MAT1030 – Diskret Matematikk

Forelesning 8: Logikk, predikatlogikk, bevisteknikker

Dag Normann

Matematisk Institutt, Universitetet i Oslo

10. februar 2010

(Sist oppdatert: 2010-02-10 13:48)

Kapittel 4: Mer predikatlogikk

Oppsummering

Vi gjentar læringsmålene for utsagnslogikk og predikatlogikk:

1. Definisjonene av utsagn og predikat, og å kunne bestemme hvilke ytringer som er utsagn, hvilke som er predikat og hvilke som faller utenfor rammene våre.
2. De logiske bindeordene $\neg$, $\wedge$, $\vee$, $\rightarrow$ og $\leftrightarrow$ og hvordan de defineres via sannhetsverditabeller.
3. Sette opp sannhetsverditabellen til et sammensatt uttrykk, og bruke denne til å bestemme om et utsagn er en tautologi, en kontradiksjon eller ingen av delene.
4. Kjenne til logikkens lover og i noen utstrekning kunne bruke dem.
5. Spesielt sentralt står deMorgans lover og de distributive lovene.
6. Kjenne definisjonene av kvantorene $\forall$ og $\exists$ og kjenne deMorgans lover for kvantorer.
7. Kunne uttrykke en sammenheng ved bruk av kvantorer og kunne "forstå" et uttrykk som inneholder kvantorer.

Anvendelser av predikatlogikk

Vi nevnte kort et område hvor predikatlogikk ikke har noen fornuftig funksjon, i tester for styring av **while**-løkker og liknende.

Vi skal se på et par eksempler hvor bruk av predikatlogikk er nyttig.

I innledningen til kapitlet om logikk trakk vi frem en rekke andre eksempler, eksempler som vi ikke kan gå detaljert inn på.

Anvendelser av predikatlogikk

Eksempel (Kvalitetssikring av databaser)

- Anta at vi skal bygge opp en base for registrering av slektskapsforhold.
- Vi vil registrere noen grunnleggende slektskapsforhold.
- For å sikre oss mot at vi lagrer data på feil måte skal vi sette opp visse aksiomer som dataene våre skal respektere.
- Hvis vi kan utlede en kontradiksjon fra de lagrede slektskapsforholdene og aksiomene, har vi foretatt en feillagring.

Anvendelser av predikatlogikk

Eksempel (Fortsatt)

- Noen aksiomer kan være
  - $\text{Far}(x,y) \land \text{Far}(x,z) \to \text{Søsken}(y,z)$
  - $\text{Mor}(x,y) \land \text{Mor}(x,z) \to \text{Søsken}(y,z)$
  - $\text{Søsken}(x,y) \land \text{Far}(x,z) \land \text{Mor}(y,z) \to \mathbf{F}$
- Dette sikrer at vi ikke lagrer søsken som foreldre til det samme barnet.
- Riktignok medfører disse aksiomene at alle er sin egen søsken, men siden ingen av oss vil kunne bli både mor og far til det samme barnet, er kvalitetssikringen ivaretatt uansett.

Anvendelser av predikatlogikk

- Hvis en datamaskin skal gi oss en feilmelding ut fra at de dataene vi har lastet inn leder til en kontradiksjon, må den være programmert til å gjøre det.
- En mulighet er å bruke et spesialkonstruert programmeringsspråk PROLOG til dette formålet.
- Et PROLOG-program vil være en liste av kvantorfrie predikater av en bestemt form, og når programmet kjøres innebærer det å vise at predikatene samlet sett er kontradiktoriske.

Anvendelser av predikatlogikk

- PROLOG har sin egen syntaks, men oversatt til vårt språk kan en PROLOG-instruks være på en av tre former:
  1. $A_1 \wedge \cdots \wedge A_n \rightarrow B$
  2. $A_1 \wedge \cdots \wedge A_n \rightarrow \mathbf{F}$
  3. B
- Svært mange relevante sammenhenger mellom data i en base kan formuleres som en PROLOG-instruks.
- Her vil A_i og B være positive grunnpredikater uten bruk av kvantorer eller bindeord, da heller ikke negasjon.
- Eksempler kan være $x < y$, $\mathrm{Far}(x, y)$, Forbudt(Promillekjøring) og "Per bedriver promillekjøring".
- Vi skal gi et veldig enkelt eksempel på hvordan PROLOG kan brukes til å søke etter data i en base.

