Databaser fra et logikkperspektiv

Evgenij Thorstensen

IFI, UiO

Høst 2013

Outline

1. Relasjonsdatabaser
2. Fra relasjonsdatabase til modell
3. Spørringer som formler
4. Query containment

Relasjonsdatabaser

Litt uformelt er en relasjonsdatabase en samling tabeller.

Brukernavn	Navn	FDato	Stillingskode
evgenit	Evgenij Thorstensen	05.07.1987	SKO1352

Kode	Tittel	Ansvarlig	Semester
INF3170	Logikk	arild	H

Relasjonsdatabaser

Litt uformelt er en relasjonsdatabase en samling tabeller.

Brukernavn	Navn	FDato	Stillingskode
evgenit	Evgenij Thorstensen	05.07.1987	SKO1352

Kode	Tittel	Ansvarlig	Semester
INF3170	Logikk	arild	H

Informasjonen hentes via spørringer, for eksempel
"SELECT Kode, Navn FROM Kurs, Personer WHERE Ansvarlig=Brukernavn AND Brukernavn='arild';"

Kode	Navn
INF3170	Arild Waaler

Skjema

Mer presist, så består en relasjonsdatabase av to deler, et *skjema* og en *instans*.

Skjemaet er en samling relasjonssymboler med navngitte attributter, hver med sitt domene.

Personer(Brukernavn, Navn, FDato, Stillingskode)
Kurs(Kode, Tittel, Ansvarlig, Semester)

Domenet lar vi være implisitt; typisk int, string, etc.

Instans

Gitt et skjema Σ, så er en instans over Σ en mengde tupler (rader) for hvert relasjonssymbol i skjemaet, med riktig aritet og domener.

Vi kaller elementene i tuplene verdier.

Hvis vi vil, kan vi dropper navngitte attributter, og bruke indekser istedenfor (aritet må angis).

Dette likner jo veldig på noe dere har sett før.

Fra database til modell

Vi kan tolke et skjema som en signatur, ved å la relasjonene være predikatsymboler.

Så kan vi tolke en instans som en modell $\mathcal{M}$, på følgende måte:

- Alle verdier er konstanter, og tolkes som seg selv.
- For predikatsymboler lar vi $P^{\mathcal{M}} = \{\langle c_1^{\mathcal{M}}, \dots, c_n^{\mathcal{M}} \rangle \mid \langle c_1, \dots, c_n \rangle$ er i relasjonen til $P\}$.

La oss skrive $DB^{\mathcal{M}}$ for modellen vi får fra databasen DB.

Fra spørringer til formler

Siden en database svarer til en (endelig) modell, så bør formler svare til spørringer.

Gitt en lukket førsteordens formel ϕ og en database DB, så kan vi sjekke om $\mathrm{DB}^{\mathcal{M}} \models \phi$.

Dette er en *boolsk* spørring, "finnes det noe i DB som passer?".

Fra spørringer til formler

Siden en database svarer til en (endelig) modell, så bør formler svare til spørringer.

Gitt en lukket førsteordens formel ϕ og en database DB, så kan vi sjekke om $\mathrm{DB}^{\mathcal{M}} \models \phi$.

Dette er en *boolsk* spørring, "finnes det noe i DB som passer?".
For eksempel, har alle kurs en ansvarlig som står i persontabellen?

$\forall$ Kode, Tittel, Ansvarlig, Semester
Kurs(Kode, Tittel, Ansvarlig, Semester) $\rightarrow$
$\exists$ Navn, FDato, Stillingskode
Personer(Ansvarlig, Navn, FDato, Stillingskode)

$$\forall x, y, z, v. \mathrm{Kurs}(x, y, z, v) \rightarrow \exists x', y', z'. \mathrm{Personer}(z, x', y', z')$$

Spørringer som returnerer data

SQL-spørringen i begynnelsen returnerer data. Slike spørringer svarer til formler med frie variable.

Vi husker at en variabel er *fri* i en formel hvis den ikke er innefor skopet til en kvantor. Vi skriver $FV(\phi)$ for mengden av de frie variablene i ϕ.

For eksempel, hvis
$\phi = \forall x, y, v.\mathrm{Kurs}(x, y, z, v) \rightarrow \exists y', z'.\mathrm{Personer}(z, x', y', z')$, så er
$FV(\phi) = \{z, x'\}$

Intuitivt, finn alle z, x' slik at...

