UiO : Institutt for informatikk
Det matematisk-naturvitenskapelige fakultet

# Relasjonsmodellen og normaliseringsteori

Til bruk i INF3100

Ragnar Normann
5. utgave, mai 2013

# Innhold

# Forord

Dette er femte utgave av et lite kompendium som gir et konsentrat av kjernestoffet rundt relasjonsmodellen. Kompendiet er ment som en hjelp ved repetisjon og som et oppslagsverk over definisjoner av noen sentrale begreper i pensum i INF3100.

Kompendiet er ikke noen lærebok, og det er ikke beregnet for selvstudium. Det skal være et *supplement* til læreboken

> Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom:
> Database Systems. The Complete Book.
> Second Edition. Pearson Education 2009.
> ISBN 0-13-1135428-0 978-0-13-135428-9

og ikke en *erstatning* for den. Kompendiet er praktisk talt uten eksempler, og det konsentrerer seg om selve teorien, og ikke om anvendelsen av teorien på praktiske oppgaver. Eksempler og oppgaver finnes i læreboken.

Fremstillingen i kompendiet er noe mer stringent enn lærebokens. Denne 5. utgaven inneholder en del nye forskningsresultater fra årene etter at læreboken kom ut. Dette er stoff som *ikke* er pensum i INF3100.

Det er tre personer jeg ønsker å takke for kommentarer og råd under mitt opprinnelige arbeid med kompendiet: Rune Djurhuus hvis konstruktive kritikk har bidratt sterkt til kvaliteten på det faglige innholdet i kompendiet, Liv Thrap-Meyer som har påpekt og rettet språklige feil og unøyaktigheter, og Dag F. Langmyhr som har vært en tålmodig veileder og rådgiver under denne min første seriøse anvendelse av LaTeX.

Etter at første utgave kom våren 1995, har en rekke studenter og lærere kommet med kommentarer og forslag til forbedringer.

Jeg takker Bjørn Kristoffersen som påviste unøyaktigheter i flere av definisjonene av relasjonsoperatorene i kapittel 3 i første utgave.

Jeg vil også takke Dr. Peter Creasy ved University of Queensland som gjennom diskusjoner fikk meg til å legge terminologien rundt relasjonsmodellen nærmere lærebokens og til å inkludere Fagins opprinnelige definisjon av PJNF i fjerde utgave.

Jeg takker Ellen Munthe-Kaas for nyttige samtaler og gode innspill under mitt arbeid med denne utgaven.

Takk til alle andre som har kommet med konstruktiv kritikk!

Gaustadbekkdalen, 7. mai 2013

Ragnar Normann

# 1 Metaspråket

Dette heftet skal gi en oppsummering av relasjonsmodellen som er den matematiske modellen bak relasjonsdatabaser. Før vi tar fatt på selve relasjonsmodellen, skal vi se litt på det språket vi skal bruke til å beskrive modellen (et slikt beskrivelsesspråk kalles et *metaspråk*). I tillegg til vanlig norsk skal vi bruke 1. ordens logikk (som består av Boolske uttrykk og kvantorer over elementer) og elementær mengdelære. Dette burde være velkjent stoff, men for fullstendighetens skyld skal vi kort repetere det.

Først skal vi se på en type matematiske strukturer som vi får bruk for i beskrivelsen av Boolske uttrykk og relasjonsmodellen:

## 1.1 Algebraer

En algebra består av en mengde *operander* og en endelig mengde *operatorer* over disse.

Operatorene er funksjoner som tar en ordnet mengde operander, kalt *argumentene* til operatoren, og produserer en operand, kalt *resultatet* til operatoren.

Hver operator tar et fast antall argumenter, kalt *ariteten* til operatoren. Har operatoren ett argument, kalles den unær. Har den to argumenter, kalles den binær; har den tre, kalles den ternær. Vi vil bare få bruk for unære og binære operatorer.

Operatorer uten argumenter kalles gjerne konstante operatorer. Vi skal imidlertid ikke betrakte konstanter som operatorer, men heller skille mellom konstante og variable operander.

Vi skal vanligvis bruke infix-notasjon for binære operatorer, dvs at vi setter operatoren mellom operandene. Dette medfører at vi må bruke parenteser for å beskrive kompliserte uttrykk. Et eksempel er uttrykket $X \circ Y \diamond Z$ der $X$, $Y$ og $Z$ er operander, og $\circ$ og $\diamond$ er to binære operatorer. Hvis vi ikke har noen definert rangordning mellom operatorene, må vi skrive $(X \circ Y) \diamond Z$ eller $X \circ (Y \diamond Z)$ for å presisere hva vi mener.

Generelt gjelder at alle lovlige algebrauttrykk kan brukes som operand ved å sette en parentes rundt uttrykket.

## 1.2 1. ordens logikk

### 1.2.1 Boolsk algebra

Boolsk algebra har to konstanter, $T$ (True) og $F$ (False), kalt *sannhetsverdier*, og to (fundamentale) operatorer: $\neg$ (negasjon (ikke)) som er unær, og $\wedge$ (konjunksjon (og)) som er binær.

Disse er definert henholdsvis ved at $\neg T$ gir $F$ og $\neg F$ gir $T$, og ved at $T \wedge T$ gir $T$, mens $F \wedge T$, $T \wedge F$ og $F \wedge F$ alle gir $F$.

Ut fra disse to operatorene kan vi definere flere binære operatorer. Her tar vi bare med de to vi skal få bruk for:

Vi definerer disjunksjon (eller), $u \vee v$, som $\neg(\neg u \wedge \neg v)$, og implikasjon, $u \Rightarrow v$, som $\neg u \vee v$.

Det er verdt å merke seg at $\neg$ er den eneste unære Boolske operatoren (vi ser da bort fra *identitetsoperatoren*, den som ikke gjør noe). Alle andre Boolske operatorer er binære.

### 1.2.2 Boolske atomer og uttrykk

Et Boolsk *atom* er et uttrykk $P$ som ikke inneholder noen Boolske operatorer, og som evaluerer til en sannhetsverdi når alle (frie) variable i $P$ blir gitt verdier. Alle Boolske variable og sannhetsverdiene $T$ og $F$ er Boolske atomer. Et annet eksempel er uttrykket $x = 2$ som evaluerer til $T$ hvis $x$ får verdien 2, og til $F$ hvis $x$ får en hvilken som helst annen verdi (som 3 eller Paris).

Et Boolsk uttrykk er bygget opp av Boolske atomer ved hjelp av Boolske operatorer og parenteser. Vi skal definere hva som er lovlige Boolske uttrykk ved å beskrive hva som er lovlige operander for de fire Boolske operatorene ovenfor. Det er verdt å merke seg at definisjonen er induktiv i den forstand at et vilkårlig komplisert Boolsk uttrykk kan ta plassen til et atom hvis vi setter en parentes rundt uttrykket.

**$\neg$-operander** Alle Boolske atomer er $\neg$-operander.
Hvis $u$ er en $\neg$-operand, så er $\neg u$ en $\neg$-operand.
Hvis $P$ er et Boolsk uttrykk, så er $(P)$ en $\neg$-operand.
Merk: $\neg\neg u$ skal tolkes som $\neg(\neg u)$.

**$\wedge$-operander** Alle $\neg$-operander er $\wedge$-operander.
Hvis $u$ og $v$ er $\wedge$-operander, så er $u \wedge v$ en $\wedge$-operand.

**$\vee$-operander** Alle $\wedge$-operander er $\vee$-operander.
Hvis $u$ og $v$ er $\vee$-operander, så er $u \vee v$ en $\vee$-operand.

**Boolske uttrykk** Alle $\vee$-operander er Boolske uttrykk.
Hvis $u$ og $v$ er $\vee$-operander, så er $u \Rightarrow v$ et Boolsk uttrykk.

Denne definisjonen gir en rangordning mellom de Boolske operatorene som reduserer den nødvendige bruken av parenteser. Rangordningen er $\neg$, $\wedge$, $\vee$ og $\Rightarrow$, der $\neg$ binder sterkest og $\Rightarrow$ svakest. Det er fullt tillatt å sette ekstra parenteser for å øke lesbarheten.

### 1.2.3 Ekvivalenser, tautologier og DeMorgans lover

To Boolske uttrykk $P$ og $Q$ som alltid evaluerer til samme sannhetsverdi, uansett hvilke verdier vi gir variablene i atomene, kalles *ekvivalente*, og vi skriver det $P \equiv Q$.

Et Boolsk uttrykk $P$ som alltid er sant, dvs at $P \equiv T$, kalles en *tautologi*.

Er $P$ alltid falskt, dvs at $P \equiv F$, kalles $P$ en *selvmotsigelse*.

Eksempler: $\neg u \vee u$ er en tautologi, mens $\neg u \wedge u$ er en selvmotsigelse.

Viktige eksempler på ekvivalenser er de assossiative lovene for $\wedge$ og $\vee$:

$$(u \wedge v) \wedge w \equiv u \wedge (v \wedge w) \text{ og } (u \vee v) \vee w \equiv u \vee (v \vee w)$$

(som er nødvendige og tilstrekkelige til å sikre at vår definisjon av Boolske uttrykk er veldefinert), de kommutative lovene for $\wedge$ and $\vee$:

$$u \wedge v \equiv v \wedge u \text{ og } u \vee v \equiv v \vee u$$

og de distributive lovene for $\wedge$ and $\vee$:

$$u \wedge (v \vee w) \equiv u \wedge v \vee u \wedge w \text{ og } u \vee v \wedge w \equiv (u \vee v) \wedge (u \vee w)$$

Andre viktige eksempler er $\neg\neg u \equiv u$ og DeMorgans lover:

$$\neg(u \vee v) \equiv \neg u \wedge \neg v \text{ og } \neg(u \wedge v) \equiv \neg u \vee \neg v$$

Ekvivalensene ovenfor kan bevises direkte fra definisjonene av $\neg$, $\wedge$ og $\vee$.

### 1.2.4 Kvantorer

Uttrykket $\exists x(P(x))$ er $T$ hvis det finnes (minst) en verdi av $x$ som gjør at $P(x)$ evaluerer til $T$. $\exists$ kalles *eksistenskvantoren*.

Uttrykket $\forall x(P(x))$ er $T$ hvis $P(x)$ evaluerer til $T$ uansett hvilken verdi vi gir $x$. $\forall$ kalles *allkvantoren* eller *universalkvantoren*.

Siden $\forall x(P(x))$ betyr at $P(x)$ skal holde for *alle* $x$ i universet, er det vanlig å dele $P$ i to deluttrykk $P_1$ og $P_2$ slik: $P = \neg P_1 \vee P_2$. Det betyr at allkvantoruttrykk vanligvis skrives på en av de to ekvivalente formene:

$$\forall x(\neg P_1(x) \vee P_2(x)) \text{ eller } \forall x(P_1(x) \Rightarrow P_2(x))$$

$P_1$ blir brukt til å skille ut den delen av universet som interesserer oss, mens $P_2$ er den betingelsen som de interessante tilfellene må oppfylle.

Vi avslutter med to nyttige ekvivalenser:

$$\neg\forall x(P(x)) \equiv \exists x(\neg P(x)) \text{ og } \neg\exists x(P(x)) \equiv \forall x(\neg P(x))$$

### 1.2.5 Frie og bundne variable

Kort sagt kan vi si at variable som er nevnt i en kvantor, er bundet av kvantoren. Variable som ikke er bundet, er fri.

Eksempel: I uttrykket $\forall x \exists y (P(x,z) \vee Q(z,y))$ er $x$ og $y$ bundet og $z$ fri.

Parentesen etter kvantoren bestemmer rekkevidden til kvantoren, dvs i hvilket område variabelen er bundet. Formelt kan vi altså «bruke» navnet på en bundet variabel om igjen utenfor rekkevidden til kvantoren som vist i følgende eksempel:

Uttrykkene $\exists x(P(x)) \wedge \exists y(Q(y))$ og $\exists x(P(x)) \wedge \exists x(Q(x))$ uttrykker det samme. I det siste uttrykket bruker vi altså symbolet $x$ om to forskjellige variable. Faren for å forveksle dette med uttrykket $\exists x(P(x) \wedge Q(x))$, som *ikke* er det samme, gjør at vi bør unngå slik gjenbruk av variabelnavn.

Helt til slutt en kommentar til dem som lurer på hva vi mener med at vi begrenser oss til 1. ordens logikk: Det betyr at det universet kvantorene går over (dvs hvilke verdier vi kan sette inn for våre (bundne) variable), består av konstanter. Kvantorene kan altså ikke tas over *mengder*.

## 1.3 Mengdelære

Vi skal her nøye oss med en kort repetisjon av de viktigste begrepene innen elementær mengdelære. Vi skal ikke forsøke å definere begrepene *element* og *mengde* skikkelig, men overlater det til filosofene. For oss er det nok å betrakte et element som en konstant og en mengde som en samling elementer hvor vi kan avgjøre om et gitt element er med i mengden eller ikke.

Hvis $a$ er et element, og $A$ er en mengde, skriver vi $a \in A$ for å betegne at $a$ er et element i $A$. Merk at $a \in A$ er et Boolsk atom som er $T$ hvis $a$ er et element i mengden $A$, og $F$ ellers. Merk også at vi ofte bruker skrivemåten $a \notin A$ i stedet for $\neg a \in A$.

Vi skal stort sett bruke to måter å angi mengder på. Den første er listeformen

$$A = \{a_1, \ldots, a_n\} \text{ der } a_k \in A \text{ for } 1 \leq k \leq n$$

som vanligvis bare brukes for endelige mengder.

