7. FORMELL SATSLOGIK (SL)

7.1 VEM BEHÖVER FORMELL LOGIK?

Ingen använder formell logik i det dagliga livet. Den logik vi använder, den naturliga eller intuitiva logiken, är, som vi sett, varierande och komplicerad. Att uttrycka den som ett formellt system är omöjligt. Vi lär oss den inte genom att memorera regler utan genom att använda den i praktiken ända från barnsben. Men många typer av slutledningar kan rekonstrueras i formella logiska system. Den moderna forskningen om sådana system började i mitten av 1800-talet och var mycket intensiv under 1900-talet.

Vem behöver formell logik, och varför? Detta är en fråga som otaliga studenter har frågat sig (men låtit bli att fråga läraren). För att få ett svar måste man känna till den historiska bakgrunden.

Aristoteles var den förste som insåg att vissa argument får sin giltighet på grund av den logiska funktionen hos några få ord. Innehållet, begreppen spelar i sådana argument ingen roll för giltigheten. Vi kan därför bortse från innehållet och bara se till formen. Vi har redan haft en mängd exempel t.ex. syllogismerna. Genom formalisering kan vi uttrycka generella regler, som sedan kan tillämpas på en oändlig mängd olika konkreta exempel. I modern formell logik går man ett steg längre och bygger upp system i vilka man inte alls använder naturliga språk utan enbart exakt definierade tecken. Man skapar därmed system av den typ vi är vana med från matematiken. Den formella logiken kallas därför ibland matematisk logik. En annan benämning är symbolisk logik. Den är dock missvisande därför att också den intuitiva logiken arbetar med symboler. All logik är i grund och botten symbolisk. Denna text t.ex. består enbart av symboler. En symbol är ett tecken av något slag som har en mening.

Vi får samma fördelar som matematiken ger, dvs klara och exakta regler för hur vi får och inte får hantera symbolerna. Naturliga språk är inte klara, entydiga och exakta. Det finns nästan alltid flera tolkningsmöjligheter. De flesta uttryck är mer eller mindre vaga. Det finns inga exakta regler för vilka slutsatser som följer. Detta är för det mesta inget problem, utan en fördel. Det gör språket ofantligt flexibelt för nästan vilka ändamål som helst.

Naturliga språk har utvecklats för att hjälpa oss i det vardagliga livet. De är oöverträffade för detta ändamål. De är levande, ständigt föränderliga, anpassas till livets och samhällets förändringar. Deras främsta fördel är att vi kan uttrycka mycket information kort och enkelt. Men att lösa komplicerade matematiska problem går inte alls om vi måste uttrycka oss i naturliga språk. En enkel naturlag, t.ex. Newtons gravitationslag blir otymplig om vi uttrycker den i ett naturligt språk. Än mer otympligt blir det att beskriva matematiska bevis. De blir mycket snabbt obegripliga.

För formella logiska språk gäller detsamma. De medger att vi uttrycker oss med klarhet och exakthet enligt noggrannt stipulerade regler. Ett formellt system ger inget rum för tolkningar. Medan naturliga språk är ytterst flexibla är formella språk helt rigida. Svagheten är att vi förlorar information. Endast vissa typer av problem kan formaliseras. Formella språk är omöjliga att använda i t.ex. normala samtal. Läsaren kommer att inse varför om han/hon bekantar sig med min presentation av några grundläggande formella system nedan.

Efter Aristoteles hände det föga nytt inom den formella logiken förrän under mitten av 1800-talet. Grundaren av den moderna formella logiken blev en tysk matematiker och filosof vid namn Gottlob Frege (1848-1925). Frege var en tillbakadragen teoretiker som arbetade med ett högst abstrakt problem som diskuterades livligt av filosofiskt sinnade matematiker under 1800-talet. Den store tyske filosofen Immanuel Kant (1724-1804) hade hävdat att artimetikens sanningar, t.ex. 5+7= 12 är syntetiska. Detta innebär att man kan förneka en matematisk sanning utan att göra sig skyldig till en motsägelse. Den inflytelserika engelska filosofen John Stuart Mill (1876-1873) gick ännu längre och påstod att matematiska sanningar är empiriska och baserade på induktion. Vi har många gånger iakttagit att två objekt som läggs ihop med två andra objekt ger fyra objekt. Denna observation har vi generaliserat till den aritmetiska sanningen att 2+2=4. Om Kant eller Mill och deras anhängare har rätt så innebär det att aritmatiken vilar på en tämligen osäker grund. Kant konstruerade en ytterst invecklad teori om att den vilade på i vår natur inbyggda sätt att uppfatta rum och tid. Mills uppfattning var enklare men innebar att det inte finns någon annan grund för matematiken än våra observationer i kombination med induktion. Som vi tidigare har sett ger induktion sannolikhet snarare än säkerhet. Enligt Mill borde vi sålunda säga att det är sannolikt att 5+7=12, inte att det är orubbligt säkert!

Denna diskussion späddes på av att matematikerna i början av 1800 hade upptäckt att det finns s.k. icke-euklidiska geometrier. Ända sedan antiken hade alla matematiker varit övertygade om att Euklides geometri är den enda möjliga beskrivningen av ytor och rum. De utgick ifrån att hur stora rum v i än tänker på, t.o.m. hela världsrymden, beskrivs korrekt av denna enda geometri. Eftersom det inte finns något annat motsägelsefritt sätt att beskriva ytor och rum än Euklides så måste han teorem vara absolut säkra. Kant var så säker på detta att han utvecklade ett invecklat bevis som innebär att vårt sätt att uppfatta världen är en del av vår natur. Vi kan inte uppfatta rum på något annat sätt än så som Euklides beskriver det.

Men några decennier efter Kants död gjorde några matematiker just det som enligt Kant var omöjligt. De konstruerade nya geometrier som beskrev rummet på ett annat sätt än Euklides. Därmed var det inte längre självklart att Euklides geometri, som i alla tider betraktats som ett ideal av absolut sanning, var sann. Det var möjligt att Euklides teorem bara gällde i våra små omständigheter och att helt andra geometrier gällde mycket stora eller extremt små rum. Och om man inte längre kunde lita på Euklides - kunde man då lita på någonting alls inom matematiken? Kan vi ens lita på vanlig enkel aritmetik?

Dessa tvivel på matematikens orubblighet bekymrade Frege i hög grad. Han koncenterade sig på det enklaste området inom matematiken, dvs aritmetiken och beslöt att en gång för alla bevisa dess orubbliga sanning. Men hur bevisa detta en gång för alla och så att ingen kunde ifrågasätta bevisen?

Detta blev ett livslångt projekt för Frege. Det krävde att han kunde utgå från helt säkra och orubbliga sanningar som axiom. Var finner man sådana? Endast inom logiken, menade han. Men hur skulle härledningen kunna göras absolut säker? Ända sedan antiken hade matematiker och filosofer producerat otaliga bevis. Gemensamt för dessa var att de helt eller delvis skrevs ut i vanligt språk och att de i hög grad byggde på intuitiv logik. Det fanns därför möjligheter till misstag. För att helt eliminera dessa behöver vi, menade Frege, ett helt formell, exakt språk med entydiga regler för hur man härleder teorem från axiomen.

