

**Løsningsforslag til**  
**Kontinuasjoneksamen i fag**  
**45160 Systemering 1**  
**Lørdag 17. august 1996**

**Oppgave 1 – Programvaredesign (15%)**

a) Hva ligger i begrepene *coupling* og *cohesion*?

(Læreboka s. 90-95)

- *Coupling* er et mål på grad av sammenkobling av moduler i en programvarestruktur. Det finnes seks grader av coupling fra lavest til høyest:
  1. Data coupling - to subprogrammer kommuniserer gjennom grensesnittet via parametre.
  2. Frimerke (stamp) coupling - to ellers uavhengige subprogrammer refererer til samme datastruktur, men bruker forskjellige deler av den.
  3. Kontroll coupling - tilstandsvariabler kommuniseres mellom subprogrammer (f.eks. flagg).
  4. Ekstern coupling - subprogrammer er bundet til eksterne enheter (f.eks. I/O enheter, spesielle formater og protokoller.)
  5. Felles (common) coupling - subprogrammer deler globale variable.
  6. Innholds (content) coupling - programmoduler refererer til hverandres implementasjon (f.eks. ved bruk av GOTO-setninger.)
- *Cohesion* er et mål på funksjonell assosiasjon til programkomponentene i et subprogram, dvs. "limet" som holder subprogrammet sammen. Det finnes sju grader av cohesion fra høyest til lavest:
  1. Funksjonell cohesion - hvert subprogram bidrar til å utføre en enkelt funksjon komplett uten å påvirke andre subprogram.

2. Sekvensiell cohesion - output fra et subprogram sendes som input til et annet. Subprogrammene utfører ikke komplette funksjoner (ellers ville vi hatt functional cohesion).
3. Kommunikativ cohesion - funksjoner som jobber mot felles data grupperes sammen.
4. Prosedural cohesion - funksjoner som må utføres i en spesiell sekvens grupperes sammen.
5. Temporal cohesion - funksjoner som er tidsrelaterte grupperes sammen (f.eks. gruppere all initialisering i et subprogram).
6. Logisk cohesion - funksjoner som er nesten like, grupperes sammen (f.eks. samle ti forskjellige utskriftsrutiner i en programmodul).
7. Tilfeldig cohesion - tilfeldig gruppering av funksjoner.

b) Hvilke retningslinjer finnes for å oppnå god strukturell kvalitet på programvare? Forklar også kort hvordan disse retningslinjene bidrar til forbedret strukturell kvalitet.

(Læreboka s. 96-102)

- Reduser coupling og forbedre cohesion. Gir bedre strukturell oppbygging av systemet og gjør vedlikeholdsarbeid mye enklere.
- Minimaliser fan-out. Strev for å oppnå høy fan-in (dvs. lag gjenbrukbare subprogram). Gjenbruk reduserer systemstørrelse og kompleksitet.
- Begrens scope-of-effect til scope-of-control, dvs. vi ønsker ikke sideeffekter som er vanskelig å holde orden på og gir dårlig struktur. Dette reduserer også kontroll coupling.
- Reduser kompleksitet i grensesnittene. Tilsynelatende urelaterte data som passerer gjennom grensesnitt indikerer lav cohesion.
- Søk å oppnå single-entry, single-exit subprogrammer og unngå content coupling.
- Unngå subrutiner med "internt" minne slik at oppførselen er avhengig av tidligere subrutinekall.
- Prøv å moderere bruken av retningslinjene for systemutvikling. Disse retningslinjene gir som oftest ikke optimal løsning m.h.p. ytelse.

## Oppgave 2 – Informasjon og kunnskap (20%)

a) Kunnskap kan klassifiseres på forskjellige måter. Diskutér ulike måter å gjøre dette på.

(Læreboka s. 368-369)

Det er mange måter å klassifisere kunnskap. Læreboka nevner følgende måter:

- Datamaskinperspektiv: Deklarativ og prosedural kunnskap.  
Deklarativ kunnskap er her klassedefinisjoner og databaseskjemaer, mens prosedural kunnskap handler om hvordan disse brukes og eksekveres.
- Fra psykologien: Eksplisitt og implisitt kunnskap.  
Eksplisitt kunnskap kan beskrives verbalt, implisitt kunnskap er ferdigheter.
- Overflatekunnskap vs. innsikt (deep knowledge).  
Overflatekunnskap er gjerne et knippe tommelfingerregler, men innsikt er kjennskap til prinsippene bak.
- Kunnskap og meta-kunnskap.  
Meta-kunnskap er kunnskap om kunnskap og hvordan den er strukturert.
- Rittel klassifiserer i fem forskjellige typer:
  - Factual knowledge: x er eller vil bli et faktum.
  - Deontic knowlegde: x bør være eller bør bli et faktum. Uttrykker menneskelig overbevisning eller målsetting.
  - Explanatory knowledge: x er et faktum fordi y...
  - Instrumental knowledge: når x er gjort, blir y konsekvensen. Hvordan-kunnskap.
  - Conceptual knowledge: meta-kunnskap eller kunnskap om kunnskap.

