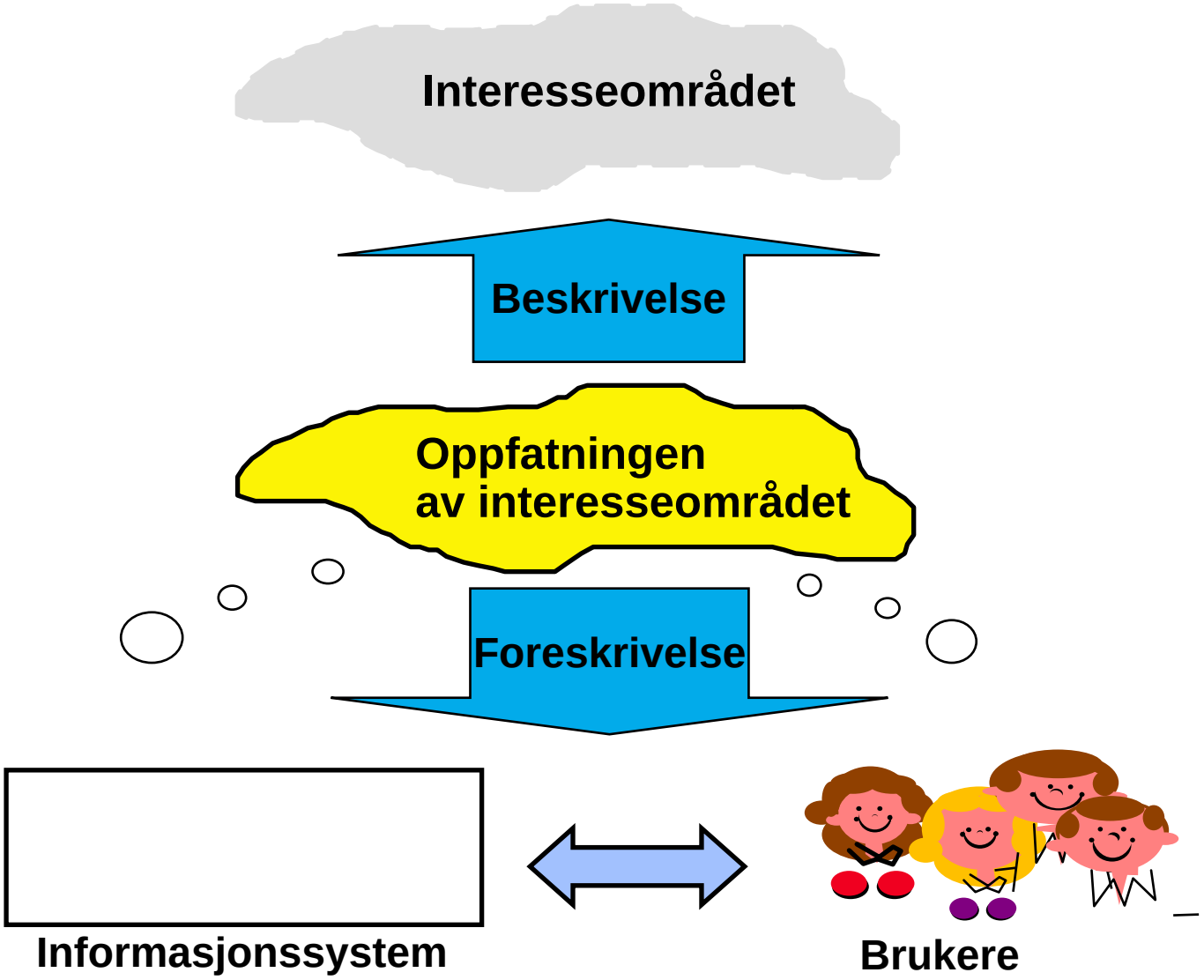
Den redundansfri datamodellen

jfr. *Systemutvikling –*
fra kjernen og ut, fra skallet og inn
kapittel 6

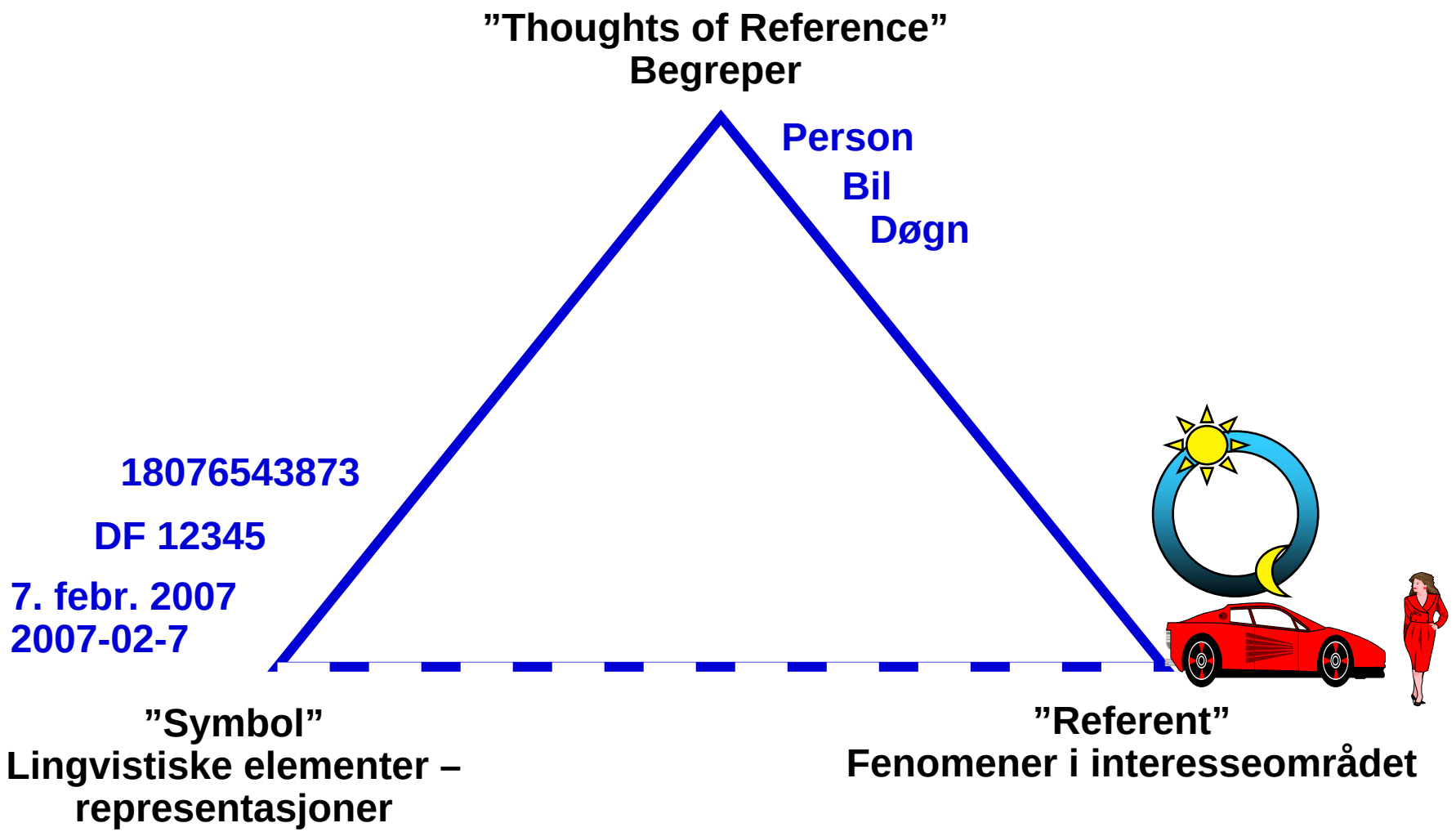
Dagens tema

- ”Individer” i interesseområdet
- Redundansfrihet
  – ingen dobbeltlagringer eller avledninger
- Gruppering, normalisering eller intuisjon?
- Begrepsdannelse
- Høyere ordens assosiasjoner
- Litt om tid

Modellenes to formål



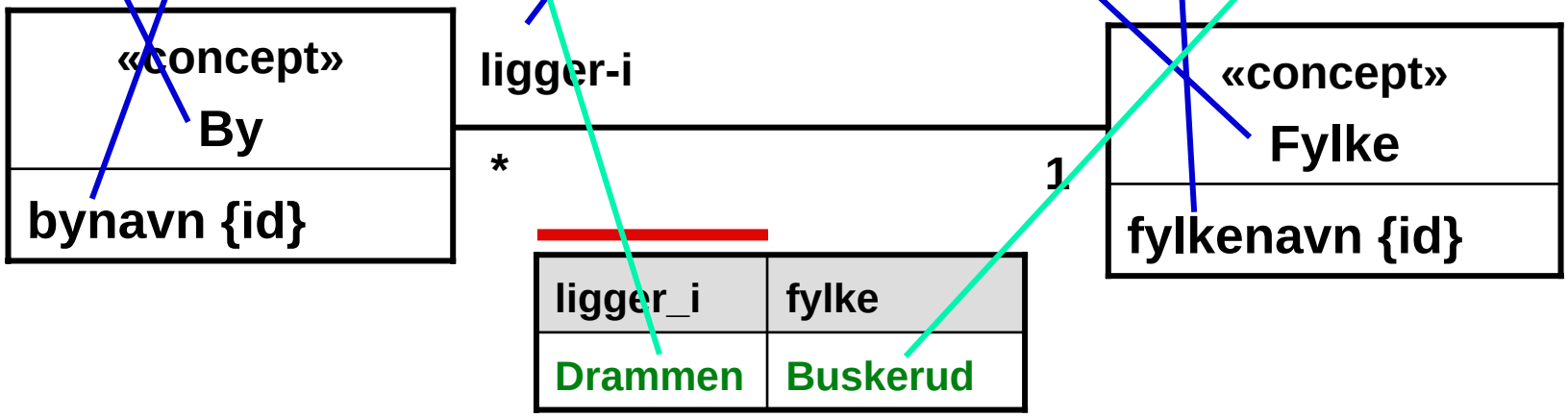
Figur 5-2. Ogdens trekant



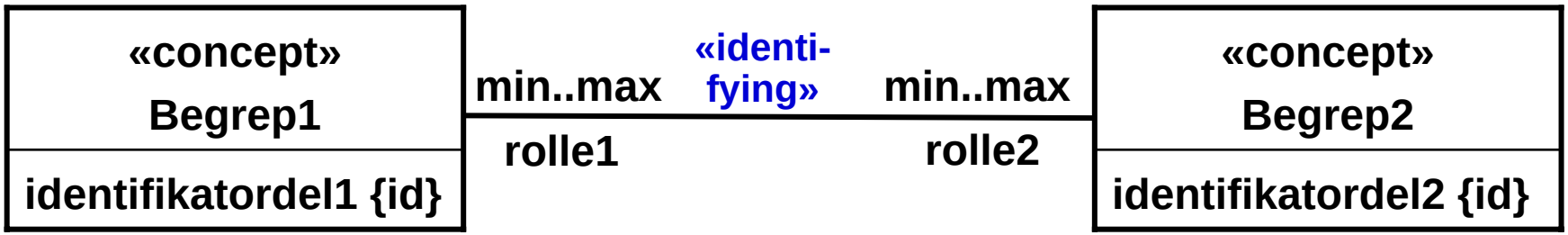
Fra naturlig språk til datamodell

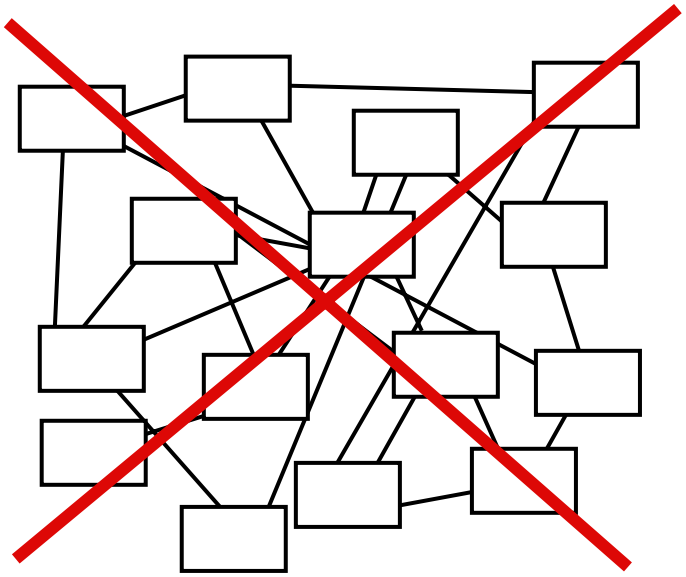
Drammen ligger i Buskerud

By med bynavn Drammen ligger i fylke med fylkenavn Buskerud

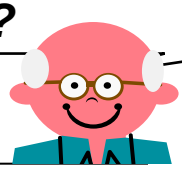


Den grunnleggende konstruksjonen – det elementære utsagnet





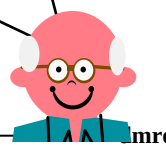
Greitt, men hvordan bygger vi modeller med den?



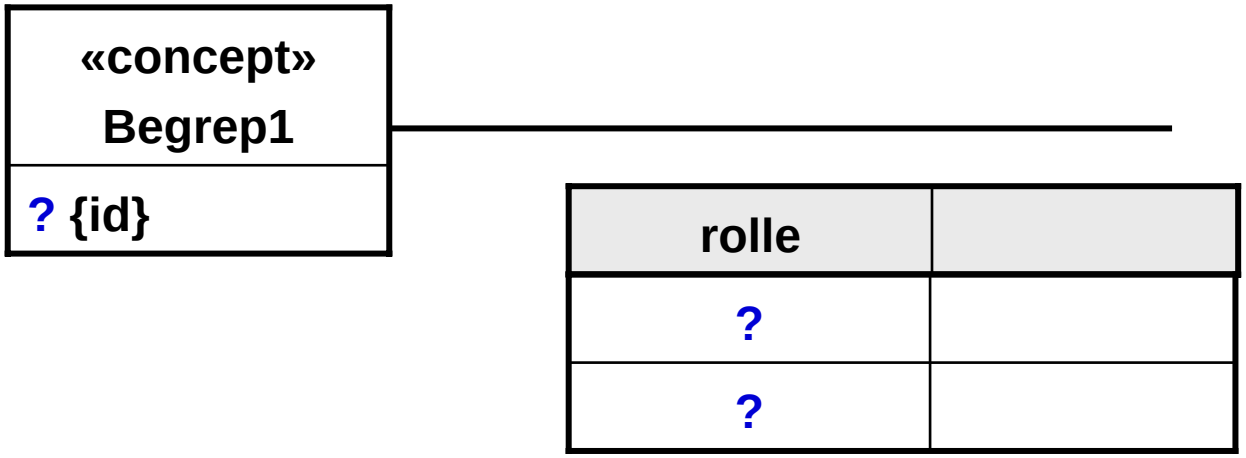
Datamodellererens tre bud

1. *Du skal kunne identifisere begrepsforekomstene!*
2. *Du skal ikke ha dobbeltlagringer!*
3. *Du skal ikke ha avledninger!*

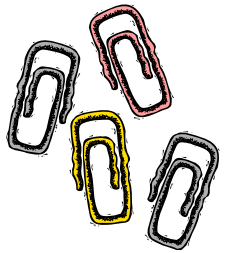
Hvis du allikevel (av effektivitetsgrunner) har dobbeltlagringer og avledninger i databasen, må de være under kontroll!



Identifiserbare begrepsforekomster



- Hvilke ”individer” (begrepsforekomster) skal vi vite noe om?
- Hvordan representerer vi disse individene?
- Eksempel: Hvordan ser vi på en haug med binders?



