

Denkende machines

Docentenhandleiding

Exact in context

Denkende machines: computers, rekenen, redeneren is ontwikkeld met subsidie van het Spinoza Logic in Action project van Johan van Benthem.

De samenstelling werd begeleid door een klankbord van leraren en beta-cultuurverspreiders, bestaande uit Johan van Benthem (hoogleraar logica), Wim Berkelmans (directeur Stichting Vierkant voor Wiskunde), Christian Bokhove (docent informatiekunde), Anne Kalde-waij (informaticus en universiteitsbestuurder), Jef Kolle (docent wiskunde), Peter Lam-mers (docent economie), Jan de Ruijter (docent wiskunde), Hubert Slings (redacteur Tekst in context), Marco Swaen (docent lerarenopleiding wiskunde).

Elektronische ondersteuning op internet

De website die hoort bij dit boek is te vinden op het volgende adres:

<http://www.science.uva.nl/denkendemachines/>

Jan Jaspars, een van de auteurs, programmeerde onder andere simulaties van enkele van de historische machines uit het eerste hoofdstuk, van een aantal logische schakelingen en van de waarheidstabellen uit hoofdstuk twee. Hij ontwikkelde ook programmatuur waarmee de gebruiker zelf de machinemodellen en de programma's van het soort dat in hoofdstuk drie aan de orde komt, kan ontwerpen en laten draaien.

Naast het speciaal voor het boek ontwikkelde materiaal geeft de website ook verwijzingen naar andere sites over de geschiedenis en theorie van computers. Bovendien kan de lezer via de site reacties en commentaar op het boek kwijt. Die reacties zien wij graag tegemoet.

De reeks *Exact in context* is ontwikkeld voor gebruik in de Tweede Fase. Met behulp van aanpassingen in de dosering en docering is de reeks op tal van manieren inzetbaar: zowel voor HAVO 4/5 als voor VWO 4/5/6, zowel voor klassikale behandeling als voor gedeeltelijk gebruik, zowel voor vrijblijvende kennismaking als voor toetsing met behulp van vragen en opdrachten op verschillende niveaus.

Denkende machines: computers, rekenen, redeneren is het eerste deel in de reeks *Exact in context*.

Denkende machines

Computers, rekenen, redeneren

EXACT IN CONTEXT 1

Docentenhandleiding

Antwoorden

AMSTERDAM UNIVERSITY PRESS

ISBN 90 5356 583 3

NUGI 952

© Amsterdam University Press, Amsterdam, 2002

Alle rechten voorbehouden. Niets uit deze uitgave mag worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopien, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voorzover het maken van kopieën uit deze uitgave is toegestaan op grond van artikel 16B Auteurswet 1912 j⁰ het Besluit van 20 juni 1974, St.b. 351, zoals gewijzigd bij het Besluit van 23 augustus 1985, Stb. 471 en artikel 17 Auteurswet 1912, dient men de daarvoor wettelijk verschuldigde vergoedingen te voldoen aan de Stichting Reprorecht (Postbus 882, 1180 AW Amstelveen). Voor het overnemen van gedeelte(n) uit deze uitgave in bloemlezingen, readers en andere compilatiewerken (artikel 16 Auteurswet 1912) dient men zich tot de uitgever te wenden.

Inhoudsopgave

Hoofdstuk 1	7
Hoofdstuk 2	12
Hoofdstuk 3	17

Hoofdstuk 1

Uitwerking van 1.1. 256 en 64 zijn beide machten van 2: $256 = 2^8$ en $64 = 2^6$. Met een byte (= 8 bits) kun je de getallen 0 tot en met 255 representeren. Ieder getal staat dan voor een kleur.

Uitwerking van 1.2. Digitaal: kassa, roulette, piano. Analoog: thermometer, liniaal, viool.

Uitwerking van 1.3. De Oude Grieken gebruikten de 24 letters van hun alfabet plus drie extra symbolen (in onbruik geraakte letters). Elke letter stond voor een vast getal.

$\alpha = 1, \beta = 2, \dots, \theta = 9$

$\iota = 10, \kappa = 20, \dots, \chi = 90$

$\rho = 100, \sigma = 200, \dots, \xi\xi = 900$

Het getal 129 werd dus geschreven als $\rho\kappa\theta$, of als $\kappa\theta\rho$: de volgorde maakt immers niets uit.

Uitwerking van 1.4. Links staat het getal $2+30+50 = 82$. Rechts staat $1+40+200+1000 = 1241$.

Uitwerking van 1.5. Het duidelijkste voorbeeld is natuurlijk de tijdsaanduiding, met een verdeling van een uur in 60 minuten, en van een minuut in 60 seconden. Dit is een erfenis van de Babyloniërs, die rekenden in het zestigtallig stelsel. Hetzelfde geldt voor de verdeling van de cirkel in 360 graden.

Het getal 12 als basis voor het tellen heeft ook zijn sporen nagelaten in onze omgeving: een jaar telt twaalf maanden, en een dag twaalf uren. De basis twaalf voor het tellen heeft waarschijnlijk een astronomische oorsprong: er gaan twaalf volle manen in een jaar. In de Nederlandse taal zijn woorden als dozijn (12) en gros ($144 = 12 \times 12$) restanten van een twaalfvallig stelsel. In Engeland is een yard verdeeld in 3 feet en een foot is weer 12 inch.

Uitwerking van 1.6.

