Logik

Dr. Johan Hagelbäck

johan.hagelback@lnu.se

http://aiguy.org

Vad är logik?

- Logik handlar om korrekta och inkorrekta sätt att resonera
- Logik är ett sätt att skilja mellan korrekt och inkorrekt tankesätt - *vetenskapen om korrekt tänkande*
- Genom historien har många matematiker och filosofer definierat mänskligt resonerande som ett logisk system
- Logik delas in i två kategorier: induktiv och deduktiv

Induktiv logik

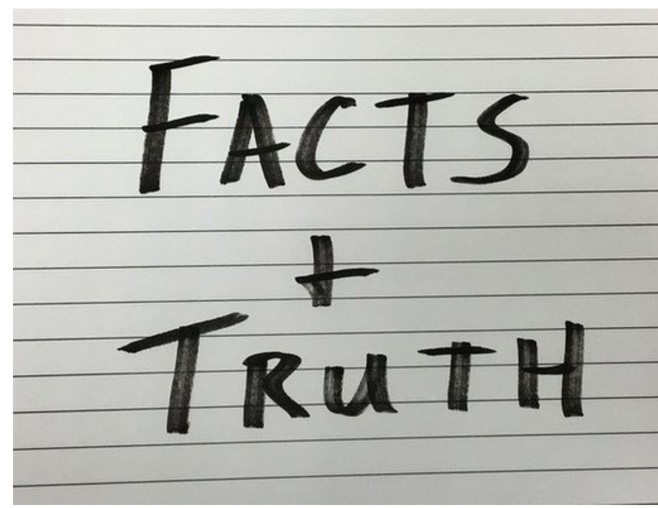
- Induktiv logik börjar med ett antal observationer eller erfarenheter
- Utifrån dessa kan slutsatser dras med viss sannolikhet

Exempel

- Du har endast ätit avokado två gånger i ditt liv
- Båda gångerna har du blivit sjuk strax efteråt
- Du drar då slutsatsen att du är allergisk mot avokado
- Det är en rimlig slutsats, men kan det inte finnas andra anledningar?
- Att du är allergisk är inte den enda slutsatsen och du kan därmed inte vara helt säker
- Slutsatser som dras i induktiva system är osäkra
- Graden av säkerhet är direkt relaterad till antalet observationer/erfarenheter

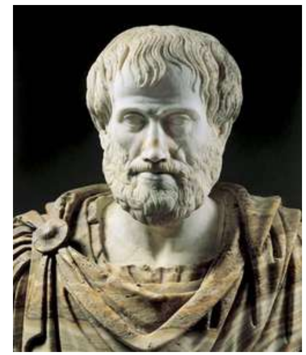
Deduktiv logik

- Deduktiv logik utgår ifrån att vissa saker garanterat är sanna
- Slutsatser som dras från dessa måste då också vara sanna



Exempel

- Alla män är dödliga
- Aristoteles är en man
- Alltså är Aristoteles dödlig



- Alla gudar är odödliga
- Tor är en gud
- Alltså är Tor odödlig

Deduktiv logik

- Aristoteles var en grekisk filosof som levde runt 350 f.Kr.
- Han beskrev ett antal logiska regler som måste följas för att dra korrekta slutsatser utifrån antaganden
- Aristoteles logiska system formulerar ett antal *premisser* utifrån *slutsatser* kan dras
- Alla premisser antas vara absolut sanna
- Premisser och slutsatser beskrivs med vanligt språk

Deduktiv - Induktiv

Deduktiv | **Induktiv**

Regler
Fakta

Observationer
Frekvens

Symbolisk logik

- Symbolisk logik är en modern utökning av Aristoteles logik där symboler används i stället för vanligt språk
- Logiska operatorer används för att uttrycka logiska tankegångar
- Det mest använda systemet är *Boolean Logic* som utvecklades av matematikern George Boole runt 1850
- Det används idag i alla moderna datorsystem

