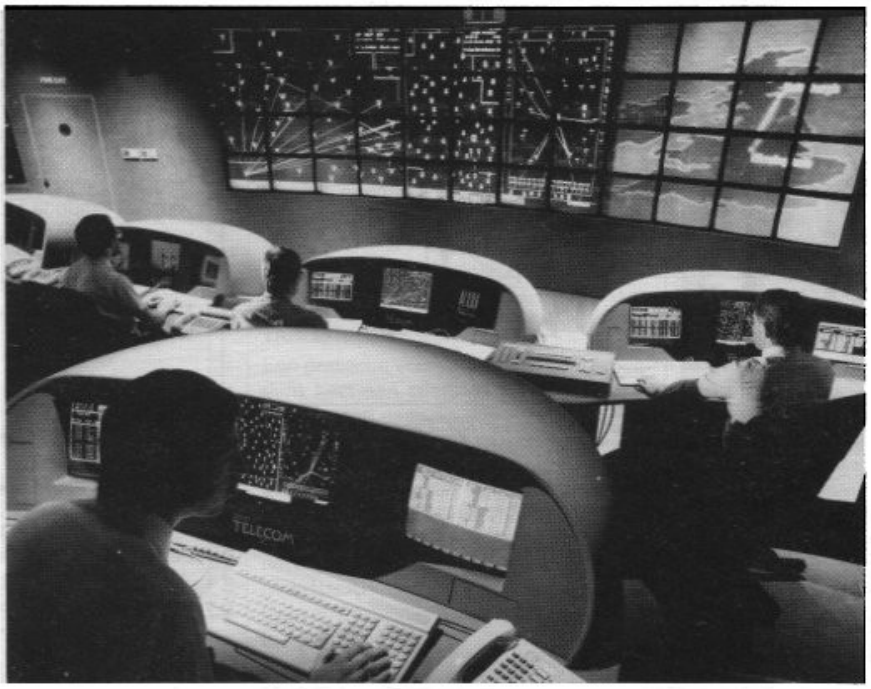
ir. J.M. VAN THIEL



Formele specificatie speelt een belangrijke rol in de telecommunicatie bij onder meer het beschrijven van communicatieprotocollen.

Formeel formuleren

Deel 2: Specificeren, predikatenlogica en betekenis

Dit is het tweede en laatste deel in een korte serie over formeel specificeren met behulp van mathematische logica. Centraal hierbij staat de vraag hoe een systeem zo goed mogelijk en op voorhand kan worden beschreven. De huidige formele specificatiemethoden en -talen vormen hier een uitvloeisel van. Zij zijn gebaseerd op de mathematische logica en worden niet alleen gebruikt voor het beschrijven van systemen maar ook in de theoretische informatica en bij de berekenbaarheidstheorie. Deze takken van de logica komen in deze serie niet aan de orde, evenmin als de diverse specificatiemethoden en -technieken zelf. Waar het hier vooral om gaat zijn de theoretische en mathematische uitgangspunten die aan deze methoden ten grondslag liggen. Zo zijn in het vorige artikel enkele logische principes geschetst die van belang zijn voor formeel specificeren. Dit tweede deel behandelt de elementen van het beschrijven zelf. Hierbij spelen verschillende niveaus van interpretatie. Het is bijvoorbeeld noodzakelijk dat de specificatie niet alleen syntactisch juist is maar dat hij ook inhoudelijk klopt. Predikatenlogica is een hulpmiddel om de geldigheid van beweringen te controleren. Problemen die daarbij optreden zijn dat ook correcte beschrijvingen nog tegenstrijdigheden kunnen bevatten of onbeslisbaar kunnen zijn. En dan is er altijd nog het probleem van de betekenis: bevat de specificatie, hoe correct en consistent ook, geen volstrekt zinloze uitspraken?

