Wat zijn decimale gehele getallen? over de shapes $\text{DNNI}_{\equiv}$ en DNNI

Jan Bergstra, Minstroom Research, Utrecht
janaldertb@gmail.com

1 november 2026

1 Inleiding

Na de introductie van DNN's (ofwel DNN-naturals) in [5] is een volgende stap dat we gaan kijken naar de minus operatie, ook wel aftrekken genoemd. Dat gaat dan op een leuke manier via een tussenstap, de zogemaande minpairs (minparen). Voor twee DNN's a en b is $a \multimap b$ een minpair. Concreet bijvoorbeeld $71 \multimap 2$ en $0 \multimap 13$. Minpairs kunnen we bijvoorbeeld gebruiken om aan te geven hoe je een combinatie van blokken kunt veranderen. De notatie voor minpairs komt uit [2].

2 Blokken en rijen van blokken

We gaan ee wereld van blokken bekijken. De blokken hebben alle de zelfde vorm, een kubus, en dezelfde kleur, zwart. Blokken B kun je naast elkaar zetten met $-||-$, dat levert reeksen van blokken van verschillende lengte zoals: B, $B||B$, $B||B||B$, $B||B||B||B$. Met $-||-$ zetten we de blokken dus alleen maar op een rij, en verder niet in de andere richtingen naast of op elkaar.

2.1 Tellen en optellen

Het tellen van blokken kan met DNN's. Het is van belang dit te leren. Eigenlijk hoef je dan niet te leren om op te tellen met DNN's. We schrijven $\#(-)$ voor "het aantal van". Als je twee rijen blokken R_1 en R_2 hebt dan is $R_1||R_2$ ook een rij en we zo dat

$$\#(R_1||R_2) = \#(R_1) + \#(R_2)$$

Met dit idee bij de hand en een grote verzameling blokken kunnen we DNN's a en b optellen: (i) maak een rij R_1 met $\#(R_1) = a$ en maak een rij R_2 met $\#(R_2) = b$ en (ii) stel dan beide rijen same: $R_1||R_2$, en (iii) bepaal de lengte $\#(R_1||R_2)$ in DNN, dat is de som $a + b$ van a en b. Eigenlijk hoeven we niet te kunnen optellen als we dit principe kennen, het is voldoende om veel blokken te

© J.A. Bergstra 2026 Licence CC BY-SA 4.0

hebben, om die blokken in de rij te kunnen zetten en om de omvang van de rij te kunnen tellen met een resultaat in DNN.

Dat veel mensen weten dat je DNN-naturals vrij gemakkelijk kunt optellen zonder met rijen van blokken te schuiven is voor het goede begrip vann de zaak misschien eerder een nadeel dan een voordeel. Het intuïieve begrip van optellen krijg je in de wereld van de blokken misschien eerder dan iin de wereld van het cijferen .

2.2 Minpairs

Een minpair is een tweetal DNN's, we schrijven zeg $12 -\!\!\bullet 34$ voor het minpair met (positief of eerste deel) 12 en tweede deel 34. Een minpair heeft niet zomaar een vaste betekenis, informeel is de betekenis van $a -\!\!\bullet b$ zoiets als a erbij en b eraf. Dat is informeel en laat nog mogelijkheden open. Het is de context die bepaalt hoe het werkt.

Waarbij en waaraf dat hangt af van de context, de volgorde, eerst erbij of eraf, hangt ook af van de context. We gaan verschillende manieren bekijken om erbij en eraf te begrijpen, d.w.z. om te begrijpen hoe erbij en eraf zou kunnen werken.

3 Blokrijen Z

Met blokrijen a bedoelen we een manier om minpairs op blokrijen toe te passen.

Bij blokrijen Z, hebben we alleen zwarte blokken. In deze situatie kun je een minpair, zeg $3 -\!\!\bullet 4$ op een reeks blokken toepassen, en dat gaat als volgt:

(i) Haal van links (d.w.z. het begin van de rij) af 4 blokken weg. Als er minder dan 4 blokken in de rij staan is het resultaat een lege rij, en dat resultaat noteren we met [ ]. Verder wegalen vam blokken verandert dan niks, de blokrij blijft leeg.

