



Skriying av hovedfagsoppgaven

Vår 2001

FORORD

Dette heftet er ment å gi deg hjelp på veien i forbindelse med skriving av hovedfagsoppgaven. Heftet har ennå mangler og svakheter. Dette vil vi kontinuerlig forsøke å arbeide med i fremtidige utgaver. Forslag til rettelser, endringer og forbedringer mottas med takk.

Bidragstere:

Ilan Eini

Jo Tryti (tegninger)

Stein Krogdahl

Anne-Margit Arntzen

Oddvar Søråsen

Stein Michael Storleer

Mikkel Orheim

Øyvind Endresen

Sara Selmark

Trond Alvern

Ingebjørg Theiss

Katinka Grønli

Håvard Holstad

Marina Maneas Bakkum

Pål André Larsen

Tone Tellevik Dahl

INNHOOLD

1. OPPGAVESKRIVNINGEN	3
1.1 Hvordan skaffe seg hovedfagsoppgave?	4
1.2 Hovedfagsoppgave med flere studenter	5
1.3 Hva er egentlig en hovedfagsoppgave?	5
1.4 Oppstartning av arbeidet med oppgaven	6
1.5 Litteraturstudier	7
1.6 Skrivning	7
1.7 Et opplegg for å starte arbeidet	8
1.8 Om skrivning av selve oppgaveteksten	8
1.9 Hvem skal man tenke seg at man skriver for?	9
1.10 Oppgaven skal være mest mulig sekvensielt lesbar	10
1.11 La problemstillingen være drivkraften i fremstillingen	10
1.12 Oppgaven må være mer enn innledning og detaljer	10
1.13 Man må ha en holdning til all kvasiengelsken	11
1.14 Vær likefrem, ærlig og beskjeden	12
1.15 Bruk av “jeg”, problemet med “han” og “hun”	12
1.16 Samarbeidet mellom veileder og student	12
2. LITTERATUR.....	14
2.1 Hvordan lage en god litteraturfortegnelse til hovedfagsoppgaven og avhandlingen	14
2.1.1 Artikkel fra tidsskrift/konferanse/bok	14
2.1.2 Bok/rapport/avhandling	15
2.2 Eksempler	15
2.3 Litteratur om å skrive hovedfagsoppgave. (listet på en annen måte)	15

1. Oppgaveskrivningen

Når en student starter med hovedfag og skal til med en hovedfagsoppgave, oppstår mange spørsmål. Det kan f.eks. være: Hvordan skaffe seg veileder? Hva forlanges av en hovedfagsoppgave? Hvordan skal man best komme i gang? Hvilken stil skal den skrives i? Svar på disse og liknende spørsmål vil studentene skaffe seg gjennom diverse formelle og uformelle kanaler, først og fremst gjennom samtaler med andre studenter og med veilederen, men også gjennom studieveileder og studiehandboken.

Det finnes en oversikt over veileder med deres interesseområde og fagfelt tilgjengelig i ekspedisjonen. Veilederne presenterer hovedfagsoppgaver på introduksjonskurset IN-IFHS, samtidig som de legges ut på web og en del henges opp på oppslagstavlen for hovedfag. Det siste gjelder spesielt oppgaver fra eksterne veiledere.

En del av det som sies her, men kanskje spesielt ting som sies i litteratur det henvises til, kan sikkert også ha interesse for folk som skal i gang med å publisere sine første arbeider. Dette vil antageligvis først og fremst gjelde folk i rekrutteringsstilling.

Det finnes mange bøker, artikler og notater som behandler ting i samme område som vi tar opp her. Emnet kan f.eks. være hvordan man skriver den gode artikkel, råd om å drive forskning generelt, og endel er også skrevet spesielt omkring det å utarbeide en hovedfagsoppgave. Det som finnes av den siste typen er gjerne, liksom dette, knyttet til et spesielt fagfelt, og det som finnes er stort sett knyttet til fag der et feltarbeide eller en undersøkelse inngår som en naturlig del.

På biblioteket finnes en del. Følgende bøker kan anbefales som en begynnelse: Rapportskrivning (Pedersen 1975), “Vitenskapelig forfatterskap” (Høeg, 1968), “Vejledning i rapportskrivning” (Bohlbro & Crossland, 1980), “Writing scientific papers in english” (Connor & Woodfor, 1975), “Effective Writing” (Turk & Kirkman, 1982), “How to

Write & Publish a Scientific Paper” (Day,1988). For å holde dette skriftet noenlunde kort, vil vi ikke uten videre gjenta ting man kan finne bra beskrevet i disse bøkene. Problemstillingene er da bare kort skissert her, sammen med en referanse. Om tingene er spesielt viktige, eller de er spesielt knyttet til en oppgave innen informatikk, er de nøye behandlet her.

Det er altså nødvendig å ta for seg en eller flere av disse bøkene for å få med seg det som er tenkt dekket med dette skriv, og vi vil på det varmeste anbefale at folk blar litt i flere av dem. Det er vel nr 1 og 2 som mest direkte går på vårt emne her, men også de andre inneholder mye av interesse. Når man leser slik litteratur, er det ofte de generelle innledningskapitlene som har interesse, sammen med kapitler om mer spesielle ting som f.eks. stil og språk, litteratursøking eller liknende. En titt i innholdsfortegnelsen gir gjerne fort nøkkelen til det som er verdt å titte på. Problemstillinger i tråd med de som taes opp her blir også behandlet i enkelte av kursene som gis ved instituttet, spesielt i kurset IN 160.

1.1 Hvordan skaffe seg hovedfagsoppgave?

IN-IFHS er obligatorisk for alle nye hovedfagsstudenter. Formålet med kurset er å få deg fortere i gang med oppgaven, samt å gjøre deg bedre i stand til å gjennomføre hovedfagsoppgaven. Intensjonen er at du skal **velge hovedfagsoppgave** i løpet av første del av kurset, og at obligatorisk oppgave velges i samråd med hovedfagsveileder. Oppgaven kan med fordel innpasses som et innledende arbeid med hovedfagsoppgaven. Hovedfagsveileder står ansvarlig for veiledning og bedømmelse av rapporten. De kursdeltakerne som ikke finner hovedfagsoppgave i løpet av kurset vil bli tilbudt en standard obligatorisk oppgave.