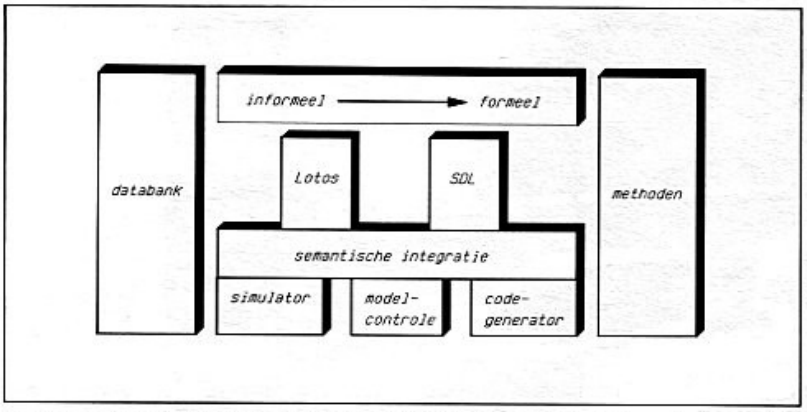
Formeel specificeren houdt in dat het te ontwikkelen systeem als een mathematische structuur wordt opgevat. In het meest algemene geval is dat een klasse objecten met diverse relaties tussen die objecten. In zijn algemeenheid kunnen dus ook fysische systemen en zelfs computerprogramma's als mathematische structuren worden onderzocht.

In de in het eerste deel beschreven voorbeelden ging het om algebraïsche specificaties. Daar bestaan de structuren uit verzamelingen, functies en constanten terwijl de beweringen vergelijkingen zijn die, met behulp van de variabelen, uitdrukken dat voor alle elementen bepaalde functiewaarden gelijk zijn aan andere. De interpretaties van dergelijke formuleringen heten *algebra's*.

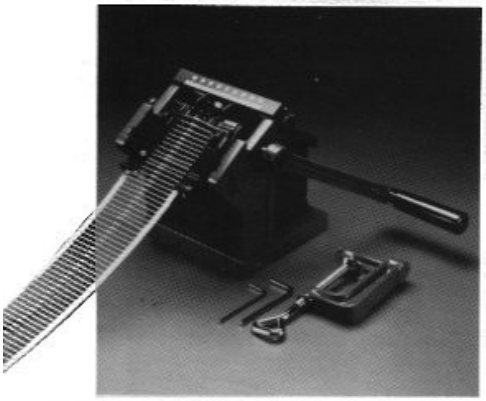
In de logica heet de verzameling van alle ware uitspraken over een structuur de *theorie* van die structuur. Bij algebraïsche specificaties wordt gesproken, in iets andere zin dus dan bij een afleidingsstelsel, van compleetheid als de theorie van de algebra uit de vergelijkingen volgt. Dat is, zoals in de inleiding al werd gesteld, het uiteindelijke doel van formeel specificeren.

INCOMPLEET

Het voorbeeld in het vorige artikel over de karakterisering van weerstanden – door middel van '@' voor een weerstandsloze draad, '#' voor de verbinding in serie en '$' voor de parallelschakeling – schiet behoorlijk tekort op het punt van compleetheid. Er is bijvoorbeeld geen mogelijkheid om uit te drukken dat er voor elke schakeling precies één weerstand bestaat die ▷



Het Race-project Specs is gebaseerd op de specificatietalen Lotos en SDL. Een tussentaal verzorgt de semantische integratie en hierbij wordt als formalisme procesalgebra gebruikt.



In het vorige artikel is gebleken dat de gepresenteerde algebra voor het karakteriseren van weerstanden niet compleet is.

Bij de interpretatie van het voorbeeldstelsel als systeem voor digitale combinatorische schakelingen is de vergelijking x # x = x wel juist, in tegenstelling tot het weerstandsmodel.

▷ hieraan gelijkwaardig is, laat staan welke weerstand dit is.

