Logik och bevisteknik
lite extra teori

Inger Sigstam

2011-04-26

1 Satslogik (eng: propositional logic)

1.1 Språket

Alfabetet består av följande symboler:
satssymbolerna p_0, p_1, p_2, ...
konnektiverna $\perp$, $\vee$, $\wedge$, $\longrightarrow$, $\longleftrightarrow$.
paranteser: (,).

PROP är mängden av "tillåtna teckensträngar" i detta alfabet, och PROP definieras induktivt enligt följande. Elementen i PROP kallas i fortsättningen för *satser*.

1.1 Definition. Teckensträngen φ tillhör mängden PROP, dvs $\varphi \in$PROP, om den enligt följande klausuler är en sats.

Bas : $\perp$ är en sats, och p_i är en sats, för varje $i \in \mathbf{N}$.

Ind$\neg$: Om φ är en sats, så är $(\neg\varphi)$ en sats.

Ind$\vee$: Om φ och ψ är satser, så är $(\varphi \vee \psi)$ en sats.

Ind$\wedge$: Om φ och ψ är satser, så är $(\varphi \wedge \psi)$ en sats.

Ind$\longrightarrow$: Om φ och ψ är satser, så är $(\varphi \longrightarrow \psi)$ en sats.

Ind$\longleftrightarrow$: Om φ och ψ är satser, så är $(\varphi \longleftrightarrow \psi)$ en sats. $\square$

De satser som kastades in i PROP i bassteget, dvs $\perp$ och alla satssymboler, kallas *atomära satser*.
Underförstått i alla induktivt definierade mängder är att ingenting annat kommer in i mängden än de objekt som ska tillhöra mängden enligt någon av klausulerna.

1.2 Exempel. Teckensträngarna $(p_8 \wedge (\neg\perp))$, $(p_1 \vee (p_0 \wedge p_0))$ och $(p_4 \longleftrightarrow (\neg p_4))$ är satser, men t ex $(p_5 \wedge \neg)$ är inte en sats. $\square$

Observera att konnektiverna $\vee$, $\wedge$, $\longrightarrow$ och $\longleftrightarrow$ är *två-ställiga*. T ex är $(p_1 \longrightarrow p_2 \longrightarrow p_3)$ inte en sats. Vi behöver välja om vi menar $((p_1 \longrightarrow p_2) \longrightarrow p_3)$ eller $(p_1 \longrightarrow (p_2 \longrightarrow p_3))$. Men för att öka läsbarheten, så ska vi använda oss av följande *parenteskonvention*.

(i) Skriv inte ut de yttre parenteserna om de är onödiga.
T ex skriver vi $p_8 \longrightarrow p_7$ i stället för $(p_8 \longrightarrow p_7)$.

(ii) Konnektivet $\neg$ binder starkare än alla andra konnektiv.
Vi skriver t ex $p_2 \wedge \neg p_6$, när vi menar $(p_2 \wedge (\neg p_6))$.

(iii) Konnektiverna $\wedge$ och $\vee$ binder starkare än $\longrightarrow$ och $\longleftrightarrow$.
T ex skriver vi $\varphi \wedge \psi \longrightarrow \neg p_4$ som en förkortning av satsen $((\varphi \wedge \psi) \longrightarrow (\neg p_4))$.

Anm: Konnektiverna $\wedge$ och $\vee$ binder *lika starkt*. Om man skriver $\varphi \wedge \psi \vee \sigma$ så kanske vi menar $\varphi \wedge (\psi \vee \sigma)$, men lika gärna $(\varphi \wedge \psi) \vee \sigma$. I denna situation måste vi använda parenteserna för att klargöra vilken sats vi menar!
På liknande sätt gäller att konnektiverna $\longrightarrow$ och $\longleftrightarrow$ binder *lika starkt*.

1.2 Semantiken

Objekten i PROP kallades satser. Man ska tänka på dessa som namn på *fullständiga meningar*. En fullständig mening är en mening med subjekt och predikat, t ex ”Kalle sitter på månen.” eller ”$x + 4 = 7$”. Ett annat kännetecken på en fullständig mening är att den kan ha ett sanningsvärde, dvs den kan vara sann eller falsk.
De minsta beståndsdelarna i satslogikens språk är de atomära satserna, dvs satssymbolerna och $\perp$. Givet att alla satssymboler har ett sanningsvärde, så blir sanningsvärdet för varje enskild sats i PROP entydigt bestämt, enligt följande definition.

1.3 Definition. Låt för varje $i \in \mathbf{N}$ sanningsvärdet av satssymbolen p_i vara $v(p_i)$. Dvs $v(p_i) = 1$ eller $v(p_i) = 0$ för alla $i \in \mathbf{N}$. Om $v(p_i) = 1$ så innebär det att p_i är sann, och $v(p_i) = 0$ innebär att p_i är falsk. Sanningsvärdet $v(\varphi)$ definieras nu enligt:

(i) $v(\perp) = 0$. Om $\varphi = p_i$, så $v(\varphi) = v(p_i)$ redan definierat. (Dvs $\perp$ är alltid falsk, och satssymbolernas sanningsvärde är givet av v.)

(ii) Om $\varphi = \neg\psi$, så sätt $v(\varphi) = 1 - v(\psi)$. (Dvs $\neg\psi$ är sann om och endast om ψ är falsk.)

(iii) Om $\varphi = \psi_1 \vee \psi_2$, så sätt $v(\varphi)$ =max$(v(\psi_1, v(\psi_2))$. (Dvs $\psi_1 \vee \psi_2$ är sann om och endast om *minst en av* satserna ψ_1 och ψ_2 är sann.)

(iv) Om $\varphi = \psi_1 \wedge \psi_2$, så sätt $v(\varphi)$ =min$(v(\psi_1, v(\psi_2))$. (Dvs $\psi_1 \wedge \psi_2$ är sann om och endast om *både* ψ_1 och ψ_2 är sanna.)

(v) Om $\varphi = \psi_1 \longrightarrow \psi_2$, så låt $v(\varphi) = 0$ om och endast om $v(\psi_1) = 1$ och $v(\psi_2) = 0$. (Dvs satsen $\psi_1 \longrightarrow \psi_2$ är falsk om och endast om ψ_1 är sann och ψ_2 är falsk.)

(vi) Om $\varphi = \psi_1 \longleftrightarrow \psi_2$, så låt $v(\varphi) = 1$ om och endast om $v(\psi_1) = v(\psi_2)$. Dvs satsen $\psi_1 \longleftrightarrow \psi_2$ är sann om och endast om satserna ψ_1 och ψ_2 har *samma sanningsvärde*. $\square$

Observera att klausulerna i denna definition gör precis det som sanningsvärdestabeller gör. Så när du vill ta reda på sanningsvärdet hos en sats, så använd sanningsvärdestabell. När vi skriver sanningsvärdestabeller, så använder vi värdet 1 för SANN och värdet 0 för FALSK.

1.4 Definition.

(i) En sats φ kallas *satisfierbar* om den är sann i någon sanningsvärdestilldelning av sanningsvärden till satssymbolerna.

(ii) En mängd Γ av satser kallas *satisfierbar* om det finns någon sanningsvärdestilldelning av sanningsvärden till satssymbolerna sådan att alla satser i Γ blir sanna. $\square$

En sats är alltså satisfierbar om det finns *minst en* 1:a i sanningsvärdestabellen för satsen. En mängd Γ av satser är satisfierbar om man kan göra alla satser i Γ sanna *samtidigt*.

1.5 Exempel. Satsen $p \vee (q \wedge \neg q)$ är satisfierbar eftersom den är sann när p är sann. Däremot är satsen $p \wedge (q \wedge \neg q)$ inte satisfierbar.
Mängden $\{p \longrightarrow q,\, p\}$ är satisfierbar, eftersom båda satserna i mängden är sanna när både p och q är sanna. Se sanningsvärdestabellen i Ex.1.9. $\square$

1.6 Definition. En sats φ kallas för en *tautologi* om den är sann i alla möjliga sanningsvärdestilldelningar av sanningsvärden till satssymbolerna. $\square$

För att ta reda på om en sats är en tautologi så kan man ställa upp sanningsvärdestabellen för den, och kontrollera om det blir värdet 1 på varje rad.