På kurset vil en eller flere lærere informere om prosjekter som er i gang eller under planlegging, og om mulige hovedfagsoppgaver forøvrig. Du kan også skaffe deg en oppgave på andre måter. På hjemmesiden til Ifi finnes det linker til de forskjellige studieretningene. Disse sidene inneholder mer eller mindre oppdaterte oversikter over tilgjengelige veiledere/hovedfagsoppgaver ved Ifi. Flere veiledere har også hjemmesider på WWW der de legger ut informasjon om oppgaver o.l. På oppslagstavlen i 2. etasje pleier det å henge noen ledige oppgaver. Har du egne interesser og id er til en oppgave, så snakk med en lærer om disse. Det er ikke utenkelig at dere sammen kan formulere et godt grunnlag for en hovedfagsoppgave.

Som en inspirasjonskilde for både studenter og potensielle veiledere skal vi imidlertid liste opp en del typer av oppgaver som kan være aktuelle. Denne listen vil forhåpentligvis bli utvidet etter hvert.

1. Man sitter med noen “nye” idéer om et felt, og vil studere og konkretisere disse idéene. Man utformer og implementerer et system som anvender idéene på en mer eller mindre fullkommen og fullstendig måte. Oppgaveteksten beskriver dette systemet, men består først og fremst av en diskusjon av idéene ut fra den erfaring man har skaffet seg ved programmeringen og eventuell bruk av programmet.
2. Man sitter med noen “nye” idéer, og konkretiserer og diskuterer disse ut fra rent teoretiske betraktninger.
3. Man sitter med en implementasjonsoppgave, der selve implementasjonen forventes å føre til så mange prinsipielle problemer at det å diskutere disse og den nærmeste omegn er en kraftig nok utfordring. Man utfører en implementasjon, og diskuterer alle sider ved denne.
4. Man skriver et program som skal brukes i en eller annen undersøkelse, men også selve undersøkelsen og vurderingen av denne er en vesentlig del av oppgaven. Eksempel: Lag analyseverktøy for en eller annen egenskap ved programmer, og gjør en undersøkelse av et antall programmer.
5. Man undersøker hvordan et gitt programsystem faktisk blir mottatt og brukt, og sammenlikner med de vyene som designerne hadde.
6. Man tar for seg et problemområde der det er skrevet en del artikler, men som ennå ikke er blitt et etablert felt.
7. Man presenterer problemområdet, summerer opp hver av artiklene, diskuterer terminologi og begrepsbruk, xkommer selv med forslag, og forsøker å peke ut hvor de sentrale gjenstående problemer ligger.

I tillegg kan man tenke seg oppgaver med historisk eller pedagogisk dreining. Om man har spesiell bakgrunn og/eller interesse her, kan man forhøre seg om dette kunne være en mulighet.

1.2 Hovedfagsoppgave med flere studenter

Det at flere studenter arrangerer seg slik at deres hovedfagsoppgaver henger mer eller mindre tett sammen har både ulemper og fordeler, og det er nok viktig at man tenker over disse før man begir seg inn på en slik løsning.

Hovedfordelen er at man får et fellesskap å arbeide i og et forum å lufte sine tanker i. I motsetning til fellesskapet med veilederen har dette den fordel at man har det samme faglige nivå, og at alle har oppgaven som hovedbeskjeftigelse. Et slikt fellesskap kan være spesielt fruktbart om man f.eks. skal sette seg inn i en maskin eller et operativsystem, eller om man må gjennom en masse artikler. Da kan man fordele oppgavene, og støtte hverandre med forståelsen.

Ulempene er imidlertid at man på forskjellige måter gjør seg avhengige av hverandre, og all erfaring viser at det å få ferdig en hovedfagsoppgave kan utvikle seg svært annerledes enn man planla, for en eller flere i en slik gruppe. Om en da er avhengig av f.eks. et program som en annen i gruppa skulle skrive, og denne enten blir forsinket eller vil lage en annen vri på oppgaven, er man straks oppe i store problemer. Så om man går inn for oppgave omkring samme problemstillingen må man altså tenke nøye over at koblingene ikke blir sterkere enn at man føler seg trygg på å slippe dramatiske problemer. Noen problemer av denne typen må man nok i alle tilfelle regne med, men dette er det jo også svært mye å lære av når det gjelder kunsten å samarbeide.

1.3 Hva er egentlig en hovedfagsoppgave?

Det som først og fremst karakteriserer en hovedfagsoppgave er at den er to ting på en gang. På den ene side er det en eksamensbesvarelse, og på den annen side er det en vitenskapelig rapport. Dermed kommer det inn to sett med kriterier man må forholde seg til mens man arbeider med oppgaven, og det er nok viktig at ingen av disse får lov å dominere fullstendig om man skal komme seg vel gjennom det hele på rimelig tid.

Det er denne blandingen av kriterier som gjør arbeidet med en hovedfagsoppgave til noe helt spesielt. Følgende poenger kan vel sies å komme ut nettopp av denne blandingen:

1. Selv om det opplagt står mye igjen å gjøre, og det fra et faglig synspunkt synes unaturlig å slutte, må man på ett eller annet tidspunkt si at "nå er det nok!", og foreta en avrunding av det man har. Man får så ta sin eksamen, og eventuelt arbeide videre med stoffet siden. Man skal i denne forbindelse også passe seg for den onde sirkelen der man nå synes at man har holdt på så lenge at man for skams skyld må levere noe enda litt bedre, osv, osv.
2. Man må i en hovedfagsoppgave gjøre mer fullstendig rede for alle momentene som inngår, og gå mer i dybden, enn man ville f.eks. i en tidsskriftsartikkel. Dette ligger i tradisjonen, men er vel slik nettopp fordi man derved hvertfall en gang får erfare hvordan det er å gjennomarbeide en ting nokså fullstendig, og også derved får vist at man behersker dette.
3. Man skal ikke tro at hovedfagsoppgaven som sådan vil bli lest rundt om i verden, og dermed la seg binde av en slik forestilling. Man må ikke la seg friste eller true til å skrive på engelsk med mindre man føler seg meget fri på dette språket, det blir bare en ekstra belastning. Om noe av det man kommer frem til har internasjonal interesse, vil man i alle tilfelle skrive det sammen i en mer konsentrert form etter eksamen, og dette vil det da være rimelig å gjøre på engelsk.
4. Selv om vi lenger opp karakteriserte en hovedfagsoppgave som en "eksamensbesvarelse", betyr det ikke at den skal ha noen likhet med en besvarelse til en korttidsksamen. Da forutsetter man at sensor har den fulle oversikt, og det viktige er å si så mye at sensor føler seg trygg på at kandidaten har tenkt riktig. En hovedfagsoppgave skal derimot ha form av en selvstendig rapport som står på egne bein.

I tillegg til disse poengene, er selvfølgelig hensikten med hovedfagsoppgaven at man skal lære noe! For det første skal man bli kjent med et avgrenset fagfelt helt frem til "forskningsfronten", for det andre skal man bli kjent med hva som kreves for å gå i dybden på et felt, og for det tredje skal man lære seg å legge frem sine resultater i en skikkelig rapport. De to siste punktene kan vel sammenfattes i at man skal lære seg "forskningens håndverk".