Er is nog op een andere manier in te zien dat het systeem niet volledig is. Dat kan met een ander model van dezelfde verzameling axioma's, namelijk een digitale combinatorische schakeling met EN-poorten, OF-poorten en invertoren. De OF-operatie wordt nu gerepresenteerd door '#', de EN door '$' terwijl '@' in dit geval staat voor de digitale '0'. Hier kan \@ eenvoudig worden geïnterpreteerd; dat is zelfs essentieel. Het gaat hier echter om een heel andere structuur, waarvoor bijvoorbeeld nog steeds geldt (zie figuur 1):

$$x \$ y = \((\x) \# (\y))$$

maar ook:

$$x \# x = x$$

en dat is bij het model van de weerstanden beslist onjuist. Die eigenschap is dus geen

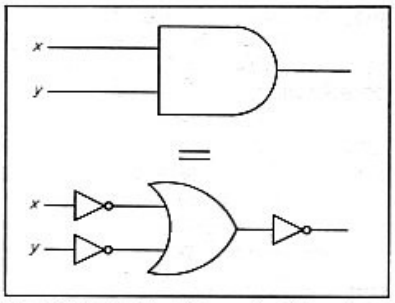


Fig. 1. Ook als het model wordt geïnterpreteerd als model voor digitale poorten is '$' nog steeds equivalent met '#' en dubbele inversie.

logische consequentie van de axioma's en die vormen daarmee dan ook geen volledige specificatie.

GELIJKHEID

Formeel specificeren is een wetenschap op zich. Het vakgebied kan tevens modulariteit en parametrisering kan omvatten maar ook die aspecten liggen niet ver van de mathematische logica af (veelsoortige logica bij modulariteit).

Een stelsel vergelijkingen zoals hierboven beschreven is een speciaal geval van de eerste-orde predicatenlogica met gelijkheid. De vergelijkingen gelden immers voor willekeurige x, y en z en daarmee voor alle x, y en z. Die variabelen heten dan *universeel gekwantificeerd*. Als 'x # y' geschreven wordt als P(x,y), is:

x # @ = x

te schrijven als:

$$\forall x\,(P(x,@) = x)$$

en dat kan worden geïnterpreteerd (in woorden) door:

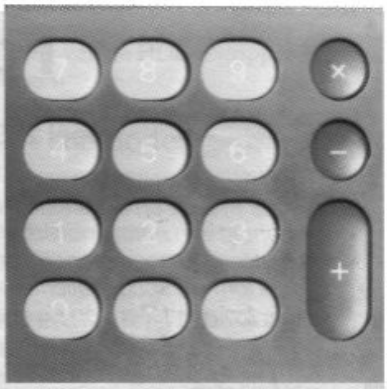
> Voor alle weerstanden x is de serieschakeling van x met weerstandsloze draad @ gelijk aan x.

Een andere interpretatie zou kunnen zijn:

> Voor alle getallen x is de som van x met het getal 0 gelijk aan x.

En weer een andere:

> Voor alle getallen x is het produkt van x met het getal 1 gelijk aan x.



Onafhankelijk of de interpretatie nu weerstanden of getallen betreft, kan predikaat P een eigenschap uitdrukken.

Wat de letter P nu precies uitdrukt is afhankelijk van de interpretatie maar blijkbaar is het steeds de een of andere eigenschap, een predikaat.

PREDIKATENLOGICA

Hoewel de interpretatie van een formalisme volledig vrij staat – het hoeft grof gezegd niet eens te kloppen – is het toch wel zo dat al die mogelijke betekenissen iets met elkaar gemeen hebben. De signatuur van een algebra bestrijkt alleen functies en constanten en de vergelijkingen hebben ook een bepaalde vorm.

Voor de eerste-orde predikatenlogica is dat niet anders. Het gaat hier steeds om objecten (welke dan ook) en eigenschappen

(eveneens willekeurige) van die objecten. Operaties kunnen een- of meerplaatsig zijn; P in het voorbeeld boven is binair.
Vervolgens kunnen beweringen gevormd worden, zoals:

> Als er een x is waarvoor geldt dat x de eigenschap A heeft of dat x de eigenschap B heeft dan is er een x met de eigenschap A, of er bestaat een x met de eigenschap B.