Boolean Logic

- Grunden i systemet är *propositioner*
- Propositioner är påståenden som kan vara antingen sanna eller falska
- En proposition har alltså ett värde som kan vara antingen sant eller falskt
- Exempel på propositioner är:
  - ”Mount Everest är världens högsta berg” – Sant
  - ”Ford är världens snabbaste bil” – Falskt

Boolean Logic

- Alla meningar är inte propositioner – de är inte sanna eller falska
- Exempel på icke-propositioner:
  - ”Hur högt är Mount Everest?”
  - ”Tvätta din Ford”
- Den första meningen är en fråga och den andra är en kommendering

Boolean Logic

- Det finns bara två möjliga värden på en proposition – sant eller falskt
- I Boolean Logic finns inga osäkerheter och sannolikheter
- Antingen är en proposition helt sann eller helt falsk
- Propositioner kan antingen vara *enkla* eller *sammansatta*
- En enkel proposition kan inte delas upp i mindre delar
- En sammansatt proposition består av flera enkla propositioner och logiska operatorer

Logiska operatorer

- Det finns fyra logiska operatorer:
  - Och (and)
  - Eller (or)
  - Medför (implies)
  - Inte (not)
- Exempel:
  - ”Jag är hungrig” – enkel
  - ”Jag är trött” – enkel
  - ”Jag är trött och jag är hungrig” – sammansatt
- Alla tre kan vara antingen sanna eller falska och är därmed propositioner

Symbolisk logik

- Att beskriva sammansatta propositioner i text blir snabbt oöverskådligt
- Därför utvecklades den symboliska logiken
- Alla enkla propositioner märks med en symbol (stor bokstav)
- Sammansatta propositioner beskrivs därmed med två eller fler symboler och logiska operatorer
- Exempel:
  – P = ”Jag är hungrig”
  – Q = ”Jag är trött”
  – P och Q

Logiska regler

- För att propositioner ska vara väl formulerade och förståeliga måste de följa några grundläggande regler

- Regel 1 – något av följande är en enkel proposition:
  - En enskild, stor bokstav
  - Sant
  - Falskt

Logiska regler

- Antag att ☐ är en proposition (enkel eller sammansatt)
- Regel 2 – om vi antar att ☐ är en proposition, är också någon av följande en proposition:
  - ☐ och ☐
  - ☐ eller ☐
  - ☐ medför ☐
  - ☐ ≡ ☐
  - inte ☐
  - (☐)

Logiska operatorer

- För att ytterligare förenkla sättet vi skriver på, och vara oberoende av språk, används symboler för alla logiska operatorer:

Svenska	Engelska	Symbol	Alternativ symbol
och	and	&	$\wedge$
eller	or	|	$\vee$
medför	implies	$\rightarrow$	$\Rightarrow$
om och endast om	if and only if	$\equiv$	$\Leftrightarrow$
inte	not	!	$\neg$

Logiska regler

- Regel 1 definierar vad som är en proposition
- Regel 2 definierar hur vi konstruerar sammansatta propositioner utifrån enkla propositioner

- Hur kan vi utifrån dessa regler kontrollera om en sammansatt proposition är välformulerad?

Logiska bevis

- Påstående: P och inte (Q eller R)
  1. P, Q och R är enkla propositioner enligt Regel 1:
     ☐ och inte (☐ eller ☐)
  2. ☐ eller ☐ är en proposition enligt Regel 2B:
     ☐ och inte (☐)
  3. (☐) är en proposition enligt Regel 2F:
     ☐ och inte ☐
  4. inte ☐ är en proposition enligt Regel 2E:
     ☐ och ☐
  5. ☐ och ☐ är en proposition enligt Regel 2A:
     ☐
- Eftersom beviset slutar med ☐ är påståendet en proposition

Logiska bevis

- Antag i stället att vi har följande påstående: P inte Q
  1. P och Q är enkla propositioner enligt Regel 1:
     □ inte □
  2. inte □ är en proposition enligt Regel 2E:
     □ □
- Nu kan vi inte göra mer för det finns ingen regel för fallet
  □ □
- Eftersom beviset inte slutar på □ är påståendet inte en välformulerad proposition