(ii) plaats dan 3 blokken terug (die blokken worden er weer bijgezet aan het begin van de rij).

We schrijven dit op met $(3 -\!\!\bullet 4) \circ_{\mathsf{Z}} (B||B||B||B||B)$ bij toepassing op een rij van 5 blokken. We krijgen zo:

$$(3 -\!\!\bullet 4) \circ_{\mathsf{Z}} (B||B||B||B||B) = B||B||B||B||B$$

$$(2 -\!\!\bullet 1) \circ_{\mathsf{Z}} (B||B||B) = B||B||B||B$$

$$(2 -\!\!\bullet 3) \circ_{\mathsf{Z}} (B||B) = [\,]$$

$$(3 -\!\!\bullet 1) \circ_{\mathsf{Z}} [\,] = B||B||B$$

$$(3 -\!\!\bullet 2) \circ_{\mathsf{Z}} B = B||B||B$$

We zien nu dat $(3 -\!\!\bullet 2) \circ_{\mathsf{Z}} B = B||B||B$ en $(2 -\!\!\bullet 1) \circ_{\mathsf{Z}} B = B||B \neq B||B||B$ en we mogen constateren dat $3 -\!\!\bullet 2$ en $2 -\!\!\bullet 1$ niet hetzelfde zijn (want zou dat zou zijn dan zou toepassen op $B||B||BB$ ook hetzelfde op moeten leveren, en dat is niet het geval). Je kunt de bewerkingen $(a -\!\!\bullet b) \circ_{\mathsf{Z}} -$ en $(c -\!\!\bullet d) \circ_{\mathsf{Z}} -$

na elkaar uitvoeren, bijvoorbeeld op X (een willekeurige rij van blokken) en dat levert (als je eerst $(c \multimap d) \circ_{\mathsf{Z}} -$ uitvoert) het volgende op:

$$(a \multimap b) \circ_{\mathsf{Z}} ((c \multimap d) \circ_{\mathsf{Z}} X)$$

en we kunnen dat ook anders noteren:

$$((a \multimap b) \circ (c \multimap d)) \circ_{\mathsf{Z}} X = (a \multimap b) \circ ((c \multimap d) \circ_{\mathsf{Z}} X)$$

We krijgen expressies voor operaties op rijen van blokken en en voor zulke expressies hebben we dan gelijkheid als twee expressies op elke rij van blokken hetzelfde opleveren. We vinden allerlei gelijkheden die je ook in de praktijk met de hand kunt controleren.

$$(2 \multimap 5) \circ (3 \multimap 8) = 2 \multimap 10$$

$$(a \multimap b) \circ (b \multimap c) = a \multimap c$$

$$a \multimap b = (a \multimap 0) \circ (0 \multimap b)$$

3.1 De monus operatie

Als je 10 blokken hebt en je haalt er drie weg dan zijn er 7 blokken over. Als je 8 blokken hebt en je haalt er 10 weg dan houd je geen blokken over. In de wereld van het weghalen van blokken: als je van a blokken er b weghaalt dan houd je er $a \dot{-} b$ over. De bewerking $\dot{-}$ heet monus. We hebben bijvoorbeeld $8 \dot{-} 5 = 3, 20 \dot{-} 7 = 13$ en $2 \dot{-} 3 = 0$. De samenstelling van twee minpairs (gezien als bewerkingen op rijen van blokken) doet het zelfde als een enkele minpair die je met behulp van de monus kunt opschrijven.

$$(a \multimap b) \circ (c \multimap d) = ((a + (c \dot{-} b)) \multimap (d + (b \dot{-} c))$$

Het controleren van deze bewering kan empirisch gebeuren door een aantal viertallen van waarden in te vullen voor a, b, c, d en door LRS en RHS die je zo krijgt toe te passen op verschillende rijen van blokken.