In formule wordt dat:

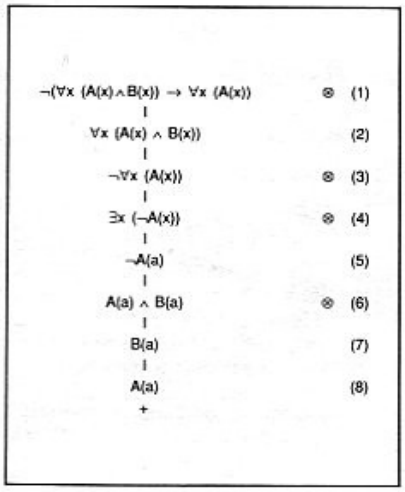
$$\exists x\,(A(x)) \vee B(x)) \rightarrow \exists x\,(A(x)) \vee \exists x\,(B(x))$$

In het algemeen kunnen dergelijke zinnen waar of onwaar zijn, afhankelijk van de interpretatie die er aan wordt gegeven, maar deze zin is altijd waar, om welke objecten en welke predikaten het ook gaat. Overigens is het omgekeerde hier niet altijd waar en evenmin altijd onwaar.

KWANTIFICEREN

Beweringen die altijd opgaan, voor welke interpretatie dan ook, heten *tautologieën*. De bewering dat, als er een object bestaat met de eigenschap A, dit op hetzelfde neerkomt dat niet voor alle objecten niet-A geldt, is eveneens een tautologie. Dergelijke algemene waarheden kunnen natuurlijk ook worden gebruikt in redeneringen met specifieke structuren (zoals boven) als interpretatie.
In die zin behoren de ware uitspraken van de logica, de theorie ervan, tot het het gereedschap bij formeel formuleren. Het gebruik van 'Voor alle..' en 'Er is..' heet *universeel kwantificeren* respectievelijk *existentieel kwantificeren*.
Om nu uit te zoeken of een bepaalde formule een tautologie is of niet, is het weer nodig om naar de semantiek te kijken.

Fig. 2. Voorbeeld van een semantisch tableau.



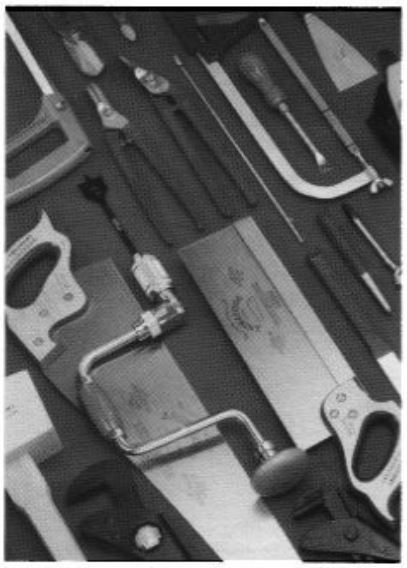
Geldt de bewering voor alle objecten en eigenschappen, welke dan ook? Zo ja, hoe valt dat dan aan te tonen, voor het oneindig grote aantal interpretaties?
Het is in elk geval zo dat als de ontkenning van die bewering altijd onwaar is, de bewering zelf altijd waar is. Op basis hiervan kan worden geprobeerd om te laten zien dat zo'n ontkenning zowel een bepaalde zin als zijn tegendeel tot gevolg heeft. Is dat het geval dan is de uitgangszin blijkbaar een tautologie.

TABLEAU

Een grafische voorstelling van zo'n procedure heet een semantisch tableau en figuur 2 geeft daarvan een voorbeeld. Hier wordt geprobeerd aan te tonen dat indien alle objecten twee eigenschappen hebben, alle objecten ook een van die eigenschappen bezitten (tautologieën zijn uit de aard van de zaak vaak triviaal). Neem aan dat het tegendeel het geval is dan wordt het uitgangspunt dus dat het gedeelte voor de pijl klopt en de conclusie niet.
In figuur 2 begint dat door bewering (1) te vervangen door (2) en (3) en zin (1) te markeren teneinde aan te geven dat die niet meer nodig is. Als niet voor alle x predikaat A geldt, is er dus een x waarvoor niet-A(x) geldt: vervang (3) door (4). Misschien is die x, als willekeurig voorbeeld, het element a: vervang (4) door (5). Daarboven staat echter dat alle x eigenschappen A en B hebben, dus ook de a die net is gekozen (uitspraak 6).
Misschien komen er verderop nog meer voorbeelden behalve element a, dus (2) moet blijven staan. Bewering (6) wordt echter vervangen door (7) en (8). Hiermee, namelijk tussen (5) en (8), is de strijdigheid gevonden en dat wordt aangegeven met de + aan het eind van het tableau.
Het algemene principe van de procedure is tweeledig en omvat enerzijds het kiezen van een willekeurig element als voorbeeld voor de variabelen en anderzijds het splitsen van samengestelde zinnen in hun componenten. Dat laatste berust op de regels van de propositielogica die op het laagste niveau alleen zinnen kent en als interpretatie alleen waarheidswaarden.

