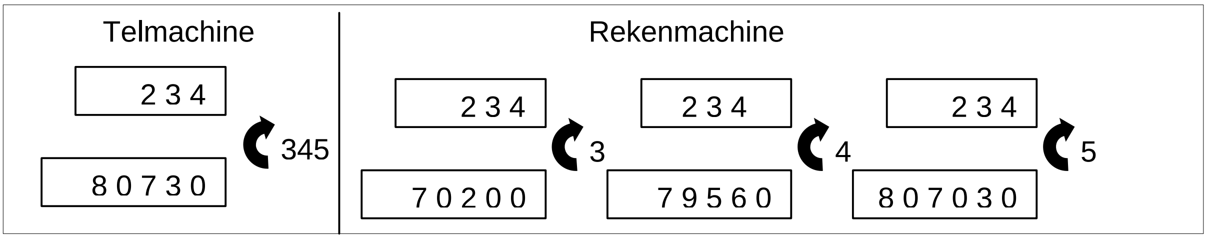
# IERSE LOGARITHMEN

Andries de Man

Mechanische rekenmachines kunnen eigenlijk alleen maar optellen en aftrekken. Vermenigvuldigen wordt bewerkstelligd door de multiplicand net zo vaak bij zichzelf op te tellen als de multiplicator groot is. Om het proces te versnellen wordt voor de hogere ordes van de multiplicator de multiplicand een positie verschoven, zodat voor een vermenigvuldiging met 345 niet 345 optellingen nodig zijn, maar slechts 3+4+5.

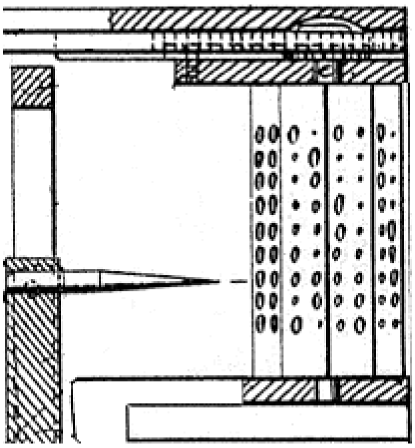
De machines van Leibnitz (1673) en Thomas de Colmar (1820) werkten al volgens dit principe. Toch was men niet tevreden met het feit dat het aantal optellingen, dus de duur van de berekening, afhangt van de waarde van de multiplicator.




**Figuur 1: Machinale vermenigvuldiging**

## Directe vermenigvuldiging

Ramón Verea, een Spanjaard die via Cuba in New York terecht was gekomen, stelde in 1878 voor om de tafels van vermenigvuldiging in “hardware” uit te voeren om daarmee een meercijferige multiplicand in één keer met een multiplicator kleiner dan 10 te kunnen vermenigvuldigen.[1] Hij maakte een tienkantig prisma met gaten waarvan de “ondiepte”[2] de eenvoudige producten[3] representeert. (Figuur 2) De diepte van een gat wordt afgetast met een pin die een telwiel aanstuurt. Bij een vermenigvuldiging wordt bij elk prisma het vlak dat correspondeert met een cijfer uit de multiplicand naar voren gedraaid. De pinnen worden op een hoogte gezet die correspondeert met één cijfer van de multiplicator. Vervolgens worden de prisma’s naar de pinnen geschoven, waardoor de pinnen zoveel bewegen als voorgeschreven door de tafels van vermenigvuldiging. Bij vermenigvuldiging met een meercijferige multiplicator moet deze handeling meerdere keren herhaald worden. Als de ondiepte van de gaten rechtstreeks de eenvoudige producten representeert zou het diepste gat 9×9=81 keer zo diep moeten zijn als het ondiepste gat. Dit vergt een aftastnauwkeurigheid die in Verea’s tijd niet te behalen was zonder een onwerkbaar groot en traag mechaniek te maken. Daarom liet Verea de eenvoudige producten door twee gaten representeren: één gat voor de eenheden en één gat voor de tientallen. De gaten worden door twee pinnen afgetast die twee opeenvolgende ordes in het telmechanisme aansturen. Bij 9×9 zou dus de ene pin 1 stapje bewegen en het eenheden-wiel in het telmechanisme met 1 ophogen, terwijl de andere pin 8 stappen beweegt en het tientallen-wiel in het telmechanisme met 8 ophoogt. Een andere reden om twee gaten te gebruiken is dat hierdoor de normale tientallen-overdracht in het telmechanisme toegepast kan worden: als het eenheden-wiel verder dan 9 draait moet het tientallen-wiel een stap verder gedraaid worden. Als één gat gebruikt was zou een overdracht van 8 voor kunnen komen en dat kunnen de gebruikelijke tientallen-overdracht-mechanismen niet aan.