	3	6	9	
1	1	2	3	4
	2	4	6	
7	2	1	3	7
	3	4	3	

Uitwerking van 1.7. Op de diagonalen staan achtereenvolgens, te beginnen rechtsonder, eenheden, tientallen, honderdtallen, . . . De getallen op de diagonalen vinden we terug in de kolommen als we optellen zoals wij het geleerd hebben.

Uitwerking van 1.8. De negenproef berust op het volgende rekenkundige feit: het verschil tussen een (positief geheel) getal en de som van de cijfers van dat getal is altijd een veelvoud van negen.

Waarom dit zo is valt het best te verduidelijken aan een voorbeeld.

$$\begin{aligned} 6413 &= 6 \times 1000 + 4 \times 100 + 1 \times 10 + 3 \\ &= 6 \times (9 \times 111 + 1) + 4 \times (9 \times 11 + 1) + 1 \times (9 \times 1 + 1) + 3 \\ &= 9 \times (6 \times 111 + 4 \times 11 + 1 \times 1) + 6 + 4 + 1 + 3. \end{aligned}$$

$$6413 - (6 + 4 + 1 + 3) = 9 \times (6 \times 111 + 4 \times 11 + 1 \times 1).$$

Als het verschil tussen twee getallen x en y een veelvoud van 9 is zeggen we dat x en y congruent zijn modulo 9. Notatie: $x = y \pmod{9}$. Twee getallen die congruent zijn modulo 9 leveren dezelfde rest op als je ze allebei deelt door 9.

Voor elk getal geldt dus dat dat getal en de som van de cijfers van dat getal congruent zijn modulo 9. Dus:

$$\begin{aligned} 6413 &= 6 + 4 + 1 + 3 = 14 \pmod{9} \\ 14 &= 5 \pmod{9}. \end{aligned}$$

Als je in een vermenigvuldiging of een optelling elk getal vervangt door een getal dat ermee congruent is modulo 9, dan krijg je een uitkomst die congruent is met de oorspronkelijke uitkomst modulo 9. Dat wil zeggen: als je in een vermenigvuldiging of optelling elk getal vervangt door de som van zijn cijfers, dan is het verschil tussen de oorspronkelijke uitkomst en de nieuwe uitkomst altijd een veelvoud van 9.

Uitwerking van 1.9. Delen correspondeert met aftrekken. Kijk maar in de logaritmetafel voor de logaritmen met basis 10, op bladzijde 14. $40/8 = 5$, en ${}^{10}\log 40 - {}^{10}\log 8 = 1,602 - 0,903 = 0,699 = {}^{10}\log 5$.

Worteltrekken correspondeert met delen door 2. Kijk maar: $\sqrt{100} = 10$, en ${}^{10}\log 100/2 = 2/2 = 1 = {}^{10}\log 10$. Of nog een voorbeeld: $\sqrt{4} = 2$, en ${}^{10}\log 4/2 = 0,602/2 = 0,301 = {}^{10}\log 2$.

Uitwerking van 1.10. Beschouw de overdracht van tientallen naar honderdtallen. Het tandwiel voor de tientallen bestaat uit twee lagen. De ene laag heeft tien tanden, de andere een. Als dit tandwiel een hele ronde heeft gemaakt, draait het ene tandje het tandwiel van de honderdtallen een cijfer verder. (Maak zelf een tekening.)

Uitwerking van 1.11. Het 9-complement van 3456789, over 7 posities, is 6543210. Dit optellen bij 10000000 geeft 16543210. Eerste 1 weglaten, en 1 bij het resultaat optellen geeft de gevraagde uitkomst: 6543211.

Uitwerking van 1.12. De volgende tabel beschrijft de overbrengverhoudingen van de de-railleur.

	12	13	14	15	17	20
38	3,167	2,923	2,714	2,533	2,235	1,900
42	3,500	3,231	3,000	2,800	2,471	2,100

Uitwerking van 1.13. Als we $f(0), f(1), f(2)$ uitrekenen kunnen we hieruit het tweede verschil afleiden, en zo de tabel afmaken. Uit $f(0) = 7, f(1) = 5$ volgt dat de eerste waarde voor het eerste verschil -2 bedraagt. Ook $f(2) = 5$, en dus is de tweede waarde van het eerste verschil 0 . Het tweede verschil is dus $0 - (-2) = 2$, en we kunnen de tabel door optellen verder uitwerken.

$$\begin{array}{cccccccc}
 7 & 5 & 5 & 7 & 11 & 17 & \cdots & \\
 -2 & 0 & 2 & 4 & 6 & \cdots & & \\
 2 & 2 & 2 & 2 & \cdots & & &
 \end{array}$$

Uitwerking van 1.14. Het derde verschil is constant, dus we weten dat we te maken hebben met een veelterm van de vorm $f(x) = ax^3 + bx^2 + cx + d$. Uit $f(0) = 3$ leiden we af dat $d = 3$. Uit $f(1) = 6$ leiden we af dat $a + b + c + 3 = 6$, d.w.z., $a + b + c = 3$. Uit $f(2) = 17$ krijgen we dat $8a + 4b + 2c + 3 = 17$, d.w.z., $8a + 4b + 2c = 14$, en we zien dat $a = b = c = 1$ voldoet. De gevraagde veelterm is dus $x^3 + x^2 + x + 3$.

Uitwerking van 1.15.