Utvärdera påståenden

- Oftast är vi mer intresserad av värdet på en sammansatt proposition än om den är välformulerad
- Värdet på en sammansatt proposition beror på värdet på de enkla propositionerna som är inblandade, och vilka logiska operatorer som används
- Först måste vi se veta hur värdet förändras för alla logiska operatorer beroende på inparametrar
- Detta kallas *sanningstabeller*

Och (konjunktion)

- Och blir endast sann om båda inparametrarna är sanna:

P	Q	P och Q
Falskt	Falskt	Falskt
Falskt	Sant	Falskt
Sant	Falskt	Falskt
Sant	Sant	Sant

Eller (disjunktion)

- Eller blir sann om någon av inparametrarna är sanna:

P	Q	P eller Q
Falskt	Falskt	Falskt
Falskt	Sant	Sant
Sant	Falskt	Sant
Sant	Sant	Sant

Inte (negering)

- Inte tar bara en inparameter, och svaret är motsatsen:

P	inte P
Falskt	Sant
Sant	Falskt

Medför (implikation)

- Implikation kräver lite mer tankeverksamhet för att förstå
- Den innebär att om något är sant, måste också något annat logiskt sett vara sant
- Exempel:
  - Om ”mitt batteri är slut” är sant, medför det att ”min telefon startar inte” också är sant
- Implikation är villkorssatser:
  - Om A så B

Medför (implikation)

- Sanningstabellen för implikation visas nedan
- Den är inte helt lätt att förstå, så vi ska gå igenom fall för fall

P	Q	P medför Q
Falskt	Falskt	Sant
Falskt	Sant	Sant
Sant	Falskt	Falskt
Sant	Sant	Sant

P = Sant, Q = Sant

- Betrakta vårt exempel:
  - P = ”mitt batteri är slut” – Sant
  - Q = ”min telefon startar inte” – Sant
- Om telefonen inte startar när batteriet är slut stämmer
- Sant medför Sant är alltså Sant

P = Sant, Q = Falskt

- Betrakta vårt exempel:
  - P = ”mitt batteri är slut” – Sant
  - Q = ”min telefon startar inte” – Falskt
- Att telefonen startar fastän batteriet är slut kan inte stämma
- Sant medför Falskt är alltså Falskt

P = Falskt, Q = Sant

- Betrakta vårt exempel:
  - P = ”mitt batteri är slut” – Falskt
  - Q = ”min telefon startar inte” – Sant
- Batteriet är inte slut men telefonen startar ändå inte
- Detta är logiskt, då det kan finnas andra anledningar till att telefonen inte startar
- Falskt medför Sant är alltså Sant

P = Falskt, Q = Falskt

- Betrakta vårt exempel:
  - P = ”mitt batteri är slut” – Falskt
  - Q = ”min telefon startar inte” – Falskt
- Batteriet är inte slut och telefonen startar vilket stämmer
- Falskt medför Falskt är alltså Sant

Om och endast om (ekvivalens)

- Propositionerna P och Q är ekvivalenta (lika) om de har samma sanningsvärde:

P	Q	P ≡ Q
Falskt	Falskt	Sant
Falskt	Sant	Falskt
Sant	Falskt	Falskt
Sant	Sant	Sant

Sammansatta propositioner

- För att kunna utvärdera sammansatta propositioner behöver vi undersöka sanningsvärdet för varje enskild operator enligt dess sanningstabell
- Antag att vi har ett påstående: P och (Q eller R)
- Vi måste då börja med att utvärdera sanningsvärdet på uttrycket inom parenteser
- Efter det utvärderar vi sanningsvärdet på Och-uttrycket
- Enklast är att skapa en sanningstabell för hela uttrycket och fylla i del för del

Sammansatta propositioner

P	Q	R	(Q eller R)	P och (Q eller R)
F	F	F		
F	F	S		
F	S	F		
F	S	S		
S	F	F		
S	F	S		
S	S	F		
S	S	S		

Skapa en tom sanningstabell

Sammansatta propositioner

P	Q	R	(Q eller R)	P och (Q eller R)
F	F	F	F	
F	F	S	S	
F	S	F	S	
F	S	S	S	
S	F	F	F	
S	F	S	S	
S	S	F	S	
S	S	S	S	