Spørringer med frie variabler

La DB være en database over Σ, og ϕ en formel med $FV(\phi) = \{x_1, \ldots, x_k\}$.

Vi definerer svarene til ϕ over DB som mengden av tupler $\langle c_1, \ldots, c_k \rangle$ med konstanter fra Σ slik at $DB^{\mathcal{M}} \models \phi[c_1/x_1, \ldots, c_k/x_k]$.

Vi skriver $Ans(\phi, DB)$ for denne mengden.

Spørringer med frie variabler

La DB være en database over Σ, og ϕ en formel med $FV(\phi) = \{x_1, \ldots, x_k\}$.

Vi definerer svarene til ϕ over DB som mengden av tupler $\langle c_1, \ldots, c_k \rangle$ med konstanter fra Σ slik at $DB^{\mathcal{M}} \models \phi[c_1/x_1, \ldots, c_k/x_k]$.

Vi skriver $Ans(\phi, DB)$ for denne mengden.

La DB være $\{Pab, Pbc\}$ og $\phi = \exists y(Pxy \land Pyz)$. Da er $Ans(\phi, DB) = \{\langle a, c \rangle\}$.

SPJ-spørringer

En veldig vanlig type spørring i SQL er select-project-join.

SELECT $v_1, v_2 \ldots$ FROM $T_1, T_2 \ldots$ WHERE
$Attr_1 = Attr_2, Attr_3 = Attr_4 \ldots$

Vi kombinerer flere tabeller med likhet mellom attributter.

Denne typen spørring kan vi enkelt karakterisere ved hjelp av det logiske perspektivet.

Konjunktive spørringer

En *konjunktiv spørring (CQ)* er bygd opp av konjunksjoner av atomer og eksistensielle kvantorer. Formelt:

- Alle atomære formler er med i CQ.
- Hvis ϕ og ψ er i CQ, så er $\phi \land \psi$ i CQ.
- Hvis ϕ er i CQ, så er $\exists x \phi$ i CQ.

Vi tillater altså ikke negasjon, disjunksjon, etc.

Siden samme variabel alltid tolkes likt, har vi likhet: $Pxy \land Pzv \land y = z$ er det samme som $Pxy \land Pyv$.

Oppsummering så langt

Vi kan tolke databaser som endelige modeller, ved å la hver tabell bli et predikat.

Spørringer blir da formler med frie variable.

En spesiell type spørringer er SPJ, som tilsvarer eksistenskvantifiserte konjunktive formler.

Optimisering av spørringer

Hvis vi har en komplisert spørring, eller flere spørringer å kjøre, kan deler være overflødige (redundant).

Da kan spørringen optimiseres. For eksempel er $Pxx \wedge \exists y Pxy$ ekvivalent til Pxx.

Vi ønsker å oppdage slike ting uten å evaluere spørringene.

Query containment

Definisjon (Query containment)

La ϕ og ψ være to spørringer slik at $FV(\phi) = FV(\psi)$. Vi sier at ϕ er inneholdt (contained) i ψ hvis $Ans(\phi, DB) \subseteq Ans(\psi, DB)$ for alle DB.

Med andre ord, svarene til ϕ er alltid en del av svarene til ψ.

Hvis $\phi \subseteq \psi$ og $\psi \subseteq \phi$, så er spørringene ekvivalente.

Med et logisk perspektiv kan vi lett finne en måte å teste query containment på.

Query containment, semantikk

La ϕ og ψ være to lukkede formler.

Theorem

Vi har $\phi \subseteq \psi$ hvis og bare hvis $\phi \models \psi$.

Hvorfor det? Jo, fordi alle databaser = alle endelige modeller.

$\phi \subseteq \psi$ betyr at for alle $\mathrm{DB}^{\mathcal{M}}$, hvis $\mathrm{DB}^{\mathcal{M}} \models \phi$, så $\mathrm{DB}^{\mathcal{M}} \models \psi$. Det er definisjonen av logisk konsekvens.

Hva hvis vi har frie variable?

QC med frie variable

Hvis vi har frie variable, kan vi bruke *universell tillukning*.

La ϕ være en formel med frie variable $\mathsf{FV}(\phi) = \{x_1, \ldots, x_k\}$. Den universelle tillukningen av ϕ er $\forall x_1, \ldots, x_k \phi$.

