routledge.

Blum, W. (2012). Quality Teaching of Mathematical Modelling: What Do We Know, What Can We Do? I S. J. Cho, *The Proceedings of the 12th International Congress on Mathematical Education: Intellectual and Attitudinal Challenges* (ss. 73–99). Springer.

Boaler, J. (2016). *Mathematical Mindsets.* San Fransisco: Jossey-Bass.

Bocconi, S., Chioccariello, A., & Earo, J. (2018). The Nordic approach to introducing Computational Thinking and programming in compulsory education. *Report prepared for the Nordic@BETT2018 Steering Group*.

Bryman, A. (2012). *Social research methods.* Oxford University Press Inc.

Bråting, K., & Kilhamn, C. (2021). Exploring the intersection of algebraic and computational thinking. *Mathematical Thinking and Learning, 23:2*, ss. 170–185.

Bueie, H. (2019). *Programmering for Matematikklærere.* Universitetsforlaget.

Collis, K. (1975). *Development of formal reasoning.* University of Newcastle.

Creswell, J. W., & Creswell, J. D. (2018). *Research design - Qualitative, Quantitative, and Mixed Methods Approaches.* Sage.

Duval, R. (2006). A Cognitive Analysis of Problems of Comprehension in a Learning of Mathematics. I A. Bakker, *Educational Studies in Mathematics* (ss. 103-131). Springer.

Egara, F. O., & Nzeadibe, A. C. (2018). *EFFECT OF COMPUTER SIMULATION ON JUNIOR SECONDARY SCHOOL STUDENTS'ACHIEVEMENT IN ALGEBRA.* Hentet fra https://www.researchgate.net/publication/340132861_EFFECT_OF_COMPUTER_SIMULATION_ON_JUNIOR_SECONDARY_SCHOOL_STUDENTS%27ACHIEVEMENT_IN_ALGEBRA.

Egara, F. O., Eseadi, C., & Nzeadibe, A. C. (2022). Efect of Computer Simulation on Secondary School Student's Interest in Algebra. *Education and Information Technologies*.

Ellemor-Collins, D., & Wright, R. (2009). Structuring Numbers 1 to 20: Developing Facile Addition and Subtraction. *Mathematics Education Research Journal, Vol 21, No. 2*, ss. 50–75.

Engel, R. J., & Schutt, R. K. (2017). *The Practise of Research in Social Work.* SAGE Publications Inc.

Field, A. (2018). *Discovering statistics using IBM SPSS statistics.* SAGE Publications Ltd.

Forskningsetikkloven. (2017). *Lov om organisering av forskningsetisk arbeid*. Hentet fra https://lovdata.no/dokument/NL/lov/2017-04-28-23.

Haraldsrud, A. D., Sveinsson, H. A., & Løvold, H. H. (2020). *Programmering i skolen.* Universitetsforlaget.

Haverbeke, M. (2009). *Eloquent Javascript.* No Starch Press.

Healy, L., Pozzi, S., & Sutherland, R. (2002). Reflections on the role of the computer in the development of algebraic thinking. I R. Sutherland, T. Rojano, A. Bell, & R. Lins, *Perspectives on school algebra* (ss. 231–249). New York: Kluwer academic publishers.

Hinna, K. R., Rinvold, R. A., & Stølen Gustavsen, T. (2011). *QED 5-10 - Matematikk for grunnskolelærerutdanningen - Bind 1.* Kristiansand: Høyskoleforlaget AS.

Høgheim, S. (2020). *Masteroppgaven i GLU*. Bergen: Fagbokforlaget.

Jakobsen, H. Ø. (2012). *Derfor er algebra vanskelig*. Hentet fra forskning.no: https://www.forskning.no/barn-og-ungdom-skole-og-utdanning-matematikk/derfor-er-algebra-vanskelig/703396

Jupri, A., & Drijvers, P. (2016). Student Difficulties in Mathematizing Word Problems in Algebra. *Eurasia Journal of Mathematics, Science & Technology Education, 12(9),* 2481–2502.

Kaput, J. J. (2008). What is Algebra? What is Algebraic Reasoning? I J. J. Kaput, D. W. Carraher, & M. L. Blanton, *Algebra in the Early Grades* (ss. 5–19). New York: Lawrence Erlabum.

Kaput, J. J., Blanton, M. L., & Moreno, L. (2008). Algebra From a Symbolization Point of View. I J. J. Kaput, D. W. Carraher, & M. L. Blanton, *Algebra in the Early Grades* (ss. 19–57). New York: Lawrence Erlabaum.

Karlsen, L. (2014). *Tenk det!* Oslo: Cappelen Damm.

Kaufmann, O. T., & Stenseth, B. (2021). Programmering in mathematics education. *International Journal of Mathematical Education in Science and Technology, 52:7*, ss. 1029–1048.

Kieran, C. (2004). Algebraic thinking in the early grades: What is it? *The Mathematics Educator, Vol.8, No.1*, ss. 139–151.

Kieran, C. (2007). Learning and teaching algebra at the middle school through college levels. I F. K. Lester, Jr., *Second handbook of research on mathematics teaching and learning* (ss. 707–736). Information Age Publishing Inc.

Kieran, C. (2018). Seeking, Using, and Expressing Structure in Numbers and Numerical Operations: A Fundamental Path to Developing Early Algebraic Thinking. I C. Kieran, *Teaching and Learning Algebraic Thinking with 5- to 12-Year-Olds* (ss. 79–106). Springer.

Kilhamn, C., & Bråting, K. (2019). Algebraic thinking in the shadow og programming. *CERME 11: Thematic Working Group 03*, ss. 566–573.

Kilpatrick, J., Swafford, J., & Findell, B. (2001). *Adding It Up: Helping Children Learn Mathematics.* Washington, DC: National Academy Press.

Kongsnes, A. L., & Wallace, A. K. (2020a). *Matemagisk 8.* Oslo: Aschehoug Undervisning.

Kongsnes, A. L., & Wallace, A. K. (2020b). *Matemagisk 9.* Oslo: Aschehoug Undervisning.

Kongsnes, A. L., & Wallace, A. K. (2021). *Matemagisk 10.* Oslo: Aschehoug Undervisning.

Kvarv, S. (2021). *Vitenskapsteori - tradisjoner, posisjoner og dikusjoner.* Oslo: Novus Forlag.

Kaarstein, H., Radišić, J., Lehre, A. C., Nilsen, T., & Bergem, O. K. (2020). *TIMSS 2019. Kortrapport.* Institutt for lærerutdanning og skoleforskning, Universitetet i Oslo.