Anvendelser av predikatlogikk

Eksempel

- Vi ser litt nærmere på slektskapsbasen vi så på i sted.
- Vi har lagret informasjon om hvem som er mor eller far til hvem. Annen informasjon må utledes.
- På samme måte som vi innfører *Søsken* som en utledet egenskap, kan vi innføre *Farfar* ved

  $\mathrm{Far}(x, y) \land \mathrm{Far}(y, z) \to \mathrm{FarFar}(x, z)$,
- Tor oppsøker denne basen og lurer på om han har noen farfar.

Anvendelser av predikatlogikk

Eksempel (Fortsatt)

- I basen er det lagret $\mathrm{Far}(\mathrm{Per}, \mathrm{Tor})$ og $\mathrm{Far}(\mathrm{Knut}, \mathrm{Per})$.
- Programmereren som styrer basen legger til aksiomet

$$\mathrm{FarFar}(x, \mathrm{Tor}) \to \mathbf{F}$$

- Dette sier at x ikke er farfar til Tor.
- Hvis det leder til en motsigelse, vet Tor at han har en farfar registrert i basen.

Anvendelser av predikatlogikk

Eksempel (Fortsatt)

- PROLOG vil nå målrettet prøve å utlede **F** fra det nye aksiomet og den lagrede informasjonen.
- Gangen vil være omtrent som følger:
  1. Fra $\text{FarFar}(x, \text{Tor}) \rightarrow \mathbf{F}$ og $\text{Far}(x, y) \wedge \text{Far}(y, z) \rightarrow \text{FarFar}(x, z)$ kan vi slutte $\text{Far}(x, y) \wedge \text{Far}(y, \text{Tor}) \rightarrow \mathbf{F}$ ved at vi setter inn Tor for z.
  2. Fra $\text{Far}(x, y) \wedge \text{Far}(y, \text{Tor}) \rightarrow \mathbf{F}$ og $\text{Far}(\text{Per}, \text{Tor})$ kan vi slutte $\text{Far}(x, \text{Per}) \rightarrow \mathbf{F}$ ved at vi setter inn Per for y.
  3. Fra $\text{Far}(\text{Knut}, \text{Per})$ og $\text{Far}(x, \text{Per}) \rightarrow \mathbf{F}$ kan vi slutte **F** ved at vi setter inn Knut for x.
- PROLOG vil ikke bare bevise at Tor har en farfar, men den virker slik at den finner frem til en farfar blant dataene.
- For de som bare leser utskriftene: Endel ble forklart muntlig på forelesningen.

Kapittel 4: Bevisteknikker

Litt repetisjon

- Vi har nå gått gjennom både utsagnslogikk og predikatlogikk.
- Vi innførte

  eksistenskvantoren $\exists$ og
  allkvantoren $\forall$.
- Vi så på en del eksempler på oversettelse mellom dagligtale og uttrykk med kvantorer.
- Vi viste noen logiske ekvivalenser:
  - deMorgans lover: $\neg\exists x A \equiv \forall x \neg A$ og $\neg\forall x A \equiv \exists x \neg A$.
  - Sammentrekning: $\exists x A \vee \exists x B \equiv \exists x(A \vee B)$ og $\forall x A \wedge \forall x B \equiv \forall x(A \wedge B)$.

Bevisteknikker

- Den siste delen av kapittel 4 handler om forskjellige bevisteknikker.
- Vi skal se på måter å strukturere et matematisk bevis på.
- Dette er et tema alle studenter i matematikk eller et annet teoretisk fag etterhvert vil kjenne seg igjen i.
- Vi skal se på direkte bevis, bevis ved tilfeller og kontrapositive bevis.
- Senere skal vi føye induksjonsbevis til vår meny av bevisteknikker.
- Vi skal eksemplifisere de forskjellige bevisformene.

Bevisteknikker

Eksempel

- Vi skal vise at differensen mellom to kvadrattall som kommer etter hverandre i tallrekken er et oddetall.
- Vi kan formulere dette mer matematisk som en påstand:

  For alle tall n er $(n+1)^2 - n^2$ et oddetall.

Bevis

- Ved 1. kvadratsetning er $(n+1)^2 = n^2 + 2n + 1$, så $(n+1)^2 - n^2 = n^2 + 2n + 1 - n^2 = 2n + 1$.
- Siden $2n + 1$ alltid er et oddetall er påstanden vist.