La ϕ og ψ være to formler med $\mathsf{FV}(\phi) = \mathsf{FV}(\psi) = \{x_1, \ldots, x_k\}$.

Theorem

Vi har $\phi \subseteq \psi$ *hvis og bare hvis* $\forall x_1, \ldots, x_k \phi \models \forall x_1, \ldots, x_k \psi$.

QC med frie variable, bevis del en

La ϕ og ψ være to formler med $FV(\phi) = FV(\psi) = \{x_1, \ldots, x_k\}$.

Theorem

Vi har $\phi \subseteq \psi$ *hvis og bare hvis* $\forall x_1, \ldots, x_k \phi \models \forall x_1, \ldots, x_k \psi$.

Hvorfor det? Vel, bare hvis-delen funker greit. Anta $\phi \subseteq \psi$. Hvis $DB^{\mathcal{M}} \models \forall x_1, \ldots, x_k \phi$, så er $Ans(\phi, DB)$ alle mulige k-tupler.

Siden $\phi \subseteq \psi$, så er alle disse også svarene til ψ, og da har vi at $DB^{\mathcal{M}} \models \forall x_1, \ldots, x_k \psi$.

QC med frie variable, bevis del to

La ϕ og ψ være to formler med $FV(\phi) = FV(\psi) = \{x_1, \ldots, x_k\}$.

Theorem

Vi har $\phi \subseteq \psi$ hvis og bare hvis $\forall x_1, \ldots, x_k \phi \models \forall x_1, \ldots, x_k \psi$.

For hvis-delen, anta at det ikke stemmer. Da har vi at $\forall x_1, \ldots, x_k \phi \models \forall x_1, \ldots, x_k \psi$, men ikke at $\phi \subseteq \psi$. Da finnes en DB slik at $Ans(\phi, DB) \not\subseteq Ans(\psi, DB)$.

QC med frie variable, bevis del to

La ϕ og ψ være to formler med $FV(\phi) = FV(\psi) = \{x_1, \ldots, x_k\}$.

Theorem

Vi har $\phi \subseteq \psi$ *hvis og bare hvis* $\forall x_1, \ldots, x_k \phi \models \forall x_1, \ldots, x_k \psi$.

For hvis-delen, anta at det ikke stemmer. Da har vi at $\forall x_1, \ldots, x_k \phi \models \forall x_1, \ldots, x_k \psi$, men ikke at $\phi \subseteq \psi$. Da finnes en DB slik at $Ans(\phi, DB) \not\subseteq Ans(\psi, DB)$.

Da kan vi ta $\langle c_1, \ldots, c_k \rangle \in Ans(\phi, DB) - Ans(\psi, DB)$. Siden $\langle c_1, \ldots, c_k \rangle \in Ans(\phi, DB)$, så har vi at $DB^{\mathcal{M}} \models \phi[c_1/x_1, \ldots, c_k/x_k]$, men at $DB^{\mathcal{M}} \not\models \psi[c_1/x_1, \ldots, c_k/x_k]$.

Ergo kan ikke $\forall x_1, \ldots, x_k \phi \models \forall x_1, \ldots, x_k \psi$ stemme, og vi har en motsigelse.

QC, algoritme

Teoremet gir oss følgende algoritme: For å sjekke $\phi \subseteq \psi$, sjekk $\forall x_1, \ldots, x_k \phi \vdash \forall x_1, \ldots, x_k \psi$.

QC, algoritme

Teoremet gir oss følgende algoritme: For å sjekke $\phi \subseteq \psi$, sjekk $\forall x_1, \ldots, x_k \phi \vdash \forall x_1, \ldots, x_k \psi$.

Theorem (Trakhtenbrot 1949)

For førsteordens formler ϕ og ψ er $\phi \models \psi$ ikke avgjørbart, selv over endelige modeller.

Hva med konjunktive spørringer?

QC for konjunktive spørringer

Theorem

For konjunktive spørringer ϕ *og* ψ *er* $\phi \subseteq \psi$ *avgjørbart, og NP-komplett.*

For å vise at dette problemet er avgjørbart, skal vi bruke *homomorfier*.

Siden en konjunktiv spørring ϕ bare inneholder eksistenskvantorer og konjunksjoner, kan vi skrive den som $\exists x_1, \ldots, x_n \phi$, hvor ϕ er åpen.