Det var ett väldigt projekt för en enda, ensam man, som för övrigt rönte föga förståelse. Filosoferna var inte intresserade ty de menade att han sysslade med matematik, inte med filosofi. Konstigt nog var matematikerna lika ointresserade. De tyckte att det fanns alltför mycket filosofi, som de inte intresserade sig för, i Freges publikationer. Frege genomförde dock sitt projekt och blev därmed småningom, men först på äldre dagar, erkänd som en av de stora inom både filosofi och matematik. Anders Wedberg skriver:

> "Trots många tidigare ansatser till formalisering - särskilt inom den logiska algebran (Leibniz, Boole m.fl.) - gav Frege med sin *Begriffsschrift, eine der aritmetischen nachgebildete Formelsprache des reinen Denkens* (1879) det första exemplet på ett strängt formaliserat och mycket uttrycksfullt språk. Inom ramen för detta språk gav han en strängt formaliserad logisk teori, som innefattade en formaliserad satslogik (den klassiska tvåvärdiga), en formaliserad kvantifikationslogik (med kvantifikation av "föremål" och "funktioner av första ordningen") och en formaliserad identitetsteori. Dessutom gav han som sagt en - låt vara fragmentarisk, och som det senare skulle visa sig, motsägelsefull - *formaliserad mängdteori.* Ingen har före

Frege lika klart som han fattat och genomfört formaliseringens idé. På detta område är han den store pionjären."[1]

Freges *Begriffsschrift* väckte till en början föga intresse. Han var t.o.m. tvungen att själv betala för publiceringen av detta verk som småningom blev en klassiker inom logiken.

I England fanns en ung filosof som i slutet av 1800-talet tänkte i alldeles samma banor som Frege. Också han grubblande över aritmetikens hållbarhet. Också han kom till slutsatsen att man måste försöka visa att aritmetiken kan härledas från enkla och självklara logiska sanningar. När han inledde sitt projekt kände han inte alls till Frege och hans insatser, men när han fick kännedom om den tyske filosofen läste han med iver och sakkännedom hans skrifter. Sålunda kom det sig att när Frege år 1903, när han stod i beråd att ge ut en ny upplaga av boken *Die Grundgesätze der Aritmetik,* fick ett brev av den då 31-årige filosofen i vilken denne påpekade att Freges system innehöll en motsägelse, som senare kom att kallas Russells paradox. Den unge filosofen hette alltså Bertrand Russell (1872- 1970). Tillsammans med matematikern A.N. Whitehead arbetade Russell senare på ett stort arbete vars mål var att formellt härleda aritmetiken från logiken. Resultatet av deras möda blev boken *Principia mathematica*, också den en klassiker inom logiken, som utkom i tre delar 1910-1913. Denna bok fick stort inflytande inom såväl matematik som filosofi. De flesta läroböcker i formell logik har sedan dess baserat sig på detta arbete. Boken behandlar satslogik, predikatlogik och mängdlära. I mängdläran införs regler som gör att Russells paradox inte kan uppstå. Författarna bevisar sedan en mängd aritmetiska sanningar, t.ex. att 2+2=4 med hjälp av sitt formella logiska språk. Boken bevisar därmed att det faktiskt är möjligt att härleda aritmetiken ur logiken. Någon egentlig praktisk betydelse fick detta bevis dock aldrig eftersom ingen någonsin på allvar betvivlat att 2 + 2 = 4 eller att 7 + 5 = 12. Däremot fick det formella språk som Frege och senare Russel och Whitehead skapade stor betydelse inom andra områden, och speciellt inom informationsteknologin. Boken stimulerade i hög grad intresset för formella språk. Mängder av nya logiska system har konstruerats under 1900-talet.

Den formella logiken uppstod sålunda som ren grundforskning och förblev detta fram till 1930-talet. Då insåg flera forskare, oberoende av varandra, att formell logik lämpar sig utmärkt när man bygger elektroniska maskiner som behandlar data, dvs elektronhjärnor, som många till en början kallade dem med en minst sagt fantasifull benämning. Förväntningar på dessa maskiner var enorma under 50- och 60-talet, men det var också dystopierna, de skrämmande framtidsscenarierna om en värld styrd av datorer. . . Som så många gånger tidigare och senare i vetenskapens historia blev det som börjat som ett högst abstrakt problem inom den rena grundforskningen med tiden av enorm praktisk betydelse.

Det enda sättet att förstå vad formell logik är och vad den handlar om är att lära sig grunderna för några system. Detta är inte svårt, men kräver att man jobbar en smula med övningar och lär sig använda exakta definitioner och regler. Man får då en viss förståelse för vad man kan och inte kan uppnå med formella metoder. Viktigt är att man hela tiden har i minnet att allt i sista hand bygger på intuitiv logik. Jag citerar logikern Daniel Bonevac: "Logiska system är matematiskt exakta modeller av hur man resonerar inom vissa språkliga områden. När logikerna konstruerar sådana modeller måste de göra antaganden om meningen hos de uttryck de studerar. Logikerna försöker skapa [formella] system i vilka just de argument blir giltiga, som de som talar språket, när de funderar över saken, anser vara giltiga."[2] Vad detta betyder konkret kommer att klarna när läsaren lär känna de formella system som jag presenterar medan.

[1] Anders Wedberg, *Filosofins historia.Från Bolzano till Wittgenstein*. Bonniers 1966, s. 104.

[2] Daniel Bonevac *Deduction. Introduction to Symbolic Logic.* Mayfield Publishing Company 1987, s. 336.

Jag börjar med ett mycket enkelt, men dock högst användbart system som kallas formell satslogik (SL). Det har blivit praxis att läroböcker i formell logik inleds med detta system. Jag behandlar detta system tämligen detaljerat. När man kan SL är det lätt att gå vidare till formell modal satslogik (MSL). Sedan behandlar jag formell deontisk satslogik (DSL). Dessa båda system behandlar jag kortfattat eftersom de är mindre viktiga än SL.

Ett betydligt mera kompliceratg system är den formella predikatlogiken (PL). Den kallas också kvatifikationslogik. Den baserar sig på SL men gör en långt noggrannare analys möjlig. Min presentation är ganska kortfattad. Dessa system kan sedan utvidgas och kompletteras på olika sätt. Man kan t.ex. utvidga PL genom att införa modala och deontiska operatorer i systemet.

Yrkeslogikerna har skapat massvis med olika system för olika ändamål. De flesta är av intresse endast för den som sysslar med ren grundforskning. De som behandlas här är de som historiskt sett spelat den viktigaste rollen och som kommit att få tillämpningar inom olika områden såsom vetenskapsteori, beslutsteori, teknologi och informationsbehandling.

7. 2 FORMALISERING INOM SL.