Utvärdera uttrycket (Q eller R)

Sammansatta propositioner

P	Q	R	(Q eller R)	P och (Q eller R)
F	F	F	F	F
F	F	S	S	F
F	S	F	S	F
F	S	S	S	F
S	F	F	F	F
S	F	S	S	S
S	S	F	S	S
S	S	S	S	S

Utvärdera resten av uttrycket

Tautologi

- En proposition som är sann oavsett inparametrar kallas en *tautologi*
- Exempel:
  – P eller inte P
- En tautologi är en omskrivning av samma sak, alltså kaka på kaka
- Tautologier används ibland i politiska debatter:
  – ”vi ska värna de offentliga finanserna, vi ska skydda kärnan i den offentliga välfärden, vi ska säkra välfärden …”
  – Sagt av vår förra finansminister Anders Borg

Kontradiktion

- En proposition som är falsk oavsett inparametrar kallas en *kontradiktion*
- Exempel:
  - P och inte P
- Kontradiktion uppstår när vi säger saker som är varandras motsatser:
  - ”Några bilar är vita”
  - ”Ingen bil är vit”

Tillämpningar

Sökning

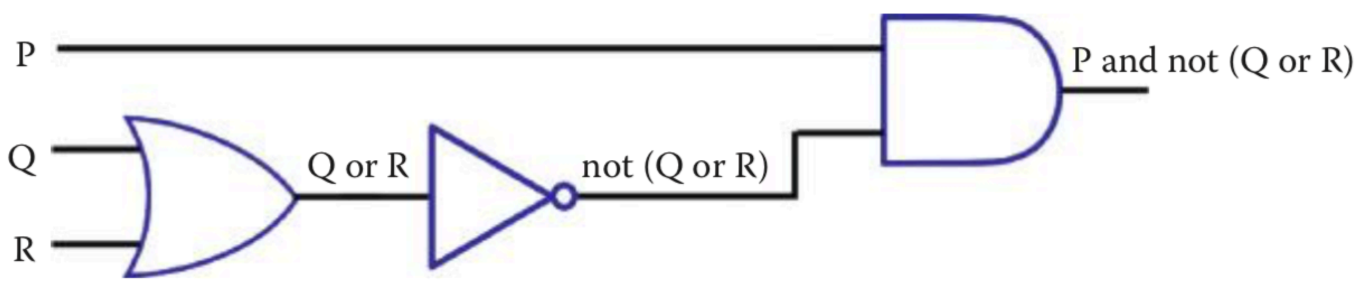
- Antag att du söker på t.ex. ”boolean logic”
- De flesta sökmotorer tolkar detta som att du söker på ”boolean *eller* logic”
- Du får då resultat från sidor som bara handlar om boolean eller bara om logic, och så klart sidor som handlar om boolean logic
- För att kunna precisera din sökning stödjer de flesta sökmotorer logiska operator
- Sökning på ”boolean *och* logic” ger sannolikt mer precisa resultat

Sökning

- Testar vi detta i Google får vi följande resultat:
  - ”boolean logic”: 5,65 miljoner sidor
  - ”boolean OR logic”: 96,5 miljoner sidor
  - ”boolean AND logic”: 6,26 miljoner sidor
- Google verkar hantera fallet utan logisk operator på något särskilt sätt som inte tydligt framgår
- Samma sökningar i Bing:
  - ”boolean logic”: 3,48 miljoner sidor
  - ”boolean OR logic”: 19,1 miljoner sidor
  - ”boolean AND logic”: 3,47 miljoner sidor
- Bing verkar hantera sökning utan logisk operator som *och*

