De Hamming-code

De wiskunde van het fouten verbeteren in digitale gegevens

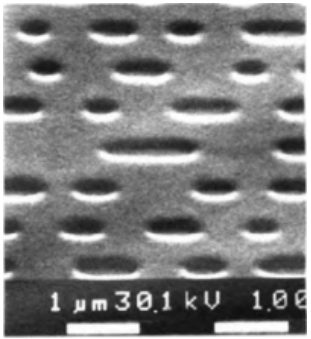
Hans Sterk, Benne de Weger

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Waar gaat coderen over?

Digitale opslag van gegevens gebeurt in bits - enen en nullen.
Dit komt omdat veel opslagtechnieken twee standen kennen: aan en uit.
Dit kan met heel veel soorten informatie:



- tekst: `.doc`, `.txt`, ...
- foto's en afbeeldingen: `.gif`, `.jpg`, ...
- muziek, video: `.mp3`, `.wmv`, ...
- software: `.exe`, ...

De uitdaging is:

- systematisch opslaan
- beschermen tegen foutjes
- efficiency

Actief werkgebied van wiskundigen in de afgelopen 50 jaar.

Beschermen tegen foutjes en efficiency

- **Zijn foutjes erg?**
  - Niet:
  - Wel:
- **Hoe bescherm je efficiënt tegen foutjes?**
  - Door listig (wiskunde!) extra informatie mee te sturen

(toppunt van domheid: elke boodschap tig keer versturen...)

Van tekst naar bits

- We bekijken vanaf nu alleen digitale opslag van teksten
- Tekst omzetten in ‘bits’ kan het beste per letter (want dan hoef je maar 26 verschillende codewoorden te hebben)
- Systematiek gewenst! (moet geautomatiseerd uit te voeren zijn)
  - Elke letter vervangen door rijtje nullen en enen van vaste lengte
  - Met rijtjes van lengte 5: $2^5 = 32$ mogelijkheden
- Iedere letter wordt zo een binair codewoord.

Voorbeeld: binaire codering alfabet

00000	↔	a	01101	↔	n	11010	↔	.
00001	↔	b	01110	↔	o	11011	↔	,
00010	↔	c	01111	↔	p	11100	↔	;
00011	↔	d	10000	↔	q	11101	↔	:
00100	↔	e	10001	↔	r	11110	↔	–
00101	↔	f	10010	↔	s	11111	↔	spatie
00110	↔	g	10011	↔	t			
00111	↔	h	10100	↔	u			
01000	↔	i	10101	↔	v			
01001	↔	j	10110	↔	w			
01010	↔	k	10111	↔	x			
01011	↔	l	11000	↔	y			
01100	↔	m	11001	↔	z			

Voorbeeld: coderen en decoderen

Coderen (binair maken) en decoderen (terugvertalen in gewone taal).

- Coderen van het woord plus:

p	l	u	s
↕	↕	↕	↕
01111	01011	10100	10010

Codering: `01111010111010010010`

- Decoderen: opsplitsen in groepen van vijf en in tabel opzoeken:

01111	01011	10100	10010
p	l	u	s

- Wat als er iets fout gaat: een `1` op het eind bijvoorbeeld?

01111	01011	10100	10011
p	l	u	t

Natuurlijke taal beschermt tegen fouten

Grondidee: als een gewoon woord qua spelling erg verschilt van elk ander woord, dan kun je in dat woord best een foutje maken.

- Bijvoorbeeld: ‘`wiskonde`’ $\longrightarrow$ ‘`wiskunde`’
  (dichtstbijzijnde woord uit woordenboek).
- Maar niet: ‘`mip`’ (hoort dit bij ‘`min`’, ‘`mep`’, bij ‘`sip`’, ... ?).

Een ‘fout’ woord verschilt op één (of weinig) plaatsen van het goede woord.
Dat ‘verschillen’ is een afstandsmaat.