Op dit moment zie ik deze bewering als een voorspelling van wat er gebeurt als je dit in de praktijk gaat proberen, met blokken op tafel. Misschien is het wel fout en dan is er een tegenvoorbeeld, misschien is dat tegenvoorbeeld zo groot dat het moeilijk is om het te vinden (als het bestaat). Pas na heel wat proberen in de praktijk komt aan de orde of de wiskunde van de zaak toelaat hierover iets te zeggen of zo men wil te bewijzen en hoe we doen dat zou moeten als we dat eventueel zouden willen.

4 Blokrijen RZ: werken met zwarte en rode blokken

We gaan de wereld van de blokken iets ingewikkelder maken, er zijn van nu af aan zwarte blokken en rode blokken. De operatie $(a \multimap b) \circ_{\mathsf{RZ}} X$ werkt nu anders op een rij X van blokken. Rode en zwarte blokken staan gewoon door elkaar.

(i) Zet er b zwarte blokken bij, en zet er a rode blokken bij..

(ii) Zo vaak als het kan: als er nog een rood blok is en nog een zwart blok is, haal een rood en een zwart blok weg. (Op het einde hiervan zijn er geen rode blokken meer, geen zwarte blokken meer of helemaal geen blokken meer.)

Met deze definitie van $\circ_{\mathsf{RZ}}$ op rijen van rode en zwarte dozen hebben we: $(a \multimap\!\bullet\, a) \circ_{\mathsf{RZ}} X = X$ en:

$$(a \multimap\!\bullet\, b) \circ_{\mathsf{RZ}} ((c \multimap\!\bullet\, d) \circ_{\mathsf{RZ}} X) = ((a+c) \multimap\!\bullet\, (b+d)) \circ_{\mathsf{RZ}} X$$

Met $\circ$ als compositienotatie voor de $\circ_{\mathsf{RZ}}$-applicatie krijgen we dan:

$$(a \multimap\!\bullet\, b) \circ_{\mathsf{RZ}} ((c \multimap\!\bullet\, d) \circ_{\mathsf{RZ}} X) = ((a \multimap\!\bullet\, b) \circ (c \multimap\!\bullet\, d)) \circ_{\mathsf{RZ}} X$$

en (als zgn. functionaalvergelijking) ook:

$$(a \multimap\!\bullet\, b) \circ (c \multimap\!\bullet\, d) = (a+c) \multimap\!\bullet\, (b+d)$$

Bovenstaande vergelijking zegt iets over hoe we willen werken met het toepassen van minpairs op een rij of constellatie van blokken. Alleen alle informatie over de keuze van die setting is uit de vergelijking verdwenen. Op het ook is hier een inconsistentie met de vergelijking die we in Paragraaf 3.1 hebben gevonden. Die vergelijking is gebasserd op een andere aanname over hoe we de blokkenrij aanpassen bij toepassing van een minpair daarop.

5 Blokrijen Z-$\bot$

De werking van minpairs op een context van blokken kan op allerelei manieren anders, bijvoorbeeld met alleen zwarte blokken maar als de rij leeg is en er nog meer uit gehaald zouden moeten worden ontstaat het resultaat $\bot$, een foutwaarde. Dit mecahnisme noemen we blokrijen Z-$\bot$. Voor gebruik van $\bot$ in elementair rekenen zie Bergstra 2024 [1]. Dan krijg je b.v.