Som en del av mange hovedfagsoppgaver innen informatikk vil det inngå å skrive et program for ett eller annet formål. Dette må imidlertid ikke forlede noen til å tro at selve programteksten, hvor godt kommentert den enn er, er å regne som en fullverdig del av selve oppgaveteksten. Et slikt program vil som regel bli lagt med som et appendiks til oppgaven, gjerne i et eget hefte.

Selve oppgaveteksten må ha form av en diskusjon av en problemstilling, som selvfølgelig vil være knyttet til programmet på en eller annen måte, men som ofte går mye videre enn det som direkte har med programmet å gjøre. Det er altså ikke nok å skrive et stort, men forøvrig standard, program, og så la oppgaven være en diskusjon bare av de valg og vurderinger som ble gjort under implementasjonen. Et slikt opplegg vil bare være brukbart om det er noe spesielt nytt, utfordrende og interessant ved selve implementasjonen.

I denne forbindelse skal man også legge merke til at man ikke er forpliktet til å forsvare det programmet man har skrevet i ett og alt. Programmet ble jo i stor grad skrevet for å få “fingertupp-følelse” med det problemet man skal diskutere, og vitsen er nettopp at man etter å ha skrevet programmet kan uttale seg sikrere om hva slags valg man bør gjøre i forbindelse med denne type programmer. At de valgene man gjorde i første omgang ikke var helt vellykkede er da nærmest en selvfølge.

Under programmeringen vil man som regel, spesielt henimot slutten når man får mer oversikt, få en mengde “aha-opplevelser” angående hvordan man egentlig burde ha grepet saken an fra begynnelsen. Om man ikke kunne diskutere slike idéer fritt, uten også å forandre programmet tilsvarende, ville man måtte velge mellom å lyve om hva man egentlig mente, eller å aldri bli ferdig, og ingen av disse alternativene virker rimelige.

Hvilket forhold selve programmeringsarbeidet skal ha til resten av oppgaven vil man som regel måtte avgjøre etterhvert som det hele utvikler seg. Det fine med en slik oppgave er imidlertid at man har en nokså konkret ende å begynne i, og at arbeid fra denne enden i stor grad vil øke ens forståelse av det problemet man siden skal skrive om.

Det kan selvfølgelig også være annet praktisk arbeid forbundet med en hovedfagsoppgave, som f.eks. en spørreundersøkelse angående programmeringsvaner eller effektivitetstester av noen programmer. I forbindelse med slike oppgaver har man en tilsvarende fordel. Denne fordelene har man ikke om man setter i gang med en oppgave av ren teoretisk karakter. Det er derfor her lettere å bli sittende lenge med en følelse av at man ikke er kommet skikkelig i gang, og at man må skaffe seg uendelig mye mer innsikt før man kan sette et eneste fornuftig ord ned på papiret. Om det inngår noe praktisk i oppgaven skal jo dette også beskrives, og det gjør at man i dette tilfellet også lettere kommer i gang med selve skrivingen.

I tillegg til dette kommer at selve oppgaveteksten ved en teoretisk oppgave gjerne blir mer “gjennomsiktig”, i den forstand at den ikke er fylt opp med beskrivelser av det arbeidet som er gjort. Det blir dermed lettere å se om de overlegninger som gjøres er fornuftige eller ikke.

For ikke å komme skjevt ut er det derfor i forbindelse med rå teoretiske oppgaver viktig at man har en bevisst holdning til dette problemet. Man kan f.eks. sette seg delmål i forbindelse med litteraturstudier, eller ved å skulle beskrive hovedproblemstillingen i et eller flere notater. Det vil bli sagt mer om det å komme i gang med arbeidet i et eget avsnitt lenger ned.

Hva slags oppgave man ender opp med styres vel i stor grad av tilfeldigheter, men den enkeltes legning og interesser vil vel ofte også gjøre at man tiltrekkes av den ene eller annen type. Det har nok også skjedd mer enn en gang at noe som var tenkt som en implementasjonsoppgave endte opp som en ren teoretisk oppgave.

1.4 Oppstartning av arbeidet med oppgaven

Selv om det i stor grad avhenger av oppgavetypen, veilederen og personlige forutsetninger, vil nok mange oppleve det å komme i gang med oppgaven som problematisk. Som før nevnt kan dette spesielt gjøre seg gjeldende i forbindelse med teoretiske oppgaver der man ikke har noe nærmere klart definert delmål enn det å “tenke ut noe genialt, og skrive det ned”. Fordi man lenge vil føle man mangler svært mye av den bakgrunnen som skal til for å komme frem med noe “genialt”, kan det hele da virke nokså håpløst. I neste kapittel er det listet opp en del punkter som kan være til hjelp for å komme i gang med oppgaven. Det er fint om punktene gjennomgås i samarbeid med veileder og på egenhånd.

Arbeidet med oppgaven vil normalt inneholde følgende elementer, der andre og tredje punktet altså kan mangle. Litteraturstudier

- Forberede programmering eller undersøkelse
- Utføre programmering eller undersøkelse
- Bearbeide materiale, vurdere og trekke konklusjoner
- Skrive selve oppgaveteksten

Legg imidlertid merke til at denne listen ikke betyr at tingene skal utføres etter hverandre i denne rekkefølgen. Det er snarere å anbefale at man begynner med de forskjellige delene så tidlig som mulig, og tar dem opp igjen periodevis under hele arbeidet.

1.5 Litteraturstudier

Om dette kan man lese i Pedersen (1975), side 2 og i Høeg (1986), side 12. Legg her spesielt merke til anbefalingen om at man ikke skal lese for mye før man begynner å gjøre noe selv. Det er viktig å utnytte skapelysten og impulsiviteten før den forsvinner i andres mer eller mindre geniale forslag og metoder. Man skal heller ta litteraturstudiene opp igjen etter en stund. Da vil den bakgrunnen man har i å ha møtt problemer på egen hånd også gjøre at man leser med en fornyet forståelse og interesse.

Dette er jo forholdsvis greit om man har noe praktisk å gå i gang med. Har man ikke det må man sørge for å skaffe seg et aktivt forhold til stoffet på en annen måte. Det blir da svært viktig å skrive små oppsummeringer av hver ting man leser, f.eks. laget over lesten:

Hva ville forfatteren gjerne ha frem?

Hva var nytt for meg her?

Hvordan har dette betydning for oppgaven?

Hva var det jeg ikke forstod ved det jeg leste?