Eksempel på dobbeltlagring av opplysning

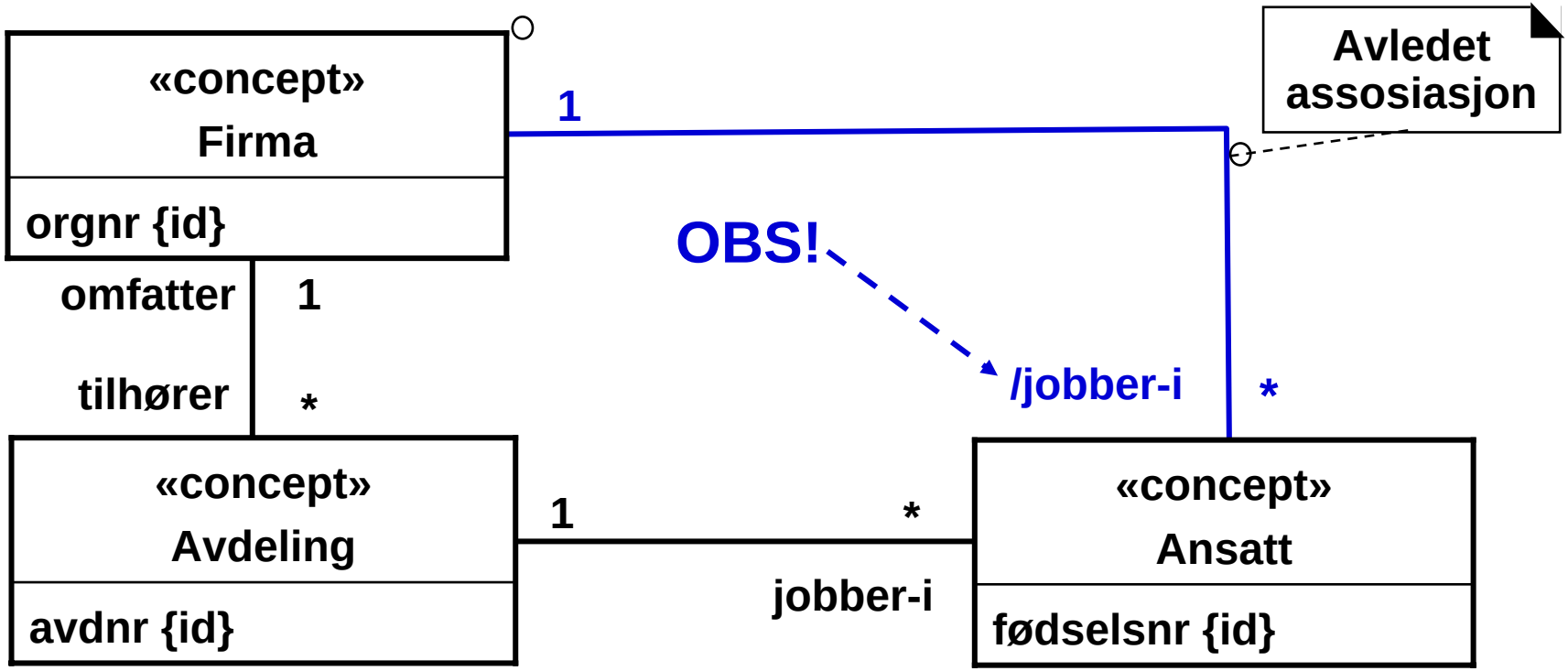
Person
navn {id}
gateadresse
postnr
poststed

innebygd elementært utsagn

navn	gateadresse	postnr	poststed
Dal	Storgata 7	1400	Ski
Li	Lilleveien 3	1400	Ski
Fjell	Nyveien 20	1500	Moss

Dobbeltlagrede opplysninger åpner for inkonsistens i forekomstene!

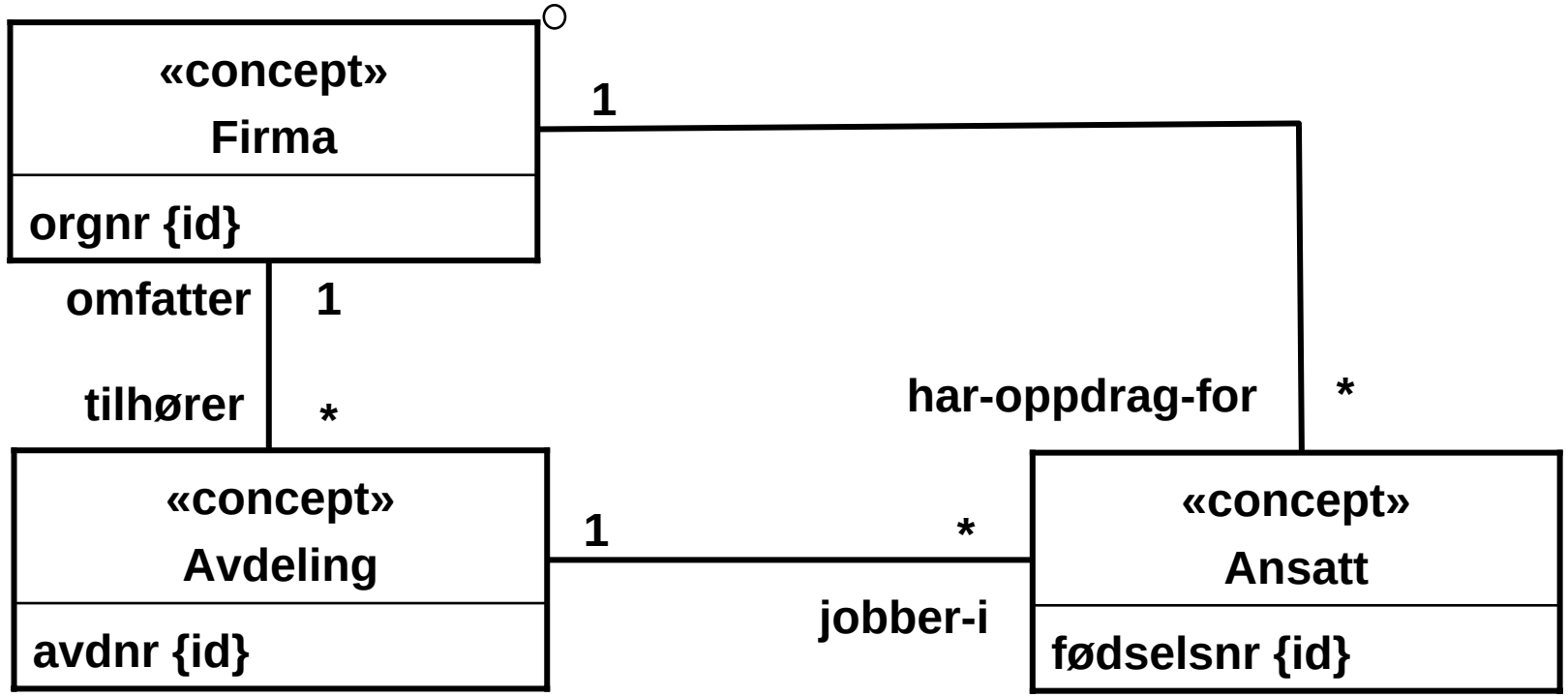
Eksempel på avledet opplysning



Ansatt jobber-i Avdeling ∧ Avdeling tilhører Firma
⇒ Ansatt jobber-i Firma

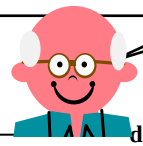
Avledede opplysninger åpner for inkonsistens i forekomstene!

Eksempel på *ikke* avledet opplysning



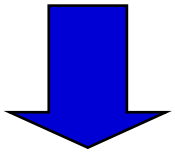
Ansatt jobber-i Avdeling ∧ Avdeling tilhører Firma
⇏ Ansatt har-oppdrag-for Firma

Det er altså ikke nok å se på bare hvordan modellen ser ut!



En unormalisert modell

Person
navn {id}
gateadresse
postnr
poststed

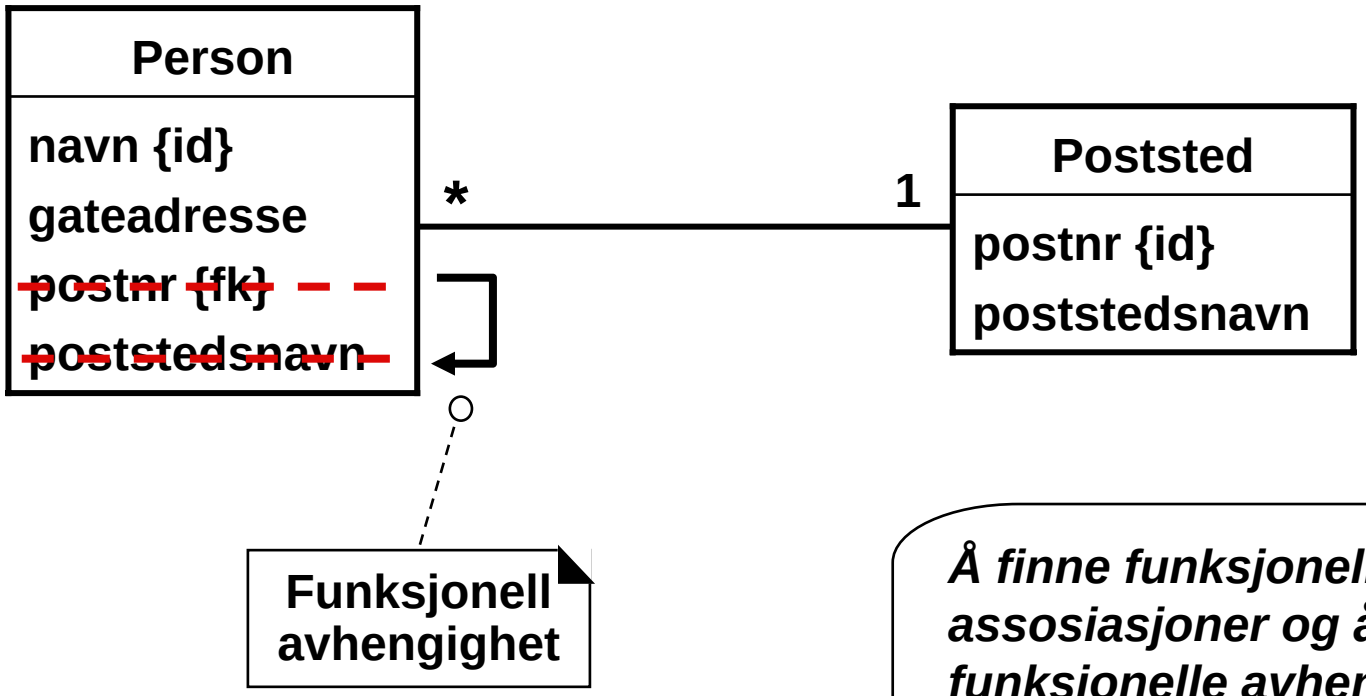


Funksjonell avhengighet

navn	gateadresse	postnr	poststed
Dal	Storgata 7	1400	Ski
Li	Lilleveien 3	1400	Ski
Fjell	Nyveien 20	1500	Moss

En datamodell (og en database) som ikke er normalisert, vil åpne for dobbeltlagring av opplysninger

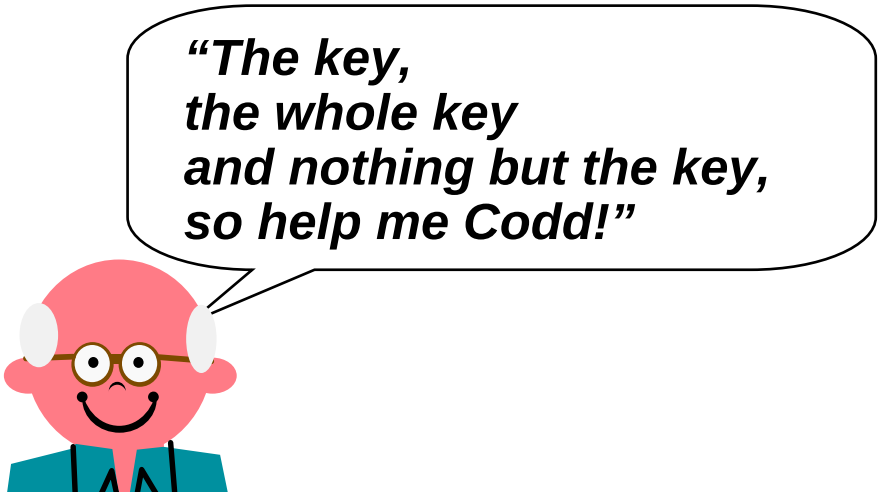
En funksjonell avhengighet skjuler en assosiasjon som skilles ut



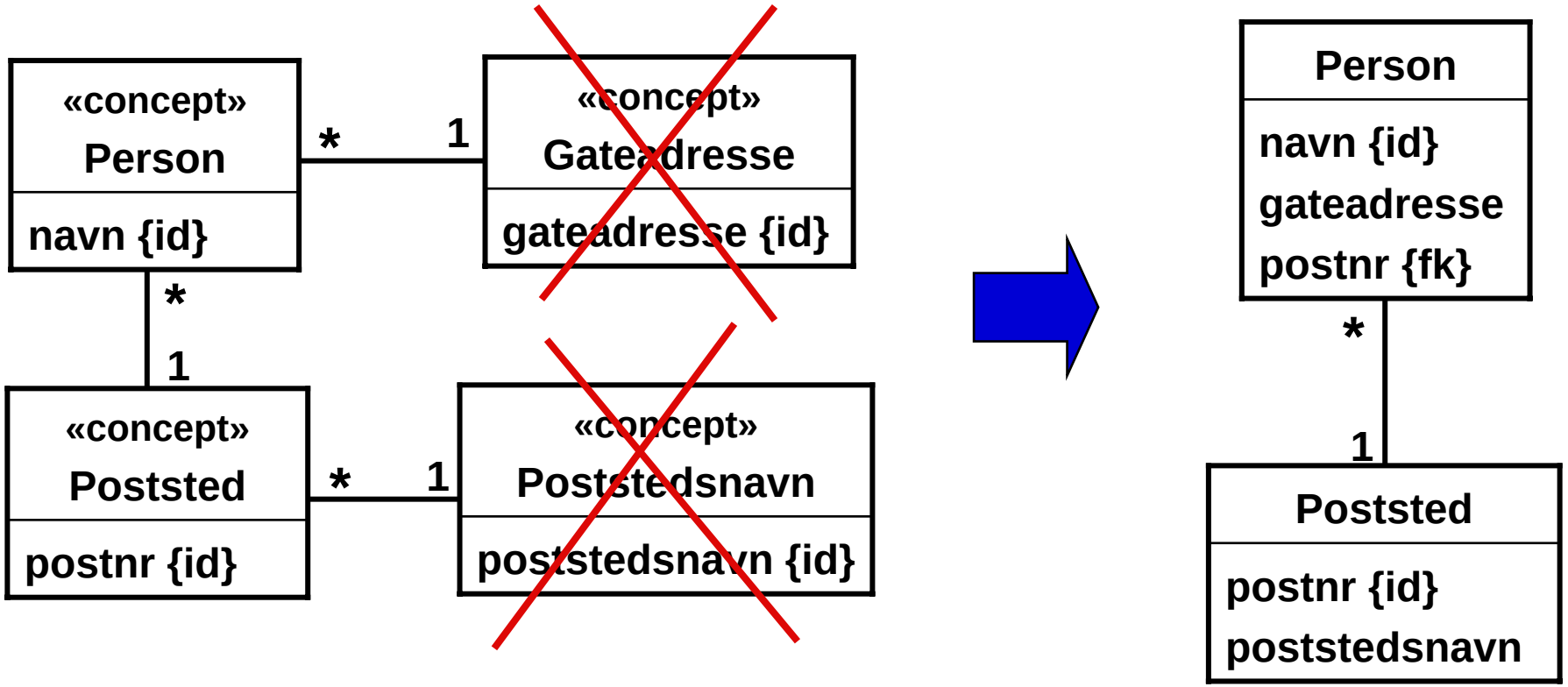
Å finne funksjonelle assosiasjoner og å finne funksjonelle avhengigheter – det er samme sak

Normaliseringsteori

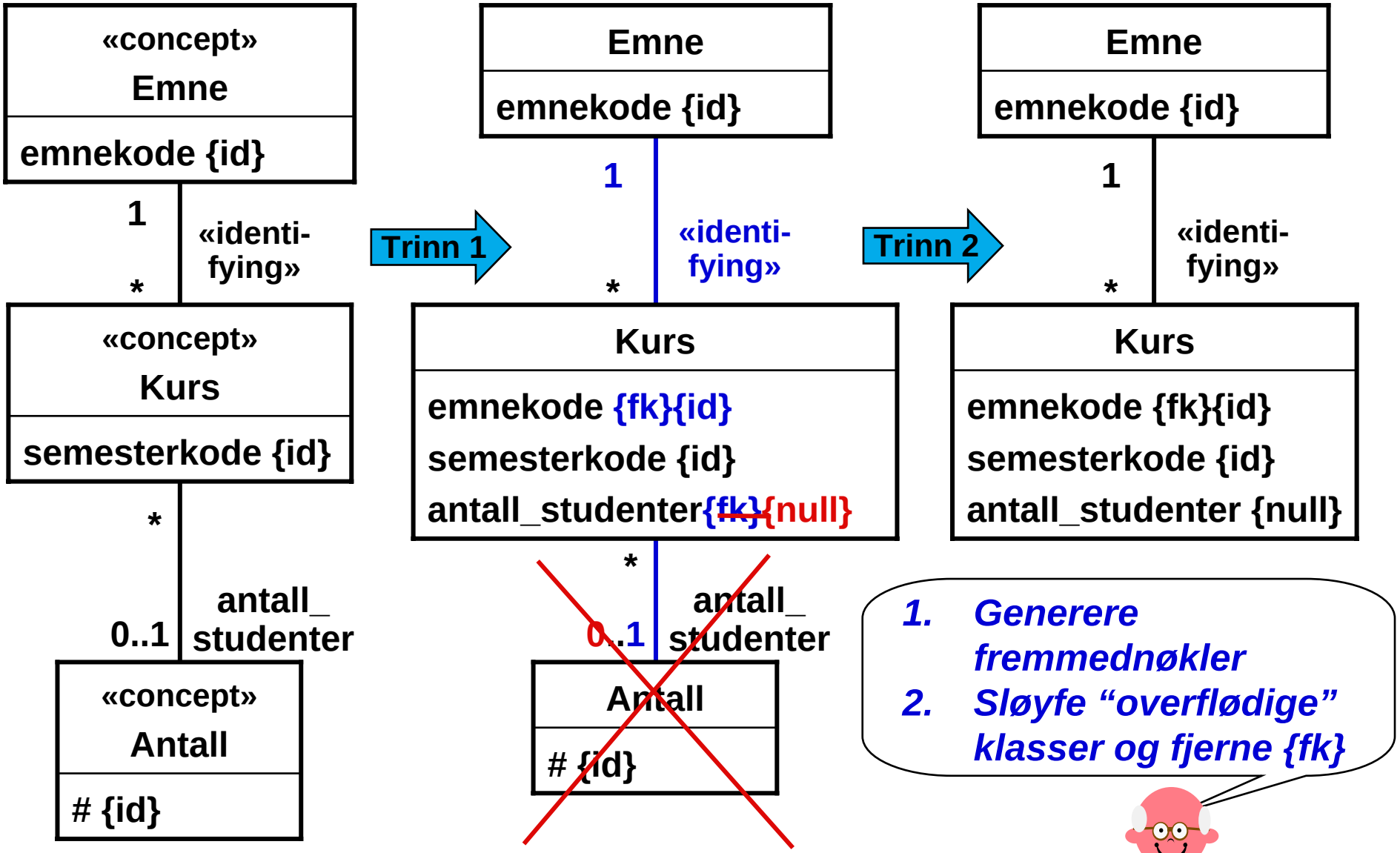
- Slike oppdelinger som vi har sett et eksempel på her er sentrale i forbindelse med såkalt *normalisering* i relasjonsdatabaseteorien.
- Hovedbudskapet er at verdien i et attributt skal være entydig bestemt av verdien på en av kandidatnøklene (i praksis primærnøkkelen)