1. Voor het representeren van de drie cijfers in getallen van 0 tot en met 999 zijn in de ENIAC representatie $3 \times 10 = 30$ buizen nodig.
2. Alle mogelijke aan/uit combinaties van 10 buizen geeft $2^{10} = 1024$ mogelijkheden. Daarvan worden er daadwerkelijk maar 10 gebruikt. De verhouding tussen gebruikte combinaties en mogelijke combinaties is dus $\frac{10}{2^{10}}$, met andere woorden, er wordt nog niet 1 procent van de mogelijkheden benut.
3. Tien buizen zijn genoeg, want dit geeft $2^{10} = 1024$ verschillende aan/uit combinaties. In plaats van de 18.000 buizen die er in de ENIAC zaten had men dus toegekund met slechts 6000.

Uitwerking van 1.16. De volgende Boolese wetten uit de tabel gaan ook op voor gewoon rekenen.

- $x + (y + z) = (x + y) + z$.
- $x + y = y + x$.
- $x + 0 = x$.
- $x \cdot (y \cdot z) = (x \cdot y) \cdot z$.

- $x \cdot y = y \cdot x$.
- $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$.
- $x \cdot 0 = 0$.
- $x \cdot 1 = x$.
- $--x = x$.

Bij de andere regels treden er verschillen op, als volgt.

- $x + x = x$. Bij gewoon rekenen geldt dit alleen als $x = 0$. Bij gewoon rekenen hebben we in het algemeen $x + x = 2 \cdot x$.
- $x + (y \cdot z) = (x + y) \cdot (x + z)$. Bij gewoon rekenen geldt $(x + y) \cdot (x + z) = (x \cdot x) + (x \cdot y) + (z \cdot x) + (z \cdot y)$.
- $x + (x \cdot y) = x$. Bij gewoon rekenen geldt $x + (x \cdot y) = x \cdot (1 + y)$.
- $-(x+y) = -x \cdot -y$. Bij gewoon rekenen gelden $-(x+y) = -x + -y$ en $-x \cdot -y = x \cdot y$.
- $x + 1 = 1$. Bij gewoon rekenen geldt dit alleen voor $x = 0$.
- $x + -x = 1$. Bij gewoon rekenen geldt dit voor geen enkele x . Bij gewoon rekenen hebben we juist $x + -x = 0$.
- $x \cdot x = x$. Bij gewoon rekenen geldt dit alleen voor $x = 1$.
- $x \cdot (x + y) = x$. Bij gewoon rekenen hebben we $x \cdot (x + y) = (x \cdot x) + (x \cdot y)$.
- $-(x \cdot y) = -x + -y$. Bij gewoon rekenen hebben we $-(x \cdot y) = -x \cdot y = x \cdot -y$.
- $x \cdot -x = 0$. Bij gewoon rekenen geldt dit voor geen enkele x . Bij gewoon rekenen hebben we juist $x \cdot -x = 1$.

Uitwerking van 1.17. Als er drie verzamelingen A , B en C zijn hebben we te maken met acht mogelijke situaties: dingen kunnen in A of niet in A (d.w.z. in \bar{A}) zitten, en voor elk van die twee situaties onderscheiden we tussen dingen die in B of in \bar{B} zitten, en voor elk van die vier situaties onderscheiden we weer tussen dingen die in C of in \bar{C} zitten. De mogelijke combinaties zijn dus:

$$ABC, ABC\bar{C}, A\bar{B}C, A\bar{B}\bar{C}, \bar{A}BC, \bar{A}B\bar{C}, \bar{A}\bar{B}C, \bar{A}\bar{B}\bar{C}.$$

Als geldt dat alle $A B$ zijn vallen de mogelijkheden $A\bar{B}C$ en $A\bar{B}\bar{C}$ af: er zijn dan immers geen dingen die zowel A als niet- B zijn.

Als geen B C is zijn er geen dingen die B en C zijn. Dus: de mogelijkheden ABC , $\bar{A}BC$ vallen af.

Onder *beide* aannamen blijven dus alleen de volgende mogelijkheden over:

$$AB\bar{C}, \bar{A}B\bar{C}, \bar{A}\bar{B}C, \bar{A}\bar{B}\bar{C}.$$

Uitwerking van 1.18. De pijlen werken negaties $-$ en conjuncties \cdot zoveel mogelijk naar binnen, en zorgen er in het algemeen voor dat elke Boolese uitdrukking er uiteindelijk uit komt te zien als een disjunctie van conjuncties van losse symbolen of symbolen met een $-$ ervoor. Iets formeler gezegd: de uitdrukkingen die we uiteindelijk krijgen hebben de vorm $C_1 + (\dots + C_n)$, waarbij elke C van de vorm $L_1 \cdot (\dots \cdot L_m)$ is, met elke L hetzij van de vorm a hetzij van de vorm $-a$, waarbij a een losse variabele is.

Uitwerking van 1.19.

$$\begin{array}{ll} -1 & \longrightarrow 0 \\ -0 & \longrightarrow 1 \\ x + 1 & \longrightarrow 1 \\ 1 + x & \longrightarrow 1 \\ x + 0 & \longrightarrow x \\ 0 + x & \longrightarrow x \\ x \cdot 1 & \longrightarrow x \\ 1 \cdot x & \longrightarrow x \\ x \cdot 0 & \longrightarrow 0 \\ 0 \cdot x & \longrightarrow 0 \end{array}$$

Hoofdstuk 2

Uitwerking van 2.1. $2000 = 2 \times 1000 = 2^2 \times 500 = 2^3 \times 250 = 2^4 \times 125 = 2^4 \times (124 + 1) = 2^5 \times 62 + 2^4 = 2^6 \times 31 + 2^4 = 2^7 \times 15 + 2^6 + 2^4 = 2^8 \times 7 + 2^7 + 2^6 + 2^4 = 2^9 \times 3 + 2^8 + 2^7 + 2^6 + 2^4 = 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^4 = 11111010000_2$.

