

Breuken II (Rekenen met gehele getallen)

V2

Jan Bergstra
Minstroom Research, Utrecht

May 25, 2025

1 Rekenen met hele getallen, RmHG

Doel van dit werk is om een systematische aanpak van het rekenen met gehele getallen te propageren waarbij logisch redeneren in beeld blijft. Heel getal en geheel getal staat voor hetzelfde en wordt afwisselend gebruikt. We werken met decimale natuurlijke getallen en decimale gehele getallen volgens de conventies besproken in [2]

1.1 Cijferseries

De cijfers die we gebruiken zijn $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$, een cijferserie is een niet-lege rij van cijfers, b.v. 83701 .

(i) Een positief geheel getal is een cijferserie die niet met 0 begint, b.v. 2 of 34 .

(ii) Een niet-negatief geheel getal is 0 dan wel een positief geheel getal. De positieve en ook de niet-negatieve hele getallen zijn alle ook termen.

(iii) Een heel getal is hetzij een niet-negatief heel getal of een term van de vorm $-p$ met p een positief heel getal.

Deze omschrijving van hele getallen is een keuze die wij voor een eerste kennismaking met hele getallen bruikbaar vinden. Gehele getallen zijn niet uniek vastgelegd maar zijn in een meer algemene aanpak elementen van structuren die aan bepaalde regels voldoen. Vast te stellen welke structuren dat zijn is geen eenvoudige opgave. Met de bovenstaande definities bouwen we aan een context waarin het begrip getal zo concreet mogelijk is in de verwachting dat daarmee de taal van het elementaire rekenen zo ver mogelijk kan worden vereenvoudigd.

1.2 Waardegelijkheid versus vormgelijkheid

Er zijn ook cijferseries die wel met een 0 beginnen, b.v. het fameuze 007. Zulke cijferseries zijn geen getallen maar zijn wel expressies voor getallen. Cijferseries kan men vereenvoudigen door de leidende nullen te verwijderen, zolang er nog een cijfer overblijft. Vereenvoudiging van 007 levert dan 7 op. Het hele getal 7 is de waarde van de cijferserie 007. We kunnen dus schrijven $007 = 07 = 7$ omdat deze drie cijferseries elk dezelfde waarde (nl. 7) hebben. Om te beginnen schrijven we $007 =_w 07 =_w 7$. Hiermee zien we het contrast met $=_v$ (vormgelijkheid, ofwel hetzelfde zijn), want $007 \neq_v 07 \neq_v 7$. ENG voor vormgelijkheid is structural equivalence met bijbehorende notatie $=_{\text{str}}$, en ook in het NL gebruiken we voor de eenvoud voortaan liever $=_{\text{str}}$ dan $=_v$.

1.3 Default gelijkheid

Met $=$ bedoelen we $=_w$ tenzij anders wordt aangegeven. Dit betekent dat de betekenis van $=$ is vast gelegd by default, ofwel bij afwezigheid van extra gegevens kiezen we voor $=_w$.

2 Optellen en vermenigvuldigen

Om te beginnen met optellen is er nog een ander idee nodig: $\perp$ staat voor “weet niet”. Dat is niet zozeer “ik weet het niet” als wel: we hebben afgesproken dat we het niet weten. $\perp$ is een zgn. perifeer getal, niet een “echt getal” om zo te zeggen. Maar werken met $\perp$ heeft allerlei voordelen.

We gaan er nu van uit dat we met een groep leerlingen leren rekenen met hele getallen. Wij gaan $\perp$ gebruiken voor tijdelijke afspraken, b.v. $8 + 3 = \perp$ die afspraak is praktisch als we willen rekenen met alleen de cijfers 1, 2, 3, 4, 5, 6, 7, 8, 9. Als een optelling meer dan 9 oplevert dan spreken we af dat er $\perp$ uitkomt. Die afspraak geldt voor de hele groep en die laten we over een tijdje vallen als alle leden van de groep kunnen rekenen met getallen met een enkel cijfer ongelijk 0. Dan gaan we rekenen met getallen met 1 of 2 twee cijfers zoals 8 en 21.

De structuur $\mathbb{Z}_{\text{dec}}^k(+, -)$ heeft als operaties optellen en minus, met getallen die met ten hoogste k decimalen kunnen worden geschreven waarbij het minusteken niet meetelt. We noemen zo’n structuur een rekenstructuur (ENG: arithmetic). Dit levert een domein $[-10^k + 1, 10^k - 1] \cup \{\perp\}$ met $2 \cdot 10^k$ elementen. Bij $k = 1$ vinden we nu de structuur $\mathbb{Z}_{\text{dec}}^1(+, -)$ met als domein de verzameling $\{-9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \perp\}$.

In $\mathbb{Z}_{\text{dec}}^1(+, -)$ hebben we b.v. $2 + 7 = 9$ en $2 + 9 = \perp$, $(5 + 5) + (-4) = \perp - 4 = \perp$, $5 + (5 + (-4)) = 5 + 1 = 6$. In deze eindige structuur is optellen nog wel commutatief maar niet meer associatief.