$$(3 \multimap\!\bullet\, 4) \circ_{\mathsf{Z},\bot} (B||B||B) = \bot$$

en

$$(1 \multimap\!\bullet\, 2) \circ_{\mathsf{Z},\bot} (\bot) = \bot$$

maar ook

$$(3 \multimap\!\bullet\, 2) \circ_{\mathsf{Z},\bot} (B||B) = B||B||B$$

5.1 Blokrijen Z-max

Zonder details te geven, met blokrijen Z-max bedoelen we een situatie waarin maximaal zeg K blokken geplaatst kunnen worden, teveel blokken worden gewoon niet geplaatst. Dan krijg je, met $K = 5$:

$$(7 \multimap\!\bullet\, 3) \circ_{\mathsf{Z},\mathsf{max},5} (B||B||B||B||B) = B||B||B||B||B$$

en

$$(1 -\!\!\bullet 2) \circ_{\mathsf{Z},\mathsf{max},5} (\bot) = \bot$$

en

$$(1 -\!\!\bullet 2) \circ_{\mathsf{Z},\mathsf{max},5} (B) = \bot$$

maar ook

$$(3 -\!\!\bullet 2) \circ_{\mathsf{Z},\mathsf{max},5} (B||B) = B||B||B$$

Je ziet dat er weer veschillende opties ontstaan, hierboven nemen we aan dat weghalen van een blok bij een lege rij $\bot$;evert maar dat zou ook [ ] kunnen zijn, en dat levert weer een iets ander mechanisme om met de blokken te werken.

6 DNNI: decimal integers ofwel decimale gehelen

Decimal integers, ofwel decimale gehelen, of ook DNNI gehele getallen, krijgen we als volgt: voor elk positief DNN-natural a is het sign $-a$ een negatief DNNI-integer. Door de negatieve DNNI integers toe te voegen aan DNN krijgt men DNNI. In totaal zijn de DNN-integers de DNN-naturals gecombineerd (verenigd) met de negative DNN-integers (zoals hierbover aangegeven, met de vorm $-a$ waarbij a een DNN (lees DNN-natural) ongelijk 0 is). We duiden DNNI gehelen ook aan met DNNI.

We zien -17 als een DNNI, en dat levert ook (de noties op van) een DNNI-term, een DNNI-sign en een DNNI-token. Als opgeschreven dan is -18 een DNNI-token, voor een DNNI-sign dat zelf een vormgeving levert voor een DNNI-term, die weer een notatie is voor een DNNI-value (en dat is dan een DNNI). Wij zien de DNNI-signs, net zoals de DN signs, tevens als DNNI-values. Dat matcht met de beschrijving van DNN als signs voor naturals.

Dit jargon is heavy going en we gebruiken de meeste van deze termen niet of nauwelijks. Waar het om gaat is te weten dat de termen bestaan voor de desbetreffende noties en dat er niet steeds gedacht hoeft te worden dat bekende termen zoals “som” en “verschil” volstaan om precies te besprekenm hoe e.e.a. in elkaar zit. Zulke termen zijn voor dagelijks werk meestal wel toereikend, maar voor een precieze uiteenzetting over het rekenen is soms meer nodig. DNNI zie ik als de meest voor de hand liggende shape voor gehele getallen. Maar met minpairs biuj de hand onstaat direct een andere idee voor een shape voor gehelen.

6.1 De shape $\mathbf{DNNI}_{\equiv}$ voor gehele getallen

De shape $\mathrm{DNNI}_{\equiv}$ is een shape voor (het label van) gehele getallen. Voor de shape $\mathrm{DNNI}_{\equiv}$ zijn de getallen equivaletieklassen van gesloten minpairs onder de volgende equivalentierelatie $\equiv$.

Definition 6.1. *Voor gesloten minpairs $a -\!\!\bullet b$ en $c -\!\!\bullet d$ definiëren we equivalentie als volgt: $a -\!\!\bullet b \equiv c -\!\!\bullet d$ is geldig precies wanneer $a + c = b + d$.*