1.7 Exempel. Satsen $\neg\bot$ är en tautologi, och t ex $p \vee \neg p$ är också en tautologi. Andra exempel är $(p \longrightarrow q) \longleftrightarrow (\neg p \vee q)$ och $p \wedge (q \vee r) \longleftrightarrow (p \wedge q) \vee (p \wedge r)$. Vi visar sanningsvärdestabeller för de två senare satserna.

p	q	$(p \longrightarrow q)$	$\longleftrightarrow$	$(\neg p$	$\vee$	$q)$
1	1	1	1	0	1	1
1	0	0	1	0	0	0
0	1	1	1	1	1	1
0	0	1	1	1	1	0
		1	5	2	4	3

Siffrorna i den understa raden anger i vilken ordning man har evaluerat kolumnerna. I detta exempel är 5 den sist evaluerade kolumnen, den som anger sanningsvärdet för hela satsen för de olika sanningsvärdestilldelningarna. I kolumn 5 finns endast 1:or, vilket betyder att satsen är sann i alla tilldelningar, dvs att satsen $(p \longrightarrow q) \longleftrightarrow (\neg p \vee q)$ är en tautologi. Vi gör även en sanningsvärdestabell för $p \wedge (q \vee r) \longleftrightarrow (p \wedge q) \vee (p \wedge r)$.

p	q	r	p	$\wedge$	$(q \vee r)$	$\longleftrightarrow$	$(p \wedge q)$	$\vee$	$(p \wedge r)$
1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	0
1	0	1	1	1	1	1	0	1	1
1	0	0	1	0	0	1	0	0	0
0	1	1	0	0	1	1	0	0	0
0	1	0	0	0	1	1	0	0	0
0	0	1	0	0	1	1	0	0	0
0	0	0	0	0	0	1	0	0	0
			1	3	2	7	4	6	5

De tre vänstra kolumnerna som listar alla möjliga tilldelningar av sanningsvärden till de ingående satssymbolerna kallas *referenskolumner* Vi fyllde alltså i kolumn 3 med hjälp av värdena i kolumn 1 och 2. Sedan fyllde vi kolumn 4 och 5 med hjälp av referenskolumnerna, därefter kolumn 6 med hjälp av kolumn 4 och 5. Sist fyllde vi i kolumn 7 med hjälp av kolumnerna 3 och 6. Eftersom kolumn 7 endast innehåller 1:or, så är satsen en tautologi. $\square$

Observera att om en sats beror av n stycken olika satssymboler $p_1, p_2, \dots, p_n$, så behöver man en sanningsvärdestabell där antalet rader är 2^n. (Multiplikationsprincipen, varje satssymbol kan ha 2 sanningsvärden.)

1.8 Definition. En sats σ kallas för en *tautolog konsekvens* av satserna $\varphi_1, \varphi_2, \dots, \varphi_n$ om σ är sann i alla sanningsvärdestilldelningar som gör alla satserna $\varphi_1, \varphi_2, \dots, \varphi_n$ sanna.
Notation: $\varphi_1, \varphi_2, \dots, \varphi_n \models \sigma$. $\square$

Mer allmänt, om Γ är en mängd satser, och σ en sats, så säger vi att σ är en tautolog konsekvens av Γ och skriver $\Gamma \models \sigma$ om σ är sann närhelst *alla* satser i Γ är sanna. Om mängden Γ är tom, så kan vi skriva $\models \sigma$, vilket innebär att σ är sann i alla sanningstilldelningar, dvs σ är en tautologi.

1.9 Exempel. Satsen q är en tautolog konsekvens av satserna $p \longrightarrow q$ och p. Dvs vi kan skriva $p \longrightarrow q,\ p \models q$. Man kan använda sanningsvärdestabell för att undersöka detta:

p	q	$p \longrightarrow q$	p	q
1	1	1	1	1
1	0	0	1	0
0	1	1	0	1
0	0	1	0	0
		1	2	3

De rader i kolumn 1 och 2 som båda har en 1:a, har en 1:a även i kolumn 3. (Det finns i detta exempel bara en sådan rad, den översta.) Därför gäller att q är en tautolog konsekvens av satserna $p \longrightarrow q$ och p. Vi kan alltså skriva $p \longrightarrow q,\ p \models q$. $\square$

1.10 Exempel. p är *inte* en tautolog konsekvens av $p \longrightarrow q,\ q$, dvs $p \longrightarrow q,\ q \not\models p$. För om p är falsk och q är sann, så är satserna $p \longrightarrow q$ och q sanna medan p är falsk.

p	q	$p \longrightarrow q$	q	p
1	1	1	1	1
1	0	0	0	1
0	1	1	1	0
0	0	1	0	0
		1	2	3

Eftersom det finns (minst) en rad där både kolumn 1 och 2 visar sant, och kolumn 3 visar falskt, så har vi visat att $p \longrightarrow q,\ q \not\models p$. $\square$

1.11 Definition. Två satser σ_1 och σ_2 kallas för *tautologt ekvivalenta* om $\sigma_1 \models \sigma_2$ och $\sigma_1 \models \sigma_2$, dvs om var och en av dem är en tautolog konsekvens av den andra.
Notation: $\sigma_1 \approx \sigma_2$. $\square$

Påståendet att satserna σ_1 och σ_2 är tautologt ekvivalenta kan även formuleras som att satsen $\sigma_1 \longleftrightarrow \sigma_2$ är en tautologi. Om två satser är tautologt ekvivalenta, så är sanningsvärdena för de båda satserna lika på varje rad i sanningsvärdestabellen. I Exempel 1.5 ser vi att $p \longrightarrow q \approx \neg p \vee q$ och att $p \wedge (q \vee r) \approx (p \wedge q) \vee (p \wedge r)$.

1.3 Syntaxen - de formella bevisen

Läs i boken kapitel 5, 6 (regler för $\neg$, $\vee$, $\wedge$) och kapitel 8 (regler för $\longrightarrow$ och $\longleftrightarrow$). Samtliga regler finns sammanställda i boken sid 557 och framåt.

Det är praktiskt, men ej nödvändigt, att även använda regeln **Reit** (för reiteration). Denna regel tillåter dig att upprepa en sats som står högre upp i samma (del-)bevis. (Ett *delbevis* är ett bevis som förekommer inne i ett annat bevis. Läs om detta i kap 6.2.)

1.12 Definition. Ett *bevis* med slutsats σ och premisser $\psi_1, \ldots, \psi_n$ är en lista

$$S_1, S_2, \ldots, S_k$$

av satser så att $\sigma = S_k$ och varje sats S_i antingen är en av premisserna $\psi_1, \ldots, \psi_n$ eller följer med hjälp av en bevisregel från en eller flera satser S_j med $j < i$. $\square$

Om man skriver sitt bevis som en lista uppifrån och ner, så ska det alltså på varje rad i listan stå antingen en av premisserna, eller en sats som följer från satser högre upp i listan från en av reglerna. Man skriver sin sats, och till höger på raden skriver man vilken regel som har använts och vilken eller vilka rader högre upp man hänvisar till.

1.13 Definition.

(i) $\psi_1, \ldots, \psi_n \vdash \sigma$ om det finns ett bevis med slutsats σ i vilket alla premisser som används är med bland $\psi_1, \ldots, \psi_n$.

(ii) $\vdash \sigma$ om det finns ett bevis *utan premisser* med slutsats σ.

$\square$

Obs att beviset inte måste använda alla satser $\psi_1, \ldots, \psi_n$. Men det får inte använda någon annan premiss än dessa! Om man t ex har visat att $A, B \vdash C$ så är det även sant att $A, B, D \vdash C$ för vilken sats D som helst.

1.14 *Sundhetssatsen för satslogiken.* *Låt $\Gamma \subseteq$PROP och $\sigma \in$PROP. Då gäller att*

$$\Gamma \vdash \sigma \Longrightarrow \Gamma \models \sigma$$

dvs allt som kan bevisas i det formella systemet gäller också semantiskt. $\square$

Sundhetssatsen säger att vi med vårt bevissystem från sanna premisser inte kan bevisa några osanna satser. Beviset går just ut på att visa att varje bevisregel bevarar sanning. Se boken avsnitt 8.3.

1.15 *Fullständighetssatsen för satslogiken.* *Låt $\Gamma \subseteq$PROP och $\sigma \in$PROP. Då gäller att*

$$\Gamma \models \sigma \Longrightarrow \Gamma \vdash \sigma$$

dvs om σ är en tautolog konsekvens av Γ, så finns det också ett formellt bevis vars alla premisser tillhör Γ och vars slutsats är σ. $\square$

Fullständighetssatsen säger att vi kan bevisa tillräckligt mycket: vi kan bevisa åtminstone allt som är sant. Observera att om Γ är tom, så har vi att $\models \sigma \Longleftrightarrow \vdash \sigma$, dvs att σ är en tautologi om och endast om σ kan bevisas utan premisser. Det krävs mer arbete för att bevisa fullständighetssatsen, en skiss av ett sådant bevis finns i pappret om sekventkalkyl. Det bygger på att om $\Gamma \models \sigma$, så går det inte att göra alla satser i Γ sanna samtidigt som man gör σ falsk.

1.16 Definition. En mängd Γ av satser kallas *inkonsistent* om $\Gamma \vdash \bot$.
Mängden Γ kallas *konsistent* om $\Gamma \nvdash \bot$. $\square$

Obs: $\Gamma \models \bot \iff \Gamma$ är inte satisfierbar.
Men denna observation kan man som följdsats till sundhets- och fullständighetssatserna visa:

1. Om Γ är satisfierbar, så är Γ konsistent.

2. Om Γ är konsistent, så är Γ satisfierbar.

Att visa att en mängd Γ är inkonsistent kan ju göras med att ange ett bevis av $\bot$ i vilket premisserna tillhör Γ. Men om man vill bevisa att Γ är konsistent, så är det mycket svårare eftersom man behöver visa att det *inte går* att göra ett formellt bevis av $\bot$ från premisser i Γ. Då har man stor nytta av sundhetssatsen och ovanstående kommentar. Man behöver bara ange en sanningsvärdestilldelning sådan att alla satser i Γ blir sanna. Då är ju Γ satisfierbar, och enligt ovan också konsistent!

1.17 Exempel. Visa att mängden $\{p \wedge \neg q,\ p \vee (q \wedge r)\}$ är konsistent.
Lösning. Vi låter p vara sann och q och r falska. Då är båda satserna i mängden sanna. Mängden är alltså satisfierbar, och det följer av sundhetssatsen att den är konsistent. $\square$

2 Predikatlogik utan kvantorer

I predikatlogik ska vi förbättra språket så att vi kan tala om element, mängder och funktioner över ett universum.