Vermenigvuldiging $(\cdot)$ kan worden toegevoegd en dat levert $\mathbb{Z}_{\text{dec}}^k(+, -, \cdot)$. Bij $k = 1$ vinden we $3 \cdot 3 = 9$ en $3 \cdot 4 = \perp$.

$$\lim_{x \rightarrow \infty} \mathbb{Z}_{\text{dec}}^k(+, -, \cdot) \cong \text{Enl}_{\perp}(\mathbb{Q})$$

Hier is $\text{Enl}_\perp(\mathbb{Q})$ de zgn. $\perp$ -enlargement ($\perp$ -vergroting) van $\mathbb{Q}$ (in de notatie van [3])

2.1 Rekenen in $\mathbb{Z}_{\text{dec}}^1(+, -, \cdot)$, in $\mathbb{Z}_{\text{dec}}^2(+, -, \cdot)$, in $\mathbb{Z}_{\text{dec}}^3(+, -, \cdot), \dots$

Het belang van werken met eindige rekenstructuren blijkt uit de volgende drie overwegingen:

(i) de keuze van het aantal decimalen k geeft duidelijk aan dat er altijd sprake is van ontwerpkeuzen, en dat men zowel het bestaan van die keuzen als de uitkomst daarvan bij voorkeur expliciet moet maken. Ook het achterwege laten van de keuze voor een specifieke k en werken met een oneindige rekenstructuur is een ontwerpkeuze;

(ii) het wordt duidelijk dat er van bootstrapping sprake is: om $\mathbb{Z}_{\text{dec}}^k(+, -)$ te verklaren is k nodig. Het is niet onredelijk om k te kiezen b.v. uit $\mathbb{Z}_{\text{dec}}^2(+, -)$ en aan te nemen dat we 2 kennen.

(iii) Als men met een klas gaat oefenen met rekenen dan is het b.v. zinvol om eerst in $\mathbb{Z}_{\text{dec}}^1(+, -)$ te werken, daarna in $\mathbb{Z}_{\text{dec}}^2(+, -)$, en daarna in $\mathbb{Z}_{\text{dec}}^3(+, -)$ en te zeggen dat het met $\mathbb{Z}_{\text{dec}}^4(+, -)$ “net zo” werkt. Van belang is nu dat door duidelijke afspraken te hanteren alle studenten in dezelfde condities werken. ook als een leerling L al weet hoe je 25 en 89 moet optellen in $\mathbb{Z}_{\text{dec}}^3(+, -)$, als de hele klas werkt aan rekenen in de rekenstructuur $\mathbb{Z}_{\text{dec}}^2(+, -)$ dan heeft de voorsprong van L qua kennis op medestudenten (die nog niet weten dat en hoe je in $\mathbb{Z}_{\text{dec}}^3(+, -)$ meer kun uitrekenen dan in $\mathbb{Z}_{\text{dec}}^2(+, -)$) geen relevantie voor de juistheid van antwoorden die moeten worden gegeven want ook voor L is het gewenste en goede antwoord: $25 + 89 = \perp$. Student L moet in de klas accepteren dat de gemeenschappelijke afspraak dat $k = 2$ en dat wordt gerekend in de rekenstructuur $\mathbb{Z}_{\text{dec}}^2(+, -)$ zwaarder weegt dan de eigen kennis van zaken waarmee L de plausibiliteit van de keuze dat $k = 2$ ter discussie kan en wil stellen. Er ontstaat op deze wijze een level playing field, en dat is niet kunstmatig, een level playing field ontstaat altijd op basis van afspraken waarbij bepaalde voordelen en nadelen worden opgeheven of geëgaliseerd.

2.2 Een familie van shapes en labels: begrensde decimale gehele getallen voor het label begrensde gehele getallen

We noemen “gehele getallen” een label, en “decimale gehele getallen” noemen we een shape voor dat label. Zo’n label geeft een soort (in informele zin, niet in de zin van de logica) van getallen aan, en de shape karakteriseert een vormgeving van (de getallen van) een label.

De getallen in $\mathbb{Z}_{\text{dec}}^k(+, -)$ horen bij een label begrensde gehele getallen vormen ook een shape: begrensde decimale gehele getallen. Label en shape hebben beide een parameter, de grootte van de grens (hier 10^k).

3 Gretige gelijkheid

Het is jammergenoeg niet zo dat alle belangrijke gelijkheden in rekenstructuren blijven gelden, met name de associativiteit van de optelling gaat verloren. Hierbij valt op te merken dat de notie van gretige gelijkheid ([4]) daar op inspeelt, want onder gretige gelijkheid zijn de gewone vergelijkingen van de rekenkunde allemaal ook geldig in elke eindige rekenstructuur van de vorm $\mathbb{Z}_{\text{dec}}^2(+, -)$ voor $k \geq 1$.

References

- [1] J.A. Bergstra. Arithmetical datatypes, fracterms, and the fraction definition problem. *Transmathematica*, ISSN 2632-9212, (published 2020-04-30), (2020). <https://doi.org/10.36285/tm.33>
- [2] J.A. Bergstra. Sumterms, summands, sumtuples, and sums and the meta-arithmetic of summation. *Scientific Annals of Computer Science*, 30 (2), 167-203, (2020), doi: 10.7561/SACS.2020.2.167.
- [3] Jan A. Bergstra and John V. Tucker. 2022. Totalising partial algebras: Teams and splinters. *Transmathematica*, <https://doi.org/10.36285/tm.57>
- [4] J.A. Bergstra and J.V. Tucker. Eager equality for rational number arithmetic.. *ACM Transactions on Computational Logic*, 24 (3), article no. 22 p. 1-28, (2023). <https://doi.org/10.1145/3580365>