UPPSALA UNIVERSITET

Föreläsningsanteckningar VT22

# Logik och Bevisteknik

*Rami Abou Zahra*

## Contents

# 1. Föreläsning - Introduktion till Satslogik

**1.1. Historia.** Vad är ett matematiskt bevis?
Vad får användas i bevis?
Är matematiken motsägelsefri (Konsistent = motsägelsefritt)?

**1.2. Vad behövs för ett matematiskt bevis?**
- Ett påstående (även kallad utsaga) (ex.vis $\sqrt{2}$ är irrationellt), dessa har sanningsvärde *sant* eller *falskt*
- (Giltigt) Argument (resonemang)

**1.3. Exempel.**
- Påstående: *Varje kvadrat är en rektangel*
- Påstående: *Det finns en fyrhörning som inte är en rektangel*

---

Alltså finns en fyrhörning som inte är kvadrat

Detta är syftet med kursen, låt oss nu göra det mer abstrakt. (Låt x vara form, K(x) kvadrat, R(x) rektangel, F(x) fyrhörning) Då blir påståenden:
- $\forall x(K(x) \Rightarrow R(x))$
- $\exists x(F(x) \wedge \neg R(x))$

---

$\exists x(F(x) \wedge \neg(K(x)))$

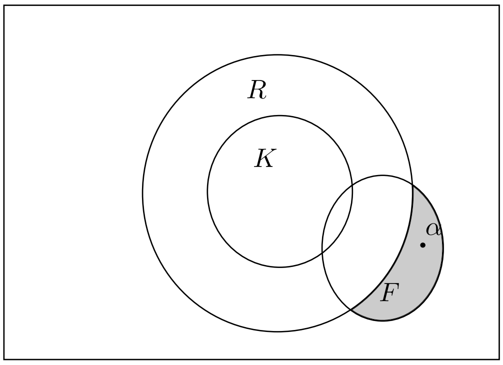


Figure 1. Grafisk tolkning

**Sats 1.1: Logiskt giltighet (Predikat-logik = 1:a ordningens logik)**

En följd av logiska steg kallas för *logiskt giltig* om det gäller $\forall$ tolkningar av $K, R, F$ (även vovvisar)

1.4. **Satslogik.** Vi behöver:

- Ett skriftligt språk (mängd av teckensträngar som betyder utsagor/satser)
- Regler/Formella bevis (Hur får man dra slutsatser av nya teckensträngar), *syntax*
- Tolka teckensträngarna i en ”verklighet” (sant eller falskt)

1.5. **Predikatlogik (1:a ordningens logik).** Skiljer sig från satslogik i och med att vi inte hanterar satser utan predikaten.

- Språk behövs, liknande med teckenmängd men vi kan även hantera elementen
- Formella bevis består av teckensträngsmanipulation, relation mellan teckensträngar
  (Exvis $A_1, A_2, A_3 \vdash B$ (A bevisar B))
- Vad betyder det att något är sant eller falskt? Om det går att visa B utan något så är det sant.

**Sats 1.2: Definiera sanning i struktur**

$\vdash B$ = Sant om det inte krävs något för att visa B:s sanningsvärde.

$A_1, A_2, A_3 \vDash B$ det vill säga om alla A krav är uppfyllda så gäller B

- Samband mellan $\vdash$ (formell bevisbarhet) och $\vDash$ (hur man tolkar något som sant eller falskt)

**Sats 1.3: Sundhetssatsen**

Detta säger att bevissystemet är sunt, allt man visar är sunt

$$A_1, A_2, A_3 \vdash B \Rightarrow A_1, A_2, A_3 \vDash B$$

**Sats 1.4: Adekvathet**

Om det är så att B är sann så fort alla A är sanna så finns det ett bevis, vi kommer kunna påstå att det finns ett formellt bevis för samma sak.

$$A_1, A_2, A_3 \vDash B \Rightarrow A_1, A_2, A_3 \vdash B$$

**Sats 1.5: Fullständighet**

Slår vi ihop dessa 2 (Sundhetssatsen och Adekvathet) får vi fullständighet

$$A_1, A_2, A_3 \vdash B \Leftrightarrow A_1, A_2, A_3 \vDash B$$

## 2. Satslogik (Propositional calculus)

2.1. **Språk (LP).**

- Satssymboler: $\sigma = \{p_0, p_1 \cdots, p_n\}$ (en *satslogisk signatur*, även kallas *språkets signatur*)

  LP($\sigma$) kallas för en *dialekt* av LP
- Alfabet i LP($\sigma$):
  - Alla symboler i $\sigma$
  - Konnektiver: $\Leftrightarrow, \Leftarrow, \Rightarrow, \vee, \wedge, \neg, \perp$
  - Paranteser: (, )

Viktigt att notera, $\perp$ är konnektiv men även sats, så den skapar inga nya formler.

**Sats 2.1: Mängden av formler i LP($\sigma$)**

Mängden definieras rekursivt (likt naturliga talen).

- Basfall:
  - Alla tecken i $\sigma$ är en formel
  - $\perp$ är en formel
  - Dessa kallas för *Atomära* formler
- Induktion:
  - Om $\varphi$ är en formel, så är $(\neg\varphi)$ en formel
- Låt $\square$ vara konnektiv:
  - Induktion $\square$:
    - Om $\varphi_1$ och $\varphi_2$ är formler, så är $(\varphi_1 \square \varphi_2)$ en formel

2.2. **Exempel.** Några exempel på formler i LP($\sigma$) där $\sigma = \{p, q, r\}$:

- $\perp, r, (p \Rightarrow (\neg r)), (p \vee (\neg p))$

Några som *inte* är formler:

- $(p\wedge, p \vee q \wedge r$ (inga paranteser!)