**b)** Kunnskap og informasjon kan ikke alltid beskrives i presise termer. Spesielt i systemutviklingssammenheng forstår vi ofte ikke problemet nok til å kunne definere konsepter presist. Diskutér hvordan vi i slike sammenhenger kan klassifisere informasjon. Hvilke egenskaper (eller attributter) har informasjon generelt?

(*Læreboka s. 370-372*)

Læreboka foreslår å klassifisere informasjon som presis, vag, usikker eller ufullstendig. I tillegg kan vi ha ukorrekt informasjon som en femte kategori.

- Presis informasjon er mest vanlig i data-verdenen og består av svart-på-hvitt fakta. Første ordens logikk brukes ofte for å beskrive slik informasjon.
- Vag informasjon finner vi i dagliglivet. Vi sier f.eks. at "været er bra" og gir en kvalitativ bedømmelse uten å presisere hva vi mener med "bra". Betydningen av termene er bl.a. gitt av konteksten de er brukt i og hvem som bruker dem.
- Usikker informasjon er vag fordi man ikke kan gi presis informasjon.
- Ufullstendig informasjon reflekterer av observatøren ikke har komplett kunnskap. Forskjellen fra usikker informasjon er at ufullstendig informasjon kan være presis. Man kan f.eks. vite at Oslo er Norges hovedstad, men ikke at Oslo ligger sør i landet.

Informasjon vil ha følgende attributter:

- Relevans - sier noe om viktigheten av informasjonen.
- Korrekthet - sier noe om hvor virkelighetsnær informasjonen er.
- Presisjon (accuracy) - sier noe om hvor vag informasjonen er.
- Innhold - sier noe om hvilken og hvor mye kunnskap informasjonen formidler.
- Originalitet - sier noe om hvor forskjellig informasjonen er fra allerede eksisterende kunnskap.
- Foreldethet (obsoleteness) - sier noe om hvor aktuell informasjonen er tidsmessig.

### Oppgave 3 – Logikk (15%)

a) Nedenfor er det gitt en liten fotballhistorie.

*Per og Pål spiller fotball på hvert sitt lag. Hvis Per er uforsiktig og takler Pål, blir Pål skadet. Men hvis Per viser seg uforsiktig og Pål i tillegg blir skadet, blir dommeren sur. Så lenge Per ikke er uforsiktig, vil han heller ikke bryte reglene. Hvis Per ikke bryter reglene eller dommeren ikke blir sur, vil Per unngå å få gult kort. Men Per ser sitt snitt når dommeren ikke er sur til å spille uforsiktig. Nå skjedde det faktisk at Per taklet Pål og fikk gult kort.*

Avgjør ved resolusjon ut i fra denne fotballhistorien om dommeren var sur eller ikke.

(Læreboka s. 243-246)

Vi formulerer først atomiske påstander:

- A. Per er uforsiktig.
- B. Per takler Pål.
- C. Pål blir skadet.
- D. Dommeren blir sur.
- E. Per bryter reglene.
- F. Per får gult kort.

Vi omskriver historien vha. disse atomiske påstandene og første ordens logikk:

Setning	Uttrykk
Hvis Per er uforsiktig og takler Pål, blir Pål skadet.	$A \wedge B \rightarrow C$
Men hvis Per viser seg uforsiktig og Pål i tillegg blir skadet, blir dommeren sur.	$A \wedge C \rightarrow D$
Så lenge Per ikke er uforsiktig, vil han heller ikke bryte reglene.	$\neg A \rightarrow \neg E$
Hvis Per ikke bryter reglene eller dommeren ikke blir sur, vil Per unngå å få gult kort.	$\neg E \vee \neg D \rightarrow \neg F$
Men Per ser sitt snitt når dommeren ikke er sur til å spille uforsiktig.	$\neg D \rightarrow A$
Nå skjedde det faktisk at Per taklet Pål og fikk gult kort.	$B \wedge F$

Vi skriver om formlene til klausal form:

$$A \wedge B \rightarrow C \Leftrightarrow \neg(A \wedge B) \vee C \Leftrightarrow \neg A \vee \neg B \vee C$$

$$A \wedge C \rightarrow D \Leftrightarrow \neg(A \wedge C) \vee D \Leftrightarrow \neg A \vee \neg C \vee D$$

$$\neg A \rightarrow \neg E \Leftrightarrow A \vee \neg E$$

$$\neg E \vee \neg D \rightarrow \neg F \Leftrightarrow \neg(\neg E \vee \neg D) \vee \neg F \Leftrightarrow (E \wedge D) \vee \neg F \Leftrightarrow (E \vee \neg F) \wedge (D \vee \neg F)$$