Den andre er ved hjelp av mengdebyggersymbolet

$$A = \{a \mid P(a)\}$$

der $P$ er et uttrykk i 1. ordens logikk som kalles *mengdepredikatet*, og som definerer $A$ ved at $a \in A \equiv P(a)$. Det er verdt å merke seg at variabelen $a$ godt kan være en liste av flere variable som i dette eksemplet:

$$\{t, u, v \mid P(t, u, v)\}$$

Merk også at $t$, $u$ og $v$ er de eneste frie variable i mengdepredikatet $P$. Alle andre variable i $P$ må være bundet. Dette er et eksempel på følgende

**Regel**

I et mengdepredikat skal alle variable som står til venstre for | i mengdebyggeren, være frie. Alle andre variable i mengdepredikatet skal være bundet av kvantorer.

### 1.3.1 Mengdeoperatorer

I mengdealgebra er operandene mengder. Vi definerer tre binære operatorer:

- $A \cup B = \{a \mid a \in A \vee a \in B\}$ (Union)
- $A \cap B = \{a \mid a \in A \wedge a \in B\}$ (Snitt)
- $A \setminus B = \{a \mid a \in A \wedge a \notin B\}$ (Differanse)

### 1.3.2 Sammenligningsoperatorer

La $A$ og $B$ være mengder. Da definerer vi inklusjon og likhet ved:

- $A \subseteq B$ betyr at $a \in A \Rightarrow a \in B$. Videre skal $A \supseteq B$ bety det samme som $B \subseteq A$.
- $A = B$ betyr at $A \subseteq B \wedge A \supseteq B$, dvs at $a \in A \Leftrightarrow a \in B$.
- $A \subset B$ betyr at $A \subseteq B \wedge A \neq B$, og $A \supset B$ betyr at $B \subset A$.

### 1.3.3 $\varnothing$ og potensmengder

Symbolet $\varnothing$ betegner den tomme mengden, dvs mengden uten elementer. Vi har altså $\varnothing = \{\} = \{a \mid F\}$. Hvis $A$ er en mengde, definerer vi *potensmengden* til A som $\mathcal{P}(A) = \{X \mid X \subseteq A\}$, mengden av delmengder i $A$.

# 2 Relasjoner

Relasjonsmodellen (E. F. Codd 1970) er en begrepsmessig modell i henhold til 3-skjemaarkitekturen.

Intuisjonen bak modellen er at alle data er organisert i navngitte tabeller, kalt *relasjoner*, og at disse tabellene er de eneste datastrukturene i det begrepsmessige skjemaet. Radene i tabellene kalles *forekomster*, og de utgjør selve dataene i strukturen. Kolonnene har navn og kalles *attributter*.

Hver kolonne (attributt) har et *domene* (= mengde av mulige verdier) tilordnet seg.

Her følger mer presise definisjoner av begrepene ovenfor:

## 2.1 Domener

Et domene, $D$, er en mengde atomære verdier. At $d \in D$ er atomær, betyr at $d$ ikke selv er en mengde.

Dersom $D = \{d \mid \Delta(d)\}$, kalles mengdepredikatet $\Delta$ *domenerestriksjonen* for $D$. Vi skal forutsette at $\Delta$ er et uttrykk i 1. ordens logikk.

Et domene svarer internt til en *datatype* (et dataformat) og eksternt til et *format* (presentasjonsformat).

## 2.2 Relasjonsskjemaer og attributter

Et relasjonsskjema, $S$, er en endelig mengde $\{A_1, \ldots, A_n\}$ av navn, kalt *attributter*, hvor det til hvert attributt, $A_i$ $(1 \leq i \leq n)$, er tilordnet et domene, $dom(A_i)$. $n$ kalles kalles *graden*, eller *ariteten*, til $S$. $A_k$ kalles et *rollenavn* for $dom(A_k)$ i $S$ $(k \in \{1, \ldots, n\})$.

Siden $S$ er en mengde, har vi at $j \neq k \Rightarrow A_j \neq A_k$

**Merknad om valg av formalisme:**

Siden alle attributter må være tilknyttet eksakt ett domene, virker det kanskje mer naturlig å definere attributter som par $(A, D)$ der $A$ er et navn og $D$ er et domene med den tilleggsbetingelsen at navnene er entydige innenfor relasjonsskjemaet. Denne alternative definisjonen brukes i noe litteratur, og da er $A : D$ den vanligste notasjonen for et attributt med navn $A$ og domene $D$. Her skal vi velge å bare bruke navnet som definisjon av et attributt. Årsaken til dette valget er at noen av definisjonene nedenfor blir enklere da.

## 2.3 Komponenter

En komponent $X$ i et relasjonsskjema $S$ er en delmengde av $S$. $X$ kalles en *atomær komponent* hvis $X$ inneholder eksakt ett attributt.

Merk at $\emptyset$ (den tomme mengden) og $S$ selv er komponenter i $S$, og at komponenter selv er relasjonsskjemaer (de er mengder av attributter).

## 2.4 Relasjoner og forekomster

En relasjon, $R$, består av et relasjonsskjema $\mathbf{R}(R) = \{A_1, \ldots, A_n\}$, kalt *intensjonen* til $R$, og en endelig mengde $r(R)$, kalt *ekstensjonen* til $R$, der hver $t \in r(R)$ er en funksjon fra $\mathbf{R}(R)$ inn i unionen av alle domener tilknyttet attributtene i $\mathbf{R}(R)$ slik at $t(A_k) \in dom(A_k)$ for $1 \leq k \leq n$. (Egentlig burde vi ikke ha skrevet at $R$ er en relasjon, men at $R$ er *navnet* på en relasjon. Men som alle andre, skal vi skrive «La $R$ være en relasjon» i stedet for det pinlig korrekte «Gitt en relasjon med navn $R$».)

Hver $t \in r(R)$ kalles en *forekomst* i $R$. Vi skriver ofte forekomstene på formen $t = (v_1, \ldots, v_n)$ (der altså $v_k = t(A_k) \in dom(A_k)$ for $1 \leq k \leq n$). Denne notasjonen gjør at forekomstene ofte kalles *tupler*.

Merk at ordet *relasjon* brukes om litt forskjellige begreper i relasjonsmodellen og matematikk. Sammenhengen er at hvis $R$ er en relasjon i relasjonsmodellen, så er ekstensjonen til $R$, $r(R)$, en endelig (matematisk) relasjon over $dom(A_1), \ldots, dom(A_n)$.

## 2.5 Nilmerker og nilverdier

I det praktiske liv kan det hende at en forekomst $t$ ikke har noen verdi i et attributt $X$. Vi sier at $t$ er **nil** i $A$ («nil» er latin for «tom»). Siden det ikke er noe som heter «tomme» bit, må vi lagre noe i maskinen, og vi kaller det vi lagrer for å markere manglende data et *nilmerke*. Ved å ha flere forskjellige nilmerker kan vi lagre *årsaken* til at data mangler.

Lovlige nilmerker og deres betydning er en del av det begrepsmessige skjemaet. Fornuftig bruk av nilmerker gir økt uttrykkskraft i relasjonsmodellen.

Som følgende eksempel viser, får vi imidlertid problemer ved bruk av **nil** i sammenligninger: Anta vi lagrer (informasjon om) personer og at attributtet «Barn» inneholder antall barn de har.

Hvis vi vil finne alle personer som har barn, tester vi på at Barn>0 ($u$).
Vil vi finne alle personer som ikke har barn, tester vi på at Barn=0 ($\neg u$).

Forekomster hvor Barn er **nil** blir ikke med i noen av løsningsmengdene. Altså er ikke $\neg u \vee u$ en tautologi, noe som strider mot Boolsk algebra.

For å kunne håndtere **nil** må vi ha (minst) en sannhetsverdi i tillegg til $T$ (true) og $F$ (false). Slike ekstra sannhetsverdier kalles *nilverdier*. Det enkleste eksemplet er en 3-verdilogikk hvor $M$ (maybe) er innført i tillegg til $T$ og $F$. Problemet med slike flerverdilogikker er at de vante slutningsreglene (som DeMorgans lover) ikke nødvendigvis er riktige lenger.

Merk forskjellen på nilmerker og nilverdier. Nilverdier får vi når vi sammenligner nilmerker med hverandre eller med vanlige verdier. Vi kan godt ha mange nilmerker selv om vi bare har én nilverdi.

Det er utført mye forskning på håndteringen av manglende informasjon i databaser, og den viser at det kompliserer teorien vesentlig hvis vi tillater nilmerker i tuplene våre. Derfor skal vi se helt bort fra nilmerker og nilverdier i vår teoretiske håndtering av relasjonsmodellen. Det betyr at vi skal forutsette at forekomstene våre er *totale funksjoner*, dvs at en forekomst alltid har en verdi i alle attributter.

### 2.6 Noen konvensjoner

La $R$ være en relasjon.

- At $t$ er en forekomst (tuppel) i $R$, betyr at $t \in r(R)$, ekstensjonen til $R$.
- At $A$ er et attributt i $R$, betyr at $A \in \mathbf{R}(R)$, intensjonen til $R$.
- At $X$ er en komponent i $R$, betyr at $X \subseteq \mathbf{R}(R)$.
- La $t$ være en forekomst og $X = \{A_1, \ldots, A_p\}$ en komponent i $R$. Da skriver vi $t[X]$ for verdien av $t$ på $X$, dvs at $t[X] = (t(A_1), \ldots, t(A_p))$. Dette viser at forekomstene i $R$ kan oppfattes som forekomster i $X$ også (tilsvarer matematisk å restriktere funksjonen $t$ til $X$). Noe slurvete skriver vi $t[X]$ for selve forekomsten i $X$ også (det matematisk korrekte er $t|_X$ som er symbolet for funksjonen $t$ restriktert til $X$).
- La $X$ og $Y$ være komponenter i $R$. Da skriver vi $XY$ for unionen av $X$ og $Y$. Siden $XY$ selv er en komponent, gir triviell induksjon at denne notasjonen kan generaliseres til vilkårlig mange komponenter. F.eks. er $XYZUV = X \cup Y \cup Z \cup U \cup V$.

### 2.7 Unionkompatibilitet

To attributter, $A$ og $B$, kalles *kompatible* (*sammenlignbare*) hvis $dom(A) = dom(B)$.

To komponenter $X = \{A_1, \ldots, A_n\}$ og $Y = \{B_1, \ldots, B_m\}$ kalles *unionkompatible* hvis $m = n$ og $dom(A_k) = dom(B_k)$ for $1 \leq k \leq n$.

To relasjoner kalles unionkompatible hvis deres relasjonsskjemaer er det.

## 3 Relasjonsalgebra

I relasjonsalgebra er operandene relasjoner. Det betyr at operatorene tar relasjoner som argumenter og produserer en ny relasjon som resultat.

## 3.1 Fundamentale relasjonsoperatorer

Relasjonsalgebraen har seks fundamentale operatorer. Tre av dem, renavning, seleksjon og projeksjon, er unære; de andre tre, union, differanse og produkt, er binære.

Det at de er fundamentale, betyr at ingen av dem kan defineres ved hjelp av de andre fem. (De kan heller ikke defineres på en fornuftig måte ved hjelp av de avledede operatorene nedenfor.)

For å definere en relasjonsoperator må vi kunne bruke argumentverdiene til å definere både intensjonen og ekstensjonen til resultatet.

Intensjonen lar seg definere ut fra attributtene i operanden(e). For å definere (noen av) ekstensjonene bruker vi:

### 3.1.1 $\vartheta$-notasjonen

For relasjoner er verdiene i forekomstene begrenset til domenene. Domenene begrenser med andre ord det universet variablene i våre Boolske uttrykk kan hente sine verdier fra.

I tillegg kan vi bare sammenligne verdier fra samme domene. (Spørsmål som: «Er 17år < 28kg ?» er meningsløse.)

Vi skal bruke symbolet $\vartheta$ om et Boolsk uttrykk basert på atomer på formen $x\theta y$ der $x$ og $y$ kommer fra samme domene og $\theta \in \{=, \neq, <, >, \leq, \geq\}$.

$\vartheta$ representerer altså det mest generelle Boolske uttrykk vi kan ha i relasjonsalgebra.

### 3.1.2 Renavning

La $R$ være en relasjon, med skjema $\mathbf{R}(R) = \{A_1, \ldots, A_n\}$. Da definerer vi *renavningsfunksjonen* fra $R$ til $S$ som funksjonen $\rho_{S(B_1,\ldots,B_n)}$ gitt ved

$$\rho_{S(B_1,\ldots,B_n)}(A_k) = B_k \text{ der } dom(B_k) = dom(A_k) \text{ for } 1 \leq k \leq n$$

$$r(S) = \{t \circ \rho_{R(A_1,\ldots,A_n)} \mid t \in r(R)\}$$

Det siste betyr at forekomstene i $S$ er definert ved at vi for hver forekomst $t$ i $R$ har en forekomst $u$ i $S$ gitt ved

$$u(B_k) = t \circ \rho_{R(A_1,\ldots,A_n)}(B_k) = t(\rho_{R(A_1,\ldots,A_n)}(B_k)) = t(A_k)$$

Hvis vi tolker forekomstene som tupler, får vi $r(S) = r(R)$.

Merk at $\rho_{S(B_1,\ldots,B_n)} : R \to S$ og $\rho_{R(A_1,\ldots,A_n)} : S \to R$ er inverse funksjoner, dvs. at $\rho_{R(A_1,\ldots,A_n)} \circ \rho_{S(B_1,\ldots,B_n)}$ er identitetsfunksjonen på $R$.

Hvis vi bare vil endre relasjonsnavnet (og ikke attributtene), kan vi skrive $\rho_S(R)$ i stedet for $\rho_{S(A_1,\ldots,A_n)}(R)$

### 3.1.3 Seleksjon

Et *seleksjonspredikat* $\vartheta$ er et Boolsk uttrykk hvor hvert atom er på en av formene $A\theta B$, $A\theta c$ eller $c\,\theta A$ der $A$ og $B$ er kompatible attributter i $\mathbf{R}(R)$, $c \in dom(A)$ og $\theta \in \{=, \neq, <, >, \leq, \geq\}$.