## 3. Föreläsning - Naturlig deduktion (syntax)

### 3.1. **Parantesers roll.**

Exempel: $p \to q \land r$. Hur skall vi placera ut paranteser, och behåller det formelns sanningsvärde? Vad händer om vi skriver $(q \land r)$ istället så att $(p \to (q \land r))$, eller motsatsen, $(p \to q)$ så att $((p \to q) \land r)$.

Det finns en konvention som hjälper oss att hålla koll på var och när och hur många paranteser som behövs.

- Skriv inte ut yttersta paranter
- $\neg$ binder starkare än $\land, \lor, \to, \leftrightarrow$
  - $\neg p \land q$ betyder $(\neg q) \land q$
- $\land, \lor$ binder starkare än pilarna $\to, \leftrightarrow$
  - $p \land q \to r \lor s \Leftrightarrow (p \land q) \to (r \lor s)$
- $\land, \lor$ binder lika hårt.
- $\to, \leftrightarrow$ binder lika hårt.
  - Ex: $p \lor q \land r$ ej klart vad som menas, här måste paranteser plaserar ut

Man kan formulera detta genom *parsingträd* där subnoderna *inte* får kommutera, detta kallas för att trädet är *ordnat.*

Ex: $(p \land (\neg q))$

Ex: $(p_1 \land p_2) \to (\neg p_2 \to (p_1 \to \perp))$. Detta är en *pilformel* och då kallas det för ett *huvudkonnektiv.*

**Sats 3.1**

Det sist tillagda konnektivet i formeln (motsvarar högsta noden i trädet) kallas för *huvudkonnektiv.*

Varje formel har ett entydigt träd. Men också tvärtom, givet ett träd så kan vi ”bygga upp” en entydig formel.

**Sats 3.2: Parsingträd**

Låt $T$ stå för ett träd. $T : \mathrm{LP}(\sigma) \to \{\text{parsingträd}\}$.

Vi definierar det induktivt, där basen är en p atom: $T(p) = \bullet p$. Sedan påbörjar induktionen:

- Om $\varphi$ formel med träd $T(\varphi)$, så $T((\neg\varphi)) =$ fig.
- Om $\varphi$ och $\phi$ formler med träd $T(\varphi)$ resp $T(\phi)$, så $T((\varphi\Box\phi)) =$ fig.

**Sats 3.3: Delformel**

En *delformel* till en formel $\varphi$ är en teckensträng från $\varphi$ som själv är en formel, då är det en delformel. Den triviala delformeln är $\varphi$ själv.

Exempel: $\varphi = (p_1 \land \neg p_2) \to ((p_2 \to p_3) \land p_4)$. Då är exempelvis $(p_2 \to p_3)$ en delformel eller $\neg p_2$. Däremot så är $p_3) \land p_4$ *inte* en delformel. I parsingträdet så motsvarar varje nod en delformel.

### 3.2. **Syntax.**

Hur man kan dra slutsatser på ett syntaktiskt sett:

Om $\Gamma$ är en mängd av formler och $\varphi$ är en formel vill vi studera relationen mellan dessa. Följer $\varphi$ av formlerna i $\Gamma$?

Vi kommer studera denna frågan på 2 sätt, syntaktiskt och semantiskt.

- Syntax:
  - Formella bevisregler, tex. $\dfrac{AB}{(a \land B)}$

  - $\Gamma \vdash \varphi$, $\Gamma$ bevisar $\varphi$, ”det finns ett bevis i naturlig deduktion som har slutsatsen $\varphi$ vars premisser/antaganden kommer från $\Gamma$”
  - Teckensträngsmanipulation
  - Naturlig deduktion
- Semantik:
  - Är $\varphi$ sann om alla formler i $\Gamma$ är sanna?
  - $\Gamma \vDash \varphi$

### 3.3. **Naturlig deduktion.**

För varje konnektiv som vi har kommer vi introducera 2 regler, en för att introducera konnektivet och en för att ta bort.

OBS! Reglerna i naturlig deduktion är *syntax*, det vill säga teckensträngsmanipulation. Alltså det spelar ingen roll vad som representeras, utan vilka regler som man får använda sig på dessa tecken. Eg. $A \wedge B \neq B \wedge A$ eftersom den första har tecknet $A$ på första platsen men det har inte den andra.

- $\wedge$ intro ($\wedge I$). Om vi har teckensträng $A$ och $B$ så kan vi $\dfrac{AB}{(A \wedge B)}$
- $\wedge$-elimination. $\dfrac{(A \wedge B)}{A}$ och $\dfrac{(A \wedge B)}{B}$
- $\rightarrow$-intro. $A \cdots B$, jag börjar med $A$ och jobbar mig mot $B$ S.T $\dfrac{A \vdots B}{(A \rightarrow B)}$. Efter $B$ får man dra vilken slutsats $A \rightarrow B$ där $A$ vilken formel som helst. Om $A$ är en premiss ovanför $B$ så får $A$ strykas.
- $\rightarrow$-elimination. Om $A$ gäller och $A \rightarrow B$ så vet vi att $B$ gäller (Modus pomens). Vi får dra $B$ som slutsats.
- $\vee$-intro. Om jag vet $A$, då vet jag $A$ eller $B$: $\dfrac{A}{A \vee B}$
- $\vee$-elimination. Om jag antar $A$ skall jag kunna bevisa exakt teckensträng som om jag antar $B$.
- $\neg$-intro. Antag $A$ så att man kommer fram till formen $\perp$ (botten kan tolkas som alltid falsk/motsägelse). Då får man dra slutsatsen $\neg A$. Här får man stryka premiss.
- $\neg$-elimination. Om man har visat $A$ och $\neg A$ så får man dra slutsatsen $\neg$ (ingen strykning).
- $\leftrightarrow$-intro. Om $A \rightarrow B$ och $B \rightarrow A$ då kan vi skriva $A \leftrightarrow B$.
- $\leftrightarrow$-elimination. Från $A \leftrightarrow B$ får vi ”2 fall”, $A \rightarrow B$ och $B \rightarrow A$