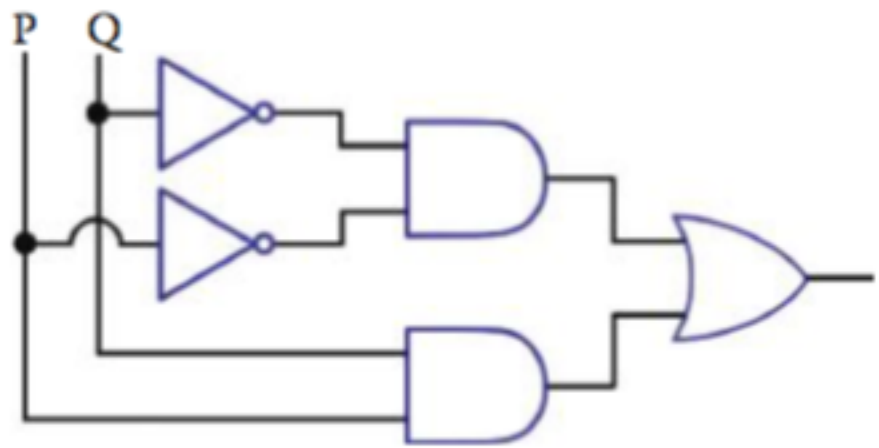
Digitala kretsar

- Digitala kretsar gör det möjligt för en dator att utföra aritmetiska operationer
- De konstrueras genom att kombinera olika logiska kretsar



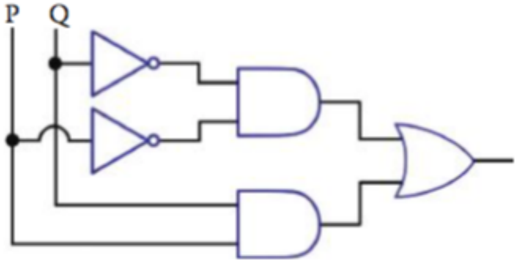
Digitala kretsar

- Ekvivalens krets:



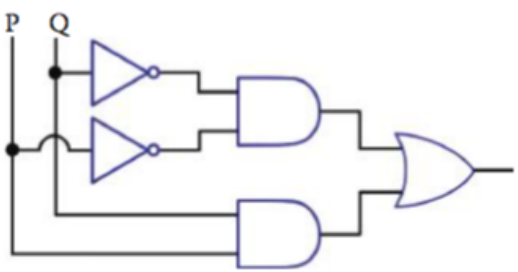
Ekvivalens krets

P	Q	!P	!Q	A = (!P och !Q)	B = (P och Q)	A eller B
F	F					
F	S					
S	F					
S	S					



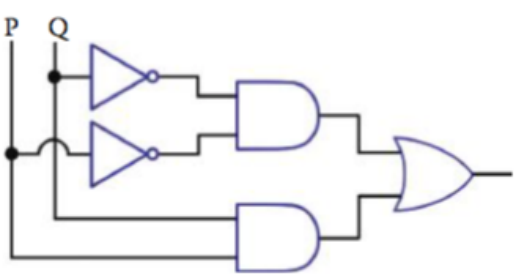
Ekvivalens krets

P	Q	!P	!Q	A = (!P och !Q)	B = (P och Q)	A eller B
F	F	S				
F	S	S				
S	F	F				
S	S	F				



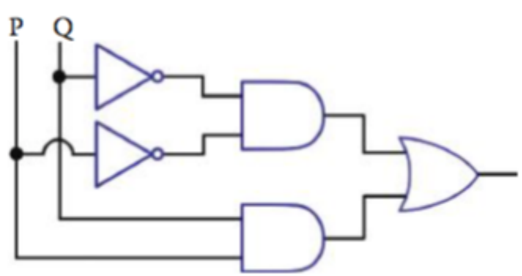
Ekvivalens krets

P	Q	!P	!Q	A = (!P och !Q)	B = (P och Q)	A eller B
F	F	S	S			
F	S	S	F			
S	F	F	S			
S	S	F	F			



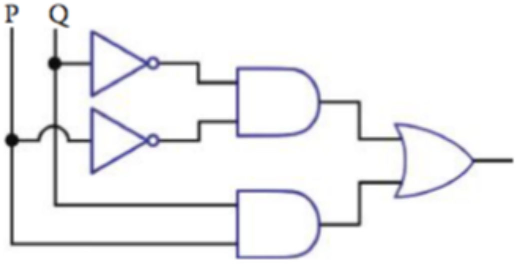
Ekvivalens krets

P	Q	!P	!Q	A = (!P och !Q)	B = (P och Q)	A eller B
F	F	S	S	S		
F	S	S	F	F		
S	F	F	S	F		
S	S	F	F	F		