Om man vil diskutere f.eks. en artikkel med veilederen kan det være en fordel å be veilederen lese gjennom ens notater først. Man må imidlertid da ikke binde seg i skrivingen ut fra dette, ved f.eks. å være redd for å dumme seg ut. Om man vil komme noen vei med stoffet må man regne med å stille svært mange "dumme" spørsmål. Vær heller litt provoserende mot deg selv og andre på dette punkt.

Det samme gjelder det å komme med "håpløse" forslag. Ikke det at man skal fortsette å forsvare slike om man får rasjonelle motforstillinger, men den reaksjonen man får er det som regel svært mye å lære av. De forslag man stiller er jo som regel av typen: "Ut fra det jeg nå har forstått, ville jeg si at så og så er en bra løsning". De innvendinger man får til dette vil jo da nettopp være ny og sentral kunnskap. Ved å komme med en serie slike "håpløse" forslag kan man få mye informasjon ut av folk som de neppe selv ville klart å formulere på en direkte måte.

Legg også merke til at dette å komme med forslag i øst og vest representerer en slags "brain storming", altså en måte å generere et forråd av forslag på, hvorav kanskje noen vil lede til noe som har noe for seg. Det å tørre å komme med egne forslag kan også motvirke den knugende følelsen man kan få av at andre vet så mye mer, og som kan virke litt lammende. Man må imidlertid passe seg for å knytte noen prestisje til slike forslag, som jo i stor grad bare er "prøveballonger".

Forøvrig er det å gå gjennom litteratur noe hver må gjøre på sin egen måte. Noen kan feie gjennom masse litteratur, og etter hvert klare å sette sammen de detaljene man får med seg fra hver artikkel til et helhetlig bilde, mens andre må arbeide mer med hver artikkel for at de i det hele tatt skal kunne nyttiggjøre seg det som står der. Det å lage et kortkartotek (eller en tilsvarende fil??) over det man leser, der man også skriver ned sine kommentarer til artiklene, er nok imidlertid en god ide for de fleste, og nesten en forutsetning når man skal begynne å skrive litteraturlisten. Så, noter ned referansen med en gang på alle artikler du leser.

1.6 Skrivning

Det å skrive selve oppgaveteksten er jo i en viss forstand det siste man gjør, men det betyr ikke at man ikke skal begynne med det tidlig. Man må fremfor alt ikke tro at man på slutten, uten forarbeider, kan skrive den endelige oppgaveteksten rett ned. Man må i alle tilfelle regne med å skrive minst en "rå-kladd", som så kan være grunnlag for skrivingen av den endelige utgaven.

Vi skal komme nærmere inn på selve skrivingen senere, men bare fremheve her at det å begynne å skissere kapittelinndeling og innholdet av sentrale kapitler ikke bare vil lette den senere skrivingen, men også vil ha en strukturerende virkning på tankegangen omkring problemene. Man bør derfor alvorlig vurdere, på et svært tidlig tidspunkt, å skissere en kapittelinndeling av oppgaven, og å fylle ut disse med en "rå-rå-kladd" ut fra en foreløpig forståelse av problemene. Dette behøver selvfølgelig ikke være mer enn noen punkter og endel mer eller mindre

fullstendige setninger. Man vil nok i alle tilfelle få dette igjen ved den senere skriving, selv om man da foretar en fullstendig omkalfatring og revisjon. Dessuten får man altså fordelen av den strukturerende virkning dette har på forståelsen.

Når man er ferdig med oppgaven og skal lage en forside så anbefaler vi deg å bruke programmet 'hfagforside'.

1.7 Et opplegg for å starte arbeidet

Se også neste punkt.

Om det i oppgaven f.eks. inngår en programmeringsjobb, kan et brukbart opplegg for å starte arbeidet være følgende:

1. Skaff deg noenlunde oversikt over det du skal jobbe med ved et aktivt forhold til veileder og studium av en del litteratur.
2. Gå gjennom problematikken i forbindelse med det programmet du skal skrive på kryss og tvers, og skisser utforming og implementasjon.
3. Foreslå en kapittelinndeling av oppgaven, og skriv en "rå-rå-kladd" utfra denne. Forsøk å la den systematiseringen du da foretar få tilbakevirkning på ditt syn på stoffet.
4. Gjør etterhvert en "endelig" programutforming, og sett i gang å programmere.

Videre blir det altså en blanding av programmering, (re)vurdering, litteraturstudier og skriving. Man bør vel kanskje ikke la seg avspore for mye i programmeringen når man først er i gang, og hver gang man får en ny forståelse av hvordan man "egentlig" burde ha lagt saken an, må man altså vurdere om man skal la dette få følger for programmet, eller om man bare skal ta det med i diskusjonen til slutt.

1.8 Om skriving av selve oppgaveteksten

Det er selve utformingen av artikler og rapporter det er skrevet mest om i den type litteratur vi har referert til i kap. 1, og vi kan bare anbefale å bla og lese i disse bøkene, spesielt i Pedersen (1975) og Høeg (1986). Her er mye interessant, selv om noe av det kanskje kan virke rigid, og være litt overdrevet i forhold til behovet omkring en hovedfagsoppgave. Det eneste vi ønsker å si noe om er kapittelinnstillingen. Den kan f.eks. være slik:

1. Innledning
2. Historikk, Bakgrunn
3. Problemstilling, Beskrivelse
4. Materiale og Metoder
5. Analyse, Diskusjon
6. Konklusjon
7. Referanser
8. Appendix

Vi skal nedenfor trekke frem en del punkter som vi videre mener er sentrale, eller som har spesielt med informatikk å gjøre.

Som tidligere nevnt er det mange gode grunner til tidlig å begynne å skissere både deler av og helheten i oppgaven, og det viktige i denne sammenheng er altså at det vil forbedre kvaliteten av de senere utkast, og gjøre skrivingen av dem lettere.

Man skal her legge merke til at enhver skriftlig formulering omkring oppgavens tema vil være med på å gjøre de senere manuskripter så gode som mulig. Dette må ikke forstås slik at man må gjemme og siden finne frem alle notater man produserer, men bare det en gang å ha "tvunget" en problemstilling ned i form av en sekvensiell tekst, og vurdert resultatet, vil gjøre at det helt ubevisst blir lettere å få det godt neste gang.

Man bør imidlertid være forsiktig med å kaste notater under arbeidet. Det koster svært lite å samle opp alle gamle notater i en usortert bunke i stedet for å kaste dem, og det kan være utrolig irriterende å ikke finne igjen en gammel formulering man husker man fikk litt sving på.