Exempel: Vi vill göra ett bevisträd som har följande slutsats, $(A \rightarrow (B \rightarrow A \wedge B))$. I detta träd kommer alla antaganden vara längst upp och slutsatser längst ner. Vi tittar på slutsatsen och märker att huvudkonnektivet är en pil, så vi kommer behöva använda pilintro. Då skall jag antag $A$ och försöka härleda $B \rightarrow (A \wedge B)$, men då måste jag visa att $B \rightarrow (A \wedge B)$ så vi måste anta B för att visa $A \wedge B$ och nu kan jag visa det jag ville visa. Vi använder $\wedge$-intro. Per vårat antagande gäller $A, B$ och därmed $A \wedge B$. Nu kan vi använda pil-elimination för att få bort pilen $B \rightarrow (A \wedge B)$

**Sats 3.4: Premiss**

En *premiss* är en formel som ej är struken och förekommer högst upp i bevisträdet.

**Sats 3.5: Slutsats**

Formeln som står längst ner i bevisträdet.

## 4. Förtydligande

**Sats 4.1: Disjunktion**

$A$ eller $B$. Uttrycker att minst en av $A$ och $B$ är fallet. Betecknas $A \vee B$ där $A, B$ kallas *disjunktionsled* eller *disjunkter*.

Det finns 2 typer av disjunktion, *uteslutande* och *icke-uteslutande*. Uteslutande disjunktion är helt enkelt A eller B (men inte båda) och icke-uteslutande är motsatt.

Disjunktionen är så kallad *inklusiv*, det vill säga det är helt okej att både $A$ och $B$ är sann. Det finns en så kallad *exklusiv* disjunktion vilket kommer lite senare (kanske $A \wedge B$?).

**Sats 4.2: Implikation**

Om $A$ så $B$ uttrycker att $B$ är fallet givet att $A$ är det. Detta betecknas $A \rightarrow B$. Här är $A$ *antecendenten* eller även *förledet* och $B$ är *konsekventen* eller *efterledet*. En implikation kallas också ivland en *materiell implikation*, *konditionalsats*, *villkorssats*.

Notera här att det kan bli lite klurigt med den näst sista raden i tabellen men! Antag att A är ”jag bajsar” och B är ”jag sitter ner”. Om jag inte bajsar så implicerar det fortfarande att jag *kan* sitta ner, varpå implikationen fortfarande gäller.

**Sats 4.3: Ekvivalens**

$A$ omm $B$ yttrycker konjunktionen av två implikationer, det vill säga om $A$ så $B$ oh om $B$ så $A$. Skrivs $A \leftrightarrow B$ och kallas även för *materiella ekvivalenser* eller *bikonditionalsatser*.

**Sats 4.4: Molekyler och Atomär**

En sats är *molekylär* om den är uppbyggd av en eller två andra satser med hjälp av ett konnektiv. I motsatt fall är satsen *atomär*. En molekylär sats innehåller minst ett konnektiv.

**Sats 4.5: n-ställig satsoperator**

En *n-ställig satsoperator* är en operator som ”tar in” $n$ variabler. Exempelvis är negationen $\neg$ en 1-ställig operator, medan $\wedge$ är en 2-ställig operator.

**Sats 4.6: Huvudoperator**

En *huvudoperator* är den operator som har tillämpats sist i uppbyggnaden av satsen. Exempelvis, i $(A \rightarrow (B \vee \neg A))$ är $\rightarrow$ huvudoperatorn. Däremot är $\vee$ huvudoperatorn i delformen $(B \vee \neg A)$.

### 4.1. **Sanningstabeller.**

För negation ¬:

¬A är sann ⇔ A är falsk ⇔ A inte är sann.

| A | ¬A |
|---|---|
| S | F |
| F | S |

För Konjunktion:

A∧B är sann ⇔ A är sann och B är sann.

| A | B | A ∧ B |
|---|---|---|
| S | S | S |
| S | F | F |
| F | S | F |
| F | F | F |

För disjunktion:

A∨B är sann ⇔ A är sann eller B är sann ⇔ minst en av A och B är sann.

| A | B | A∨B |
|---|---|---|
| S | S | S |
| S | F | S |
| F | S | S |
| F | F | F |

För implikation:

A→B är sann ⇔ om A är sann så B är sann ⇔ A är falsk eller B är sann. (Tips, vad är sista sanningsvärdet, dvs sanningsvärdet på B?)

| A | B | A→B |
|---|---|---|
| S | S | S |
| S | F | F |
| F | S | S |
| F | F | S |

För ekvivalens:

A↔B är sann ⇔ om A är sann så är B saan och om B är sann så är A sann. ⇔ A→B är sann och B→A är sann ⇔ A och B har samma sanningsvärde.

| A | B | A↔B |
|---|---|---|
| S | S | S |
| S | F | F |
| F | S | F |
| F | F | S |

**Sats 4.7: Satsparametrar**

Antalet satsparametrar är antalet ”variabler” i vår utsaga. Exvis, i $A \to (B \wedge C \leftrightarrow \neg A)$ har 3st satsparametrar.

## 5. Föreläsning - Naturlig deduktion forts.

Exempel: Visa att för alla heltal $n$ gäller att $n^2 + n$ är jämn.

**Bevis 5.1: Exempel**

Ett heltal är antingen jämn *eller* udda, så vi gör en falluppdelning:

- Fall 1 (jämn):
  - $n = 2k \Rightarrow (2k)^2 + 2k = 4k^2 + 2k = 2(2k^2 + k)$ alltså jämn
- Fall 2 (udda):
  - $n = 2k + 1 \Rightarrow (2k+1)^2 + 2k + 1 = 4k^2 + 4k + 1 + 2k + 1 = 2(2k^2 + 3k + 1)$ alltså jämn.