2.1 Strukturer

Relationer

Låt A vara en icke-tom mängd. En *ett-ställig relation* på A är en delmängd av A. Om $R \subseteq A$ så kan vi skriva $R(a)$ för $a \in R$. Om $\mathbf{N}$ är mängden av naturliga tal, och J de jämna naturliga talen, så är alltså J en ett-ställig relation på $\mathbf{N}$. Vi kan t ex skriva $J(4)$ för att säga att 4 är ett jämnt naturligt tal.

En *två-ställig relation* på A är en delmängd av $A \times A$, dvs en mängd av ordnade par av element från A. T ex är "Mindre än"-relationen på $\mathbf{N}$ en tvåställig relation, som vi brukar ge namnet $<$. Vi har t ex $Mindre(3, 100)$ eftersom $3 < 100$. Två-ställiga relationer kallas även för *binära relationer*. Vid binära relationer skrivs ofta, men inte alltid, relationstecknet mellan de två argumenten.

Mer generellt: En *n-ställig relation* på mängden A är en delmängd av $A^n = A \times A \times \cdots \times A$. Om R är en n-ställig relation på A, så är alltså R en mängd av ordnade n-tupler med element från A. T ex kan vi ha den tre-ställiga relationen $Mellan$ på $\mathbf{N}$ definierad av $Mellan(a, b, c)$ om och endast om a är mellan b och c på tallinjen.

Funktioner

Låt A och B vara två icke-tomma mängder. Att f är en funktion från A till B betyder att till varje $a \in A$ har tilldelats precis ett element $b = f(a) \in B$. Notation: $f : A \longrightarrow B$. Observera att vi kräver att funktioner är *totala*, dvs $f(a)$ ska vara definierad för varje element a i A, och $f(a)$ ska vara ett element i mängden B.

T ex är $f : \mathbf{N} \longrightarrow \mathbf{N}$ definierad av $f(n) = 3n+1$ en funktion. Men i sambandet $x^2+y^2 = 1$ (ekvationen för en cirkel med radien 1 och medelpunkt i origo) är inte y en funktion av x eftersom det finns flera y-värden som passar till varje x-värde $(-1 < x < 1)$.

Låt A vara en icketom mängd. En *ett-ställig funktion* på A är en funktion $f : A \longrightarrow A$. En *två-ställig funktion* på A är en funktion $f : A \times A \longrightarrow A$, dvs en funktion som tar två argument från mängden A, och svarar med ett element i A. Exempelvis $P : \mathbf{N} \times \mathbf{N} \longrightarrow \mathbf{N}$ given av $P(n, m) = n + m$. P är en tvåställig funktion, som given två naturliga tal n och m svarar med deras summa.
Vi måste vara försiktiga med mängderna, funktionens svar måste alltid vara i mängden A. T ex definierar inte regeln $M(n, m) = n - m$ någon funktion på $\mathbf{N}$. Det beror på att det finns naturliga tal n och m så att $n - m$ inte är ett naturligt tal, t ex $3 - 5 \notin \mathbf{N}$. I detta fall säger vi att mängden $\mathbf{N}$ inte är sluten under operationen M.

En *n-ställig funktion* på A är en funktion $f : A^n \longrightarrow A$, dvs f tar n stycken argument $a_1, \ldots, a_n$ och svarar med ett element $f(a_1, \ldots, a_n) \in A$.
Vi ska nu definiera vad vi menar med en *struktur*.

2.1 Definition. En *struktur* är en icketom mängd A, kallad strukturens *universum*, utrustad med ett antal relationer, funktioner och utpekade element i A, kallade *konstanter*. Man brukar skriva

$$\mathcal{A} = \langle A, R_1, \ldots R_n, f_1, \ldots f_m, a_1, \ldots a_k \rangle$$

för strukturen med universum A, relationerna $R_1, \ldots R_n$, funktionerna $f_1, \ldots f_m$ och konstanterna $a_1, \ldots a_k$. $\square$

Exempel 1
Strukturen $\mathcal{N} = \langle \mathbf{N}, <, S, 0 \rangle$. Universum är de naturliga talen. Relationen $<$ (den vanliga "mindre än"), den ett-ställiga funktionen $S(n) = n + 1$. (S för eng. successor, på svenska efterföljarfunktionen.) Konstanten 0 tillhör $\mathbf{N}$.

Exempel 2
Strukturen $\mathcal{R}_1 = \langle \mathbf{R}, +, -, \cdot, 0, 1 \rangle$ med universum de reella talen. Två-ställiga funktioner $+$ och $\cdot$, den ett-ställiga funkitonen $-$, och konstanterna 0 och 1.

Exempel 3
Strukturen $\mathcal{R}_2 = \langle \mathbf{R}^+, \cdot, \surd, 0, 1 \rangle$ med universum de icke-negativa reella talen. Två-ställiga funktionen $\cdot$, den ett-ställiga funktionen $\surd$, och konstanterna 0 och 1.

Exempel 4
Strukturen $\mathcal{Z} = \langle \mathbf{Z}, +, -, 0, 1 \rangle$ med universum alla heltal. Två-ställiga funktionen $+$, ett-ställiga funktionen $-$ (additiv invers) och konstanterna 0.

Exempel 5
Strukturen $\mathcal{Z}_5 = \langle \mathbf{Z}_5, \oplus, \ominus, 0, 1 \rangle$ med universum restklasserna modulo 5. Dvs $\mathbf{Z}_5 = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \overline{4}\}$ och den två-ställiga funktionen $\oplus$ är addition modulo 5. T ex $\overline{3} \oplus \overline{1} = \overline{4}$, $\overline{3} \oplus \overline{4} = \overline{2}$. Den ett-ställiga funktionen $\ominus$ är invers: $\ominus\overline{3} = \overline{2}$ eftersom $\overline{3} \oplus \overline{2} = \overline{0}$, och $\ominus\overline{1} = \overline{4}$.

Exempel 6
Strukturen $\mathcal{M} = \langle M, Yngre, Mor, Anna \rangle$ där M är mängden av alla människor (som lever nu eller har levt tidigare). Den två ställiga relationen $Yngre(a, b)$ om a är yngre än b. Den ett-ställiga funktionen $Mor : M \longrightarrow M$ där $Mor(a)$ är mamma till a, om $a \in M$.

Exempel 7
Strukturen $\mathcal{N}_0 = \langle \mathbf{N} \rangle$ har universum de naturliga talen. Strukturen har ingen relation, funktion eller konstant.

2.2 Språk

Vi ska använda predikatlogik för att resonera i strukturer som ovan. Beroende på vilken sorts struktur man är intreserad av, behöver man olika sorters språk. För varje relation R behöver vi en relationssymbol $\mathbf{R}$, och för varje funktion f behöver vi en funktionssymbol $\mathbf{f}$. För varje konstant a behöver vi en konstantsymbol $\mathbf{a}$. Man brukar räkna upp symbolerna i språket i samma ordning som motsvarande relation anges i strukturen, därför anges språket som en ordnad följd

$$\langle \mathbf{R_1}, \ldots, \mathbf{R_n}, \mathbf{f_1}, \ldots, \mathbf{f_m}, \mathbf{a_1}, \ldots, \mathbf{a_k} \rangle$$

av symboler. Vi tar som definition att alla språk innehåller en symbol $\doteq$ för likhet $=$. Symbolen är en två-ställig relationssymbol och finns i alla språk, men vi skriver inte upp den i listan.

Exempel För strukturen $\mathcal{N}$ i Exempel 1 ovan kan vi använda språket

$$L_1 = \langle \mathbf{M}, \mathbf{S}, \mathbf{0} \rangle$$

där $\mathbf{M}$ är en 2-ställig relationssymbol, $\mathbf{S}$ är en 1-ställig funktionssymbol och $\mathbf{0}$ är en konstantsymbol. Samma språk fungerar bra också till strukturen i $\mathcal{M}$ i Exempel 6, eftersom det består av en två-ställig relationssymbol, en ett-ställig funktionssymbol och en konstantsymbol.

Exempel För strukturerna $\mathcal{Z}$ och $\mathcal{Z}_5$ i Exempel 4 och 5 ovan kan vi använda språket

$$L_2 = \langle \mathbf{P}, \mathbf{Inv}, \mathbf{0}, \mathbf{1} \rangle.$$

Exempel För strukturen $\mathcal{N}_0$ i Exempel 7 har vi inga extra symboler; den enda relationssymbolen är likhetsteknet $\doteq$.

$$L_7 = \langle \qquad \rangle.$$

2.2 Definition. Låt L vara ett språk. Mängden av *slutna termer* i språket definieras induktivt av klausulerna

(i) Om $\mathbf{a}$ är en konstantsymbol, så är $\mathbf{a}$ en sluten term.

(ii) Om $\mathbf{f}$ är en n-ställig funktionssymbol och $t_1, \ldots, t_n$ är slutna termer, så är $\mathbf{f}(t_1, \ldots, t_n)$ en sluten term.

□

Slutna termer i språket L_1 är t ex $\mathbf{0}$, $\mathbf{S}(\mathbf{0})$ och $\mathbf{S}(\mathbf{S}(\mathbf{S}(\mathbf{0})))$.

I språket L_2 har vi t ex de slutna termerna $\mathbf{0}$, $\mathbf{P}(\mathbf{0}, \mathbf{1})$, $\mathbf{Inv}(\mathbf{1})$ och $\mathbf{P}(\mathbf{Inv}(\mathbf{1}), \mathbf{1})$.