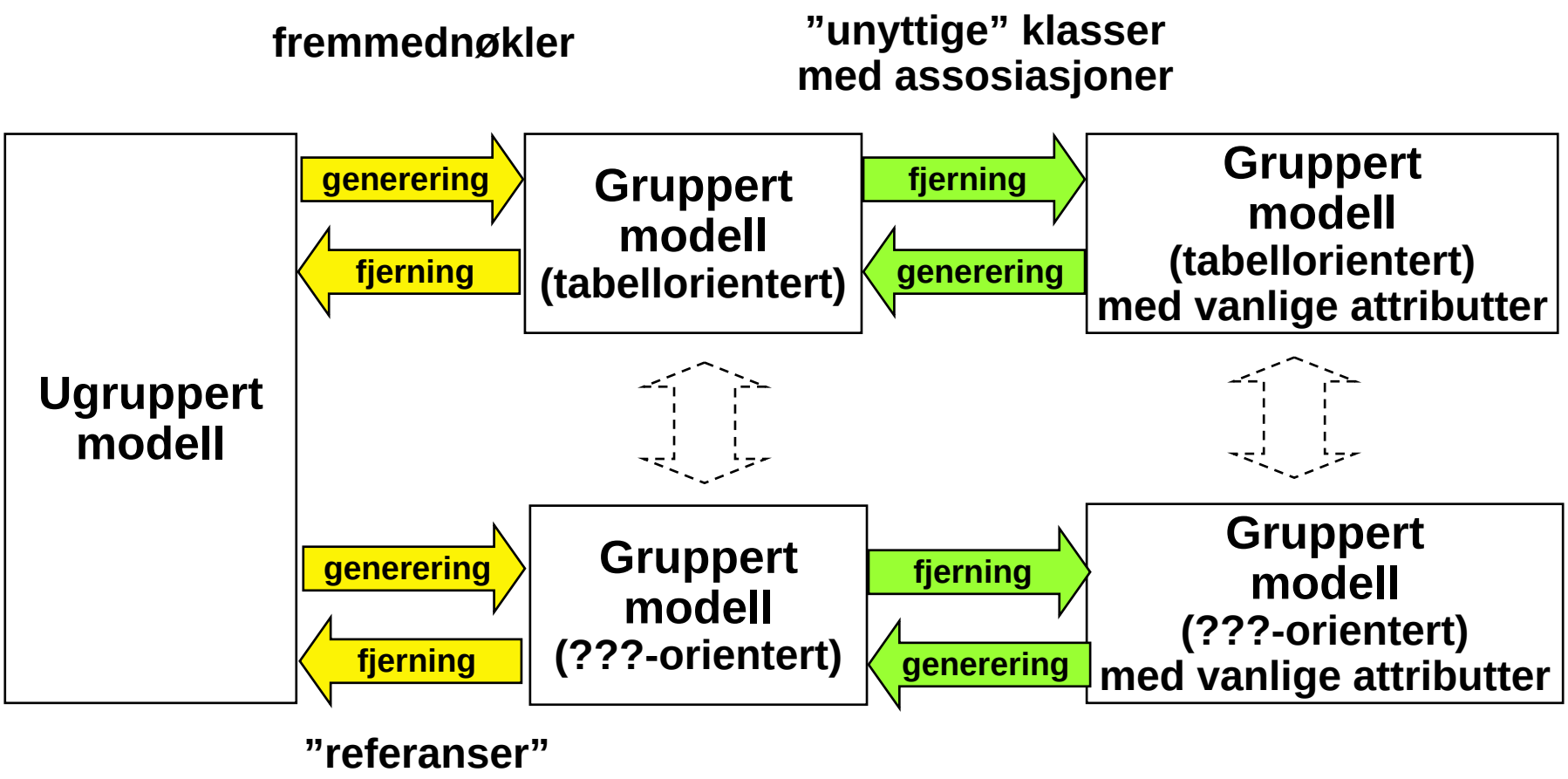
Alternativet: Gruppering av ugruppert modell



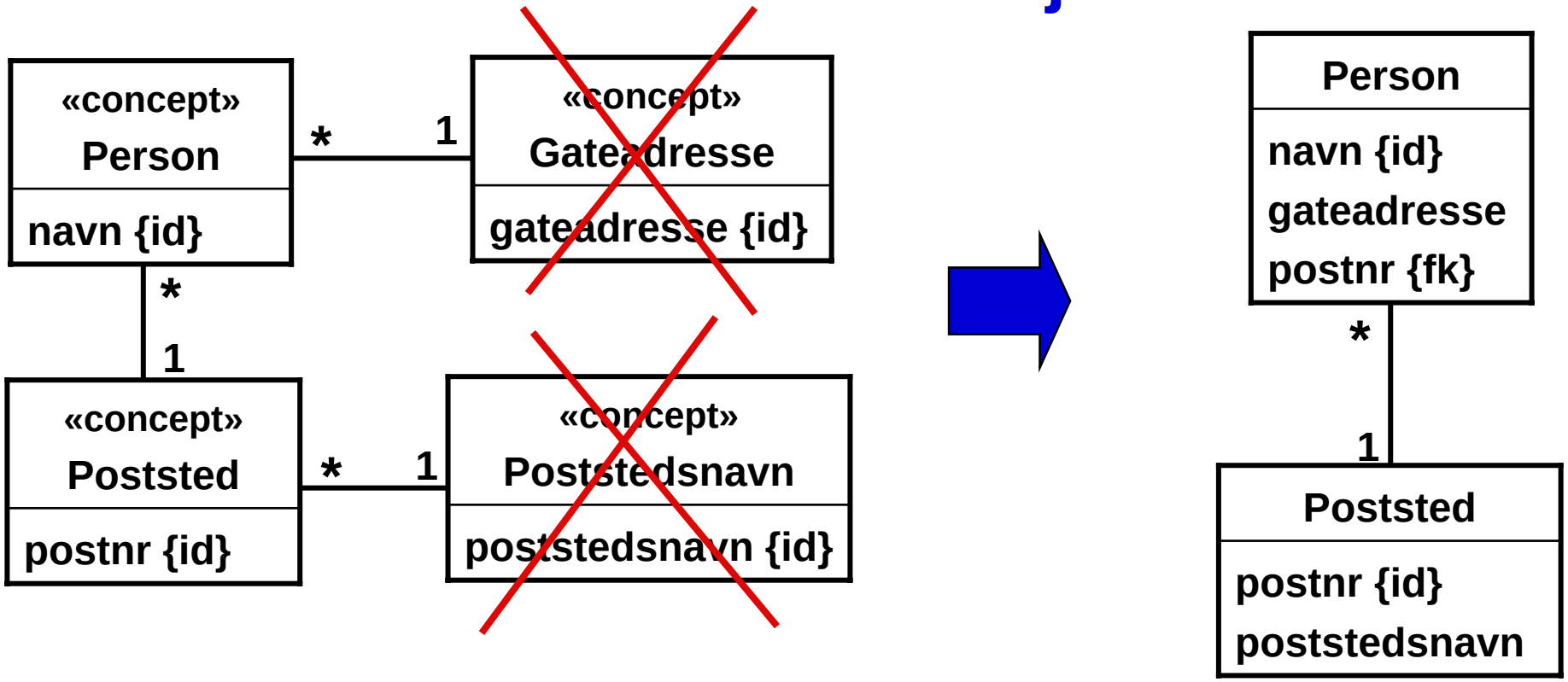
Ugruppert og gruppert modell - eksempel



Ugrupperte og grupperte modeller



Fra normalisert modell til relasjonsdatabase

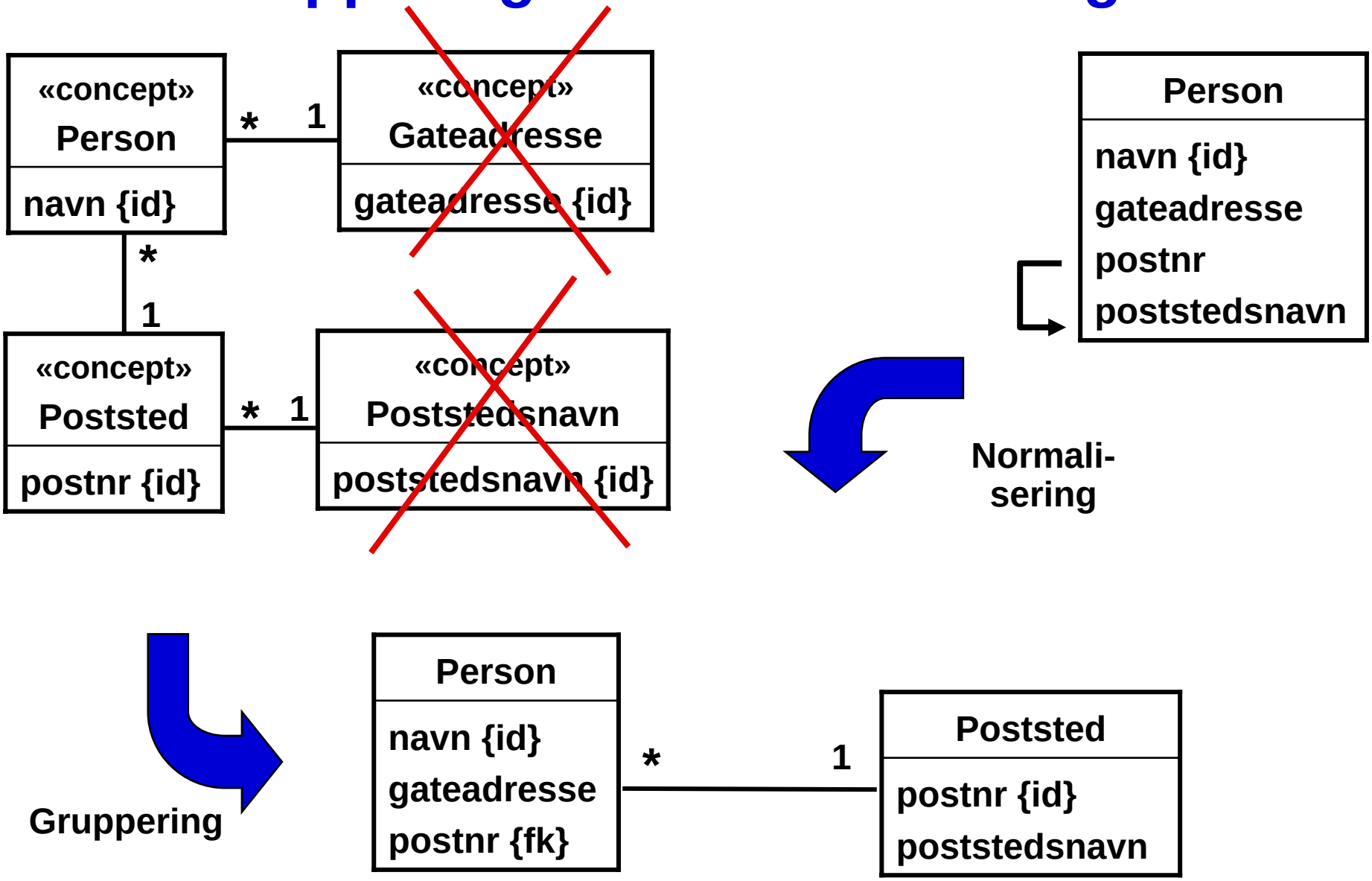


navn	gateadresse	postnr
Dal	Storgata 7	1400
Li	Lilleveien 3	1400
Fjell	Nyveien 20	1500

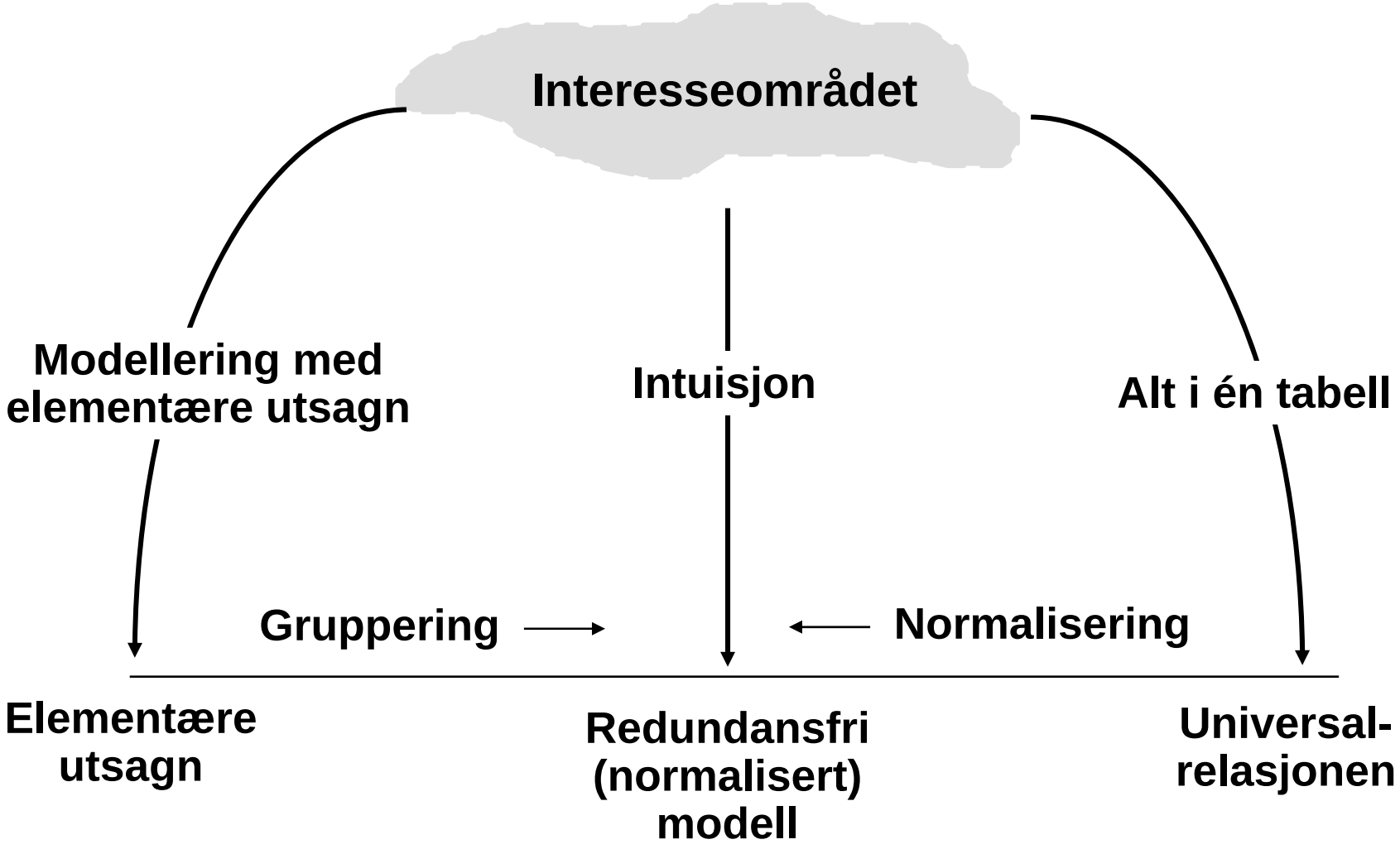
{subset}

postnr	poststed
1400	Ski
1500	Moss

Gruppering eller normalisering?