Lysø, K. O. (2015). Statistikk og kvantitativ metode. I T. S. Gustavsen, K. R. Hinn, I. C. Borge, & P. S. Andersen, *QED 5-10 - Matematikk for lærerutdanningen - Bind 2* (ss. 405–529). Cappelen Damm.

Lær Kidsa Koding. (uå). *Koding i skolen*. Hentet fra https://www.kidsakoder.no/skole/.

Mason, J. (2018). How Early Is Too Early for Thinking Algebraically. I C. Kieran, *Teaching and Learning Algebraic Thinking with 5- to 12-Year-Olds* (ss. 321–350). Springer.

Meld. St. 28. ((2015–2016)). *Fag - Fordypning - Forståelse: En fornyelse av Kunnskapsløftet.*

Misfeldt, M., & Ejsing-Duun, S. (2016). Learning mathematics through programming: An instrumental approach to potentials and pitfalls. *CERME 9 - Ninth Congress of the European Society for Research*, 2524–2530.

NESH. (2021). *Forskningsetiske retningslinjer for samfunnsvitenskap og humaniora*. Hentet fra https://www.forskningsetikk.no/retningslinjer/hum-sam/forskningsetiske-retningslinjer-for-samfunnsvitenskap-og-humaniora/.

Niss, M., Blum, W., & Galbraith, P. (2007). Introduction. I W. Blum, P. L. Galbraith, H. Henn, & M. Niss, *Modelling and Applications in Mathematics Education* (ss. 3-33). Springer.

Noss, R., & Hoyles, C. (1996). *Windows on mathematical meanings - learning cultures and computers.* Kluwer Academic Publishers.

NSD. (uå.). *Hvordan gjennomføre et prosjekt uten å behandle personopplysninger*. Hentet fra NSD: Norsk senter for forskningsdata: https://www.nsd.no/personverntjenester/oppslagsverk-for-personvern-i-forskning/hvordan-gjennomfore-et-prosjekt-uten-a-behandle-personopplysninger/

OECD. (2016). *Skills for a digital world.* Hentet fra https://www.oecd.org/els/emp/Skills-for-a-Digital-World.pdf

Opplæringsloven. (2008). *Lov om grunnskolen og videregående opplæring*. Hentet fra https://lovdata.no/dokument/NL/lov/1998-07-17-61/

Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education 128*, 356–376.

Pramesti, T. I., & Retnawati, H. (2019). Difficulties in learning algebra: An analysis of students' errors. *Journal of Pyhsics: Conference Series 1320 012061*.

Radford, L. (2010). Algebraic thinking from a cultural semiotic perspective. *Mathematics Education, 12:1*, 1–19.

Radford, L. (2018). Theoretical Perspectives for Developing Early Algebraic Thinking. I C. Kieran, *Teaching and Learning Algebraic Thinking With 5- to 12-Year-Olds* (ss. 3–27). Springer.

Richardson, M., Isaacs, T., Barnes, I., Swensson, C., Wilkinson, D., & Golding, J. (2020). *TIMS 2019: National report for England.* Department for Education.

Rystedt, E., Kilhamn, C., & Helenius, O. (2016). What's there in an n? Investigating contextual resources in small group discussions concerning an algebraic expression. *Nordic Studies in Education, 21 (1)*, 5–26.

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Educational and Information Technologies 22*, ss. 469–495.

Sevik, K. (2016). *Programmering i skolen.* Senter for IKT i urdanningen.

Sfard, A., & Linchevski, L. (1994). The gains and the pitfalls of reification - The case of algebra. *Educational Studies in Mathematics, 26*, ss. 191–228.

Skemp, R. R. (1976). Relational Understanding and Instrumental Understanding. *Mathematics Teaching, 77*, ss. 20–26.

Smith, J. P., & Thompson, P. W. (2008). Quantitative Reasoning and the Development of Algebraic Reasoning. I J. J. Kaput, D. W. Carraher, & M. L. Blanton, *Algebra in the Early Grades* (ss. 95–133). New York: Lawrence Erlbaum.

Utdanningsdirektoratet. (2019a). *Dybdelæring*. Hentet fra https://www.udir.no/laring-og-trivsel/dybdelaring/.

Utdanningsdirektoratet. (2019b). *Algoritmisk tenkning*. Hentet fra https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/.

Utdanningsdirektoratet. (2020a). *Matematikk 1-10 (MAT01-05) Fagrelevans og sentrale verdier*. Hentet fra https://www.udir.no/lk20/mat01-05/om-faget/fagets-relevans-og-verdier?lang=nno.

Utdanningsdirektoratet. (2020b). *Matematikk 1–10 (MAT01-05) Kompetansemål og vurdering*. Hentet fra https://www.udir.no/lk20/mat01-05/kompetansemaal-og-vurdering/kv15?lang=nob

Utdanningsdirektoratet. (2020c). *Matematikk 1-10 (MAT01-05) Kjerneelementer*. Hentet fra https://www.udir.no/lk20/mat01-05/om-faget/kjerneelementer.

Utdanningsdirektoratet. (2020d). *Matematikk 1–10 (MAT01-05) Tverrfaglige temaer*. Hentet fra https://www.udir.no/lk20/mat01-05/om-faget/tverrfaglige-temaer?lang=nob.

Utdanningsdirektoratet. (2020e). *Overordnet del - Verdier og prinsipper for grunnopplæringen*. Hentet fra https://www.udir.no/lk20/overordnet-del?kode=mat01-05&lang=nob.

Utdanningsdirektoratet. (2020f). *Hva er nytt i matematikk?* Hentet fra https://www.udir.no/laring-og-trivsel/lareplanverket/fagspesifikk-stotte/nytt-i-fagene/hva-er-nytt-i-matematikk/.

Williamson, B. (2016). Political computational thinking: policy networks, digital governance and 'learning to code'. *Critical Policy Studies, vol 10*, 39–58.

Wing, J. M. (2017). Computational thinking's influence on research and education. *Italian Journal of Education Technology, 25(2),* 7–14.

7 Vedlegg

7.1 Vedlegg 1: Eksempeloppgave fra eksamen etter ny læreplan av utdanningsdirektoratet med hjelpemidler – oppgave 7

Hentet fra: https://www.udir.no/eksamen-og-prover/eksamen/eksempeloppgaver/eksempeloppgaver-i-matematikk-grunnskolen/

Oppgave 7

Siden 2018 har pant på plastflasker vært 2 kr for små flasker og 3 kr for store flasker.