La $R$ være en relasjon og la $\vartheta$ være et seleksjonspredikat over $R$.
Da definerer vi den seleksjonen av $R$ som tilfredsstiller $\vartheta$, $\sigma_\vartheta(R)$, ved

$$\mathbf{R}(\sigma_\vartheta(R)) = \mathbf{R}(R)$$

$$r(\sigma_\vartheta(R)) = \{t \mid t \in r(R) \wedge \vartheta(t)\}$$

Siden vi får forskjellige seleksjoner avhengig av hvilken $\vartheta$ vi har, har vi strengt tatt ikke bare én seleksjonsoperator, men en hel familie av dem. Det er likevel vanlig å betrakte alle seleksjonsoperatorer som én unær operator.

### 3.1.4 Projeksjon

La $R$ være en relasjon og la $S$ være en komponent i $R$.
Da definerer vi projeksjonen av $R$ på $S$, $\pi_S(R)$, ved

$$\mathbf{R}(\pi_S(R)) = S$$

$$r(\pi_S(R)) = \{t[S] \mid t \in r(R)\}$$

Også her har vi strengt tatt en familie unære operatorer, én for hver verdi av $S$.

### 3.1.5 Union

La $R$ og $S$ være to unionkompatible relasjoner.
Da definerer vi unionen av $R$ og $S$, $R$ union $S$, ved

$$\mathbf{R}(R \text{ union } S) = \mathbf{R}(R)$$

$$r(R \text{ union } S) = r(R) \cup \{t \circ \rho_{S(B_1,\ldots,B_n)} \mid t \in r(S)\}$$

der $\rho_{S(B_1,\ldots,B_n)}$ renavner $R$ til $S$. Merk at union ikke er kommutativ.

### 3.1.6 Differanse

La $R$ og $S$ være to unionkompatible relasjoner.
Da definerer vi differansen mellom $R$ og $S$, $R$ minus $S$, ved

$$\mathbf{R}(R \text{ minus } S) = \mathbf{R}(R)$$

$$r(R \text{ minus } S) = r(R) \setminus \{t \circ \rho_{S(B_1,\ldots,B_n)} \mid t \in r(S)\}$$

der $\rho_{S(B_1,\ldots,B_n)}$ renavner $R$ til $S$.

### 3.1.7 Produkt

La $R$ og $S$ være to relasjoner med skjemaer $\mathbf{R}(R) = \{A_1, \ldots, A_n\}$ og $\mathbf{R}(S) = \{B_1, \ldots, B_m\}$.

Dersom skjemaene overlapper, dvs. at det finnes $i$ og $k$ med $A_i = B_k$, må vi forandre navn på ett av disse attributtene. Vi velger konsekvent å forandre $B_k$ til $B'_k$, dvs at vi lager en renavningsfunksjon $\rho_{S'(B'_1,\ldots,B'_m)}$ fra $S$ til $S'$ slik at

$$\{A_1, \ldots, A_n\} \cap \{B'_1, \ldots, B'_m\} = \varnothing$$

Da kan vi definere produktet av $R$ og $S$, $R \,\mathsf{prod}\, S$, ved

$$\mathbf{R}(R \,\mathsf{prod}\, S) = \{A_1, \ldots, A_n, B'_1, \ldots, B'_m\}$$

$$r(R \,\mathsf{prod}\, S) = \{t \mid t[\mathbf{R}(R)] \in r(R) \wedge t \circ \rho_{S'(B'_1,\ldots,B'_m)}[\mathbf{R}(S)] \in r(S)\}$$

## 3.2 Avledbare relasjonsoperatorer

Det er vanlig å inkludere tre avledbare operatorer i relasjonsalgebraen: snitt, divisjon og join. Vi skal imidlertid definere fire avledbare operatorer, idet vi skal skille mellom «vanlig» join og naturlig join.

### 3.2.1 Snitt

La $R$ og $S$ være to unionkompatible relasjoner.
Da definerer vi snittet mellom $R$ og $S$ ved

$$R \,\mathsf{snitt}\, S = R \,\mathsf{minus}\, (R \,\mathsf{minus}\, S)$$

Merk at snitt ikke er kommutativ.

### 3.2.2 Divisjon

La $R$ og $S$ være to relasjoner der $\mathbf{R}(S)$ er en komponent i $R$.
Sett $X = \mathbf{R}(R) \setminus \mathbf{R}(S)$. Da definerer vi $R$ dividert på $S$ ved

$$R \,\mathsf{div}\, S = \pi_X(R) \,\mathsf{minus}\, \pi_X((\pi_X(R) \,\mathsf{prod}\, S) \,\mathsf{minus}\, R)$$

### 3.2.3 $\vartheta$-join og ekvijoin

La $R$ og $S$ være to relasjoner. Skift, om nødvendig, navn på (noen av) attributtene i $B$ slik at alle attributtene i $R$ og $S$ har forskjellige navn. Da definerer vi en $\vartheta$-join av $R$ og $S$ ved

$$R \bowtie_\vartheta S = \sigma_\vartheta(R \,\mathsf{prod}\, S)$$

der join-predikatet $\vartheta$ er et Boolsk uttrykk hvor alle atomene er på formen $A\,\theta\,B$ der $A$ og $B$ er kompatible attributter i henholdsvis $R$ og $S$, og hvor $\theta \in \{=, \neq, <, >, \leq, \geq\}$.

Dersom $\vartheta$ er på formen $A_1 = B_1 \wedge \cdots \wedge A_k = B_k$, sier vi at vi har en *ekvijoin*. Ekvijoiner skrives ofte på formen $R \bowtie_{X=Y} S$ der $X$ og $Y$ er unionkompatible komponenter i henholdsvis $R$ og $S$.

Legg merke til at alle atomene i et join-predikat sammenligner verdier i to (kompatible) attributter, ett fra hver operand.

### 3.2.4 Naturlig join

La $R$ og $S$ være to relasjoner som er slik at hvis $R$ og $S$ har to attributter med samme navn, så har disse to attributtene samme domene. Da definerer vi den naturlige join mellom $R$ og $S$, $R \bowtie S$, på følgende måte:

Sett $Y = \mathbf{R}(R) \cap \mathbf{R}(S)$ (mengden av felles attributter i $R$ og $S$).
Sett $X = \mathbf{R}(R) \setminus Y$ og $Z = \mathbf{R}(S) \setminus Y$.
Skift navn på alle attributtene i $S.Y$ til noe som hverken finnes i $X$, $Y$ eller $Z$, kall $S.Y$ etter renavningen for $W$, og sett

$$R \bowtie S = \pi_{XYZ}(R \bowtie_{Y=W} S)$$

**Merk**:
Navneskiftet av $S.Y$ er utelukkende et teknisk knep for å få definisjonen formelt korrekt. Den har ingen innvirkning på resultatet og kan trygt ignoreres i all praktisk bruk.

## 4 Relasjonskalkyle

**Merk**:
Relasjonskalkyle er *ikke* pensum i INF3100 og er heller ikke omtalt i læreboken, men er likevel en viktig del av relasjonsmodellen. Dette kapitlet er derfor et tilbud til interesserte lesere.

Relasjonsalgebraen er et *prosedyreorientert* språk. Det vil si at et relasjonsalgebrauttrykk er en algoritme for *hvordan* vi finner svaret.

Relasjonskalkylen er et *deklarativt* språk. Det betyr at et relasjonskalkyleuttrykk bare beskriver *hva* svaret er, men ikke *hvordan* vi skal finne det.

Vi har to varianter av relasjonskalkyle. Begge definerer en relasjon som en mengde med et mengdepredikat uttrykt i 1. ordens logikk. Forskjellen består i hvilket univers variablene varierer over, noe som har innvirkning både på hvordan mengdeuttrykkene og de Boolske atomene ser ut.

## 4.1 Tuppelrelasjonskalkyle

En *tuppelvariabel* varierer over universet av tupler, dvs over alle mulige forekomster av alle mulige relasjoner.

Lovlige atomer i tuppelrelasjonskalkyle har en av formene:

(1) $R(t)$ der $R$ er (navnet på) en relasjon og $t$ er en tuppelvariabel.

Atomet er $T$ hvis $t \in r(R)$ (og altså $F$ hvis $t \notin r(R)$).

(2) $t.A\,\theta\,u.B$ der $t$ og $u$ er tuppelvariable, $A$ og $B$ er navn (på attributter), og $\theta \in \{=, \neq, <, >, \leq, \geq\}$.

Atomet er $T$ hvis, og bare hvis, følgende holder:
$A$ er et attributt i en relasjon $R \wedge R(t)$
$\wedge$ $B$ er et attributt i en relasjon $S \wedge S(u)$
$\wedge$ $dom(A) = dom(B)$
$\wedge$ $t$ og $u$ har verdier slik at $t.A\,\theta\,u.B$ evaluerer til $T$

(3) $t.A\,\theta\,c$ eller $c\,\theta\,t.A$ der $t$ er en tuppelvariabel, $A$ er et navn (på et attributt), $c$ er en konstant, og $\theta \in \{=, \neq, <, >, \leq, \geq\}$.

Atomet er $T$ hvis, og bare hvis, følgende holder:
$A$ er et attributt i en relasjon $R \wedge R(t)$
$\wedge$ $c \in dom(A)$
$\wedge$ $t$ og $c$ har verdier slik at atomet evaluerer til $T$

Et *tuppelpredikat* er et 1. ordens uttrykk bygget opp av slike atomer.

Et *tuppelrelasjonskalkyleuttrykk* er på formen

$$\{t_1.A_1, \ldots, t_n.A_n \mid P(t_1, \ldots, t_n, u_1, \ldots, u_m)\}$$

der $t_1, \ldots, t_n, u_1, \ldots, u_m$ er tuppelvariable, $A_1 \ldots, A_n$ er attributter og $P$ er et tuppelpredikat slik at:

- $t_1, \ldots, t_n$ er de frie og $u_1, \ldots, u_m$ de bundne variablene i $P$
- For alle $k$ med $1 \leq k \leq n$ finnes en relasjon $R_k$ slik at
  (1) $R_k(t_k)$ opptrer som atom i $P$ på en slik måte at vi kan spalte det ut fra $P$, dvs at det finnes et tuppelpredikat $P_k$ slik at $P(t_1, \ldots, t_n, u_1, \ldots, u_m) = R_k(t_k) \wedge P_k(t_1, \ldots, t_n, u_1, \ldots, u_m)$
  (2) $A_k$ er attributt i $R_k$

Det er verdt å merke seg at vi *ikke* forlanger at alle tuppelvariablene $t_1, \ldots, t_n$ er forskjellige.

## 4.2 Domenerelasjonskalkyle

En *domenevariabel* varierer over universet av domener, dvs alle mulige forekomster i alle mulige domener.

**Merk**: Det er kutyme å sløyfe kommaene i lister av domenevariable.

Lovlige atomer i domenerelasjonskalkyle har en av formene:

(1) $R(x_1 \dots x_k)$ der $R$ er (navnet på) en relasjon, og $x_1, \dots, x_k$ er domenevariable eller konstanter.

Atomet er $T$ hvis, og bare hvis, $(x_1, \dots, x_k) \in r(R)$.

(2) $x\,\theta\,y$ eller $y\,\theta\,x$ der $x$ er en domenevariabel, $y$ er en domenevariabel eller en konstant, og $\theta \in \{=, \neq, <, >, \leq, \geq\}$.

Atomet er $T$ hvis, og bare hvis, $x$ og $y$ er i samme domene og har verdier slik at atomet evaluerer til $T$

Et *domenepredikat* er et 1. ordens uttrykk bygget opp av slike atomer.

Et *domenerelasjonskalkyleuttrykk* er på formen

$$\{x_1 \dots x_n \mid P(x_1 \dots x_m)\}$$

der $x_1, \dots, x_m$ er domenevariable, $m \geq n$, og $P$ er et domenepredikat slik at:

- $x_1, \dots, x_n$ er $n$ forskjellige frie variable i $P$
- Hvis $x_k$ forekommer i et atom av type 2, og $n < k \leq m$, så er $x_k$ en bundet variabel i $P$
- For alle $k$ med $1 \leq k \leq m$ forekommer $x_k$ i et atom $R_k(x_{i_1} \dots x_k \dots x_{i_p})$ i $P$ på en slik måte at vi kan spalte atomet ut fra $P$, dvs at det finnes et domenepredikat $P_k$ slik at vi kan skrive $P$ på formen:
  $P(x_1 \dots x_m) = \exists x_{j_1} \cdots \exists x_{j_q}(R_k(x_{i_1} \dots x_k \dots x_{i_p}) \wedge P_k(x_1 \dots x_m))$

## 4.3 Sikre kalkyleuttrykk

Siden relasjoner har både endelig mange attributter og endelig mange forekomster, bør vi unngå kalkyleuttrykk med uendelige løsningsmengder. Vi ønsker også at løsningsmengdene skal være begrenset til de relasjonene som er nevnt i kalkyleuttrykket. Kalkyleuttrykk som oppfyller dette, kalles *sikre*.

Siden vi ikke skal gjøre bruk av den, skal vi ikke ta med den fullstendige definisjonen, men nøye oss med et sett av betingelser som garanterer at vi får sikre uttrykk.

Her er notasjonen for domenekalkyle brukt, men de tre tilsvarende kravene (med de nødvendige syntaktiske endringer) holder også for tuppelkalkyle.