Bevisteknikker

- I dette eksemplet formulerte vi først det vi skulle vise i en matematisk språkdrakt, deretter regnet vi litt på den differensen vi skulle bevise var et oddetall, og endte opp med at det var akkurat et oddetall det var.
- Hvis vi analyserer beviset litt nærmere ser vi at alle oddetallene kan fremkomme som en slik differens, for å få $2n + 1$ som verdi, kan vi velge kvadrattallene $(n + 1)^2$ og n^2.
- Når vi først har funnet et bevis, kan vi undersøke om samme metode kan gi oss mer innsikt. Det vil ofte være tilfelle, men kan kreve ekstra innsats.
- La oss se om vi kan bruke samme resonnement til å si noe om differensen mellom kubikktall.

Bevisteknikker

Teorem

- For alle naturlige tall n er $(n+1)^3 - n^3$ et oddetall.

Bevis

- Vi har at $(n+1)^3 = n^3 + 3n^2 + 3n + 1$, så

$$(n+1)^3 - n^3 = 3n^2 + 3n + 1.$$

- Hvis n er et partall, er $3n^2 + 3n$ også et partall, så $(n+1)^3 - n^3$ er et oddetall.
- Hvis n er et oddetall, er både $3n^2$ og $3n$ oddetall, så $3n^2 + 3n$ er fortsatt et partall, og også i dette tilfellet er $(n+1)^3 - n^3$ et oddetall.
- Dermed er påstanden bevist.

Bevisteknikker

- Som vi ser, brukte vi akkurat samme resonement i starten av disse to bevisene.
- I det andre beviset måtte vi imidlertid etterhvert dele argumentet opp i to tilfeller, ett for at n er et oddetall og ett for at n er et partall. Siden dette dekker alle mulighetene, er beviset fullstendig.
- La oss gi et tredje eksempel på et bevis hvor vi må dele argumentet opp i tilfeller.

Bevisteknikker

Eksempel

- La oss bevise følgende påstand:

  Hvis n er et helt tall, så kan $n^2 - n$ deles på 6 eller så kan $n^2 + n$ deles på 6.

Bevisteknikker

Bevis

- Vi har at $n^2 - n = (n-1)n$ og at $n^2 + n = n(n+1)$.
- Nøyaktig ett av tallene $n-1$, n eller $n+1$ kan deles på 3.
- Hvis $n-1$ kan deles på 3 er ett av tallene $(n-1)$ eller n et partall, og da er $(n-1)n$ delelig med 6.
- Hvis $n+1$ er delelig med 3 er ett av tallene n eller $n+1$ partall, så $n(n+1)$ er delelig med 6.
- Hvis n er delelig med 3 ser vi ved samme argument at både $(n-1)n$ og $n(n+1)$ er delelige med 6.
- Tilsammen beviser dette påstanden.

Bevisteknikker

- Dette eksemplet viser hvordan man enkelte ganger må dele et argument opp i tilfeller.
- Det viser imidlertid også at man av og til må få anta at leseren henger med i noen av svingene, uten at alle detaljene som ligger til grunn for beviset blir tatt med.
- Vi har for eksempel tatt det for gitt at leseren er med på at 6 er faktor i $(n-1)n$ når $n-1$ kan deles på 3 og $n-1$ eller n er et partall.
- Vi har heller ikke minnet om det er fordi $n^2 - n = (n-1)n$ at vi har vist påstanden når vi i realiteten viser at $(n-1)n$ eller $n(n+1)$ kan deles på 6.
- Hvor mange detaljer man tar med er en vurderingsak, og vil være avhengig av målgruppen.

Kontrapositive bevis

- De bevisene vi har sett på til nå kalles direkte bevis.
- Dette er for å skille dem fra såkalte kontrapositive eller indirekte bevis.
- Hvis vi går tilbake til utsagnslogikken, ser vi at utsagnene

$$p \to q$$

og

$$\neg q \to \neg p$$

er logisk ekvivalente.

- Det betyr at hvis vi ønsker å vise en påstand på formen $A \to B$, kan vi like gjerne anta $\neg B$ og bevise $\neg A$.
- Et bevis på denne formen kalles kontrapositivt.