Wanneer gaat fouten verbeteren goed?
Als de goede woorden ‘ver’ uit elkaar liggen.

Conclusie: bij een goede code kies je de woorden zo dat ze
altijd op meerdere plekken verschillen.

Nu binair...

Afstandsbegrip in de binaire wereld

De afstand tussen twee binaire rijtjes van dezelfde lengte is het aantal plaatsen waarop ze verschillen.

- $d(1001, 0011) = 2$ is kort voor:
  de rijtjes 1001 en 0011 verschillen op twee plaatsen
  (d van distance)
- Stel: 0000 en 1111 zijn de enige twee goede rijtjes (afstand ?). Je ontvangt 0100. Wat zou dan verzonden zijn? Wat zijn de afstanden tot 0000 en 1111?
- Stel je voor: drie rijtjes a, b en c met $d(a, c) = 4$, $d(a, b) = 1$. Wat kun je zeggen van $d(b, c)$?

Opgave A

Hoe helpt afstand tussen codewoorden?

Minimum-afstand 2:

- Stel je voor: je gebruikt een set codewoorden die twee aan twee steeds op minimaal 2 plaatsen verschillen.
- Bijvoorbeeld: $u = 110011$, $v = 110110$, ...
- Veronderstel: Bij verzending verandert één bit (niet meer) van zo'n woord, bijv. $u' = 110111$
- Is er dan een codewoord (goed woord) dat het dichtste bij u' ligt? Zeker weten?
- Wat is je conclusie?

Een code met minimumafstand 2 maken

Startpunt: $2^4 = 16$ rijtjes ter lengte 4, zoals 0000, 0001,... Afstand tussen twee van die rijtjes kan 1 zijn, jammer genoeg.

- **Dom:** codewoord = elk rijtje 1 maal herhalen, bijv $0010\ 0010$
- **Slim:** voeg aan elk rijtje 1 bit toe, een 0 of 1 zó dat:
  *) elk resulterend 5-bits rijtje een even aantal 1'en bevat.
  *) Bijv: 1000 verandert in 10001 en 1001 in 10010.
- Samen:
  *) 16 codewoorden (ter lengte 5)
  *) afstand tussen tweetal codewoorden: ≥ 2.
- Als er 1 fout gemaakt is, kun je die detecteren.
  Bijvoorbeeld: 10011 is fout...
- Maar corrigeren...

Opgave B

Hoe kan de afstand verder helpen?

Minimumafstand 3:

- Stel je voor: je gebruikt een set codewoorden die twee aan twee steeds op minimaal 3 plaatsen verschillen.
- Bijvoorbeeld: $u = 1100110$, $v = 1101011$, ...
- Bij verzending verandert één bit van zo'n woord, bijv $u' = 1101110$
- Is er dan een codewoord (goed woord) dat het dichtste bij ligt? Zeker weten?
- Wat is je conclusie?

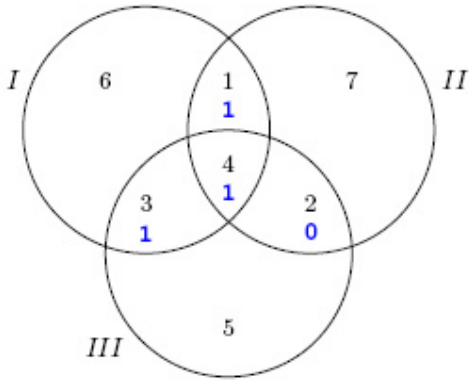
Een code met minimumafstand 3 maken

Startpunt: $2^4 = 16$ rijtjes ter lengte 4, zoals $\mathtt{0000}$, $\mathtt{0001}$,...