La $P$ være et domenepredikat. Dersom

(1) $\{x_1 \cdots x_n \mid P(x_1 \cdots x_m)\}$ er endelig

(2) Alle delutrykk i $P$ på formen $\exists u(\phi(\cdots u \cdots))$ ser slik ut:
$\exists u(R(\cdots u \cdots) \wedge \phi_1(\cdots u \cdots))$

(3) Alle delutrykk i $P$ på formen $\forall u(\phi(\cdots u \cdots))$ ser slik ut:
$\forall u(\neg R(\cdots u \cdots) \vee \phi_1(\cdots u \cdots))$

holder, er $\{x_1 \cdots x_n \mid P(x_1 \cdots x_m)\}$ et sikkert domenerelasjonskalkyleuttrykk.

Det er også verdt å merke seg at alle sikre domenerelasjonskalkyleuttrykk kan omformes slik at (1)–(3) ovenfor holder.

## 4.4 Relasjonell kompletthet

Et språk som har (minst) samme uttrykkskraft som relasjonsalgebraen, kalles *relasjonelt komplett*.

Både tuppelrelasjonskalkyle og domenerelasjonskalkyle er relasjonelt komplette. Helt presist holder følgende

**Teorem**
Nøyaktig de samme relasjonene kan beskrives med

(1) Relasjonsalgebra

(2) Sikre tuppelrelasjonskalkyleuttrykk

(3) Sikre domenerelasjonskalkyleuttrykk

For å bevise teoremet er det enkleste å bevise (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (1). Den midterste implikasjonen er forholdsvis grei å bevise. De to andre bevises med induksjon på antall operatorer i uttrykket. De som er interessert i beviset, henvises til Ullman: «Principles of Database Systems» eller til Mittra: «Principles of Relational Database Systems.»

Den historiske bakgrunnen for relasjonskalkylen er at tuppelkalkylen gir en matematisk beskrivelse av hvordan WHERE-klausulen i SQL burde ha vært, mens domenekalkylen gir en matematisk beskrivelse av spørrespråket QBE (Query-By-Example).

# 5 Nøkler og avhengigheter

Vi skal her se litt på noen av de integritetsreglene (restriksjonene) som kan gjelde for en relasjon. Vi tar først for oss entydighet. Deretter generaliserer vi til funksjonelle avhengigheter og avslutter med flerverdiavhengigheter som er den mest generelle formen for avhengighet vi har mellom to komponenter i samme relasjon.

## 5.1 Nøkler

En nøkkel er en komponent som kan brukes til å identifisere forekomstene i en relasjon. I litteraturen brukes ordet «nøkkel» i flere betydninger. Vi skal unngå ordet og alltid bruke ett av følgende tre spesialiserte begreper:

### 5.1.1 Supernøkler

La $R$ være en relasjon. En komponent $X \subseteq \mathbf{R}(R)$ kalles en supernøkkel i $R$ hvis

$$t_1 \in r(R) \wedge t_2 \in r(R) \wedge t_1[X] = t_2[X] \Rightarrow t_1 = t_2$$

Merk at vi alltid har at $\mathbf{R}(R)$ er supernøkkel for $R$.

### 5.1.2 Kandidatnøkler

Minimale supernøkler kalles kandidatnøkler:

La $R$ være en relasjon, og la $X$ være en supernøkkel for $R$. Da er $X$ en kandidatnøkkel for $R$ hvis vi har at

$$\forall Y((Y \text{ er supernøkkel i } R) \wedge Y \subseteq X \Rightarrow Y = X)$$

Siden alle relasjoner har (minst) en supernøkkel, har alle relasjoner minst en kandidatnøkkel.

### 5.1.3 Primærnøkler

Primærnøkkelen i en relasjon er en kanonisk kandidatnøkkel for relasjonen, dvs at vi velger en av kandidatnøklene og kaller den primærnøkkel.

I en relasjonsdatabase skal alle relasjoner ha eksakt én primærnøkkel.

## 5.2 Fremmednøkler

Fremmednøkler er ikke nøkler i egentlig forstand. De brukes til (logisk) sammenkobling av relasjonene i en relasjonsdatabase:

La $R$ og $S$ være relasjoner, og la $P$ være primærnøkkel i $R$. En fremmednøkkel i $S$ med hensyn på $R$ er en komponent $X$ i $S$ som er unionkompatibel med $P$, og hvor hensikten er å kunne sammenligne verdier av $X$ og $P$.

En relasjon kan godt ha fremmednøkler til seg selv, men primærnøkkelen regnes aldri som en slik. I konteksten ovenfor betyr det at hvis $R = S$, så er $X \neq P$.

## 5.3 Funksjonelle avhengigheter (FDer)

La $X$ og $Y$ være to komponenter i en relasjon $R$. Da sier vi at vi har en FD (Functional Dependency) $X \rightarrow Y$ hvis

$$(t_1 \in r(R) \wedge t_2 \in r(R) \wedge t_1[X] = t_2[X]) \Rightarrow t_1[Y] = t_2[Y]$$

Vi sier at $Y$ er funksjonelt avhengig av $X$, eller at $X$ bestemmer $Y$. Hvis $Y$ er atomær, dvs $Y = \{A\}$, skriver vi som oftest $X \rightarrow A$ i stedet for det formelt korrekte $X \rightarrow \{A\}$.

Hvis $Y \subseteq X$, kalles FDen $X \rightarrow Y$ *triviell* fordi den alltid er oppfylt og ikke legger noen restriksjon på forekomstene i $R$.

### 5.3.1 $F^+$ og Armstrongs slutningsregler

La $F$ være en mengde FDer over en relasjon $R$. Da betegner $F^+$ mengden av alle FDer som kan utledes fra $F$ ved hjelp av 1. ordens logikk og mengdelære.

Armstrongs slutningsregler:

(1) Refleksivitet: Hvis $X \supseteq Y$, så har vi at $X \rightarrow Y$

(2) Utvidelse av FDer: Hvis $X \rightarrow Y$, så har vi at $XZ \rightarrow YZ$

(3) Transitivitet for FDer: Hvis $X \rightarrow Y$ og $Y \rightarrow Z$, så har vi at $X \rightarrow Z$

er sunne (dvs at vi ikke kan utlede selvmotsigelser fra dem) og komplette (dvs at alle FDer i $F^+$ kan utledes fra FDene i $F$ ved hjelp av dem).

Merk at en FD er triviell hvis, og bare hvis, den følger av refleksivitetsregelen.

### 5.3.2 Elementære FDer og determinanter

En ikke-triviell FD med atomær høyreside og minimal venstreside kalles *elementær*. Presist har vi at $X \to A$ er elementær med hensyn på en mengde FDer $F$ i en relasjon $R$ hvis følgende holder:

(1) $X$ er en komponent og $A$ et attributt i $R$
(2) $X \to A \in F^+$ ($X \to A$ er en FD)
(3) $A \notin X$ (FDen er ikke triviell)
(4) $Y \subseteq X \wedge Y \to A \in F^+ \Rightarrow Y = X$ ($X$ er minimal)

Hvis $X \to A$ er elementær, kalles $X$ en *determinant* for $A$. (Enkelte forfattere bruker ordet *determinant* om venstresiden i enhver FD. De kaller våre determinanter for *minimale determinanter*.)

Merk at et attributt $A$ kan ha flere ulike determinanter. Vi kan altså ha flere elementære FDer med samme høyreside.

Hvis $X \to A$ er elementær, sier vi at $A$ er *fullt avhengig* av $X$.

**En kommentar om terminologien**:
Begrepet *elementær FD* skyldes Zaniolo (C. Zaniolo: 'A new normal form for the design of relational database schemas', ACM TODS vol. 7 no. 3, 1982, side 489–499). Vi har valgt å bruke hans terminologi selv om *essensielle* eller *signifikante* FDer trolig er mer beskrivende. Læreboken benytter hverken termen «elementær FD» eller «determinant».

## 5.4 Flerverdiavhengigheter (MVDer)

La $X$ og $Y$ være to komponenter i en relasjon $R$. Da sier vi at vi har en MVD (Multi-Valued Dependency) $X \twoheadrightarrow Y$ hvis

$$\begin{aligned}(t_1 \in r(R) \wedge t_2 \in r(R) \wedge t_1[X] = t_2[X]) \Rightarrow \exists u_1 \exists u_2 (u_1 \in r(R) \wedge u_2 \in r(R) \\ \wedge u_1[X] = u_2[X] = t_1[X] = t_2[X] \wedge u_1[Y] = t_1[Y] \wedge u_2[Y] = t_2[Y] \\ \wedge u_1[\mathbf{R}(R) \setminus XY] = t_2[\mathbf{R}(R) \setminus XY] \wedge u_2[\mathbf{R}(R) \setminus XY] = t_1[\mathbf{R}(R) \setminus XY])\end{aligned}$$

Uttrykt med ord sier denne formelen at hvis vi har to forekomster $t_1$ og $t_2$ som er like på $X$, så finnes to tupler $u_1$ og $u_2$ slik at $u_1$ er lik $t_1$ på $Y$ og lik $t_2$ utenfor $Y$, mens $u_2$ er lik $t_2$ på $Y$ og lik $t_1$ utenfor $Y$.

Vi sier at $Y$ er fleravhengig av $X$, eller at $X$ flerbestemmer $Y$.

Hvis $Y \subseteq X$, eller hvis $XY = \mathbf{R}(R)$, har vi alltid at $X \twoheadrightarrow Y$. Slike MVDer kalles trivielle.

I tillegg til Armstrongs regler (avsnitt 5.3.1) har vi følgende slutningsregler:

(4) Komplementregelen: Hvis $X \twoheadrightarrow Y$, så har vi at $X \twoheadrightarrow (\mathbf{R}(R) \setminus XY)$

(5) Utvidelse av MVDer:
Hvis $X \twoheadrightarrow Y$ og $W \supseteq Z$, så har vi at $XW \twoheadrightarrow YZ$

(6) Transitivitet for MVDer:
Hvis $X \twoheadrightarrow Y$ og $Y \twoheadrightarrow Z$, så har vi at $X \twoheadrightarrow (Z \setminus Y)$

(7) FDer er MVDer: Hvis $X \rightarrow Y$, så har vi at $X \twoheadrightarrow Y$

(8) Sammensmelting:
Hvis $X \twoheadrightarrow Y$, $W \rightarrow Z$, $W \cap Y = \emptyset$ og $Y \supseteq Z$, så har vi at $X \rightarrow Z$

(1)–(8) utgjør et sunt og komplett sett med slutningsregler for MVDer og FDer.

# 6 Normaliseringsteori

Hensikten med normaliseringsprosessen er å unngå utilsiktet tap av informasjon og inkonsistente data. En viktig kilde til det siste er dobbeltlagring av informasjon. Hvis vi oppdaterer dobbeltlagret informasjon, må vi rette alle data som bærer denne informasjonen. Det at data må rettes flere steder samtidig, er et eksempel på en *oppdateringsanomali*.

For å unngå slike anomalier prøver vi å splitte relasjonen på en slik måte at de komponentene som inneholder dobbeltlagrede data, kommer alene i en egen relasjon, slik at vi blir kvitt dobbeltlagringen. Vi begynner beskrivelsen av normalisering og normalformer med å se på hva vi mener med å splitte relasjoner:

## 6.1 Dekomposisjoner

La $R$ være en relasjon, og la $\mathcal{D}$ være en mengde komponenter i $R$.

Hvis unionen av komponentene i $\mathcal{D}$ utgjør hele $\mathbf{R}(R)$, sier vi at $\mathcal{D}$ er en *dekomposisjon* av $R$.

### 6.1.1 Tapsfrie dekomposisjoner og JDer

La $\mathcal{D} = \{X_1, \ldots, X_n\}$ være en dekomposisjon av relasjonen $R$.

Da sier vi at $\mathcal{D}$ er *tapsfri* hvis

$$R = \pi_{X_1}(R) \bowtie \cdots \bowtie \pi_{X_n}(R)$$

holder for alle lovlige tilstander i $R$, dvs for alle mulige $r(R)$.

Betegnelsen «tapsfri» er egentlig misvisende, og vi bruker den utelukkende av historiske årsaker. Vi har alltid at

$$R \subseteq \pi_{X_1}(R) \bowtie \cdots \bowtie \pi_{X_n}(R)$$

Det er den motsatte inklusjonen som må garanteres for at vi skal ha en tapsfri dekomposisjon. Den naturlige joinen kan altså gi oss noen ekstra (falske) tupler når dekomposisjonen ikke er tapsfri.

En betingelse som sikrer oss at en dekomposisjon $\mathcal{D}$ er tapsfri, kalles en *join-avhengighet* eller JD (Join Dependency) i $R$. Hvis $\mathbf{R}(R) \in \mathcal{D}$, er $\mathcal{D}$ automatisk tapsfri, og vi sier at vi har en triviell JD. Hvis $\mathbf{R}(R) \notin \mathcal{D}$, må det en ikke-triviell JD til for å sikre at $\mathcal{D}$ er tapsfri.

En komponent $U$ i en tapsfri dekomposisjon $\mathcal{D}$ av $R$ kalles *overflødig* hvis $\mathcal{D} \setminus \{U\}$ også er en tapsfri dekomposisjon av $R$.

En tapsfri dekomposisjon uten overflødige komponenter kalles *ureduserbar* (irreducible).

Normaliseringsprosessen består i å kvitte oss med oppdateringsanomalier ved å foreta tapsfri dekomposisjoner. Tapsfrihet er nødvendig for at vi skal kunne gjenskape den opprinnelige relasjonen ut fra projeksjonene på komponentene.

Det er dessuten åpenbart sløsing med plass å ha overflødige komponenter. Vi skal derfor bare benytte ureduserbare dekomposisjoner.

**En kommentar om terminologi**:
I nyere normaliseringslitteratur skilles det mellom FDer og JDer. Det er strengt tatt ikke riktig da ikketrivielle FDer som ikke har en supernøkkel som venstreside, gir opphav til en ikketriviell JD. Her skal vi reservere betegnelsen JD for «ekte» JDer, dvs. JDer som *ikke* er FDer. Vi kommer tilbake til dette i avsnitt 6.3.1.