Deze uitkomst krijg je ook met de procedure die in de tekst wordt beschreven.

2000	0
1000	00
500	000
250	0000
125	10000
62	010000
31	1010000
15	11010000
7	111010000
3	1111010000
1	11111010000

Uitwerking van 2.2. $2048 = 2^{11}$, dus de binaire representatie van 2048 is 100000000000, een 1 met elf nullen. Reden, wellicht, voor een groot (carnavals)feest?

Uitwerking van 2.3.

1. Bijvoorbeeld: 1983. Unair: 1983 symbolen. Binair: 11 symbolen. Decimaal: 4 symbolen.
2. Unair: n symbolen. Binair: $^2 \log n + 1$. Decimaal: $^{10} \log n + 1$.

Uitwerking van 2.4.

1. $435 = 1 \cdot 3^5 + 2 \cdot 3^4 + 1 \cdot 3^3 + 0 \cdot 3^2 + 1 \cdot 3^1 + 0 \cdot 3^0$. Dus: 121010.
2. $4B7F = 4 \cdot 16^3 + 11 \cdot 16^2 + 7 \cdot 16^1 + 15 \cdot 16^0 = 19327$.
3. $435 = 1 \cdot 16^2 + 11 \cdot 16^1 + 3 \cdot 16^0$. Dus: 1B3.

Uitwerking van 2.5.

$$\begin{array}{r} 1011 \\ 10011 \\ \hline 11110 \end{array}$$

Uitwerking van 2.6.

waar	NOCH	waar	=	onwaar
waar	NOCH	onwaar	=	onwaar
onwaar	NOCH	waar	=	onwaar
onwaar	NOCH	onwaar	=	waar

Uitwerking van 2.7.

	waar	NEN	waar	=	onwaar
	waar	NEN	onwaar	=	waar
1.	onwaar	NEN	waar	=	waar
	onwaar	NEN	onwaar	=	waar

2. Het symbool voor NEN is een combinatie van het EN symbool en de NIET streep.

Uitwerking van 2.8. $11/101001\backslash 1101.\overline{10}$. De $\overline{10}$ staat daarbij voor het gedeelte dat oneindig vaak wordt herhaald. We hebben hier kennelijk te maken met een repeterende binaire breuk.

Uitwerking van 2.9 en 2.10: overgelaten aan de lezer.

Uitwerking van 2.11.

1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

Uitwerking van 2.12. 1010.1010 is binaire complementnotatie voor het negatieve getal $(2^5 + 2^3 + 2) - 2^7 = (32 + 8 + 2) - 128 = -86$. $\text{COM}_2^7 1010.1010 = 0101.0101 + 1 = 0101.0110$. Dit is binaire complementnotatie voor $2^6 + 2^4 + 2^2 + 2 = 64 + 16 + 4 + 2 = 86$.

Uitwerking van 2.13. Vier bytes is $4 \times 8 = 32$ bits. Het hoogste bit is voor de tekeninformatie, dus

1000 0000 0000 0000 0000 0000 0000 0000

staat voor $-2^{31} =$ (met behulp van de zakrekenmachine) -2147483648 . Dit is het kleinste getal dat gerepresenteerd kan worden.

Uitwerking van 2.14.

1. p : Jan heeft een auto gekocht, q : Jans vrouw heeft een rijbewijs, r : Jan heeft een rijbewijs. Vertaling: $p \wedge \neg q \wedge \neg r$. Ware situatie: Jan heeft een auto gekocht, Jan heeft geen rijbewijs, zijn vrouw heeft geen rijbewijs. Onware situatie: Jan heeft geen auto gekocht.
2. p : Piet steekt zijn lichten aan, q : het is donker, r : het regent. Vertaling: $(q \rightarrow p) \wedge (r \rightarrow \neg p)$. Ware situatie: het is donker en het regent niet, Piet steekt zijn lichten aan. Onware situatie: het is niet donker, het regent, Piet steekt zijn lichten aan. Verder is de zin onwaar in elke situatie waarin het donker is en regent!
3. p : Sandra rijdt naar zee, q : het regent, r : het is te warm. De zin kan op twee manieren worden geïnterpreteerd.
 - Vertaling 1: $(\neg q \vee r) \rightarrow p$. Ware situatie: Het regent, het is koud en Sandra blijft thuis. Onware situatie: Het is te warm en Sandra blijft thuis.
 - Vertaling 2: $\neg(q \vee r) \rightarrow p$. Ware situatie: Het is te warm en Sandra blijft thuis. Onware situatie: Het regent niet, het is koud en Sandra blijft thuis.
4. p : Kees steekt zijn lichten aan, q : het is donker, r : het regent, s : het mist. Vertaling: $(q \vee r \vee s) \rightarrow p$. Ware situatie: Kees steekt zijn lichten aan. Onware situatie: Het mist, maar Kees steekt zijn lichten niet aan.
5. p : Johan rijdt met de auto, q : de auto is kapot, r : het is mooi weer. Vertaling: $(q \vee r) \leftrightarrow \neg p$. Ware situatie: De auto is kapot en Johan fietst. Onware situatie: Johan rijdt met de auto (hij is dus niet kapot) en het is mooi weer.

Uitwerking van 2.15.