Ulike veier til den redundansfri modellen



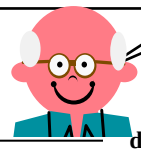
Kontroll av grupperte modeller

Hvis du går direkte på en gruppert modell, sjekk for hvert attributt:

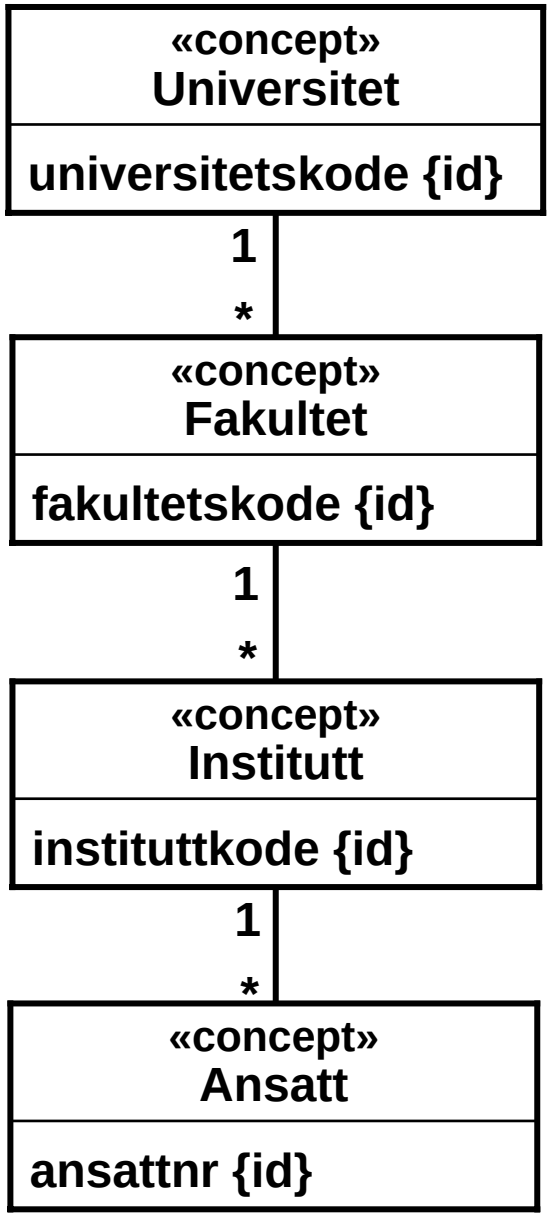
- **Er det høyst én verdi for attributtet?**
- **Er verdien bestemt av primærnøkkelen, hele primærnøkkelen og intet annet enn primærnøkkelen?**

Person
navn {id}
telefon
e-post
gateadresse
postnr
poststed

Hvordan faller denne kontrollen ut for klassen Person?



Assosiasjoner i et hierarki



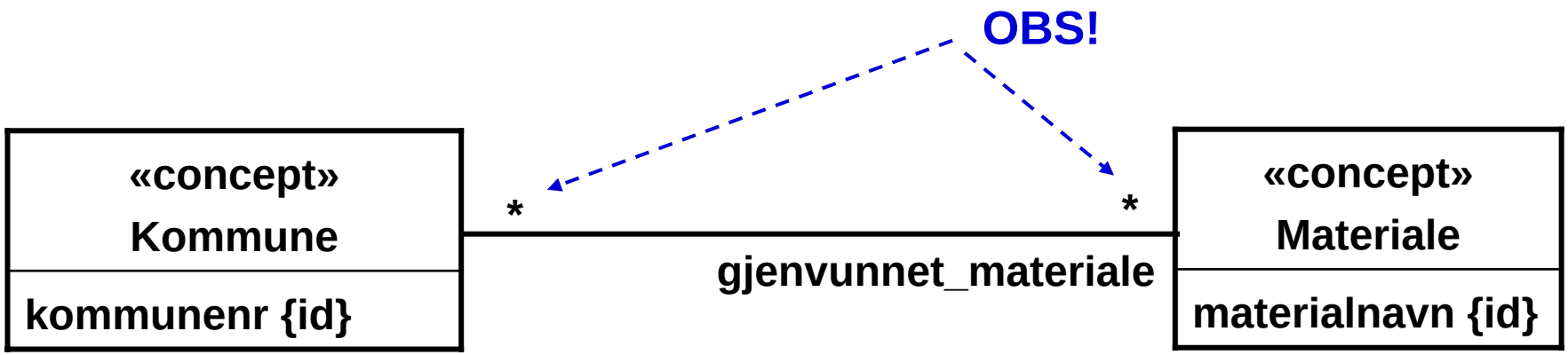
- Med hva skal vi assosiere
  - Universitetets sentralbordnummer
  - ansatts jobbadresse
  - ansatts telefonnummer
  - antall studenter

?

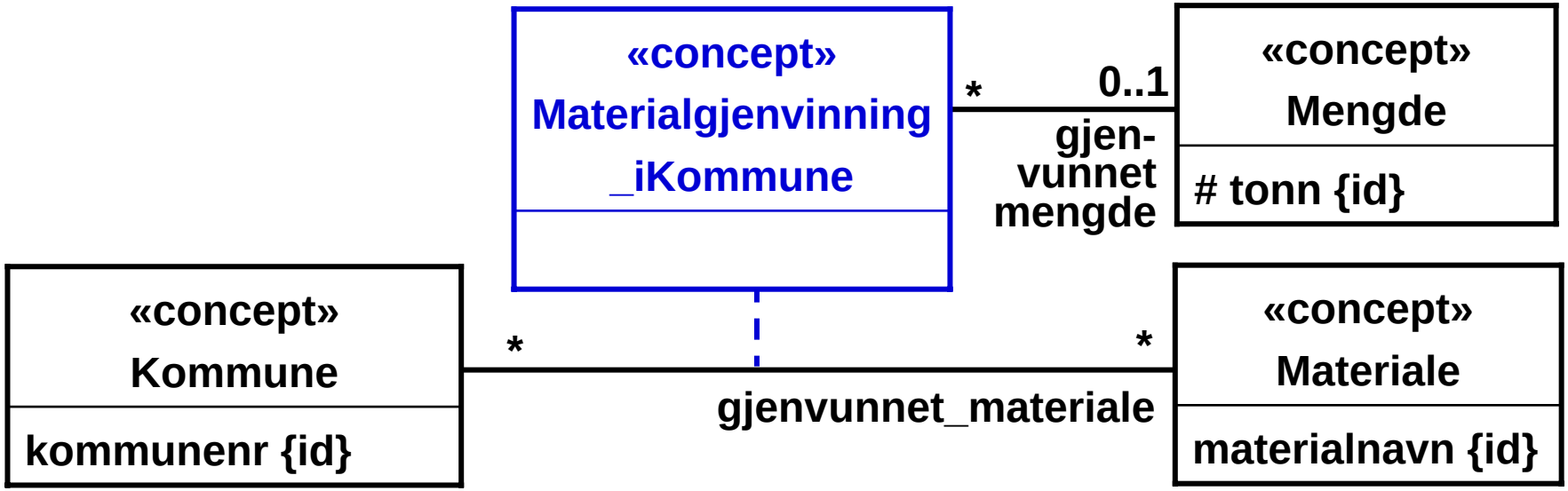
Legg assosiasjonene så høyt opp i hierarkiet som mulig!

Begrepsdannelse

En mange-til-mange-assosiasjon kan tolkes som et begrep

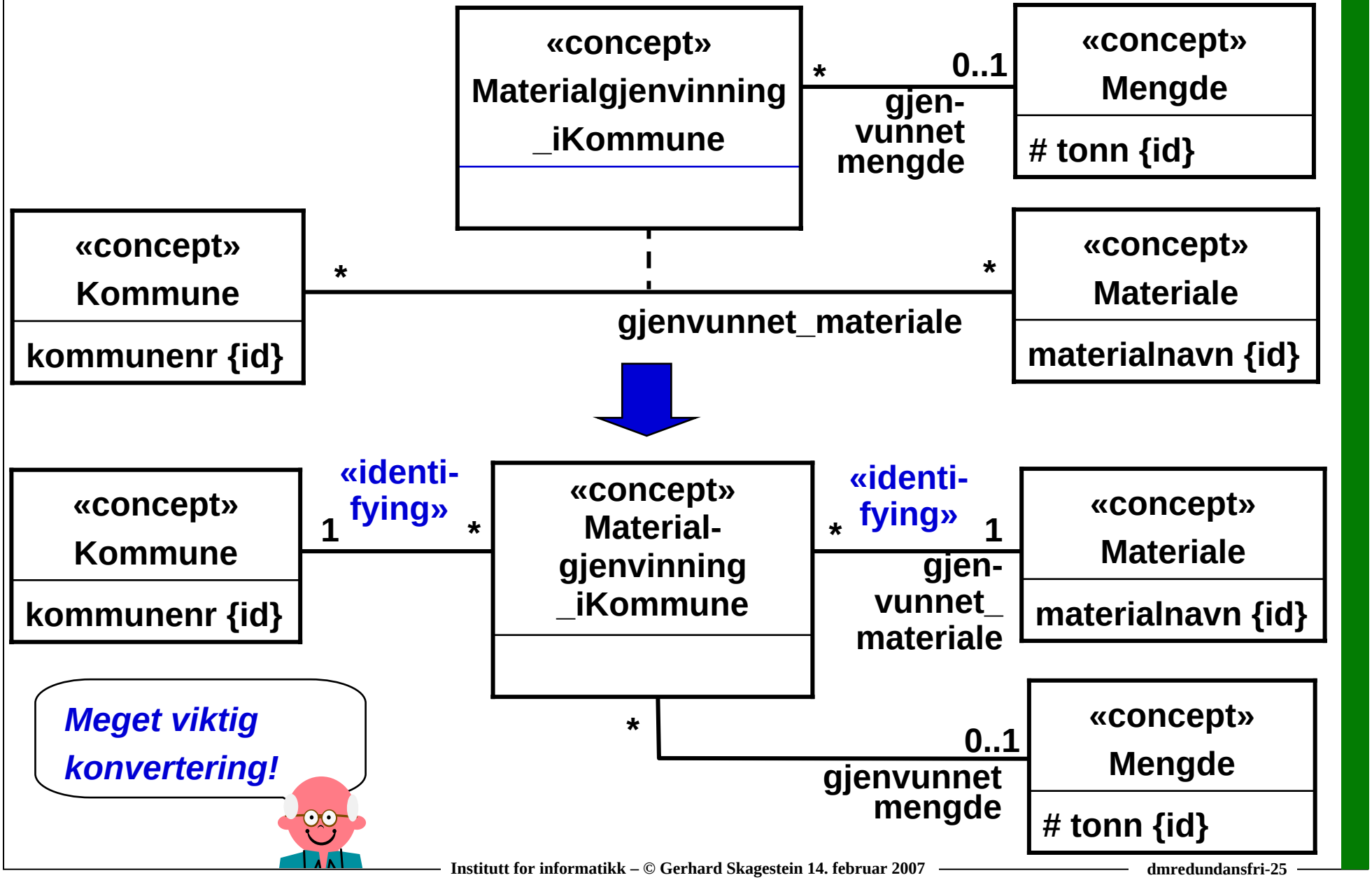


Mange-til-mange-assosiasjon med assosiasjonsklasse



jfr. lærebokas figur 5-7 og 5-8

Figur 5-9. Assosiasjonsklassen erstattes med en vanlig klasse



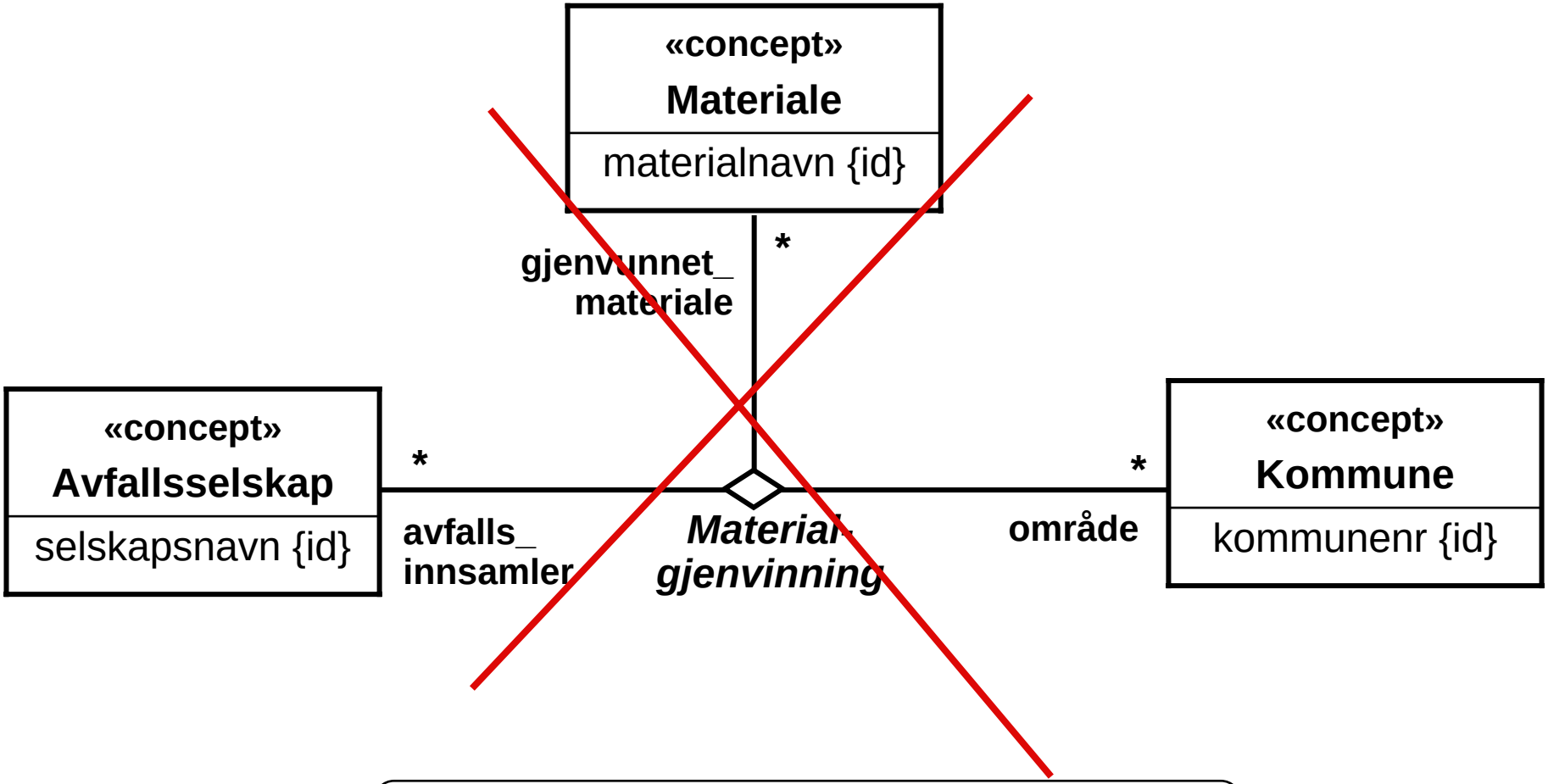
Høyere ordens assosiasjoner

- Noen ganger er det behov for assosiasjoner mellom mer enn to klasser/begreper
- Typisk dreier det seg da om begreper av typen Hvem, Hva, Hvor, Når
- Eksempel:
  - AS Gjenvinning gjenvinner papir i Oslo

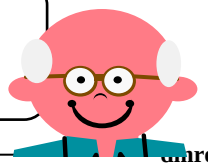
  kan vanligvis ikke brytes ned til

  - AS Gjenvinning gjenvinner papir
  - AS Gjenvinning gjenvinner i Oslo
  - papir gjenvinnes i Oslo

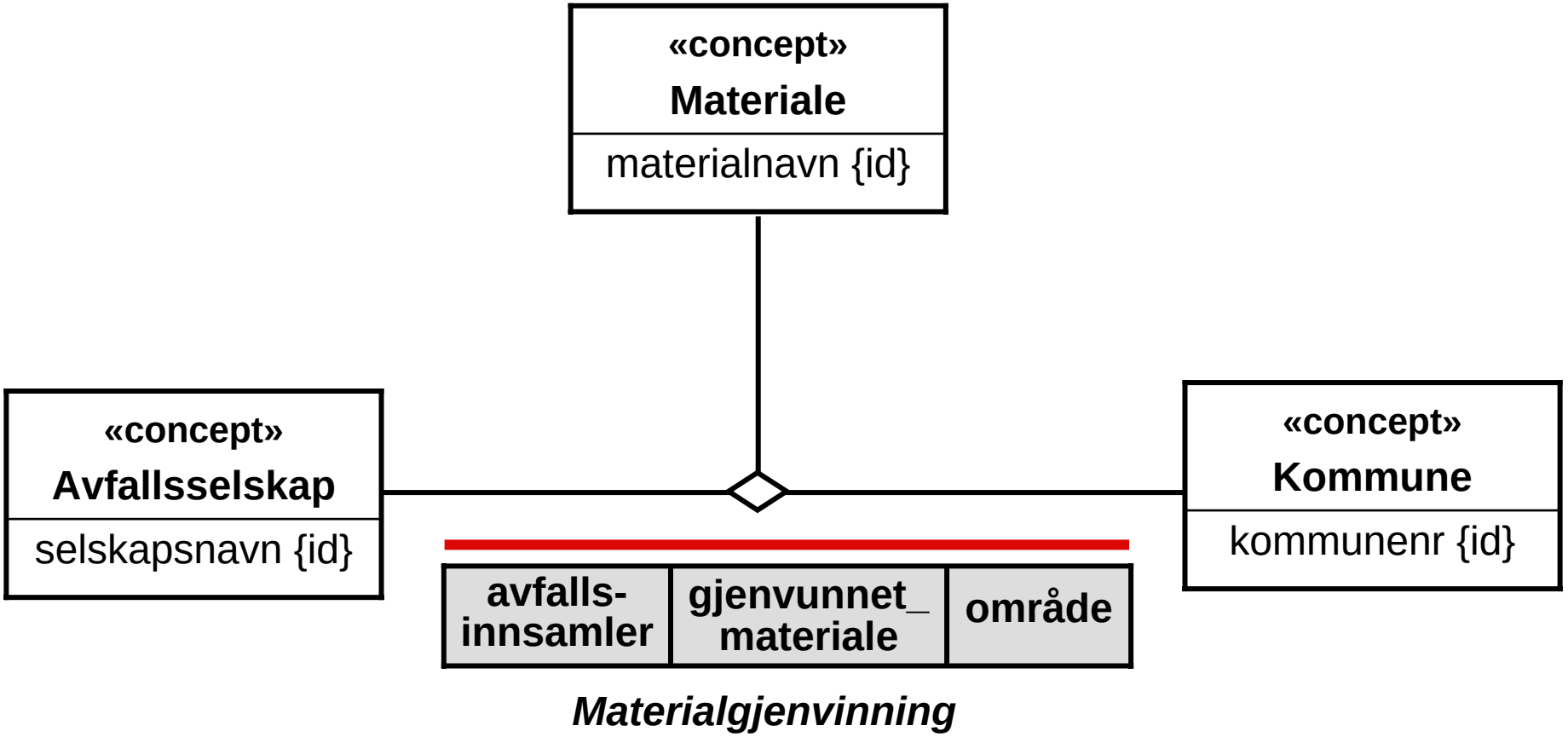
En ternær assosiasjon



Unngå UML-assosiasjoner mellom mer enn to klasser!