## 6.2 Normalformer

Data som lagres i relasjoner, kalles *normaliserte*. Vi sier de er på *første normalform* (1NF).

Graden av vellykkethet i prosessen måles i *normalformer*. Dersom vi ikke har noen oppdateringsanomalier, har vi den høyeste normalformen, 5NF, også kalt PJ/NF (Project-Join Normal Form), den som er målet for våre bestrebelser. Dessverre gjelder ikke det omvendte. Selv om vi har oppnådd 5NF, kan vi fortsatt ha oppdateringsanomalier, men disse anomaliene kan vi ikke fjerne ved å foreta tapsfrie dekomposisjoner.

I avsnitt 6.6 skal vi vise hvordan FDer og MVDer kan brukes til å finne dekomposisjoner som gir oss høyere normalformer. Men først skal vi beskrive normalformene, både de som er pensum i INF3100,og noen som ikke er pensum.

Vi vil få bruk for følgende to enkle definisjoner:

### 6.2.1 Nøkkelattributter og elementære nøkler

Attributter som ligger i kandidatnøkler, kalles *nøkkelattributter*. Et ikke-nøkkelattributt er altså ikke element i noen kandidatnøkkel.

En kandidatnøkkel $X$ kalles *elementær* hvis $X$ er determinant for en elementær FD.

### 6.2.2 1NF (Codd 1972)

Det å si at en datastruktur er på første normalform (1NF), er det samme som å si at datastrukturen er en relasjon.

Alle relasjoner er altså på 1NF.

### 6.2.3 2NF (Codd 1972)

En relasjon $R$ er på andre normalform (2NF) hvis alle ikke-nøkkelattributter er fullt avhengige av alle kandidatnøkler, dvs. at ingen ikke-nøkkelattributter er funksjonelt avhengig av en del av en kandidatnøkkel. En annen måte å si dette på er at 2NF betyr at alle FDer på formen $K \rightarrow A$ der $K$ er en kandidatnøkkel og $A$ er et ikke-nøkkelattributt, er elementære.

Dette kan oppsummeres slik:

En relasjon $R$ er på 2NF hvis alle ikke-trivielle FDer i $R$ på formen $X \rightarrow A$ (der $A$ er et attributt i $R$) tilfredsstiller minst ett av følgende tre krav:

(1) $X$ er en supernøkkel i $R$

(2) $A$ er et nøkkelattributt i $R$

(3) $X \nsubseteq K$ for noen kandidatnøkkel $K$ i $R$

Siden alle relasjoner er på 1NF, er spesielt alle 2NF-relasjoner på 1NF.

**Eksempel**

La $R$ være en relasjon med skjema $\{A, B, C\}$ hvor vi har FDen $A \rightarrow B$ (f.eks. la $A$ være fødselsnummer, $B$ navn og $C$ emnekode).

Her er $AC$ eneste kandidatnøkkel, og $A \rightarrow B$ strider mot (1), (2) og (3).

Eksemplet viser at det finnes relasjoner som ikke er på 2NF.

### 6.2.4 3NF (Codd 1972)

En relasjon $R$ er på tredje normalform (3NF) hvis alle ikke-trivielle FDer i $R$ på formen $X \rightarrow A$ (der $A$ er et attributt i $R$) tilfredsstiller minst ett av følgende to krav:

(1) $X$ er en supernøkkel i $R$

(2) $A$ er et nøkkelattributt i $R$

Siden (1) og (2) ovenfor er identiske med (1) og (2) i kravet til 2NF, er alle 3NF-relasjoner også på 2NF.

I den klassiske definisjonen av 3NF het det at *transitive FDer* ikke var tillatt i 3NF-relasjoner. En transitiv FD ble definert som en FD fra noe som ikke er en supernøkkel til et ikke-nøkkelattributt.

Den tredje muligheten under 2NF sier nettopp at transitive FDer er tillatt i 2NF-relasjoner (men altså ikke i 3NF-relasjoner).

**Eksempel**

La $R$ være en relasjon med skjema $\{A, B, C\}$ hvor vi har FDene $A \rightarrow B$ og $B \rightarrow C$ (f.eks. la $A$ være fødselsnummer, $B$ postnummer og $C$ postdistrikt).

Her er $A$ eneste kandidatnøkkel, og $A \rightarrow B$ strider mot (1) og (2), men ikke mot (3) i 2NF-definisjonen.

Eksemplet viser at det finnes 2NF-relasjoner som ikke er på 3NF.

### 6.2.5 EKNF (Zaniolo 1982)

En relasjon $R$ er på elementær nøkkelnormalform (EKNF = Elementary Key Normal Form) hvis alle ikke-trivielle FDer i $R$ på formen $X \rightarrow A$ (der $A$ er et attributt i $R$) tilfredsstiller minst ett av følgende to krav:

(1) $X$ er en supernøkkel i $R$

(2) $A$ er et attributt i en elementær kandidatnøkkel i $R$

Forskjellen på denne definisjonen og definisjonen av 3NF er at mulighet (2) er skjerpet til bare å gjelde elementære kandidatnøkler i $R$. Følgelig er alle EKNF-relasjoner på 3NF.

**Eksempel** (C. Zaniolo)

La $R$ være en relasjon med skjema $\{A, B, C\}$ der $AB$ og $BC$ er kandidatnøkler, og hvor vi i tillegg har de to FDene $A \rightarrow C$ og $C \rightarrow A$

(f.eks. la $A$ være fødselsnummer, $B$ emnekode og $C$ emailadresse på studenter).

Siden alle attributtene i $R$ er nøkkelattributter, er krav (2) for 3NF alltid oppfylt, så $R$ er åpenbart på 3NF.

Derimot har ikke $R$ noen elementære kandidatnøkler, så både $A \rightarrow C$ og $C \rightarrow A$ bryter mot EKNF.

Eksemplet viser at det finnes 3NF-relasjoner som ikke er på EKNF.

### 6.2.6 BCNF (Boyce, Codd 1974)

En relasjon $R$ er på Boyce-Codd normalform (BCNF) hvis vi for alle ikke-trivielle FDer i $R$ på formen $X \rightarrow A$ (der $A$ er et attributt i $R$), har at $X$ er en supernøkkel i $R$.

Siden alle BCNF-relasjoner tilfredsstiller det første alternativet i EKNF-definisjonen, er alle BCNF-relasjoner på EKNF.

**Eksempel**

La $R$ være en relasjon med skjema $\{A, B, C\}$ der $AB$ er kandidatnøkkel, og hvor vi i tillegg har FDen $C \rightarrow A$ (f.eks. la $A$ være fagforening, $B$ verv og $C$ person). Altså er $C^+ = AC$, og $BC$ er en kandidatnøkkel for $R$.

Siden $C$ ikke er en kandidatnøkkel, bryter $C \rightarrow A$ mot BCNF.

Observer imidlertidig at $AB$ er en elementær kandidatnøkkel (i motsetning til $BC$ som *ikke* er elementær), så $C \rightarrow A$ bryter ikke mot EKNF.

I definisjonen av BCNF forutsatte vi at høyresidene i FDene var atomære. Det gjør åpenbart ingen forskjell å tillate generelle komponenter som høyresider. Vi har altså:

En relasjon $R$ er på BCNF hvis (og bare hvis) vi for alle ikke-trivielle FDer $X \rightarrow Y$ i $R$ har at $X$ er en supernøkkel i $R$.

### 6.2.7 4NF (Fagin 1977)

En relasjon $R$ er på fjerde normalform (4NF) hvis vi for alle ikke-trivielle MVDer $X \twoheadrightarrow Y$ i $R$ har at $X$ er en supernøkkel i $R$.

**Påstand** Alle 4NF-relasjoner er også på BCNF

**Bevis**

Anta at $R$ er på 4NF og at vi har en ikke-triviell FD $X \rightarrow Y$ i $R$. Vi skal vise at $X$ er en supernøkkel i $R$.

Siden alle FDer er MVDer, har vi at $X \twoheadrightarrow Y$ i $R$. Da har vi to muligheter:

- Den første er at MVDen ikke er triviell. Siden $R$ er på 4NF, medfører det at i så fall er $X$ en supernøkkel i $R$.
- Den andre er at MVDen er triviell, dvs. at $\mathbf{R}(R) \setminus XY = \emptyset$. Men da er $\mathbf{R}(R) = XY$, så $X$ er en supernøkkel for $R$ i dette tilfellet også.

Observer at alle relasjoner som inneholder en ikke-triviell MVD, men ingen ikke-trivielle FDer, er på BCNF, men ikke på 4NF.

#### 6.2.8 5NF (Fagin 1979)

En relasjon $R$ er på femte normalform (5NF) hvis alle JDer kan utledes fra kandidatnøklene.

Mer presist er $R$ på 5NF hvis alle tapsfrie dekomposisjoner er *nøkkelsammenhengende* i henhold til følgende:

**Algoritme** (Fagin)

(1) Sett $\mathcal{D}$ initielt lik den tapsfrie dekomposisjonen

(2) Utfør inntil intet forandrer seg

- Hvis det finnes $U, V \in \mathcal{D}$ med $U \subset V$, fjern $U$ fra $\mathcal{D}$
- Hvis det finnes $U, V \in \mathcal{D}$ og en kandidatnøkkel $K$ i $R$ med $K \subseteq U \cap V$, erstatt $U$ og $V$ med $U \cup V$ i $\mathcal{D}$

(3) Hvis $\mathcal{D} = \{\mathbf{R}(R)\}$, er den opprinnelige dekomposisjonen nøkkelsammenhengende, ellers ikke

Opprinnelig brukte Fagin betegnelsen PJ/NF (Project-Join Normal Form) på 5NF, og disse betegnelsene brukes om hverandre i litteraturen. Vi anbefaler å bruke 5NF som betegnelse på denne normalformen. Dette begrunnes i avsnitt 6.5.4 nedenfor.

### 6.3 FDer, MVDer og dekomposisjoner

#### 6.3.1 Alle MVDer er JDer

Vi skal nå bevise at alle MVDer er JDer. Det gjør vi ved hjelp av følgende

**Teorem** (Fagin 1977)
La $R$ være en relasjon, og la $\mathcal{D} = \{U, V\}$ være en dekomposisjon av $R$. Da er $\mathcal{D}$ tapsfri hvis, og bare hvis, $U \cap V \twoheadrightarrow (U \setminus V)$ i $R$.

**Bevis:** Sett $X = U \cap V$, $Y = U \setminus V$ og $Z = V \setminus U$, og merk at da er $X$, $Y$ og $Z$ parvis disjunkte med $U = YX$ og $V = XZ$.

La oss først se på tilfellet hvor $\mathcal{D}$ har en triviell JD. Observer at

$$\mathcal{D} \text{ har en triviell JD} \Leftrightarrow U = \mathbf{R}(R) \vee V = \mathbf{R}(R)$$
$$\Leftrightarrow Y = \emptyset \vee Z = \emptyset \Leftrightarrow X \twoheadrightarrow Y \text{ er triviell}$$

Anta så at $\mathcal{D}$ ikke har noen triviell JD, dvs at $Y \neq \emptyset \wedge Z \neq \emptyset$.

At $\mathcal{D}$ er tapsfri betyr at $r(\pi_{YX}(R) \bowtie \pi_{XZ}(R)) = r(R)$. Når vi beregner venstresiden, får vi alle kombinasjoner av forekomster i de to projeksjonene som er like på $X$. Dermed er $\mathcal{D}$ tapsfri hvis, og bare hvis:

Hver gang $(y_1, x, z_1)$ og $(y_2, x, z_2)$ er to forekomster i $R$ som er like på $X$, så er også «krysskoblingene» $(y_1, x, z_2)$ og $(y_2, x, z_1)$ forekomster i $R$ (de blir nemlig forekomster i $\pi_{YX}(R) \bowtie \pi_{XZ}(R)$).

Men dette er identisk med definisjonen av at $X \twoheadrightarrow Y$ i $R$.

qed

Teoremet sier at en MVD er det samme som en JD for en dekomposisjon i to komponenter, så MVDer er JDer.

Siden en FD er en MVD, er strengt tatt også FDer JDer, men som sagt i terminologikommentaren i avsnitt 6.1.1, skal ikke FDer regnes som JDer. Det betyr at bare MVDer som ikke er FDer, er JDer.

### 6.3.2 4NF og dekomposisjoner

Vi skal nå bevise at 4NF også kan defineres ved hjelp av ureduserbare tapsfrie dekomposisjoner, men for å gjøre det trenger vi følgende

**Lemma**

La $R$ være en relasjon, og la $\{U, V\}$ være en ureduserbar tapsfri dekomposisjon av $R$ hvor både $U$ og $V$ er supernøkler. Da er $U \cap V$ en supernøkkel i $R$.

**Bevis** (ved kontradiksjon):

Sett $X = U \cap V$, $Y = U \setminus V$ og $Z = V \setminus U$, og anta at $X$ ikke er en supernøkkel, men at $U = YX$ og $V = XZ$ er det.

Siden $X$ ikke er supernøkkel, kan det finnes to forekomster $(y_1, x, z_1)$ og $(y_2, x, z_2)$ i $R$ som er like på $X$, og siden $YX$ og $XZ$ er supernøkler, må vi i så fall ha $(y_1 \neq y_2)$ og $(z_1 \neq z_2)$.