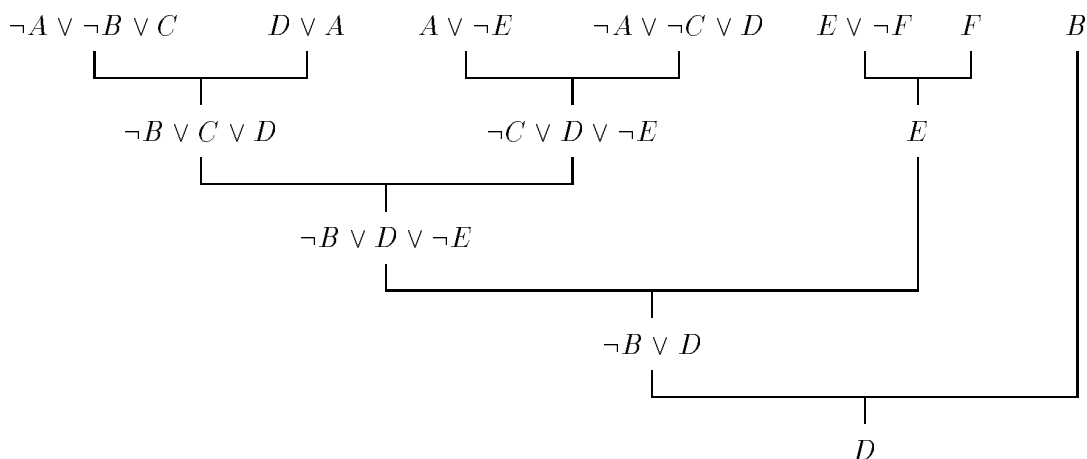
$$\neg D \rightarrow A \Leftrightarrow D \vee A$$

$$B \wedge F$$

og får følgende formel som beskriver fotballhistorien:

$$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg C \vee D) \wedge (A \vee \neg E) \wedge (E \vee \neg F) \wedge (D \vee \neg F) \wedge (D \vee A) \wedge B \wedge F$$

Bruker resolusjon og grupperer termene slik at vi får eliminert så mange atomiske påstander som mulig.



Termen  $(D \vee \neg F)$  er ikke brukt over. Ved å bruke  $F$ -termen på nytt<sup>1</sup> vil vi ha  $(D \vee \neg F) \wedge F \Leftrightarrow D$  ved resolusjon, dvs. konklusjonen er fortsatt  $D$  eller at dommeren ble sur.

<sup>1</sup>Vi kan alltid bruke termer flere ganger fordi  $X \Leftrightarrow X \wedge X$

Vi kunne også vise dette ved å anta at dommeren ikke ble sur  $\neg D$  og vise inkonsistensen som da oppstår.

b) I pensumboka påstås det at det er umulig å bevise konsistens i predikatlogikk, men at inkonsistens kan detekteres. Hvorfor er dette tilfelle, og hva kaller vi denne egenskapen til predikatlogikk?

(Læreboka s. 246)

Man kan ikke bevise konsistens i et sett av logiske formler som allerede er konsistent fordi bevisførselen vil ikke terminere. Man kan eventuelt bevise inkonsistens ved å detektere selvmotsigelser. Egenskapen kalles *semi-decidability*.

c) Hvis vi ønsker å legge til et nytt predikat til et eksisterende sett av predikater som vi vet er konsistent, hvordan kan vi avgjøre om det nye predikatet er konsistent med resten av regelsettet?

(Læreboka s. 247)

Det nytter ikke å bevise konsistens direkte, så vi legger heller til negasjonen av det nye predikatet og ser om vi får inkonsistens. Får vi det, var det opprinnelige predikatet konsistent med resten av regelsettet.

## Oppgave 4 – Modellering (50%)

### Flykontrollsystem

*Et flykontrollsystem består av 21 lufttrafikk-kontrollsentra som alle er delt opp i trafikksektorer. Flykontrollen utføres av mennesker som hver har ansvar for en trafikksektor og som blir betjent av en datamaskin i vedkommende kontrollsenter. Flyene registreres i systemet ved meldinger som beskriver flykaptainens planer og som bl.a. inneholder: destinasjoner, rute, hastighet, anmodning om flyhøyde og avgangstid, estimert flytid og forventet ankomsttid.*

*Denne ruteplanen legges inn i datasystemet to timer før flyavgang og oppbevares i startstedets lufttrafikk-kontrollsentra. Fem minutter før planlagt flyavgang oversendes en forkortet versjon av ruteplanen til kontrollørene i den første trafikksektoren som flyet skal gjennom. Den forkortede flyplanen blir overført til kontrollørene i de etterfølgende sektorer som berøres av flyet fem minutter før flyet kommer inn i vedkommende sektor. Når flyet går over fra ett kontrollsenter til et annet, blir den komplette flyplanen overført til vedkommende kontrollsenter fem minutter før flyet kommer inn i kontrollområdet.*

a) Beskriv systemet nedenfor ved hjelp av dataflytdiagram. Dekomponér systemet ett nivå ned.

Dataflytdiagram er foreløbig ikke tilgjengelig. Fokus ligger på riktig syntaks, kompletthet, oversiktighet, konsistens og riktighet i forhold til domenenet.

b) Beskriv de statiske aspektene vha. ER-diagram.

I dette tilfellet modellerer vi domenet - ikke et tenkt databasesystem! Men ikke alt er like viktig. F.eks. det at kontrollsentret inneholder en datamaskin som lagrer flyplaner anses som implisitt i diagrammet nedenfor.