De fleste slike notater vil altså forberede skriveingen omkring de enkelte problemstillingene. Ved siden av dette bør man tidlig lage skisser av helheten i oppgaveteksten, med kapittelinnndelingen, og så mye innmat i hvert kapittel som man til enhver tid finner fornuftig. Man må imidlertid passe seg for å regne med at de tidlige versjonene blir særlig like den endelige.

På et eller annet tidspunkt synes man så man har så at mye oversikt og materiale at man vil sette sammen en fullstendig oppgavetekst. Det er da på tide å lage en "rå-kladd" til hele oppgaven, og mens man gjør det bør man tenke litt annerledes enn når man senere formulerer den endelige versjonen. Man bør nå tenke lite på detaljformuleringer, men i stedet forsøke å gjøre arbeidet mest mulig sammenhengende for å få flyt i fremstillingen.

Spesielt innledninger og avsnitt kan være vonde å få til på dette stadiet, og da kan det være greit å sette opp punktvis de tingene man tenker å nevne, og så gå rett løs på innmaten. Dermed mister man ikke flyten i det hele, og innledninger må man likevel nesten alltid skrive om igjen til slutt.

Legg imidlertid merke til at de tankene man tvinges inn i, i forbindelse med utformingen av innledninger etc. fører med seg en vurdering av det man driver med i en større ramme enn man vanligvis ser det, og ethvert incitament i denne retning skal man ta vare på!

Med snev av manglende moralske skrupler tar vi også med følgende forslag:

Om man synes det er vanskelig å få pennen på gli under skriveing av råkladden, har enkelte med hell forsøkt å senke selvkritikken noe v.h.a. en pils. (Mer enn en anbefales ikke!) Da får man hvertfall noe på papiret, og kan siden arbeide videre ut fra dette.

Man kan her selvfølgelig også putte inn i teksten notater man har gjort tidligere, eventuelt ved å tilpasse dem litt. Det kan imidlertid være lurt først å forsøke å formulere ting om igjen ut fra den stilen man nå bruker. Det er slående hvor mye bedre en ting kan bli når man reformulerer den fritt, enn når man forsøker å fikse på et gammelt manuskript.

1.9 Hvem skal man tenke seg at man skriver for?

Mange spørsmål omkring hva man skal ta med, hvor nøye ting må forklares, hvilken stil ting skal skrives i osv. kan lettere besvares om man har en "typisk leser" å tenke på. I forbindelse med hovedfagsoppgaver er dette selvfølgelig i en viss forstand sensor, men ved siden av at sensor gjerne selv vil forstå det som står i oppgaven, blir den også vurdert ut fra en slags tenkt vedtatt leser.

Det fine er imidlertid at den "typiske" sensor ofte også tilfredsstiller dette kravet, nemlig at det skal være en som har rimelig god peiling på informatikk generelt, som har endel peiling på det feltet det dreier seg om (f.eks. kompilatorer eller verifikasjon av programmer), men som ikke er kjent med detaljene i det problemområdet oppgaven omhandler. Man kan nok også tenke på en medstudent som har omtrent samme fagkrets som en selv, og som f.eks. også har oppgave omkring kompilatorer, men med et annet språk og en annen problemstilling enn det en selv har.

Det er nok mer vanlig å forestille seg en leser som vet for mye, enn omvendt. Grunnen til dette er vel at man selv er så dypt nede i detaljene at det tar litt tid å komme opp igjen. Dessuten er det lettere å skrive for en detaljkyndig leser, og mange studenter har nok også noe overdrevne forestillinger om hva en fag-informatiker kan utover sitt spesialområde. Det å arbeide seg ut av dette er derfor en av de ting man spesielt må ha i tankene når man reviderer manuskriptet fra versjon til versjon, ikke minst når man arbeider med innledninger og ansatser til kapitlene.

En tenkemåte man her eksplisitt må passe seg for er den man har når man skriver besvarelser på korttidsksamener. Da antar man at sensor ser tvers gjennom hele problemet, og man skriver bare nok til å overbevise sensor om at man også har forstått problemet. En hovedfagsoppgave skal derimot være en rapport som står på egne bein, og man må ha klart for seg at denne rapporten gjerne blir det eneste leseren får vite om disse problemstillingene, mens man selv vet en masse mer. Man bør tenke seg inn i leserens sted og se om man får et dekkende og klart bilde av det man vil ha sagt.

Opgaven skal altså gjennom innledninger og oversikter føre den tenkte leser fra det antatte kunnskapsnivå frem til de problemstillinger som oppgaven behandler, og på dette grunnlaget skal så ting beskrives og diskuteres. Som hovedregel skal slike oversikter stå på egne bein, men av og til ville dette føre for vidt. Man må da i stedet gi et antall

henvisninger, og i oppgaven bare gi en ufullstendig oppsummering. Hvordan dette skal gjøres må diskuteres i hvert enkelt tilfelle med veilederen.

1.10 Oppgaven skal være mest mulig sekvensielt lesbar

I likhet med de fleste andre tekster skal man tilstrebe at oppgaven kan leses sekvensielt tvers gjennom. Dette betyr selvsagt ikke at ting skal forklares så omstendelig at man aldri behøver å lese en forklaring to ganger for å forstå den, men at selve logikken i oppgaven er slik at man til enhver tid bare bygger på det som er sagt tidligere.

Det å få en ting sekvensielt lesbart kan sjelden oppnås ved å produsere det sekvensielt, og tro man er ferdig. Man vil stadig komme på ting man skulle ha forklart tidligere, og man vil stadig måtte skrive om innledninger for å få dem til å sette scenen riktig for det som blir sagt etterpå. Svært mye av arbeidet fra en versjon til den neste vil ha forbindelse med dette.

Idealet om sekvensiell lesbarhet kan sjelden gjennomføres fullstendig. Man skal imidlertid ikke bryte det før man har bevisstgjort seg grunnen til at det er riktig å gjøre det nettopp her. Man bør da også opplyse om dette for leseren, og angi hvordan det er rimelig å legge an lesingen (f.eks. foreslå at man skimleser et gitt avsnitt, for så å lese det nøyer etter at et annet kapittel er lest). Tilsvarende bør man si fra de stedene man går inn på detaljer eller forklaringer som ikke har betydning for den videre logikk i oppgaven.



1.11 La problemstillingen være drivkraften i fremstillingen

Når man skal skrive et kapittel og ikke sitter med meget begrunnede meninger om hvordan det hele skal struktureres, kan det være en god ide å stille seg spørsmål som: Hva er hovedproblemstillingen i kapittelet? Hvilke hovedgrupper av argumenter finnes? Hva er de iboende motsetninger i dette stoffet? Slike motsetninger finnes nesten alltid, da det er disse som gjør problemet verdt å diskutere! Det kan f.eks. være tid kontra plass, menneskevennlighet kontra effektivitet eller to forskjellige interessegrupper som står mot hverandre.