Ali har pantet flasker for 109 kr.

Til sammen pantet han 51 flasker.

Sett opp, forklar og løs et likningssett som vil gi svar på hvor mange små og store flasker Ali pantet.

7.2 Vedlegg 2: Eksempeloppgave fra eksamen etter ny læreplan av utdanningsdirektoratet med hjelpemidler – oppgave 9

Hentet fra: https://www.udir.no/eksamen-og-prover/eksamen/eksempeloppgaver/eksempeloppgaver-i-matematikk-grunnskolen/

Oppgave 9

Fakta
Påfølgende heltall er heltall som kommer rett etter hverandre. For eksempel er 3, 4 og 5 tre påfølgende heltall.

Bruk påstandene ovenfor som et utgangspunkt for å vise din kompetanse innen abstraksjon og generalisering.

7.3 Vedlegg 3: Eksempeloppgave fra eksamen etter ny læreplan av utdanningsdirektoratet med hjelpemidler – oppgave 10

Hentet fra: https://www.udir.no/eksamen-og-prover/eksamen/eksempeloppgaver/eksempeloppgaver-i-matematikk-grunnskolen/

Oppgave 10

Anne er 15 år, og ønsker å ta førerkort for moped.
Hun skal kjøpe moped når hun blir 16 år.
Hun planlegger å selge den når hun blir 18 år.

Følgende er obligatorisk opplæring når du skal ta førerkort for moped:

Grunnkurs moped – 3 timer	1000,-
Trinnvurdering trinn 2	700,-
Sikkerhetskurs trafikk – 4 timer	2040,-
Trinnvurdering trinn 3	700,-
Sikkerhetskurs vei – 4 timer	2040,-

Samlet pris: All obligatorisk opplæring + 3 kjøretimer: kr. 8800,-

Gebyr førerkort moped:

Gebyr teoriprøve	660,-
Gebyr utstedelse av førerkort	310,-
Fakturagebyr	65,-

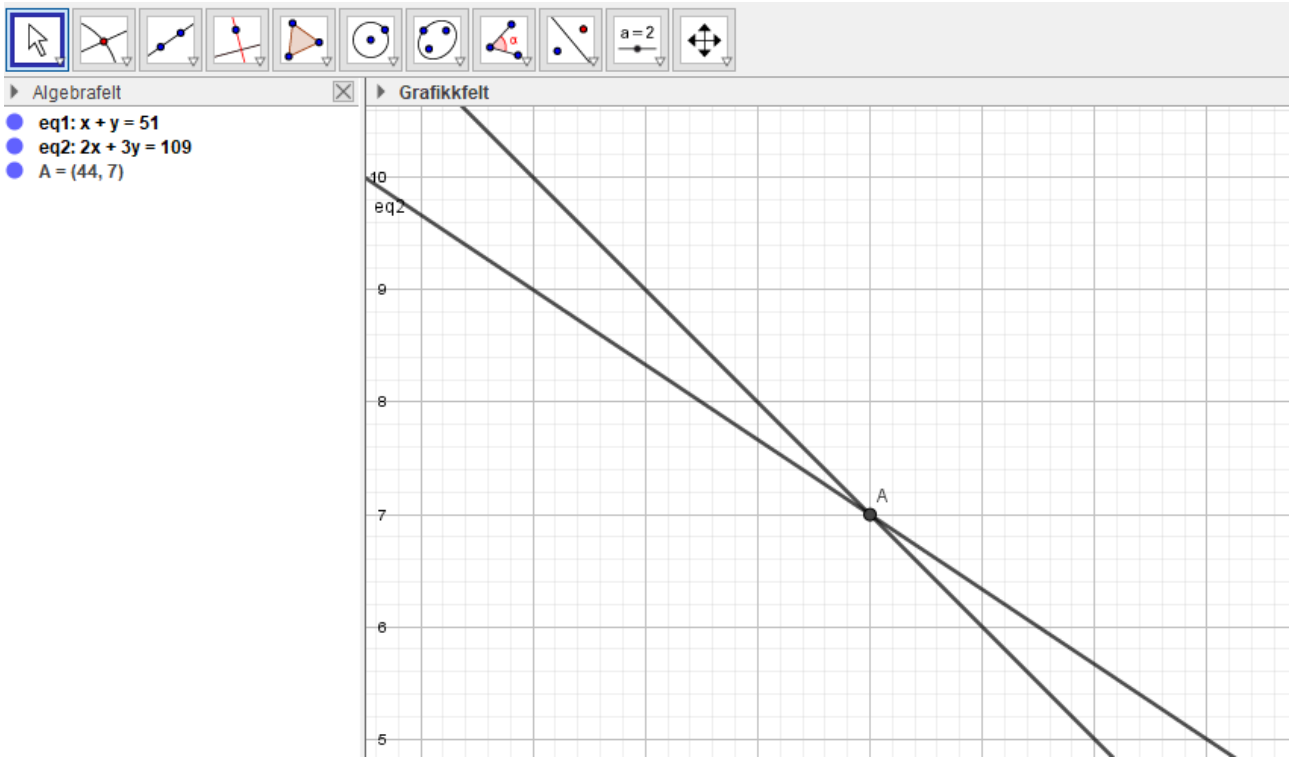
♡ Legg til favoritt

Peugeot Speedfight 4 Pure

Pris
16 000 kr

Bruk informasjonen ovenfor til å vise din kompetanse innen modellering og anvendelse.

7.4 Vedlegg 4: Løsning av praktisk situasjon med likning med to ukjente. Dette er et løsningsforslag av oppgaven i vedlegg 1



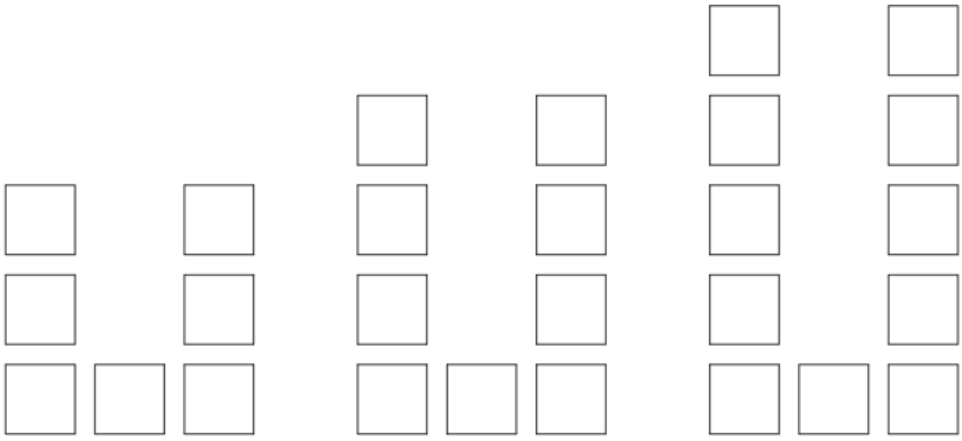
7.5 Vedlegg 5: Matemagisk sitt forslag til terminprøve for 8.trinn etter den nye læreplanen – oppgave 10

STANDARD TERMINPRØVE 8. TRINN – VÅREN 2021

NAVN: _____ KLASSE:_____

OPPGAVE 10

Ulrik lager bokstaven **U** i ulike størrelser ved å bruke brikker.