1. De prinses zit achter deur 2.

p_1	p_2	$deur1$	$deur2$	$(deur1 \wedge deur2) \vee (\neg deur1 \wedge \neg deur2)$
0	0	0	1	0
0	1	1	1	1
1	0	1	0	0
1	1	1	0	0

2. Er zitten prinsessen achter beide deuren.

p_1	p_2	$deur1$	$deur2$	$(deur1 \wedge deur2) \vee (\neg deur1 \wedge \neg deur2)$
0	0	1	0	0
0	1	1	0	0
1	0	0	1	0
1	1	1	1	1

3. De prinses zit achter deur 2.

p_1	p_2	$deur1$	$deur2$	$(p1 \leftrightarrow deur1) \wedge (\neg p2 \leftrightarrow deur2)$
0	0	0	0	0
0	1	0	0	1
1	0	0	0	0
1	1	1	1	0

Uitwerking van 2.16.

1. klopt niet (tegenvoorbeeld: p waar, r onwaar);
2. klopt niet (tegenvoorbeeld: q waar, p en r onwaar);
3. klopt;
4. klopt.

Uitwerking van 2.18. 1 en 3 zijn tautologieën, 2 niet (tegenvoorbeeld: p, q en r waar).

Uitwerking van 2.19. Alleen de twee formules in 3 zijn niet equivalent.

Uitwerking van 2.20 $x = y \equiv (x \rightarrow y) \wedge (y \rightarrow x)$, $x < y \equiv \neg x \wedge y$ en $x \leq y \equiv x \rightarrow y$

Uitwerking van 2.21.

1. $\neg \top = \perp$

$$\neg \top = \neg \top \wedge \top = \perp$$

2. $x \vee (y \wedge \neg y) = x$

$$x \vee (y \wedge \neg y) = x \vee \perp = x$$

3. $x \vee x = x$

$$x = x \vee \perp = x \vee (x \wedge \neg x) = (x \vee x) \wedge (x \vee \neg x) = (x \vee x) \wedge \top = x \vee x$$

4. $x \vee \top = \top$

$$x \vee \top = x \vee (x \vee \neg x) = (x \vee x) \vee \neg x = x \vee \neg x = \top$$

5. $x \vee (x \wedge y) = x$

$$x = x \wedge \top = x \wedge (y \vee \top) = (x \wedge y) \vee (x \wedge \top) = (x \wedge y) \vee x = x \vee (x \wedge y)$$

6. Laat zien dat als $x \vee y = x \wedge y$ dan $x = y$.

$$\begin{aligned} x &= x \vee \perp = x \vee (y \wedge \neg y) = (x \vee y) \wedge (x \vee \neg y) = (x \wedge y) \wedge (x \vee \neg y) \\ &= (x \wedge y \wedge x) \vee (x \wedge y \wedge \neg y) = x \wedge y \wedge x = x \wedge y \end{aligned}$$

Op dezelfde manier is $y = x \wedge y$, en dus $x = y$.

7. Laat zien dat als $x \wedge y = \perp$ en $x \vee y = \top$ dan $y = \neg x$.

$$y = \perp \vee y = (x \wedge \neg x) \vee y = (x \vee y) \wedge (\neg x \vee y) = \top \wedge (\neg x \vee y) = \neg x \vee y$$

$$y = \top \wedge y = (x \vee \neg x) \wedge y = (x \wedge y) \vee (\neg x \wedge y) = \perp \vee (\neg x \wedge y) = \neg x \wedge y$$

Dus $\neg x \vee y = \neg x \wedge y$, en uit de vorige opgave volgt dus dat $y = \neg x$.

8. Laat zien dat $\neg\neg x = x$. Omdat $\neg x \wedge x = \perp$ en $\neg x \vee x = \top$, $\neg\neg x = x$.

9. Laat zien dat $\neg(x \vee y) = \neg x \wedge \neg y$ en $\neg(x \wedge y) = \neg x \vee \neg y$.

$$(x \vee y) \wedge (\neg x \wedge \neg y) = (x \wedge (\neg x \wedge \neg y)) \vee (y \wedge (\neg x \wedge \neg y)) = \perp \vee \perp = \perp$$

$$(x \vee y) \vee (\neg x \wedge \neg y) = (x \vee y \vee \neg x) \wedge (x \vee y \vee \neg y) = \top \wedge \top = \top$$

Toepassing van het eerdere resultaat geeft $\neg(x \vee y) = (\neg x \wedge \neg y)$. Analoog voor de andere vergelijking.

Uitwerking van 2.22.

1. $\neg(x \wedge \neg x)$ is een tautologie

$$\neg(x \wedge \neg x) = \neg\perp = \top$$

2. $x \vee y \vee (\neg x \wedge \neg y)$ is een tautologie

$$x \vee y \vee (\neg x \wedge \neg y) = (x \vee y) \vee \neg(x \vee y) = \top$$

3. $x \wedge y \models x$

$$\neg(x \wedge y) \vee x = \neg x \vee \neg y \vee x = (\neg x \vee x) \vee y = \top \vee y = \top$$

4. $x \models x \vee y$

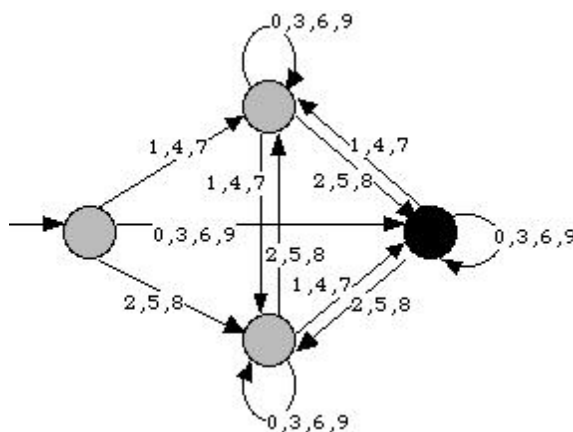
$$\neg x \vee x \vee y = \top \vee y = \top$$