Om man ikke finner tilfredsstillende svar på slike problemer bør man sikkert bruke litt tid på å lete etter svarene, eller eventuelt konkludere med at det er noe feil f.eks. med kapittelinnstillingen. Har man svarene har man imidlertid også et godt utgangspunkt for skrivingen. Man beskriver først hovedmotsetningene ut fra det som er sagt tidligere, og går så videre med løsningsalternativer som så i tur og orden skaper nye problemstillinger som behandles tilsvarende, og dette gjøres i beste pedagogiske stil.

1.12 Oppgaven må være mer enn innledning og detaljer

Etter første utkast vil man ofte sitte med et produkt som i alt for stor grad bare er en løs innledning, og så en masse detaljer. Utfordringen blir da å få fylt ut det mellomliggende, som altså skal være en diskusjon som setter ting inn i sammenheng der detaljene er spesialtilfeller og som samtidig utsier noe av interesse.

Dette er selvfølgelig først og fremst en faglig utfordring, der valg av fruktbare begreper står sentralt. Problemet henger imidlertid også sammen med det å arbeide seg ut av sin snevre verden av detaljer, og å forestille seg en leser som ser det hele mer utenfra. Dette er igjen en av de ting man må ha i tankene når man reviderer utkastene til oppgaven. Bruk figurer der dette sier mer enn tekst. Ofte vil skjemaer, grafiske fremstillinger eller andre figurer få ting mye klarere frem enn man kan med tekst, og da skal man så absolutt benytte figurer. Det er også en avveksling for leseren å studere figurer innimellom lesingen, og det hele blir dermed mere leservennlig.

Det er mye som kan sies om bruk av figurer, og det anbefales å lese i (f.eks. den refererte) litteratur om dette. Det viktigste er at enhver figur skal ha en ledsagende tekst som sammen med figuren danner en selvstendig enhet som kan forstås ut fra generell kunnskap om det teksten dreier seg om. Her skal også slike ting som enheter på akser og alle slags forkortelser være forklart.

Man skal altså ikke behøve å lete seg frem i den alminnelige teksten for å forstå figuren, men det skal likevel være en henvisning til figuren og en oppsummering av den betydning den har i fremstillingen der dette hører naturlig hjemme i teksten.

Mange hovedfagsoppgaver i informatikk blir skrevet på tekstbehandlingssystemer, og de færreste av disse kan lagre og tegne ut figurer utover det å bruke de vanlige tegnene i spesielle kombinasjoner. For å ha alt på ett sted er det derfor fristende å benytte seg av slike figurer, men de er sjelden vakre å se på, og er som regel store i forhold til den informasjon de inneholder. Man bør derfor motstå denne fristelsen, og heller klistre tegninger utført på gamlemåten på originalen når den er ferdig.

En annen sak er det selvfølgelig om man har et godt grafisk system, og det er vel å håpe at dette etter hvert blir mer og mer vanlig.

1.13 Man må ha en holdning til all kvasiengelsken

Et problem som ofte dukker opp i hovedfagsoppgaver i informatikk er hvordan man skal forholde seg til all den engelske terminologien som databehandlere nokså kritikkløst tar i bruk, og som de i dagligtale bruker litt med norske og litt med engelske endelser. Det kan dreie seg om tradisjonelle ord som "array" (og "arrayen") eller "byte", men det blir gjerne enda verre om man f.eks. har med en spesiell maskin å gjøre, og er avhengig av å bruke det mangfold av begreper leverandøren benytter i sin dokumentasjon. Det kan f.eks. være navn på alle slags måter å adressere på, eller navn på maskin-instruksjonene.

Tilsvarende problemer får man om man bygger på artikler der verden er gruppert på ymse måter, og der hver av gruppene har mer eller mindre anerkjente engelske betegnelser. Dette kunne f.eks. typisk oppstå om man arbeider med nettverksproblemer.

Det finnes vel ingen entydig vei ut av dette (om man da ikke benytter problemet som det avgjørende argument for å skrive oppgaven på engelsk). Man må imidlertid i det minste ha en bevisst holdning til dette problemet, og det vil nesten alltid være på sin plass å si noen ord i begynnelsen av oppgaven et sted om hvordan man har valgt å takle det.

En løsning er selvfølgelig å oversette begrepene til norsk, og i denne forbindelse bør man konsultere Norsk Dataordbok (Heggstad, 1981). I den grad man velger denne metoden kan det være riktig å gi en liste over de norske og de tilsvarende engelske betegnelser man har brukt. Om man danner sine egne begreper for å diskutere den aktuelle problemstillingen, bør disse normalt velges på norsk. Det kan imidlertid være mange argumenter mot å gå over til selvalgte norske betegnelser. Hvis det dreier seg om en mengde begreper brukt f.eks. i en dokumentasjon som oppgaven bygger på, kan det bli mye vanskeligere for leseren å forstå sammenhengen dersom det ikke er en helt direkte korrespondanse i betegnelserne. Det kan også være at de engelske betegnelserne er så innarbeidet på norsk at det å bruke en norskere variant virker nokså kunstig.

Konklusjonen må bli at man i hvertfall må bruke et visst minstemål av energi på dette problemet om man møter det, og altså ikke uten videre gi seg ukritisk hen til engelske betegnelser. Hvor mye arbeid man spanderer på det vil avhenge av ens egen legning, hvor sentralt problemet står i oppgaven og hvor håpløst det ser ut i utgangspunktet. Problemet bør selvfølgelig diskuteres med veilederen. Om man bruker engelske betegnelser, må man velge mellom norske og engelske endelser, og her har vel de fleste tross alt fått minst problemer med norske. Man kan jo også sette den engelske delen av ordet i gåseøyne eller med kursiv, men det kan virke noe slitsomt dersom ordene går igjen til stadighet.

For å komme i gang med skriving i begynnelsen må man imidlertid ikke la seg hemme av dette problemet, men bare registrere på veien at man har et problem og forsøke å få en følelse av omfanget. På et visst tidspunkt må man så sette seg ned og vurdere i hvilken grad norske betegnelser er et reelt alternativ, og så lage seg en "standard" som man forsøker å følge siden.

Når man vurderer å gå over til norske betegnelser vil man oppleve også å måtte vurdere hva begrepene egentlig står for, og om det egentlig er dem man er ute etter. Denne gjennomtenkningen kan være meget sunn og fruktbar, og er altså en av de få fordelene vi har av denne språksituasjonen. Forsøk derfor å utnytte den godt.