$\square$

Exempel: $A \vee B \vdash B \vee A$ (sanningstabellen för dessa är likadana). Vi skall producera ett bevisträd som har en premiss $A \vee B$ där slutsatsen är $B \vee A$

### 5.1. **Botten ($\perp$).**

Reglerna här är inte riktigt "intro/elimination" utan det har med motsägelser osv att göra.

Antag att vi vill visa $A$, då har vi premissen $\neg A$ så att vi kommer fram till $\perp$. Då kan vi dra slutsatsen $A$. Detta kallas för RAA = Reductio ad absurdum. Då får vi stryka $\neg A$

**Bevis 5.2: Exempel: Det finns oändligt många primtal**

Vi antar motsatsen (att det finns ändligt många primtal) och försöker visa motsägelse.

Låt $A$ = det finns oändligt många primtal. Då antar vi $\neg A$ och visar $\perp$, och sen drar slutsatsen $A$. $\square$

Bevisträd är inte entydiga.

**Sats 6.1**

En mängd av formler $\Gamma$ kallas *konsistent* om $\Gamma \nvdash \bot$. Det finns inget bevisträd som visar $\varphi$ från premisser i $\Gamma$. Kallas *inkonsistent* om $\Gamma \vdash \bot$

Exempel: $\{p_0, \neg p_0\}$ är inkonsistent, eftersom vi kan visa botten genom en $\neg$-elimination.

Exempel: $\{A \wedge B, \neg A \vee \neg B\}$ är inkonsistent.

Exempel: $\{A, B, C\}$ är konsistent (kan ej visa botten). Detta visas senare i kursen.

Vi har fortfarande vår signatur $\sigma$. Vi vill ge ett värde till varje formel i LP($\sigma$). Detta värde är ett sant eller falskt värde (sanningsvärde). I predikatlogik har värdet ”mer värde”, det är lite rikare och beskriver mer vad det betyder att något är sant eller falskt gentemot i satslogik där det är binärt.

**Sats 6.2**

En $\sigma$-struktur är en funktion $A : \sigma \to \{0, 1\}$ som tilldelar sanningsvärde till vare satssymbol där 0 betyder falsk och 1 sann.

Exempel: $\sigma = \{p, q\}$. Här finns det 4 olika $\sigma$-strukturer eftersom $p$ kan antingen vara falsk eller sann, och samma för $q$ alltså $2 \cdot 2 = 4$:

| | | |
|---|---|---|
| $A_1$ | 1 | 1 |
| $A_2$ | 1 | 0 |
| $A_3$ | 0 | 1 |
| $A_4$ | 0 | 0 |

Men vi vill ge sanningsvärden till *varje* formel i LP($\sigma$).

Antag att $A$ är en $\sigma$-struktur. Vi definierar en funktion $A^* : \{\text{formler i LP}(\sigma)\} \to \{0, 1\}$. Vi definierade formlerna i LP($\sigma$) genom induktion, vi får göra liknande här:

- Bas:
  - $A^*(\bot) = 0$
  - $A^*(p) = A(p)$ om $p \in \sigma$
- Induktion ($\neg$):
  - $A^*((\neg\varphi)) = 1 - A^*(\varphi)$
- Induktion ($\wedge$):
  - $A^*((\varphi \wedge \psi)) = 1 \Leftrightarrow A^*(\varphi) = A^*(\psi) = 1$
- Induktion ($\vee$):
  - $A^*((\varphi \vee \psi)) = 1 \Leftrightarrow$ minst en av $A^*(\varphi)$ och $A^*(\psi) = 1$
- Induktion ($\to$):
  - $A^*((\varphi \to \psi)) = 0 \Leftrightarrow A^*(\varphi) = 1$ och $A^*(\psi) = 0$
- Induktion ($\leftrightarrow$):
  - $A^*((\varphi \leftrightarrow \psi)) = 1 \Leftrightarrow A^*(\varphi) = A^*(\psi)$

OBS: $A^*(\varphi)$ beroe endast på $A(p)$ för de satssymboler $p$ som ingår i $\varphi$.

OBS: Olika värden på $A^*$ fås från sanningsvärdestabeller.

Exempel: $\sigma = \{p, q\}$, $\sigma$-strukturerna $A_1, A_2, A_3, A_4$ från tabell. Vi låter $\varphi = (\neg p) \wedge (p \to q)$. Nu vill vi ta reda på vad $\varphi$ får för värden på olika $A^*$:

| | $p$ | $q$ | $(\neg p)$ | $\wedge$ | $(p \to q)$ |
|---|---|---|---|---|---|
| $A_1$ | 1 | 1 | 0 | 0 | 1 |
| $A_2$ | 1 | 0 | 0 | 0 | 0 |
| $A_3$ | 0 | 1 | 1 | 1 | 1 |
| $A_4$ | 0 | 0 | 1 | 1 | 1 |

Då blir sanningsvärdestabellen för $A_n^*(\varphi)$ samma som kolonnen under $\wedge$

**Sats 6.3: Modell av en formell**

$\sigma$-strukturen $A$ kallas *modell* för $\varphi$ om $A^*(\varphi) = 1$, alltså en modell för när $\varphi$ är sann i $A$. Notation:

$$A \vDash \varphi$$

Satisfierbar är en egenskap som en formel kan ha, medan $\sigma$-strukturen kan vara en modell om den är sann.