5. $x \wedge (y \vee z) \models (x \wedge y) \vee z$

$$\begin{aligned} \neg(x \wedge (y \vee z)) \vee ((x \wedge y) \vee z) &= \neg x \vee (\neg y \wedge \neg z) \vee (x \wedge y) \vee z \\ &= ((\neg x \vee x) \wedge (\neg x \vee y)) \vee ((z \vee \neg y) \wedge (z \vee \neg z)) \\ &= (\neg x \vee y) \vee (z \vee \neg y) = \neg x \vee z \vee \top = \top \end{aligned}$$

Hoofdstuk 3

Uitwerking van 3.1.



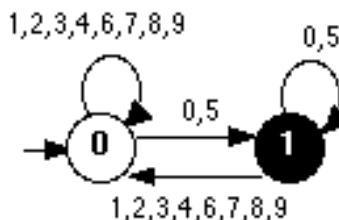
We hebben hier de pijlen tussen twee toestanden samengetrokken. Deze verkorte manier van tekeningen wordt elders ook gebruikt.

Uitwerking van 3.2. Rechtsboven: de getallen met rest 1 na deling door 3 (technische terminologie: 1 modulo 3). Rechtsonder: de getallen met rest 2 na deling door 3 (2 modulo 3).

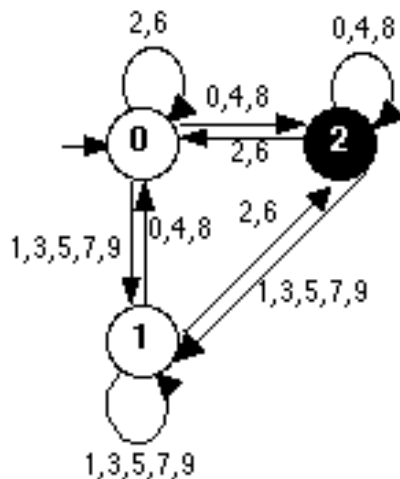
Uitwerking van 3.3 Voor elk getal geldt dat het een drievoud is precies wanneer de volgende eigenschap geldt: als je de cijfers van decimale representatie optelt, en vervolgens de cijfers van het resultaat optelt, enzovoort, tot je een enkel cijfer overhoudt, dan is het resultaat deelbaar door 3. Bijvoorbeeld 1632 is deelbaar door 3 want $1+6+3+2 = 12$, en $1+2 = 3$.

Net zo'n eigenschap hebben we voor 9 — bijvoorbeeld: 111111111 is deelbaar door 9, want $1+1+1+1+1+1+1+1+1 = 9$ — maar niet, bijvoorbeeld, voor 5: 10 is deelbaar door 5, maar de som van de cijfers van 10 is niet deelbaar door 5.

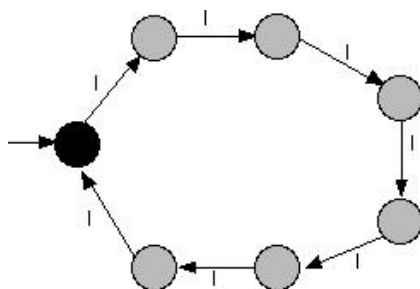
Uitwerking van 3.4.



Uitwerking van 3.5.



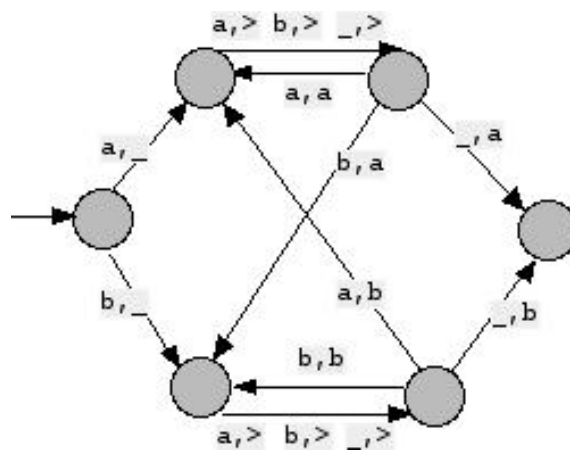
Uitwerking van 3.6.



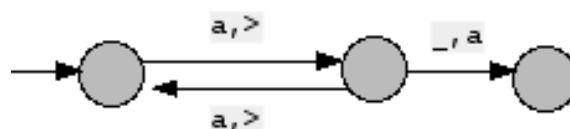
Uitwerking van 3.7. Stel dat er een automaat is die alle rijtjes $_, 01, 0011, 000111, \dots$ zou herkennen. Neem nou een rijtje die hierbij hoort maar dat langer is dan het aantal toestanden uit de automaat. Zeg dat het rijtje n tekens lang is. Dan moet er een toestand T zijn die ten minste tweemaal bezocht wordt tijdens de herkenning. Laat nu m_1 het aantal symbolen zijn dat ingelezen is voor het eerste bezoek aan T en m_2 het aantal symbolen dat is ingelezen voor het tweede bezoek aan T . Maak nu een rijtje van 0- en 1-en dat tot aan het m_2^e symbool identiek is aan de originele rij. Vervolgens herhalen we het stuk van het $m_1 + 1$ -de t/m het m_2 -de symbool en daarna gaan we op de oude manier verder en kopiëren het $m_2 + 1$ -de symbool tot het einde van de originele rij. Ook deze rij wordt herkend. Het verschil in de herkenning bestaat er slechts uit dat de $m_1 - m_2$ -lus twee keer gemaakt wordt. Als dit verdubbelde stuk alleen maar 0-en bevat, dan heeft de rij echter te veel 0-en. Hetzelfde geldt als het alleen maar uit 1-en bestaat. Als het uit 0-en en 1-en bestaat, dan zou na verdubbeling er nu 0-en na 1-en in de rij voorkomen. Al deze resultaten behoren niet bij de verzameling van rijtjes die herkend dienden te worden.