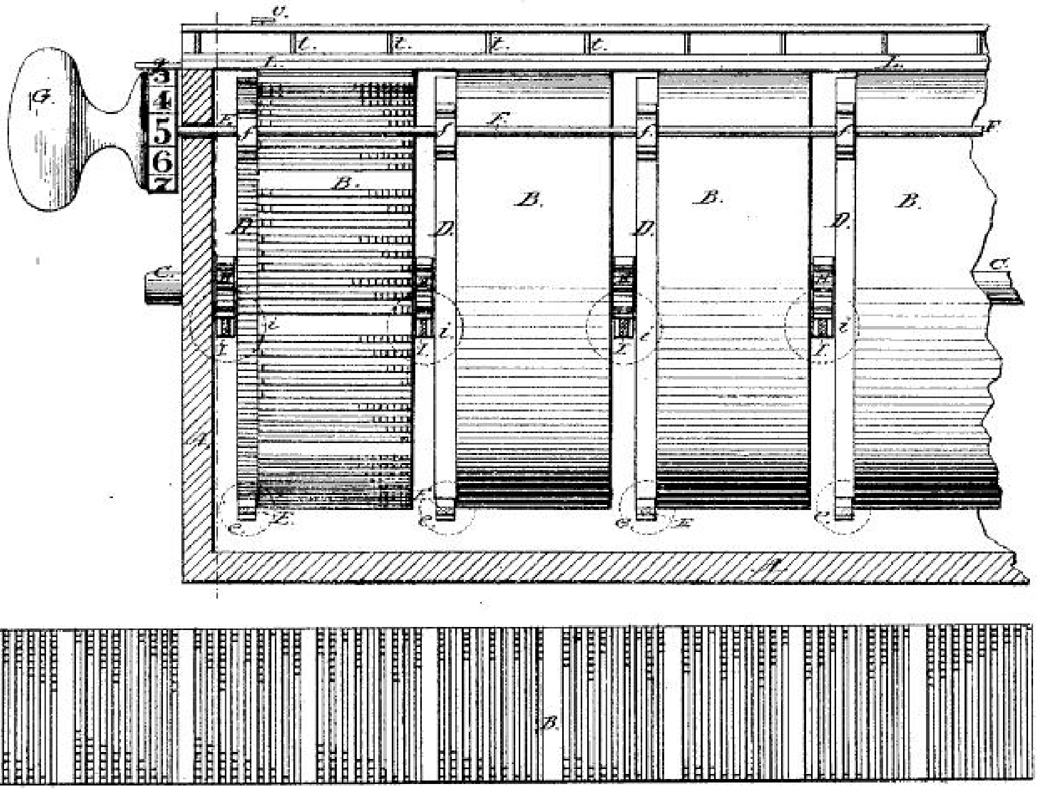
**Figuur 2: Verea's tafel van vermenigvuldiging**

[1] Ramón Verea, “Improvement in calculating machines”, US Patent 207918, 10 September 1878, http://www.google.com/patents?q=patent:207918

[2] Ondiepte = de afstand van de bodem van een gat tot de as van het prisma.