Det enklaste och mest grundläggande formella systemet är den s.k. satslogiken (i fortsättningen SL). Redan under antiken studerade en del filosofer logiken hos ord som *inte, och, eller* och *om-så*. SL är ett formellt system för dessa ord. Läroböckerna i formell logik börjar vanligen med formell SL

För ovanlighetens skull är det inte Aristoteles som är upphovsmannen. Satslogik började utforskas först efter hans död. Det var främst inom den stoiska skolan, en filosofisk riktning med stort inflytande under antiken, som man intresserade sig för hur dessa ord fungerar. Stoikerna utgick från begreppet *axioma*. (Ordet har förstås gett upphov till vårt ord *axiom*, men det betyder alltså sats). Det finns, menade de två typer av satser, enkla och sammansatta. Exempel på enkla satser är *Det är dag. Solen skiner. Det är ljust.* Exempel på sammansatta är *Det är dag och solen skiner. Om det är dag så är det ljust. Det är dag eller det är natt.* Man diskuterade sedan vilka slutsatser som följer av olika typer av sammansatta satser.

I ett naturligt språk kan vi urskilja flera huvudkomponenter. För det första har vi ordförrådet, som vi här kallar *vokabulären*. Vokabulären i svenskan består av tiotusentals ord. För att förstå vardaglig text måste man känna flera tusen ord. Men orden kan inte radas efter varandra hur som helst. Man måste ha en *grammatik*. Den ger reglerna för hur vi får och inte får kombinera ord. Slutligen har vi en *semantik*. Denna talar om vad som menas med sanning och under vilka villkor en sats är sann. Varje språk, också formella sådana, måste ha något som motsvarar vokabulär, grammatik och semantik hos naturliga språk. Det enklaste sättet att kort presentera ett formellt språk är därför att ange dessa. För SL är detta enkelt.

Vokabulären för SL.

Vokabulären består av s.k. satsbokstäver och konnektiver (sammanbindare). Vi låter små bokstäver a,b,c osv stå för enkla satser. Man kan vid behov använda a^1, a^2 osv. Att en sats är enkel betyder i SL att den inte innehåller något av orden *inte, och, eller, om-så* och *om och endast om*, eller något ord som används synonymt med dessa. Vidare ingår i vokabulären ett antal specialtecken som är kännetecknande för SL. Olika böcker använder något olika tecken. Här använder vi följande: **-, &, v, →, ↔.** Dessa tecken kallas konnektiver (sammanbindare). De binder ihop enkla satser till sammansatta. Tekniskt uttryckt sett anger de funktioner på samma sätt som tecknen +, - , x , : i aritmetiken. Därtill innehåller vokabulären parenteser **( )** och klamrar **{ }.** Andra tecken än dessa finns inte i standardversionen av SL.

När vi resonerar inom SL får vi endast använda dessa tecken och reglerna för dem. Men när vi beskriver SL och diskuterar dess för och nackdelar etc får vi förstås använda det naturliga språket och vilka andra system som helst. SL kallas då objektspråk. Det är objektet för vårt intresse. Det språk vi använder när vi talar om SL, dvs det naturliga språket, kallas metaspråk. Objektspråket har alltså en högst begränsad vokabulär. Metaspråkets vokabulär är däremot mycket stor.

Grammatiken för SL.

Grammatiken i SL talar om för oss vilka räckor av tecken som godkänns som formler i SL. Den består av blott tre regler. (En enklare grammatik kan man knappast tänka sig!) När jag skriver ner dessa regler använder jag A och B som variabler för formler i SL, s.k. metavariabler. I stället för A och B kan man sätta in vilken formel som helst i SL. A och B ingår sålunda inte i vokabulären för SL, utan används i det metaspråk vi använder när vi talar om SL. Här är reglerna:

- Varje satsbokstav är en formel.
- Om A är en formel så är -A en formel.
- Om A och B är formler så är A & B, A v B, A→B och A↔B formler.

Exempel: Enligt regel 1) är **a** en formel, men också **b** och **c.** Om vi tillämpar regel 2) på dessa så får vi att **-a, -b,** och **-c** är formler. Om vi tillämpar regel 3) på **a,b,c** får vi en rad nya formler t.ex. **a & b, a v b, a→c, b↔c.** Enligt 2) får vi alltid en formel när vi skriver - framför en formel. Alltså är också t.ex. **- a & b)** och **-(a v b)** formler. Om vi tillämpar regel 3) på dessa får vi att **-(a & b) v -(a v b)** är en formel. Man kan sålunda bygga hur långa formler som helst med hjälp av vokabulären och grammatiken. Märkväl att reglerna tillåter konstiga formler som **a & a & a & a.** Parenteserna och klamrarna använder vi för att markera vad som hör ihop. Om en formel enbart innehåller **&** eller **v** eller så behöver vi inte skriva ut paranteser. Men det är inte heller förbjudet. Om en formel innehåller → eller ↔ måste vi däremot alltid skriva ut paranteser för att det skall vara klart vad som hör ihop. Vi får alltså inte skriva **A→B→C**. I stället måste vi antingen skriva **(A→B)→C** eller **A→(B→C)**.

Semantiken för SL.

Semantiken berättar för oss vad tecknen betyder och under vilka villkor en formel betraktas som sann i SL. Detta system är av den typ som kallas klassisk logik. Det betyder att det finns två sanningsvärden. Eftersom jag ofta måste använda ordet sanningsvärde i fortsättningen förkortar jag det till SV. Dessa betecknar vi med bokstäverna S och F. S = sant, F = falskt, osant. För SL gäller förstås logikens grundlagar. Varje formel antas sålunda ha värdet S eller F, och ingen formel kan ha båda värdena samtidigt. T.ex. formeln **-{(a & b) v -(b v -c)}** har ett bestämt SV. Hur får vi reda på detta. Här kommer vi in på själva grundidén med SL.

Om man känner till SV för de enkla formlerna a,b och c så kan man rent mekaniskt räkna ut SV för hela den sammansatta formeln. Uttryckt på ett annat sätt: SV hos varje komplex formel är en funktion av SV för de enkla (atomära) formler av vilka den komplexa formeln är uppbyggd. Anta att vi har följande SV a = S, b = F och c = S. Vilket SV får då hela formeln? Den får värdet F.

Hur räknar man ut SV för en godtycklig formel? För att kunna göra detta måste vi ge exakta definitioner av konnektiverna. Vi måste ge dessa tecken en exakt mening. Det gör vi genom att ange deras sanningsvillkor, dvs. under vilka förhållanden de är sanna respektive falska.