Men siden $\{U, V\}$ er tapsfri, har vi at $X \twoheadrightarrow Y$ i $R$. Det betyr at $(y_1, x, z_2)$ og $(y_2, x, z_1)$ begge må være forekomster i $R$. Som vist ovenfor, er $(z_1 \neq z_2)$, så $(y_1, x, z_1)$ og $(y_1, x, z_2)$ er to forskjellige forekomster i $R$, noe som strider mot at $YX$ er supernøkkel i $R$.

qed

Vi er nå i stand til å bevise følgende

**Teorem**

La $R$ være en relasjon. Da er $R$ på 4NF hvis, og bare hvis, begge komponentene i alle ureduserbare tapsfrie dekomposisjoner med to komponenter er supernøkler i $R$.

**Bevis:**

Anta først at $R$ er på 4NF. La $\{U, V\}$ være en ureduserbar tapsfri dekomposisjon av $R$. Da er $U \cap V \twoheadrightarrow (U \setminus V)$ en ikke-triviell MVD, så $U \cap V$ er en supernøkkel i $R$. Men da er opplagt både $U$ og $V$ supernøkler.

Anta så at $R$ ikke er på 4NF. Da finnes en ikke-triviell MVD $X \twoheadrightarrow Y$ i $R$ hvor $X$ ikke er en supernøkkel. Siden $X \twoheadrightarrow X$ er triviell, gir den transitive loven for MVDer at $X \twoheadrightarrow Y \setminus X$, så vi kan uten tap av generalitet anta at $X \cap Y = \emptyset$. Hvis vi setter $Z = \mathbf{R}(R) \setminus XY$, så er $\{XY, XZ\}$ en ureduserbar tapsfri dekomposisjon av $R$ med $X = XY \cap XZ$ (at dekomposisjonen er tapsfri, følger av at $X \twoheadrightarrow Y$). Siden X ikke er en supernøkkel, ville det stride mot lemmaet ovenfor hvis både $XY$ og $XZ$ var supernøkler.

qed

## 6.4 Overflødige tupler og oppdateringsanomalier

Dette avsnittet er *ikke* pensum i INF3100.

### 6.4.1 Delvis overflødige tupler

La $R$ være en relasjon, la $r$ være en instans av $R$, og la $t \in r$ være et tuppel. Da sier vi at $t$ er *delvis overflødig* i $r$ hvis det finnes et tuppel $t' \in r$, hvor $t' \neq t$, og en ikke-triviell FD $X \rightarrow A$ hvor $t'[X] = t[X]$.

Grunnen til denne terminologien er at FDen sikrer at $t(A) = t'(A)$. Det betyr at $A$-verdien bestemt av $X$ lagres både i $t$ og $t'$.

**Teorem** (Fagin 1981)

En relasjon er på BCNF hvis, og bare hvis, ingen lovlig instans har et delvis overflødig tuppel.

**Bevis:**

Anta først at $R$ er på BCNF. Dersom en instans $r$ har et delvis overflødig tuppel $t$, skal det i følge definisjonen finnes en FD $X \rightarrow A$ og et tuppel $t' \in r$ slik at $t[X] = t'[X]$ og $t' \neq t$. Men da er ikke $X$ en supernøkkel, noe som strider mot BCNF.

Anta så at $R$ ikke er på BCNF. Da finnes en ikke-triviell FD $X \rightarrow Y$ i $R$ hvor $X$ ikke er en supernøkkel, og siden $X$ ikke er en supernøkkel, må det finnes et attributt $A$ slik at $X \rightarrow A$ ikke er en FD i $R$.

Vi lager oss nå to tupler $s$ og $s'$ med de samme attributtene som $R$ hvor $s[X] = s'[X]$ og $s(B) \neq s'(B)$ for alle attributter $B$ utenfor $X$. Så bruker vi chase-algoritmen på instansen som består av tuplene $s$ og $s'$. I løpet av chase-prosessen sørger vi for at vi ikke endrer på $s$, men bytter ut noen av $s'(B)$-verdiene med $s(B)$. Kall sluttresultatet av $s'$ for $t$ og sett $r = \{s, t\}$. Chase-algoritmen sikrer at $r$ er en lovlig instans av $R$, og at $t[X] = s'[X] = s[X]$. Siden $X \rightarrow A$ ikke er en FD i $R$, har vi at $t(A) = s'(A) \neq s(A)$ som viser at $t \neq s$. Men da er $t$ et delvis overflødig tuppel i $r$.

qed

### 6.4.2 Fullstendig overflødige tupler

La $\mathbf{R}(R)$ være et relasjonsskjema, la $S$ være en mengde tupler, og la $t$ være et tuppel. Da sier vi at $t$ er *en logisk konsekvens av* $S$ (ut fra avhengighetene i $\mathbf{R}(R)$) hvis hver instans av $\mathbf{R}(R)$ som inneholder alle tuplene i $S$, også må inneholde $t$.

La $R$ være en relasjon med skjema $\mathbf{R}(R)$, la $r$ være en instans av $R$, og la $t \in r$ være et tuppel. Da sier vi at $t$ er *fullstendig overflødig* i $r$ hvis det finnes en mengde tupler $S$ i $r$ hvor $t \notin S$ slik at $t$ er en logisk konsekvens av $S$.

Merk at «delvis overflødig» og «fullstendig overflødig» er ortogonale begreper. Et tuppel kan være fullstendig overflødig uten å være delvis overflødig, og omvendt.

### 6.4.3 Essensielle tupler

Begrepet *essensielle tupler* er inspirert av begrepet *essensielle data* introdusert av E. F. Codd i artikkelen «Codd and Date: Interactive support for non-programmers: the relational and network approaches.» (publisert i «Proceedings of the 1974 ACM SIGFIDET, pages 11–41»), hvor han sier at essensielle data er data som ikke kan fjernes uten tap av informasjon.

La $t$ være et tuppel i en instans $r$ av en relasjon $R$. Da sier vi at $t$ er *essensiell* i $r$ hvis $t$ hverken er delvis eller fullstendig overflødig i $r$.

### 6.4.4 Delete-anomalier (Fagin 1981)

Et relasjonsskjema $\mathbf{R}(R)$ sies å ha en *delete-anomali* hvis det finnes to relasjoner $r_1$ og $r_2$ som begge har de samme attributtene som $\mathbf{R}(R)$, slik at

(1) $r_2$ inneholder de samme tuplene som $r_1$ pluss eksakt ett tuppel $t$ i tillegg

(2) $r_1$ tilfredstiller ikke avhengighetene i $\mathbf{R}(R)$

(3) $r_2$ tilfredstiller avhengighetene i $\mathbf{R}(R)$

Dette betyr at vi har en delete-anomali når vi ved å fjerne ett tuppel fra en lovlig instans av $\mathbf{R}(R)$ får noe som ikke er en lovlig instans av $\mathbf{R}(R)$.

Merk at vi kan bytte ut krav (1) ovenfor med $r_1 \subset r_2$ (ekte inklusjon) uten at definisjonen blir endret. Fordi $r_2$ er en lovlig instans, og vi ved å fjerne ett og ett tuppel fra $r_2$ kommer til $r_1$ som ikke er en lovlig instans, må vi et sted på veien ha fjernet et tuppel $t$ som tilfredstiller den opprinnelige definisjonen.

**Teorem**

La $\mathbf{R}(R)$ være et relasjonskjema uten andre avhengigheter enn FDer og JDer. Da har $\mathbf{R}(R)$ en delete-anomali hvis, og bare hvis, $\mathbf{R}(R)$ har en instans med et fullstendig overflødig tuppel.

**Bevis:**

Anta først at $\mathbf{R}(R)$ har en delete-anomali, og la $r_1$, $r_2$ og $t$ være som i definisjonen. Sett $S = r_1$ og $r = r_2$. Da er $S \subseteq r$, og siden $r$ er en instans av $\mathbf{R}(R)$ og dermed overholder alle FDer i $\mathbf{R}(R)$, så må også $S$ overholde disse FDene. Altså må det være en JD i $\mathbf{R}(R)$ som holder i $r$, men ikke i $S$. Det betyr at $t$ er en logisk konsekvens av $S$ med hensyn på $\mathbf{R}(R)$. Siden $t \notin S$, følger det av definisjonen at $t$ er fullstendig overflødig i $r$ med hensyn på $\mathbf{R}(R)$.

Anta så at $\mathbf{R}(R)$ har en instans med et fullstendig overflødig tuppel, og la $t$, $r$ og $S$ være som i definisjonen i avsnitt 6.2.8. Siden $t$ er logisk bestemt av $S$, og $t \notin S$, kan ikke $S$ tilfredstille alle avhengighetene i $\mathbf{R}(R)$. Sett $S = r_1$ og $r = r_2$. Da følger det av kommentaren etter definisjonen at $\mathbf{R}(R)$ har en delete-anomali.

qed

### 6.4.5 Insert-anomalier (Fagin 1981)

Et relasjonsskjema $\mathbf{R}(R)$ sies å ha en *insert-anomali* hvis det finnes to relasjoner $r_1$ og $r_2$ som begge har de samme attributtene som $\mathbf{R}(R)$, slik at

(1) $r_2$ inneholder de samme tuplene som $r_1$ pluss eksakt ett tuppel $t$ i tillegg

(2) $r_1$ tilfredstiller avhengighetene i $\mathbf{R}(R)$

(3) $r_2$ tilfredstiller ikke avhengighetene i $\mathbf{R}(R)$

(4) $r_2$ tilfredstiller alle nøkkel- og domeneavhengigheter i $\mathbf{R}(R)$

Dette betyr at vi har en insert-anomali når vi ved å legge til ett tuppel $t$ til en lovlig instans $r_1$ av $\mathbf{R}(R)$ får noe som ikke er en lovlig instans av $\mathbf{R}(R)$ selv om $t$ tilfredstiller domenerestriksjonene i $\mathbf{R}(R)$ og ikke er lik noe tuppel i $r_1$ på noen kandidatnøkkel i $\mathbf{R}(R)$.

**Teorem**

La $\mathbf{R}(R)$ være et relasjonskjema uten andre avhengigheter enn FDer og JDer (spesielt skal det ikke være noen begrensende domenerestriksjoner). Da har ikke $\mathbf{R}(R)$ noen insert-anomalier hvis, og bare hvis, $\mathbf{R}(R)$ er på 5NF.

**Bevis:**

Anta først at $\mathbf{R}(R)$ er på 5NF og la $r_2$ være en lovlig instans av $\mathbf{R}(R)$. Da må $r_2$ tilfredstille alle nøkkelavhengighetene i $\mathbf{R}(R)$, og siden alle avhengigheter i $\mathbf{R}(R)$ er en logisk konsekvens av nøkkelavhengighetene i $\mathbf{R}(R)$, må $r_2$ tilfredstille alle avhengighetene i $\mathbf{R}(R)$. Altså kan ikke $\mathbf{R}(R)$ ha noen insert-anomali.

Anta så at $\mathbf{R}(R)$ ikke er på 5NF. Da finnes en relasjon $r$ med de samme attributtene som $\mathbf{R}(R)$, og som tilfredstiller alle nøkkelavhengighetene i $\mathbf{R}(R)$, men som bryter en FD eller JD i $\mathbf{R}(R)$. Hvis $r$ har $k$ tupler, finnes relasjoner $s_0, s_1, \ldots, s_k$ hvor $s_0 = \emptyset$ og $s_k = r$ hvor hver $s_{i+1}$ er laget ved å legge til eksakt ett tuppel til $s_i$ for $0 \leq i \leq k-1$. Siden $s_0$ ikke bryter med noen FD eller JD i $\mathbf{R}(R)$, mens $s_k$ gjør det, må det finnes en $j$ slik at $s_j$ tilfredstiller alle FDer og JDer i $\mathbf{R}(R)$, mens $s_{j+1}$ bryter minst en FD eller JD. Sett $r_1 = s_j$ og $r_2 = s_{j+1}$. Siden $r_2 \subseteq r$, og $r$ tilfredstiller alle nøkkelavhengighetene i $\mathbf{R}(R)$, må også $r_2$ overholde alle nøkkelenhetene. Dette valget av $r_1$ og $r_2$ viser at $\mathbf{R}(R)$ har en insert-anomali.

qed

## 6.5 Normalformer mellom 4NF og 5NF

Dette avsnittet er *ikke* pensum i INF3100.

Terminologien (og mye av innholdet) er hentet fra en artikkel av Hugh Darwen, C. J. Date og Ronald Fagin: «A Normal Form for Preventing Redundant Tuples in Relational Databases» presentert på ICDT 2012 (ICDT = International Conference on Database Theory).

I dette avsnittet skal vi kalle den *ETNF-artikkelen*.

### 6.5.1 SKNF (Maier 1983)

En relasjon $R$ er på SKNF (Super Key Normal Form) hvis alle ureduserbare tapsfrie dekomposisjoner av $R$ bare består av supernøkler.

Det betyr at hvis $R$ er på SKNF, og $\mathcal{D} = \{U_1, \ldots, U_k\}$ er en ureduserbar tapsfri dekomposisjon av $R$, så er hver $U_i$ en supernøkkel i $R$.

**Teorem (Ragnar Normann 1998)**
Alle 5NF-relasjoner er på SKNF, men ikke omvendt

**Bevis:**
Anta først at $R$ er på 5NF og la $\mathcal{D} = \{X_1, \ldots, X_m\}$ være en ureduserbar tapsfri dekomposisjon av $R$. Hvis $m = 1$, så er $X_1 = \mathbf{R}(R)$ som er en supernøkkel, så anta $m \geq 2$. Velg $k$ vilkårlig med $1 \leq k \leq m$, definer $\mathcal{D}_k = \mathcal{D} \setminus \{X_k\}$, og sett $U_k = \bigcup_{X \in \mathcal{D}_k} X$.

Siden $\mathcal{D}$ er ureduserbar, er ikke $\mathcal{D}_k$ noen tapsfri dekomposisjon av $R$. Altså er ikke $\mathcal{D}_k$ en nøkkelsammenhengende dekomposisjon, mens $\mathcal{D}$ er. Men det kan bare skje hvis $X_k \cap U_k$ inneholder en kandidatnøkkel. Altså er $X_k$ en supernøkkel, og $R$ er på SKNF.