Deze equivalentierelatie kunnen we ons als volgt voorstellen. Een oneindige lijn bevat punten op gelijke afstand. In het midden is een punt p_0, naar rechts zijn er de punten q_1, q_2, q_3 etc., naar links zijn er vanaf p_0 geteld, de punten r_1, r_2, r_3 etc. Een object A staat steeds op één van de genoemde punten. Een minpair $a -\!\!\bullet b$ kunnen we toepassen op A, notatie $(n -\!\!\bullet m) \circ A$ by first moving A, n positions naar rechts en vervolgens m positions naar links. Nu is de equivalentie $n -\!\!\bullet m \equiv u -\!\!\bullet v$ aan de orde als voor elke positie van A, $(n -\!\!\bullet m) \circ A = (u -\!\!\bullet v) \circ A$. Een equivalentieklasse van een minpair noteren we als $[5 -\!\!\bullet 9]_{\equiv}$. De constanten en operaties werken als volgt:

$$[\![0]\!]_{\mathsf{DNNI}_{\equiv}} = [0 -\!\!\bullet 0]_{\equiv}, [\![1]\!]_{\mathsf{DNNIE}} = [1 -\!\!\bullet 0]_{\equiv}, \text{ for } a \in \mathrm{DNN}\ [\![a]\!]_{\mathsf{DNNI}_{\equiv}} = [a -\!\!\bullet 0]_{\equiv}$$

$$[a -\!\!\bullet b]_{\equiv} + [c -\!\!\bullet d]_{\equiv} = [(a + c) -\!\!\bullet (b + d)]_{\equiv}$$

$$[a -\!\!\bullet b]_{\equiv} - [c -\!\!\bullet d]_{\equiv} = [(a + d) -\!\!\bullet (b + c)]_{\equiv}$$

$$[a -\!\!\bullet b]_{\equiv} \cdot [c -\!\!\bullet d]_{\equiv} = [(a \cdot c + b \cdot d) -\!\!\bullet (a \cdot d + b \cdot c)]_{\equiv}$$

Claim 6.1. *Equivalentieklassen van minpairs onder $\equiv$ zijn een bruikbaar domein voor (het label van) de gehele getallen. (In de terminologie van [4] leveren deze equivalentieklassen een shape voor het label integer, en deze shape noemen we $DNNI_{\equiv}$.)*

6.2 Reshaping naturals

Nu hebben we een tweede notatie voor, zeg 3 te weten $[3 \cdot 0]_{\equiv}$. Leidt dit tot een infinite regress, ofwel, de suggestie dat we het eigenlijk over $[[3 \cdot 0]_{\equiv} \cdot 0]_{\equiv}$ hebben, en over $[[[3 \cdot 0]_{\equiv} \cdot 0]_{\equiv} \cdot 0]_{\equiv}$ enz? Nee, dat laatste is niet de situatie want in $[3 \cdot 0]_{\equiv}$ staat de 3 voor een natuurlijk getal van shape DNN. Wanneer we met de shape DNNIE gaan werken en daarmee een nieuwe shape invoeren voor naturals en voor integers dan blijft op de achtergrond de shape DNN bestaan en deze wordt ook gebruikt.

Reshaping voor een label L is de introductie van een andere of nieuwe shape daarvoor. Bij reshaping gebeurt het volgende: (i) neem aan dat met sshape S_L voor label L wordt gewerkt, (ii) met behulp van bestaande shapes, waaronder S_L, wordt een nieuwe shape, zeg S_L^n gemaakt voor label L, (ii) de values van termen worden nu andere objecten: het is plausibel dat voor een gesloten term t,

$$[\![t]\!]_{S_L^n} \neq [\![t]\!]_{S_L}$$

De ongelijkheid hierboven is op ontologisch niveau, waarbij objecten uit verschillende shapes met elkaar wordden vergeleken. In gewoon rekenen komt dat niet voor omdat je steeds werkt onder de aanname dat een shape vastligt, ook al doet het er meestal niet toe welke shape je dan kiest, en is het ook geen probleem als verschillende mensen die samenwerken verschillende shapes hanteren.

7 Wat hiervan te denken?

Waarom is dit allemaal van belang, en hebben we met $\text{DNNI}_{\equiv}$ nu een "goede" shape voor gehelen in handen? Ik merk dit op:

(i) $\text{DNNI}_{\equiv}$-integers zijn minpair equivalentieklassen, met de drie hierboven beschreven operaties. Dit levert een bruikbare shape voor het label integer.