Språket L_7 saknar slutna termer eftersom det inte finns någon konstantsymbol.

2.3 Definition. Låt L vara ett språk. Mängden av *slutna atomära formler* i språket definieras induktivt av klausulerna

(i) Om t_1 och t_2 är slutna termer, så är $t_1 \doteq t_2$ en sluten atomär formel.

(ii) Om $\mathbf{R}$ är en n-ställig relationssymbol och $t_1, \ldots, t_n$ är slutna termer, så är $\mathbf{R}(t_1, \ldots, t_n)$ en sluten atomär formel.

(iii) $\perp$ är en sluten atomär formel.

□

Några slutna atomära formler i språket L_1 ovan är $\mathbf{0} \doteq \mathbf{S}(\mathbf{0})$ och $\mathbf{M}(\mathbf{0}, \mathbf{S}(\mathbf{0}))$ och $\mathbf{M}(\mathbf{S}(\mathbf{S}(\mathbf{S}(\mathbf{0}))), \mathbf{S}(\mathbf{0}))$.

Några slutna atomära formler i språket L_2 är $\mathbf{0} \doteq \mathbf{P}(\mathbf{0}, \mathbf{1})$ och $\mathbf{Inv}(\mathbf{1}) \doteq \mathbf{P}(\mathbf{Inv}(\mathbf{1}), \mathbf{1})$.

I språket L_7 är $\perp$ den enda slutna atomära formeln.

Anm: I bokens första del gör författarna ”logik utan kvantorer”. I definition av PROP (Definition 1.1), om man byter ut satssymbolerna $p_0, p_1, p_2, \ldots$ mot de atomära slutna satserna, så får vi det språk som boken använder.

2.3 Semantik

Betrakta ett första ordningens språk

$$L = \langle \mathbf{R_1}, \ldots, \mathbf{R_n}, \mathbf{f_1}, \ldots, \mathbf{f_m}, \mathbf{a_1}, \ldots, \mathbf{a_k} \rangle$$

och en struktur

$$\mathcal{A} = \langle A, R_1, \ldots R_n, f_1, \ldots f_m, a_1, \ldots a_k \rangle$$

som passar för språket L. Strukturen kallas då för en L-struktur.

Vi ska ge de slutna termerna och de slutna formlerna en tolknig i strukturen $\mathcal{A}$. Varje sluten term t ska tolkas som ett element $t^{\mathcal{A}}$ i strukturens universum A.

Varje sluten atomär sats σ ska ges ett sanningsvärde i $\mathcal{A}$. Vi kommer att skriva $\mathcal{A} \models \sigma$ för att uttrycka att σ är sann i strukturen $\mathcal{A}$. Om σ är falsk i strukturen $\mathcal{A}$ skriver vi $\mathcal{A} \not\models \sigma$. Ibland utläses $\mathcal{A} \models \sigma$ som ”$\mathcal{A}$ är en modell för σ”, vilket alltså betyder att σ är sann i $\mathcal{A}$.

2.4 Definition. (*Tolkning av slutna termer.*)
Låt $\mathcal{A}$ vara en struktur som passar för språket L.

(1) Om $\mathbf{a}$ är en konstantsymbol, med motsvarande konstant $a \in A$, så låt

$$\mathbf{a}^{\mathcal{A}} = a.$$

(2) Om $\mathbf{f}$ är en n-ställig funktionssymbol och den motsvarande funktionen i strukturen är $f : A^n \longrightarrow A$, och $t_1, \ldots, t_n$ är slutna termer, så låt

$$\big(\mathbf{f}\,(t_1, \ldots, t_n)\big)^{\mathbf{A}} = f\big(t_1^{\mathbf{A}}, \ldots, t_n^{\mathbf{A}}\big).$$

□

Exempel. Vi tolkar några slutna termer i språket L_1, dels i strukturen $\mathcal{N}$ från Exempel 1, dels i strukturen $\mathcal{M}$ från Exempel 6.

$$\begin{array}{rcl} (\mathbf{0})^{\mathcal{N}} = 0 & \text{och} & (\mathbf{0})^{\mathcal{M}} = \mathit{Anna} \\ (\mathbf{S}(\mathbf{0}))^{\mathcal{N}} = 1 & \text{och} & (\mathbf{S}(\mathbf{0}))^{\mathcal{M}} = \mathit{Annas}\ \text{mamma} \\ (\mathbf{S}(\mathbf{S}(\mathbf{0})))^{\mathcal{N}} = 2 & \text{och} & (\mathbf{S}(\mathbf{S}(\mathbf{0})))^{\mathcal{M}} = \mathit{Annas}\ \text{mormor} \end{array}$$

Exempel. Vi tolkar några slutna termer i språket L_2, i strukturerna $\mathcal{Z}$ och $\mathcal{Z}_5$ från Exempel 4 och 5.

$$\begin{array}{rcl} (\mathbf{0})^{\mathcal{Z}} = 0 & \text{och} & (\mathbf{0})^{\mathcal{Z}_5} = \overline{0} \\ (\mathbf{P}(\mathbf{1}, \mathbf{1}))^{\mathcal{Z}} = 1 + 1 = 2 & \text{och} & (\mathbf{P}(\mathbf{1}, \mathbf{1}))^{\mathcal{Z}_5} = \overline{1} \oplus \overline{1} = \overline{2} \\ (\mathbf{Inv}(\mathbf{1}))^{\mathcal{Z}} = -1 & \text{och} & (\mathbf{Inv}(\mathbf{1}))^{\mathcal{Z}_5} = \ominus\overline{1} = \overline{4} \\ (\mathbf{P}(\mathbf{Inv}(\mathbf{1}), \mathbf{1}))^{\mathcal{Z}} = -1 + 1 = 0 & \text{och} & (\mathbf{P}(\mathbf{Inv}(\mathbf{1}), \mathbf{1}))^{\mathcal{Z}_5} = \ominus\overline{1} \oplus \overline{1} = \overline{4} \oplus \overline{1} = \overline{0} \end{array}$$

2.5 Definition. (*Tolkning av slutna atomära formler.*)
Låt $\mathcal{A}$ vara en struktur som passar för språket L.

(i) Om t_1 och t_2 är slutna termer, så

$$\mathcal{A} \models t_1 \doteq t_2 \Longleftrightarrow t_1^{\mathcal{A}} = t_2^{\mathcal{A}}.$$

(ii) Om $\mathbf{R}$ är en n-ställig relationssymbol, som i strukturen tolkas som den n-ställiga relationen $R \subseteq A^n$, och $t_1, \ldots, t_n$ är slutna termer, så

$$\mathcal{A} \models \mathbf{R}\,(t_1, \ldots, t_n) \Longleftrightarrow R\big(t_1^{\mathcal{A}}, \ldots, t_n^{\mathcal{A}}\big).$$

(iii) $\mathcal{A} \not\models \perp$. $\square$

Exempel. Med språket L_1 och strukturerna $\mathcal{N}$ och $\mathcal{M}$ från Exempel 1 och 6. Vi har

(i) $\mathcal{N} \not\models \mathbf{0} \doteq \mathbf{S}(\mathbf{0})$ eftersom $0 \neq 1$.
$\mathcal{M} \not\models \mathbf{0} \doteq \mathbf{S}(\mathbf{0})$ eftersom *Anna* och *Annas* mamma inte är samma person.

(ii) $\mathcal{N} \models \mathbf{M}\,(\mathbf{0}, \mathbf{S}(\mathbf{0}))$ eftersom $0 < 1$.
$\mathcal{M} \not\models \mathbf{M}\,(\mathbf{0}, \mathbf{S}(\mathbf{0}))$ eftersom *Anna* inte är äldre än sin mamma.

(iii) $\mathcal{N} \not\models \mathbf{M}\,(\mathbf{S}(\mathbf{S}(\mathbf{S}(\mathbf{0}))), \mathbf{S}(\mathbf{0}))$ eftersom utsagan "$3 < 1$" är falsk.
$\mathcal{M} \models \mathbf{M}\,(\mathbf{S}(\mathbf{S}(\mathbf{S}(\mathbf{0}))), \mathbf{S}(\mathbf{0}))$ eftersom *Annas* gammelmormor verkligen är äldre än *Annas* mamma.

Exempel. Med språket L_2 och strukturerna $\mathcal{Z}$ och $\mathcal{Z}_5$ från Exempel 4 och 5 har vi:

(i) $\mathcal{Z} \models \mathbf{1} \doteq \mathbf{P}(\mathbf{0}, \mathbf{1})$ eftersom $1 = 0 + 1$.
$\mathcal{Z}_5 \models \mathbf{1} \doteq \mathbf{P}(\mathbf{0}, \mathbf{1})$ eftersom $\overline{1} = \overline{0} \oplus \overline{1}$.

(ii) $\mathcal{Z} \not\models \mathbf{Inv}(\mathbf{1}) \doteq \mathbf{P}(\mathbf{Inv}(\mathbf{1}), \mathbf{1})$ eftersom $-1 \neq 0$.
$\mathcal{Z}_5 \not\models \mathbf{Inv}(\mathbf{1}) \doteq \mathbf{P}(\mathbf{Inv}(\mathbf{1}), \mathbf{1})$ eftersom $\overline{4} \neq \overline{0}$.