För att räkna ut SV för en komplex formel behöver vi, som jag nämnde ovan, känna SV för de atomära formler av vilka den är uppbyggd. Hur får vi reda på dessa? *Det får vi inte! Detta problem kan inte lösas inom ramen för SL.* Tag som exempel påståendet *Aristoteles hade tre barn.* Vi vet inte om det är sant eller inte. Det som vi vet är att det antingen är sant eller inte. SL ger oss alltså ingen metod att bestämma SV för atomära satser. Men den ger en metod att bestämma SV för komplexa satser förutsatt att vi känner värdena för de atomära. Men det viktigaste är att SL ger oss en metod att *systematiskt gå igenom alla tänkbara möjligheter.* Detta gör vi genom att rita upp något som kallas sanningsvärdetabeller, i fortsättningen SV-tabeller. Betydelsen hos varje konnektiv definieras exakt genom en sådan tabell.

Tabellen för - är enkel och självklar. (När jag skriver ner dessa regler använder jag igen metavariabler, bokstäver som är variabler för formler i SL). Om en formel A har värdet S så har - A förstås motsatt värde, dvs. F. Och om A har värder F så har - A självklart värdet S. Detta konnektiv kallas *negation*. Regeln för negationen är sålunda att den ändrar, kastar om, SV på den formel som den styr. Här är SVT för de övriga konnektiverna:

Sanningsvärdetabellerna för konnektiverna i SL.

A B	A & B	A v B	A → B	A ↔ B
S S	S	S	S	S
S F	F	S	F	F
F S	F	S	S	F
F F	F	F	S	S

Detta är grundtabellen för konnektiverna i SL. Man utgår alltid från denna tabell när man räknar ut sanningsvärdena för alla andra formeler i SL.

För två formler A och B finns det fyra och blott fyra möjliga kombinationer av SV. Dessa fyra kombinationer ser vi i kolumnen längst till vänster. För var och en av dessa kombinationer får varje formel, som innehåller ett konnektiv, ett bestämt värde. Dessa SV finns i kolumnen under respektive konnektiv. Ex: I den översta raden ser vi längst till vänster att både A och B är sanna. När vi går mot höger ser vi vilket SV en formel med vart och ett av de olika konnektiven får i detta fall. Vi ser att alla blir sanna om både A och B är sanna.

En formel där huvudkonnektivet är & kallas *konjunktion*. (Av lat. conjugere= att sammabinda). Tecknet & i SL motsvarar många, men inte alla, användningar av *och* i naturliga språk. Det motsvarar också många användningar av *men*. I tabellen ser vi att en formel A & B är sann om och endast om (som vi i fortsättningen förkortar till omm) både A och B är sanna. Om någondera eller båda är falska så blir också konjunktionen falsk. Denna regel kan vi tillämpa på varje formel som är en konjunktion, hur många satsbokstäver den än innehåller. Sålunda är en formel A & B & C sann omm alla tre atomära satser är sanna. För att en formel A & B & C & D skall vara sann måste alltså var och en av formlerna A,B,C och D ha värdet S. Annars är den osann.

En formel där huvudkonnektivet är v kallas *disjunktion*. (Av lat. disjungere = åtskilja). Tecknet v motsvarar användningen av *eller* i naturliga språk förutsatt att man avser det s.k. *inklusiva eller*. Detta används när A och B inte utesluter varandra. Det motsvarar däremot inte *antingen eller* som kallas för *exklusivt*. För att uttrycka *antingen eller* kan vi skriva (A v B) & -(A & B). Men vi kan, om vi så vill, använda ett skilt tecken $\underline{v}$ för *antingen eller*. I kolumnen under v ser vi att en disjunktion är sann omm både A och B eller någondera är sann. Det räcker m.a.o. att en enkel formel i en disjunktion är sann för att hela den sammansatta formeln skall vara sann. Om t.ex. A har värdet S i formeln A v B v C v D så blir hela formeln sann. Däremot är en formel som innehåller det exklusiva *eller*, t.ex. A $\underline{v}$ B $\underline{v}$ C , sann endast om inte alla tre är sanna eller alla tre falska. För att en formel A v B v C v D skall vara falsk måste sålunda alla de atomära formlerna vara falska.

En formel där huvudkonnektivet är → kallas *materiell implikation*.(Av lat. implicare = att förbinda). Detta konnektiv används som motsvarighet till uttrycket *om-så* eller synonyma uttryck i naturliga språk. Det är dock ingen exakt motsvarighet. SV för en sats som innehåller *om-så* i ett naturligt språk kan inte bestämmas enbart utgående från en SVT. Av SVT ovan ser vi att formeln A → B är sann i alla andra fall utom när A = S och B = F. Sålunda är formeln alltid sann när A = F, dvs. när försatsen är falsk. Detta stämmer inte med intuitiv logik. Tag t.ex. satsen *Om Hans Rosing är 50 år gammal år 2003 (falskt), så är han född år 1944 (sant).* Ännu konstigare är att t.ex. satsen *Om Hans Rosing är 100 år gammal så är Finland en del av Ryssland* är sann enligt SL. Satserna är sanna enligt SL, men knappast enligt intuitiv logik. I själva verket är det omöjligt att konstruera en motsvarighet till *om-så* inom ramen för SL. Den materiella implikationen är det närmaste man kan komma.

En formel där huvudkonnektivet är ↔ kallas *ekvivalens*.(Av lat. aequivalere = att ha samma kraft). Detta tecken motsvarar uttrycket *om och endast om* i naturliga språk. Av SV-tabellen ser vi att formeln A↔B är sann omm båda har samma sanningsvärde.

I läroböckerna i logik talar man om *formalisering* av satser i naturliga språk. Denna påminner något om en översättning från ett språk till ett annat. Formalisering innebär att man analyserar satser (eller påståenden, argument, teorier etc) med hjälp av ett formellt system. Det som vi gör när vi formaliserar med hjälp av SL är, exakt uttryckt, att med formler visa upp hur de enkla satserna är "kopplade" till varandra. Dessa kopplingar har de logiska egenskaper som framgår av SVT.

När man formaliserar en sats i SL börjar man med att söka efter motsvarigheter till konnektiverna i satsen. Man sätter sedan in konnektiver för ord som *inte, icke, ej, och, men, eller, antingen eller, om-så, ifall, endast om* och *om och endast om*. Detta är inte en mekanisk procedur utan kräver förståelse. Översättningen kan därför inte mekaniseras, den kan inte skötas av ett datorprogram. Sedan sätter man in satsbokstäver för de enkla satserna.

Exempel: formalisera satsen "**Om** man äter hälsosamt **och** motionerar mycket **så** blir man **inte** sjuk." Jag markerar uttryck som motsvarar konnektiver med fetstil. Vi sätter in konnektiver för dessa uttryck och får "Man äter hälsosamt **&** motionerar mycket → **-** man blir sjuk." Nu ersätter vi de enkla satser som är kvar med godtyckligt valda bokstäver, t.ex. så här

h = Man äter hälsosamt.
m = Man motionerar mycket.
s = Man blir sjuk.

När vi sätter in dessa bokstäver i st.f. satserna får vi den slutgiltiga formeln (**h & m)** → **- s**. Vi anger med en parantes att h och m hör ihop. Vi utläser formeln: h och m implicerar icke s.