Det gjenstår å vise at det finnes relasjoner i SKNF som ikke er i 5NF. Som eksempel skal vi bruke følgende ternære relasjon:

| $R$ | $A_1$ | $A_2$ | $A_3$ |
|---|---|---|---|
| | $a_1$ | $u_2$ | $u_3$ |
| | $a_1$ | $v_2$ | $v_3$ |
| | $x_1$ | $a_2$ | $x_3$ |
| | $y_1$ | $a_2$ | $y_3$ |
| | $z_1$ | $z_2$ | $a_3$ |
| | $w_1$ | $w_2$ | $a_3$ |

Vi skal bevise at dette er en lovlig tilstand for en relasjon $R$ med skjema $\mathbf{R}(R) = \{A_1, A_2, A_3\}$, tre kandidatnøkler, $\{A_1, A_2\}$, $\{A_2, A_3\}$ og $\{A_3, A_1\}$, og en JD som sikrer at dekomposisjonen $\mathcal{D} = \{\{A_1, A_2\}, \{A_2, A_3\}, \{A_3, A_1\}\}$ er tapsfri, og så finne alle ureduserbare tapsfrie dekomposisjoner og vise at de bare består av supernøkler.

Vi har (som alltid) den trivielle dekomposisjonen $\{\mathbf{R}(R)\}$ som er en ureduserbar tapsfri dekomposisjon bestående av én supernøkkel.

På den annen side kan ikke en ureduserbar tapsfri dekomposisjon ha noen atomær komponent, f.eks. $\{A_1\}$. Fordi en ureduserbar dekomposisjon ikke kan ha noen komponent som er inneholdt i en annen komponent, måtte $\{A_1\}$ i så fall være disjunkt fra alle andre komponenter. Men da er det bare to mulige dekomposisjoner, nemlig $\{\{A_1\}, \{A_2, A_3\}\}$ og $\{\{A_1\}, \{A_2\}, \{A_3\}\}$, og ingen av dem er tapsfrie.

Det gjenstår å se på dekomposisjoner hvor alle komponentene har to attributter. Når vi danner $\pi_{\{A_1,A_2\}}(R) \bowtie \pi_{\{A_2,A_3\}}(R)$, får vi de to falske tuplene $(x_1, a_2, y_3)$ og $(y_1, a_2, x_3)$. Disse blir begge borte når vi «joiner» resultatet med $\pi_{\{A_3,A_1\}}(R)$. På grunn av symmetrien har vi dermed bevist at $\mathcal{D}$ er tapsfri og ureduserbar. Vi observerer at alle komponentene i $\mathcal{D}$ er kandidatnøkler, og dermed supernøkler, i $R$, og har dermed bevist at $R$ er på SKNF.

På den annen side er hverken $A_1$, $A_2$ eller $A_3$ supernøkler. Følgelig er ikke $\mathcal{D}$ nøkkelsammenhengende, så $R$ er ikke på 5NF.

qed

### 6.5.2 RFNF og KCNF (Vincent 1995)

RFNF (Redundancy-Free Normal Form) og KCNF (Key-Complete Normal Form) ble beskrevet av M. W. Vincent i artikkelen «Redundancy elimination and a new normal form for relational database design» i bind 1358 av Lecture Notes in Computer Science, side 247–264, Springer 1995.

Før vi kan definere RFNF, trenger vi litt terminologi:

La $\mathbf{R}(R)$ være et relasjonsskjema, la $r$ være en instans av $\mathbf{R}(R)$, la $t$ være et tuppel i $r$, og la $A$ være et attributt i $\mathbf{R}(R)$.

Fordi samme verdi kan forekomme flere ganger i samme tuppel, skal vi skrive en verdi $t(A)$ som paret $(t, A)$.

Vi sier at paret $(t, A)$ er *overflødig i $r$ (med hensyn på $\mathbf{R}(R)$)* hvis vi hver gang $t'$ er et tuppel der $t'(A) \neq t(A)$ og $t'(B) = t(B)$ for alle attributter $B$ bortsett fra $A$ og lar $r'$ betegne det vi får ved å erstatte tuplet $t$ i $r$ med $t'$, så er ikke $r'$ en instans av $\mathbf{R}(R)$.

Intuitivt sier denne definisjonen at verdien av tuplet $t$ i attributtet $A$ er overflødig hvis det er entydig bestemt av resten av relasjonen.

Legg merke til at delvis og fullt overflødige tupler er spesialtilfeller av overflødige verdier. For hvis $r$, $t$ og $A$ er som i definisjonen av delvis overflødige tupler (se avsnitt 6.4.1), så er åpenbart $(t, A)$ overflødig i $r$, og hvis $r$ og $t$ er som i definisjonen av delvis overflødige tupler (se avsnitt 6.4.2), så er $(t, A)$ like åpenbart overflødig i $r$ for alle attributter $A$.

**Definisjon av RFNF**

> Et relasjonsskjema $\mathbf{R}(R)$ er på RFNF hvis det ikke finnes noen instans $r$ av $\mathbf{R}(R)$, tuppel $t$ i $r$ og attributt $A$ slik at $(t, A)$ er overflødig i $r$ med hensyn på $\mathbf{R}(R)$.

**Definisjon av KCNF**

> Et relasjonsskjema $\mathbf{R}(R)$ som er spesifisert utelukkende av FDer og JDer, er på KCNF hvis den er på BCNF, og vi for alle JDer $J$ i $\mathbf{R}(R)$ (eksplisitte og implisitte) har at unionen av de komponentene i $J$ som er supernøkler inneholder alle attributtene i $\mathbf{R}(R)$.

Det følgende teoremet viser at KCNF er en syntaktisk karakterisering av RFNF.

**Teorem (Vincent 1995)**

> La $\mathbf{R}(R)$ være et relasjonsskjema som er spesifisert utelukkende av FDer og JDer. Da er $\mathbf{R}(R)$ på RFNF hvis, og bare hvis, den er på KCNF.

At SKNF $\Rightarrow$ KCNF er opplagt. Dermed har vi at SKNF $\Rightarrow$ RFNF. Vincent viser også at den motsatte implikasjonen ikke holder.

### 6.5.3 ETNF (Darwen, Date, Fagin 2012)

Dette avsnittet er hentet fra ETNF-artikkelen nevnt i innledningen til dette hovedavsnittet.

Et relasjonsskjema $\mathbf{R}(R)$ er på ETNF (Essential Tuple Normal Form) hvis alle tupler i alle instanser av $\mathbf{R}(R)$ er essensielle. (Se avsnitt 6.4.3)

**Teorem (Darwen, Date, Fagin 2012)**
La $\mathbf{R}(R)$ være et BCNF relasjonsskjema som er spesifisert utelukkende av FDer og JDer. Da er $\mathbf{R}(R)$ på ETNF hvis, og bare hvis, den er på BCNF og alle eksplisitte JDer i $\mathbf{R}(R)$ har (minst) en supernøkkel som komponent.

**Korollar**
La $\mathbf{R}(R)$ være et relasjonsskjema som er spesifisert utelukkende av FDer og JDer, og som har (minst) ett attributt som er kandidatnøkkel. Da er $\mathbf{R}(R)$ på ETNF.

Beviset for teoremet er for komplisert til å ta med her.

Korollaret er imidlertid opplagt: Hvis attributtet $A$ er en kandidatnøkkel og $J$ er en JD, så må $A$ være med i en av komponentene i $J$, og denne komponenten er en supernøkkel.

Her er et eksempel som viser hvorfor betingelsen i teoremet om at $\mathbf{R}(R)$ utelukkende skal være spesifisert av FDer og JDer er nødvendig:

**Eksempel (Ellen Munthe-Kaas 2012)**
La $R(A,B,C,D,E,F)$ være en relasjon med skjema $\mathbf{R}(R)$ der alle attributtene har de naturlige tall som domene. La

$$\mathcal{F} = \{ABC \rightarrow DEF, DEF \rightarrow ABC\}$$

være mengden av eksplisitte FDer i $\mathbf{R}(R)$ og sett

$$\Sigma = \{\text{ABDE, ABDF, ABEF, ACDE, ACDF, ACEF, BCDE, BCDF, BCEF}\}$$

Da er $\Sigma$ den størst mulige dekomposisjonen av $\mathbf{R}(R)$ som tilfredsstiller følgende to krav:

(1) Ingen komponent er inneholdt i en annen.

(2) Ingen komponent er supernøkkel i $\mathbf{R}(R)$.

Sett $\sigma = \bowtie \Sigma$. Vi ønsker å se om $\mathcal{F} \vdash \sigma$ (dvs. om $\sigma$ er en logisk konsekvens av $\mathcal{F}$). Vi bruker chase-algoritmen (se læreboken, avsnitt 3.4.2) med den konvensjonen at vi ved likhet alltid bytter til den laveste indeksen. Vi får følgende tablå:

| $\mathbf{R}(R)$ | $A$ | $B$ | $C$ | $D$ | $E$ | $F$ |
|---|---|---|---|---|---|---|
| $ABDE$ | $a_0$ | $b_0$ | $c_1$ | $d_0$ | $e_0$ | $f_1$ |
| $ABDF$ | $a_0$ | $b_0$ | $c_2$ | $d_0$ | $e_1$ | $f_0$ |
| $ABEF$ | $a_0$ | $b_0$ | $c_3$ | $d_1$ | $e_0$ | $f_0$ |
| $ACDE$ | $a_0$ | $b_1$ | $c_0$ | $d_0$ | $e_0$ | $f_2$ |
| $ACDF$ | $a_0$ | $b_2$ | $c_0$ | $d_0$ | $e_2$ | $f_0$ |
| $ACEF$ | $a_0$ | $b_3$ | $c_0$ | $d_2$ | $e_0$ | $f_0$ |
| $BCDE$ | $a_1$ | $b_0$ | $c_0$ | $d_0$ | $e_0$ | $f_3$ |
| $BCDF$ | $a_2$ | $b_0$ | $c_0$ | $d_0$ | $e_3$ | $f_0$ |
| $BCEF$ | $a_3$ | $b_0$ | $c_0$ | $d_3$ | $e_0$ | $f_0$ |

Her ser vi at ingen av de to FDene i $\mathcal{F}$ kan anvendes, så algoritmen terminerer uten å gjøre noen endringer, så $\mathcal{F} \nvdash \sigma$. Dermed har vi bevist at det ikke finnes noen tapsfri dekomposisjon av $\mathbf{R}(R)$ hvor ingen av komponentene er en supernøkkel. Altså er $\mathbf{R}(R)$ i ETNF.

La oss så legge domenerestriksjonen $\Delta = (\mathrm{dom}(A) = \mathrm{dom}(B) = \{0,1\})$ til $\mathbf{R}(R)$. Med bare to mulige A-verdier må (minst) to av de tre nederste radene bli like i de tre første kolonnene. I følge $\mathcal{F}$ må de da bli like i de tre siste kolonnene også, og disse tre kolonnene får alle indeks 0 i de to tuplene. Det samme resonnementet på B-kolonnen i de tre midterste radene i tablået gir oss to nye rader med indeks 0 i de siste kolonnene. Den ubrukte FDen i $\mathcal{F}$ gir oss hele fire rader med indeks 0 i alle kolonnene. Altså har vi at $\mathcal{F}, \Delta \vdash \sigma$ som viser at $\mathbf{R}(R)$ ikke lenger er i ETNF.

I ETNF-artikkelen bevises det også at RFNF ⇒ ETNF ⇒ 4NF og at ingen av de motsatte implikasjonene holder. Den første implikasjonen er triviell. Den andre og de to bevisene for at ingen av implikasjonspilene kan snus, er for kompliserte til å gjengi her.

### 6.5.4 Oppsummering med advarsel mot termen PJ/NF

Vi har definert tre normalformer mellom 4NF og 5NF der vi har

$$\text{5NF} \Rightarrow \text{SKNF} \Rightarrow \text{RFNF} \Rightarrow \text{ETNF} \Rightarrow \text{4NF}$$

hvor ingen av implikasjonene kan snus.

Opprinnelig brukte Fagin betegnelsen PJ/NF (Project-Join Normal Form) for 5NF. (Se avsnitt 6.2.8)

Men vi mener at PJ/NF bør betegne den høyest mulige normalformen vi kan få ved bare å bruke projeksjoner som danner en ureduserbar tapsfri dekomposisjon.

Anta at vi har et relasjon $R$ med skjema $\mathbf{R}(R)$ på RFNF (eller høyere). Da vet vi at alle tapsfri dekomposisjoner av $\mathbf{R}(R)$ inneholder komponenter $U_1, \ldots, U_k$ som alle er supernøkler og som utgjør en (ikke nødvendigvis tapsfri) dekomposisjon av $\mathbf{R}(R)$. Men da har hver $\pi_{U_i}(R)$ like mange

forekomster som $R$, og det kan umulig være noe plass å tjene på å foreta dekomposisjonen.

Så hvis termen PJ/NF skal brukes, er ETNF den beste kandidaten. Men av historiske grunner er trolig det beste å ikke bruke PJ/NF som navn på noen normalform.

Til slutt, et lite hjertesukk: Da Codd i 1972 introduserte begrepet normalformer, hadde han ikke tenkt på at relasjoner kunne ha delvis overlappende kandidatnøkler. Da han, sammen med Boyce, kom med BCNF i 1974, burde han ha kalt det en rettelse av definisjonen av 3NF i stedet for å gi den et nytt navn. (Det var den gamle, nåværende, definisjonen av 3NF som burde ha fått nytt navn) Men nå er det for sent. Vi kan ikke forandre 40 år gamle godt innarbeidede begreper.