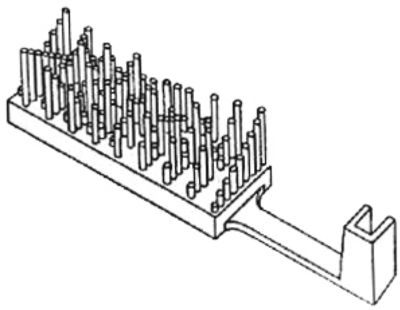
[3] Eenvoudige producten: 1×1, 1×2, … 9×8, 9×9

In 1872, voordat Verea zijn machine patenteerde, had Edmund Barbour al een patent gekregen op een vermenigvuldiger[4]. De eenvoudige producten zijn hier uitgevoerd als korte tandheu-gels op grote cylinders.



**Figuur 3: Barbour's tafels van vermenigvuldiging**

De Fransman Leon Bollée heeft in 1896 een vermenigvuldiger ontwikkeld volgens een vergelijkbaar principe.[5] In plaats van prisma's met gaten gebruikte hij staafjes op een blok om de tafels te representeren (Figuur 4). Bollée beschrijft in zijn patent duidelijk de arbeidsgang van zijn machine: na instelling van de multiplicand en van één cijfer van de multiplicator worden eerst alle eenheden van de eenvoudige producten aan het telmechanisme toegevoegd, daarna worden de eventueel hierdoor veroorzaakte tientallen-overdrachten uitgevoerd, vervolgens worden de tientallen van de eenvoudige producten aan het (één positie verschoven) telmechanisme toegevoegd en de daardoor veroorzaakte tientallenoverdrachten uitgevoerd, waarna het telmechanisme weer terugschuift.



**Figuur 4: Bollée's tafel van vermenigvuldiging**

Bollée heeft dus maar één aftaster per resultaatcijfer nodig, terwijl Verea er twee heeft. Het is niet duidelijk hoe in Verea's machine twee aftasters, de eenhedenaftaster van de ene orde en de tientallen-aftaster van een lagere orde, tegelijk op één telwiel kunnen werken.

Vermenigvuldigingstafels volgens Bollée's ontwerp zijn succesvol toegepast door Otto Steiger en Hans Egli in de "Millionaire" rekenmachine. Er zijn nog enkele andere directe vermenigvuldigers op de markt gebracht, maar het gros van de mechanische rekenmachines vermenigvuldigde door herhaaldelijk op te tellen.

## Logaritmen

De doorgewinterde rekenliniaalverzamelaar zal zich nu afvragen waarom men voor directe-vermenigvuldiging-door-optellen geenllogaritmen gebruikt. Daarvoor is een mechanische reden.

[4] Edmund Barbour, "Improvements in Calculating Machines", 13 Augustus 1872, http://www.google.com/patents?q=patent:130404

[5] Leon Bollée, "Calculating Machine", US Patent 556720, 17 Maart 1896, http://www.google.com/patents?q=patent:556720

De kleinste afstand tussen de logaritmen van de eenvoudige producten is de log(64) - log(63) = 0.00684. Dit betekent dat op een schaal van log(81)=1.909 een aftastnauwkeurigheid van 0.00684 gehaald moet worden, dus 0.4%, en dat is nog nauwkeuriger dan de 1/81=1.2% die voor de "één-gat-vermenigvuldigingstafel" nodig is die eerder besproken is.
Getaltheorie geeft echter een alternatief: de Jacobi-index van een geheel getal[6].
Neem een priemgetal $p$. De Jacobi-index Ind($z_i$) van een geheel getal $z_i$ dat tussen 1 en $p$ - 1 ligt is een geheel getal tussen 0 en $p$ - 2 en heeft de eigenschap:

$$\mathrm{Ind}(z_1 \times z_2) = \mathrm{Ind}(z_1) + \mathrm{Ind}(z_2)$$

Voor elk getal getal $z_i$ tussen 1 en $p$ - 1 is een unieke Jacobi-index te bepalen.
Dus ook met behulp van Ind() kunnen we vermenigvuldigen door op te tellen. In Bijlage 1 wordt nader op de theorie en constructie van de indices ingegaan.