Exempel: Från föregående exempel blev det sant för $A_3$ och $A_4$, alltså $A_3 \vDash \varphi$ och $A_4 \vDash \varphi$. Däremot, såg vi att $A_1 \nvDash \varphi$ och $A_2 \nvDash \varphi$ ($\varphi$ inte sann i $A_2$)

**Sats 6.4: Tautologi**

$\varphi$ kallas för *tautologi* om $A \vDash \varphi$ för varje $\sigma$-struktur $A$. Alltså om sanningsvärdestabellen har värdet 1 i varje rad. Notation:

$$\vDash \varphi$$

**Sats 6.5: Satisfierbar**

$\varphi$ kallas *satisfierbar* om $A \vDash \varphi$ för minst en struktur $A$. I föregående exempel är $\varphi$ satisfierbar.

**Sats 6.6: Falsifierbar**

$\varphi$ kallas *falsifierbar* om det är möjligt att göra den falsk. $A \nvDash \varphi$ för minst en struktur falsk. I föregående exempel är $\varphi$ falsifierbar

**Sats 6.7: Osatisfierbar**

$\varphi$ kallas *osatisfierbar* om formen är falsk i alla strukturer, det vill säga $A \nvDash \varphi$ i alla strukturer.

# 7. Logisk ekvivalens och konsekvens

## 7.1. **Logisk ekvivalens.**

**Sats 7.1: Logisk ekvivalens**

En relation som 2 formler har till varandra, de kan kallas för *logiskt ekvivalenta.*

Låt $\sigma$ vara en struktur, $\varphi$ och $\psi$ i LP($\sigma$). $\varphi$ och $\psi$ *kallas logiskt ekvivalenta* om $\vDash \varphi \leftrightarrow \psi$

Notation: $\varphi$ eq $\psi$, alltså samma sanningsvärde.

## 7.2. **Exempel.**

$p \to q$ eq $\neg p \vee q$. Här får man rita upp sanningstabellen:

| $p$ | $q$ | $p \to q$ | $\neg p \vee q$ |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 |

**Sats 7.2: Eq**

Eq är en ekvivalensrelation

**Bevis 7.1: Eq**

- Reflexiv ($\varphi$ eq $\varphi$)
- Symetrisk ($\varphi$ eq $\psi \Leftrightarrow \psi$ eq $\varphi$)
- Transitiv ($\varphi$ eq $\psi$ och $\psi$ eq $\xi \Rightarrow \varphi$ eq $\xi$)

$\square$

## 7.3. **Några viktiga ekvivalenser.**

$$\left.\begin{array}{l} p \vee (q \wedge r) \text{ eq } (p \vee q) \wedge (p \vee r) \\ p \wedge (q \vee r) \text{ eq } (p \wedge q) \vee (p \wedge r) \end{array}\right\} = \text{Distributiva lagar}$$

$$\left.\begin{array}{l} p \vee (q \vee r) \text{ eq } (p \vee q) \vee r \\ p \wedge (q \wedge r) \text{ eq } (p \wedge q) \wedge r \end{array}\right\} = \text{Associativa lagar}$$

$$\left.\begin{array}{l} p \vee q \text{ eq } q \vee p \\ p \wedge q \text{ eq } q \wedge p \end{array}\right\} = \text{Kommutativa lagar}$$

$$\left.\begin{array}{l} \neg(p \wedge q) \text{ eq } \neg p \vee \neg q \\ \neg(p \vee q) \text{ eq } \neg(p) \wedge \neg(q) \end{array}\right\} = \text{de Morgans lagar}$$

$$\left.\begin{array}{l} p \vee p \text{ eq } p \\ p \wedge p \text{ eq } p \end{array}\right\} = \text{Idempotenslagar}$$

$$\neg\neg p \text{ eq } p\} = \text{Lagen om dubbel-negation}$$

### 7.4. **Logisk konsekvens.**

**Sats 7.3: Logisk konsekvens**

Låt $\Gamma$ vara en mängd av formler i LP($\sigma$) och låt $\varphi$ en formel i $\Gamma$. $\varphi$ är en *logisk konsekvens* av $\Gamma$, skrivet $\Gamma \vDash \varphi$, om $\varphi$ är sann i varje modell för $\Gamma$.

Dvs, för varje $\sigma$-struktur gäller: Om $A \vDash \gamma$ för varje $\gamma \in \Gamma$, så kräver vi att $\varphi$ (den logiska konsekvensen) också ska vara sann. Här betyder $\vDash$ "när hälst varje struktur i $\Gamma$".

### 7.5. **Exempel.**

Visa att $\{p_1 \to p_2, p_1\} \vDash p_2$ (visa att $p_2$ är en logisk konsekvens av mängden).

Vi visar detta genom att låta $A$ vara en modell för $p_1 \to p_2$ och $p_1$, alltså $A$ är en struktur där de två är sanna. Detta betyder att $A$ är en $\sigma$-struktur, dvs $A^*(p_1 \to p_2) = 1$ och $A^*(p_1) = 1$. Det vi måste visa är att $A^*(p_2) = 1$. Detta kan vi göra genom att anta att $A^*(p_2) = 0$, vi vill få en motsägelse. Då blir $A^*(p_1 \to p_2) = 0$, men detta mostäger antagandet att $A^*(p_1 \to p_2) = 1$, alltså $A^*(p_2) = 1$

Om inte $\Gamma \vDash \varphi$ gäller, så skriver man $\Gamma \nvDash \varphi$, dvs "$\varphi$ är inte en logisk konsekvens av $\Gamma$" $\Leftrightarrow \neg(\varphi$ sann i varje modell för $\Gamma) \Leftrightarrow$ det finns någon (minst 1) modell för $\Gamma$ i vilken $\varphi$ är falsk.

### 7.6. **Exempel.**

$\{p_1 \to p_2, p_2\} \nvDash p_1$. Vi måste ge en motexempelstruktur $A$, så att $A^*(p_1 \to p_2) = A^*(p_2) = 1$ och $A^*(p_1) = 0$. Kom ihåg att struktur betyder tilldeling av sanningsvärde till satssymbolerna.