Kontrapositive bevis

- Før vi gir et eksempel, skal vi se på et generelt spesialtilfelle.
- Det å bevise en påstand A er det samme som å vise $\mathbf{T} \to A$.
- Den kontrapositive varianten vil være å vise $\neg A \to \neg\mathbf{T}$, det vil si $\neg A \to \mathbf{F}$.
- $\mathbf{F}$ er selvmotsigelsen i sin reneste form, så en måte å bevise A på vil være å anta $\neg A$, og så utlede en selvmotsigelse.
- Dette er en vanlig måte å bevise teoremer på, enten ved at man antar at det man skal bevise er usant eller som en hjelp til å bevise påstander man trenger underveis i beviset.

Kontrapositive bevis

- Det klassiske eksemplet på et kontrapositivt bevis er Pythagoreernes argument for at $\sqrt{2}$ ikke er et rasjonalt tall.
- Dette er gitt som en oppgave i boka, og vi skal la det være med det.
- Vi skal gi tre eksempler på kontrapositive bevis.
- Det første er bare for å illustrere metoden, mens det siste viser et meget viktig resultat som berører forståelsen av informatikkens begrensninger.

Kontrapositive bevis

Teorem

Alle naturlige tall > 1 er primtall eller kan faktoriseres i primtall.

Bevis

- Anta at $n > 1$ hverken er et primtall eller kan faktoriseres i primtall.
- Siden n ikke er et primtall, kan n skrives som et produkt $n = ab$ hvor $1 < a < n$ og $1 < b < n$.
- Hvis både a og b enten er primtall eller kan faktoriseres i primtall, vil n kunne faktoriseres i primtall.

Kontrapositive bevis

Bevis (Fortsatt)

- Det har vi antatt at ikke er tilfelle.
- Derfor fins det $n_1 < n$ slik at $1 < n_1$ og slik at n_1 hverken er et primtall eller kan faktoriseres i primtall.
- Da kan vi fortsette dette resonementet n ganger, og får $1 < n_n < n_{n-1} < \ldots < n_1 < n$ slik at ingen av disse tallene er primtall eller kan faktoriseres i primtall.
- Dette er umulig, siden det ikke fins så mange tall mindre enn n.
- Derfor må forutsetningen, som var at teoremet vårt ikke holder, være feil.

Når vi har lært om induksjonsbevis, vil vi kunne bevise dette på en måte som virker mer overbevisende på matematikere.

Kontrapositive bevis

Teorem

La n og m være positive heltall, og la c være den største felles faktoren i n og m.

Da finnes det hele tall a og b slik at

$$c = an + bm.$$

Bevis

La d være det minste positive hele tallet slik at d kan skrives på formen

$$d = an + bm$$

hvor a og b er heltall.

Kontrapositive bevis

Bevis (fortsatt)

Siden c er en faktor i både n og m, er c en faktor i d, så $c \leqslant d$.

Det er derfor nok å vise at d er en faktor både i n og i m.

Anta at det ikke er tilfelle.

Ved symmetri kan vi anta at d ikke er faktor i n, og hvis vi deler n på d får vi en rest $r < d$.

Da finnes det k, a og b slik at

$$r = n - kd = n - k(an + bm) = (1 - ka)n + (-kb)m.$$

Dette er umulig siden d er det minste positive tallet som kan skrives som en kombinasjon av n og m.

Dermed er teoremet bevist.

Et langt bevis

- Vi skal vise at under svært generelle forutsetninger er det ikke mulig å løse noen helt grunnleggende problemer om egenskaper ved programmer.
- Vi skal la $\mathcal{P}$ være et programmeringsspråk som har følgende egenskaper:
  - Det er mulig å skrive et program for en prosedyre som i teorien aldri stopper (mens $x \geqslant 0$, sett $x = x + 1$).
  - Vi kan la et program P_2 etterfølge et program P_1 ved en konstruksjon som
  $$P_1; P_2.$$
  - Vi kan skille mellom tilfeller som i **If** ... **then** ... **else**.
  - En hvilken som helst tekstfil, eksempelvis et program, kan tjene som input.
  - Det fins et program for kopiering av en fil, det vil si, som til en inputtekst t gir output tt, dvs t og en kopi av t.

Et langt bevis

Teorem

- La $\mathcal{P}$ være et programmeringsspråk med egenskapene på forrige side.
- Da fins det ikke noe program Q for å avgjøre om et annet program P med input t vil stoppe eller fortsette i det uendelige.