Figur nr. 1 Figur nr. 2 Figur nr. 3

a Fargelegg brikkene over slik at det kommer fram hvordan figurene vokser.

b Beskriv med ord hvordan mønstret utvikler seg fra en figur til den neste.

Løs oppgave **b** her.

c Lag et algebraisk uttrykk for antall brikker i figur nr. *n*.

Løs oppgave **c** her.

7.6 Vedlegg 6: Enkel sannsynlighetsfordeling med kast av to terninger. Eksempel fra egen undervisning i matematikk på 9.trinn

```
1  import random
2
3  resultat = [0] * 11
4
5  for n in range(1, 10001):
6      terning1 = random.randint(1, 6)
7      terning2 = random.randint(1, 6)
8      resultat[terning1 + terning2 - 2] += 1
9
10 for n in range(len(resultat)):
11     print("Verdi:", n+2, ". Frekvens: ", resultat[n])
12
```

```
Verdi: 2 . Frekvens:  263
Verdi: 3 . Frekvens:  583
Verdi: 4 . Frekvens:  852
Verdi: 5 . Frekvens:  1110
Verdi: 6 . Frekvens:  1389
Verdi: 7 . Frekvens:  1606
Verdi: 8 . Frekvens:  1432
Verdi: 9 . Frekvens:  1113
Verdi: 10 . Frekvens:  864
Verdi: 11 . Frekvens:  514
Verdi: 12 . Frekvens:  274
```

7.7 Vedlegg 7: Hjørnespillet

Et eksempel fra egen undervisning i programmering valgfag på bruk av pygame og koordinater for å få et rektangel til å bevege seg på skjermen. Programmet avslutter når rektangelet treffer hjørnet.

```
import pygame
import random

window = pygame.display.set_mode((1000, 600))
pygame.display.set_caption("Hjørnespillet!")
clock = pygame.time.Clock()
fps = 500
run = True

x = random.randint(0, 500 - 200) * 2
y = random.randint(0, 300 - 100) * 2
x_endring = 1
y_endring = 1

vegg_treff = 0

while run:
    clock.tick(fps)
    window.fill((1, 45, 66))

    if x > 1000 - 200 or x < 0:
        x_endring *= -1
        vegg_treff += 1
        print(vegg_treff)
    if y > 600 - 100 or y < 0:
        y_endring *= -1
        vegg_treff += 1
        print(vegg_treff)


    x += x_endring
    y += y_endring

    if x == 800 and y == 500:
        run = False
    elif x == 0 and y == 0:
        run = False
    elif x == 800 and y == 0:
        run = False
    elif x == 0 and y == 500:
        run = False

    rektangel = pygame.Rect(x, y, 200, 100)
    pygame.draw.rect(window, (255, 255, 255), rektangel)

    pygame.display.update()

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            run = False

print("Total: ", vegg_treff)
pygame.quit()

```

7.8 Vedlegg 8: Samtykkeerklæring

Du er invitert til å delta på et forskningsprosjekt i matematikkfaget!

Formål og hensikt

I forbindelse med min masteroppgave skal jeg gjennomføre et forskningsprosjekt i algebra med elevene i klasse [redacted] Formålet er å undersøke to ulike undervisningsmetoders effekt på læring. Målet er å finne nyere og mer hensiktsmessige undervisningsmetoder der elevene arbeider med oppgaver som er mer relevante for dem og deres deltakelse i samfunnet. Undervisningsmetodene er tenkt å gi algebra mer hensikt og mening.

Ifølge retningslinjene gitt av Den nasjonale forskningsetiske komité, må jeg informere om hensikten med forskningsprosjektet, og jeg må hente inn samtykke fra deltakerne. Det **er frivillig** å være med på prosjektet, og du kan når som helst trekke deg. Jeg må også innhente samtykke fra foresatte når det forskes på barn under 16 år – personopplysningsloven artikkel 8.

Gjennomføring

Elevene vil først gjennomføre en før-test der de løser noen algebraoppgaver. Dette er for å kartlegge algebraferdighetene. Elevene vil deretter bli delt inn i to like grupper, der gruppene videre får to ulike undervisningsopplegg. Avslutningsvis tar elevene en etter-test slik at forbedringer i algebrakompetanse kan måles.

Personvern

Full anonymitet vil bli ivaretatt, og ingen personopplysninger vil bli behandlet elektronisk. Elevenes tester vil bli identifisert med koder, og være en del av slutningsstatistikken. Før- og etter-testen vil makuleres så fort slutningsstatistikken er utført.

Med vennlig hilsen

Anders Holm – Lærer ved [redacted] og masterstudent ved universitetet i Sørøst-Norge.

Samtykkeerklæring

Jeg samtykker til at _____ (barnets navn) kan delta på dette forskningsprosjektet.

Dato: _____

_____ _____

Elev **Foresatt**

7.9 Vedlegg 9: Pretest

Algebra før-test. Deltakernummer:

Tid: 45 min

Oppgave 1

Hvorfor bruker vi ofte bokstaver/variabler i matematikk?

_____

_____

_____

Oppgave 2

a) Lag et algebraisk uttrykk for hvor mye Anders tjener på å selge **t** t-skjorter. Han selger t-skjorter for 150 kr per stk.

b) Hver t-skjorte koster 100 kr å lage, og det koster også 1500 kr per dag for å leie t-skjorte-maskinen. Lag et algebraisk uttrykk for hvor mye det koster å lage **t** t-skjorter per dag. Bruk også uttrykket for å regne ut hvor mye det koster å lage 50 t-skjorter på én dag.

c) Dette algebraiske uttrykket viser hva Anders sitter igjen med per dag han lager t-skjorter: $\mathbf{50}\boldsymbol{t} - \mathbf{1500}$. Forklarer uttrykket, og bruk uttrykket til å forklare hvor mange t-skjorter Anders må lage for at han ikke skal tape penger!