(iii) $\mathcal{Z} \not\models \mathbf{P}\,(\mathbf{P}(\mathbf{1}, \mathbf{1}), \mathbf{P}(\mathbf{1}, \mathbf{1})) \doteq \mathbf{Inv}(\mathbf{1})$ eftersom $2 + 2 = -1$ är falskt.
Men $\mathcal{Z}_5 \models \mathbf{P}\,(\mathbf{P}(\mathbf{1}, \mathbf{1}), \mathbf{P}(\mathbf{1}, \mathbf{1})) \doteq \mathbf{Inv}(\mathbf{1})$ eftersom $\overline{2} \oplus \overline{2} = \overline{4} = \ominus\overline{1}$.

Alla slutna formler som fås genom att tillämpa konnektiverna på mängden av slutna atomära formler kan nu ges sanningsvärde i strukturen $\mathcal{A}$ på samma sätt som vi gjorde i PROP, dvs med sanningsvärdestabeller, där vi hanterar de atomära slutna formlerna på samma sätt som satssymbolerna. T ex har vi $\mathcal{N} \models \mathbf{M}\,(\mathbf{0}, \mathbf{S}(\mathbf{0})) \vee \mathbf{M}\,(\mathbf{S}(\mathbf{0}), \mathbf{0})$ om och endast $\mathcal{N} \models \mathbf{M}\,(\mathbf{0}, \mathbf{S}(\mathbf{0}))$ eller $\mathcal{N} \models \mathbf{M}\,(\mathbf{S}(\mathbf{0}), \mathbf{0})$, dvs om "$0 < 1$ eller $1 < 0$" gäller. Eftersom detta gäller har vi alltså visat $\mathcal{N} \models \mathbf{M}\,(\mathbf{0}, \mathbf{S}(\mathbf{0})) \vee \mathbf{M}\,(\mathbf{S}(\mathbf{0}), \mathbf{0})$.

2.6 Definition. En sluten formel σ utan kvantorer kallas *tautologi* om den har värdet sann i alla rader i sanningsvärdestabellen. Notation: $\models_T \sigma$. $\square$

Anm: Notationen $\models_T$ används för att betona att man bara använder ren satslogik; man bortser från ev information som kanske är inrymd i symbolerna i vårt rikare språk. När man ska undersöka $\models_T \sigma$ så ersätter varje förekomst av atomär sluten formel med en satssymbol, och ställer sedan upp sanningsvärdestabellen.

2.7 Definition. En sluten formel utan kvantorer kallas *logiskt sann* om den är sann i alla strukturer som har samma typ som språket. Notation: $\models \sigma$. $\square$

Alltså gäller att $\models \sigma$ om och endast om $\mathcal{A} \models \sigma$ för varje L-struktur $\mathcal{A}$.

Exempelvis är den slutna formeln $\mathbf{M}(\mathbf{0}, \mathbf{S}(\mathbf{0})) \vee \neg \mathbf{M}(\mathbf{0}, \mathbf{S}(\mathbf{0}))$ en tautologi eftersom den har samma sanningsvärdestabell som satsen $p_0 \vee \neg p_0$. Den är också en logisk sanning. Däremot är satsen $\mathbf{M}(\mathbf{0}, \mathbf{S}(\mathbf{0})) \vee \mathbf{M}(\mathbf{S}(\mathbf{0}), \mathbf{0})$ inte en tautologi; den får samma sanningsvärdestabell som satsen $p_0 \vee p_1$.

Observera att varje tautologi är en logisk sanning. Men det finns logiska sanningar som *inte* är tautologier. Formeln $\mathbf{a} \doteq \mathbf{a}$ är ett exempel på det. Den är sann i alla strukturer, eftersom $a = a$ gäller för alla element a i ett universum A. Men den är inte en tautologi.

Vi ska nu betrakta begreppen *tautolog konsekvens* och *logisk konsekvens.* Vi repeterar begreppet *tautolog konsekvens* från den rena satslogiken.

2.8 Definition. *Tautolog konsekvens.*
Satsen σ är en *tautolog konsekvens* av satserna $\psi_1, \ldots \psi_n$ om σ är sann på varje rad i sanningsvärdestabellen där alla satser $\psi_1, \ldots \psi_n$ är sanna. Notation: $\psi_1, \ldots \psi_n \models_T \sigma$. □

2.9 Definition. *Logisk konsekvens.*
Satsen σ är en *logisk konsekvens* av satserna $\psi_1, \ldots \psi_n$ om σ är sann i varje struktur (av rätt typ) som gör alla satserna $\psi_1, \ldots \psi_n$ sanna. Notation: $\psi_1, \ldots \psi_n \models \sigma$. □

Man kan alltså uttrycka $\psi_1, \ldots \psi_n \models \sigma$ som att σ är sann i varje modell för satserna $\psi_1, \ldots \psi_n$.

Observera att varje tautolog konsekvens också är en logisk konsekvens. *Men* det finns logiska konsekvenser som inte är tautologa: Låt ψ_1, ψ_2 och σ vara satserna $\mathbf{a} \doteq \mathbf{b}$, $\mathbf{b} \doteq \mathbf{c}$ och $\mathbf{a} \doteq \mathbf{c}$ respektive. Då är σ en logisk konsekvens av ψ_1, ψ_2, dvs $\psi_1, \psi_2 \models \sigma$, eftersom $\doteq$ alltid tolkas som likhet (som är en transitiv relation). Men σ är inte en tautolog konsekvens av ψ_1, ψ_2, dvs $\psi_1, \psi_2 \not\models_T \sigma$ eftersom om vi ersätter atomerna med satssymboler så har vi att avgöra huruvida $p_1, p_2 \models_T p_3$. När satssymbolerna p_1 och p_2 är sanna och symbolen p_3 falsk så har vi ett motexempel. Alltså $\psi_1, \psi_2 \not\models_T \sigma$.

2.4 Syntaxen - de formella bevisen

Utöver de regler vi använde för att göra formella bevis i satslogik tillkommer regler för likhetstecknet, = **Elim** och = **Intro**. Läs om dem i kapitel 2.

3 Predikatlogik (med kvantorer)

3.1 Språket

Vi bygger på det språk som introducerades i avsnitt 2.2 och får ett så kallat första ordningens språk, *First order language, FOL.*
Vi lägger till variabler $x_1, x_2, \ldots$ och kvantorer $\forall$ och $\exists$ till vårt alfabet. Vi har som vanligt likhetstecken $\doteq$ och alla konnektiver samt eventuellt s.k. ickelogiska symboler. Vi skriver fortfarande

$$\mathcal{L} = \langle \mathbf{R_1}, \ldots, \mathbf{R_n}, \mathbf{f_1}, \ldots, \mathbf{f_m}, \mathbf{a_1}, \ldots, \mathbf{a_k} \rangle$$

för vårt språk.

3.1 Definition. *Termer.*
Mängden av *termer* definieras induktivt genom följande klausuler.

(1a) Om x är en variabel, så är x en term.

(1b) Om $\mathbf{c}$ är en konstantsymbol, så är $\mathbf{c}$ en term.

(2) Om $\mathbf{f}$ är en n-ställig funktionssymbol, och $t_1, \ldots, t_n$ är termer, så är $\mathbf{f}(t_1, \ldots, t_n)$ en term.

$\square$

3.2 Definition. *Formler (Wffs).*
Mängden av *formler* i ett språk $\mathcal{L}$ definieras induktivt genom följande klausuler.

(1a) Om t och s är termer, så är $t \doteq s$ en formel.

(1b) Om $\mathbf{R}$ är en n-ställig relationssymbol, och $t_1, \ldots, t_n$ är termer, så är $\mathbf{R}(t_1, \ldots, t_n)$ en formel.

(1c) $\perp$ är en formel.

(2a) Om φ och ψ är formler, så är $(\neg\varphi)$, $(\varphi \vee \psi)$, $(\varphi \wedge \psi)$, $(\varphi \longrightarrow \psi)$ och $(\varphi \longleftrightarrow \psi)$ formler.

(2b) Om φ är en formel och x en variabel, så är $\forall x\varphi$ och $\exists x\varphi$ formler.

$\square$

3.3 Exempel. Låt $\mathcal{L} = \langle \mathbf{P}, \mathbf{E}, \mathbf{S}, \mathbf{c} \rangle$, där $\mathbf{P}$ är en 2-ställig relationssymbol, $\mathbf{E}$ en 1-ställig relationssymbol, $\mathbf{S}$ en 1-ställig funktionssymbol och $\mathbf{c}$ en konstantsymbol.
Några termer: x, $\mathbf{c}$, $\mathbf{S}(x)$, $\mathbf{S}(\mathbf{c})$, $\mathbf{S}(\mathbf{S}(x))$, $\mathbf{S}(\mathbf{S}(\mathbf{c}))$.
Några formler:

- $x \doteq \mathbf{S}(x)$
- $\mathbf{c} \doteq \mathbf{S}(x)$
- $\mathbf{P}\left(\mathbf{S}(x), \mathbf{S}(\mathbf{S}(\mathbf{c}))\right)$
- $\mathbf{E}(\mathbf{S}(x))$
- $\exists x\,(x \doteq \mathbf{S}(\mathbf{c}))$
- $\forall x\,(\mathbf{P}(x, \mathbf{S}(x)) \wedge \mathbf{E}(\mathbf{S}(\mathbf{S}(\mathbf{c}))))$
- $\exists x \forall y\, \mathbf{P}(x, y)$
- $\forall x \exists y\, \mathbf{P}(x, y)$
- $\forall x \mathbf{E}(\mathbf{S}(x)) \vee x \doteq \mathbf{S}(\mathbf{c})$

$\square$

Fria och bundna förekomster av variabler.