Exempel: formalisera satsen "Man bör äta mycket frukt **och** bär, **och** vidare mörkt bröd, fisk **och** magert kött, **men inte** mycket socker **och** fett." Här är ett *och* underförstått mellan mörkt bröd, fisk. Också här sätter vi alltså ut ett &. Vi ersätter de enkla satserna med bokstäver. Självklart får samma bokstav inte användas för två olika satser. Om samma sats förekommer flera gånger skall den givetvis ha samma bokstav.

f= Man bör äta mycket frukt.
b= Man bör äta mycket bär.
m= Man bör äta mörkt bröd.
i= Man bör äta fisk.
k= Man bör äta magert kött.
s= Man bör äta mycket socker.
t=Man bör äta mycket fett.

Vi får då formeln **f & b & m & i & k &-s &-t**. Vi behöver inga parenteser därför att det bara finns en möjlig tolkning av formeln när vi endast använder konnektivet &.

Ytterligare ett exempel. Formalisera följande berömda uttalande av Jesus i Bergspredikan. "**Om** någon slår dig på den högra kinden **så** vänd den andra till åt honom; **och om** någon vill gå till rätta med dig för att beröva dig din livklädnad, **så** låt honom få manteln med; **och om** någon tvingar dig att till hans tjänst gå med en mil, **så** gå två med honom." Vi markerar alla motsvarigheter till konnektiver, sätter in symbolerna för dessa och ersätter de enkla satserna med satsbokstäver, n = Någon slår etc; v = Vänd den andra etc; g = Någon vill gå till rätta. etc; l = Låt honom få etc; t = Någon tvingar dig etc; v = Gå två med honom.
Formaliseringen i SL blir alltså   **(n→v) & (g→l) & (t→v)**

Läsaren bör själv försöka formalisera följande satser.

1) Det var Runeberg eller Topelius, men inte Lönnrot som skrev Fänrik Ståls sägner.
2) Regeringen måste ta upp mera lån om den inte minskar statens utgifter eller höjer skatterna.
3) Den sökande får den lediganslagna tjänsten om och endast om hon är kompetent, ogift och inte har alltför höga lönekrav.
4) Antingen är slutsatsen i ett deduktivt argument sann eller så är argumentet ogiltigt eller minst en av premisserna falsk.
5) Ett argument är deduktivt och sunt om och endast om det är omöjligt att alla premisser är sanna och slutsatsen falsk samt premisserna de fakto är sanna.

7.3 DEN AXIOMATISKA BEVISMETODEN

Frege och Russell skapade SL för att kunna bevisa satser på ett exakt sätt. Det var förstås grekerna som hittade på begreppet bevis. Aristoteles skrifter vimlar av bevis. Han var också den förste som konstruerade ett axiomatiskt system genom att axiomatisera sin teori om syllogismerna. Historiens mest inflytelserika axiomatiska system konstruerades emellertid av en grekisk matematiker som hette Euklides. Om hans liv har de bevarade källorna inget att berätta. Vi vet inte ens med säkerhet när han levde, men troligen omkring 300 f.Kr. Euklides sammanfattade sin tids matematiska vetande i ett stort arbete *Elementa*. Han utgick från ett fåtal definitioner av begrepp samt från ett antal axiom, grundläggande, självklara sanningar. Från axiomen härledde han sedan en stor mängd teorem, t.ex. det berömda Pytagoras teorem.

Den axiomatiska metoden har sedan antiken framstått som den bästa bevismetod som finns. Anledningen till detta är lätt att förstå. Axiomen bör väljas så att det inte kan råda några tvivel om deras sanning. Euklides utgår från 10 axiom. (Fem av dem kallar han postulat, men de är av samma karaktär som axiomen). Några exempel på axiom:

1. Man kan alltid dra en rät linje från en punkt till en annan.
2. Varje begränsad linje kan förlängas obegränsat.
3. Kring en medelpunkt kan man alltid beskriva en cirkel med en bestämd radie.
4. Alla räta vinklar är lika.
5. Storheter som täcker varandra är lika stora.
6. Det hela är större än sin del.

Från axiomen härleder man sedan teorem med hjälp av deduktivt giltiga regler. Som vi tidigare har sett definieras deduktiv giltighet så att man från sanna premisser aldrig kan komma till en falsk slutsats. Regeln är som man säger *sanningsbevarande*. Detta betyder att om axiomen faktiskt är sanna och härledningsreglerna giltiga så är också alla teorem sanna.
Det fanns dock ett problem med denna metod. En noggrann analys har visat att Euklides, säkerligen omedvetet, ibland använde antaganden som han inte explicit formulerat. När

han gjorde slutledningar förlitade han sig på intuitiv logik. Det fanns sålunda ett visst spelrum för tolkningar och misstag.

Det tyska universalgeniet G.W. Leibniz (1646-1716) var den första som föreslog att man borde försöka skapa ett exakt, formellt språk i vilket också slutledningsreglerna kunde formuleras. När två filosofer var oeniga i någon fråga skulle de, menade han, kunna formulera frågan som en formel i ett exakt formellt språk. Sedan kunde de med hjälp av samma typ av räkneregler som finns i matematiken räkna fram det rätta svaret. För mångsysslaren Leibniz, som för övrigt var mycket förtjust i Aristoteles logik, blev detta dock blott en vision. Han försökte aldrig själv skapa den *characteristica universalis,* det universella formella språk, som han drömde om.

Det skulle gå mer än hundra år efter hans död innan någon tog ett första steg mot förverkligandet av denna vision. Denne någon var en engelsk matematiker vid namn George Boole (1815-1864). Han lade märke till likheter mellan vissa regler i algebran och regler inom satslogiken. Han skapade ett slags logisk algebra som har samma struktur som den SL som Frege och Russell utvecklade. Sådana strukturer går i dag under benämningen *boolsk algebra*.

Booles upptäckt väckte inget större intresse. Han hade inte samma storslagna vision som Frege och Russell. Han utvecklade varken något formellt språk, predikatlogik eller mängdlära. Han hade föga intresse för de grundläggande filosofiska frågor som de senare kämpade med. Det är därför helt korrekt att ge Frege äran av att vara den moderna formella logikens fader.

Frege och Russell använde förstås den axiomatiska metoden i sina system. Vi skall här mycket kort se hur Russell arbetade. Axiomen måste självklart väljas med yttersta omsorg. Det är ju på dem hela systemet grundar sig. Tanken är att axiomen i sig implicit skall innehålla allt som är sant i systemet. Teoremens funktion är blott att tydligt uttrycka det som redan implicit finns i mängden av axiom. Axiomen måste vara sanna. Därför måste de vara så enkla att det inte kan råda någon oenighet om deras sanning. De skall förstås vara av logisk natur. Meningen var ju att härleda matematiken, åtminstone aritmetiken, från logiken. Vidare måste axiomen vara så få som möjligt.