Oppgave 3

f_1 f_2 f_3

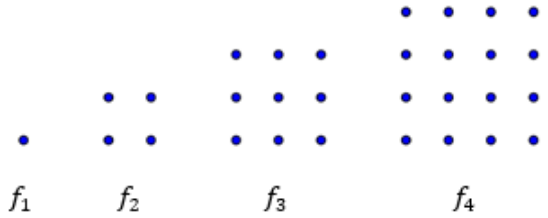
a) Lag et algebraisk uttrykk for å finne antall klosser i den **n**-te figuren i tallfølgen. (**PS**: Når du vet hva figuren øker med, kan du lage figurnummer 0 for å finne konstanten)

b) Bruk det algebraiske uttrykket som du lagde for å finne hvor mange klosser det er på figur-nummer 100

Oppgave 4

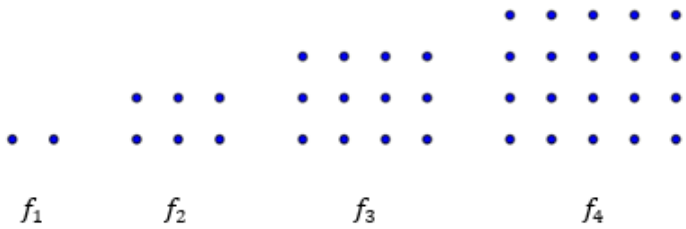
a) Kryss av for riktig algebraisk uttrykk for figurtallene nedenfor. Flere kan være riktig.

- ☐ $2n$
- ☐ $n * n$
- ☐ n^2



b) Kryss av for riktig algebraisk uttrykk for figurtallene nedenfor. Flere kan være riktig.

- ☐ $n * n + 1$
- ☐ $n(n + 1)$
- ☐ $n^2 + n$
- ☐ $n * 1 + n$



Oppgave 5

Vi vet at partall kan skrives på formen: $\boldsymbol{2 * n = 2n}$. Fordi alle partall har 2 som faktor.

Hvis vi legger sammen (adderer) to partall, kan vi skrive det på generell form slik:

$2n + 2n =$

Kan du bruke dette til å forklare og bevise at to partall lagt sammen også blir et partall?

Forklar:

7.10 Vedlegg 10: Posttest

Algebra etter-test. Deltakernummer:

Tid: 45 min

Oppgave 1

Hvorfor bruker vi ofte bokstaver/variabler i matematikk?

_____

_____

_____

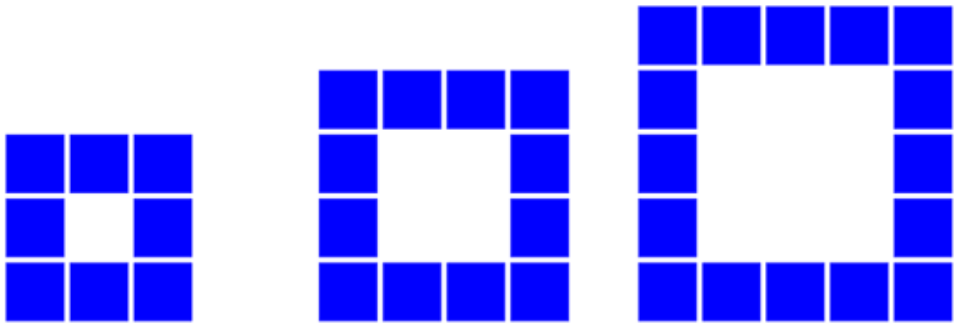
Oppgave 2

a) Lag et algebraisk uttrykk for hvor mye Anders tjener på å selge **b** bananer. Han selger bananer for 5 kr per stk.

b) Anders pleier å kjøre til Oslo for å kjøpe bananer på en båt for 2 kr per stk. Han tar buss til/fra Oslo for 500kr. Lag et algebraisk uttrykk for hva det koster for en tur til Oslo per **b** banan han kjøper.

c) Anders trenger et algebraisk uttrykk for hva han tjener på dette. Lag et algebraisk uttrykk for hva han TJENER per tur til Oslo for å hente bananer. (*HINT: Du vet at han tjener* $5 - 2 = 3\ kr$ *per banan. Men husk at turen til Oslo koster litt*)

Oppgave 3



f_1 f_2 f_3

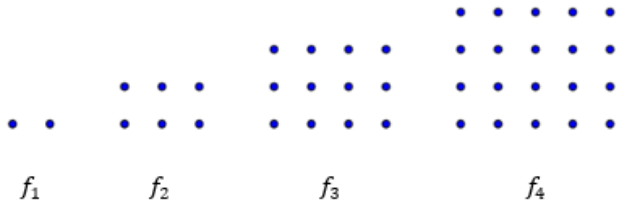
a) Lag et algebraisk uttrykk for å finne antall klosser i den **n**-te figuren i tallfølgen. (**PS**: Når du vet hva figuren øker med, kan du lage figurnummer 0 for å finne konstanten)

b) Bruk det algebraiske uttrykket som du lagde for å finne hvor mange klosser det er på figur-nummer 10

Oppgave 4

a) Kryss av for riktig algebraisk uttrykk for figurtallene nedenfor. Flere kan være riktig.

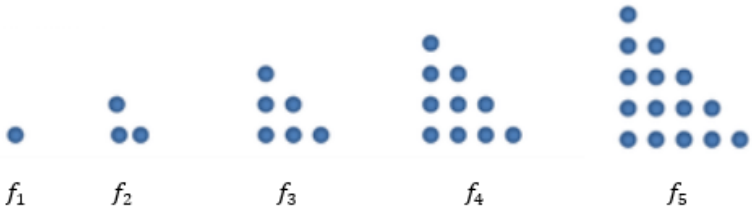
- ☐ $n * n + 1$
- ☐ $n(n + 1)$
- ☐ $n^2 + n$
- ☐ $n * 1 + n$

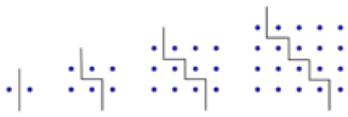


Hint: $side * side$

b) Kryss av for riktig algebraisk uttrykk for figurtallene nedenfor. Flere kan være riktig.

