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:
- $\forall x(K(x) \Rightarrow R(x))$
- $\exists x(F(x) \wedge \neg R(x))$

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

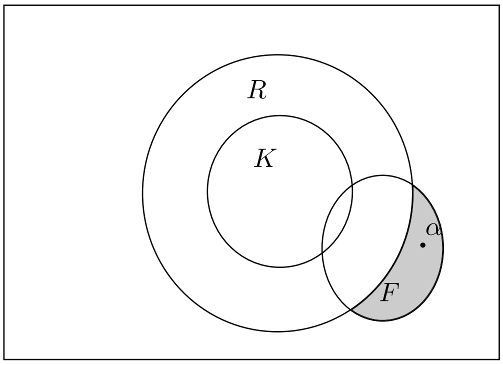

Figure 1. Grafisk tolkning

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(σ) kallas för en *dialekt* av LP
- Alfabet i LP(σ):
  - Alla symboler i σ
  - 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(σ)

Mängden definieras rekursivt (likt naturliga talen).

- Basfall:
  - Alla tecken i σ är en formel
  - $\perp$ är en formel
  - Dessa kallas för *Atomära* formler
- Induktion:
  - Om φ är en formel, så är $(\neg\varphi)$ en formel
- Låt $\square$ vara konnektiv:
  - Induktion $\square$:
    * Om φ_1 och φ_2 är formler, så är $(\varphi_1 \square \varphi_2)$ en formel

2.2. **Exempel.** Några exempel på formler i LP(σ) 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: $p \to q \wedge r$. Hur skall vi placera ut paranteser, och behåller det formelns sanningsvärde? Vad händer om vi skriver $(q \wedge r)$ istället så att $(p \to (q \wedge r))$, eller motsatsen, $(p \to q)$ så att $((p \to q) \wedge r)$.

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 $\wedge, \vee, \to, \leftrightarrow$
  - $\neg p \wedge q$ betyder $(\neg q) \wedge q$
- $\wedge, \vee$ binder starkare än pilarna $\to, \leftrightarrow$
  - $p \wedge q \to r \vee s \Leftrightarrow (p \wedge q) \to (r \vee s)$
- $\wedge, \vee$ binder lika hårt.
- $\to, \leftrightarrow$ binder lika hårt.
  - Ex: $p \vee q \wedge r$ 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: $(p \wedge (\neg q))$

Ex: $(p_1 \wedge p_2) \to (\neg p_2 \to (p_1 \to \perp))$. 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 φ formel med träd $T(\varphi)$, så $T((\neg\varphi)) =$ fig.
- Om φ och ϕ formler med träd $T(\varphi)$ resp $T(\phi)$, så $T((\varphi \Box \phi)) =$ fig.

Sats 3.3: Delformel

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

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

3.2. Syntax.

Hur man kan dra slutsatser på ett syntaktiskt sett:

Om Γ är en mängd av formler och φ är en formel vill vi studera relationen mellan dessa. Följer φ av formlerna i Γ?

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

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

  - $\Gamma \vdash \varphi$, Γ bevisar φ, ”det finns ett bevis i naturlig deduktion som har slutsatsen φ vars premisser/antaganden kommer från Γ”
  - Teckensträngsmanipulation
  - Naturlig deduktion
- Semantik:
  - Är φ sann om alla formler i Γ 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 $A \to B$. 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 $(A \to (B \vee \neg A))$ är $\to$ 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 $A \to (B \land C \leftrightarrow \neg A)$ 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 Γ kallas *konsistent* om $\Gamma \nvdash \bot$. Det finns inget bevisträd som visar φ från premisser i Γ. 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 σ. Vi vill ge ett värde till varje formel i LP(σ). 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 σ-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 σ-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(σ).

Antag att A är en σ-struktur. Vi definierar en funktion $A^* : \{\text{formler i LP}(\sigma)\} \to \{0, 1\}$. Vi definierade formlerna i LP(σ) 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 φ.

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

Exempel: $\sigma = \{p, q\}$, σ-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 φ 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

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

$$A \vDash \varphi$$

Satisfierbar är en egenskap som en formel kan ha, medan σ-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 $A_2 \nvDash \varphi$ (φ inte sann i A_2)

Sats 6.4: Tautologi

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

$$\vDash \varphi$$

Sats 6.5: Satisfierbar

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

Sats 6.6: Falsifierbar

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

Sats 6.7: Osatisfierbar

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

Notation: φ eq ψ, alltså samma sanningsvärde.

7.2. Exempel.

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

p	q	$p \rightarrow q$	$\neg p \vee q$
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 (φ eq φ)
- Symetrisk (φ eq $\psi \Leftrightarrow \psi$ eq φ)
- Transitiv (φ eq ψ och ψ eq $\xi \Rightarrow \varphi$ eq ξ)

$\square$

7.3. Några viktiga ekvivalenser.

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

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

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

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

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

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

7.4. **Logisk konsekvens.**

Sats 7.3: Logisk konsekvens

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

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

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 σ-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 ”φ är inte en logisk konsekvens av Γ” $\Leftrightarrow \neg$(φ sann i varje modell för Γ) $\Leftrightarrow$ det finns någon (minst 1) modell för Γ i vilken φ 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 Γ och för φ. För de rader där alla i Γ är sanna, kolla om φ 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	1	1
1	0	1	1	0	0
1	1	1	0	1	1

Rad 3 säger ju att VL = 1 men HL = 0, alltså finns modell för Γ så att $(A \wedge C) \to B$ är falsk, alltså $\Gamma \nvDash (A \wedge C) \to B$