BETEKENIS

Het is heel gebruikelijk om een verzameling te definiëren door middel van een eigenschap, bijvoorbeeld de verzameling oneven natuurlijke getallen of de verzameling handgereedschappen. Als dit altijd zou kunnen, zou de predikatenlogica eigenlijk samenvallen met verzamelingenleer.
De gelijkstelling van predikaat met verzameling leidt echter tot paradoxen die het noodzakelijk hebben gemaakt om dit op zich zeer aantrekkelijke idee los te laten.



Verzamelingen worden vaak gedefinieerd op basis van een eigenschap, bijvoorbeeld die van handgereedschappen.

Deze paradoxen ontstaan als gevolg van de mogelijkheid tot recursie: een klasse objecten kan namelijk dezelfde eigenschap hebben als de objecten zelf. Zo kan een groep programmaprocedures heel goed zelf een programmaprocedure zijn.
Meestal is zoiets niet het geval; een verzameling gereedschappen is geen gereedschap. De onmogelijkheid ontstaat wanneer een verzameling wordt gedefinieerd door de eigenschap dat de elementen daarvan verzamelingen zijn die zichzelf niet als element hebben. Deze verzameling is zelf zowel wel als niet element van zichzelf (Russell's paradox). In formule:

$$x \in x \leftrightarrow x \notin x$$

VERZAMELINGEN

Deze en andere soortgelijke paradoxen die aan het begin van de twintigste eeuw werden gevonden, hebben geleid tot een herziening van het concept *verzameling*. In de herziene formulering wordt veel voorzichtiger omschreven wat een verzameling is. Het grondprincipe is dat een verzameling wordt samengesteld uit zijn elementen en dat hij dus niet beschikbaar kan zijn als element voordat die opbouw is voltooid.
Een verzameling kan dus geen element zijn van zichzelf. De consequentie hiervan is dat er entiteiten zijn, zoals de klasse programma's, die geen verzameling vormen. In dergelijke gevallen worden de termen *klasse* en *collectie* gebruikt. Het laatste woord is hiermee uiteraard niet gezegd; verzamelingenleer bestrijkt de studie van dit hele gebied en is zelfs een aparte tak van de logica. ▷

▷ Een tweede punt dat aan de orde komt bij de semantiek van predikatenlogica is de *beslisbaarheid*. Het is niet ondenkbaar dat bepaalde formules aanleiding geven tot een semantisch tableau dat oneindig lang kan worden. In zo'n geval is de uitgangsformule onbeslisbaar. Er ontstaat nooit tegenstrijdigheid maar het is ook niet op mechanistische wijze uit te maken dat er geen tegenstrijdigheden inzitten. Soortgelijke vraagstukken zijn ook aan te treffen bij de Turing-machine. Als gevolg van de complexiteit van het onderwerp en het belang ervan, beslaat de problematiek met betrekking tot de berekenbaarheid een aanzienlijk deel van het onderzoek naar formele methoden.