Ta tex följande:

- $A(p_1) = 0$
- $A(p_2) = 1$

Nu har vi hittat en struktur som gör VL sann och HL falsk.

### 7.7. **Algoritm för att avgöra $\Gamma \vDash \varphi$.**

Använd sanningsvärdestabeller för alla formler i $\Gamma$ och för $\varphi$. För de rader där alla i $\Gamma$ är sanna, kolla om $\varphi$ är sann.

### 7.8. **Exempel.**

Avgör om följande gäller $\neg A \vee \neg B$, $B \vee C \vDash (A \wedge C) \to B$:

| $A$ | $B$ | $C$ | $\neg A \vee \neg B$ | $B \vee C$ | $A \wedge C \to B$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 |

Rad 6 säger ju att VL = 1 men HL = 0, alltså finns modell för $\Gamma$ så att $(A \wedge C) \to B$ är falsk, alltså $\Gamma \nvDash (A \wedge C) \to B$

7.9. **Exempel.**
Avgör om följande gäller: $\neg(A \to B) \vDash A \wedge \neg B$. OBS, ekvivalent omformulering: $\neg(A \to B)$ eq $\neg(\neg A \vee B)$

Vi kan använda de Morgans lagar och arbeta in icke-tecknet:

$$\neg(\neg A \vee B) \text{ eq } (\neg\neg A \wedge \neg B) \text{ eq } A \wedge \neg B$$

Vi skulle dock visa om den var logiskt konsekvent men vi visade att den var logiskt ekvivalent, då är den alltid sann! Dvs, om den är logiskt ekvivalent så är den automatiskt logiskt konsekvent.

Vi kan resonera på följande sätt: Anta att $S$ är en $\sigma$-struktur där $\neg(A \to B)$ är sann.
Dvs, $S^*(\neg(A \to B)) = 1$. Då följer att $S^*(A \to B) = 0$. Stjärna av en pil är bara falsk om förledet är sant men efterledet är falskt, dvs $S^*(A) = 1$ och $S^*(B) = 0$. Nu vet vi vad atomerna måste ha för värde i $S$.
Alltså $S^*(\neg(B)) = 1$ och sedan $S^*(A \wedge \neg B) = 1$, då har vi visat att $\neg(A \to B) \vDash A \wedge \neg B$

7.10. **Kommentar.**
Det är inte alltid självklart att hela formeln kan substitueras om det är eq.

# 8. Substitution

**Sats 8.1: Substitution**

Syftet är att byta ut en delformel i en formel, med en annan formel. Vi börjar med att byta ut satssymboler dock.

Låt $\varphi$ vara en formel, $p$ en satssymbol, och $\psi$ en formel. Notation: $\varphi\,[\psi/p]$. Utläses "$\varphi$ med $\psi$ för $p$", eler alt "i formeln $\varphi$, byt ut $\psi$ mot $p$". Definieras med induktion över hur formeln $\varphi$ är uppbyggd ty vi vet att $\varphi$ är konstruerad mha induktion.

För att definiera med induktion:

- Bas:
  - $\varphi$ atom: $\varphi = q$:
    - $\varphi[\psi/p] = q[\psi/p]$:
      - $\psi$ om $q = p$
      - Annars $q$
  - $\varphi$ atom: $\varphi =\perp$:
    - $\varphi[\psi/p] =\perp [\psi/p] =\perp$
- Induktion $\neg$:
  - $\varphi$ är $(\neg\varphi_1)$:
    - $\varphi[\psi/p] = (\neg\varphi_1)[\psi/p] = \neg(\varphi_1[\psi/p])$
- Induktion $\square$:
  - $\varphi$ är $(\varphi_1\square\varphi_2)$, $\varphi[\psi/p] = (\varphi_1\square\varphi_2)[\psi/p] = (\varphi_1[\psi/p]\square\varphi_2[\psi/p])$

8.1. **Exempel.**

$(p_q \to p_2)[p_7 \wedge p_8/p_2] = (p_1 \to (p_7 \wedge p_8))$

Här behöver vi 2 satser:

**Sats 8.2**

Anta $\varphi_1$ eq $\varphi_2$. Då gäller $\varphi_1[\psi/p]$ eq $\varphi_2[\psi/p]$

Exempel på sats 8.2, tag $\varphi_1 = p_1 \to p_2$ och $\varphi_2 = \neg p_1 \vee p_2$. Vi vet att $p_1 \to p_2$ eq $\neg p_1 \vee p_2$ från tidigare. Sats 8.2 ger $(p_1 \to p_2)[p_7 \wedge p_8/p_2]$ eq $(\neg p_1 \vee p_2)[p_7 \wedge p_8/p_2]$, det vill säga
$p_1 \to (p_7 \wedge p_8)$ eq $\neg p_1 \vee (p_7 \wedge p_8)$

**Sats 8.3**

ntag $\psi_1$ eq $\psi_2$ och $\varphi$ en formel, $p$ en satssymbol. Då gäller: $\varphi[\psi_1/p]$ eller om vi gör $\varphi[\psi_2/p]$ så är de eq. Det vill säga $\varphi[\psi_1/p]$ eq $\varphi[\psi_2/p]$

Exempel på sats 8.3, visa att $\neg(A \to B)$ eq $\neg(\neg A \vee B)$. Låt $\psi_1 = A \to B$ och $\psi_2 = \neg A \vee B$. Här har vi $\psi_1$ eq $\psi_2$.
Låt $\varphi = \neg p$, då ger sats 8.3 att $\varphi[\psi_1/p]$ eq $\varphi[\psi_2/p]$, det vill säga $(\neg p)[\psi_1/p]$ eq $(\neg p)[\psi_2/p]$. Nu byter vi ut $p$ mot $\psi_1$, dvs $\neg\psi_1$ eq $\neg\psi_2$, dvs $\neg(A \to B)$ eq $\neg(\neg A \vee B)$

## 8.2. **Normalformer.**

**Sats 8.4: DNF**

En formel är på *disjunktiv normalform (DNF)* om den är på formen:

$$\varphi_1 \vee \varphi_2 \vee \cdots \vee \varphi_n$$

Där varje $\varphi_i$ är en konjunktion av atomer och/eller negerade atomer.