Ekvivalens krets

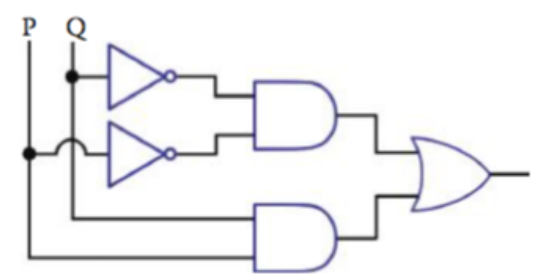
P	Q	!P	!Q	A = (!P och !Q)	B = (P och Q)	A eller B
F	F	S	S	S	F	
F	S	S	F	F	F	
S	F	F	S	F	F	
S	S	F	F	F	S	



Ekvivalens krets

P	Q	!P	!Q	A = (!P och !Q)	B = (P och Q)	A eller B
F	F	S	S	S	F	S
F	S	S	F	F	F	F
S	F	F	S	F	F	F
S	S	F	F	F	S	S

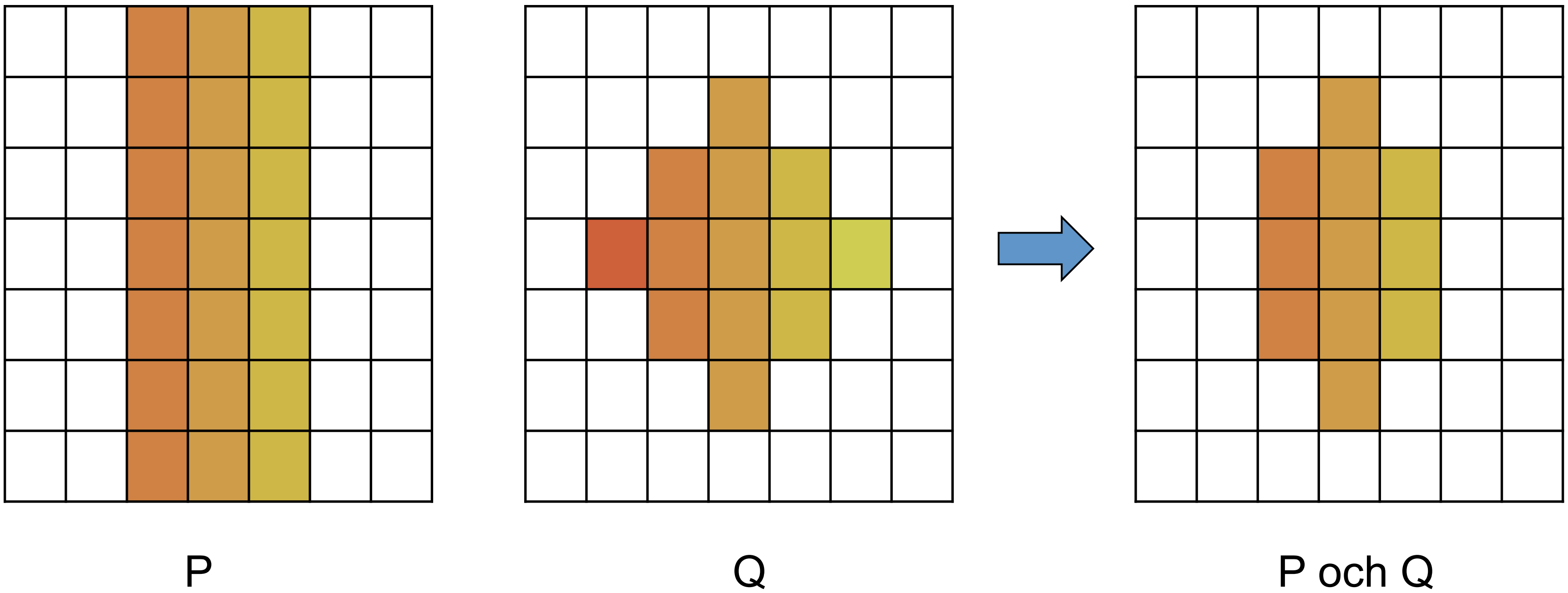
Sant om P och Q är lika, vilket skulle bevisas!