(ii) $\text{DNNI}_{\equiv}$-integers zijn fundamenteel omdat de beschrijving van de operaties meer uniform is.

(iii) Als we de $\text{DNNI}_{\equiv}$ als shape voor het label integer kiezen, dan krijgen de elementen van DNN (te weten signs voor integers) bij elkaar de status van een additionele soort, een soort die gebruikt wordt en om zo te zeggen meehelpt om $\text{DNNI}_{\equiv}$ te beschrijven. Het is niet zinvol om signs die lijken op ontologische uitdrukkingen zoals $[7 -\!\!\bullet\, 3]_{\equiv}$ te gaan ontwerpen en gebruiken, dat is voor de praktijk veel te omslachtig. De ambiguïteit van DNNI-tokens neemt daarmee toe, want deze kunnen ook staan voor een entiteit van de vorm $[a -\!\!\bullet\, b]_{\equiv}$.

(iv) De getallen in $\text{DNNI}_{\equiv}$ zijn tamelijk abstracte "objecten", want $[7 -\!\!\bullet\, 3]_{\equiv}$ is een oneindige verzameling, een entiteit waarvan men zich het bestaan alleen binnen een universum van verzamelingen kan voorstellen. Die abstracties worden toch als dingen gezien, en dat proces, van een abstractie tot object te maken, noemt men ook reïficatie.

(v) Het belang van $\text{DNNI}_{\equiv}$ ligt met name in het feit dat DNN met optelling een een semigroep vormt en dat het werken met equivalentieklassen van paren (wat gebeurt bij $\text{DNNI}_{\equiv}$) dan een algemene, d.w.z. altijd toepasbare methode levert om een inverse toe te voegen. De inverse is dan de verwisseling van het paar, terwijl de groepsbewerking per element wordt gedaan. Als we de bewerking van de semigroep met $- \star -$ schrijven (en $-^{-1}$ voor de inverse daarvan) dan krijg je $(a, b) \star (c, d) = (a \star c, b \star d)$ en $(a, b)^{-1} = (b, a)$. Bij DNN speelt $+$ de rol van $\star$ en schrijven we $-a$ voor a^{-1} en $a - b$ voor $a + (-b)$.

(vi) Als je met DNN begint dan vergt de introductie van shape DNNI geen reshaping van de naturals. Bij introductie van $\text{DNNI}_{\equiv}$ treedt zulke reshaping juist wel op, en dat zie ik als een significant nadeel van de shape $\text{DNNI}_{\equiv}$.

(vii) Minpare kan men vergelijken met de somparen uit [6], maar een verschil is toch wel dat met de shape $\text{DNNI}_{\equiv}$ en vergelijkbare constructies in de afgelopen 150 jaar het gebruik van minparen, doorgaans zonder gebruik van een specifieke notatie steeds weer optreedt, terwijl somparen (of plusparen) veel minder productief ingezet zijn. Men kan leren rekenen zonder over klusparen te denkn, bij minparen is dat nog niet zo duidelijk.

7.1 DNNI-reïficatie en blending

In $\text{DNNI}_{\equiv}$ treedt reificatie op: een abstact object, in casu een oneindige equivalentieklasse, wordt gezien als een concreet object. $\text{DNNI}_{\equiv}$ suggereert een opmerkelijke optie: dat een numtoken als 7 gelijktijdig zowel naar een DNN verwijst (en dus naar een DNNI) en naar een $\text{DNNI}_{\equiv}$ verwijst. In dit geval is er sprake van blending, DNNI en $\text{DNNI}_{\equiv}$ raken gecombineerd als ware het zo dat de een tegelijk ook de ander kan zijn. Dit noem ik blending, maar of, en zo

ja in welke mate, blending voor het rekenen relevant is weet ik op dit moment niet.