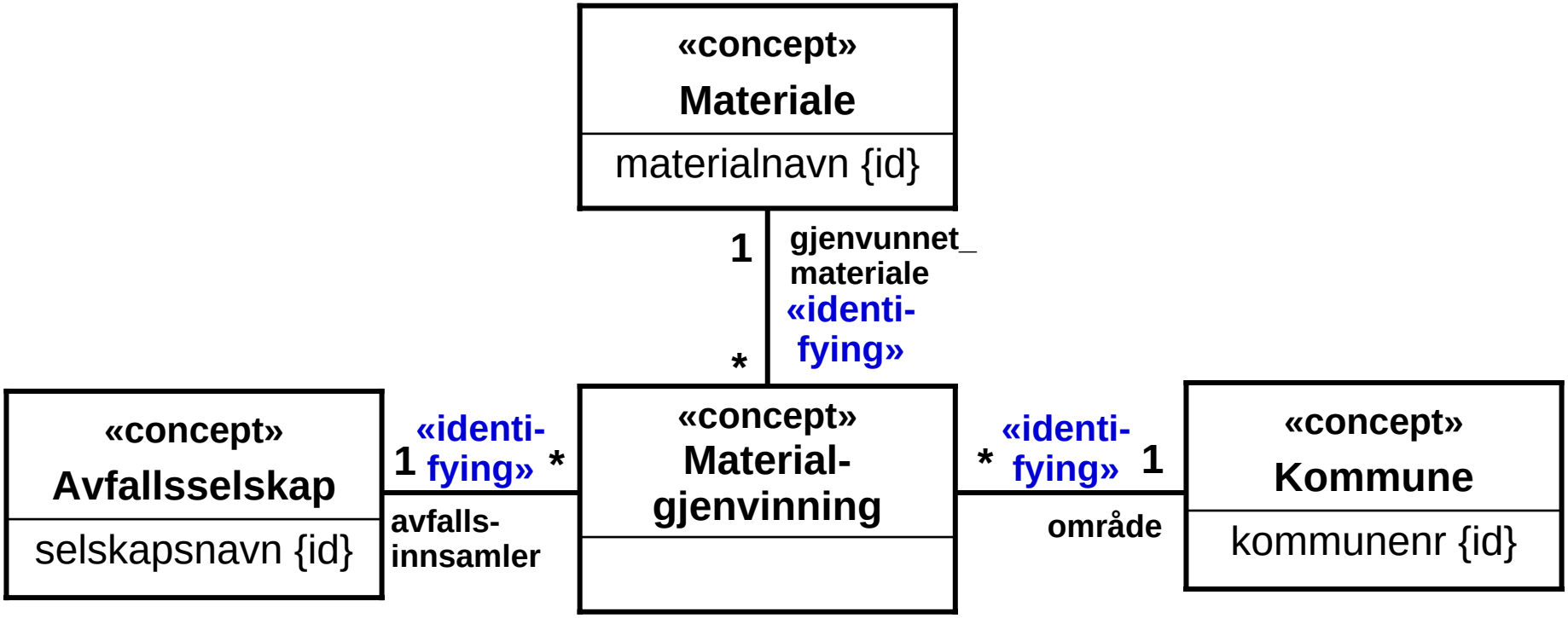


En ternær assosiasjon sett som tabell/relasjon



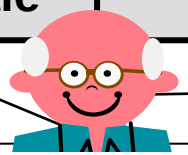
”Alle avfallsinnsamlere kan gjenvinne alle materialer i alle områder”

Fjern høyere ordens assosiasjoner vha. nye begreper

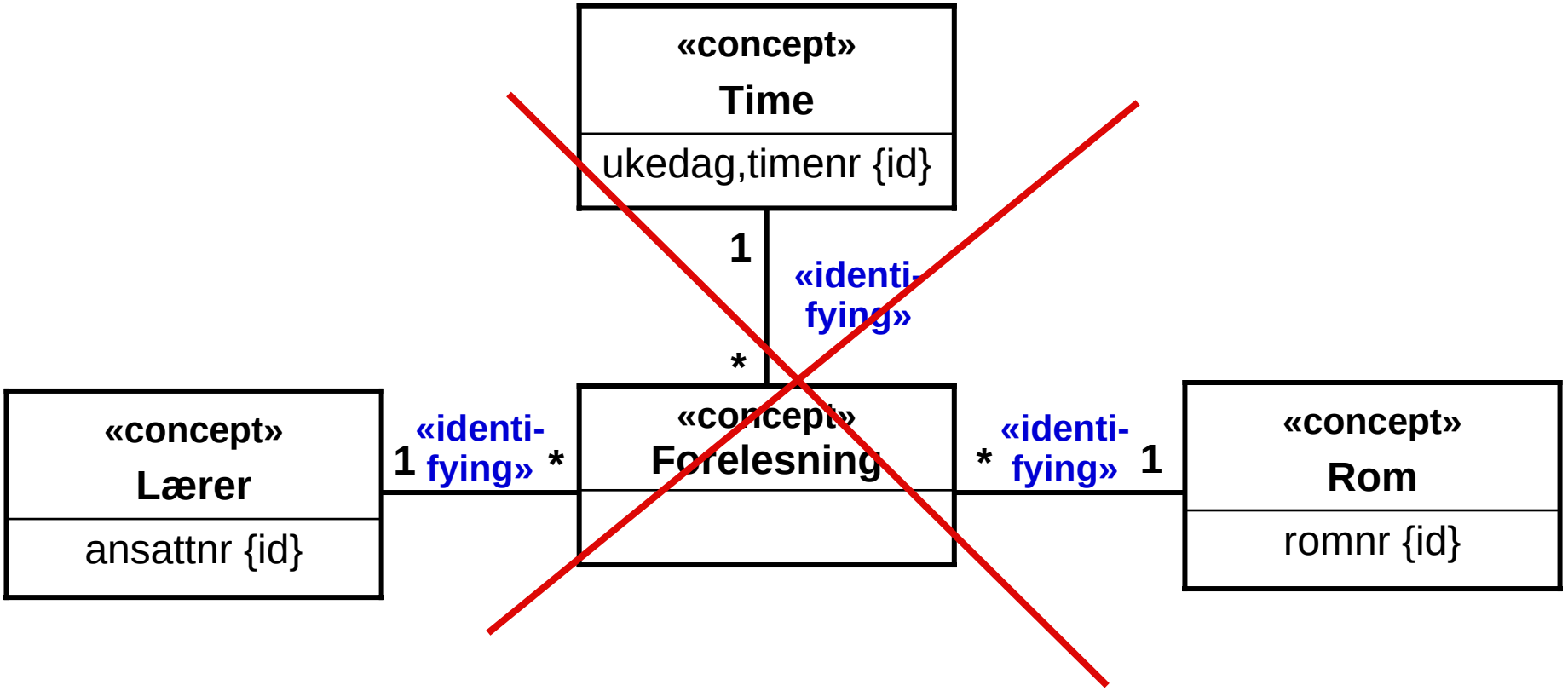


Lang entydighetsskranke konverteres til identifiserende assosiasjoner

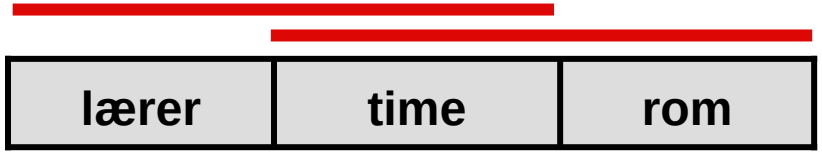
avfalls-innsamler	gjenvunnet_materiale	område



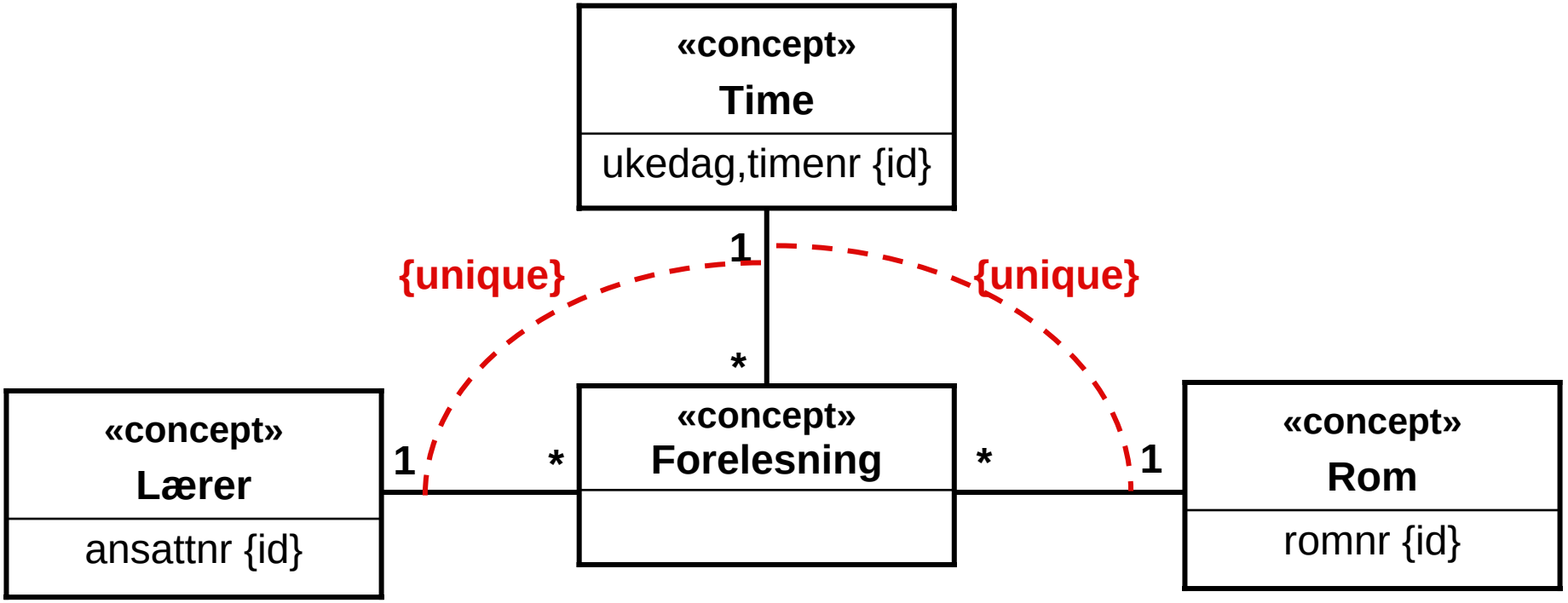
Overlappende entydighetsskranker



Hvordan viser vi dette i UML?



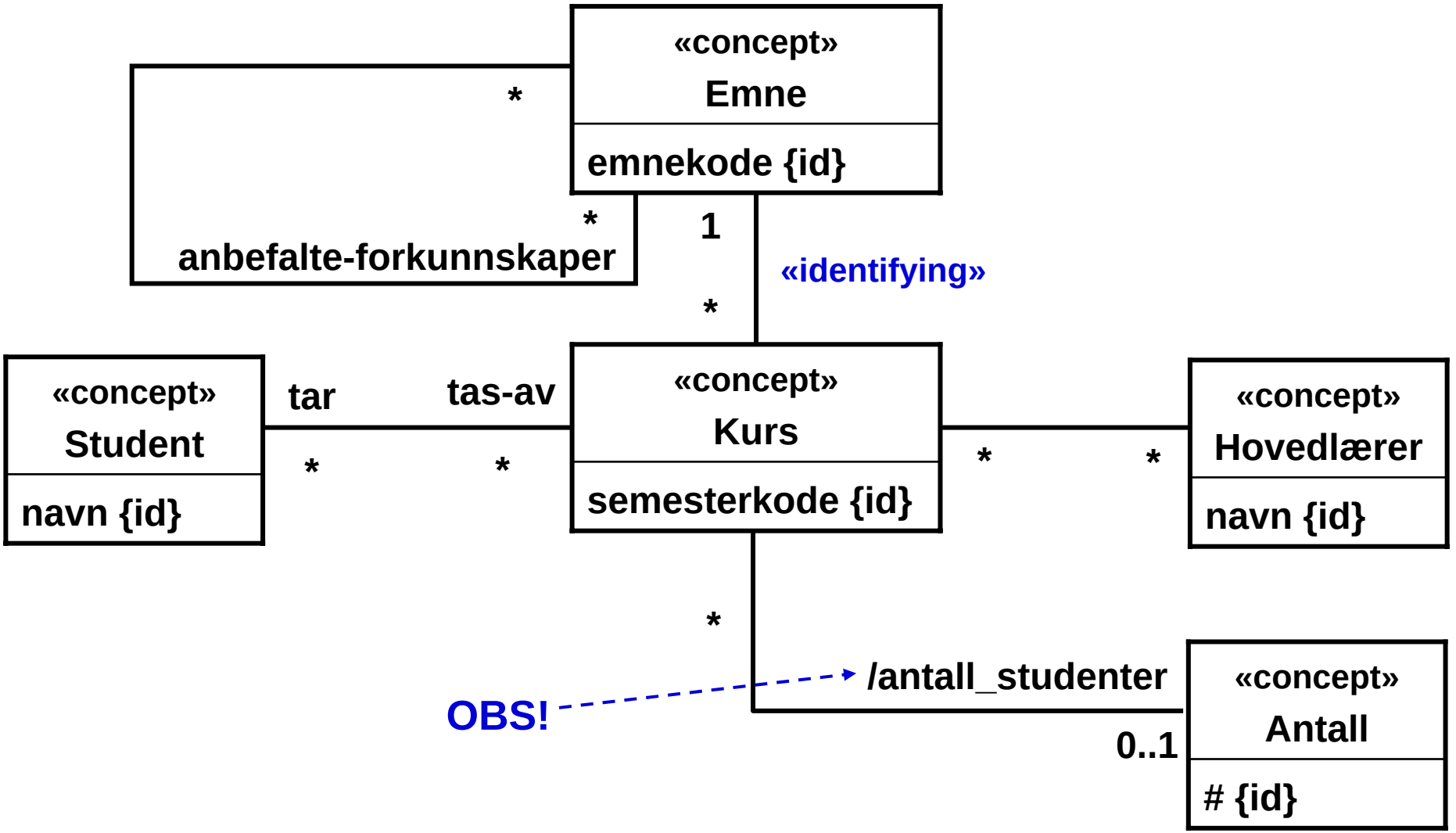
Den eksterne entydighetsskranken



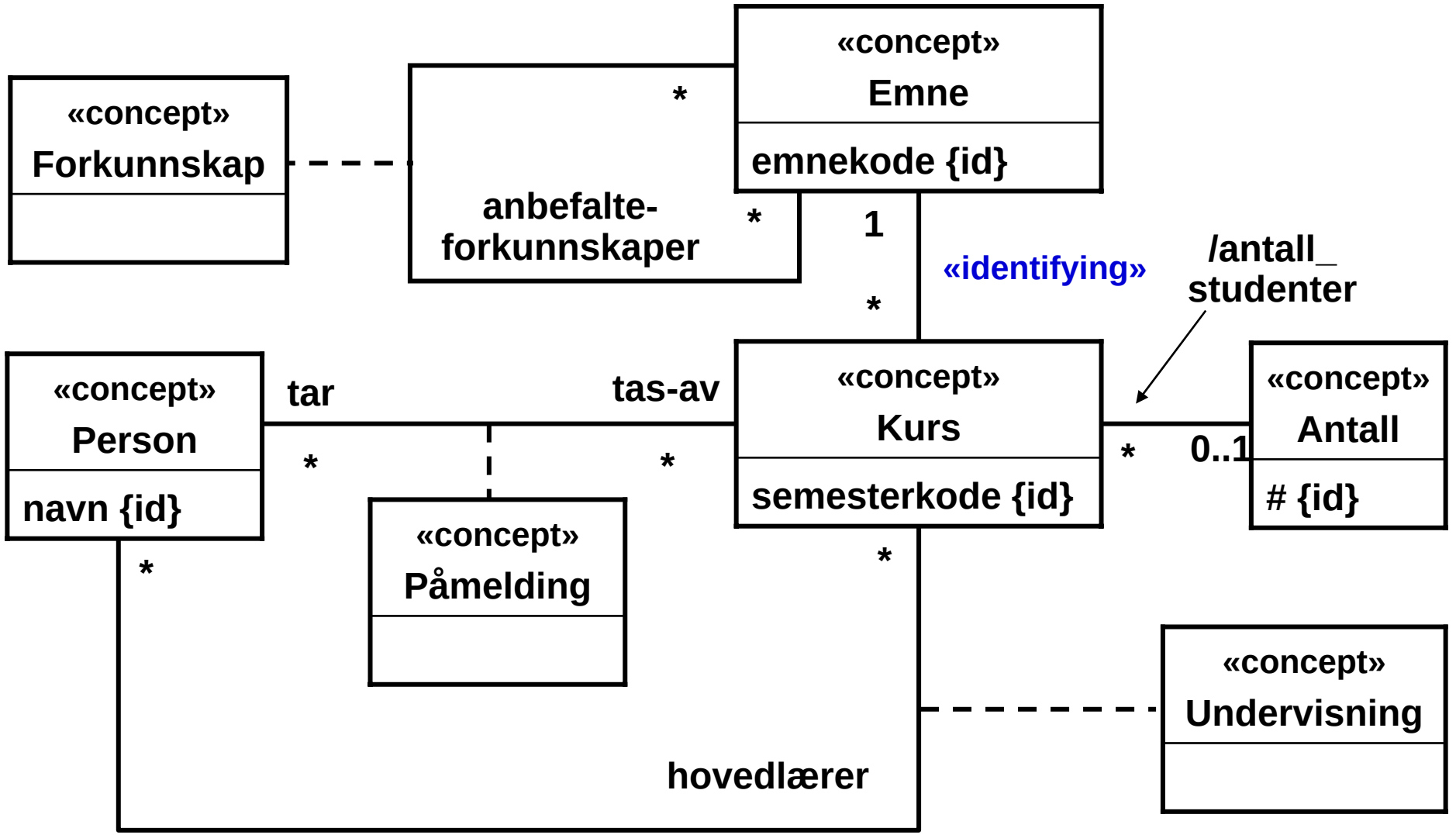
Warmer & Kleppe foreslår lignende grafiske notasjoner for lignende skranker