Ingen variabel i en term förekommer bundet.
Variabeln x förekommer *bundet* i formler med utseendet

$$\forall x\,(\ldots x \ldots) \text{ eller } \exists x\,(\ldots x \ldots).$$

Om x förekommer, men inte tillsammans med en kvantor, så kallas förekomsten *fri*. Några exempel från språket $\mathcal{L} = \langle \mathbf{P}, \mathbf{E}, \mathbf{S}, \mathbf{c} \rangle$ ovan:

(1) Variabeln x förekommer fritt i formeln $\mathbf{c} \doteq \mathbf{S}(x)$.

(2) Variabeln x förekommer bundet i formeln $\exists x\,(\mathbf{c} \doteq \mathbf{S}(x))$.

(3) Variabeln x förekommer bundet och variabeln y förekommer fritt i formeln $\forall x \mathbf{E}(\mathbf{S}(x)) \vee y \doteq \mathbf{S}(\mathbf{c})$.

(4) Variabeln y förekommer både bundet och fritt i formeln $\forall y \mathbf{E}(\mathbf{S}(y)) \vee y \doteq \mathbf{S}(\mathbf{c})$.

3.4 Definition. En *sluten formel* är en formel som saknar fria variabelförekomster. □

Anm: Ibland används ordet *sats* som synonym till ”sluten formel”.
Anm: Först i kursen definierade vi endast slutna formler, vi hade inga variabler eller kvantorer i språket. Observera att dessa formler som då kallades slutna, verkligen är slutna även med den här nya definitionen.
Exempel på satser (slutna formler) i språket $\mathcal{L} = \langle \mathbf{P}, \mathbf{E}, \mathbf{S}, \mathbf{c} \rangle$ är $\forall y \mathbf{E}(\mathbf{S}(y))$ och $\mathbf{P}(\mathbf{c}, \mathbf{S}(\mathbf{c}))$.

3.2 Semantiken

Tolkning av slutna formler i en struktur av rätt typ.

Låt $\mathcal{L}$ vara ett FOL, och låt $\mathcal{A}$ vara en struktur som passar för språket $\mathcal{L}$. Vi vill nu definiera för en godtycklig sats σ i språket vad det innebär att den är sann i $\mathcal{A}$, dvs vi vill definiera vad $\mathcal{A} \models \sigma$ betyder. Om σ inte innehåller någon kvantor, så har vi redan definierat detta i avsnitt 2. För att klara av att tolka satser med kvantorer så behöver vi utvidga språket så att vi får möjlighet att tala om varje individ i strukturens universum. Låt A vara universum i strukturen $\mathcal{A}$. Vi definierar *det utvidgade språket* $\mathcal{L}(\mathcal{A})$ genom att till $\mathcal{L}$ lägga *nya konstantsymboler*, en ny konstantsymbol för varje element i universum A:

$$\mathcal{L}(\mathcal{A}) = \mathcal{L} \cup \{\overline{a} : \ a \in A\}.$$

Konstantsymbolen $\overline{a}$ kallas för *namn* på elementet $a \in A$.

3.5 Exempel. Låt $\mathcal{L} = \langle \mathbf{P}, \mathbf{E}, \mathbf{S}, \mathbf{c} \rangle$, där $\mathbf{P}$ är en 2-ställig relationssymbol, $\mathbf{E}$ en 1-ställig relationssymbol, $\mathbf{S}$ en 1-ställig funktionssymbol och $\mathbf{c}$ en konstantsymbol.
Betrakta strukturen $\mathcal{N} = \langle \mathbf{N}, <, J, S, 3 \rangle$ med de naturliga talen som universum, mindre än-relationen, $J(n)$ om och endast om n är jämn, funktionen $S(n) = n+1$ och konstanten 3. Det utvidgade språket är

$$\begin{aligned} \mathcal{L}(\mathcal{N}) &= \{\mathbf{P},\ \mathbf{E},\ \mathbf{S},\ \mathbf{c}\} \cup \{\overline{n} : \ n \in \mathbf{N}\} \\ &= \{\mathbf{P},\ \mathbf{E},\ \mathbf{S},\ \mathbf{c},\ \overline{0},\ \overline{1},\ \overline{2},\ \overline{3},\ \overline{4},\ \ldots\}. \end{aligned}$$

□

I Definition 2.4 står hur slutna termer i språket $\mathcal{L}$ tolkas som element i strukturen $\mathcal{A}$. För att tolka de slutna termerna i språket $\mathcal{L}(\mathcal{A})$, så behöver vi till den definitionen lägga till hur de nya konstantsymbolerna $\overline{a}$, för $a \in A$, ska tolkas. Vi får då följande definition.

3.6 Definition. (*Tolkning av slutna termer i det utvidgade språket.*)
Låt $\mathcal{A}$ vara en $\mathcal{L}$-struktur, med mängden A som universum.

(1a) Om $\mathbf{a}$ är en konstantsymbol, med motsvarande konstant $a \in A$, så låt

$$\mathbf{a}^{\mathcal{A}} = a.$$

(1b) Om $a \in A$, med namnet $\overline{a}$ i det utvidgade språket, så

$$(\overline{a})^{\mathcal{A}} = a.$$

(2) Om $\mathbf{f}$ är en n-ställig funktionssymbol och den motsvarande funktionen i strukturen är $f : A^n \longrightarrow A$, och $t_1, \ldots, t_n$ är slutna termer, så låt

$$\big(\mathbf{f}\,(t_1, \ldots, t_n)\big)^{\mathbf{A}} = f\big(t_1^{\mathbf{A}}, \ldots, t_n^{\mathbf{A}}\big).$$

□

Vi ska nu tolka *de slutna formlerna*, dvs satserna, i språket $\mathcal{L}(\mathcal{A})$.

3.7 Definition. Låt $\mathcal{L}$ vara ett FOL, och låt $\mathcal{A}$ vara en struktur som passar för språket $\mathcal{L}$. Låt σ en sluten formel i det utvidgade språket $\mathcal{L}(\mathcal{A})$. Vi definierar $\mathcal{A} \models \sigma$ med induktion på satsen σ.

(1a) $\mathcal{A} \models t \doteq s$ omm $t^{\mathcal{A}} = s^{\mathcal{A}}$, om t och s är slutna termer.

(1b) Om $\mathbf{R}$ är en n-ställig relationssymbol som svarar mot den n-ställiga relationen R, och $t_1, \ldots, t_n$ är slutna termer, så $\mathcal{A} \models \mathbf{R}(t_1, \ldots, t_n)$ omm $R\left(t_1^{\mathcal{A}}, \ldots, t_n^{\mathcal{A}}\right)$.

(1c) $\mathcal{A} \not\models \perp$.

(2a) Om φ och ψ är slutna formler, så

(i) $\mathcal{A} \models \neg\varphi$ omm $\mathcal{A} \not\models \varphi$
(ii) $\mathcal{A} \models \varphi \vee \psi$ omm $\mathcal{A} \models \varphi$ eller $\mathcal{A} \models \psi$
(iii) $\mathcal{A} \models \varphi \wedge \psi$ omm $\mathcal{A} \models \varphi$ och $\mathcal{A} \models \psi$
(iii) $\mathcal{A} \models \varphi \longrightarrow \psi$ omm (om $\mathcal{A} \models \varphi$, så $\mathcal{A} \models \psi$)
(iv) $\mathcal{A} \models \varphi \longleftrightarrow \psi$ omm ($\mathcal{A} \models \varphi$ omm $\mathcal{A} \models \psi$)

(2b) Anta att σ är den slutna formeln $\forall x \varphi(x)$ (eller den slutna formeln $\exists x \varphi(x)$). Då är x den enda variabeln som ev förekommer fritt i formeln φ. Om man substituerar in en konstantsymbol $\overline{a}$ för varje förekomst av x i φ, så blir resultatet $\varphi(\overline{a})$ en sluten formel, som är mindre komplex än σ (eftersom den har en kvantor mindre). Induktivt vet vi därför vad $\mathcal{A} \models \varphi(\overline{a})$ betyder. Vi definierar:

$$\mathcal{A} \models \forall x \varphi \quad \text{omm} \quad \mathcal{A} \models \varphi(\overline{a}) \text{ för varje element } a \in A$$

och

$$\mathcal{A} \models \exists x \varphi \quad \text{omm} \quad \mathcal{A} \models \varphi(\overline{a}) \text{ för minst ett element } a \in A.$$

□

Observera att vi endast har definierat $\mathcal{A} \models \sigma$ för *slutna* formler.

3.8 Exempel. Låt $\mathcal{L} = \langle \mathbf{P}, \mathbf{E}, \mathbf{S}, \mathbf{c} \rangle$, där $\mathbf{P}$ är en 2-ställig relationssymbol, $\mathbf{E}$ en 1-ställig relationssymbol, $\mathbf{S}$ en 1-ställig funktionssymbol och $\mathbf{c}$ en konstantsymbol.
Betrakta strukturen $\mathcal{N} = \langle \mathbf{N}, <, J, S, 3 \rangle$ med de naturliga talen som universum, mindre än-relationen, $J(n)$ om och endast om n är jämn, funktionen $S(n) = n+1$ och konstanten 3. Det utvidgade språket är

$$\begin{aligned} \mathcal{L}(\mathcal{N}) &= \{\mathbf{P},\ \mathbf{E},\ \mathbf{S},\ \mathbf{c}\} \cup \{\overline{n} :\ n \in \mathbf{N}\} \\ &= \{\mathbf{P},\ \mathbf{E},\ \mathbf{S},\ \mathbf{c},\ \overline{0},\ \overline{1},\ \overline{2},\ \overline{3},\ \overline{4},\ \ldots\}. \end{aligned}$$

Observera att det kan finnas flera slutna termer som tolkas sam samma tal, exempelvis tolkas både $\mathbf{c}$ och $\overline{3}$ på talet 3. Observera också att om vi inte hade utvidgat språket, så hade vi i detta fall ingen sluten term som tolkades som 0 (inte heller 1 eller 2). Vi undersöker om $\mathcal{N} \models \sigma$ för några satser σ.