La $Q = \exists x_1, \ldots, x_n \phi$ og $R = \exists y_1, \ldots, y_m \psi$ være to konjunktive spørringer med $FV(Q) = FV(R)$. En *homomorfi* fra Q til R er en substitusjon θ slik at $\phi\theta$ er en delformel av ψ.

Homomorfien skal ikke endre de frie variablene i Q og R!

Homomorfiteoremet del en

Theorem (Homomorfiteoremet)

La $Q = \exists y_1, \ldots, y_n \phi$ *og* $R = \exists y_1, \ldots, y_m \psi$ *være konjunktive spørringer med* $FV(Q) = FV(R) = \{x_1, \ldots, x_k\}$. *Vi har* $Q \subseteq R$ *hvis og bare hvis det finnes en homomorfi fra* R *til* Q.

Hvis-delen: Anta at vi har en homomorfi θ. La DB være en database, og $\langle c_1, \ldots, c_k \rangle \in Ans(Q, DB)$. Da finnes det elementer $d_1, \ldots, d_n$ i $DB^{\mathcal{M}}$ slik at $DB^{\mathcal{M}} \models \phi[c_1/x_1, \ldots, c_k/x_k, d_1/y_1, \ldots, d_n/y_n]$.

Siden $\psi\theta$ er en delformel av ϕ, og begge er konjunksjoner av atomer, så må $DB^{\mathcal{M}} \models \psi\theta[c_1/x_1, \ldots, c_k/x_k, d_1/y_1, \ldots, d_n/y_n]$.

Ergo er $\langle c_1, \ldots, c_k \rangle \in Ans(R, DB)$.

Homomorfiteoremet del to

Theorem (Homomorfiteoremet)

La $Q = \exists y_1, \ldots, y_n \phi$ *og* $R = \exists y_1, \ldots, y_m \psi$ *være konjunktive spørringer med* $FV(Q) = FV(R) = \{x_1, \ldots, x_k\}$*. Vi har* $Q \subseteq R$ *hvis og bare hvis det finnes en homomorfi fra* R *til* Q*.*

Bare hvis-delen: La oss lage en database Q_{DB} ved å la hvert atom av ϕ være et tuppel i sin tabell. For eksempel, $Pxy \land Pxx$ blir tuplene $\langle x, x \rangle$ og $\langle x, y \rangle$ i P.

Homomorfiteoremet del to

Theorem (Homomorfiteoremet)

La $Q = \exists y_1, \ldots, y_n \phi$ *og* $R = \exists y_1, \ldots, y_m \psi$ *være konjunktive spørringer med* $FV(Q) = FV(R) = \{x_1, \ldots, x_k\}$. *Vi har* $Q \subseteq R$ *hvis og bare hvis det finnes en homomorfi fra* R *til* Q.

Bare hvis-delen: La oss lage en database Q_{DB} ved å la hvert atom av ϕ være et tuppel i sin tabell. For eksempel, $Pxy \wedge Pxx$ blir tuplene $\langle x, x \rangle$ og $\langle x, y \rangle$ i P.

Hva er $Ans(Q, Q_{DB})$? Jo, $\{\langle x_1, \ldots, x_k \rangle\}$. Siden $Q \subseteq R$, må $\langle x_1, \ldots, x_k \rangle \in Ans(R, Q_{DB})$. Men da finnes det konstanter $c_1, \ldots, c_m$ i Q_{DB} slik at $Q_{DB}^{\mathcal{M}} \models \psi[c_1/y_1, \ldots, c_m/y_m]$.

Vi skal vise at $[c_1/y_1, \ldots, c_m/y_m]$ er en homomorfi fra R til Q.

Homomorfiteoremet del to, fortsettelse

Theorem (Homomorfiteoremet)

La $Q = \exists y_1, \ldots, y_n \phi$ *og* $R = \exists y_1, \ldots, y_m \psi$ *være konjunktive spørringer med* $FV(Q) = FV(R) = \{x_1, \ldots, x_k\}$. *Vi har* $Q \subseteq R$ *hvis og bare hvis det finnes en homomorfi fra* R *til* Q.