Warmer & Kleppe: The Object Constraint Language. Addison-Wesley 1999

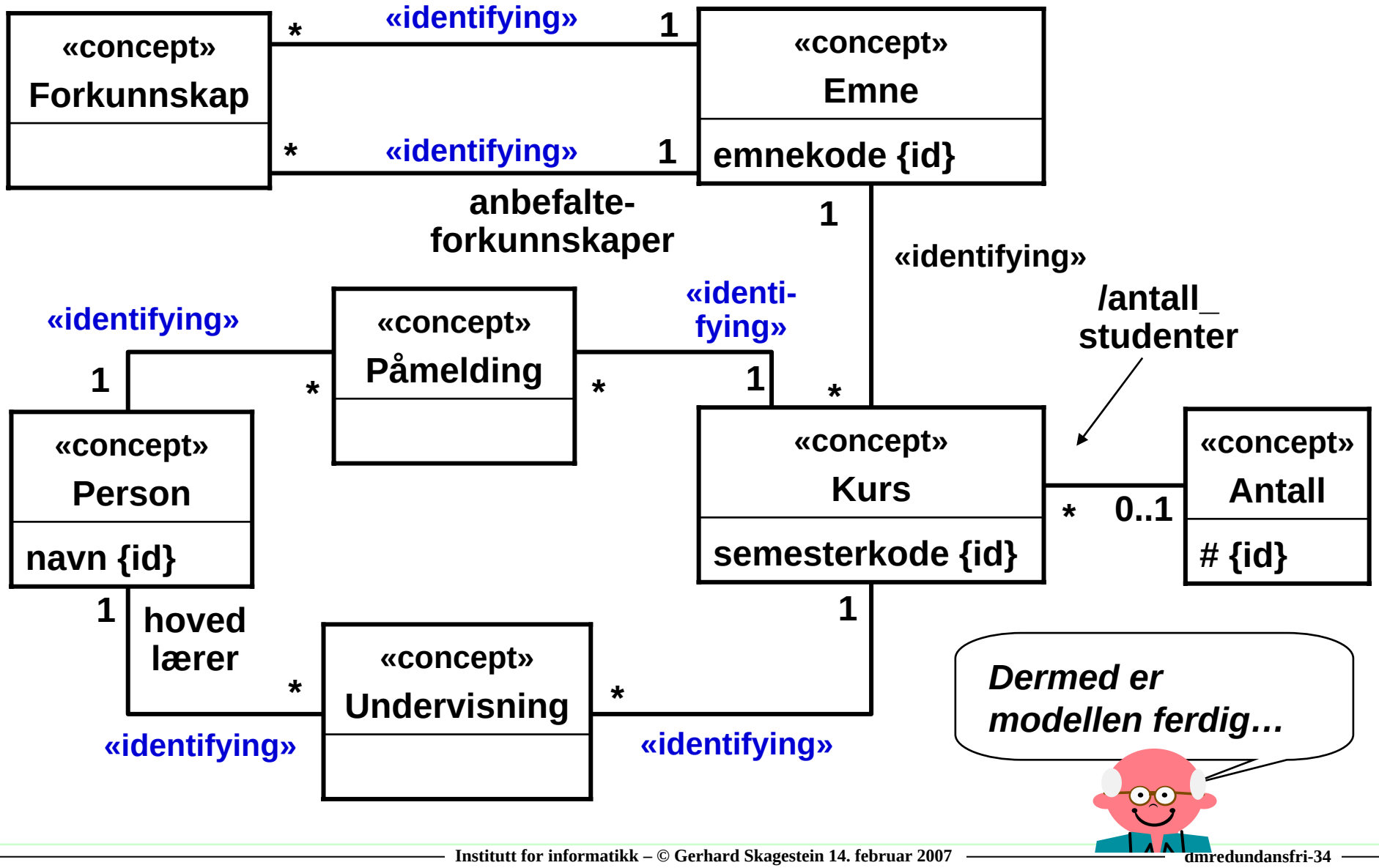
Kurssystemet (forts.)



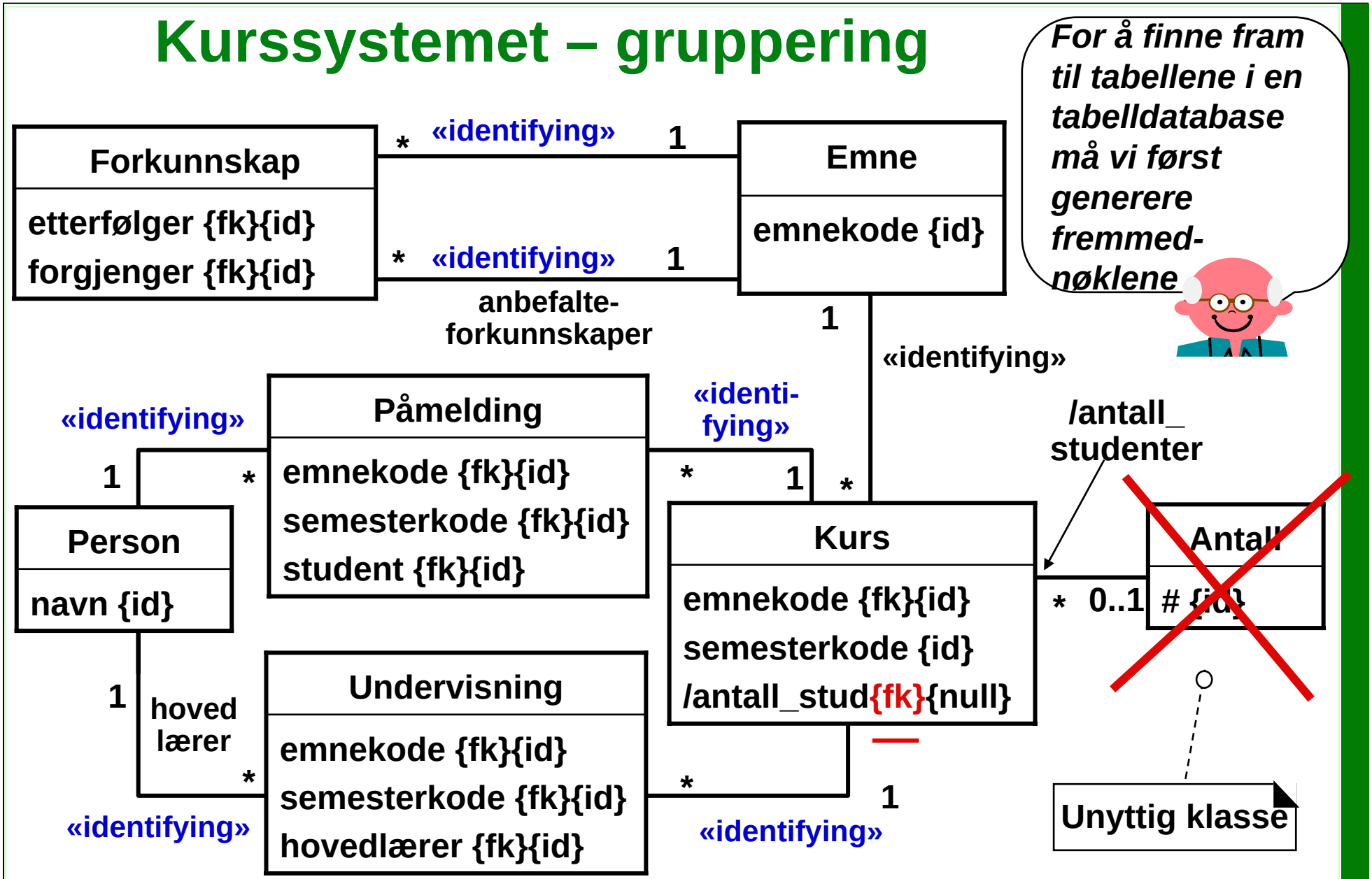
Kurssystemet – begrepsdannelse



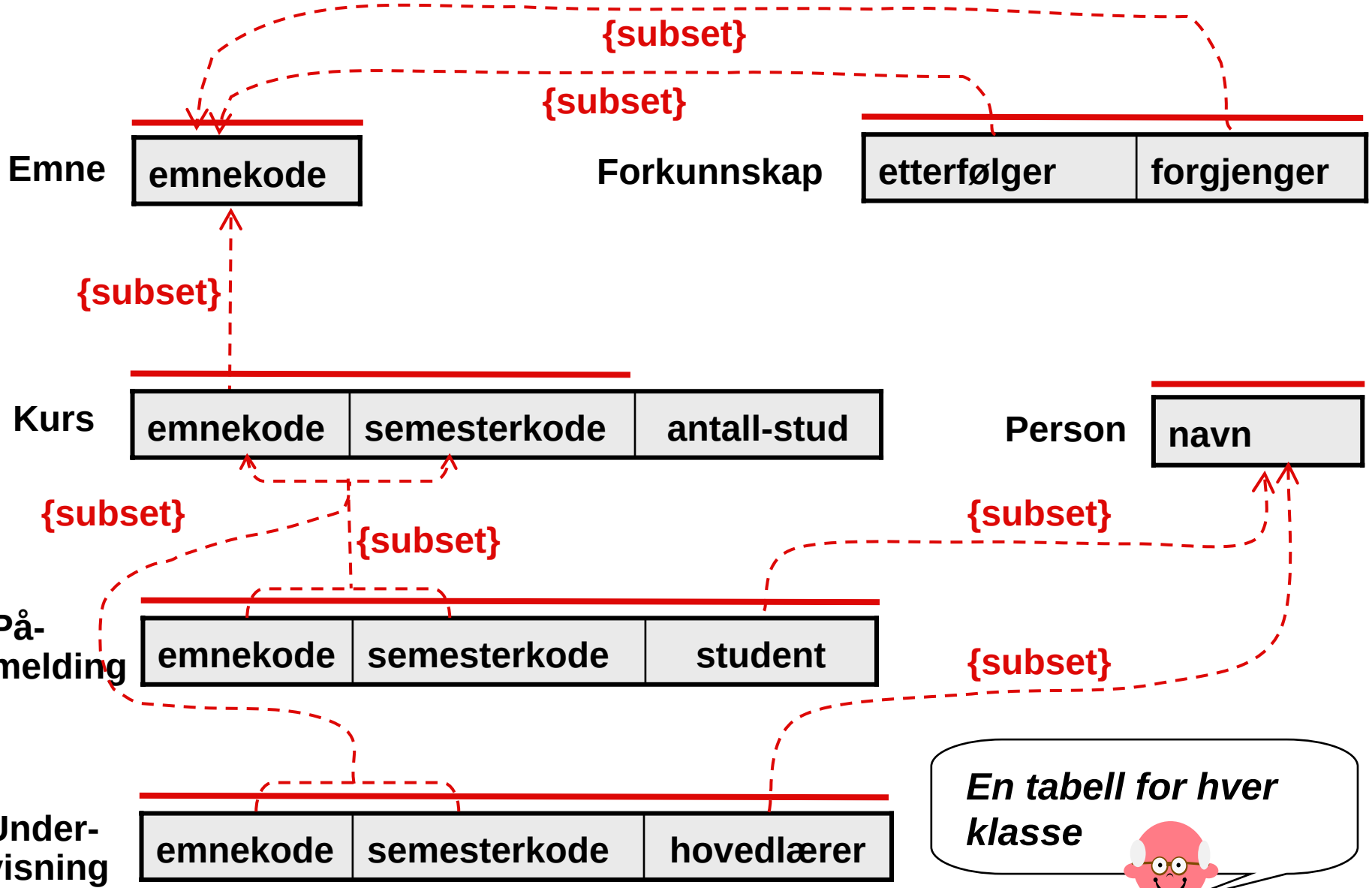
Kurssystemet – konvertere assosiasjonsklasser



Kurssystemet – gruppering



Kurssystemet – tabelldatabasen



Kurssystemet – tabelldatabasen

{subset}

{subset}

Emne

emnekode
INF1010
INF1040
INF1050

Forkunnskap

etterfølger	forgjenger
~~INF1050~~	~~INF1000~~
INF1050	INF1040
INF1010	INF1000

{subset}

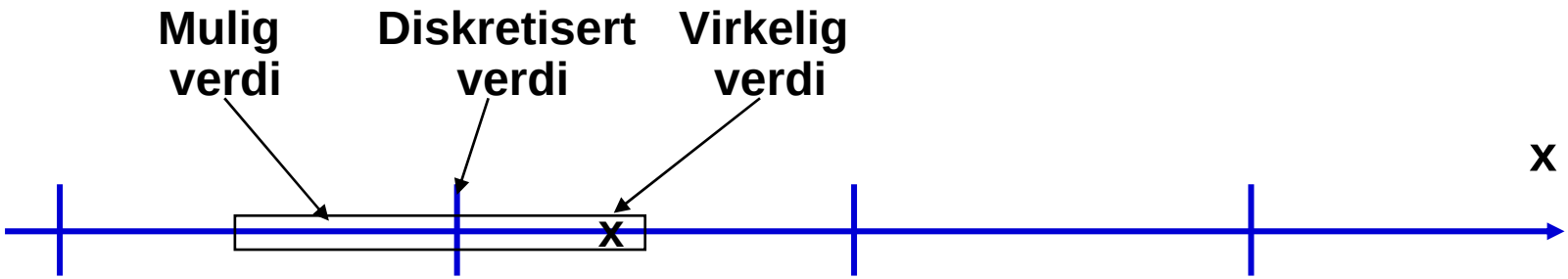
Kurs

emnekode	semesterkode	antall-stud
INF1010	vår2006	457
INF1050	vår2006	277

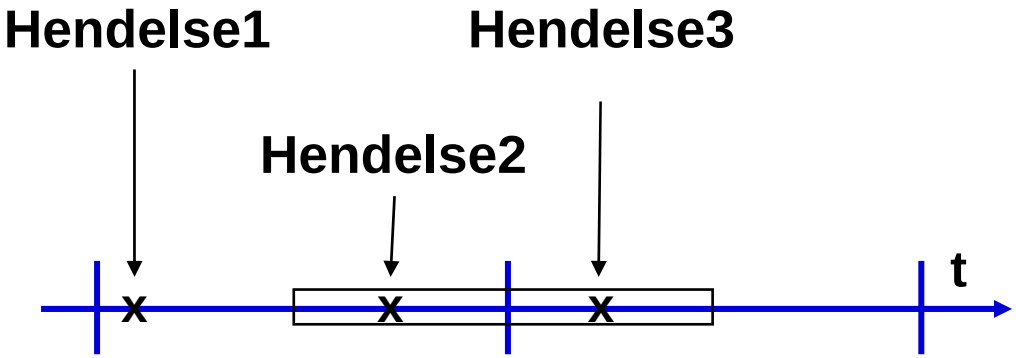
Eksempler på forekomster

Litt om tid

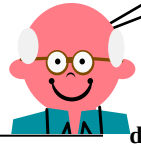
- **Tiden er kontinuerlig – derfor må den deles opp i små stykker – den må *diskretiseres*** (jf. læreboka figur 12-3)



Størrelsen på stykkene bestemmer *oppløsningsevnen*
(jf. læreboka figur 12-4)