1.14 Vær likefrem, ærlig og beskjeden

Om man f.eks. på et sent tidspunkt kommer på en ide som ville gjort oppgaven annerledes om man hadde tenkt på det tidligere, er det naturlig å forsøke å innarbeide denne i teksten. Det kan imidlertid føre for langt å gjøre dette fullstendig. Da kan det være like greit å la tingene stå som de er, og så legge på noen egne avsnitt der man rett og slett sier at dette kom sent, men at man gjerne vil ha det med. Her kan man så forklare idéen i forhold til hovedfremstillingen. Det samme kan man gjøre om idéen bare er at fremstillingen skulle vært lagt an annerledes, men da er det vel rimelig å begrense seg til noen setninger i en innledning.

I det hele tatt vil man så og si alltid tjene på å fremstille tingene som de er, og man skal heller ikke gjøre det i en tone som vitner om noen slags dårlig samvittighet. Det at idéer kommer sent, og at man en gang vil bli ferdig er helt greie argumenter. Det å komme med slike “meta-utsagn”, uten å overdrive antallet, vil også gjøre teksten mer tilgjengelig for leseren i og med at man får bakgrunn for strukturen. En mer eller mindre vellykket innarbeidelse av en ide på et sent tidspunkt vil som regel virke motsatt.

Man skal også være klar på hva som er ens egne idéer, og hva man har fra andre, og ikke minst hva som er alminnelig kunnskap innen dette spesielle feltet. Ens holdning til egne idéer forøvrig bør være høyst kjølig, selv om man holder på å boble over av iver og stolthet. I den grad man karakteriserer dem i det hele tatt skal det være med uttrykk som “idéene synes å være fruktbare, i den forstand at ...”. Andres idéer kan man uttrykke hårfint mer begeistring om, men enhver overdrivelse virker forunderlig og stort sett mot sin hensikt. Stort sett skal ting bare karakteriseres ved faglige termer.

1.15 Bruk av “jeg”, problemet med “han” og “hun”

Mange har en følelse av at det å bruke “jeg” i oppgaven blir for direkte, at det er uvitenskapelig eller liknende, og man forsøker alle slags omskrivninger som “forfatteren”, “vi” eller bruk av passiv form. Dette problemet er behandlet både i Pedersen (1975) og Høeg (1986), og konklusjonen er at man stort sett kan bruke “jeg” friere enn mange tror, men at det dog ikke skal brukes i ett sett. Om man er flere om oppgaven er det selvfølgelig naturlig å bruke “vi”, og “vi” kan også brukes ellers i betydningen “leseren og forfatteren”, f.eks. som når man sier: “Vi skal nå gå over til å se på”.

Mange studenter vil legge vekt på symmetrisk behandling av kvinner og menn, og får dermed problemer når man skal betegne f.eks. en programmerer som “han” eller “hun”. Man kan selvfølgelig unngå problemet med å skrive om alle slike setninger, men dette kan ofte være unaturlig, og ikke minst kreve mye energi. En annen mulighet er konsekvent å bruke “han/hun” eller “han eller hun”, men om det går igjen ofte kan dette virke omstendelig. En mulig løsning er at kvinnelige forfattere konsekvent bruker “hun”, mens mannlige bruker “han”. Denne idéen synes å være brukt i Bohlbro & Crossland (1980) .

1.16 Samarbeidet mellom veileder og student

Det er neppe tvil om at et godt samarbeid mellom student og veileder er grunnlaget for en vellykket hovedfagsoppgave. Som samarbeid har det imidlertid den skjevhet at det er hovedbeskjeftigelsen til studenten, mens det for veilederen er en av mange gjøremål.

Et visst minimum av tid, omtanke og oppmerksomhet har imidlertid studenten krav på, og veilederen må nok sette av minst en time pr. uke i gjennomsnitt. Om en av partene har forventninger eller planer som skiller seg vesentlig fra dette, bør de si fra før man starter opp et samarbeid. Hvor mye tid som til enhver tid brukes til veiledning kan imidlertid variere med i hvilken fase av arbeidet studenten er, og med mange andre hensyn. Som regel vil det være noe mer behov for veiledning i begynnelsen, og kanskje enda mer mot slutten når selve oppgaveteksten skal utarbeides. Det vil da også gå mye tid for veilederen til lesing av utkast til oppgaven.

Det kan være fornuftig å avtale faste tider for veiledning f.eks. hver uke eller hver 14. dag. Noen vil foretrekke å holde på slike faste møter (for kontinuitetens skyld) selv om man ikke har presserende spørsmål å diskutere, mens andre kan foretrekke å ta det mer etter behov. Studenten har imidlertid krav på en viss kontinuitet i den grad hun/han ønsker det.

I begynnelsen vil det ubetinget være veilederen som er "eksperten", men etter hvert vil dette ofte jevne seg ut på den måten at studenten blir eksperten på detaljene, mens det fremdeles er veilederen som lettest kan sette tingene i en større sammenheng. Det er da viktig at begge bestreber seg på å utnytte denne situasjonen best mulig.

Det er veilederens plikt å sørge for at studenten kommer ordentlig i gang med arbeidet. Dette innebærer først og fremst at veilederen må forsikre seg om at problemstillingen er forstått, slik at studenten stadig har grunnlag for å arbeide videre med den. Dessuten må veilederen hjelpe til med å finne frem nøkkellitteratur, og til å opprette kontakt med personer som studenten må samarbeide med (f.eks. ved å ta en telefon først).

På den annen side har selvfølgelig studenten en forpliktelse til å følge opp det man diskuterer og avtaler under veiledningen, og veilederen bør informeres dersom innsatsen blir liten en periode (f.eks. pga. eksamenslesing).

Oppgavens kvalitet er selvfølgelig til syvende og sist studentens ansvar, men det er viktig at veilederen løpende vurderer omfanget og kvaliteten av arbeidet, og sammenholder det med kravene til hovedfagsoppgaver. Dette må også jevnlig taes opp under veiledningen, og spesielt bør selvfølgelig studenten informeres dersom veilederen mener oppgaven er for tynn. Det omvendte kan imidlertid være like aktuelt. Veilederen må si fra dersom studenten har vanskelig for å avslutte arbeidet, og må også løpende gi råd om hvilke problemstillinger som bør og ikke bør følges for at oppgaven skal få et rimelig omfang.

Som nevnt tidligere er det ofte fruktbart at studenten leverer små notater som veilederen kan lese gjennom før selve veiledningen. Da har man noe konkret å snakke ut fra, og studenten får stadig forsøkt seg litt på egen hånd. Slike notater kan gjerne benyttes nesten fra starten av arbeidet, og de tidlige vil da naturlig inneholde like mye spørsmål som svar. Studenten må dog være klar over at veilederens tid er begrenset, og at de ikke kan belastes med særlig mye lesing på en regelmessig basis.