HOGERE ORDEN

In de eerste-orde predikatenlogica wordt alleen gekwantificeerd over de objecten. In hogere-orde logica's gebeurt dit ook ten aanzien van natuurlijke getallen, relaties en relaties van relaties. Gelijkheid (' = ') is zo'n belangrijke eigenschap dat die wel aan de predikatenlogica wordt toegevoegd. Is dat het geval dan wordt de aanduiding *predikatenlogica met gelijkheid* gehanteerd.
Het blijkt dat hogere-orde logica's in bepaalde gevallen absoluut noodzakelijk zijn om bepaalde mathematische structuren te specificeren. Sommige definities zijn bijvoorbeeld gewoon niet mogelijk zonder kwantificatie over getallen. Anderzijds zijn deze logica's en is ook het predikaat 'gelijkheid' wel weer te baseren op de eerste-orde variant, al zou dat in de praktijk ondoenlijk zijn.

Termen

Zowel de mathematische logica als de specificatieleer kennen een groot aantal technische uitdrukkingen. Hieronder volgt een (losse) verklaring van de in deze artikelen gebruikte termen.

Interpretatie: De betekenis die aan een formeel stelsel wordt gegeven. Er zijn altijd verschillende interpretaties mogelijk.

Model: Een interpretatie waarvoor de zinnen van het formele stelsel gelden. Zij zijn waar. Een stelsel zinnen kan meerdere modellen hebben.

Consistent: Een zin of verzameling zinnen is consistent als er een model voor bestaat. In andere woorden: de zinnen zijn niet per sé altijd onwaar voor welke interpretatie dan ook.

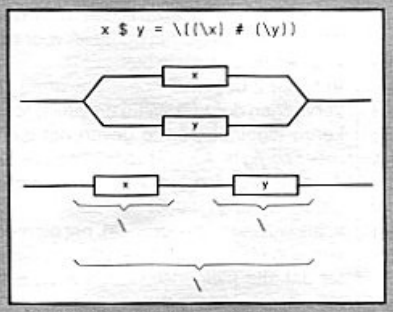
G ⊨ S: De bewering s geldt voor alle modellen van de verzameling zinnen G, al zit s zelf niet in de verzameling G. De bewering s is een logische consequentie van G.
Ook: s is een semantische consequentie van G.
En eveneens: G maakt s waar.

Afleiding: Een constructie volgens welbepaalde regels van een zin uit (andere) zinnen. De regels behoren tot een formeel deductief systeem (FDS) en er zijn meerdere FDS'en mogelijk. Een afleiding doet geen beroep op de diverse mogelijke interpretaties.

G ⊢ s: De bewering s is afleidbaar uit de verzameling zinnen G. Afleidbaarheid kan afhangen van het toegepaste FDS.

Soundness: Correctheid, soliditeit. In een solide stelsel geldt, als s afleidbaar is (uit G), dat s dan ook een logische consequentie is (van G).
Soliditeit van een algebraïsche specificatie betekent dat de vergelijkingen en alle logische consequenties gelden voor de hiermee te specificeren algebra.

Compleetheid: Ook: volledigheid. Als s een logische consequentie is van G dan is s (in het betreffende FDS) afleidbaar uit G.



De voorbeeldalgebra uit het eerste deel is een minimale en solide Σ-algebra maar wel incompleet.

Compleetheid van een algebraïsche specificatie betekent ruwweg dat alle relevante informatie over de algebra met behulp van de algebraïsche 'rekenregels' uit de specificatie is te halen. Er bestaan echter onderscheidingen tussen diverse soorten compleetheid. De betekenis van de twee begrippen verschilt dus in meerdere opzichten van de gebruikelijke uit de logica.

Adequaat: Correct en compleet:
G ⊨ s dan en slechts dan als G ⊢ s.

Structuur: Niet nader omschreven mathematische term voor een wiskundig object, meestal een verzameling met 'bijbehorende' relaties.

Theorie: De theorie van een structuur is de verzameling ware beweringen over die structuur.

Functie: Gezien als verzameling: een verzameling geordende paren waarbij voor elk eerste element het tweede element uniek is.

Relatie: Hetzelfde als functie zonder de beperking van uniekheid. Een argument kan dus meerdere relatiewaarden hebben, maar slechts een functiewaarde.

∀(..x..): Voor alle x... De technische term is universele kwantificatie.

∃(..x..): Er is een x... existentiële kwantificatie. Door kwantificatie ontstaan uit formules met variabelen beweringen die waar of onwaar kunnen zijn.