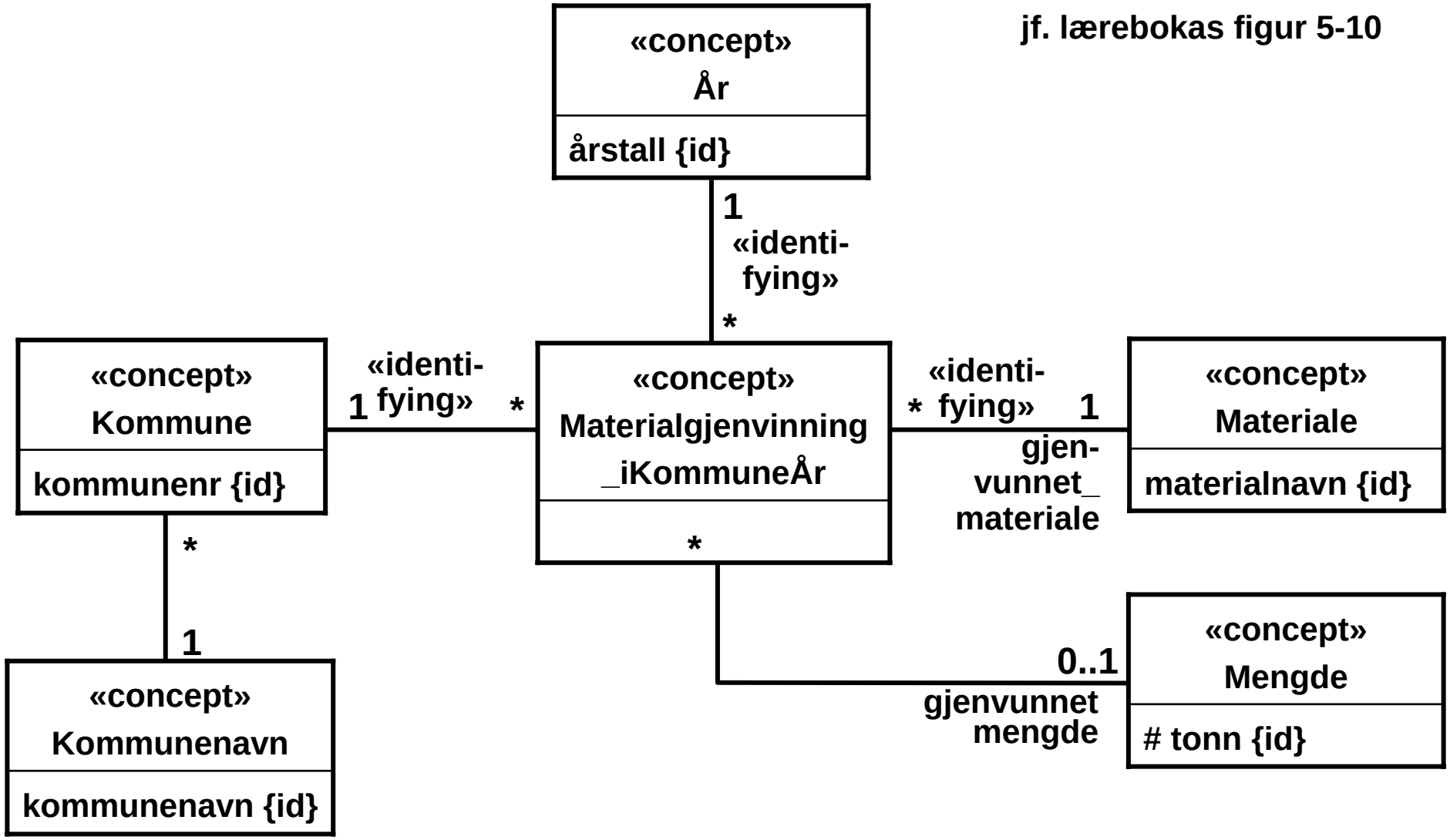


Som regel velger vi inndelinger som er kjent fra kalender og klokke

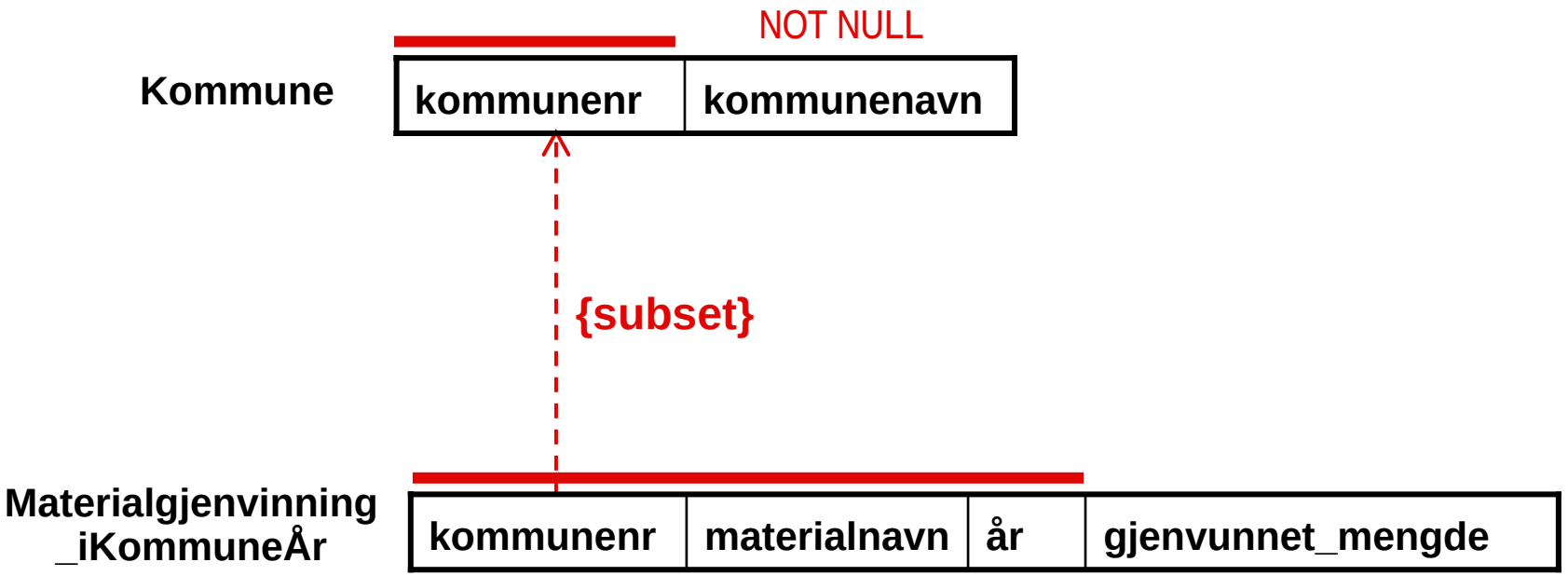


Datamodell med tidsdimensjon

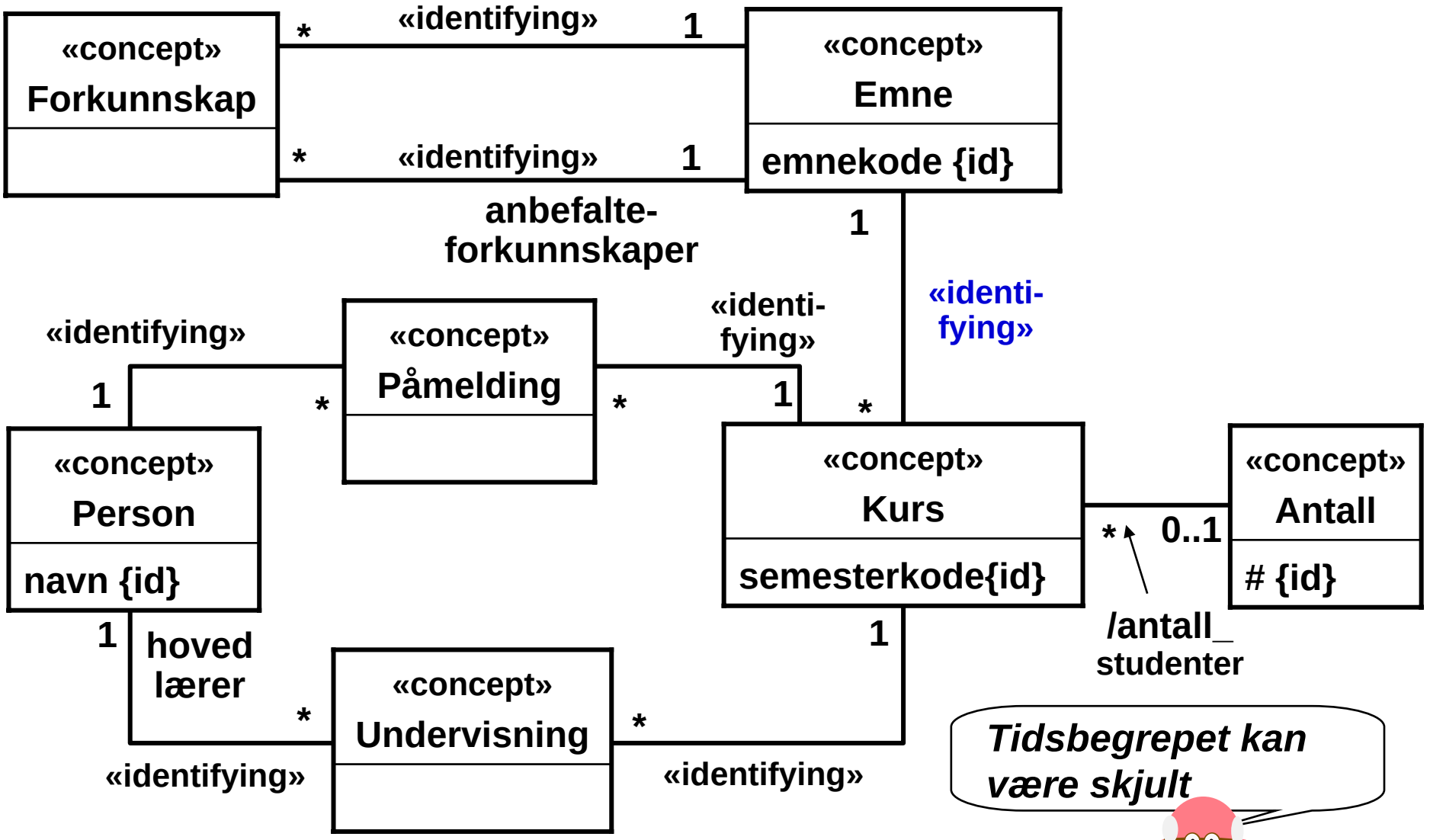
jf. lærebokas figur 5-10



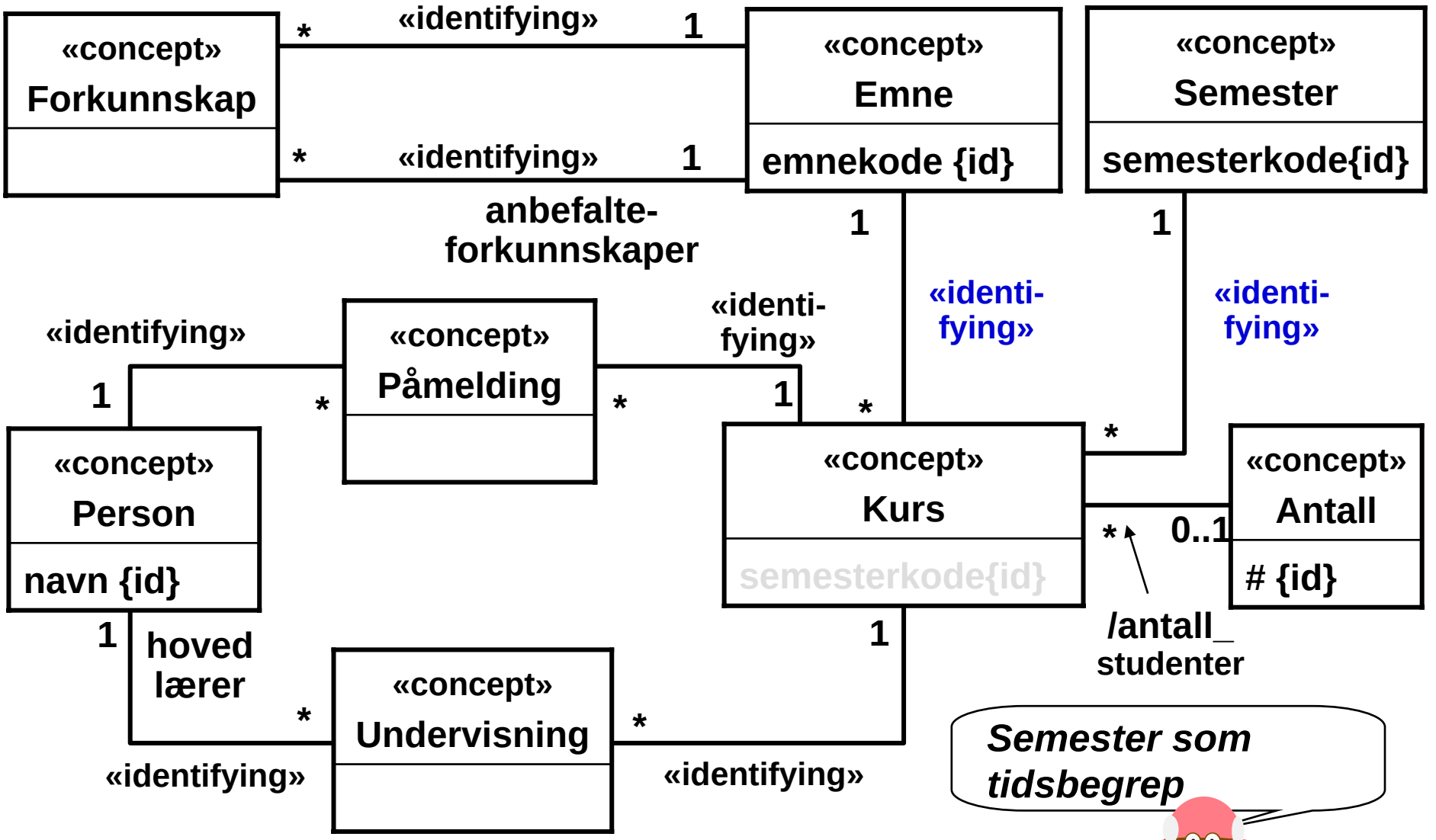
Relasjonsdatabase med tidsdimensjon



Kurssystemet (forts.)



Kurssystemet – med tidsbegrep



Underbegreper

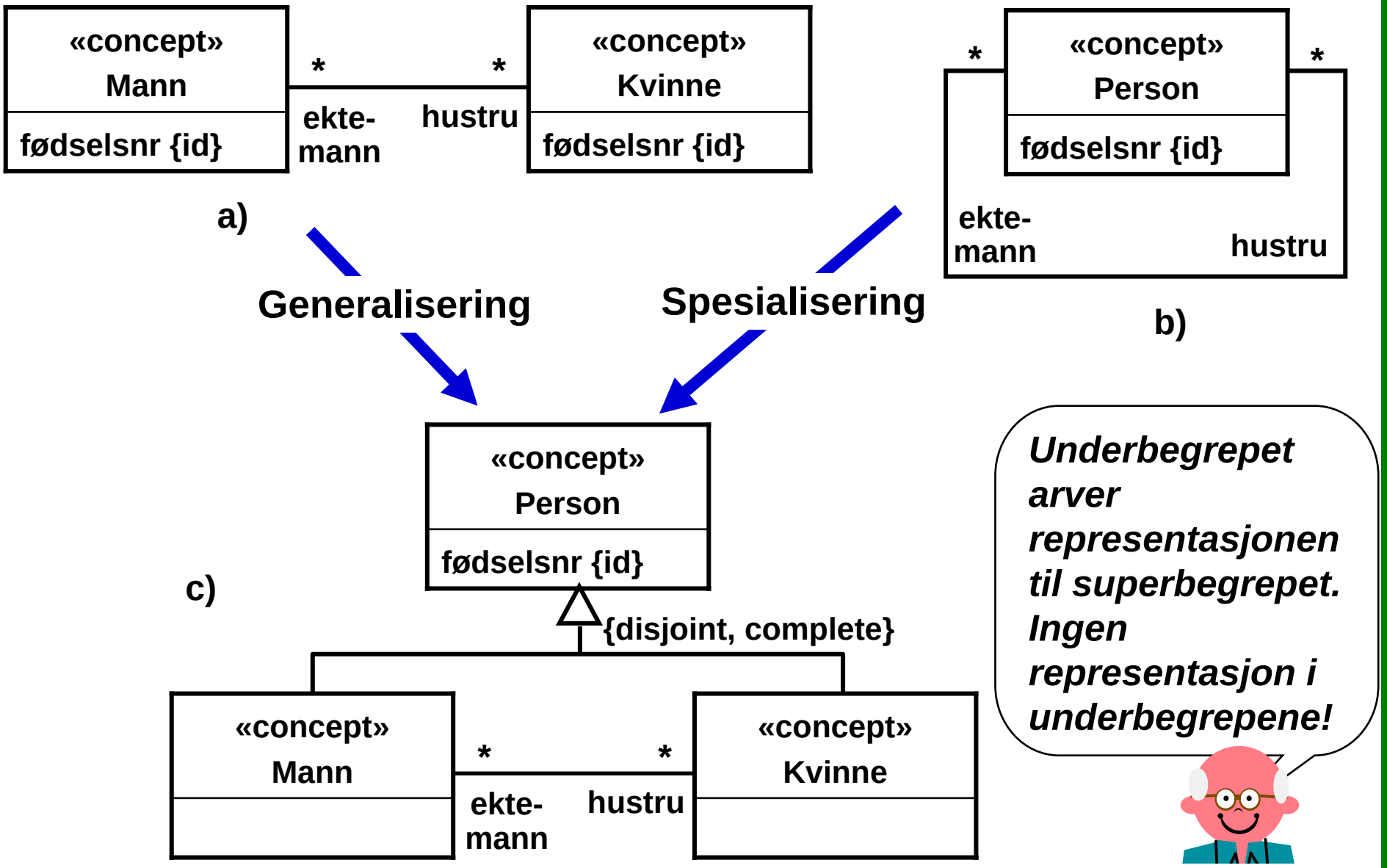
Jfr. “Fra kjernen og ut, fra skallet og inn”
avsnitt 5.3 og 5.4.4