- ☐ $\frac{n^2+n}{2}$
- ☐ $n(n + 1)$
- ☐ $\frac{n*n}{2}$
- ☐ $\frac{1}{2} * n(n + 1)$



Hint: 

Oppgave 5

Vi vet at partall kan skrives på formen: $2 * n = \mathbf{2n}$. Fordi alle partall har 2 som faktor.

Vi vet at alle oddetall kan skrives på formen: $2 * n - 1 = \mathbf{2n - 1}$

Vi kan legge sammen to oddetall slik: $\mathbf{2n - 1 + 2n - 1}$

Kan du bruke dette til å forklare og bevise at to oddetall lagt sammen blir et partall? Forklar:

Hint: Sjekk om svaret ditt kan deles på to! Da er det et partall!

7.11 Vedlegg 11: Pedagogisk plan kontrollgruppe

Pedagogisk plan kontrollgruppe

Tid	*Aktivitet*
10 min	Oppstart: Si god morgen, dele opp i to grupper, og plan for dagen Elevene går i læringspar
15 min	Felles gjennomgang av variabler og algebraiske uttrykk Tankekart + «konsensus» i plenum på hva det er og hvorfor vi lager dem
65 min	Arbeid med realistiske saker i forhold til algebraiske uttrykk. Elevene får et oppgaveark som de skal arbeide med.
15 min	Pause
15 min	Felles gjennomgang av figurtall og tallfølger
45 min	Arbeid med figurtall og tallfølger fra oppgavearket

7.12 Vedlegg 12: Pedagogisk plan eksperimentell gruppe

Pedagogisk plan eksperimentell gruppe

Tid	*Aktivitet*
10 min	Oppstart: Si god morgen, plan for dagen og klargjøring av datamaskiner. Alle elevene får datamus slik at arbeidet på datamaskinene går smertefritt.
15 min	Felles gjennomgang av variabler og print()-funksjonen og hvordan vi kan bruke dette til å lage eller regne ut algebraiske uttrykk. Også et eksempel med input()-funksjonen. I tillegg til å jobbe med programmering, jobber de også med algebra.
20 min	Gruppeaktiviteter der elevene bruker variabler og print()-funksjonen relatert til utforming av algebraiske uttrykk. (**Oppgaver på arket**)
45 min	**Oppgaver på arket i grupper**: Lage lønningskalkulator, drivstoffkalkulator og valutakalkulator
15 min	Pause - friminutt
15 min	Forklare løkker og bruken av dem med eksempler fra oppgavene ovenfor
45 min	Finne formler på tallfølger og figurtall ved å jobbe med løkker. Elevene får mulighet til å manipulere algebraiske uttrykk og direkte se resultatet av dette. De skal også få bruke dette til å lage matematiske bevis.

7.13 Vedlegg 13: Oppgaveark kontrollgruppe

Algebra

Oppgave 1

Sett et kryss ved det algebraiske uttrykket som ikke passer inn. Skriv også en begrunnelse under på hvorfor det ikke passer inn.

a)

$a+b$	ab	$b+a$

b)

ab	$b+a$	ba

c)

$20b$	$20a$	$20+a$

d)

$20b+20a$	$20(a+b)$	$20ab$

e)

$3n-2$	$\frac{1}{2}*6n-4$	$\frac{6n-4}{2}$

f)

$a*a$	$2a$	a^2

g)

$(-a)*(-a)$	a^2	$-a^2$

h)

$4*4+2$	$2x+2$	10	$4+4+2$

i)

$2b-2a$	$a-b$	$b-a$

Oppgave 2 – Valutakalkulator

På google kan man du søke opp ulike valutakurser. Finn ut hvor mange norske kroner det koster for én dollar eller én euro. PS: Søk på «dollar to nok» eller «euro to nok».

a) Lag et algebraisk uttrykk for hvor mange kroner det koster per **d** dollar eller **e** euro.

b) Bruk uttrykket til å regne ut hvor mye det koster å kjøpe en vare til 250 dollar eller 250 euro.

c) Dere har jobbet en del med likninger. Tenk deg at du har tjent 5000kr på en sommerjobb. Du lurer på hvor mange dollar eller euro det blir. Sett opp en likning med inspirasjon fra uttrykket du lagde i oppgave a, og løs den.

d) Anders har startet en valutabutikk. Han vil tjene penger på at folk veksler mellom ulike valutaer. Han bestemmer seg for at han kan la folk veksle fra norske kroner til euro eller dollar. Han bestemmer seg for at hvis han skal tjene noe på dette, så må han ta 150kr ekstra per transaksjon. Lag et algebraisk uttrykk på hvor mange kroner det koster per **d** dollar eller **e** euro i butikken til Anders.

Oppgave 3

Bilen til Anders bruker 0.75l bensin per mil.

a) Lag et algebraisk uttrykk for hvor mye bensin han bruker per **m** mil han kjører.

b) Bruk uttrykket til å regne ut hvor mye bensin han bruker tur/retur Ringshaug-Oslo. Ca 10 mil per vei.

Det koster 20kr per liter for bensin.

c) Lag et algebraisk uttrykk for hvor mye det koster per **m** mil for Anders å kjøre bil.

Siden Anders har en gammel bil, så stiger den i verdi. Anders anslår at den vil stige med 8000kr per år. Likevel må han betale 5000kr per år i forsikring og årsavgift.

d) Lag et algebraisk uttrykk for **x** verdien til bilen etter **å** år.

Oppgave 4 – Matematiske bevis (Oppgave fra førtesten)

Vi vet at partall kan skrives på formen: $\boldsymbol{2 * n = 2n}$. Fordi alle partall har 2 som faktor.

Hvis vi legger sammen (adderer) to partall, kan vi skrive det på generell form slik:

$2n + 2n =$

Kan du bruke dette til å forklare og bevise at to partall lagt sammen også blir et partall?