Russell utgick från blott fem axiom. Senare kunde det dock visas att ett av dessa fem i själva verket kunde härledas som ett teorem ur de fyra övriga. Sålunda kom SL i *Principia* att bygga på blott fyra axiom. (För predikatlogiken och mängdläran måste ytterligare exiom förstås antas). Dessa kan formuleras i det formella språk jag ovan beskrivit. De ser ut så här:

A1) **(a v a) → a**
A2) **a → (b v a)**
A3) **(a v b) → (b v a)**
A4) **(a → b) →[(c v a) → (c v b)]**

För att kunna härleda teorem ur dessa axiom använda Russell endast tre grundregler. Av dessa har läsaren redan bekantat sig med *modus ponens*. Dessutom använde han *substitutionsregeln* som säger att en bokstav A kan ersätta en annan förutsatt att detta utbyte sker på varje ställe i formeln där A förekommer. Genom denna regel kan man t.ex. från A3) härleda **(c v d) → (d v c)**. Den sista regeln säger att ekvivalenta formler kan ersätta varandra.

Dessutom använde han följande definitioner. Uttrycket = df utläses: "är enligt definition detsamma som".

D 1) **a →b = df -a v b**
D 2) **a & b = df -(-a v -b)**
D 3) **a ↔ b = df (a → b) & (b → a)**

Här ett enkelt exempel på ett bevis.
Bevisa att **(a → -a)** → **-a** är ett teorem.
Bevis
Vi utgår från axiom A1). Genom att tillämpa substitutionsregeln ersätter vi alla förekoster av **a** med **-a.** Vi får **(-a v -a)** → **-a**. Vi tillämpar D1) på disjunktionen inom parentesen och får ett uttryck som innehåller → i st.f. **v,** dvs **(--a** → **-a).** Men två negationstecken efter varandra tar ut varandra. Så vi får **(a** → **-a).** Enligt ekvivalensregeln får vi nu sätta in denna implikation i st.f. för disjunktionen. Därmed har vi kommit fram till det teorem som skulle bevisas. Q.E.D.

Denna bevismetod har flera nackdelar. Den viktigaste är att det krävs stor skicklighet för att kunna använda den. Man måste kunna en stor mängd formler utantill och ha klara begrepp om hur man skall kunna, steg för steg, komma fram till önskat resultat. Självklart får man utnyttja teorem som redan bevisats i de nya bevis man konstruerar. Men för att göra detta måste man komma ihåg vad man redan bevisat. Det finns sålunda inga enkla regler att följa och allra minst någon mekanisk procedur som ger korrekta resultat. Anta att vi vill veta huruvida en viss formel är sann, dvs ett teorem, i SL. Vi bör då söka finna ett bevis för formeln. I beviset får vi förstås använda axiomen, definitionerna och tidigare bevisade teorem samt reglerna. Anta att vi inte lyckas komma på något bevis. Betyder detta att formeln inte är ett teorem? Inte alls. Det kan lika gärna vara så att vår förmåga att finna ett bevis inte räcker till.

7.4. BEVIS MED HJÄLP AV SANNINGSTABELLER.

Det innebar därför ett stort framsteg när flera logiker oberoende av varandra under 1920-talet utvecklade en helt ny bevismetod för SL. Den nya metoden innebar att man bevisar genom att göra upp tabeller över alla möjliga kombinationer av sanningsvärden. Vi kallar den *tabellmetoden*. Med denna metod behöver vi varken axiom eller slutledningsregler. Skillnaden mellan axiom och teorem försvinner. Metoden gör det möjligt att, genom en mekanisk räkneoperation, för varje (ändlig) formel avgöra om den är en logisk sanning eller inte.

En SV-tabell för en formel i SL består av alla kombinationer av sanningsvärden som är möjliga för formeln i fråga. *I denna metod fungerar grundtabellerna för konnektiverna på samma sätt som axiomen i den axiomatiska metoden*. De anger de exakta villkor under vilka en formel innehållande ett visst konnektiv är sann resp. falsk. Låt oss ta implikationen **a** → **b** som exempel. Den innehåller två satsbokstäver (variabler). För två satser finns det exakt fyra möjliga kombinationer av sanningsvärden: båda är sanna, den första sann och den andra falsk, den första falsk och den andra sann, och båda falska. Skrivet i tabellform får vi två kolumner och fyra rader.
SS
SF
FS
FF
Om vi har tre variabler som i formeln **a & b & c** så ökar antralet möjliga kombinationer till exakt åtta. Nämligen
SSS
SSF
SFF
SFS
FSF
FFS

FSS
FFF
Om vi har fyra variabler får vi igen en fördubbling av antalet möjliga kombinationer till sexton. För fem variabler blir det följaktligen 32 rader i tabellen osv. Tabellerna blir snabbt så stora att de är helt ohanterliga för en människa. För en dator går det däremot förstås blixtsnabbt att göra upp tabeller med många variabler. Metoden är alltså i praktiken ohanterlig, men principiellt är den ytterst elegant och filosofiskt är den enkel och lätt att förstå.

Denna metod innebar att Leibniz´ vision om att mekaniskt kunna räkna ut svaret på ett problem förverkligades. Visserligen blott för en mycket enkel och begränsad logik. *Med SVT kan man nämligen för varje problem, som kan formuleras i SL, mekaniskt räkna ut lösningen*. Det tråkiga är bara att mycket få verkligt intressanta filosofiska problem kan formuleras i SL!

Exempel: Bevisa att Russells axiom A2) faktiskt är sant i SL. Vi gör upp SVT för axiomet. Det innehåller två variabler. Vi får alltså fyra rader i tabellen. Sedan räknar vi ut SV för parentesen **(a v b).** Dessa värden ser vi rakt under **v.** Vi har på s. 89 sett att en disjunktion får värdet F omm både **a** och **b** har värdet F, annars S. Slutligen räknar vi ut SV för implikationen. Den får, som vi sett värdet F omm **a** har värdet S och **b** värdet F, annars S. Värdena ser vi rakt under pilen. Tabellen ser ut så här.

a b	**a → (b v a)**	
SS	S	S
SF	S	S
FS	S	S
FF	S	F
12	3	4

Jag har numrerat kolumnerna (längst ner) för att kunna uttrycka mig kort. Först räknade vi ut kolumn 4 genom att kombinera värdena i kolumn 1 och 2 med hjälp av regeln för disjunktion. (Vi börjar alltid räkna kolumnen för den innersta parentesen och går sedan utåt). Sedan räknade vi ut kolumn 3 genom att kombinera värdena i kolumn 1 med de värden vi fått i kolumn 4. Då använder vi förstås regeln för implikation.

Kolumn 3 innehåller endast S. Vad betyder det? Det betyder att vilka sanningsvärden de enkla formlerna **a** och **b** än har så blir formeln **a → (a v b)** alltid sann. Det är m.a.o. omöjligt att denna formel är falsk. Formeln är giltig, en nödvändig sanning, en tautologi. På samma sätt kan vi enkelt bevisa att de övriga axiomen också är giltiga. Läsaren bör försöka göra detta för axiom A1) och A3).