Predikatenlogica: Formalisme met (algemene structuur van) objecten en eigenschappen alsmede beweringen daarover als interpretatie. In de eerste-orde predikatenlogica wordt alleen over objecten gekwantificeerd, in hogere orden logica's ook over natuurlijke getallen (tweede-orde) of over functies en relaties.

Propositie: Een bewering die waar of onwaar kan zijn. De interpretatie van een propositie is een waarheidswaarde (twee mogelijkheden).

Logische connectieven: Operaties als en, of, niet, impliceert, etcetera die uit een of meer proposities een nieuwe propositie maken. De interpretatie van het resultaat volgt uit de delen door toepassing van de bekende waarheidstabellen.

Propositielogica: Formalisme met alleen proposities (geen predikaten).

Tautologie: Uitspraak die waar is voor alle mogelijke interpretaties.

Algebra: Verzameling met functies en constanten. Beweringen over een algebra zijn opgesteld als vergelijkingen van (samenstellingen van) functies.

Signatuur: Declaratie van de verzameling, functies en constanten voor een mogelijke algebra. Afkorting Σ.

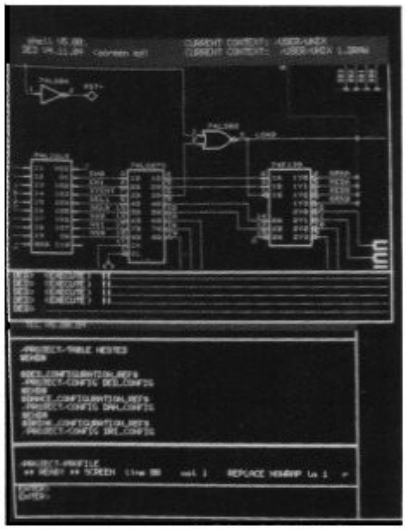
Algebraïsche specificatie: Signatuur E samen met stelsel vergelijkingen Σ.

```
File  Run  Windows  Analysis  Display  Memory  Configure

G>declare                          ; Declare the macro variables ..............
G>local  string xVerifacation      ; "Test O.K." string. Used to compare memory.
G>                                 ; location 0f000 hex after each test.
G>local  int    xPumpPressure      ; Controls the Pump Pressure for each test.
G>global int    xFinalPressure     ; Returns the final pump pressure.
G>enddeclare
G>
G>; Initialise the test verify string to "Test O.K." and pressure to 2 bar.
G>strcpy xVerifacation, "Test O.K."
G>xPumpPressure = 2
G>
G>g 0 t main
G>if ( SP == STACK_START )         ; Verify that the SP is set up correctly.
G>   g f 1s                       ; Stack is o.k. so start the test program.
G>
G>   if ( PC == PROGRAM_END )     ; Check if the PC is at the end.
G>
G>      ; Test for "Test O.K." in location 0f000h using memcmp command.
G>      memcmp 0f000h, xVerifacation
G>      if (xCOMPARE == 0)                ; "Test O.K" is at 0f000h.
G>      endif

1  ALT-F1   F2   F7   F8   F9   F10   ALT Q
```

Als een uitgangsformule onbeslisbaar is dan kan niet mechanistisch worden bewezen dat er geen tegenstrijdigheden in zitten. Dergelijke problemen doen zich vooral voor bij programmatuur en ze duiken dan ook vaak op bij abstracties als de Turing-machine.



Digitale combinatorische schakelingen kunnen uitstekend propositielogica simuleren.

Hetzelfde geldt in nog sterkere mate voor de veelsoortige logica die, zoals de naam al zegt, niet een maar meerdere domeinen als interpretatie neemt. Deze logica is bij uitstek geschikt voor het onderzoek naar modulariteit en daarmee voor specificatietheorie en informatica. Juist op het niveau van de logische fundering staat hier echter nog veel open.

FDS

Voor de eerste-orde predikatenlogica zijn verschillende formeel deductieve systemen (FDS) ontwikkeld. De ene hoofdgroep heet *Hilbert-type systemen*. Alle tautologieën, de axioma's van de gelijkheid en twee generalisatieregels worden vooronderstelt. Verder is alleen het gebruik van *modus ponens* toegestaan: als het antecedent van een implicatie bewezen wordt geacht, evenals de implicatie zelf, dan is ook de conclusie bewezen.