Forklar:

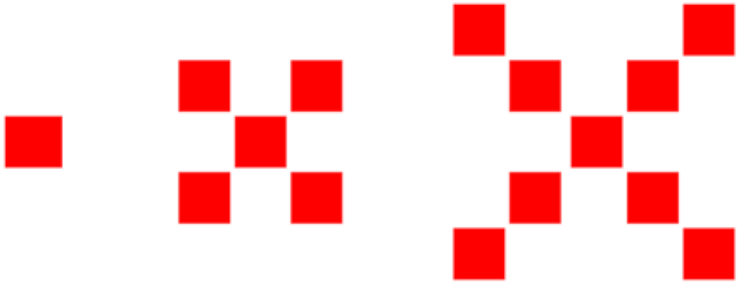
Oppgave 5 – Figurtall (Fra førtesten)

f_1 f_2 f_3

a) Lag et algebraisk uttrykk for å finne antall klosser i den **n**-te figuren i tallfølgen. (**PS:** Når du vet hva figuren øker med, kan du lage figurnummer 0 for å finne konstanten)

b) Bruk det algebraiske uttrykket som du lagde for å finne hvor mange klosser det er på figur-nummer 100

Oppgave 6 – Figurtall

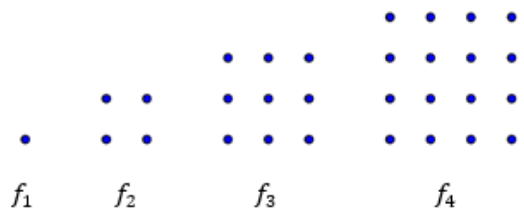


f_1 f_2 f_3

a) Lag et algebraisk uttrykk for å finne antall klosser i den **n**-te figuren i tallfølgen. (**PS**: Når du vet hva figuren øker med, kan du lage figurnummer 0 for å finne konstanten)

b) Bruk det algebraiske uttrykket som du lagde for å finne hvor mange klosser det er på figur-nummer 10

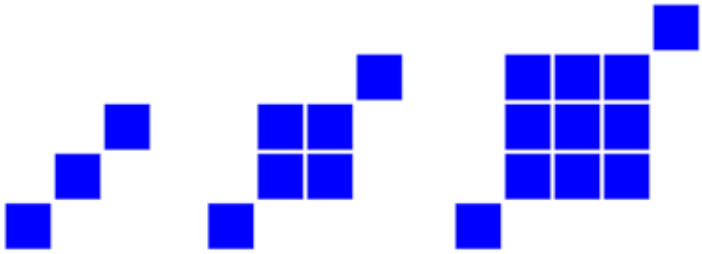
Oppgave 7 – Figurtall (Fra førtesten)



a) Lag et algebraisk uttrykk for å finne antall klosser i den **n**-te figuren i tallfølgen.

b) Bruk det algebraiske uttrykket som du lagde for å finne hvor mange klosser det er på figur-nummer 10

Oppgave 8 – Figurtall



f_1 f_2 f_3

a) Lag et algebraisk uttrykk for å finne antall klosser i den **n**-te figuren i tallfølgen.

b) Bruk det algebraiske uttrykket som du lagde for å finne hvor mange klosser det er på figur-nummer 10

7.14 Vedlegg 14: Oppgaveark eksperimentell gruppe

Programmering og algebra

Oppgave 1 – variabler og print()

Hvis du er usikker, **prøv gjerne i Thonny**!

a)

```
1 a = 25
2 print(2*a)
```

Hva vil dette dataprogrammet skrive ut? Algebraisk uttrykk: $\mathbf{2a}$

_____

b)

```
1 a = 10
2 print(3*a-5)
```

Hva vil dette dataprogrammet skrive ut? Algebraisk uttrykk: $\mathbf{3a - 5}$

_____

c)

```
1 a = 10
2 b = 5
3 print(10*a-5*b)
```

Hva vil dette dataprogrammet skrive ut? Algebraisk uttrykk: $\mathbf{10a - 5b}$

_____

d)

```
1 a =
2 print(a-5)
```

```
>>> %Run oppgaver.py
20
```

Dette programmet skriver ut 20. Hvilken verdi har variabelen? Fyll inn.

e)

```
1 x = -5
2 print(x**2)
```

Hva vil dette dataprogrammet skrive ut? Algebraisk uttrykk: x^2

_____

f)

```
1 timelønn = 200
2 timer = 30
3 total = timelønn * timer
4 print("Lønnen til Anders:", total)
```

Hva vil dette dataprogrammet skrive ut?

_____

Oppgave 2 – Anders sine eventyr

a) Dyreparken – lag et program

Anders har fått seg jobb som dyrepasser på Kristiansand dyrepark, og har sluttet på jobben sin på Ringshaug. Han tjener 250kr i timen. Han ønsker seg et dataprogram som kan regne ut lønnen hans!

Man kan få brukere til å skrive inn verdi til en variabel ved å bruke input()-funksjonen. Vi må også sette den i en int()-funksjon for å gjøre om det vi taster inn til heltall.

```
1 t = int(input("Skriv inn antall timer: "))
2 print(      )
```

Verdien som brukeren skriver inn, i dette tilfellet 25, blir verdien til variabelen!

```
>>> %Run oppgaver.py
Skriv inn antall timer: 25
6250
```

Skriv ferdig programmet i Thonny, og prøv det ut med forskjellige verdier. Hva blir det algebraiske uttrykket for **t** timer som skal stå i print()-funksjonen?

_____

b) Bilkjøring – lag et program

Bilen til Anders bruker 0,75 liter bensin per mil. Anders ønsker at noen kan lage et dataprogram som regner ut hvor mye bensin han bruker over flere mil, spesielt siden det blir mye pendling til Kristiansand. Lag et dataprogram slik som ovenfor som gjør nettopp det, og prøv ut med forskjellige verdier. Hva blir det algebraiske uttrykket for **m** mil som skal stå inne i print()-funksjonen?

_____

c) Pris på bilkjøring – lag et program

På bensinstasjonen koster det 20kr per liter for bensin (husk at bilen bruker 0,75 liter per mil). Anders ønsker et program som regner ut hvor mye penger han må betale for bilturene sine. Lag et program som tar inn en verdi fra brukeren, og som skriver ut hvor mye han må betale. Hva blir det algebraiske uttrykket per **m** mil?

---

d) Anders skal til utlandet

Anders skal til Texas for å kikke på hester. Men han er ikke så god i matematikk, og trenger derfor et dataprogram som kan gjøre om fra dollar til norske kroner (valutakalkulator). Søk opp «dollar to nok» på google. Da finner du ut hvor mye man må betale i norske kroner per dollar. **I tillegg til dette, så tar banken 100 kr ekstra hver gang man endrer valuta**. Lag dataprogrammet. Hva blir det algebraiske uttrykket per **d** dollar?