Med denna metod är det mycket enkelt att bevisa teoremet som vi ovan bevisade axiomatiskt. Teoremet innehåller endast en bokstav. Det finns alltså endast två möjligheter och tabellen blir denna.

a	**(a → -a) → -a**	
S	F	S
F	S	S
1	2	3

Först räknar vi ut värdena inne i parentesen, dvs kolumn 2, och sedan kolumn 3. Den innehåller endast S vilket betyder att formeln blir sann oberoende av om **a** är sann eller falsk. Formeln är alltså nödvändigt sann.
Låt oss ännu bevisa att A4) är en logisk sanning. Formeln innehåller tre variabler. Vi får därför åtta rader i SVT.

a b c	**(a → b) → [(c v a) → (c v b)]**				
SSS	S	S	S	S	S

SSF	S	S	S	S	S
SFF	F	S	S	F	F
SFS	F	S	S	S	S
FSF	S	S	F	S	S
FFS	S	S	S	S	S
FSS	S	S	S	S	S
FFF	S	S	F	S	F
123	4	5	6	7	8

Vi börjar inifrån och går utåt. Först får vi kolumnerna 6 och 8 genom att använda regeln för disjunktion. När vi kombinerar värdena i 6 och 8 genom regeln för implikation får vi kolumn 7. Denna kolumn ger SV för formeln inom klammern. Sedan räknar vi ut kolumn 4. Och slutligen kombinerar vi kolumnerna 4 och 7 genom regeln för implikation och får kolumn 5. Denna kolumn, som innehåller värdena för den sammansatta formeln, består endast av S, vilket betyder att denna formel blir sann oberoende av hur vi tolkar **a,b,c**, dvs oberoende av vilka SV variablerna har.

Men vi kan göra mycket mera med tabellmetoden än bevisa att en formel är en logisk sanning. Genom att studera slutkolumnen för en formel kan vi dessutom avgöra om formeln är en motsägelse (en kontradiktion, nödvändigt falsk, en logisk falskhet), om den är satisfierbar, och om den är kontingent. Följande regler gäller.
Om slutkolumnen för en formel innehåller endast S så är formeln en logisk sanning.
Om slutkolumnen för en formel innehåller endast F så är formeln en logisk falskhet.
Om slutkolumnen för en formel innehåller både S och F så är formeln kontingent.
Om slutkolumnen för en formel innehåller minst ett S så är formeln satisfierbar.

Slutligen kan vi använda tabellmetoden för att avgöra om en slutledningsregel (argumentform) är giltig eller inte. Vi definierade giltighet i avsnitt 3.7. *Definitionen säger att en regel är giltig omm det inte kan inträffa att alla premisser är sanna, men slutsatsen ändå falsk.* Om vi finner en sådan tilldelning av SV att premisserna är sanna, men slutsatsen ändå falsk, så har vi ett motbevis (ett motexempel).

Det är enkelt att söka motbevis genom tabellmetoden. Metoden innebär ju att vi går igenom alla möjliga kombinationer av SV. Om vi inte kan finna någon rad där alla premisser är sanna, men slutsatsen falsk, så har vi bevisat att regeln är giltig.

Jag har tidigare flera gånger nämnt två av de viktigaste logiska reglerna, nämligen *modus ponens* och *modus tollens*. Nu skall vi visa att dessa regler är giltiga i SL med hjälp av en SV-tabeller. Vi börjar med tabellen för modus ponens. Premisserna är **a → b** och **a.** Slutsatsen är **b.** Vi skriver dessa formler efter varandra och gör upp tabellen.

a b	**a → b**	**a**	**b**	
SS	S	S	S	1
SF	F	S	F	2
FS	S	F	S	3
FF	S	F	S	4
1 2	3	4	5	

För att kunna uttrycka mig så klart som möjligt har jag numrerat både rader och kolumner. Ett motbevis är en rad där alla premisser är sanna men slutsatsen falsk. Vi ser att det inte finns någon rad där detta gäller. Alltså finns det inga motexempel, och regeln är giltig. Vi ser att båda premisserna är sanna endast på rad 1, men på denna rad är också slutsatsen sann. Detta betyder att så fort vi sätter in sådana SV i modus ponens att premisserna blir sanna så blir också slutsatsen sann. Regeln är alltså sanningsbevarande.

Märk att vi inte alls behöver bry oss om de rader där en eller båda premisserna är falska, dvs raderna 2, 3 och 4. Meningen med en giltig regel är att den skall garantera att **om** alla premisser är sanna så måste också slutsatsen vara det. Med en giltig regel skall det vara omöjligt att härleda falskhet från sanning. Vad som sker om minst en av premisserna är falsk är därför likgiltigt. Men vi ser av tabellen att slutsatsen i så fall kan vara falsk (rad 2) eller sann (rad 3, 4).
Vi fortsätter med att bevisa att regeln *modus tollens* är giltig i SL. Regeln säger att av premisserna **a → b och -b** följer slutsatsen **-a**. Man kan också skriva regeln så här
a → b
<u>- b</u>
-a
Vi gör upp SVT för premisserna och slutsatsen

a b	**a → b**	**-b**	**-a**
S S	S	F	F
S F	F	S	F
F S	S	F	S
F F	S	S	S

Vi ser att den enda rad där båda premisserna är sanna är den sista. Men i denna rad är också slutsatsen sann. Det kan alltså inte existera något motexempel. Q.E.D.

Den uppmärksamma läsaren undrar kanske vad som händer om det inte alls finns någon rad där båda premisserna är sanna. Låt oss undersöka. Vi testar om följande regel är giltig.
-(a → b)
<u>-a v b</u>
a & b Vi gör upp SVT.

a b	**-(a**	**→ b)**	**-a**	**v b**	**a & b**
S S	F	S	F	S	S
S F	S	F	F	F	F
F S	F	S	S	S	F
F F	F	S	S	S	F
	1	2	3	4	5

Först räknar vi ut kolumn 2. Eftersom vi har ett negationstecken framför parentesen ändrar vi värdena i kolumn 2 till de motsatta och får kolumn 1. Detta är kolumnen för den första premissen. I kolumn 3 har vi kastat om värdena för **a**. Kolumn 4 får vi genom att kombinera kolumn 3 med kolumnen för **b**. När vi jämför kolumn 1 (den första premissen) med kolumn 4 (den andra premissen) märker vi något intressant. Vaddå? Jo att det inte finns någon rad där båda premisserna är sanna. Tabellen visar att det är omöjligt att tilldela SV till formlerna så att båda premisserna blir sanna. Alltid när den ena blir sann, blir den andra falsk. Detta betyder att premisserna motsäger varandra. Tillsammans bildar de en kontradiktion, en logisk motsägelse. Vi säger att mängden av premisser är *inkonsistent*.