## 6.6 Dekomposisjonsalgoritmer

### 6.6.1 Minimale overdekninger

La $R$ være en relasjon, la $F$ og $G$ være to mengder med FDer over $R$. Da sier vi at $G$ *dekker* $F$ hvis $F \subseteq G^+$ (det er ekvivalent med at $F^+ \subseteq G^+$), og at $F$ og $G$ er *ekvivalente* hvis de dekker hverandre, dvs at $F^+ = G^+$. Hvis $G$ dekker $F$, kalles $G$ en *overdekning* av $F$.

Algoritme for å finne minimale overdekninger:

(1) Initialiser:
$G := F$;

(2) Lag atomære høyresider:
For hver $X \to Y$ i $G$ hvor $Y = \{A_{k_1}, \ldots, A_{k_p}\}$ ikke er atomær,
erstatt $X \to Y$ i $G$ med de $p$ FDene $X \to A_{k_i}$, $1 \leq i \leq p$.

(3) Gjør venstresidene minimale:
For hver $X \to A$ i $G$ og hver $B \in X$:
Beregn $(X \setminus \{B\})^+$ med hensyn på $G$.
Hvis $A \in (X \setminus \{B\})^+$, så erstatt $X \to A$ med $(X \setminus \{B\}) \to A$ i $G$.

(4) Fjern overflødige FDer:
For hver $X \to A$ i $G$:
Beregn $X^+$ med hensyn på $G \setminus \{X \to A\}$.
Hvis $A \in X^+$, fjern $X \to A$ fra $G$.

En minimal overdekning av $F$ er altså en minimal mengde $G$ av elementære FDer slik at $G^+ = F^+$.

### 6.6.2 Tapsfri FD-bevarende dekomposisjon til EKNF

La $U$ være en komponent i $R$, og la $F$ være en mengde FDer over $R$.

Da definerer vi mengden av *lokalt kontrollerbare FDer* i $U$ som

$$F|_U = \{X \to Y \mid X \to Y \in F^+ \wedge XY \subseteq U\}$$

Merk at det står $F^+$, og ikke $F$, i definisjonen ovenfor.

La $\mathcal{D} = \{U_1, \ldots, U_m\}$ være en dekomposisjon av $R$. Dersom

$$(\bigcup_{k=1}^{m} F|_{U_k})^+ = F^+$$

sier vi at $\mathcal{D}$ er en *FD-bevarende* dekomposisjon. Som navnet sier, mister vi ingen FDer ved FD-bevarende dekomposisjoner.

Algoritme for å konstruere en tapsfri FD-bevarende dekomposisjon til EKNF fra en relasjon $R$ og en mengde FDer $F$ over $R$ (algoritme 3.26 på side 103 i læreboken):

(1) Finn en minimal overdekning $G$ for $F$ (se ovenfor).

(2) For hver $X = \{B_1, \ldots, B_q\}$ som opptrer som venstreside i $G$, lag en komponent $U_X = \{B_1, \ldots, B_q, A_1, \ldots, A_p\}$ der $\{X \to A_i \mid 1 \leq i \leq p\}$ er mengden av alle FDer i $G$ som har $X$ som venstreside.

(3) Samle de komponentene i $R$ som ikke er med i noen $U_X$, i en egen komponent, $U_0$ (ofte vil $U_0 = \emptyset$).

(4) Hvis ingen av komponentene er en supernøkkel, utvid $U_0$ til en kandidatnøkkel (alle supernøkler må omfatte $U_0$).

Merk at denne algoritmen ikke nødvendigvis finner en ureduserbar tapsfri dekomposisjon av $R$. Det kan godt tenkes at vi kan fjerne noen komponenter og fortsatt ha en tapsfri dekomposisjon, men da trenger den ikke lenger å være FD-bevarende.

### 6.6.3 De elementære FDene i $F^+$, $F_E^+$

Selv om $F$ inneholder få FDer, er $F^+$ en stor mengde. Eksempelvis er $\emptyset^+$ alle trivielle FDer i $R$. Hvis ariteten til $R$ er $n$, så er $2^n \leq \#(F^+) \leq 2^{2n}$ der $\#(F^+)$ er antall FDer i $F^+$. Heldigvis er det tilstrekkelig å sjekke de elementære FDene i $F^+$ når vi skal vise noe for alle FDer i $F^+$.

Husk: Elementære FDer har atomære høyresider og minimale venstresider.

Hvis $F$ er en mengde FDer i en relasjon $R$, betegner vi mengden av elementære FDer i $F^+$ med $F_E^+$. (Merk at $F_E^+$ er entydig bestemt av $F^+$ som

igjen er entydig bestemt av $F$. I motsetning til minimale overdekninger, finnes det bare én $F_E^+$.)

For å spare skriving, slår vi gjerne sammen alle elementære FDer med samme determinant. Hvis $X \to A_i$, $1 \leq i \leq m$, er alle elementære FDer med $X$ som determinant, lar vi $X \to A_1 A_2 \cdots A_m$ representere dem i beskrivelsen av $F_E^+$. Merk imidlertid at dette bare er en kompakt skrivemåte for $m$ forskjellige elementære FDer.

En av årsakene til at vi introduserer $F_E^+$, er ligningen

$$(F|_U)_E^+ = \{X \to A \mid XA \subseteq U \wedge X \to A \in F_E^+\}$$

som gjør det lett å bestemme de lokalt kontrollerbare FDene.

Algoritme for å finne $F_E^+$ gitt $F$:

(1) Initialiser:
Sett $F_E^+$ lik $F$.

(2) Rensk ut de ikke-elementære FDene:
Splitt alle FDene med sammensatte høyresider slik at alle høyresider i $F_E^+$ blir atomære. Fjern alle trivielle FDer fra $F_E^+$. Mens det fortsatt finnes $X \to A$ og $Y \to A$ som begge er i $F_E^+$, og $X \subset Y$, fjern $Y \to A$ fra $F_E^+$.

(3) Finn avledbare elementære FDer med kjent determinant:
Begynn med de atomære venstresidene i $F_E^+$, deretter de med to attributter, så de med tre osv, og beregn $X^+$ for hver venstreside $X$ i $F_E^+$.

For hver $A \in X^+ \setminus X$, sjekk om $X \to A$ er ny og elementær (det er nok å sjekke mot FDene som allerede er i $F_E^+$), og føy den i så fall til $F_E^+$.

(4) Finn avledbare elementære FDer med nye determinanter:
For hvert par $X \to A$ og $Y \to B$ i $F_E^+$ med $A \in Y$ og $B \notin X^+$:

(a) Sett $Y' = Y \setminus \{A\}$ (dvs $Y = AY'$) og $Z = XY'$

(b) Sjekk om det finnes noen $W \to B$ i $F_E^+$ med $W \subseteq Z$.

(c) Hvis ingen slik $W$ finnes, legg $Z \to B$ inn i $F_E^+$, beregn $Z^+$, sjekk om $Z$ er determinant for flere elementære FDer, og legg i så fall også disse inn i $F_E^+$.

Gjenta (4) til det ikke blir generert flere elementære FDer.

**Merk**:
I (4) bruker vi Armstrongs regler til å finne nye determinanter: Vi har de to FDene $X \to A$ og $Y \to B$. Vi bruker utvidelsesregelen på den første og får $Z = XY' \to AY' = Y$. Den transitive regelen brukt på $Z \to Y$ og $Y \to B$ gir så at $Z \to B$.

### 6.6.4 Å avgjøre om en dekomposisjon er FD-bevarende

La $F$ være en mengde FDer over en relasjon $R$ og la $\mathcal{D}$ være en dekomposisjon av $R$. For å avgjøre om $\mathcal{D}$ bevarer $F$, gjør vi følgende:

(1) Beregn $F_E^+$

(2) Sett $G = \{X \rightarrow A \mid \exists U(XA \subseteq U \in \mathcal{D} \wedge X \rightarrow A \in F_E^+)\}$

(3) Beregn $G_E^+$

(4) $\mathcal{D}$ er FD-bevarende hvis, og bare hvis, $G_E^+ = F_E^+$

### 6.6.5 Tapsfri dekomposisjon til BCNF

La $F$ være en mengde FDer over en relasjon $R$.

(1) Initialiser:
Beregn $F_E^+$ og sett $\mathcal{D} = \{\mathbf{R}(R)\}$

(2) Mens det finnes en $Q \in \mathcal{D}$ som ikke er på BCNF

  (a) Velg en FD $X \rightarrow Y$ i $F_E^+$ med $XY \subseteq Q$ som bryter med BCNF i $Q$ (Velg $Y$ så stor som mulig)

  (b) Erstatt $Q$ med de to komponentene $Q \setminus Y$ og $XY$ (i $\mathcal{D}$)

### 6.6.6 Tapsfri dekomposisjon til 4NF

La $R$ være en relasjon.

(1) Bruk algoritmen ovenfor og lag en tapsfri dekomposisjon $\mathcal{D}$ av $R$ hvor alle komponentene i $\mathcal{D}$ er på BCNF

(2) Mens det finnes en $Q \in \mathcal{D}$ som inneholder en ikketriviell MVD $X \twoheadrightarrow Y$ (med $X$ og $Y$ disjunkte) som *ikke* er en FD, erstatt $Q$ med de to komponentene $Q \setminus Y$ og $XY$ (i $\mathcal{D}$)

## 6.7 Usammenhengende dekomposisjoner

Vi skal her ta for oss noen spesialtilfeller som forholdsvis sjelden forekommer i praksis.

### 6.7.1 Sammenhengskomponenter

La $R$ være en relasjon, og la $\mathcal{D} = \{X_1, \ldots, X_n\} \subseteq \mathcal{P}(\mathbf{R}(R))$. Vi definerer *sammenhengskomponenten* til en $X_k \in \mathcal{D}$, $\overline{X}_k$, ved følgende lille algoritme:

(1) Sett initielt $\overline{X}_k := X_k$

(2) Så lenge det finnes en $X_i$ som har minst et felles attributt med $\overline{X}_k$ uten at $X_i \subseteq \overline{X}_k$, utvid $\overline{X}_k$ med $X_i$

Denne definisjonen gir umiddelbart at $\overline{X}_i \cap \overline{X}_k \neq \emptyset \Rightarrow \overline{X}_i = \overline{X}_k$, så sammenhengskomponentene i $\mathcal{D}$ er parvis disjunkte.

En dekomposisjon som bare har én sammenhengskomponent, kalles *sammenhengende.*

La så $\mathcal{D}$ være en tapsfri dekomposisjon som ikke er sammenhengende, og la $U_1, \ldots, U_k$ være sammenhengskomponentene i $\mathcal{D}$, der $k \geq 2$. Siden $\mathcal{D}$ er tapsfri, gir det at

$$R = \pi_{U_1}(R) \bowtie \cdots \bowtie \pi_{U_k}(R) = \pi_{U_1}(R) \,\mathsf{prod}\, \cdots \,\mathsf{prod}\, \pi_{U_k}(R)$$

Dette viser at $r(R)$ er et kartesisk produkt. Altså er det nok å lagre hver av komponentene og (svært) dumt å lagre hele $R$.

### 6.7.2 MVDer og FDer med $\emptyset$ som venstreside

I en ikke-sammenhengende dekomposisjon med to komponenter, $U_1$ og $U_2$, er $U_1 \cap U_2 = \emptyset$. Dersom $\{U_1, U_2\}$ i tillegg er tapsfri, har vi en MVD $\emptyset \twoheadrightarrow U_1$.

Det omvendte gjelder også. Hvis $\emptyset \twoheadrightarrow U$ i en relasjon $R$, er $U$ en sammenhengskomponent i en tapsfri dekomposisjon av $R$. Merk at vi alltid har en triviell MVD $\emptyset \twoheadrightarrow \mathbf{R}(R)$.

Hvis vi har en FD $\emptyset \rightarrow Y$ i en relasjon $R$, betyr det at alle forekomstene i $R$ må ha samme $Y$-verdi.

# 7 Relasjonsdatabaser

En relasjonsdatabase er en database hvor dataene lagres som relasjoner. På et gitt tidspunkt er databasen i en *databasetilstand* som består av en mengde relasjonsekstensjoner.

Hva som er lovlige tilstander og transisjoner (overgang mellom to tilstander), beskrives i databasens begrepsmessige skjema:

## 7.1 Relasjonsdatabaseskjemaer

Et relasjonsdatabaseskjema (RDBS) er et kvadruppel $(\mathcal{D}, \mathcal{R}, \mathcal{N}, \mathcal{I})$, der $\mathcal{D}$ er en endelig mengde domener, $\mathcal{R}$ er en endelig mengde relasjonsskjemaer, $\mathcal{N}$ er en endelig mengde med nilmerker, og $\mathcal{I}$ er en endelig mengde regler, kalt *integritetsreglene* i skjemaet, slik at

- for alle $D \in \mathcal{D}$ gjelder at $D \cap \mathcal{N} = \emptyset$
- domenene til alle attributter i alle relasjonsskjemaeme i $\mathcal{R}$ er med i $\mathcal{D}$
- alle integritetsreglene i $\mathcal{I}$ er 1. ordens logiske uttrykk over domenene i $\mathcal{D}$ og deres atomære verdier, attributtene i relasjonsskjemaene i $\mathcal{R}$ og ordning i tid (i betydningen: før – etter)

## 7.2 Relasjonsdatabasetilstander

En relasjonsdatabasetilstand for et RDBS $(\mathcal{D}, \mathcal{R}, \mathcal{N}, \mathcal{I})$ er en mengde ekstensjoner $\{r(R_1), \ldots, r(R_q)\}$ som tilfredsstiller integritetsreglene i $\mathcal{I}$, og hvor $\mathcal{R} = \{\mathbf{R}(R_1), \ldots, \mathbf{R}(R_q)\}$.