Exempel, $(p_1 \wedge p_2 \wedge \neg p_3) \vee (p_8 \wedge p_7) \vee (\neg p_8)$ är på DNF. Samma sak med $p_1 \wedge p_2 \wedge \neg p_3$ (här är $n = 1$). Ett exempel på något som inte är på DNF är $\neg(p_1 \wedge p_2)$ eftersom negationen sker inte framför en atom. Däremot är $\neg p_1 \vee \neg p_2$ som är på DNF, och enligt de Morgans lag är de eq.

Exempel, skriv $p \leftrightarrow q$ på DNF. Då vill vi helt enkelt hitta en eq formel som vi kan substituera och som är på DNF.
Vi får $p \leftrightarrow q$ eq $(p \to q) \wedge (q \to p)$ eq $(\neg p \vee q) \wedge (\neg q \vee p)$. Här kan vi använda distributiva lagar så vi får att det är eq $((\neg p \vee q) \wedge \neg q) \vee ((\neg p \vee q) \wedge p)$. Men här har vi tyvärr ”eller”, vi vill ha och! Vi kör distributiva lagar så vi får
eq $((\neg p \wedge \neg q) \vee (q \wedge \neg q)) \vee ((\neg p \wedge p) \vee (q \wedge p))$ eq $(\neg p \wedge \neg q) \vee (q \wedge p)$ är på DNF.

**Sats 8.5: Varje formel kan skrivas på DNF**

Låt $\varphi$ vara en formel. Då finns formel $\psi$ på DNF så att $\varphi$ eq $\psi$. Detta gäller även för KNF. Generellt, varje formel $\varphi$ i LP($\sigma$) har formel $\varphi_1$ och $\varphi_2$ så att:

- $\varphi$ eq $\varphi_1$ eq $\varphi_2$
- $\varphi_1$ på DNF
- $\varphi_2$ på KNF

Detta visas med ett exempel som ger en algoritm.

**Bevis 8.1: Varje formel kan skrivas på DNF**

$\varphi = ((p \leftrightarrow q) \to r) \to q$. Skriv på DNF. Skriv sanningstabellen:

| $p$ | $q$ | $r$ | $(p \leftrightarrow q)$ | $\to r$ | $\to$ | $q$ |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | | | 1 | |
| 1 | 1 | 0 | | | 1 | |
| 1 | 0 | 1 | | | 0 | |
| 1 | 0 | 0 | | | 0 | |
| 0 | 1 | 1 | | | 1 | |
| 0 | 1 | 0 | | | 1 | |
| 0 | 0 | 1 | | | 0 | |
| 0 | 0 | 0 | | | 1 | |

Markera raderna där huvudkonnektivet är 1. (Fyll i tabellen)

Vi ser av tabellen att $\varphi$ eq $(p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (\neg p \wedge q \wedge r) \vee (\neg p \wedge q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge \neg r)$ där varje parantes motsvarar varje rad där huvudkonnektivet är 1 enligt tabellen.

Notering, $\perp$ är på DNF $\square$

**Sats 8.6: KNF**

*Konjunktiv normalform.* Då är det helt enkelt som DNF men med och istället:

$$\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n$$

Där $\varphi_i$ = disjunktion av atom/¬atom

**Lemma 8.1: Följdsats av KNF**

Låt $\varphi$ vara en formel. Då finns $\psi$ på KNF så att $\varphi$ eq $\psi$

Kan vi använda en liknande algoritm som i beviset för DNF? Om vi vänder på algoritmen, det vill säga vi markerar raderna med 0 och för varje rad med noll tar vi tvärtom så där det är 1 skriver vi exempelvis $\neg p$. Detta funkar eftersom det vi har gjort är använt de Morgans lagar och vänt på lite saker.

## 9. Funktionellt kompletta mängder av konnektiv

Vi har observerat att varje formel kan skrivas på DNF. Vi har samtidigt visat att varje formel har en annan formel som endast skrivs med $\wedge, \vee, \neg$. Detta kallas för en *funktionellt komplett mängd av konnektiv.*

**Sats 9.1: Funktionellt komplett mängd av konnektiv**

En mängd av konnektiver kallas *funktionellt komplett* om varje formel i LP($\sigma$) kan skrivas ekvivalent med en formel som endast innehåller konnektiver från mängden. "De räcker till för att säga allt som går att säga".

Exempel:
Mängden $\{\vee, \wedge, \neg\}$ är funktionellt komplett.
Vi påstår även att $\{\wedge, \neg\}$ är också funktionellt komplett. För att visa detta påstående behöver vi att varje formel i LP($\sigma$) är ekvivalent med att bara använda dessa konnektiv.