Och nu följer en viktig, men egendomlig slutsats. En regel är giltig omm det inte finns något motexempel, dvs någon rad där båda premisserna är sanna och slutsatsen falsk. I denna tabell finns ingen rad där båda premisserna är sanna. Då kan det ju inte heller finnas någon rad där båda premisserna är sanna och slutsatsen falsk. *Enligt denna formella definition av giltighet är regeln ovan alltså giltig!!* Vi ser att det inte spelar någon roll vilken formel vi har som slutsats. Vad vi än har som slutsats kan det inte finnas något motexempel och regeln är giltig. Inom formell logik gäller det sålunda att
ur en motsägelse kan man härleda vilken formel som helst!!

Detta gäller förstås inte inom intuitiv logik. Om någon säger att det ur påståendena *Åbo är en stad* och *Åbo är inte en stad* följer att *Månen är en ost* så tar vi detta som ett skämt. Men i SL och i formell logik generellt är det blodigt allvar. Vi har just bevisat att man i SL kan härleda vilken formel som helst ur en motsägelse. Eftersom SL är grundläggande för all mer komplicerad formell logik så gäller detta generellt för formella system. Vad lär vi oss av detta? Borde vi försöka skapa en formell logik där detta inte gäller? Det går inte därför att vi i så fall måste utesluta disjunktionen ur systemet och då blir det ingenting av något värde kvar. Slutsatsen blir i stället att *när man arbetar med formella system måste man vara mycket noga med att inga motsägelser finns med i systemet.*

Läsaren bör nu försöka lösa följande uppgifter.
Avgör genom tabellmetoden för var och en av formlerna om den är a) nödvändigt sann, b) nödvändigt falsk, eller c) kontingent.
1) $\mathbf{(a \leftrightarrow b) \rightarrow -(a \,\&\, -b)}$
2) $\mathbf{-(-a \lor b) \leftrightarrow (a \rightarrow b)}$
3) $\mathbf{(a \rightarrow b) \lor (a \,\&\, b)}$
4) $\mathbf{(a \rightarrow b) \,\&\, (-c \rightarrow -b) \rightarrow (a}$
5) $\mathbf{(a \,\&\, b \,\&\, c) \rightarrow (a \lor b \lor c)}$
Avgör genom tabellmetoden vilka av följande slutledningsregler som är giltiga.

1) **a → b**	2) **a ↔ b**	3) **a v b**	4) **-a → -b**
b	**-a**	**b v c**	**-(-b v a)**
a	**-b**	**a & b**	**a & b &c**

7.5. BEVIS GENOM SANNINGSTRÄD.

Detta avsnitt saknas tillsvidare.

7.6. EGENDOMLIGHETER I SL

Det finns många egendomligheter t.o.m. i ett så enkelt system som SL. (I logikböckerna talar man mycket litet om detta). Det lilla ordet *och* betecknas i SL med **&**. Men det finns många skillnader mellan **&** och *och*. (Det gäller att hänga med!) Vi tar som exempel satsen “Pelle öppnade dörren och steg in”. Vi låter **p** vara “Pelle öppnade dörren” och **s** “Pelle steg in”. Motsvarande formel i PL ser då ut såhär **p & s**. Men enligt reglerna i PL är **p & s** exakt detsamma som **s & p**. Formlerna är ekvivalenta. Detta stämmer inte med intuitiv logik för satsen “Pelle steg in och öppnade dörren” betyder något helt annat än satsen som vi utgick ifrån. Detta betyder att ordet *och* i denna sats inte överhuvudtaget kan formaliseras i SL. I själva verket finns det en hel rad användningar av *och* som inte kan formalieras. Ett annat exempel är “Pelle och Oskar lyfte upp kanoten”. Enligt reglerna för SL kan man av denna sats sluta sig till att “Pelle lyfte upp kanoten” vilket ju är fel. Inte heller denna sats kan sålunda formaliseras korrekt i SL. Den intuitiva logiken för det lilla ordet *och* är alltså mycket mer komplicerad än för **&** i SL.

De formella systemen, inklusive matematiken, är mycket enkla i jämförelse med vanliga språk och inituitivt tänkande. Anledningen till att vi uppfattar dem som svåra är att vår hjärna inte är någon datamaskin utan fungerar enligt helt andra principer. Vårt tänkande är alltså av annan typ än datorernas sätt att processa information. Det är enkelt att skriva ett program som spelar schack bättre än de flesta vuxna människor, men mycket svårt att skriva ett program som kan föra ett helt tvivialt samtal av den typ som ett barn utan minsta besvär klarar av.

Jag skall nämna ytterligare ett par exempel på hur SL, och givetvis alla system som innehåller SL, skiljer sig från intuitiv logik. Som exempel tar vi satsen "Om Åbo är en stad så är månen en ost." Är denna sats sann eller falsk? Eller är den rent av meningslös? Enligt SL är den sann! Men också satsen " Om Åbo inte är en stad så är månen en ost" är sann enligt SL. Detta följer av den SV-tabell som definierar → . Enligt intuitiv logik är dessa satser nonsens. Det finns inget logiskt samband mellan Åbo och månen. Enligt intuitiv logik kan endast saker som har något att göra med varandra kombineras genom om-så. Detta kan vi kalla *relevansregeln*. Den gäller stenhårt i intuitiv logik, men inte alls i formell logik.

Men kan man inte införa en sådan regel? Nej, därför att det strider mot grundidéen i formell logik, nämligen att man bara studerar formen och ignorerar innehållet. Man kan avgöra om det finns något slags relevant samband mellan två satser bara om man förstår innehållet. Eftersom en maskin fungerar enligt ett formellt system och eftersom formella system inte kan förstå mening och därför inte innehålla relevansregler följer det att maskiner inte kan avgöra om utsagor är relevanta i ett visst sammanhang eller inte. En maskin som "talar" kan därför gladeligen säga helt fel saker i sammanhanget. Att konstruera en digital maskin baserad på formell logik som verkligen förstår vad den säger förefaller vara omöjligt.

I intuitiv logik spelar motsägelser en viktig roll därför att de visar att någonting är fel. Om en person motsäger sig själv förstår vi att han måste missta sig eller ljuga. Men i formell logik har motsägelser, som vi bevisade i föregående avsnitt, en ytterligare konsekvens som förefaller helt absurd ur intuitiv synvinkel. Om en person motsäger sig själv, t.ex. genom att påstå att han var hemma vid ett visst klockslag och en stund senare förneka detta, så inser vi att man inte kan dra några andra slutsatser av detta än just att han ljuger eller misstar sig. Men i formell logik kan man från en motsägelse logiskt korrekt, dvs. enligt reglerna, inom systemet härleda *vilket påstående som helst*, t.ex. att världen går under i morgon.

Denna egendomlighet kan, som vi såg, inte elimineras utan att SL utarmas så mycket att det blir ointressant. Men SL ingår i alla mer omfattande system. Därför gäller att om ett formellt system innehåller motsägelser så kan vilken formel som helst inom systemet bevisas. Därför är motsägelser "dödligt farliga" i varje formell logik. I intuitiv logik kan man däremot inte härleda något alls från en motsägelse.