---

e) God mode (For de som er trygge på programmering)

Denne oppgaven introduserer enda flere elementer: vilkår. Den gjør følgende: **Hvis** valuta er dollar, så gjør den noe, **eller hvis** valuta er euro, så gjør den noe. **Tolk programmet og skriv inn hva som kan stå inne i print()-funksjonen**. Husk at det også koster 100 kr å veksle.

```
1 valuta = input("Hvilken valuta trenger du? (dollar, euro): ").lower()
2
3 if valuta == "dollar":
4     antall = float(input("Hvor mange dollar skal du ha?: "))
5     print(                              )
6 elif valuta == "euro":
7     antall = float(input("Hvor mange euro skal du ha?: "))
8     print(                              )
```

Oppgave 3 - Figurtall og bevis

a)

Vi vet at alle partall kan skrives på formen: $\boldsymbol{2 * n = 2n}$. Nå har du mulighet til å prøve ut dette! Lag en løkke, og skriv inn det algebraiske uttrykket, og se resultatet!

```
1 for n in range(1, 101):
2     print(    )
```

Hva forteller dette deg om uttrykket $2n$?

_____

_____

_____

b)

Forsøk å endre på det algebraiske uttrykket ditt slik at det bare skriver ut **oddetall**. Hvordan blir det algebraiske uttrykket da?

_____

c)

Forsøk å legge sammen (adder) to generelle partall inne i print()-funksjonen. Se hva som skjer. Hva forteller resultatet deg?

_____

_____

_____

d) (Fra førtesten)

Her kan du prøve ut og manipulere uttrykket ditt i en for-løkke for å «sjekke» at det passer med figurene.

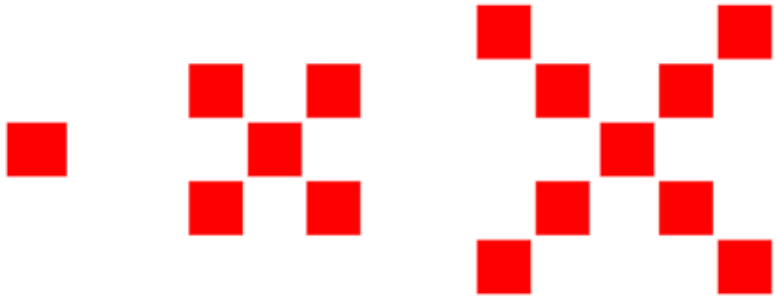
f_1 f_2 f_3

a) Lag et algebraisk uttrykk for å finne antall klosser i den **n**-te figuren i tallfølgen. (**PS**: Når du vet hva figuren øker med, kan du lage figurnummer 0 for å finne konstanten)

b) Bruk det algebraiske uttrykket som du lagde for å finne hvor mange klosser det er på figur nummer 100

e)

Her kan du prøve ut og manipulere uttrykket ditt i en for-løkke for å «sjekke» at det passer med figurene.



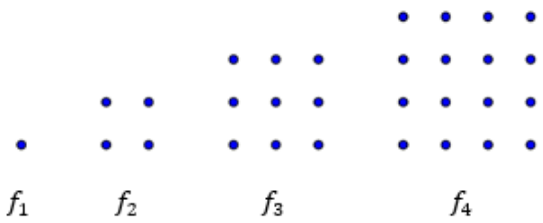
f_1 f_2 f_3

a) Lag et algebraisk uttrykk for å finne antall klosser i den **n**-te figuren i tallfølgen. (**PS:** Når du vet hva figuren øker med, kan du lage figurnummer 0 for å finne konstanten)

b) Bruk det algebraiske uttrykket som du lagde for å finne hvor mange klosser det er på figur nummer 10

f)

Her kan du prøve ut og manipulere uttrykket ditt i en for-løkke for å «sjekke» at det passer med figurene.

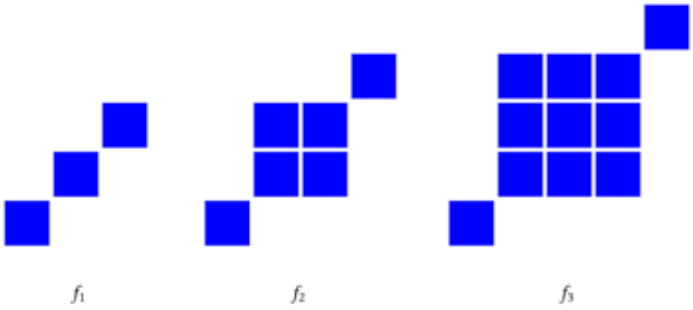


a) Lag et algebraisk uttrykk for å finne antall klosser i den **n**-te figuren i tallfølgen.

b) Bruk det algebraiske uttrykket som du lagde for å finne hvor mange klosser det er på figur nummer 10

g)

Her kan du prøve ut og manipulere uttrykket ditt i en for-løkke for å «sjekke» at det passer med figurene.



a) Lag et algebraisk uttrykk for å finne antall klosser i den **n**-te figuren i tallfølgen.

b) Bruk det algebraiske uttrykket som du lagde for å finne hvor mange klosser det er på figur nummer 10

7.15 Vedlegg 15: Økonomiprogram som viser hensiktsmessig bruk av for-løkke eller while-løkke

```python
import matplotlib.pyplot as p

sparebeløp = 0 # Beløpet vi starter med
innskudd = 25000 # Det vi setter inn hvert år
rente = 3.5 # Renten som sparebeløpet vårt skal øke med hvert år
vekstfaktor = 1 + rente/100 # Lager vekstfaktor
år = 10 # Antall år vi skal spare

x = [0] # Lager en liste med verdier som skal være x-aksen. Dette vil være antall år
y = [sparebeløp] # Lager en liste som skal være tilsvarende y-verdi for hvert år (x).

for n in range(1, år +1): # En løkke som gjentar seg selv over antall år vi skal spare
    sparebeløp = sparebeløp + innskudd # Legger til innskuddet
    sparebeløp = sparebeløp * vekstfaktor # Legger til rentene på slutten av året
    print("Etter", n, "år. Har jeg:", round(sparebeløp, 2))
    x.append(n) # Legger til x-verdier til lista vår, 1.. 2.. 3.. 4 osv
    y.append(sparebeløp) # Legger til tilsvarende y-verdi, altså sparebeløpet

p.plot(x, y) # Lager en graf av utviklingen på sparekontoen vår over et x antall år
p.show() # Viser kordinatsystemet med grafen på skjermen :)
```

```python
sparebeløp = 300000
innskudd = 25000
konto = 0
rente = 3.5
vekstfaktor = rente/100 + 1
år = 0

while konto < sparebeløp:
    konto = konto + innskudd
    konto = konto * vekstfaktor
    år = år + 1
    print("Etter", år, "år. Har jeg", round(konto, 2), "kr på BSU.")

```