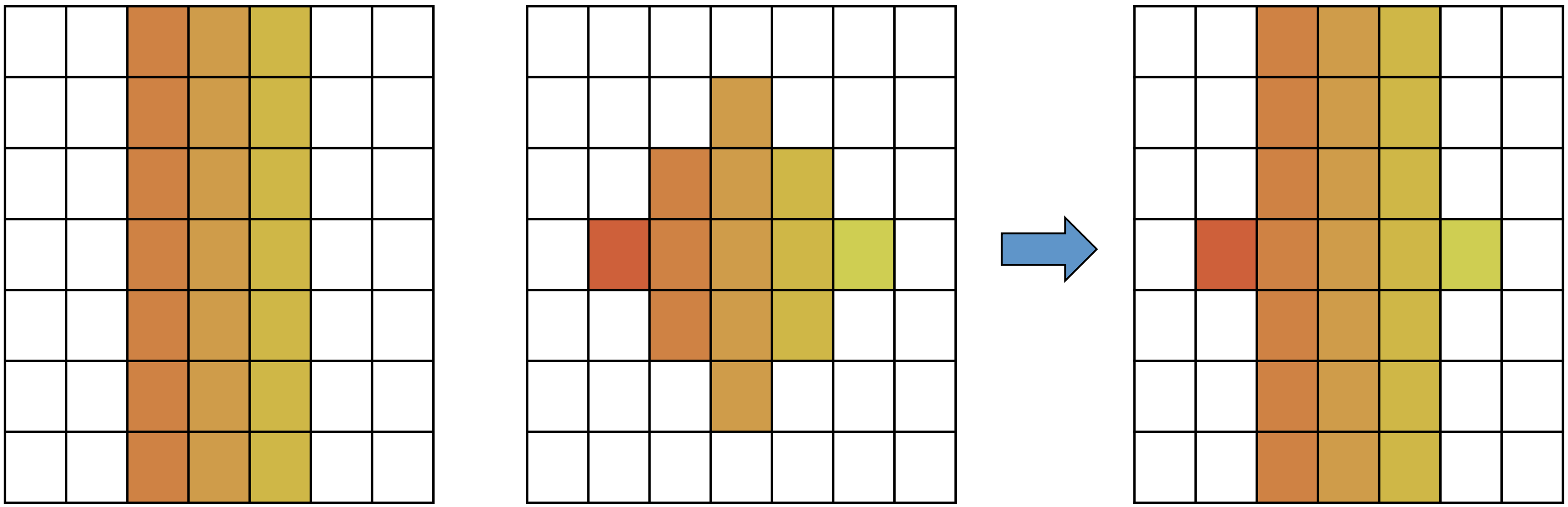
Bildbehandling

- Flera filter som används för bildbehandling eller bildkomposition använder logiska operatorer
- Ett exempel är att vi kan kombinera två bilder till en ny bild
- Vi kan anta att vit pixel är Falskt, alla andra färger är Sant