Voor een verbeterde Verea-achtige rekenmachine hoeven we alleen de indices te bepalen van de 36 unieke getallen die voorkomen als eenvoudig product. We geven alleen de indices van de priemgetallen kleiner dan 10, dus van 1,2,3,5 en 7. De indices van de overige 31 unieke getallen volgen eenvoudig uit de indices van de priemgetallen.
Onderstaande tabel toont de indices van de priemgetallen voor de eerste twee sets die we volgens de constructie-methode van Bijlage 1 vinden:

| *z* | 1 | 2 | 3 | 5 | 7 |
|---|---|---|---|---|---|
| Ind(*z*) | 0 | 1 | 18 | 44 | 7 |
| Ind(*z*) | 0 | 1 | 8 | 44 | 27 |

Voor beide oplossingen is grootste index 88 (voor 5×5), en dat is dus een verslechtering ten opzichte van de primitieve één-gat-vermenigvuldigingstafel.

De Ierse accountant Percy Ludgate (1883-1922) heeft in 1909 een rekenmachine voorgesteld waarin ook een soort indices voorkomen.[7] Hij heeft deze indices zonder bovenstaande theoretische basis bepaald. De indices van de priemgetallen zijn:

| *z* | 1 | 2 | 3 | 5 | 7 |
|---|---|---|---|---|---|
| Ind(*z*) | 0 | 1 | 7 | 23 | 33 |

De grootste index is 66 (voor 7×7), en dat is dus een verbetering. Ludgate heeft niet de methode van Bijlage 1 gebruikt. Sterker nog, ik heb zijn indices niet via de methode van Bijlage 1 kunnen reconstrueren. Dat wil niet zeggen dat dat niet mogelijk is, maar ik weet ook niet of dat (getaltheoretisch) ook *nodig* is voor de indices van de "rare" set van 36 eenvoudige producten.
Waarschijnlijk heeft Ludgate de indices als volgt bepaald: vanzelfsprekend is Ind(1) = 0, vervolgens neemt hij Ind(2) = 1, dus Ind(4) = 2, Ind(8) = 3, ... Ind(64) = 6. De eerstvolgende ongebruikte index is 7. Het ligt voor de hand dit te kiezen als Ind(3). Dus Ind(3) = 7, Ind(2×3) = 8, ... Ind(3×16) = 11 (we gaan niet verder dan $z$ = 81). Ook de index-waarden 14, 21 en 28 = Ind(81) zijn nu bezet.
De waarden voor Ind(5) en Ind(7) zal Ludgate door trial-and-error hebben verkregen. Hij schrijft niet voor niets dat het hem "enige moeite" heeft gekost. In de Angelsaksische literatuur worden deze indices "Irish Logarithms" genoemd.

In 1913 publiceerde K. Hoecken een uitgebreid artikel over vermenigvuldigings-machines waarin wel de getaltheoretische basis wordt genoemd. Hoecken komt uiteindelijk op de zelfde tabel als Ludgate uit en geeft toe dat deze tabel puur empirisch is ontstaan. In een voetnoot die bij het ter perse gaan van zijn artikel is toegevoegd vermeldt hij nog twee andere empirische sets van indices:

| *z* | 1 | 2 | 3 | 5 | 7 | max Index | auteur |
|---|---|---|---|---|---|---|---|
| Ind(*z*) | 0 | 1 | 13 | 21 | 30 | 60 | Remak |
| Ind(*z*) | 0 | 8 | 13 | 1 | 30 | 60 | A. Korn |

[6] K. Hoecken, "Die Rechenmaschinen von Pascal bis zur Gegenwart, unter besonderer Berücksichtigung der Multiplikationsmechanismen", Sitzungsberichte Berliner Math. Gesellsch. Vol. 13, Feb. 1913, p.8-29. http://www.rechnerlexikon.de/files/HoeckMult.pdf
[7] B. Randell, "Ludgate's analytical machine of 1909", The Computer Journal, Vol. 14, No. 3, 317-326.