Gentzen-type systemen bevatten vrijwel geen axioma's maar wel een zeker aantal nauwkeurig omschreven transformatieregels die op formules mogen worden toegepast. Het bekendste is het systeem van zogenoemde *natuurlijke deductie*. Van beide type systemen is aan te tonen dat zij logisch adequaat zijn in de zin zoals in het vorige artikel beschreven (Volledigheidsstelling van Gödel).

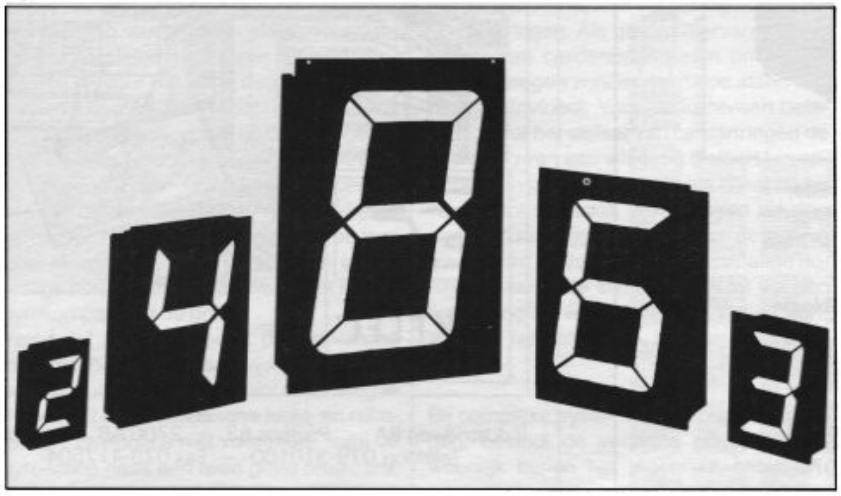
De eenvoudigste logica is de propositielogica. Hierin is elke formule een zin of een samengestelde zin, zonder kwantificatie, met slechts twee interpretatiemogelijkheden: waar of onwaar. Zinnen worden samengesteld met behulp van logische connectieven zoals 'en', 'of', 'niet', 'impliceert' enzovoort. De berekening van de waarheidswaarde gebeurt met *waarheidswaardetabellen*. De logische connectieven worden precies zo gebruikt in predikatenlogica.

Voor propositielogica bestaan formele afleidingsstelsels die logisch adequaat zijn. Propositielogica is bovendien beslisbaar. Digitale combinatorische schakelingen vormen een Boole-algebra maar zij kunnen tevens zinnen uit de propositielogica simuleren omdat de waarheidstabellen voor de logische operaties samenvallen.

In plaats van semantische tableaus of waarheidstabellen zou dus ook een de proefschakeling als verificatiemethode kunnen worden gebruikt: neem de variabelen als ingang, bouw de samengestelde zin met de betreffende poorten en kijk naar de uitgang. Als die hoog is voor alle ingangscombinaties dan is de formule een tautologie. Wordt de uitgang minstens een keer hoog wordt dan heeft de zin een model.

Is dat laatste ook niet het geval dan kan nooit aan de formule worden voldaan. De negatie van de zin is dan een tautologie. Als alle ingangen hoog zijn en de uitgang is hoog dan is de samengestelde zin een logische consequentie van de proposities op de ingangen. □

Sommige definities zijn niet mogelijk zonder kwantificatie over getallen.



Referenties:

[1.] Jon Barwise (red.): *Handbook of Mathematical Logic*, North Holland, 1977.

[2.] S. Reeves, M. Clarke: *Logic for Computer Science*, Addison Wesley, 1990.

[3.] G.S. Boolos, R.C. Jeffrey: *Computability and Logic*, Cambridge University Press, 1989.

[4.] W. Bouma: *Algebraïsche Specificaties*, Kluwer Bedrijfswetenschappen, 1991.