8 Decimale gehele getallen, de shape DNNI

Voor het onderwijs in EAwD zie ik $\text{DNNI}_{\equiv}$ niet als een gunstige en goed bruikare shape omdat het gebruik van equivalentieklassen en reïficatie een te groot beroep doet op abstracties die een leerling/student in eerste instantie nog niet heeft.

In de shape DNNI zijn de natuurlijke getallen precies als in DNN terwijl de negatieve getallen signs zijn van de vorm $-a$ voor a een positief (d.w.z. niet 0) element van DNN. Alle elementen van DNNI zijn daarmee signs (terwijl de elementen van $\text{DNNI}_{\equiv}$ juist geen signs zijn maar oneindige verzamelingen).

Het kost nu enige moeite om optellen (en daarmee ook aftrekken, want $a-b = a+(-b)$) te beschrijven op DNNI maar dat kan men op allerelei manieren doen, en daarover heb ik weinig nieuws te vertellen. Het punt van DNNI is niet dat de operaties daarin zo gemakkelijk kunnen worden beschreven maar dat de objecten zo duidelijk zijn, veel duielijker en concreter dan in $\text{DNNI}_{\equiv}$.

De shape $\text{DNNI}_{\equiv}$ kent, los van de precieze notatie een lange geschiedenis en een eerste voorloper ervan vindt men in een destijds niet gepubliceerde maar pothuum gevonden tekst uit 1872 van de Duitse wiskundige Richard Dedekind. Ik noem $\text{DNNI}_{\equiv}$ om die reden de DNN gebaseerde Dedekind shape voor gehele getallen (zie ook Bergstra 2026 [3]).

Een vergelijkbare constructie kan leiden tot de VNNN gebaseerde Dedekind shape voor gehelen ($\text{VNNNI}_{\equiv}$), waarin de voor wiskundigen bekende von Neumann ordinalen de rol van naturals (Von Neumann natural numbers, VNNN) worden gebruikt in paats van DNN. Deze laatste shape is tegenwoordig gebruikelijk in academisch onderwijs.

Claim 8.1. *DNNI is een bruikbare shape voor rekenonderwijs op het vlak van EAwD.*

Ik zie DNNI als een "preferred shape" voor gehele getallen (in een educatieve setting) ook al komt $\text{DNNI}_{\equiv}$ dichter bij de shapes die in een academische meestal de voorkeur hebben. De shape DNNI duid ik daarom ook aan als (de shape van) "de decimale gehele getallen". Daarmee is/wordt een (negatief) decimaal geheel getal zoiets as de sign -25 en blijft een positief geheel getal (ofwel positief natuurlijk getal) zoiets als de sign 16.

Net als bij het label natuurlijk getal ga ik er van uit dat " de gehele getallen" niet bestaan, maar dat er verschillende shapes voor het label geheel getal zijn. Met decimaal geheel getal bedoel ik dan uitdrukkelijk de elementen van de shape DNNI. Als we shape $\text{DNNI}_{\equiv}$ gebruiken dan spreek ik liever van $\text{DDNI}_{\equiv}$- integers of $\text{DNNI}_{\equiv}$-gehelen of $\text{DNNI}_{\equiv}$ gehele getallen.

Referenties

[1] J. A. Bergstra. Errors, Failures, and Faults in the Context of Elementary Arithmetic. *Transmathematica*, 2024. DOI.

[2] J. A. Bergstra. Fracpairs and Quantitative Mereology. *Transmathematica*, 2026. DOI.

[3] J. A. Bergstra. Shapes for naturals, integers and rationals. Submitted to Transmathematica, *Transmathematica*, 2026. (forthcoming).

[4] J.A. Bergstra and J.V. Tucker. On Ambiguity: The case of fraction, its meanings and roles. arXiv 2604.04647 URL, (2026).

[5] Jan Bergstra. Zijn er natuurlijke getallen? over decimale natuurlijke getallen. Minstroom Research website, URL, (2026).

[6] Jan Bergstra. Wat is een som? over somparen en somtupels. Minstroom Research website. URL, (2026).