Bevis

- Med et program vil vi her mene tekstfilen som utgjør programmet.
- Hvis P er et program og t er et mulig input, skriver vi $P(t)$ for tilsvarende output.
- Anta at teoremet er feil, og at det fins et program Q slik at om P er et annet program og t er et mulig input, så vil
  - $Q(Pt) = 1$ om $P(t)$ har en verdi, dvs P med input t stopper.
  - $Q(Pt) = 0$ om $P(t)$ i teorien aldri stopper.

Et langt bevis

Bevis (Fortsatt)

- La C være et program slik at $C(t) = tt$ for alle t og la U være et program som aldri stopper uansett input.
- La R være et program som svarer til
- **If** $Q(C(t)) = 1$ **then** U **else** output 1.
- Husk at R også er tekstfilen til R.
- Anta at $R(R)$ stopper. La oss se på hvordan beregningen til $R(R)$ må se ut.
- Vi har at $Q(C(R)) = Q(RR) = 1$ fordi $R(R)$ stopper.
- Men etter den testen, fortsetter R med U, det vil si at $R(R)$ ikke stopper.

Et langt bevis

Bevis (Fortsatt)

- Men hvis $R(R)$ ikke stopper, er $Q(C(R)) = 0$, så R gir output 1, hvilket betyr at $R(R)$ stopper likevel.
- Når vi konstruerer R på denne måten, vil beregningen av $R(R)$ stoppe hvis og bare hvis beregningen aldri stopper.
- Dette er en motsigelse, og konklusjonen må være at det ikke fins noe program Q som følger spesifikasjonene.

Konstruktive bevis

- I informatikk kan det ofte være lurt å bruke konstruktive bevis.
- Det betyr at man ikke ukritisk gjør bruk av tautologier som $p \vee \neg p$, men at man skal ha kontroll på hvilken av de to delene som holder.
- Kontrapositive bevis har heller ingen plass i konstruktiv matematikk.
- Fordelen med konstruktive bevis er at man kan trekke algoritmer og annen form for informasjon ut av bevisene.
- Det klassiske beviset for at det alltid fins et større primtall er konstruktivt.
- Vi skal se et eksempel på et bevis som ikke er konstruktivt.

Konstruktive bevis

Teorem

Det fins to irrasjonale tall a og b slik at $a^b \in \mathbb{Q}$.

Bevis

Vi deler beviset opp i to tilfeller.

- Tilfelle 1: $\sqrt{2}^{\sqrt{2}} \in \mathbb{Q}$.

  Da lar vi $a = b = \sqrt{2}$ og $a^b \in \mathbb{Q}$.
- Tilfelle 2: $\sqrt{2}^{\sqrt{2}} \notin \mathbb{Q}$.

  Da lar vi $b = \sqrt{2}$ og $a = \sqrt{2}^{\sqrt{2}}$.
- Da får vi

$$a^b = (\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{(\sqrt{2}\cdot\sqrt{2})} = \sqrt{2}^2 = 2 \in \mathbb{Q}.$$

Konstruktive bevis

- Dette eksemplet blir ofte brukt til å illustrere forskjellen på konstruktive bevis og bevis basert på klassisk logikk.
- I klassisk logikk kan vi gjøre uhemmet bruk av antagelsen at enten gjelder en påstand P eller så gjelder negasjonen $\neg P$.
- I konstruktiv matematikk kreves det at vi i tillegg har noe informasjon om hvilken av de to som gjelder, eller i det minste en metode for å avgjøre hvilken av de to som gjelder i en gitt situasjon.
- Ut fra det beviset vi har sett på kan vi ikke si noe sikkert om hvilket par a, b av irrasjonale tall det er som er slik at $a^b \in \mathbb{Q}$, bare at det fins et slikt par.

Konstruktive bevis

- Vi ga tidligere et bevis for at hvis c er største felles faktor til n og m, så finnes det heltall a og b slik at

$$c = an + bm.$$

- Hvis vi ser nærmere på det beviset, ser vi at vi ikke har noen informasjon om hvilke tall a og b er.
- Dette teoremet vises vanligvis ved å ta utgangspunkt i Euklids algoritme som vi har eksemplifisert ved en pseudokode.
- Da blir beviset mer konstruktivt, og vi kan utlede en algoritme for å finne a og b fra det beviset.
- Skal vi være helt ærlige, er det egentlig det samme beviset i to utgaver.