1. Låt σ_1 vara satsen $\exists x\,(\mathbf{P}(\mathbf{c}, x) \wedge \mathbf{E}(x))$. Vi har
   $\mathcal{N} \models \exists x\,(\mathbf{P}(\mathbf{c}, x) \wedge \mathbf{E}(x))$
   omm det finns $n \in \mathbf{N}$ så att $\mathcal{N} \models \mathbf{P}(\mathbf{c}, \overline{n}) \wedge \mathbf{E}(\overline{n})$
   omm det finns $n \in \mathbf{N}$ så att $\big(\ \mathcal{N} \models \mathbf{P}(\mathbf{c}, \overline{n})$ och $\mathcal{N} \models \mathbf{E}(\overline{n})\ \big)$
   omm det finns $n \in \mathbf{N}$ så att $3 < n$ och n är jämn.
   Eftersom det finns naturliga tal som är jämna och större än 3, så har vi har alltså visat att $\mathcal{N} \models \sigma_1$

2. Låt σ_2 vara satsen $\forall x\, \exists y\, \mathbf{P}(x, y)$. Då gäller att
   $\mathcal{N} \models \sigma_2$ omm för varje naturligt tal n vi har $\mathcal{N} \models \exists y\, \mathbf{P}(\overline{n}, y)$
   omm för varje naturligt tal n det finns ett naturligt tal m så att $\mathcal{N} \models \mathbf{P}(\overline{n}, \overline{m})$
   omm det för varje naturligt tal n finns ett naturligt tal m så att $n < m$.
   Eftersom det till varje naturligt tal finns ett större naturligt tal, så är även denna sats sann i $\mathcal{N}$, dvs $\mathcal{N} \models \sigma_2$.

3. Låt σ_3 vara satsen $\exists x\, \forall y\ (\mathbf{P}(x, y) \vee x \doteq y)$. Då gäller att
   $\mathcal{N} \models \sigma_3$ omm det finns något naturligt tal n så att $\mathcal{N} \models \forall y\ (\mathbf{P}(\overline{n}, y) \vee \overline{n} \doteq y)$
   omm det finns något naturligt tal n sådant att för varje naturligt tal m gäller $\mathcal{N} \models \mathbf{P}(\overline{n}, \overline{m}) \vee \overline{n} \doteq \overline{m}$
   omm det finns något naturligt tal n sådant att för varje naturligt tal m gäller att $n < m$ eller $n = m$
   omm det finns något naturligt tal n sådant att för varje naturligt tal m gäller att $n \leq m$.
   Eftersom det finns ett minsta naturligt tal (0), så är satsen σ_3 sann i $\mathcal{N}$, dvs $\mathcal{N} \models \sigma_3$.

4. Låt σ_4 vara satsen $\forall x\, \forall y\, (\mathbf{P}(x, y) \longrightarrow \exists z\, (\mathbf{P}(x, z) \wedge \mathbf{P}(z, y)))$. Då gäller
   $\mathcal{N} \models \sigma_4$ omm för alla naturliga tal n och m det gäller att om $n < m$, så finns naturligt tal k så att $n < k$ och $k < m$.
   Vi ser att om t ex $n = 4$ och $m = 5$ så gäller $n < m$, men det finns inget naturligt tal k så att $4 < k$ och $k < 5$. Alltså är σ_4 falsk i $\mathcal{N}$, dvs $\mathcal{N} \not\models \sigma_4$.

Låt $\mathcal{Q}$ och $\mathcal{Z}$ vara som $\mathcal{N}$ men med andra universa. Universum i $\mathcal{Q}$ är de rationella talen $\mathbf{Q}$, och universum i $\mathcal{Z}$ är heltalen $\mathbf{Z}$. Symbolerna $\mathbf{P},\ \mathbf{E},\ \mathbf{S},\ \mathbf{c}$ tolkas på samma sätt som i $\mathcal{N}$, dvs $\mathbf{P}$ tolkas som $<$, $\mathbf{E}$ tolkas som mängden J av jämna heltal, $\mathbf{S}$ tolkas som funktionen S som adderar 1, och $\mathbf{c}$ tolkas som konstanten 3. Vi undersöker satserna σ_1, σ_2, σ_3 och σ_4 i dessa strukturer.

1. Vi har både $\mathcal{Q} \models \sigma_1$ och $\mathcal{Z} \models \sigma_1$. (Eftersom det finns jämna heltal som är större än 3 i både $\mathbf{Q}$ och $\mathbf{Z}$.)

2. Eftersom ingen av våra strukturer har ett största element, så gäller även $\mathcal{Q} \models \sigma_2$ och $\mathcal{Z} \models \sigma_2$.

3. Varken $\mathcal{Q}$ eller $\mathcal{Z}$ har ett minsta element. Därför är σ_3 falsk i båda strukturerna: $\mathcal{Q} \not\models \sigma_3$ och $\mathcal{Z} \not\models \sigma_3$.

4. Satsen σ_4 är falsk i $\mathcal{Z}$, eftersom det t ex inte finns något heltal mellan 4 och 5. Satsen σ_4 är däremot sann i $\mathcal{Q}$ eftersom det mellan två olika rationella tal alltid finns ett tredje rationellt tal. (Om $p = \frac{a}{b}$ och $q = \frac{c}{d}$ är två rationella tal, och $p < q$, så får vi $p < r < q$ om vi t ex tar $r = \frac{p+q}{2}$.) Vi har alltså att $\mathcal{Q} \models \sigma_4$ och $\mathcal{Z} \not\models \sigma_4$.

5. Låt $\sigma_5 = \neg\sigma_4$. Då har vi i stället $\mathcal{Q} \not\models \sigma_5$ och $\mathcal{Z} \models \sigma_5$.

$\square$

Logisk konsekvens och logisk sanning

Låt $\mathcal{L}$ vara ett FOL, och låt $\mathcal{A}$ vara en struktur som passar till $\mathcal{L}$. Strukturen $\mathcal{A}$ kallas då för en *$\mathcal{L}$-struktur*. Låt σ vara en sats i språket $\mathcal{L}$, och låt Γ vara en mängd av satser. Vi säger att $\mathcal{A}$ är en *modell* för Γ om $\mathcal{A} \models \gamma$ för alla $\gamma \in \Gamma$. Om Γ bara innehåller en sats, $\Gamma = \{\gamma\}$, så säger vi att $\mathcal{A}$ är en modell för γ.

Vi har redan givit begreppen logisk konsekvens och logisk sanning, men vi repeterar dem nu, när vi har hela FOL.

3.9 Definition. *Logisk sanning.*
Låt $\mathcal{L}$ vara ett FOL, och låt σ vara en sats i språket $\mathcal{L}$. Satsen σ är en *logisk sanning* om σ är sann i varje struktur (som passar språket). Notation: $\models \sigma$. $\square$

Anm: Att σ är en logisk sanning kallas också synonymt för att σ är *valid*. De valida satserna är alltså de satser som alltid är sanna, dvs de är sanna oberoende av hur vi tolkar symbolerna i språket och vilket universum som väljs. Exempel på valida satser i språket $\mathcal{L} = \langle \mathbf{P}, \mathbf{E}, \mathbf{S}, \mathbf{c} \rangle$ (som användes i flera tidigare exempel):

1. $\models \exists x\, (x \doteq x)$, dvs $\exists x\, (x \doteq x)$ är valid, eftersom vi har krävt att alla strukturer ska ha ett universum som innehåller minst ett element.

2. $\models \forall x\, (\mathbf{P}(x,x) \vee \neg\mathbf{P}(x,x))$, eftersom för varje element a i ett universum A så gäller antingen $P(a,a)$ eller inte.

3. $\models \forall x \mathbf{E}(x) \longrightarrow \mathbf{E}(\mathbf{c})$ eftersom om alla element i universum har egenskapen E (där E är tolkningen av $\mathbf{E}$), så har ju c egenskapen E (där c är tolkningen av $\mathbf{c}$).

4. $\models \forall x \mathbf{P}(x,x) \vee \neg\forall x \mathbf{P}(x,x)$. Satsen utsäger att antingen är P (den relation som $\mathbf{P}$ tolkas som) reflexiv eller så är den inte reflexiv.

Satserna i exemplen 1, 2 och 3 är om man tittar på deras satslogiska form, inte tautologier. (Deras satslogiska form är respektive p, p, $p \longrightarrow q$) Satsen i exempel 4 är valid eftersom dess form är tautologin $p \vee \neg p$.

Alla satser vars satslogiska form är en tautologi, är valida. Men vi ser också att det finns satser som är valida men som inte är tautologier.