Bildbehandling

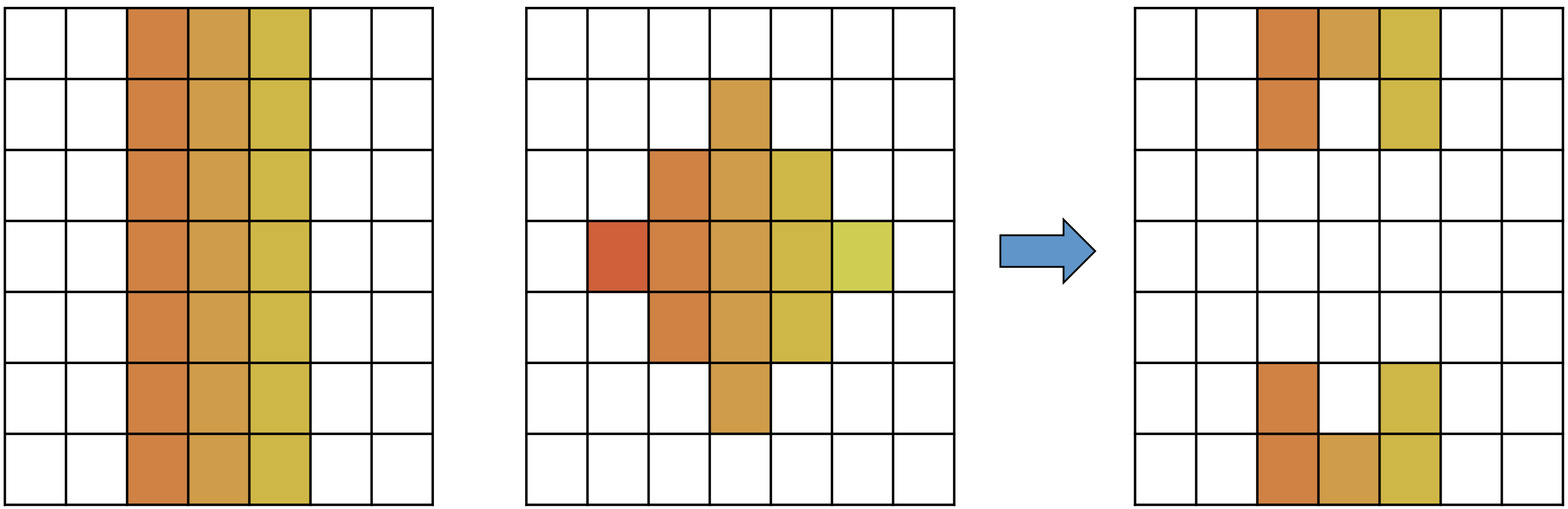


Bildbehandling



P Q P eller Q

Bildbehandling



P

Q

P och !Q

Databaser

- Databaser består (oftast) av fält där varje fält har en specifik typ av data
- Som exempel kan vi ha en databas med olika filmer
- Vilken information om en film är intressant att lagra?
- Det beror på vad vi ska använda databasen till!
- I vårt fall har vi:
  - Titel
  - År
  - Regissör
  - IMDB poäng

Filmdatabas

Titel	År	Regissör	IMDB
Nycken till frihet	1994	Frank Darabont	9,3
Inglorious Basterds	2009	Quentin Tarantino	8,3
Pulp Fiction	1994	Quentin Tarantino	8,9
Star Wars: Episod I – Det mörka hotet	1999	George Lucas	6,5
The Dark Knight Rises	2012	Christopher Nolan	8,4

Databaser

- SQL databaser är en av de äldsta och vanligaste typerna av databas
- För att hämta information från en SQL databas ställer vi sökfrågor med ett speciellt språk: SQL
- Alla sökfrågor har strukturen:
  SELECT fält1, fält2, … FROM tabell WHERE sökkriterier

Sökning i filmdatabasen

- Hämta titeln på alla filmer som regisserats av Quentin Tarantino:
  - SELECT Titel FROM FilmDB WHERE Regissör = ’Quentin Tarantino’

Titel
Inglorious Basterds
Pulp Fiction

Sökning i filmdatabasen

- Hämta titel och regissör på alla filmer med IMDB poäng över 8:
  - SELECT Titel, Regissör FROM FilmDB WHERE IMDB > 8

Titel	Regissör
Nycken till frihet	Frank Darabont
Inglorious Basterds	Quentin Tarantino
Pulp Fiction	Quentin Tarantino
The Dark Knight Rises	Christopher Nolan

Sökning i filmdatabasen

- Hämta titel och årtal på alla filmer med IMDB poäng över 8 och som kommit ut på 2000-talet:
  - SELECT Titel, År FROM FilmDB WHERE IMDB > 8 AND År >= 2000

Titel	År
Inglorious Basterds	2009
The Dark Knight Rises	2012