**Bevis 9.1: Föregående mängd är funk. komp**

$\varphi$ i LP($\sigma$). Låt $\varphi_1$ eq $\varphi$ och $\varphi_1$ på KNF (eller DNF, spelar ej roll). Men i KNF kanske det finns eller tecken, men vi vill inte ha det i vårat bevis. Vi måste göra något för att få bort eventuella eller tecken. OBS, varje gång vi använder "eller" så kan vi använda de Morgans lag och skriva om den med "och" och "icke":

$$A \vee B \text{ eq } \neg(\neg A \wedge \neg B)$$

Använder vi andra varianten av de Morgans lag så kan vi göra samma sak med DNF och därmed visa att mängden $\{\vee, \neg\}$ är funktionellt komplett. $\square$

Exempel på en mängd som *inte* är funktionellt komplett:
Mängden $\{\vee, \wedge\}$ är *inte* funktionellt komplett. Det räcker med att visa att en formel inte går att skriva med hjälp av dessa konnektiv. Vi låter mängden $A$ vara mängden av alla formler som går att skriva med konnektiven $\vee, \wedge$. Vi vill definiera den konkret, så vi härmar definitionen av formler:

- Bas
  - $p \in A$ om $p$ satssymbol ($p \in \sigma$)
- Induktion
  - Om $\varphi_1$ och $\varphi_2$ tillhör $A$, så $(\varphi_1 \wedge \varphi_2) \in A$ och $(\varphi_1 \vee \varphi_2) \in A$

Betrakta följande $\sigma$-strukturen A: a$(p) = 1 \forall p \in \sigma$

Om $\varphi \in A$, så gäller A$^*(\varphi) = 1$. (Beviset ges av induktion på $A$). Men då kan vi betrakta $\psi = \perp \in$ LP($\sigma$). Per definition så är A$^* \perp = 0$, men vi har ju definierat $\sigma$-strukturen att alltid vara sann, och då kan vi inte skriva alla formler med hjälp av enbart $\wedge, \vee$. Alltså är mängden inte funktionellt komplett.

Exempel: Konnektivet | (sheffers streck) har följande sanningstabell:

| $p$ | $q$ | $p\|q$ |
|---|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 1 |

Vi påstår att mängden $\{|\}$ är funktionellt komplett. Idé, skriv något vi vet är funktionellt komplett med hjälp av strecket och använd *eq*.

Vi vill knyta ihop detta med semantiken och syntax i satslogiken:

| Syntax | Semantik |
|---|---|
| $\Gamma \vdash \varphi$ (det finns ett bevissträd i naturlig deduktion som visar $\varphi$ med premisser i $\Gamma$) | $\Gamma \vDash \varphi$ ($\varphi$ sann i varje modell för $\Gamma$) |

**Sats 9.2: Sundhetssatsen**

Om $\Gamma \vdash \varphi$ så $\Gamma \vDash \varphi$.
Specialfall, $\vdash \varphi \Rightarrow \vDash \varphi$

Beviset nämns kort:

**Bevis 9.2: Sundhetssatsen**

Detta görs egentligen av induktion på alla bevissträd. Det innebär att i basfallet har man korta bevis som bara har en formel. Bevisträdet $\varphi$ bevisar att $\varphi \vdash \varphi$. Om vi försöker baka ihop till ett induktionsbevis får vi $\varphi$ som bas och induktionssteg som $\wedge$-intro. Kortfattat, bevisreglerna ”bevarar sanning”. □

### 9.1. **Användning av sundhetssatsen.**

När vi pratade om logisk konsekvens så fanns det en formel vi kunde använda för att avgöra om saker var sant eller ej genom att skriva upp sanningsvärdestabeller. En anvädning av denna sats är för att visa att något inte är bevisbart i naturlig deduktion.

Exempel: Visa att $\nvdash \underbrace{(p \to q) \to (p \vee \neg q)}_{\varphi}$. Det hela går ut på att negera sundhetssatsen i någon mening, om vi kan visa att något inte är logiskt konsekvent så är det inte heller bevisbart, dvs om $\Gamma \nvDash \varphi$ så $\Gamma \nvdash \varphi$. Vi visar $\nvDash \varphi$, dvs $\nvDash (p \to q) \to (p \vee \neg q)$, alltså inte en tautologi. Om $p = 0, q = 1$ så är påståendet falskt, dvs vi har hittat en struktur så att den inte är en tautologi, alltså kan vi mha sundhetssatsen dra slutsatsen att $\nvdash \varphi$ eftersom om det inte är en tautologi så är $\vDash \varphi$ men det är den inte.

Påminnelse, en mängd kallas *konsistent* om $\Gamma \nvdash \perp$. Nu har vi en metod för detta!

Exempel: Visa att $\Gamma = \{p_1, p_2, p_3\}$ är konsistent. Vi vill visa att vi inte kan använda dessa premisser för att komma fram till botten. Vi börjar med att visa på semantik sidan att $\Gamma$ inte är en logisk konsekvens av $\perp$ dvs $\Gamma \nvDash \perp$. Låt $A$ vara en $\sigma$-struktur så att $A(p_1) = A(p_2) = A(p_3) = 1$, då bör $A^* \vDash \Gamma$ men $A^* \nvDash \perp$. Sundhetssatsen säger då att $\Gamma \nvdash \perp$.

Sundhetssatsen kan man säga är ”pilen från syntax till semantik”. Pilen från andra hållet kallas för adekvathetssatsen:

**Sats 9.3: Adekvathetssatsen**

Om man vet att något är en logisk konsekvens så får man dra slutsatsen i naturlig deduktion, mer formellt, om $\Gamma \vDash \varphi$ så $\Gamma \vdash \varphi$.

Vi får slutgiltigen att Sundhetssatsen + Adekvathetssatsen = Fullständighetssatsen När vi pratade om logisk konsekvens så fanns det en formel vi kunde använda för att avgöra om saker var sant eller ej genom att skriva upp sanningsvärdestabeller. En anvädning av denna sats är för att visa att något inte är bevisbart i naturlig deduktion.

Påstående: Varje konsistent mängd av formler i LP($\sigma$) har en modell. Här ger ”konsistent” en lettråd om att den befinner sig på semantik-sidan. Om man vill visa adekvathetssatsen så vill vi visa påståendet och att det är ekvivalent med adekvathetssatsen.