Homogenitetsregelen

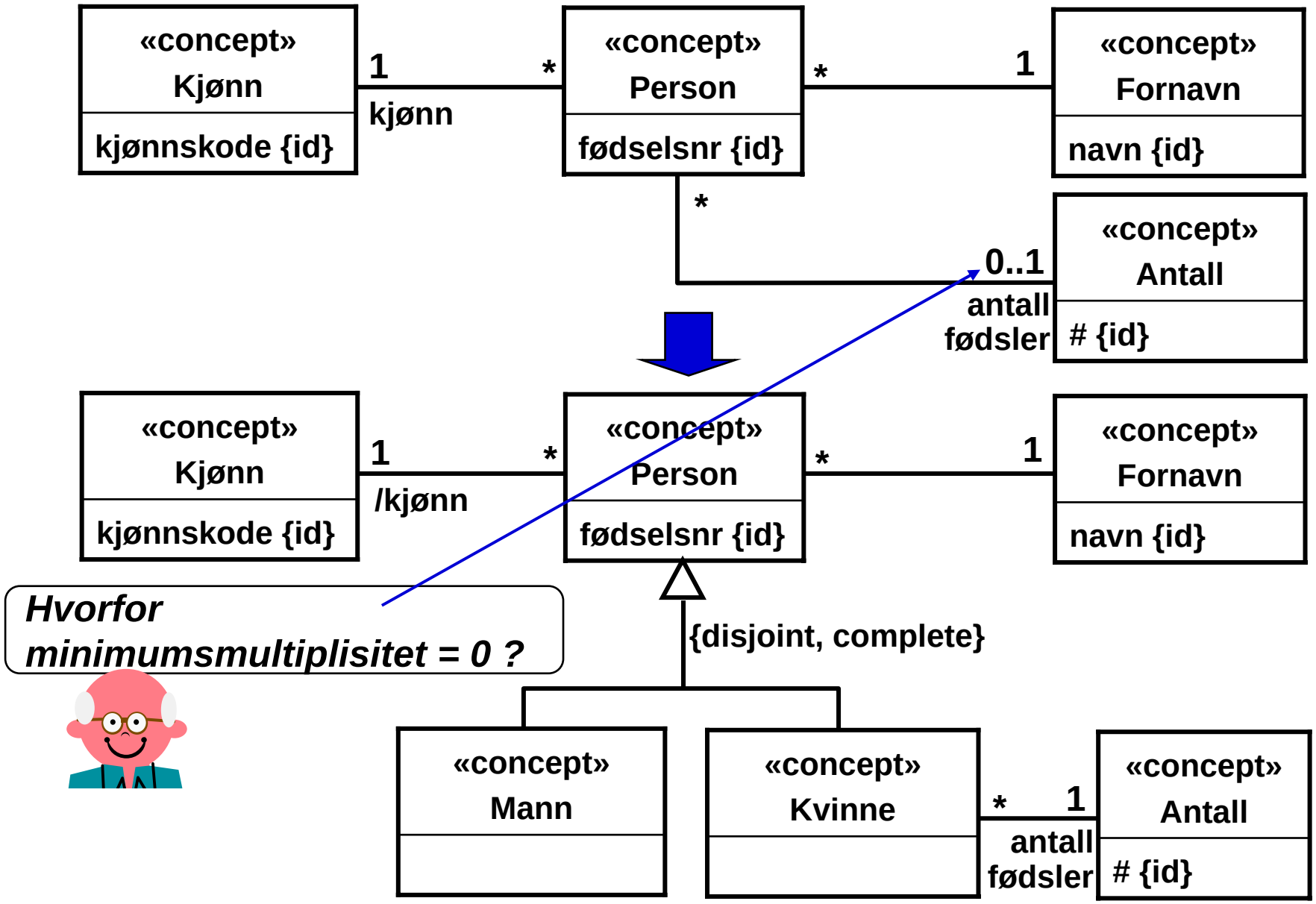
Alle tenkelige forekomster av et begrep
skal kunne spille alle roller
som er tilknyttet begrepet.

Vi krever ikke alltid at datamodellen
tilfredsstiller homogenitetsregelen,
men den blir mer presis hvis den gjør det.

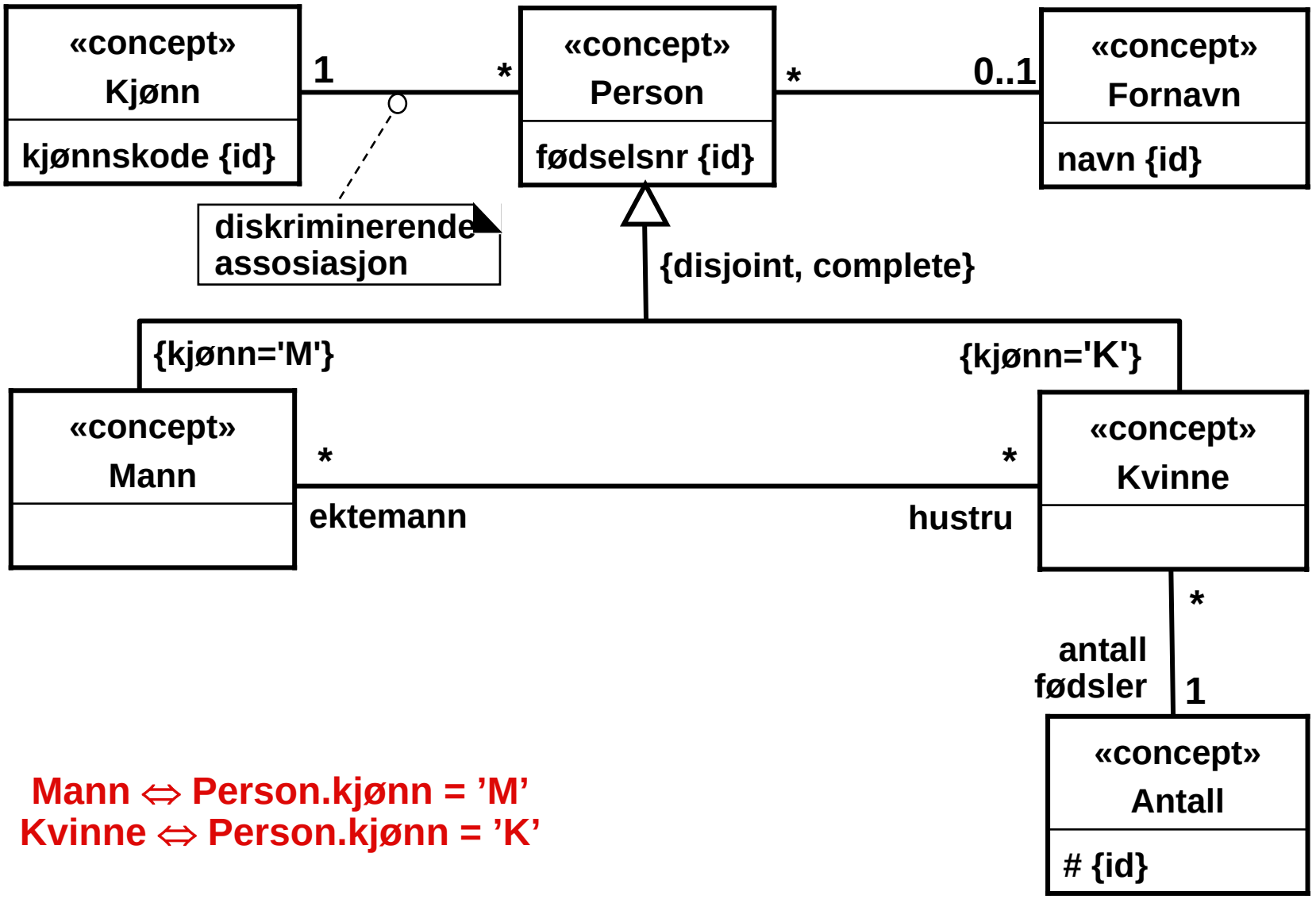
Figur 5-11. Spesialisering og generalisering – Underbegreper



Bruk av homogenitetsregelen

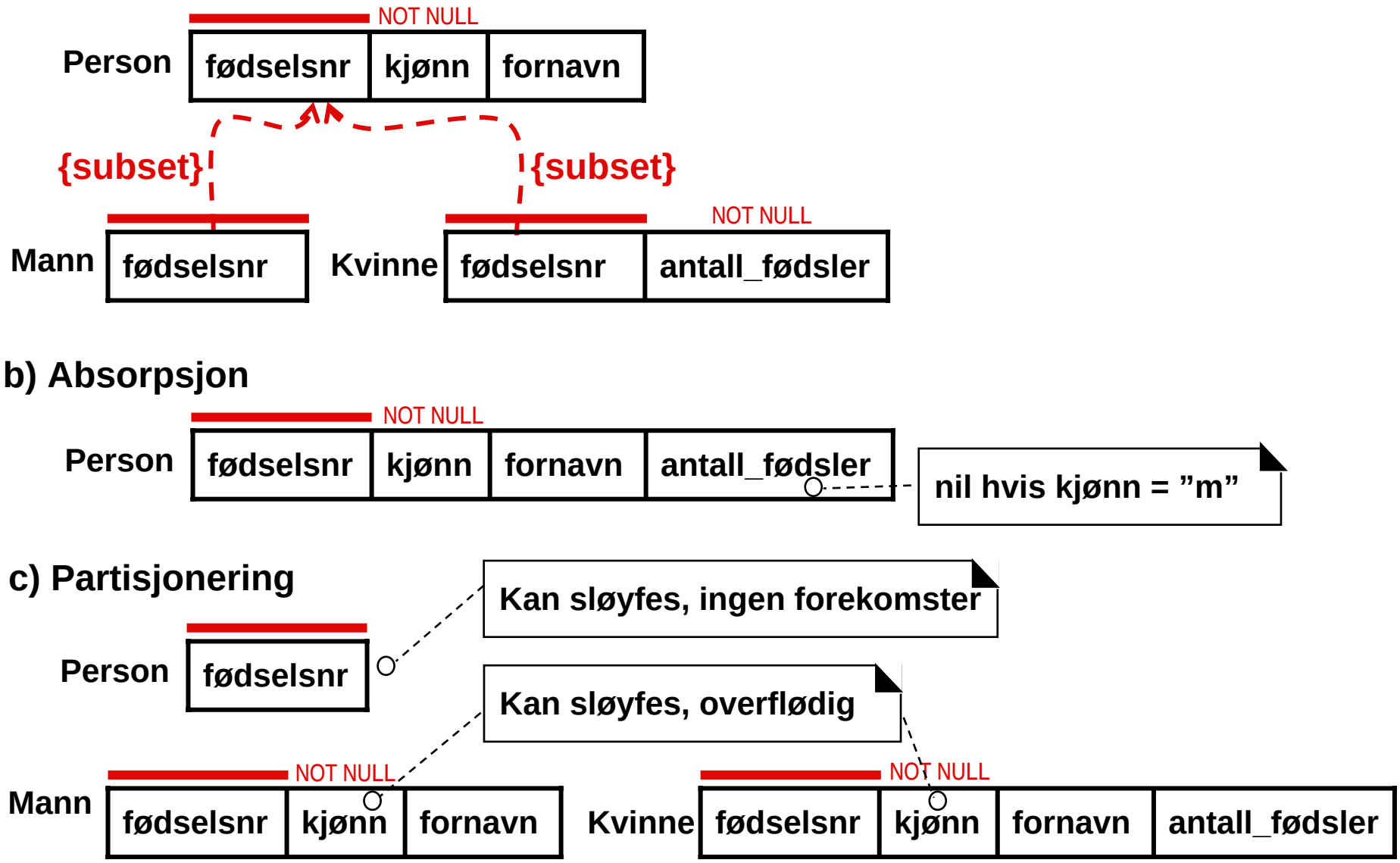


Figur 7-25. Underbegrep med diskriminerende assosiasjon

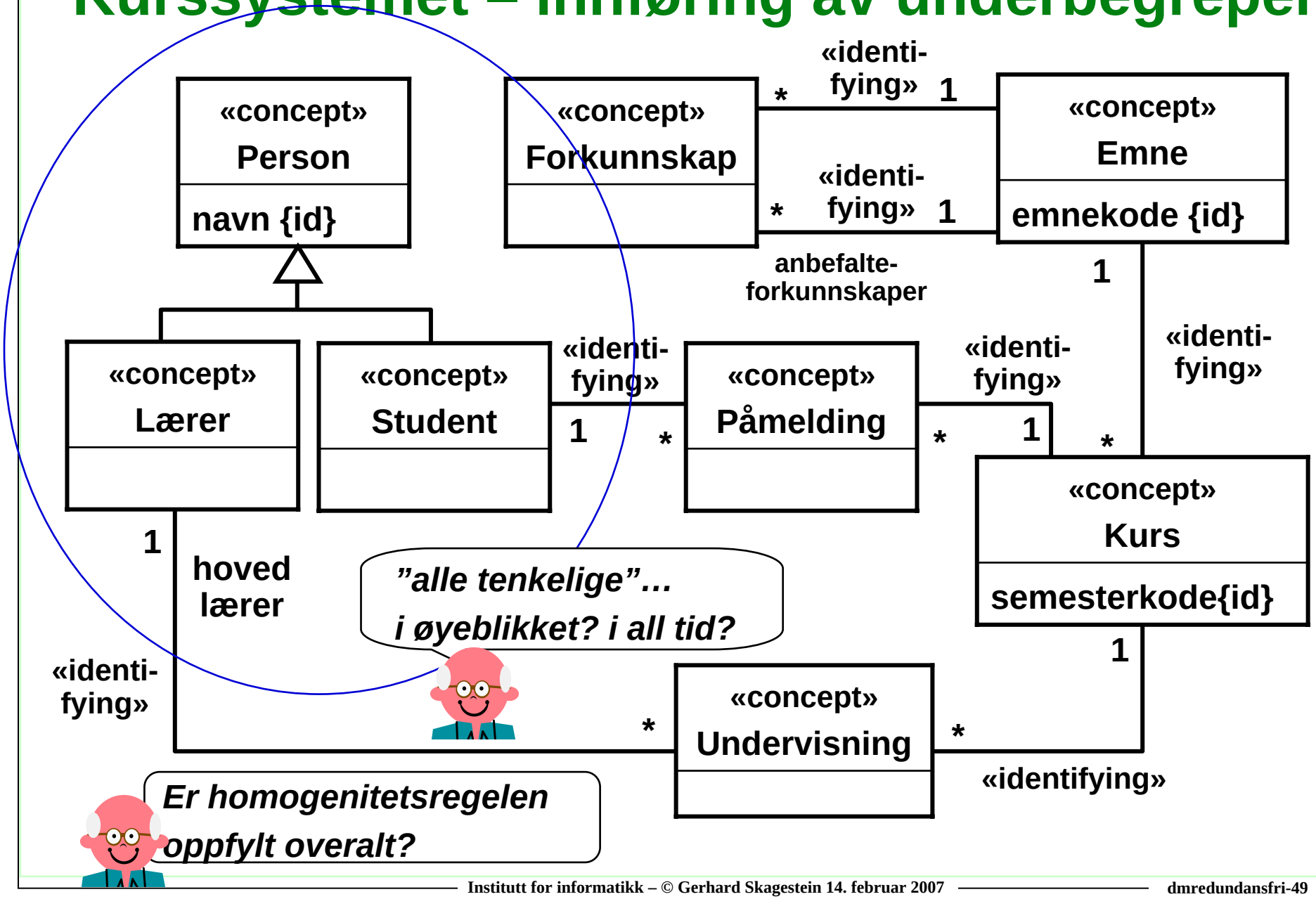


Mann ⇔ Person.kjønn = ’M’
Kvinne ⇔ Person.kjønn = ’K’

Figur 5-19. Håndtering av underbegreper



Kurssystemet – innføring av underbegreper



Oppsummering

- Vær bevisst på hvilke ”individer” vi skal vite noe om
- I en redundansfri datamodell skal det ikke finnes
  - dobbeltlagrede opplysninger
  - avledede opplysninger
- Redundansfrihet kan oppnås ved
  - gruppering av elementære utsagn
  - normalisering
  - intuisjon (”gå rett på”)
- Begrepsdannelse:
  Mange-til-mange-assosiasjoner kan oppfattes som et begrep
- Tid krever diskretisering
- Underbegreper kan avdekkes ved å bruke homogenitetsregelen