Hva er $Ans(Q, Q_{DB})$? Jo, $\{\langle x_1, \ldots, x_k \rangle\}$. Siden $Q \subseteq R$, må $\langle x_1, \ldots, x_k \rangle \in Ans(R, Q_{DB})$. Men da finnes det konstanter $c_1, \ldots, c_m$ i Q_{DB} slik at $Q_{DB}^{\mathcal{M}} \models \psi[c_1/y_1, \ldots, c_m/y_m]$.

Disse konstantene er termer fra Q. Siden ψ er en konjunksjon av atomer, betyr dette at Q_{DB} inneholder hvert atom i $\psi[c_1/y_1, \ldots, c_m/y_m]$.

Pga. måten vi konstruerte Q_{DB} på, betyr dette at også Q inneholder alle disse atomene, og ergo er $\psi[c_1/y_1, \ldots, c_m/y_m]$ en delformel av ϕ.

QC for konjunktive spørringer, kompleksitet

Siden det er endelig mange homomorfier, er problemet avgjørbart.

Siden det å sjekke om en substitusjon er en homomorfi kan gjøres i lineær tid, er vi i klassen NP.

NP-hardhet følger av hardhet for homomorfi-problemet.

QC for konjunktive spørringer, konklusjon

For konjunktive spørringer ϕ og ψ med like frie variable er følgende påstander ekvivalente:

- $\phi \subseteq \psi$
- $\phi \models \psi$
- Det finnes en homomorfi fra ψ til ϕ

Den i midten viste vi for arbitrære formler.

Den siste følger av homomorfiteoremet.

Databaser som spørringer

Det å representere en konjunktiv spørring som en database kan snus.

Vi kan representere en database DB som en boolsk konjunktiv spørring, DB_Q, ved lage en stor konjunksjon av alle tupler i alle relasjoner.

$$DB_Q = \bigwedge_{P \text{ relasjon i DB}} \bigwedge_{\langle c_1, \ldots, c_k \rangle \in P} P(c_1, \ldots, c_n)$$

Databaser som spørringer

Det å representere en konjunktiv spørring som en database kan snus.

Vi kan representere en database DB som en boolsk konjunktiv spørring, DB_Q, ved lage en stor konjunksjon av alle tupler i alle relasjoner.

$$\mathrm{DB}_Q = \bigwedge_{P \text{ relasjon i DB}} \bigwedge_{\langle c_1,\ldots,c_k\rangle \in P} P(c_1, \ldots, c_n)$$

Da kan vi vise følgende: Hvis ϕ er en boolsk konjunktiv spørring, så har vi $\mathrm{DB}^{\mathcal{M}} \models \phi$ hvis og bare hvis $\mathrm{DB}_Q \subseteq \phi$.

Databaser som spørringer

Det å representere en konjunktiv spørring som en database kan snus.

Vi kan representere en database DB som en boolsk konjunktiv spørring, DB_Q, ved lage en stor konjunksjon av alle tupler i alle relasjoner.

$$\mathrm{DB}_Q = \bigwedge_{P \text{ relasjon i DB}} \; \bigwedge_{\langle c_1,\ldots,c_k\rangle \in P} P(c_1,\ldots,c_n)$$

Da kan vi vise følgende: Hvis ϕ er en boolsk konjunktiv spørring, så har vi $\mathrm{DB}^{\mathcal{M}} \models \phi$ hvis og bare hvis $\mathrm{DB}_Q \subseteq \phi$.

Bevis: Ukeopgave!

Bonus: NP-hardhet for konjunktive spørringer

Definisjon (Trefargeproblemet)

Gitt en urettet graf G, kan den farges med tre farger slik at nabonoder har ulik farge?

Vi skal lage en spørring og en database som uttrykker dette problemet.

Bonus: NP-hardhet for konjunktive spørringer

Definisjon (Trefargeproblemet)

Gitt en urettet graf G, kan den farges med tre farger slik at nabonoder har ulik farge?

Vi skal lage en spørring og en database som uttrykker dette problemet.

Databasen inneholder en binær relasjon E med seks tupler: $\langle r, g \rangle$, $\langle r, b \rangle$, og $\langle b, g \rangle$, samt deres symmetrier.

Spørringen lager vi ut fra G. Hver node er en variabel, og hver kant $\langle v, w \rangle$ er et atom $E(v, w)$. Formelen blir da

$$\exists V \bigwedge_{\langle v,w \rangle \in E} E(v, w)$$