3.10 Definition. *Logisk konsekvens.*
Låt $\mathcal{L}$ vara ett FOL, och låt σ vara en sats och Γ en mängd av satser i språket $\mathcal{L}$. Satsen σ är en *logisk konsekvens* av satserna i mängden Γ om σ är sann i varje struktur (som passar språket) som gör alla satserna i Γ sanna. Notation: $\Gamma \models \sigma$. $\square$

Man kan alltså uttrycka $\Gamma \models \sigma$ som att σ är sann i varje modell för Γ.
Om Γ är ändlig, säg $\Gamma = \{\psi_1, \ldots \psi_n\}$, så skriver vi $\psi_1, \ldots \psi_n \models \sigma$ (i stället för $\{\psi_1, \ldots \psi_n\} \models \sigma$).
Observera att om Γ är den tomma mängden, så betyder $\Gamma \models \sigma$ och $\models \sigma$ samma sak, dvs att σ är valid. Med andra ord, en sats σ är valid om den är en logisk konsekvens av tomma mängden.

3.11 Exempel. Låt $\mathcal{L} = \langle \mathbf{P}, \mathbf{Q} \rangle$ där $\mathbf{P}$ och $\mathbf{Q}$ är ett-ställiga relationssymboler. Låt ψ_1, ψ_2 och σ vara följande formler:

$$\begin{aligned}
\psi_1 &= \forall x\, (\mathbf{P}(x) \longrightarrow \mathbf{Q}(x)) \\
\psi_2 &= \exists x\, \neg \mathbf{Q}(x) \\
\sigma &= \exists x\, \neg \mathbf{P}(x)
\end{aligned}$$

Vi ska visa att $\psi_1, \psi_2 \models \sigma$.
Lösning. Låt $\mathcal{A} = \langle A, P, Q \rangle$ vara en godtycklig struktur, där $A \neq \emptyset$ är universum och P och Q är två delmängder av A. Anta att $\mathcal{A}$ är en modell för ψ_1 och ψ_2. Vi måste visa att $\mathcal{A}$ även är en modell för σ.
Att $\mathcal{A} \models \psi_1$ innebär att för varje element $a \in A$ så gäller att om $a \in P$ så är $a \in Q$. Dvs från $\mathcal{A} \models \varphi_1$ drar vi slutsatsen att $P \subseteq Q \subseteq A$.
Att $\mathcal{A} \models \psi_2$ innebär att det finns något element $a \in A$ sådant att $a \notin Q$.
Låt a vara ett element i A sådant att $a \notin Q$ (sådant a finns eftersom ψ_2 är sann i $\mathcal{A}$). Om $a \in P$, så följer av $P \subseteq Q$ att $a \in Q$. Motsägelse, ty vi valde ju ett element a utanför Q. Alltså kan det inte gälla att $a \in P$. Alltså är $a \notin P$. Det finns således något element i universum som är utanför P, dvs $\mathcal{A} \models \sigma$.
Vi har visat att σ är sann i varje modell för ψ_1 och ψ_2. Alltså gäller $\psi_1, \psi_2 \models \sigma$. $\square$

3.12 Exempel. Låt $\mathcal{L} = \langle \mathbf{P}, \mathbf{R}, \mathbf{c} \rangle$ där $\mathbf{P}$ är en ett-ställig relationssymbol, $\mathbf{R}$ är en två-ställig relationssymbol och $\mathbf{c}$ är en konstantsymbol. Låt ψ_1, ψ_2, ψ_3 och σ vara följande formler:

$$\begin{aligned}
\psi_1 &= \mathbf{P}(\mathbf{c}) \\
\psi_2 &= \forall y\, \mathbf{R}(\mathbf{c}, y) \\
\psi_3 &= \forall x \forall y\, (\mathbf{P}(x) \wedge \mathbf{R}(x, y) \longrightarrow \mathbf{P}(y)) \\
\sigma &= \forall y\, \mathbf{P}(y)
\end{aligned}$$

Visa att $\psi_1, \psi_2, \psi_3 \models \sigma$.
Lösning. Låt $\mathcal{A} = \langle A, P, R, c \rangle$ vara en godtycklig struktur, där $A \neq \emptyset$ är universum, $P \subseteq A$, $Q \subseteq A \times A$ är en två-stälig relation på A och $c \in A$ en konstant. (Vi skriver som vanligt aRb för $(a, b) \in R$.) Anta att $\mathcal{A}$ är en modell för ψ_1, ψ_2 och ψ_3. Vi måste visa att $\mathcal{A}$ även är en modell för σ.
Att $\mathcal{A} \models \psi_1$ innebär att $c \in P$.
Att $\mathcal{A} \models \psi_2$ innebär att för varje element $a \in A$ så gäller cRa.
Att $\mathcal{A} \models \psi_3$ innebär att för alla a och b i A så gäller att om $a \in P$ och aRb så följer $b \in P$. (Man skulle kunna säga att mängden P är uppåt sluten med avseende på relationen R.)

Vi vill visa att $\mathcal{A} \models \sigma$, vilket innebär att alla element i universum är med i mängden P. Vi måste alltså visa att $A \subseteq P$.
Låt a vara ett godtyckligt element i A. Vi vet enligt ψ_2 att cRa, och enligt ψ_1 att $c \in P$. Då följer det av ψ_3 (med c för x och a för y) att $a \in P$. Eftersom a var godtycklig i A så har vi nu visat att $A \subseteq P$. Eftersom $\mathcal{A}$ var en godtycklig $\mathcal{L}$-struktur så har vi visat att σ sann i varje modell för ψ_1, ψ_2, ψ_3. Alltså gäller $\psi_1, \psi_2, \psi_3 \models \sigma$. $\square$

3.3 Syntaxen

Låt $\mathcal{L}$ vara ett FOL. Kom ihåg att en sats är en sluten formel, dvs en formel som saknar fria variabelförekomster. Låt Γ vara en mängd av satser, och σ en sats, i $\mathcal{L}$. För att göra formella bevis i predikatlogik använder vi alla bevisregler som vi hade i satslogik, dvs en introduktionsregel och en eliminationsregel för varje konnektiv och för $\doteq$. Vi behöver lägga till regler för $\exists$ och $\forall$. Se boken till LPL för formuleringar av dessa fyra regler. Var noga med att lära dig restriktionerna för de olika reglerna:

1. ($\forall$-elim): Om vi har bevisat $\forall x\, \varphi(x)$ och $\mathbf{t}$ är en sluten term, så får vi dra slutsatsen $\varphi(\mathbf{t})$.

2. ($\forall$-intro): Om vi har ett delbevis utan premisser med konklusionen $\varphi(\mathbf{c})$, och $\mathbf{c}$ är en konstantsymbol som inte förekommer utanför detta delbevis, så får vi dra slutsatsen $\forall x\, \varphi(x)$.

3. ($\exists$-intro): Om vi har bevisat den slutna formeln $\varphi(\mathbf{t})$, där $\mathbf{t}$ är en sluten term, så får vi dra slutsatsen $\exists x\, \varphi(x)$.

4. ($\exists$-elim): Om vi har (den slutna formeln) $\exists x\, \varphi(x)$ *och* vi har ett delbevis med premiss $\varphi(\mathbf{c})$ och konklusion $\mathbf{A}$, där $\mathbf{A}$ är en sluten formel som *inte* innehåller konstantsymbolen $\mathbf{c}$, och $\mathbf{c}$ inte heller förekommer utanför detta delbevis, så får vi avsluta delbeviset och dra slutsatsen $\mathbf{A}$.

Vi säger att σ är formellt bevisbar från premisser i Γ och skriver $\Gamma \vdash \sigma$, om det finns ett formellt bevis vars premisser tillhör Γ och vars slutsats är σ.
Kom även ihåg att alla formler i de formella bevisen är slutna, man får t ex inte göra $\forall$-elim av $\forall x \varphi(x)$ och dra slutsatsen $\varphi(x)$. Man måste dra slutsaten $\varphi(\mathbf{t})$ för någon sluten term $\mathbf{t}$.
Läs mer om formella bevis och gör många övningsuppgifter, t ex från kap 13 i kursboken. På sid 557 och framåt finns en sammanställning av alla bevisregler.

3.4 Samband mellan semantik och syntax

Sundhetssatsen säger att man med våra bevisregler inte kan bevisa något som inte är sant.

3.13 Sundhetssatsen. *Om* $\Gamma \vdash \sigma$*, så* $\Gamma \models \sigma$.

Bevis Beviset är med induktion över formella bevis, och man visar att alla regler bevarar sanning. $\square$

Om $\Gamma = \emptyset$, så får vi följande specialfall av sundhetssatsen:

$$\text{Om} \vdash \sigma, \text{ så} \models \sigma.$$

Dvs om en sats kan bevisas utan premisser, så är den valid (sann i alla strukturer).

3.14 Fullständighetssatsen. *Om* $\Gamma \models \sigma$, *så* $\Gamma \vdash \sigma$.

Bevis Se texten om sekventkalkyl för en skiss av ett bevis. $\square$

Om $\Gamma = \emptyset$, så får vi följande specialfall av fullständighetssatsen:

$$\text{Om } \models \sigma, \text{ så } \vdash \sigma.$$

Dvs om en sats är valid, så har den också ett formellt bevis utan premisser.
Sundhetssatsen och fullständighetssatsen kan naturligtvis sammanfattas som en enda sats:

$$\Gamma \vdash \sigma \iff \Gamma \models \sigma$$

Referenser

1. van Dalen, D. *Logic and Structure*, 4th ed. Springer 2004.
2. Barwise, J. och Etchemendy, J. *Language, proof and logic*, CSLI 2003.