Deze oplossingen zijn nog iets beter. De oplossing van Korn wijkt af van alle anderen doordat Ind(2) ? 1. Ook deze indices heb ik niet via de methode uit Bijlage 1 kunnen reconstrueren. Bij Ludgate, Remak en Korn is Ind(49) maximaal terwijl bij “mijn” indices Ind(25) maximaal is.

## Rekenliniaal

Joh. Schumacher, Professor bij het Beierse Kadettenkorps, publiceerde in 1909, dus in hetzelfde jaar als Ludgate, een ontwerp voor een rekenliniaal met indices.[8] Deze liniaal is als Model 366 door A.W. Faber gefabriceerd[9] en bevat indices voor **alle** getallen tussen 1 en 100, dus niet alleen voor de 36 eenvoudige producten. In onderstaande tabel staan alleen de indices voor de priemgetallen < 101.

| *z* | 1 | 2 | 3 | 5 | 7 | 11 | 13 | 17 | 19 | 23 | 29 | 31 | 37 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ind(*z*) | 0 | 1 | 69 | 24 | 9 | 13 | 66 | 30 | 96 | 86 | 91 | 84 | 56 |
| *z* | 41 | 43 | 47 | 53 | 59 | 61 | 67 | 71 | 73 | 79 | 83 | 89 | 97 |
| Ind(*z*) | 45 | 42 | 58 | 23 | 29 | 77 | 81 | 44 | 61 | 64 | 89 | 21 | 52 |

Op de rekenliniaal zijn alle indices “modulo 100” getekend: Ind(9) = Ind(3×3) = 69 + 69 = 138 mod 100 = 38, dus bij het 39e streepje staat een 9 (bij Ind(1) = 0 staat het eerste streepje).
De gebruiker moet er rekening mee houden dat het product modulo 101 wordt genomen, en de som van de indices modulo 100:
Ind(23×23) = Ind(23) + Ind(23) = 86 + 86 = 172 = 72 mod 100 = Ind(24 mod 101)
De “modulo 100” wordt met de Schumacher-rekenliniaal op de zelfde manier uitgevoerd als een overgang naar een hogere orde op een normale rekenliniaal. Men leest dus af 23×23=24, en dat klopt “modulo 101”. In de praktijk wil men het echte product hebben, dus zal bij het resultaat nog een paar keer 101 opgeteld moeten worden. In dit geval verwachten we dat het product eindigt op 3×3=9. Er moet dus 9 - 4=5 keer 101 bij het resultaat worden opgeteld: 529. Het moge duidelijk zijn dat de educatieve waarde van deze rekenliniaal groter is dan de praktische.

Er zijn mij geen rekenlinialen bekend die alleen de “Ierse Logaritmen” voor de eenvoudige producten bevatten. Het nut ervan is ook beperkt: er kunnen alleen producten van twee getallen < 10 mee berekend worden. Om de lezer toch in staat te stellen hiermee te experimenteren heb ik rekenlinialen volgens Ludgate, Korn, Remak en mijn twee oplossingen als bijlage toegevoegd.[10] Ter vergelijking zijn ook de Schumacher-liniaal en de eenvoudige producten op een rekenliniaal met één of twee decades beschikbaar.[11]

## Epiloog

De “Ierse Logaritmen” zijn nooit in een echte rekenmachine toegepast. Ook de bruikbaarheid van de Schumacher rekenliniaal is beperkt. Het is des te verbazingwekkender dat A.W. Faber de moeite heeft genomen deze rekenliniaal in productie te nemen. Er is zelfs een schijfvormige versie van deze rekenliniaal ontworpen.[12] Deze rekenschijf is nooit in productie genomen[13]. De wereld was blijkbaar te klein voor twee Schumacher rekenlinialen.