Uitwerking van 3.8. We gaan vrijwel op dezelfde manier te werk als bij de vorige opdracht. Als er een automaat bestaat die alle l -rijtjes van kwadratische lengte herkent, dan moet die automaat ook een rijtje van lengte n^2 herkennen waarbij n groter is dan het aantal toestanden van de automaat. Dit betekent weer dat er zich zo'n m_1 - m_2 -lus moet voordoen: we spreken hierbij wel af dat deze lus zich voordoet bij de eerste toestand die twee keer bezocht wordt. Als we dit gedeelte uit de rij weghalen ontstaat er een nieuw rijtje dat korter is dan n^2 . Het m_1 - m_2 -stuk kan echter niet langer zijn dan n , want n is groter dan het aantal toestanden en voordat er n symbolen zijn ingelezen moet er dus al een toestand zijn die twee keer is bezocht. Kortom, de lengte van het overgebleven stuk is langer dan $n^2 - n$ maar korter dan n^2 . Het overgebleven stuk wordt weer herkend door de automaat, maar dat rijtje kan nooit kwadratische lengte hebben, want het kwadraat dat voorafgaat aan n^2 is $(n - 1)^2 = n^2 - 2n + 1$, en dat is kleiner dan $n^2 - n$. Kortom, onze automaat moet rare fouten maken.

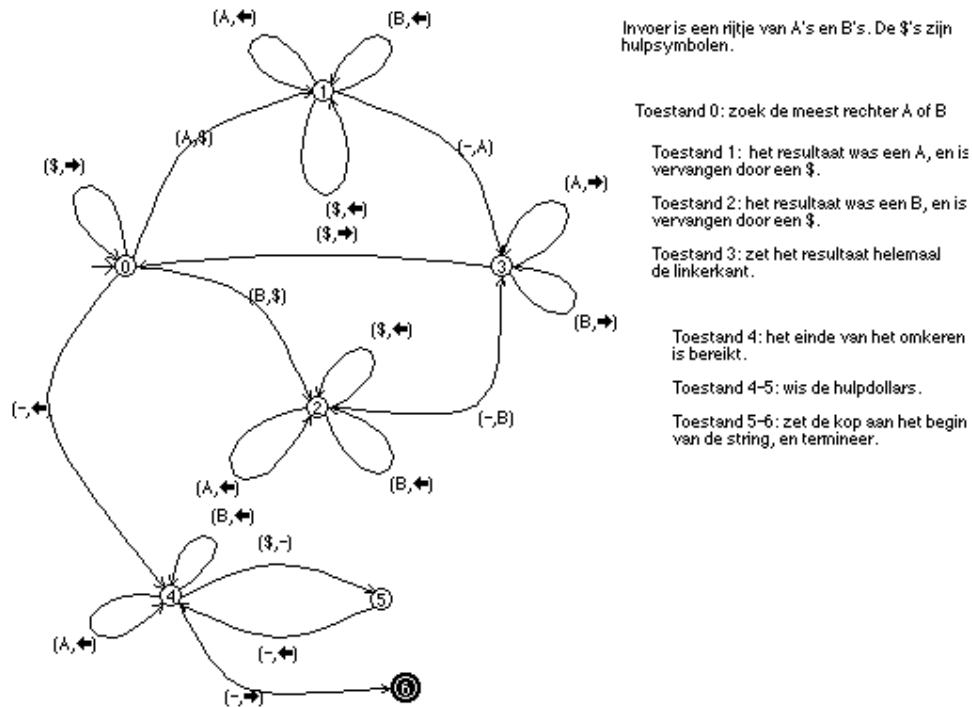
Uitwerking van 3.9.



Uitwerking van 3.10.



Uitwerking van 3.11.



Uitwerking van 3.12. De vierkantjes bevatten een kwadraat, zeg n^2 , aan stippen. De laatst bijgevoegde hoeveelheid, om van $(n - 1)^2$ naar n^2 te komen, is $2n - 1$: het n -de oneven getal.

Uitwerking van 3.13. Overgelaten aan de lezer. Hint: gebruik de Turing machine applet op de internetsite bij dit boek.

Uitwerking van 3.14. Maak zelf een tekening. Bestudeer ook de kwadraterende Turing machine op de internetsite bij dit boek.

Uitwerking van 3.15. Over de haalbaarheid van de Turing test verschillen de meningen zeer, zoals je gemakkelijk kunt constateren door op internet te zoeken met behulp van *Turing test*.

Uitwerking van 3.16. z blijft 0 wanneer het programma wordt opgestart in een beginsituatie met $y \geq x$. In dat geval bevat x na afloop van het programma het verschil van y en x in de starttoestand.