- 1 bit toevoegen is te weinig (afstand 2 komt voor)
- Kan het met 2 bits toevoegen? Nee.
  - stel aan $\mathtt{pqrs}$ voeg je $\mathtt{xy}$ toe, dus $a = \mathtt{pqrsxy}$ is een codewoord
  - noteer $\overline{\mathtt{x}}$ voor de bit die niet gelijk is aan $\mathtt{x}$
  - aan $\mathtt{pqr}\overline{\mathtt{s}}$ moet je dan wel $\overline{\mathtt{xy}}$ toevoegen, dus $b = \mathtt{pqr}\overline{\mathtt{s}}\overline{\mathtt{xy}}$ is een codewoord, want $d(a, b)$ moet ≥ 3
  - aan $\mathtt{pq}\overline{\mathtt{r}}\mathtt{s}$ moet je nu ook $\overline{\mathtt{xy}}$ toevoegen, dus $c = \mathtt{pq}\overline{\mathtt{r}}\mathtt{s}\overline{\mathtt{xy}}$ is een codewoord, want $d(a, c)$ moet ≥ 3
  - maar nu is $d(b, c) = 2$, en de minimumafstand kan dus niet 3 zijn
- Kan het eigenlijk wel? Ja, maar dus met toevoegen van ten minste 3 bits.
- **Dom:** codewoord = elk rijtje 2 maal herhalen, bijv $\mathtt{0010\ 0010\ 0010}$
- **Slim:** het kan met 3 bits extra (wiskunde!)

De Hamming-code: codewoorden ter lengte 7

Aan 4 (informatie)bits 3 controlebits toevoegen
Regel: in elk van de drie cirkels moet een even aantal enen komen.
Probeer maar met 1011.



(Richard Hamming)

Levert $2^4 = 16$ codewoorden ter lengte 7. Bijv: 1011010

En nu in formules...

0 en 1 vormen een eigen rekenwereld: binair rekenen.

- **Optellen:**
  - *) $0+0=0$, $1+0=1$, $0+1=1$ en $1+1=0$
  - *) Bijvoorbeeld: $1+1+1=(1+1)+1=0+1=1$.
    En: $1+1+1+1+1+1=0$.
- Even aantal enen tussen x_1, x_2, x_3, x_4, x_5 betekent:

$$x_1+x_2+x_3+x_4+x_5=0$$

Tel links en rechts x_5 op: $x_5=x_1+x_2+x_3+x_4$
(want $x_5+x_5=0$ ongeacht of $x_5=0$ of $x_5=1$)

Binair rekenen met rijtjes

Je kunt ook rijtjes (van gelijke lengte) optellen:

- $0101 + 1100 = 1001$ (pleksgewijs optellen dus)

Wanneer verschijnt er een 1 in de som?

Opgave C

Binair rekenen met rijtjes

Je kunt ook rijtjes (van gelijke lengte) optellen:

- $0101 + 1100 = 1001$ (pleksgewijs optellen dus)

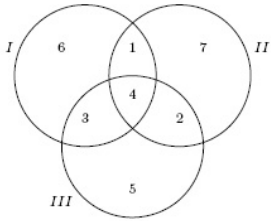
Wanneer verschijnt er een 1 in de som?

Opgave C

Conclusie: afstand tussen twee rijtjes is het aantal enen in de som

- $11110011 + 10110001 = 01000010$,
  dus $d(11110011, 10110001) = 2$

De Hamming-code: met vergelijkingen



Elke cirkel levert een vergelijking op. Samen:

$$
\begin{aligned}
I: \quad & x_1 + x_3 + x_4 + x_6 = 0 \longrightarrow x_6 = x_1 + x_3 + x_4 \\
II: \quad & x_1 + x_2 + x_4 + x_7 = 0 \longrightarrow x_7 = x_1 + x_2 + x_4 \\
III: \quad & x_2 + x_3 + x_4 + x_5 = 0 \longrightarrow x_5 = x_2 + x_3 + x_4
\end{aligned}
$$

Hieruit volgt:

Opgave D

- de som van twee codewoorden is ook een codewoord:

$$1100110 + 0010110 = 1110000$$

- Start maar met 2 codewoorden $(x_1, \ldots, x_7)$ en $(y_1, \ldots, y_7)$. Noem de som $(s_1, \ldots, s_7)$.
  Bijv. uit $x_5 = x_2 + x_3 + x_4$ en $y_5 = y_2 + y_3 + y_4$ volgt
  $s_5 = x_5 + y_5 = (x_2 + x_3 + x_4) + (y_2 + y_3 + y_4) = (x_2 + y_2) + (x_3 + y_3) + (x_4 + y_4) = s_2 + s_3 + s_4$.
  Net zo voor s_6 en s_7.

De Hamming-code heeft minimum-afstand 3

Als twee codewoorden afstand 1 of 2 zouden hebben, dan

- is hun som een codewoord,
- bevat hun som precies hoeveel 1'n ?
- en kijk nu eens in de vergelijkingen of dat kan...

$$\begin{array}{ccccccccccccc} & & x_2 & + & x_3 & + & x_4 & + & x_5 & & & & = 0 \\ x_1 & + & & + & x_3 & + & x_4 & + & & + & x_6 & & = 0 \\ x_1 & + & x_2 & + & & + & x_4 & + & & & & + \; x_7 & = 0 \end{array}$$

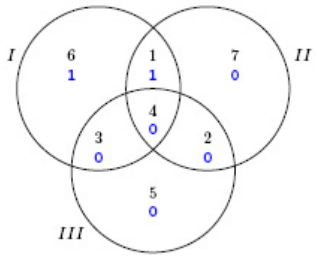
Wat heb je ook al weer aan afstand 3?

Met afstand 3 kun je 1 fout corrigeren, want

- als een codewoord op 1 plaats verandert is de afstand tot dat woord gelijk aan 1.
- En de afstand tot elk ander codewoord is minstens...
- Conclusie: De Hamming-code is 1-foutverbeterend.

En nu: decoderen

- Ontvangen: `1000010` (geen codewoord, want...)
- Vul in in de vergelijkingen...



$$
\begin{array}{lrcl}
I: & x_1 + x_3 + x_4 + x_6 & = & 0 \\
 & 1 + 0 + 0 + 1 & = & 0 \text{ ok} \\
II: & x_1 + x_2 + x_4 + x_7 & = & 0 \\
 & 1 + 0 + 0 + 0 & = & 1 \text{ fout} \\
III: & x_2 + x_3 + x_4 + x_5 & = & 0 \\
 & 0 + 0 + 0 + 0 & = & 0 \text{ ok}
\end{array}
$$

- Je mag 1 bit veranderen om een codewoord te krijgen...
- Conclusie: ...

Decoderen in formules:

Het volgende recept werkt:

- bereken de **bits**:

$$\begin{aligned} b_1 &= x_4 + x_5 + x_6 + x_7 \\ b_2 &= x_2 + x_3 + x_6 + x_7 \\ b_3 &= x_1 + x_3 + x_5 + x_7 \end{aligned}$$

(slechts één van de x_i'tjes is 1)

	b_1	b_2	b_3
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

(dus: uitkomst is drie nullen of enen)

- in het foute codewoord moet bit nummer $4b_1 + 2b_2 + b_3$ worden veranderd

Voorbeeld: `0001001` geeft $b_1 = 0, b_2 = 1, b_1 = 1$
en $4 \times 0 + 2 \times 1 + 1 = 3$, dus het derde bit was fout
het juiste codewoord was `0011001`

Opgave E

Echte codes

Bij muziek-CD's gebruikt men een listige combinatie van twee codes: de ene code heeft codewoorden ter lengte 28, de andere ter lengte 32. Samen hebben ze een enorme foutenverbeterende capaciteit.

Eindhovense inbreng (aan de basis van de CD):

- dr Kees Schouhamer Immink (destijds Philips, Emmy Award 2003!),
- prof. Jack van Lint † (TU/e)