## Bijlage 1: Jacobi Indices

De Jacobi-index Ind(*z*) van een geheel getal *z* kan worden gegeven door:

$$z = g^{\mathrm{Ind}(z)} \text{ modulo } p \qquad (1)$$

[8] Dr. Joh. Schumacher, “Ein Rechenschieber mit Teilung in gleiche Intervalle auf der Grundlage der zahlentheoretischen Indizes. Für den Unterricht konstruiert”, München, 1909.
[9] Dieter von Jezierski, Detlef Zerfowski, Paul Weinmann: “A.W. Faber Model 366 - System Schumacher. A Very Unusual Slide Rule”, Journal of the Oughtred Society Vol. 13, No. 2, 2004, p. 10-17.
[10] Ook online als PDF beschikbaar: http://home.telfort.nl/ajmdeman/irishlogrule.pdf
[11] Ook online als PDF beschikbaar: http://home.telfort.nl/ajmdeman/schumacherrule.pdf
[12] “Technik Geschichte: Beiträge zur Geschichte der Technik und Industrie”, VDI, 1933, p. 153-154.
[13] Dieter von Jezierski, Detlef Zerfowski, Paul Weinmann: “A.W. Faber Model 366 - System Schumacher. A Very Unusual Slide Rule”, Journal of the Oughtred Society Vol. 13, No. 2, 2004, p. 10-17.

waarbij het de kunst is waarden van *g* en *p* te vinden waarvoor deze vergelijking geldt en Ind(*z*) uniek is voor elke waarde *z* in een gekozen verzameling getallen. Als *p* een priemgetal is en de verzameling *z* uit alle getallen tussen 1 en *p* - 1 bestaat, dan is er een *g* waarvoor elke *z* een unieke Index tussen 0 en *p* - 2 heeft. Als dat gelukt is, dan geldt:

| | |
|---|---|
| $\mathrm{Ind}(z_1 \times z_2) = \mathrm{Ind}(z_1) + \mathrm{Ind}(z_2)$ | (2) |

Dus ook met behulp van Ind() kunnen we vermenigvuldigen door op te tellen[14].

Voor een Verea-achtige rekenmachine hebben we alleen de indices nodig van de 36 getallen die in de eenvoudige tafels van vermenigvuldiging voorkomen. Voor een rekenmachine wijkt de verzameling *z* dus af van de verzameling van alle getallen tussen 1 en *p* - 1: er zitten “gaten” in de verzameling en de verzameling eindigt niet bij een priemgetal ( *p* - 1 = 81).
Omdat we in de rekenmachine alleen getallen kleiner dan 10 vermenigvuldigen hoeft vergelijking (2) niet voor alle $z_1$ en $z_2$ te gelden, maar alleen voor $z_1$, $z_2$<10. De resulterende index $\mathrm{Ind}(z_1 \times z_2)$ moet wel uniek zijn voor alle unieke eenvoudige producten.
We hoeven de Jacobi-indices dus niet letterlijk toe te passen. We kunnen wel proberen of we via vergelijking (1) indices kunnen genereren die aan ons doel voldoen. Er is echter geen garantie dat de indices die we vinden de **kleinste** getallen opleveren waarvoor geldt $\mathrm{Ind}(z_1 \times z_2) = \mathrm{Ind}(z_1) + \mathrm{Ind}(z_2)$, en we willen juist dat de grootste index zo klein mogelijk is.
Met *p* = 11 en *g* = 2 heb ik via vergelijking (1) twee sets met indices gevonden waarbij de grootste index kleiner is dan 100. Onderstaande tabel toont de indices voor de priemgetallen < 10:

| *z* | 1 | 2 | 3 | 5 | 7 |
|---|---|---|---|---|---|
| Ind(*z*) | 0 | 1 | 18 | 44 | 7 |
| Ind(*z*) | 0 | 1 | 8 | 44 | 27 |

Andere keuzes van *p* en *g* leveren misschien betere (dus kleinere) indices op.
De indices van Schumacher’s rekenliniaal zijn gegenereerd met *p* = 101 en *g* = 2.

[14] Strikt genomen: de som van de indices modulo *p* - 1.