2. Litteratur

Dette kapittel er utgitt av informatikkbiblioteket (1992) og kan fås der som egen trykksak.

2.1 Hvordan lage en god litteraturfortegnelse til hovedfagsoppgaven og avhandlingen.

Hvorfor har du litteraturfortegnelser i hovedfagsoppgaven/avhandlingen din? Bl.a. fordi leseren skal kunne kikke deg i kortene, se hvilke tidligere arbeider du bygger på, hva som er dine egne idéer og om du har oversett litteratur som ville ha vært relevant for dine problemstillinger. Fortegnelsen skal inneholde alle trykte arbeider som er nevnt i teksten, men heller ikke flere.

En litteraturhenvisning (referanse) må derfor inneholde opplysninger nok til at det blir enkelt å identifisere og finne frem dokumentet for andre. Det er også viktig at disse opplysningene blir presentert på en konsekvent måte slik at de enkelte opplysningene i referansen kan identifiseres. Men det finnes forskjellige typer dokumenter som krever forskjellige sett av opplysninger: en vanlig bok, tidsskriftartikler, bidrag i bøker, bidrag på konferanser, tekniske rapporter, avhandlinger, offentlige utredninger, osv. Nedenfor vil du finne eksempler på hva forskjellige typer referanser bør inneholde av opplysninger og hvordan de kan oppstilles. Dette må leses som veiledning. Slike veiledninger finnes også i mange tidsskrifter. Det som er viktig er at du er konsekvent og holder deg til et valgt mønster. Helt til slutt i denne veiledningen finner du en eksempelsamling.

Først noen generelle kommentarer:

Litteraturfortegnelsen bør stilles opp i alfabetisk rekkefølge, dvs. alfabetisert på forfatters etternavn (første forfatter dersom de er flere om verket).

Hver enkelt referanse bør merkes, slik at du har noe å henvise til fra hovedteksten. Dette merket kan være en fortløpende nummerering ([1], [2], ..), men dersom du ønsker å beholde en alfabetisk ordning av referansene under arbeidet med oppgaven vil det være mer hensiktsmessig med et merke som utnytter den alfabetiske ordningen (ved f.eks. å bruke forfatternavnet og årstallet for publikasjonen [Knuth 80]).

LaTeX har mekanismer for å håndtere referanser og referanselister.

Navnet på tidsskrifter kan skrives fullt ut, men det er vanlig å forkorte dem - ikke til hva du måtte finne på - men til en standardisert forkortelse. Er du tvil, kan du spørre i biblioteket hva standardisert forkortelse er for det aktuelle tidskriftet. Du finner også forkortelsene i Computer & Control Abstracts :

List of Journals.

Du kan bruke engelske betegnelser som «vol.», «no.» og «pp.», men vær konsekvent.

Vi har valgt å sette tittelen på omslaget av dokumentet i kursiv. Den kan selvsagt utheves på andre måter (fete typer, eller bare store bokstaver). Videre har vi valgt å skrive forfatternavn med fornavn (evt. initialer) og deretter etternavn, fordi dette gir en mer leservennlig referanse når det er snakk om flere forfattere til samme verk.

2.1.1 Artikkel fra tidsskrift/konferanse/bok

En komplett referanse til en artikkel fra tidsskrift/konferanse/bok bør inneholde:

forfatter(e)

artikkeltittel (kan settes i anførsel)

navn på tidsskriftet/konferansen/boka (bør utheves)

tidsskriftartikkel: årgang, år, heftenummer, sidetall [Isloor 80] [West 89]

konferanse: sted, år, sidetall [Gray 81], [West 89]

bok: sted, forlag, år, sidetall. Vanlig bok [Lindsay 80], bokserie [Steel 91]

2.1.2 Bok/rapport/avhandling

En komplett referanse til en bok/rapport/avhandling bør inneholde:

forfatter(e)

tittel på boka/rapporten/avhandlingen (bør utheves)

bok: (evt. utgave), sted, forlag, år [Knuth 80]

rapport: avdeling, universitet, sted, rapport serie, tittel og nummer [Fisher 79]

avhandling: avhandlingen grad, universitet, år [Jaffe 79]

Det fins også andre typer dokumenter (offentlige utredninger, proposisjoner, lover, osv). Spør i biblioteket hvordan du skal føre opp slike.

2.2 Eksempler

[Fisher 79]

A.L. Fisher. «Implementation issues for algorithmic VLSI processor arrays». Dep. Comput. Sci., Carnegie-Mellon University, Pittsburgh, PA. Rep. CMU-CS-79-115, 1979.

[Gray 81]

J.N. Gray. «The transaction concept virtues and limitations». I: Proc. 7th int. conf. very large databases, Cannes, Frankrike, Sept. 1981, s. 144-154.

[Isloor 80]

S.S. Isloor og T.A. Marsland. «The deadlock problem : an overview». IEEE Comput. Mag., årg. 13 (1980), nr 9, s. 58-78.

[Jaffe 79]

J.M. Jaffe. «Parallel computation : synchronization, scheduling and schemes», Ph.D. disseration, Massachusetts Institute of Technology, August. 1979.

[Knuth 80]

D.E. Knuth. «The art of computer programming, vol. 1: Fundamental algortihms». 2nd ed., Reading, Mass. : Addison-Wesley, 1980. [Lindsay 80] B. Lindsay, P.Selinger, C. Galtieri, J. Gray, R. Lorie, F. Putzolu, I. Traiger og B. Wade. «Single and multi-site recovery facilities». I: W. Draffa og F. Poole (red), Distributed data bases. Cambridge, Mass. : Cambridge University Press, 1980, s.1-10.

[Steel 91]

S. Steel. «Knowledge subgoals in plans». I: J. Hertzberg (red), Proc. European Workshop on planning : EWSP '91, Sankt Augustin, FRG, March 18-19, 1991. Berlin : Springer, 1991, s. 112-121. (Lecture notes in computer science ; 522).

[West 89]

Colin H. West. "Protocol validation in complex systems". I: SIGCOMM «89 Symposium: Communications Architectures & Protocols, Austin, Texas, Sept. 19-20, 1989. Comput. comm. rev., årg 19 (1989), nr 4, s.303-12.

2.3 Litteratur om å skrive hovedfagsoppgave. (listet på en annen måte)

Bohlbro, Merete & Crossland, Hanne. 1980: "Veiledning i rapportskrivning" (Dansk), Polyteknisk Forlag .

Day, Robert A. 1989: "How to write publish & a scientific paper". Cambridge university press, Cambridge, 211pp.