Uitwerking van 3.17.

if (T) {P} ~ if (T) {P} else {x+;x-}

Uitwerking van 3.18.

if (T) {P} else if (S) {Q} else {R} ~ if (T) {P} else { if (S) {Q} else {R}}

Uitwerking van 3.19.

```
repeat P until T ~ P; while (T) {P}
```

Uitwerking van 3.20.

```
[TIMES](..) {
  Y := EMPTY();
  while (X2?) { Y := PLUS }
}

z := EMPTY();
z := z+;
while (y?) {
  y-;
  z := TIMES(z,x);
}
```

Uitwerking van 3.21. De functie `MODULO` roept zichzelf aan. Dit is niet erg omdat `X1` steeds kleiner wordt en dan gaat op een gegeven moment de test `LARGER(X2,X1)` slagen en wordt `X1` naar het resultaat gekopieerd. In het geval dat `X2` leeg is wordt `X1` niet kleiner en blijft `MODULO` zichzelf aanroepen. M.a.w., het programma stopt niet. Je kan dit ondervangen door te definiëren:

```
[MODULO](..) { if (X2?) {P} else {Y := EMPTY()}},
```

waarbij `P` identiek is aan de functie `MODULO` in de tekst.

Uitwerking van 3.22.

```
[DIV](..) {
  if (MODULO(X2,X1)?) {
    Y: EMPTY(); Y+;
  } else { Y:= EMPTY()}
}
```

Uitwerking van 3.23.

```
[PRIME](.) {
  Z := EMPTY();Z+;Z+
  if (LARGER(Z,X1?)) {
    Y := EMPTY()          /* getallen kleiner dan 2 zijn niet priem */
  } else {
    Y := EMPTY(); Y+      /* neem aan dat X1 een priem */
    while (LARGER(X1,Z?)) { /* totdat het tegendeel blijkt */
      if (DIV(Z,X1?)) {
        Z := COPY(X1);    /* deler gevonden die kleiner */
        Y-                /* is dan X1 => geen priem en stop */
      } else {
        Z+                /* (nog) geen deler gevonden. */
      }
    }
  }
}
```

Uitwerking van 3.24.

```
[AND](..) {
  Y := EMPTY();
  if (X1?) {
    if (X2?) {
      Y+;
    }
  }
}
```

```

[OR](..) {
  Y := EMPTY();
  if (X1?) {
    Y+;
  } else {
    if (X2?) { Y+ }
  }
}

```

Uitwerking van 3.25.

```

[EQUALS](..) {
  Y := NOT(OR(LARGER(X1,X2),LARGER(X2,X1)));
}

```

Uitwerking van 3.26. Als HALT correct zou werken, dan zou nadat $x:=q$ is doorgevoerd HALT het stopgedrag van Q goed beantwoorden. Met andere woorden, na uitvoering van $x:=q$;HALT wordt een leeg y-register afgeleverd als Q niet stopt en een gevuld register indien Q stopt. Nou is Q gedefinieerd als een toevoeging van dit programma. Het stopt nadat $x:=q$; HALT y leeg heeft achtergelaten, en blijft doordraaien indien y juist niet gelegeerd is na $x:=q$;HALT. En daarmee heeft HALT een verkeerd antwoord afgeleverd.

Uitwerking van 3.27. Er moet uitkomen (niet iets om met de hand te doen):

27,82,41,124,62,31,94,47,142,71,214,107,322,161,484,242,121,
 364,182,91,274,137,412,206,103,310,155,466,233,700,350,175,
 526,263,790,395,1186,593,1780,890,445,1336,668,334,167,502,
 251,754,377,1132,566,283,850,425,1276,638,319,958,479,1438,
 719,2158,1079,3238,1619,4858,2429,7288,3644,1822,911,2734,
 1367,4102,2051,6154,3077,9232,4616,2308,1154,577,1732,866,
 433,1300,650,325,976,488,244,122,61,184,92,46,23,70,35,106,
 53,160,80,40,20,10,5,16,8,4,2,1

Uitwerking van 3.28. Bij een invoer van lengte $n = 1$ is $2^n = 2 > n^8 = 1$, maar dat is flauw. Voor invoer van lengte $1 < n \leq 43$ geldt $2^n < n^8$, maar daarna verandert de zaak. Met de rekenmachine krijgen we: $2^{44} = 17592186044416 > 44^8 = 14048223625216$,

$2^{45} = 35184372088832 > 45^8 = 16815125390625$, en de verschillen lopen nu snel op tot letterlijk astronomische grootte.

Uitwerking van 3.29. Dit programma werkt iets sneller. We beginnen weer met 1 en kijken of hij iemand kent onder de andere 999. Bij nummer 2 wordt dit weer gedaan, maar we testen niet meer of hij 1 kent. Dit lijkt tot een foute oplossing te kunnen leiden als 2 niemand kent maar wel 1 en zouden we 2 onterecht als de beroemdheid aanzien. Echter, 1 is niet beroemd dus de beroemde persoon moet een rugnummer hoger dan 1 hebben. Als 2 niet beroemd is, dan moeten we die beroemdheid ook tegenkomen bij het testen en moet 2 hem kennen: voldoende om 2 als niet beroemd te laten afvallen. Enfin, om te testen dat nummer n beroemd is hoeven we dus maar $1000 - n$ tests uit te voeren. In het totaal dus $999 + 998 + \dots + 2 + 1$. En, in het algemeen, $(N - 1) + \dots + 1$. Dit is een kwadratische functie (verschil steeds 1): $\frac{N(N-1)}{2}$.

Uitwerking van 3.30.

```
x := 0;
y := 1000;
while (x < y) { if (KENT(x,y)) { x := x+1 } else { y := y-1 } }
```